Nelson Mandela
Metropolitan
University

for tomorrow

South Africa

Faculty of Engineering, the Built Environment and IT
Department of Mechatronics

SIMULATION AND VISUALIZATION PLATFORM INTEGRATED UNDER HARDWARE CONTROL SYSTEMS FOR A RECONFIGURABLE PROCESS CONTROL

Compiled by : Ngonidzashe Munyaradzi Zata [212324241]

Supervisor : Professor Theo Ian van Niekerk

Co-Supervisor : Mr John Manuel Fernandes

A dissertation presented as the full requirement
for the degree of Masters in
Mechatronics Engineering

JULY 2017

Declaration

I, NGONIDZASHE MUNYARADZI ZATA [212 324 241], declare that this material submitted for assessment, is entirely my own work; and it has not been taken from the work of others. The work presented in this document is based solely on my own study and/or research.

I fully understand that plagiarism, collusion, and copying are grave and serious offences in the university; and I accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. I have identified and included all sources of all facts, ideas, opinions and view points of others in the bibliographical list.

I have read and understood the NMMU's library referencing guidelines found at:
http://ebeitlibrarian.nmmu.ac.za /Reference-Techniques-(citing-sources)
I have also read and understood the EBEIT Faculty's plagiarism policy found at:
http://ebeit.nmmu.ac.za/ebeit/media/Store/documents/ResearchGuidelines/
-PlagiarismandTerminology/plagiarism2.pdf

Date of submission: 26 July 2017

Signature:

Life is a priceless gift from God.

The day we are born, she endures pain and joy at the same time.

This dissertation is dedicated to my kind and unconditionally loving mother, who has dedicated her life to guiding me to become a hard-working and responsible person.

Abstract

Engineering students often face a challenge in relating advanced course work taught in universities to the systems they face in industry. Such a challenge is due to a number of factors, including the fact that the number of universities that offer engineering qualifications are quite few; and the resources of industry to demonstrate real engineering principles are quite expensive for tertiary institutions.

This dissertation discusses the development of a platform that strives to assist engineering institutions in engaging students with real industry equipment in the most cost-effective way, by allowing the re-use and sharing of resources. In this research, Programmable Logic Controller (PLC) controlled simulated plant models are developed. The user can operate and monitor the process-control system (PLC and plant model) via Supervisory Control and Data Acquisition (SCADA). The use of a PLC and SCADA is acceptable, in order to give users an understanding of industrial-standard tools.

Simulating a plant model reduces costs; and it also gives room for the development of systems in various engineering disciplines, such as electrical, marine, mechanical and mechatronics compared to building real systems. The platform can be remotely accessed on a managed website. The SCADA developed for the platform is accessible upon a successful login on the website; and during the assigned time session by the website administrator. Remote accessing the SCADA is done in an attempt to teach and demonstrate Industry 4.0. It is the operation of industrial systems through the use of the internet; and it is currently considered as the next phase in industrial enhancement.

The research was implemented in phases, which involved:

- Programming Simulink models (dual tank and submarine) based on real and theoretical concepts;
- Inter-linking PLC to Simulink models to communicate in real-time;
- Setting the Input and Output (I/O) parameters between PLC and Simulink models;
- Programming PLC and SCADA to have control and feedback from the plant models;
- Designing a user-defined and managed website that is capable of registration, user-unique login, user access to SCADA in real-time, as well as the monitoring of user activity by the administrator.
- Interfacing SCADA to the website; and finally, the overall outcome of the entire platform when fully integrated. This was looked at on the basis of the outlined objectives.

Acknowledgments

To implement the research and complete this document my sincere gratitude goes to the following individuals and institutions:

- The merSETA in collaboration with AMTC(NMMU) for providing all the funding and resources required to implement this research.

- My supervisor – Professor Theo Ian van Niekerk, who tirelessly gave me feedback and guidance throughout the research. From this research, I wrote my first publication, which I presented at the ROBMECH Conference (November 2016).

- My Co-Supervisor – Mr John Manuel Fernandes, a man with a strict work ethic, from whom I have learnt a lot from during the research. Weekly, we had progress-meeting updates, which kept me motivated until the completion of my research project.

- Mr Bruce Smith and Mr Connie Van Zyl of the NMMU ICT services department, who assisted me in setting up the platform website.

- To my family – Dr G.S, Zata, Irene Zata, Catherine Zata, Masiiwa Zata, Emilia Kanyemba, Winfred Kanyemba and Rosemary Kanyemba, who have always been my source of motivation and moral support.

- To my friends – H.A Hapatel, M.T. Haspatel, L. Haspatel, A. Msimang, L. Lebea – during times when I needed to refresh my mind; these gentlemen would take me out for drinks or to the soccer field. As fellow engineers, they also gave me insights on the research.

Table of Contents

List of Figures

List of Tables

Nomenclature

List of Abbreviations

Acronym	Definition
AC	Alternating Current
AMTC	Advanced Mechatronics Technology Centre
CPU	Central-Processing Unit
DC	Direct Current
GUI	Graphical-User Interface
HMI	Human-Machine Interface
I/O	Input and Output
IP	Internet Protocol
LabView	Laboratory Virtual-Instrument Engineering Workbench
Matlab	Matrix Laboratory
Mbps	Megabits per second
NMMU	Nelson Mandela Metropolitan University
ONT	Optical-Network Terminal
OPC	Open Platform Communication
PC	Personal Computer
PID	Proportional-Integral-Derivative
PLC	Programmable-Logic Controller
PROFIBUS	Process Field Bus
RTT	Round-Trip Time
SCADA	Supervisory Control And Data Acquisition
TCP	Transmission-Control Protocol
TIA	Totally Integrated Automation

Chapter 1

Introduction

This chapter discusses the overall platform's capabilities; it outlines the motivation, the aim and objectives, as well as the research approach followed to develop the platform.

1.1 Motivation

In South Africa, there seems to be a gap between the universities and the industries, whereby students face challenges to apply the knowledge acquired during their university studies in the real world. This is a consequence of three critical problems, which comprise of:

Costs – some of the equipment used in universities may not meet industry standards. This is because it is expensive to set up and maintain real industry systems for learning purposes only.

Safety – some of the equipment used in industry cannot be used by non-qualified or untrained individuals; while such regulations are stipulated by the law.

Limited systems – most practical systems used in learning institutions are built for a specific course or discipline. Hence, some of the courses may not offer adequate knowledge through practicals.

The outlined factors have motivated the development of a platform for this research.

1.2 Background

Process control is an engineering discipline that is often practised, in order to ensure that a process is predictable, stable and consistent. With the technological advances, many industrial processes have grown in complexity – resulting in mass production and the improved quality of products. However, in most engineering environments (e.g. vehicle-manufacturing plants, universities, water-vessel building), there are factors to be taken into consideration to set up, monitor and operate process-control systems. Some of these factors include:

- Simulation – using virtual tools to identify the specifications for components to be used in building the systems.
- Selection of user-friendly hardware and software that can be integrated to give the requisite functionality of the system.
- Communication methods of the system devices (controller, software, actuators, sensors, etc). This is an important part of the system; since it needs to operate in real-time.
- Method of control and monitoring the system by the user – In the modern era, process-control systems are commonly operated and monitored via SCADA and/or HMI systems.

1.3 Aim and Objectives of the Research

The aim of the research is to develop one or more reconfigurable simulated-process control plant models, integrated with an industrial controller (PLC), which can be remotely accessed via a user-defined website, as shown in Figure 1.1. The platform is to be deployed for engineering practical courses.

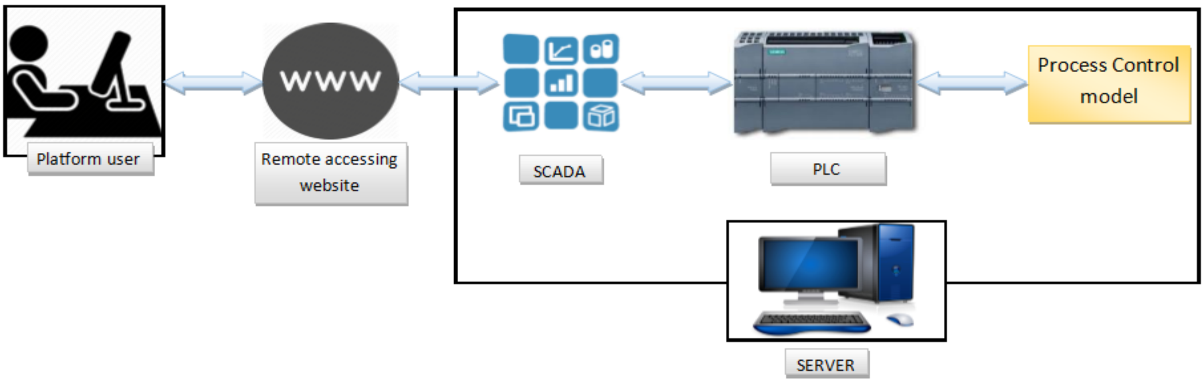


Figure 1.1: Architecture of overall system

The platform to be designed must meet the following objectives:

(i) The platform must have reconfigurable simulated-process parameters based on theoretical and real-plant models.

(ii) A process-control system with multi-(input/output) parameters that can display the linear and/or non-linear characteristics.

(iii) It makes use of industrial standard hardware (PLC) and software (SCADA).

(iv) A website, which enables users (students and lecturers) to remotely access the process control SCADA at allocated time sessions.

(v) Accommodate and enable students from different engineering departments to perform experiments on different models (i.e. Mechatronics, Electrical, Mechanical and Marine Engineering).

1.4 Delimitations of the Research

This section discusses the aspects of how the platform is developed, as well as its functionalities. The user may set parameters for the assigned task; but s/he cannot alter the platform programs. For instance: on Proportional-Integral-Derivative (PID) control of a motor/pump, the user may change the PID parameters, the set-point and other plant parameters. The process control SCADA will be accessed on a website running on the NMMU internet domain. The following have been considered in developing the platform website:

(i) Using the NMMU domain would have the advantage of substantial internet speed. Accessing the platform by other means could result in users experiencing poor responses from the platform.

(ii) Security assurance by using the NMMU domain: only registered users may access the platform.

(iii) Users are to be monitored when they log in to the website. Login times will be allocated for them to work on the platform. The following have been considered when the user logs onto the website:

   - The website administrator is to assign different time sessions for each user; since having a lot of users working simultaneously on the platform could result in poor operation and lagging of the platform.
   - A user may be allocated a different task, depending on what the administrator has allocated to him/her.

1.5 Hypothesis

The implementation of a simulation and visualization platform that makes use of industrial hardware (PLC) and software (SCADA) would give students more exposure to complex integrated process-control systems. This would teach them how to approach any given system, in spite of not having any experience or knowledge of it. The platform would also engage students with a virtual laboratory and remote laboratory, which would give them an insight on Industry 4.0. The platform would need to be re-usable and shared by students, in addition to saving space and costs within the learning environment; since the models are built purely in software.

1.6 The Research Methodology

The platform to be designed will have numerous sub-systems (plant models, PLC, SCADA and website) running on different software and hardware, which are to be selected, such that they can be interfaced together. A dedicated PC (server) will be used specifically for this project; and it can only be accessed by the platform designer and the administrators. The software to be used is in the following categories of:

(i) Simulation software that will be used to set up the process-control plant models.

(ii) Industrial programming software for designing SCADA will be used to give plant visualization and PLC programs.

(iii) Website building software to set up a website that can easily be integrated with the simulation and visualization platform.

1.6.1 Setting up Simulated Plant Models

The research will commence with programming the process-control plant models. These plant models will be purely software designs that are based on real and theoretical process-control concepts. Different software packages may be used to implement the process-control plant models (e.g. Matlab Simulink, LabView).

1.6.2 Integrating Simulated Plant Models with PLC

The simulated plant models and PLC must communicate (send and receive variables) between each other. Achieving this will mean that the simulated plant models can be manipulated by a PLC passing parameters via SCADA. TIA-Totally Integrated Automation software is used to run the PLC and the SCADA for the entire system.

1.6.3 Setting up Remote Accesing Website

The website is designed, such that the SCADA can be remotely accessed; and each user (student) has a set session, when it is accessible. The website is to be used as an online approach for teaching students and for monitoring their progress individually.

The software that may be used for the website design must integrate with the SCADA platforms. This includes Visual Studio, Java script, as well as Python.

1.6.4 Performance Evaluation

The designed reconfigurable process-control platform will be tested under various conditions. The parameters that may be altered on the platform will necessarily include the following: valve control, flow rate, pump speed, set-point, etc. Since this platform is developed for educational purposes; it is important to find out the end-users' opinions with regard to working on the platform. The following tests will be conducted to evaluate the students' learning experience and the platform's performance:

- A test of response time (i.e. how the platform responds to the altering of parameters). This will be done, in order to see how to improve the platform's performance.
- Student feedback will be received through a questionnaire and practical tasks. This evaluation will be carried out after the platform has been fully developed; and it will be carried out at a selected department in the Engineering Faculty of NMMU.

1.7 Significance of the Research

(i) The research platform may be used by various NMMU Engineering departments (Electrical, Mechanical, Mechatronics and Marine) for educational and further research purposes.

(ii) The system will assist students to carry out the right engineering approach when implementing a design or experiment, making use of industrial tools following the outlined approaches:

  - Simulation and visualization will be used, in order to acquire the right parameters before applying to a built model.

- The implementation of programs and parameters on a real model, in order to justify the simulation and visualization.

(iii) The platform will be remotely accessed by students in the comfort of the computer laboratories. This ensures that the students are safe at all times; since they will not be exposed to machinery that could cause harm to them; and consequently, there is no need for supervision.

(iv) There will be less financial stress; since building models in software eliminates the problems faced when physically built models are damaged. In these instances, costs are incurred – to repair or replace them.

1.8 Conclusion and Structure of the dissertation

This chapter has discussed the overall research implemented with all the critical aspects outlined. The structure of the dissertation is shown in Figure 1.2.

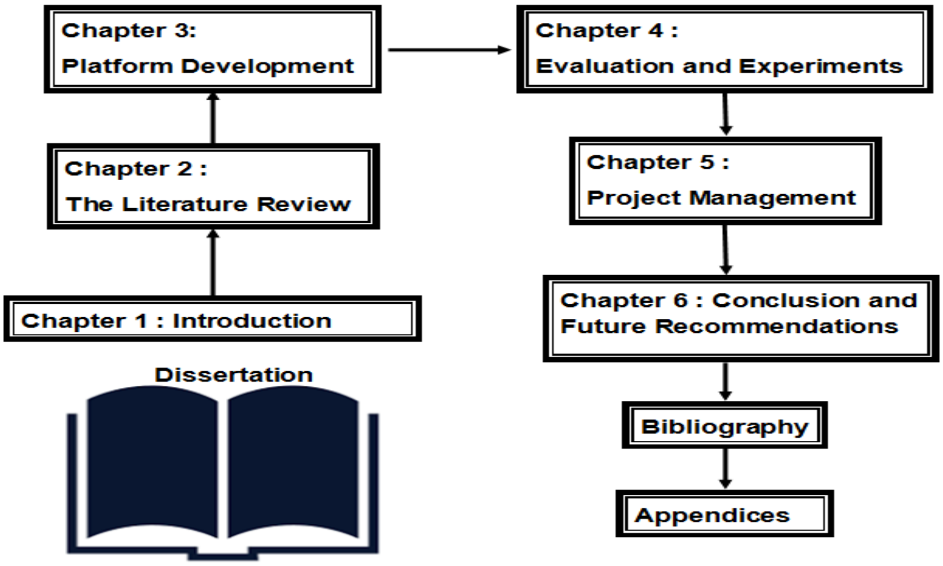


Figure 1.2: Structure of dissertation

This dissertation consists of eight chapters. **Chapter 1** is a brief introduction of the overall research. **Chapter 2** discusses the current work researched, in order to gather and identify the state-of-the art technology used to develop the system.

Upon reaching a point, in which the tools to set up the system are identified and selected; there then follows the actual development of the platform in **Chapter 3**. Looking back at the outlined objectives, the platform functionalities are then evaluated in **Chapter 4**. **Chapter 5** discusses the time spent on the entire research, the costs and some aspects that were encountered during the development of the platform. Following the system's development and evaluation, there is a conclusion and some future recommendations in **Chapter 6**. The development of the platform was done by looking at the feasible means and the tools used by different individuals and institutions. These have all been included in the **Bibliography** list. Last, but not least, there is **Appendices**, which consist of all the programs designed for the platform (PLC, plant models, SCADA and website), the specifications of the system's set-up and the platform outcomes.

Chapter 2

The Literature Review

This chapter discusses the definition and importance of process-control systems and their various applications in the engineering field. The designs and set-ups of these systems are looked at, in order to establish the state-of-the-art hardware and the software used on remotely accessed process-control systems. The educational platform set-ups used for teaching engineering students are also described; and how the entire teaching approach is carried out, such that students can work on process-control systems.

2.1 Process-Control Systems

As the world population is growing exponentially, the demand for the quantity of products and services for consumers has become high. Hence, the development of automated systems that are capable of mass productions; this has made it possible to meet these demands. Vagia M. et al. state that "automation refers to a system that will do exactly what it is programmed to do by the programmer, without much choice, or [the] possibility to act in any different way. Its actions are predefined (logical instructions, such as: if, while, and, or statements); and it has no ability to change them; unless the programmer makes [some] changes [1, 2]."

A process control (Figure 2.1, [1]) involves measures in which mass, energy and information are transferred, converted or stored. A typical process-control system consists of sensors and actuators that are assigned to the process. Sensors are used for monitoring and diagnosis of the process. Actuators serve the purpose of intervening in the process, in order to achieve the desired task [3, 4]. The importance of automation in the process industry has increased dramatically in recent years.

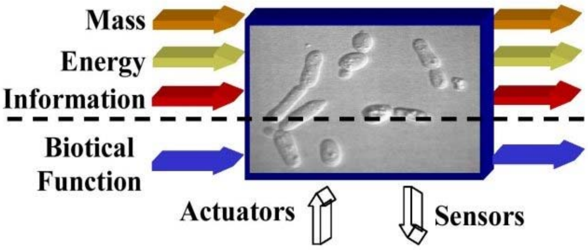


Figure 2.1: Automated Process Control structure

Process automation accommodates the following:

- Improving product quality;
- Mastering the whole range of products;
- Improving process safety and plant availability;
- Efficiently utilizing the resources and reducing the emission of toxic materials.

2.1.1 Automated Process-Control Systems in Agriculture

The agricultural sector is currently undergoing fundamental technological modifications and changes. This is due to a number of factors that include demands for increased efficiency in the production processes; while at the same time, there is a need to meet the requirements from consumers for lower costs, increased quality, as well as reliable data on the origin and handling of the food items: from "Farm-to-Fork" [5]. The automated control of farming machinery, environmental sensors to collect weather parameters, integration with sensors and farming systems, in order to collect the relevant data for traceability – these are some of the technologies currently being developed and used. Most food-production processes are non-linear, time invariant and unstable due to factors such as: the weather, disease, pests and other unpredictable factors. The automation of food production has to handle these properties properly; and also be able to employ them actively [5, 6].

2.1.1.1 Automated-Irrigation System

For all living organisms, water is considered to be a natural necessity that is vital for their existence and well-being. The regulation of the quantity of water in cultivated fields is a process, which affects the sustainability and the productivity of crops. If there is insufficient or excessive irrigation, there may not only be obstruction, but also the destruction of crops. Without irrigation, crops would never have been grown in some parts of the world that have limited natural water sources, such as deserts [5] .

A system shown in Figure 2.2 [5] was developed by Nikolidakis SA and colleagues. This has proven to be one of the systems that sets the standards for automated irrigation systems. The proposed system considers climate values, such as humidity, temperature and wind speed – in order to calculate the quantity of water that is needed for proper irrigation. The proposed automated irrigation-management system comprises two sub-systems [5]. The first sub-system concerns the WSN (Wireless Sensor Network), which collects the data from the cultivated fields. The proposed WSN structure includes a set of sensors that monitor the humidity and the temperature of the soil, the speed of air, and the duration of sunshine hours per day. The individual sensors are placed at appropriate locations in the crop field. The collected information is then transmitted to a base station [5, 7].

Each of the sensors is self-powered by a battery. The main application is based on the collected inputs it receives from the sensors. It then acts on the basis of the configuration that is applied to the system. For the place that needs to be monitored, the type of crops that are planted in the field (corn, potatoes, etc), the chemical soil components, such as the sodium content and the season of the year (i.e. winter or summer). These factors must also be taken into consideration [5].
The platform offers advantages such as :

- Improved estimation and planning of field irrigation, based on the available water supply.

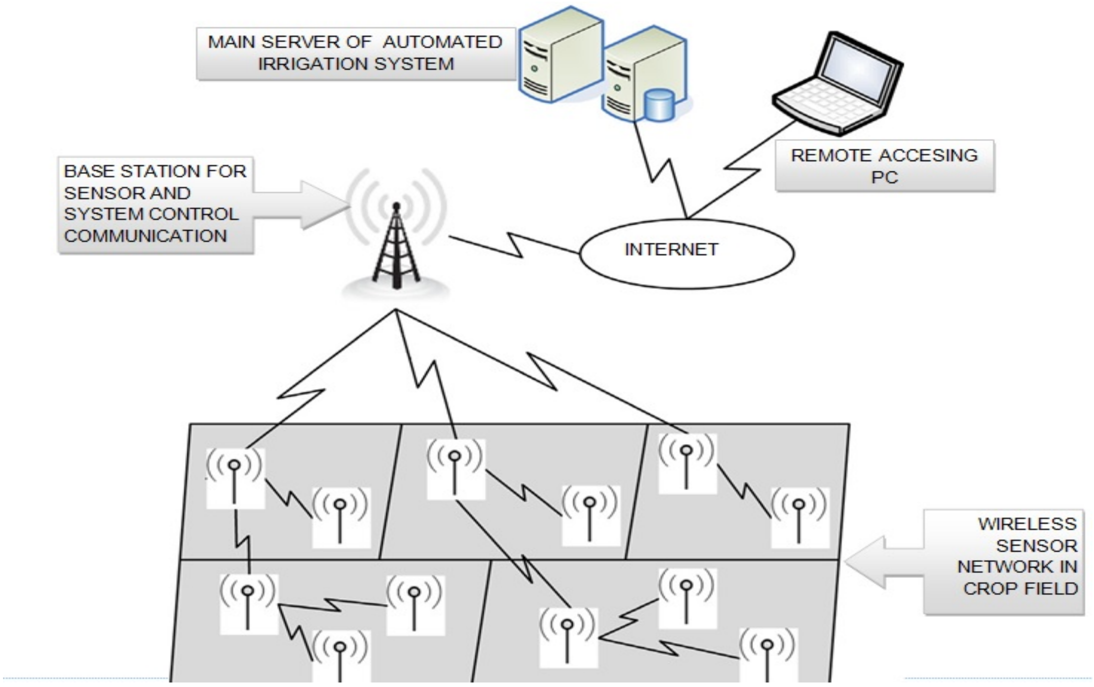

MAIN SERVER OF AUTOMATED IRRIGATION SYSTEM
REMOTE ACCESING PC
BASE STATION FOR SENSOR AND SYSTEM CONTROL COMMUNICATION
INTERNET
WIRELESS SENSOR NETWORK IN CROP FIELD

Figure 2.2: The architecture of the Automated Irrigation System with remote access

- Minimization of the required human resources, time and effort in the agricultural production.
- Early detection of possible floods in the field that could be destructive for the crops, as well as proper pumping of the water to mitigate such cases.

The second sub-system involves the decision-making system, as shown in Figure 2.2, which makes use of a computer as the main controller with dedicated software. The proposed decision-making system is based on the acquired data from the sensors and the historical data values, which are used to calculate the water quantity necessary for irrigation [5].

2.1.2 Automated Process-Control Systems in the Production of Pharmaceutics

As the world is going through fast and broad advances in technology; it also faces pandemics in diseases. Hence, with large populations, a lot of medical drugs are always needed to help those victims suffering from illnesses. Thus, through automation in the processes of drug production, it has been made possible to save thousands of lives in good time [8, 9].

Automated process-control systems are also desired by the pharmaceutical industry; as these enable them to satisfy regulatory constraints and flexible market demands. Traditional pharmaceutical industries operate on the basis of inflexible batch operation processes that are often characterized by little or no use of advanced on-line measurements and limited process control. Such “traditional” pharmaceuticals’ production is also associated with a long quality control [8, 9, 10]. A framework has been proposed, which could be used to design an automated pharmaceutical plant. The framework focuses on:

- Providing stepwise guidelines for the integration of advanced control strategies, control hardware/software and sensing technologies, such as PAT (Process-Analytical Technology) tools.
- Supporting the design and implementation of the control strategies.

PAT is an element of the framework that gives a better understanding of the manufacturing process, and how to use that knowledge online, in order to attain improved control of the process and consistency in product quality [11, 12]. The block diagram in Figure 2.3 illustrates the framework of how a typical automated pharmaceutical system may be set up.

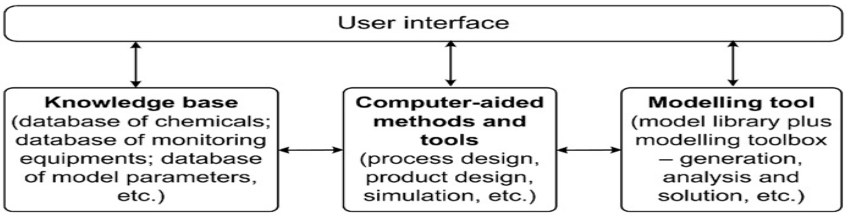

User interface
Knowledge base
(database of chemicals; database of monitoring equipments; database of model parameters, etc.)
Computer-aided methods and tools
(process design, product design, simulation, etc.)
Modelling tool
(model library plus modelling toolbox – generation, analysis and solution, etc.)

Figure 2.3: Guide for setting up a Pharmaceutical Automated System

2.1.2.1 A Continuous Tablet Manufacturing System

The designed system shown in Figure 2.4, [8] was designed at the Department of Chemical and Biochemical Engineering, Rutgers, the State University of New Jersey. A closed-loop operation of the tablet manufacturing plant using a NIR (Near Infrared) sensor has been demonstrated in this system. This was a significant advancement in pharmaceutical manufacturing. A direct-compaction continuous tablet-manufacturing process was built. It has three gravimetric feeders that provide the necessary lubricant, API and excipient. Each feeder has a hopper that can hold a certain amount of material and a rotating screw to change the flow rate. The feeds are then passed on to a blender, in order to generate a homogeneous mixer. After the blender, the blended powder is transferred directly to the tablet press through a feed frame [8].

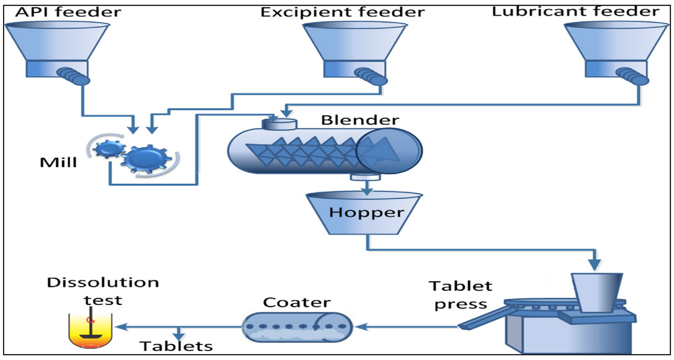


Figure 2.4: Automated Tablet Manufacturing System

The onsite design of the control system, as well as the control loops are set up on a real control platform (Siemens PLC); and the input and output of each control loop is connected with the process model simulated. However, this connecting methodology is generic; and it can be applied to any simulation tool and control platform.

2.2 Process Control Technology

In the preceding Section 2.1, all the mentioned process-control systems make use of some form of controllers. These run on their respective software. There is a wide range of controllers for specific systems and applications. The selection of a controller for an automated system depends on numerous factors. These include the following [13, 14, 15] :

(i) Peripherals of the system being controlled: These include the sensors, actuators, power supply, etc. An I/O card controller that has the capability to acquire the data and manipulate the data for the system must be used.

(ii) Physical size and operating environment of the system: If a system is to be portable and operate at different places, a PLC is advisable; whereas for a system that is to be installed at one fixed vicinity, an industrial PC maybe used.

(iii) Ease of interfacing controller with other platforms and software: A flexible controller usually saves costs to any workplace; as one does not need to buy a new controller for a new project, or for expanding the existing systems.

(iv) Maintenance and operation: A controller that is commonly used and recommended is usually the best option. Users can easily access information from other users, who have used the same controller, without the need to go back to the manufacturer.

(v) Speed of tasks to be carried out: Depending on how fast the control system needs to be, a controller that has a good response time is required. When it comes to speed, the Industrial PC and PLC can meet most fast-processing tasks.

Controllers mostly used in industry are divided into two main categories. These are PLCs and Industrial PCs. In this section, the PLC and the Industrial PC will be discussed, looking mainly at their structures and some applications in simulated process-control systems.

2.2.1 Hardware PLCs

A programmable logic controller (PLC) is an industrial-control device that monitors the state of the input devices; and it then executes signals to the output devices, based on a custom program to control the state-of-output devices. Thus, for a process-control system, a PLC is often used and engineers working on such systems need the knowledge for the operation of PLCs [14, 16]. Figure 2.5 shows the basic structure of a PLC. The different sub-systems that make up the PLC are described as follows:

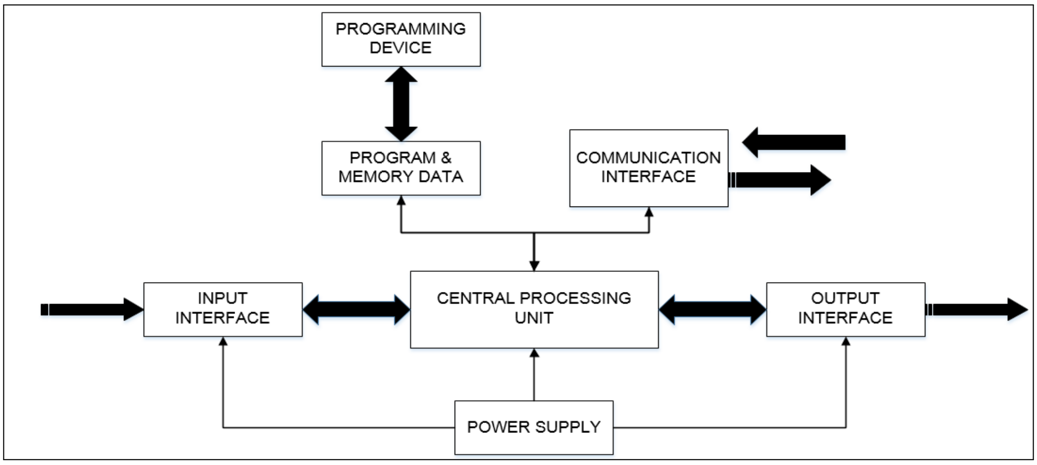


Figure 2.5: PLC basic structure

(i) The central-processing unit (CPU): This contains the micro-processor. It interprets the input signals; and it carries out the control and the calculation of actions, according to the coded program; and it passes signals to the outputs.

(ii) Power supply: This converts the mains AC voltage to the low DC voltage needed for the processor, and the circuits in the input, output interface and the communication modules.

(iii) Programming device: This is used to code the required program into the memory of the CPU. After the coding is complete, the program is then transferred to the memory unit.

(iv) Program and Memory Data: This stores the program of the PLC and the input data from a process; and it also buffers the data for output.

(v) Input /Output sections: The CPU processor receives information from the external devices; and it communicates this information to the external devices. Input and output devices can be classified by signal type, such as digital or analog. The inputs might be from switches or sensors, such as temperature sensors, or flow sensors. The outputs might be connected to solenoid valves, or other actuators.

(vi) Communication interface: This is used to receive and transmit the data on the communication networks. It carries out the following tasks: device verification, data acquisition, synchronization, as well as the connection management.

2.2.2 Soft PLCs

A soft PLC is basically a software package, which emulates the functionality of a standard PLC upon a PC. Instead of having a PLC, there are a number of I/O-modules connected to a PC, which run the soft PLC package. Online connections can be established via the Ethernet, via PROFIBUS DP, and directly on the same PC [16, 17]. Figure 2.6 illustrates the basic structure of a Soft PLC.

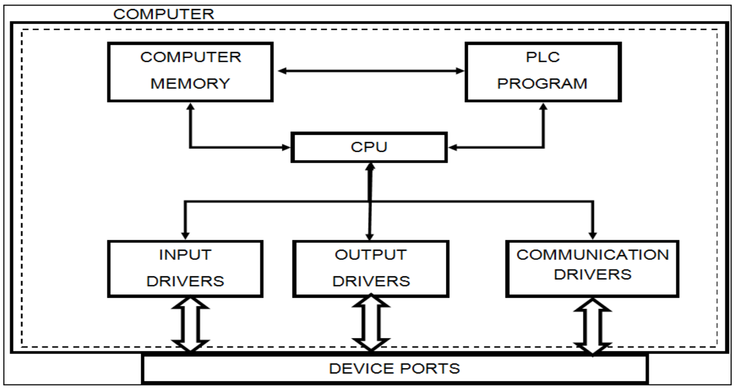


Figure 2.6: Soft PLC basic structure

The different sub-systems that make up a Soft PLC described are similar to those of a physical PLC:

(i) A central-processing unit (CPU): It contains the micro-processor, which is in the form of a program that runs on a computer's memory. It interprets the input signals and carries out the control and calculation of actions, according to the coded program.

(ii) PLC program: Coded on the computer; and it comes integrated with the software PLC. The program is stored on the computer's memory.

(iii) Computer Memory: This stores the program of the PLC, CPU of Soft PLC and input (data and drivers) from a process; and it also buffers the data for output.

(iv) Input and Output drivers: The drivers interpret the input and output signals to the CPU and the system being operated by the PLC.

(v) Communication drivers: These are used to receive and transmit the data on the communication networks, via communication devices to the CPU. They carry out the following tasks: device verification; data acquisition; synchronisation; and the connection management.

(vi) Device Ports: These are the physical connectors for the Soft PLC drivers to the hardware of the system being operated. Data bus – this transfers the data being manipulated by the CPU, the CPU executable instructions; and furthermore, it returns the status of the inputs and outputs.

2.2.3 Industrial Computers

Often referred to as Industrial PCs, Industrial Computers are commonly used in industry; and they have more unique features than the ordinary computer used in an Internet café or library. They can run on custom programs, depending on the software used to code the program. An Industrial Computer may be considered as the most powerful controller available for industry. An Industrial Computer renders any process-control system much more efficient in terms of control, monitoring and set-up [18]. They are usually manufactured relative to what the customer requires in terms of inputs, outputs, communication modules and power for the peripherals on the system to be operated [5]. Figure 2.7 shows the structure of the Industrial Computer which is also similar to the basic computer. The sub-systems that make up an Industrial Computer are described below:

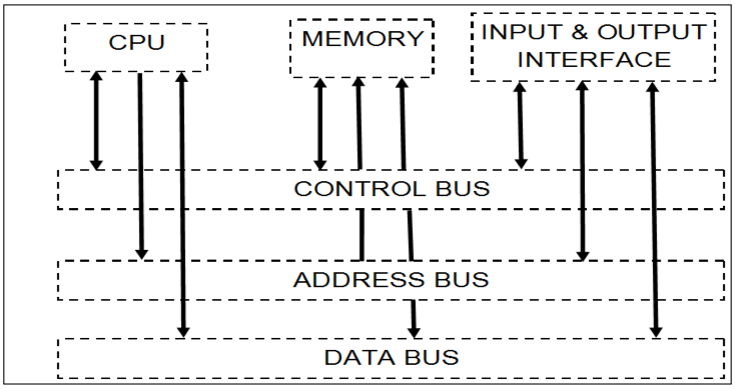


Figure 2.7: Industrial Computer basic structure

(i) CPU, Memory, Input/Output Interface: The CPU interprets the input signals; and it carries out the control and calculation of actions, according to the coded program; and it passes these signals to the outputs. The memory is where the program is stored, and from which it can be retrieved.

(ii) Control bus: It is used by the CPU to link with other peripherals within the entire computer.

(iii) Address bus: This transfers the information, to which device, the CPU is linked during the execution of the program.

(iv) Data bus: This transfers the data being manipulated by the CPU, as well as the CPU executable instructions; and it returns the status of the inputs and the outputs.

2.2.4 PLC and Industrial Computer Comparison

Table 2.1 describes a comparison for PLCs and Industrial Computers. The critical aspects of the industrial controllers are summarized in the table. Depending on the designed control system, these factors are considered.

Table 2.1: PLC vs Industrial Computer

Parameter	PLC	Industrial Computer
Applications	A PLC can be used in a variety number of projects such as student projects, domestic systems and large scale industries. A PLC can be installed on mobile systems such as AGVs.	Industrial PC is limited to industrial systems that are of a large scale. It may be used for small scale systems; but is not as commonly used as the PLC. The Industrial PC is usually used on non-mobile systems and is fixed to a station.
Cost	A PLC is relatively affordable, it may be easily purchased off the shelf.	An Industrial PC is relatively expensive compared to a PLC.
Durability	A PLC has no moving parts; so it can withstand harsh environments. A standard PC contains moving parts; such as fans or hard disk drives and is less suitable for environments with high vibration levels.	Industrial PCs have peripherals such as solid-state drives, fan-less cooling systems, and in-cabinet mounting. These features make them just as durable as PLCs, to withstand the toughest industrial or environmental conditions.
Programming	PLCs can execute code that is event-driven or scan-driven. At some instances scan-driven software takes much more time executing than an event-driven equivalent.	Industrial PCs are like generic PCs in that much of the software written for them is event-driven. Their memory can be expanded as well, hence they execute programs faster than PLCs.

2.3 Simulation of Process-Control Systems

This section discusses two integrated platforms that have an industrial controller interfaced to a simulated plant model.

2.3.1 Three-Tank System operated in Real-time with Matlab

A three-tank system operated by a standard industry PLC and Matlab Simulink in real-time was developed as a practical laboratory tool (Figure 2.8). In some instances, laboratory experiments of such a nature (having multiple inputs and outputs) can be operated from Matlab Simulink by some real-time software packages – combined with an analog and/or digital I/O card [19].

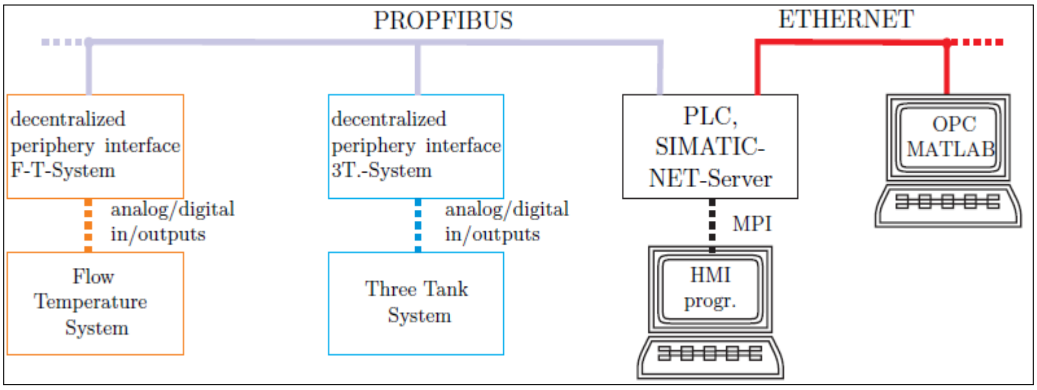

PROPFIBUS
ETHERNET
decentralized periphery interface F-T-System
decentralized periphery interface 3T.-System
PLC, SIMATIC-NET-Server
OPC MATLAB
analog/digital in/outputs
analog/digital in/outputs
MPI
Flow Temperature System
Three Tank System
HMI progr.

Figure 2.8: A simulated three tank system controlled by a PLC

The central SIMATIC PLC is acting as a bus master; and the tank system is connected by PROFIBUS via a decentralized-periphery interface as a bus slave. The PLC is equipped with a communication processor for Ethernet communication via TCP/IP. This enables standard automation tasks, like remote control or error reports by email [19, 20]. The deployment of an OPC server WINDOWS application can be addressed through the Ethernet. This feature has been utilized to connect the Matlab Simulink to the Ethernet; and thereby, also to communicate with the PLC and its business clients. The experimental set-up may be structured into the three-tank system, the SIMATIC equipment, and the OPC-server with Matlab connection, as shown in (Figure 2.8 [19]).

2.3.2 Automated Generation of Simulation Models for Controlled Code Tests

The system has been developed to ensure the conformity of the implemented control functions, with the customer's specifications and test activities. For the past several years, control coded tests have increasingly been carried out on simulation models to increase test coverage and timeliness. For this platform, there are two simulated systems. These are:

- Hardware-in-the-Loop simulation – This uses a real controller in combination with the simulation model (Figure 2.9)

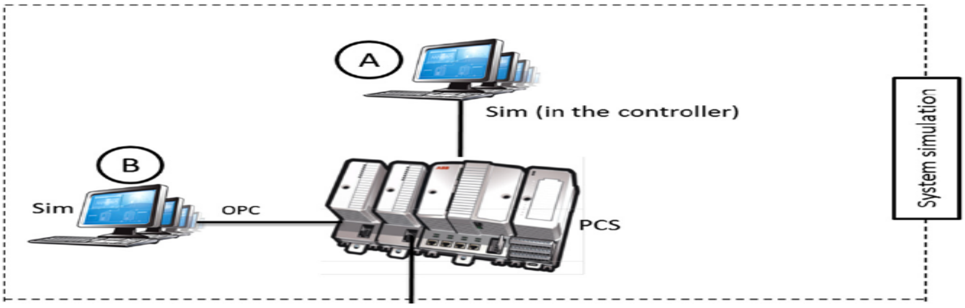


Figure 2.9: Hardware-in-the-Loop simulation

- System simulation – This uses a soft controller (Figure 2.10)

The soft controller runs on the same computer as the simulation model. Therefore, it is intended to operate a communication set-up that is based on the virtual exchange of variables; instead of communicating via a hard-wired digital system or by analog signals. In this research, a new Modelica OPC library object was developed. It sends the tank level to the controller and the receiver that can be used for manual modelling [21].

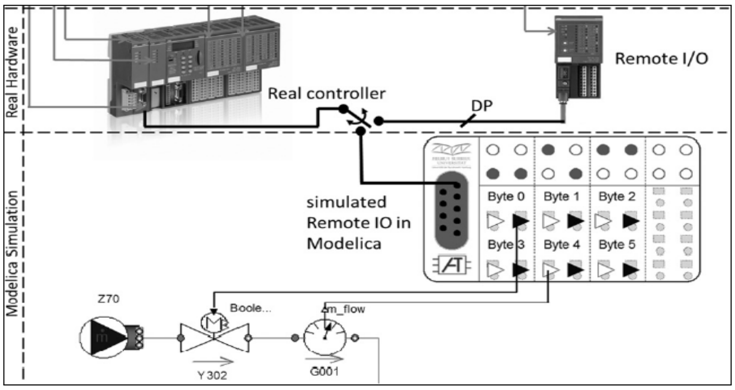


Figure 2.10: System simulation

2.4 The significance of Simulation in the Engineering Field

In modern industry, automated process-control systems are now being set up and monitored through modelling and simulation. This is done; as it has a definite cost-saving benefit; and it also ensures that:

- project specifications are fully met within the desired objectives project budget;
- and costly mistakes are thereby avoided.

Simulation is the emulation of a real or theoretical system that is done by using computer software. Simulation also gives a justification of how the theoretical models work [22, 23]. Process simulation and visualization systems have to provide solutions for two different problems. These are:

- To support the definition of the model; and its interfacing with the process.
- To provide the features of a user-interface development and help to interconnect the generated user interface with the model designed.

Hence, the knowledge on simulation platforms is important for engineers and plant designers, in order for them to be able to set up, monitor and control the systems used in industry [24, 25, 26].

2.4.1 Simulation Software used in Industry

The selection of software for simulation depends on how complex the simulated process is. Industry makes use of simulation platforms for different tasks. These include the prediction of future operations, the required materials, safety reasons, etc. Two examples of simulation applications in the process-control industry, namely: Aspen Plus and ChemCAD, are discussed in sections 2.4.1.1 and 2.4.1.2.

2.4.1.1 The Kinetics-based Aspen Plus Model

ASPEN Plus is a software product of Aspen Technology that is a well-known provider of software and services for the process industries. Process-modelling analysis and design tools are accessible through process simulators, like Aspen HYSYS and Aspen Plus. Some of the features for these process designs include [27, 28, 29]:

- Energy use;
- Capital and operating costs;
- Product yield through the use of activated energy;
- Equipment design during the modelling process.

This study was designed to investigate the entrained-flow gasification of wood waste with the ASPEN Plus model and experimental diagnostics; while using the Kentucky coal as a baseline. Figure 2.11 depicts the developed ASPEN Plus kinetic-based model that simulates the operation of the entrained-flow gasifier [27]. The following assumptions were made in the development of the entrained flow-gasification model:

- The pressure drop in the gasifier is neglected;
- The particles are assumed to be spherical and of uniform size;
- The temperature inside the particle is assumed to be uniform.

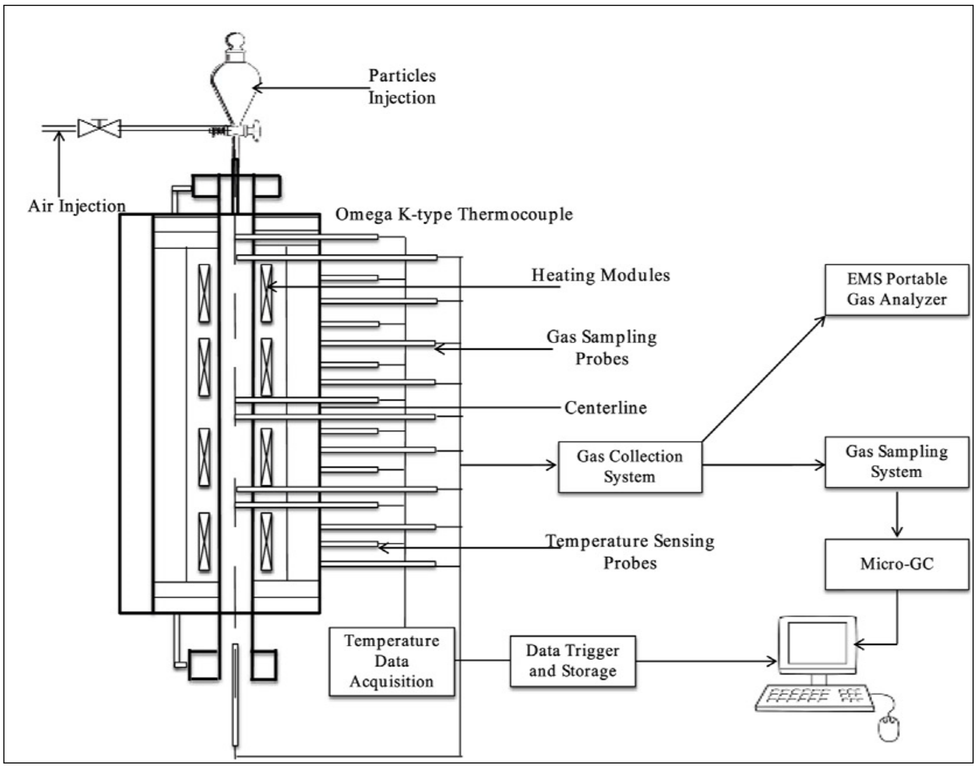


Figure 2.11: The Retrofitted Kinetics-Based ASPEN PLUS MODEL

2.4.1.2 A Systematic Process Retrofit of Complex Chemical Processes

For this designed simulator, the focus was on the improvement of operating conditions, the analysis of bottlenecks, alterations in production methods and the influence of additional variables. This simulator makes use of two software platforms that are integrated together. The software platforms used are: Matlab Simulink and ChemCAD. The ChemCAD is an integrated suite of intuitive process-engineering software. It has a wide variety of capabilities that meet most of the chemical-process simulation needs, from small-scale projects to large plant projects. One can model these by using ChemCAD [28, 30]. In developing this simulator, a real model was first analyzed; and then a methodology had to be followed. The methodology was for a plant-wide retrofit of complex processes divided into five steps, as shown in Figure 2.12.

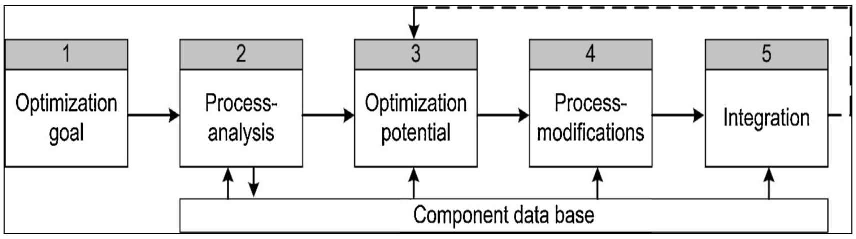


Figure 2.12: Process-Modification Guide for ChemCAD model

The retrofit tool-box can deal with complex processes; and it uses a well-promising optimization algorithm in co-operation with a flexible and fast data transfer. The simulator and the programming software Matlab have been interfaced together, using the OPC standard as the communication platform; as shown in Figure 2.13 [28].

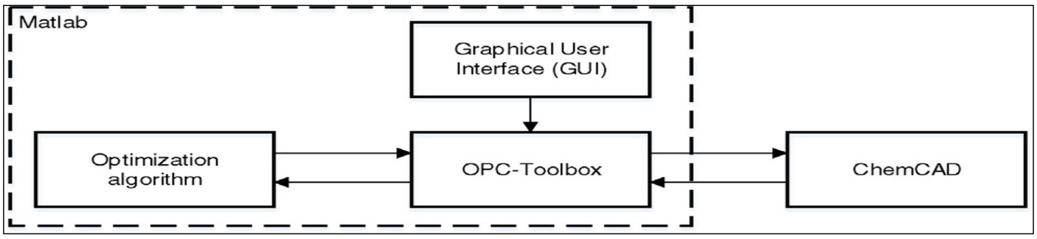


Figure 2.13: Control System for ChemCAD model

2.4.2 Simulation Software used in Academia

A university is the place where an engineer's critical knowledge is learnt and acquired. Hence, it is important for any engineering graduate to have had some form of exposure to industry-standard systems. Thus, simulations and visualizations are done in universities to teach and give the students an understanding of the basics that are required on systems that they are likely to face in industry. For most universities, not all equipment (software and hardware) is available to accommodate all the students, or to render a real-industry practice; hence, there are cheaper and more feasible platforms that may be used to teach the students. In this section, the commonly used software (Matlab Simulink, LabView) for simulation in universities will be discussed [31, 32]. These software platforms are not only used for university purposes, but also for some industrial applications.

2.4.2.1 Matlab Simulink

Matlab, also known as Matrix Laboratory, is a multi-variable and complex software that is optimized for solving engineering and scientific problems. Simulink is a graphical programming environment for modelling, simulating and analyzing the multi-domain dynamic systems that work on the Matlab platform. It enables the rapid construction of virtual prototypes, in order to explore design concepts at any level of detail, with minimal effort. It includes a comprehensive library of predefined blocks to be used to construct graphical models of systems using drag-and-drop mouse operations. It supports linear and non-linear systems, modelled in continuous-time, sampled time, or a hybrid of the two [31].

Since students learn efficiently with frequent feedback, the interactive nature of Simulink encourages one to try things out, change parameters; and immediately, to see what happens, for "what if" exploration [31]. The other key features that are offered by the software are:

- Simulation engine with fixed-step and variable-step solvers;
- Scopes and data displays for viewing the simulation results;
- Project and data-management tools for managing model files and data;
- Model-analysis tools for refining model architecture and increasing simulation speed.

Applications of Matlab Simulink

(i) Matlab Simulink can also be used to program PLCs; as it has a code generator, known as SCL (Structured Control Language) code; and it can be used to run Siemens S7-Controller family integrated into the TIA Portal-engineering framework. This feature of Matlab makes it possible for users to operate industrial standard hardware [33].

(ii) Advanced Mathematics and Mechanics Applications – it can carry out large calculations with a lot of variables, provided the parameters and the algorithms have been programmed correctly. For instance, it would require a lot of work and time for students to be able to calculate multi-dimensional vectors (adding, subtracting, multiplying, etc); but with Matlab Simulink one can develop a program to carry out all these required calculations [31].

(iii) Artificial Neural Network programming – Neural Networks are used for a number of important applications in both industry and universities; as they are used to predict non-linear inputs and outputs (image-processing systems, the stock markets, the weather forecast, etc). In Matlab Simulink, the neural network toolbox provides algorithms, as well as functions and applications to create, train, visualize and simulate the neural networks. It can perform classification, regression, clustering, dimensionality reduction, time-series forecasting and dynamic system modelling and control [31].

2.4.2.2 LabView

The Laboratory-Virtual Instrument-Engineering Workbench (LabView) is a National Instruments software designed for a variety of projects for engineers and scientists. It has a graphical programming syntax that makes it possible to visualize the set-up and the code-engineering systems. From building automated machines to ensuring the quality of connected devices, LabView is an ideal platform that may be used to create, deploy and test the Internet of Things. In LabView, one can acquire accurate timely measurements for industrial monitoring and control systems. Using LabView, it is possible to quickly connect to industrial sensors and specialty I/O to acquire high-speed, high-resolution data in the most demanding industrial environments [32]. LabView can be interfaced with programmable logic controllers (PLCs). However, this means that one may implement modular expansion I/O for time-critical applications over deterministic Ethernet, or otherwise to connect to an industrial fieldbus, such as Ethernet/IP, PROFIBUS and OPC [34]. Other features available on the LabView platform include [32]:

- Parallel programming – LabView is an inherently concurrent language; so it is very easy to program multiple tasks that are performed in parallel by means of multi-threading.
- Large libraries – Many libraries with a large number of functions for data acquisition, signal generation, mathematics, statistics, signal conditioning, analysis, etc.
- Interfacing to Devices – this includes extensive support for interfacing to devices, instruments, cameras and other devices. Users can interface to hardware, by either writing direct bus commands (USB, GPIB, and Serial) or using high-level, device-specific, drivers that provide native LabView function nodes for controlling the device.

Applications of LabView

(i) LabView is an ideal mechatronics software for developing systems that merge mechanical, electrical, control and embedded software design. With tight integration between LabView and SolidWorks software, one can create digital prototypes and simulate the electromechanical performance of the designed machine within minutes. LabView helps in software investment; because it allows for the use of the same code created for simulating the machine and when changing from prototyping to high-volume deployment [35].

(ii) With the LabView Control Design and Simulation Module, simulation of dynamic systems, designing sophisticated controllers, and deploying control systems to real-time hardware can all be implemented. The use of both classical and state-space approaches to design controllers and estimators is also possible. When integrating this module with the LabView Math Script RT Module; one can perform textual mathematics and algorithm design in LabView [36].

(iii) R Series Application – Sensor simulation is the process of providing realistic sensor signals to the inputs of a device under test (DUT) and then evaluating how a piece of equipment responds across a broad range of operating conditions. This is particularly useful in hardware-in-the-loop (HIL) testing and rapid-control prototyping (RCP). The greatest benefit to simulating sensors is the ability to push past the operational limits of a specific environment; and to test for fault conditions that would otherwise be damaging or dangerous [32].

2.5 The Internet of Things and Remote Access in the Modern World

In the modern world, there are some tasks and activities that are now carried out by using applications and devices from any location. Figure 2.14 shows the set-up of a remotely accessed and operated platform.

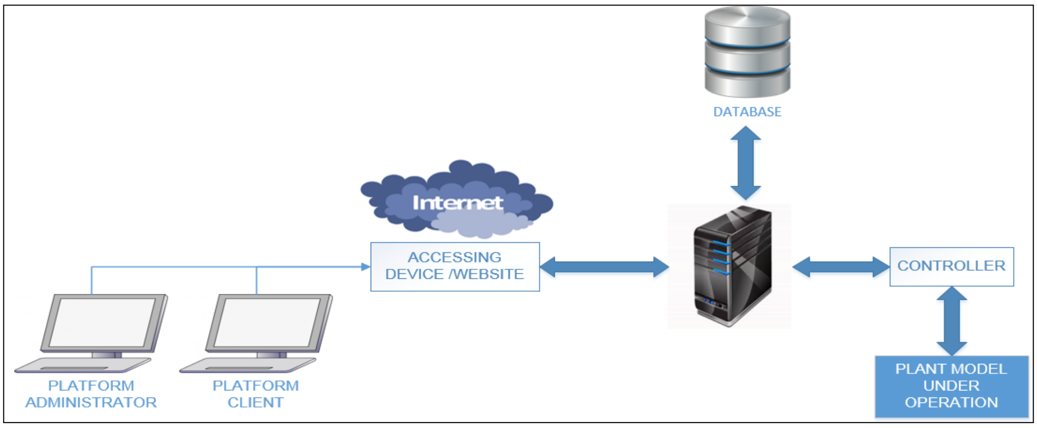


Figure 2.14: A Typical Remote-Accessing Architecture

Some examples that involve remote-accessing activities include the following:

- Mobile and Internet banking (purchasing of goods is now possible without having cash, or a card, or going to the bank).
- The use of the remote in the household (adjusting things, such as audio volume, air-conditioner temperature, etc).
- In industry, engineers can now monitor and operate plants from different places around the globe. They can be notified of any faults and carry out assigned tasks. The operation of such systems allows the re-using of hardware, whereby other company workers in different locations can also make use of the same plant.

However, there are a number of communication methods that may be used to achieve remote access and the control of systems. The commonly known communication means used for remote accessing are accomplished through the internet, Bluetooth, infrared, cable connections, etc. In this section, the main focus will be on the internet-access methods that are currently being used. Upcoming engineers also need the knowledge of the internet of things; hence, remote laboratories are also discussed in this section [37].

2.5.1 Industry 4.0

First introduced by the German government to promote the computerization of manufacturing, the term Industry 4.0 refers to a fourth industrial revolution – following those ushered in by steam power, electrical power and computing (Figure 2.15). Industry 4.0 is a collective term consisting of a number of different technologies used in the 21st century, such as contemporary automation, data exchange and manufacturing technologies [38].

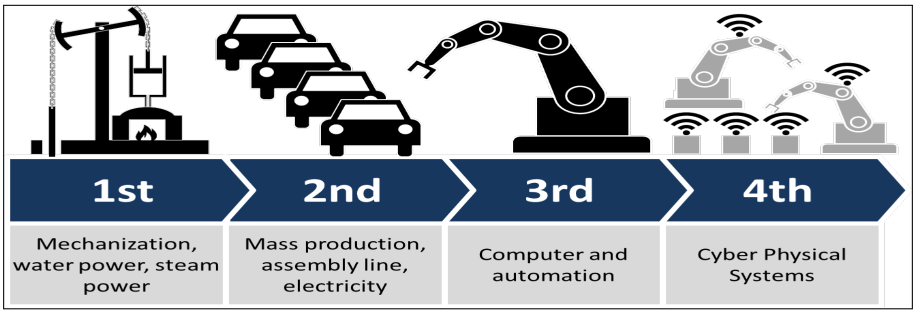

1st
Mechanization, water power, steam power
2nd
Mass production, assembly line, electricity
3rd
Computer and automation
4th
Cyber Physical Systems

Figure 2.15: The Four Phases of Industrial Revolution

It involves the operation of industrial systems, through the use of the internet. This includes setting up systems, monitoring systems and numerous other activities being carried out. Such software-based system and service platforms will play a major role in tomorrow's manufacturing; as they are the only way to bring connectivity, including data analysis to machines and workpieces in production [24, 39, 40]. Looking at how Industry 4.0 will work; it does bring a great change of how the world operates in terms of production, working environments, sales, purchasing and data capturing [40].

Automation in process-control itself has had a bad impact in some instances on plant workers. Workers feel they may lose their jobs; because automated systems bring the following advantages:

- Longer working hours – as long as they are powered up; they do not get tired;
- The handling of goods is made much easier; and they are less vulnerable to damage. More products are manufactured to meet the market demands.

It is impossible to have an industrial plant without the need for any form of human interaction. This is due to the fact that there are tasks that robots or machines cannot do. In certain situations that may involve:

- Installation and connecting wires, pipes, power terminals;
- Fault repairs of peripherals, robots, worn-out tools;
- Fatal emergency situations that require power cut-offs; at times, programs or machines may malfunction; hence, there will always be the need to stop the system from running.

Industry 4.0 involves the operation, manipulation and monitoring of automated systems using the internet. Thus, there is a need for workers who carry out these tasks to be equipped with the necessary knowledge of working via the internet. This is also an advantage to the skills that workers have beforehand (mechanical, electrical, etc). A feasible way to prepare plant workers and engineers for a future industry that is oriented around Industry 4.0 is required. Such a training and teaching method would involve the use of remotely accessed laboratories to various systems (real and virtual) that emulate those used in industry. Universities and other learning institutions do provide courses for students to learn the basic operations of a computer: simulation, process control and automated systems. However, the stated learning courses are offered separately, which does help students to understand and grasp the relevant concepts better. An integration of these courses is, however, needed; and the best way is via the use of remote and virtual laboratories [41, 42] .

2.5.2 Methods of Remote Accessing via the Internet

The interfacing of different software and hardware for any process control system directly affects the performance and functionality of the platform. Most engineers prefer using hardware and software from one source. This usually seems to be an assurance that the system to be designed will work (i.e. hardware and software communication) [43, 44]. However, the use of different software and hardware has some added advantages. These include:

- Reduced costs on a system – In certain instances, purchasing the entire package (hardware and software) from one company may be quite expensive; as either the software or the hardware may be replaced with a lower-costing substitute that would work the same way, or even better.

- Heterogeneous networks – a system may have two different sub-systems running on different software platforms. For instance, they may have a vision-based sub-system and a motor/actuator sub-system. For these two sub-systems, the data should be received and sent to the main controller; thus, the use of special communication protocols is needed.

- Reuse of software – the same software may be used for the development of other projects in the future.

There are ways and means of communication that are commonly used to interface process-control systems (hardware and simulating software) with a chosen controller; and these include [44]:

(i) OPC Protocol – Commonly known as Open-Platform Communications. This is a standard way that is used to set up communication between numerous data sources, including the devices on a factory floor, laboratory equipment, test-system fixtures and databases [45]. OPC usually has three platforms communicating with each other. These are the server, the client and the plant [46]. The OPC server runs a software program that converts the

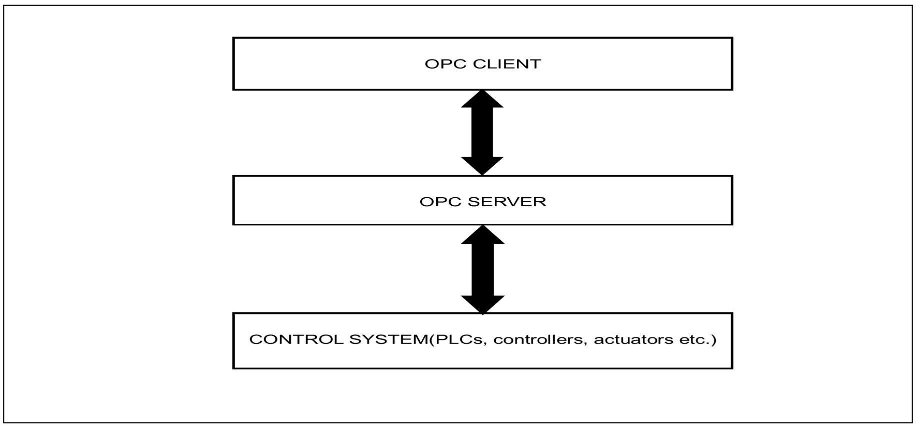


Figure 2.16: OPC Protocol

hardware/software communication protocol that is used by the system controller into the OPC protocol. The OPC client software is any program that needs to connect to the hardware, such as an HMI. The OPC client uses the OPC server to get the data from, or to send commands to the hardware/software. Figure 2.16 shows the flow of information between the client and the control system [45, 46].

(ii) TCP/IP Protocol – The framework of the TCP/IP protocol defines the actual source and set-up of the internet, as shown in Figure 2.17. Hence, it is also employed in private communication networks for the remote accessing of automated process control systems [47]. The communication protocol is a

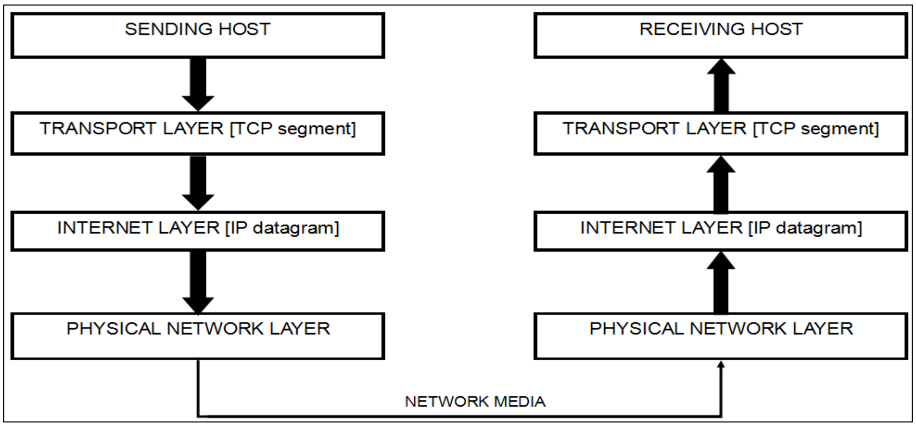

SENDING HOST
TRANSPORT LAYER [TCP segment]
INTERNET LAYER [IP datagram]
PHYSICAL NETWORK LAYER
RECEIVING HOST
TRANSPORT LAYER [TCP segment]
INTERNET LAYER [IP datagram]
PHYSICAL NETWORK LAYER
NETWORK MEDIA

Figure 2.17: TCP/IP Protocol

combination of two protocols that are defined as [47, 48]:

- The TCP (Transmission Control Protocol) – gives a service with the desired connection for transmission of the data. The TCP window control increases or decreases the traffic, according to the speed required to make a return.
- IP (Internet Protocol) – network level that gives a connectionless service that is defined with a unique address number for each unique device to be used on the network e.g. (193.89.71.0). The IP protocol was designed as an interconnection Protocol defining a data block of a well-defined format and containing an address, but no other functionality.

The discussed methods of communication between the different devices (i.e. actuators, PLCs, controllers, simulators) for process control systems are applied in industry. The methods have proven to be efficient and effective; thus they may be applied for any automated process-control system.

2.5.3 Internet Connection Types

The use of the internet has made the world a smaller place; as information and applications are readily available to people across the globe. However, the internet is accessible through different types of connections. These connections are different in terms of speed, cost and security. Usually, with internet connections, the different attributes and factors involved are directly related to each other. The faster the connection, the more expensive the connection. This applies to security also: the more secure the connection, the more expensive it is. Thus, to operate and access any system in real-time via the internet, the best connection would be the fastest connection type [49]. Some of the commonly used and reliable internet connection types include [50]:

(i) Ethernet over cable – Ethernet Over Cable is very similar to DSL; however, for this service it is provided over a coax cable infrastructure, instead of a phone line to connect to the Internet. Ethernet Internet has a speed range that goes up to

1000MB/s.

(ii) Fibre-optic Internet – Fibre-optic cables transmit the data through laser-generated pulses of light. The signal is ultimately understood by a computer; because it is converted by an Optical Network Terminal (ONT). The ONT is a media converter that can deliver phone, Internet and video services with a speed range that goes up to 2500MB/s.

(iii) Mobile Internet – This is mostly used on cellphones; and it comes with 3G and 4G networks, which are provided by mobile-phone operators. 3G mobile-phone networks are the most common; and they have the widest coverage area; whereas the newer 4G networks are less common with restricted coverage area. The download speed ranges from 1.5-25MB/s.

(iv) Wireless Internet – The wireless technology used is similar to home wireless networking; hence, for a device, such as a laptop/pda equipped for connection to a home or office wireless network, it would also work on a public wireless network. The download speed ranges from 50-500MB/s.

2.5.4 Remote and Virtual Laboratories in Universities

A remote and virtual laboratory is a laboratory which may be accessed via the internet from another computer. The laboratory has a programmed plant system (thus, it is termed virtual) with parameters that are being manipulated by the user to carry out the required tasks. The use of remote and virtual laboratories has become a norm in terms of training and teaching engineering students [51, 52].

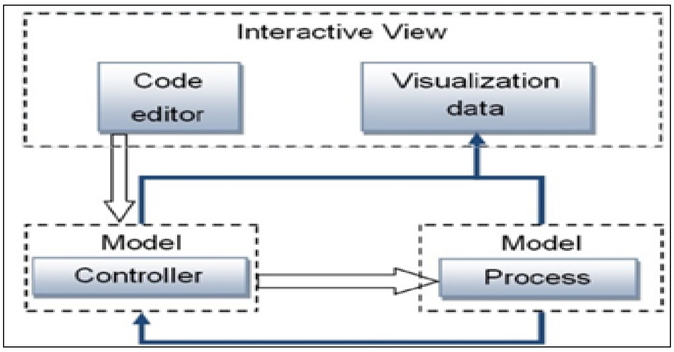


Figure 2.18: Basic Architecture used to setup a Remote Laboratory

The designing of remote and virtual laboratories follows the architecture shown in Figure 2.18. This consists of three main components:

- Model – In this architecture, the model represents the process system under study, having different variables and states. Such models are designed using simulation software, such as Matlab Simulink or LabView.
- View – This provides the graphical display and user interface of the model.
- Controller – The brain between the view and the model, whereby it responds to the user interactions; and it converts them into parameters for the model.

Some of the advantages that have been introduced by remote and virtual laboratories are [35, 36, 52] :

(i) Safety – constraining operations that avoid dangerous situations; hence students are safe at all times during the practicals.

(ii) Physical experimentation increases the costs incurred; as physical systems are expensive to acquire and maintain. Remote and virtual laboratories only cost a lot when setting up the platform; and they do not cost as much when it comes to maintenance.

(iii) On physical systems, there is also a constraint of location, some systems may be located in a place that is difficult to access for all students. However, when it comes to a remote and virtual remote laboratory, the students would need access to a computer that is connected to the internet.

(iv) Each student can actively participate on the platform.

(v) Testing of the same hardware on different plants or models. Remote and virtual laboratories save space; as there would be so many real models needed to accommodate all the students.

2.6 Industry Standards for Process Control Systems

In any field of production or professional work environment, there are standards that have to be met by all means. Standards ensure that clients receive satisfactory services and products. Standards also pave the way for any innovation that leads to better research and development. For any system that is developed on a commercial basis, the manufacturer has to meet certain standards. Standards are usually outlined by engineering boards; and they determine whether the product may be distributed [53]. Some of the engineering boards that set standards globally for process-control systems include:

- The American Society of Mechanical Engineers (ASME) [54]
- The International Society of Automation (ISA) [55]
- The International Council on Systems Engineering (INCOSE) [56]

The stated engineering boards have some common elements that an engineer must be aware of at all times. A system that is designed or operated by an engineer must adhere to the following criteria [57, 58]:

(i) Safety – This is the most important element; as it ensures that the operator of any system is safe at all times. Usually, the following are to be noted: Operating time of a system; noise levels; radiation levels and brightness levels.

(ii) The environmental effect of a product – The amount of toxic emissions from a system should always be known during manufacturing, as well as the lifetime of the product and its disposal. Hence, a certain critical amount (e.g. hydrocarbons) of materials used or emitted are usually set by the board.

(iii) Product management – The product management plan deals with how the product will be produced, used, distributed and disposed of. Most engineering boards now require a product that can be recycled to produce new products.

Thus, for developing a process-control teaching tool, it would be necessary to develop one that comprises hardware and software, which adhere to the engineering standards. A platform of such nature would also give insight to students on some of the engineering standards, which they are expected to learn and understand.

2.7 Fundamentals to teach Engineers Process-Control Systems

In most engineering universities, students are taught concepts from theory in the normal lectures (lecturer-student face-to-face in class). Many students face challenges in understanding what the lecturer is trying to put across to them – due to a number of factors, such as [59]:

- Language barrier – Some students face problems when learning in another language, other than their mother-tongue; hence they get confused, or they lose their enthusiasm to learn the subject.
- Concept complexity – Some concepts being taught in classes may be too complex for the students, for instance deriving a third-order Equation for a given system. In this scenario, students are bound to make mistakes; and they would never know their mistakes – until they get their test scripts back from the lecturer.
- Not enough physical evidence of concepts – At times students attend classes for the sake of having a good attendance record. This is because the theory taught to them may not show any physical impact on a particular machine or system.

The outlined problems have largely been addressed in most universities. In modern times, for any engineering or scientific course; there are three methods that students go through when tested on the course concepts [59]. These methods are shown in Figure 2.19; and they are applied in teaching and giving knowledge to engineering students in process-control systems. In sections 2.7.1 and 2.7.2 the theory and approach on how practicals for process control should be implemented are discussed, respectively.

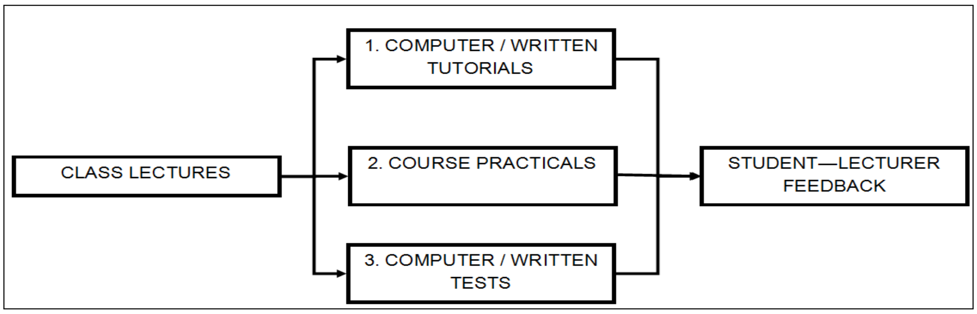

1. COMPUTER / WRITTEN TUTORIALS
CLASS LECTURES
2. COURSE PRACTICALS
STUDENT—LECTURER FEEDBACK
3. COMPUTER / WRITTEN TESTS

Figure 2.19: University Approach of Teaching Process-Control Courses

2.7.1 Theory for Process Control Systems

Students who are to complete process control courses are to receive a good grounding in process control principles, strategies, controller tuning and loop dynamics. Engineers and technicians with little or no experience often face challenges in industry; when they are given complex process-control systems to work with [60]. Most process control systems are designed around pneumatics, electronic instruments, analog devices, digital logic devices and other computer-based process-control devices. Hence, the theory that any process-control engineer should understand and know, is based on the devices used to design the systems and the fundamentals, which include [60, 61]:

(i) Pressure, Level and Flow of Fluids – In any system that handles a fluid (gas/liquid), the pressure, level and flow of the fluid should be known at all times. This is to ensure that the system functions normally; and that it is safe for any operator to control and monitor.

(ii) Temperature and Heat – The operating temperature of any system or device needs to be known, to ensure that the system operates without giving any problems. The knowledge of a system's temperature is necessary; as it helps in material selection, device selection and to know whether a cooling system is required for the plant.

(iii) Humidity, Density, Viscosity, and pH – The properties of any fluid and/or operating environment should be known at all times. For any process-control system; the product to be manufactured/treated may need specific fluid properties; hence, these properties need to be known – and how they affect one another (linear and non-linear) relationships.

(iv) Sensors, Actuators and Control – for any automated process-control system, monitoring and control are quite important. For these to be achieved, sensors and actuators are required in the system. A sensor is used for monitoring and measuring a process value and/or condition; while the actuators implement the action of the task required by the user.

(v) Controller design (PI, PD, PID, PI [Phase Lag/Lead], etc) – depending on the system hardware and parameters the controller for the effective response to a user's request is required. Hence; for the process-control system, a controller that executes in real-time is always required to have a stable working system.

The outlined fundamentals are usually introduced to students in separate courses – from the time they are enrolled in university. These fundamentals are then merged in process-control courses, with any other additions to the knowledge they have at hand. Such theory work and concepts may not all be understood by the students in depth in their previous classes. Hence; in process-control courses, it is important to ensure that they refresh and understand these concepts, through a series of practical courses that build up from the basic concepts to the more complex ones. This may only be possible by making use of a reconfigurable teaching platform.

2.7.2 Practicals for Process-Control Systems

Figure 2.20 illustrates the basic architecture of any process control practical system that most universities have. The user interface shows the student-system response and status.

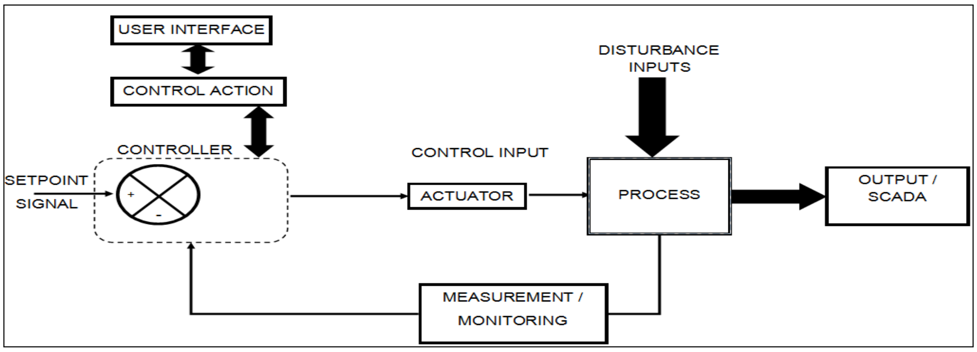


Figure 2.20: Process Control Practical setup

The student may set parameters that commonly involve integers or setting signals on and off (control action) via the interface (GUI/SCADA). The controller will then activate the actuator (pump, piston, motor); thus the process will be in action. For any real-time process, control system external disturbances are usually experienced. Such disturbances include (noise, temperature changes, heat, etc). From the process, the critical parameters are then monitored and measured through sensors giving feedback to the user through a GUI platform [61].

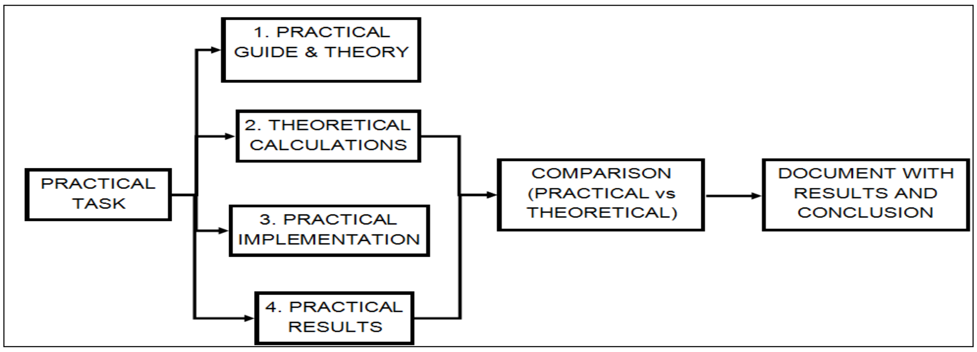

PRACTICAL TASK
1. PRACTICAL GUIDE & THEORY
2. THEORETICAL CALCULATIONS
3. PRACTICAL IMPLEMENTATION
4. PRACTICAL RESULTS
COMPARISON (PRACTICAL vs THEORETICAL)
DOCUMENT WITH RESULTS AND CONCLUSION

Figure 2.21: Student Approach to Process Control Practical

Carrying out a practical course is usually done in the manner shown in Figure 2.21.The lecturer usually explains to the students what the expected outcomes of the practical are; and the practical guide is then provided to the students. In the practical guide, the students would then be able to relate and identify what resources they may need to carry out the tasks at hand. Depending on the nature of the practical, there may be a need for the students to do their own research and mathematical calculations – before working on the actual practical equipment. This is to ensure that they set the right inputs to the system and achieve feasible results from the system.

After carrying out the practical, students will then carry out a comparison of the theoretical results with the real-time results from the practical. The students will then submit a document that is well-written that describes all the steps taken in the practical and visual evidence (graphs, tables, charts, etc) that are well-explained. This document is then evaluated by the lecturer; and in some cases, the student may be required to demonstrate the practical to the lecturer [55]. However, during the practical; it is never guaranteed that the system will respond in a friendly manner. At times, students may have miscalculated the parameters, or set them wrongly. Hence, this would have an impact on the system. The system may be damaged; or it may malfunction – to the extent that it may cause harm to the students. Thus, there is a need to ensure that the students are safe at all times. The introduction of remote and /or virtual laboratories has brought more safety for students compared to working on real hardware systems.

2.8 Literature Review Conclusion

A thorough research of the work has been done; and it is currently in progress, in order to give this research a significance and definition. In this chapter, the basic definition of automation in process control was looked at, as well as the different fields, in which it is applied. Examples have been given; so that there is a clear understanding of how automated process-control systems are set up and function.

The hardware and software used in industry for real and simulated process-control systems were also discussed; hence, the importance of simulation has been identified. The process-control software used in universities is also looked at; as this gives a foundation of what a teaching platform should achieve; as well as the limitations that the existing ones may have. The operation and monitoring of industrial systems is currently moving to the phase of being implemented through the internet. Hence Industry 4.0 and the current methods of using the internet on automated process-control systems were looked at.

In the last section, the critical elements and methods used for teaching engineering students process-control systems were looked at; and the justification for these are shown. To conclude, this chapter has discussed the information, which is used as a guide in the selection of hardware and software for setting up the final platform for the research. Chapter 3 discusses how the platform has been developed to meet the outlined objectives, as stated in Chapter 1.

Chapter 3

Platform Development

This chapter discusses the development of the platform based on the outlined requirements. Figure 3.1 shows the industry standard technologies used to set up the system that incorporates all the hardware and software. These are discussed in sections 3.1 and 3.2, respectively.

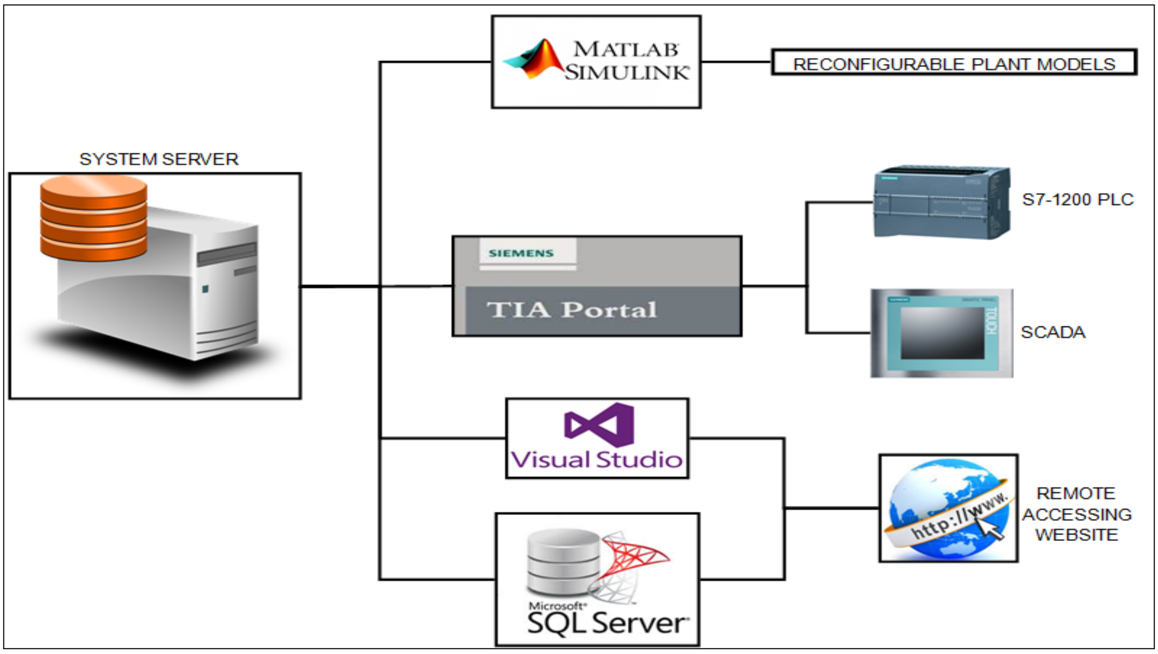


Figure 3.1: Platform Hardware and Software

The system server used for this research has the following specifications:

- Random Access Memory - 8GB, System type - 64-bit
- Hard Disk Speed - 7 200 RPM, Graphics Card Size - 1GB
- Processor Type - i7, Processor Speed - 3.4 GHz

The platform server was used for programming and interfacing the different subsystems that make up the platform; and the tasks implemented include:

- Plant Modeling
- PLC programming
- Interfacing plant models with PLC
- SCADA design
- Website design and interfacing with SCADA

3.1 Technology for developed Online Platform

This section discusses the selection of the hardware used in the development of the platform, the industry controller and the ethernet switch. Figure 3.2 shows the architecture of the fully integrated platform. Each component is discussed in the sections that follow.

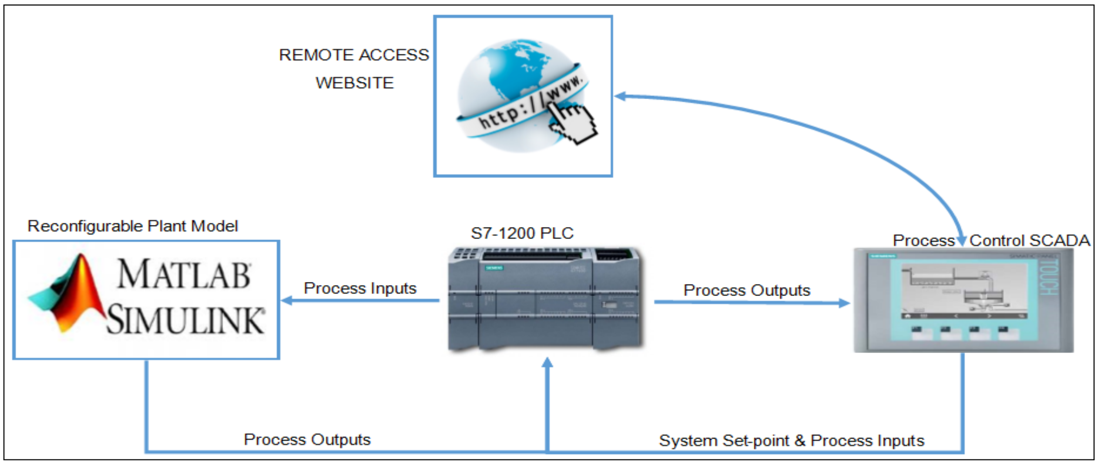


Figure 3.2: Architecture for accessing SCADA via Website

3.1.1 System Controller

The controller for the system was selected on the basis of matrix Table 3.1; and the PLC gave the highest overall score. The rating of the matrix table is defined as:

- Cost – The higher the cost of the controller, the lower the score given, and the lower the cost, the higher the score given.
- Industry use – The more the applications in industry of the controller, the higher the score awarded.
- Integration to external systems – The more applications with various systems (hardware and software), the higher the score given.
- Processing power – A controller that makes use of the lowest processing power is awarded a high score.

Table 3.1: Controller-selection matrix

Factors considered	Weighting	Micro-controller	PLC	Industrial PC
Cost	25	22	17	13
Industry use	35	15	30	25
Integration to external systems	25	17	15	16
Processing power	15	14	12	10
Overall Score	**100**	**68**	**74**	**64**

PLCs are commonly used in industry compared to micro-controllers and industrial PCs. Even though a micro-controller would work on the system, its use in industry is compliant compared to a PLC or Industrial PC [62, 63]. This factor is one of the outlined requirements for the research, namely, the use of an industry standard controller. The Industrial PC could be used for the platform; but it could have limitations, when it comes to costs; as it is more expensive than a PLC or micro-controller. Some Industrial PCs come with their operating system software;

hence, it might not be compatible with external systems' software [64]. The integration factor contributes to the reconfigurability of the system; as it should be able to be interfaced with different plant models or software.

From the three controllers considered; the PLC consumes less processing power than an Industrial PC; hence it is a better option than the micro-controller. An educational platform would not need high processing power; but it would need real-time response; as micro-controllers tend to be slower than PLCs; as the programs become complex. For this research; the Siemens S7-1200 PLC is used as the system controller, as shown in Figure 3.2. The controller offers the following functionalities that were found suitable for the platform [65]:

- It allows remote connection and access to other software and hardware systems, using communication protocols, such as UDP, TCP/IP Protocols, Ethernet, Profibus, etc.
- Real and virtual instrumentation can be implemented using the PLC; and thus it is possible to have a fully functional SCADA [66].
- It offers adequate computational power for the platform (processing-plant model variables), such that the platform operates in real-time.

3.1.2 Ethernet Switch

The designed platform has various sub-systems, which communicate with each other (platform server, PLC, plant models, SCADA). Hence, a private network for the platform was set up by the NMMU ICT services. The stated sub-systems are interconnected on the network via a D-Link Ethernet switch. The devices' TCP/IP configurations are set up to the same subnet of the private network. The network domain for this platform is only limited to the NMMU domain. The selected Ethernet switch has 16 ports, which provide room for future expansion on the platform. It also provides a substantial internet speed of 1000Mbps; consequently, the sub-systems can communicate in real-time.

3.2 Platform Software

This section discusses the different software packages used to program the simulated plant models, PLC, SCADA and website.

3.2.1 Plant-Modeling Software

Various software packages were considered for programming the plant models. The two commonly used software packages for designing complex mathematical and engineering plant models are Matlab Simulink and LabView. These are used both in industry and largely in learning institutions. However, the plant-model designing software used to set up the plant models is Matlab Simulink. Matlab Simulink is a graphical design software that allows for the development of simple mathematical and complex modelling of various systems that include electrical, mechanical and chemical [31]. The selection criteria used is shown in matrix Table 3.2, whereby Matlab Simulink has a higher overall score and meets the requirements for the system. The selection criteria in the matrix table are defined, by using factors, such as:

- Connectivity to external systems – A software package used in various applications is given a higher score.
- Educational use – The more the software applications in engineering learning, the higher the score awarded.
- Programming difficulty – A software that is more difficult to program is awarded a lower score.
- Model development complexity – A software that provides room for the development of models with various parameters; and more than one engineering discipline is awarded a higher score.

Table 3.2: Plant Modeling Software Selection Matrix

Factors considered	Weighting	Matlab Simulink	LabView
Connectivity to external systems	30	25	20
Educational use	40	35	30
Programming difficulty	20	12	16
Model development complexity	10	8	6
Overall Score	**100**	**80**	**72**

The interfacing of Matlab Simulink to external systems is done in a number of engineering disciplines, such as aeronautics, chemical, electrical, mechanical, mechatronics and many more. However, LabView seems to be limited to the chemical and process-control disciplines [32, 33]. Hence; this automatically proves why Matlab Simulink is used more for educational purposes.

3.2.2 Controller Programming Software

TIA Portal

Totally Integrated Automation Portal (TIA Portal) software provides a large platform that offers digitalized automation services, from digital planning and integrated engineering to transparent operation. The software package has been used to program the S7-1200 PLC, design SCADA and to set up communication between PLC and plant models [65, 66].

System SCADA

The SCADA for the platform is designed in WinCC, which is a Siemens product that comes with TIA Portal Software Package. WinCC SCADA offers the platform merits that include [66]:

(i) Reducing costs; since it replaces hardware that may be expensive to maintain or replace. This ensures that the platform users are also safe at all times; as they do not handle dangerous hardware.

(ii) Allowing the designing of different systems by using the same software package. Hence, the platform can accommodate different plant models (dual-tank model and submarine ballast tank control model) designed with their different monitoring and operating tools.

(iii) Operation is in real-time; because it is programmed in the same environment as the PLC program interfaced to a plant model – the data is written and read through PLC tags; and the user can visualize the system status and response.

(iv) Smart Service – this is an option that is available in WinCC that is flexible for communication between HMI systems or SCADA systems via TCP/IP connections (e.g. Internet, LAN). This feature allows the SCADA to be accessed on the network that is configured on the computer running the SCADA [67]. The other features offered by the Smart Service, which are used in the set-up of the platform, include:

  - The remote access to SCADA systems and process data on a PLC from any location.
  - Multiple SCADA systems can be accessed on a single network.

3.2.3 Remote Accessing Software

To remotely access and operate the platform on a managed and controlled system; a user-defined website was designed for the platform. Remote desktop software was considered; as it offers the use of a PC on another PC remotely.

The factors outlined in matrix Table 3.3 give the justification of developing the platform website. The connectivity of a remote desktop software is better than a website; as the user operates directly on the system's PC [68]. However, remote desktop software is an off-the-shelf product, which comes at a high cost. For the outlined online operation deliverables using a remote desktop software would result in more costs; as

Table 3.3: Remote Accessing Software selection matrix

Factors considered	Weighting	Remote Desktop Software	User-defined Website
Connectivity to external platforms	30	20	18
Cost	30	10	25
Programming difficulty	25	20	15
User access limitation	15	7	14
Overall Score	**100**	**57**	**72**

the software providers have to set up a platform that meets the system's requirements [69]. The users are to access the SCADA without having access to the controller or any plant-model programs.

Remote desktop software gives full access to users; hence for the developed platform, the user access limitations would only be met by developing a website that is built around the designed SCADA and the plant-model system. The user-defined website was designed by using Visual Studio and SQL Server.

Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as websites, web applications and web services. Visual Studio uses Microsoft software development platforms, such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight [70]. Hence; with these outlined features, Visual Studio 2015 is used for the development of the remote accessing website.

SQL Server

SQL Server is a Microsoft product used to manage and store information. The SQL Server is a “relational database management system” (RDMS). The SQL Server platform is designed to integrate with multiple programs that are programmed and run in Visual Studio [71]. Consequently, to store and retrieve information for the platform website, Microsoft SQL Server 2014 is used as the database designing and management tool.

3.3 Hardware and Software Selection

The hardware and software used to set up the platform discussed were selected; because they provide the required functionalities that meet the platform requirements stated in Chapter 1. The selection was also done, based on the past, present and future work on industrial and educational systems discussed in Chapter 2. Sections 3.4 to 3.9 discuss the steps and methods used to build the platform sub-systems and their integration.

3.4 Plant Modeling

Two plant models are designed and discussed in this section, with the SCADA homepage shown in Figure 3.3. Having two plant models ensures that the platform can be shared by different engineering disciplines; and it proves that the platform is reconfigurable. Sub-sections 3.4.1 and 3.4.2 discuss a dual-tank model and a ballast tank-control model for a submarine, respectively. The mathematical Equations are derived and then programmed into Matlab Simulink models.

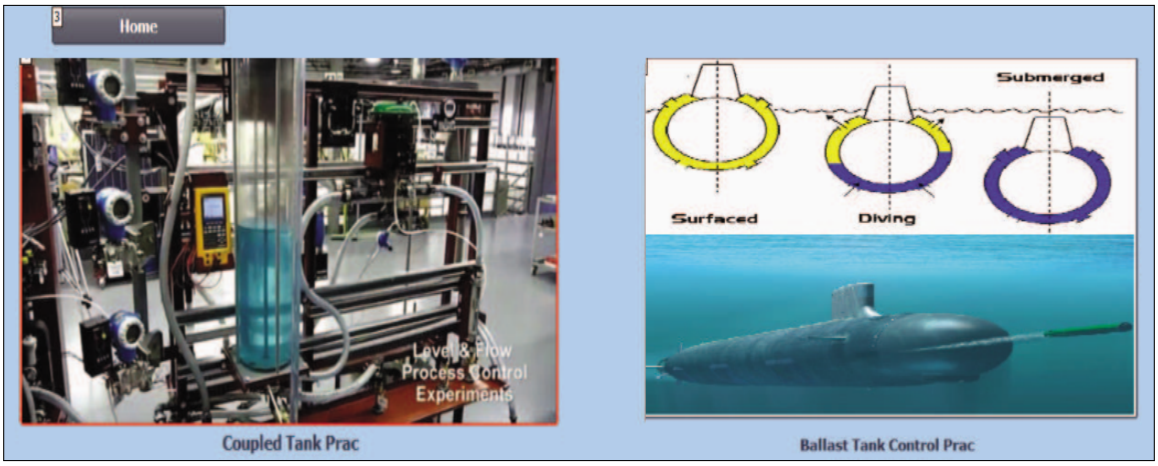


Figure 3.3: Platform SCADA for plant models

3.4.1 Dual-Tank Model

The model shown in Figure 3.4 is a dual-tank plant that consists of two tanks that are of variable parameters. A model of this nature is often used to teach engineering students concepts of process-control systems in both theoretical and practical classes.

The following parameters have been considered in designing the model:

- Valve – A valve is a device that regulates, directs and/or controls the flow of a fluid (gases, liquids, etc) by opening, closing, or partially obstructing various passageways. In an open valve, fluid flows in a direction from higher pressure to one of lower pressure.

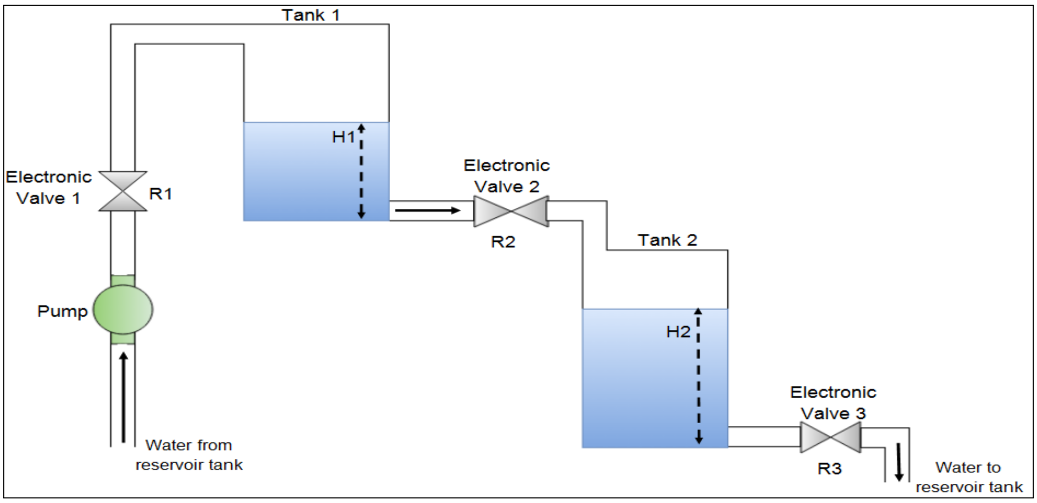


Figure 3.4: Schematic of Dual Tank Process

- Tank surface area – Fluid tanks are large vessels for storing fluids. They come in a variety of styles, including horizontal cylinders, vertical cylinders and rectangles. Thus, to determine the volume capacity of any tank, the surface area needs to be known at all times. Usually, for changing volume, it is considered as a constant.

- Tank height – The height of any vertical tank is usually fixed; as the surface area, but it is used to measure the volume of a fluid inside the tank at any point in time.

- Pump discharge – A pump is a device that moves fluids (liquids or gases) by electro-mechanical action. It is used for transferring fluids/gases from one vessel to another. Hence, pump discharge is the rate at which the pump moves the fluid from one vessel into the desired vessel.

The first tank receives water directly from the pump; and an interconnecting pipe between the two tanks allows for the transfer of water from the first tank to the second tank. Each of the tanks has a variable surface area that may be set in the program. In this model (Figure 3.4), there are three valves located at the feed-input from pump to tank 1, between tank 1 and tank 2, and at the outlet of tank 2.

3.4.1.1 Mathematical Background for Dual Tank Model

In this section, the main focus is on the theoretical Equations and the justifications used to set up the model. The model is designed with the following assumptions:

- There is no pressure loss within the system (transfer pipes, valves).
- The effect of flow rate is from the valves only (pipe lengths have been excluded).

The pump parameters selected for the model are based on an NMMU Dual-tank system [72] in Table 3.4 and these are also represented in the schematic shown in Figure 3.4. They are described below.

Table 3.4: Variables Used to model Dual-Tank Plant

Parameter	**Variable**	**Units**
Pump Speed	0-2900	rpm
Pump Discharge	0-40	l/min
Tank Dimensions $L \times W \times H$	0-140 ,0-130 ,0-300	mm
Gravitational acceleration (g)	9.81	ms^{-2}

Q $=$ flow rate$(\mathrm{m}^3\mathrm{s}^{-1})$

A$_c$ $=$ valve cross sectional area(m^2)

d $=$ valve diameter (m)

ΔH $=$ pump head (m)

Pa $=$ pressure in pascals

A$_1$ $=$ surface area of Tank 1 (m^2)

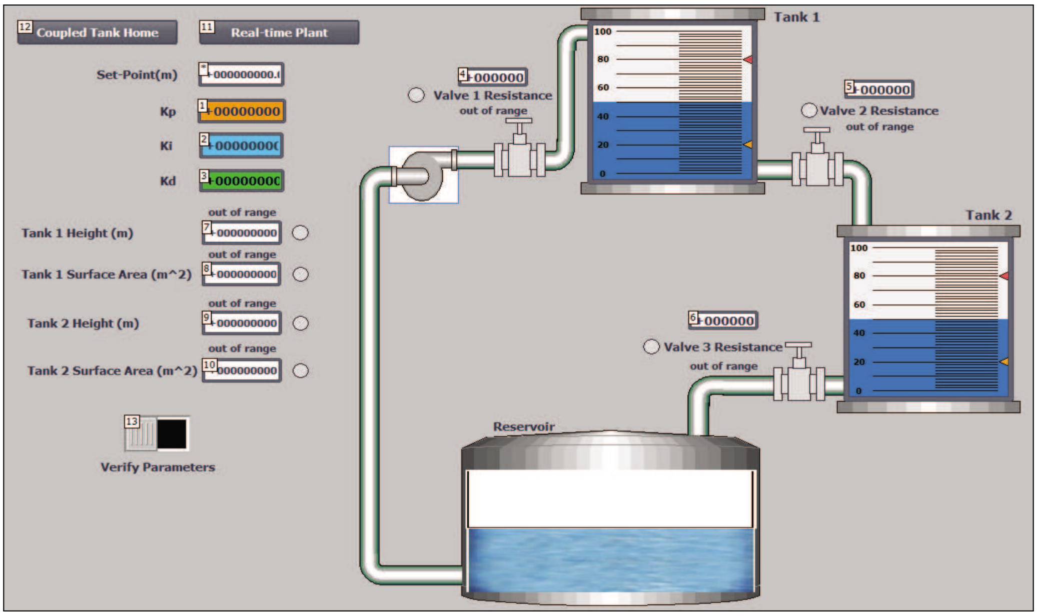
A$_2$ $=$ surface area of Tank 2 (m^2)

R$_1$ $=$ resistance of valve 1 $(\frac{\mathrm{Pa}}{\mathrm{m}^3\mathrm{s}^{-1}})$

R$_2$ $=$ resistance of valve 2 $(\frac{\mathrm{Pa}}{\mathrm{m}^3\mathrm{s}^{-1}})$

R$_3$ $=$ resistance of valve 3 $(\frac{\mathrm{Pa}}{\mathrm{m}^3\mathrm{s}^{-1}})$

The user can change model parameters via SCADA, as illustrated in Figure 3.5. Parameters that the user can change for the dual-tank model are the pump head which is set-point, valve resistances, tank 1 (area and height), tank 2 (area and height) and PID parameters. For any given fluid that is influenced by the atmospheric



Set-Point(m) * +000000000.
Kp 1 +00000000
Ki 2 +0000000(
Kd 3 +0000000C
out of range
Tank 1 Height (m) 7 +000000000 ○
out of range
Tank 1 Surface Area (m^2) 8 +000000000 ○
out of range
Tank 2 Height (m) 9 +000000000 ○
out of range
Tank 2 Surface Area (m^2) 10 +000000000 ○
4 +000000
○ Valve 1 Resistance
out of range
5 +000000
○ Valve 2 Resistance
out of range
6 +000000
○ Valve 3 Resistance
out of range

Figure 3.5: Dual tank parameters on SCADA

pressure and gravity, the flow rate from one vessel to another vessel is given as: Assuming that the valve is circular in shape [73] :

$$A_c = \frac{\pi d^2}{4} \text{ thus } Q = A_c \sqrt{(2gh)} \tag{3.1}$$

Thus, the pump head per unit time is equivalent to h and maybe given as:

$$\Delta H(t) = \frac{Q^2}{2gA_c^2} \tag{3.2}$$

3.4.1.2 Tank 1

Tank 1 is modelled into Matlab Simulink following the differential Equation formulated to give the height of the fluid in the tank at any point in time $h_1(t)$ Tank 1 differential Equation is derived as [73, 74]: Flow rate into tank 1 ,

$$A_1 \frac{dh_1(t)}{dt} = \frac{\Delta H - h_1(t)}{R_1} - \frac{h_1(t)}{R_2} \tag{3.3}$$

Thus the final differential Equation for Tank 1 is given as [73, 74]:

$$\frac{dh_1(t)}{dt} = \frac{\Delta H - h_1(t)}{A_1 R_1} - \frac{h_1(t)}{A_1 R_2} \tag{3.4}$$

3.4.1.3 Tank 2

The differential Equation formulated to give the height of the fluid in the tank at any point in time $h_2(t)$,

$$A_2 \frac{dh_2(t)}{dt} = \frac{h_1(t)}{R_2} - \frac{h_2(t)}{R_3} \tag{3.5}$$

Thus the final differential Equation for Tank 2 is;

$$\frac{dh_2(t)}{dt} = \frac{h_1(t)}{A_2 R_2} - \frac{h_2(t)}{A_2 R_3} \tag{3.6}$$

3.4.1.4 Matlab Simulink Modeling for Dual-Tank

The dual-tank model Matlab Simulink code shown in Figure 3.6 has been modeled using the derived Equations 3.4 and 3.6. The model and all its input and output parameters are transferred to and from the PLC via the TCP/IP Protocol.

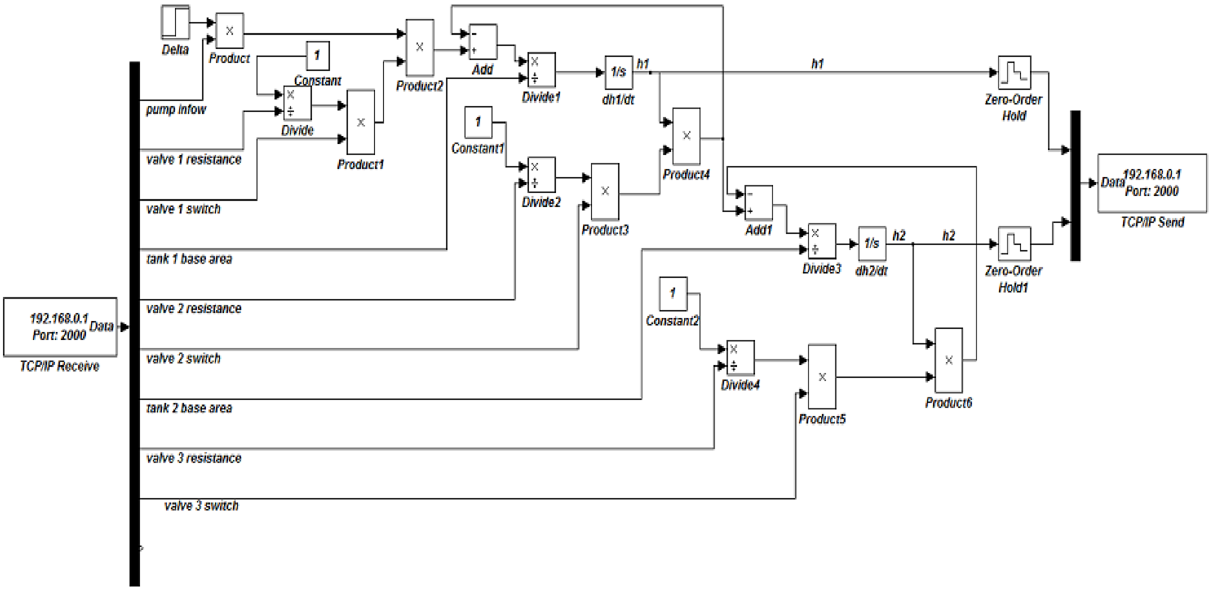


Figure 3.6: Dual-tank model integrated to PLC

3.4.2 Submarine Ballast Tank Control Model

In this section, a model to control and monitor a submarine's depth underwater is developed. A submarine makes use of a ballast tank to achieve its movement under water to reach the required depths. Hence, there are parameters that may be manipulated to have full control of the submarine's depth, which could be the volume of air in the ballast tank, the volume of water in the ballast tank, the vessel's shape, etc.

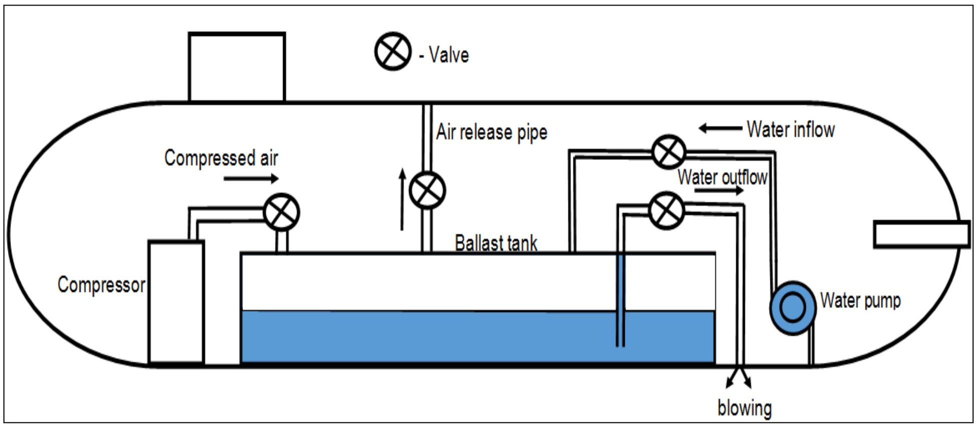


Figure 3.7: Structure of a Submarine's ballast tank

Submarines and other large water vessels depend on ballast tanks for depth and stability control, as shown in Figure 3.7. These tanks are filled with air and water in different proportions – for the vessel to be heavier or lighter in weight. In instances where the submarine is to ascend, compressed air is pumped into the tank under high pressure, so as to displace the water in the tank. When submerging, the air outlet valve is opened, releasing high pressure air in the vessel; and water from the surrounding sea is pumped into the tank. Hence, the model has been developed using linear and non-linear parameters that affect the depth and buoyancy of a submarine in its normal environment [75, 76]. The model is assumed to have an elliptical cross-sectional area, constant gravitational force acting on the submarine's dry mass, one ballast tank and small variations in the density of the surrounding water [76].

The section that follows describes the mathematical modelling done to derive the Equations used to program the model in Simulink.

3.4.2.1 Mathematical Background for Ballast-Tank Model

Figure 3.8 shows the main forces that act on a submarine. These forces determine the submarine's rising and submerging capabilities. Hence, Equations (3.7, 3.8, 3.9, 3.10) have been used to derive the final Equation of motion used to model the depth control of a submarine [77].

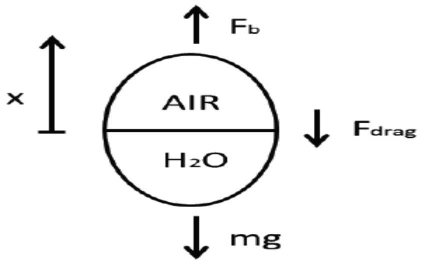


Figure 3.8: Main forces that act on a Submarine

$$\sum F = \text{-mg} + F_b - F_{drag} \tag{3.7}$$

$$F_{drag} = 0.5(C_d)(\rho_{H2O})(A)(x')^2 \tag{3.8}$$

$$F_b = (\rho_{H2O} - \rho_{air})(V_{air})(g) \tag{3.9}$$

$$\text{Tank-area}(A) = \Pi * b * c \tag{3.10}$$

The parameters described in Table 3.5,can be changed by the user, except for the gravitational acceleration constant, the density of water and the density of air. The user has to enter all the required parameters in the fields shown in Figure 3.9 before running the plant model. If invalid values are given, the system controller will notify the user and set the parameter to a default value.

Time domain Equation 3.11 of motion for submarine,

$$(m_{total})\frac{d^2x(t)}{dt^2} = \text{-(mg)+ } (\rho_{H2O} - \rho_{air})(V_{air})(g) - \frac{1}{2}(C_d)(\rho_{H2O})(A)(\frac{dx(t)}{dt})^2 \tag{3.11}$$

Table 3.5: Variables Used to model Ballast-Tank Plant

Variable Parameter	Value	Units
m(dry mass)	1 400 000	kg
ρ_{H20}(density of water)	1025	kgm^{-3}
ρ_{air}(density of air)	1.225	kgm^{-3}
C(total volume of ballast tank)	2000	m^{3}
g(gravitational acceleration constant)	9.81	ms^{-2}
C_d(drag coefficient)	0.7	
a- length of tank	50	m
b - short diameter of tank	2	m
d - long diameter of tank	4	m

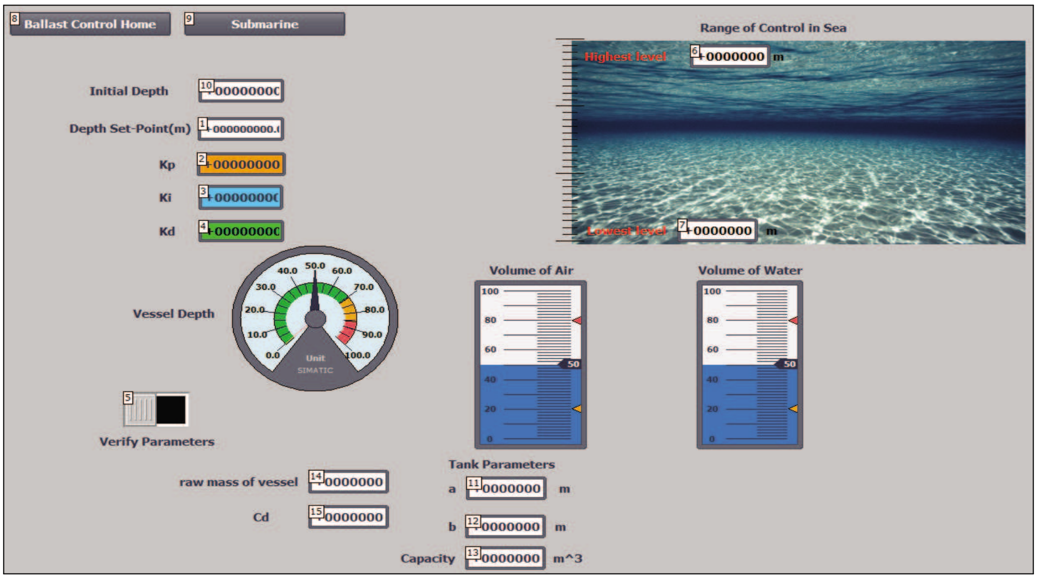


Figure 3.9: Ballast-tank parameters on SCADA interface

whereby

$$\mathrm{m_{total}} = \mathrm{m} + (\rho_{\mathrm{H2O}})(\mathrm{C} - \mathrm{V_{air}(t)}) + (\rho_{\mathrm{air}})(\mathrm{V_{air}(t)}) \tag{3.12}$$

Thus, the final second-order differential Equation for the submarine depth control model is given as Equation 3.13. This Equation is used in the design of the Matlab Simulink model shown in Figure 3.10.

$$\frac{\mathrm{d^2x(t)}}{\mathrm{dt^2}} = \frac{-(\mathrm{mg}) + (\rho_{\mathrm{H2O}} - \rho_{\mathrm{air}})(\mathrm{V_{air}})(\mathrm{g}) - \frac{1}{2}(\mathrm{C_d})(\rho_{\mathrm{H2O}})(\mathrm{A})(\frac{\mathrm{dx(t)}}{\mathrm{dt}})^2}{\mathrm{m} + (\rho_{\mathrm{H2O}})(\mathrm{C} - \mathrm{V_{air}(t)}) + (\rho_{\mathrm{air}})(\mathrm{V_{air}(t)})} \tag{3.13}$$

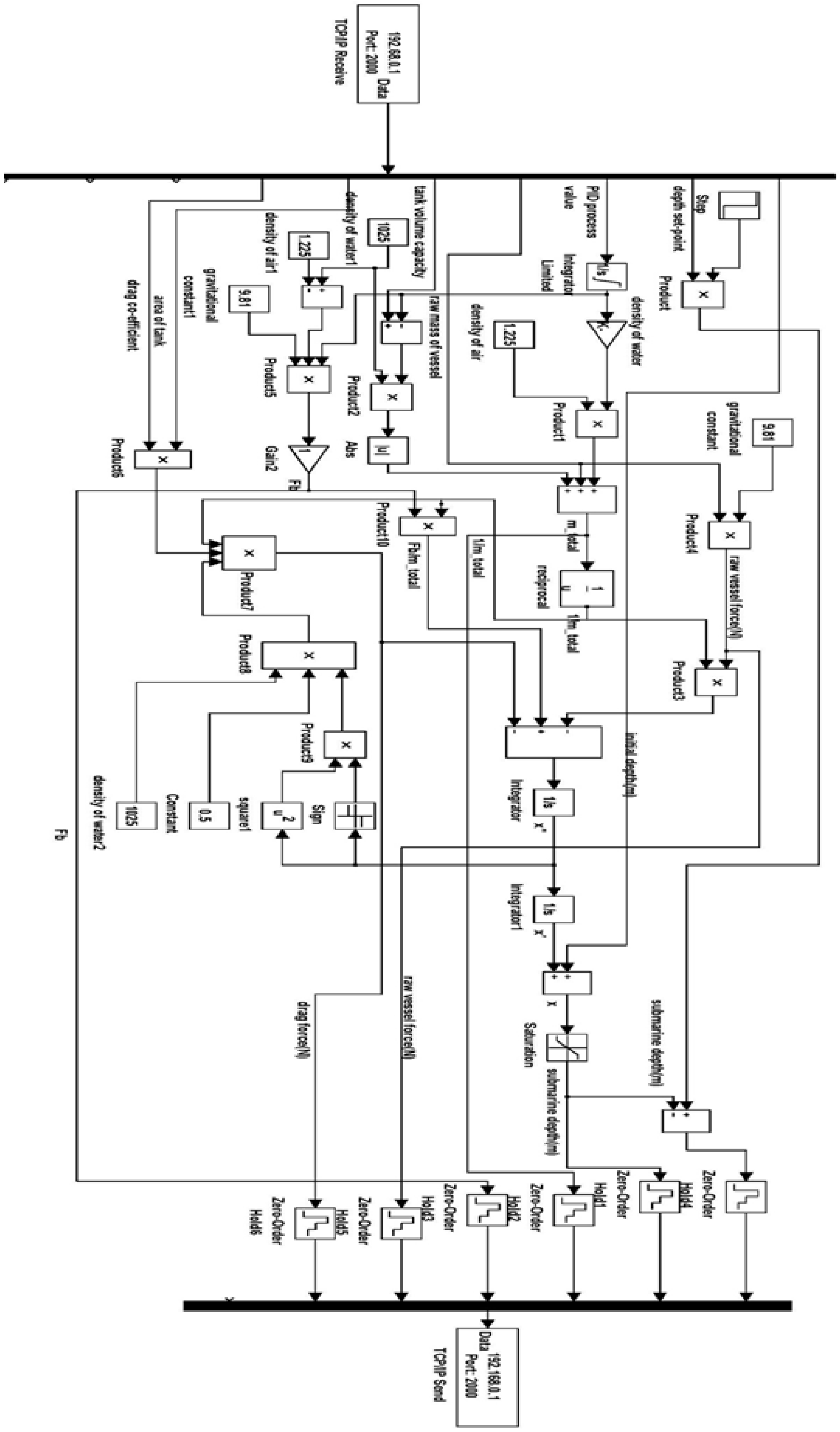


Figure 3.10: Ballast-tank control model integrated to PLC

Upon testing the models to be functional and giving feasible responses, the models were integrated to the PLC, such that the various parameters shown in Table 3.4 and Table 3.5 may be changed by the user via SCADA and processed by the PLC. This is discussed in section 3.5. The development of two plant models on this platform has been done to ensure that the users can understand the basic operation of the platform by working with a basic dual-tank system and a more complex ballast-tank control model. From these models, it may be concluded that the platform may also make use of other plant models designed in Matlab Simulink. Hence, the developed platform can be reconfigured for numerous plant models (motor control, actuator control, etc.).

3.5 Integration of Matlab Simulink Models to PLC

This section focuses on the connection implemented between the system controller (PLC) and the Matlab Simulink. Since this is an integration of different software, an effective method of communication is needed to ensure that there is real-time communication and minimal data distortion. The communication set-ups for both models and PLC are discussed. The TCP/IP communication Protocol has been used to interface the Matlab Simulink models with the PLC. The communication architecture is shown in Figure 3.11. The IP address and port values are set on the PLC.

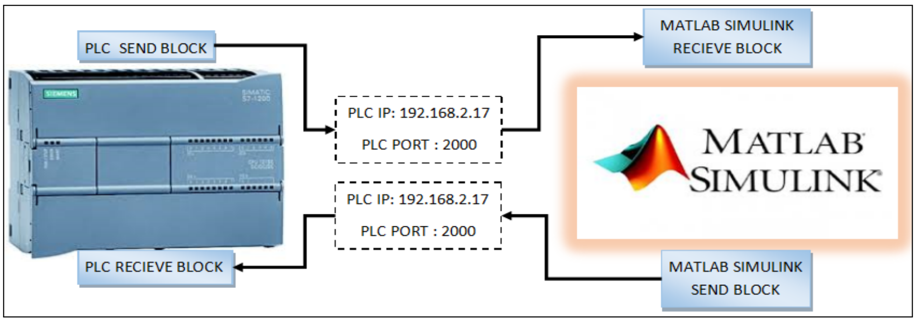


Figure 3.11: Communication architecture between PLC and plant models

The integrated PLC and plant models transfer input and response parameters on the platform, as illustrated in Figure 3.12. The PLC receives and sends the plant response to the user via the SCADA. The user inputs are read on the SCADA and sent to the PLC, which then sends these values to the plant model.

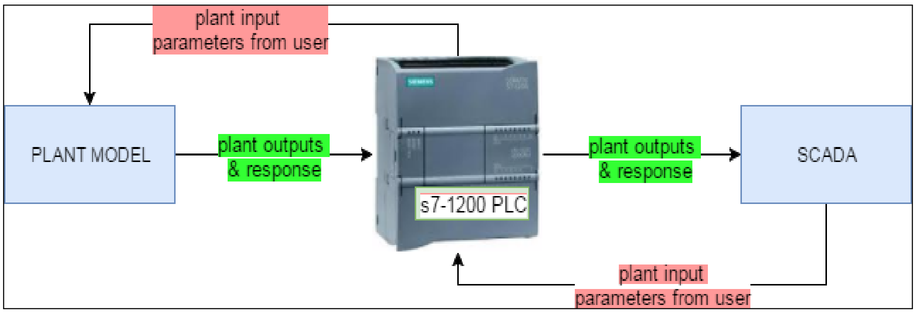


Figure 3.12: Transfer of data during PLC and plant model communication

3.5.1 Matlab Simulink Set-up for Communication with PLC

In Matlab Simulink, the TCP/IP protocol makes use of two blocks that receive the data and send the data, respectively. The configuration and initialization of these blocks occur once, which is at the start of the model's execution. The TCP/IP Receive block (Figure 3.13) connects to an interface of a specified remote address using the TCP/IP protocol. During the model's run time, the block acquires the data – either in blocking mode or non-blocking mode. In non-blocking mode, the block would have two output ports, the data port and the status port. At every time step, the data port outputs the requested values from the buffer. In non-blocking mode, the status port indicates whether the block has received new data [31]. The TCP/IP Send block sends the data from the model to remote machines. This data is sent at the end of the simulation or at fixed intervals during a simulation. The TCP/IP Send block (Figure 3.14) has one input port and the size of the input port is dynamic and is inherited from the driving block [31].

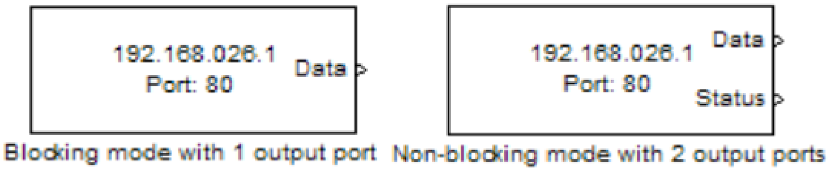

192.168.026.1
Port: 80
Data
Blocking mode with 1 output port
192.168.026.1
Port: 80
Data
Status
Non-blocking mode with 2 output ports

Figure 3.13: Matlab Simulink TCP/IP Receive Block

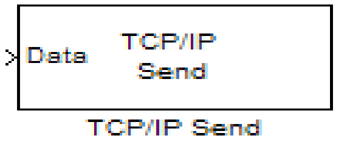
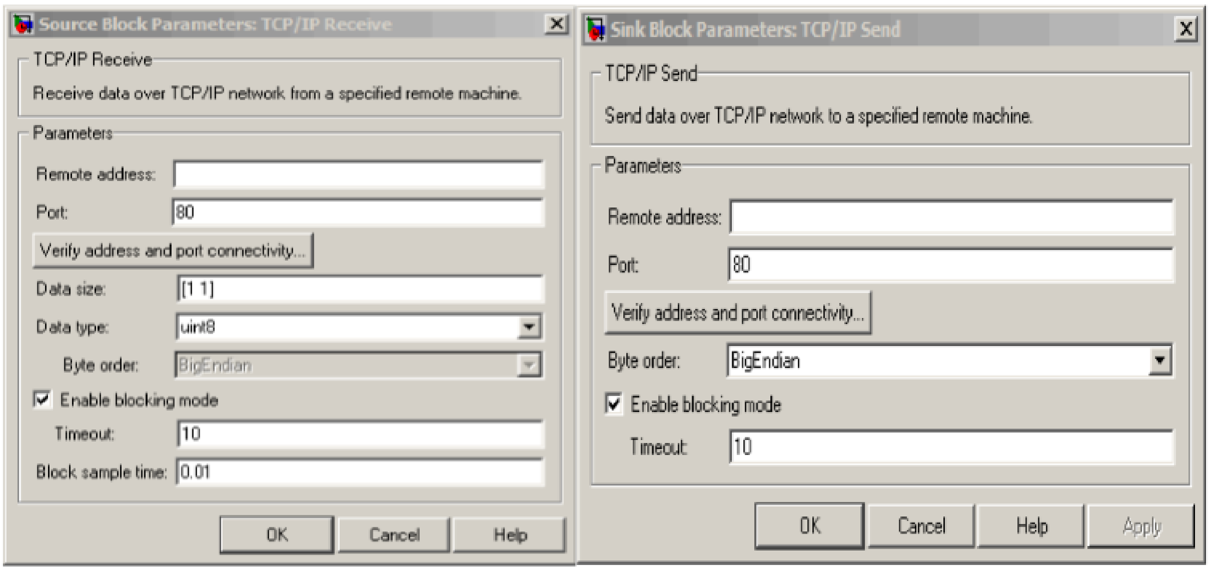


Figure 3.14: Matlab Simulink TCP/IP Send Block

The configuration of the Matlab send-and-receive communication block parameters is shown in Figure 3.15; are discussed in Table 3.6.



Source Block Parameters: TCP/IP Receive

TCP/IP Receive

Receive data over TCP/IP network from a specified remote machine.

Parameters

Remote address:

Port: 80

Data size: [1 1]

Data type: uint8

Byte order: BigEndian

☑ Enable blocking mode

Timeout: 10

Block sample time: 0.01

Sink Block Parameters: TCP/IP Send

TCP/IP Send

Send data over TCP/IP network to a specified remote machine.

Parameters

Remote address:

Port: 80

Byte order: BigEndian

☑ Enable blocking mode

Timeout 10

Verify address and port connectivity...
OK
Cancel
Help
Verify address and port connectivity...
OK
Cancel
Help
Apply

Figure 3.15: Source Block Parameters: TCP/IP Receive and Send

Table 3.6: Matlab TCP/IP Protocol Configuration Parameters

Parameter	Receive block	Send block
Data size	Specifies the output data size, or the number of values that should be read at every simulation time step. Usually these are arrays whose size depends on the number of values being received	The data size for the Send block depends on the number of outputs from the model being sent to the block
Data type	The output data type to be received from the block can be selected. The data types available include: single, double, int8, uint8, int16, uint32	The data type for the Send block depends on the output values from the model
Remote address	This refers to (the IP address, the name or the Web server address) of the machine, from which the data is received from and sent	
Port	The remote port on the remote machine you need to connect to. The port value for the (receive and send) communication blocks is the same for both host and client (i.e. PLC and Matlab Model)	
Verify and port connectivity	This button is used to check whether the specified remote address and port are correct. After verification, an establishment of the connection is made to the remote machine	
Byte order	When using binary or bin-block format with more than 8 bits, one can specify the instrument's byte order for the data. The options available are Big Endian or Little Endian. Both send and receive data blocks must have the same byte order if they are connected to the same machine	
Enable blocking mode	Selection to block the simulation while receiving/sending data. This option is selected by default. This check box can be cleared if the read/write operation is not needed to block the simulation	

3.5.2 PLC Setup for Communication with Matlab Simulink

The S7-1200 PLC has two communication blocks that can be used to implement the TCP/IP Protocol. The TRCV C DATA BLOCK (Figure 3.16) initiates the receiving Protocol of the PLC. The data is received in form of array(s) by RECEIVED_Data.table_DI of selected data types that are then used for different calculations and decision-making instructions for the controlled plant/system [65].

The TSEND C DATA BLOCK (Figure 3.17) initiates the sending Protocol of the PLC.

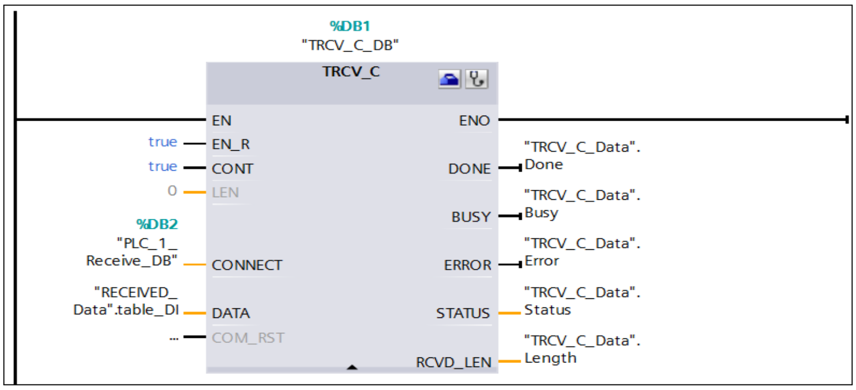


Figure 3.16: S7-1200 TCP/IP Receive Block

The data is sent in form of array(s) by the SEND_Data.table_AO of selected data types that are set as inputs to the controlled system. The block writes data over Ethernet (native TCP or UDP).

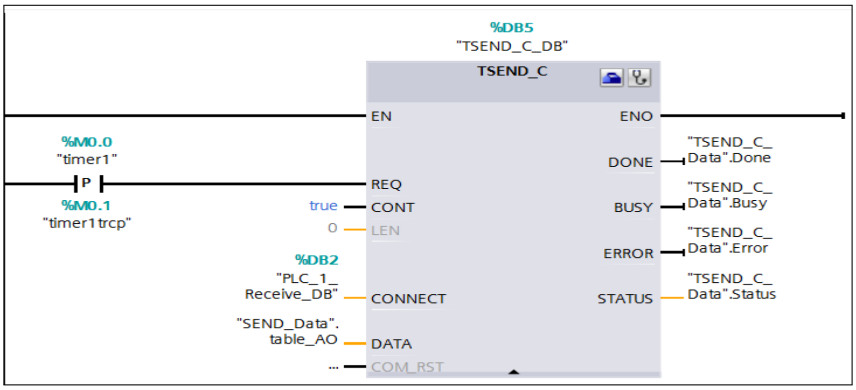
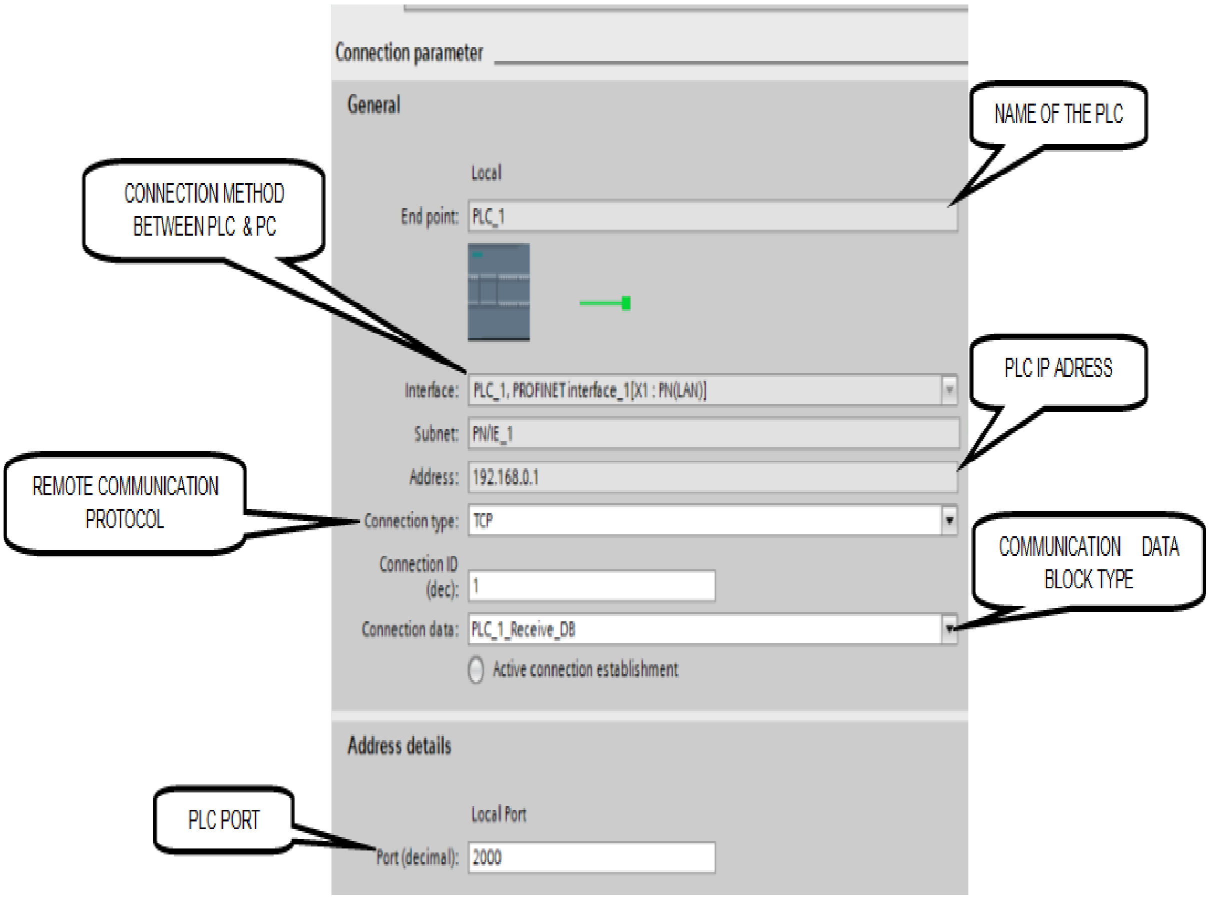


Figure 3.17: S7-1200 TCP/IP Send Block

The common parameters on the send and receive communication blocks are described as follows [65]:

- REQ – This is the control parameter request, which executes the terminating connection.
- DONE –This is a status parameter that indicates whether sending or receiving has been initiated or implemented without any error. It is represented by a 1 for job executed without error, or 0 the job has been initiated.
- BUSY – This is a status parameter that indicates whether sending or receiving has been completed. It is represented by a 1 for job complete, or 0 for job incomplete.
- ERROR – This is an error status parameter; and it is only represented by a 1; and it also provides details of the error incurred during the communication.
- STATUS – Gives various error information.

Figure 3.18 shows the configuration of the s7-1200 (TRCV C DATA and TSEND C DATA) blocks. The set-up is basically identical for both blocks. This configuration is done, when coding the data block into the PLC ladder program. The parameters (IP address and Port values) in the configuration are used for the entire TCP/IP Protocol between the PLC and the Matlab Simulink model [65].



Connection parameter

General

Local

End point: PLC_1

Interface: PLC_1, PROFINET interface_1[X1 : PN(LAN)]

Subnet: PN/IE_1

Address: 192.168.0.1

Connection type: TCP

Connection ID (dec): 1

Connection data: PLC_1_Receive_DB

○ Active connection establishment

Address details

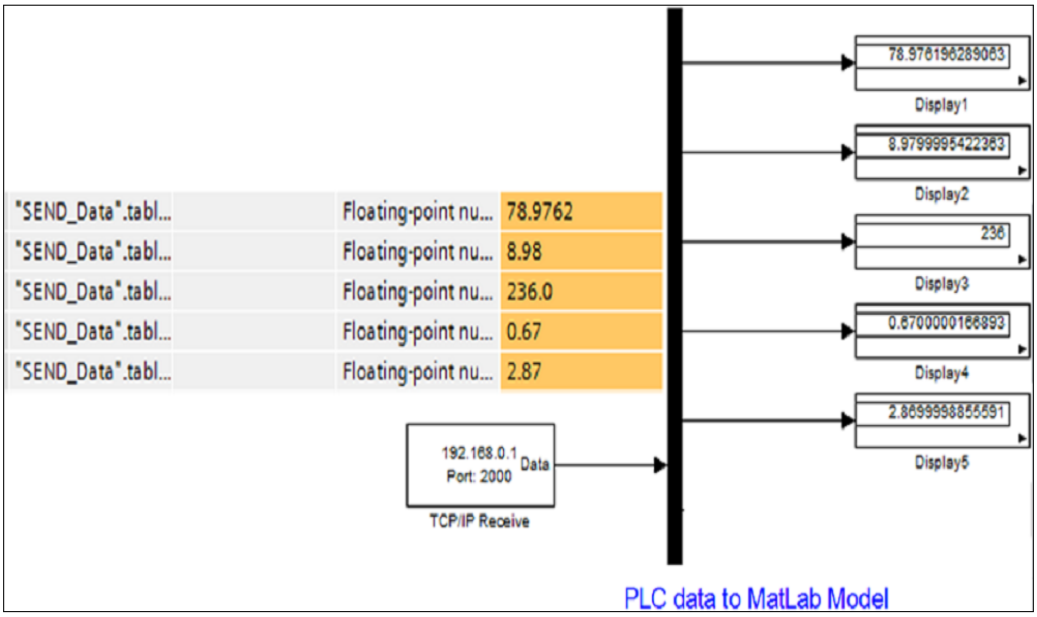
Local Port

Port (decimal): 2000

NAME OF THE PLC
CONNECTION METHOD BETWEEN PLC & PC
PLC IP ADRESS
REMOTE COMMUNICATION PROTOCOL
COMMUNICATION DATA BLOCK TYPE
PLC PORT

Figure 3.18: s7-1200 TCP/IP Protocol configuration of communication blocks

The communication implemented for the plant models and the PLC is then justified and tested to observe how the two work in real-time [65]. Refer to Chapter 4: **Real-time Response of Platform** section. Figure 3.19 shows an example of the TCP/IP protocol communication between the PLC and the model. The communication test shows that there is minimum deviation of the data between the two; and this communication method is feasible.



"SEND_Data".tabl...	Floating-point nu...	78.9762
"SEND_Data".tabl...	Floating-point nu...	8.98
"SEND_Data".tabl...	Floating-point nu...	236.0
"SEND_Data".tabl...	Floating-point nu...	0.67
"SEND_Data".tabl...	Floating-point nu...	2.87

Figure 3.19: S7-1200 -Matlab TCP/IP Communication Illustration

3.6 Controller Software Design

This section discusses the programmed controller software with blocks (ladder syntax), shown in Figure 3.20 for the platform. The software is programmed to carry out numerous tasks and to do calculations between plant models and SCADA.

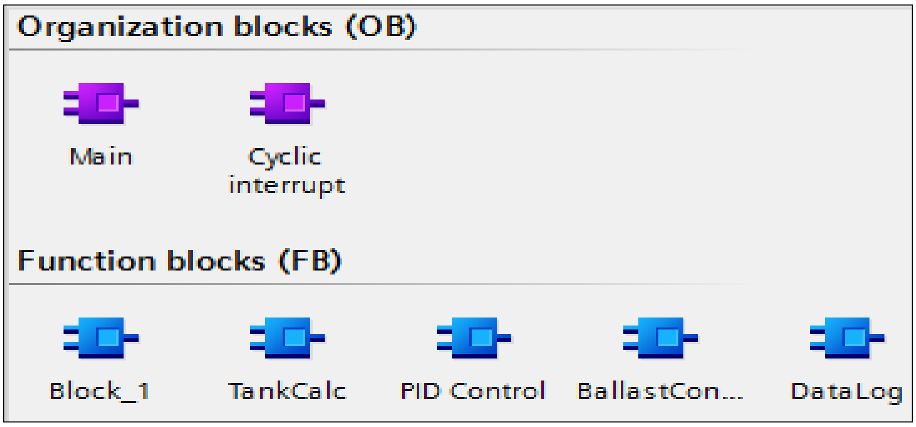


Figure 3.20: PLC program blocks

The controller software executes the following tasks on the PLC:

(i) Communication with plant models using TCP/IP protocol on a private network, which involves sending and receiving data. The code for this communication protocol is executed in the Main OB.

(ii) Calculations on plant models – There are different parameters that may be set by the user and that need to be scaled to meet the plant requirements; and for each model, these calculations are done in the function blocks:

- TankCalc (FB) – receives the Dual Tank Parameters from user (SCADA) and sends them to the plant model (i.e. tanks 1 and 2 (surface area, height), valves (1, 2 and 3) resistances and activation modes, PID parameters, set-point). The response of the plant model is received by the TankCalc block, which then passes on the status of the plant model to the SCADA.

- Ballast Control – receives the Ballast Tank Control model parameters from the user and sends them to the plant model (i.e. tank size, vessel mass, activation of ballast control, initial depth, depth set-point, cross-sectional area of vessel and PID parameters). The response of the plant model is read back into the block, such that the user can observe it on SCADA.

For the plant model function blocks, the parameters are checked in the program so that they meet the model requirements. If an input value is out of range, the user is informed that the value is wrong; and it automatically sets it to a default value, so that the plant model continues to operate.

(iii) PID Control (FB) – This is often used to give smooth and stable control for various actuators and devices in process-control systems; and the PID block is capable of performing such a task. The Compact PID block used for the plant models is a block that comes within the TIA software package; and on this platform, it is used to control the fluid level of tank 1 in the dual-tank model and the depth of the submarine in the ballast-tank control model. Users are to change the PID parameters via SCADA and observe the system's response. The model PID blocks are executed in the Cyclic interrupt (OB); so that the plant models are fine-tuned at constant rates for achieving the response control.

(iv) DataLog (FB) – This FB serves the purpose of saving the data of all the inputs and outputs at a chosen rate. The set-up and programming of this block is further described in Section 3.6.1: **Data Logging of activity on system**.

The programmed algorithms and set parameters for the plant models are all represented on the SCADA; and the SCADA operation is further discussed in Section 4.4: **Website and Process Control Interaction**. For the controller software, refer to **Appendix A : PLC Programs**.

3.6.1 Data Logging of activity on system

This section discusses the approach implemented to have the I/O parameters from the SCADA written onto a Data-Log file and made accessible to the user. The importance of this functionality of the system helps the user to analyze and see how the system behaves when they enter or control different parameters on the system. The PLC is programmed in such a manner, that the user can record and reset the data logging; and this is only possible when the SCADA system is available on the student page. Figure 3.21 illustrates the algorithm for the data logging procedure implemented on the PLC. Table 3.7 gives the steps for the user to download the file with I/O data recorded by the PLC during the practical session. The output file of the CSV recorded by the data logging procedure is shown in Figure 3.22.

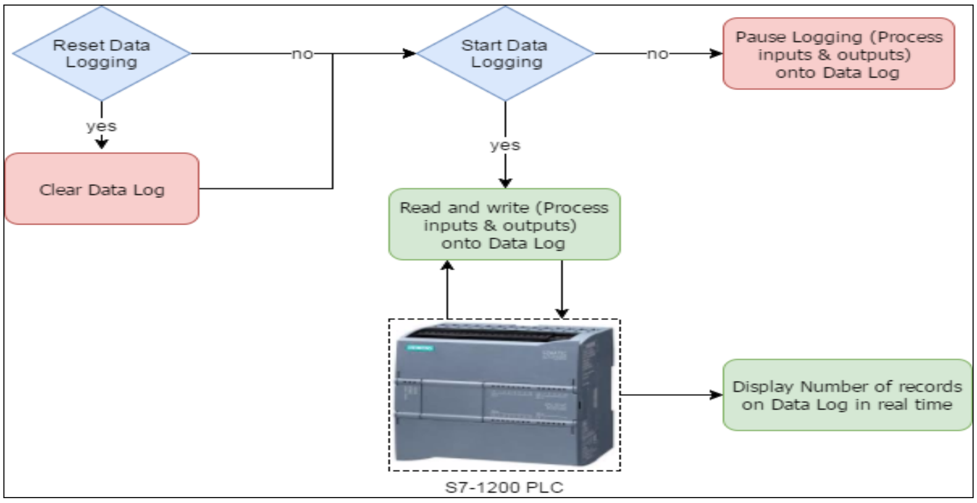
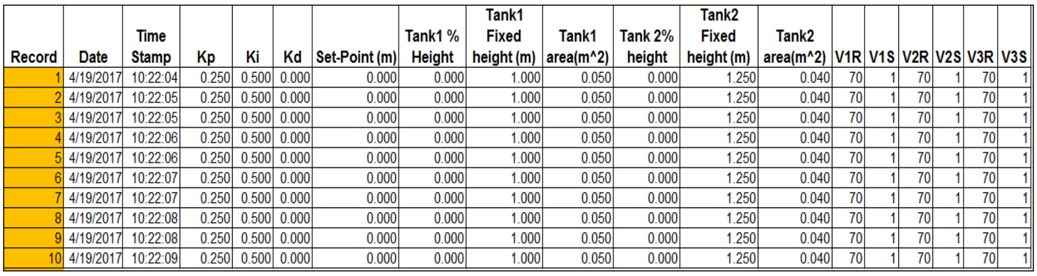

Reset Data Logging
no
Start Data Logging
no
Pause Logging (Process inputs & outputs) onto Data Log
yes
Clear Data Log
yes
Read and write (Process inputs & outputs) onto Data Log
Display Number of records on Data Log in real time
S7-1200 PLC

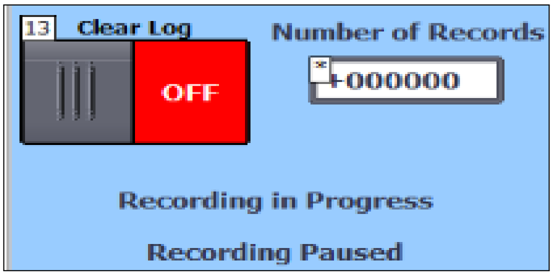
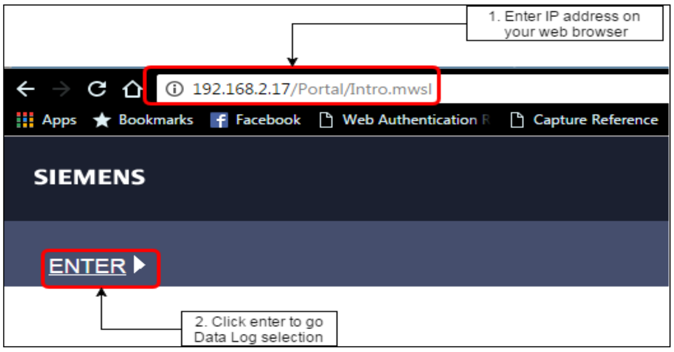
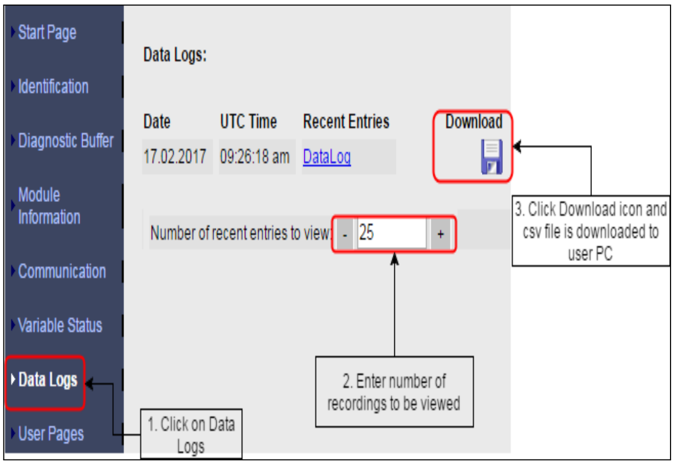
Figure 3.21: Flow diagram for the data logging



Record	Date	Time Stamp	Kp	Ki	Kd	Set-Point (m)	Tank1 % Height	Tank1 Fixed height (m)	Tank1 area(m^2)	Tank 2% height	Tank2 Fixed height (m)	Tank2 area(m^2)	V1R	V1S	V2R	V2S	V3R	V3S
1	4/19/2017	10:22:04	0.250	0.500	0.000	0.000	0.000	1.000	0.050	0.000	1.250	0.040	70	1	70	1	70	1
2	4/19/2017	10:22:05	0.250	0.500	0.000	0.000	0.000	1.000	0.050	0.000	1.250	0.040	70	1	70	1	70	1
3	4/19/2017	10:22:05	0.250	0.500	0.000	0.000	0.000	1.000	0.050	0.000	1.250	0.040	70	1	70	1	70	1
4	4/19/2017	10:22:06	0.250	0.500	0.000	0.000	0.000	1.000	0.050	0.000	1.250	0.040	70	1	70	1	70	1
5	4/19/2017	10:22:06	0.250	0.500	0.000	0.000	0.000	1.000	0.050	0.000	1.250	0.040	70	1	70	1	70	1
6	4/19/2017	10:22:07	0.250	0.500	0.000	0.000	0.000	1.000	0.050	0.000	1.250	0.040	70	1	70	1	70	1
7	4/19/2017	10:22:07	0.250	0.500	0.000	0.000	0.000	1.000	0.050	0.000	1.250	0.040	70	1	70	1	70	1
8	4/19/2017	10:22:08	0.250	0.500	0.000	0.000	0.000	1.000	0.050	0.000	1.250	0.040	70	1	70	1	70	1
9	4/19/2017	10:22:08	0.250	0.500	0.000	0.000	0.000	1.000	0.050	0.000	1.250	0.040	70	1	70	1	70	1
10	4/19/2017	10:22:09	0.250	0.500	0.000	0.000	0.000	1.000	0.050	0.000	1.250	0.040	70	1	70	1	70	1

Figure 3.22: CSV file output from the data logging of dual-tank model

Table 3.7: Data Logging for user

Directions to follow	Implementation Illustration
To record/pause, clear or reset the data log. The instance a user selects a practical on the SCADA homepage the Data Log is reset. This is to ensure previous values may not be copied by the current user. For a user to manipulate data logging, they may select the real plant tab or graphical view tab.	 13 Clear Log OFF Number of Records +000000 Recording in Progress Recording Paused
To retrieve the Data Log (CSV file), the user must enter the IP address (192.168.2.17) on the same web browser. NB! The user cannot login onto the PLC to change tags or the running program. The administrator is capable of accessing the PLC program.	 1. Enter IP address on your web browser 192.168.2.17/Portal/Intro.mwsl Apps Bookmarks Facebook Web Authentication R Capture Reference SIEMENS ENTER 2. Click enter to go Data Log selection
Upon clicking the Enter tab on the Siemens remote server, the user must click on the Data Logs tab; and then the file will be visible on this page. The user is advised to enter at least double the number of entries indicated on the SCADA, in order to obtain all the information. Having the entries entered, the user must click the download button; and a CSV file is downloaded directly onto user PC. A CSV file can be opened in Microsoft Excel, Matlab or Notepad. Hence, users can group the data, tabulate and graph it in different software packages.	 Start Page Identification Diagnostic Buffer Module Information Communication Variable Status Data Logs User Pages Data Logs: Date: 17.02.2017 UTC Time: 09:26:18 am Recent Entries: DataLog Download Number of recent entries to view: - 25 + 1. Click on Data Logs 2. Enter number of recordings to be viewed 3. Click Download icon and csv file is downloaded to user PC

3.7 Remote Accessing Website Design

After programming the PLC software and designing the SCADA interface, the platform website was then designed and set up, as discussed in this section. The website designed for this platform was one of the key requirements to define and give full functionality to the developed process-control system. The key elements defined for the website included:

- A managed website, whereby the administrator has full access and can manage the website (user details, activity on website and manage the process control system).
- Security and privacy – each user may access only their information, if registered on the website.
- An open communication platform – any website visitor (registered or non-registered) must be able to inquire the information they need that is related to the use of the platform.
- Each user is to access the SCADA at an allocated time.

3.7.1 Website Functionalities and Structure

The structure of the designed website (Figure 3.23) and the pages are fully described in this section. On the developed website, there are the following pages:

(i) Home – the Home page incorporates a navigation bar, whereby the website user can select a desired tab.

(ii) Membership – The membership tab incorporates three sub-tabs, as shown in Figure 3.24. These are:

  - Register – this allows the user to input their credentials in the outlined fields and upon successful registration, the user and two platform administrators

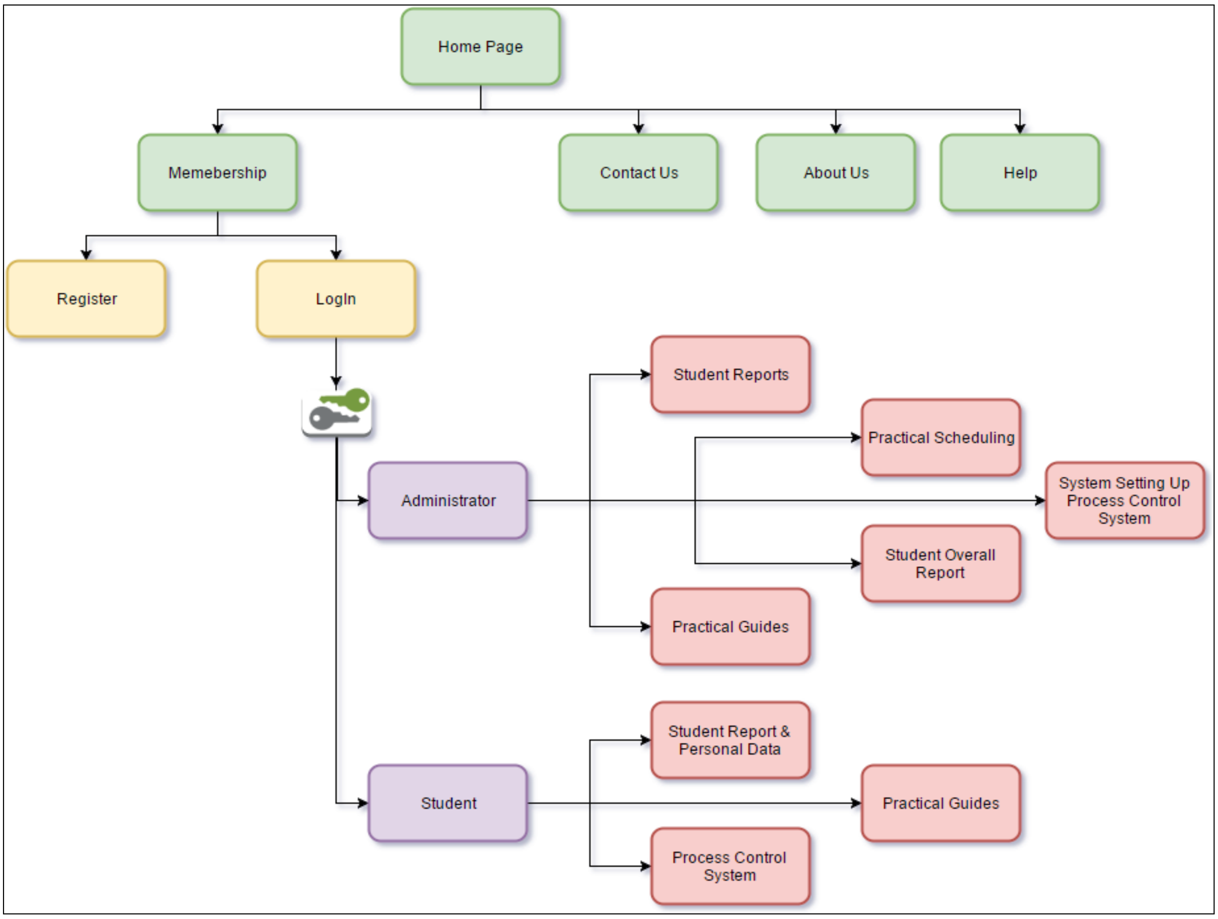


Figure 3.23: Remote Website Structure

receive an email notification to verify that they are now registered on the website. The outcome of a successful registration is illustrated in Figure 3.25.

- Login for Administrator – upon clicking this tab, the user is directed to a login page; where they input their username and password to be granted access to the administrator page.
- Login for Student – upon clicking this tab, the user is directed to a login page; where they enter their student number and password, in order to be granted access to the student page.

(iii) Contact Us – For any queries or agendas related to the platform, any website user is able to fill in a contact form that is submitted and sent to the platform

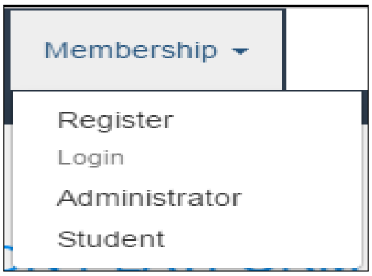
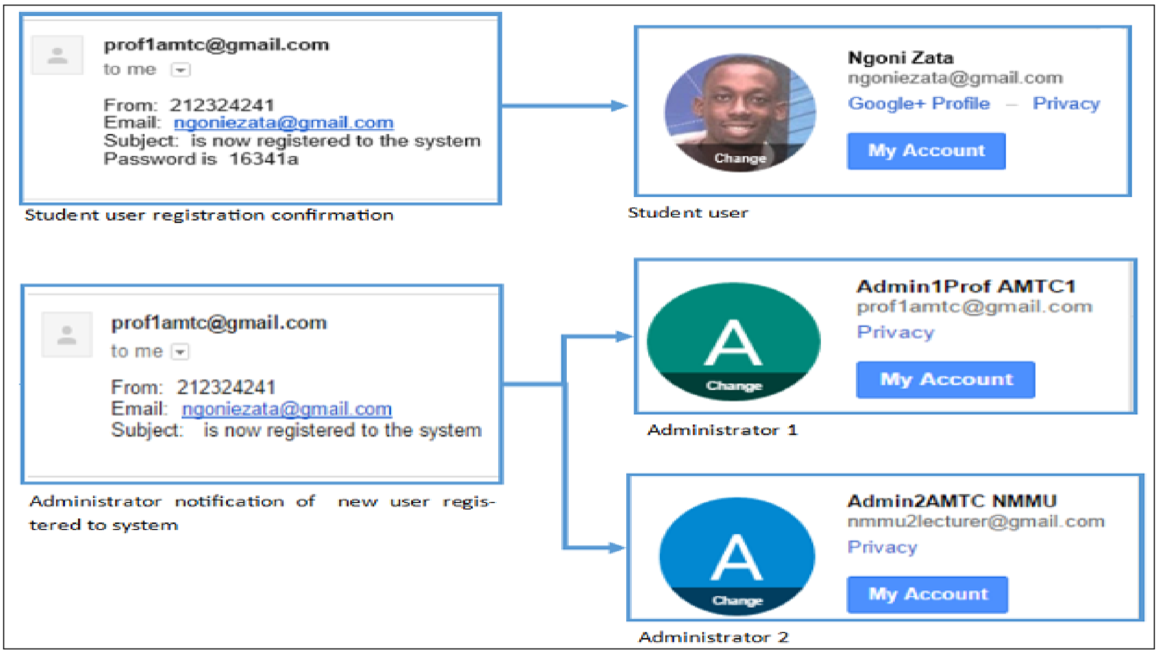


Figure 3.24: Membership tab



prof1amtc@gmail.com
to me

From: 212324241
Email: ngoniezata@gmail.com
Subject: is now registered to the system
Password is 16341a

Student user registration confirmation

Ngoni Zata
ngoniezata@gmail.com

Student user

prof1amtc@gmail.com
to me

From: 212324241
Email: ngoniezata@gmail.com
Subject: is now registered to the system

Administrator notification of new user registered to system

Admin1Prof AMTC1
prof1amtc@gmail.com

Administrator 1

Admin2AMTC NMMU
nmmu2lecturer@gmail.com

Administrator 2

Figure 3.25: Successful Registration on Website

administrators; and they are able to attend to the user via email.

(iv) About Us – The page simply shows the information with regard to the location where this research was implemented, as well as the team that was involved in bringing this research to fruition.

(v) Help – The page is of assistance to any user to carry out certain tasks to give clarity and guidance for the following:

How to register onto the system; How to Log onto the system; How to upload a document; Which tab is active when logged on to the student page.

3.7.2 Data Flow on Website

The website dataflow and functionality are defined as the transfer and execution of information, when the user is carrying out a specific task on the website. On a typical website, there are certain procedures and algorithms that are implemented, in order to keep both the site administrator and the user updated at all times, while keeping the relevant information shared and secure at all times [70, 71]. Figure 3.26 illustrates this concept of dataflow and functionality on the website. On the platform website, the two pages with the most dataflow and functionality are: the user page (student page) and the administrator page. The website code and calculating algorithms are those functions, which carry out the reading, writing and transferring of data to the database; and they give the results to the website user. The algorithms that have been programmed for the website pages include:

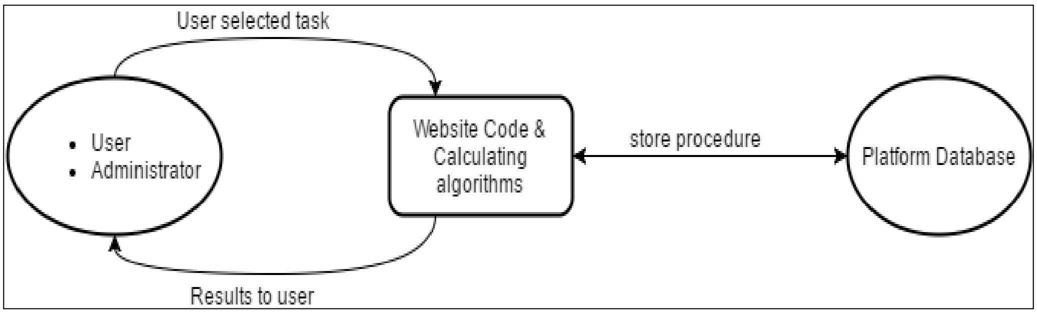


Figure 3.26: Data-Flow on Website

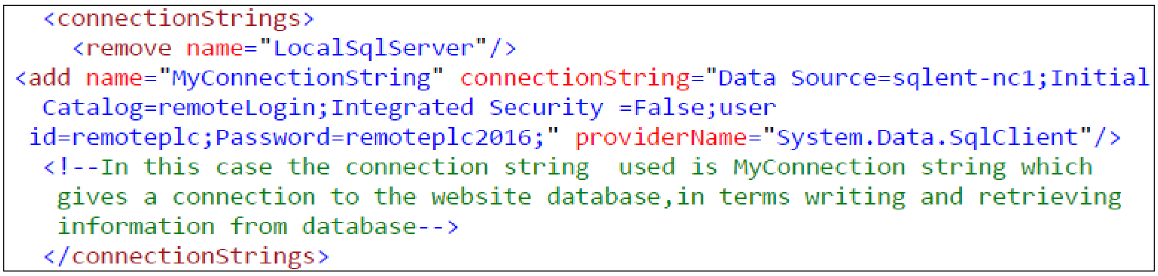
(i) Administrator page

- Login/ Logout
- Mark updating
- Practical guide uploading
- Practical session for scheduling
- Student activity on website (registration date and last login date)

(ii) Student page – Login/Logout, Student Documentation Uploading, Accessing SCADA system.

For data storage and manipulation on the website, store procedures are implemented. A store procedure is a form of syntax/method that can be implemented in either the SQL data base or C# in Visual Studio. The store procedure serves three main purposes, which are: the data storage; update and retrieval. For a store procedure to access a particular set of information in the database of the platform website, it requires the following [70]:

- Connection string shown in Figure 3.27 – that consists of Database Source, Database name and user credentials (id and password)



```
  <connectionStrings>
    <remove name="LocalSqlServer"/>
<add name="MyConnectionString" connectionString="Data Source=sqlent-nc1;Initial
  Catalog=remoteLogin;Integrated Security =False;user
 id=remoteplc;Password=remoteplc2016;" providerName="System.Data.SqlClient"/>
  <!--In this case the connection string  used is MyConnection string which
   gives a connection to the website database,in terms writing and retrieving
   information from database-->
  </connectionStrings>
```

Figure 3.27: C# Code snippet for connection string

- Table name and attributes tabulated in Website-Critical Functions – the different columns that store the information of different data types (e.g. strings, integers, date time).

For the Administrator, Registration and Student pages, the outlined activities are summarised in section 3.7.2.1 **Website-Critical Functions** table, together with the store procedures used for manipulating the different attributes. Upon the completion of the website development, it was then integrated with the SCADA; and this is discussed in section 3.8.

3.7.2.1 Website-Critical Functions

Activity	Description	C# Function	Store Procedures involved
Student Login	Student number is the Username and Password are used in the Store Procedure	Compares values from database with user inputs upon login	Select * from Users where Username='" + txtusername.Text + "' and Password ='" + txtpassword.Text
Mark updating	On the administrator page whereby administrator can update each student's marks for Practical 1 , Practical 2 and Final Mark	all students registered on the system are loaded with their student numbers with mark fields	"update adminisd set StudentName='" + StudentName.Text + "', PRAC1 ='" + PRAC1.Text + "',PRAC2 ='" + PRAC2.Text + "',Fmark ='" + Fmark.Text + "' where studentNo=" + studentNo
Practical guide uploading	On the administrator page: Upon the assignment of practicals for students the administrator may upload the documents	All documents already uploaded are loaded and a dialog box for uploading is available	"INSERT INTO SaveDoc1(DocName, Type,DocData,CreatedDate)" + " VALUES (@DocName,@Type, @DocData,@CreatedDate)";

Practical session scheduling	On the administrator page, all students registered on system are loaded and fields are made available for practical name ,date, start time and end time	A date picker that reads the system current calendar is activated and a time setter field is available. Prac name is available on a dropdown list. These functions have been implemented to ensure administrator does not enter incorrect values in the data fields.	"PracSet11"
			UPDATE [dbo].[adminisd]
			SET StudentNo = @studentNo, PracName=@PracName ,PracDate=@PracDate,StartTime=@startTime,EndTime=@ENDtIME
			WHERE StudentNo =@studentNo

Student activity on website (registration date and last login date)	This is implemented to give the administrator a clear view on the date and time student registered and logged onto the system	Simply loads the student list with their registration and last login dates and times	A simple griedview is loaded onto admin page whereby column values are linked to Users data table with selected fields via a connection string of the database
Registration	On the registration page ,user is to fill in the required fields	Upon completing the required fields, username and email are compared to values in database to ensure user does not exist, and password must be entered twice. Upon successful registration database is instantly updated and student can log onto the system. Administrators are notified that a new user is now registered and user receives their user name and password on their email address	Insert_User3 BEGIN INSERT INTO [Users] ([Username] ,[Password] ,[Email] ,[CreatedDate]) VALUES (@Username ,@Password ,@Email ,GETDATE()) iNSERT INTO adminisd(studentNo ,StudentName) values (SCOPE_IDENTITY() , @Username)
Student Documentation Uploading	On the student page: Upon the completion of a practical each student is to upload a document that will be evaluated by administrator	Store procedure implemented	"INSERT INTO StudentDocs(Student,DocName,Type,DocData, CreatedDate)" + " VALUES (@Student,@DocName, @Type,@DocData,@CreatedDate)";
Implementing process control task	On the student page, when the set session is ready for implementation ,the SCADA link is opened via an Iframe	IP address of SCADA is loaded by Iframe	N/A

3.8 Website Integration to Process-Control Platform

The process-control platform is made accessible to the user via SCADA with the architecture shown in Figure 3.28. The IP address of the PC running the SCADA application is loaded in an I-frame on the website, given that the student is logged on and a practical has been assigned (Practical Name, Practical Date, Practical Start Time and End time) have all been set by the administrator. A timing-session algorithm has been developed to ensure that all users have access to the platform at their allocated times, following what the administrators have set (Prac Name and Times).

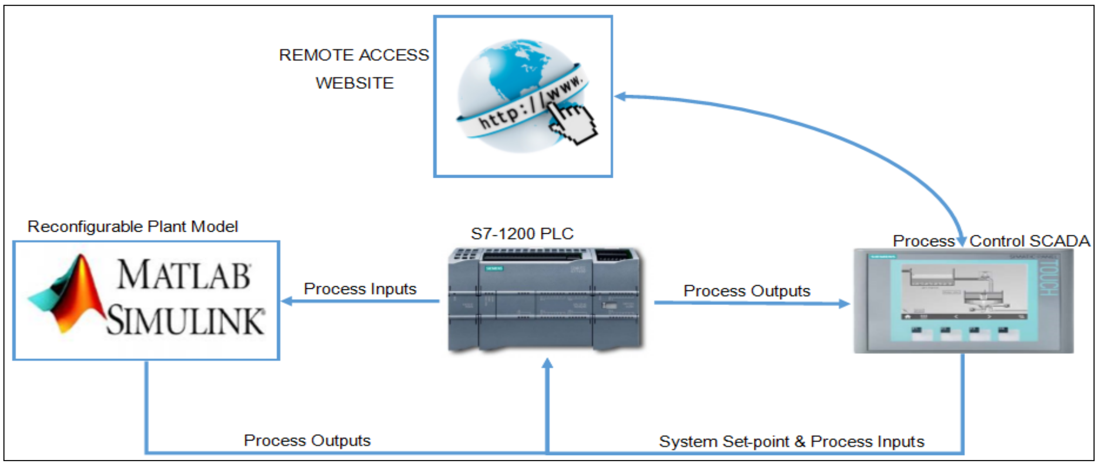
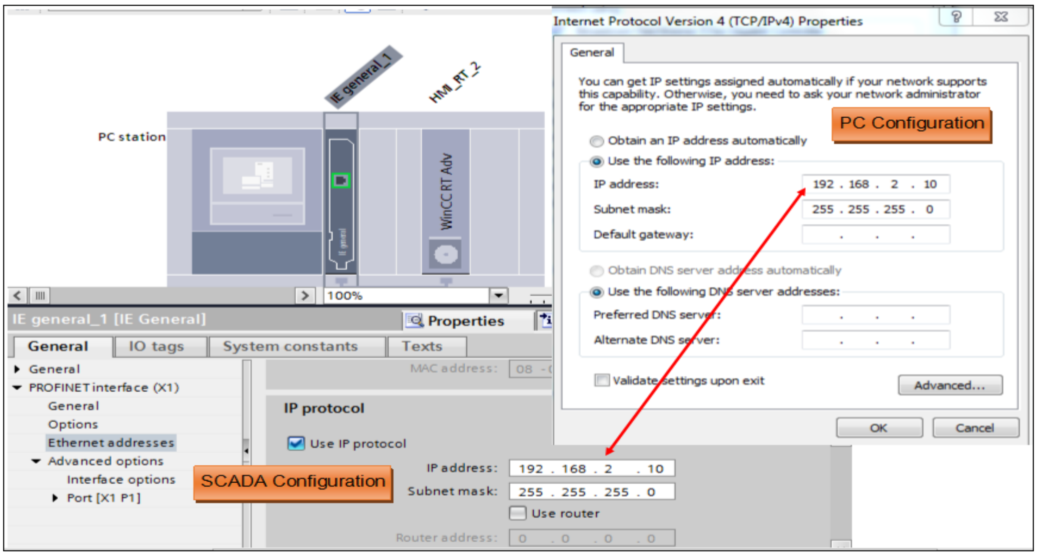


Figure 3.28: Architecture for accessing SCADA via Website

Figure 3.29 shows the configuration settings that have been implemented on the system (i.e. SCADA and platform server) for remote accessing the SCADA. The PLC runtime platform is on a private network that is connected via an Ethernet switch. It uses the IP of the PC, upon running the PLC runtime application. The website code snippet shown in Figure 3.30 for the Iframe loads the SCADA, when a user session is active. The timing-session algorithm has been implemented on the website; and it controls what is loaded on the Iframe. When a user logs into the website, the first step is to read the user's student number. The student number is then used in a store



You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

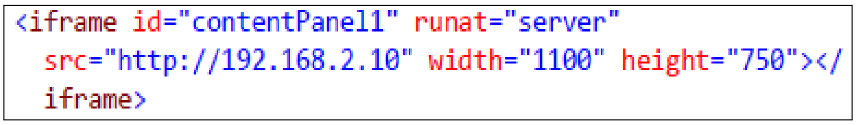
○ Obtain an IP address automatically
● Use the following IP address:
IP address: 192 . 168 . 2 . 10
Subnet mask: 255 . 255 . 255 . 0
Default gateway: . . .

○ Obtain DNS server address automatically
● Use the following DNS server addresses:
Preferred DNS server: . . .
Alternate DNS server: . . .

□ Validate settings upon exit

IP protocol
☑ Use IP protocol
IP address: 192 . 168 . 2 . 10
Subnet mask: 255 . 255 . 255 . 0
□ Use router
Router address: 0 . 0 . 0 . 0

Figure 3.29: IP Setup on PC running SCADA



```
<iframe id="contentPanel1" runat="server"
  src="http://192.168.2.10" width="1100" height="750"></
  iframe>
```

Figure 3.30: Iframe C# code snippet

procedure that will acquire the user-specific details, which comprise the Practical Name, Practical Date, Starting Time and End Time – from the SQL database. The Practical Date, Starting Time and End Time are then compared with the user's current system (date and time); so that the algorithm will be able to inform the user when their practical is available, or is overdue. When the practical is ready, the SCADA is made accessible to the user. The flowchart diagram Figure 3.31 shows how the website code executes step-by-step to give the SCADA access to the user. The operation and functionality of the SCADA integrated with the website is further discussed and shown in Section 4.4 **Website and Process-Control Interaction**.

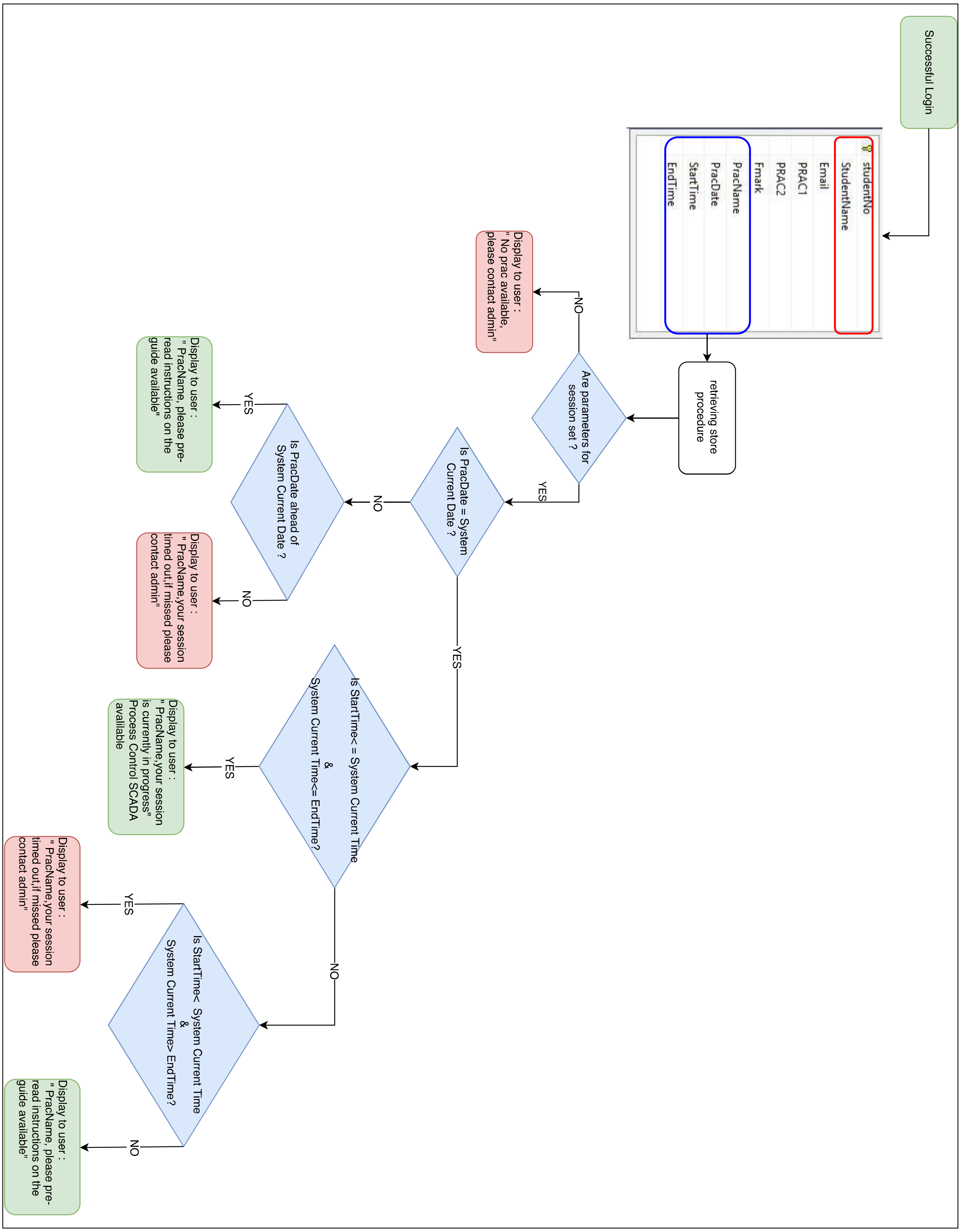

Successful Login
studentNo
StudentName
Email
PRAC1
PRAC2
Fmark
PracName
PracDate
StartTime
EndTime
retrieving store procedure
Are parameters for session set ?
NO
Display to user : " No prac available, please contact admin"
YES
Is PracDate = System Current Date ?
NO
Is PracDate ahead of System Current Date ?
YES
Display to user : " PracName, please pre-read instructions on the guide available"
NO
Display to user : " PracName,your session timed out,if missed please contact admin"
YES
Is StartTime< = System Current Time & System Current Time<= EndTime?
YES
Display to user : " PracName,your session is currently in progress" Process Control SCADA availiable
NO
Is StartTime< System Current Time & System Current Time> EndTime?
YES
Display to user : " PracName,your session timed out,if missed please contact admin"
NO
Display to user : " PracName, please pre-read instructions on the guide available"

Figure 3.31: Timing algorithm for SCADA access on website

3.8.1 Sources of Code

In this section, the different sources and the tools of the code are used to develop the website, as they are listed. The different attributes and purposes they served in designing the website include: file downloading and uploading, database management, navigation panels, website visual appearance, website access and integration.
Code Sources:

(i) https://github.com/twbs/bootstrap [Used For front end of the website].

(ii) http://asphelps.com/Topics/GridView.aspx [Used for deleting /adding/updating on grid views].

(iii) https://www.youtube.com/watch?v=y780MwhY70s [Used for data grid-view].

(iv) https://msdn.microsoft.com/en-us/library/ms972814.aspx

(v) http://stackoverflow.com/

(vi) http://www.codeproject.com/

(vii) Visual Studio 2015 Pro [Development language used]

(viii) http://www.w3schools.com/asp/

(ix) https://www.tutorialspoint.com/asp.net/

3.9 Platform Development Conclusion

After the entire platform was developed; it was then tested to verify that it gives all the functionalities. This was done so that the platform could be fine-tuned to meet the outlined objectives. The various tests that were implemented are discussed in Chapter 4.

Chapter 4

Evaluation and Experiments

For any mechatronic system that is set up to execute programmed instructions and data acquisition, there is always a need to investigate and observe its operation. This is done, in order to ensure that the system is fully functional to deliver the expected results; and it may be fine-tuned to reach the desired outputs. Hence, this entire chapter focuses on the critical tests and the data analysis that were carried out on the sub-systems that make up the developed platform and its overall functionality.

4.1 Real-time Response of Platform

The test was conducted to verify that the platform may be operated on the NMMU network domain; and the user gets real-time response, when working on the platform. The test is divided into two sections, whereby the communication between Simulink and PLC was done; and then a test for SCADA and the website response on the NMMU network follows.

4.1.1 PLC - Simulink Communication Test

The communication between the two devices/software interlinked together should be set up in such a manner, that there is minimal data distortion. However, the interlinked systems are to transfer signals and data in real-time, so as to have the entire system operate efficiently. This test was implemented, in order to observe and adjust the send/receive time value on the PLC that would give the overall system a real-time response from the simulated plant models. The test was implemented by generating a wave signal in Simulink that is sent to the PLC; and then sent back to the Simulink environment, as illustrated in Figure 4.1.

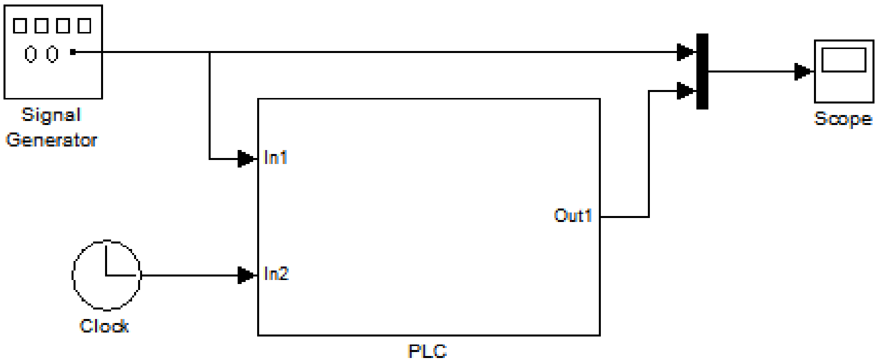


Figure 4.1: PLC-Simulink communication test procedure

Figure 4.2 is a flow diagram of the algorithm implemented to observe the difference between the Simulink time and the PLC recorded time for sending and receiving the generated waves. The simulation time is independent of the PLC recorded time; while the PLC send/receive time affects how responsive the PLC is to the Simulink model. Hence, this value had to be adjusted and monitored in such a way that the PLC gives a recorded time value close to the set simulation time in Simulink.

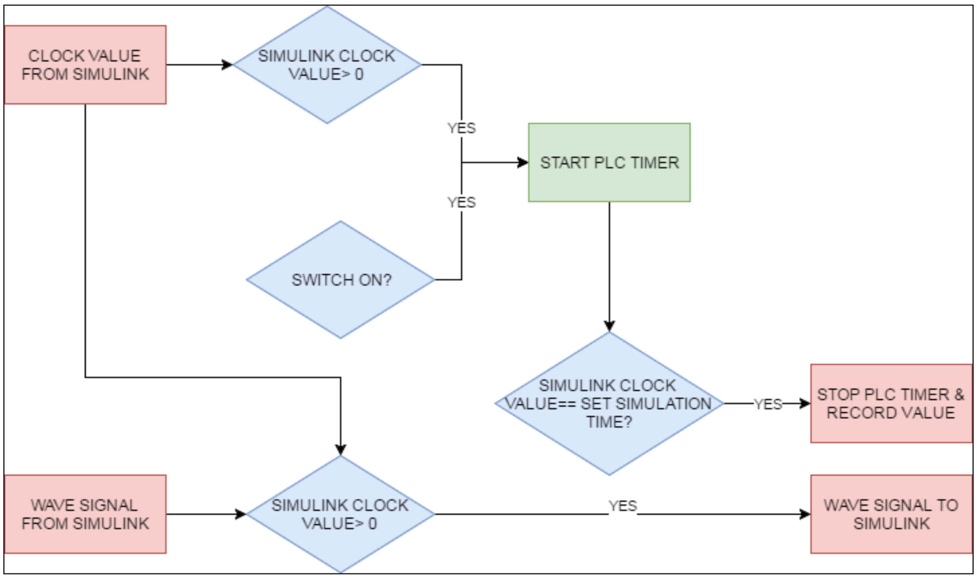

CLOCK VALUE FROM SIMULINK
SIMULINK CLOCK VALUE> 0
YES
START PLC TIMER
YES
SWITCH ON?
SIMULINK CLOCK VALUE== SET SIMULATION TIME?
YES
STOP PLC TIMER & RECORD VALUE
WAVE SIGNAL FROM SIMULINK
SIMULINK CLOCK VALUE> 0
YES
WAVE SIGNAL TO SIMULINK

Figure 4.2: PLC-Simulink real-time measuring algorithm

The signal peak values were observed; while on the PLC, the times for the sending and receiving of the signals were recorded, as shown in Figure 4.3. The same signal was also recorded on another channel, in order to compare it with the signal sent to and received from the PLC (Figure 4.1).

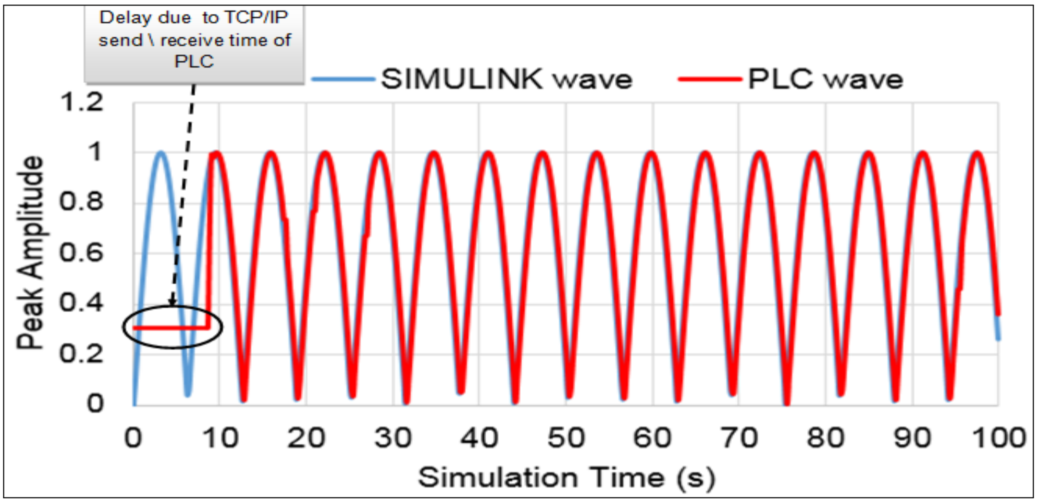
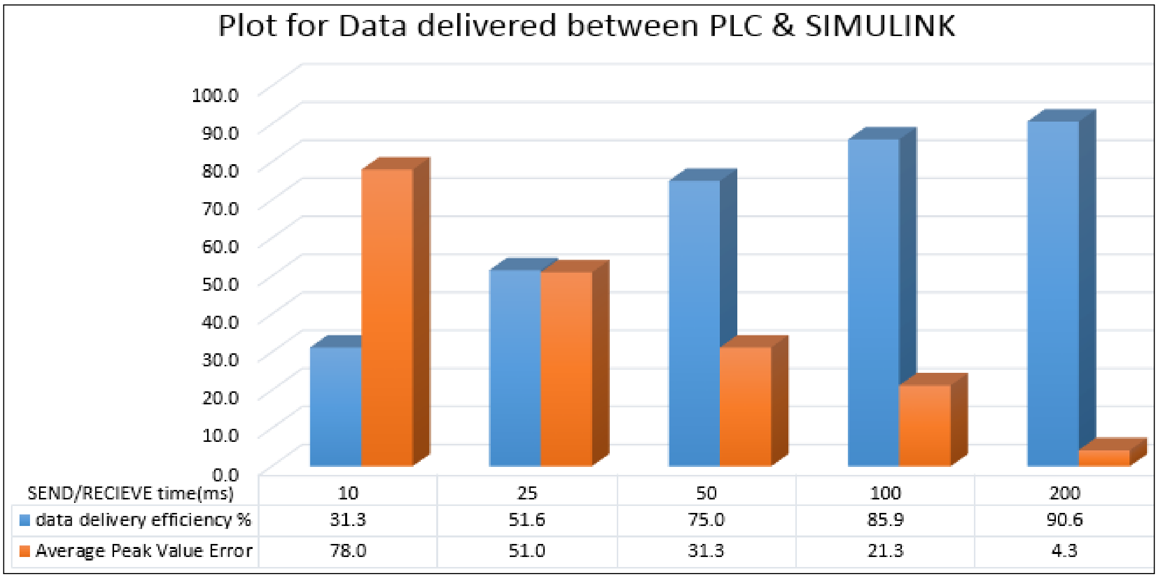


Figure 4.3: PLC-Simulink transferred signal waves

The peaks (as shown in Figure 4.3) from the PLC represent the amount of data sent back to Simulink by PLC. Hence, the efficiency of the data is also observed in this test, depending on the set send/receive value on PLC. From Figure 4.4, it was clear that send/receive value for PLC communication affects data distortion and the amount of data ($\frac{\text{PLC peaks}}{\text{Simulink peaks}}$).This observation denotes that as the PLC send/receive time is increased, the transfer of the generated signal from Simulink to PLC – and then back to the PLC, yields favourable communication responses. The average peak value error (Simulink to PLC to Simulink), as shown in Figure 4.4 reduces as the send/receive time is increased. The data delivery efficiency ($\frac{\text{PLC peaks}}{\text{Simulink peaks}} * 100$) also increased, as more signals (peaks) were transferred between PLC and Simulink during the test.

However, to ensure that the PLC and the Simulink environment operate in synchrony, the value that gave this requirement was 100ms. The values below 100ms showed that the PLC was lagging behind the Simulink model; while those above 100ms showed that the Simulink model was lagging behind the PLC time. The difference between the PLC and Simulink operation time with the variation of the send/receive time is shown in Figure 4.5. Hence the setup communication for the PLC and Simulink models, the PLC send/recieve value was set to 100ms.



SEND/RECIEVE time(ms)	10	25	50	100	200
🟦 data delivery efficiency %	31.3	51.6	75.0	85.9	90.6
🟧 Average Peak Value Error	78.0	51.0	31.3	21.3	4.3

Figure 4.4: PLC-Simulink transferred signal waves

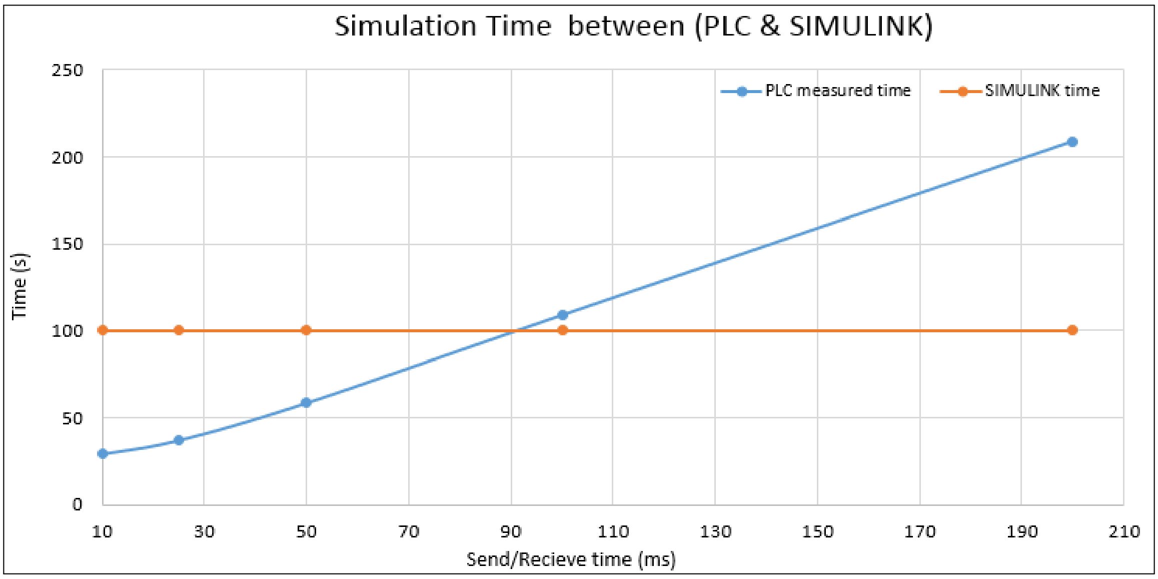
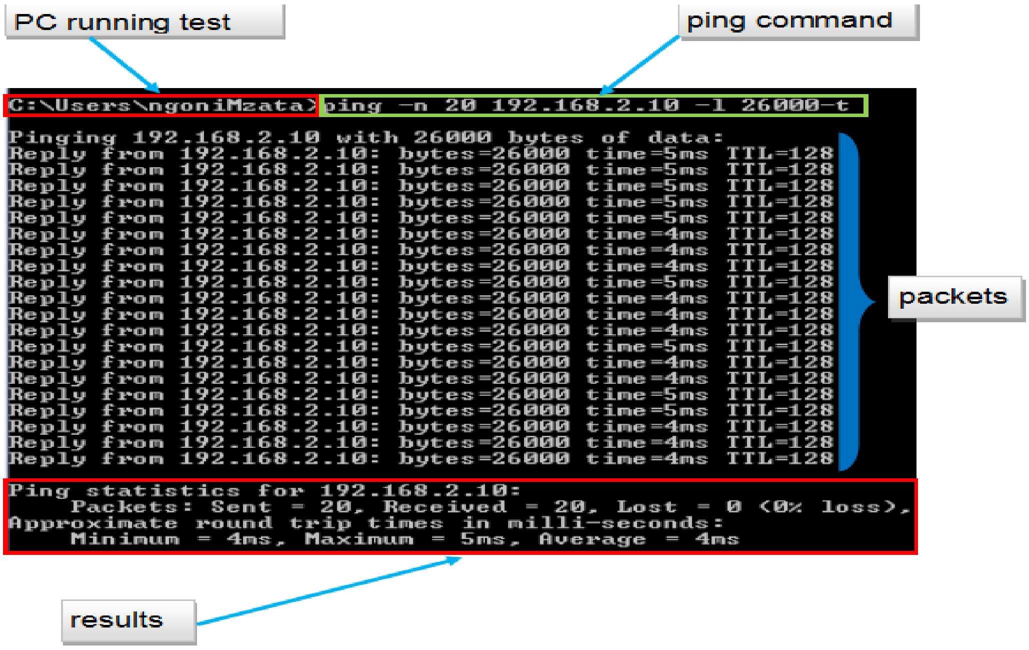


Figure 4.5: PLC-Simulink real-time response

4.1.2 SCADA and Website Response on NMMU network

This test was done at two of the most used and high traffic computer laboratories at the NMMU North and South Campuses. The test was to verify how the SCADA and the website receive data and send data on the network, in order to observe the different speeds at which the data are transferred. Ping testing is often used to check the speed of an internet connection. It works by using a similar principle of sensors that send a beam to an object and receive it back. Thus, the time taken for this procedure can be used to measure the speed of the network. The implementation of this test is carried out using the cmd.exe application that is in-built within the windows operating system software [78]. The variables shown in Figure 4.6 are described as follows:



```
C:\Users\ngoniMzata>ping -n 20 192.168.2.10 -l 26000-t

Pinging 192.168.2.10 with 26000 bytes of data:
Reply from 192.168.2.10: bytes=26000 time=5ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=5ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=5ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=5ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=5ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=4ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=4ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=4ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=5ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=4ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=4ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=4ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=5ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=4ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=4ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=5ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=5ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=4ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=4ms TTL=128
Reply from 192.168.2.10: bytes=26000 time=4ms TTL=128

Ping statistics for 192.168.2.10:
    Packets: Sent = 20, Received = 20, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 5ms, Average = 4ms
```

Figure 4.6: Ping Test Procedure Implemented

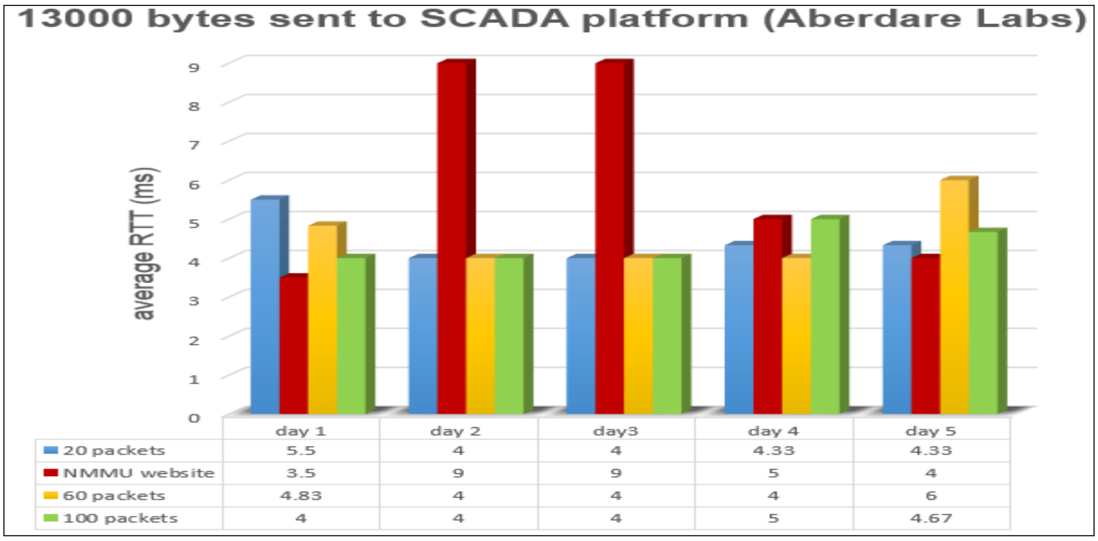
(i) Ping Command – The ping command is used to verify the IP-level connectivity. When this command is executed, the data are sent to the external machine and they bounce back to the source from which they were sent. In the command the number of packets are declared, the IP address of the remote device/website is set and the number of bytes(data size) can also be set.

(ii) Packets – These are termed as the amount of data that are sent over a digital network. When the data are being transmitted between two or more devices on the same network, they are sub-divided into packets, and then rebuilt at the final destination point.

(iii) Results – When the ping test is complete, an auto-generated report giving the (shortest, longest and average) ping time, the packets that were sent from the machine and received by the machine, are displayed, as shown in Figure 4.6.

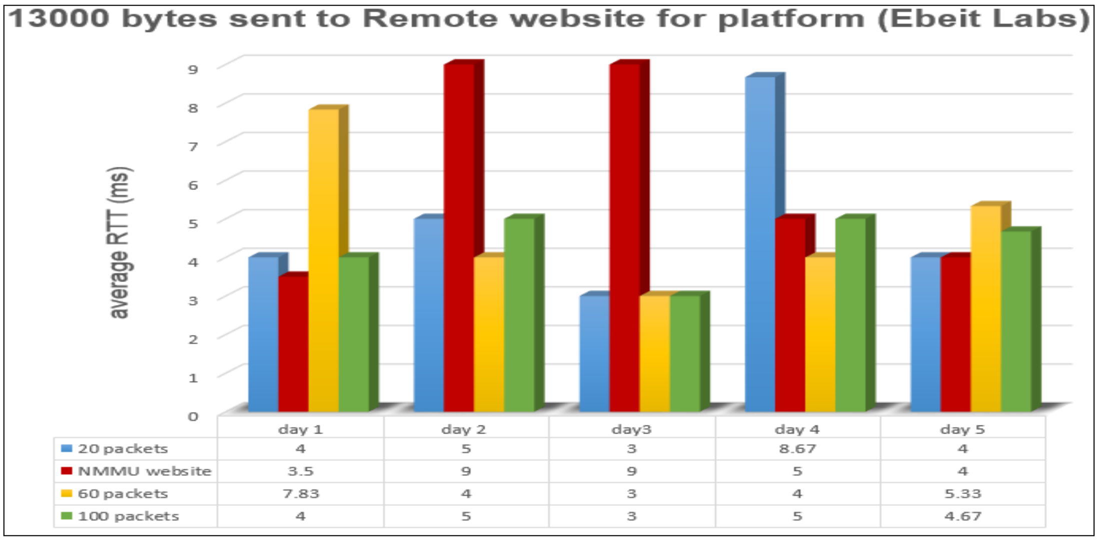
The tests were done over 5 days, during the week (Monday to Friday) when the students are on campus working in the laboratories. This was to observe the response under a lot of traffic on the network. The variables that were altered during the test were the number of packets and the bytes sent. However, the tests are divided into two sub-tests, which were done by:

- Changing the number of packets; while keeping the amount of data sent constant
- Changing the amount of data sent; while keeping the number of packets constant



	day 1	day 2	day3	day 4	day 5
🟦 20 packets	5.5	4	4	4.33	4.33
🟥 NMMU website	3.5	9	9	5	4
🟨 60 packets	4.83	4	4	4	6
🟩 100 packets	4	4	4	5	4.67

Figure 4.7: Packet Variation for Ping test in Aberdare Labs



	day 1	day 2	day3	day 4	day 5
🟦 20 packets	4	5	3	8.67	4
🟥 NMMU website	3.5	9	9	5	4
🟨 60 packets	7.83	4	3	4	5.33
🟩 100 packets	4	5	3	5	4.67

Figure 4.8: Packet Variation for Ping test in Ebeit Labs

From Figures(4.7 and 4.8), the variation of packets clearly did not affect the rate of transfer of data Round Trip Time (RTT). Hence for the platform, the main focus that followed was on the amount of data sent to the remote website and SCADA.

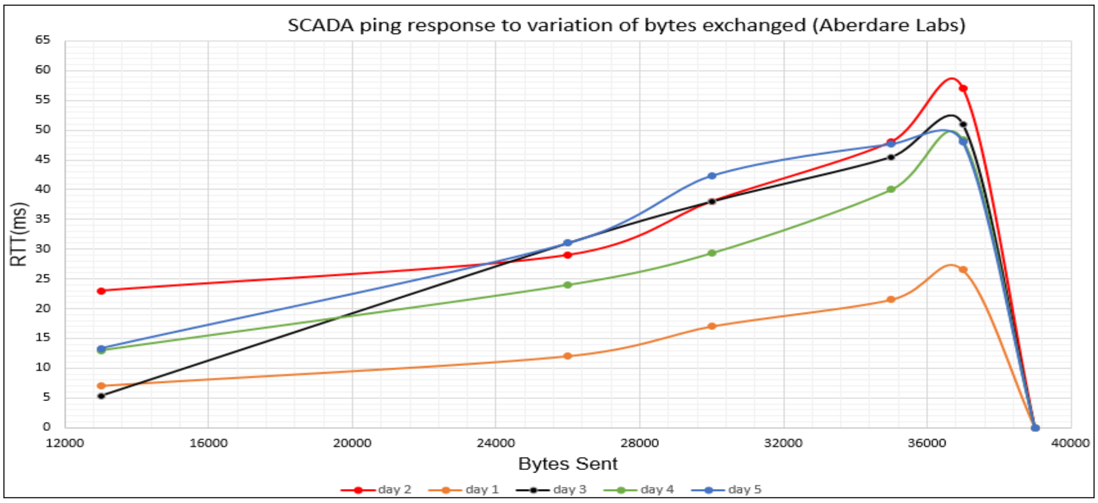


Figure 4.9: Bytes Variation for Ping test in Aberdare Labs

Looking at the graphs in Figures 4.9 and 4.10; there is a clear relationship between the increase in the number of bytes sent to the SCADA system from remote locations and the RTT, which also increases. However, there was found to be a limit to the number of bytes the SCADA can take and respond to. The bytes were within the range of (36000-

39000) bytes. To observe the actual throughput from the measured and recorded RTT

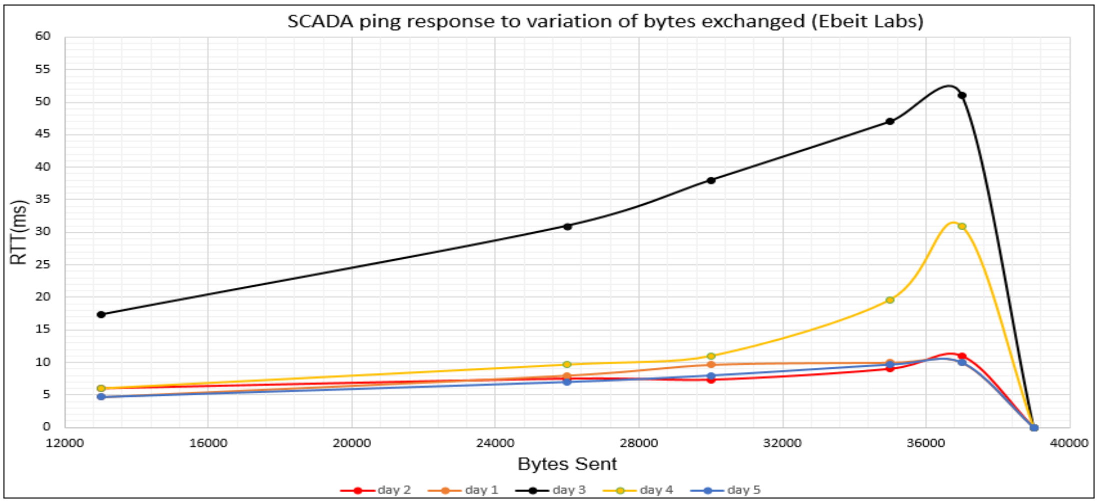


Figure 4.10: Bytes Variation for Ping test in Ebeit Labs

values from (Appendix: B) for the SCADA and remote website, the calculations were implemented using Equation 4.1.

$$\text{Throughput(Mbps)} = (\frac{\text{window size}}{\text{RTT}}) = (\frac{\text{bytes * 8}}{\text{round trip time}}) \tag{4.1}$$

Throughput is defined as the measure of the amount of data transferred successfully between two machines on the same network in a given time period, given as megabits per second (Mbps). Throughput is indirectly proportional to RTT; since the higher the RTT value, the lower the throughput of the network. This is calculated by using Equation 4.1 [78].

Taking a sample calculation with a value from graph shown in Figure 4.9, which is for day 4 with 28000 bytes sent in the ping and RTT approximated to 25 ms:

$$\text{Throughput(Mbps)} = \left(\frac{28000 * 8}{25 * 10^{-3}}\right) = 8.96 \text{ Mbps} \tag{4.2}$$

The overall throughput values observed from the test implemented are tabulated in Table 4.1. These values are ranging from 5 Mbps to 35 Mbps. Having the remote website for the platform running from an NMMU server, the throughput was observed; so that it could be used as a comparison to the internet response of (SCADA and PLC) that run from the platform server. The average throughput of the developed website was found to be 16.09 Mbps; while that of the SCADA was 15.20 Mbps. Thus, from this particular test with the observed throughput values, it is clear that the platform meets the following elements:

- Operation in real-time and the concept of remote accessing via the internet works for the developed platform.
- The merging of the website and SCADA is possible and it works.
- The selection of NMMU network is reliable for the platform to be remotely operated.

Table 4.1: Overall Throughput of SCADA and Remote Website

	Throughput (Mbps)				
	Day 1	**Day 2**	**Day 3**	**Day 4**	**Day 5**
Aberdare Labs					
SCADA	14.10	5.81	8.90	7.59	6.44
Website	15.00	21.81	10.30	11.22	7.10
Ebeit Labs					
SCADA	26.82	10.36	13.93	24.82	33.20
Website	21.69	20.45	8.26	23.45	21.66

4.2 Ballast-Tank Control Model Results

The results discussed in the section were the experiments implemented to illustrate the depth control of a submarine to the users. This plant model shows that the developed platform can be reconfigured; since it is of a different nature to that of the dual-tank model. The ballast-tank control model is applicable in the Marine Engineering field. A summary of what the graphs plotted represent for this model, as well as the numerous concepts/activities that can be taught, based on this model, are discussed.

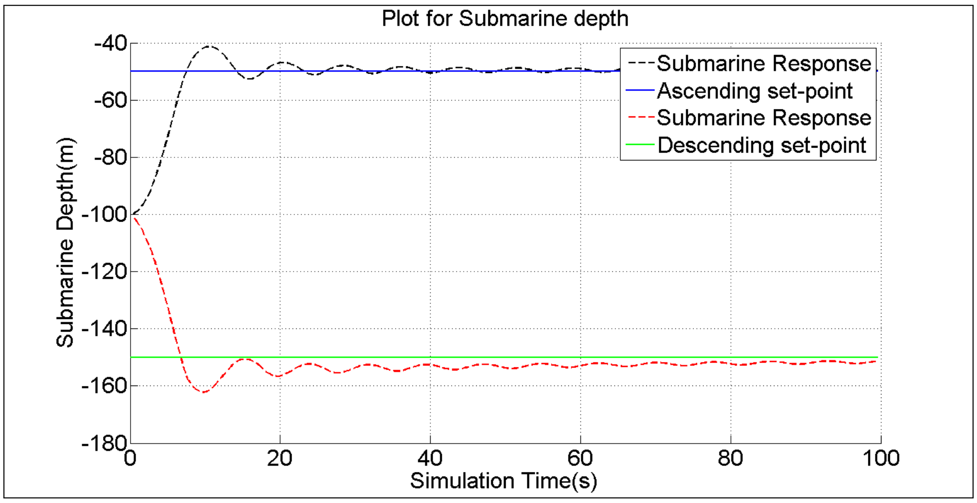


Figure 4.11: Submarine Depth Response from Model

Figure 4.11 shows the depth set-point; when the submarine is ascending and descending, whereby, the submarine is initially 100m below the surface. This graph gives the user a basic illustration of what really takes place when a submarine moves at different depths. Figure 4.12 depicts the error values between the depth set-point and the actual response of the model. Such a graph gives the users feedback on whether they are implementing the practical in the right manner, or not. Concepts like statistical analysis can be implemented by the users, whereby they are to repeat the same set-point to be reached more than once; and they can observe the effect of changing certain parameters.

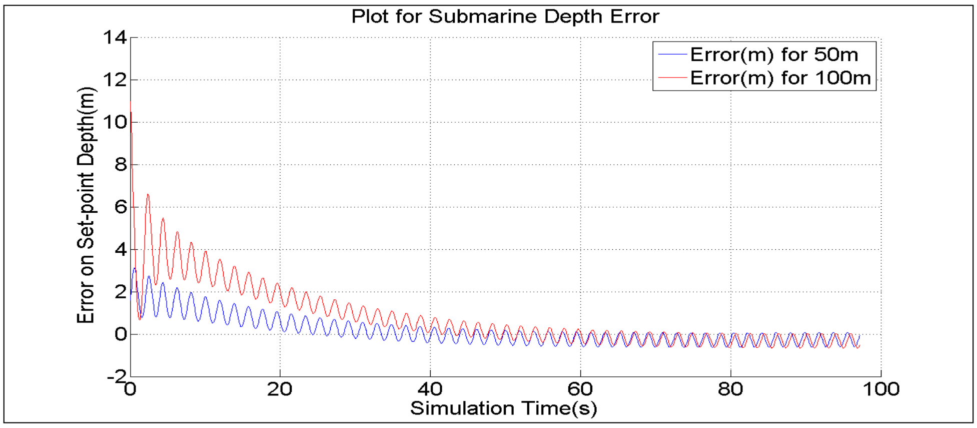


Figure 4.12: Error of submarine depth from set-point

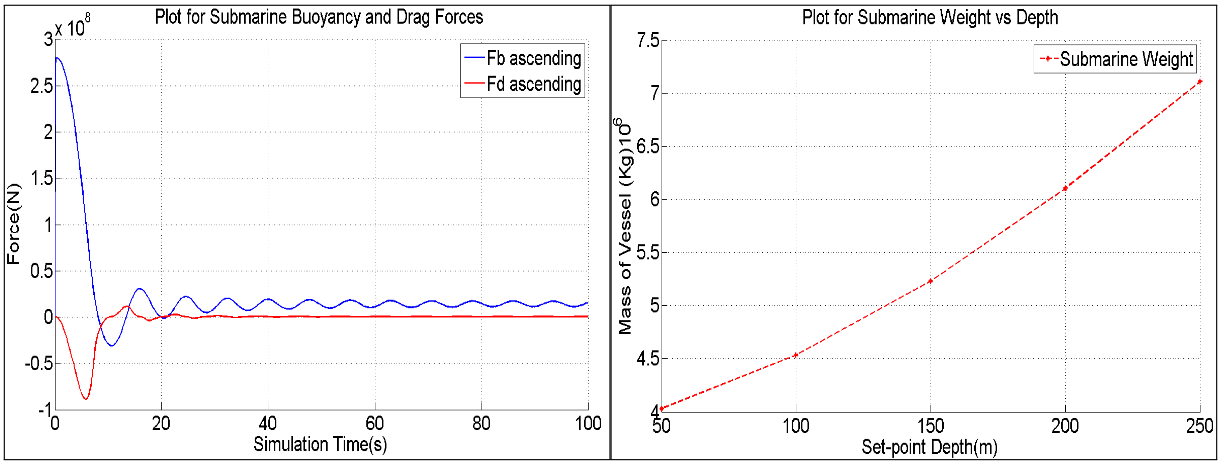


Figure 4.13: Force and Weight variations of Submarine due to Ballast Control

For any body in motion; there are always forces acting in the same and/or the opposite direction of the motion. The graphs in Figure 4.13 give an illustration of what a submarine comes across when ascending, whereby the buoyancy force is always lifting the vessel and the drag force opposing the direction of motion. As for the weight with depth, the two are directly proportional to each other. When the submarine is descending, the overall weight of the vessel increases. This is done through pumping more water into the ballast tank. When the submarine is ascending water is pumped out of the tanks by opening the outlet valves of the tank; while pumping in high pressurized air. The SCADA operation of this plant model is further discussed in section 4.4.

4.3 Dual-Tank Model Results

The model is tested with different parameters, in order to observe its response, as well as to identify the various activities used to teach and evaluate users from the platform. The main concept and process-control knowledge that can be taught from this model is the use of PID control. The parameters that were varied during the evaluation of the model included the gain, the integral and the derivative.

4.3.1 Mathematical analysis on model

When implementing the practical on the model, the user must be capable to identify and derive Equations 3.4 and 3.6, given the schematic digram shown Figure 3.4. Compact PID parameter calculations is to be provided to the users, in order to identify parameters for the controller and understand the functionality thereof. The parameter Equations for the PID block are given as [65]:

$$\mathrm{y} = \mathrm{K_p}[(\mathrm{b.w - x}) + \frac{1}{\mathrm{T_1.s}}(\mathrm{w - x}) + \frac{\mathrm{T_D.s}}{\mathrm{aT_D.s + 1}}(\mathrm{c.w - x})] \tag{4.3}$$

$$\mathrm{K_I} = \frac{\mathrm{K_P}}{\mathrm{T_i}} \tag{4.4}$$

$$\mathrm{K_D} = \mathrm{K_P} * \mathrm{T_D} \tag{4.5}$$

The variables used in Equations 4.3, 4.4 and 4.5 are defined in Table 4.2

Table 4.2: Dual Tank Compact PID parameter definitions

y	Output value	x	Process value
w	Set-point value	s	Laplace operator
K_p	Proportional gain (P-component)	a	Derivative delay coefficient (D component)
T_I	Integral action time (I-component)	b	Proportional action weighting (P component)
T_D	Derivative action time (D-component)	c	Derivative action weighting (D component)

The PID control block runs in a cyclic interrupt; and this denotes that the plant model operation is in real-time; and it is periodically updated. Any changes of the plant model parameters that are done need to be monitored at a constant time interval, in order to keep the system functional. Figure 4.14 shows the PID block that receives the plant set-point and the plant-process value. Upon receiving these values, the PID block executes Equation 4.3, until the plant model reaches a value that is close to the set-point and it depends on the user parameters (i.e. $K_P; K_I; K_D$), which are entered on the SCADA by the user (Figure 7.8). An illustration of a typical test and operation of the plant model response is shown in Figure 4.15, whereby the user is to vary the set-point of plant model to three different states. From a plot of this nature, the user can be able to calculate various parameters and draw a conclusion from the system's response.

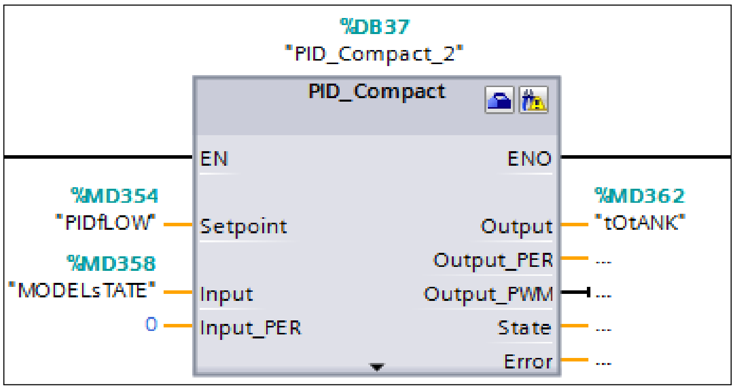


Figure 4.14: Compact PID block

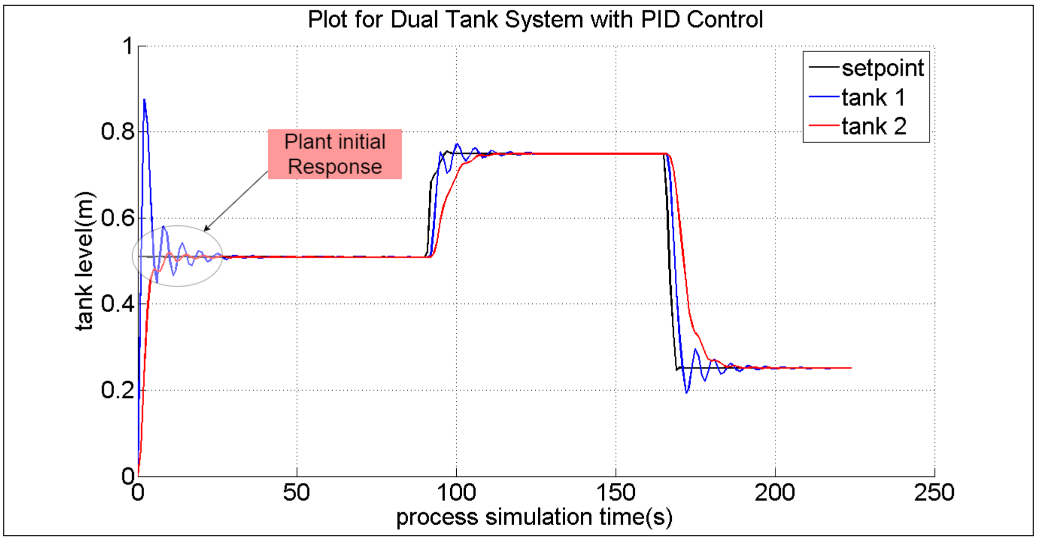


Figure 4.15: Dual-Tank operation illustration

The critical parameters to observe from such a plot include:

- The maximum overshoot – this is the point at which the model's response is at its highest, relative to the set-point.
- Rise time (t_r) – this is the time it takes for the plant model to reach a certain portion of the set-point.
- T_{max} – this is the time it takes for the plant model to reach the maximum overshoot.
- Settling time (t_s) – this is the time it takes for the plant model to reach a steady state, at which the process value has stabilized.

The critical parameters stated are identified, as shown in Figure 4.16, for plant's initial response. These assist the user to select or adjust the parameters, in order to fine-tune the system and reach the required outputs. The users will also gain knowledge and understand the intelligent control methods that are used in industry and their relevance from a theoretical perspective.

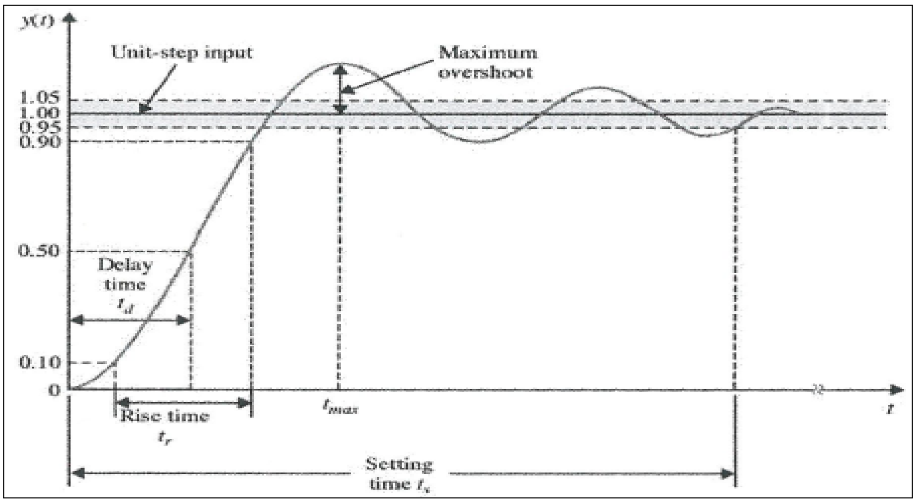


Figure 4.16: Identification of Response values from PID controlled system

4.3.2 Application and analysis of varied PID parameters

In Table 4.4, there are eight tests that were done and observed. From these tests, there are different responses and outcomes. It is clear that variation of (K_P, K_I, K_D) parameters has an impact on system's response and fluid flow through the two tanks. Depending on the task set out for the user-specific controller, the correct parameters should be used to meet the requirements. Examples of user applications are explained in the two scenarios below.

4.3.2.1 Scenario Cases on Dual Tank system

Case 1

A plant that mixes chemicals in different proportions/states and temperatures is to be implemented. Tank 1 receives the finished product; and it will discharge the finished product into tank 2, after some time for bottle filling or further mixing. Consequently, valve 2 is closed; while filling up tank 1. Such a scenario will need a quick response control and a low overshoot. From Table 4.4, samples 4 and 5 would have a PID controller that meets such requirements.

Case 2

In a scenario where the product is being filtered/processed in tank 1; and then stored in tank 2 after a lengthy period; tank 2 is to have an overshoot of less than 10% (0.05m) from the set-point; and a response of less than 30 seconds. For such a case, samples (5, 6, 8) from Table 4.4 would have a PID controller that meets such requirements.

4.3.3 Dual Tank overall response

The results in Table 4.4 show certain trends between the two tanks. The overshoot from set-point (Figure 4.17) of tank 1 is always higher than tank 2; because tank 1 inflow comes directly from the pump; and tank 1 feeds into tank 2. This observation is seen on the rise times in Figure 4.18, whereby tank 2 rise time lags behind tank 1. The settling times are similar; but tank 2 reaches the steady state before tank 1; and this is shown in Figure 4.19. These numerous responses justify that the model works similarly to how a real model would work; and it can be used in place of a real model. The SCADA operation of this plant model is discussed in section 4.4.

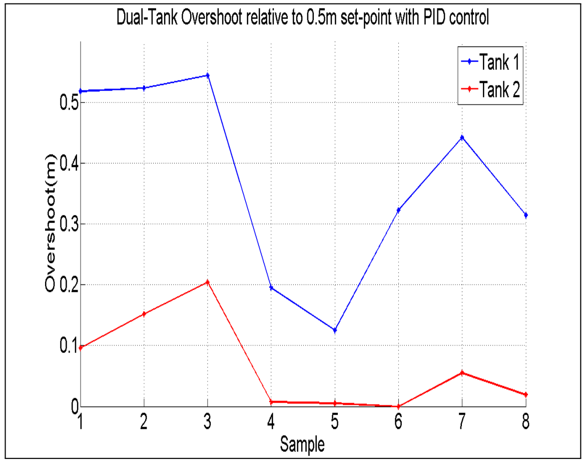


Figure 4.17: Overshoot of system using PID control

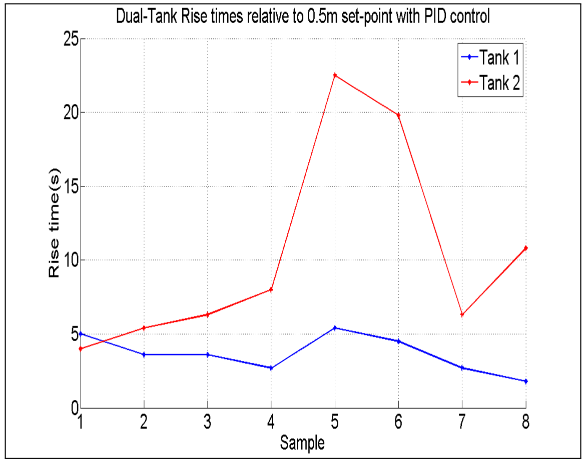


Figure 4.18: Rise-time of system using PID control

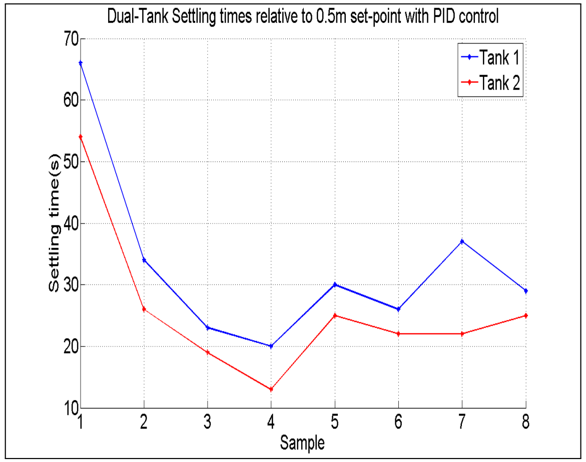


Figure 4.19: Settling-time of system using PID control

Table 4.3: Dual Tank Simulation Test Parameters

	Tank 1	Tank 2
Surface Area (m^2)	0.055	0.055
Fixed height (m)	1.125	1.125
Valves [1, 2, 3] resistance (ms^{-2})	60	
T1 - Tank 1	T2 - Tank 2	
T1 R-time (s)	Tank 1 rise time	
T2 R-time (s)	Tank 2 rise time	
T1 S-time (s)	Tank 1 settling time	
T2 S-time (s)	Tank 2 settling time	
Tn OS (m)	Tank n Overshoot (m)	n=1,2,..

Table 4.4: Dual Tank Simulation Test Results

Sample	Kp	Ki	Kd	Ti	Td	Set-point(m)	T1 OS(m)	T2 OS(m)	T1 R-time(s)	T2 R-time(s)	T1 S-time(s)	T2 S-time(s)
1	0.5	1	1	0.5	2	0.5	0.518	0.096	5	4	66	54
2	1	1	1	1	1	0.5	0.523	0.152	3.6	5.4	34	26
3	1.5	1	1	1.5	0.67	0.5	0.544	0.204	3.6	6.3	23	19
4	1	0.5	1	2	0.5	0.5	0.195	0.07	2.7	8	20	13
5	1	1.5	1	0.67	1.5	0.5	0.125	0.05	5.4	22.5	30	25
6	1	1	0.5	1	0.5	0.5	0.322	0	4.5	19.8	26	22
7	1	0.67	0	1.5	0	0.5	0.443	0.055	2.7	6.3	37	22
8	1.5	1	0	1.5	0	0.5	0.314	0.019	1.8	10.8	29	25

4.4 Website and Process-Control Interaction

This section discusses how the interfaced SCADA and website can be operated by any user. Figure 4.20 illustrates what the user will go through, when logged onto the website; and how the SCADA is accessed.

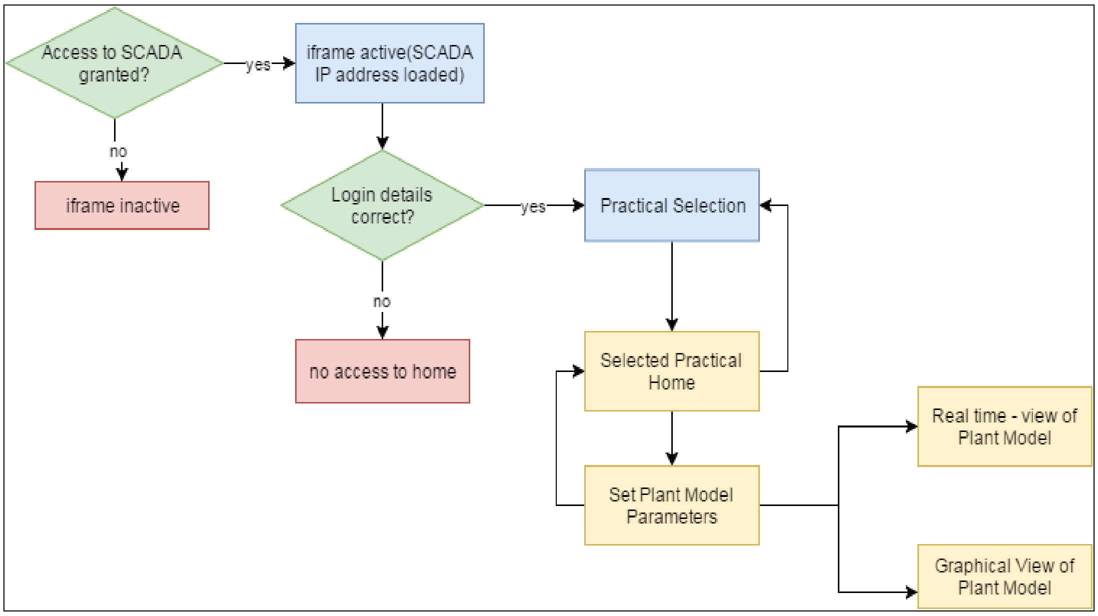
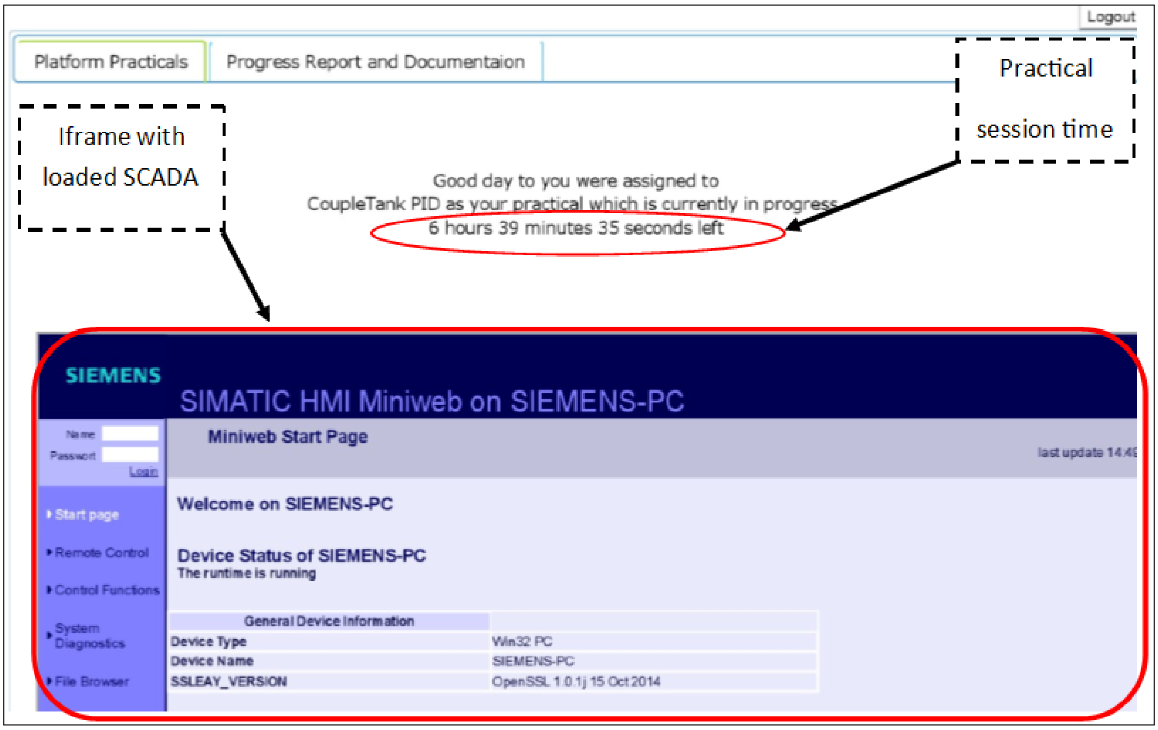


Figure 4.20: SCADA access to user flowchart diagram

If a practical session set by the administrator is running, a user is granted access to the SCADA application. The user is to enter his/her login credentials on the SCADA application, as shown in Figure 4.21. However, when the user successfully logs, on he/she may select a practical (Figure 4.22), according to what the administrator had initially assigned for them. The user would then be directed to the selected practical home page, which gives a brief introduction on how to carry out the practical; and it also gives them an idea of what they may face during the practical session. The user then sets the plant model parameters that are outlined for them in the practical guide. The parameter values are then checked to verify whether they are fit enough for the plant model to work.

If the user enters invalid values, they are warned and then they can run the plant model if the parameters are valid. When the plant model is running, the user can visualize the plant itself and graphically; while adjusting the parameters (PID parameters, set-point, valve switches, etc) and get real-time feedback from the model.



Good day to you were assigned to
CoupleTank PID as your practical which is currently in progress
6 hours 39 minutes 35 seconds left

SIMATIC HMI Miniweb on SIEMENS-PC

Miniweb Start Page

last update 14:49

Welcome on SIEMENS-PC

Device Status of SIEMENS-PC
The runtime is running

General Device Information	
Device Type	Win32 PC
Device Name	SIEMENS-PC
SSLEAY_VERSION	OpenSSL 1.0.1j 15 Oct 2014

Figure 4.21: SCADA login on student page

The website functionalities are illustrated in **Appendix B: Platform Outcomes**. These include:

(i) Document uploading and downloading on the website database, (refer to Figure 7.9).

(ii) Accessing of student details from the database, (refer to Figure 7.10).

(iii) Practical scheduling by an administrator, (refer to Figure 7.11).

(iv) File uploading by students on the system and updating administrator page, (refer to Figure 7.12).

(v) Successful login and activity recording, (refer to Figure 7.13).

The C# and HMTL code for the website are documented in **Appendix C**

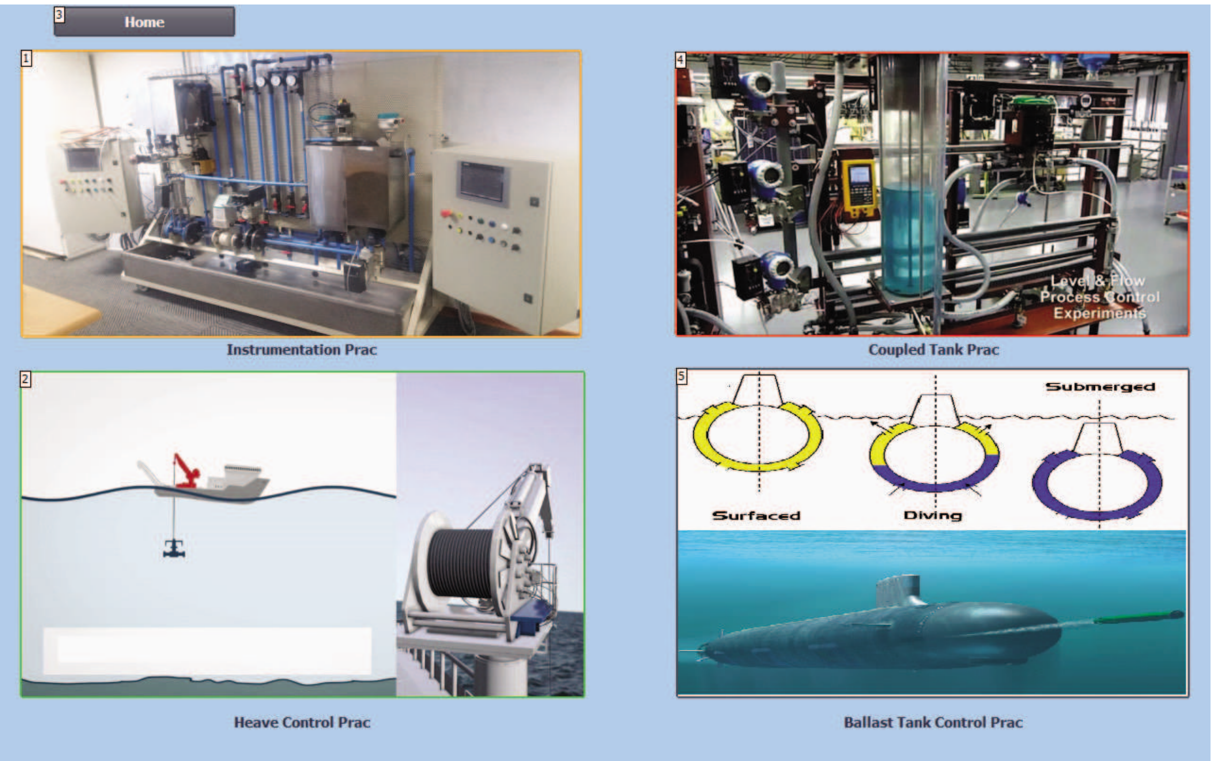


Figure 4.22: SCADA Home

Table 4.5 gives a full description of each page on the SCADA and the different tasks that the user may carry out. This also gives an outcome of what capabilities the designed platform offers to the teaching environment. After testing the platform and its subsystems, the final test that was implemented was assigning students to work on the dual-tank model and give feedback. This test is discussed in section 4.5.

Table 4.5: SCADA functionalities

Activity	Representation	Description
Ballast-Tank Control		
Home	Figure 7.1	A brief description of what the user should be aware of when implementing the practical.
Parameters	Figure 7.2	The user is to input valid values that are outlined in the practical guide.
Submarine Hovering	Figure 7.3	Upon entering valid parameters users may start running the platform and get a response of the submarine submerging or ascending, parameters may be altered while the user gets real-time response.
Graphical View	Figure 7.4	Depth of submarine, set-point and initial depth are represented on the graph. The user may also change PID parameters to get real-time response. Forces experienced by the vessel are also displayed to the user.
Dual-Tank		
Home	Figure 7.5	A brief description of what the user should be aware of when implementing the practical.
Parameters	Figure 7.6	User is to input valid values that are outlined in the practical guide.
Response in real-time	Figure 7.7	Upon entering valid parameters user may start running the model, by setting the switch on the pump and (opening/closing) the valves and the tanks will fill up or discharge in real-time.
Graphical View	Figure 7.8	Fluid levels in the tanks and set-point are represented on the graph, the user can change set-point and PID values.

4.5 Student Feedback

After working on the fully integrated platform, the students' responses are discussed in this section. The selected group of students who gave feedback are 3rd year BEng: Mechatronics students. Working on the platform was part of their control systems course, which incorporates various systems. Hence, the feedback from the students was a comparison of the developed system with the hardware systems (Figure 4.23) they used for other practical activities, and also a verification on the functionalities of the developed system.

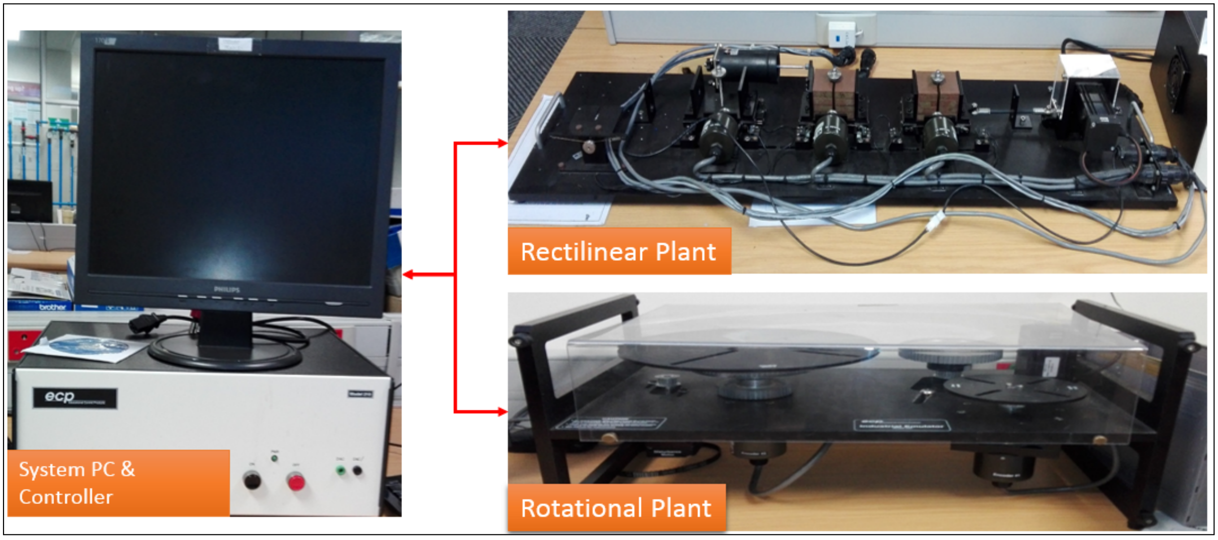


Figure 4.23: Control Systems Plants @ NMMU Mechatronics Department

A questionnaire was provided for the evaluation of the platform. The questionnaire (refer to **Appendix B : Questionnaire for users**) is divided into five main sections, which consist of 20 questions. The five sections are described as:

(i) Section 1: Graphical, visualization of platform presentation and real-time response – This section focuses on how the users rate the platform's SCADA, the real-time response and if they feel that the system should be improved. There are five questions, which are documented as questions (1 to 5) in the Questionnaire.

(ii) Section 2: Online practical Implementation – The section tries to get what the users feel on operating devices remotely without them being in physical presence. There are four questions which are (6 to 9) in the questionnaire.

(iii) Section 3: Content of given practical guides – For the provided document (refer to **Appendix B: Practical Guide for users**) that the users were given before attempting the given practical on the platform. This section gets feedback on whether the document had sufficient information to guide them to work on the platform. Section 3 consists of questions (10 to 13) in the questionnaire.

(iv) Section 4: This section focuses on how the users feel about using the platform and what they would feel is the best way to give knowledge on control systems with different methods. The questions asked are (14 to 16) in the Questionnaire.

(v) Section 5: From the outlined deliverables, the developed system had to meet, certain functionalities. These were verified and tested through the users. Hence, this section looks at some functions offered by the system and an overall impression the users have on the platform. The factors/questions asked are (17 to 20) in the Questionnaire.

From the questionnaire, the feedback responses were documented and are presented in the Feedback Results section that follows.

4.5.1 Feedback Results

The operation and feedback on the platform had 25 participants and Figure 4.24 shows all the participants' responses to the questions in the Questionnaire. These responses are expressed in their respective sections and outlined in the Questionnaire.

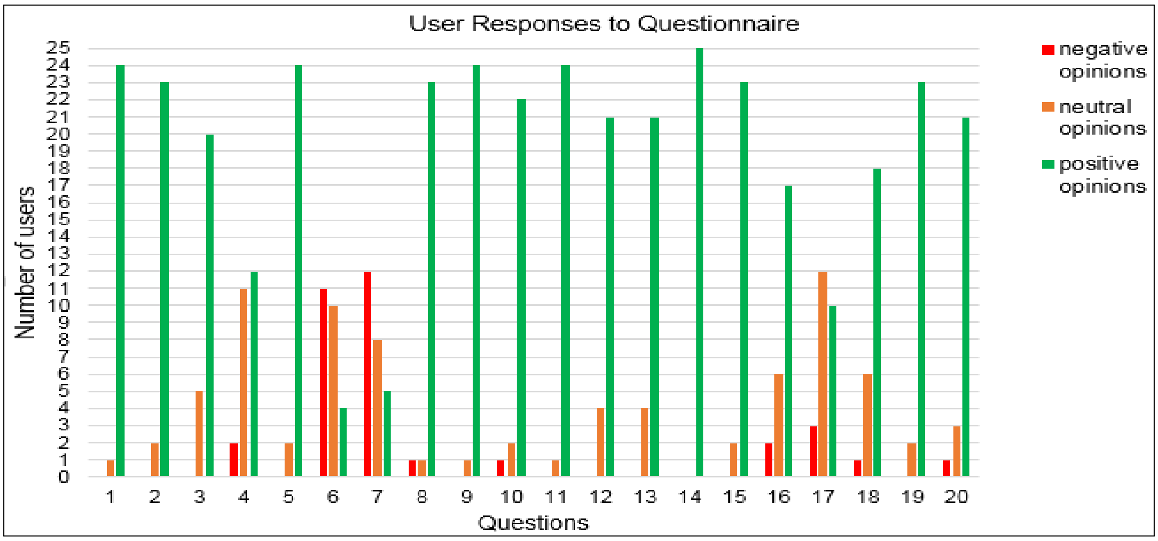


Figure 4.24: Overall feedback from users

Figure 4.25 represents the feedback on Section 1 in the questionnaire. Figure 4.25(a) shows that 92% of the students agreed that the graphical, visualization of platform presentation and real-time response all worked and were satisfactory. They also found the SCADA quite interesting and motivating; as it gave a feeling of operating a real system. Figure 4.25(b) shows that 48% of the students felt that the system could be improved by adding more functions and activities; while 44% had a neutral feeling. The neutral feeling is due to the fact that the system could have been a first-time experience for them; hence they could not be able to give an actual judgment on the platform when compared with any similar system.

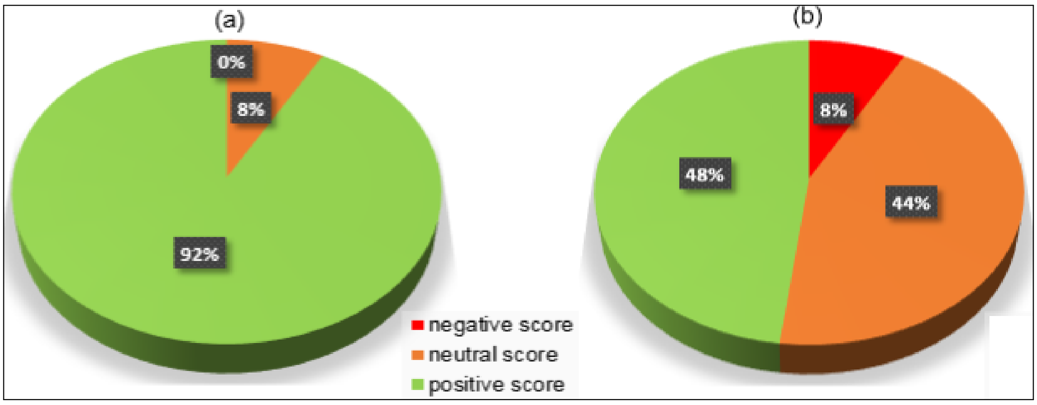


Figure 4.25: Section 1 feedback

Figure 4.26 represents the feedback on the operation of the system remotely via the internet. From Figure 4.26(a), 94% of the students agreed that the operation of the platform via the internet is a safer method than operating real hardware systems. Figure 4.26(b) shows that 46% of the students disagreed that working on the platform was challenging, and not as difficult, compared with other practical courses they have taken up during their university studies. As many as 36% of the students had a neutral opinion; and it may be concluded that they found it manageable to do.

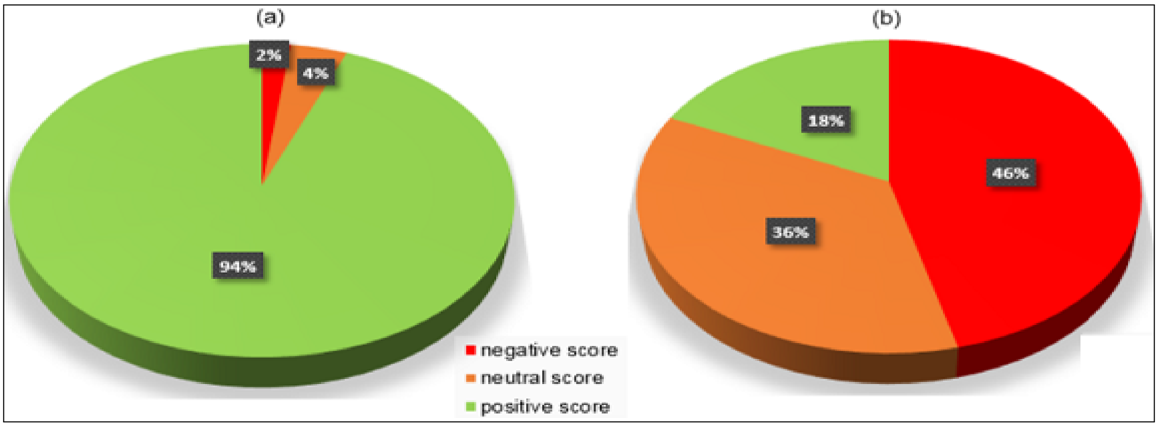


Figure 4.26: Section 2 feedback

Figure 4.27 shows that 88% of the students found the material provided in the practical guide useful and informative. They also found that the content in the document related to the theory given to them in class. Hence, the positive feedback on the practical guide may conclude that the users worked on the platform without facing a lot of challenges.

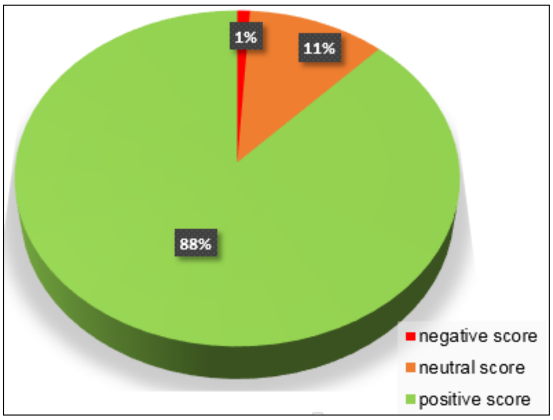


Figure 4.27: Section 3 feedback

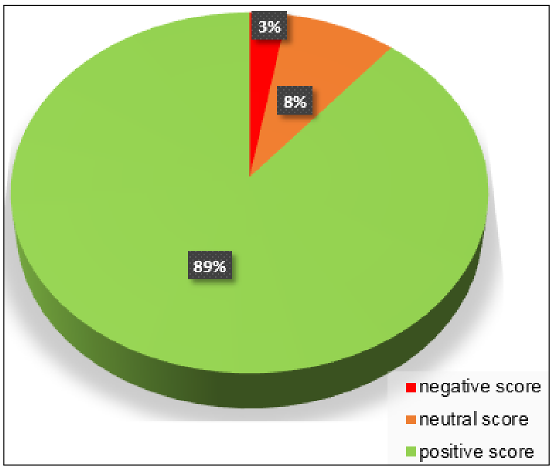


Figure 4.28: Section 4 feedback

An important part of analyzing the platform was to ensure that it adds significance to engineering education. The learning success of the platform is represented in Figure 4.28. A total of 89% of the students agreed that the platform meets the following success factors:

- The developed system illustrates the application of the theoretical concepts on real systems.
- The platform gives the users more understanding to control systems.

Some of the platform functionalities that the users had to give feedback on were the data acquisition from the system and an overall rating that represents whether the entire platform may be seen as useful for teaching purposes.

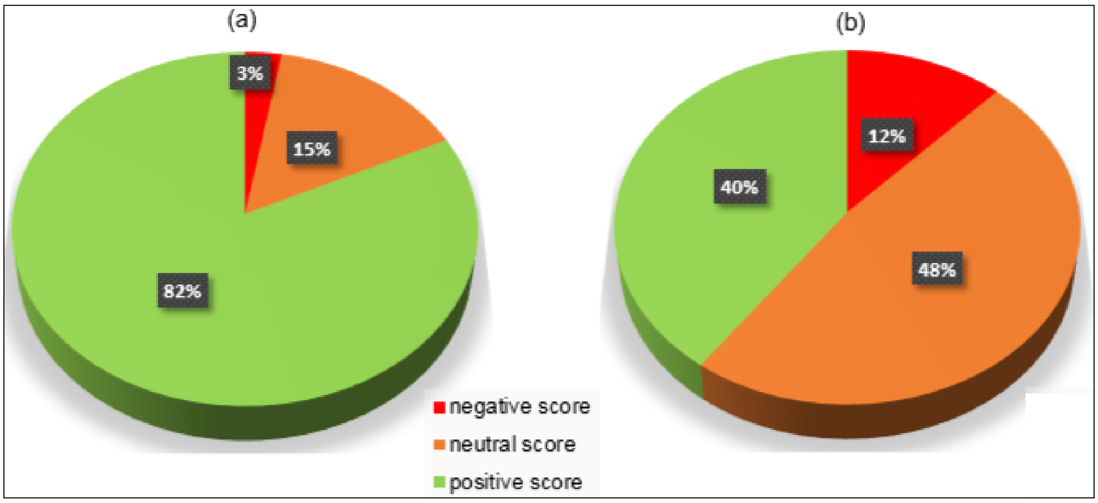


Figure 4.29: Section 5 feedback

From Figure 4.29(a) most students (82%) found the developed platform's data logging procedure fully functional; and the entire platform as usable for practical courses. Figure 4.29(b) represents the feedback on accessing the platform website; and specific functionalities were asked if they were a challenge to attempt, which include (registration, login and SCADA access). As many as 48% of the users found it not so challenging to navigate on the website and to work on the SCADA system; while 40% found it quite easy.

4.6 Evaluation and Experiments: Conclusion

Through the various tests and verifications discussed in this chapter, it may be noted that the developed platform is a useful system. The main aspects of real-time operation, the user operation of the platform and the educational significance are satisfactory, according to the acquired results. An overall conclusion of the system is further discussed in Chapter 6.

Chapter 5

Project Management

This chapter discusses the activities, the time-line and the costs incurred for the development of the platform.

5.1 Project Schedule

When the research project was defined and the deliverables were outlined, a work-bench structure was set up to ensure that system was developed in an organized manner. The platform-development procedure is defined by the work-bench structure shown in Figure 5.2. There are four main tasks, as well as their subtasks on the work bench structure, which included:

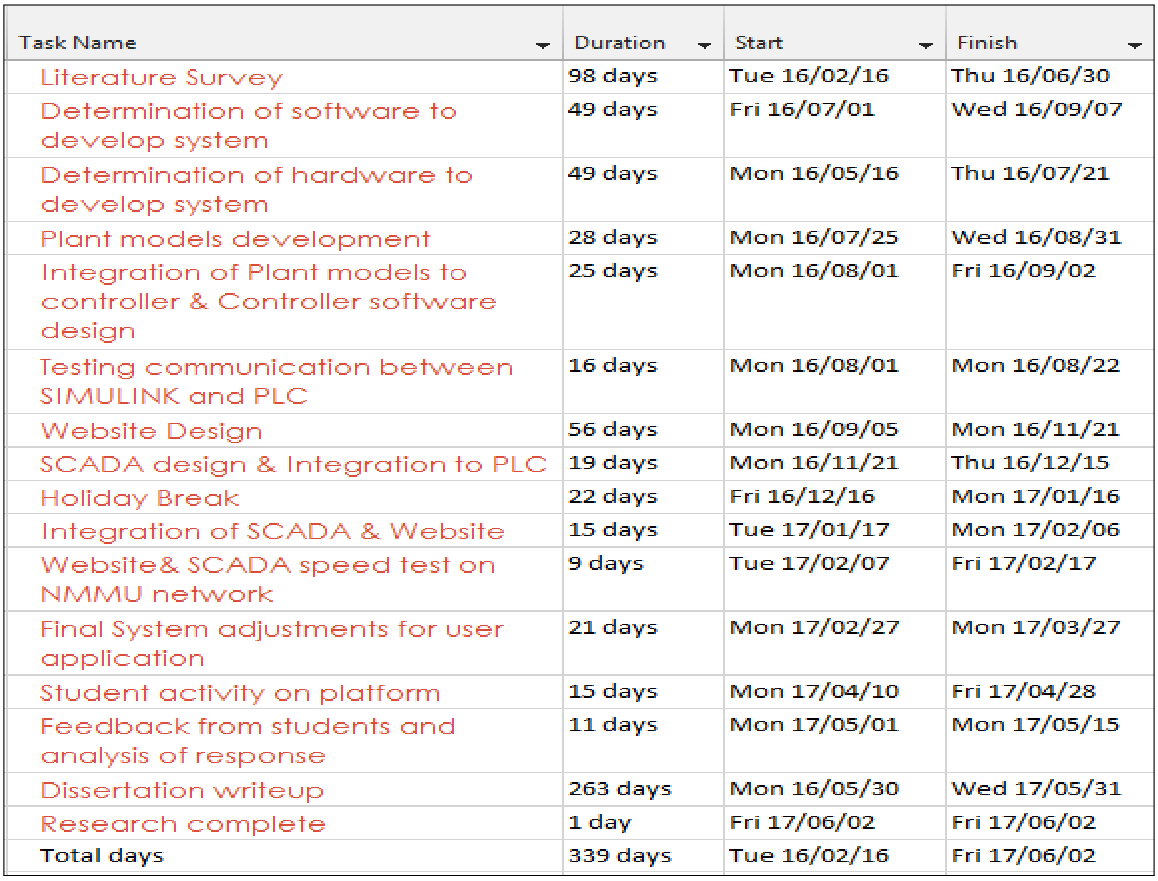
(i) **System Hardware** – this involved the selection of a server that is capable of running all the software for the designing and set-up of the system's software. The controller for the platform was selected; so that it provided communication with the plant models and was up to the industry standard.

(ii) **System Software and Integration** Integration – the entire platform was programmed on a Windows 7 (server software) operating system that accommodates all the various software packages used in the development of the platform. In Chapter 3, the sub-systems that make up the platform were programmed in the following order:
Designing plant models in Matlab Simulink, and then the integration of the plant models to the PLC. The PLC was then programmed to have full control of the plant models, as discussed in section **3.6 Controller Software Design**. After the PLC and the plant models were interfaced, the SCADA was then designed. Hence, after developing the SCADA system, the website development commenced. The platform website was developed in Visual Studio 2015 and

this is discussed in section 3.7 of the Remote Accessing Website Design. The website was then integrated to the SCADA, as discussed in section **3.8 Website Integration to Process Control Platform**.

(iii) **System Evaluation** – During the development of each sub-system, after the required functionalities had been met, the tests were implemented. This was to ensure that the platform works, according to the outlined requirements. The evaluation of the developed platform is discussed in Chapter 4: **Evaluation and Experiments**.

The time-line for the research is clearly shown in Figure 5.1. The number of days spent on each activity were outlined; and they follow the preceding activities, respectively. This time-line follows the activities shown on the work-bench structure.



Task Name	Duration	Start	Finish
Literature Survey	98 days	Tue 16/02/16	Thu 16/06/30
Determination of software to develop system	49 days	Fri 16/07/01	Wed 16/09/07
Determination of hardware to develop system	49 days	Mon 16/05/16	Thu 16/07/21
Plant models development	28 days	Mon 16/07/25	Wed 16/08/31
Integration of Plant models to controller & Controller software design	25 days	Mon 16/08/01	Fri 16/09/02
Testing communication between SIMULINK and PLC	16 days	Mon 16/08/01	Mon 16/08/22
Website Design	56 days	Mon 16/09/05	Mon 16/11/21
SCADA design & Integration to PLC	19 days	Mon 16/11/21	Thu 16/12/15
Holiday Break	22 days	Fri 16/12/16	Mon 17/01/16
Integration of SCADA & Website	15 days	Tue 17/01/17	Mon 17/02/06
Website& SCADA speed test on NMMU network	9 days	Tue 17/02/07	Fri 17/02/17
Final System adjustments for user application	21 days	Mon 17/02/27	Mon 17/03/27
Student activity on platform	15 days	Mon 17/04/10	Fri 17/04/28
Feedback from students and analysis of response	11 days	Mon 17/05/01	Mon 17/05/15
Dissertation writeup	263 days	Mon 16/05/30	Wed 17/05/31
Research complete	1 day	Fri 17/06/02	Fri 17/06/02
Total days	339 days	Tue 16/02/16	Fri 17/06/02

Figure 5.1: Activities on the research time-line

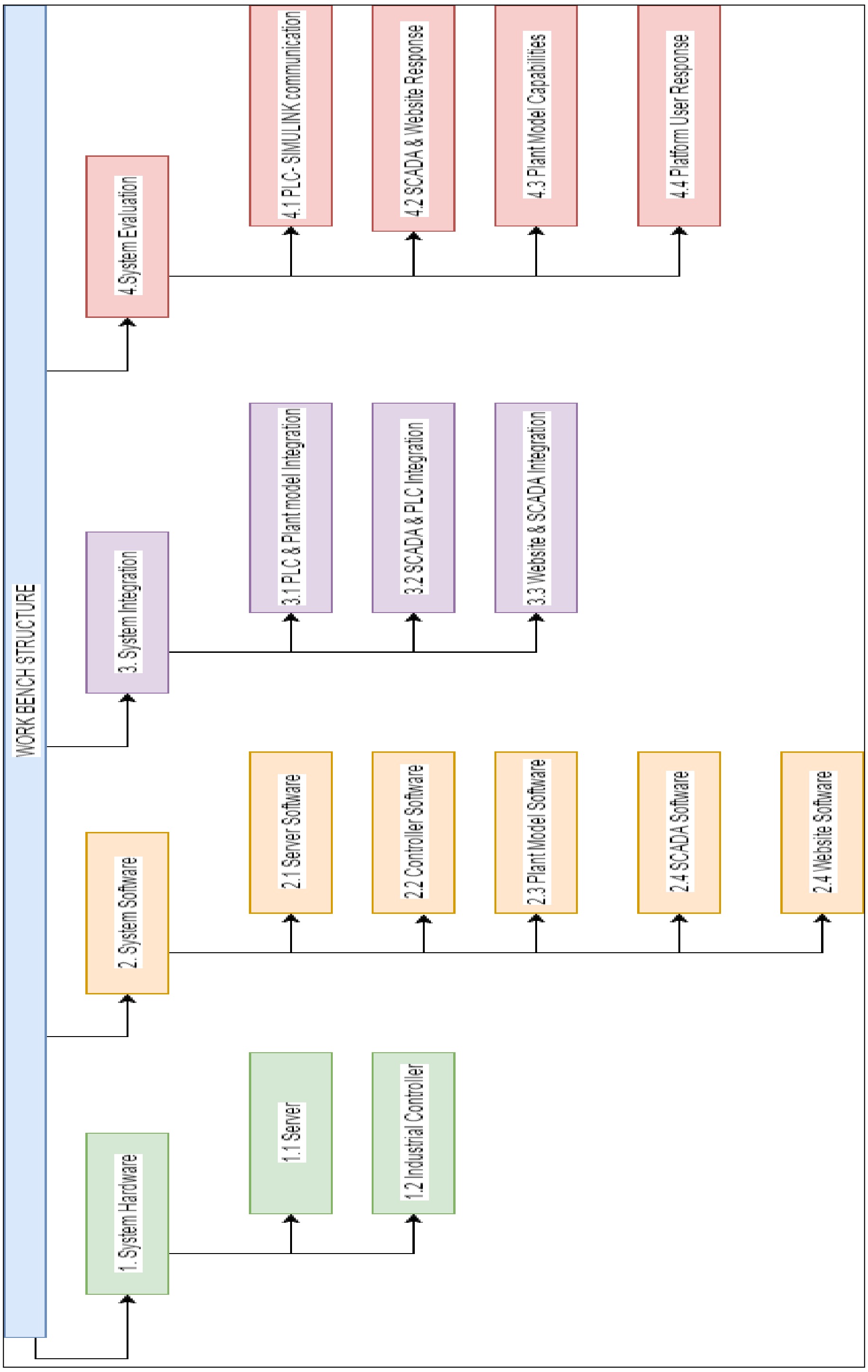

WORK BENCH STRUCTURE
1. System Hardware
1.1 Server
1.2 Industrial Controller
2. System Software
2.1 Server Software
2.2 Controller Software
2.3 Plant Model Software
2.4 SCADA Software
2.4 Website Software
3. System Integration
3.1 PLC & Plant model Integration
3.2 SCADA & PLC Integration
3.3 Website & SCADA Integration
4.System Evaluation
4.1 PLC- SIMULINK communication
4.2 SCADA & Website Response
4.3 Plant Model Capabilities
4.4 Platform User Response

Figure 5.2: Work Bench Structure

5.2 Cost Analysis

The developed platform is an NMMU project that was funded by merSETA in collaboration with AMTC (Advanced Mechatronic Training Centre). The cost of the construction of the entire system is summarised in Table 5.1. The components, without any stated prices, were all provided by the AMTC and NMMU ICT services.

Table 5.1: Cost of System

Component	**Unit Price (ZAR)**	**Quantity**	**Total Cost (ZAR)**
S7-1200, CPU 1217C (PLC)	8 220	1	8 220
S7-1200, power module	876	1	876
S7-1200, analog I/O	3 252	1	3 252
Control Box	- - - - -	1	- - - - -
TIA PORTAL(license)	- - - - -	1	- - - - -
PC Server	13 000	1	13 000
D-LINK ethernet switch	500	1	500
Matlab Simulink (license)	- - - - -	1	- - - - -
VISUAL STUDIO and SQL (license)	- - - - -	1	- - - - -
System Total Cost			**25 848**

Chapter 6

Conclusion

The platform's performance was based on the outlined objectives and the various evaluation tests implemented. The designed platform has two plant models, which proves that it is reconfigurable; and these plant models are interfaced to a PLC and SCADA system. The real-time response of the platform was done by ensuring that the TCP/IP refresh time on the PLC gives the same time as the run time of the plant models in Matlab Simulink. After ensuring that the plant models and the PLC communicate in real-time, the SCADA system and website were evaluated on the NMMU network by implementing ping tests. The two sub-systems gave good responses; as they had similar data-transfer rates (Table 4.1); and this proved that they work in real-time on the NMMU network. The real-time response of the platform was also justified by the students, having 92% (Figure 4.25(a)) of them agreeing that the platform works in real-time.

From the student feedback, the platform may be regarded as a feasible and functional teaching platform in engineering learning. The learning success seen by the students had a positive score of 89%(Figure 4.28) of them finding the platform to have a close relationship with the theory given in the class. SCADA operation, data logging, website access and navigation were also found to be fully functional by most (82%) of the students (Figure 4.29(a)). Hence, it may be concluded that a fully functional mechatronic platform was developed for teaching purposes in different disciplines (marine and mechatronics). The platform was designed by using real industry hardware (PLC) to control the plant models developed in the software, giving a real-time response similar to that of real hardware plants. Thus, the developed platform met its outlined requirements. The various contributions the platform offers are further discussed in the section that follows.

6.1 Developed-Platform Contributions

The platform has been evaluated with numerous tests on the different functionalities it offers in Chapter 4. However, looking at the entire platform's capabilities and limitations, its significance is discussed in this section. The platform will offer various activities that are often used to teach engineering concepts on process control. These include:

(i) The implementation of intelligent controllers (PD, PID);

(ii) The control and feedback on plant models;

(iii) Determining plant parameters and plotting response values from the plant models;

(iv) Data acquisition via the internet;

(v) Deriving mathematical equations and relating them to the plant model I/Os;

(vi) Drawing logical observations noted after working on the platform.

Looking at the exponentially growing number of students in universities; there is a need to ensure that all students acquire sufficient knowledge and exposure to real-world systems. Hence, by having a lot of students working on one real hardware system cannot be accomplished without more costs being incurred to maintain it; as it is more vulnerable to damage. Thus, the developed platform eliminates this challenge; because the plant models are designed purely in software (Matlab Simulink); and they can be altered without any additionally costs being incurred (re-programmed).

Another observed challenge is the rapid growth in technology, which results in learning institutions becoming financially strained; as they try to upgrade their hardware and the software components to keep up with the current industry systems. However, virtual instrumentation offered by the SCADA on the developed platform allows the expansion or upgrade of the different plant models with minimal costs; and in the shortest time.

The system and its capabilities also strive to eradicate the fact that there are “impossible” activities to implement and demonstrate to students. For instance, it is impossible for Marine engineering students to go to the harbour and carry out practicals on a real submarine; but they also need to relate the theory to a system that gives some response. From the platform, this challenge is counteracted by the ballast-tank control model for a submarine. The model has been developed in such a way that it emulates how a submarine would ascend and descend relative to the set-up parameters. Hence, Marine engineering students can apply their theoretical knowledge on a system built with industry-standard software and hardware.

The developed website on the platform offers a feasible and well-managed interaction for both lecturers and students. Students can remotely access and work on the platform SCADA at allocated time sessions. This means that every user gets an equal opportunity to work on the platform – without having to wait for their peers to finish if they take too long. The approach of remotely operating the system prepares students to have an insight on Industry 4.0.

During the development of the research an onsite system was also developed, which had a PLC and a plant model interfaced for practical and design purposes. From this, a publication was presented at the 2016 ROBMECH Conference entitled:
“A Process Control Learning Factory with a Plant Simulation integrated to Industry Standard Control Hardware” (refer to **Appendix B: Platform Outcomes 7.2.6**). The future work based on the platform’s limitations is discussed in the section that follows.

6.2 Future Work based on the Developed Platform

After evaluation and meeting the outlined deliverables, it was seen that future work that can be done, based on the platform. Three critical aspects to improve the existing platform are discussed in this section.

6.2.1 Expansion of System to Multiple Controllers

This section deals with accessing the platform server by more than one PLC to assigned Simulink models, so that they may work simultaneously during a session. The set-up is implemented by using an Ethernet switch that accommodates multiple devices on the same network. The set-up of multiple PLCs having access to the platform server enables the reuse and sharing of resources. As shown in Figure 6.1, more than one user can work and control the designed Simulink plant models. However, for each station to be functional, the following steps have to be considered to ensure that there is no interference among the users during their sessions.

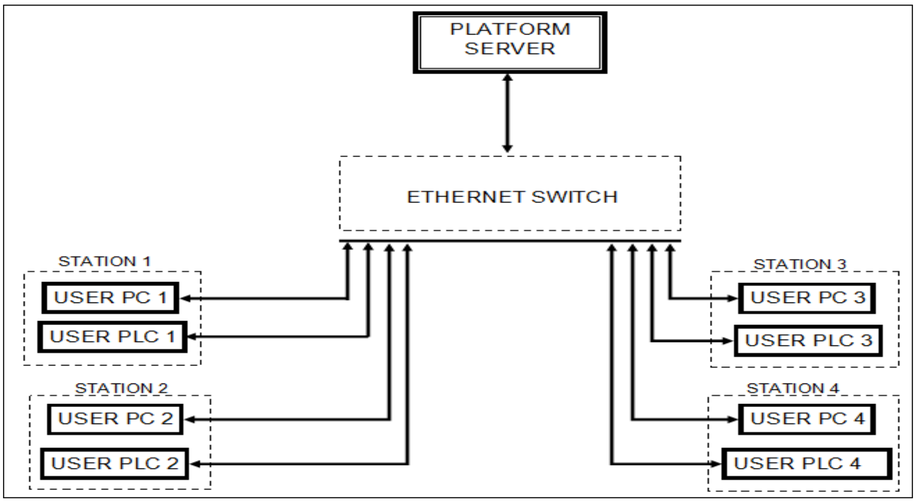


Figure 6.1: Multiple PLC Access to Matlab Simulink Server

These steps include:

- Each PLC should be connected to its unique model(s) using the TCP/IP Protocol.
- Each station must have a unique TCP/IP address, i.e. IP address and Port value.
- When running each plant model on the server; it must be in a separate simulation. This is done to ensure that if there is a failure experienced by one user, it would not interrupt other users by having the entire Simulink program malfunction or crash.

Expanding the platform in this manner will mean that the platform can be used without the need of an internet network. This also means that the platform can be used for students' tests or exam evaluations; as each student has his/her own working station.

6.2.2 Automated practical-time selection

The developed platform's current website scheduling system is done by the administrator/lecturer. Hence, to make things easier for the lecturer and the students, an improved scheduling system may be implemented. The website for the platform must be set up, so that every user can work on the platform at their desired time within the time-frame of the practical course. The website scheduling system must be developed in a manner, whereby time slots are made available to the users and locked to one user, when selected. A user is to have a limited number of slots to select. This scheduling system should also be controlled by the lecturers/administrators. Thus, the system would have better management that accommodates every user.

6.2.3 Development of a Real and Virtual Laboratory HUB

The reconfigurability of the developed platform, which is to accommodate more than one plant model, may be expanded. This may be done by interfacing real hardware systems to the PLC running on the platform. It would also be necessary to add a real-time view of the hardware systems by deploying an IP camera, as illustrated in Figure 6.2 and integrated to the website. Adding hardware systems would also give the students a feeling of both real and virtual systems. A well-designed HUB would bring an environment, in which knowledge and resources are shared in the most cost-effective and efficient way.

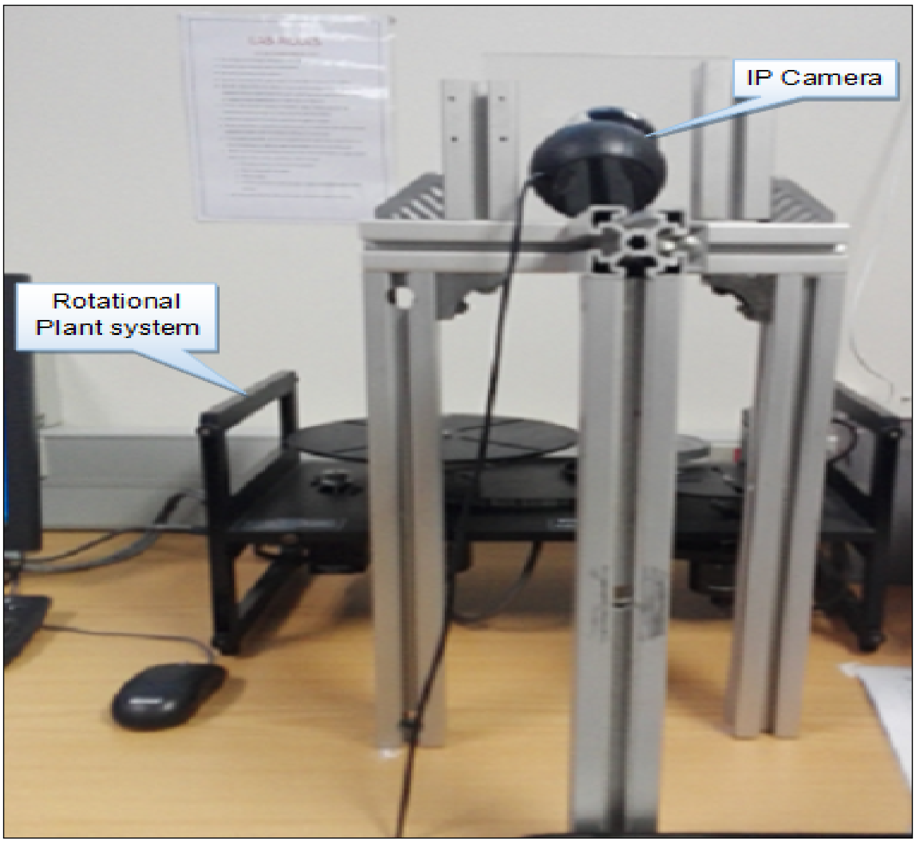


Figure 6.2: Setup for remote accessing Rotational Plant @ AMTC NMMU Laboratory

Bibliography

[1] Marialena Vagia, Aksel A., Transeth and Sigurd A. Fjerdingen. A literature review on the levels of automation during the years. what are the different taxonomies that have been proposed? *Applied Ergonomics*, 53, Part A:190–202, 2016.

[2] Process control systems-automation technology us- siemens url: http://www.industry.usa.siemens.com/automation/us/en/process-control-system/pages/process-control-system.aspx.

[3] Antonio Delgado. Automation and control in food production. *CONTROL SYSTEMS, ROBOTICS AND AUTOMATION*, 19, 2002.

[4] Daniel Hodouin. Methods for automatic control, observation, and optimization in mineral-processing plants. *Journal of Process Control*, 21(2):211–225, 2011.

[5] Stefanos A. Nikolidakis, Dionisis Kandris, Dimitrios D. Vergados, and Christos Douligeris. Energy efficient automated control of irrigation in agriculture by using wireless sensor networks. *Computers and Electronics in Agriculture*, 113:154–163, 2015.

[6] Yoshisada Nagasaka, Katsuhiko Tamaki, Kentaro Nishiwaki, Masahiro Saito, Yutaka Kikuchi, and Kota Motobayashi. 4th ifac conference on modelling and control in agriculture, horticulture and post-harvest industry a global positioning system guided automated rice transplanter. *IFAC Proceedings Volumes*, 46(18):41–46, 2013.

[7] Vu Van Tan, Dae-Seung Yoo, and Myeong-Jae Yi. An approach to remote monitoring and control based on opc connectivity. *IFAC Proceedings Volumes*, 42(3):259–262, 2009.

[8] Ravendra Singh, Marianthi Lerapetritou, and Rohit Ramachandran. System- wide hybrid mpc–pid control of a continuous pharmaceutical tablet manufacturing process via direct compaction. *European Journal of Pharmaceutics and Biopharmaceutics*, 85(3, Part B):1164–1182, 2013.

[9] Paraskevas D. Tzanavaras and Demetrius G. Themelis. Automated determination of flutamide by a validated flow-injection method: Application to dissolution studies of pharmaceutical tablets. *Journal of Pharmaceutical and Biomedical Analysis*, 43(5):1820–1824, 2007.

[10] Krist V. Gernaey and Rafiqul Gani. A model-based systems approach to pharmaceutical product-process design and analysis. *Chemical Engineering Science*, 65(21):5757–5769, 2010.

[11] Ali N. Saleemi, Gerry Steele, Nicholas I. Pedge, Anthony Freeman, and Zoltan K. Nagy. Enhancing crystalline properties of a cardiovascular active pharmaceutical ingredient using a process analytical technology based on crystallization feedback control strategy. *International Journal of Pharmaceutics*, 430(1–2):56–64, 2012.

[12] Ravendra Singh, Marianthi Lerapetritou, and Rohit Ramachandran. An engineering study on the enhanced control and operation of continuous manufacturing of pharmaceutical tablets via roller compaction. *International Journal of Pharmaceutics*, 438(1–2):307–326, 2012.

[13] S. Da’na, A. Sagahyroon, A. Elrayes, A. R. Al-Ali, and R. Al-Aydi. Development of a monitoring and control platform for plc-based applications. *Computer Standards and Interfaces*, 30(3):157–166, 2008.

[14] Ramazan Bayindir and Yucel Cetinceviz. A water pumping control system with a programmable logic controller (plc) and industrial wireless modules for industrial plants—an experimental setup. *ISA Transactions*, 50(2):321–328, 2011.

[15] Pc versus plc url:http://www.controleng.com/single- article/inside-machines-pc-versus-plc-comparing-control-options/9bf8690c6f23b11370bec90b52cb15c9.html.

[16] Entertron Industries http://www.entertron.com. Entertron industries, inc., manufacturing programmable logic controllers (plcs) for oems, since 1978. 2016.

[17] Chen Ran, Quan Liang, and Li Li. Ceis (2011) the study of soft plc running system. *Procedia Engineering*, 15:1234–1238, 2011.

[18] Maria L. Botero, Anders Selmer, Roger Watson, Mukta Bansal, and Markus Kraft. Cambridge weblabs: A process control system using industrial standard simatic pcs 7. *Education for Chemical Engineers*, 16:1–8, 2016.

[19] Bernhard Voglauer, Romain Garcia, and H. Peter Jörgl. {IMPROVEMENTS} {OF} a three-tank-system {OPERATED} {IN} {REAL} {TIME} {WITH} {MATLAB} {IN} a plc-profibus-network. *{IFAC} Proceedings Volumes*, 38(1):43 – 48, 2005. 16th {IFAC} World Congress.

[20] S. Ozana and M. Pies. 9th ifac workshop on programmable devices and embedded systems design and implementation of embedded systems in matlab & simulink environment. *IFAC Proceedings Volumes*, 42(1):73 – 79, 2009.

[21] Mike Barth and Alexander Fay. Automated generation of simulation models for control code tests. *Control Engineering Practice*, 21(2):218 – 230, 2013.

[22] Robert K. Cox, Julie F. Smith, and Yiannis Dimitratos. Can simulation technology enable a paradigm shift in process control? Modelling for the rest of us. *Computers and Chemical Engineering*, 30(10–12):1542–1552, 2006.

[23] ACT Operations Research. Scenario navigator | systems navigator - simulation - optimization - forecasting - process control - act operations research. 2016.

[24] Arnout Pool, Jacob Wijngaard, and Durk-Jouke van der Zee. Lean planning in the semi-process industry, a case study. *International Journal of Production Economics*, 131(1):194–203, 2011.

[25] László Szirmay-Kalos, Gábor Márton, Tibor Fóris, and Tamás Horváth. Development of process-visualization systems: An object-oriented approach. *Journal of Systems Architecture*, 46(3):275–296, 2000.

[26] Esteban Arroyo, Mario Hoernicke, Pablo Rodríguez and Alexander Fay. Automatic derivation of qualitative plant simulation models from legacy piping andinstrumentation diagrams. *Computers and Chemical Engineering*, 92:112–132, (2016).

[27] Idowu Adeyemi and Isam Janajreh. Modelling of the entrained flow gasification: Kinetics-based aspen plus model. *Renewable Energy*, 82:77–84, (2015).

[28] William X. Meng, Subhodeep Banerjee, Xiao Zhang, and Ramesh K. Agarwal. Process simulation of multi-stage chemical-looping combustion using aspen plus. *Energy*, 90, Part 2:1869–1877, (2015).

[29] H. A. El Mawgoud, T. M. Elshiekh, and S. A. Khalil. Process simulation for revamping of a dehydration gas plant. *Egyptian Journal of Petroleum*,24(4):475–482, (2015).

[30] José O. Valderrama, Luis A. Toselli, and Claudio A. Faúndez. Advances on modelling and simulation of alcoholic distillation. part 2: Process simulation. *Food and Bioproducts Processing*, 90(4):832–840, (2012).

[31] Matlab url: http://www.mathworks.com/products/matlab/.

[32] Labview system design software url: http://www.ni.com/labview/.

[33] Bernhard Voglauer, Romain Garcia, and H. Peter Jörgl. Improvements of a three-tank-system operated in real time with matlab in a plc-profibus-network. *IFAC Proceedings Volumes*, 38(1):43–48, 2005.

[34] A. Lakshmi Sangeetha, B. Naveenkumar, A. Balaji Ganesh, and N. Bharathi. Experimental validation of pid based cascade-control system through scada–plc–opc and internet architectures. *Measurement*, 45(4):643–649, 2012.

[35] Vinod Patel, Anand Joshi, B. R. Poorna Chandra, K. P. Geevarghese, and K. V. Gangadharan. 2nd international conference on innovations in automation and mechatronics engineering, iciame (2014) design and implementation of remote mechatronics laboratory for e-learning using labview and smartphone and cross-platform communication toolkit (scct). *Procedia Technology*, 14:108–115, 2014.

[36] Antonio Visioli, Manuel Berenguel, Alessandro Beghi, Andrea Cervato, and Mirco Rampazzo. Ifac workshop on internet based control education ibce15a remote refrigeration laboratory for control engineering education. *IFAC-PapersOnLine*, 48(29):25–30, (2015).

[37] Zhang Lixiang, Xiaomei Chen, and Hongyi Gao. International conference on advances in computational modeling and simulation: a remote plc laboratory design and realization. *Procedia Engineering*, 31:1168–1172, 2012.

[38] Ademir F. da Silva, Ricardo L. Ohta, Marcelo N. dos Santos, and Alecio P. D. Binotto. A cloud-based architecture for the internet of things targeting industrial devices remote monitoring and control. *IFAC-PapersOnLine*, 49(30):108–113, (2016).

[39] A critical look on industry 4.0 | allaboutlean.com. (2015).

[40] Frank Rennung, Caius Tudor Luminosu, and Anca Draghici. Service provision in the framework of industry 4.0. *Procedia - Social and Behavioral Sciences*, 221:372–377, (2016).

[41] Tina Krausser, Lars Evertz, and Ulrich Epple. 9th ifac symposium advances in control educational hands-on laboratory on industrial hardware, process control and advanced automation. *IFAC Proceedings Volumes*, 45(11):372–377, (2012).

[42] Alexandre Dolgui, Jurek Sasiadek, Marek Zaremba, Stephan Weyer, Mathias Schmitt, Moritz Ohmer, and Dominic Gorecky. 15th ifac symposium on information control problems in manufacturing towards industry 4.0 - standardization as the crucial challenge for highly modular, multi-vendor production systems. *IFAC-PapersOnLine*, 48(3):579–584, (2015).

[43] Albert M. Cook and Janice M. Polgar. *Chapter 8 – Accessing Mainstream Information and Communication Technologies: The Technology and the Web*, pages 171–195. Mosby, St. Louis (MO), (2015).

[44] M. Kozovsky. Remote programming and monitoring of industrial devices using nat traversal and usb over ip**this work was supported by grant no. fekt-s-14-2429 - the research of new control methods, measurement procedures and intelligent instruments in automation", which was funded by the internal grant agency of Brno university of technology. *IFAC-PapersOnLine*, 49(25):254–259, (2016).

[45] B. R. Mehta and Y. J. Reddy. *Chapter 15 - OPC communications*, pages 459–477. Butterworth-Heinemann, Oxford, 2015.

[46] Miriam Schleipen, Syed-Shiraz Gilani, Tino Bischoff, and Julius Pfrommer. Opc ua and industrie 4.0 - enabling technology with high diversity and variability. *Procedia CIRP*, 57:315–320, (2016).

[47] Wang hong you and Zhao san ping. The predigest project of tcp/ip protocol communication system based on dsp technology and ethernet. *Physics Procedia*, 25:1253–1257, (2012).

[48] César D. Maciel and Carsten M. Ritter. Tcp/ip networking in process control plants. *Computers and Industrial Engineering*, 35(3):611–614, 1998.

[49] Derrick Rountree. *Chapter 5 – Internet Connection Security*, pages 123–166. Syngress, Boston, (2013).

[50] Wei Wei, Bing Wang, Chun Zhang, Jim Kurose, and Don Towsley. Classification of access network types: Ethernet, wireless lan, adsl, cable modem or dialup? *Computer Networks*, 52(17):3205–3217, (2008).

[51] M. Domínguez, J. J. Fuertes, P. Reguera, A. Morán, S. Alonso, and M. A. Prada. Remote laboratory for learning of ac drive control. *IFAC Proceedings Volumes*, 44(1):8539–8544, 2011.

[52] Tijana Djukic, Vesna Mandic, and Nenad Filipovic. Virtual reality aided visualization of fluid flow simulations with application in medical education and diagnostics. *Computers in Biology and Medicine*, 43(12):2046–2052, (2013).

[53] Zafer Bekirogullari, Melis Yagmur Minas, O. M. Zamyatina, M. G. Minin, D. S. Denchuk, and V. O. Sadchenko. 5th iceepsy international conference on education and educational psychology analysis of engineering invention competencies in standards and programmes of engineering universities. *Procedia - Social and Behavioral Sciences*, 171:1088–1096, (2015).

[54] The American society of mechanical engineers @ONLINE, ”https://www.asme.org/”, (June 2016).

[55] Control systems 2016 @ONLINE, ”http://www.hse.gov.uk/comah/sragtech/techmeascontsyst.ht (June 2016).

[56] What is systems engineering@ONLINE, ”http://www.incose.org/aboutse/whatisse”, (June 2016).

[57] Luiz Eduardo G. Martins and Tony Gorschek. Requirements engineering for safety-critical systems: A systematic literature review. *Information and Software Technology*, 75:71–89, (2016).

[58] Pierre Mauborgne, Samuel Deniaud, Eric Levrat, Eric Bonjour, Jean-Pierre Micaëlli, and Dominique Loise. Operational and system hazard analysis in a safe systems requirement engineering process – application to automotive industry. *Safety Science*, 87:256–268, (2016).

[59] Alejandra J. Magana, Michael L. Falk, Camilo Vieira, and Michael J. Reese Jr. A case study of undergraduate engineering students' computational literacy and self-beliefs about computing in the context of authentic practices. *Computers in Human Behavior*, 61:427–442, 2016.

[60] António Dourado. 5th ifac conference on management and control of production logistics teaching production planning and control to informatics engineering students. *IFAC Proceedings Volumes*, 43(17):303–306, (2010).

[61] Fundamentals of industrial instrumentation and process

control@ONLINE,"http://mhebooklibrary.com/doi/pdf/10.1036/0071457356", (June 2016).

[62] https://www.bastiansolutions.com/blog/index.php/2013/04/24/pc-based-controls-vs-plc-based-controls-for-machine-automation/, 2013-04-24.

[63] http://www.innovasic.com/news/industrial-ethernet/comparing-plcs-and-industrial-pcs-for-automation-and-control-systems-part-2/, 2015-03-03.

[64] http://www.controleng.com/new-products/plcs-and-pacs/single-article/inside-machines-pc-versus-plc-comparing-control-options/9bf8690c6f.html, (2017).

[65] Simatic s7-1200 url http://w3.siemens.com/mcms/programmable-logic-controller/en/basic-controller/s7-1200/pages/default.aspx.

[66] Simatic wincc (tia portal) engineering software,

url:http://w3.siemens.com/mcms/automation-software/en/tia-portal-software/wincc- tia-portal/wincc-tia-portal-es/pages/default.aspx.

[67] Simatic wincc sm@rtserver (tia portal),url: http://w3.siemens.com/mcms/automation-software/en/tia-portal-software/wincc-tia-portal/wincc-tia-portal-options/simatic-wincc-smartserver/pages/default.aspx.

[68] https://nurelm.com/web-based-vs-desktop-software/, (2017).

[69] http://www.optimalnetworks.com/cost-remote-access/.

[70] https://www.visualstudio.com/, 2017-04-28.

[71] https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms, (2017).

[72] David Minne 207022526. Development of coupled tank level control system. *EMP 4110 Mechatronics Project*, (2010).

[73] Yunus Cengel & Robert Turner. *Fundamentals of Thermal Fluid Sciences*. McGraw-Hill, Singapore, (2012).

[74] Farid Golnaraghi & Benjamin C. Kuo. *Automatic Control Systems 9th edition*. John Wiley & Sons ,Inc, United States of America, (2010).

[75] Carolina Grob and Bruno G. Pollet. Regrowth in ship’s ballast water tanks: Think again! *Marine Pollution Bulletin*, 109(1):46–48, (2016).

[76] Zhixin Qi, Ian Eames and Alistair Greig. Flushing ballast tanks. *Ocean Engineering*, 89:157–172, (2014).

[77] Sae Yong Jang Mike Carter. Submarine buoyancy control ,http://charm.stanford.edu/engr1052016/mikecartersaeyongjang, (2016).

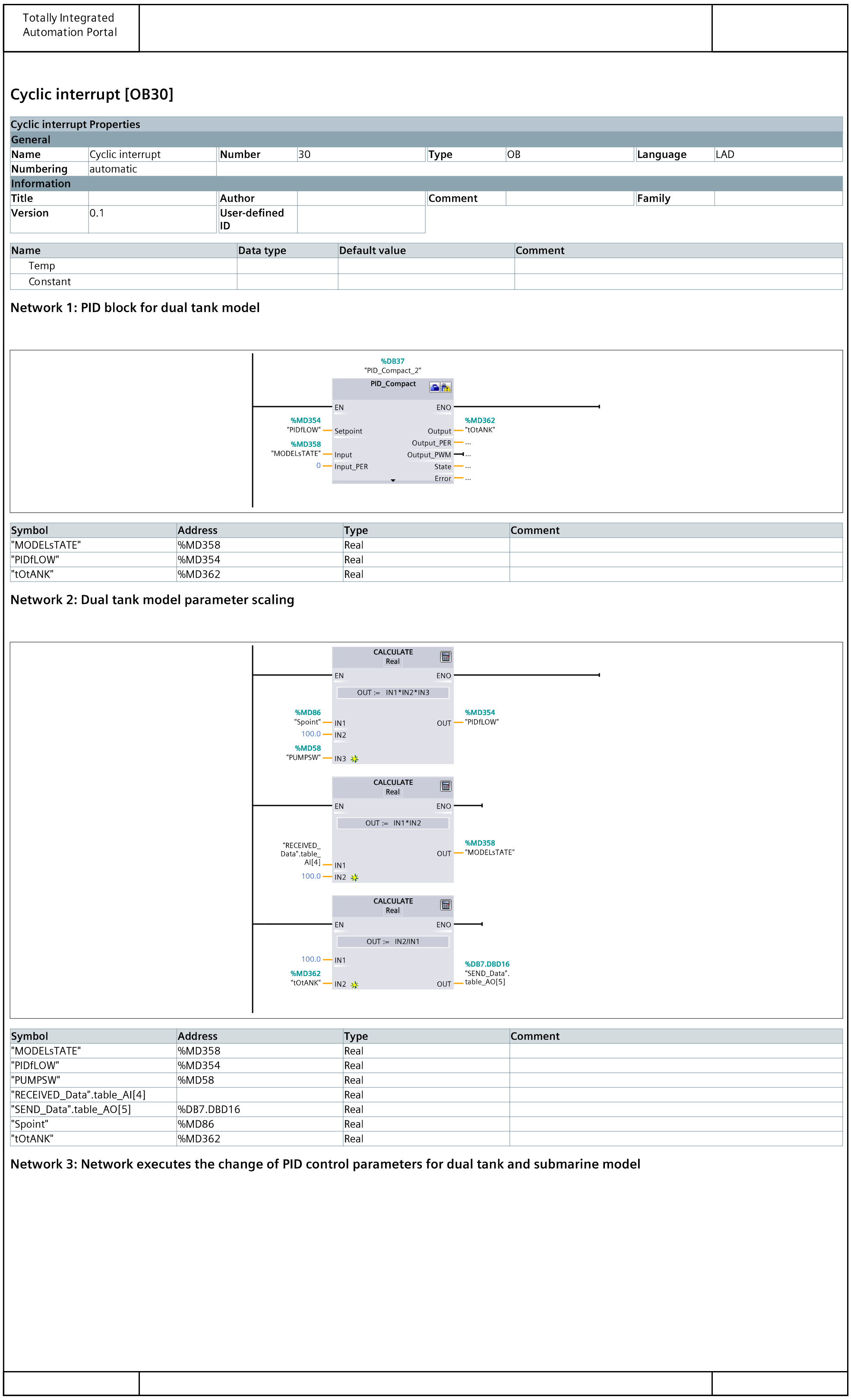
[78] How to run a ping test url https://iihelp.iinet.net.au/.

Appendices

7.1 Appendix A : PLC Programs

In this section the PLC program that executes communication and process response to plant models is documented. The various blocks discussed in Section 3.6 are all shown with their tags.

7.1.1 PLC Ladder Code



Totally Integrated
Automation Portal

Cyclic interrupt [OB30]

Cyclic interrupt Properties							
General							
Name	Cyclic interrupt	Number	30	Type	OB	Language	LAD
Numbering	automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value	Comment
Temp			
Constant			

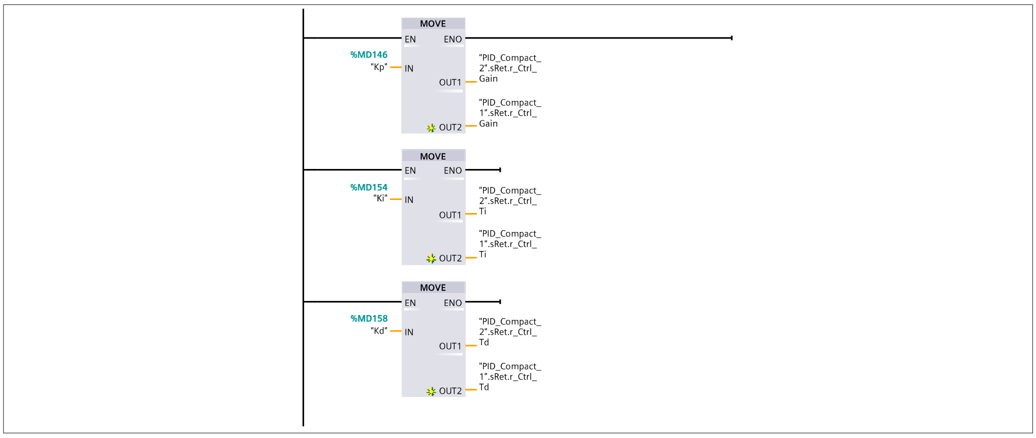
Network 1: PID block for dual tank model

Symbol	Address	Type	Comment
"MODELsTATE"	%MD358	Real	
"PIDfLOW"	%MD354	Real	
"tOtANK"	%MD362	Real	

Network 2: Dual tank model parameter scaling

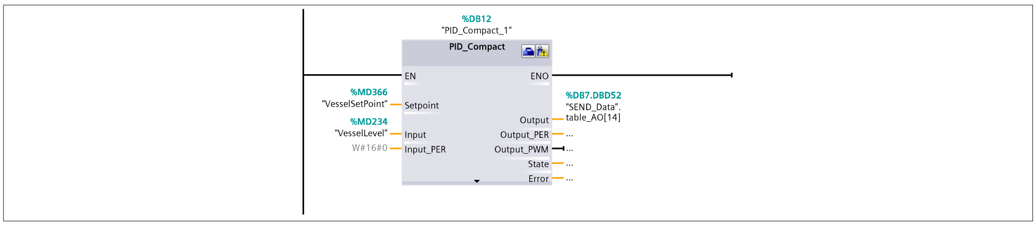
Symbol	Address	Type	Comment
"MODELsTATE"	%MD358	Real	
"PIDfLOW"	%MD354	Real	
"PUMPSW"	%MD58	Real	
"RECEIVED_Data".table_AI[4]		Real	
"SEND_Data".table_AO[5]	%DB7.DBD16	Real	
"Spoint"	%MD86	Real	
"tOtANK"	%MD362	Real	

Network 3: Network executes the change of PID control parameters for dual tank and submarine model

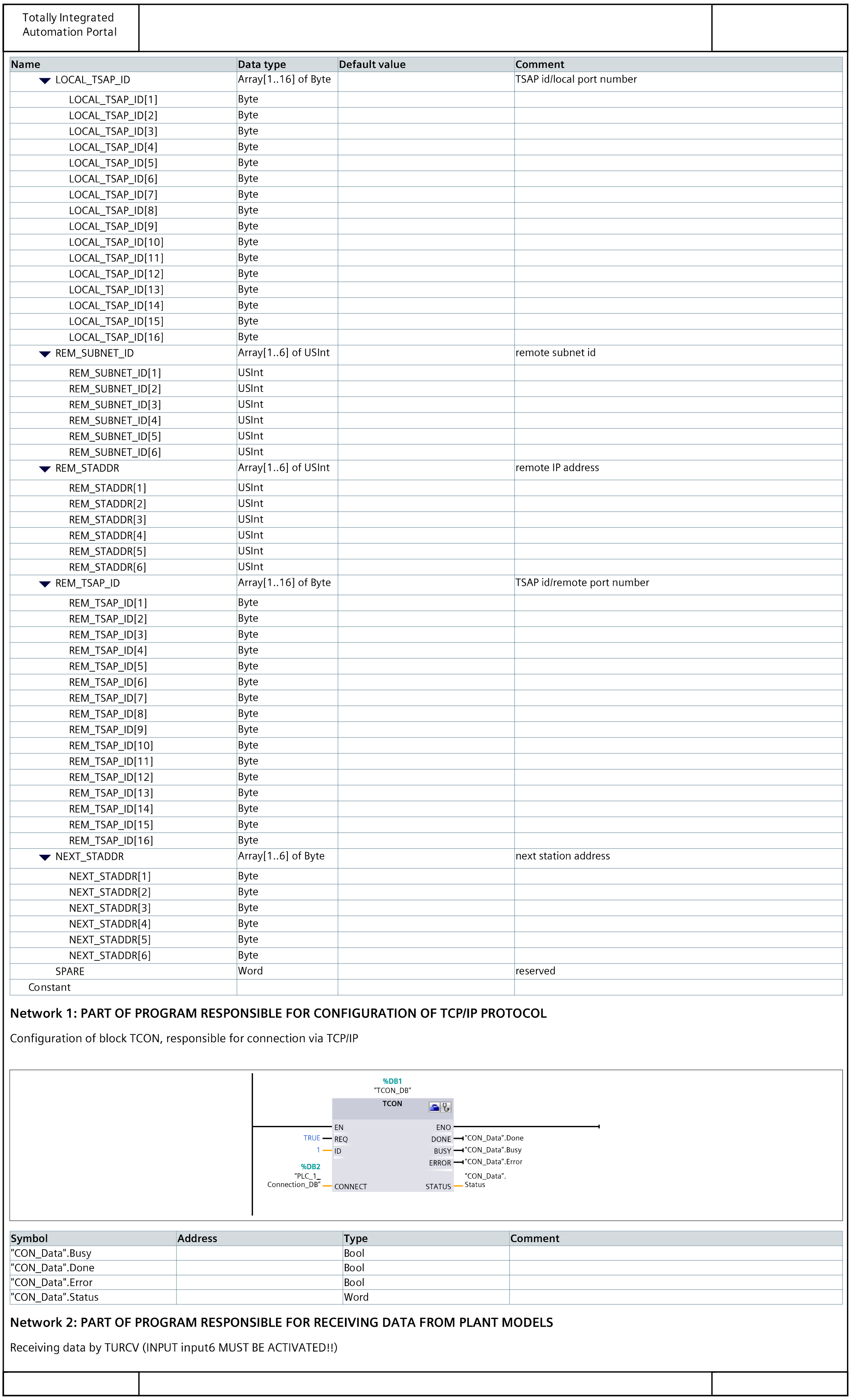

MOVE
EN
ENO
%MD146
"Kp"
IN
OUT1
"PID_Compact_2".sRet.r_Ctrl_Gain
OUT2
"PID_Compact_1".sRet.r_Ctrl_Gain
MOVE
EN
ENO
%MD154
"Ki"
IN
OUT1
"PID_Compact_2".sRet.r_Ctrl_Ti
OUT2
"PID_Compact_1".sRet.r_Ctrl_Ti
MOVE
EN
ENO
%MD158
"Kd"
IN
OUT1
"PID_Compact_2".sRet.r_Ctrl_Td
OUT2
"PID_Compact_1".sRet.r_Ctrl_Td

Symbol	Address	Type	Comment
"Kd"	%MD158	Real	
"Ki"	%MD154	Real	
"Kp"	%MD146	Real	
"PID_Compact_1".sRet.r_Ctrl_Gain		Real	actual proportional gain
"PID_Compact_1".sRet.r_Ctrl_Td		Real	actual derivative time
"PID_Compact_1".sRet.r_Ctrl_Ti		Real	actual integration time
"PID_Compact_2".sRet.r_Ctrl_Gain		Real	actual proportional gain
"PID_Compact_2".sRet.r_Ctrl_Td		Real	actual derivative time
"PID_Compact_2".sRet.r_Ctrl_Ti		Real	actual integration time

Network 4: PID block for ballast tank control model



Symbol	Address	Type	Comment
"SEND_Data".table_AO[14]	%DB7.DBD52	Real	
"VesselLevel"	%MD234	Real	
"VesselSetPoint"	%MD366	Real	



Totally Integrated
Automation Portal

Name	Data type	Default value	Comment
LOCAL_TSAP_ID	Array[1..16] of Byte		TSAP id/local port number
LOCAL_TSAP_ID[1]	Byte		
LOCAL_TSAP_ID[2]	Byte		
LOCAL_TSAP_ID[3]	Byte		
LOCAL_TSAP_ID[4]	Byte		
LOCAL_TSAP_ID[5]	Byte		
LOCAL_TSAP_ID[6]	Byte		
LOCAL_TSAP_ID[7]	Byte		
LOCAL_TSAP_ID[8]	Byte		
LOCAL_TSAP_ID[9]	Byte		
LOCAL_TSAP_ID[10]	Byte		
LOCAL_TSAP_ID[11]	Byte		
LOCAL_TSAP_ID[12]	Byte		
LOCAL_TSAP_ID[13]	Byte		
LOCAL_TSAP_ID[14]	Byte		
LOCAL_TSAP_ID[15]	Byte		
LOCAL_TSAP_ID[16]	Byte		
REM_SUBNET_ID	Array[1..6] of USInt		remote subnet id
REM_SUBNET_ID[1]	USInt		
REM_SUBNET_ID[2]	USInt		
REM_SUBNET_ID[3]	USInt		
REM_SUBNET_ID[4]	USInt		
REM_SUBNET_ID[5]	USInt		
REM_SUBNET_ID[6]	USInt		
REM_STADDR	Array[1..6] of USInt		remote IP address
REM_STADDR[1]	USInt		
REM_STADDR[2]	USInt		
REM_STADDR[3]	USInt		
REM_STADDR[4]	USInt		
REM_STADDR[5]	USInt		
REM_STADDR[6]	USInt		
REM_TSAP_ID	Array[1..16] of Byte		TSAP id/remote port number
REM_TSAP_ID[1]	Byte		
REM_TSAP_ID[2]	Byte		
REM_TSAP_ID[3]	Byte		
REM_TSAP_ID[4]	Byte		
REM_TSAP_ID[5]	Byte		
REM_TSAP_ID[6]	Byte		
REM_TSAP_ID[7]	Byte		
REM_TSAP_ID[8]	Byte		
REM_TSAP_ID[9]	Byte		
REM_TSAP_ID[10]	Byte		
REM_TSAP_ID[11]	Byte		
REM_TSAP_ID[12]	Byte		
REM_TSAP_ID[13]	Byte		
REM_TSAP_ID[14]	Byte		
REM_TSAP_ID[15]	Byte		
REM_TSAP_ID[16]	Byte		
NEXT_STADDR	Array[1..6] of Byte		next station address
NEXT_STADDR[1]	Byte		
NEXT_STADDR[2]	Byte		
NEXT_STADDR[3]	Byte		
NEXT_STADDR[4]	Byte		
NEXT_STADDR[5]	Byte		
NEXT_STADDR[6]	Byte		
SPARE	Word		reserved
Constant			

Network 1: PART OF PROGRAM RESPONSIBLE FOR CONFIGURATION OF TCP/IP PROTOCOL

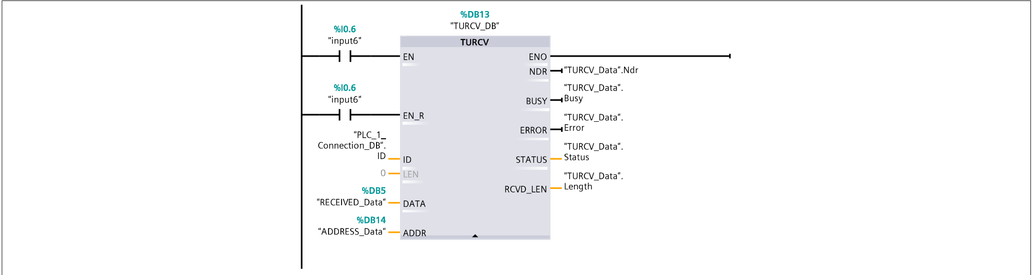
Configuration of block TCON, responsible for connection via TCP/IP

Symbol	Address	Type	Comment
"CON_Data".Busy		Bool	
"CON_Data".Done		Bool	
"CON_Data".Error		Bool	
"CON_Data".Status		Word	

Network 2: PART OF PROGRAM RESPONSIBLE FOR RECEIVING DATA FROM PLANT MODELS

Receiving data by TURCV (INPUT input6 MUST BE ACTIVATED!!)

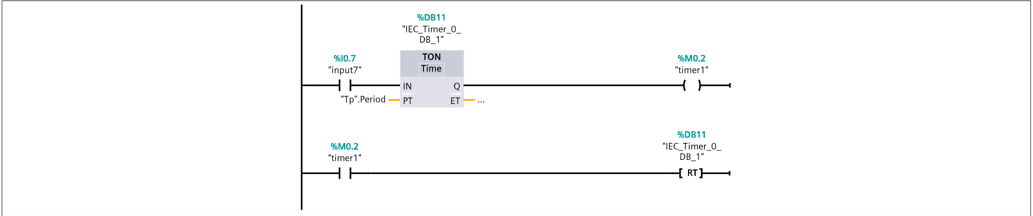
Totally Integrated Automation Portal



Symbol	Address	Type	Comment
"input6"	%I0.6	Bool	
"PLC_1_Connection_DB".ID		CONN_OUC	reference to the connection
"TURCV_Data".Busy		Bool	
"TURCV_Data".Error		Bool	
"TURCV_Data".Length		UInt	
"TURCV_Data".Ndr		Bool	
"TURCV_Data".Status		Word	

Network 3: PART OF PROGRAM RESPONSIBLE FOR SAMPLING

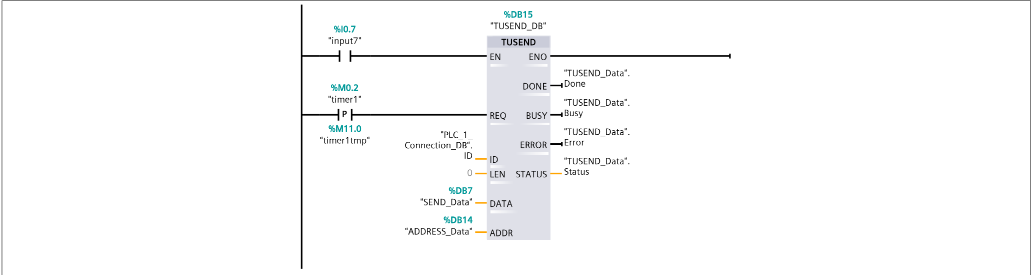
Activation of TUSEND controlled by timers (INPUT input7 MUST BE ACTIVATED!!).



Symbol	Address	Type	Comment
"input7"	%I0.7	Bool	
"timer1"	%M0.2	Bool	
"Tp".Period		Time	

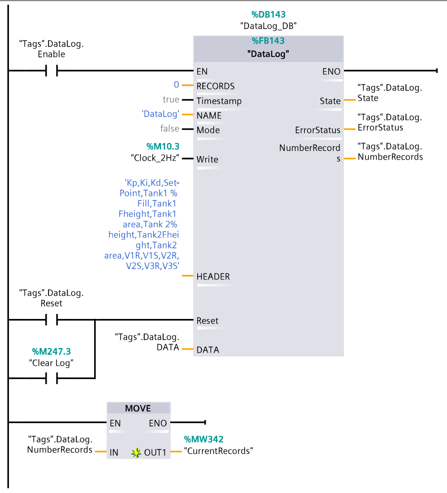
Network 4: PART OF PROGRAM RESPONSIBLE FOR SENDING DATA

Block TSEND_C - cyclic sending data (INPUT input7 MUST BE ACTIVATED!!).



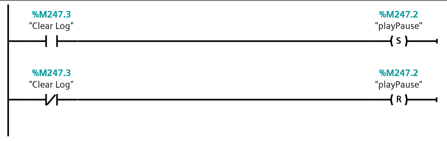
Symbol	Address	Type	Comment
"input7"	%I0.7	Bool	
"PLC_1_Connection_DB".ID		CONN_OUC	reference to the connection
"timer1"	%M0.2	Bool	
"timer1tmp"	%M11.0	Bool	
"TUSEND_Data".Busy		Bool	
"TUSEND_Data".Done		Bool	
"TUSEND_Data".Error		Bool	
"TUSEND_Data".Status		Word	

Network 5: Data logging

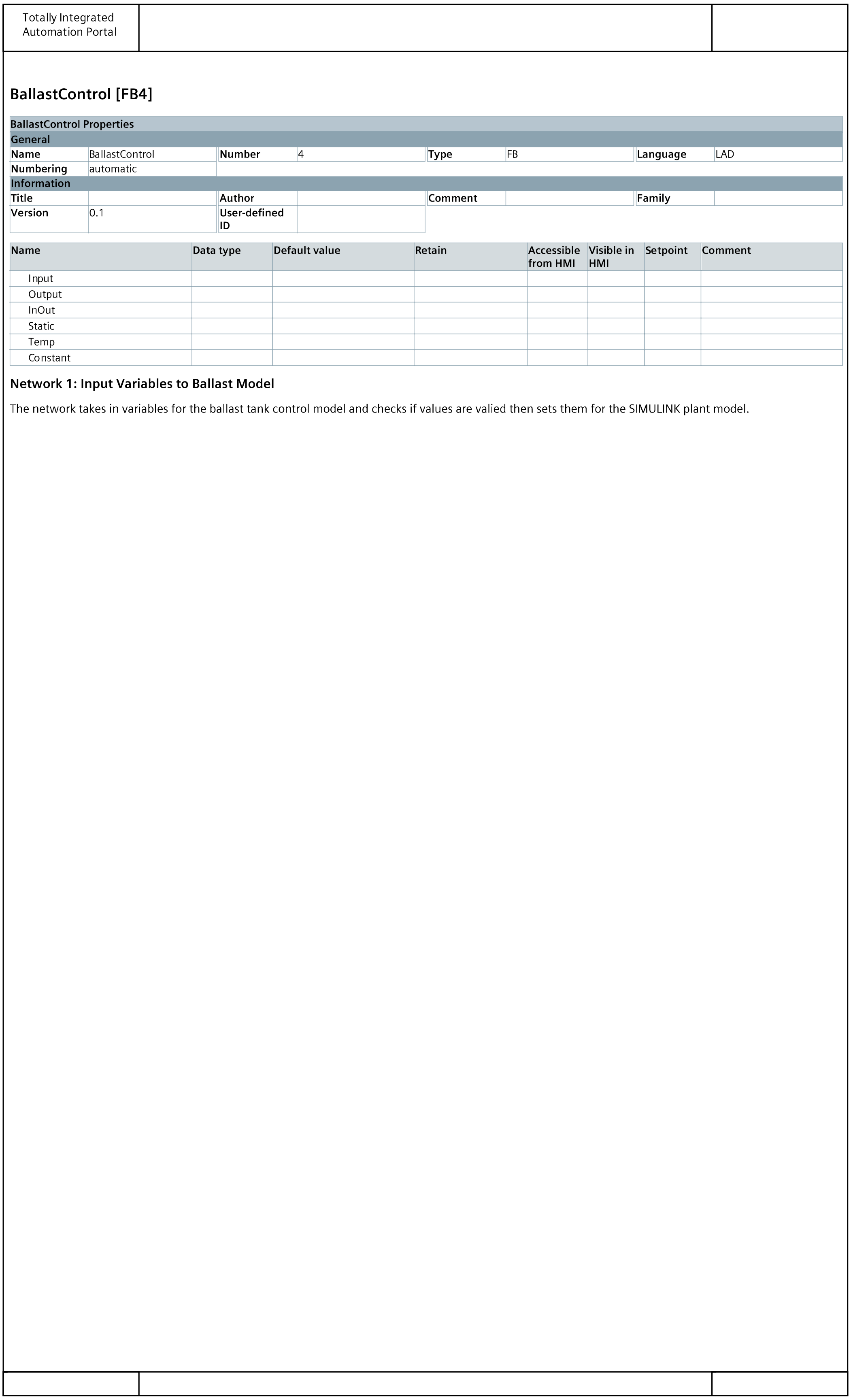


Symbol	Address	Type	Comment
"Clear Log"	%M247.3	Bool	
"Clock_2Hz"	%M10.3	Bool	
"CurrentRecords"	%MW342	Int	
"Tags".DataLog.DATA		Struct	
"Tags".DataLog.Enable		Bool	
"Tags".DataLog.ErrorStatus		Word	
"Tags".DataLog.NumberRecords		UDInt	
"Tags".DataLog.Reset		Bool	
"Tags".DataLog.State		USInt	

Network 6: logVariables



Symbol	Address	Type	Comment
"Clear Log"	%M247.3	Bool	
"playPause"	%M247.2	Bool	



Totally Integrated
Automation Portal

BallastControl [FB4]

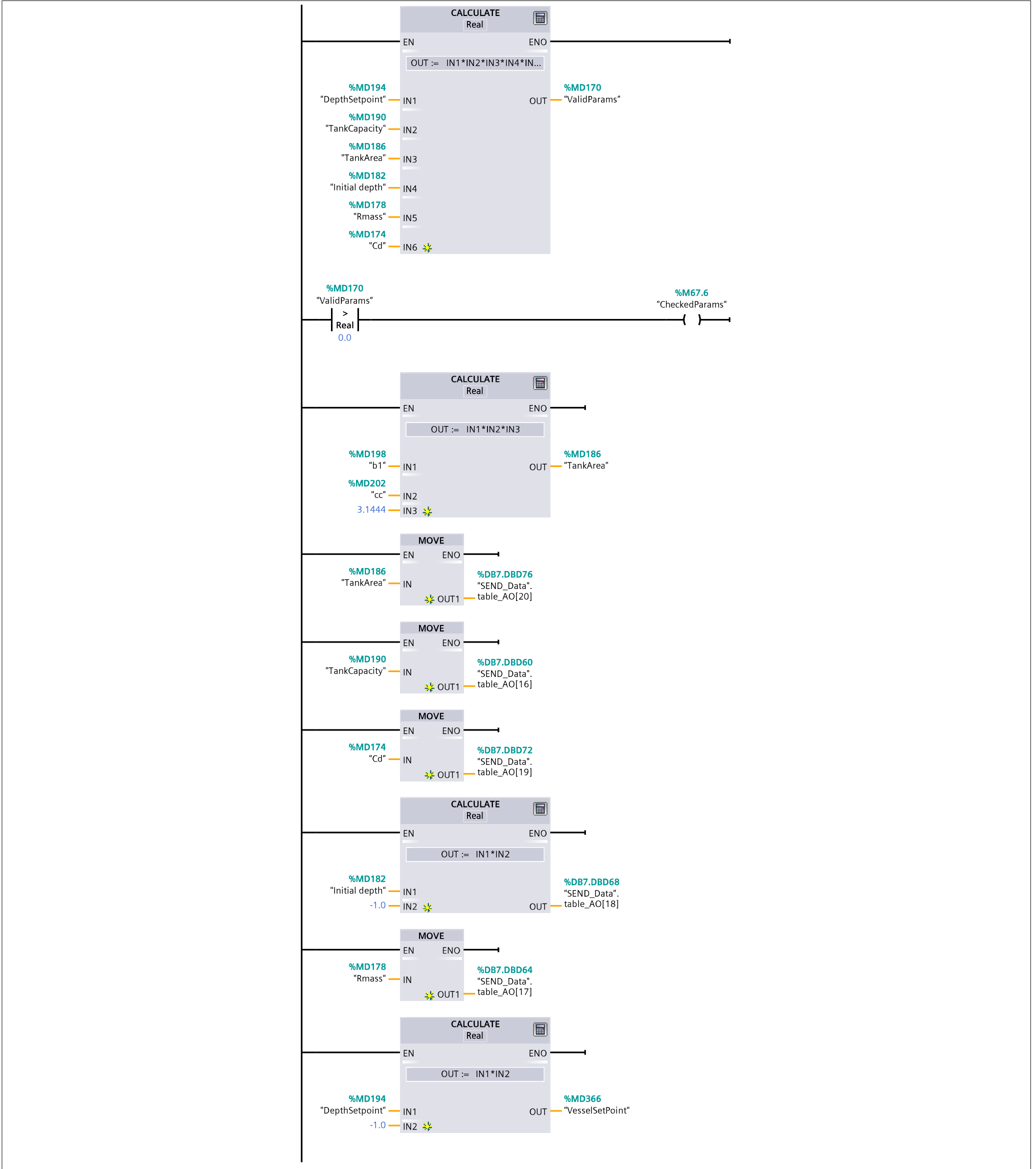
BallastControl Properties							
General							
Name	BallastControl	**Number**	4	**Type**	FB	**Language**	LAD
Numbering	automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
Input							
Output							
InOut							
Static							
Temp							
Constant							

Network 1: Input Variables to Ballast Model

The network takes in variables for the ballast tank control model and checks if values are valied then sets them for the SIMULINK plant model.

Totally Integrated Automation Portal

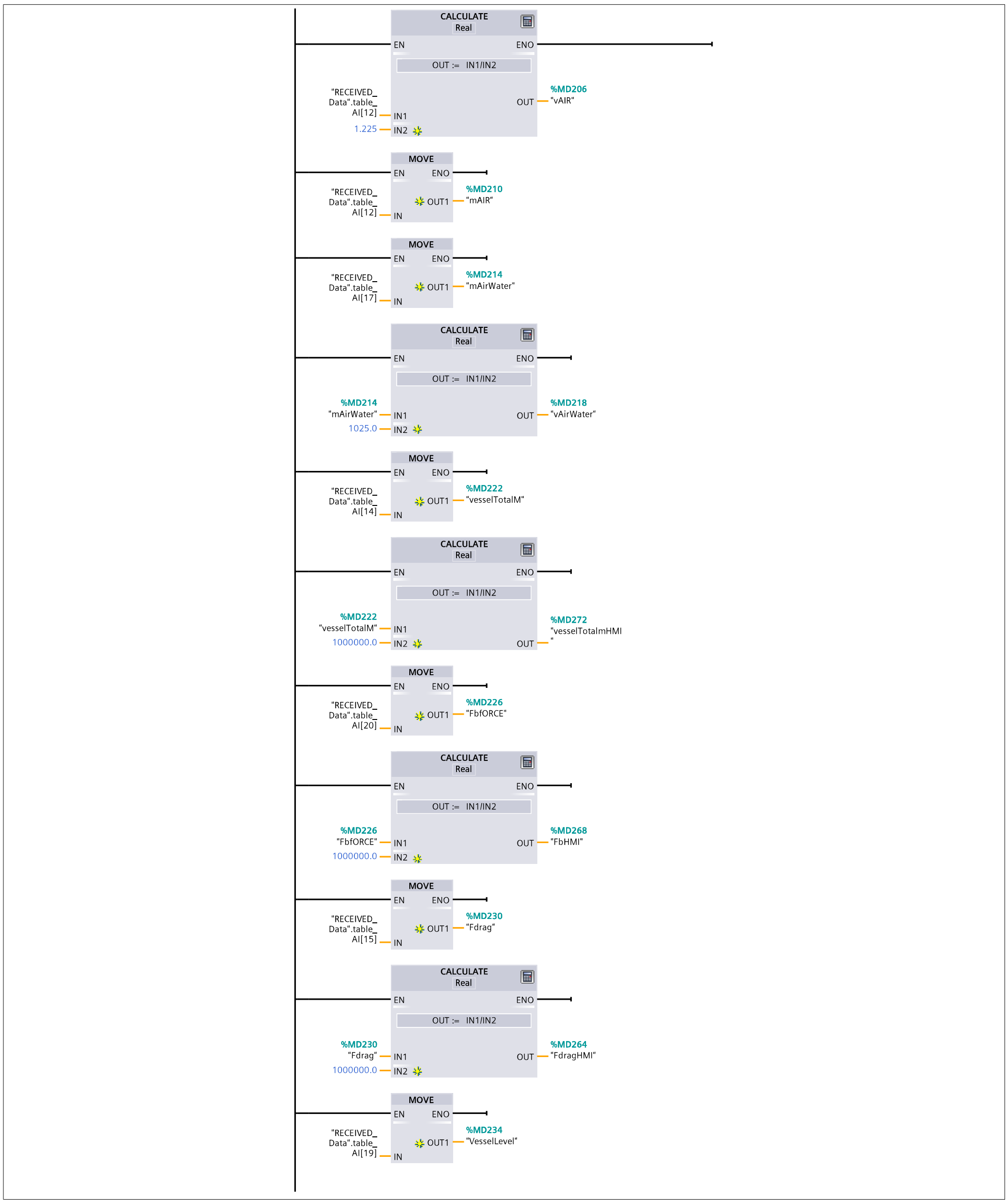


Symbol	Address	Type	Comment
"b1"	%MD198	Real	
"cc"	%MD202	Real	
"Cd"	%MD174	Real	
"CheckedParams"	%M67.6	Bool	
"DepthSetpoint"	%MD194	Real	
"Initial depth"	%MD182	Real	
"Rmass"	%MD178	Real	
"SEND_Data".table_AO[16]	%DB7.DBD60	Real	
"SEND_Data".table_AO[17]	%DB7.DBD64	Real	
"SEND_Data".table_AO[18]	%DB7.DBD68	Real	
"SEND_Data".table_AO[19]	%DB7.DBD72	Real	
"SEND_Data".table_AO[20]	%DB7.DBD76	Real	
"TankArea"	%MD186	Real	
"TankCapacity"	%MD190	Real	
"ValidParams"	%MD170	Real	
"VesselSetPoint"	%MD366	Real	

Network 2: Output Variable from ballast model

The response and different variables of the submarine model are received in this network and scaled so that they are displayed on the platform SCADA

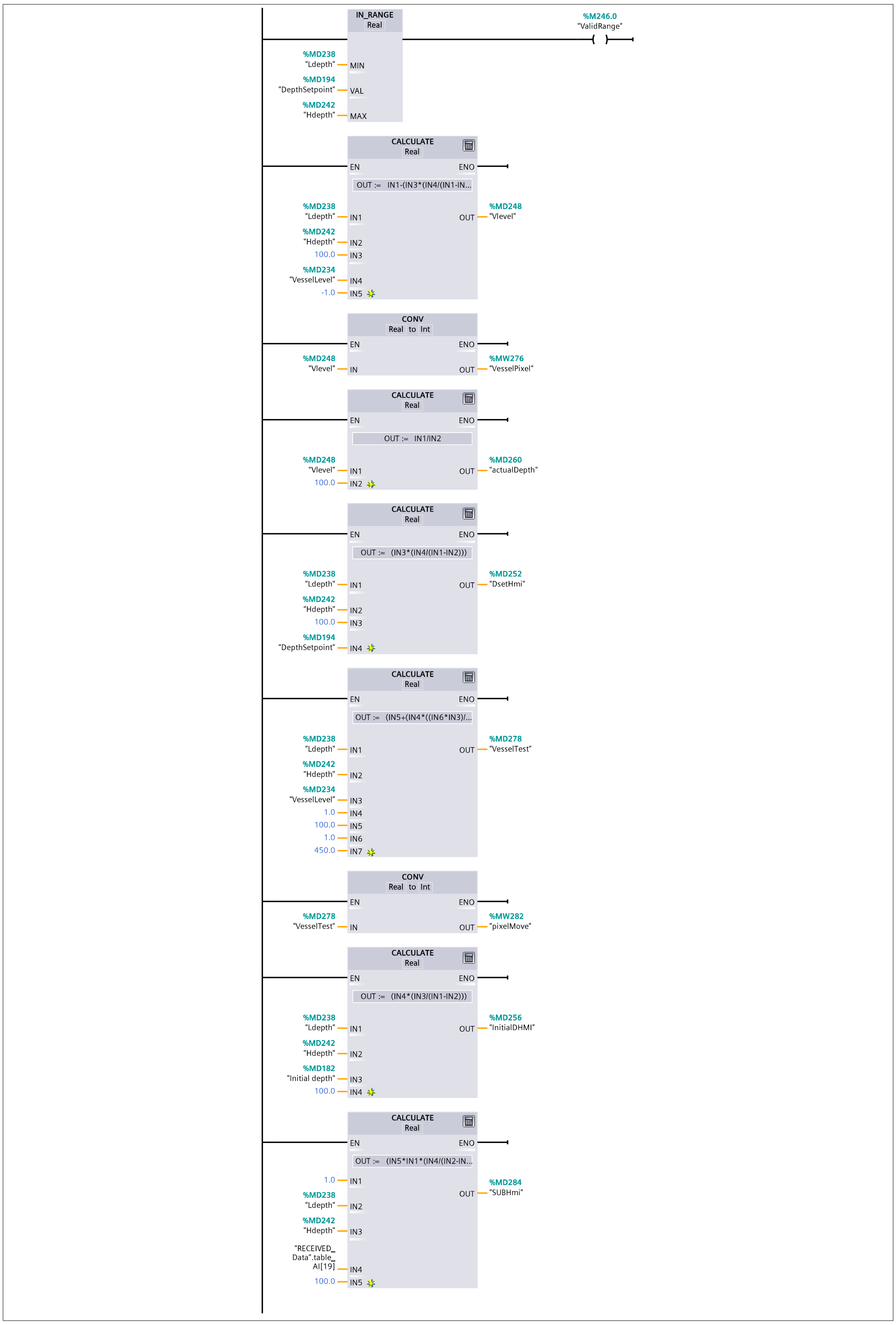
Totally Integrated Automation Portal



Symbol	Address	Type	Comment
"FbfORCE"	%MD226	Real	
"FbHMI"	%MD268	Real	
"Fdrag"	%MD230	Real	
"FdragHMI"	%MD264	Real	
"mAIR"	%MD210	Real	
"mAirWater"	%MD214	Real	
"RECEIVED_Data".table_AI[12]		Real	
"RECEIVED_Data".table_AI[14]		Real	
"RECEIVED_Data".table_AI[15]		Real	
"RECEIVED_Data".table_AI[17]		Real	
"RECEIVED_Data".table_AI[19]		Real	
"RECEIVED_Data".table_AI[20]		Real	
"vAIR"	%MD206	Real	
"vAirWater"	%MD218	Real	
"VesselLevel"	%MD234	Real	
"vesselTotalM"	%MD222	Real	
"vesselTotalmHMI"	%MD272	Real	

Network 3: SCADA representation of Submarine

The network calculates the depth of the submarine relative to (depth set-point, the region where vessel is located). It scales the position of the submarine such that it can be seen hovering on the SCADA emulating the ascending or descending of a submarine.

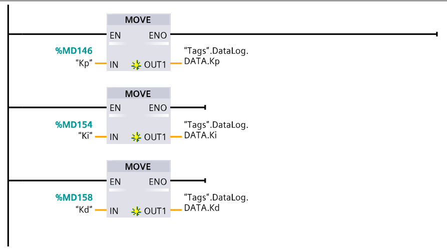


PID Control [FB3]

PID Control Properties							
General							
Name	PID Control	**Number**	3	**Type**	FB	**Language**	LAD
Numbering	automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
Input							
Output							
InOut							
Static							
Temp							
Constant							

Network 1: Assigns PID control paramters onto the datalog file



Symbol	Address	Type	Comment
"Kd"	%MD158	Real	
"Ki"	%MD154	Real	
"Kp"	%MD146	Real	
"Tags".DataLog.DATA.Kd		Real	
"Tags".DataLog.DATA.Ki		Real	
"Tags".DataLog.DATA.Kp		Real	

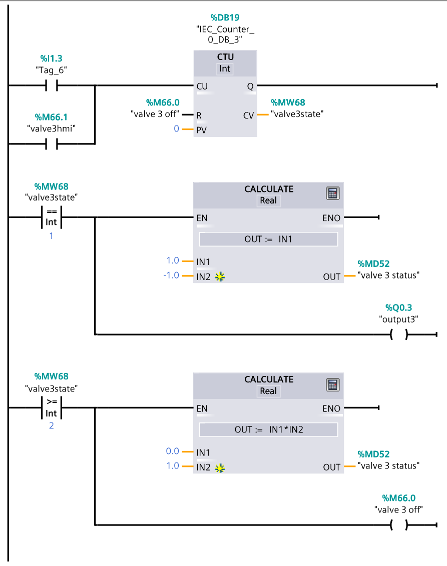
Totally Integrated
Automation Portal

TankCalc [FB2]

TankCalc Properties							
General							
Name	TankCalc	**Number**	2	**Type**	FB	**Language**	LAD
Numbering	automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
Input							
Output							
InOut							
Static							
Temp							
Constant							

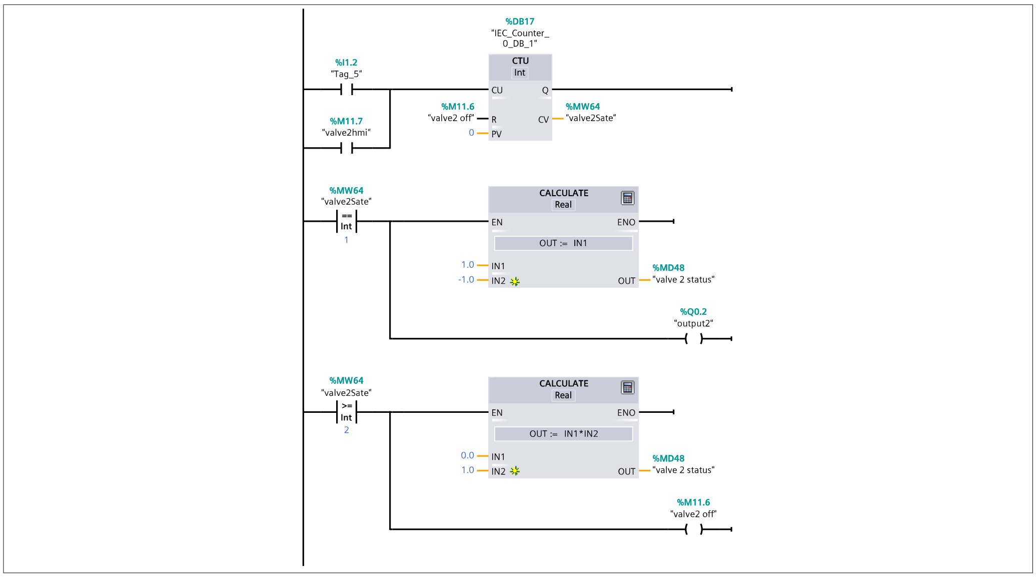
Network 1: Valve 3 Control via SCADA or Control Box



Symbol	Address	Type	Comment
"output3"	%Q0.3	Bool	
"Tag_6"	%I1.3	Bool	
"valve3hmi"	%M66.1	Bool	
"valve3state"	%MW68	Int	
"valve 3 off"	%M66.0	Bool	
"valve 3 status"	%MD52	Real	

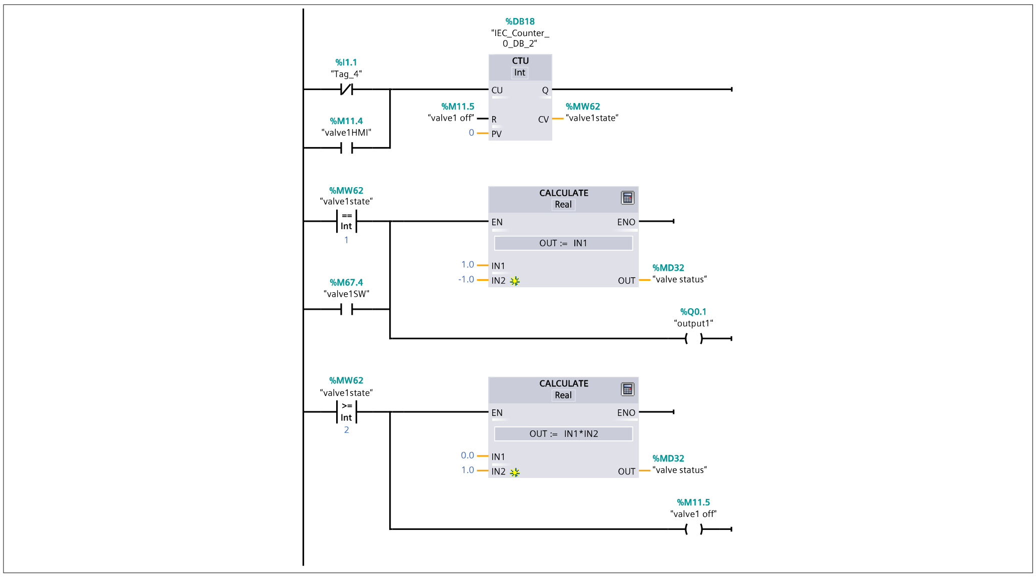
Network 2: Valve 2 Control via SCADA or Control Box

Totally Integrated Automation Portal



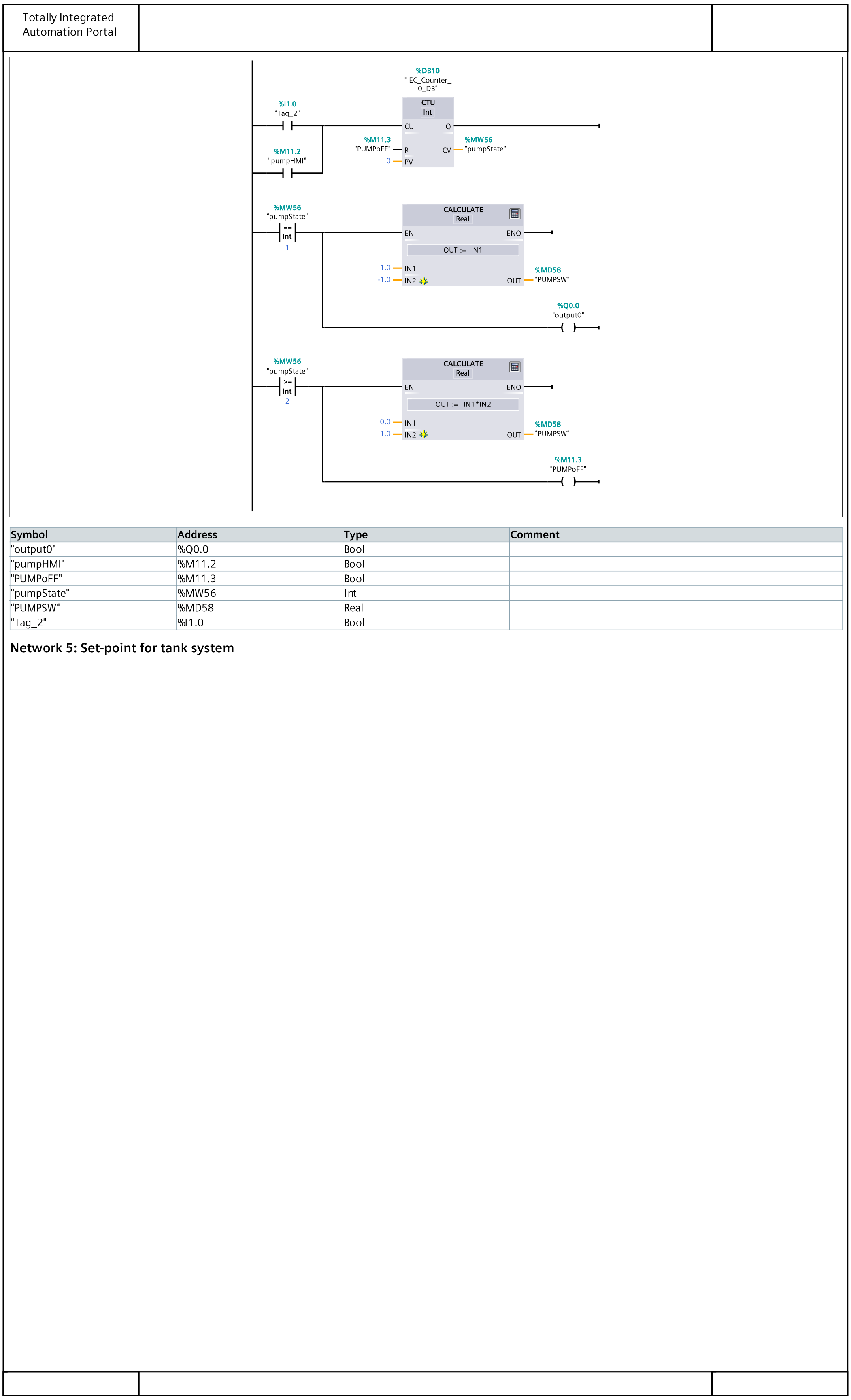
Symbol	Address	Type	Comment
"output2"	%Q0.2	Bool	
"Tag_5"	%I1.2	Bool	
"valve2 off"	%M11.6	Bool	
"valve2hmi"	%M11.7	Bool	
"valve2Sate"	%MW64	Int	
"valve 2 status"	%MD48	Real	

Network 3: Valve 1 Control via SCADA or Control Box



Symbol	Address	Type	Comment
"output1"	%Q0.1	Bool	
"Tag_4"	%I1.1	Bool	
"valve1 off"	%M11.5	Bool	
"valve1HMI"	%M11.4	Bool	
"valve1state"	%MW62	Int	
"valve1SW"	%M67.4	Bool	
"valve status"	%MD32	Real	

Network 4: PUMP Control via SCADA or Control Box

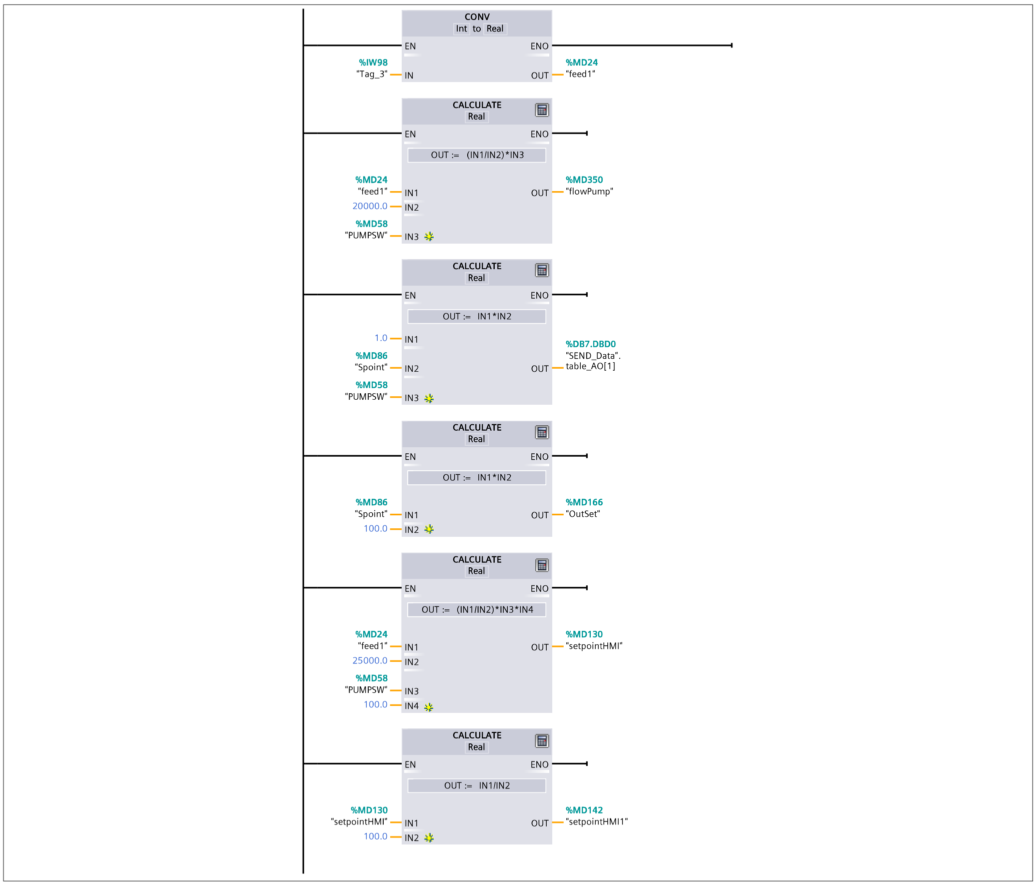


Totally Integrated Automation Portal

Symbol	Address	Type	Comment
"output0"	%Q0.0	Bool	
"pumpHMI"	%M11.2	Bool	
"PUMPoFF"	%M11.3	Bool	
"pumpState"	%MW56	Int	
"PUMPSW"	%MD58	Real	
"Tag_2"	%I1.0	Bool	

Network 5: Set-point for tank system

Totally Integrated Automation Portal

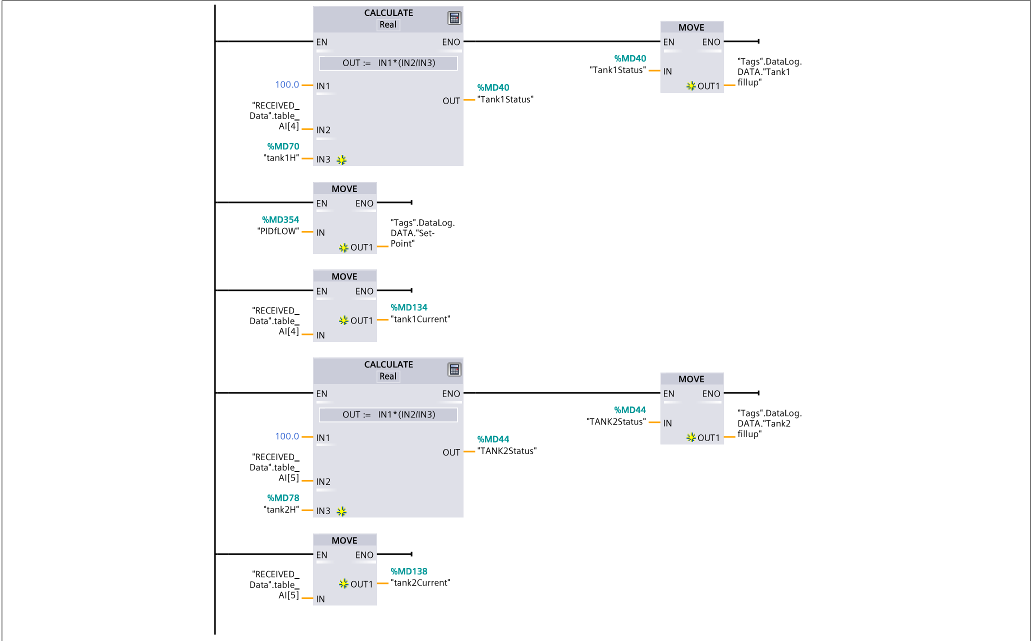


Symbol	Address	Type	Comment
"feed1"	%MD24	Real	
"flowPump"	%MD350	Real	
"OutSet"	%MD166	Real	
"PUMPSW"	%MD58	Real	
"SEND_Data".table_AO[1]	%DB7.DBD0	Real	
"setpointHMI1"	%MD142	Real	
"setpointHMI"	%MD130	Real	
"Spoint"	%MD86	Real	
"Tag_3"	%IW98	Word	

Network 6: Response values from tank system

In this network the response values of the tank system are receieved and then assigned to variables that are written on the datalog file and displayed on the platform SCADA

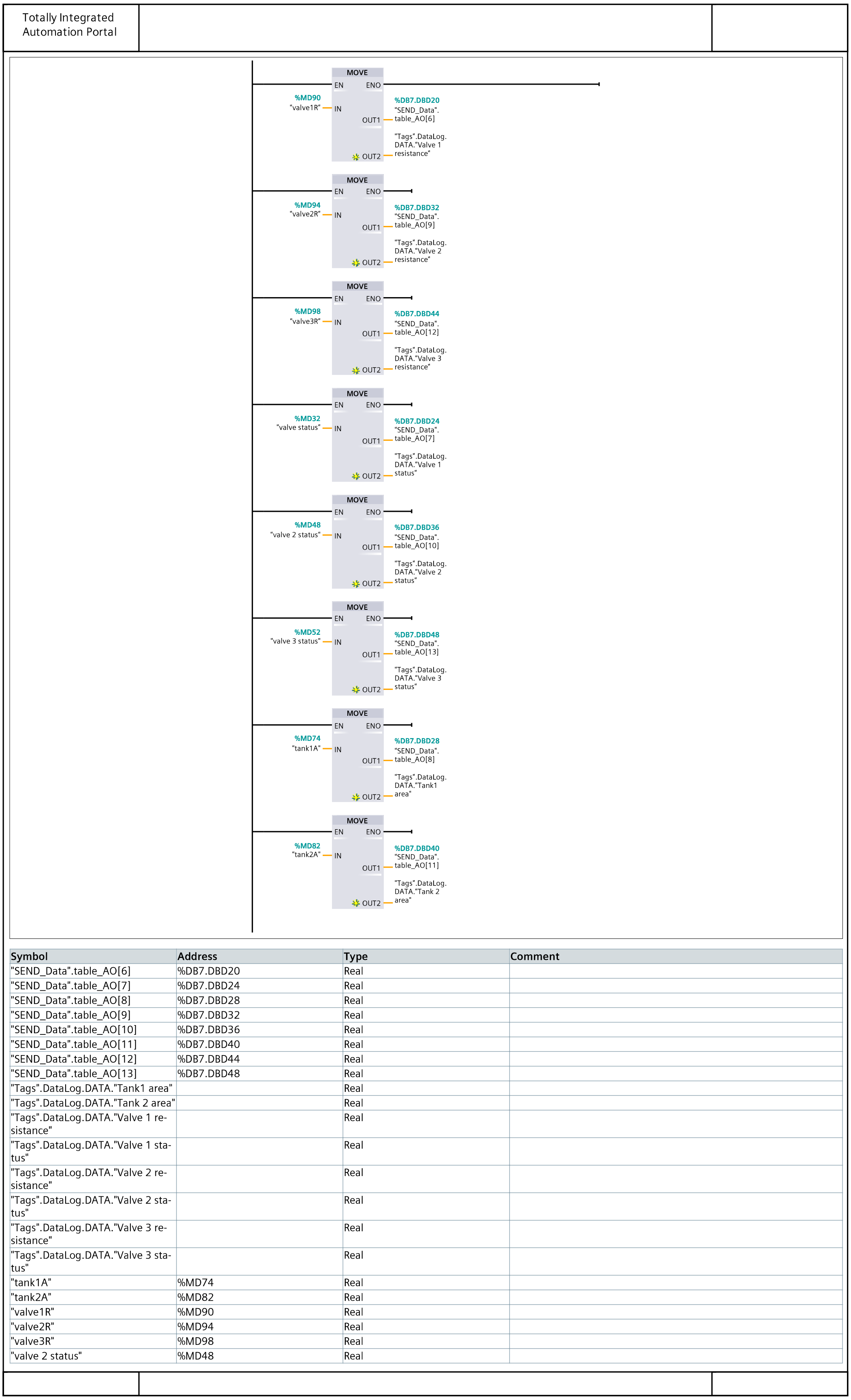
Totally Integrated
Automation Portal



Symbol	Address	Type	Comment
"PIDfLOW"	%MD354	Real	
"RECEIVED_Data".table_AI[4]		Real	
"RECEIVED_Data".table_AI[5]		Real	
"Tags".DataLog.DATA."Set-Point"		Real	
"Tags".DataLog.DATA."Tank1 fill-up"		Real	
"Tags".DataLog.DATA."Tank2 fill-up"		Real	
"tank1Current"	%MD134	Real	
"tank1H"	%MD70	Real	
"Tank1Status"	%MD40	Real	
"tank2Current"	%MD138	Real	
"tank2H"	%MD78	Real	
"TANK2Status"	%MD44	Real	

Network 7: Network updates valve conditions and other plant parameters to plant model and assigns them to log file

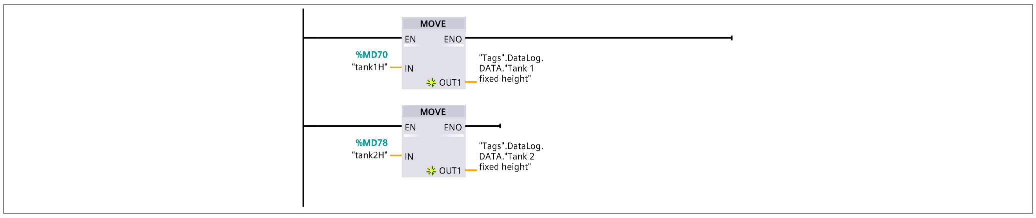
Totally Integrated Automation Portal



Symbol	Address	Type	Comment
"SEND_Data".table_AO[6]	%DB7.DBD20	Real	
"SEND_Data".table_AO[7]	%DB7.DBD24	Real	
"SEND_Data".table_AO[8]	%DB7.DBD28	Real	
"SEND_Data".table_AO[9]	%DB7.DBD32	Real	
"SEND_Data".table_AO[10]	%DB7.DBD36	Real	
"SEND_Data".table_AO[11]	%DB7.DBD40	Real	
"SEND_Data".table_AO[12]	%DB7.DBD44	Real	
"SEND_Data".table_AO[13]	%DB7.DBD48	Real	
"Tags".DataLog.DATA."Tank1 area"		Real	
"Tags".DataLog.DATA."Tank 2 area"		Real	
"Tags".DataLog.DATA."Valve 1 resistance"		Real	
"Tags".DataLog.DATA."Valve 1 status"		Real	
"Tags".DataLog.DATA."Valve 2 resistance"		Real	
"Tags".DataLog.DATA."Valve 2 status"		Real	
"Tags".DataLog.DATA."Valve 3 resistance"		Real	
"Tags".DataLog.DATA."Valve 3 status"		Real	
"tank1A"	%MD74	Real	
"tank2A"	%MD82	Real	
"valve1R"	%MD90	Real	
"valve2R"	%MD94	Real	
"valve3R"	%MD98	Real	
"valve 2 status"	%MD48	Real	

Symbol	Address	Type	Comment
"valve 3 status"	%MD52	Real	
"valve status"	%MD32	Real	

Network 8: Data assignment of tank level values to log file



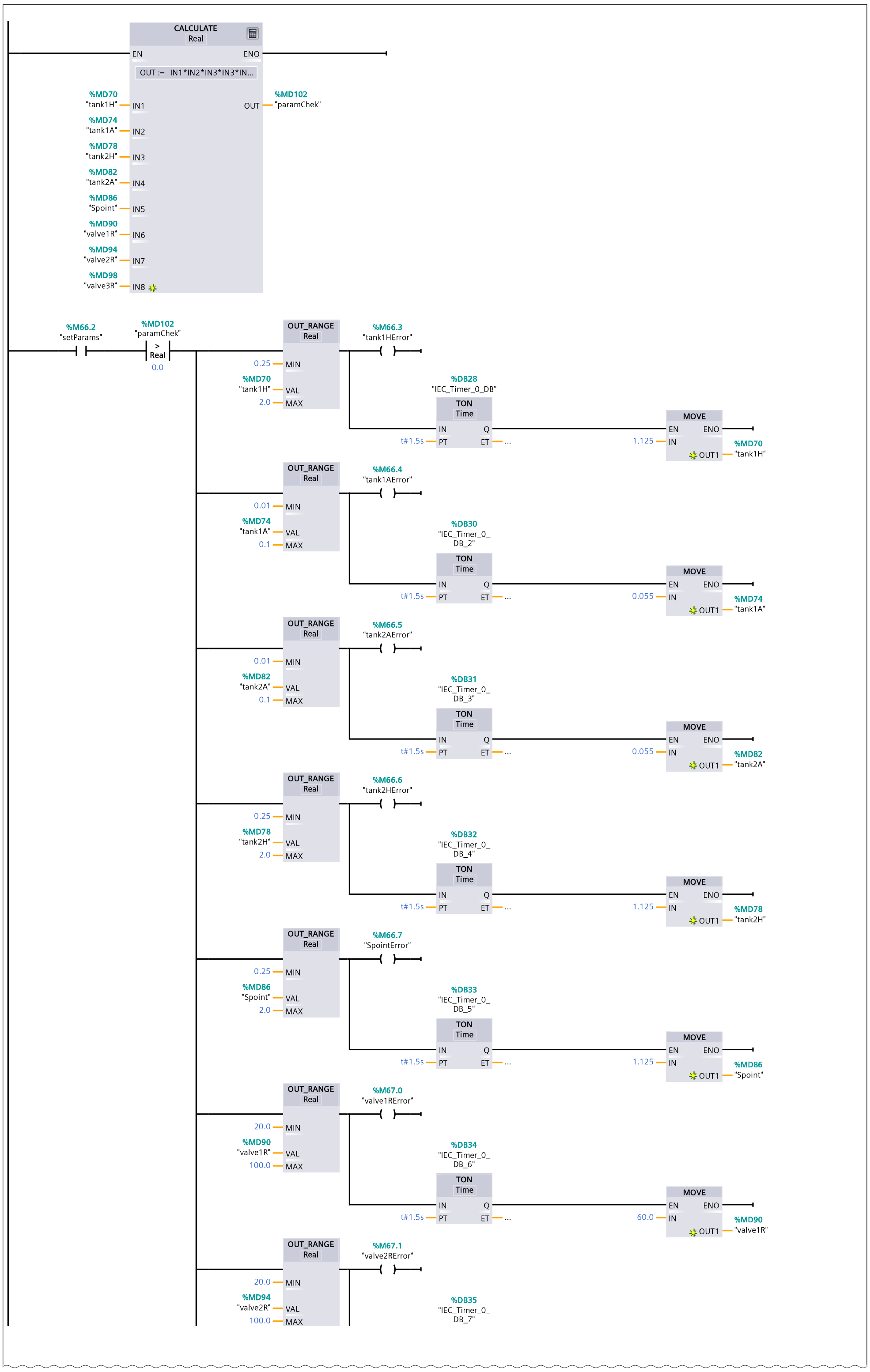
Symbol	Address	Type	Comment
"Tags".DataLog.DATA."Tank 1 fixed height"		Real	
"Tags".DataLog.DATA."Tank 2 fixed height"		Real	
"tank1H"	%MD70	Real	
"tank2H"	%MD78	Real	

Network 9: Variable checker

When user is working on the platform, if they enter values that are out of range or the plant model cannot it handle, it warns the user and sets that parameter to a default value that will ensure the system does not fail.

Totally Integrated
Automation Portal

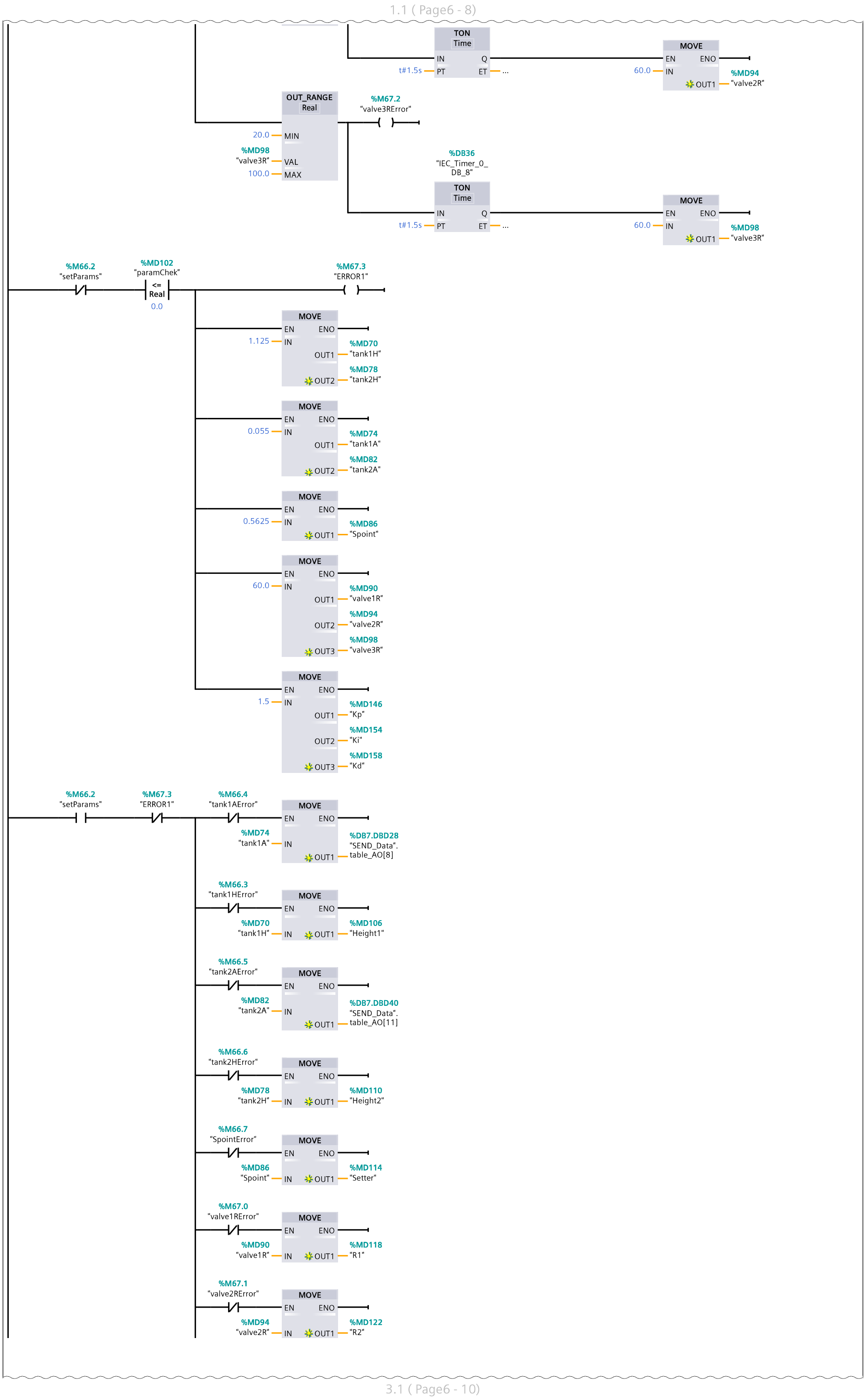
Network 9: Variable checker (1.1 / 3.1)



2.1 (Page6 - 9)

Totally Integrated Automation Portal

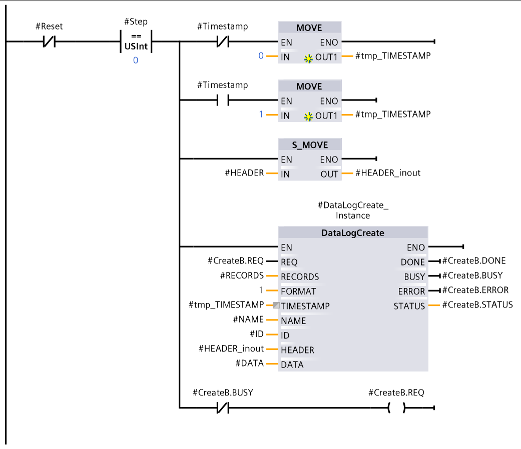
Network 9: Variable checker (2.1 / 3.1)



Totally Integrated
Automation Portal

Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
STATUS	Word	0	Non-retain	True	True	False	
▼ InOut							
ID	DWord	0	Non-retain	True	True	False	
Static							
▼ DataLogWrite_Instance	DataLogWrite			True	True	False	
▼ Input							
REQ	Bool	false	Non-retain	True	True	False	
▼ Output							
DONE	Bool	false	Non-retain	True	True	False	
BUSY	Bool	false	Non-retain	True	True	False	
ERROR	Bool	false	Non-retain	True	True	False	
STATUS	Word	0	Non-retain	True	True	False	
▼ InOut							
ID	DWord	0	Non-retain	True	True	False	
Static							
tmp_TIMESTAMP	USInt	0	Non-retain	False	False	False	
tmp_MODE	USInt	0	Non-retain	False	False	False	
Temp							
Constant							

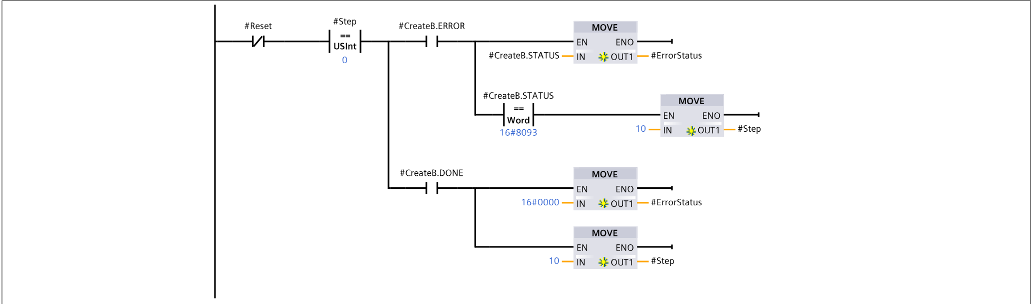
Network 1: Command 0: Create



Symbol	Address	Type	Comment
#CreateB.BUSY		Bool	
#CreateB.DONE		Bool	
#CreateB.ERROR		Bool	
#CreateB.REQ		Bool	
#CreateB.STATUS		Word	
#DATA		Variant	
#DataLogCreate_Instance		Multi_SFB	
#HEADER		String	
#HEADER_inout		String	
#ID		UDInt	
#NAME		String	
#RECORDS		UDInt	
#Reset		Bool	
#Step		USInt	
#Timestamp		Bool	
#tmp_TIMESTAMP		USInt	

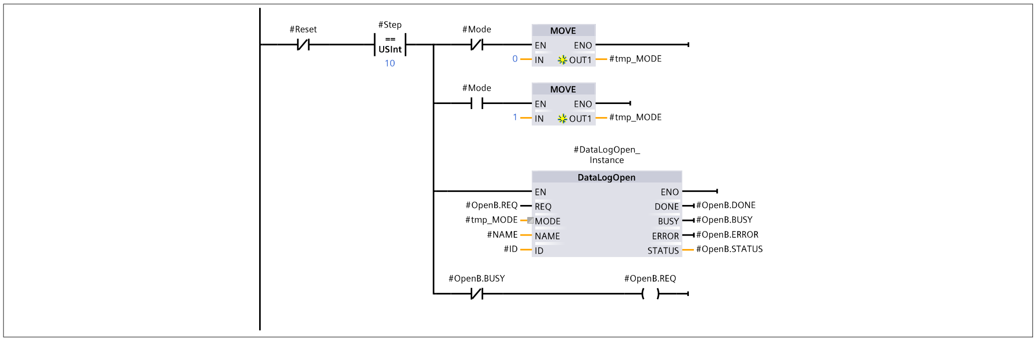
Network 2: Transition 0

8093=Data log already exists.



Symbol	Address	Type	Comment
#CreateB.DONE		Bool	
#CreateB.ERROR		Bool	
#CreateB.STATUS		Word	
#ErrorStatus		Word	
#Reset		Bool	
#Step		USInt	

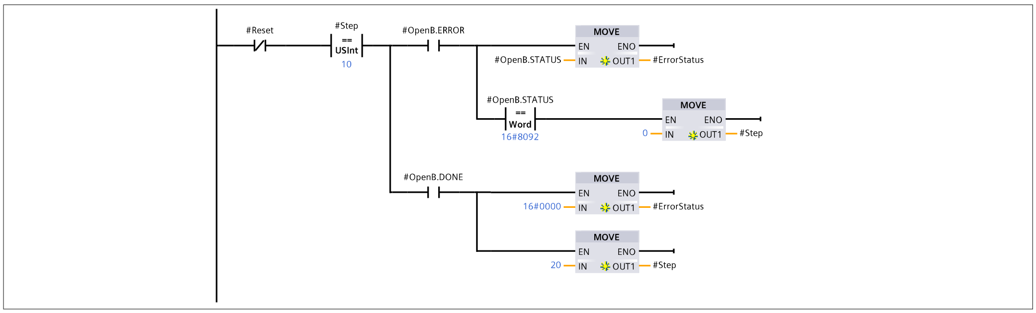
Network 3: Command 10: Open



Symbol	Address	Type	Comment
#DataLogOpen_Instance		Multi_SFB	
#ID		UDInt	
#Mode		Bool	
#NAME		String	
#OpenB.BUSY		Bool	
#OpenB.DONE		Bool	
#OpenB.ERROR		Bool	
#OpenB.REQ		Bool	
#OpenB.STATUS		Word	
#Reset		Bool	
#Step		USInt	
#tmp_MODE		USInt	

Network 4: Transition 10

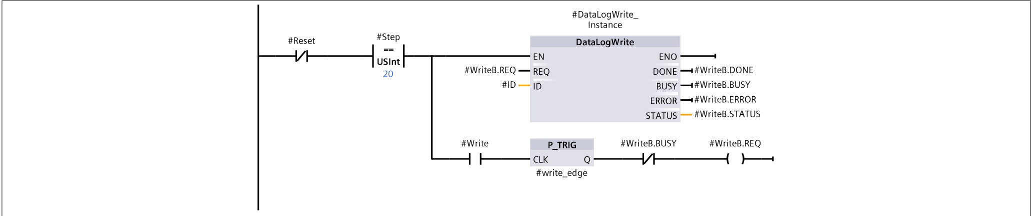
8092=Data log does not exist



Symbol	Address	Type	Comment
#ErrorStatus		Word	
#OpenB.DONE		Bool	
#OpenB.ERROR		Bool	
#OpenB.STATUS		Word	
#Reset		Bool	

Symbol	Address	Type	Comment
#Step		USInt	

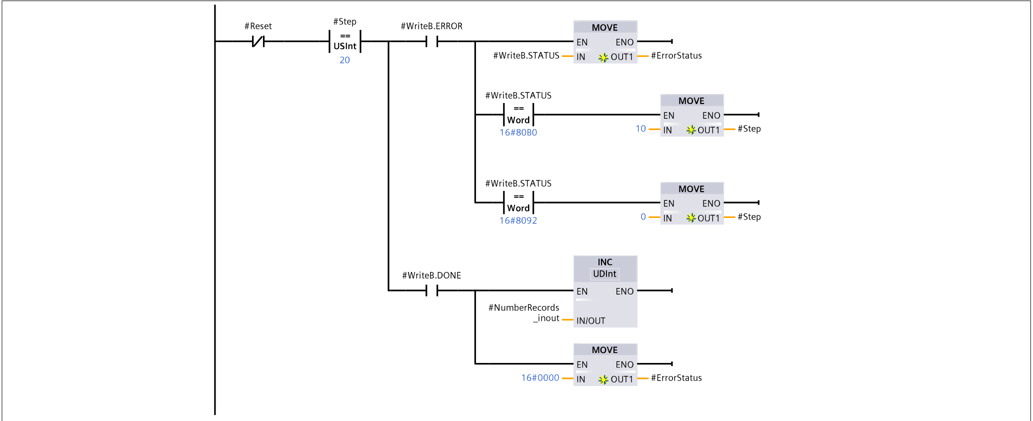
Network 5: Command 20: Write



Symbol	Address	Type	Comment
#DataLogWrite_Instance		Multi_SFB	
#ID		UDInt	
#Reset		Bool	
#Step		USInt	
#Write		Bool	
#write_edge		Bool	
#WriteB.BUSY		Bool	
#WriteB.DONE		Bool	
#WriteB.ERROR		Bool	
#WriteB.REQ		Bool	
#WriteB.STATUS		Word	

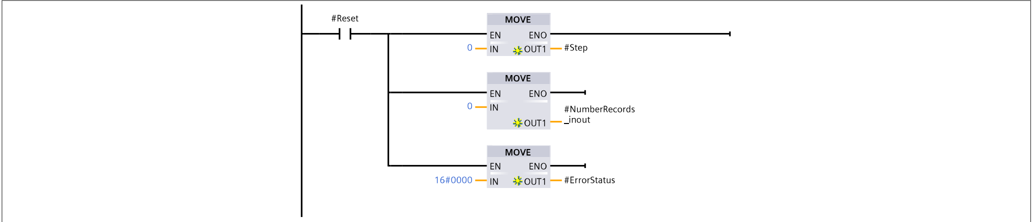
Network 6: Transition 20

80B0=Data log not open.
8092=Data log does not exist.



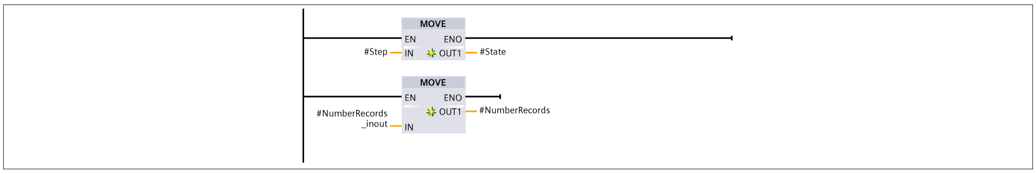
Symbol	Address	Type	Comment
#ErrorStatus		Word	
#NumberRecords_inout		UDInt	
#Reset		Bool	
#Step		USInt	
#WriteB.DONE		Bool	
#WriteB.ERROR		Bool	
#WriteB.STATUS		Word	

Network 7: Reset



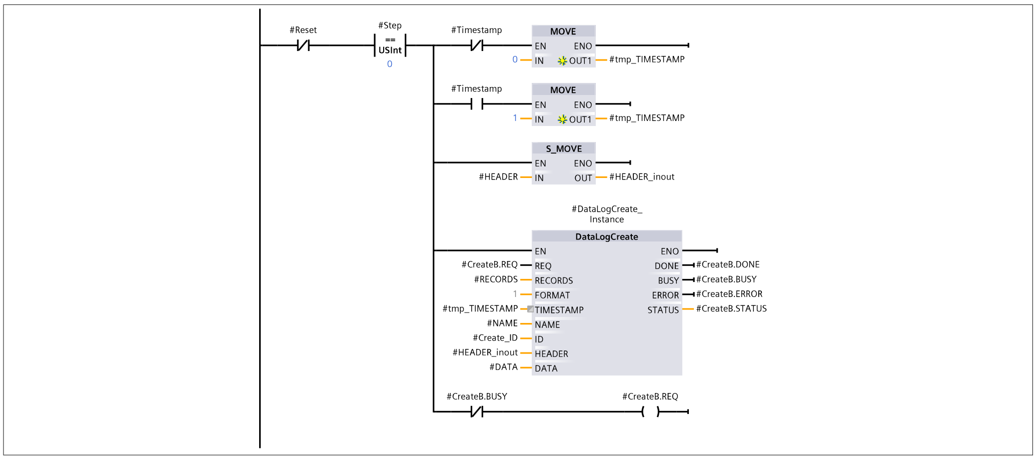
Symbol	Address	Type	Comment
#ErrorStatus		Word	
#NumberRecords_inout		UDInt	
#Reset		Bool	
#Step		USInt	

Network 8: output



Symbol	Address	Type	Comment
#NumberRecords		UDInt	
#NumberRecords_inout		UDInt	
#State		USInt	
#Step		USInt	

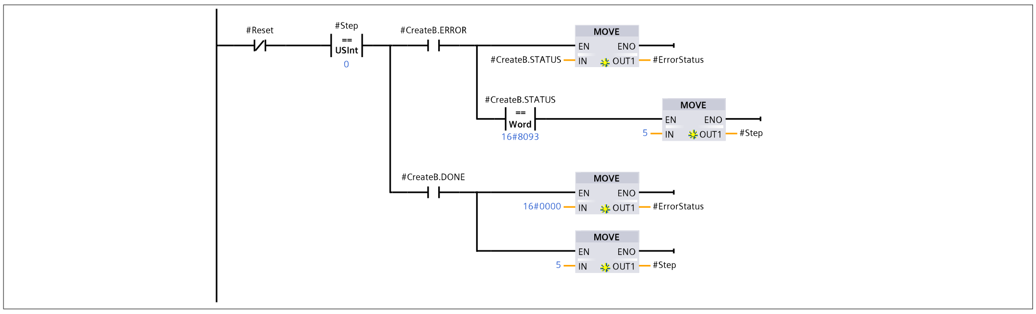
Totally Integrated
Automation Portal



Symbol	Address	Type	Comment
#Create_ID		UDInt	
#CreateB.BUSY		Bool	
#CreateB.DONE		Bool	
#CreateB.ERROR		Bool	
#CreateB.REQ		Bool	
#CreateB.STATUS		Word	
#DATA		Variant	
#DataLogCreate_Instance		Multi_SFB	
#HEADER		String	
#HEADER_inout		String	
#NAME		String	
#RECORDS		UDInt	
#Reset		Bool	
#Step		USInt	
#Timestamp		Bool	
#tmp_TIMESTAMP		USInt	

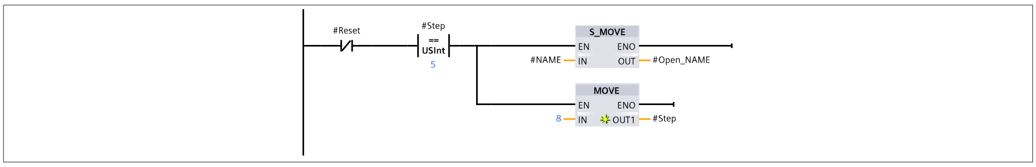
Network 2: Transition 0

8093=Data log already exists.



Symbol	Address	Type	Comment
#CreateB.DONE		Bool	
#CreateB.ERROR		Bool	
#CreateB.STATUS		Word	
#ErrorStatus		Word	
#Reset		Bool	
#Step		USInt	

Network 3: Command + Transition 5: Open_NAME = Create_NAME

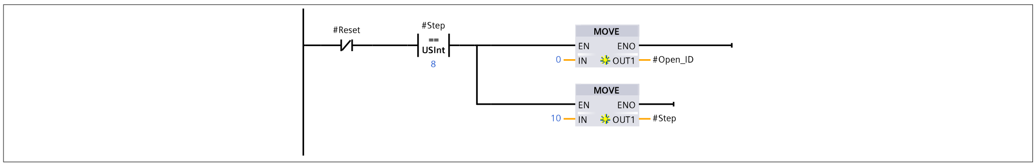


Symbol	Address	Type	Comment
#NAME		String	
#Open_NAME		String	

Totally Integrated
Automation Portal

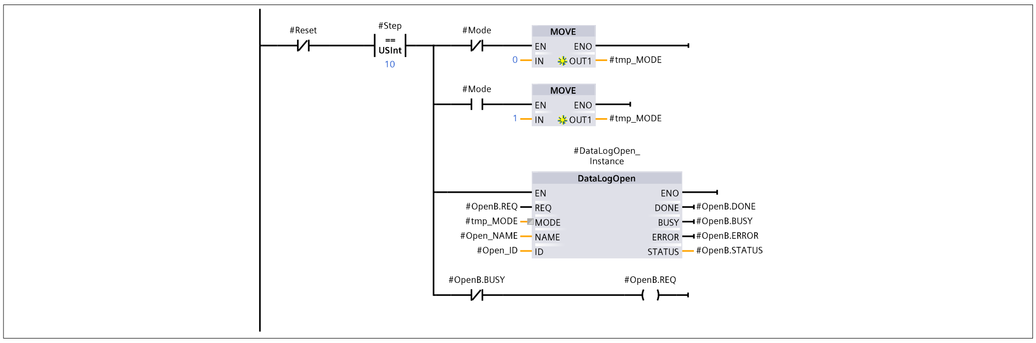
Symbol	Address	Type	Comment
#Reset		Bool	
#Step		USInt	

Network 4: Command + Transition 8: Clear Open_ID



Symbol	Address	Type	Comment
#Open_ID		UDInt	
#Reset		Bool	
#Step		USInt	

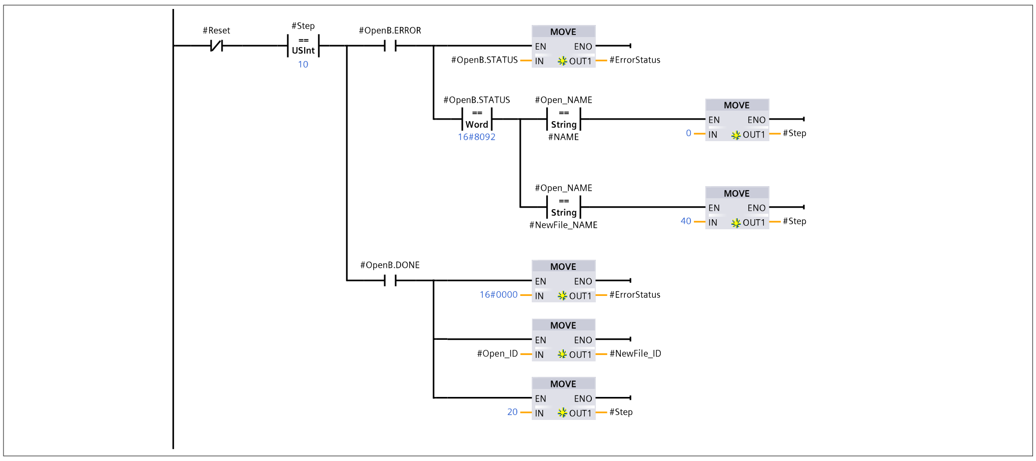
Network 5: Command 10: Open



Symbol	Address	Type	Comment
#DataLogOpen_Instance		Multi_SFB	
#Mode		Bool	
#Open_ID		UDInt	
#Open_NAME		String	
#OpenB.BUSY		Bool	
#OpenB.DONE		Bool	
#OpenB.ERROR		Bool	
#OpenB.REQ		Bool	
#OpenB.STATUS		Word	
#Reset		Bool	
#Step		USInt	
#tmp_MODE		USInt	

Network 6: Transition 10

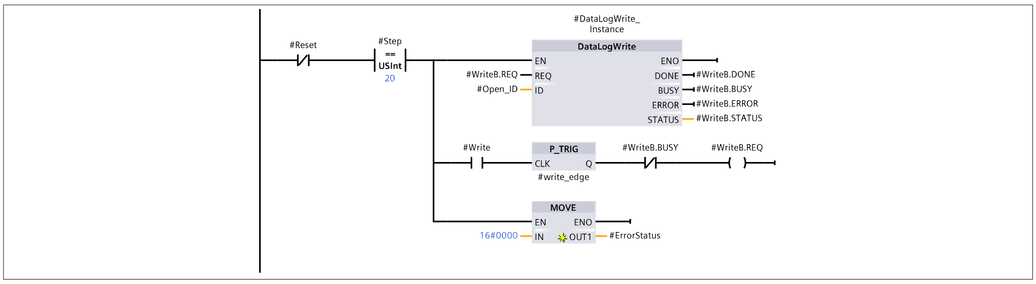
8092=Data log does not exist.



Symbol	Address	Type	Comment
#ErrorStatus		Word	

Symbol	Address	Type	Comment
#NAME		String	
#NewFile_ID		UDInt	
#NewFile_NAME		String	
#Open_ID		UDInt	
#Open_NAME		String	
#OpenB.DONE		Bool	
#OpenB.ERROR		Bool	
#OpenB.STATUS		Word	
#Reset		Bool	
#Step		USInt	

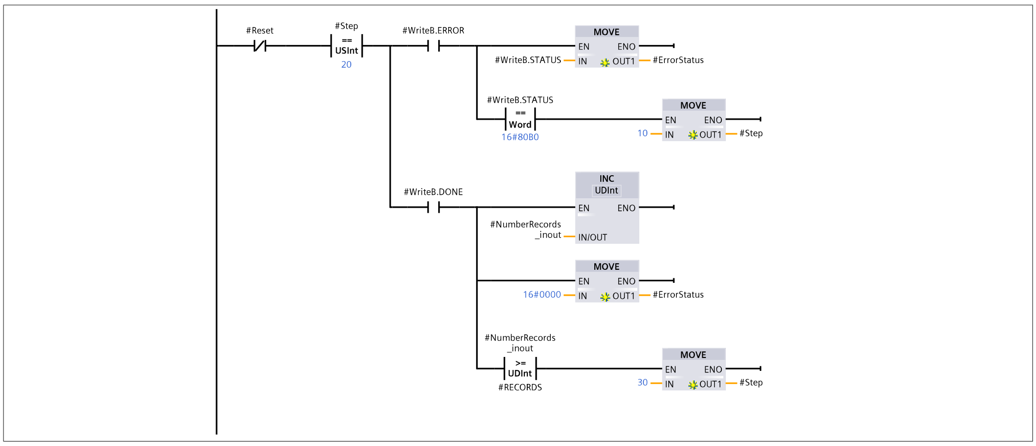
Network 7: Command 20: Write



Symbol	Address	Type	Comment
#DataLogWrite_Instance		Multi_SFB	
#ErrorStatus		Word	
#Open_ID		UDInt	
#Reset		Bool	
#Step		USInt	
#Write		Bool	
#write_edge		Bool	
#WriteB.BUSY		Bool	
#WriteB.DONE		Bool	
#WriteB.ERROR		Bool	
#WriteB.REQ		Bool	
#WriteB.STATUS		Word	

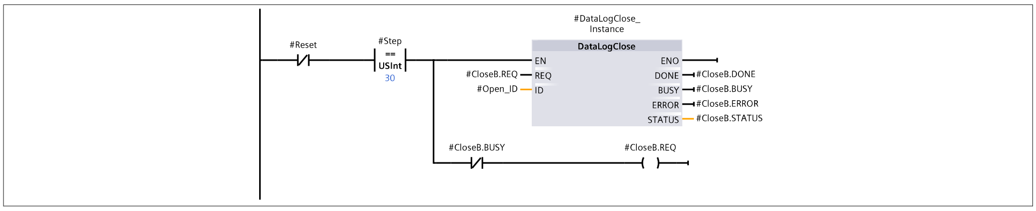
Network 8: Transition 20

80B0=Data log not open.



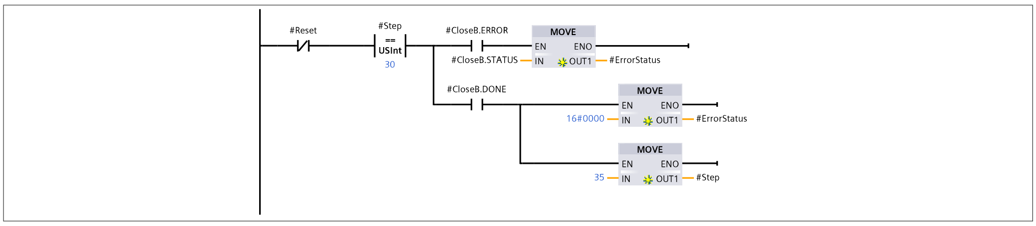
Symbol	Address	Type	Comment
#ErrorStatus		Word	
#NumberRecords_inout		UDInt	
#RECORDS		UDInt	
#Reset		Bool	
#Step		USInt	
#WriteB.DONE		Bool	
#WriteB.ERROR		Bool	
#WriteB.STATUS		Word	

Network 9: Command 30: Close



Symbol	Address	Type	Comment
#CloseB.BUSY		Bool	
#CloseB.DONE		Bool	
#CloseB.ERROR		Bool	
#CloseB.REQ		Bool	
#CloseB.STATUS		Word	
#DataLogClose_Instance		Multi_SFB	
#Open_ID		UDInt	
#Reset		Bool	
#Step		USInt	

Network 10: Transition 30

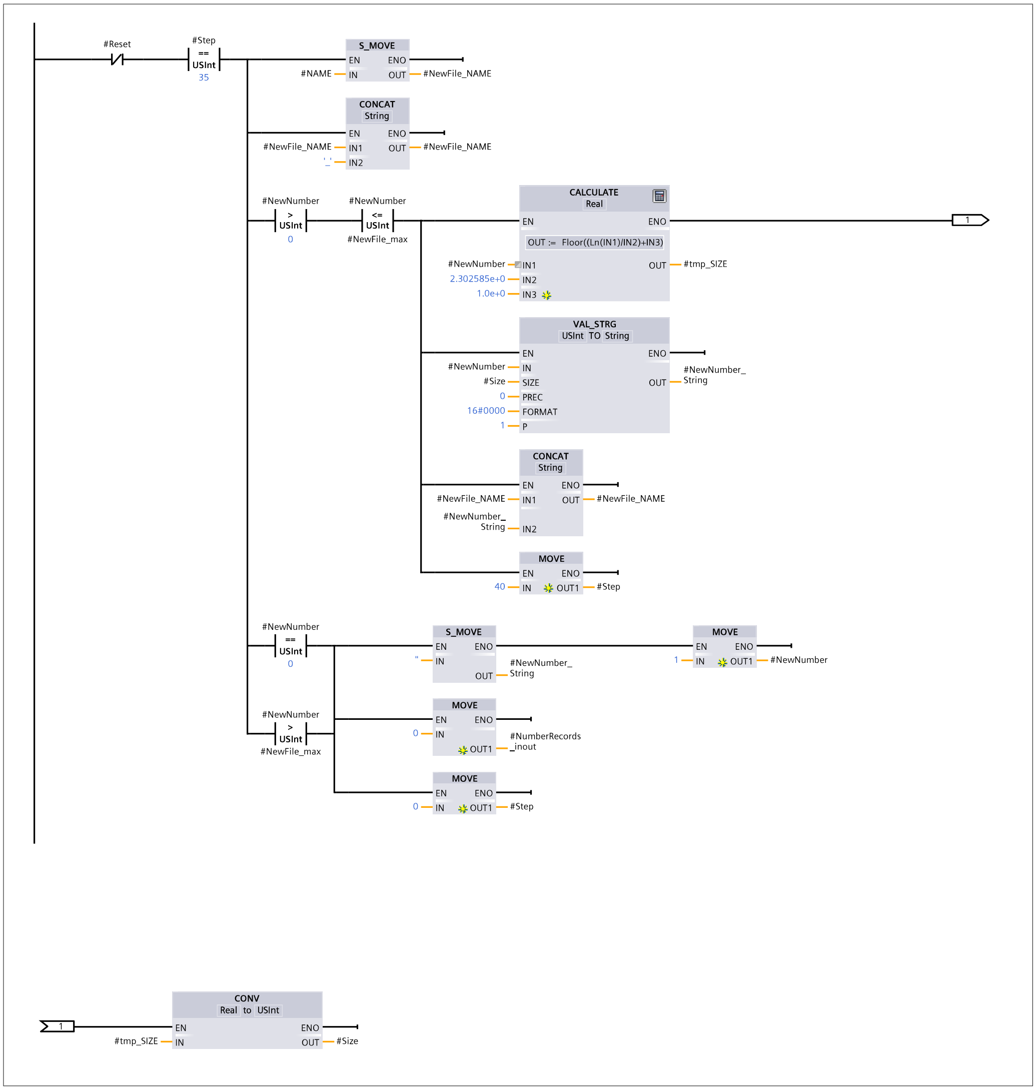


Symbol	Address	Type	Comment
#CloseB.DONE		Bool	
#CloseB.ERROR		Bool	
#CloseB.STATUS		Word	
#ErrorStatus		Word	
#Reset		Bool	
#Step		USInt	

Network 11: Command + Transition 35: Create new file NAME

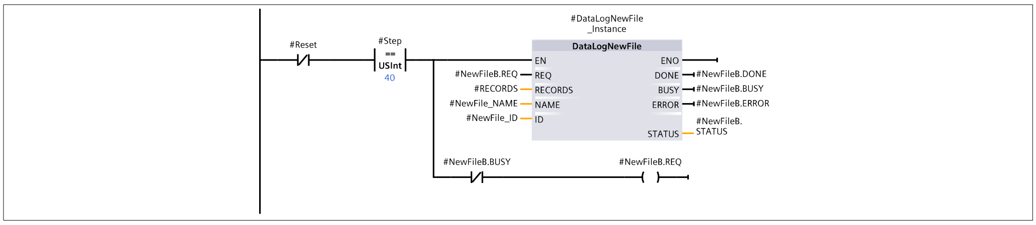
.

Network 11: Command + Transition 35: Create new file NAME



Symbol	Address	Type	Comment
#NAME		String	
#NewFile_max		USInt	
#NewFile_NAME		String	
#NewNumber		USInt	
#NewNumber_String		String	
#NumberRecords_inout		UDInt	
#Reset		Bool	
#Size		USInt	
#Step		USInt	
#tmp_SIZE		Real	

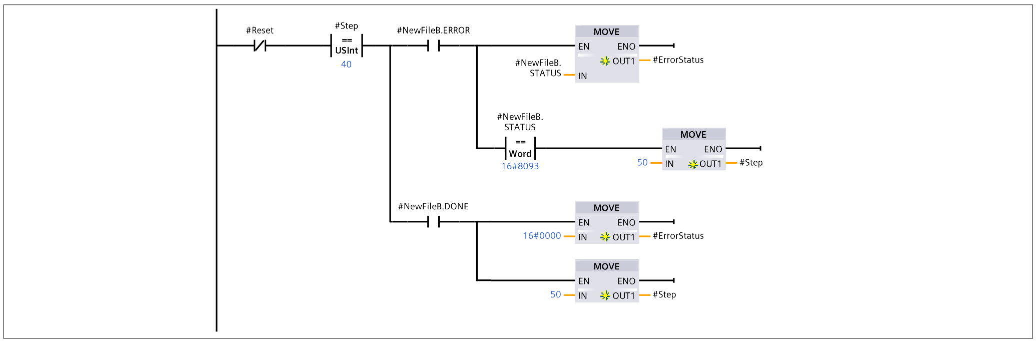
Network 12: Command 40: Create new file



Symbol	Address	Type	Comment
#DataLogNewFile_Instance		Multi_SFB	
#NewFile_ID		UDInt	
#NewFile_NAME		String	
#NewFileB.BUSY		Bool	
#NewFileB.DONE		Bool	
#NewFileB.ERROR		Bool	
#NewFileB.REQ		Bool	
#NewFileB.STATUS		Word	
#RECORDS		UDInt	
#Reset		Bool	
#Step		USInt	

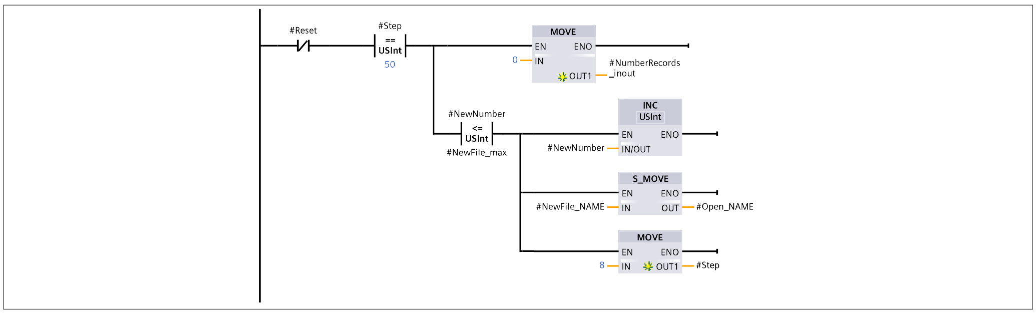
Network 13: Transition 40

8093=Data log already exists.



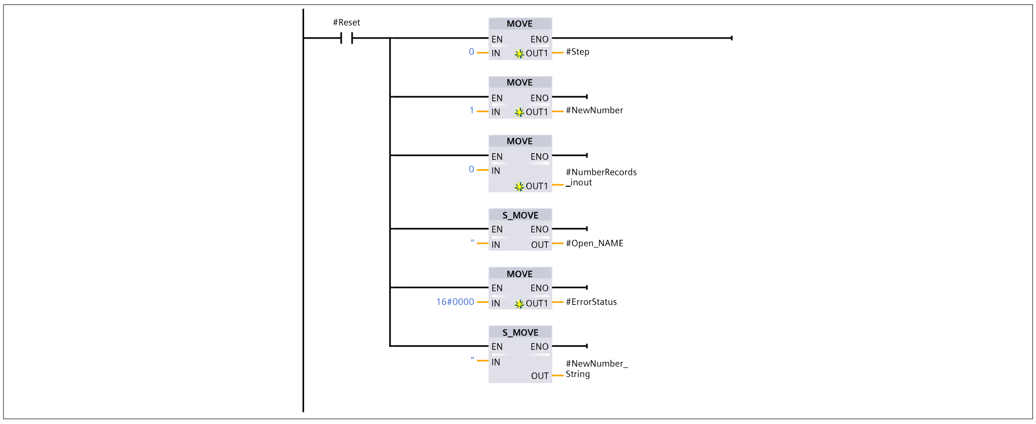
Symbol	Address	Type	Comment
#ErrorStatus		Word	
#NewFileB.DONE		Bool	
#NewFileB.ERROR		Bool	
#NewFileB.STATUS		Word	
#Reset		Bool	
#Step		USInt	

Network 14: Command + Transition 55: Open_NAME = NewFile_NAME & jump back



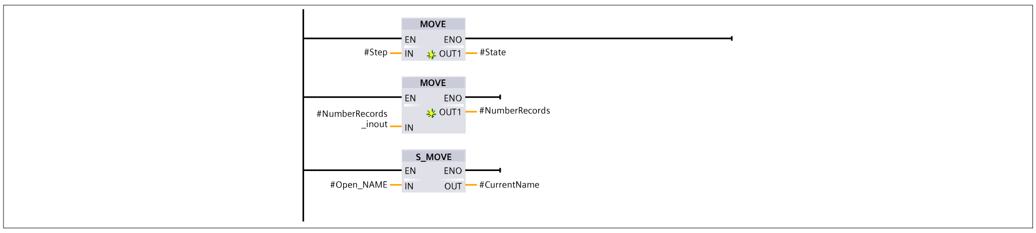
Symbol	Address	Type	Comment
#NewFile_max		USInt	
#NewFile_NAME		String	
#NewNumber		USInt	
#NumberRecords_inout		UDInt	
#Open_NAME		String	
#Reset		Bool	
#Step		USInt	

Network 15: Reset



Symbol	Address	Type	Comment
#ErrorStatus		Word	
#NewNumber		USInt	
#NewNumber_String		String	
#NumberRecords_inout		UDInt	
#Open_NAME		String	
#Reset		Bool	
#Step		USInt	

Network 16: output


MOVE
EN ENO
#Step IN OUT1 #State
MOVE
EN ENO
#NumberRecords_inout IN OUT1 #NumberRecords
S_MOVE
EN ENO
#Open_NAME IN OUT #CurrentName

Symbol	Address	Type	Comment
#CurrentName		String	
#NumberRecords		UDInt	
#NumberRecords_inout		UDInt	
#Open_NAME		String	
#State		USInt	
#Step		USInt	

7.2 Appendix B: Platform Outcomes

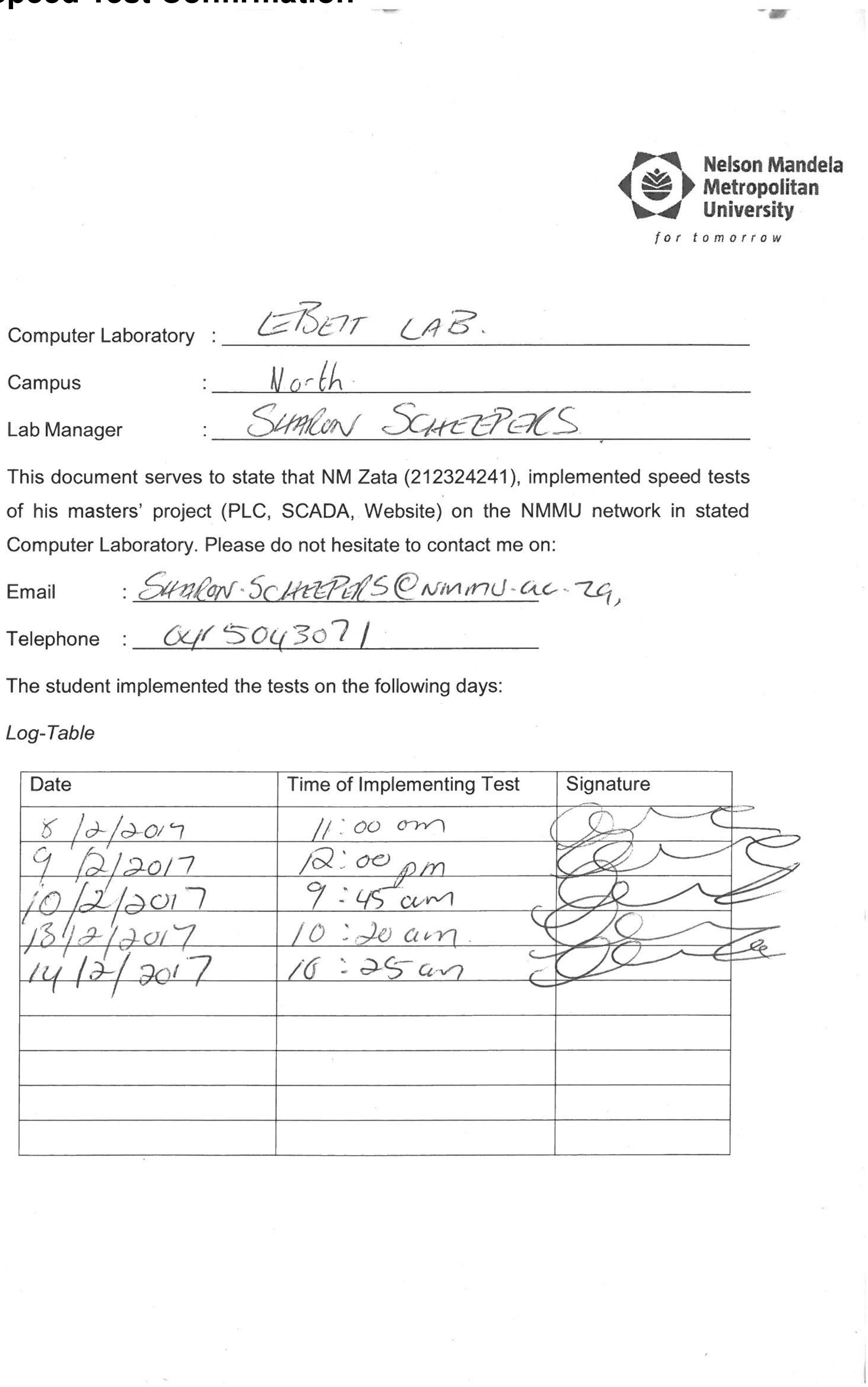
This section discusses the various outcomes from the platform and some documents used to evaluate the integrated platform. The outcomes include:

- Throughput Calculated - Response values of the ping tests done on the SCADA and website of the platform.
- SCADA for plant models - the real-time display pages that the user will access and operate the platform on are documented and described in Table 4.5.
- Website functionalities - the web-pages and critical activities that a user navigates through are illustrated and are described in **Section 4.4 Website and Process Control Interaction**.
- Publication from research - The interfacing and cost reduction on systems that make use of PLCs and simulated plant models is the main focus in the paper that was presented at the ROBMECH Conference 2016.

The documents that were used to test the platform's perfomance include:

- Speed Test Confirmation - Ping tests that were done to test the internet connectivity of the platform SCADA and website. Hence, the document is proof that the tests were actually done and monitored by the laboratory manager.
- Questionnaire for users - The document was used to get the learning experience and verify that the system was fully functional and met the outlined objectives.
- Practical guide for users - The document was used to guide and prepare the students to implement the assigned tasks on the developed platform.

7.2.1 Speed Test Confirmation



Nelson Mandela Metropolitan University
for tomorrow

Computer Laboratory : EBETT LAB.

Campus : North.

Lab Manager : SHARON SCHEEPERS

This document serves to state that NM Zata (212324241), implemented speed tests of his masters' project (PLC, SCADA, Website) on the NMMU network in stated Computer Laboratory. Please do not hesitate to contact me on:

Email : SHARON.SCHEEPERS@NMMU.AC.ZA,

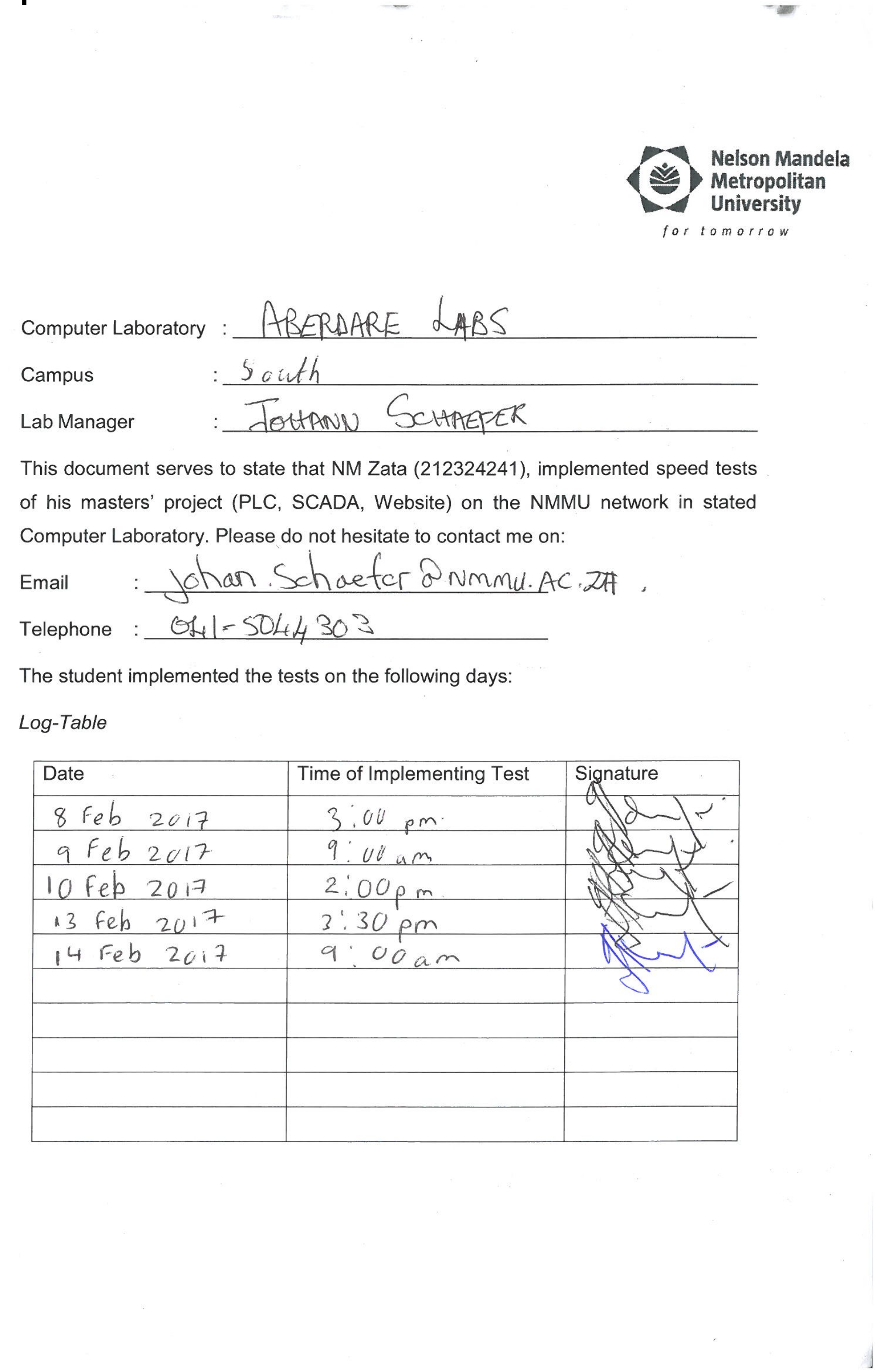
Telephone : 041 5043071

The student implemented the tests on the following days:

Log-Table

Date	Time of Implementing Test	Signature
8/2/2017	11:00 am	
9/2/2017	12:00 pm	
10/2/2017	9:45 am	
13/2/2017	10:20 am	
14/2/2017	10:25 am	

7.2.2 Speed Test Confirmation



Nelson Mandela
Metropolitan
University
for tomorrow

Computer Laboratory : ABERDARE LABS

Campus : South

Lab Manager : JOHANN SCHAEFER

This document serves to state that NM Zata (212324241), implemented speed tests of his masters' project (PLC, SCADA, Website) on the NMMU network in stated Computer Laboratory. Please do not hesitate to contact me on:

Email : johan.Schaefer@NMMU.AC.ZA

Telephone : 041-5044303

The student implemented the tests on the following days:

Log-Table

Date	Time of Implementing Test	Signature
8 Feb 2017	3:00 pm.	
9 Feb 2017	9:00 am	
10 Feb 2017	2:00pm	
13 Feb 2017	3:30 pm	
14 Feb 2017	9:00am	

7.2.3 Throughput Calculated

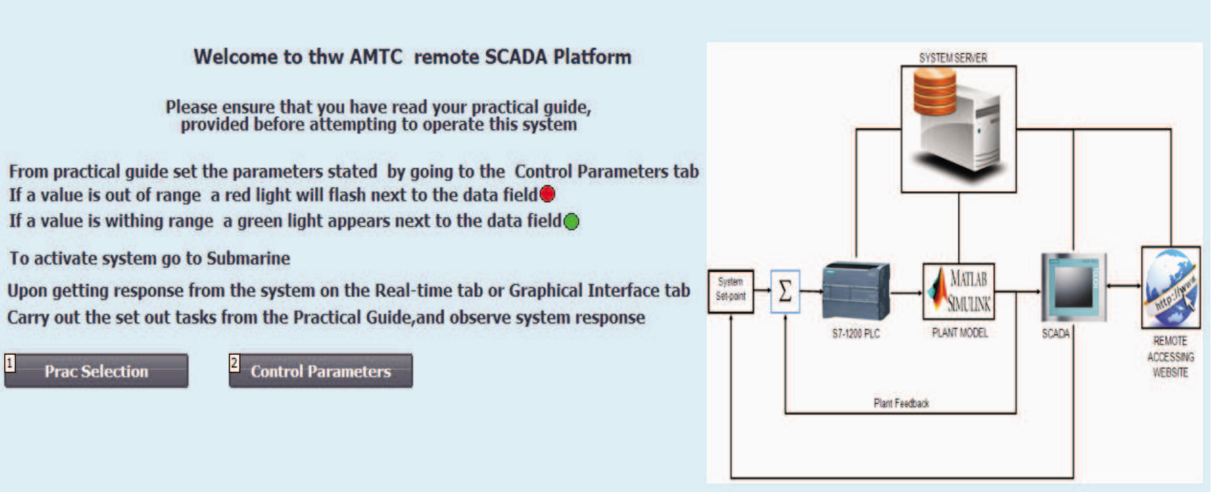
Aberdare Labs: SCADA										
	day 1		day 2		day 3		day 4		day 5	
bytes	max time(ms)	Throughput (Mbps)	max time(ms)	Throughput (Mbps)	max time(ms)	Throughput (Mbps)	max time(ms)	Throughput (Mbps)	max time(ms)	Throughput (Mbps)
13000	7	14.86	23	4.52	5	19.50	13	8.00	13	7.80
26000	12	17.33	29	7.17	31	6.71	24	8.67	31	6.71
30000	17	14.12	38	6.32	38	6.32	29	8.18	42	5.67
35000	22	13.02	48	5.83	46	6.15	40	7.00	48	5.87
37000	27	11.17	57	5.19	51	5.80	48	6.12	48	6.17

Aberdare Labs: Remote website										
bytes	max time(ms)	Throughput (Mbps)	max time(ms)	Throughput (Mbps)	max time(ms)	Throughput (Mbps)	max time(ms)	Throughput (Mbps)	max time(ms)	Throughput (Mbps)
13000	13	8.21	10	10.76	7	14.18	15	6.93	34	3.03
39000	18	17.33	14	21.77	36	8.75	22	14.40	36	8.59
42000	20	16.80	14	24.59	44	7.69	29	11.46	46	7.25
64000	29	17.66	17	30.12	48	10.59	42	12.10	54	9.54

Ebeit Labs: SCADA										
	day 1		day 2		day 3		day 4		day 5	
bytes	max time(ms)	Throughput (Mbps)	max time(ms)	Throughput (Mbps)	max time(ms)	Throughput (Mbps)	max time(ms)	Throughput (Mbps)	max time(ms)	Throughput (Mbps)
13000	6	17.33	17	6.00	6	17.33	5	22.29	3.7	28.36
26000	8	26.00	8	27.73	31	6.71	10	21.52	7.0	29.71
30000	7	32.73	38	6.32	11	21.82	9	26.67	7.3	32.73
35000	9	31.11	47	5.96	20	14.24	10	28.97	7.3	38.18
37000	11	26.91	51	5.80	31	9.55	12	24.67	8.0	37.00

Ebeit Labs: Remote website										
	day 1		day 2		day 3		day 4		day 5	
bytes	max time(ms)	Throughput (Mbps)	max time(ms)	Throughput (Mbps)	max time(ms)	Throughput (Mbps)	max time(ms)	Throughput (Mbps)	max time(ms)	Throughput (Mbps)
13000	5	22.29	6	17.33	17	6.00	6	17.33	5	22.29
39000	13	24.63	14	24.63	36	8.75	14	21.77	18	17.66
42000	17	20.16	15	20.16	44	7.69	14	24.59	17	19.76
64000	26	19.69	30	19.69	48	10.59	17	30.12	19	26.95

7.2.4 SCADA for Ballast Tank Control Model and Dual Tank Model



Welcome to thw AMTC remote SCADA Platform

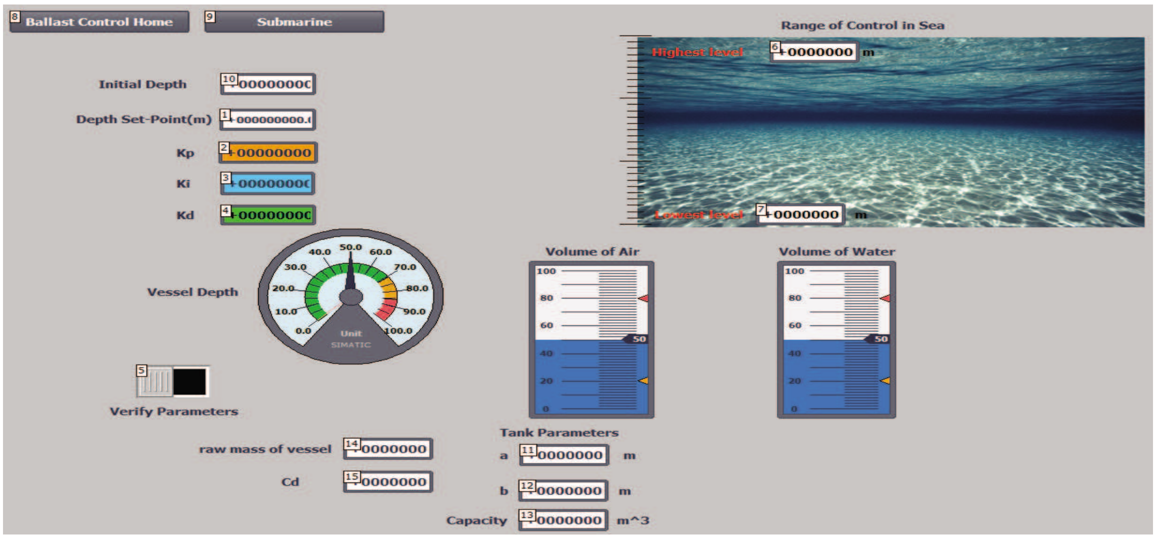
Please ensure that you have read your practical guide,
provided before attempting to operate this system

From practical guide set the parameters stated by going to the Control Parameters tab
If a value is out of range a red light will flash next to the data field●
If a value is withing range a green light appears next to the data field●

To activate system go to Submarine

Upon getting response from the system on the Real-time tab or Graphical Interface tab
Carry out the set out tasks from the Practical Guide,and observe system response

Figure 7.1: Ballast Tank Control Home



Highest level 6 0000000 m

Lowest level 7 0000000 m

Initial Depth 10 00000000

Depth Set-Point(m) 1 000000000.

Kp 2 00000000

Ki 3 00000000

Kd 4 00000000

raw mass of vessel 14 0000000

Cd 15 0000000

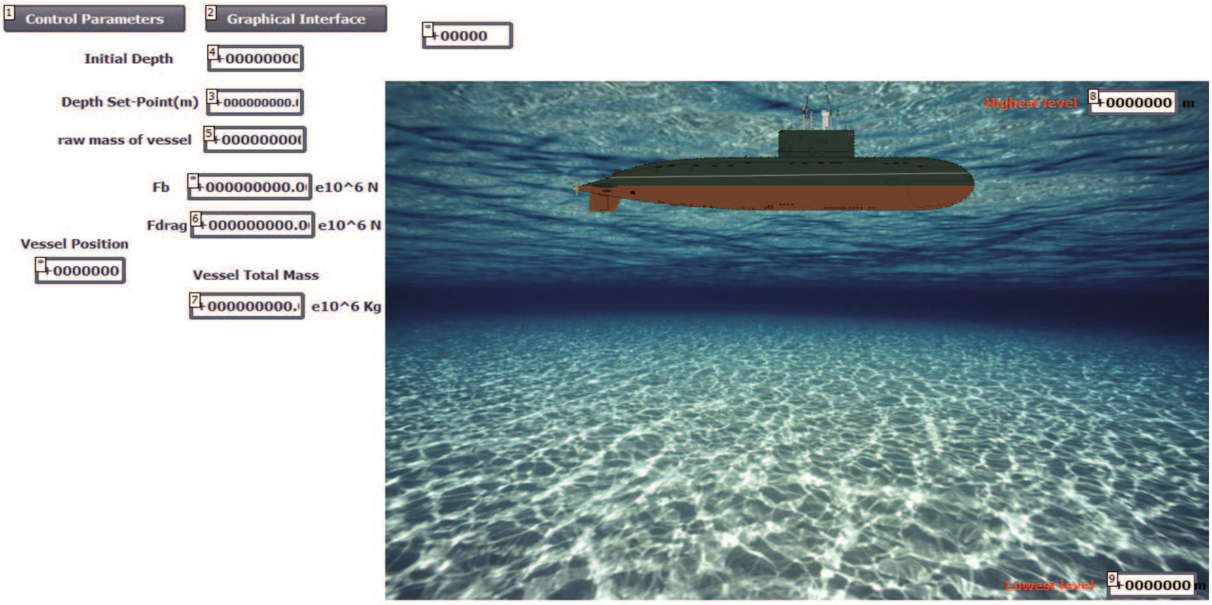
Tank Parameters

a 11 0000000 m

b 12 0000000 m

Capacity 13 0000000 m^3

Figure 7.2: Ballast Tank Parameters



+00000

Initial Depth 4 +0000000C

Depth Set-Point(m) 3 +000000000.(

raw mass of vessel 5 +00000000(

Fb +000000000.0(e10^6 N

Fdrag 6 +000000000.0(e10^6 N

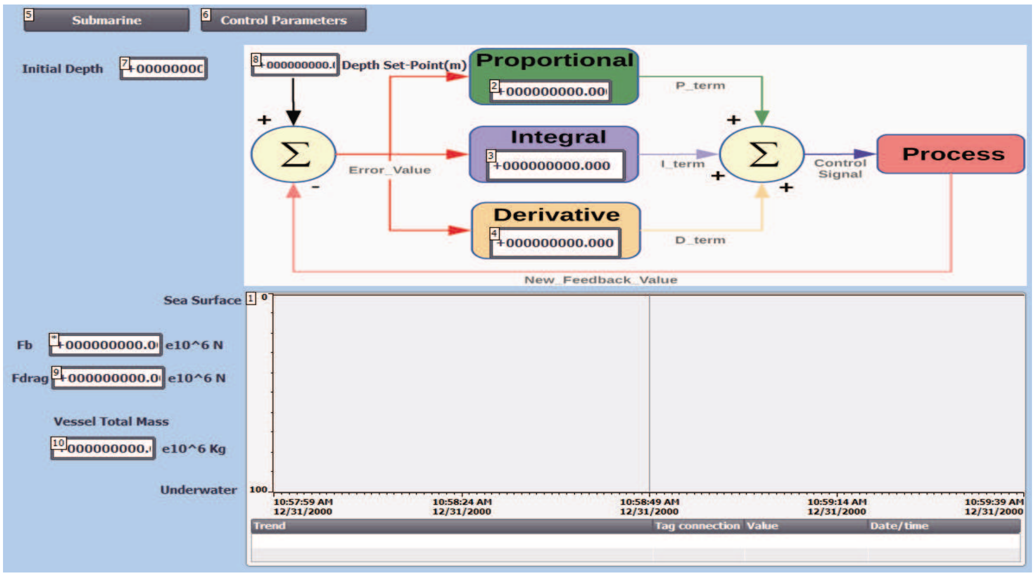
Vessel Position +0000000

Vessel Total Mass 7 +000000000.(e10^6 Kg

Highest level 8 +0000000 m

Lowest level 9 +0000000 m

Figure 7.3: Submarine Hovering underwater



Initial Depth 7 +0000000C

8 +000000000.(Depth Set-Point(m)

Proportional 2 +000000000.00(

Integral 3 +000000000.000

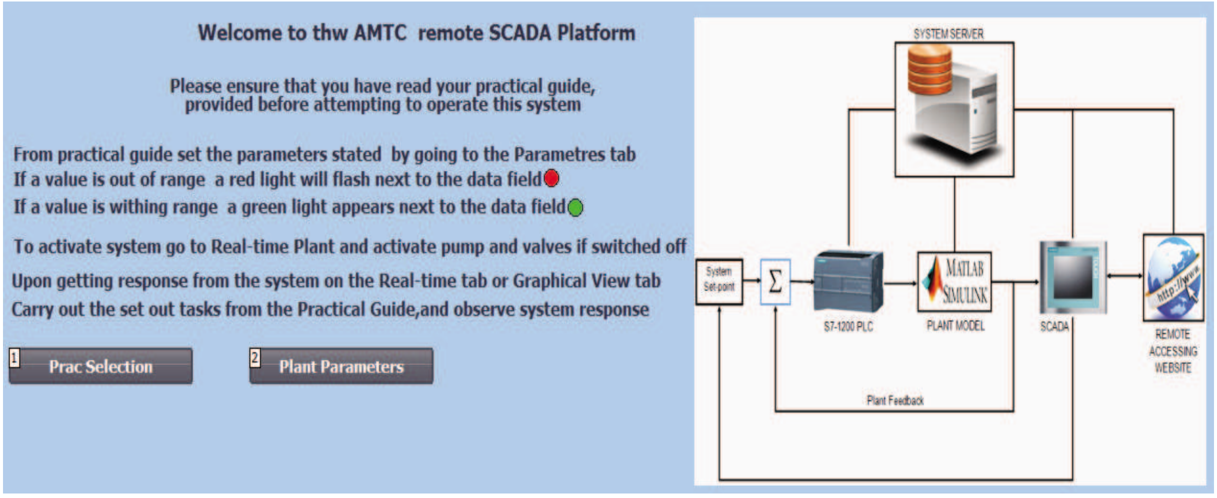
Derivative 4 +000000000.000

Fb +000000000.0(e10^6 N

Fdrag 9 +000000000.0(e10^6 N

Vessel Total Mass 10 000000000.(e10^6 Kg

Figure 7.4: Submarine Graphical View



Welcome to thw AMTC remote SCADA Platform

Please ensure that you have read your practical guide,
provided before attempting to operate this system

From practical guide set the parameters stated by going to the Parametres tab
If a value is out of range a red light will flash next to the data field🔴
If a value is withing range a green light appears next to the data field🟢

To activate system go to Real-time Plant and activate pump and valves if switched off

Upon getting response from the system on the Real-time tab or Graphical View tab

Carry out the set out tasks from the Practical Guide,and observe system response

Figure 7.5: Dual Tank Home

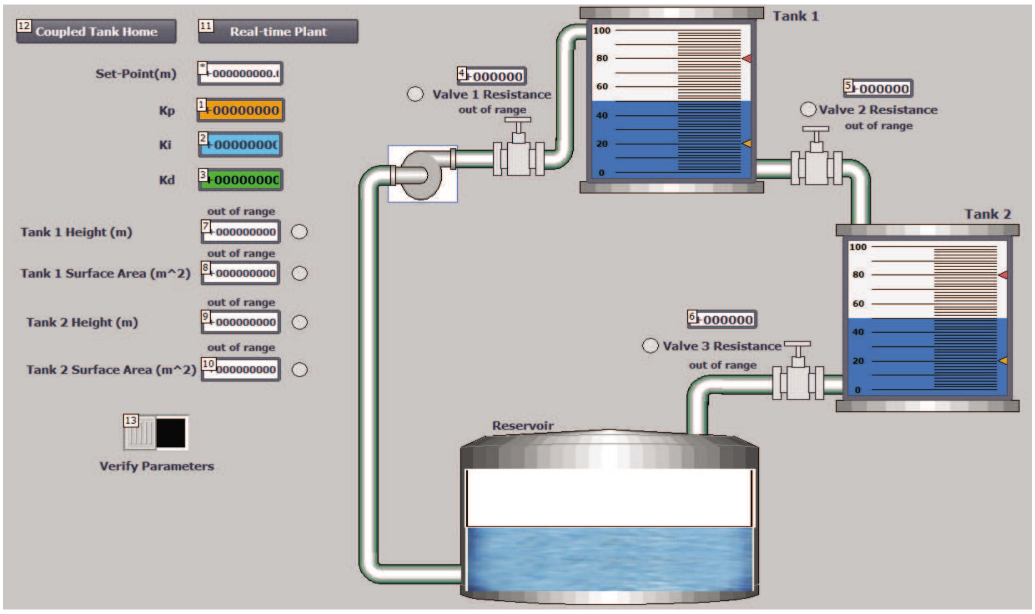


Figure 7.6: Dual Tank Parameters

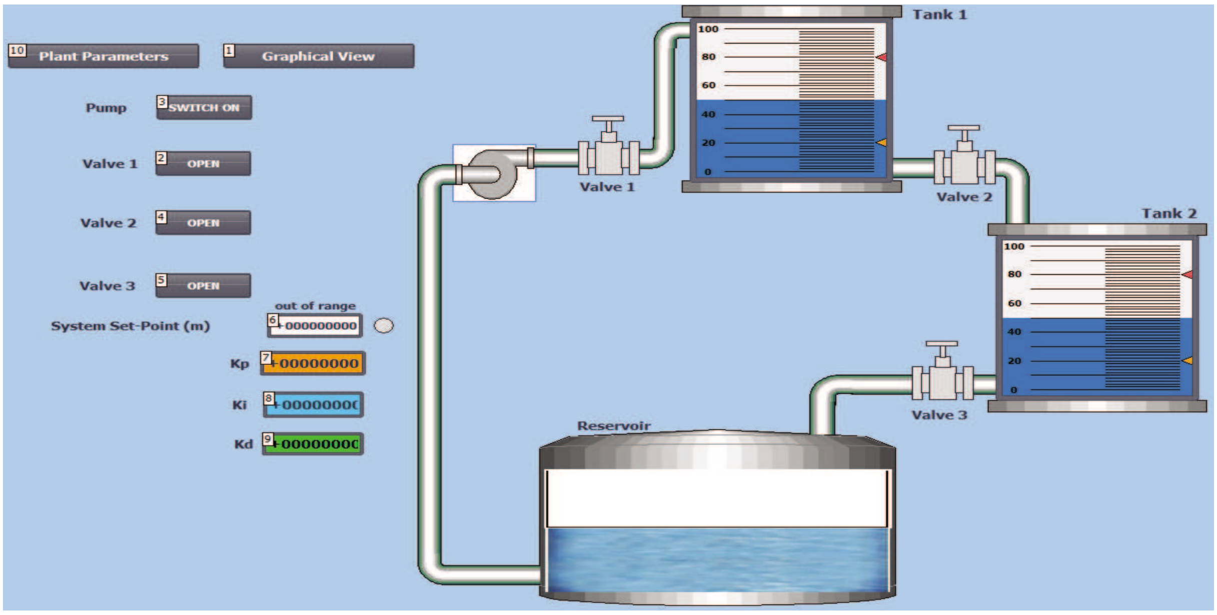


Figure 7.7: Dual Tank Response System

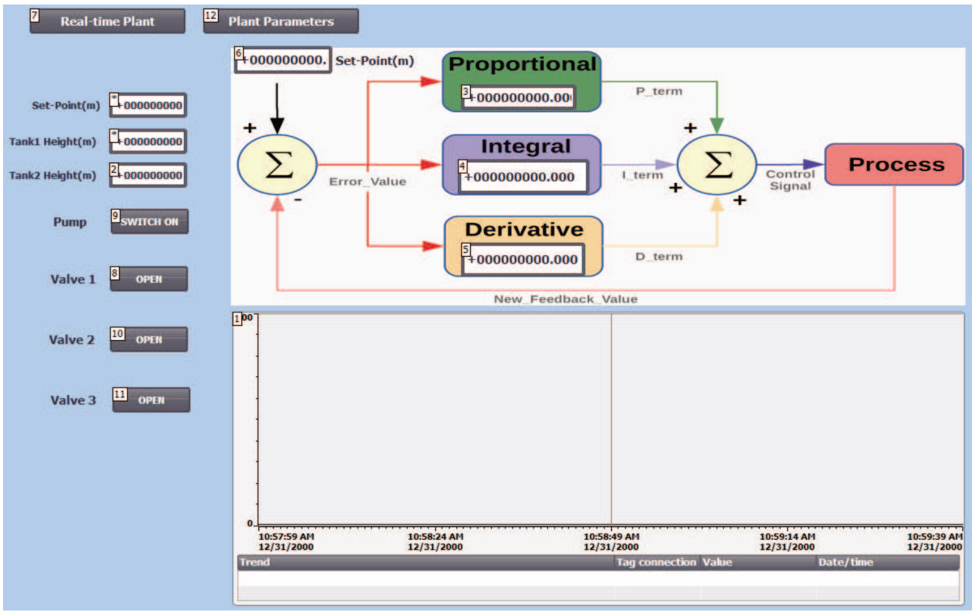
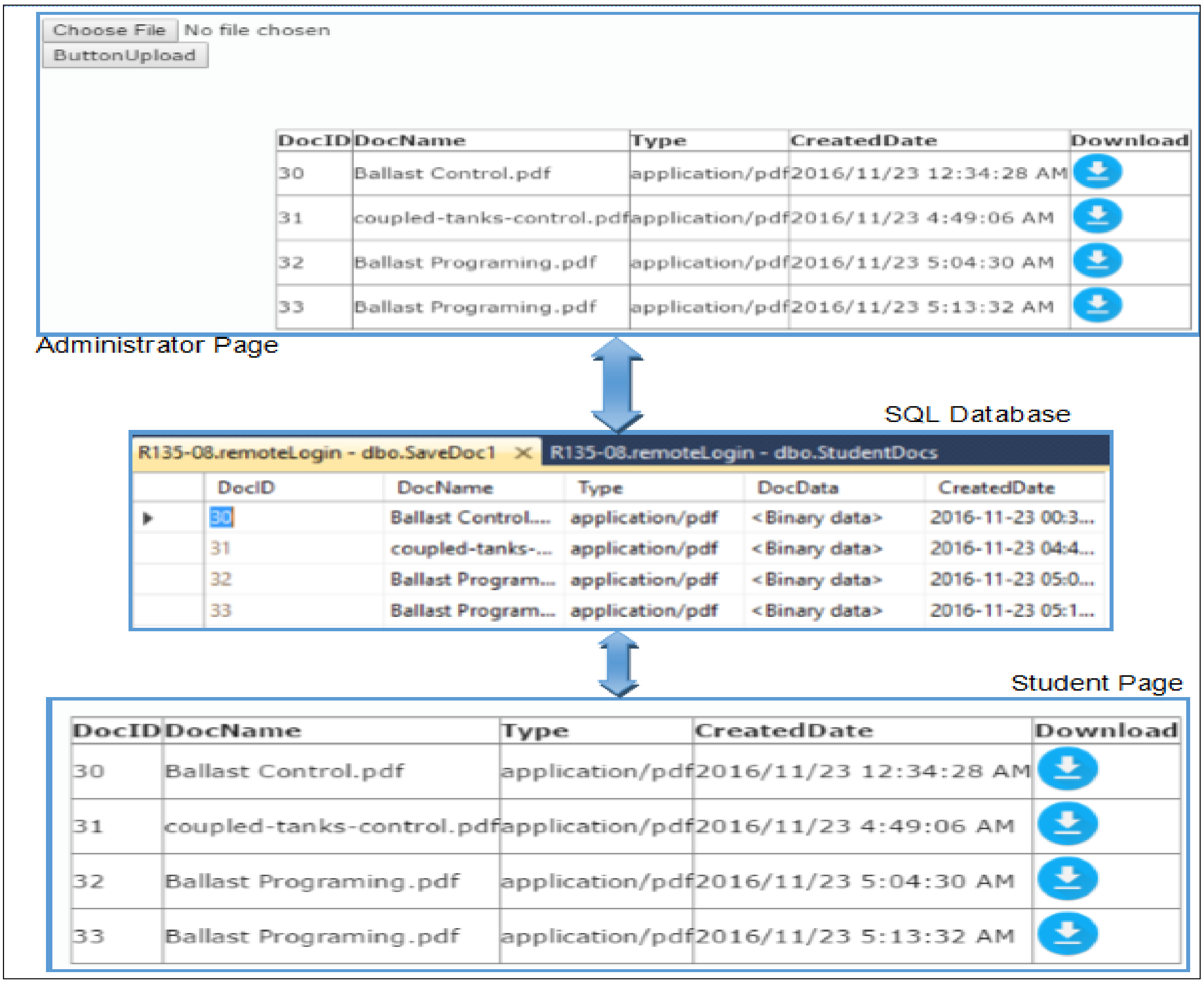


Figure 7.8: Dual Tank Graphical View

7.2.5 Remote Accessing Website Outcomes

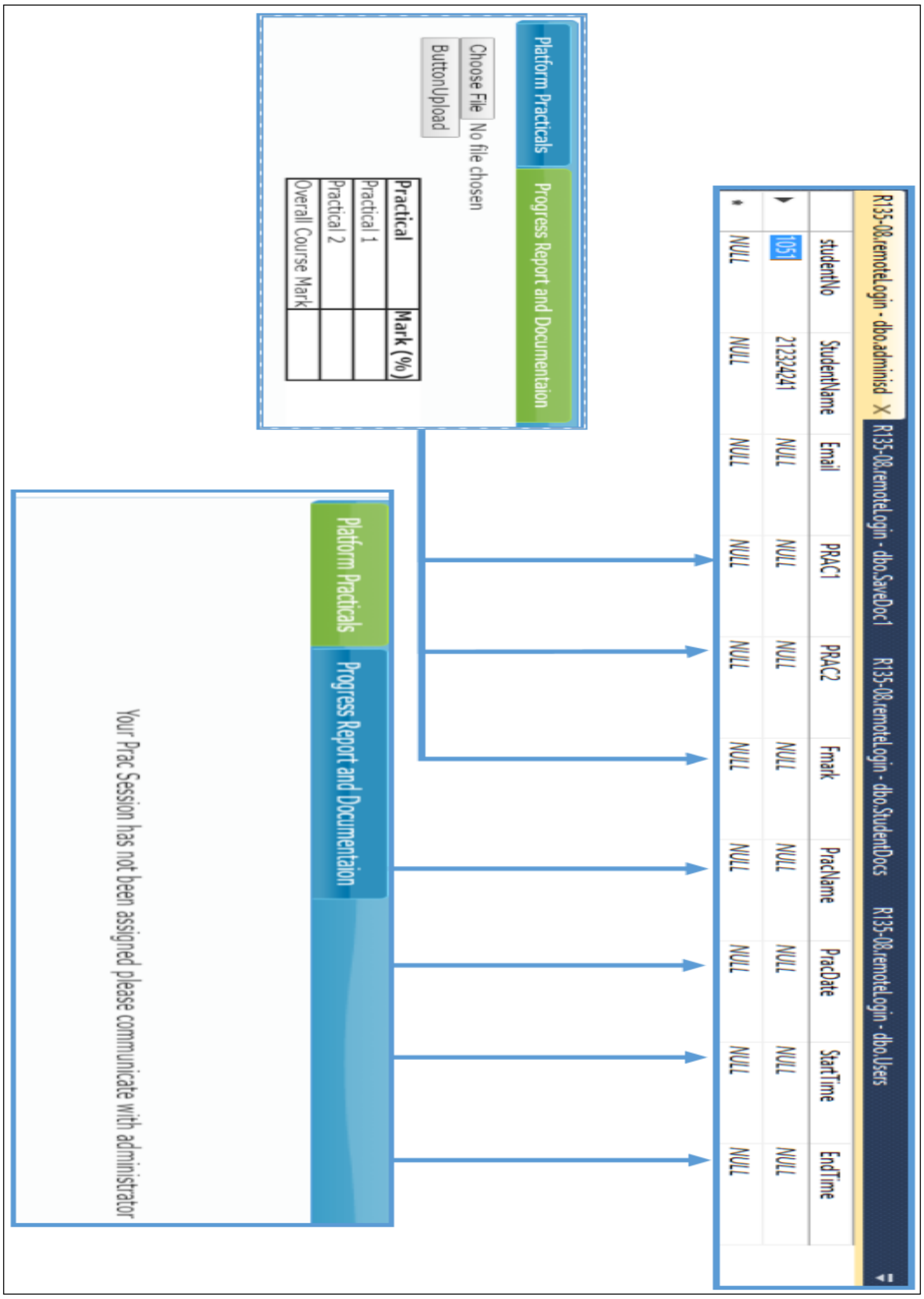


DocID	DocName	Type	CreatedDate	Download
30	Ballast Control.pdf	application/pdf	2016/11/23 12:34:28 AM	⬇
31	coupled-tanks-control.pdf	application/pdf	2016/11/23 4:49:06 AM	⬇
32	Ballast Programing.pdf	application/pdf	2016/11/23 5:04:30 AM	⬇
33	Ballast Programing.pdf	application/pdf	2016/11/23 5:13:32 AM	⬇

	DocID	DocName	Type	DocData	CreatedDate
▶	30	Ballast Control....	application/pdf	<Binary data>	2016-11-23 00:3...
	31	coupled-tanks-...	application/pdf	<Binary data>	2016-11-23 04:4...
	32	Ballast Program...	application/pdf	<Binary data>	2016-11-23 05:0...
	33	Ballast Program...	application/pdf	<Binary data>	2016-11-23 05:1...

DocID	DocName	Type	CreatedDate	Download
30	Ballast Control.pdf	application/pdf	2016/11/23 12:34:28 AM	⬇
31	coupled-tanks-control.pdf	application/pdf	2016/11/23 4:49:06 AM	⬇
32	Ballast Programing.pdf	application/pdf	2016/11/23 5:04:30 AM	⬇
33	Ballast Programing.pdf	application/pdf	2016/11/23 5:13:32 AM	⬇

Figure 7.9: Practical Guides for students

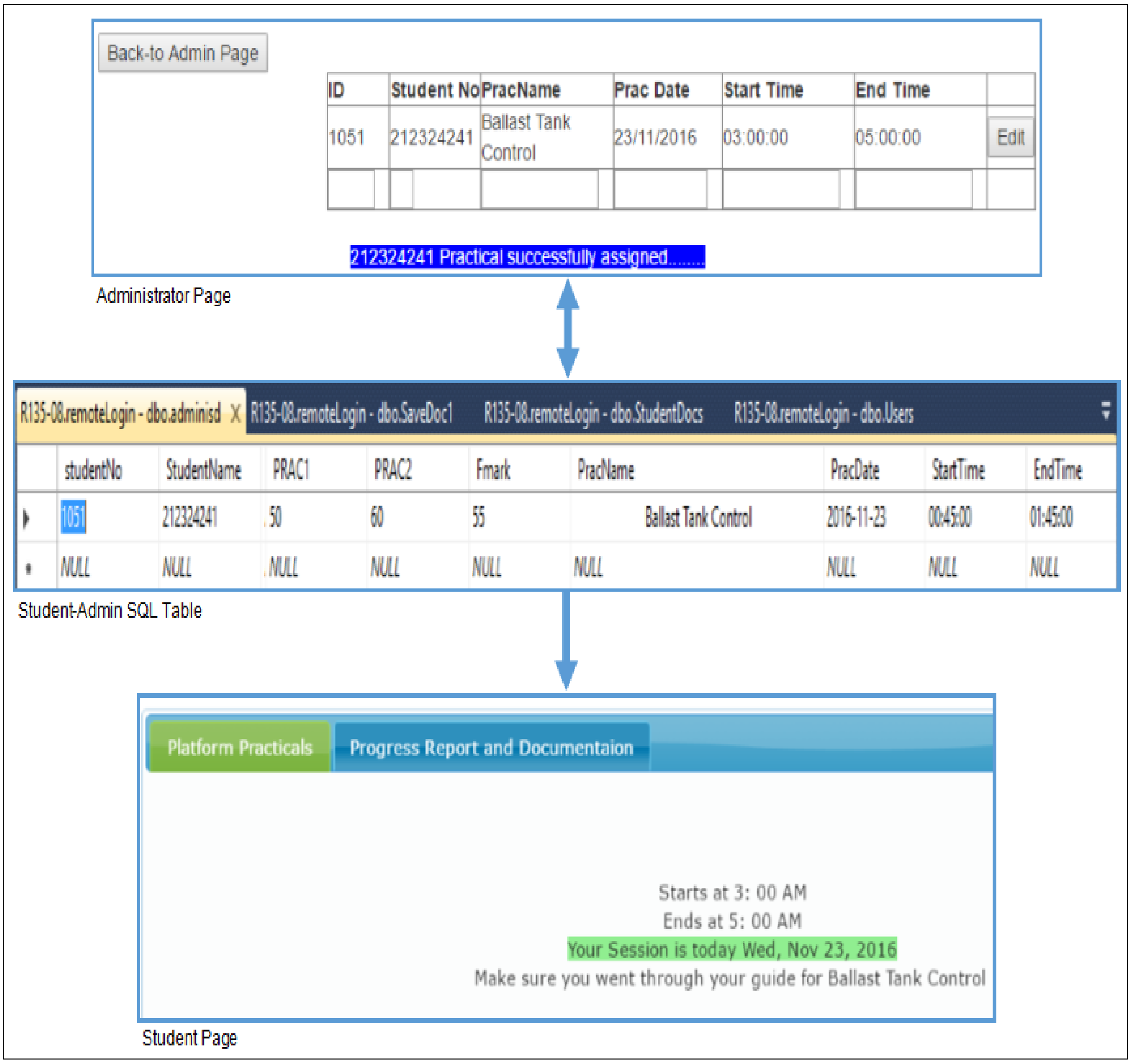

R135-08.remoteLogin - dbo.adminisd X
R135-08.remoteLogin - dbo.SaveDoc1
R135-08.remoteLogin - dbo.StudentDocs
R135-08.remoteLogin - dbo.Users
Platform Practicals
Progress Report and Documentaion
Choose File
No file chosen
ButtonUpload

studentNo	StudentName	Email	PRAC1	PRAC2	Fmark	PracName	PracDate	StartTime	EndTime
1051	21232421	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Practical	Mark (%)
Practical 1	
Practical 2	
Overall Course Mark	

Your Prac Session has not been assigned please communicate with administrator

Figure 7.10: User registered onto website


Back-to Admin Page
Edit
Administrator Page
R135-08.remoteLogin - dbo.adminisd X
R135-08.remoteLogin - dbo.SaveDoc1
R135-08.remoteLogin - dbo.StudentDocs
R135-08.remoteLogin - dbo.Users
Student-Admin SQL Table
Platform Practicals
Progress Report and Documentaion
Student Page

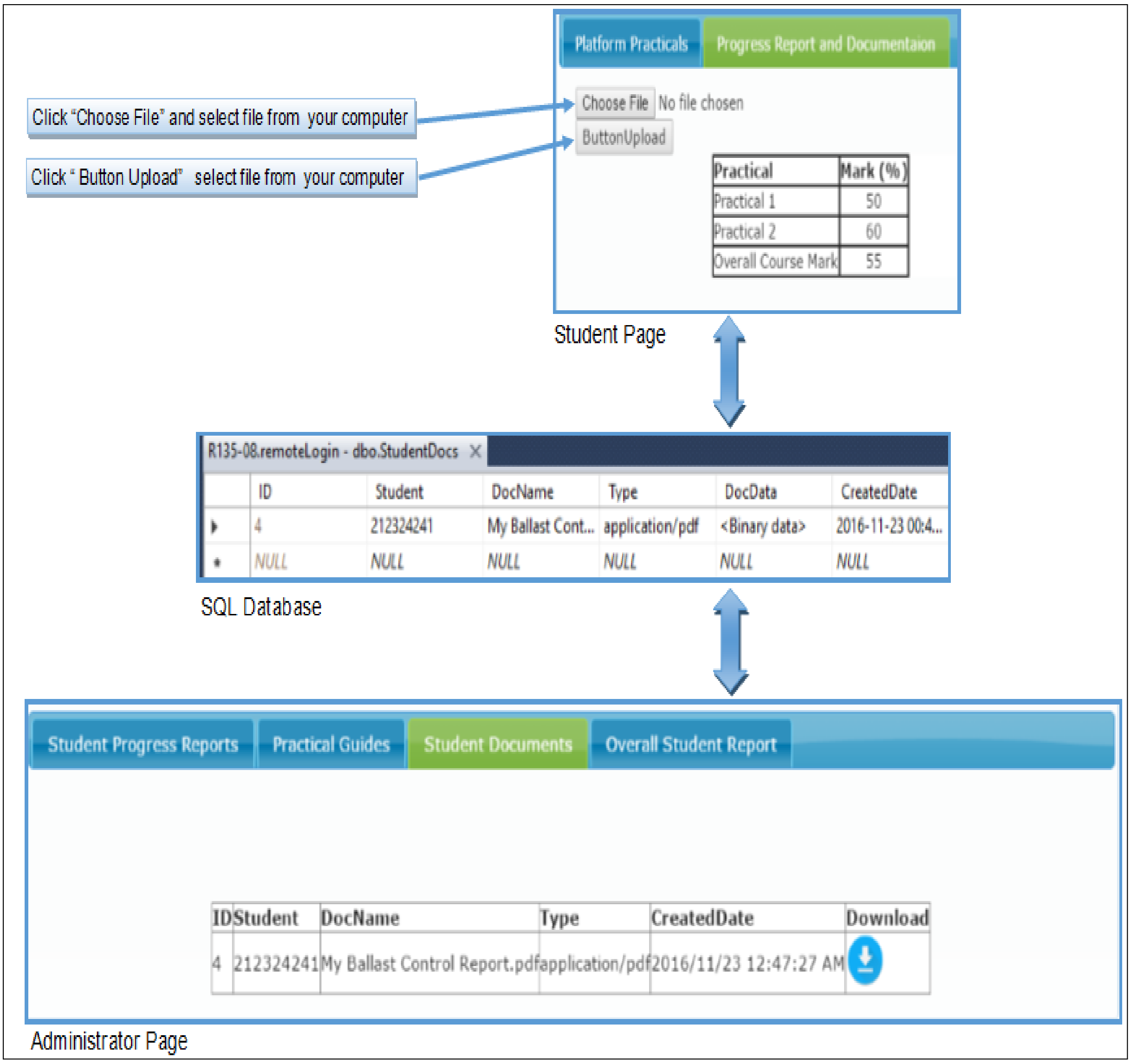
ID	Student No	PracName	Prac Date	Start Time	End Time	
1051	212324241	Ballast Tank Control	23/11/2016	03:00:00	05:00:00	Edit

212324241 Practical successfully assigned........

	studentNo	StudentName	PRAC1	PRAC2	Fmark	PracName	PracDate	StartTime	EndTime
▶	1051	212324241	50	60	55	Ballast Tank Control	2016-11-23	00:45:00	01:45:00
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Starts at 3: 00 AM
Ends at 5: 00 AM
Your Session is today Wed, Nov 23, 2016
Make sure you went through your guide for Ballast Tank Control

Figure 7.11: Successful Practical Scheduling



Click "Choose File" and select file from your computer

Click " Button Upload" select file from your computer

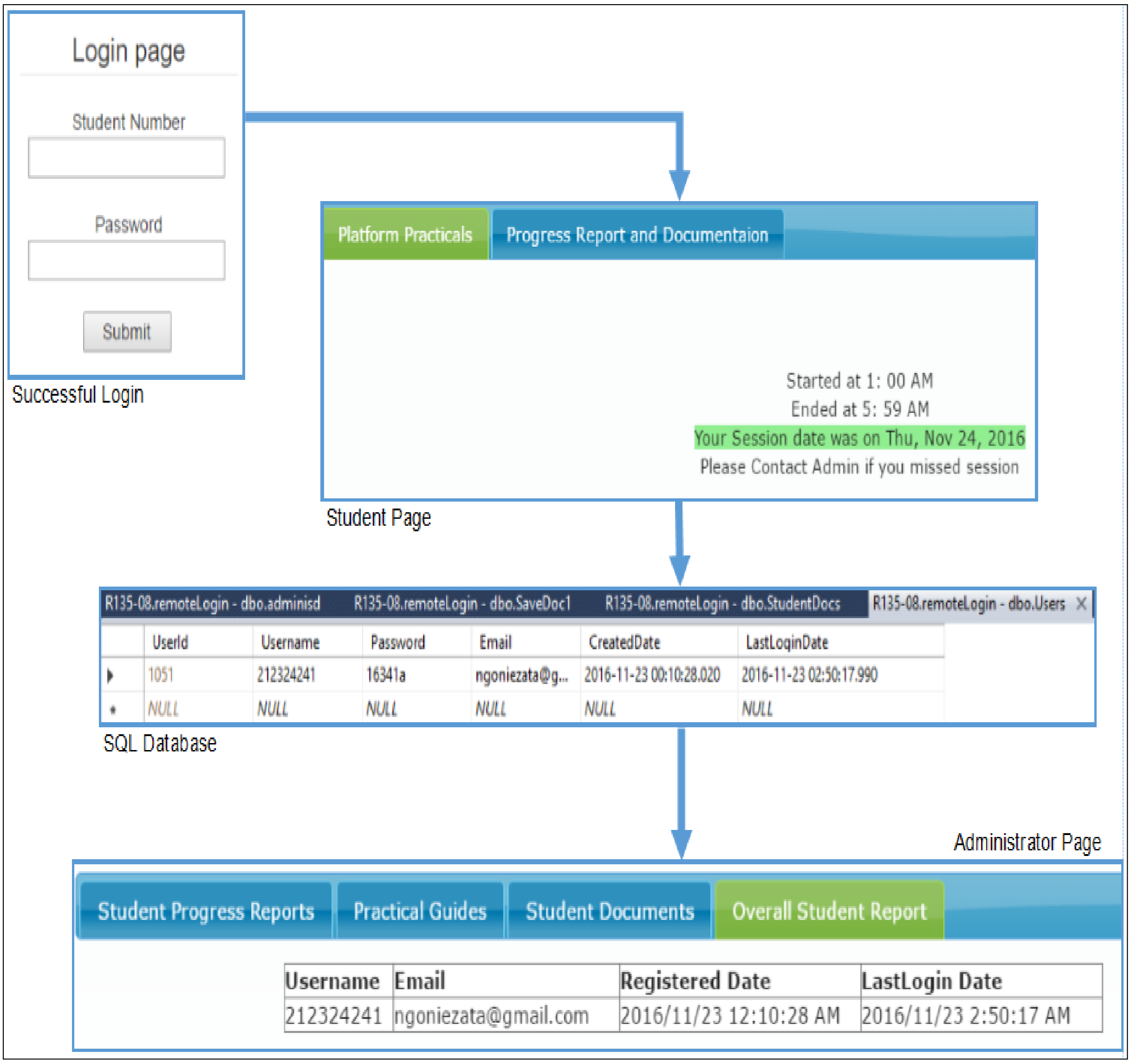
Practical	Mark (%)
Practical 1	50
Practical 2	60
Overall Course Mark	55

R135-08.remoteLogin - dbo.StudentDocs

	ID	Student	DocName	Type	DocData	CreatedDate
▸	4	212324241	My Ballast Cont...	application/pdf	<Binary data>	2016-11-23 00:4...
*	NULL	NULL	NULL	NULL	NULL	NULL

ID	Student	DocName	Type	CreatedDate	Download
4	212324241	My Ballast Control Report.pdf	application/pdf	2016/11/23 12:47:27 AM	⬇

Figure 7.12: Successful file uploading on student page



Started at 1: 00 AM
Ended at 5: 59 AM
Your Session date was on Thu, Nov 24, 2016
Please Contact Admin if you missed session

	UserId	Username	Password	Email	CreatedDate	LastLoginDate
▶	1051	212324241	16341a	ngoniezata@g...	2016-11-23 00:10:28.020	2016-11-23 02:50:17.990
*	NULL	NULL	NULL	NULL	NULL	NULL

Username	Email	Registered Date	LastLogin Date
212324241	ngoniezata@gmail.com	2016/11/23 12:10:28 AM	2016/11/23 2:50:17 AM

Figure 7.13: Successful Login and Login time capture

7.2.6 Publication

A Process Control Learning Factory with a Plant Simulation integrated to Industry Standard Control Hardware

Ngonidzashe Munyaradzi Zata, Theo Ian van Niekerk and John Manuel Fernandes
Mechatronics Department, Faculty of Engineering, Built Environment and Information Technology,
Nelson Mandela Metropolitan University, Port Elizabeth , 6001, South Africa
ngoniezata@gmail.com, Theo.vanNiekerk@nmmu.ac.za, John.Fernandes@nmmu.ac.za

***Abstract*—In this paper, a PLC controlled simulation of a coupled tank system based on an existing dual-tank educational process control plant is developed. A critical analysis of the plant is carried out in order to obtain all the required parameters to setup the simulated model. Unlike the actual plant, the virtual process control model is not limited to two tanks only but may be reconfigured with ease in numerous ways. The implementation of this kind of simulated platform that makes use of actual standard industrial software and hardware not only gives engineering students the necessary exposure to make a successful transition in to industry but also enhances learning capability in that what is taught is no longer limited to physically available equipment only. Furthermore, such a platform solves other problems commonly faced in learning environments such as limited resources and space constraints by opening up the possibility of remote laboratories.**

***Keywords*—Simulation, MATLAB SIMULINK ,TIA - Totally Integrated Automation, S7-1200, PLC, Coupled Tank System, Process Control, I/Os (Inputs & Outputs), Remote Labs, Virtual Labs**

I. Introduction

Simulation is the emulation of a system, including its dynamic processes within a model one can experiment with. The prospect of simulating industrial processes without having to construct them physically is becoming a more attractive option [1], [2]. There has recently been extensive development in simulation platforms that emulate automated cells (manufacturing robots, assembling robots, welding robots, etc.) There are also other platforms that are used to simulate process control systems but seem to be limited compared to those used to simulate automated cells. In universities simulation platforms are quite important as they bring quite a number of advantages in the learning environment. Some of the advantages include [3], [4]:

- Reduced concerns about safety; the chance of dangerous situations are reduced to a minimum
- Physical experimentation comes at a high cost as physical systems are expensive to acquire and maintain whereas simulation systems only cost during the setup; maintenance costs are also kept low
- Each student can actively participate in lab work and the possibility of copying is reduced
- Reusing the same control hardware on different virtual plant models reduces costs and saves space

Most simulation platforms are designed in a manner that follows the architecture shown in Fig. 1 [5], [6].

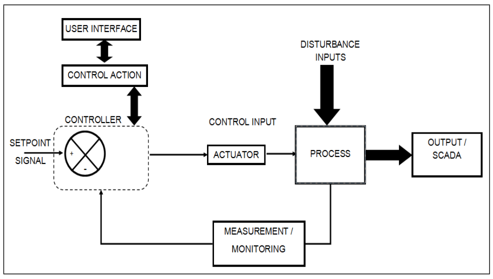


Fig. 1. Basic Architecture of a Simulation System

The subsystems shown in Fig.1 include:

- The user interface is a bridge of information of the process between the user and the platform
- User's commands are variables converted into control actions which are then converted into controller directives
- Controller - manipulates the plant and runs on the program and instructions coded in its memory
- The process is the system that is being operated to give the desired outputs
- The measurement and monitoring block represents the feedback of the system's current status during operation; various sensors (depending on the nature of the process) may be used to achieve this feedback

The architecture shown in Fig. 1 also depicts what a typical industrial process control system looks like.

In this paper, an educational simulated dual-tank system is presented [7]. The system is manipulated and controlled by a Siemens S7-1200 PLC that is programmed using TIA Portal

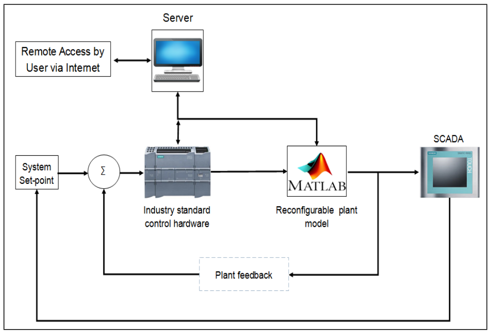

Server
Remote Access by User via Internet
SCADA
System Set-point
Σ
MATLAB
Industry standard control hardware
Reconfigurable plant model
Plant feedback

Fig. 2. Proposed Solution Architecture

from Siemens. The platform Fig. 2 is designed to be used in various configurations that will give different principles to be taught without having to setup a new system but make use of the available hardware and software. The system also leaves room for further development and research. This paper is structured following the steps that were taken to develop the platform as a whole, this involved:

- Mathematical modeling of tank system
- Programing of model in MATLAB SIMULINK based on Mathematical modeling
- Communication setup for SIMULINK model with PLC
- System tests and results(real-time responsiveness of system and plant control & parameter manipulation)

II. Background on Process Control Practicals in Universities

In most universities, carrying out a process control practical is usually done in the manner whereby the lecturer explains to the students what the expected outcomes of the practical are and a practical guide is then provided to the students. In the practical guide students will then be able to relate and identify what resources they need to carry out the task at hand. Depending on the nature of the practical there may be need for students to do their own research and mathematical calculations before working on the system. This is to ensure that they set the right inputs to the system and achieve feasible results after completing the practical [8], [9].

However during the practical it is never guaranteed that the system will respond in a friendly manner, at times students may have miscalculated parameters or set them wrongly. Hence this will have an impact on the system as it may be damaged or malfunction to the extent that it may cause harm to the students. Thus there is need to ensure that students are safe at all times.Virtual simulation laboratories have made it possible for students to remain safe while conducting practicals [10].

The research presented in this paper seeks to address some of the highlighted problems as encountered in engineering education.

III. Mathematical Modeling

In this paper a coupled tank system is used as the process to be controlled. This particular process was chosen mainly because of its applicability to the process industry. To give an idea, coupled tank systems are used in quite a number of applications including [11]:

- Beverage industry - cleaning of packaging bottles & mixing of products
- Coal power station - boilers used to produce steam
- Water treatment plant - cleaning, purification and storage of water

The numerous applications of coupled tank systems make it important to give engineering students a strong foundation in aspects such as system modeling and control related to coupled tanks. The layout of the developed model (based on an actual educational dual-tank plant) is shown in Fig. 3 and consists of two tanks that have equal capacity, three electronic valves, water level sensors and one pump that feeds the first tank [7], [12]. The actual dual-tank parameters are given in TABLE I.

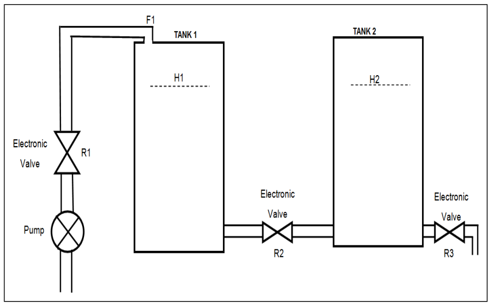


Fig. 3. Coupled Tank System

A. Equations & Parameter Identification

In this section, the main focus is on the theoretical equations and justifications used to setup the model. The model is designed with the following assumptions [13]:

- No pressure loss within the system (transfer pipes, valves, etc.)
- The effect of flow rate is from valves only (pipe lengths have been excluded)

The plant parameters are defined as follows:
Q = flowrate (m^3s^{-1})
A_c = valve cross sectional area (m^2)
d = valve diameter (m)
ΔH = pump head (m)
A_1 = surface area of Tank 1 (m^2)
A_2 = surface area of Tank 2 (m^2)
R_1 = linear resistance of valve 1 (sm^{-2})
R_2 = linear resistance of valve 2 (sm^{-2})
R_3 = linear resistance of valve 3 (sm^{-2})

TABLE I. VARIABLES USED TO MODEL THE PLANT

Parameter	Value	Units
Pump Speed	2900	rpm
Pump Discharge	40	l/min
Tank Dimensions $L \times W \times H$	140, 130, 300	mm
gravitational acceleration (g)	9.81	ms^{-2}

1) Pump: The default pump parameters (TABLE I.) have been used for the simulated model [7].

For any given vessel that is under the influence of atmospheric pressure or gravity, the flow rate from one vessel to another is given as (assuming that the valve is circular in shape [13]):

$$A_c = \frac{\pi d^2}{4}\ thus\ Q = A_C\sqrt{(2gh)} \qquad (1)$$

Thus the pump head per unit time is equivalent to h and maybe given as:

$$\Delta H(t) = \frac{Q^2}{2gA_c^2} \qquad (2)$$

2) Tank 1: Tank 1 is modeled into MATLAB SIMULINK following the differential equation formulated to give the height of the fluid in the tank at any point in time $h_1(t)$. Tank 1 differential equation is derived as [13], [14]: Flow rate into tank 1 ,

$$A_1\frac{dh_1(t)}{dt} = \frac{\Delta H - h_1(t)}{R_1} - \frac{h_1(t)}{R_2} \qquad (3)$$

Thus the final differential equation for Tank 1 is given as [13], [14]:

$$\frac{dh_1(t)}{dt} = \frac{\Delta H - h_1(t)}{A_1 R_1} - \frac{h_1(t)}{A_1 R_2} \qquad (4)$$

3) Tank 2: The differential equation formulated to give the height of the fluid in the tank at any point in time $h_2(t)$

$$A_2\frac{dh_2(t)}{dt} = \frac{h_1(t)}{R_2} - \frac{h_2(t)}{R_3} \qquad (5)$$

Thus the final differential equation for Tank 2 is

$$\frac{dh_2(t)}{dt} = \frac{h_1(t)}{A_2 R_2} - \frac{h_2(t)}{A_2 R_3} \qquad (6)$$

IV. MATLAB SIMULINK MODELING

MATLAB is a multi-variable and complex platform that is optimized for solving engineering and scientific problems. SIMULINK is a graphical programming environment for modeling, simulating and analyzing multi-domain dynamic systems that works on the MATLAB platform [15]. It includes a comprehensive library of predefined blocks to be used to construct graphical models of systems using drag-and-drop mouse operations. It supports linear and nonlinear systems, modeled in continuous-time, sampled time, or a hybrid of the two [8], [16].

Since students learn efficiently with frequent feedback, the interactive nature of SIMULINK encourages one to experiment by changing parameters and observing the immediate effects. The SIMULINK model of the coupled tank system is shown in Fig. 4 where Tank 1 has a green border and Tank 2 has a red boarder. For control and monitoring of the model, an S7-1200 PLC was used [17]. The interfacing of the controller to the model is discussed further in the next section.

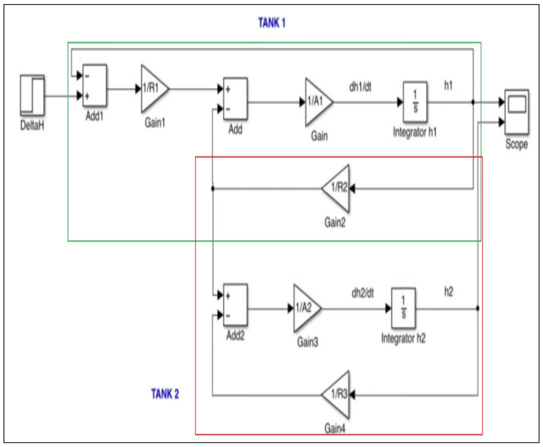


Fig. 4. Coupled Tank SIMULINK Model

V. COMMUNICATION OF MATLAB MODEL & PLC

The interfacing of different software and hardware for any process control system directly affects the performance and functionality of the system [18]. For this particular system the TCP/IP communication protocol has been used and the setup is shown in Fig. 5 [17], [15]. The framework of the TCP/IP protocol defines the actual source and setup of the Internet. Hence it is also employed in private communication networks for remote accessing in automated process control systems [16], [19]. The respective communication blocks have been programmed and set with identical IP and port values (taken from PLC). The communication architecture is illustrated in Fig. 5. PLC and SIMULINK model data is grouped in an array that is passed to and from the communication blocks. To keep and use a specific parameter/variable the index of the variable in the array

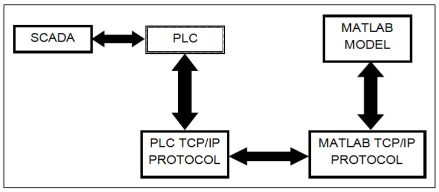


Fig. 5. Communication Architecture between PLC & Coupled Tank Model

must be noted at all times. Array size and dimension of data being transferred between the PLC & SIMULINK must be the same and are declared in both TIA Portal and Simulink. The validation and feasibility of this communication protocol was carried out with a Real-Time Interaction of PLC & SIMULINK Model test.

VI. System Tests & Results

A. *Real-Time Interaction of PLC & SIMULINK Model*

Having the PLC & SIMULINK communicate with one another there is a need to monitor the response and transfer of data between the two for the sake of measuring repeatability, accuracy and reliability. To achieve this a calculation test was carried out whereby both the PLC and SIMULINK model where implementing an algorithm following the setup shown in Fig. 6. The algorithm involved counting, adding and multiplication, whereby both PLC and Model are triggered by a switch once the TCP/IP connection has been established. This arithmetic algorithm is basically an incrementing and decrementing calculation which depends on the range of a counter value. After every calculation the previous value is stored and the new value is updated after checking the counter value at that point in time. Fig. 7 shows the flow diagram of how the algorithm is implemented.

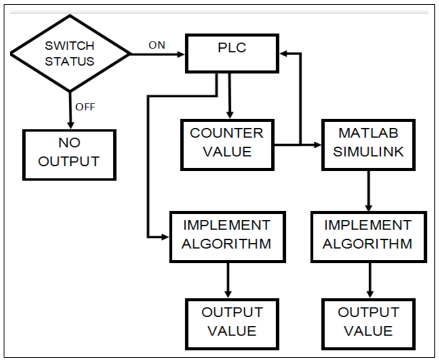


Fig. 6. Setup for PLC & MATLAB Communication Test

Parameters:
$x(t)$ - Output value calculated changing with time
s_i - Sign value for incrementing or decrementing
c_v - Counter value from PLC
a - Chosen point of counter value used to determine s_i
s_i = -1 if $c_v < a$, s_i = 1 if $c_v > a$
Algorithm

$$x(t+1) = (x(t) * s_i) + 1 \qquad (7)$$

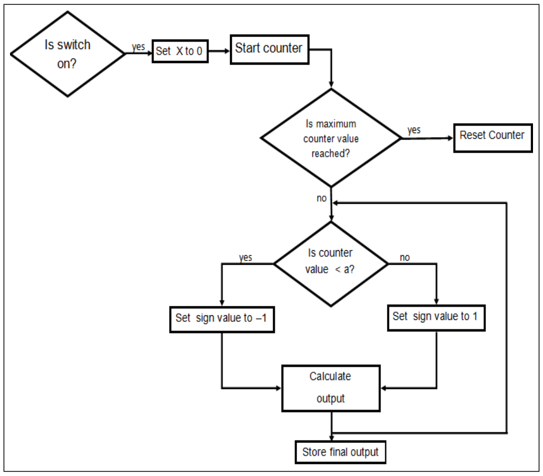


Fig. 7. Arithmetic Program for testing PLC & MATLAB Communication

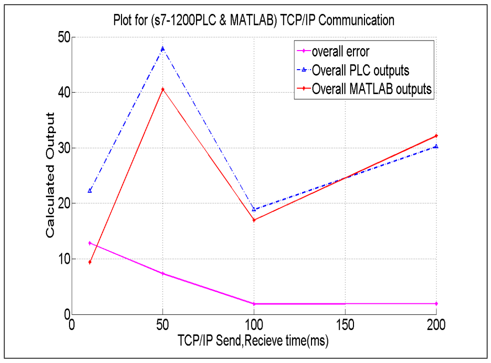


Fig. 8. PLC & MATLAB Overall Calculated Outputs

For each PLC TCP/IP SEND time (TABLE II) five tests were done and values within the same range were taken. The results of the test show that the PLC output is always greater than that of SIMULINK, as seen in Fig. 9 & Fig. 10. This is due to the fact that the PLC algorithm was sending calculated output to SIMULINK and receiving the updated value while in the SIMULINK environment the only variable being received was the counter value from PLC. Hence this results in the PLC implementing the algorithm at a slower rate and may continuously add on or subtract a value such that it is of larger magnitude than that of SIMULINK.

Looking at TABLE II & Fig. 8, it is clear that as the value of the PLC TCP/IP SEND increases the error between PLC and SIMULINK output reduces and both SIMULINK and the PLC are communicating in less time. Values that gave a good

TABLE II. REAL-TIME TEST OVERALL OUTPUTS

PLC TCP/IP SEND (ms)	Overall PLC output	Overall MATLAB output	Overall Error of Output
10	22.24	9.4143	12.829
50	47.847	40.519	7.329
100	18.841	16.985	1.856
200	32.135	30.235	1.899

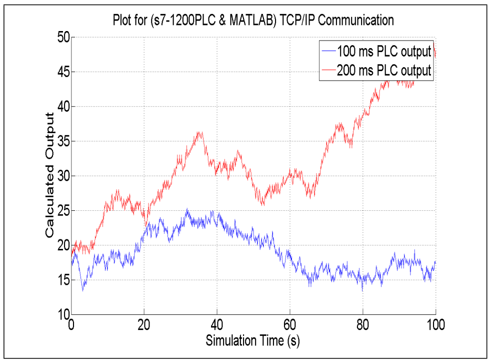


Fig. 9. PLC Calculated Outputs for 100ms & 200ms PLC TCP Send time

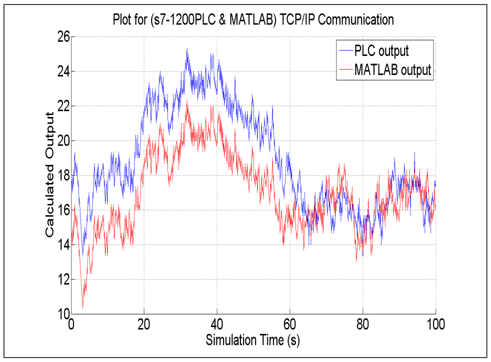


Fig. 10. PLC and MATLAB 100ms Calculated Outputs

communication for the protocol are 100ms and 200ms while much large values made the actual simulation much slower. However from the two values , the 100ms time had a lower overall magnitude as shown in Fig. 9 and the simulation itself took less time than that of the 200ms value. Hence the PLC TCP/IP protocol was set to update every 100ms.

TABLE III. PLC & MATLAB MODEL CONTROL I/OS

Parameter	SIMULINK Environment	TIA Environment
Fluid level in Tanks(1&2)	read & written	read & written
K_p, K_d	read	read & written
Pump(activation/deactivation)	read	read & written
Set Point for tank levels	read	read & written
Valve (1,2&3) open/close	read	read & written

B. MATLAB Model Control with PLC

After setting up the feasible communication parameters between the TIA and SIMULINK environment, the coupled tank model was then interfaced to the PLC. The setup was done in a manner that allows the PLC to have full control of the model and the I/0s are all shown in TABLE III, each parameter may either be read and/or written in the defined environment. A parameter that is read can only be used for a certain calculation in that environment and cannot be altered. A parameter which can be written can be altered in the environment it has been transferred to. Fig. 11. shows the hardware used to control the model which consists of PLC , (PPB - pump push button,V1PB - valve 1 push button,V2PB - valve 2 push button,V3PB - valve 3 push button) and their LED indicators that are directly above them. In the instance that the pump LED is on this indicates that the pump is running and for the valve LEDs the valves are open.

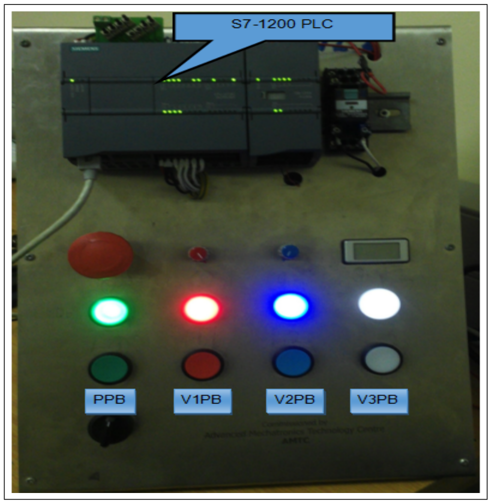


Fig. 11. PLC Control I/Os

In any process control system there is need for feedback to monitor and ensure stability in the implementation of the process. Hence for this particular model a PD controller was implemented via the PLC.

PD controllers (Fig. 12) use two basic behavior parameters: P - proportional and D - derivative. A stand-alone P controller cannot stabilize higher order processes, hence it is used in different combinations with other parameters. When P controller is used, large gain is needed to improve steady state error. Stable systems do not have problems when large gain is used [14]. If constant steady state error can be accepted with such processes, then P controller can be used. Small steady state errors can be accepted if sensor will give measured value with error or if importance of measured value is not too great anyway. For a PD Controller the D parameter is used when prediction of the error can improve control or when it necessary to stabilize the system.

Often derivative is not taken from the error signal but from the system output variable. This is done to avoid effects of the sudden change of the reference input that will cause sudden change in the value of error signal [14]. Sudden change in error signal will cause sudden change in control output. To avoid that it is suitable to design D parameter to be proportional to the change of the output variable. PD controller is often used in control of moving systems such as submarines, ships, rockets etc. Concept is however not limited to PD control. This controller is used as a simple means of testing and proving the concept.

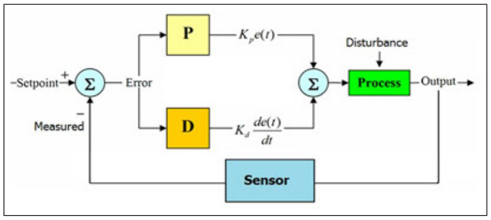


Fig. 12. PD Controller [14]

Fig.13, shows the PLC program layout which has six blocks that serve different tasks in the control of the coupled tank system:

- Cyclic interrupt block implements the PD control of the model
- Main block runs the entire program and implements all the program calculations
- RECEIVED Data block stores data transferred from the SIMULINK environment in the form of an array
- SEND Data block stores data transferred to the SIMULINK environment in the form of an array
- TRCV C Data block implements TCP/IP communication with SIMULINK model and writes data transferred from the model to the RECEIVED Data block
- TSEND C Data block implements TCP/IP communication with SIMULINK model and writes data transferred to the model to the SEND Data block

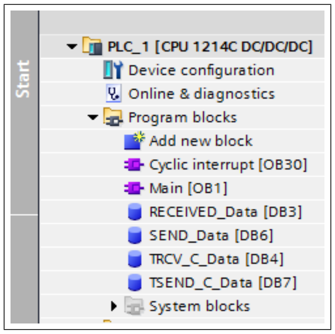

Start
PLC_1 [CPU 1214C DC/DC/DC]
Device configuration
Online & diagnostics
Program blocks
Add new block
Cyclic interrupt [OB30]
Main [OB1]
RECEIVED_Data [DB3]
SEND_Data [DB6]
TRCV_C_Data [DB4]
TSEND_C_Data [DB7]
System blocks

Fig. 13. PLC Program Layout

Hence the PLC and SIMULINK model were fully integrated together and gave a functionality that presents a typical process control system. Fig. 14 depicts how the coupled tank system will operate without any form of control and feedback, once the pump is switched on fluid is pumped into the tank continuously and such a system is unstable (overflow of fluid and not meeting set point required for the process). Hence the stability of the system was achieved by implementing the PD controller as shown in Fig. 15.

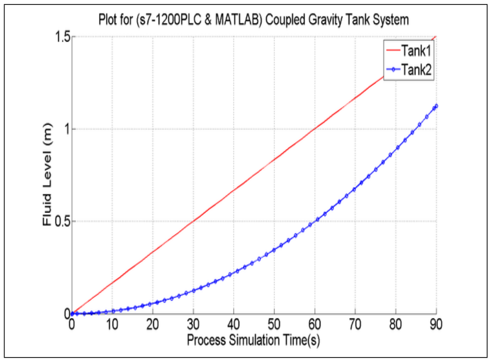


Fig. 14. PLC & MATLAB Model Control Test without control

Tank1 and Tank2 fluid levels rise up until they reach steady state mode close to the set-point. Tank1 reaches steady state faster than Tank2 due to the fact that there is direct inflow from the pump while Tank2 is directly dependent on the outflow of Tank1 .

The responsiveness of valve manipulation by the control box to the system was done and is shown in Fig. 16 and Fig. 17. These tests that were implemented are described in TABLE IV.

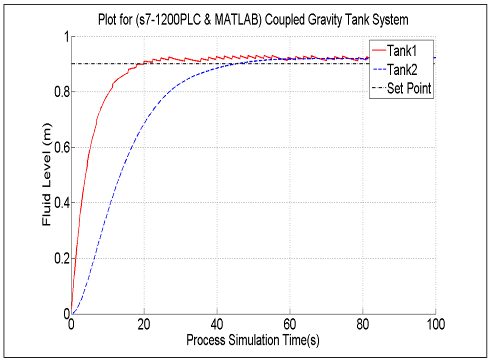


Fig. 15. PLC & MATLAB Model control Test 1

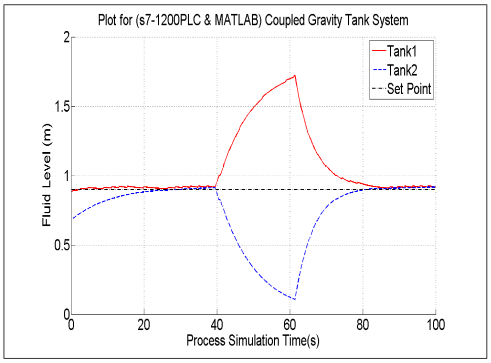


Fig. 16. PLC & MATLAB Model control Test 2

VII. Conclusion

In conclusion common problems often encountered in Engineering institutions which involve **"the teaching of students with real industry equipment in the most cost effective way were looked at"**[10]. Hence a list of objectives that try to reduce some of the problems was derived in the development of this platform [20], [17]. The platform that has been developed as an alternative to a real system that is currently being used for Process Control classes at NMMU. However this platform can offer typical process control exposure to students, as they can do both theoretical calculations and input parameters getting output that gives a clear picture of the system operation. The validation of this platform can be done by running similar tests on the real model and adjusting parameters to obtain similar results. Some of the learning concepts that can be offered from the platform include:

- The manipulation of multi - I/O systems to meet the desired output

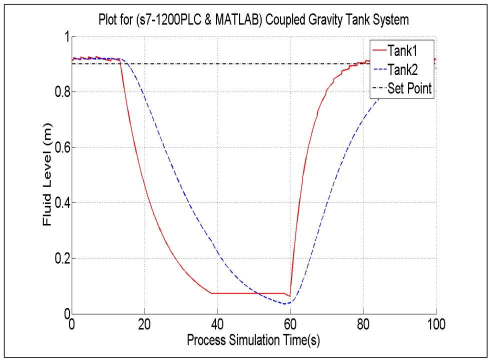


Fig. 17. PLC & MATLAB Model control Test 3

TABLE IV. TEST RESULTS FOR PLC & SIMULINK MODEL CONTROL BOX MANIPULATION

Simulation Test	**Time range(s)**	**Valve(s) Status**	**Tank Fluid Levels**
Fig. 15	0-100	All valves are open	Tank1 rises and reaches a steady state point after 20 seconds which is above the set-point 0.9m , while Tank2 also reaches steady at approximately 60 seconds
Fig. 16	40 - 80	Valve1 open,Valve2 closed ,Valve3 open	Initially Tank1 and Tank2 had reached the steady state points in the first 40 seconds, after closing Valve 2 Tank1 level immediately rose which resulted in an overflow as the fixed tank level is 1m while Tank2 level drops discharging all the fluid through Valve 3 that is open,at the point just after 60 seconds Valve 2 was then opened and both tanks reached their initial steady state levels after 80 seconds
Fig. 17	18-60	Valve1 closed,Valve2 open,Valve3 open	Initially Tank1 and Tank2 had reached the steady state points in the first 18 seconds,Valve 1 was then closed,both tank levels dropped .At 60 seconds the Valve 1 was then opened and both tank levels began to rise as there was inflow to the system

- Understand outputs from curves of input values/peripherals and observe response of system
- Carry out statistical & data analysis (repeat tests,change variables ,compare the differences observed and state variables that give feasible results)
- Design and implement a variety of controllers(e.g. PD, PI, PID, Intelligent controllers,State feedback)
- The system can be re-configured (designing different tank systems in MATLAB SIMULINK) such that students submit original work

However the system can further be improved by adding a safety program that will also give a clear understanding of how to operate a real process control system. For any automated process control system there is always need to emphasize on safety as this a universal standard in the engineering field. Having this platform developed will mean that multiple stations equivalent to real models may be developed for different engineering disciplines and courses (e.g. Chemical, Marine, Mechanical , Mechatronics).

Acknowledgments

The authors would like to thank the MEERSETA in collaboration with AMTC (Advanced Mechatronic Training Centre) NMMU for providing all the equipment and resources used to carry out this research.

References

[1] E. Arroyo, M. Hoernicke, P. Rodrguez, and A. Fay, “Automatic derivation of qualitative plant simulation models from legacy piping and instrumentation diagrams,” *Computers and Chemical Engineering*, vol. 92, pp. 112–132, 2016.

[2] R. K. Cox, J. F. Smith, and Y. Dimitratos, “Can simulation technology enable a paradigm shift in process control?: Modeling for the rest of us,” *Computers and Chemical Engineering*, vol. 30, no. 1012, pp. 1542–1552, 2006.

[3] A. Delgado, “Automation and control in food production,” *CONTROL SYSTEMS ,ROBOTICS AND AUTOMATION*, vol. 19, 2002.

[4] D. Hodouin, “Methods for automatic control, observation, and optimization in mineral processing plants,” *Journal of Process Control*, vol. 21, no. 2, pp. 211–225, 2011.

[5] A. Lakshmi Sangeetha, B. Naveenkumar, A. Balaji Ganesh, and N. Bharathi, “Experimental validation of pid based cascade control system through scadaplcopc and internet architectures,” *Measurement*, vol. 45, no. 4, pp. 643–649, 2012.

[6] A. Visioli, M. Berenguel, A. Beghi, A. Cervato, and M. Rampazzo, “Ifac workshop on internet based control education ibce15a remote refrigeration laboratory for control engineering education,” *IFAC-PapersOnLine*, vol. 48, no. 29, pp. 25–30, 2015.

[7] D. M. 207022526, “Development of coupled tank level control system,” *EMP 4110 Mechatronics Project*, 2010.

[8] F. Ke, S. Lee, and X. Xu, “Teaching training in a mixed-reality integrated learning environment,” *Computers in Human Behavior*, vol. 62, pp. 212 – 220, 2016.

[9] C. Wagner, A. Schller, C. Fleischacker, and U. Epple, “An educational framework for process control theory and engineering tools,” *IFAC-PapersOnLine*, vol. 49, no. 6, pp. 46 – 51, 2016. 11th {IFAC} Symposium on Advances in Control Education {ACE} 2016Bratislava, Slovakia, 13 June 2016.

[10] F. Rodrguez, J. Guzmn, M. Castilla, J. Snchez-Molina, and M. Berenguel, “A proposal for teaching {SCADA} systems using virtual industrial plants in engineering education,” *IFAC-PapersOnLine*, vol. 49, no. 6, pp. 138 – 143, 2016. 11th {IFAC} Symposium on Advances in Control Education {ACE} 2016Bratislava, Slovakia, 13 June 2016.

[11] B. Voglauer, R. Garcia, and H. P. Jrgl, “{IMPROVEMENTS} {OF} a three-tank-system {OPERATED} {IN} {REAL} {TIME} {WITH} {MATLAB} {IN} a plc-profibus-network,” *{IFAC} Proceedings Volumes*, vol. 38, no. 1, pp. 43 – 48, 2005. 16th {IFAC} World Congress.

[12] Z. Bekirogullari, M. Y. Minas, O. M. Zamyatina, M. G. Minin, D. S. Denchuk, and V. O. Sadchenko, “5th iceepsy international conference on education and educational psychologyanalysis of engineering invention competencies in standards and programmes of engineering universities,” *Procedia - Social and Behavioral Sciences*, vol. 171, pp. 1088–1096, 2015.

[13] Y. C. . R. Turner, *Fundamentals of Thermal Fluid Sciences*. Singapore: McGraw-Hill, 2012.

[14] F. G. . B. C.Kuo, *Automatic Control Systems 9th edition*. United States of America: John Wiley & Sons ,Inc, 2010.

[15] “Matlab url: http://www.mathworks.com/products/matlab/.”

[16] S. Ozana and M. Pies, “9th ifac workshop on programmable devices and embedded systems design and implementation of embedded systems in matlab&simulink environment,” *IFAC Proceedings Volumes*, vol. 42, no. 1, pp. 73 – 79, 2009.

[17] “Siemens url: https://ewon.biz/support/product/cosy-141-plc-area-siemens/siemens-0.”

[18] B.-J. Youm and J. Park, “Tcp/ip {PROTOCOL} {OVER} ieee-1394 {NETWORK} {FOR} real-time {CONTROL} {APPLICATIONS},” *{IFAC} Proceedings Volumes*, vol. 38, no. 1, pp. 37 – 42, 2005. 16th {IFAC} World Congress.

[19] O. Krejcar, P. Konarik, and T. Michal, “9th ifac workshop on programmable devices and embedded systems winplc control from matlab/simulink environment throw the multiplatform control system rex by secured wireless network,” *IFAC Proceedings Volumes*, vol. 42, no. 1, pp. 13 – 16, 2009.

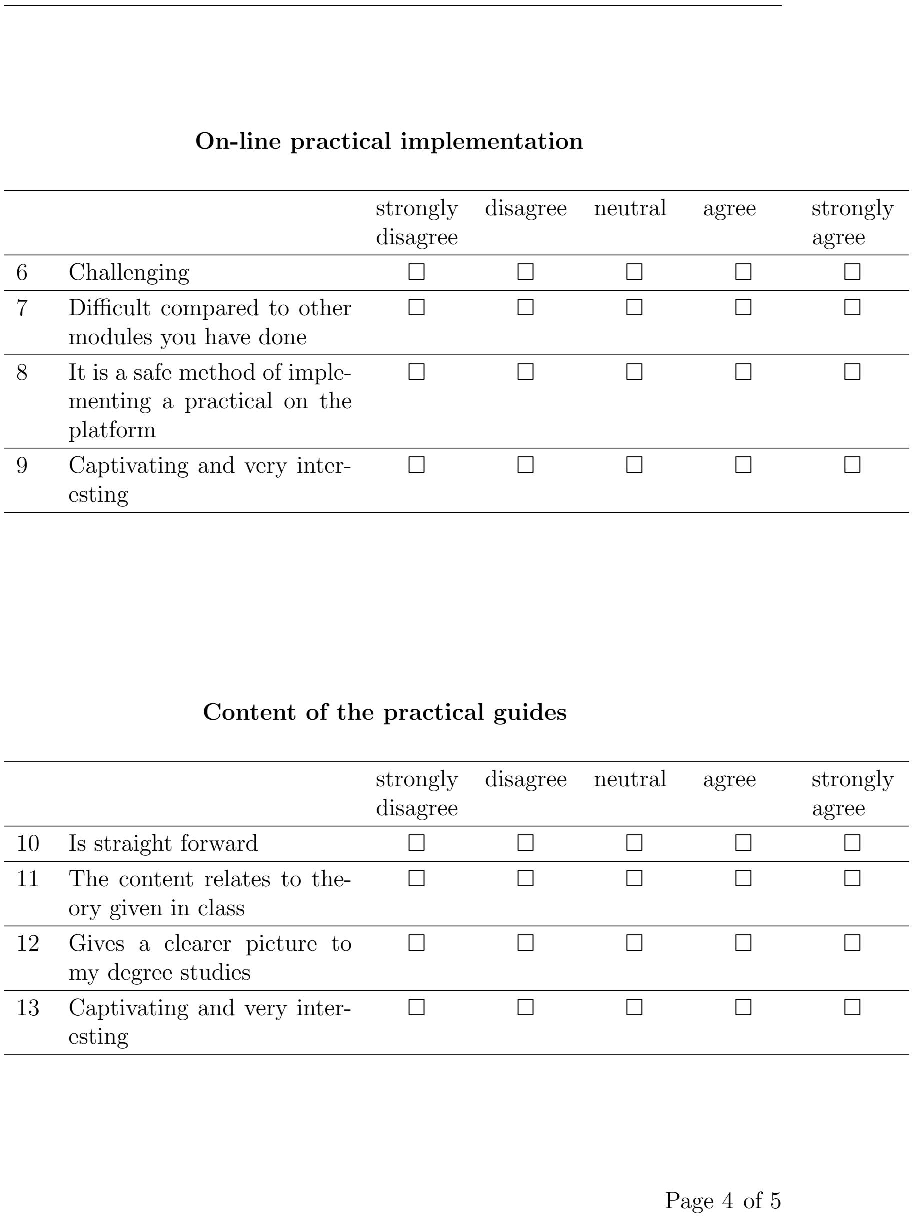
[20] S. Da’na, A. Sagahyroon, A. Elrayes, A. R. Al-Ali, and R. Al-Aydi, “Development of a monitoring and control platform for plc-based applications,” *Computer Standards and Interfaces*, vol. 30, no. 3, pp. 157–166, 2008.

7.2.7 Questionnaire for users

Feed back - questions for Lab 2

Graphical and Visualization of Platform

		strongly disagree	disagree	neutral	agree	strongly agree
1	Emulates a real system	□	□	□	□	□
2	Responds in real-time when parameters are changed	□	□	□	□	□
3	Is easy to navigate through and access different functions	□	□	□	□	□
4	Should further be improved to give more functions	□	□	□	□	□
5	Motivating and captivating	□	□	□	□	□



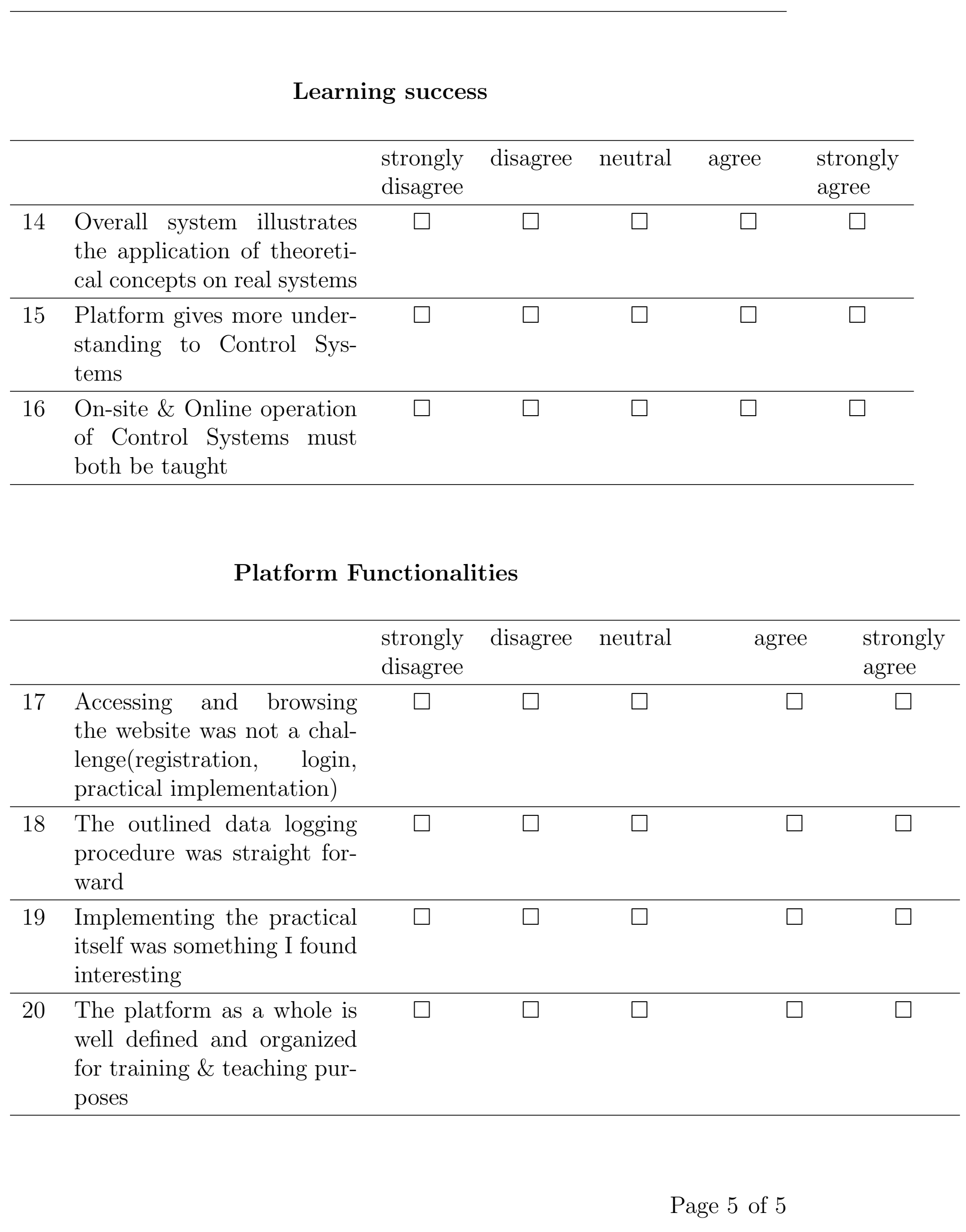
On-line practical implementation

		strongly disagree	disagree	neutral	agree	strongly agree
6	Challenging	□	□	□	□	□
7	Difficult compared to other modules you have done	□	□	□	□	□
8	It is a safe method of implementing a practical on the platform	□	□	□	□	□
9	Captivating and very interesting	□	□	□	□	□

Content of the practical guides

		strongly disagree	disagree	neutral	agree	strongly agree
10	Is straight forward	□	□	□	□	□
11	The content relates to theory given in class	□	□	□	□	□
12	Gives a clearer picture to my degree studies	□	□	□	□	□
13	Captivating and very interesting	□	□	□	□	□

Page 4 of 5



Learning success

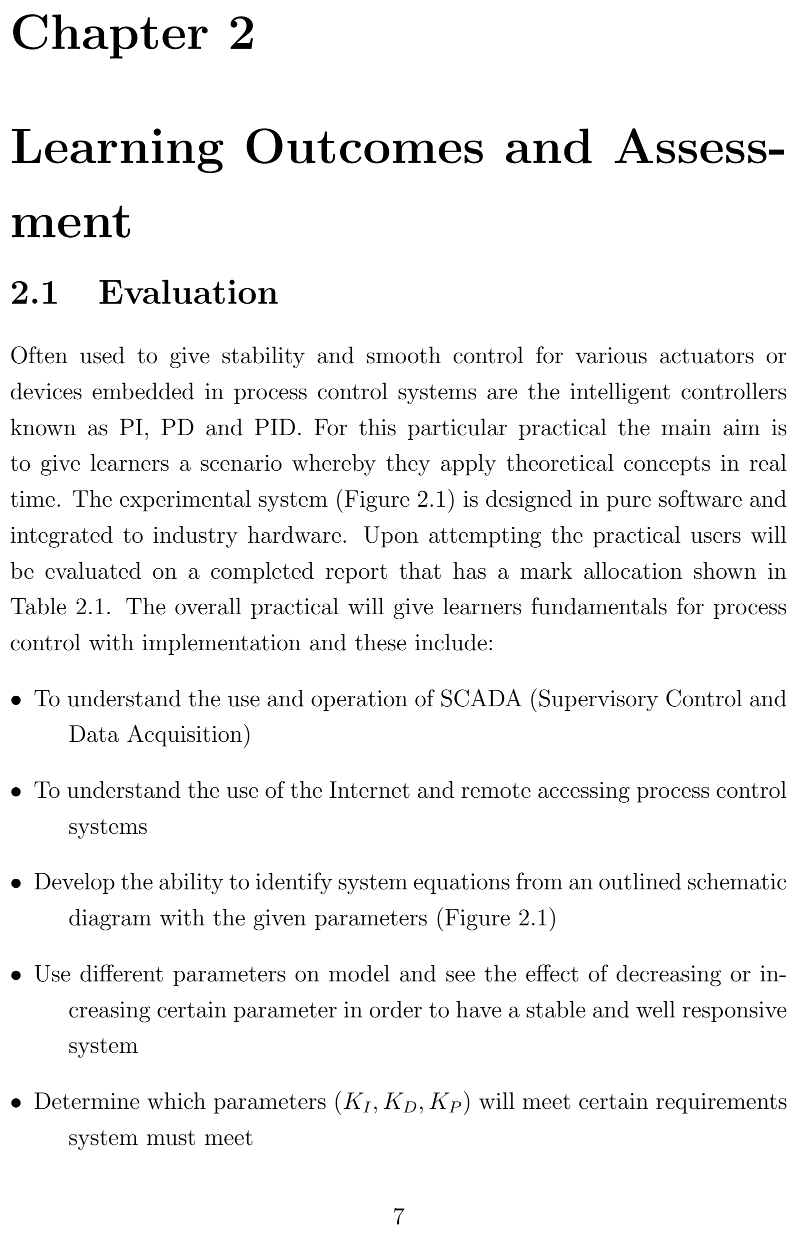
		strongly disagree	disagree	neutral	agree	strongly agree
14	Overall system illustrates the application of theoretical concepts on real systems	□	□	□	□	□
15	Platform gives more understanding to Control Systems	□	□	□	□	□
16	On-site & Online operation of Control Systems must both be taught	□	□	□	□	□

Platform Functionalities

		strongly disagree	disagree	neutral	agree	strongly agree
17	Accessing and browsing the website was not a challenge(registration, login, practical implementation)	□	□	□	□	□
18	The outlined data logging procedure was straight forward	□	□	□	□	□
19	Implementing the practical itself was something I found interesting	□	□	□	□	□
20	The platform as a whole is well defined and organized for training & teaching purposes	□	□	□	□	□

Page 5 of 5

7.2.8 Practical Guide for users



Chapter 2

Learning Outcomes and Assessment

2.1 Evaluation

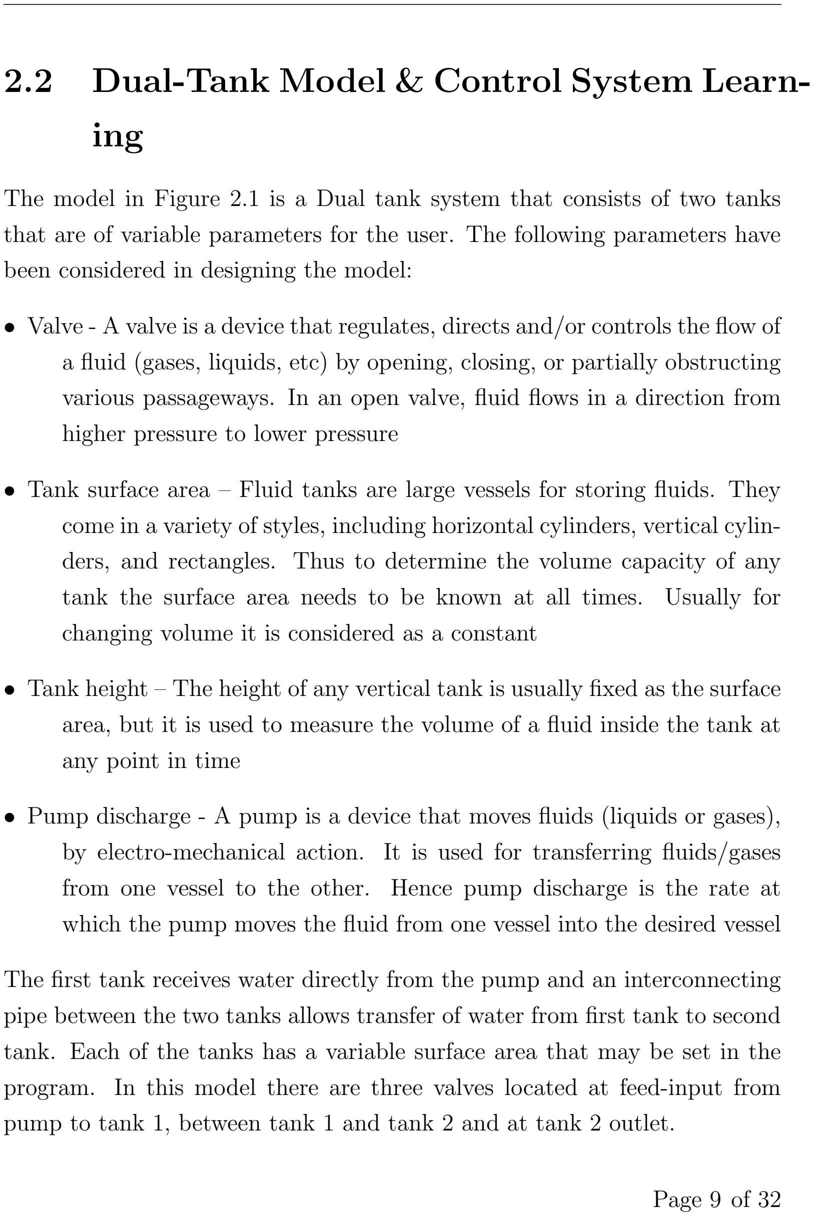
Often used to give stability and smooth control for various actuators or devices embedded in process control systems are the intelligent controllers known as PI, PD and PID. For this particular practical the main aim is to give learners a scenario whereby they apply theoretical concepts in real time. The experimental system (Figure 2.1) is designed in pure software and integrated to industry hardware. Upon attempting the practical users will be evaluated on a completed report that has a mark allocation shown in Table 2.1. The overall practical will give learners fundamentals for process control with implementation and these include:

- To understand the use and operation of SCADA (Supervisory Control and Data Acquisition)
- To understand the use of the Internet and remote accessing process control systems
- Develop the ability to identify system equations from an outlined schematic diagram with the given parameters (Figure 2.1)
- Use different parameters on model and see the effect of decreasing or increasing certain parameter in order to have a stable and well responsive system
- Determine which parameters (K_I, K_D, K_P) will meet certain requirements system must meet

Table 2.1: Practical Mark Allocation

Activity	**Student Mark**	**Marks on activity**
Use given parameters for practical		5
Calculate your T_I values relative to K_P values		6
Determine the system response parameters		36
Graph all activities (Set-point, Tank 1, Tank 2)		30
Derive Tank 1 time domain equation		5
Derive Tank 2 time domain equation		5
Derive Tank 1 Laplace equation with PI Controller		6
Give a solid conclusion on the effect of the variation of different parameters on the system response		8
Total		101

NB : Please ensure that you submit the CSV or excel file that has all the data you recorded during your practical. This is to be done 24 hours before you hand in your final report.
Send this file via email to ngoniezata@gmail.com. Email subject should be : ”Online Dual” Tank If file is not submitted you will be penalized 10% of your final mark.



2.2 Dual-Tank Model & Control System Learning

The model in Figure 2.1 is a Dual tank system that consists of two tanks that are of variable parameters for the user. The following parameters have been considered in designing the model:

- Valve - A valve is a device that regulates, directs and/or controls the flow of a fluid (gases, liquids, etc) by opening, closing, or partially obstructing various passageways. In an open valve, fluid flows in a direction from higher pressure to lower pressure
- Tank surface area – Fluid tanks are large vessels for storing fluids. They come in a variety of styles, including horizontal cylinders, vertical cylinders, and rectangles. Thus to determine the volume capacity of any tank the surface area needs to be known at all times. Usually for changing volume it is considered as a constant
- Tank height – The height of any vertical tank is usually fixed as the surface area, but it is used to measure the volume of a fluid inside the tank at any point in time
- Pump discharge - A pump is a device that moves fluids (liquids or gases), by electro-mechanical action. It is used for transferring fluids/gases from one vessel to the other. Hence pump discharge is the rate at which the pump moves the fluid from one vessel into the desired vessel

The first tank receives water directly from the pump and an interconnecting pipe between the two tanks allows transfer of water from first tank to second tank. Each of the tanks has a variable surface area that may be set in the program. In this model there are three valves located at feed-input from pump to tank 1, between tank 1 and tank 2 and at tank 2 outlet.

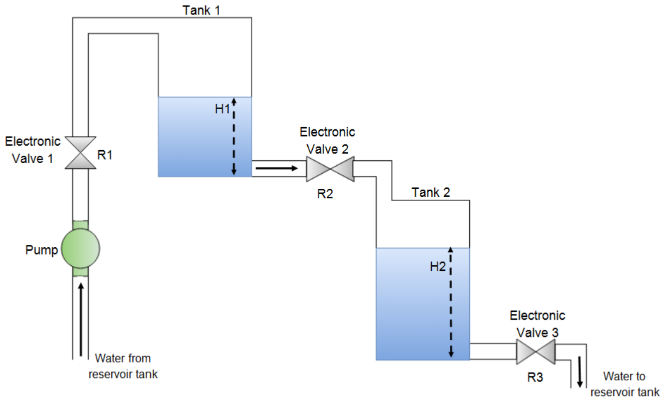


Figure 2.1: Sketch model of Dual Tank System

2.3 Mathematical Modeling for Experiment

The main concept and process control knowledge that can be taught from this model is the use of PID control. The parameter calculation for the PID block is given as:

$$y = K_p[(b.w - x) + \frac{1}{T_1.s}(w - x) + \frac{T_D.s}{a.T_D.s+1}(c.w - x)] \tag{2.1}$$

$$K_I = \frac{K_P}{T_i} \tag{2.2}$$

$$K_D = K_P * T_D \tag{2.3}$$

The variables used in equations 4.3, 4.4 and 4.5 are defined in Table 2.2 An illustration of a typical test and operation of the plant model response is shown in Figure 2.2 , whereby the user is to vary the set-point of plant model to three different states. From a plot of this nature the user can be able to calculate and an overview of the system's response. The critical parameters

Table 2.2: Dual Tank Compact PID parameter definitions

y	Output value	x	Process value
w	Set-point value	s	Laplace operator
K_p	Proportional gain(P component)	a	Derivative delay coefficient(D component)
T_I	Integral action time(I component)	b	Proportional action weighting(P component)
T_D	Derivative action time (D component)	c	Derivative action weighting (D component)

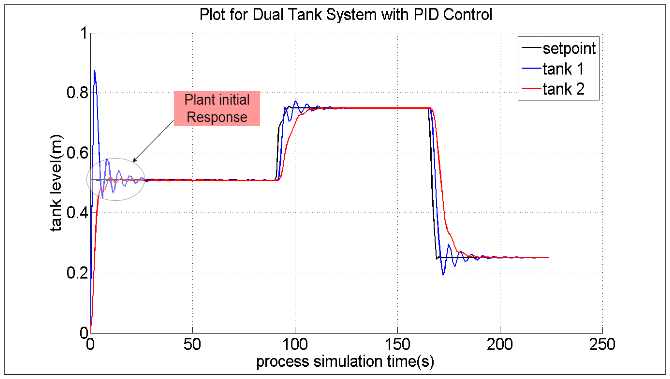


Figure 2.2: Dual Tank operation Illustration

to observe from such data include:

- Maximum overshoot - this is the point at which the model response is highest relative to the set-point
- Rise time(t_r) - the time it takes for the plant model to reach a certain portion of the set point
- T_{max} - the time it takes for the plant model to reach the maximum over-

shoot

- Settling time t_s - the time it takes for the plant model to reach steady state whereby the process value has stabilized

The critical parameters stated are identified as shown in Figure 2.3 for plant initial response. These aid the user to select or adjust parameters to fine-tune the system and reach required and satisfactory functionality. User will also gain knowledge and understand intelligent control methods that are used in Industry and their relevance from a theoretical perspective.

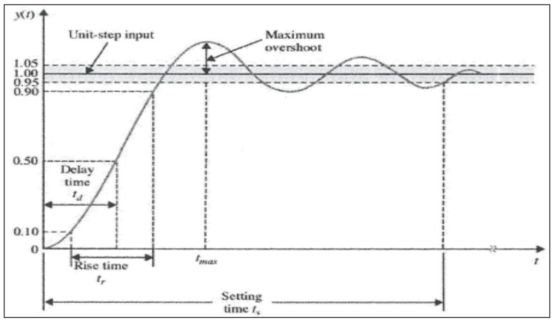


Figure 2.3: Identification of Response values from PID controlled system

Chapter 3

Experiment Setup & User activities

3.1 Input Parameters

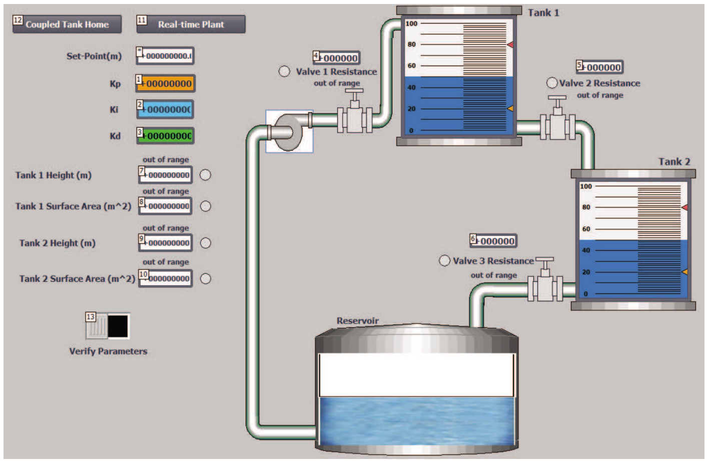


Figure 3.1: Dual Tank Parameters

3.1.1 Step 1

On the parameters tab, please enter the given practical parameters as there will be default values running on the system in Figure 3.1. Once values are entered and verified by system please switch off the pump until all tanks are empty. Reset and keep data recording deactivated. Throughout the

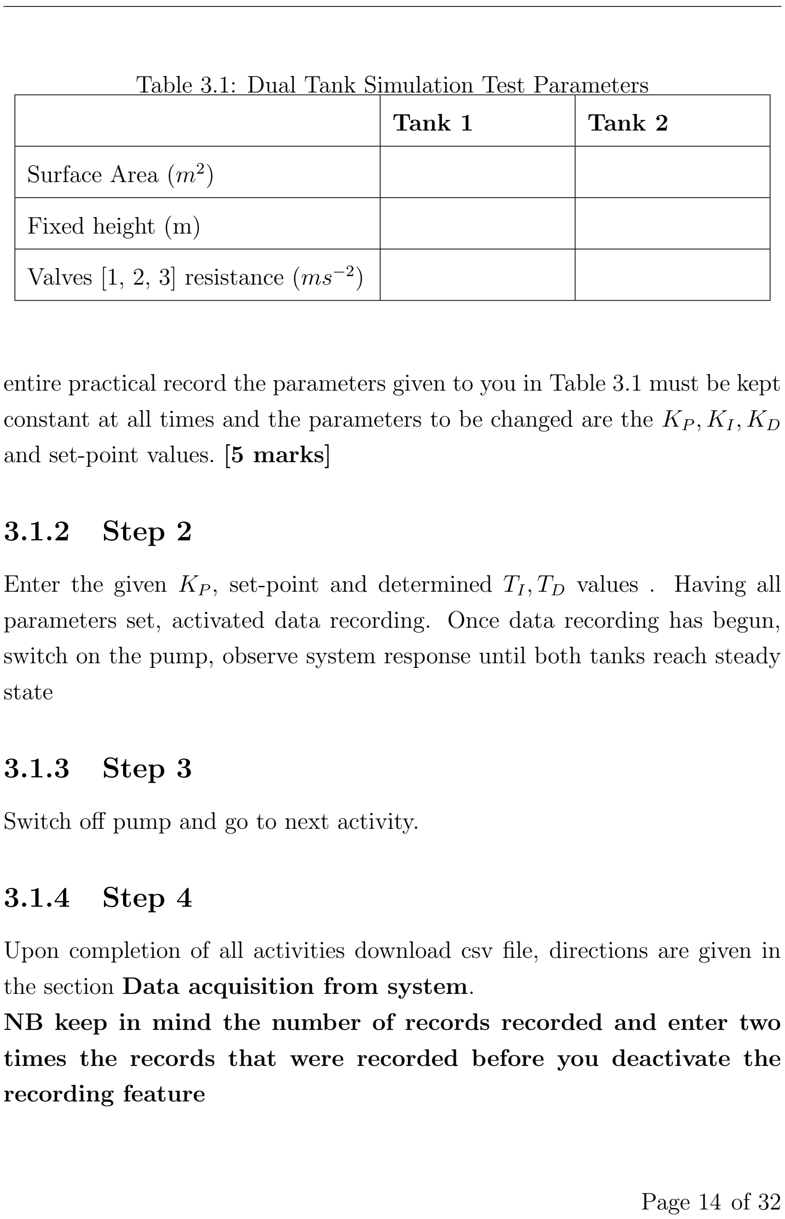


Table 3.1: Dual Tank Simulation Test Parameters

	Tank 1	**Tank 2**
Surface Area (m^2)		
Fixed height (m)		
Valves [1, 2, 3] resistance (ms^{-2})		

entire practical record the parameters given to you in Table 3.1 must be kept constant at all times and the parameters to be changed are the K_P, K_I, K_D and set-point values. [**5 marks**]

3.1.2 Step 2

Enter the given K_P, set-point and determined T_I, T_D values . Having all parameters set, activated data recording. Once data recording has begun, switch on the pump, observe system response until both tanks reach steady state

3.1.3 Step 3

Switch off pump and go to next activity.

3.1.4 Step 4

Upon completion of all activities download csv file, directions are given in the section **Data acquisition from system**.
NB keep in mind the number of records recorded and enter two times the records that were recorded before you deactivate the recording feature

3.2 Data acquisition from system

This section discusses the approach implemented to have I/O parameters from the SCADA written onto a Data Log file and made accessible to the user. The importance of this functionality on the system helps the user to analyze and see how the system behaved when they entered or controlled different parameters on the system.
Figure 3.2 illustrates the algorithm for the data logging procedure implemented on the PLC. Table 3.2 gives the steps for the user to download the file with I/O data recorded by the PLC during practical session.

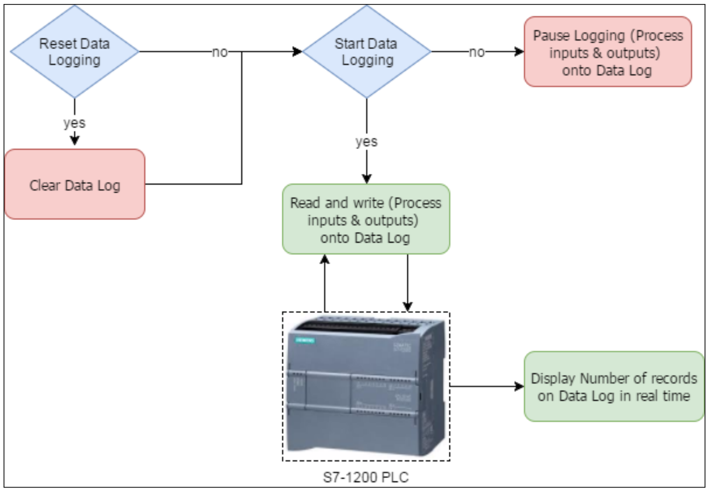
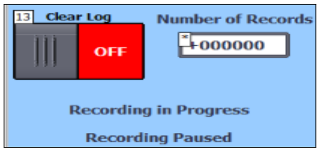
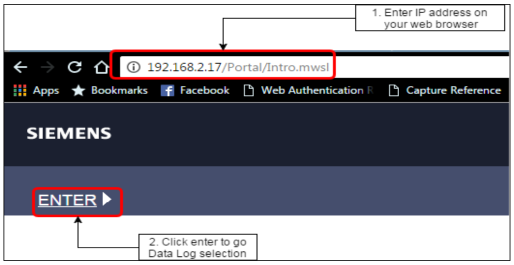
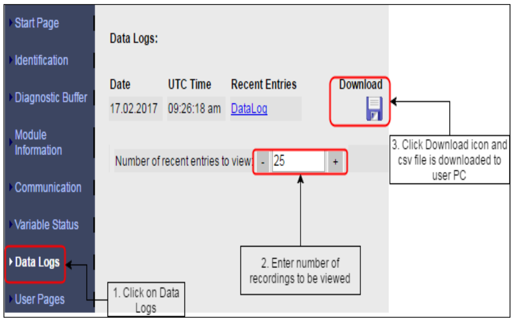
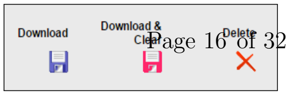
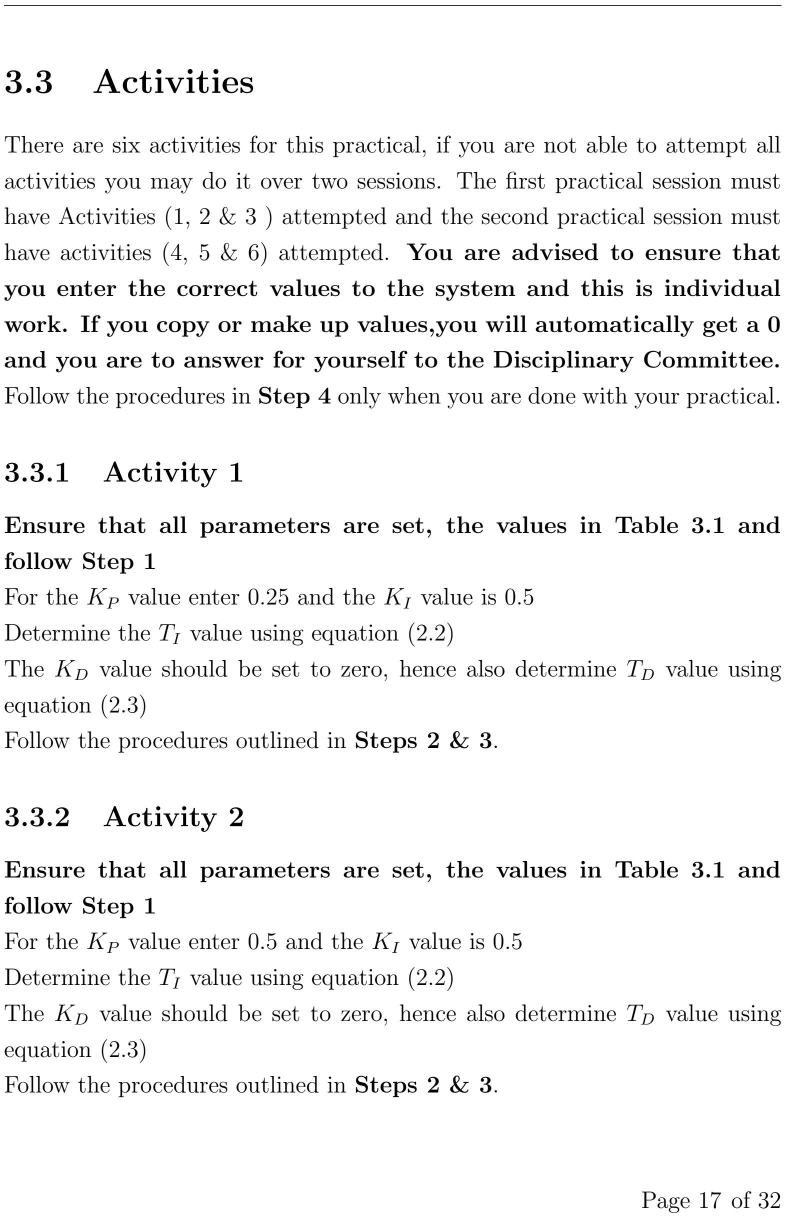

Reset Data Logging
no
yes
Clear Data Log
Start Data Logging
no
Pause Logging (Process inputs & outputs) onto Data Log
yes
Read and write (Process inputs & outputs) onto Data Log
Display Number of records on Data Log in real time
S7-1200 PLC

Figure 3.2: Flow diagram for data logging

Table 3.2: Data Logging for user

Directions to follow	Implementation Illustration
To record/pause, clear or reset the data log. The instance you access the SCADA ensure that the platform is recording select the real plant tab or graphical view tab.	 13 Clear Log OFF Number of Records +000000 Recording in Progress Recording Paused
To retrieve Data Log user must enter the IP address (192.168.2.17) on the same web browser. NB! Download your log file before your practical session is over..	 1. Enter IP address on your web browser 2. Click enter to go Data Log selection 192.168.2.17/Portal/Intro.mwsl Apps Bookmarks Facebook Web Authentication Capture Reference SIEMENS ENTER
Upon clicking the the Enter tab on the Siemens remote server, select the Data Logs tab and the file will be visible on this page. You are advised to enter atleast double the number of entries indicated on the SCADA to obtain all the information. Having the entries entered user must click the download button and a CSV file is downloaded directly onto user PC.	 Data Logs: Date: 17.02.2017, UTC Time: 09:26:18 am, Recent Entries: DataLog, Download Number of recent entries to view: 25 1. Click on Data Logs 2. Enter number of recordings to be viewed 3. Click Download icon and csv file is downloaded to user PC Start Page Identification Diagnostic Buffer Module Information Communication Variable Status Data Logs User Pages
To clear the Data Log on the PLC, Logon to the PLC by entering admin on the user tab, go to Data Logs then select clear and download option	



3.3 Activities

There are six activities for this practical, if you are not able to attempt all activities you may do it over two sessions. The first practical session must have Activities (1, 2 & 3) attempted and the second practical session must have activities (4, 5 & 6) attempted. **You are advised to ensure that you enter the correct values to the system and this is individual work. If you copy or make up values,you will automatically get a 0 and you are to answer for yourself to the Disciplinary Committee.** Follow the procedures in **Step 4** only when you are done with your practical.

3.3.1 Activity 1

Ensure that all parameters are set, the values in Table 3.1 and follow Step 1

For the K_P value enter 0.25 and the K_I value is 0.5

Determine the T_I value using equation (2.2)

The K_D value should be set to zero, hence also determine T_D value using equation (2.3)

Follow the procedures outlined in **Steps 2 & 3**.

3.3.2 Activity 2

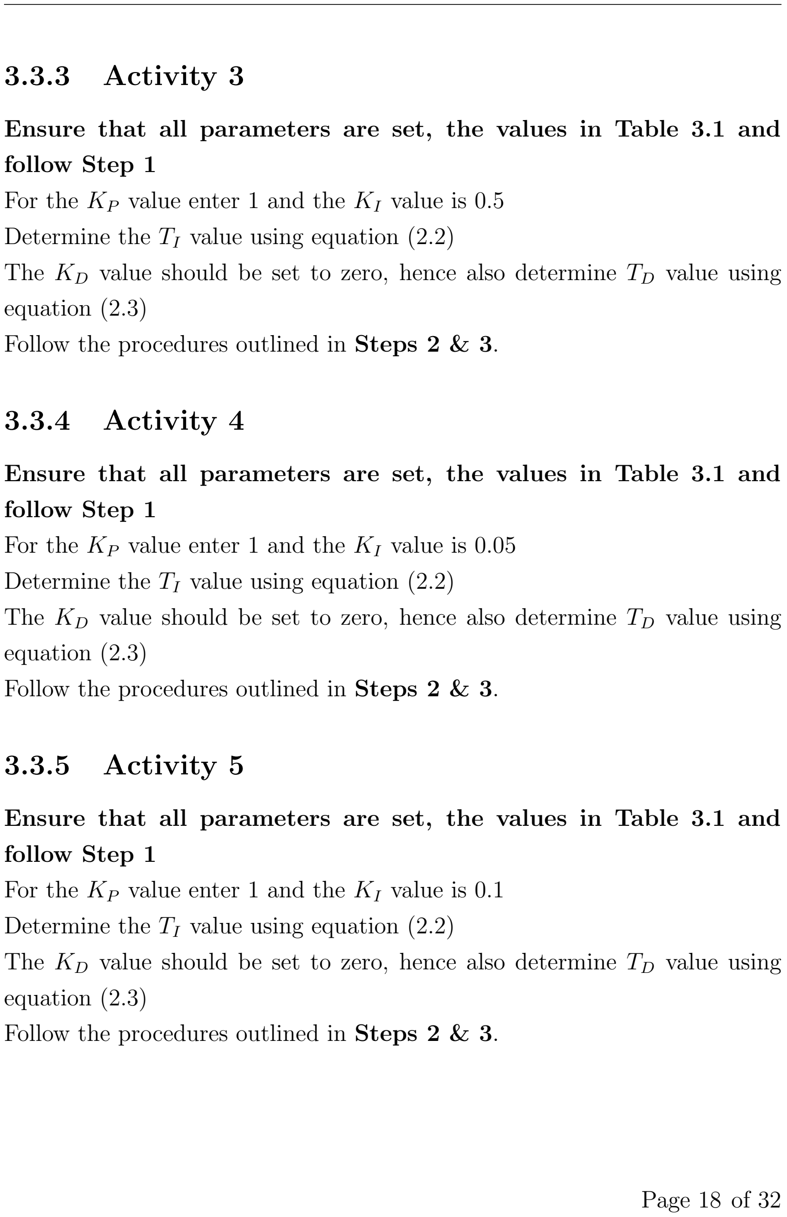
Ensure that all parameters are set, the values in Table 3.1 and follow Step 1

For the K_P value enter 0.5 and the K_I value is 0.5

Determine the T_I value using equation (2.2)

The K_D value should be set to zero, hence also determine T_D value using equation (2.3)

Follow the procedures outlined in **Steps 2 & 3**.



3.3.3 Activity 3

Ensure that all parameters are set, the values in Table 3.1 and follow Step 1

For the K_P value enter 1 and the K_I value is 0.5

Determine the T_I value using equation (2.2)

The K_D value should be set to zero, hence also determine T_D value using equation (2.3)

Follow the procedures outlined in **Steps 2 & 3**.

3.3.4 Activity 4

Ensure that all parameters are set, the values in Table 3.1 and follow Step 1

For the K_P value enter 1 and the K_I value is 0.05

Determine the T_I value using equation (2.2)

The K_D value should be set to zero, hence also determine T_D value using equation (2.3)

Follow the procedures outlined in **Steps 2 & 3**.

3.3.5 Activity 5

Ensure that all parameters are set, the values in Table 3.1 and follow Step 1

For the K_P value enter 1 and the K_I value is 0.1

Determine the T_I value using equation (2.2)

The K_D value should be set to zero, hence also determine T_D value using equation (2.3)

Follow the procedures outlined in **Steps 2 & 3**.

3.3.6 Activity 6

Ensure that all parameters are set, the values in Table 3.1 and follow Step 1
For the K_P value enter 1 and the K_I value is 1
Determine the T_I value using equation (2.2)
The K_D value should be set to zero, hence also determine T_D value using equation (2.3)
Follow the procedures outlined in **Steps 2 & 3**.

3.4 Results

In this section you are to calculate and show various parameters used on system and system response values discussed in the **Mathematical Modeling for Experiment** section.

Table 3.3: Controller Parameters used

Activity	K_P	K_I	K_D	T_I	T_D	**Set-point (m)**
1						
2						
3						
4						
5						
6						

3.4.1 Controller Values

Please record activity values required in Table 3.3 [**6 marks**]

3.4.2 System Response parameters

This section is to be done with all your values for the various activities recorded on a CSV file. You are to determine and show calculations for each activity and system response parameters outlined in Figure 2.3.

$$Tank\ fluid\ level\ (m) = \frac{tank\ level\ CSV}{100} * (Fixed\ tank\ height) \tag{3.1}$$

$$Set\ point\ (m) = \frac{Set\ point\ CSV}{100} \tag{3.2}$$

To obtain the actual plant response parameters , please use equation 3.1 to get the actual tank heights from the recorded data as these are percentage values of the tank. As for the set-point use equation 3.2.

Upon converting these values to measure values you may determine the system response values for tank 1 and tank 2 which are:
Tank overshoot value (m)
Tank overshoot value %
Tank rise-time (s)
Tank settling-time (s)

Activity 1 Response Values

Tank 1 overshoot value (m): _________ [**0.5 mark**]

Tank 1 overshoot value %: _________ [**0.5 mark**]

Tank 1 rise-time (s): _________ [**1 mark**]

Tank 1 settling-time (s): _________ [**1 mark**]

Tank 2 overshoot value (m): _________ [**0.5 mark**]

Tank 2 overshoot value %: _________ [**0.5 mark**]

Tank 2 rise-time (s): _________ [**1 mark**]

Tank 2 settling-time (s): _________ [**1 mark**]

Activity 2 Response Values

Tank 1 overshoot value (m): __________ [**0.5 mark**]

Tank 1 overshoot value %: __________ [**0.5 mark**]

Tank 1 rise-time (s): __________ [**1 mark**]

Tank 1 settling-time (s): __________ [**1 mark**]

Tank 2 overshoot value (m): __________ [**0.5 mark**]

Tank 2 overshoot value %: __________ [**0.5 mark**]

Tank 2 rise-time (s): __________ [**1 mark**]

Tank 2 settling-time (s): __________ [**1 mark**]

Activity 3 Response Values

Tank 1 overshoot value (m): __________ [**0.5 mark**]

Tank 1 overshoot value %: __________ [**0.5 mark**]

Tank 1 rise-time (s): __________ [**1 mark**]

Tank 1 settling-time (s): __________ [**1 mark**]

Tank 2 overshoot value (m): __________ [**0.5 mark**]

Tank 2 overshoot value %: __________ [**0.5 mark**]

Tank 2 rise-time (s): __________ [**1 mark**]

Tank 2 settling-time (s): __________ [**1 mark**]

Activity 4 Response Values

Tank 1 overshoot value (m): __________ [**0.5 mark**]

Tank 1 overshoot value %: __________ [**0.5 mark**]

Tank 1 rise-time (s): __________ [**1 mark**]

Tank 1 settling-time (s): __________ [**1 mark**]

Tank 2 overshoot value (m): __________ [**0.5 mark**]

Tank 2 overshoot value %: __________ [**0.5 mark**]

Tank 2 rise-time (s): __________ [**1 mark**]

Tank 2 settling-time (s): __________ [**1 mark**]

Activity 5 Response Values

Tank 1 overshoot value (m): __________ [**0.5 mark**]

Tank 1 overshoot value %: __________ [**0.5 mark**]

Tank 1 rise-time (s): __________ [**1 mark**]

Tank 1 settling-time (s): _________ [**1 mark**]

Tank 2 overshoot value (m): _________ [**0.5 mark**]

Tank 2 overshoot value %: _________ [**0.5 mark**]

Tank 2 rise-time (s): _________ [**1 mark**]

Tank 2 settling-time (s): _________ [**1 mark**]

Activity 6 Response Values

Tank 1 overshoot value (m): _________ [**0.5 mark**]

Tank 1 overshoot value %: _________ [**0.5 mark**]

Tank 1 rise-time (s): _________ [**1 mark**]

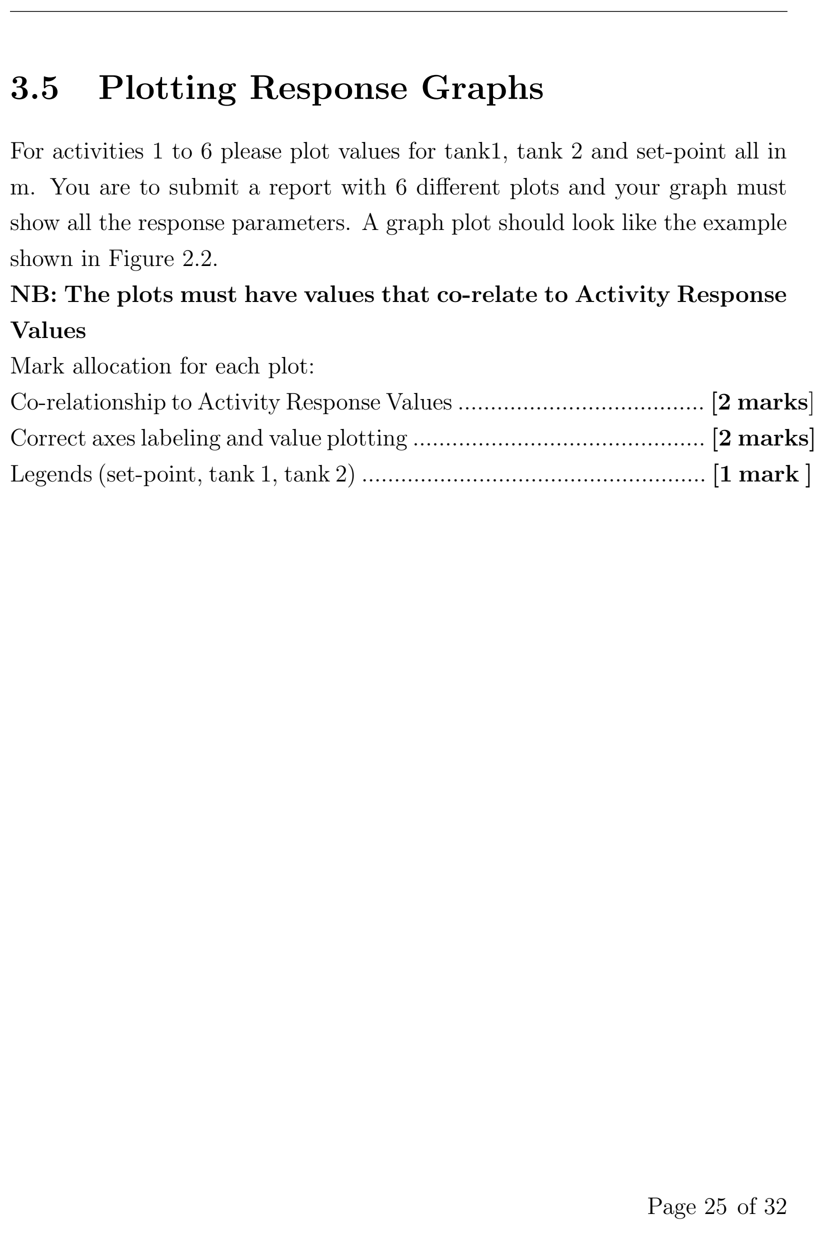
Tank 1 settling-time (s): _________ [**1 mark**]

Tank 2 overshoot value (m): _________ [**0.5 mark**]

Tank 2 overshoot value %: _________ [**0.5 mark**]

Tank 2 rise-time (s): _________ [**1 mark**]

Tank 2 settling-time (s): _________ [**1 mark**]



3.5 Plotting Response Graphs

For activities 1 to 6 please plot values for tank1, tank 2 and set-point all in m. You are to submit a report with 6 different plots and your graph must show all the response parameters. A graph plot should look like the example shown in Figure 2.2.

NB: The plots must have values that co-relate to Activity Response Values

Mark allocation for each plot:

Co-relationship to Activity Response Values	[**2 marks**]
Correct axes labeling and value plotting	[**2 marks**]
Legends (set-point, tank 1, tank 2)	[**1 mark**]

3.6 Deriving Tank 1 & Tank 2 time equations

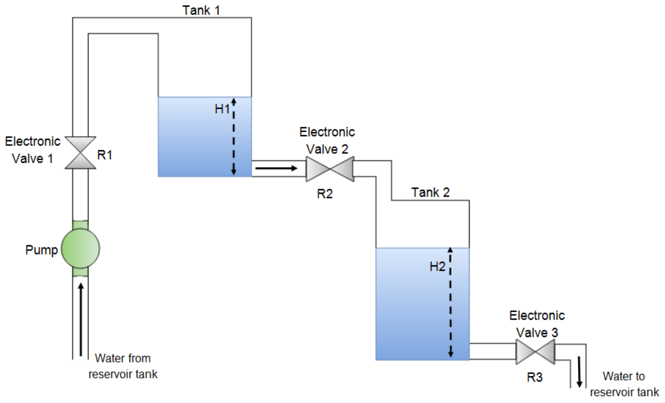


Figure 3.3: Sketch model of Dual Tank System

The differential equations for tank 1 and tank 2 using Figure 3.3:

[10 marks]

3.7 Deriving Tank 1 Laplace equation and Integrating to PI Controller

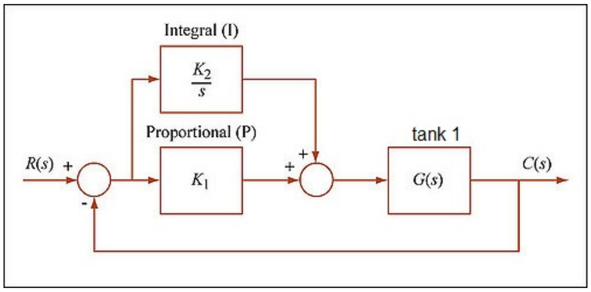


Figure 3.4: System with implemented PI Controller

The Laplace equation for tank 1 with a PI Controller using Figure 3.4:

[6 marks]

3.8 Conclusion on experiment

Please give a thorough conclusion on the system which includes the following (relationship between tank 1 and tank 2, response to variation of parameters and differences observed).

[8 marks]

7.3 Appendix C : Remote Accessing Website Code

The remote access website code with structure shown in Figure 7.14 is documented in this section. The website was programmed using Visual Studio ASP.NET web application developer. For the front end display that the user navigates through or widgets they click or drag, these are all coded in HTML syntax. The code that executes certain instructions reading inputs or responses from the front end was coded in C#. The C# interface and HTML code make up each functional page. For each page the HTML code is documented followed by the C# code. Certain functions and critical algorithms are described by the green font.

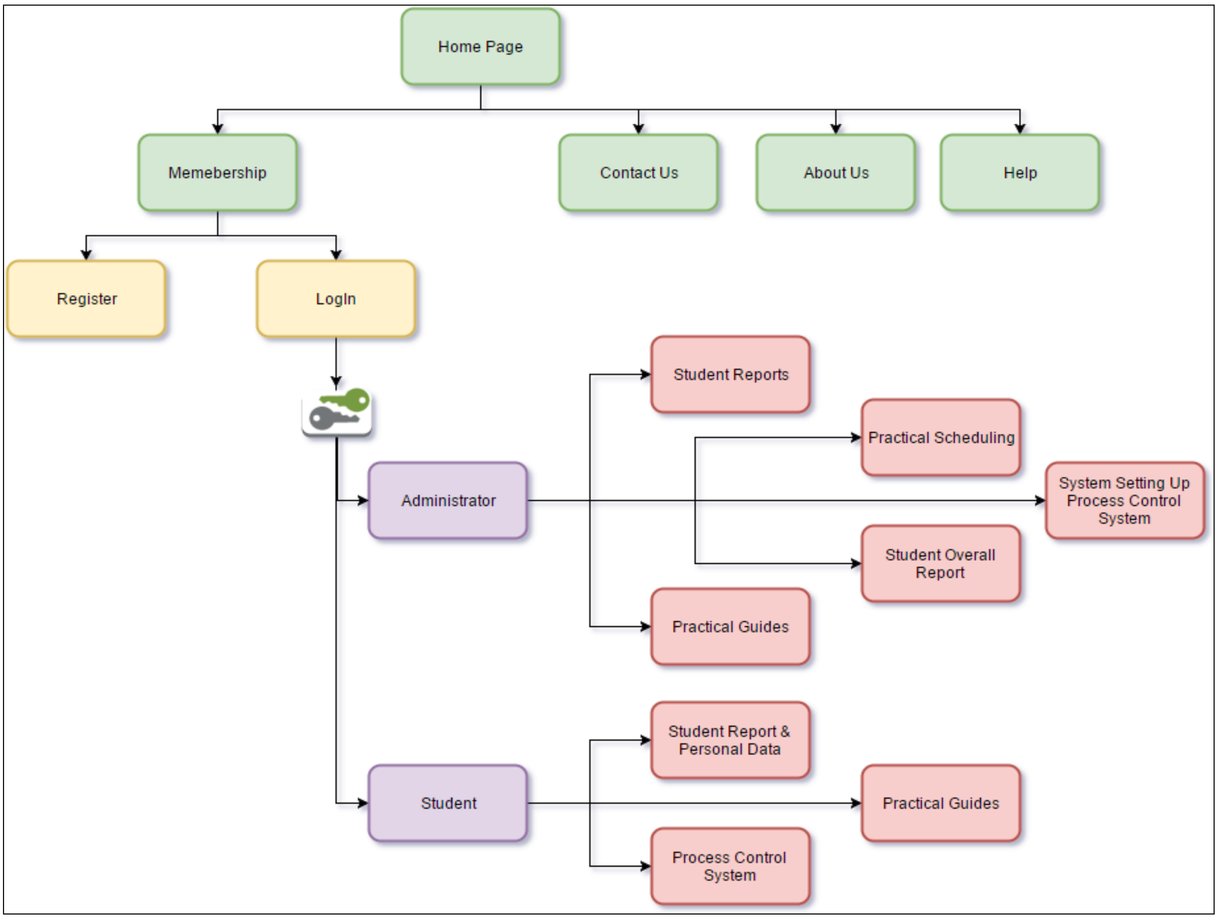
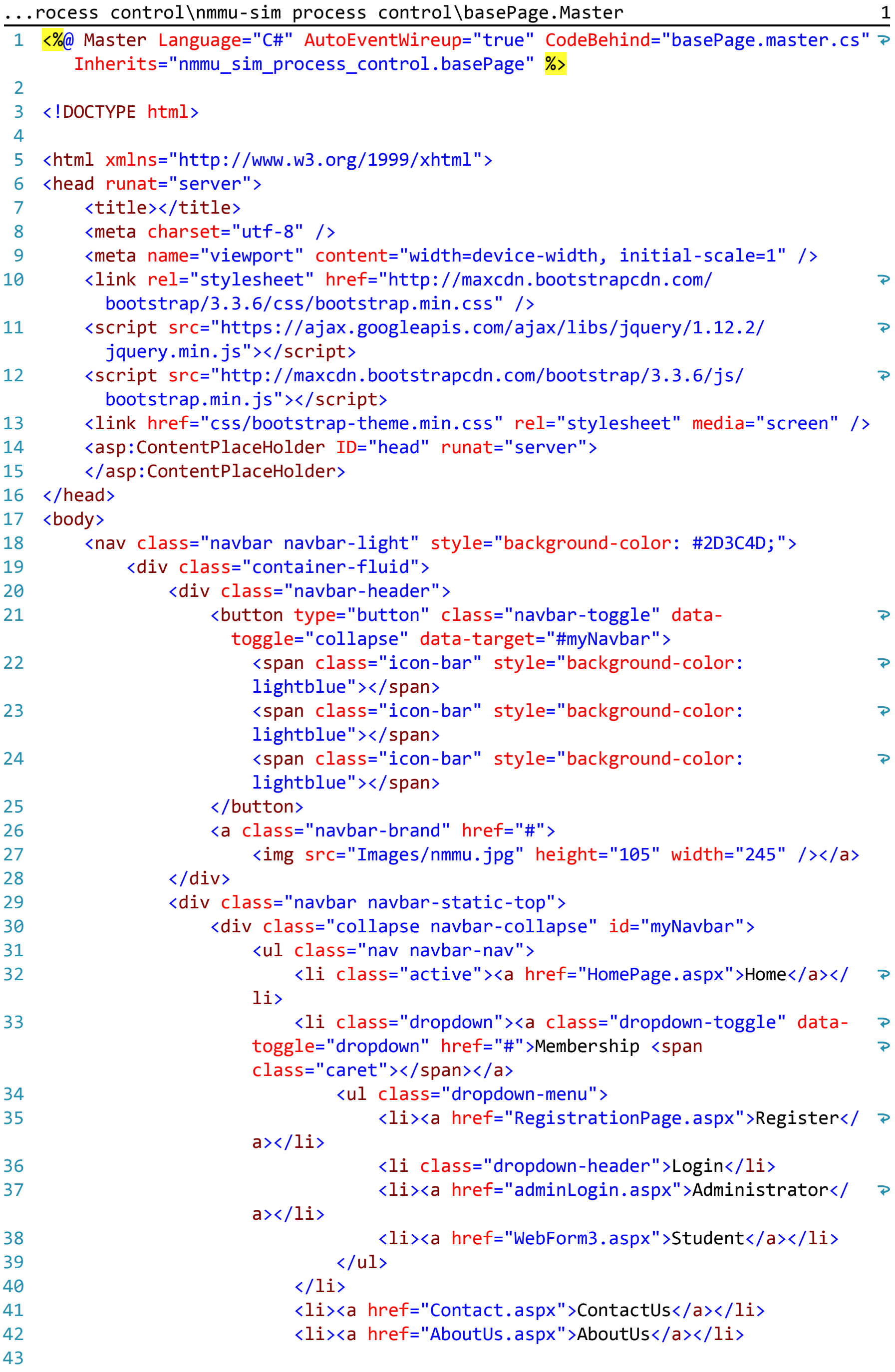


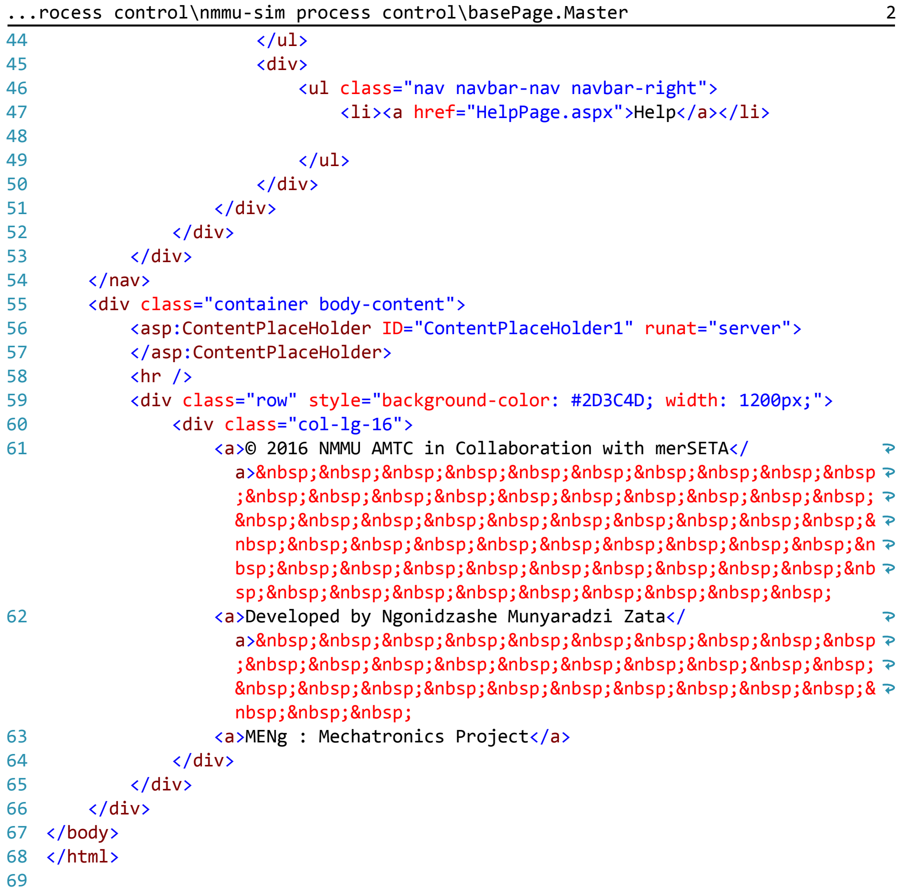
Figure 7.14: Remote Website Structure

7.3.1 Master Page : HTML Mark-Up



...rocess control\nmmu-sim process control\basePage.Master 1

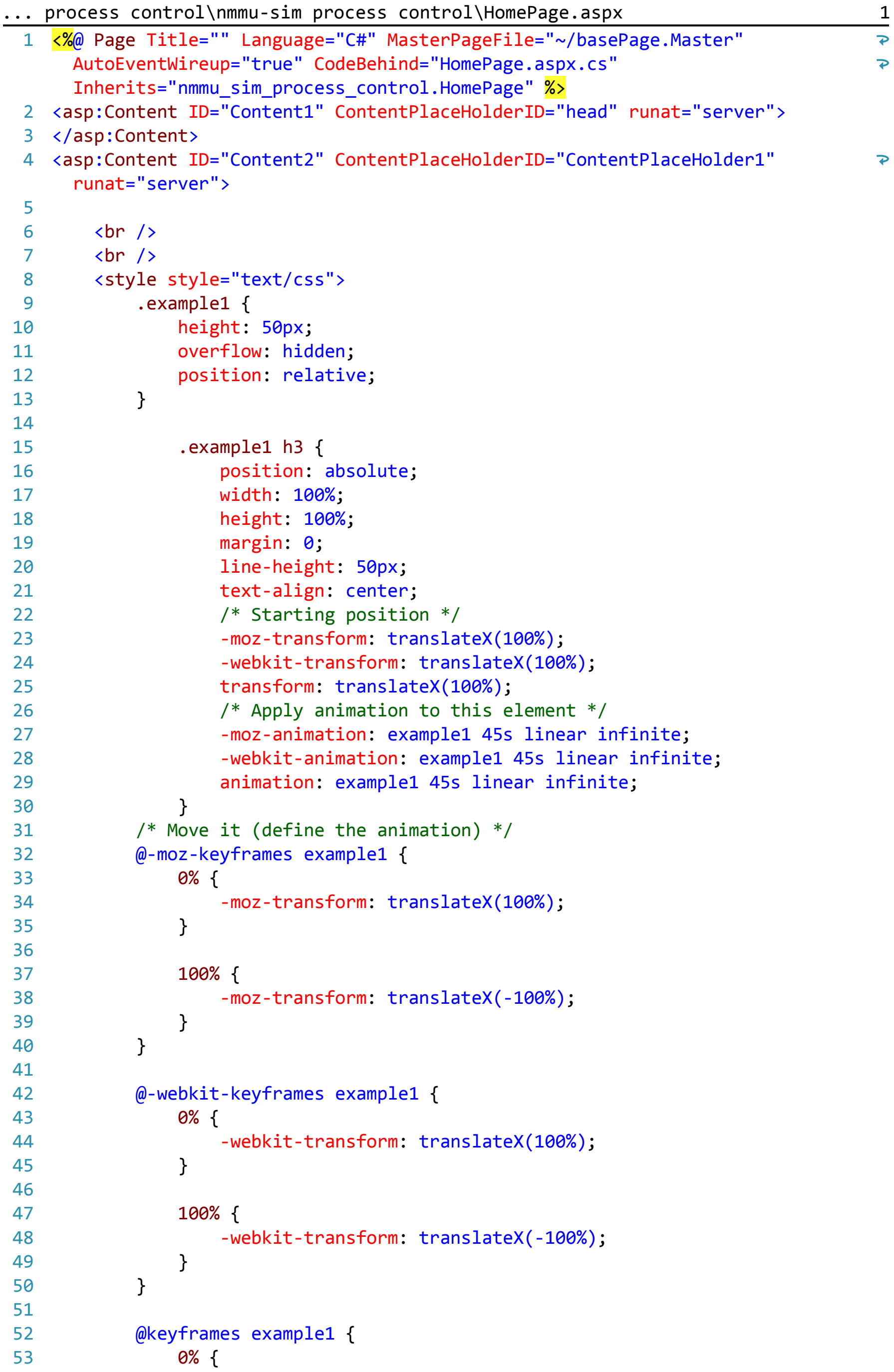
```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="basePage.master.cs" Inherits="nmmu_sim_process_control.basePage" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.min.js"></script>
    <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
    <link href="css/bootstrap-theme.min.css" rel="stylesheet" media="screen" />
    <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <nav class="navbar navbar-light" style="background-color: #2D3C4D;">
        <div class="container-fluid">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar">
                    <span class="icon-bar" style="background-color: lightblue"></span>
                    <span class="icon-bar" style="background-color: lightblue"></span>
                    <span class="icon-bar" style="background-color: lightblue"></span>
                </button>
                <a class="navbar-brand" href="#">
                    <img src="Images/nmmu.jpg" height="105" width="245" /></a>
            </div>
            <div class="navbar navbar-static-top">
                <div class="collapse navbar-collapse" id="myNavbar">
                    <ul class="nav navbar-nav">
                        <li class="active"><a href="HomePage.aspx">Home</a></li>
                        <li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown" href="#">Membership <span class="caret"></span></a>
                            <ul class="dropdown-menu">
                                <li><a href="RegistrationPage.aspx">Register</a></li>
                                <li class="dropdown-header">Login</li>
                                <li><a href="adminLogin.aspx">Administrator</a></li>
                                <li><a href="WebForm3.aspx">Student</a></li>
                            </ul>
                        </li>
                        <li><a href="Contact.aspx">ContactUs</a></li>
                        <li><a href="AboutUs.aspx">AboutUs</a></li>

```



...rocess control\nmmu-sim process control\basePage.Master 2

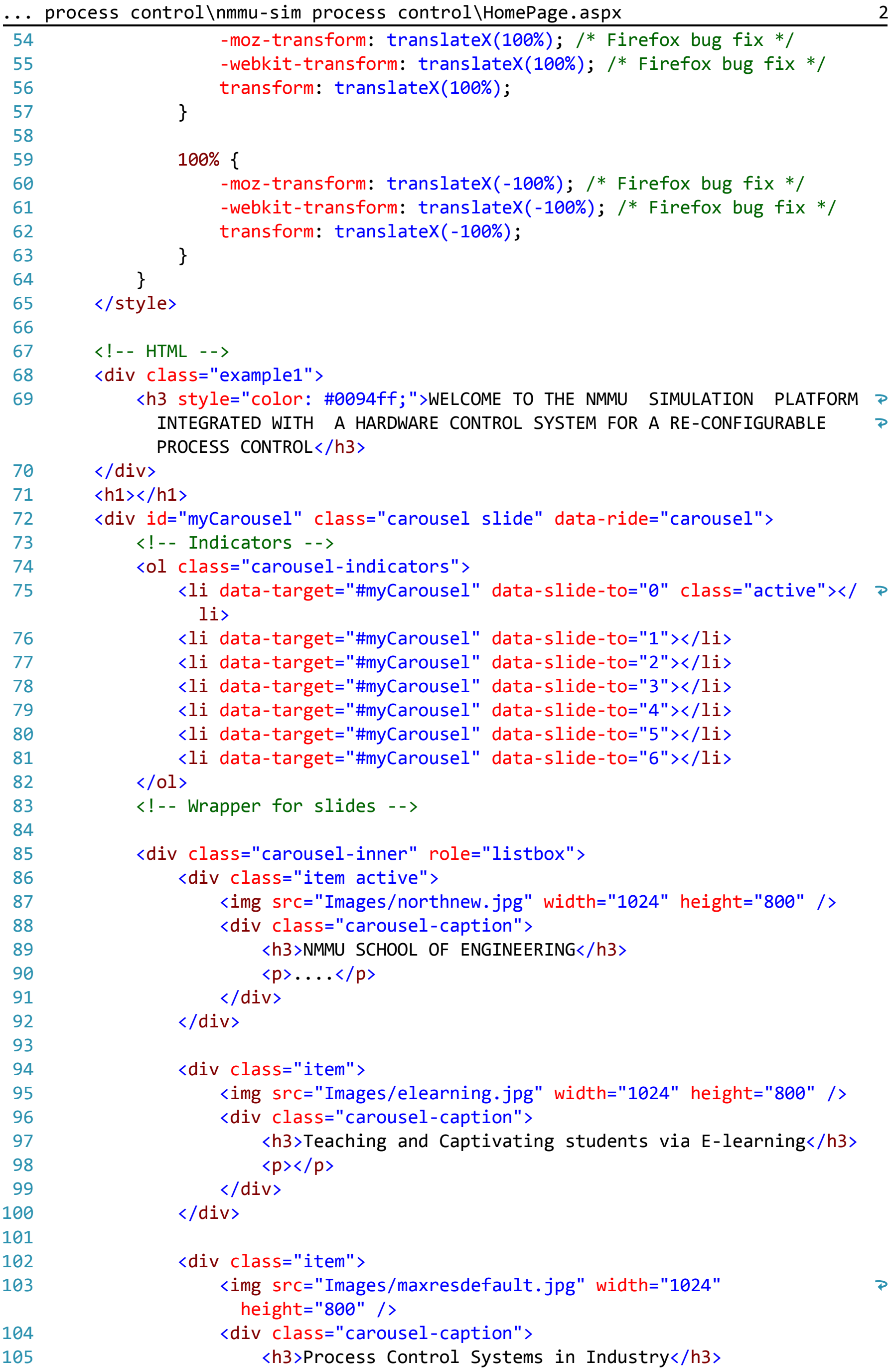
```
                    </ul>
                    <div>
                        <ul class="nav navbar-nav navbar-right">
                            <li><a href="HelpPage.aspx">Help</a></li>

                        </ul>
                    </div>
                </div>
            </div>
        </div>
    </nav>
    <div class="container body-content">
        <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
        </asp:ContentPlaceHolder>
        <hr />
        <div class="row" style="background-color: #2D3C4D; width: 1200px;">
            <div class="col-lg-16">
                <a>© 2016 NMMU AMTC in Collaboration with merSETA</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
                <a>Developed by Ngonidzashe Munyaradzi Zata</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
                <a>MENg : Mechatronics Project</a>
            </div>
        </div>
    </div>
</body>
</html>

```

7.3.2 Home Page: HTML Mark-Up



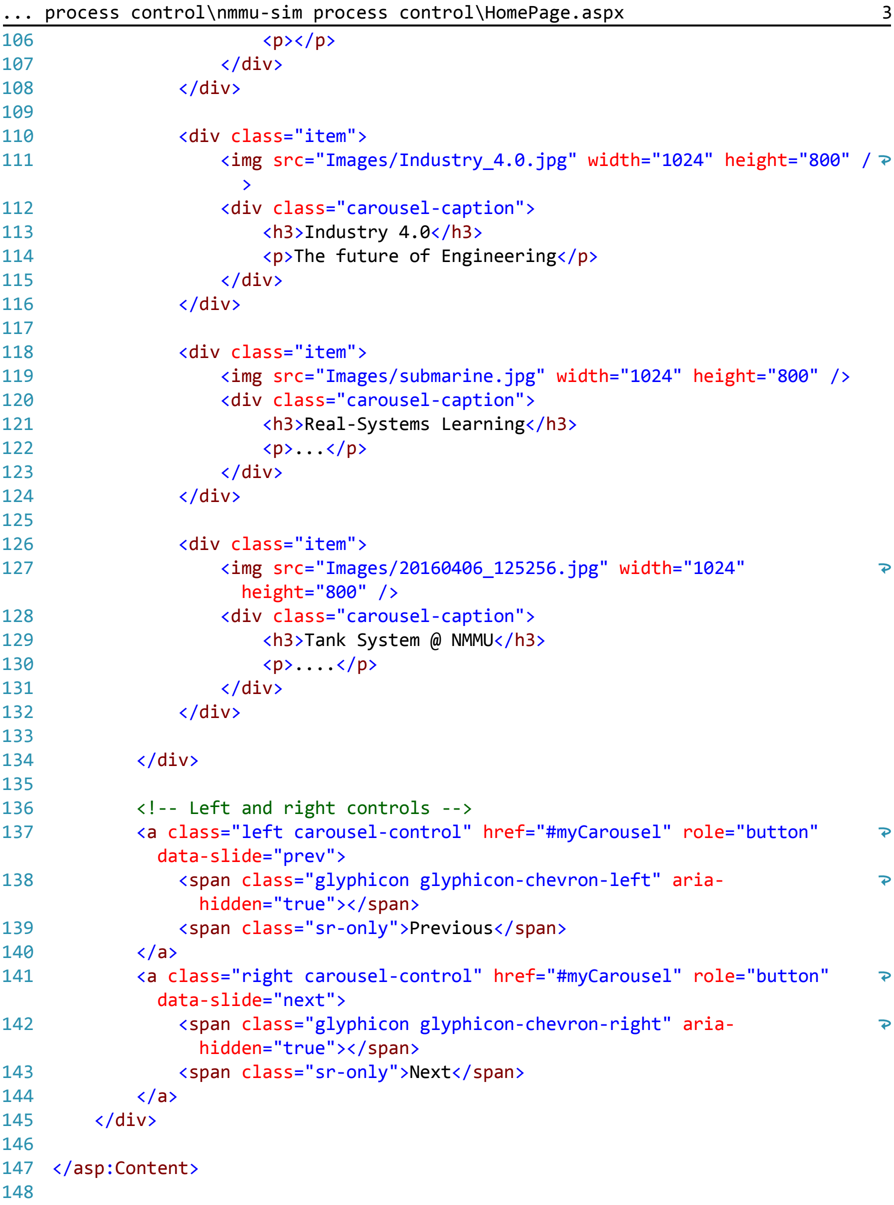
... process control\nmmu-sim process control\HomePage.aspx 1

```
<%@ Page Title="" Language="C#" MasterPageFile="~/basePage.Master" AutoEventWireup="true" CodeBehind="HomePage.aspx.cs" Inherits="nmmu_sim_process_control.HomePage" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

    <br />
    <br />
    <style style="text/css">
        .example1 {
            height: 50px;
            overflow: hidden;
            position: relative;
        }

            .example1 h3 {
                position: absolute;
                width: 100%;
                height: 100%;
                margin: 0;
                line-height: 50px;
                text-align: center;
                /* Starting position */
                -moz-transform: translateX(100%);
                -webkit-transform: translateX(100%);
                transform: translateX(100%);
                /* Apply animation to this element */
                -moz-animation: example1 45s linear infinite;
                -webkit-animation: example1 45s linear infinite;
                animation: example1 45s linear infinite;
            }
        /* Move it (define the animation) */
        @-moz-keyframes example1 {
            0% {
                -moz-transform: translateX(100%);
            }

            100% {
                -moz-transform: translateX(-100%);
            }
        }

        @-webkit-keyframes example1 {
            0% {
                -webkit-transform: translateX(100%);
            }

            100% {
                -webkit-transform: translateX(-100%);
            }
        }

        @keyframes example1 {
            0% {
```



... process control\nmmu-sim process control\HomePage.aspx 2

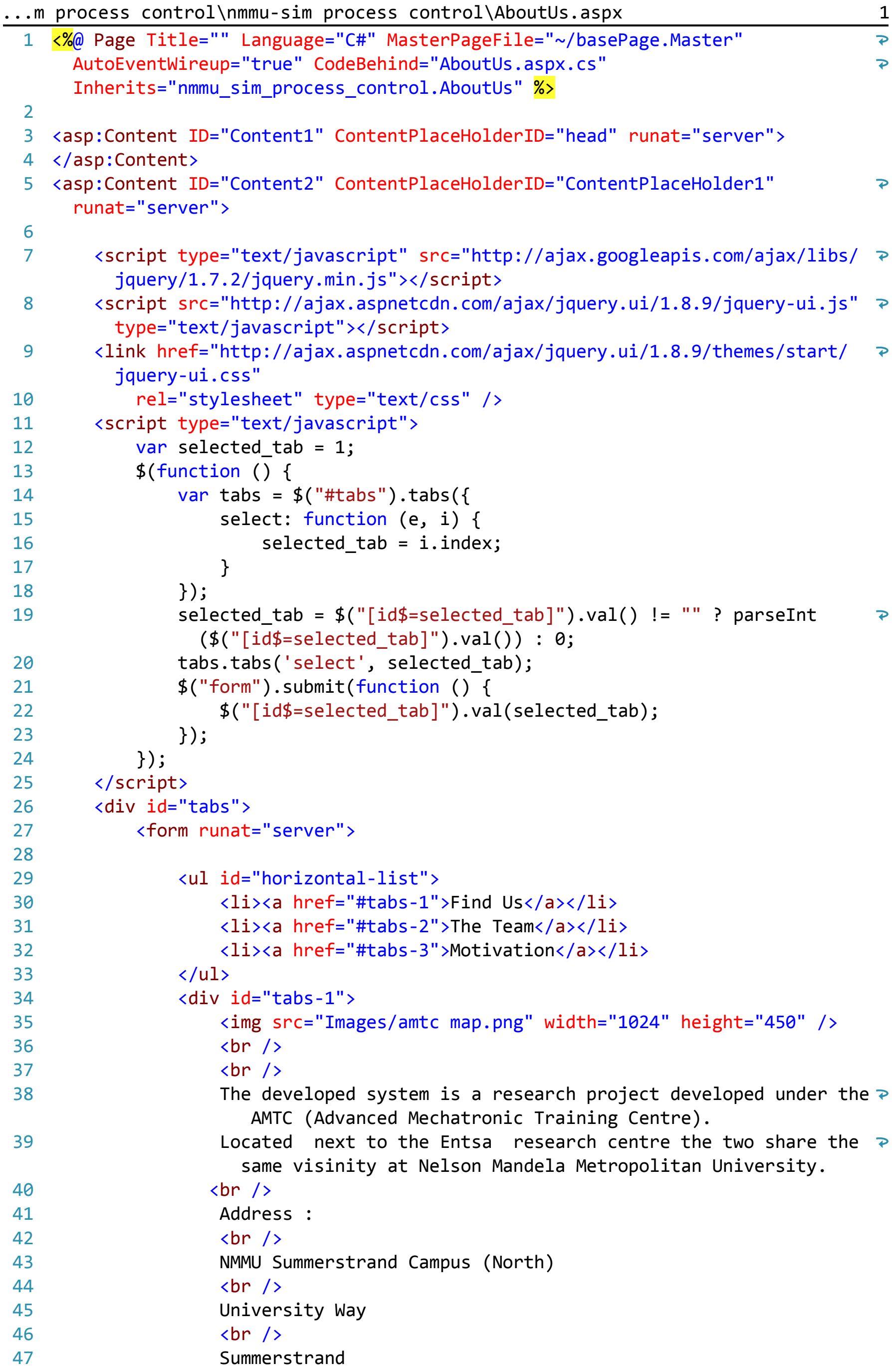
```
                -moz-transform: translateX(100%); /* Firefox bug fix */
                -webkit-transform: translateX(100%); /* Firefox bug fix */
                transform: translateX(100%);
            }

            100% {
                -moz-transform: translateX(-100%); /* Firefox bug fix */
                -webkit-transform: translateX(-100%); /* Firefox bug fix */
                transform: translateX(-100%);
            }
        }
    </style>

    <!-- HTML -->
    <div class="example1">
        <h3 style="color: #0094ff;">WELCOME TO THE NMMU  SIMULATION  PLATFORM INTEGRATED WITH  A HARDWARE CONTROL SYSTEM FOR A RE-CONFIGURABLE PROCESS CONTROL</h3>
    </div>
    <h1></h1>
    <div id="myCarousel" class="carousel slide" data-ride="carousel">
        <!-- Indicators -->
        <ol class="carousel-indicators">
            <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
            <li data-target="#myCarousel" data-slide-to="1"></li>
            <li data-target="#myCarousel" data-slide-to="2"></li>
            <li data-target="#myCarousel" data-slide-to="3"></li>
            <li data-target="#myCarousel" data-slide-to="4"></li>
            <li data-target="#myCarousel" data-slide-to="5"></li>
            <li data-target="#myCarousel" data-slide-to="6"></li>
        </ol>
        <!-- Wrapper for slides -->

        <div class="carousel-inner" role="listbox">
            <div class="item active">
                <img src="Images/northnew.jpg" width="1024" height="800" />
                <div class="carousel-caption">
                    <h3>NMMU SCHOOL OF ENGINEERING</h3>
                    <p>....</p>
                </div>
            </div>

            <div class="item">
                <img src="Images/elearning.jpg" width="1024" height="800" />
                <div class="carousel-caption">
                    <h3>Teaching and Captivating students via E-learning</h3>
                    <p></p>
                </div>
            </div>

            <div class="item">
                <img src="Images/maxresdefault.jpg" width="1024" height="800" />
                <div class="carousel-caption">
                    <h3>Process Control Systems in Industry</h3>
```



... process control\nmmu-sim process control\HomePage.aspx 3

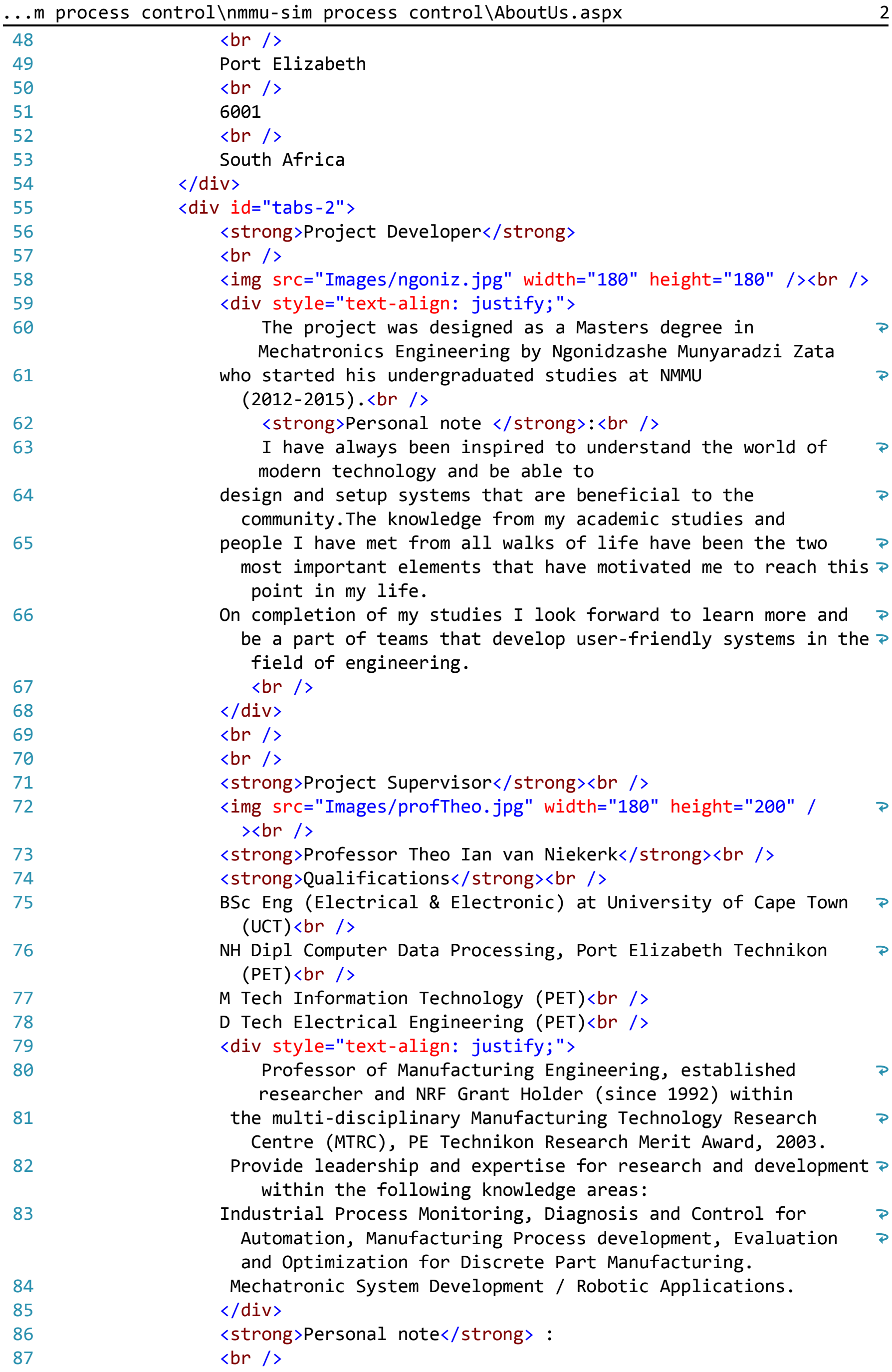
```
                    <p></p>
                </div>
            </div>

            <div class="item">
                <img src="Images/Industry_4.0.jpg" width="1024" height="800" />
                <div class="carousel-caption">
                    <h3>Industry 4.0</h3>
                    <p>The future of Engineering</p>
                </div>
            </div>

            <div class="item">
                <img src="Images/submarine.jpg" width="1024" height="800" />
                <div class="carousel-caption">
                    <h3>Real-Systems Learning</h3>
                    <p>...</p>
                </div>
            </div>

            <div class="item">
                <img src="Images/20160406_125256.jpg" width="1024" height="800" />
                <div class="carousel-caption">
                    <h3>Tank System @ NMMU</h3>
                    <p>....</p>
                </div>
            </div>

        </div>

        <!-- Left and right controls -->
        <a class="left carousel-control" href="#myCarousel" role="button" data-slide="prev">
            <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
            <span class="sr-only">Previous</span>
        </a>
        <a class="right carousel-control" href="#myCarousel" role="button" data-slide="next">
            <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
            <span class="sr-only">Next</span>
        </a>
    </div>

</asp:Content>
```

7.3.3 About Us Page : HTML Mark-Up



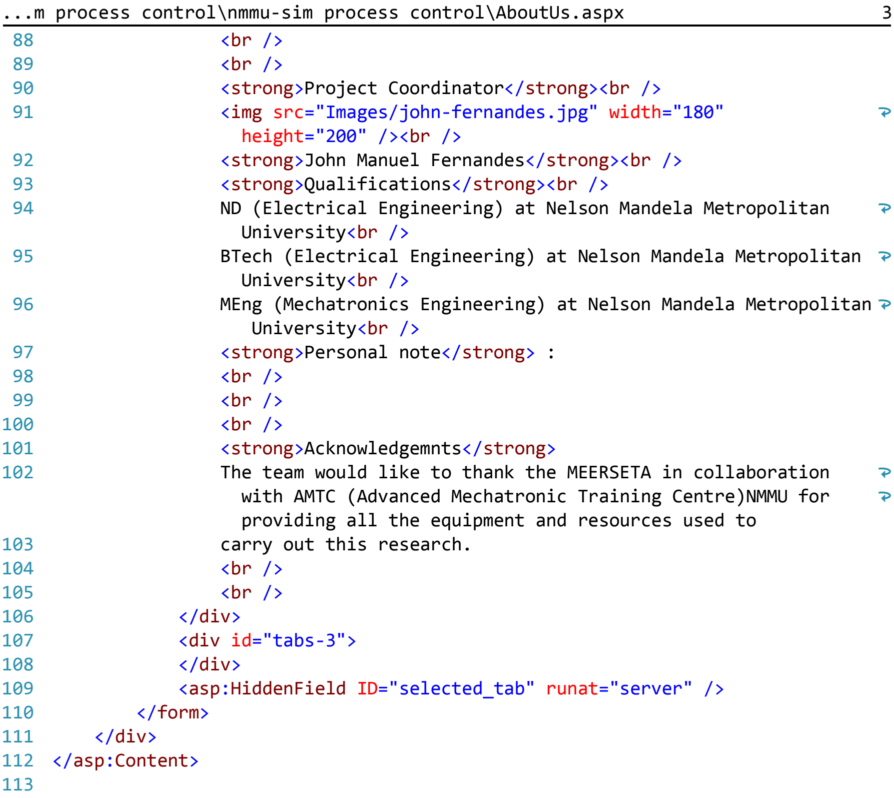
...m process control\nmmu-sim process control\AboutUs.aspx 1

```
<%@ Page Title="" Language="C#" MasterPageFile="~/basePage.Master" AutoEventWireup="true" CodeBehind="AboutUs.aspx.cs" Inherits="nmmu_sim_process_control.AboutUs" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

    <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>
    <script src="http://ajax.aspnetcdn.com/ajax/jquery.ui/1.8.9/jquery-ui.js" type="text/javascript"></script>
    <link href="http://ajax.aspnetcdn.com/ajax/jquery.ui/1.8.9/themes/start/jquery-ui.css"
        rel="stylesheet" type="text/css" />
    <script type="text/javascript">
        var selected_tab = 1;
        $(function () {
            var tabs = $("#tabs").tabs({
                select: function (e, i) {
                    selected_tab = i.index;
                }
            });
            selected_tab = $("[id$=selected_tab]").val() != "" ? parseInt($("[id$=selected_tab]").val()) : 0;
            tabs.tabs('select', selected_tab);
            $("form").submit(function () {
                $("[id$=selected_tab]").val(selected_tab);
            });
        });
    </script>
    <div id="tabs">
        <form runat="server">

            <ul id="horizontal-list">
                <li><a href="#tabs-1">Find Us</a></li>
                <li><a href="#tabs-2">The Team</a></li>
                <li><a href="#tabs-3">Motivation</a></li>
            </ul>
            <div id="tabs-1">
                <img src="Images/amtc map.png" width="1024" height="450" />
                <br />
                <br />
                The developed system is a research project developed under the AMTC (Advanced Mechatronic Training Centre).
                Located  next to the Entsa  research centre the two share the same visinity at Nelson Mandela Metropolitan University.
               <br />
                Address :
                <br />
                NMMU Summerstrand Campus (North)
                <br />
                University Way
                <br />
                Summerstrand
```



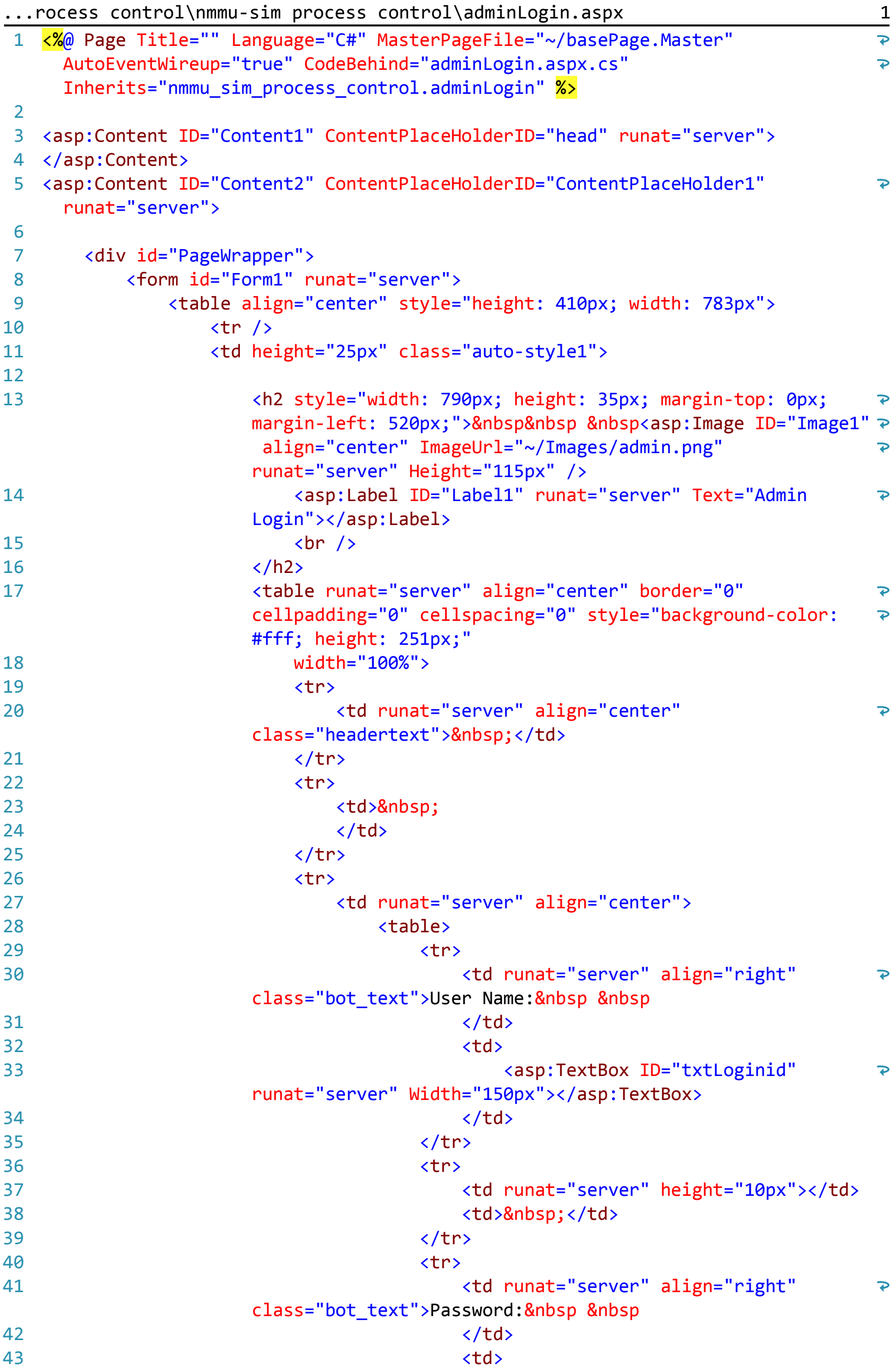
...m process control\nmmu-sim process control\AboutUs.aspx 2

```
                <br />
                Port Elizabeth
                <br />
                6001
                <br />
                South Africa
            </div>
            <div id="tabs-2">
                <strong>Project Developer</strong>
                <br />
                <img src="Images/ngoniz.jpg" width="180" height="180" /><br />
                <div style="text-align: justify;">
                    The project was designed as a Masters degree in Mechatronics Engineering by Ngonidzashe Munyaradzi Zata
                who started his undergraduated studies at NMMU (2012-2015).<br />
                    <strong>Personal note </strong>:<br />
                    I have always been inspired to understand the world of modern technology and be able to
                design and setup systems that are beneficial to the community.The knowledge from my academic studies and
                people I have met from all walks of life have been the two most important elements that have motivated me to reach this point in my life.
                On completion of my studies I look forward to learn more and be a part of teams that develop user-friendly systems in the field of engineering.
                   <br />
                </div>
                <br />
                <br />
                <strong>Project Supervisor</strong><br />
                <img src="Images/profTheo.jpg" width="180" height="200" /><br />
                <strong>Professor Theo Ian van Niekerk</strong><br />
                <strong>Qualifications</strong><br />
                BSc Eng (Electrical & Electronic) at University of Cape Town (UCT)<br />
                NH Dipl Computer Data Processing, Port Elizabeth Technikon (PET)<br />
                M Tech Information Technology (PET)<br />
                D Tech Electrical Engineering (PET)<br />
                <div style="text-align: justify;">
                    Professor of Manufacturing Engineering, established researcher and NRF Grant Holder (since 1992) within
                 the multi-disciplinary Manufacturing Technology Research Centre (MTRC), PE Technikon Research Merit Award, 2003.
                 Provide leadership and expertise for research and development within the following knowledge areas:
                Industrial Process Monitoring, Diagnosis and Control for Automation, Manufacturing Process development, Evaluation and Optimization for Discrete Part Manufacturing.
                 Mechatronic System Development / Robotic Applications.
                </div>
                <strong>Personal note</strong> :
                <br />
```



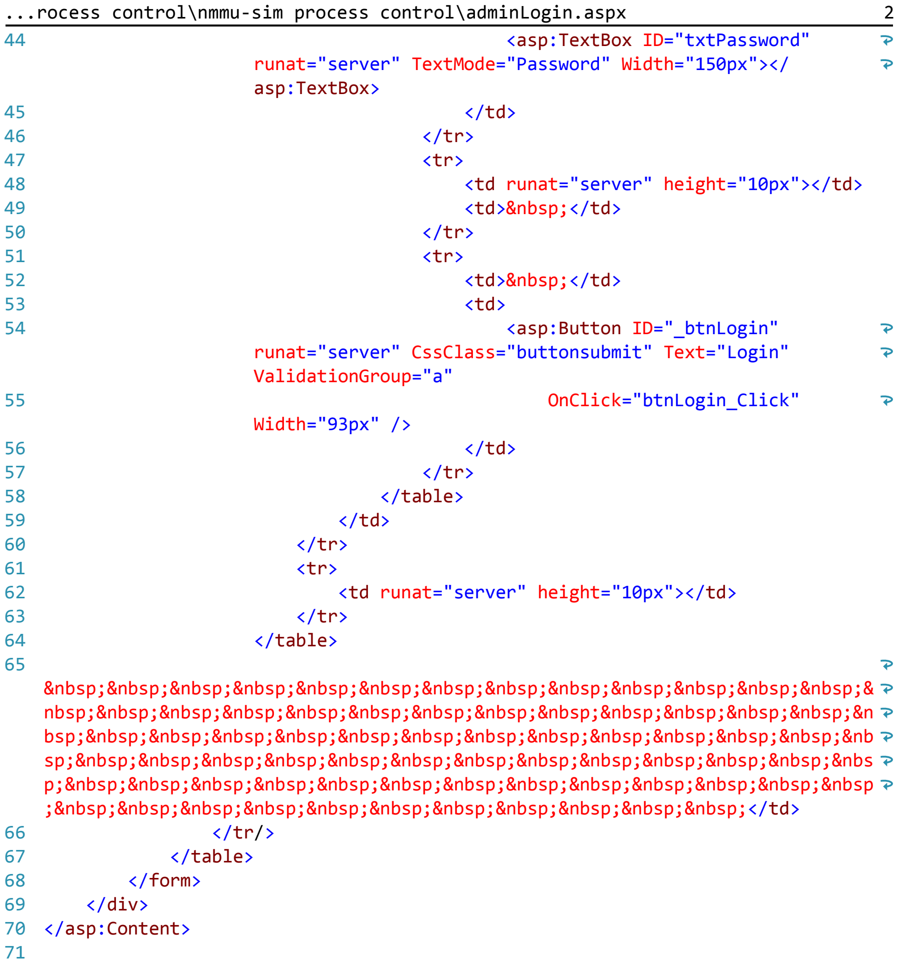
```
...m process control\nmmu-sim process control\AboutUs.aspx                    3
                <br />
                <br />
                <strong>Project Coordinator</strong><br />
                <img src="Images/john-fernandes.jpg" width="180"
                  height="200" /><br />
                <strong>John Manuel Fernandes</strong><br />
                <strong>Qualifications</strong><br />
                ND (Electrical Engineering) at Nelson Mandela Metropolitan
                  University<br />
                BTech (Electrical Engineering) at Nelson Mandela Metropolitan
                  University<br />
                MEng (Mechatronics Engineering) at Nelson Mandela Metropolitan
                   University<br />
                <strong>Personal note</strong> :
                <br />
                <br />
                <br />
                <strong>Acknowledgemnts</strong>
                The team would like to thank the MEERSETA in collaboration
                  with AMTC (Advanced Mechatronic Training Centre)NMMU for
                  providing all the equipment and resources used to
                carry out this research.
                <br />
                <br />
            </div>
            <div id="tabs-3">
            </div>
            <asp:HiddenField ID="selected_tab" runat="server" />
        </form>
    </div>
</asp:Content>

```

7.3.4 Administrator Login : HTML Mark-Up



...rocess control\nmmu-sim process control\adminLogin.aspx 1

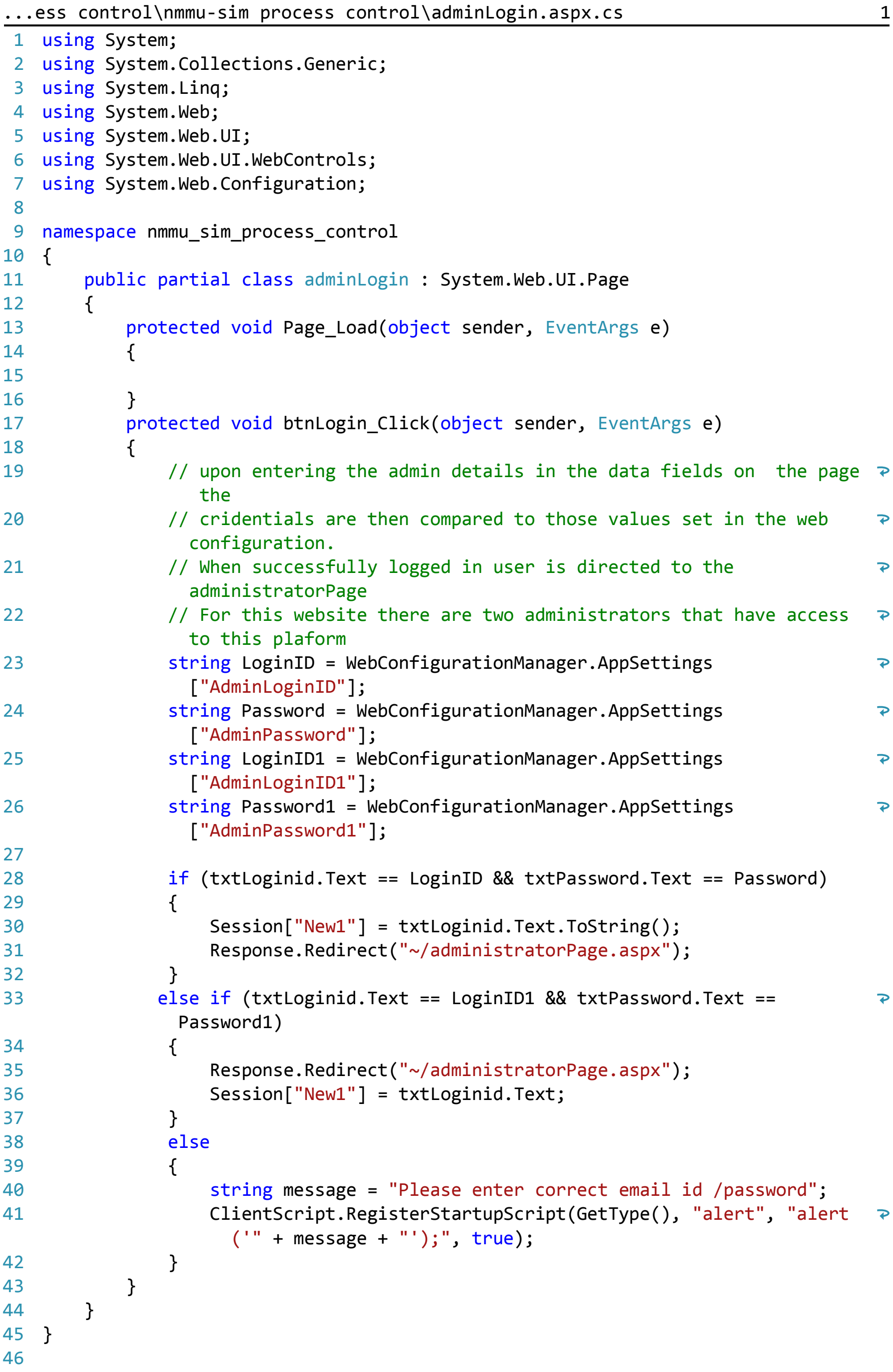
```
<%@ Page Title="" Language="C#" MasterPageFile="~/basePage.Master" AutoEventWireup="true" CodeBehind="adminLogin.aspx.cs" Inherits="nmmu_sim_process_control.adminLogin" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

    <div id="PageWrapper">
        <form id="Form1" runat="server">
            <table align="center" style="height: 410px; width: 783px">
                <tr />
                <td height="25px" class="auto-style1">

                    <h2 style="width: 790px; height: 35px; margin-top: 0px; margin-left: 520px;">&nbsp&nbsp &nbsp<asp:Image ID="Image1" align="center" ImageUrl="~/Images/admin.png" runat="server" Height="115px" />
                        <asp:Label ID="Label1" runat="server" Text="Admin Login"></asp:Label>
                        <br />
                    </h2>
                    <table runat="server" align="center" border="0" cellpadding="0" cellspacing="0" style="background-color: #fff; height: 251px;"
                        width="100%">
                        <tr>
                            <td runat="server" align="center" class="headertext">&nbsp;</td>
                        </tr>
                        <tr>
                            <td>&nbsp;
                            </td>
                        </tr>
                        <tr>
                            <td runat="server" align="center">
                                <table>
                                    <tr>
                                        <td runat="server" align="right" class="bot_text">User Name:&nbsp &nbsp
                                        </td>
                                        <td>
                                            <asp:TextBox ID="txtLoginid" runat="server" Width="150px"></asp:TextBox>
                                        </td>
                                    </tr>
                                    <tr>
                                        <td runat="server" height="10px"></td>
                                        <td>&nbsp;</td>
                                    </tr>
                                    <tr>
                                        <td runat="server" align="right" class="bot_text">Password:&nbsp &nbsp
                                        </td>
                                        <td>
```



...rocess control\nmmu-sim process control\adminLogin.aspx 2

```
                                            <asp:TextBox ID="txtPassword" runat="server" TextMode="Password" Width="150px"></asp:TextBox>
                                        </td>
                                    </tr>
                                    <tr>
                                        <td runat="server" height="10px"></td>
                                        <td>&nbsp;</td>
                                    </tr>
                                    <tr>
                                        <td>&nbsp;</td>
                                        <td>
                                            <asp:Button ID="_btnLogin" runat="server" CssClass="buttonsubmit" Text="Login" ValidationGroup="a"
                                                OnClick="btnLogin_Click" Width="93px" />
                                        </td>
                                    </tr>
                                </table>
                            </td>
                        </tr>
                        <tr>
                            <td runat="server" height="10px"></td>
                        </tr>
                    </table>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>
                </tr/>
            </table>
        </form>
    </div>
</asp:Content>

```

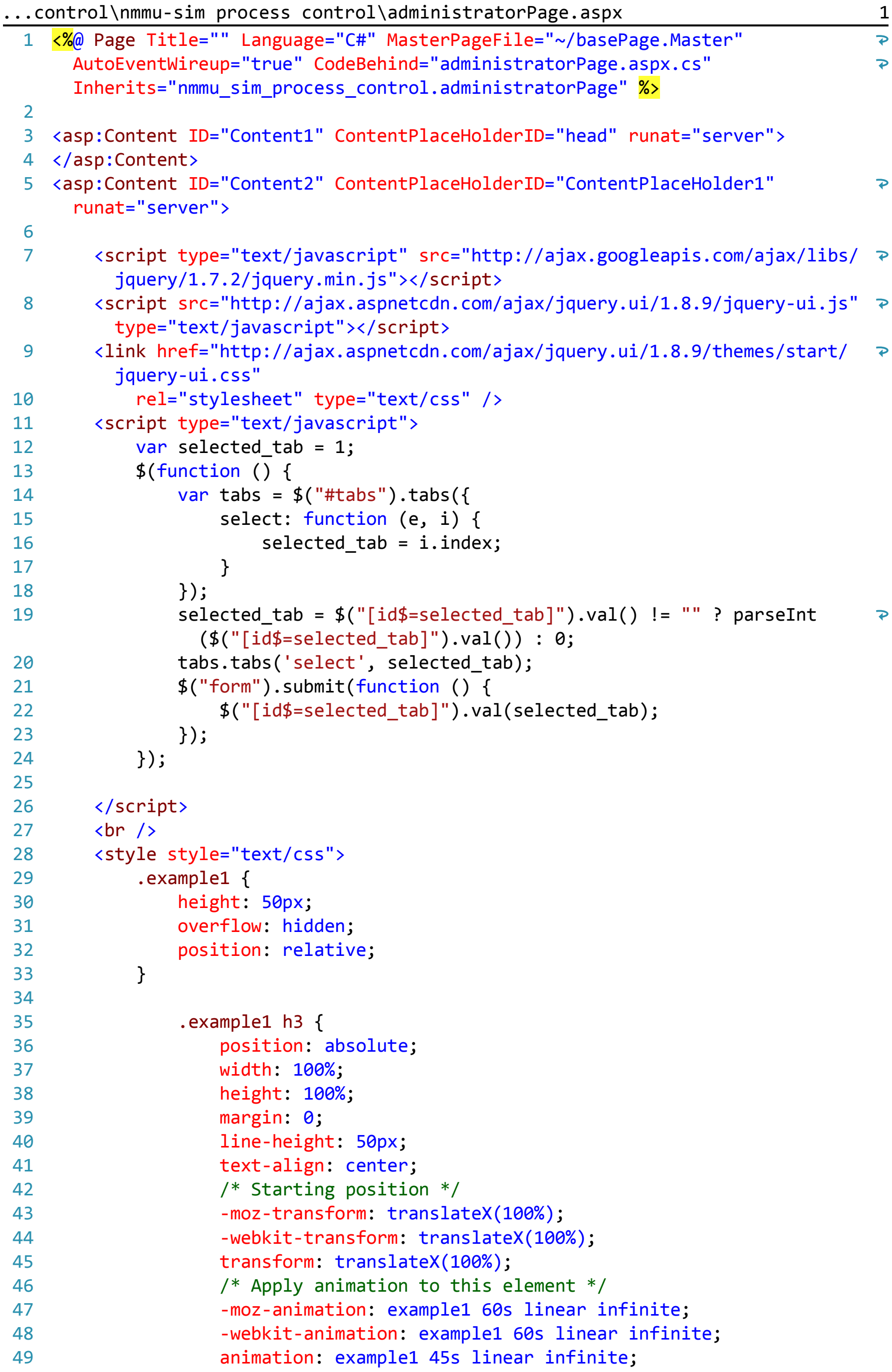
7.3.5 Administrator Login : C# Code



...ess control\nmmu-sim process control\adminLogin.aspx.cs 1

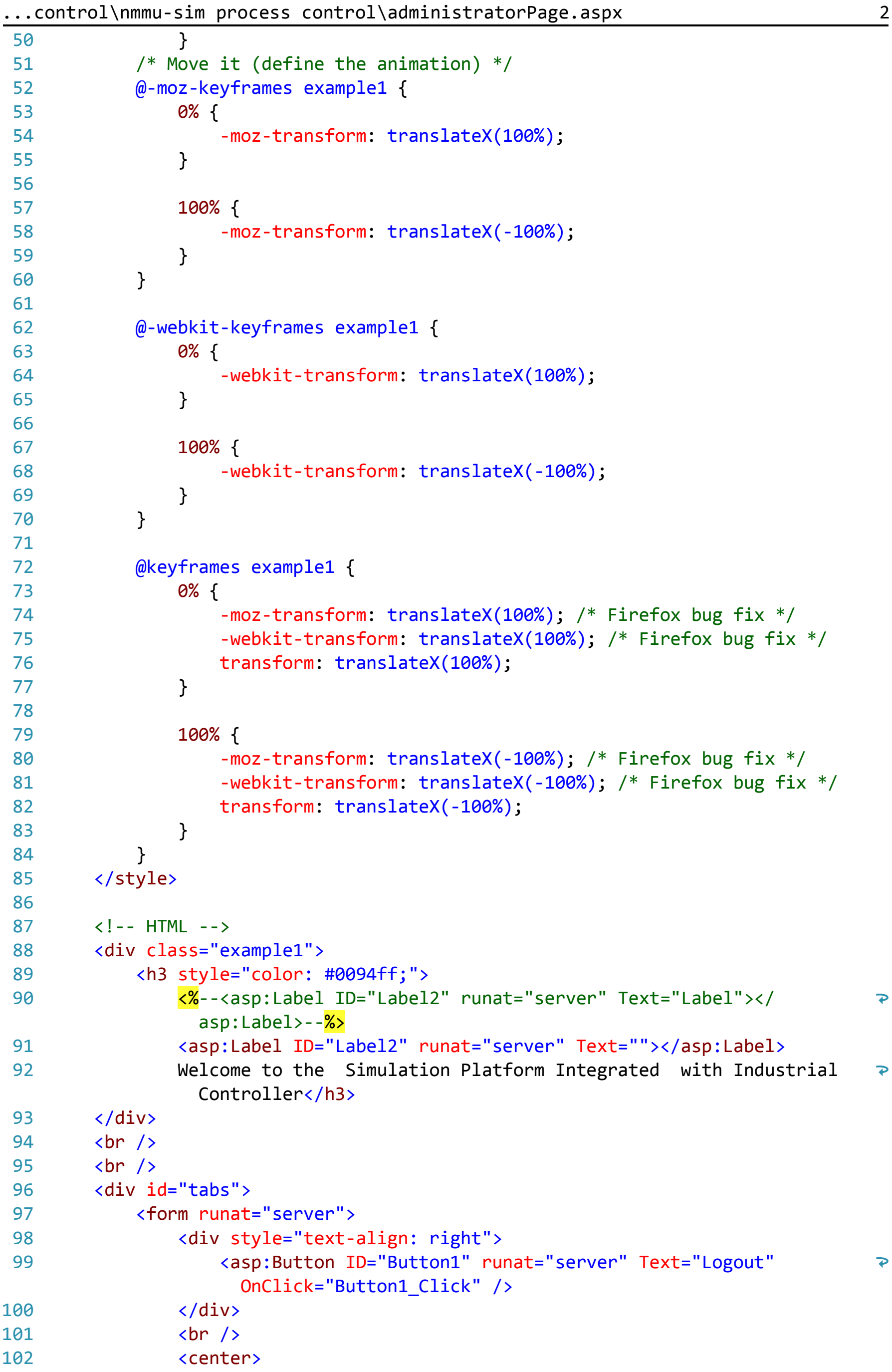
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Configuration;

namespace nmmu_sim_process_control
{
    public partial class adminLogin : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
        protected void btnLogin_Click(object sender, EventArgs e)
        {
            // upon entering the admin details in the data fields on  the page the
            // cridentials are then compared to those values set in the web configuration.
            // When successfully logged in user is directed to the administratorPage
            // For this website there are two administrators that have access to this plaform
            string LoginID = WebConfigurationManager.AppSettings["AdminLoginID"];
            string Password = WebConfigurationManager.AppSettings["AdminPassword"];
            string LoginID1 = WebConfigurationManager.AppSettings["AdminLoginID1"];
            string Password1 = WebConfigurationManager.AppSettings["AdminPassword1"];

            if (txtLoginid.Text == LoginID && txtPassword.Text == Password)
            {
                Session["New1"] = txtLoginid.Text.ToString();
                Response.Redirect("~/administratorPage.aspx");
            }
           else if (txtLoginid.Text == LoginID1 && txtPassword.Text == Password1)
            {
                Response.Redirect("~/administratorPage.aspx");
                Session["New1"] = txtLoginid.Text;
            }
            else
            {
                string message = "Please enter correct email id /password";
                ClientScript.RegisterStartupScript(GetType(), "alert", "alert('" + message + "');", true);
            }
        }
    }
}

```

7.3.6 Administrator Page : HTML Mark-Up



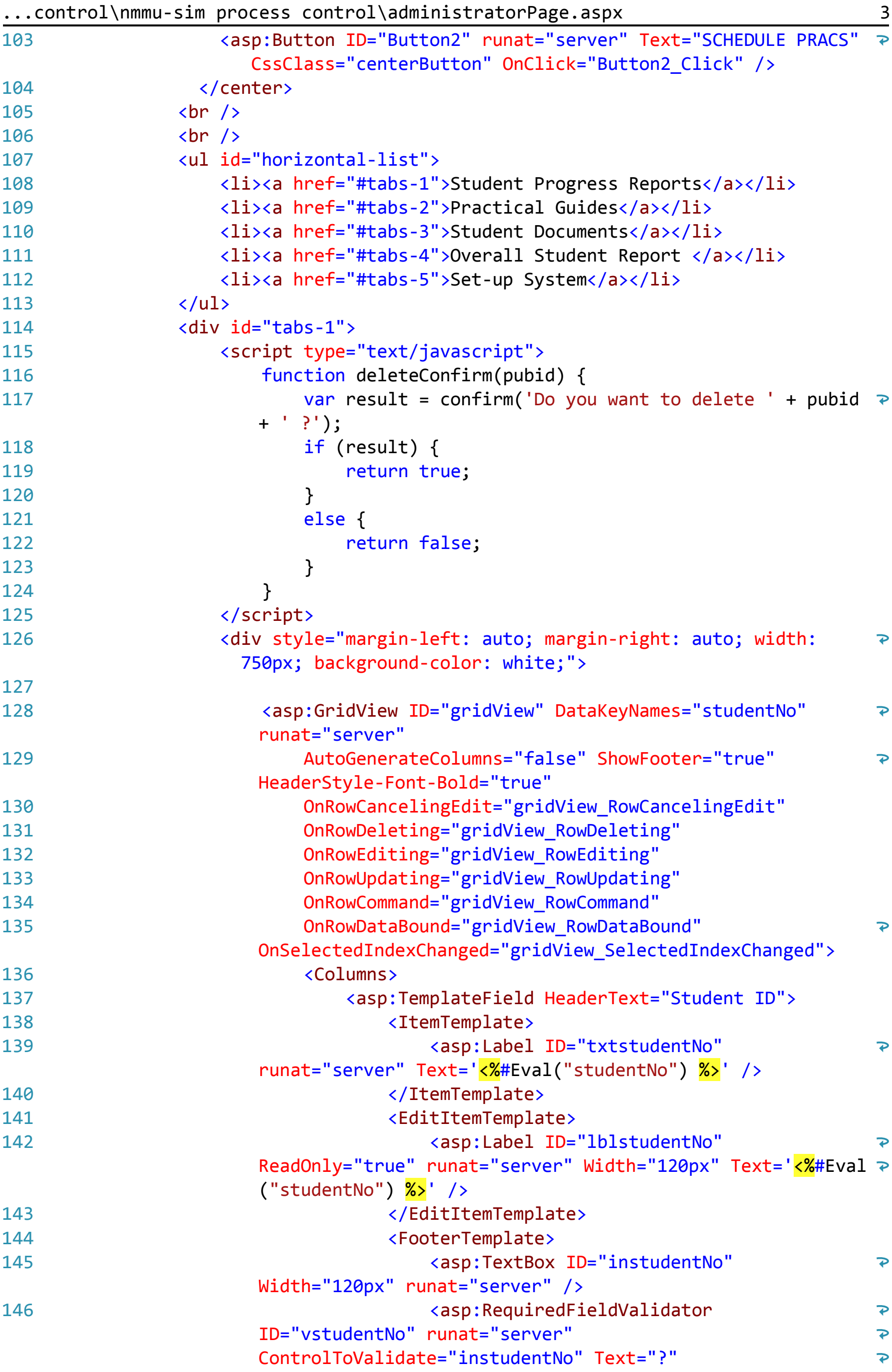
...control\nmmu-sim process control\administratorPage.aspx 1

```
<%@ Page Title="" Language="C#" MasterPageFile="~/basePage.Master" AutoEventWireup="true" CodeBehind="administratorPage.aspx.cs" Inherits="nmmu_sim_process_control.administratorPage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

    <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>
    <script src="http://ajax.aspnetcdn.com/ajax/jquery.ui/1.8.9/jquery-ui.js" type="text/javascript"></script>
    <link href="http://ajax.aspnetcdn.com/ajax/jquery.ui/1.8.9/themes/start/jquery-ui.css"
        rel="stylesheet" type="text/css" />
    <script type="text/javascript">
        var selected_tab = 1;
        $(function () {
            var tabs = $("#tabs").tabs({
                select: function (e, i) {
                    selected_tab = i.index;
                }
            });
            selected_tab = $("[id$=selected_tab]").val() != "" ? parseInt($("[id$=selected_tab]").val()) : 0;
            tabs.tabs('select', selected_tab);
            $("form").submit(function () {
                $("[id$=selected_tab]").val(selected_tab);
            });
        });

    </script>
    <br />
    <style style="text/css">
        .example1 {
            height: 50px;
            overflow: hidden;
            position: relative;
        }

            .example1 h3 {
                position: absolute;
                width: 100%;
                height: 100%;
                margin: 0;
                line-height: 50px;
                text-align: center;
                /* Starting position */
                -moz-transform: translateX(100%);
                -webkit-transform: translateX(100%);
                transform: translateX(100%);
                /* Apply animation to this element */
                -moz-animation: example1 60s linear infinite;
                -webkit-animation: example1 60s linear infinite;
                animation: example1 45s linear infinite;
```



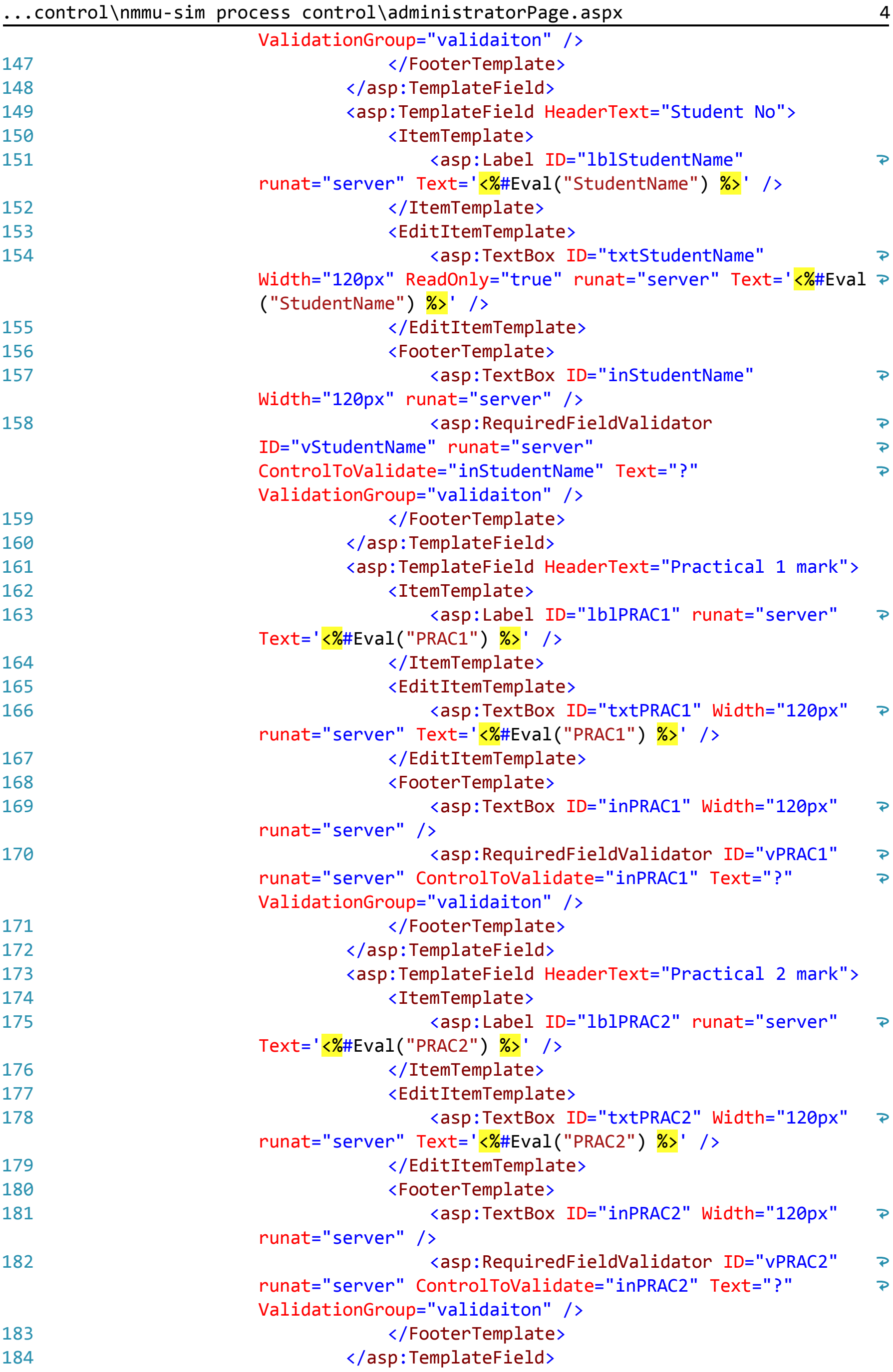
...control\nmmu-sim process control\administratorPage.aspx 2

```
            }
        /* Move it (define the animation) */
        @-moz-keyframes example1 {
            0% {
                -moz-transform: translateX(100%);
            }

            100% {
                -moz-transform: translateX(-100%);
            }
        }

        @-webkit-keyframes example1 {
            0% {
                -webkit-transform: translateX(100%);
            }

            100% {
                -webkit-transform: translateX(-100%);
            }
        }

        @keyframes example1 {
            0% {
                -moz-transform: translateX(100%); /* Firefox bug fix */
                -webkit-transform: translateX(100%); /* Firefox bug fix */
                transform: translateX(100%);
            }

            100% {
                -moz-transform: translateX(-100%); /* Firefox bug fix */
                -webkit-transform: translateX(-100%); /* Firefox bug fix */
                transform: translateX(-100%);
            }
        }
    </style>

    <!-- HTML -->
    <div class="example1">
        <h3 style="color: #0094ff;">
            <%--<asp:Label ID="Label2" runat="server" Text="Label"></asp:Label>--%>
            <asp:Label ID="Label2" runat="server" Text=""></asp:Label>
            Welcome to the  Simulation Platform Integrated  with Industrial Controller</h3>
    </div>
    <br />
    <br />
    <div id="tabs">
        <form runat="server">
            <div style="text-align: right">
                <asp:Button ID="Button1" runat="server" Text="Logout" OnClick="Button1_Click" />
            </div>
            <br />
            <center>
```



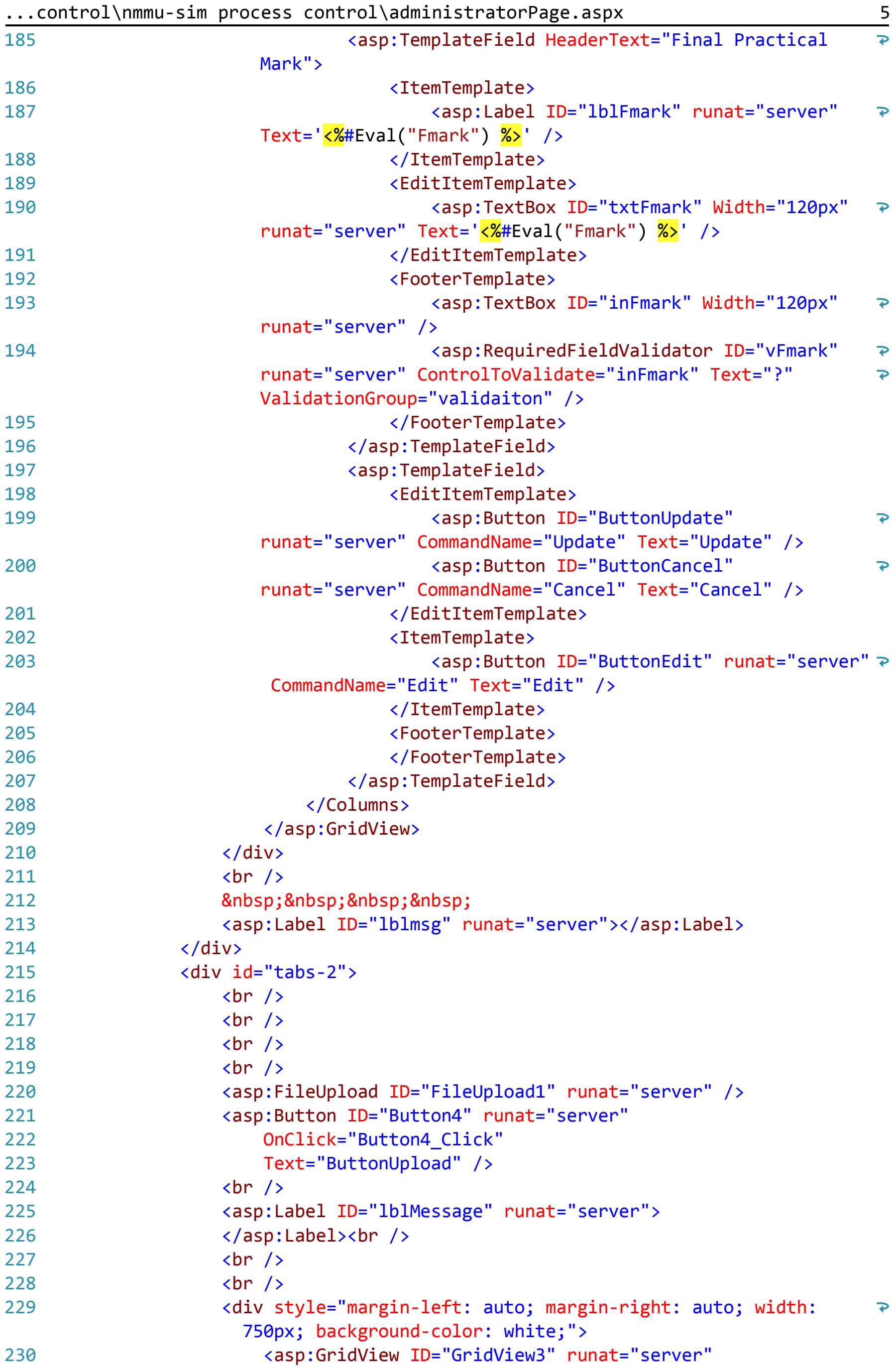
...control\nmmu-sim process control\administratorPage.aspx 3

```
                <asp:Button ID="Button2" runat="server" Text="SCHEDULE PRACS" CssClass="centerButton" OnClick="Button2_Click" />
              </center>
            <br />
            <br />
            <ul id="horizontal-list">
                <li><a href="#tabs-1">Student Progress Reports</a></li>
                <li><a href="#tabs-2">Practical Guides</a></li>
                <li><a href="#tabs-3">Student Documents</a></li>
                <li><a href="#tabs-4">Overall Student Report </a></li>
                <li><a href="#tabs-5">Set-up System</a></li>
            </ul>
            <div id="tabs-1">
                <script type="text/javascript">
                    function deleteConfirm(pubid) {
                        var result = confirm('Do you want to delete ' + pubid + ' ?');
                        if (result) {
                            return true;
                        }
                        else {
                            return false;
                        }
                    }
                </script>
                <div style="margin-left: auto; margin-right: auto; width: 750px; background-color: white;">

                    <asp:GridView ID="gridView" DataKeyNames="studentNo" runat="server"
                        AutoGenerateColumns="false" ShowFooter="true" HeaderStyle-Font-Bold="true"
                        OnRowCancelingEdit="gridView_RowCancelingEdit"
                        OnRowDeleting="gridView_RowDeleting"
                        OnRowEditing="gridView_RowEditing"
                        OnRowUpdating="gridView_RowUpdating"
                        OnRowCommand="gridView_RowCommand"
                        OnRowDataBound="gridView_RowDataBound" OnSelectedIndexChanged="gridView_SelectedIndexChanged">
                        <Columns>
                            <asp:TemplateField HeaderText="Student ID">
                                <ItemTemplate>
                                    <asp:Label ID="txtstudentNo" runat="server" Text='<%#Eval("studentNo") %>' />
                                </ItemTemplate>
                                <EditItemTemplate>
                                    <asp:Label ID="lblstudentNo" ReadOnly="true" runat="server" Width="120px" Text='<%#Eval("studentNo") %>' />
                                </EditItemTemplate>
                                <FooterTemplate>
                                    <asp:TextBox ID="instudentNo" Width="120px" runat="server" />
                                    <asp:RequiredFieldValidator ID="vstudentNo" runat="server" ControlToValidate="instudentNo" Text="?"
```



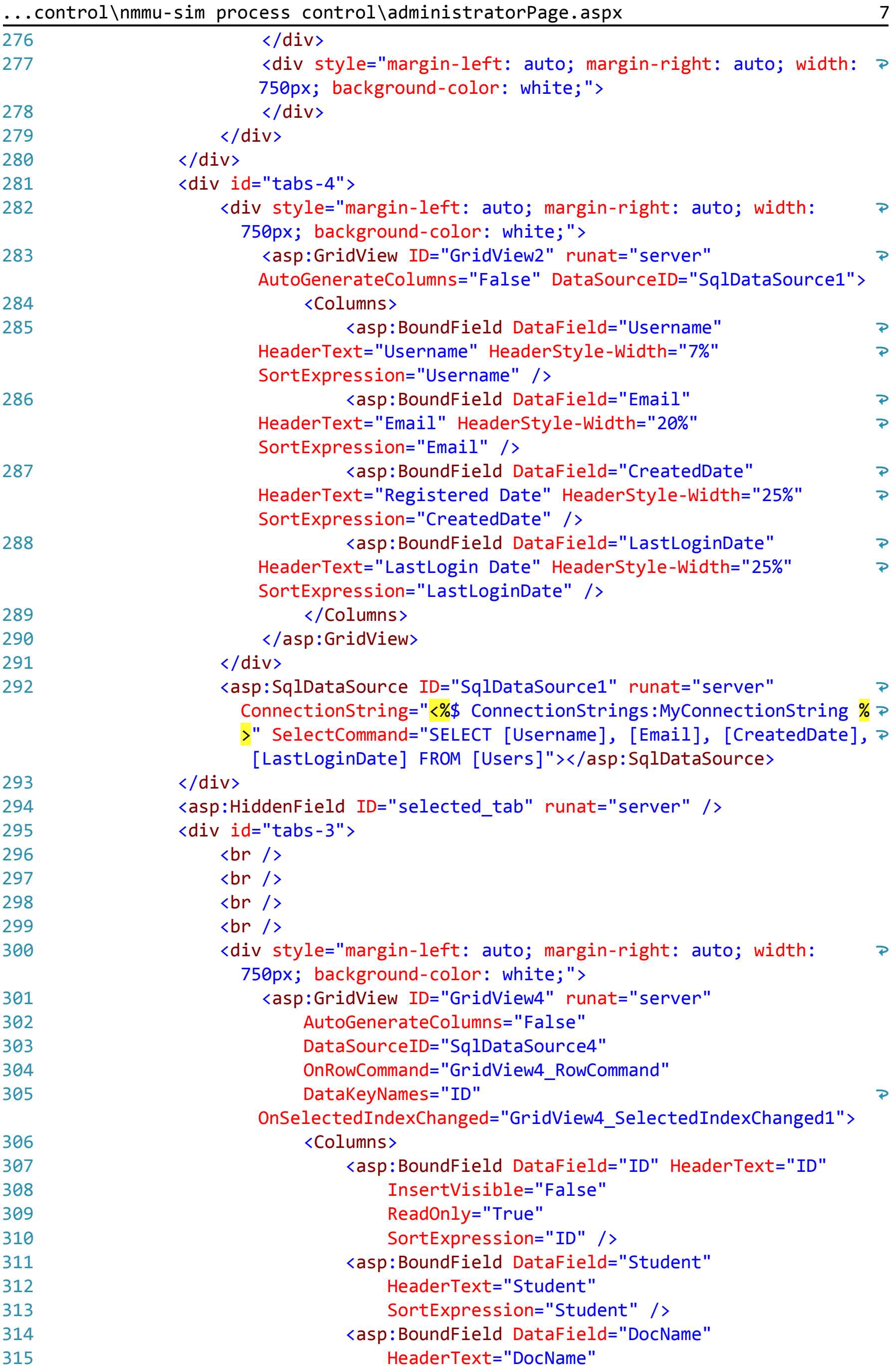
...control\nmmu-sim process control\administratorPage.aspx 4

```
ValidationGroup="validaiton" />
                    </FooterTemplate>
                </asp:TemplateField>
                <asp:TemplateField HeaderText="Student No">
                    <ItemTemplate>
                        <asp:Label ID="lblStudentName" runat="server" Text='<%#Eval("StudentName") %>' />
                    </ItemTemplate>
                    <EditItemTemplate>
                        <asp:TextBox ID="txtStudentName" Width="120px" ReadOnly="true" runat="server" Text='<%#Eval("StudentName") %>' />
                    </EditItemTemplate>
                    <FooterTemplate>
                        <asp:TextBox ID="inStudentName" Width="120px" runat="server" />
                        <asp:RequiredFieldValidator ID="vStudentName" runat="server" ControlToValidate="inStudentName" Text="?" ValidationGroup="validaiton" />
                    </FooterTemplate>
                </asp:TemplateField>
                <asp:TemplateField HeaderText="Practical 1 mark">
                    <ItemTemplate>
                        <asp:Label ID="lblPRAC1" runat="server" Text='<%#Eval("PRAC1") %>' />
                    </ItemTemplate>
                    <EditItemTemplate>
                        <asp:TextBox ID="txtPRAC1" Width="120px" runat="server" Text='<%#Eval("PRAC1") %>' />
                    </EditItemTemplate>
                    <FooterTemplate>
                        <asp:TextBox ID="inPRAC1" Width="120px" runat="server" />
                        <asp:RequiredFieldValidator ID="vPRAC1" runat="server" ControlToValidate="inPRAC1" Text="?" ValidationGroup="validaiton" />
                    </FooterTemplate>
                </asp:TemplateField>
                <asp:TemplateField HeaderText="Practical 2 mark">
                    <ItemTemplate>
                        <asp:Label ID="lblPRAC2" runat="server" Text='<%#Eval("PRAC2") %>' />
                    </ItemTemplate>
                    <EditItemTemplate>
                        <asp:TextBox ID="txtPRAC2" Width="120px" runat="server" Text='<%#Eval("PRAC2") %>' />
                    </EditItemTemplate>
                    <FooterTemplate>
                        <asp:TextBox ID="inPRAC2" Width="120px" runat="server" />
                        <asp:RequiredFieldValidator ID="vPRAC2" runat="server" ControlToValidate="inPRAC2" Text="?" ValidationGroup="validaiton" />
                    </FooterTemplate>
                </asp:TemplateField>
```



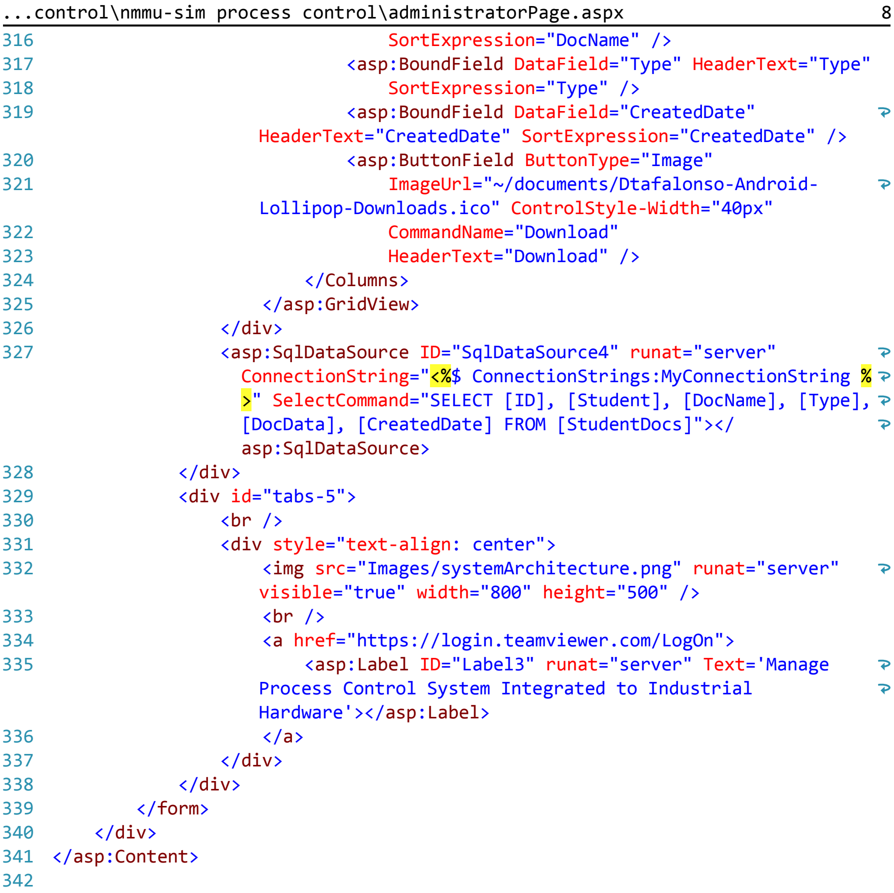
...control\nmmu-sim process control\administratorPage.aspx 5

```
                        <asp:TemplateField HeaderText="Final Practical Mark">
                            <ItemTemplate>
                                <asp:Label ID="lblFmark" runat="server" Text='<%#Eval("Fmark") %>' />
                            </ItemTemplate>
                            <EditItemTemplate>
                                <asp:TextBox ID="txtFmark" Width="120px" runat="server" Text='<%#Eval("Fmark") %>' />
                            </EditItemTemplate>
                            <FooterTemplate>
                                <asp:TextBox ID="inFmark" Width="120px" runat="server" />
                                <asp:RequiredFieldValidator ID="vFmark" runat="server" ControlToValidate="inFmark" Text="?" ValidationGroup="validaiton" />
                            </FooterTemplate>
                        </asp:TemplateField>
                        <asp:TemplateField>
                            <EditItemTemplate>
                                <asp:Button ID="ButtonUpdate" runat="server" CommandName="Update" Text="Update" />
                                <asp:Button ID="ButtonCancel" runat="server" CommandName="Cancel" Text="Cancel" />
                            </EditItemTemplate>
                            <ItemTemplate>
                                <asp:Button ID="ButtonEdit" runat="server" CommandName="Edit" Text="Edit" />
                            </ItemTemplate>
                            <FooterTemplate>
                            </FooterTemplate>
                        </asp:TemplateField>
                    </Columns>
                </asp:GridView>
            </div>
            <br />
            &nbsp;&nbsp;&nbsp;&nbsp;
            <asp:Label ID="lblmsg" runat="server"></asp:Label>
        </div>
        <div id="tabs-2">
            <br />
            <br />
            <br />
            <br />
            <asp:FileUpload ID="FileUpload1" runat="server" />
            <asp:Button ID="Button4" runat="server"
                OnClick="Button4_Click"
                Text="ButtonUpload" />
            <br />
            <asp:Label ID="lblMessage" runat="server">
            </asp:Label><br />
            <br />
            <br />
            <div style="margin-left: auto; margin-right: auto; width: 750px; background-color: white;">
                <asp:GridView ID="GridView3" runat="server"
        AutoGenerateColumns="False"
        DataSourceID="SqlDataSource2"
        OnRowCommand="GridView3_RowCommand"
        DataKeyNames="DocID"
    OnSelectedIndexChanged="GridView3_SelectedIndexChanged1">
        <Columns>
            <asp:BoundField DataField="DocID"
    HeaderText="DocID"
                InsertVisible="False"
                ReadOnly="True"
                SortExpression="DocID" />

            <asp:BoundField DataField="DocName"
                HeaderText="DocName"
                SortExpression="DocName" />

            <asp:BoundField DataField="Type" HeaderText="Type"
                SortExpression="Type" />

            <asp:BoundField DataField="CreatedDate"
    HeaderText="CreatedDate" SortExpression="CreatedDate" />
            <asp:ButtonField ButtonType="Image"
                ImageUrl="~/documents/Dtafalonso-Android-
    Lollipop-Downloads.ico" ControlStyle-Width="40px"
                CommandName="Download"
                HeaderText="Download" />
        </Columns>
    </asp:GridView>

    <asp:SqlDataSource ID="SqlDataSource2" runat="server"
    ConnectionString="<%$ ConnectionStrings:MyConnectionString
     %>" SelectCommand="SELECT [DocID], [DocName], [Type],
    [DocData],[CreatedDate] FROM [SaveDoc1]"></
    asp:SqlDataSource>
</div>
<div>
    <br />
    <br />
</div>
<div>
    <h2 style="color: #0066FF; font-weight: bold;">
        <u></u></h2>
</div>
<div>
    <asp:FileUpload ID="FileUploadToServer" Width="300px"
    runat="server" Visible="false" />
    <br />
    <br />
    <asp:Label ID="Label1" runat="server" ForeColor="Green"
    Text=""></asp:Label>
    <br />
    <div style="margin-left: auto; margin-right: auto; width:
    750px; background-color: white;">
    </div>

    <div>
```



...control\nmmu-sim process control\administratorPage.aspx 7

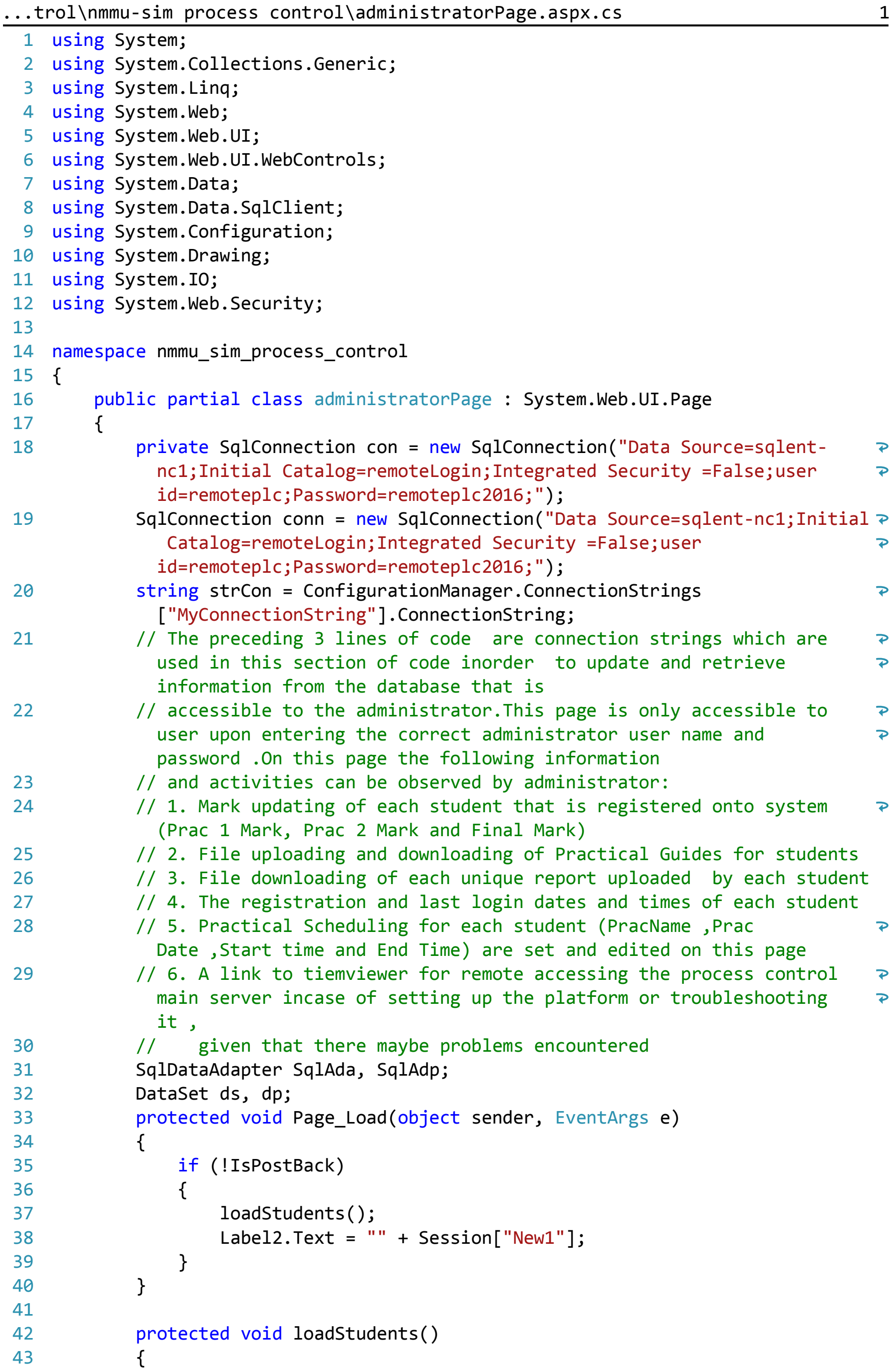
```
                    </div>
                    <div style="margin-left: auto; margin-right: auto; width: 750px; background-color: white;">
                    </div>
                </div>
            </div>
            <div id="tabs-4">
                <div style="margin-left: auto; margin-right: auto; width: 750px; background-color: white;">
                    <asp:GridView ID="GridView2" runat="server" AutoGenerateColumns="False" DataSourceID="SqlDataSource1">
                        <Columns>
                            <asp:BoundField DataField="Username" HeaderText="Username" HeaderStyle-Width="7%" SortExpression="Username" />
                            <asp:BoundField DataField="Email" HeaderText="Email" HeaderStyle-Width="20%" SortExpression="Email" />
                            <asp:BoundField DataField="CreatedDate" HeaderText="Registered Date" HeaderStyle-Width="25%" SortExpression="CreatedDate" />
                            <asp:BoundField DataField="LastLoginDate" HeaderText="LastLogin Date" HeaderStyle-Width="25%" SortExpression="LastLoginDate" />
                        </Columns>
                    </asp:GridView>
                </div>
                <asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:MyConnectionString %>" SelectCommand="SELECT [Username], [Email], [CreatedDate], [LastLoginDate] FROM [Users]"></asp:SqlDataSource>
            </div>
            <asp:HiddenField ID="selected_tab" runat="server" />
            <div id="tabs-3">
                <br />
                <br />
                <br />
                <br />
                <div style="margin-left: auto; margin-right: auto; width: 750px; background-color: white;">
                    <asp:GridView ID="GridView4" runat="server"
                        AutoGenerateColumns="False"
                        DataSourceID="SqlDataSource4"
                        OnRowCommand="GridView4_RowCommand"
                        DataKeyNames="ID" OnSelectedIndexChanged="GridView4_SelectedIndexChanged1">
                        <Columns>
                            <asp:BoundField DataField="ID" HeaderText="ID"
                                InsertVisible="False"
                                ReadOnly="True"
                                SortExpression="ID" />
                            <asp:BoundField DataField="Student"
                                HeaderText="Student"
                                SortExpression="Student" />
                            <asp:BoundField DataField="DocName"
                                HeaderText="DocName"
```



...control\nmmu-sim process control\administratorPage.aspx 8

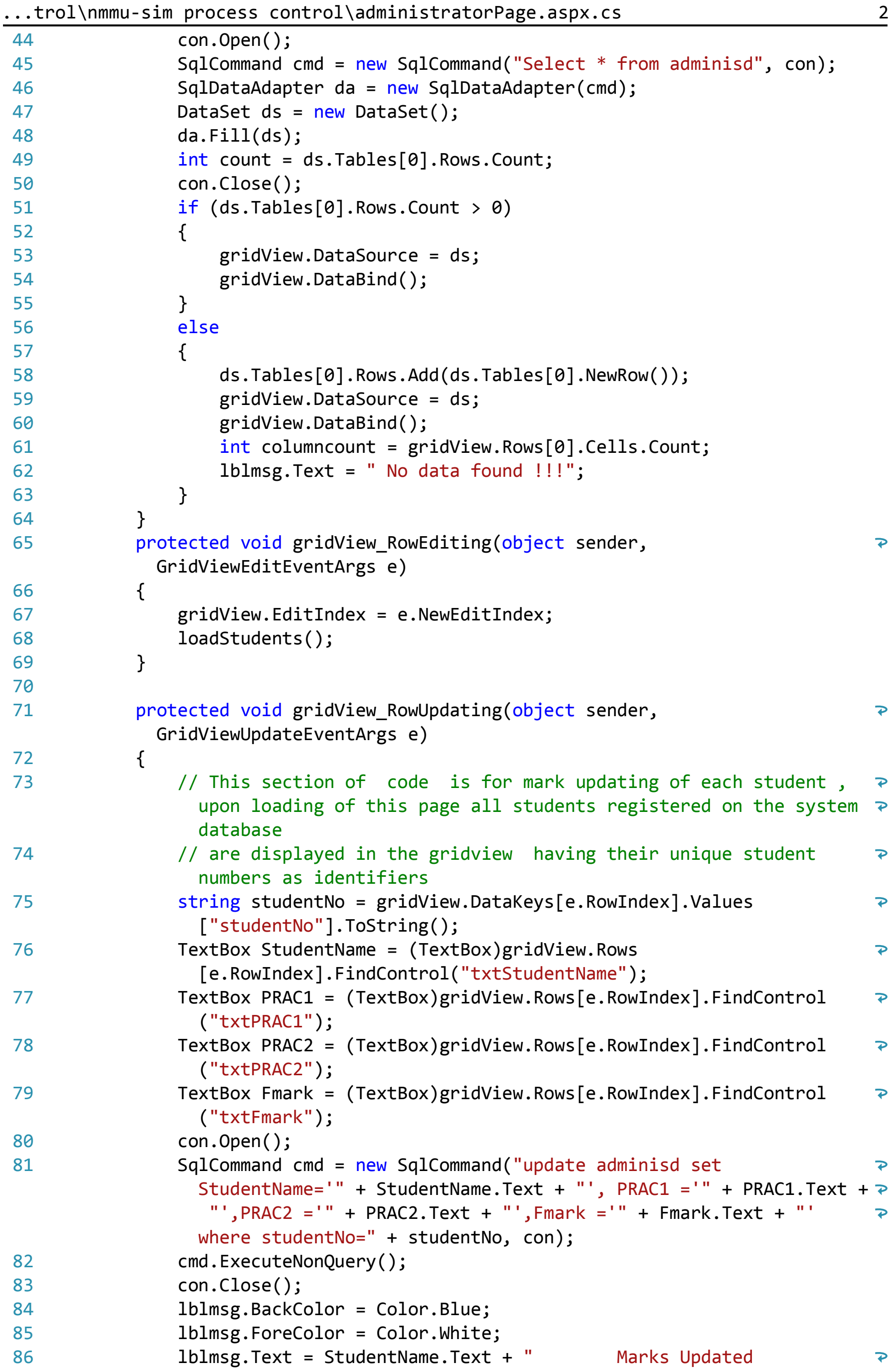
```
                                SortExpression="DocName" />
                            <asp:BoundField DataField="Type" HeaderText="Type"
                                SortExpression="Type" />
                            <asp:BoundField DataField="CreatedDate" HeaderText="CreatedDate" SortExpression="CreatedDate" />
                            <asp:ButtonField ButtonType="Image"
                                ImageUrl="~/documents/Dtafalonso-Android-Lollipop-Downloads.ico" ControlStyle-Width="40px"
                                CommandName="Download"
                                HeaderText="Download" />
                        </Columns>
                    </asp:GridView>
                </div>
                <asp:SqlDataSource ID="SqlDataSource4" runat="server" ConnectionString="<%$ ConnectionStrings:MyConnectionString %>" SelectCommand="SELECT [ID], [Student], [DocName], [Type], [DocData], [CreatedDate] FROM [StudentDocs]"></asp:SqlDataSource>
            </div>
            <div id="tabs-5">
                <br />
                <div style="text-align: center">
                    <img src="Images/systemArchitecture.png" runat="server" visible="true" width="800" height="500" />
                    <br />
                    <a href="https://login.teamviewer.com/LogOn">
                        <asp:Label ID="Label3" runat="server" Text='Manage Process Control System Integrated to Industrial Hardware'></asp:Label>
                    </a>
                </div>
            </div>
        </form>
    </div>
</asp:Content>
```

7.3.7 Administrator Page : C# Code



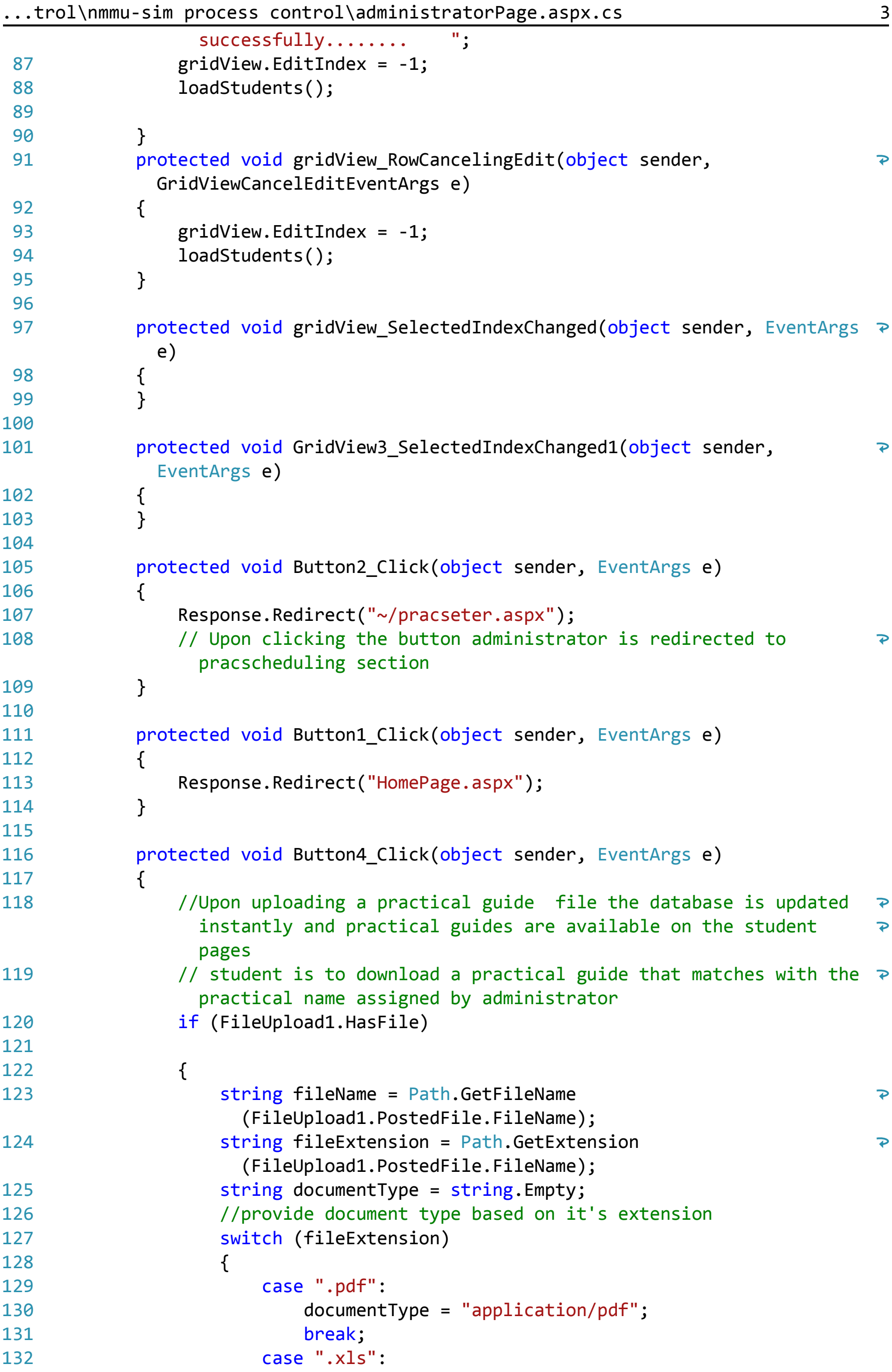
...trol\nmmu-sim process control\administratorPage.aspx.cs 1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Drawing;
using System.IO;
using System.Web.Security;

namespace nmmu_sim_process_control
{
    public partial class administratorPage : System.Web.UI.Page
    {
        private SqlConnection con = new SqlConnection("Data Source=sqlent-nc1;Initial Catalog=remoteLogin;Integrated Security =False;user id=remoteplc;Password=remoteplc2016;");
        SqlConnection conn = new SqlConnection("Data Source=sqlent-nc1;Initial Catalog=remoteLogin;Integrated Security =False;user id=remoteplc;Password=remoteplc2016;");
        string strCon = ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString;
        // The preceding 3 lines of code  are connection strings which are used in this section of code inorder  to update and retrieve information from the database that is
        // accessible to the administrator.This page is only accessible to user upon entering the correct administrator user name and password .On this page the following information
        // and activities can be observed by administrator:
        // 1. Mark updating of each student that is registered onto system (Prac 1 Mark, Prac 2 Mark and Final Mark)
        // 2. File uploading and downloading of Practical Guides for students
        // 3. File downloading of each unique report uploaded  by each student
        // 4. The registration and last login dates and times of each student
        // 5. Practical Scheduling for each student (PracName ,Prac Date ,Start time and End Time) are set and edited on this page
        // 6. A link to tiemviewer for remote accessing the process control main server incase of setting up the platform or troubleshooting it ,
        //    given that there maybe problems encountered
        SqlDataAdapter SqlAda, SqlAdp;
        DataSet ds, dp;
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                loadStudents();
                Label2.Text = "" + Session["New1"];
            }
        }

        protected void loadStudents()
        {
```



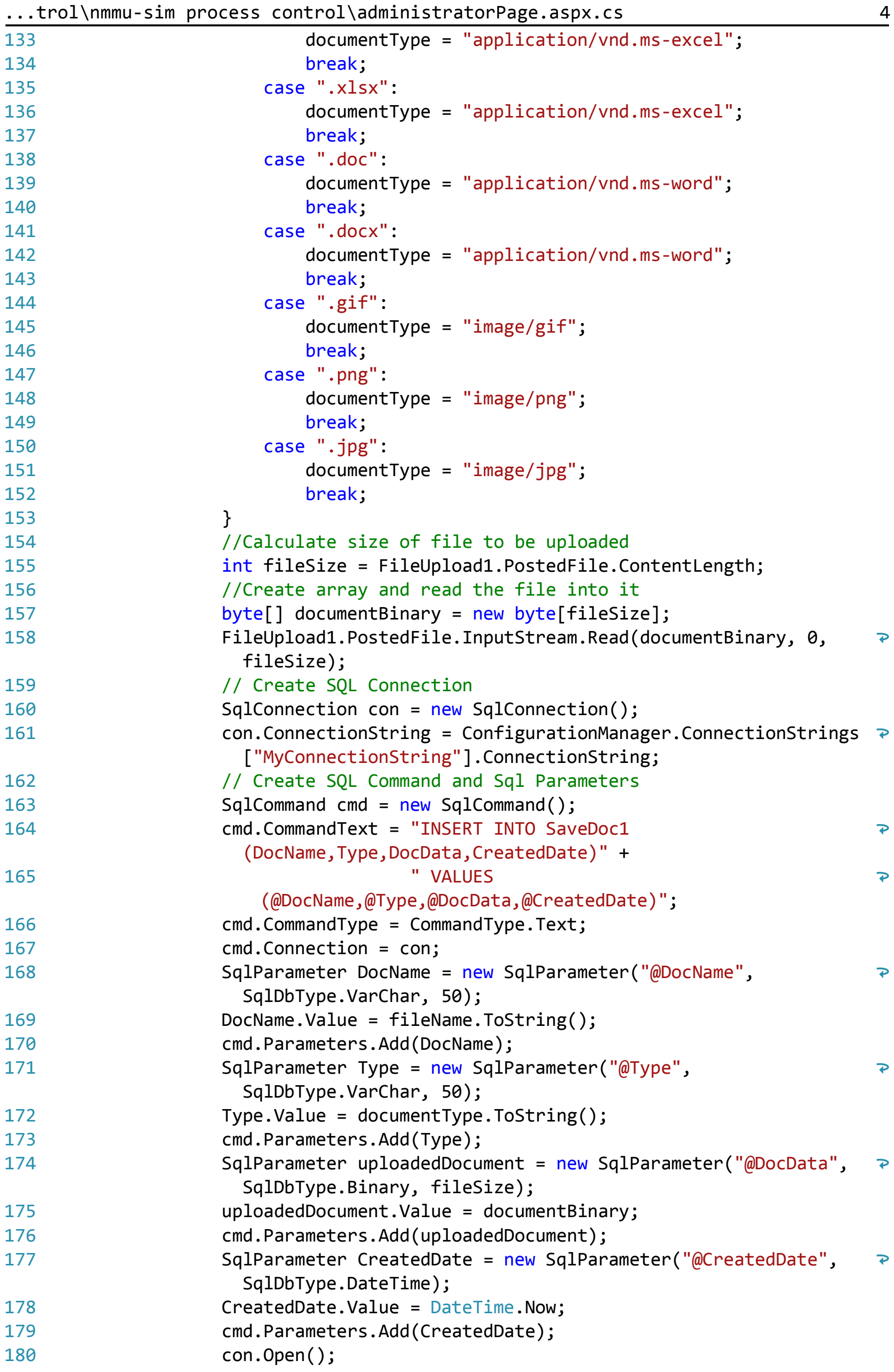
...trol\nmmu-sim process control\administratorPage.aspx.cs 2

```
            con.Open();
            SqlCommand cmd = new SqlCommand("Select * from adminisd", con);
            SqlDataAdapter da = new SqlDataAdapter(cmd);
            DataSet ds = new DataSet();
            da.Fill(ds);
            int count = ds.Tables[0].Rows.Count;
            con.Close();
            if (ds.Tables[0].Rows.Count > 0)
            {
                gridView.DataSource = ds;
                gridView.DataBind();
            }
            else
            {
                ds.Tables[0].Rows.Add(ds.Tables[0].NewRow());
                gridView.DataSource = ds;
                gridView.DataBind();
                int columncount = gridView.Rows[0].Cells.Count;
                lblmsg.Text = " No data found !!!";
            }
        }
        protected void gridView_RowEditing(object sender, GridViewEditEventArgs e)
        {
            gridView.EditIndex = e.NewEditIndex;
            loadStudents();
        }

        protected void gridView_RowUpdating(object sender, GridViewUpdateEventArgs e)
        {
            // This section of  code  is for mark updating of each student , upon loading of this page all students registered on the system database
            // are displayed in the gridview  having their unique student numbers as identifiers
            string studentNo = gridView.DataKeys[e.RowIndex].Values["studentNo"].ToString();
            TextBox StudentName = (TextBox)gridView.Rows[e.RowIndex].FindControl("txtStudentName");
            TextBox PRAC1 = (TextBox)gridView.Rows[e.RowIndex].FindControl("txtPRAC1");
            TextBox PRAC2 = (TextBox)gridView.Rows[e.RowIndex].FindControl("txtPRAC2");
            TextBox Fmark = (TextBox)gridView.Rows[e.RowIndex].FindControl("txtFmark");
            con.Open();
            SqlCommand cmd = new SqlCommand("update adminisd set StudentName='" + StudentName.Text + "', PRAC1 ='" + PRAC1.Text + "',PRAC2 ='" + PRAC2.Text + "',Fmark ='" + Fmark.Text + "' where studentNo=" + studentNo, con);
            cmd.ExecuteNonQuery();
            con.Close();
            lblmsg.BackColor = Color.Blue;
            lblmsg.ForeColor = Color.White;
            lblmsg.Text = StudentName.Text + "        Marks Updated
```



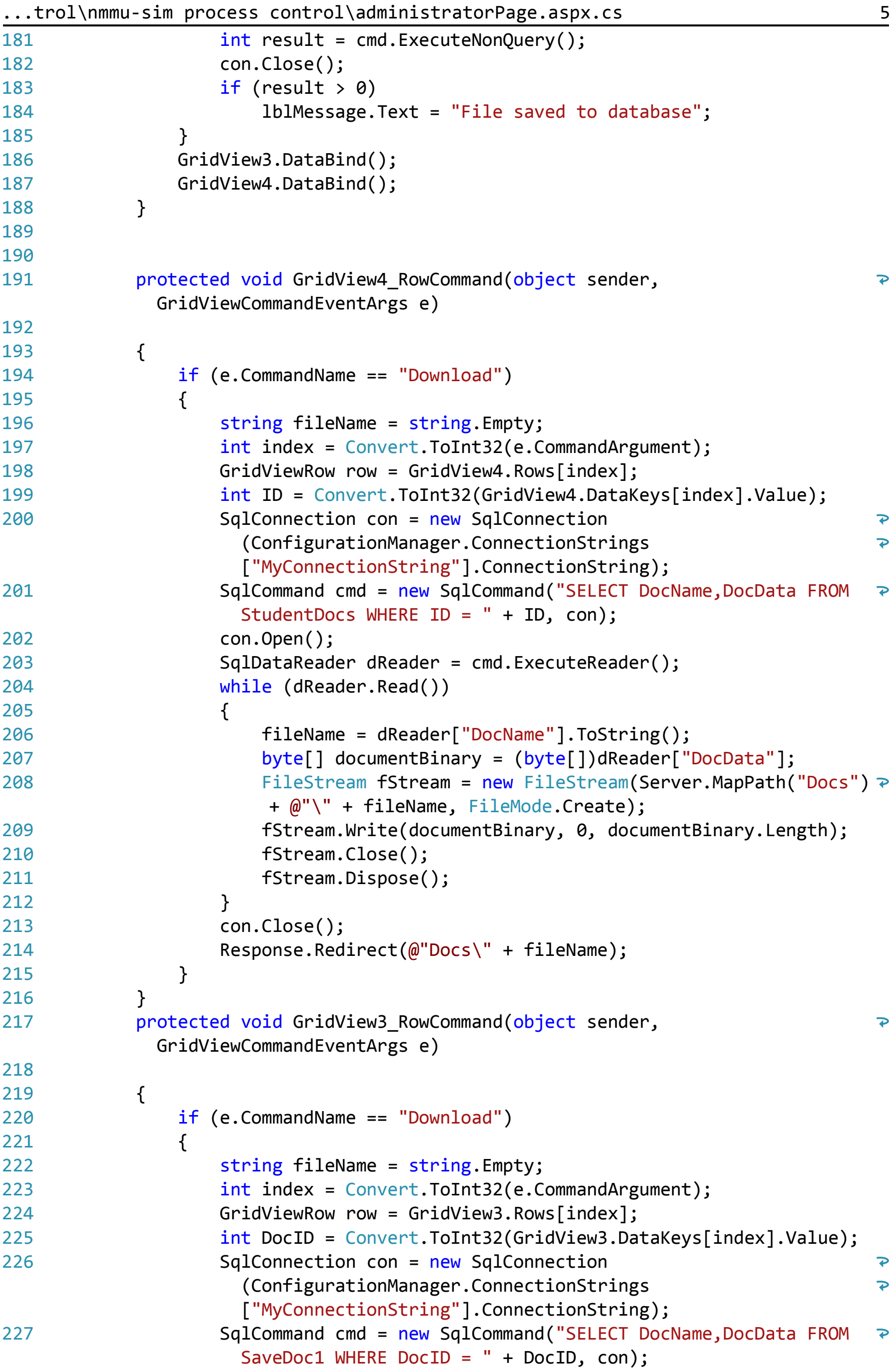
...trol\nmmu-sim process control\administratorPage.aspx.cs 3

```
              successfully........    ";
            gridView.EditIndex = -1;
            loadStudents();

        }
        protected void gridView_RowCancelingEdit(object sender, GridViewCancelEditEventArgs e)
        {
            gridView.EditIndex = -1;
            loadStudents();
        }

        protected void gridView_SelectedIndexChanged(object sender, EventArgs e)
        {
        }

        protected void GridView3_SelectedIndexChanged1(object sender, EventArgs e)
        {
        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            Response.Redirect("~/pracseter.aspx");
            // Upon clicking the button administrator is redirected to pracscheduling section
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Redirect("HomePage.aspx");
        }

        protected void Button4_Click(object sender, EventArgs e)
        {
            //Upon uploading a practical guide  file the database is updated instantly and practical guides are available on the student pages
            // student is to download a practical guide that matches with the practical name assigned by administrator
            if (FileUpload1.HasFile)

            {
                string fileName = Path.GetFileName(FileUpload1.PostedFile.FileName);
                string fileExtension = Path.GetExtension(FileUpload1.PostedFile.FileName);
                string documentType = string.Empty;
                //provide document type based on it's extension
                switch (fileExtension)
                {
                    case ".pdf":
                        documentType = "application/pdf";
                        break;
                    case ".xls":
```

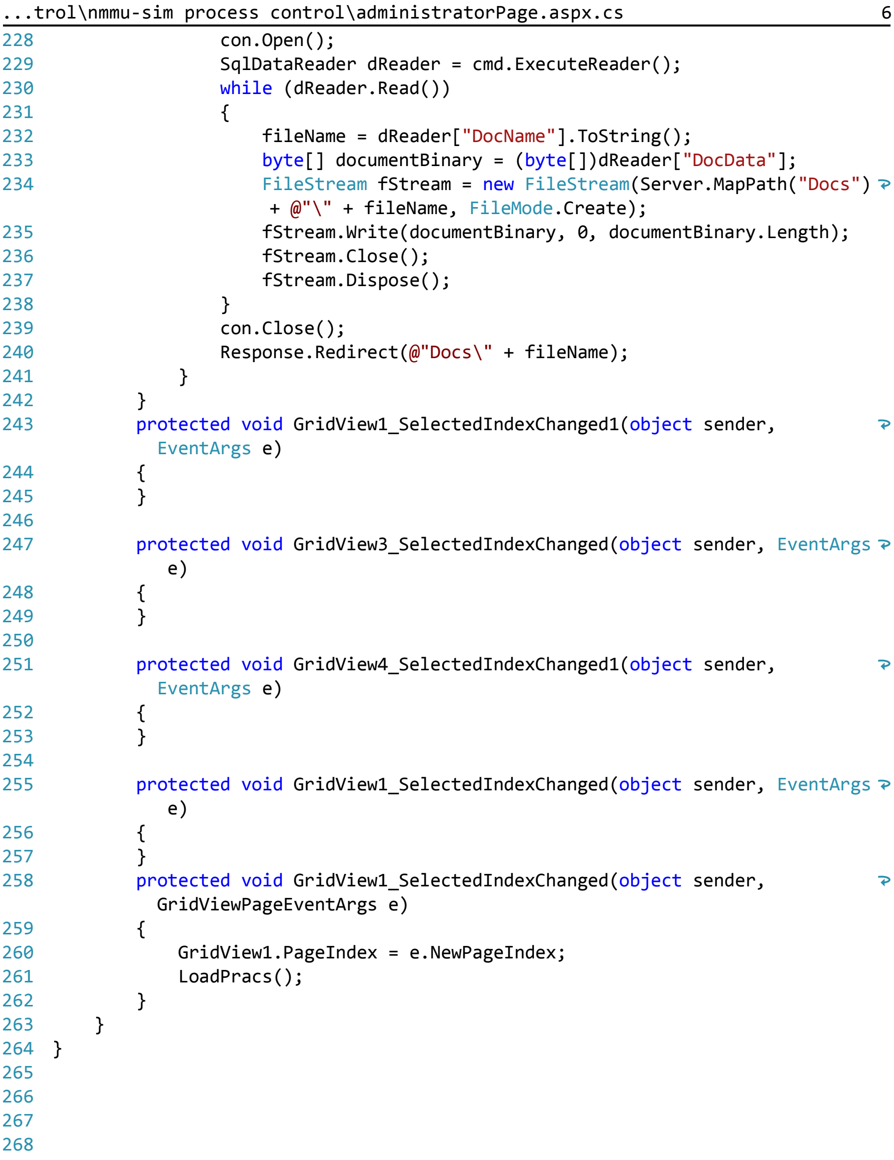


...trol\nmmu-sim process control\administratorPage.aspx.cs 4

```
                        documentType = "application/vnd.ms-excel";
                        break;
                    case ".xlsx":
                        documentType = "application/vnd.ms-excel";
                        break;
                    case ".doc":
                        documentType = "application/vnd.ms-word";
                        break;
                    case ".docx":
                        documentType = "application/vnd.ms-word";
                        break;
                    case ".gif":
                        documentType = "image/gif";
                        break;
                    case ".png":
                        documentType = "image/png";
                        break;
                    case ".jpg":
                        documentType = "image/jpg";
                        break;
                }
                //Calculate size of file to be uploaded
                int fileSize = FileUpload1.PostedFile.ContentLength;
                //Create array and read the file into it
                byte[] documentBinary = new byte[fileSize];
                FileUpload1.PostedFile.InputStream.Read(documentBinary, 0, fileSize);
                // Create SQL Connection
                SqlConnection con = new SqlConnection();
                con.ConnectionString = ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString;
                // Create SQL Command and Sql Parameters
                SqlCommand cmd = new SqlCommand();
                cmd.CommandText = "INSERT INTO SaveDoc1 (DocName,Type,DocData,CreatedDate)" +
                                  " VALUES (@DocName,@Type,@DocData,@CreatedDate)";
                cmd.CommandType = CommandType.Text;
                cmd.Connection = con;
                SqlParameter DocName = new SqlParameter("@DocName", SqlDbType.VarChar, 50);
                DocName.Value = fileName.ToString();
                cmd.Parameters.Add(DocName);
                SqlParameter Type = new SqlParameter("@Type", SqlDbType.VarChar, 50);
                Type.Value = documentType.ToString();
                cmd.Parameters.Add(Type);
                SqlParameter uploadedDocument = new SqlParameter("@DocData", SqlDbType.Binary, fileSize);
                uploadedDocument.Value = documentBinary;
                cmd.Parameters.Add(uploadedDocument);
                SqlParameter CreatedDate = new SqlParameter("@CreatedDate", SqlDbType.DateTime);
                CreatedDate.Value = DateTime.Now;
                cmd.Parameters.Add(CreatedDate);
                con.Open();
```



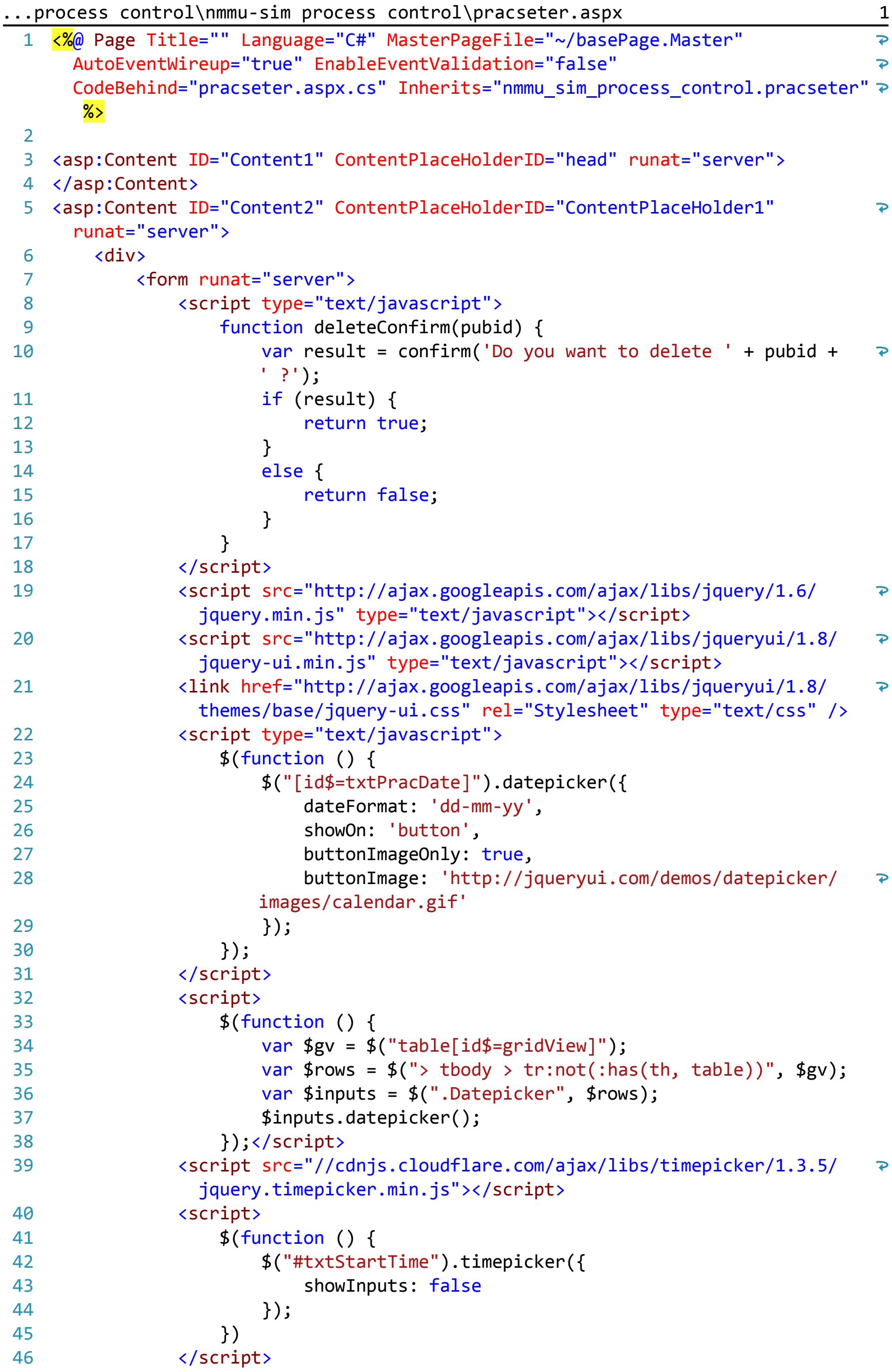
```
...trol\nmmu-sim process control\administratorPage.aspx.cs                    5
                int result = cmd.ExecuteNonQuery();
                con.Close();
                if (result > 0)
                    lblMessage.Text = "File saved to database";
            }
            GridView3.DataBind();
            GridView4.DataBind();
        }


        protected void GridView4_RowCommand(object sender,
          GridViewCommandEventArgs e)

        {
            if (e.CommandName == "Download")
            {
                string fileName = string.Empty;
                int index = Convert.ToInt32(e.CommandArgument);
                GridViewRow row = GridView4.Rows[index];
                int ID = Convert.ToInt32(GridView4.DataKeys[index].Value);
                SqlConnection con = new SqlConnection
                  (ConfigurationManager.ConnectionStrings
                  ["MyConnectionString"].ConnectionString);
                SqlCommand cmd = new SqlCommand("SELECT DocName,DocData FROM
                  StudentDocs WHERE ID = " + ID, con);
                con.Open();
                SqlDataReader dReader = cmd.ExecuteReader();
                while (dReader.Read())
                {
                    fileName = dReader["DocName"].ToString();
                    byte[] documentBinary = (byte[])dReader["DocData"];
                    FileStream fStream = new FileStream(Server.MapPath("Docs")
                     + @"\" + fileName, FileMode.Create);
                    fStream.Write(documentBinary, 0, documentBinary.Length);
                    fStream.Close();
                    fStream.Dispose();
                }
                con.Close();
                Response.Redirect(@"Docs\" + fileName);
            }
        }
        protected void GridView3_RowCommand(object sender,
          GridViewCommandEventArgs e)

        {
            if (e.CommandName == "Download")
            {
                string fileName = string.Empty;
                int index = Convert.ToInt32(e.CommandArgument);
                GridViewRow row = GridView3.Rows[index];
                int DocID = Convert.ToInt32(GridView3.DataKeys[index].Value);
                SqlConnection con = new SqlConnection
                  (ConfigurationManager.ConnectionStrings
                  ["MyConnectionString"].ConnectionString);
                SqlCommand cmd = new SqlCommand("SELECT DocName,DocData FROM
                  SaveDoc1 WHERE DocID = " + DocID, con);
```



...trol\nmmu-sim process control\administratorPage.aspx.cs 6

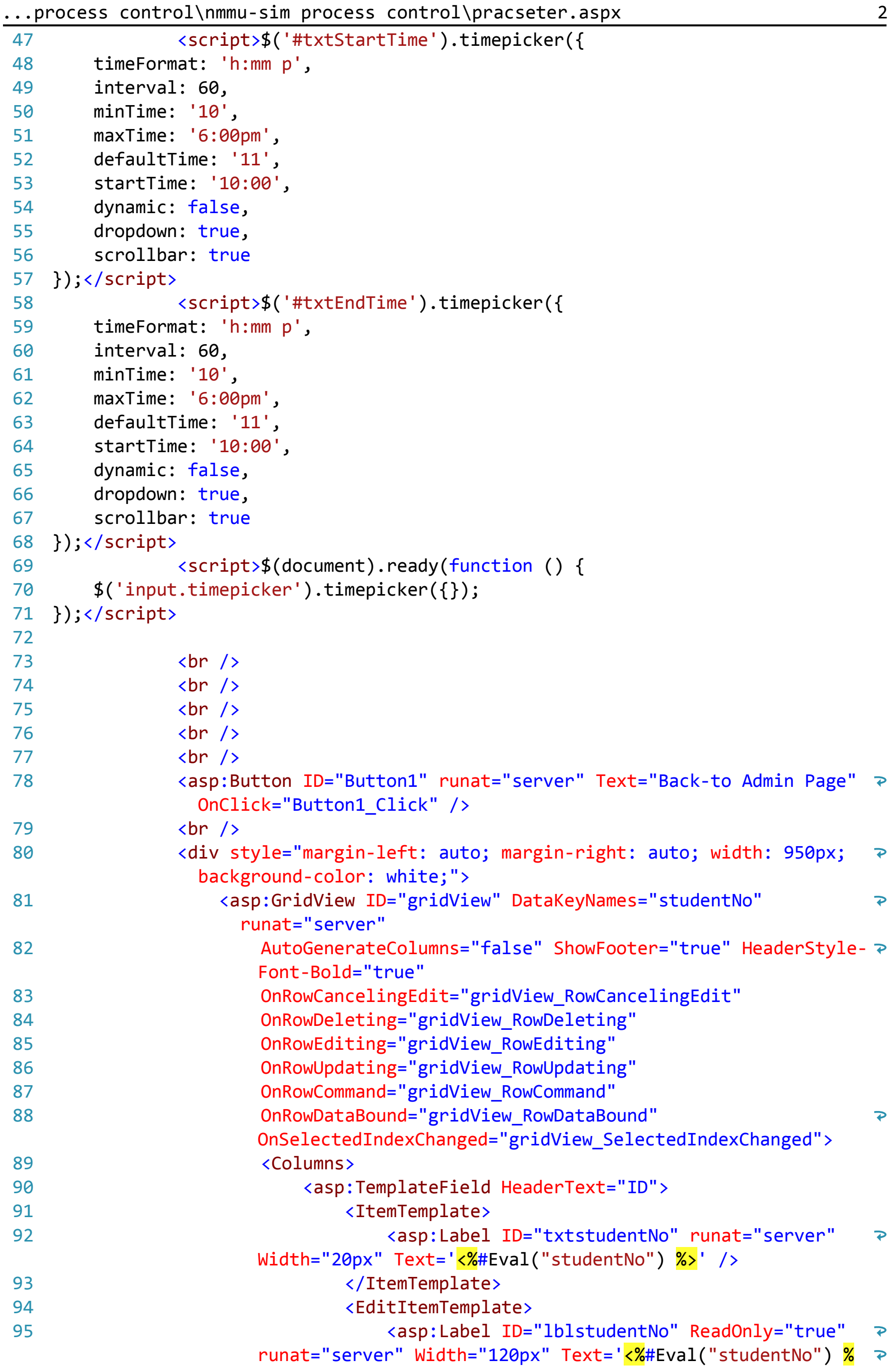
```
                con.Open();
                SqlDataReader dReader = cmd.ExecuteReader();
                while (dReader.Read())
                {
                    fileName = dReader["DocName"].ToString();
                    byte[] documentBinary = (byte[])dReader["DocData"];
                    FileStream fStream = new FileStream(Server.MapPath("Docs") + @"\" + fileName, FileMode.Create);
                    fStream.Write(documentBinary, 0, documentBinary.Length);
                    fStream.Close();
                    fStream.Dispose();
                }
                con.Close();
                Response.Redirect(@"Docs\" + fileName);
            }
        }
        protected void GridView1_SelectedIndexChanged1(object sender, EventArgs e)
        {
        }

        protected void GridView3_SelectedIndexChanged(object sender, EventArgs e)
        {
        }

        protected void GridView4_SelectedIndexChanged1(object sender, EventArgs e)
        {
        }

        protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
        {
        }
        protected void GridView1_SelectedIndexChanged(object sender, GridViewPageEventArgs e)
        {
            GridView1.PageIndex = e.NewPageIndex;
            LoadPracs();
        }
    }
}
```

7.3.8 Admin-Practical Scheduling : HTML Mark-Up

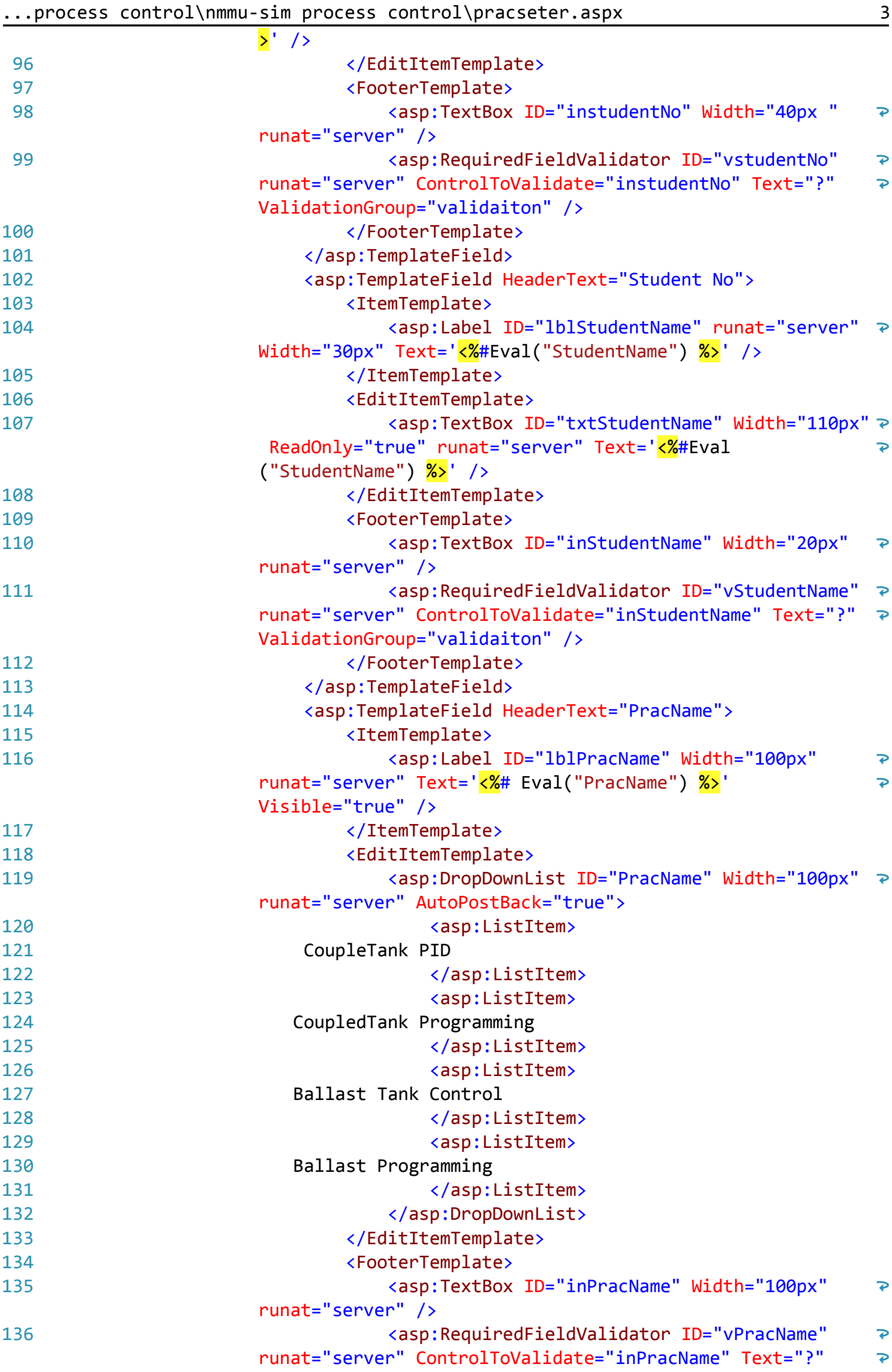


...process control\nmmu-sim process control\pracseter.aspx 1

```
<%@ Page Title="" Language="C#" MasterPageFile="~/basePage.Master" AutoEventWireup="true" EnableEventValidation="false" CodeBehind="pracseter.aspx.cs" Inherits="nmmu_sim_process_control.pracseter" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <div>
        <form runat="server">
            <script type="text/javascript">
                function deleteConfirm(pubid) {
                    var result = confirm('Do you want to delete ' + pubid + ' ?');
                    if (result) {
                        return true;
                    }
                    else {
                        return false;
                    }
                }
            </script>
            <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.6/jquery.min.js" type="text/javascript"></script>
            <script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/jquery-ui.min.js" type="text/javascript"></script>
            <link href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/themes/base/jquery-ui.css" rel="Stylesheet" type="text/css" />
            <script type="text/javascript">
                $(function () {
                    $("[id$=txtPracDate]").datepicker({
                        dateFormat: 'dd-mm-yy',
                        showOn: 'button',
                        buttonImageOnly: true,
                        buttonImage: 'http://jqueryui.com/demos/datepicker/images/calendar.gif'
                    });
                });
            </script>
            <script>
                $(function () {
                    var $gv = $("table[id$=gridView]");
                    var $rows = $("> tbody > tr:not(:has(th, table))", $gv);
                    var $inputs = $(".Datepicker", $rows);
                    $inputs.datepicker();
                });</script>
            <script src="//cdnjs.cloudflare.com/ajax/libs/timepicker/1.3.5/jquery.timepicker.min.js"></script>
            <script>
                $(function () {
                    $("#txtStartTime").timepicker({
                        showInputs: false
                    });
                })
            </script>
```

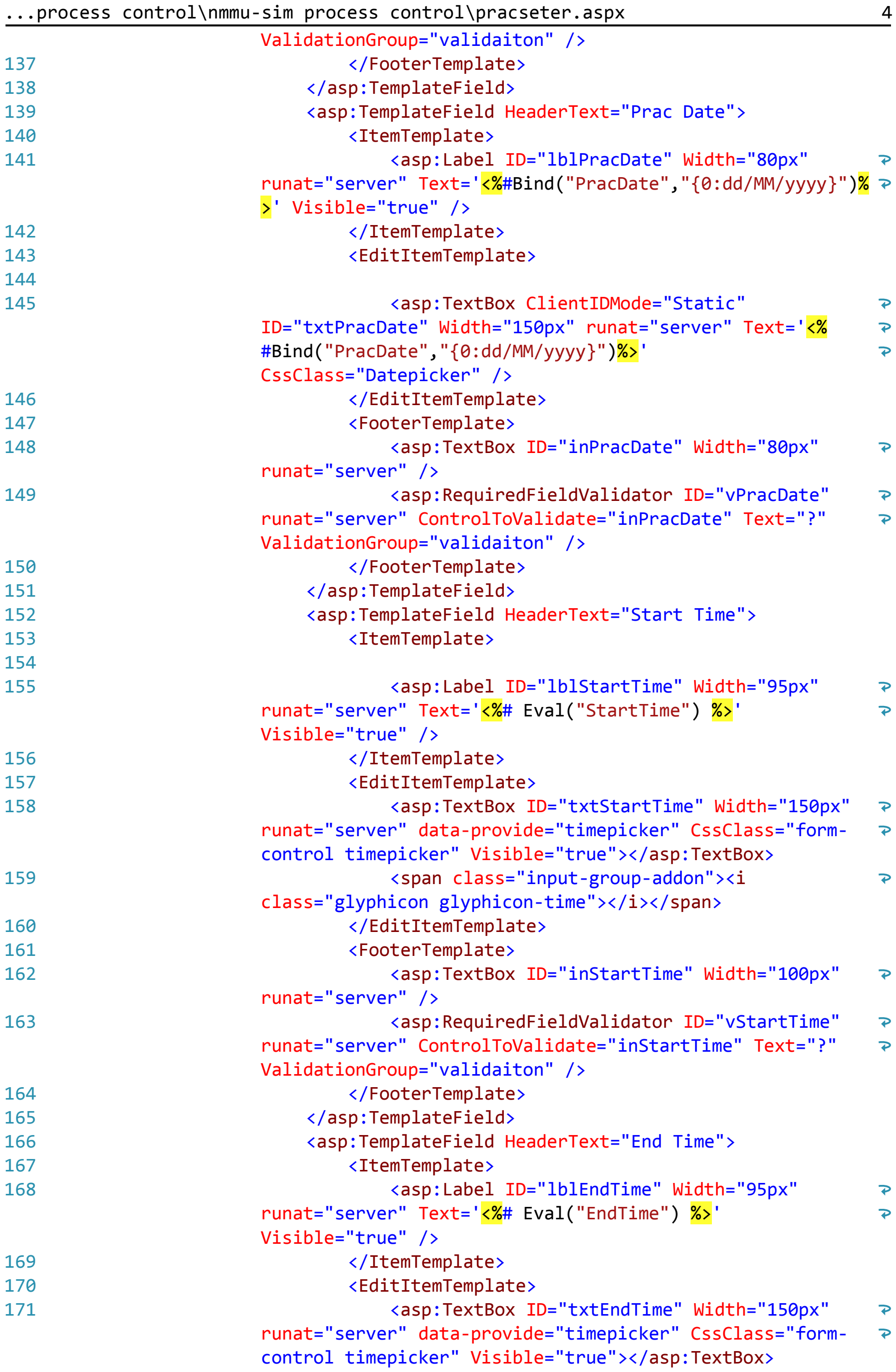


```
      <script>$('#txtStartTime').timepicker({
  timeFormat: 'h:mm p',
  interval: 60,
  minTime: '10',
  maxTime: '6:00pm',
  defaultTime: '11',
  startTime: '10:00',
  dynamic: false,
  dropdown: true,
  scrollbar: true
});</script>
      <script>$('#txtEndTime').timepicker({
  timeFormat: 'h:mm p',
  interval: 60,
  minTime: '10',
  maxTime: '6:00pm',
  defaultTime: '11',
  startTime: '10:00',
  dynamic: false,
  dropdown: true,
  scrollbar: true
});</script>
      <script>$(document).ready(function () {
  $('input.timepicker').timepicker({});
});</script>

      <br />
      <br />
      <br />
      <br />
      <br />
      <asp:Button ID="Button1" runat="server" Text="Back-to Admin Page" OnClick="Button1_Click" />
      <br />
      <div style="margin-left: auto; margin-right: auto; width: 950px; background-color: white;">
        <asp:GridView ID="gridView" DataKeyNames="studentNo" runat="server"
          AutoGenerateColumns="false" ShowFooter="true" HeaderStyle-Font-Bold="true"
          OnRowCancelingEdit="gridView_RowCancelingEdit"
          OnRowDeleting="gridView_RowDeleting"
          OnRowEditing="gridView_RowEditing"
          OnRowUpdating="gridView_RowUpdating"
          OnRowCommand="gridView_RowCommand"
          OnRowDataBound="gridView_RowDataBound" OnSelectedIndexChanged="gridView_SelectedIndexChanged">
          <Columns>
            <asp:TemplateField HeaderText="ID">
              <ItemTemplate>
                <asp:Label ID="txtstudentNo" runat="server" Width="20px" Text='<%#Eval("studentNo") %>' />
              </ItemTemplate>
              <EditItemTemplate>
                <asp:Label ID="lblstudentNo" ReadOnly="true" runat="server" Width="120px" Text='<%#Eval("studentNo") %
```



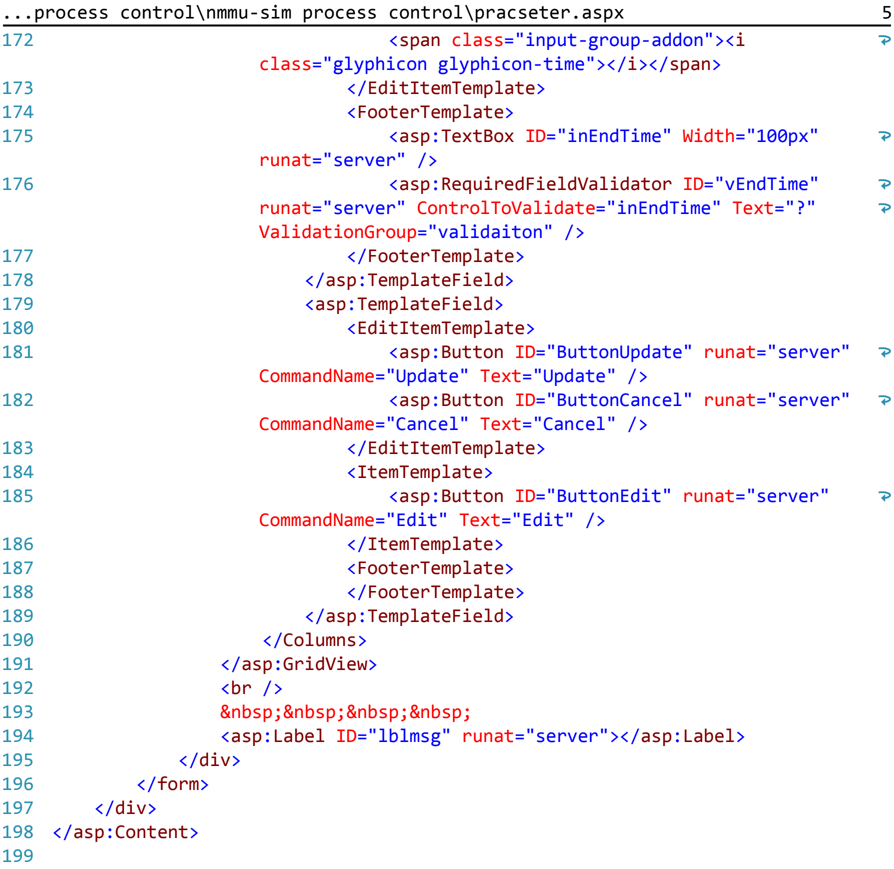
...process control\nmmu-sim process control\pracseter.aspx 3

```
%>' />
        </EditItemTemplate>
        <FooterTemplate>
            <asp:TextBox ID="instudentNo" Width="40px " runat="server" />
            <asp:RequiredFieldValidator ID="vstudentNo" runat="server" ControlToValidate="instudentNo" Text="?" ValidationGroup="validaiton" />
        </FooterTemplate>
    </asp:TemplateField>
    <asp:TemplateField HeaderText="Student No">
        <ItemTemplate>
            <asp:Label ID="lblStudentName" runat="server" Width="30px" Text='<%#Eval("StudentName") %>' />
        </ItemTemplate>
        <EditItemTemplate>
            <asp:TextBox ID="txtStudentName" Width="110px" ReadOnly="true" runat="server" Text='<%#Eval("StudentName") %>' />
        </EditItemTemplate>
        <FooterTemplate>
            <asp:TextBox ID="inStudentName" Width="20px" runat="server" />
            <asp:RequiredFieldValidator ID="vStudentName" runat="server" ControlToValidate="inStudentName" Text="?" ValidationGroup="validaiton" />
        </FooterTemplate>
    </asp:TemplateField>
    <asp:TemplateField HeaderText="PracName">
        <ItemTemplate>
            <asp:Label ID="lblPracName" Width="100px" runat="server" Text='<%# Eval("PracName") %>' Visible="true" />
        </ItemTemplate>
        <EditItemTemplate>
            <asp:DropDownList ID="PracName" Width="100px" runat="server" AutoPostBack="true">
                <asp:ListItem>
    CoupleTank PID
                </asp:ListItem>
                <asp:ListItem>
   CoupledTank Programming
                </asp:ListItem>
                <asp:ListItem>
   Ballast Tank Control
                </asp:ListItem>
                <asp:ListItem>
   Ballast Programming
                </asp:ListItem>
            </asp:DropDownList>
        </EditItemTemplate>
        <FooterTemplate>
            <asp:TextBox ID="inPracName" Width="100px" runat="server" />
            <asp:RequiredFieldValidator ID="vPracName" runat="server" ControlToValidate="inPracName" Text="?"
```



...process control\nmmu-sim process control\pracseter.aspx 4

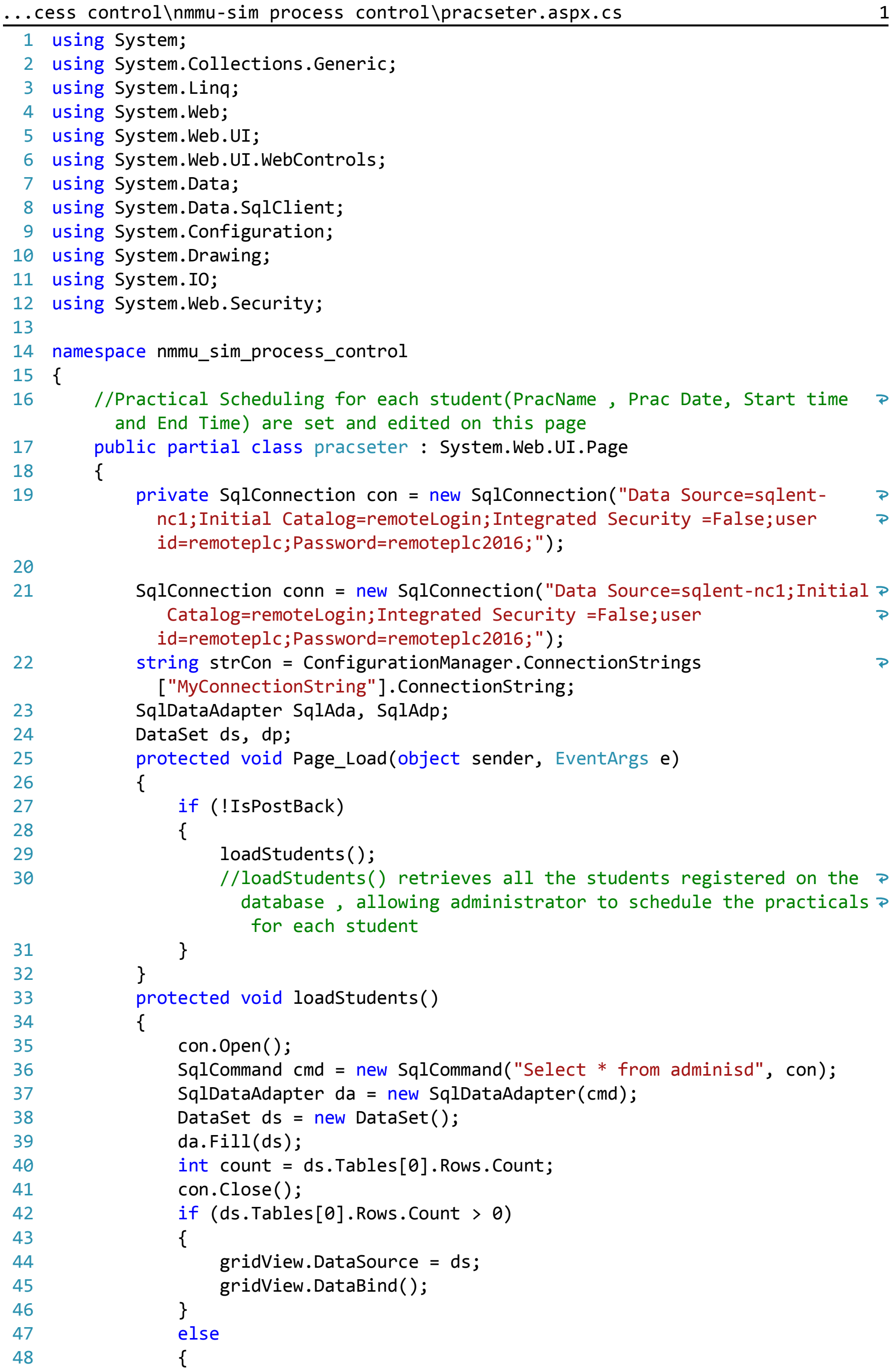
```
ValidationGroup="validaiton" />
                    </FooterTemplate>
                </asp:TemplateField>
                <asp:TemplateField HeaderText="Prac Date">
                    <ItemTemplate>
                        <asp:Label ID="lblPracDate" Width="80px" runat="server" Text='<%#Bind("PracDate","{0:dd/MM/yyyy}")%>' Visible="true" />
                    </ItemTemplate>
                    <EditItemTemplate>

                        <asp:TextBox ClientIDMode="Static" ID="txtPracDate" Width="150px" runat="server" Text='<%#Bind("PracDate","{0:dd/MM/yyyy}")%>' CssClass="Datepicker" />
                    </EditItemTemplate>
                    <FooterTemplate>
                        <asp:TextBox ID="inPracDate" Width="80px" runat="server" />
                        <asp:RequiredFieldValidator ID="vPracDate" runat="server" ControlToValidate="inPracDate" Text="?" ValidationGroup="validaiton" />
                    </FooterTemplate>
                </asp:TemplateField>
                <asp:TemplateField HeaderText="Start Time">
                    <ItemTemplate>

                        <asp:Label ID="lblStartTime" Width="95px" runat="server" Text='<%# Eval("StartTime") %>' Visible="true" />
                    </ItemTemplate>
                    <EditItemTemplate>
                        <asp:TextBox ID="txtStartTime" Width="150px" runat="server" data-provide="timepicker" CssClass="form-control timepicker" Visible="true"></asp:TextBox>
                        <span class="input-group-addon"><i class="glyphicon glyphicon-time"></i></span>
                    </EditItemTemplate>
                    <FooterTemplate>
                        <asp:TextBox ID="inStartTime" Width="100px" runat="server" />
                        <asp:RequiredFieldValidator ID="vStartTime" runat="server" ControlToValidate="inStartTime" Text="?" ValidationGroup="validaiton" />
                    </FooterTemplate>
                </asp:TemplateField>
                <asp:TemplateField HeaderText="End Time">
                    <ItemTemplate>
                        <asp:Label ID="lblEndTime" Width="95px" runat="server" Text='<%# Eval("EndTime") %>' Visible="true" />
                    </ItemTemplate>
                    <EditItemTemplate>
                        <asp:TextBox ID="txtEndTime" Width="150px" runat="server" data-provide="timepicker" CssClass="form-control timepicker" Visible="true"></asp:TextBox>
```



...process control\nmmu-sim process control\pracseter.aspx 5

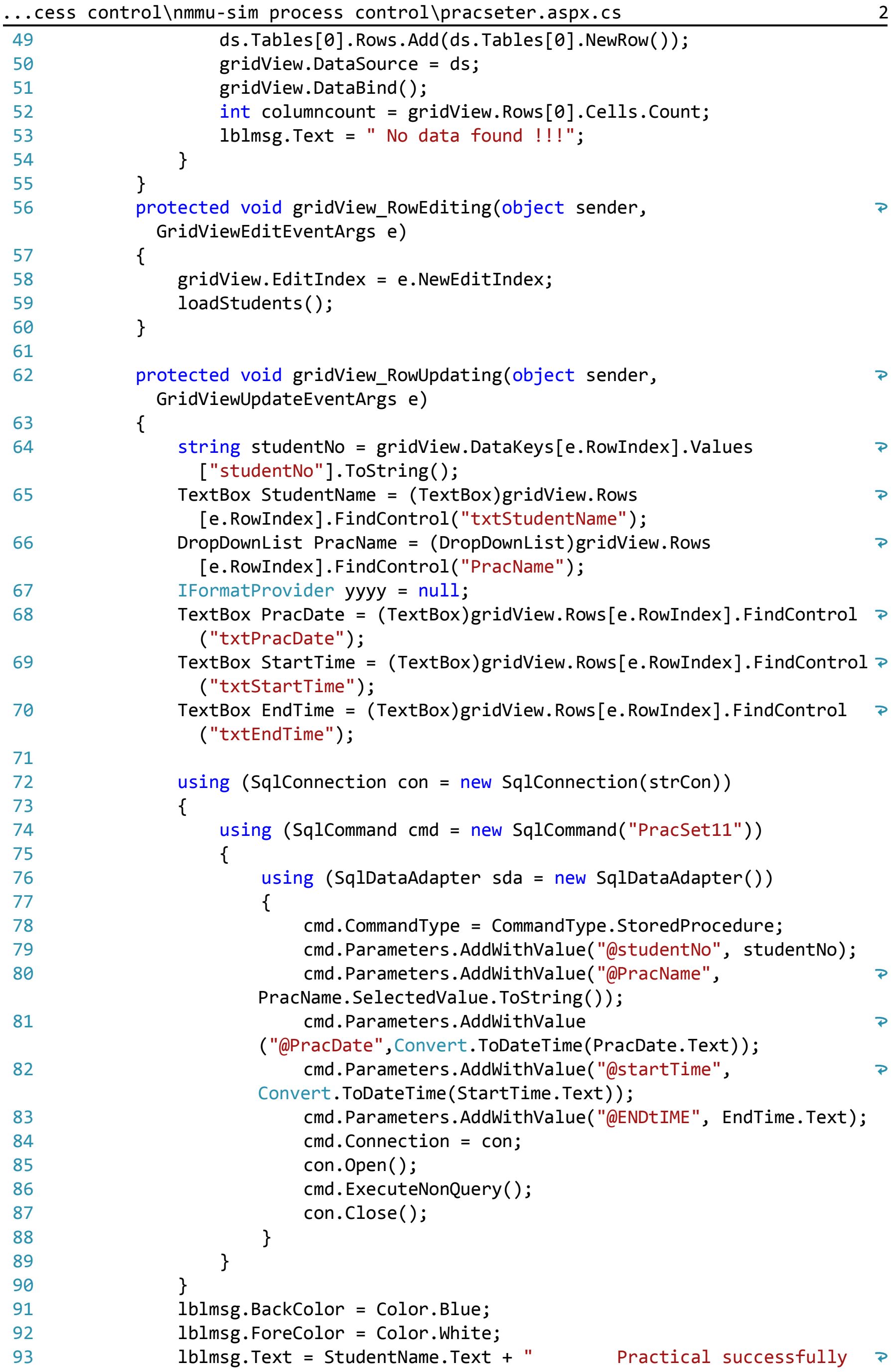
```
                                <span class="input-group-addon"><i class="glyphicon glyphicon-time"></i></span>
                            </EditItemTemplate>
                            <FooterTemplate>
                                <asp:TextBox ID="inEndTime" Width="100px" runat="server" />
                                <asp:RequiredFieldValidator ID="vEndTime" runat="server" ControlToValidate="inEndTime" Text="?" ValidationGroup="validaiton" />
                            </FooterTemplate>
                        </asp:TemplateField>
                        <asp:TemplateField>
                            <EditItemTemplate>
                                <asp:Button ID="ButtonUpdate" runat="server" CommandName="Update" Text="Update" />
                                <asp:Button ID="ButtonCancel" runat="server" CommandName="Cancel" Text="Cancel" />
                            </EditItemTemplate>
                            <ItemTemplate>
                                <asp:Button ID="ButtonEdit" runat="server" CommandName="Edit" Text="Edit" />
                            </ItemTemplate>
                            <FooterTemplate>
                            </FooterTemplate>
                        </asp:TemplateField>
                    </Columns>
                </asp:GridView>
                <br />
                &nbsp;&nbsp;&nbsp;&nbsp;
                <asp:Label ID="lblmsg" runat="server"></asp:Label>
            </div>
        </form>
    </div>
</asp:Content>

```

7.3.9 Admin-Practical Scheduling : C# Code



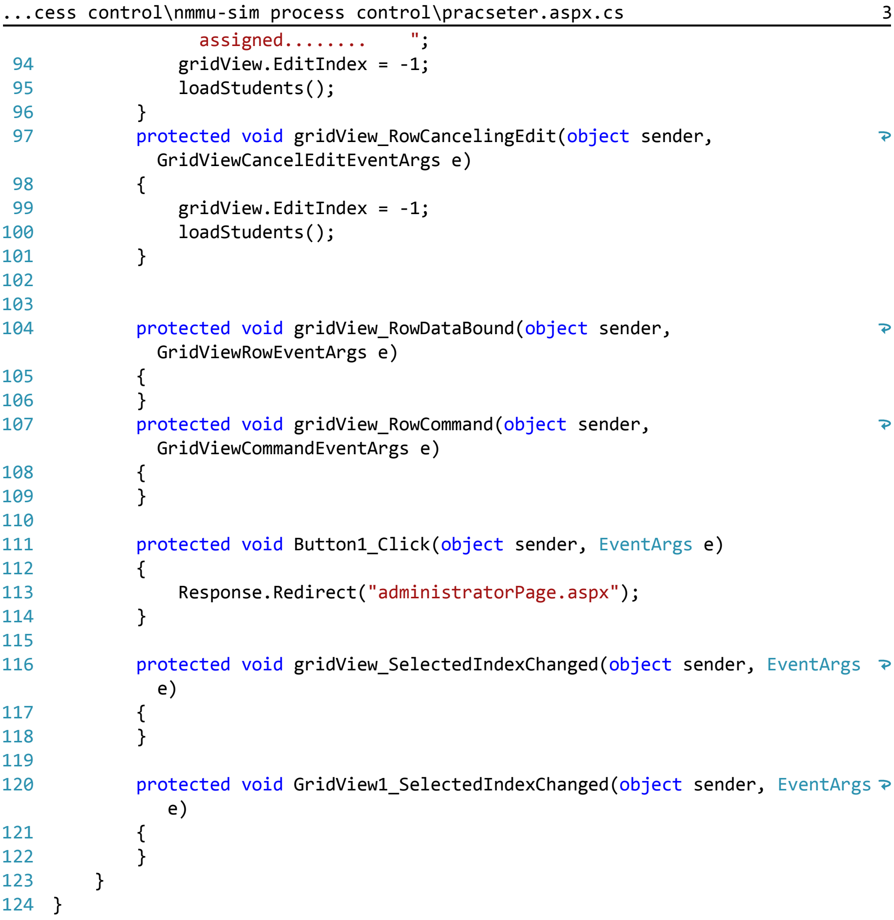
...cess control\nmmu-sim process control\pracseter.aspx.cs 1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Drawing;
using System.IO;
using System.Web.Security;

namespace nmmu_sim_process_control
{
    //Practical Scheduling for each student(PracName , Prac Date, Start time and End Time) are set and edited on this page
    public partial class pracseter : System.Web.UI.Page
    {
        private SqlConnection con = new SqlConnection("Data Source=sqlent-nc1;Initial Catalog=remoteLogin;Integrated Security =False;user id=remoteplc;Password=remoteplc2016;");

        SqlConnection conn = new SqlConnection("Data Source=sqlent-nc1;Initial Catalog=remoteLogin;Integrated Security =False;user id=remoteplc;Password=remoteplc2016;");
        string strCon = ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString;
        SqlDataAdapter SqlAda, SqlAdp;
        DataSet ds, dp;
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                loadStudents();
                //loadStudents() retrieves all the students registered on the database , allowing administrator to schedule the practicals for each student
            }
        }
        protected void loadStudents()
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("Select * from adminisd", con);
            SqlDataAdapter da = new SqlDataAdapter(cmd);
            DataSet ds = new DataSet();
            da.Fill(ds);
            int count = ds.Tables[0].Rows.Count;
            con.Close();
            if (ds.Tables[0].Rows.Count > 0)
            {
                gridView.DataSource = ds;
                gridView.DataBind();
            }
            else
            {
```



...cess control\nmmu-sim process control\pracseter.aspx.cs 2

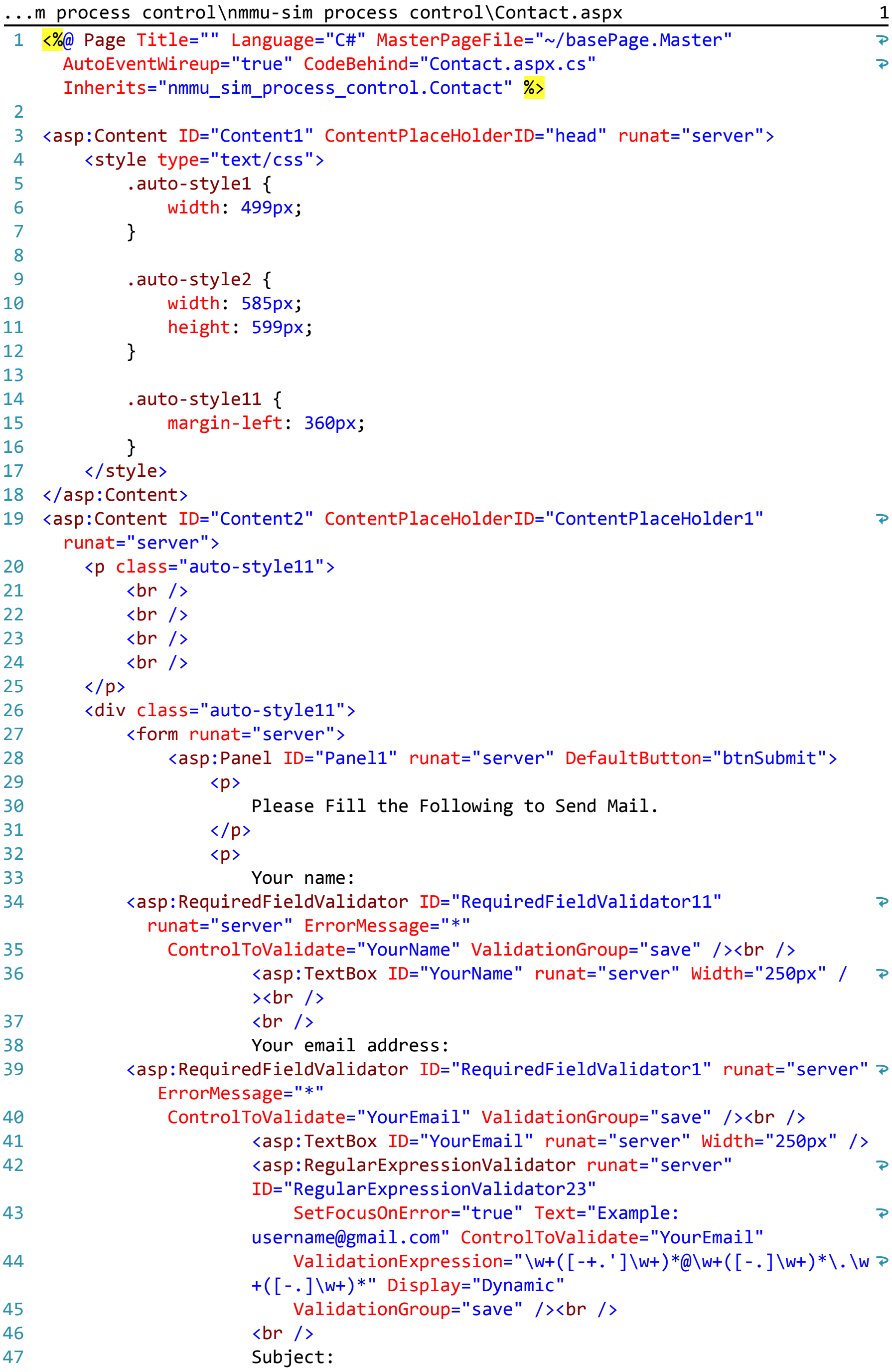
```
                ds.Tables[0].Rows.Add(ds.Tables[0].NewRow());
                gridView.DataSource = ds;
                gridView.DataBind();
                int columncount = gridView.Rows[0].Cells.Count;
                lblmsg.Text = " No data found !!!";
            }
        }
        protected void gridView_RowEditing(object sender,
          GridViewEditEventArgs e)
        {
            gridView.EditIndex = e.NewEditIndex;
            loadStudents();
        }

        protected void gridView_RowUpdating(object sender,
          GridViewUpdateEventArgs e)
        {
            string studentNo = gridView.DataKeys[e.RowIndex].Values
              ["studentNo"].ToString();
            TextBox StudentName = (TextBox)gridView.Rows
              [e.RowIndex].FindControl("txtStudentName");
            DropDownList PracName = (DropDownList)gridView.Rows
              [e.RowIndex].FindControl("PracName");
            IFormatProvider yyyy = null;
            TextBox PracDate = (TextBox)gridView.Rows[e.RowIndex].FindControl
              ("txtPracDate");
            TextBox StartTime = (TextBox)gridView.Rows[e.RowIndex].FindControl
              ("txtStartTime");
            TextBox EndTime = (TextBox)gridView.Rows[e.RowIndex].FindControl
              ("txtEndTime");

            using (SqlConnection con = new SqlConnection(strCon))
            {
                using (SqlCommand cmd = new SqlCommand("PracSet11"))
                {
                    using (SqlDataAdapter sda = new SqlDataAdapter())
                    {
                        cmd.CommandType = CommandType.StoredProcedure;
                        cmd.Parameters.AddWithValue("@studentNo", studentNo);
                        cmd.Parameters.AddWithValue("@PracName",
                    PracName.SelectedValue.ToString());
                        cmd.Parameters.AddWithValue
                    ("@PracDate",Convert.ToDateTime(PracDate.Text));
                        cmd.Parameters.AddWithValue("@startTime",
                    Convert.ToDateTime(StartTime.Text));
                        cmd.Parameters.AddWithValue("@ENDtIME", EndTime.Text);
                        cmd.Connection = con;
                        con.Open();
                        cmd.ExecuteNonQuery();
                        con.Close();
                    }
                }
            }
            lblmsg.BackColor = Color.Blue;
            lblmsg.ForeColor = Color.White;
            lblmsg.Text = StudentName.Text + "        Practical successfully
```



...cess control\nmmu-sim process control\pracseter.aspx.cs 3

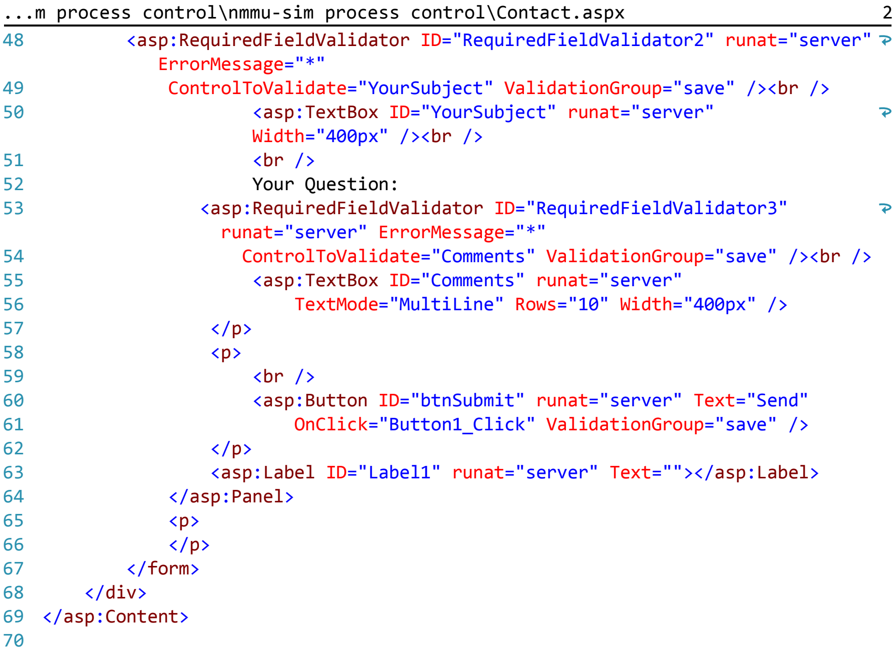
```
              assigned........    ";
            gridView.EditIndex = -1;
            loadStudents();
        }
        protected void gridView_RowCancelingEdit(object sender, GridViewCancelEditEventArgs e)
        {
            gridView.EditIndex = -1;
            loadStudents();
        }


        protected void gridView_RowDataBound(object sender, GridViewRowEventArgs e)
        {
        }
        protected void gridView_RowCommand(object sender, GridViewCommandEventArgs e)
        {
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Redirect("administratorPage.aspx");
        }

        protected void gridView_SelectedIndexChanged(object sender, EventArgs e)
        {
        }

        protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
        {
        }
    }
}
```

7.3.10 Contact Us Page : HTML Mark-Up



...m process control\nmmu-sim process control\Contact.aspx 1

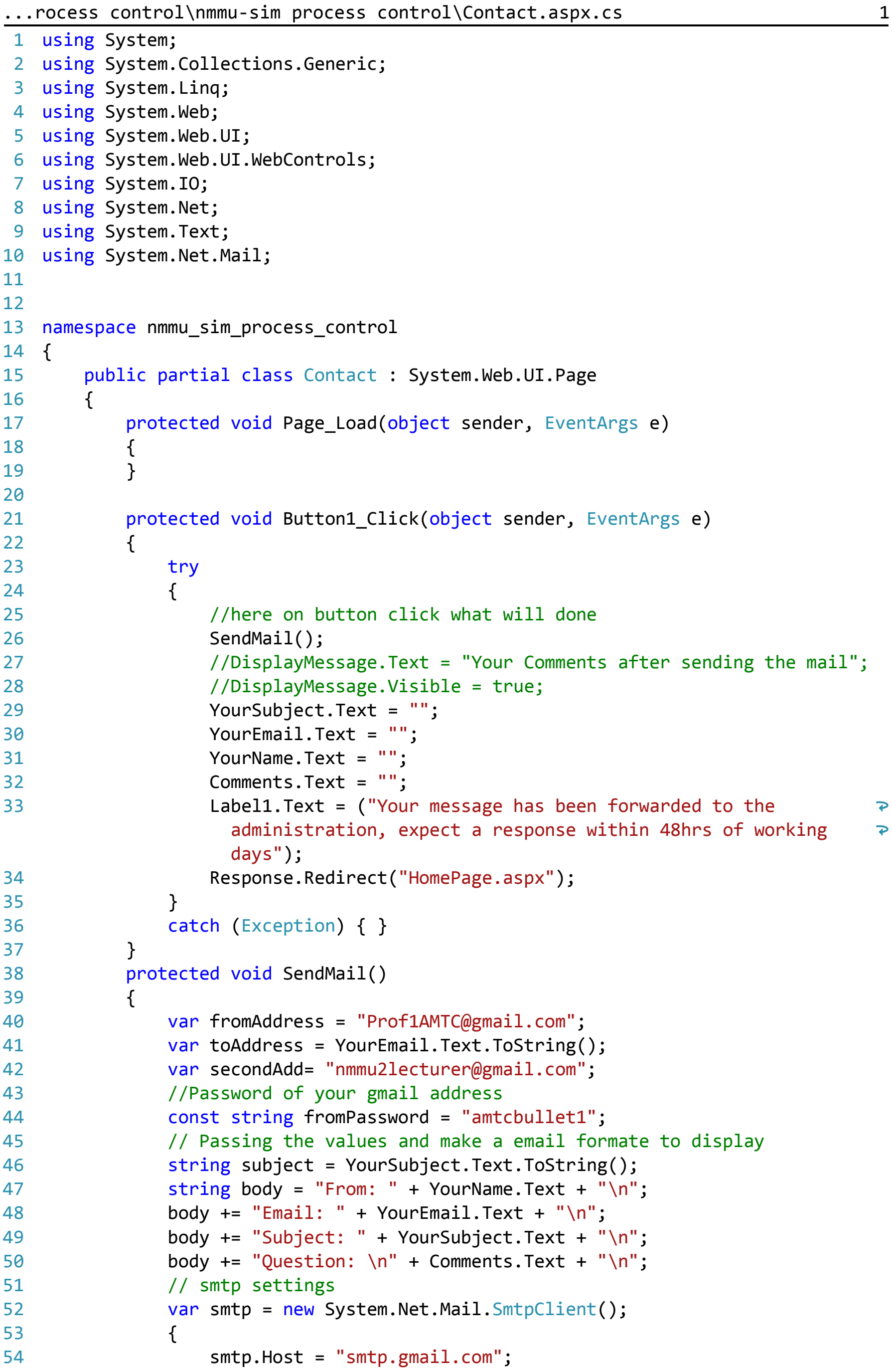
```
<%@ Page Title="" Language="C#" MasterPageFile="~/basePage.Master" AutoEventWireup="true" CodeBehind="Contact.aspx.cs" Inherits="nmmu_sim_process_control.Contact" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
    <style type="text/css">
        .auto-style1 {
            width: 499px;
        }

        .auto-style2 {
            width: 585px;
            height: 599px;
        }

        .auto-style11 {
            margin-left: 360px;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <p class="auto-style11">
        <br />
        <br />
        <br />
        <br />
    </p>
    <div class="auto-style11">
        <form runat="server">
            <asp:Panel ID="Panel1" runat="server" DefaultButton="btnSubmit">
                <p>
                    Please Fill the Following to Send Mail.
                </p>
                <p>
                    Your name:
        <asp:RequiredFieldValidator ID="RequiredFieldValidator11" runat="server" ErrorMessage="*"
            ControlToValidate="YourName" ValidationGroup="save" /><br />
                    <asp:TextBox ID="YourName" runat="server" Width="250px" /><br />
                    <br />
                    Your email address:
        <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ErrorMessage="*"
            ControlToValidate="YourEmail" ValidationGroup="save" /><br />
                    <asp:TextBox ID="YourEmail" runat="server" Width="250px" />
                    <asp:RegularExpressionValidator runat="server" ID="RegularExpressionValidator23"
                        SetFocusOnError="true" Text="Example: username@gmail.com" ControlToValidate="YourEmail"
                        ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*" Display="Dynamic"
                        ValidationGroup="save" /><br />
                    <br />
                    Subject:
```



...m process control\nmmu-sim process control\Contact.aspx 2

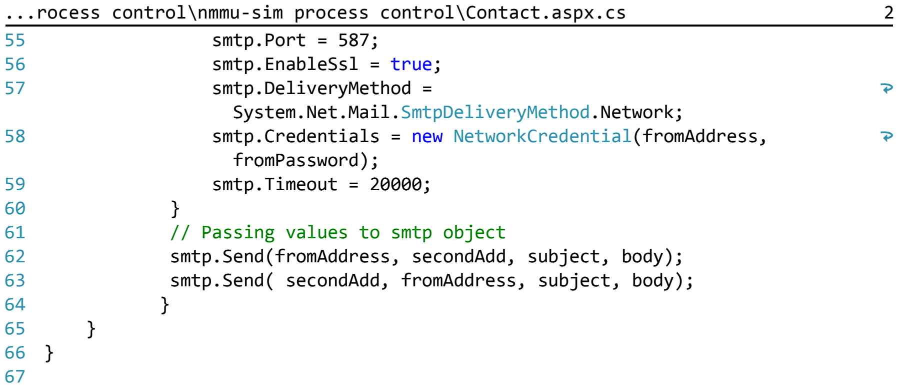
```
        <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ErrorMessage="*"
            ControlToValidate="YourSubject" ValidationGroup="save" /><br />
                    <asp:TextBox ID="YourSubject" runat="server" Width="400px" /><br />
                    <br />
                    Your Question:
               <asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server" ErrorMessage="*"
                   ControlToValidate="Comments" ValidationGroup="save" /><br />
                    <asp:TextBox ID="Comments" runat="server"
                        TextMode="MultiLine" Rows="10" Width="400px" />
                </p>
                <p>
                    <br />
                    <asp:Button ID="btnSubmit" runat="server" Text="Send"
                        OnClick="Button1_Click" ValidationGroup="save" />
                </p>
                <asp:Label ID="Label1" runat="server" Text=""></asp:Label>
            </asp:Panel>
            <p>
            </p>
        </form>
    </div>
</asp:Content>

```

7.3.11 Contact Us Page : C# Code



...rocess control\nmmu-sim process control\Contact.aspx.cs 1

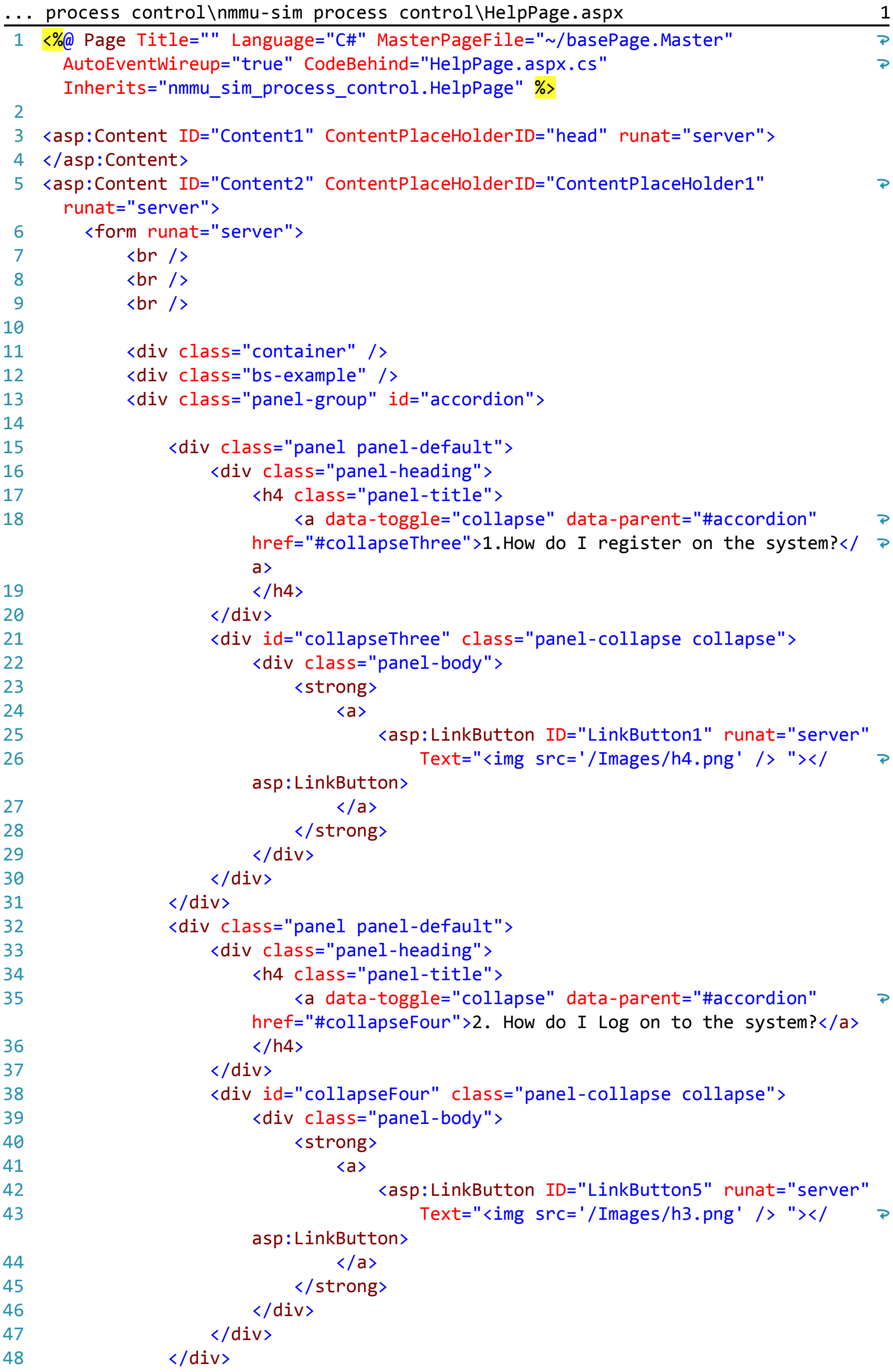
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.IO;
using System.Net;
using System.Text;
using System.Net.Mail;


namespace nmmu_sim_process_control
{
    public partial class Contact : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            try
            {
                //here on button click what will done
                SendMail();
                //DisplayMessage.Text = "Your Comments after sending the mail";
                //DisplayMessage.Visible = true;
                YourSubject.Text = "";
                YourEmail.Text = "";
                YourName.Text = "";
                Comments.Text = "";
                Label1.Text = ("Your message has been forwarded to the administration, expect a response within 48hrs of working days");
                Response.Redirect("HomePage.aspx");
            }
            catch (Exception) { }
        }
        protected void SendMail()
        {
            var fromAddress = "Prof1AMTC@gmail.com";
            var toAddress = YourEmail.Text.ToString();
            var secondAdd= "nmmu2lecturer@gmail.com";
            //Password of your gmail address
            const string fromPassword = "amtcbullet1";
            // Passing the values and make a email formate to display
            string subject = YourSubject.Text.ToString();
            string body = "From: " + YourName.Text + "\n";
            body += "Email: " + YourEmail.Text + "\n";
            body += "Subject: " + YourSubject.Text + "\n";
            body += "Question: \n" + Comments.Text + "\n";
            // smtp settings
            var smtp = new System.Net.Mail.SmtpClient();
            {
                smtp.Host = "smtp.gmail.com";
```



...rocess control\nmmu-sim process control\Contact.aspx.cs 2

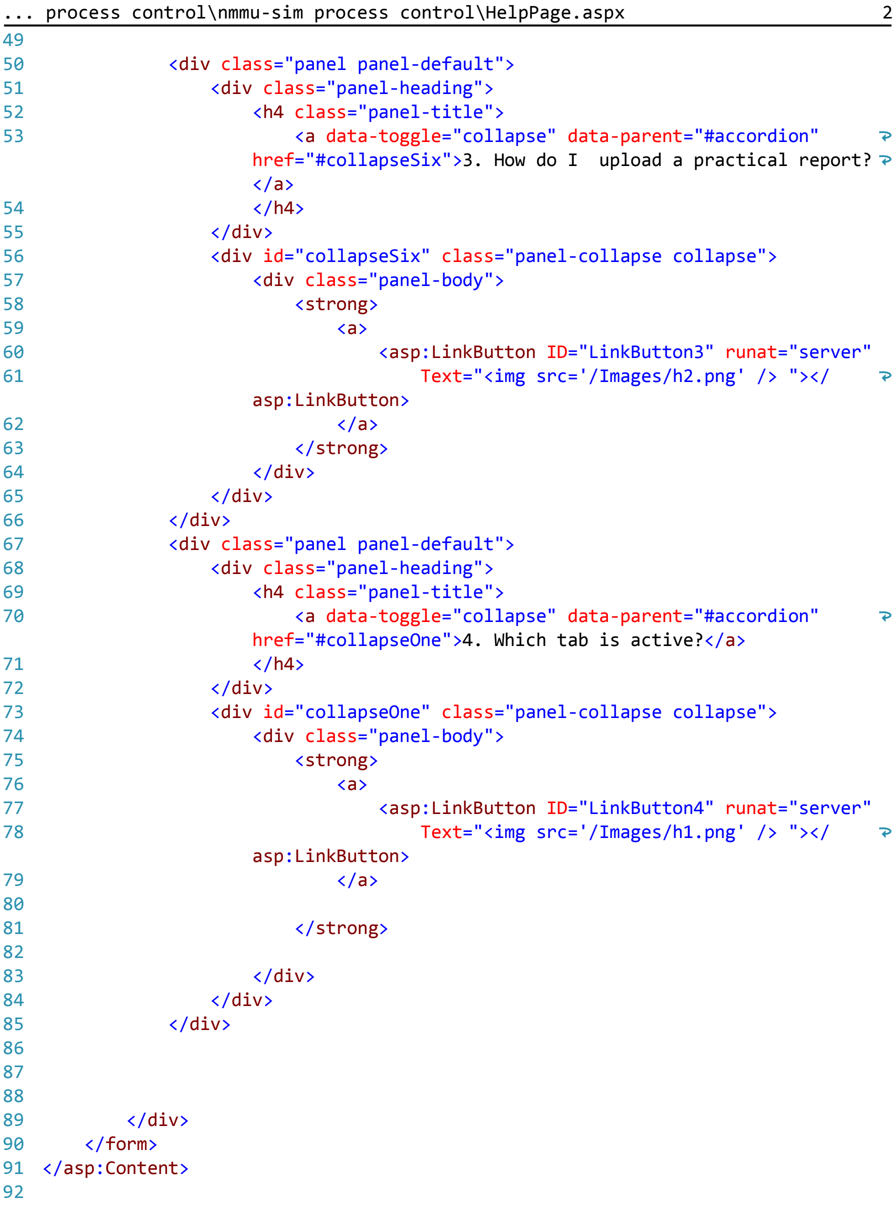
```
                smtp.Port = 587;
                smtp.EnableSsl = true;
                smtp.DeliveryMethod =
                  System.Net.Mail.SmtpDeliveryMethod.Network;
                smtp.Credentials = new NetworkCredential(fromAddress,
                  fromPassword);
                smtp.Timeout = 20000;
            }
            // Passing values to smtp object
            smtp.Send(fromAddress, secondAdd, subject, body);
            smtp.Send( secondAdd, fromAddress, subject, body);
           }
    }
}
```

7.3.12 Help Page : HTML Mark-Up



... process control\nmmu-sim process control\HelpPage.aspx 1

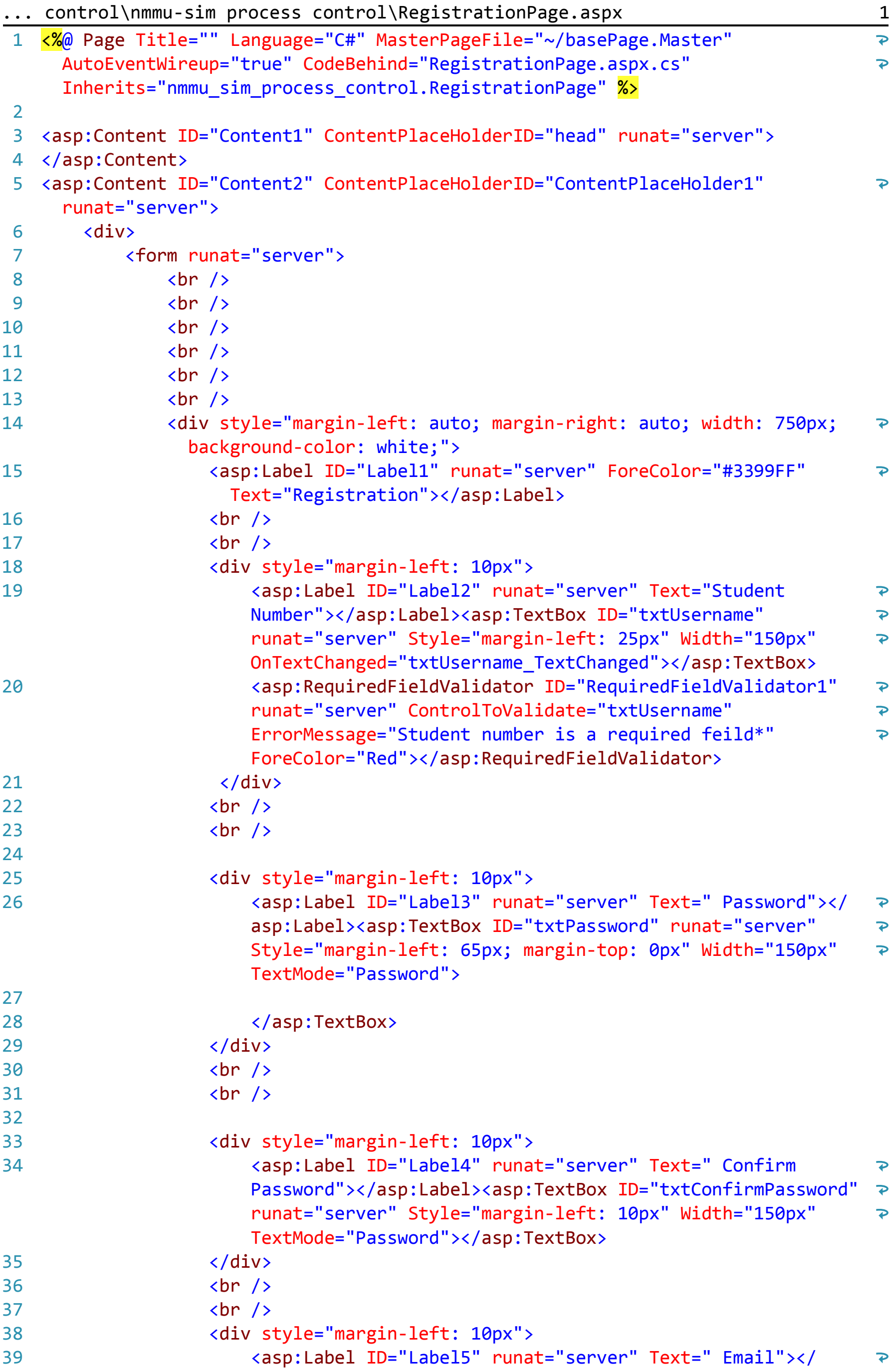
```
<%@ Page Title="" Language="C#" MasterPageFile="~/basePage.Master" AutoEventWireup="true" CodeBehind="HelpPage.aspx.cs" Inherits="nmmu_sim_process_control.HelpPage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <form runat="server">
        <br />
        <br />
        <br />

        <div class="container" />
        <div class="bs-example" />
        <div class="panel-group" id="accordion">

            <div class="panel panel-default">
                <div class="panel-heading">
                    <h4 class="panel-title">
                        <a data-toggle="collapse" data-parent="#accordion" href="#collapseThree">1.How do I register on the system?</a>
                    </h4>
                </div>
                <div id="collapseThree" class="panel-collapse collapse">
                    <div class="panel-body">
                        <strong>
                            <a>
                                <asp:LinkButton ID="LinkButton1" runat="server"
                                    Text="<img src='/Images/h4.png' /> "></asp:LinkButton>
                            </a>
                        </strong>
                    </div>
                </div>
            </div>
            <div class="panel panel-default">
                <div class="panel-heading">
                    <h4 class="panel-title">
                        <a data-toggle="collapse" data-parent="#accordion" href="#collapseFour">2. How do I Log on to the system?</a>
                    </h4>
                </div>
                <div id="collapseFour" class="panel-collapse collapse">
                    <div class="panel-body">
                        <strong>
                            <a>
                                <asp:LinkButton ID="LinkButton5" runat="server"
                                    Text="<img src='/Images/h3.png' /> "></asp:LinkButton>
                            </a>
                        </strong>
                    </div>
                </div>
            </div>
```



... process control\nmmu-sim process control\HelpPage.aspx 2

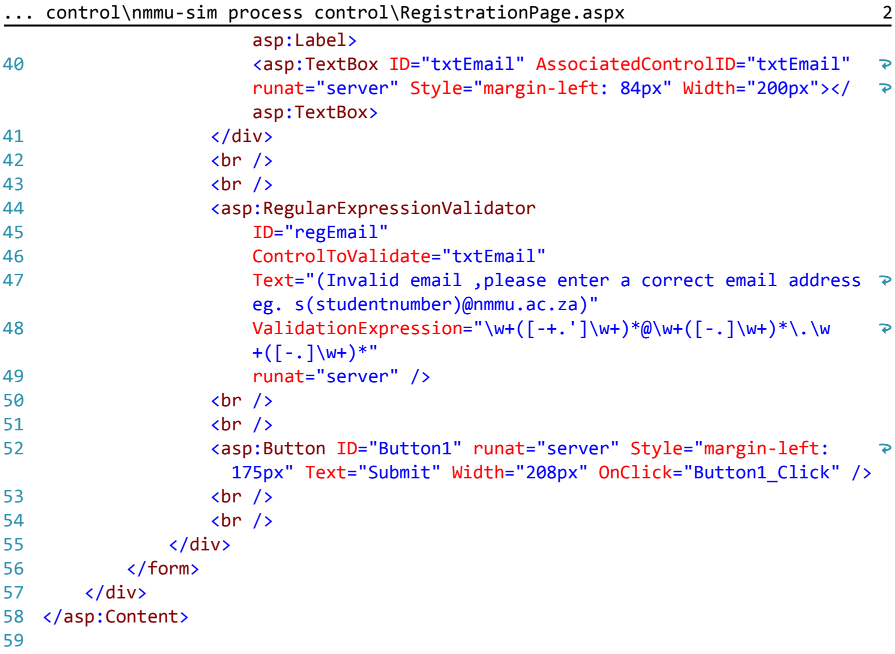
```

            <div class="panel panel-default">
                <div class="panel-heading">
                    <h4 class="panel-title">
                        <a data-toggle="collapse" data-parent="#accordion" href="#collapseSix">3. How do I  upload a practical report?</a>
                    </h4>
                </div>
                <div id="collapseSix" class="panel-collapse collapse">
                    <div class="panel-body">
                        <strong>
                            <a>
                                <asp:LinkButton ID="LinkButton3" runat="server"
                                    Text="<img src='/Images/h2.png' /> "></asp:LinkButton>
                            </a>
                        </strong>
                    </div>
                </div>
            </div>
            <div class="panel panel-default">
                <div class="panel-heading">
                    <h4 class="panel-title">
                        <a data-toggle="collapse" data-parent="#accordion" href="#collapseOne">4. Which tab is active?</a>
                    </h4>
                </div>
                <div id="collapseOne" class="panel-collapse collapse">
                    <div class="panel-body">
                        <strong>
                            <a>
                                <asp:LinkButton ID="LinkButton4" runat="server"
                                    Text="<img src='/Images/h1.png' /> "></asp:LinkButton>
                            </a>

                        </strong>

                    </div>
                </div>
            </div>



        </div>
    </form>
</asp:Content>

```

7.3.13 Registration Page : HTML Mark-Up



... control\nmmu-sim process control\RegistrationPage.aspx 1

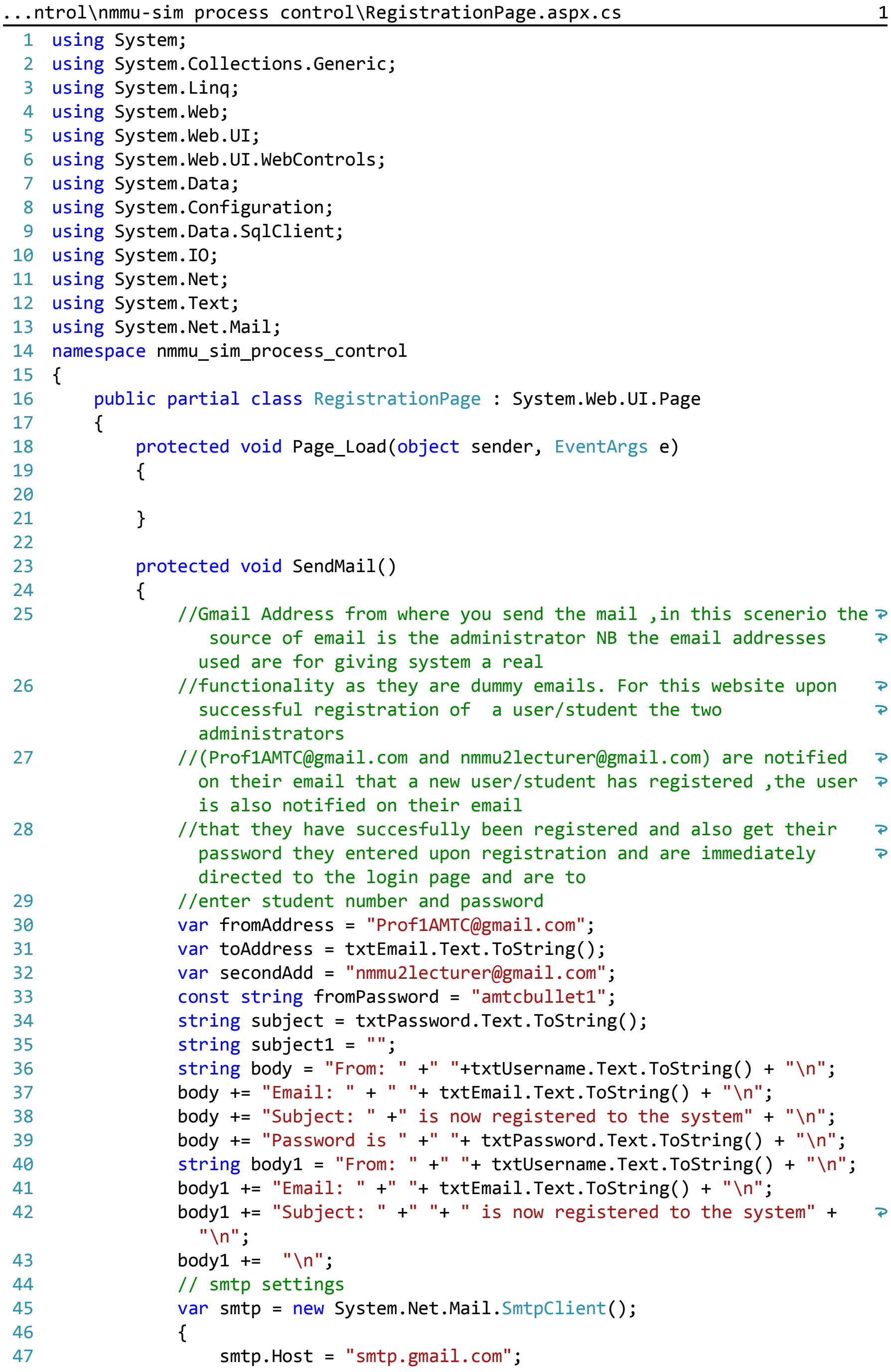
```
<%@ Page Title="" Language="C#" MasterPageFile="~/basePage.Master" AutoEventWireup="true" CodeBehind="RegistrationPage.aspx.cs" Inherits="nmmu_sim_process_control.RegistrationPage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <div>
        <form runat="server">
            <br />
            <br />
            <br />
            <br />
            <br />
            <br />
            <div style="margin-left: auto; margin-right: auto; width: 750px; background-color: white;">
                <asp:Label ID="Label1" runat="server" ForeColor="#3399FF" Text="Registration"></asp:Label>
                <br />
                <br />
                <div style="margin-left: 10px">
                    <asp:Label ID="Label2" runat="server" Text="Student Number"></asp:Label><asp:TextBox ID="txtUsername" runat="server" Style="margin-left: 25px" Width="150px" OnTextChanged="txtUsername_TextChanged"></asp:TextBox>
                    <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ControlToValidate="txtUsername" ErrorMessage="Student number is a required feild*" ForeColor="Red"></asp:RequiredFieldValidator>
                 </div>
                <br />
                <br />

                <div style="margin-left: 10px">
                    <asp:Label ID="Label3" runat="server" Text=" Password"></asp:Label><asp:TextBox ID="txtPassword" runat="server" Style="margin-left: 65px; margin-top: 0px" Width="150px" TextMode="Password">

                    </asp:TextBox>
                </div>
                <br />
                <br />

                <div style="margin-left: 10px">
                    <asp:Label ID="Label4" runat="server" Text=" Confirm Password"></asp:Label><asp:TextBox ID="txtConfirmPassword" runat="server" Style="margin-left: 10px" Width="150px" TextMode="Password"></asp:TextBox>
                </div>
                <br />
                <br />
                <div style="margin-left: 10px">
                    <asp:Label ID="Label5" runat="server" Text=" Email"></
```



... control\nmmu-sim process control\RegistrationPage.aspx 2

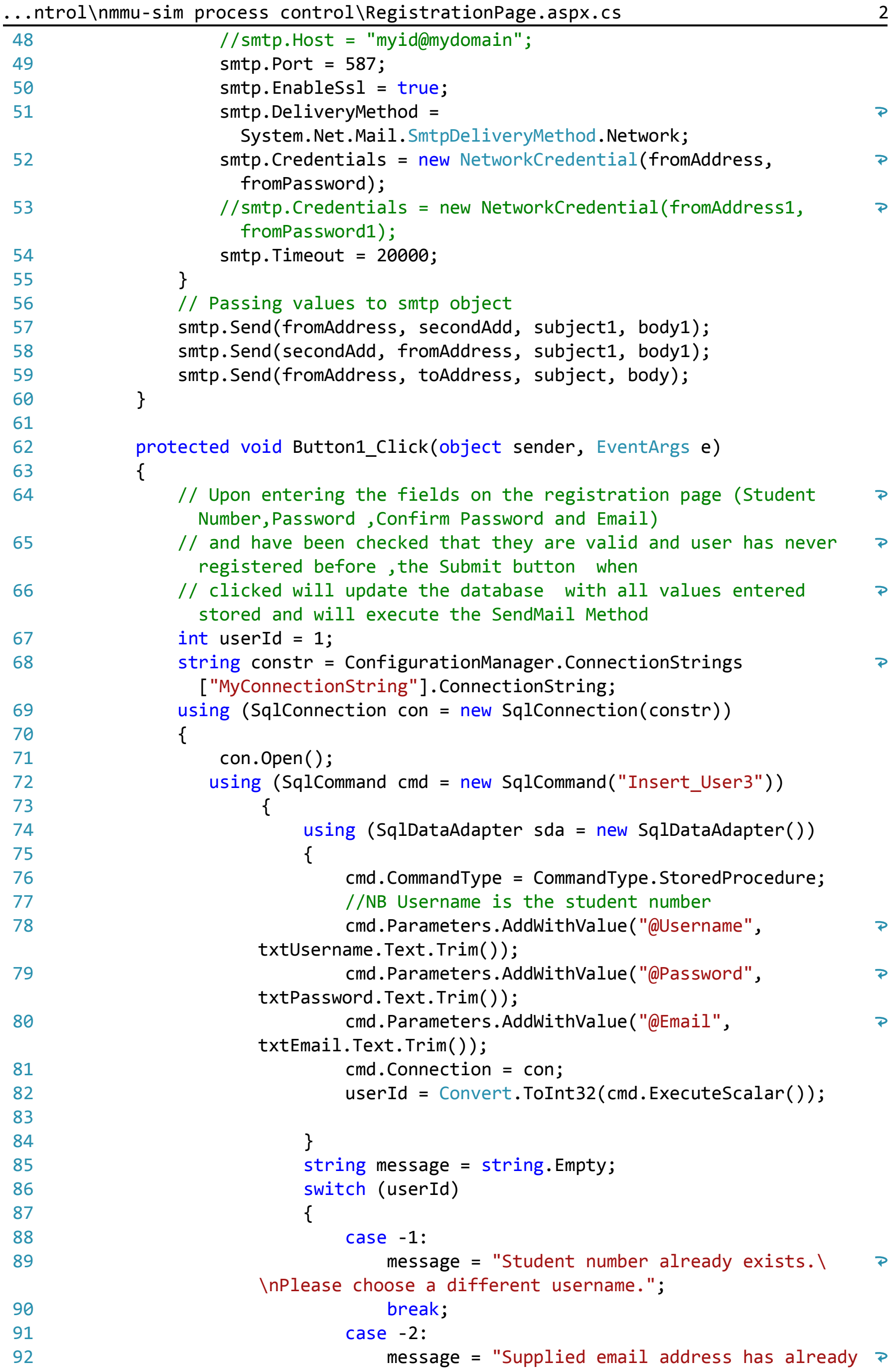
```
                    asp:Label>
                    <asp:TextBox ID="txtEmail" AssociatedControlID="txtEmail" runat="server" Style="margin-left: 84px" Width="200px"></asp:TextBox>
                </div>
                <br />
                <br />
                <asp:RegularExpressionValidator
                    ID="regEmail"
                    ControlToValidate="txtEmail"
                    Text="(Invalid email ,please enter a correct email address eg. s(studentnumber)@nmmu.ac.za)"
                    ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*"
                    runat="server" />
                <br />
                <br />
                <asp:Button ID="Button1" runat="server" Style="margin-left: 175px" Text="Submit" Width="208px" OnClick="Button1_Click" />
                <br />
                <br />
            </div>
        </form>
    </div>
</asp:Content>

```

7.3.14 Registration Page : C# Code



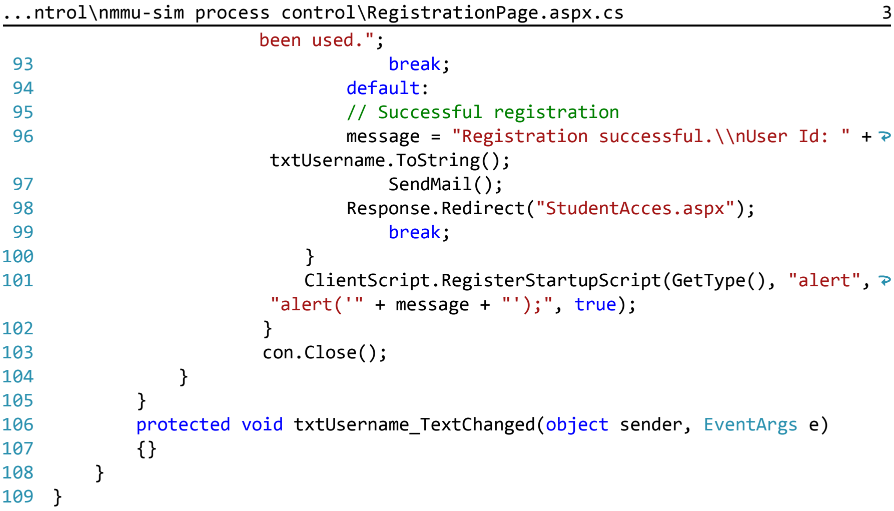
...ntrol\nmmu-sim process control\RegistrationPage.aspx.cs 1

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Configuration;
using System.Data.SqlClient;
using System.IO;
using System.Net;
using System.Text;
using System.Net.Mail;
namespace nmmu_sim_process_control
{
    public partial class RegistrationPage : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void SendMail()
        {
            //Gmail Address from where you send the mail ,in this scenerio the source of email is the administrator NB the email addresses used are for giving system a real
            //functionality as they are dummy emails. For this website upon successful registration of  a user/student the two administrators
            //(Prof1AMTC@gmail.com and nmmu2lecturer@gmail.com) are notified on their email that a new user/student has registered ,the user is also notified on their email
            //that they have succesfully been registered and also get their password they entered upon registration and are immediately directed to the login page and are to
            //enter student number and password
            var fromAddress = "Prof1AMTC@gmail.com";
            var toAddress = txtEmail.Text.ToString();
            var secondAdd = "nmmu2lecturer@gmail.com";
            const string fromPassword = "amtcbullet1";
            string subject = txtPassword.Text.ToString();
            string subject1 = "";
            string body = "From: " +" "+txtUsername.Text.ToString() + "\n";
            body += "Email: " + " "+ txtEmail.Text.ToString() + "\n";
            body += "Subject: " +" is now registered to the system" + "\n";
            body += "Password is " +" "+ txtPassword.Text.ToString() + "\n";
            string body1 = "From: " +" "+ txtUsername.Text.ToString() + "\n";
            body1 += "Email: " +" "+ txtEmail.Text.ToString() + "\n";
            body1 += "Subject: " +" "+ " is now registered to the system" + "\n";
            body1 +=  "\n";
            // smtp settings
            var smtp = new System.Net.Mail.SmtpClient();
            {
                smtp.Host = "smtp.gmail.com";
```



...ntrol\nmmu-sim process control\RegistrationPage.aspx.cs 2

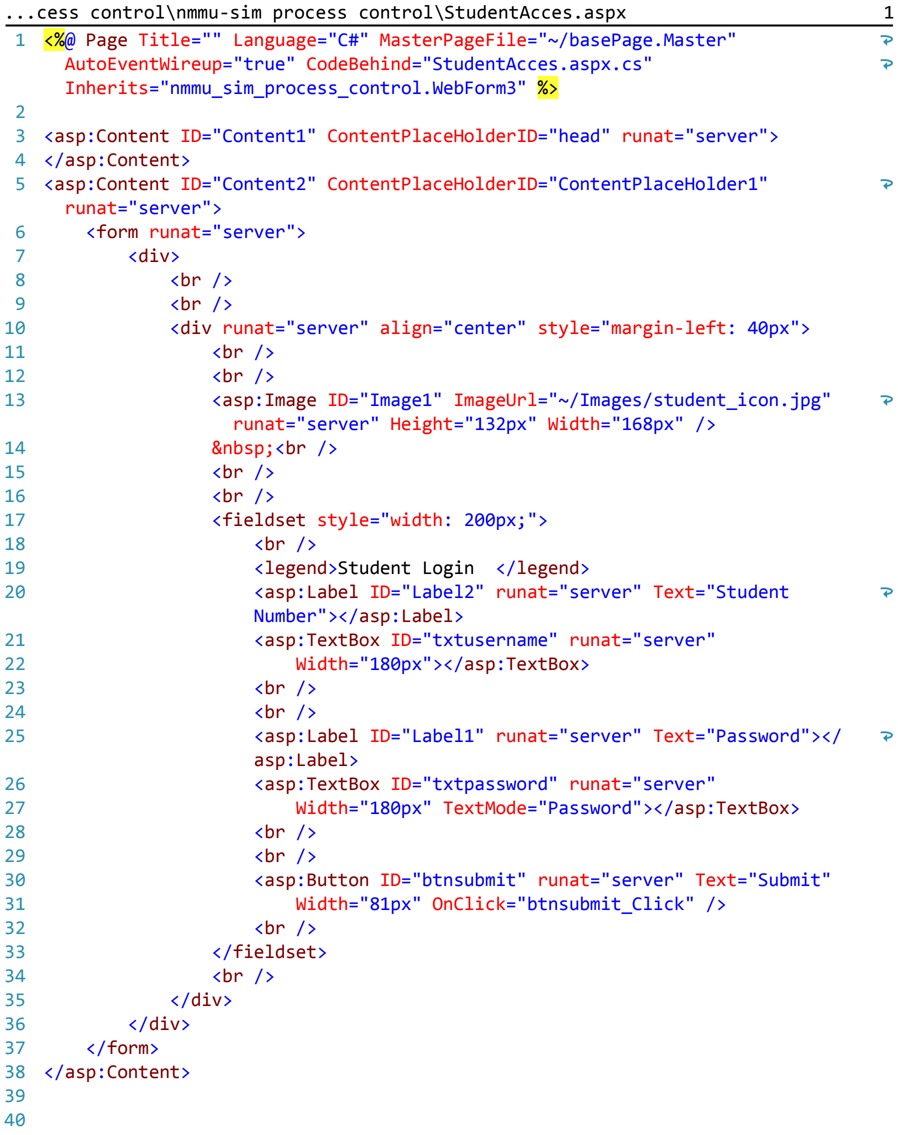
```
                //smtp.Host = "myid@mydomain";
                smtp.Port = 587;
                smtp.EnableSsl = true;
                smtp.DeliveryMethod = System.Net.Mail.SmtpDeliveryMethod.Network;
                smtp.Credentials = new NetworkCredential(fromAddress, fromPassword);
                //smtp.Credentials = new NetworkCredential(fromAddress1, fromPassword1);
                smtp.Timeout = 20000;
            }
            // Passing values to smtp object
            smtp.Send(fromAddress, secondAdd, subject1, body1);
            smtp.Send(secondAdd, fromAddress, subject1, body1);
            smtp.Send(fromAddress, toAddress, subject, body);
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            // Upon entering the fields on the registration page (Student Number,Password ,Confirm Password and Email)
            // and have been checked that they are valid and user has never registered before ,the Submit button  when
            // clicked will update the database  with all values entered stored and will execute the SendMail Method
            int userId = 1;
            string constr = ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString;
            using (SqlConnection con = new SqlConnection(constr))
            {
                con.Open();
               using (SqlCommand cmd = new SqlCommand("Insert_User3"))
                    {
                        using (SqlDataAdapter sda = new SqlDataAdapter())
                        {
                            cmd.CommandType = CommandType.StoredProcedure;
                            //NB Username is the student number
                            cmd.Parameters.AddWithValue("@Username", txtUsername.Text.Trim());
                            cmd.Parameters.AddWithValue("@Password", txtPassword.Text.Trim());
                            cmd.Parameters.AddWithValue("@Email", txtEmail.Text.Trim());
                            cmd.Connection = con;
                            userId = Convert.ToInt32(cmd.ExecuteScalar());

                        }
                        string message = string.Empty;
                        switch (userId)
                        {
                            case -1:
                                message = "Student number already exists.\\nPlease choose a different username.";
                                break;
                            case -2:
                                message = "Supplied email address has already
```



...ntrol\nmmu-sim process control\RegistrationPage.aspx.cs 3

```
                    been used.";
                                break;
                            default:
                            // Successful registration
                            message = "Registration successful.\\nUser Id: " +
                     txtUsername.ToString();
                                SendMail();
                            Response.Redirect("StudentAcces.aspx");
                                break;
                        }
                        ClientScript.RegisterStartupScript(GetType(), "alert",
                     "alert('" + message + "');", true);
                    }
                    con.Close();
            }
        }
        protected void txtUsername_TextChanged(object sender, EventArgs e)
        {}
    }
}
```

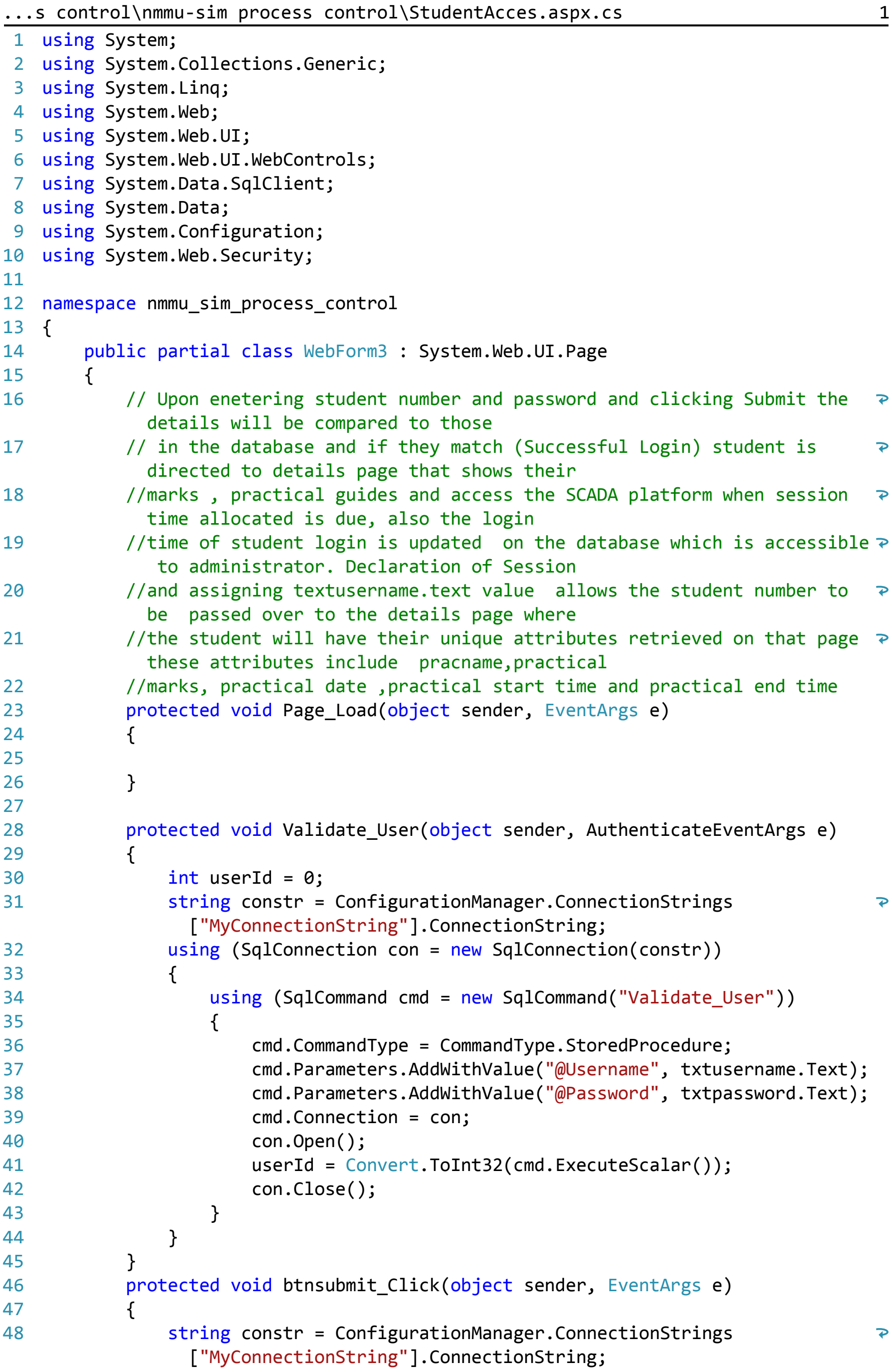
7.3.15 Student Login : HTML Mark-Up



...cess control\nmmu-sim process control\StudentAcces.aspx 1

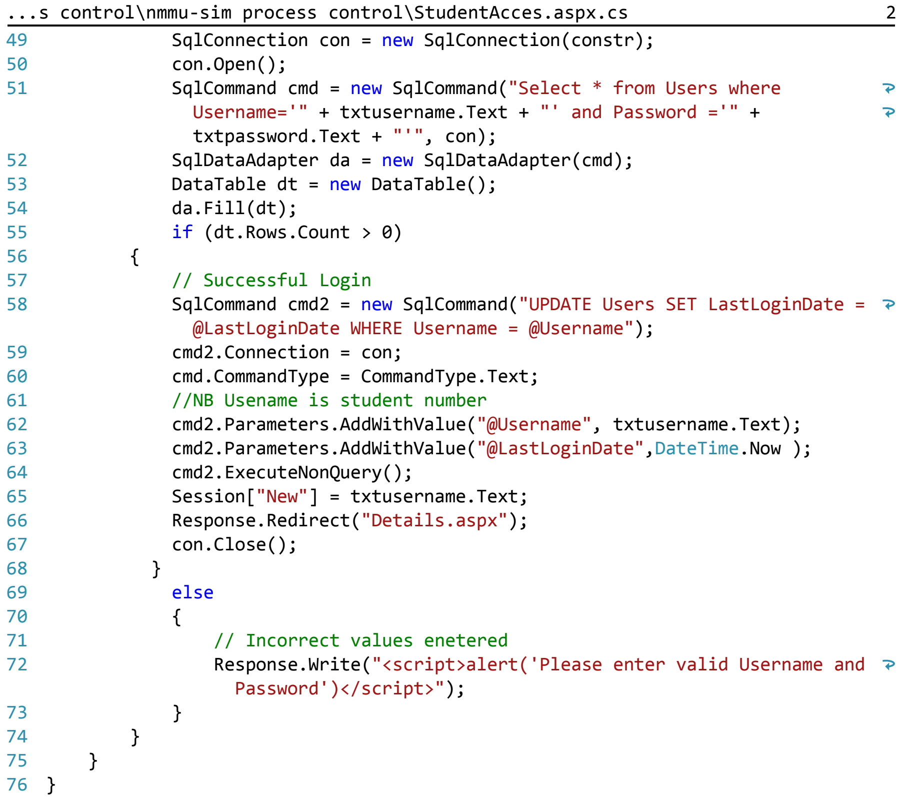
```
<%@ Page Title="" Language="C#" MasterPageFile="~/basePage.Master" AutoEventWireup="true" CodeBehind="StudentAcces.aspx.cs" Inherits="nmmu_sim_process_control.WebForm3" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <form runat="server">
        <div>
            <br />
            <br />
            <div runat="server" align="center" style="margin-left: 40px">
                <br />
                <br />
                <asp:Image ID="Image1" ImageUrl="~/Images/student_icon.jpg" runat="server" Height="132px" Width="168px" />
                &nbsp;<br />
                <br />
                <br />
                <fieldset style="width: 200px;">
                    <br />
                    <legend>Student Login  </legend>
                    <asp:Label ID="Label2" runat="server" Text="Student Number"></asp:Label>
                    <asp:TextBox ID="txtusername" runat="server"
                        Width="180px"></asp:TextBox>
                    <br />
                    <br />
                    <asp:Label ID="Label1" runat="server" Text="Password"></asp:Label>
                    <asp:TextBox ID="txtpassword" runat="server"
                        Width="180px" TextMode="Password"></asp:TextBox>
                    <br />
                    <br />
                    <asp:Button ID="btnsubmit" runat="server" Text="Submit"
                        Width="81px" OnClick="btnsubmit_Click" />
                    <br />
                </fieldset>
                <br />
            </div>
        </div>
    </form>
</asp:Content>


```

7.3.16 Student Login : C# Code



...s control\nmmu-sim process control\StudentAcces.aspx.cs 1

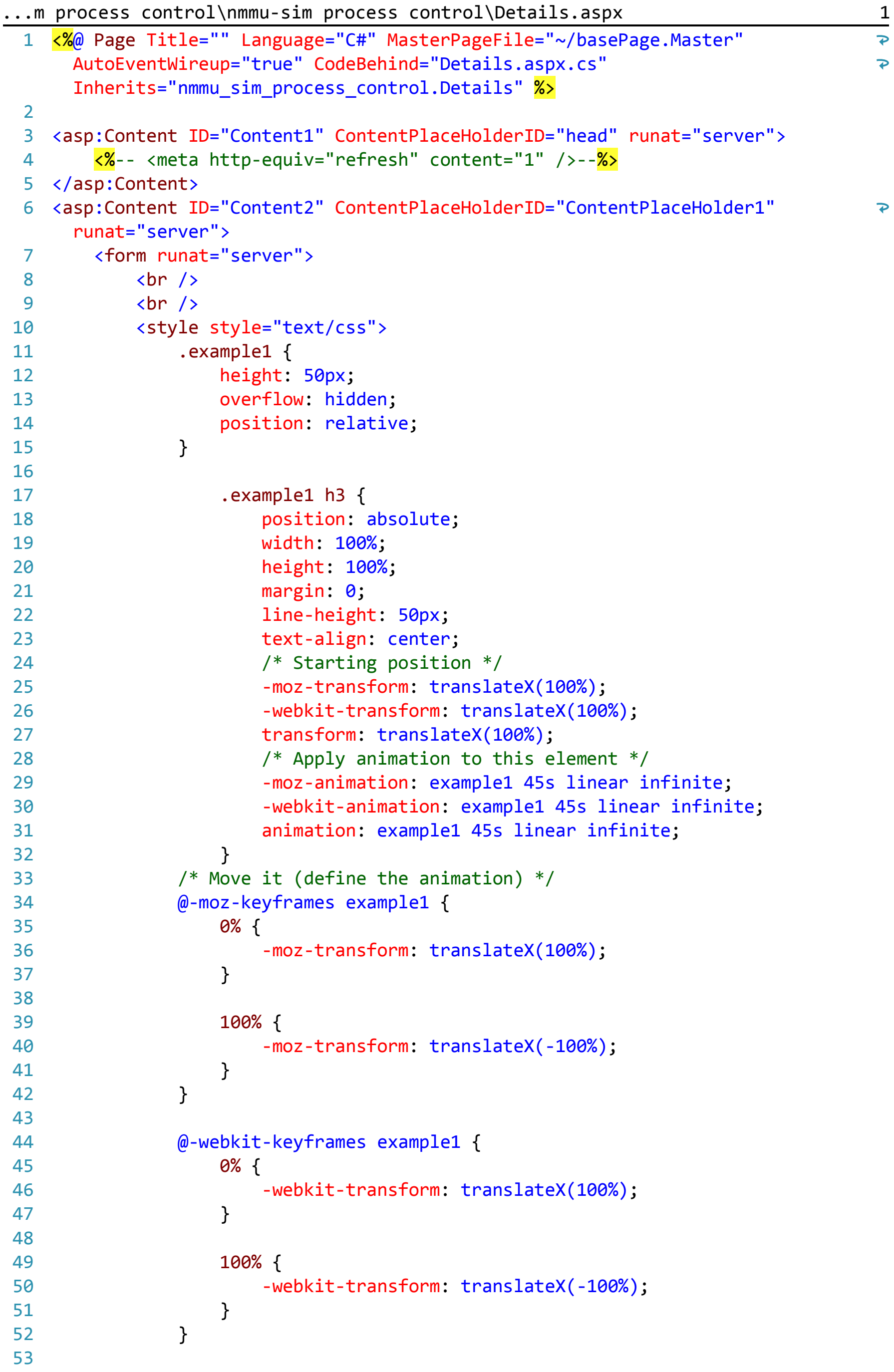
```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;
using System.Configuration;
using System.Web.Security;

namespace nmmu_sim_process_control
{
    public partial class WebForm3 : System.Web.UI.Page
    {
        // Upon enetering student number and password and clicking Submit the details will be compared to those
        // in the database and if they match (Successful Login) student is directed to details page that shows their
        //marks , practical guides and access the SCADA platform when session time allocated is due, also the login
        //time of student login is updated  on the database which is accessible to administrator. Declaration of Session
        //and assigning textusername.text value  allows the student number to be  passed over to the details page where
        //the student will have their unique attributes retrieved on that page these attributes include  pracname,practical
        //marks, practical date ,practical start time and practical end time
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Validate_User(object sender, AuthenticateEventArgs e)
        {
            int userId = 0;
            string constr = ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString;
            using (SqlConnection con = new SqlConnection(constr))
            {
                using (SqlCommand cmd = new SqlCommand("Validate_User"))
                {
                    cmd.CommandType = CommandType.StoredProcedure;
                    cmd.Parameters.AddWithValue("@Username", txtusername.Text);
                    cmd.Parameters.AddWithValue("@Password", txtpassword.Text);
                    cmd.Connection = con;
                    con.Open();
                    userId = Convert.ToInt32(cmd.ExecuteScalar());
                    con.Close();
                }
            }
        }
        protected void btnsubmit_Click(object sender, EventArgs e)
        {
            string constr = ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString;
```



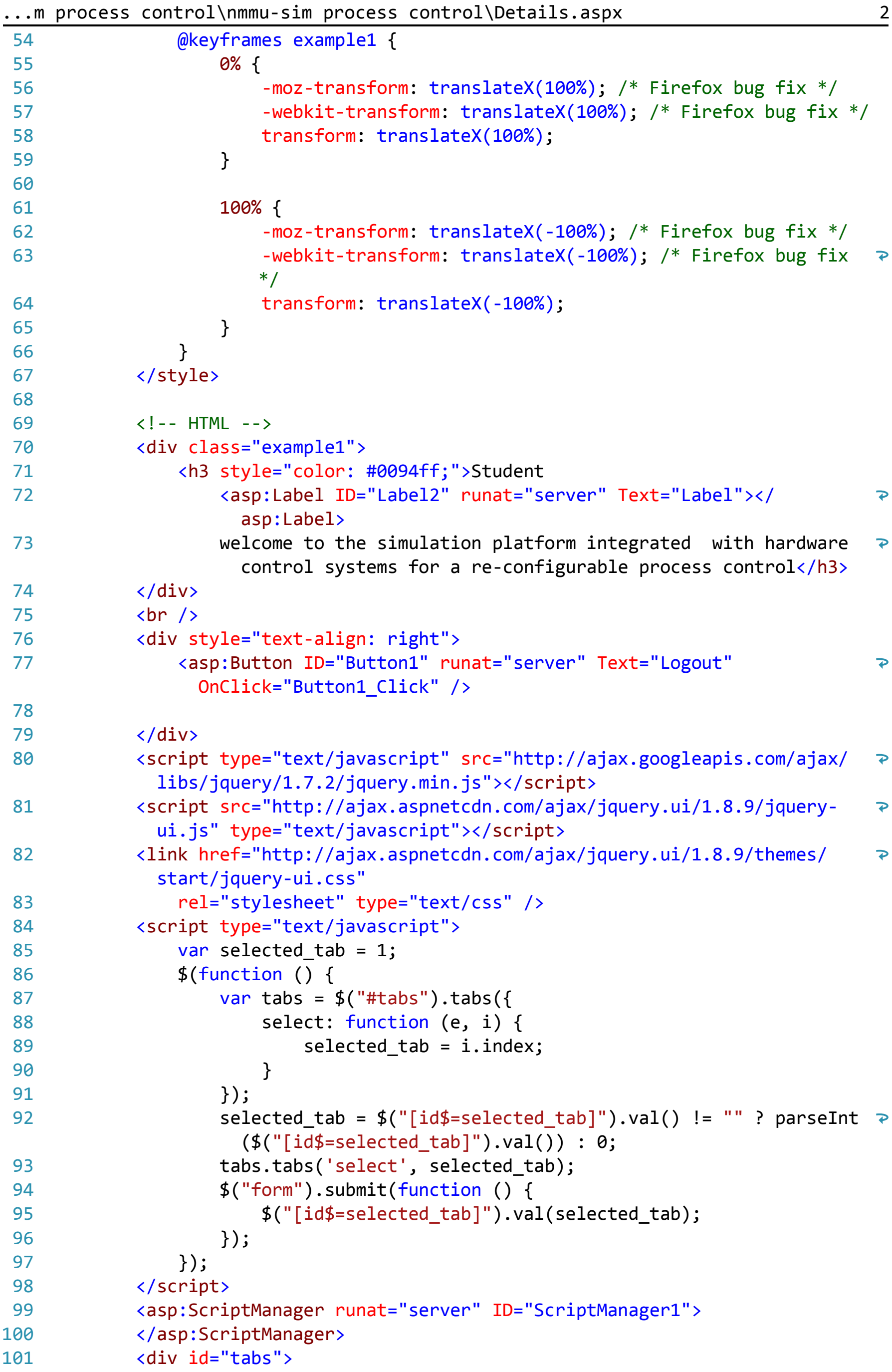
...s control\nmmu-sim process control\StudentAcces.aspx.cs 2

```
            SqlConnection con = new SqlConnection(constr);
            con.Open();
            SqlCommand cmd = new SqlCommand("Select * from Users where Username='" + txtusername.Text + "' and Password ='" + txtpassword.Text + "'", con);
            SqlDataAdapter da = new SqlDataAdapter(cmd);
            DataTable dt = new DataTable();
            da.Fill(dt);
            if (dt.Rows.Count > 0)
        {
            // Successful Login
            SqlCommand cmd2 = new SqlCommand("UPDATE Users SET LastLoginDate = @LastLoginDate WHERE Username = @Username");
            cmd2.Connection = con;
            cmd.CommandType = CommandType.Text;
            //NB Usename is student number
            cmd2.Parameters.AddWithValue("@Username", txtusername.Text);
            cmd2.Parameters.AddWithValue("@LastLoginDate",DateTime.Now );
            cmd2.ExecuteNonQuery();
            Session["New"] = txtusername.Text;
            Response.Redirect("Details.aspx");
            con.Close();
          }
            else
            {
                // Incorrect values enetered
                Response.Write("<script>alert('Please enter valid Username and Password')</script>");
            }
        }
    }
}
```

7.3.17 Student Page : HTML Mark-Up

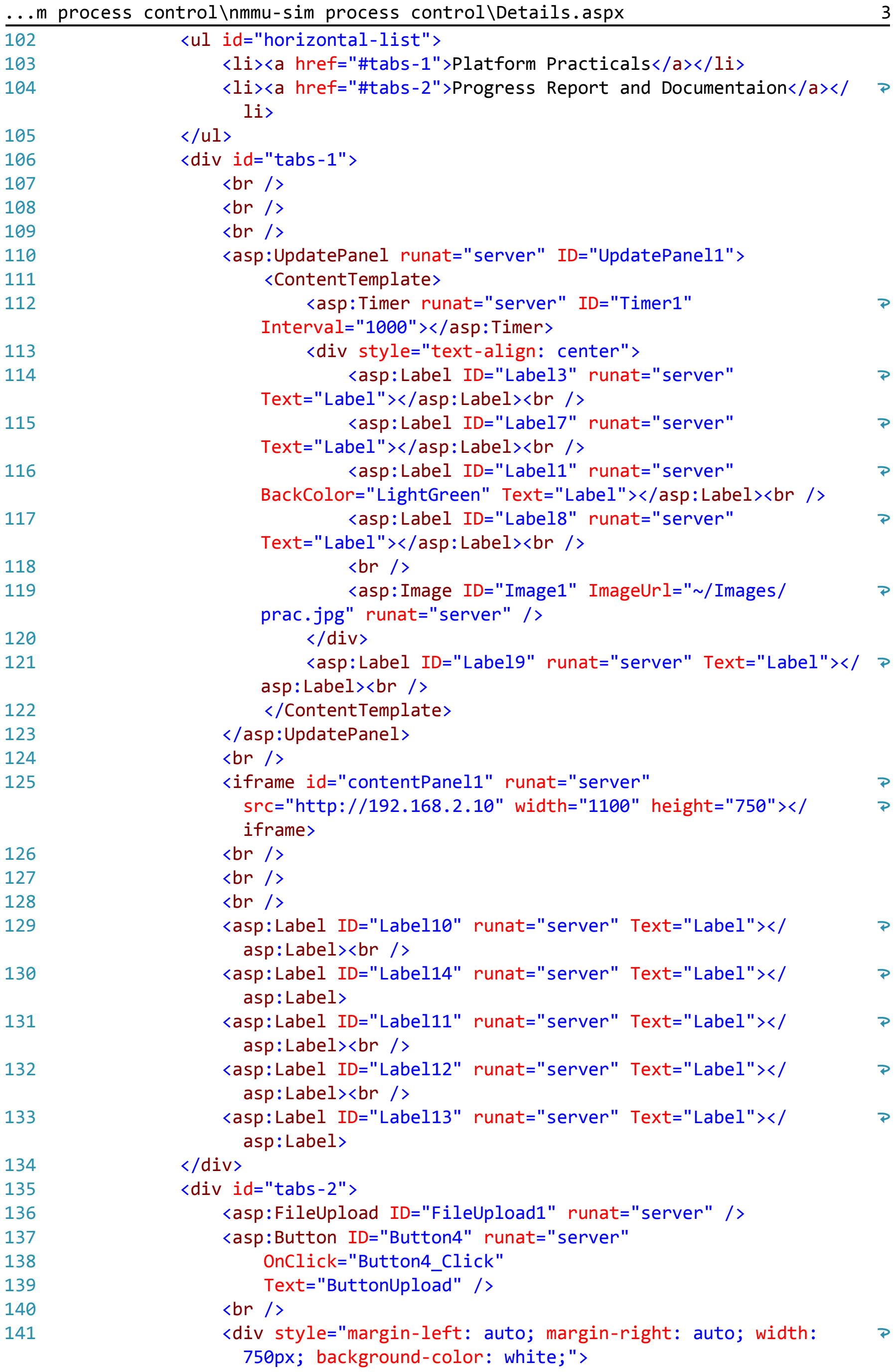


```
...m process control\nmmu-sim process control\Details.aspx                    1
<%@ Page Title="" Language="C#" MasterPageFile="~/basePage.Master"
  AutoEventWireup="true" CodeBehind="Details.aspx.cs"
  Inherits="nmmu_sim_process_control.Details" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
    <%-- <meta http-equiv="refresh" content="1" />--%>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
  runat="server">
    <form runat="server">
        <br />
        <br />
        <style style="text/css">
            .example1 {
                height: 50px;
                overflow: hidden;
                position: relative;
            }

                .example1 h3 {
                    position: absolute;
                    width: 100%;
                    height: 100%;
                    margin: 0;
                    line-height: 50px;
                    text-align: center;
                    /* Starting position */
                    -moz-transform: translateX(100%);
                    -webkit-transform: translateX(100%);
                    transform: translateX(100%);
                    /* Apply animation to this element */
                    -moz-animation: example1 45s linear infinite;
                    -webkit-animation: example1 45s linear infinite;
                    animation: example1 45s linear infinite;
                }
            /* Move it (define the animation) */
            @-moz-keyframes example1 {
                0% {
                    -moz-transform: translateX(100%);
                }

                100% {
                    -moz-transform: translateX(-100%);
                }
            }

            @-webkit-keyframes example1 {
                0% {
                    -webkit-transform: translateX(100%);
                }

                100% {
                    -webkit-transform: translateX(-100%);
                }
            }

```



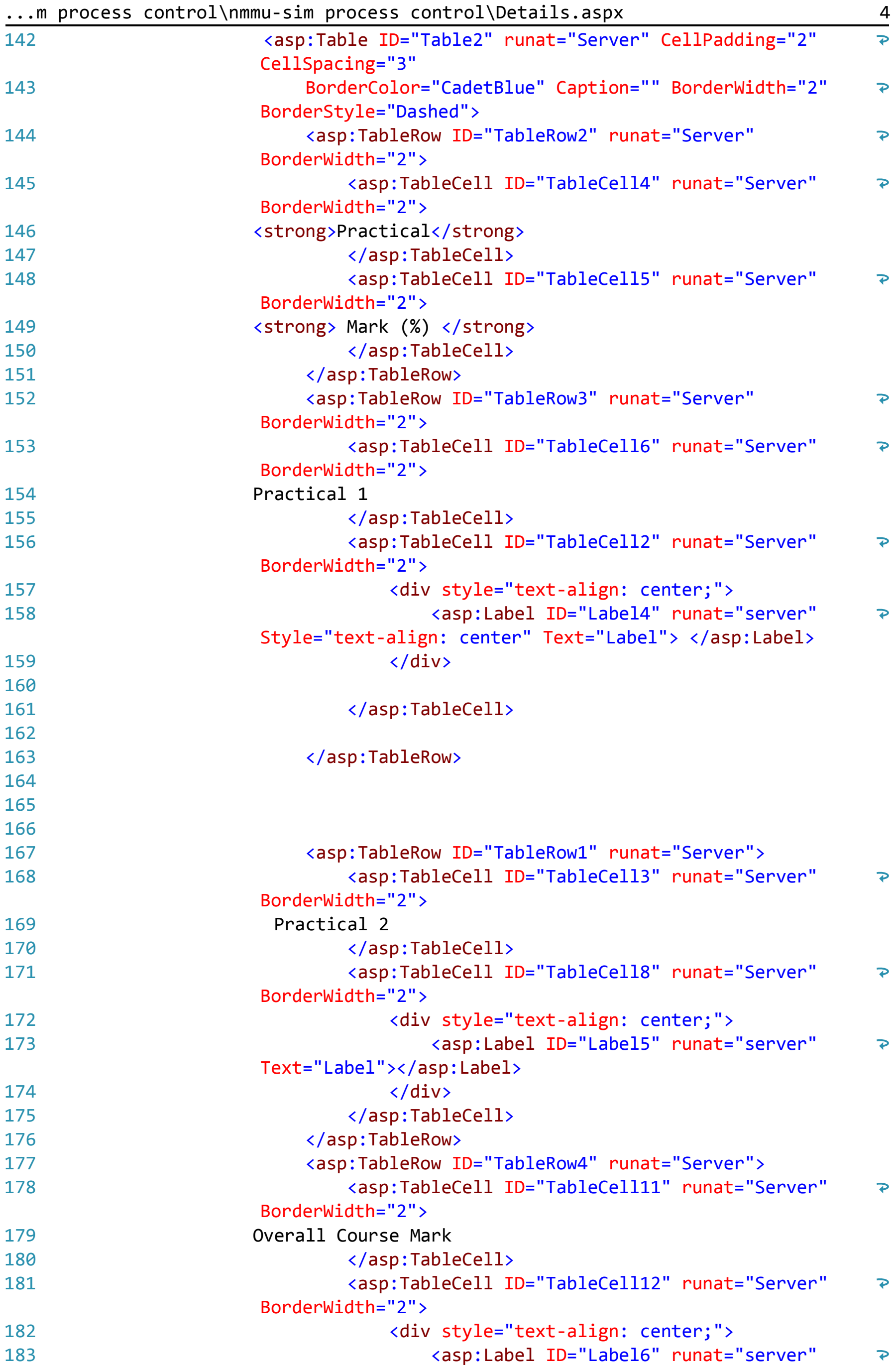
...m process control\nmmu-sim process control\Details.aspx 2

```
            @keyframes example1 {
                0% {
                    -moz-transform: translateX(100%); /* Firefox bug fix */
                    -webkit-transform: translateX(100%); /* Firefox bug fix */
                    transform: translateX(100%);
                }

                100% {
                    -moz-transform: translateX(-100%); /* Firefox bug fix */
                    -webkit-transform: translateX(-100%); /* Firefox bug fix */
                    transform: translateX(-100%);
                }
            }
        </style>

        <!-- HTML -->
        <div class="example1">
            <h3 style="color: #0094ff;">Student
                <asp:Label ID="Label2" runat="server" Text="Label"></asp:Label>
                welcome to the simulation platform integrated  with hardware control systems for a re-configurable process control</h3>
        </div>
        <br />
        <div style="text-align: right">
            <asp:Button ID="Button1" runat="server" Text="Logout" OnClick="Button1_Click" />

        </div>
        <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>
        <script src="http://ajax.aspnetcdn.com/ajax/jquery.ui/1.8.9/jquery-ui.js" type="text/javascript"></script>
        <link href="http://ajax.aspnetcdn.com/ajax/jquery.ui/1.8.9/themes/start/jquery-ui.css"
            rel="stylesheet" type="text/css" />
        <script type="text/javascript">
            var selected_tab = 1;
            $(function () {
                var tabs = $("#tabs").tabs({
                    select: function (e, i) {
                        selected_tab = i.index;
                    }
                });
                selected_tab = $("[id$=selected_tab]").val() != "" ? parseInt($("[id$=selected_tab]").val()) : 0;
                tabs.tabs('select', selected_tab);
                $("form").submit(function () {
                    $("[id$=selected_tab]").val(selected_tab);
                });
            });
        </script>
        <asp:ScriptManager runat="server" ID="ScriptManager1">
        </asp:ScriptManager>
        <div id="tabs">
```



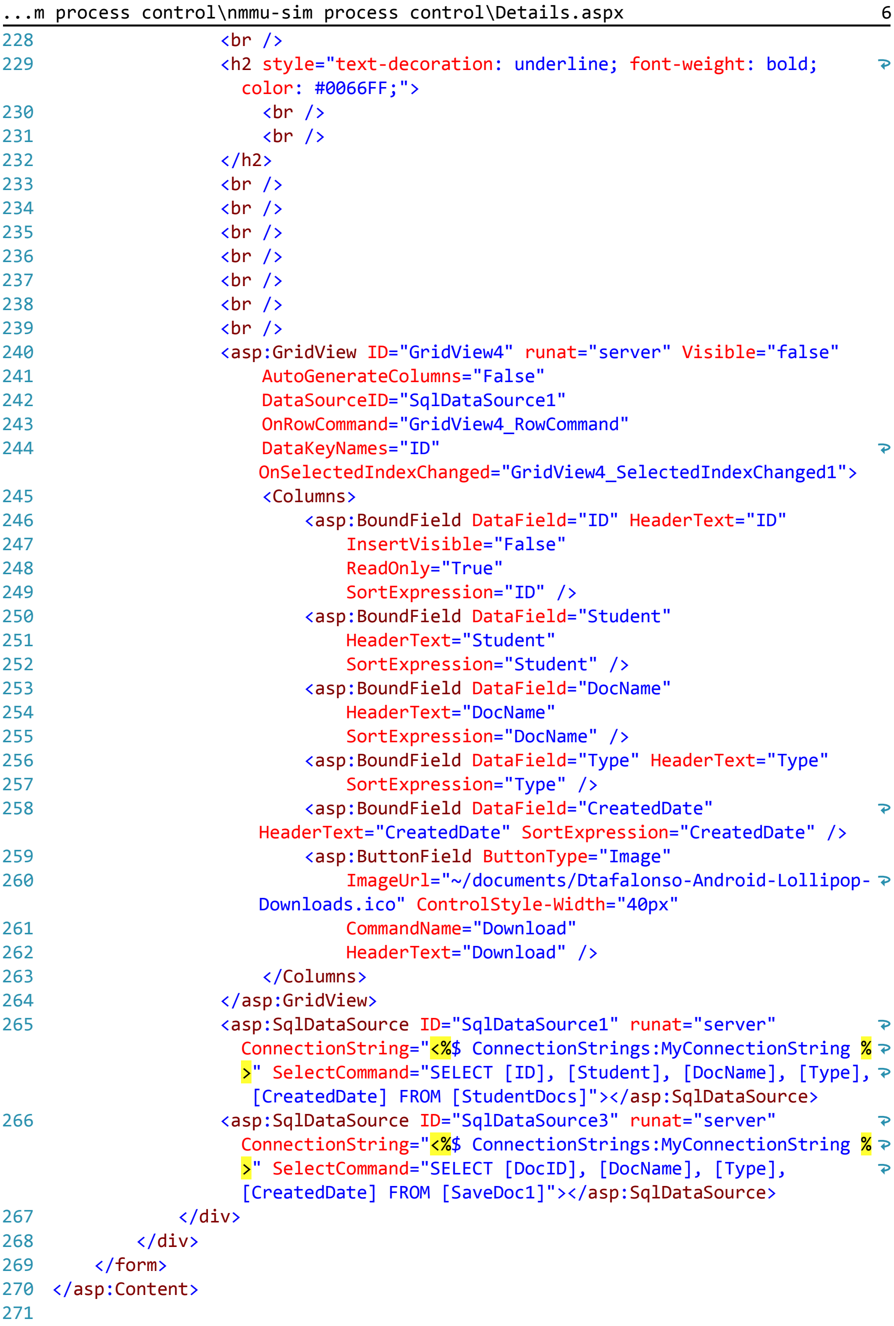
...m process control\nmmu-sim process control\Details.aspx 3

```
            <ul id="horizontal-list">
                <li><a href="#tabs-1">Platform Practicals</a></li>
                <li><a href="#tabs-2">Progress Report and Documentaion</a></li>
            </ul>
            <div id="tabs-1">
                <br />
                <br />
                <br />
                <asp:UpdatePanel runat="server" ID="UpdatePanel1">
                    <ContentTemplate>
                        <asp:Timer runat="server" ID="Timer1" Interval="1000"></asp:Timer>
                        <div style="text-align: center">
                            <asp:Label ID="Label3" runat="server" Text="Label"></asp:Label><br />
                            <asp:Label ID="Label7" runat="server" Text="Label"></asp:Label><br />
                            <asp:Label ID="Label1" runat="server" BackColor="LightGreen" Text="Label"></asp:Label><br />
                            <asp:Label ID="Label8" runat="server" Text="Label"></asp:Label><br />
                            <br />
                            <asp:Image ID="Image1" ImageUrl="~/Images/prac.jpg" runat="server" />
                        </div>
                        <asp:Label ID="Label9" runat="server" Text="Label"></asp:Label><br />
                    </ContentTemplate>
                </asp:UpdatePanel>
                <br />
                <iframe id="contentPanel1" runat="server" src="http://192.168.2.10" width="1100" height="750"></iframe>
                <br />
                <br />
                <br />
                <asp:Label ID="Label10" runat="server" Text="Label"></asp:Label><br />
                <asp:Label ID="Label14" runat="server" Text="Label"></asp:Label>
                <asp:Label ID="Label11" runat="server" Text="Label"></asp:Label><br />
                <asp:Label ID="Label12" runat="server" Text="Label"></asp:Label><br />
                <asp:Label ID="Label13" runat="server" Text="Label"></asp:Label>
            </div>
            <div id="tabs-2">
                <asp:FileUpload ID="FileUpload1" runat="server" />
                <asp:Button ID="Button4" runat="server"
                    OnClick="Button4_Click"
                    Text="ButtonUpload" />
                <br />
                <div style="margin-left: auto; margin-right: auto; width: 750px; background-color: white;">
```



```
...m process control\nmmu-sim process control\Details.aspx                    4
 <asp:Table ID="Table2" runat="Server" CellPadding="2"
CellSpacing="3"
     BorderColor="CadetBlue" Caption="" BorderWidth="2"
BorderStyle="Dashed">
     <asp:TableRow ID="TableRow2" runat="Server"
BorderWidth="2">
         <asp:TableCell ID="TableCell4" runat="Server"
BorderWidth="2">
<strong>Practical</strong>
         </asp:TableCell>
         <asp:TableCell ID="TableCell5" runat="Server"
BorderWidth="2">
<strong> Mark (%) </strong>
         </asp:TableCell>
     </asp:TableRow>
     <asp:TableRow ID="TableRow3" runat="Server"
BorderWidth="2">
         <asp:TableCell ID="TableCell6" runat="Server"
BorderWidth="2">
Practical 1
         </asp:TableCell>
         <asp:TableCell ID="TableCell2" runat="Server"
BorderWidth="2">
             <div style="text-align: center;">
                 <asp:Label ID="Label4" runat="server"
Style="text-align: center" Text="Label"> </asp:Label>
             </div>

         </asp:TableCell>

     </asp:TableRow>



     <asp:TableRow ID="TableRow1" runat="Server">
         <asp:TableCell ID="TableCell3" runat="Server"
BorderWidth="2">
  Practical 2
         </asp:TableCell>
         <asp:TableCell ID="TableCell8" runat="Server"
BorderWidth="2">
             <div style="text-align: center;">
                 <asp:Label ID="Label5" runat="server"
Text="Label"></asp:Label>
             </div>
         </asp:TableCell>
     </asp:TableRow>
     <asp:TableRow ID="TableRow4" runat="Server">
         <asp:TableCell ID="TableCell11" runat="Server"
BorderWidth="2">
Overall Course Mark
         </asp:TableCell>
         <asp:TableCell ID="TableCell12" runat="Server"
BorderWidth="2">
             <div style="text-align: center;">
                 <asp:Label ID="Label6" runat="server"
```

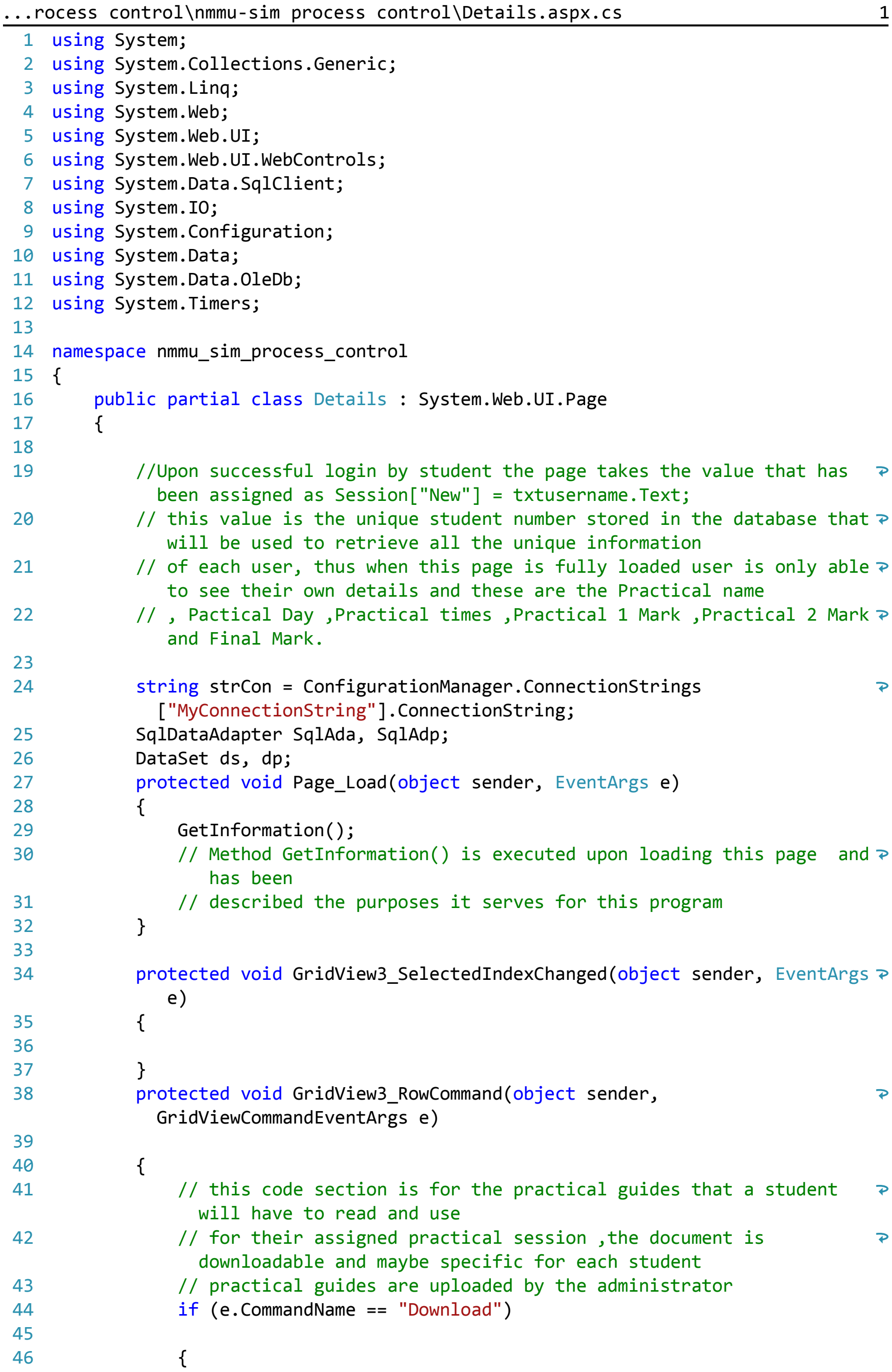
```
...m process control\nmmu-sim process control\Details.aspx                5
    Text="Label"></asp:Label>
                </div>
            </asp:TableCell>
        </asp:TableRow>
    </asp:Table>
</div>
<br />
<br />
<div style="margin-left: auto; margin-right: auto; width: 
  750px; background-color: white;">
    <asp:GridView ID="GridView3" runat="server"
        AutoGenerateColumns="False"
        DataSourceID="SqlDataSource2"
        OnRowCommand="GridView3_RowCommand"
        DataKeyNames="DocID" 
    OnSelectedIndexChanged="GridView3_SelectedIndexChanged1">
        <Columns>
            <asp:BoundField DataField="DocID" 
    HeaderText="DocID"
                InsertVisible="False"
                ReadOnly="True"
                SortExpression="DocID" />
            <asp:BoundField DataField="DocName"
                HeaderText="DocName"
                SortExpression="DocName" />
            <asp:BoundField DataField="Type" HeaderText="Type"
                SortExpression="Type" />
            <asp:BoundField DataField="CreatedDate" 
    HeaderText="CreatedDate" SortExpression="CreatedDate" />
            <asp:ButtonField ButtonType="Image"
                ImageUrl="~/documents/Dtafalonso-Android-
    Lollipop-Downloads.ico" ControlStyle-Width="40px"
                CommandName="Download"
                HeaderText="Download" />
        </Columns>
    </asp:GridView>
</div>
<asp:SqlDataSource ID="SqlDataSource2" runat="server" 
  ConnectionString="<%$ ConnectionStrings:MyConnectionString % 
  >" SelectCommand="SELECT [DocID], [DocName], [Type], 
  [CreatedDate] FROM [SaveDoc1]"></asp:SqlDataSource>
<br />
<br />
<br />
<asp:Label ID="Label15" runat="server" Text=""></asp:Label>
<br />
<br />
<br />
<asp:FileUpload ID="FileUploadToServer1" Width="300px" 
  runat="server" Visible="false" />
<asp:Button ID="btnUpload" runat="server" Text="Upload File" 
  OnClick="btnUpload_Click" Visible="false"
    ValidationGroup="vg" /><br />
<br />
<asp:Label ID="lblMsg" runat="server" ForeColor="Green" 
  Text=""></asp:Label>
```



...m process control\nmmu-sim process control\Details.aspx 6

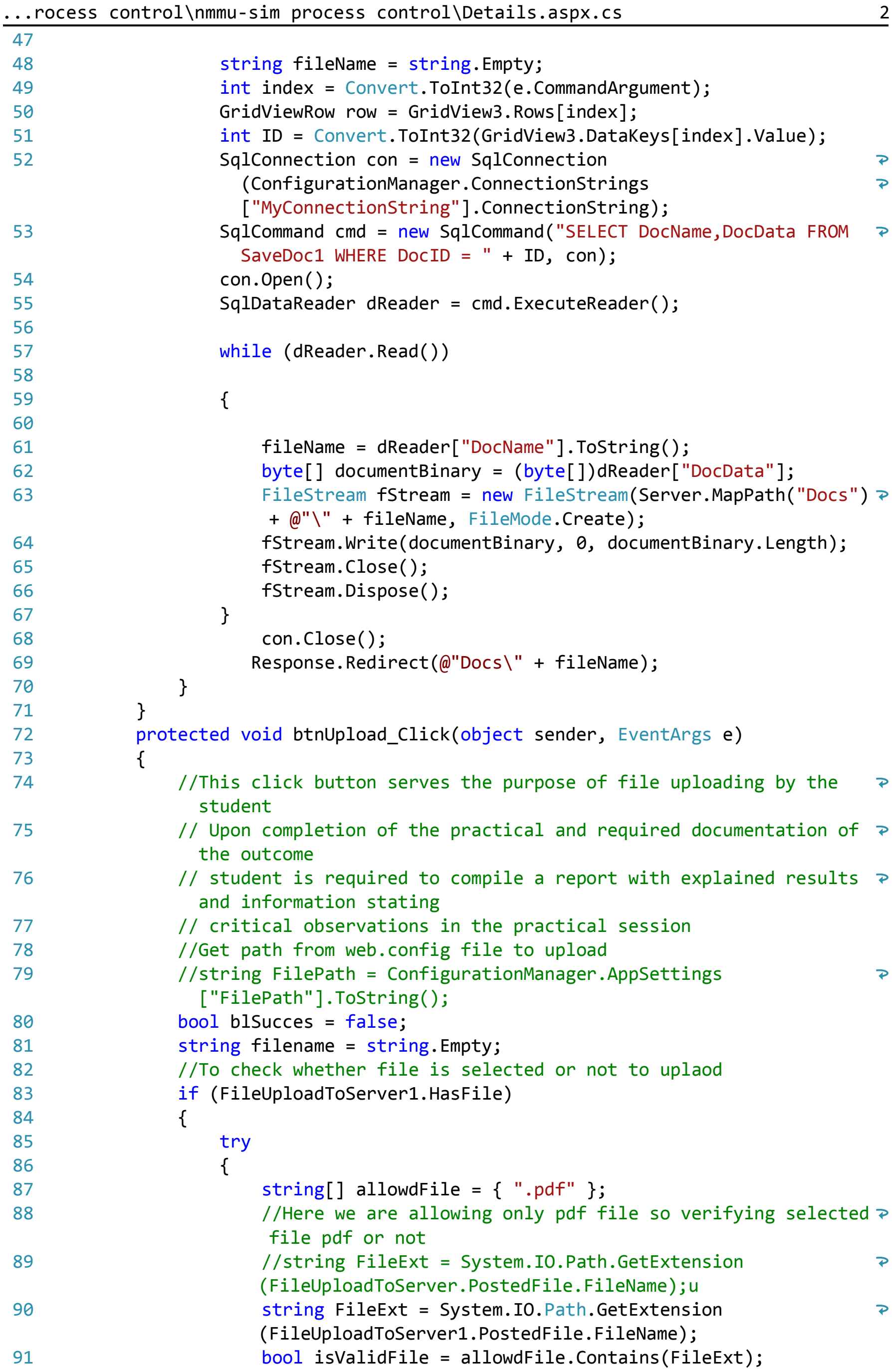
```
                <br />
                <h2 style="text-decoration: underline; font-weight: bold; color: #0066FF;">
                    <br />
                    <br />
                </h2>
                <br />
                <br />
                <br />
                <br />
                <br />
                <br />
                <br />
                <asp:GridView ID="GridView4" runat="server" Visible="false"
                    AutoGenerateColumns="False"
                    DataSourceID="SqlDataSource1"
                    OnRowCommand="GridView4_RowCommand"
                    DataKeyNames="ID" OnSelectedIndexChanged="GridView4_SelectedIndexChanged1">
                    <Columns>
                        <asp:BoundField DataField="ID" HeaderText="ID"
                            InsertVisible="False"
                            ReadOnly="True"
                            SortExpression="ID" />
                        <asp:BoundField DataField="Student"
                            HeaderText="Student"
                            SortExpression="Student" />
                        <asp:BoundField DataField="DocName"
                            HeaderText="DocName"
                            SortExpression="DocName" />
                        <asp:BoundField DataField="Type" HeaderText="Type"
                            SortExpression="Type" />
                        <asp:BoundField DataField="CreatedDate" HeaderText="CreatedDate" SortExpression="CreatedDate" />
                        <asp:ButtonField ButtonType="Image"
                            ImageUrl="~/documents/Dtafalonso-Android-Lollipop-Downloads.ico" ControlStyle-Width="40px"
                            CommandName="Download"
                            HeaderText="Download" />
                    </Columns>
                </asp:GridView>
                <asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:MyConnectionString %>" SelectCommand="SELECT [ID], [Student], [DocName], [Type], [CreatedDate] FROM [StudentDocs]"></asp:SqlDataSource>
                <asp:SqlDataSource ID="SqlDataSource3" runat="server" ConnectionString="<%$ ConnectionStrings:MyConnectionString %>" SelectCommand="SELECT [DocID], [DocName], [Type], [CreatedDate] FROM [SaveDoc1]"></asp:SqlDataSource>
            </div>
        </div>
    </form>
</asp:Content>

```

7.3.18 Student Page : C# Code



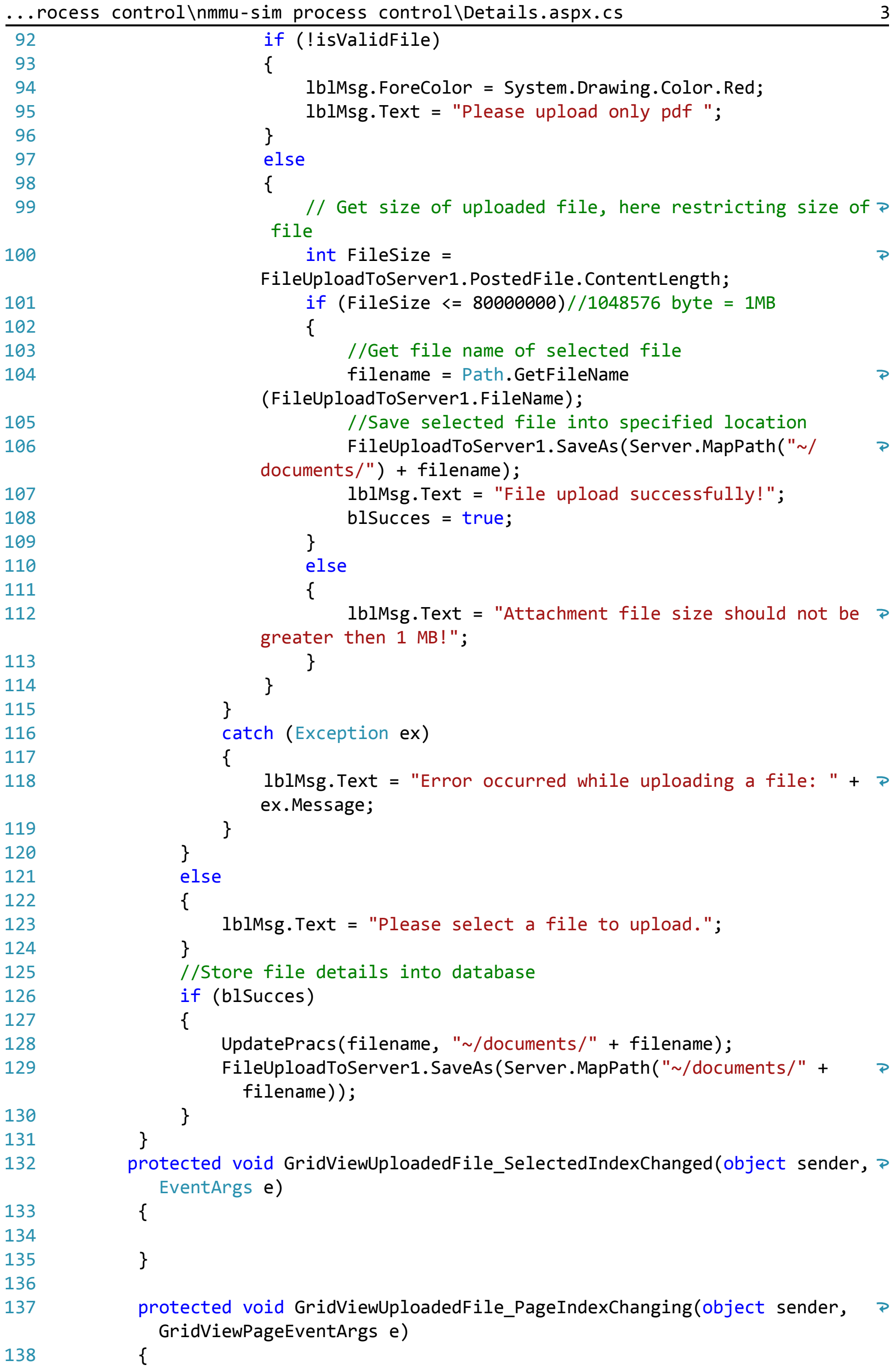
...rocess control\nmmu-sim process control\Details.aspx.cs 1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.IO;
using System.Configuration;
using System.Data;
using System.Data.OleDb;
using System.Timers;

namespace nmmu_sim_process_control
{
    public partial class Details : System.Web.UI.Page
    {

        //Upon successful login by student the page takes the value that has been assigned as Session["New"] = txtusername.Text;
        // this value is the unique student number stored in the database that will be used to retrieve all the unique information
        // of each user, thus when this page is fully loaded user is only able to see their own details and these are the Practical name
        // , Pactical Day ,Practical times ,Practical 1 Mark ,Practical 2 Mark and Final Mark.

        string strCon = ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString;
        SqlDataAdapter SqlAda, SqlAdp;
        DataSet ds, dp;
        protected void Page_Load(object sender, EventArgs e)
        {
            GetInformation();
            // Method GetInformation() is executed upon loading this page  and has been
            // described the purposes it serves for this program
        }

        protected void GridView3_SelectedIndexChanged(object sender, EventArgs e)
        {

        }
        protected void GridView3_RowCommand(object sender, GridViewCommandEventArgs e)

        {
            // this code section is for the practical guides that a student will have to read and use
            // for their assigned practical session ,the document is downloadable and maybe specific for each student
            // practical guides are uploaded by the administrator
            if (e.CommandName == "Download")

            {
```



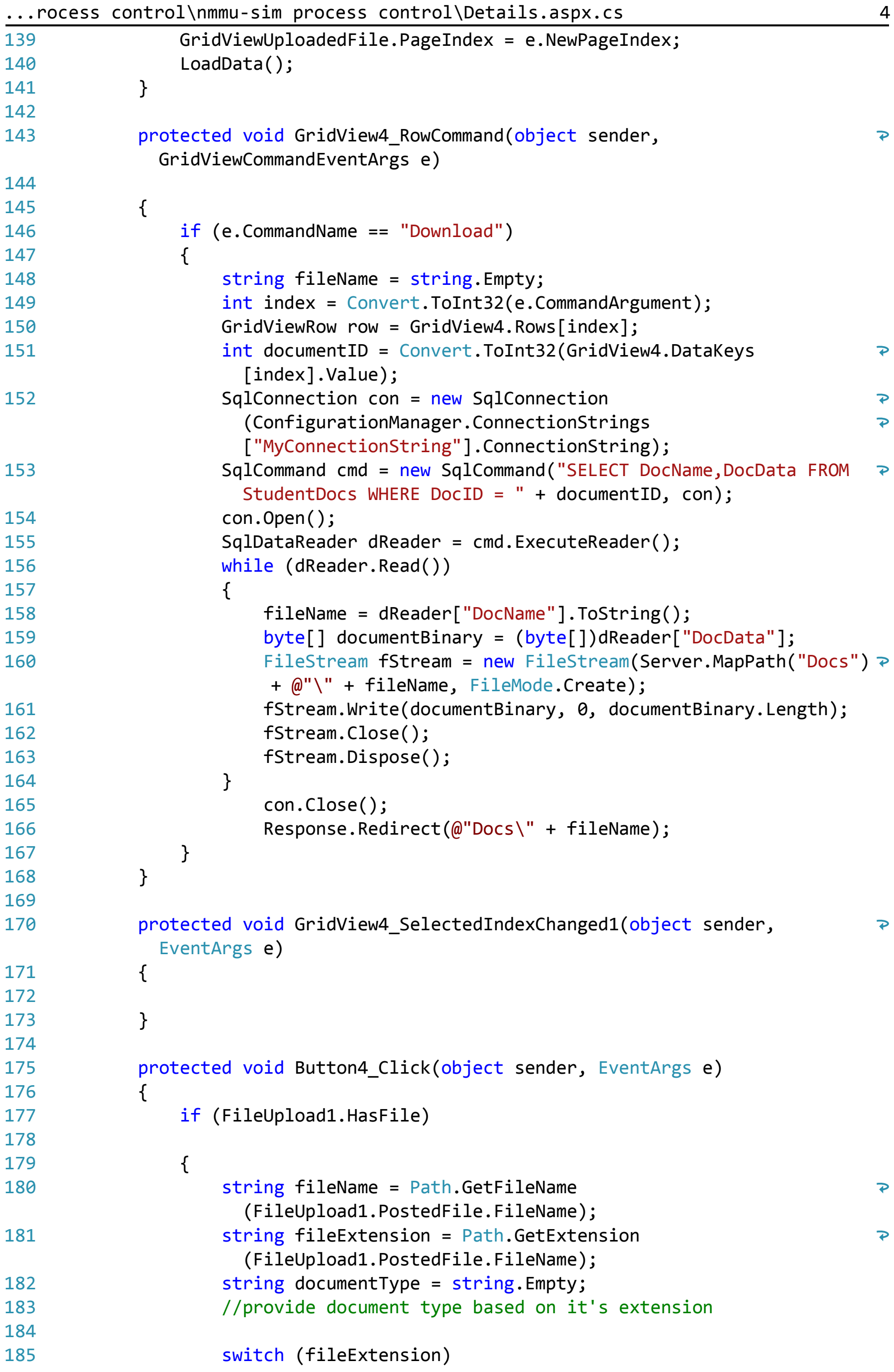
...rocess control\nmmu-sim process control\Details.aspx.cs 2

```

                string fileName = string.Empty;
                int index = Convert.ToInt32(e.CommandArgument);
                GridViewRow row = GridView3.Rows[index];
                int ID = Convert.ToInt32(GridView3.DataKeys[index].Value);
                SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString);
                SqlCommand cmd = new SqlCommand("SELECT DocName,DocData FROM SaveDoc1 WHERE DocID = " + ID, con);
                con.Open();
                SqlDataReader dReader = cmd.ExecuteReader();

                while (dReader.Read())

                {

                    fileName = dReader["DocName"].ToString();
                    byte[] documentBinary = (byte[])dReader["DocData"];
                    FileStream fStream = new FileStream(Server.MapPath("Docs") + @"\" + fileName, FileMode.Create);
                    fStream.Write(documentBinary, 0, documentBinary.Length);
                    fStream.Close();
                    fStream.Dispose();
                }
                    con.Close();
                   Response.Redirect(@"Docs\" + fileName);
            }
        }
        protected void btnUpload_Click(object sender, EventArgs e)
        {
            //This click button serves the purpose of file uploading by the student
            // Upon completion of the practical and required documentation of the outcome
            // student is required to compile a report with explained results and information stating
            // critical observations in the practical session
            //Get path from web.config file to upload
            //string FilePath = ConfigurationManager.AppSettings["FilePath"].ToString();
            bool blSucces = false;
            string filename = string.Empty;
            //To check whether file is selected or not to uplaod
            if (FileUploadToServer1.HasFile)
            {
                try
                {
                    string[] allowdFile = { ".pdf" };
                    //Here we are allowing only pdf file so verifying selected file pdf or not
                    //string FileExt = System.IO.Path.GetExtension(FileUploadToServer.PostedFile.FileName);u
                    string FileExt = System.IO.Path.GetExtension(FileUploadToServer1.PostedFile.FileName);
                    bool isValidFile = allowdFile.Contains(FileExt);
```

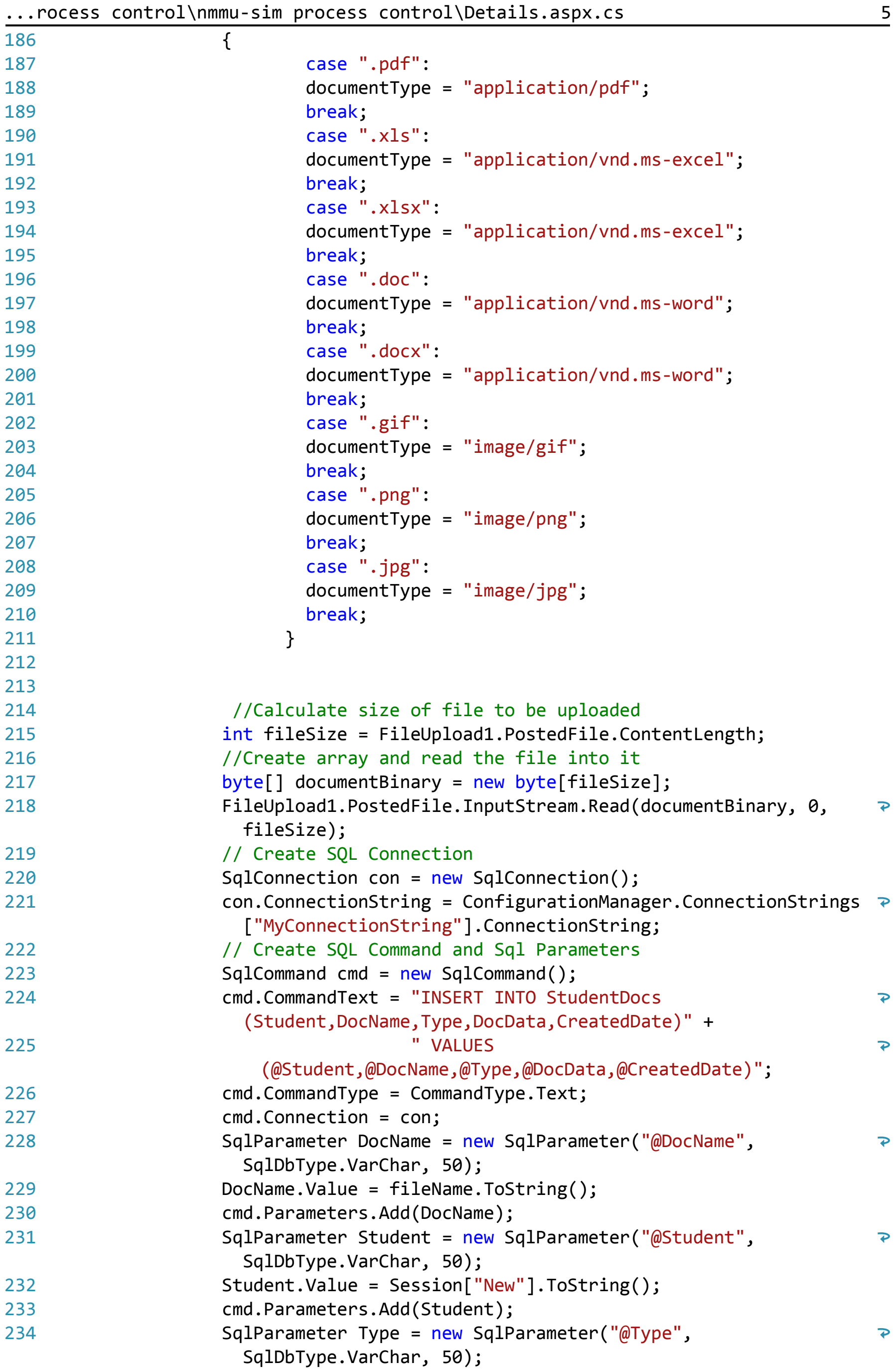


...rocess control\nmmu-sim process control\Details.aspx.cs 3

```
                    if (!isValidFile)
                    {
                        lblMsg.ForeColor = System.Drawing.Color.Red;
                        lblMsg.Text = "Please upload only pdf ";
                    }
                    else
                    {
                        // Get size of uploaded file, here restricting size of file
                        int FileSize = FileUploadToServer1.PostedFile.ContentLength;
                        if (FileSize <= 80000000)//1048576 byte = 1MB
                        {
                            //Get file name of selected file
                            filename = Path.GetFileName(FileUploadToServer1.FileName);
                            //Save selected file into specified location
                            FileUploadToServer1.SaveAs(Server.MapPath("~/documents/") + filename);
                            lblMsg.Text = "File upload successfully!";
                            blSucces = true;
                        }
                        else
                        {
                            lblMsg.Text = "Attachment file size should not be greater then 1 MB!";
                        }
                    }
                }
                catch (Exception ex)
                {
                    lblMsg.Text = "Error occurred while uploading a file: " + ex.Message;
                }
            }
            else
            {
                lblMsg.Text = "Please select a file to upload.";
            }
            //Store file details into database
            if (blSucces)
            {
                UpdatePracs(filename, "~/documents/" + filename);
                FileUploadToServer1.SaveAs(Server.MapPath("~/documents/" + filename));
            }
        }
       protected void GridViewUploadedFile_SelectedIndexChanged(object sender, EventArgs e)
        {

        }

        protected void GridViewUploadedFile_PageIndexChanging(object sender, GridViewPageEventArgs e)
        {
```

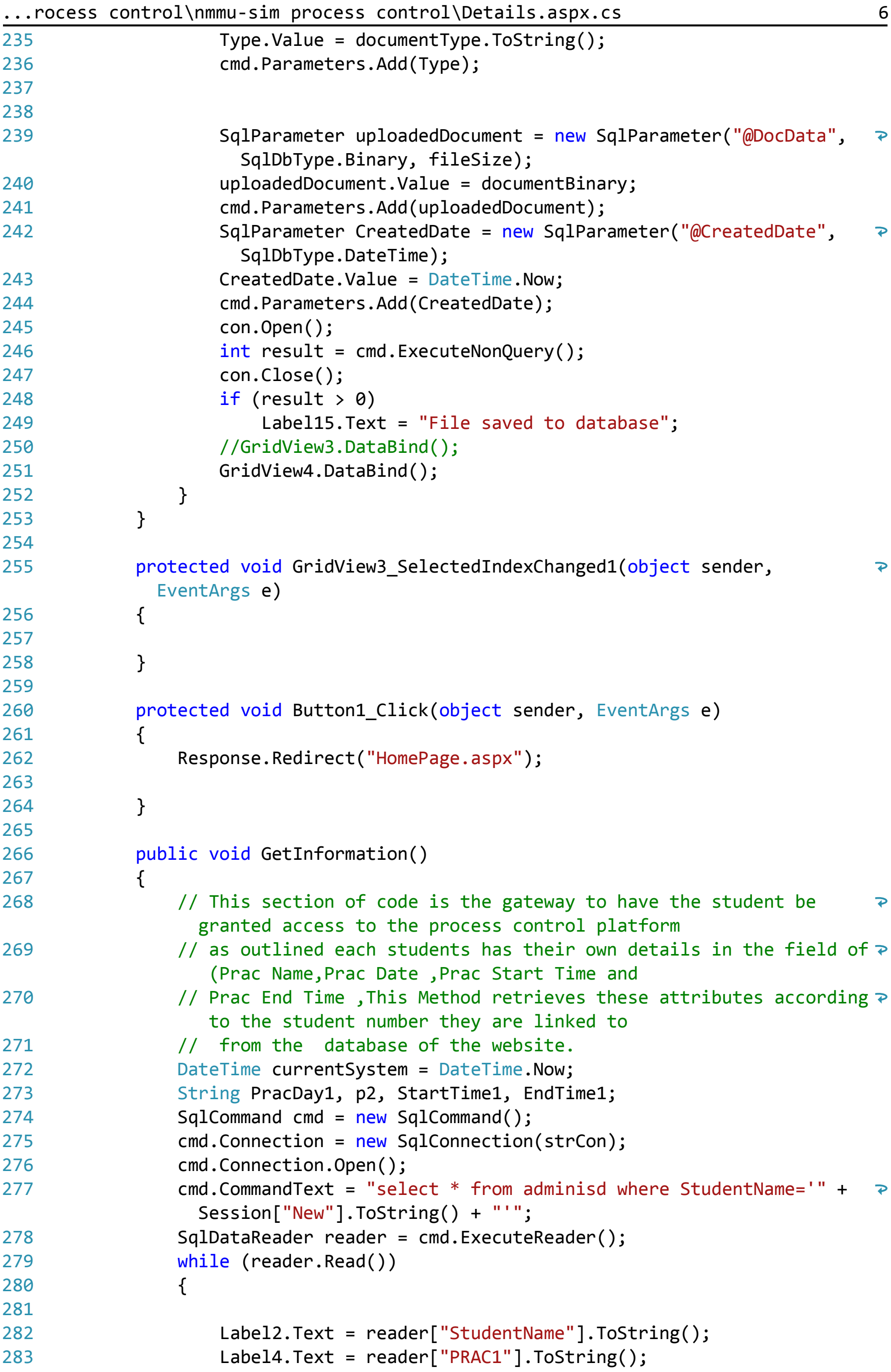


```
            GridViewUploadedFile.PageIndex = e.NewPageIndex;
            LoadData();
        }

        protected void GridView4_RowCommand(object sender, GridViewCommandEventArgs e)

        {
            if (e.CommandName == "Download")
            {
                string fileName = string.Empty;
                int index = Convert.ToInt32(e.CommandArgument);
                GridViewRow row = GridView4.Rows[index];
                int documentID = Convert.ToInt32(GridView4.DataKeys[index].Value);
                SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString);
                SqlCommand cmd = new SqlCommand("SELECT DocName,DocData FROM StudentDocs WHERE DocID = " + documentID, con);
                con.Open();
                SqlDataReader dReader = cmd.ExecuteReader();
                while (dReader.Read())
                {
                    fileName = dReader["DocName"].ToString();
                    byte[] documentBinary = (byte[])dReader["DocData"];
                    FileStream fStream = new FileStream(Server.MapPath("Docs") + @"\" + fileName, FileMode.Create);
                    fStream.Write(documentBinary, 0, documentBinary.Length);
                    fStream.Close();
                    fStream.Dispose();
                }
                    con.Close();
                    Response.Redirect(@"Docs\" + fileName);
            }
        }

        protected void GridView4_SelectedIndexChanged1(object sender, EventArgs e)
        {

        }

        protected void Button4_Click(object sender, EventArgs e)
        {
            if (FileUpload1.HasFile)

            {
                string fileName = Path.GetFileName(FileUpload1.PostedFile.FileName);
                string fileExtension = Path.GetExtension(FileUpload1.PostedFile.FileName);
                string documentType = string.Empty;
                //provide document type based on it's extension

                switch (fileExtension)
```



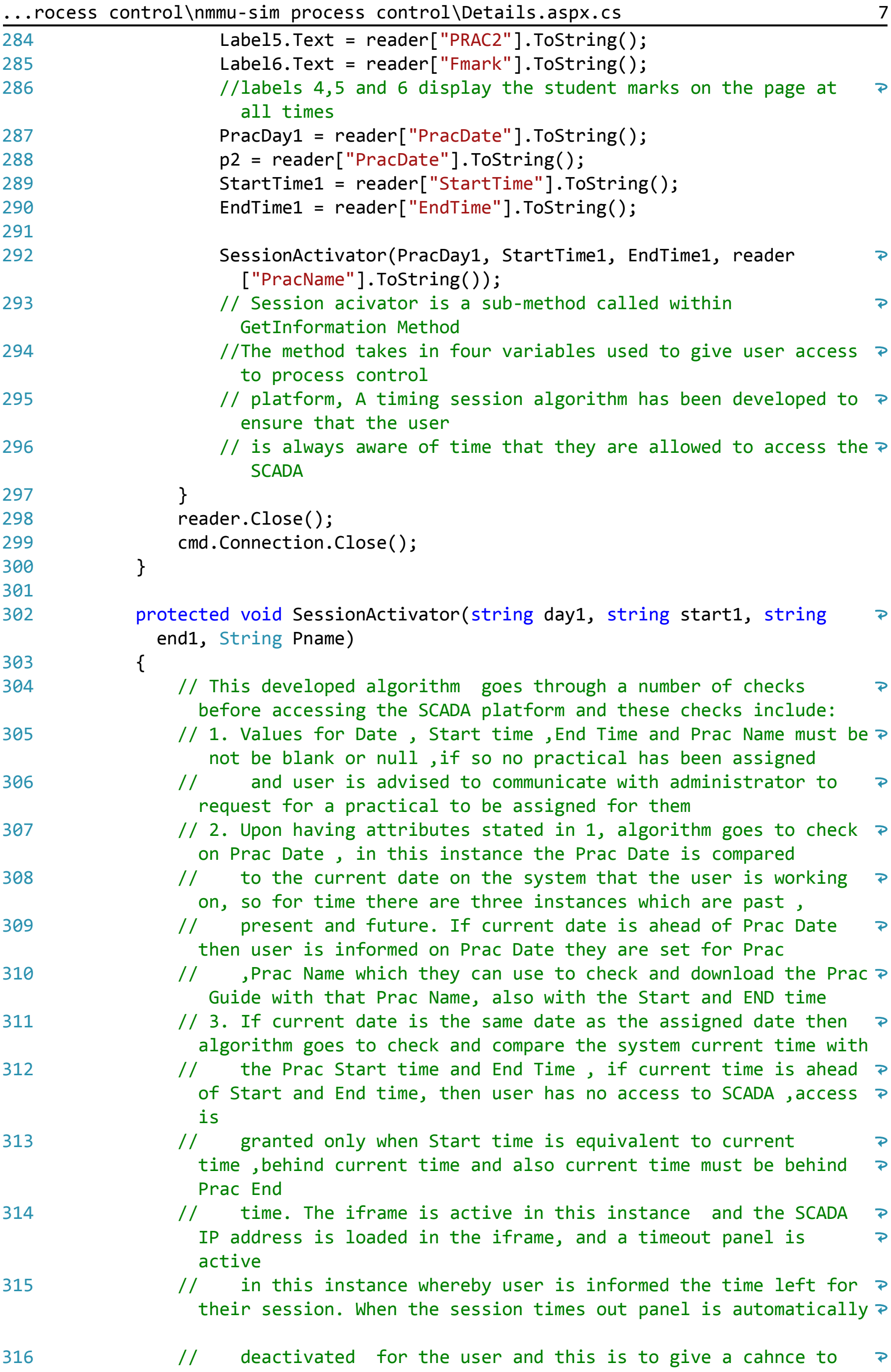
...rocess control\nmmu-sim process control\Details.aspx.cs 5

```
{
        case ".pdf":
        documentType = "application/pdf";
        break;
        case ".xls":
        documentType = "application/vnd.ms-excel";
        break;
        case ".xlsx":
        documentType = "application/vnd.ms-excel";
        break;
        case ".doc":
        documentType = "application/vnd.ms-word";
        break;
        case ".docx":
        documentType = "application/vnd.ms-word";
        break;
        case ".gif":
        documentType = "image/gif";
        break;
        case ".png":
        documentType = "image/png";
        break;
        case ".jpg":
        documentType = "image/jpg";
        break;
      }


 //Calculate size of file to be uploaded
int fileSize = FileUpload1.PostedFile.ContentLength;
//Create array and read the file into it
byte[] documentBinary = new byte[fileSize];
FileUpload1.PostedFile.InputStream.Read(documentBinary, 0, fileSize);
// Create SQL Connection
SqlConnection con = new SqlConnection();
con.ConnectionString = ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString;
// Create SQL Command and Sql Parameters
SqlCommand cmd = new SqlCommand();
cmd.CommandText = "INSERT INTO StudentDocs (Student,DocName,Type,DocData,CreatedDate)" +
                  " VALUES (@Student,@DocName,@Type,@DocData,@CreatedDate)";
cmd.CommandType = CommandType.Text;
cmd.Connection = con;
SqlParameter DocName = new SqlParameter("@DocName", SqlDbType.VarChar, 50);
DocName.Value = fileName.ToString();
cmd.Parameters.Add(DocName);
SqlParameter Student = new SqlParameter("@Student", SqlDbType.VarChar, 50);
Student.Value = Session["New"].ToString();
cmd.Parameters.Add(Student);
SqlParameter Type = new SqlParameter("@Type", SqlDbType.VarChar, 50);
```



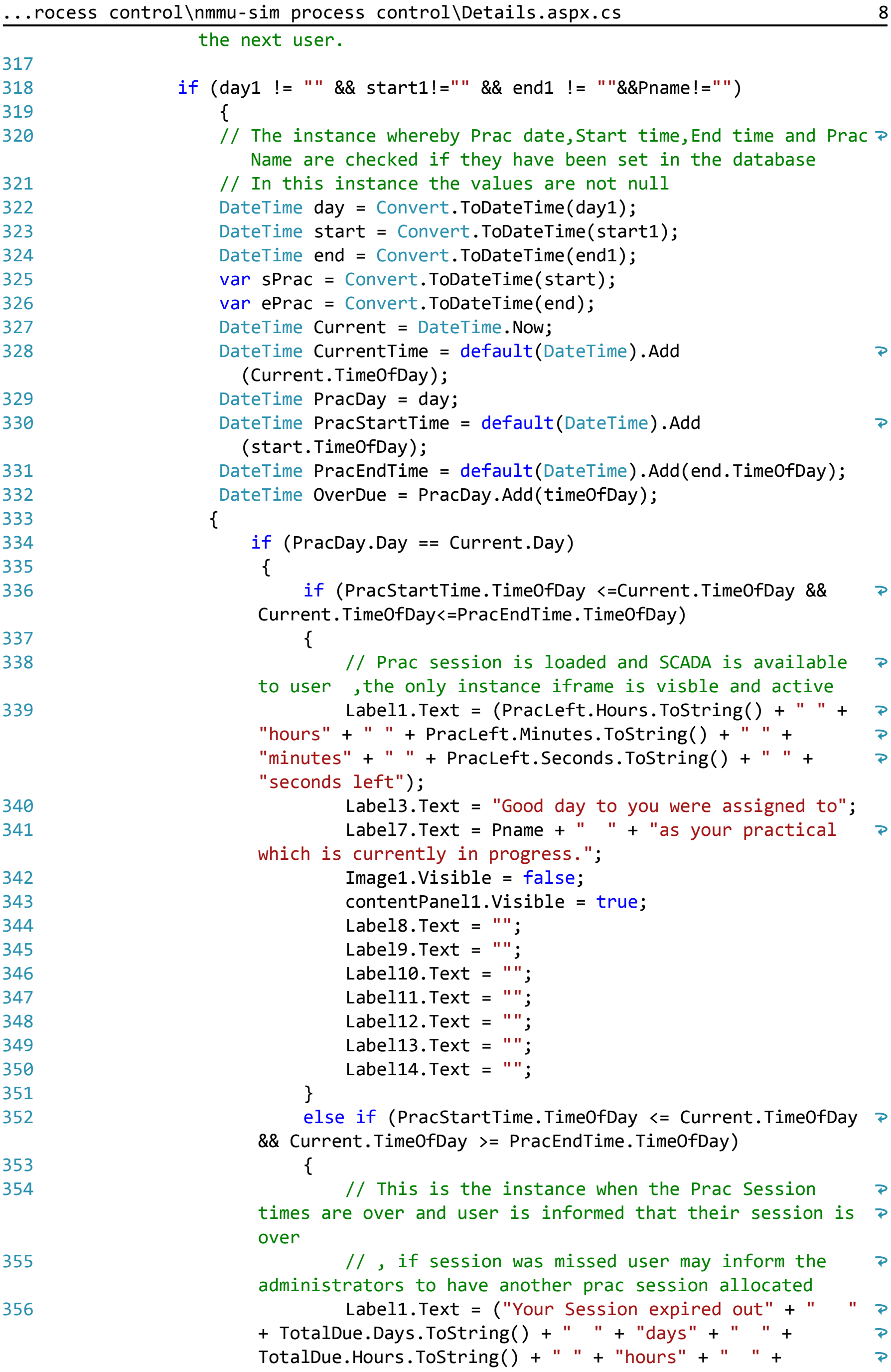
...rocess control\nmmu-sim process control\Details.aspx.cs 6

```
                Type.Value = documentType.ToString();
                cmd.Parameters.Add(Type);


                SqlParameter uploadedDocument = new SqlParameter("@DocData", SqlDbType.Binary, fileSize);
                uploadedDocument.Value = documentBinary;
                cmd.Parameters.Add(uploadedDocument);
                SqlParameter CreatedDate = new SqlParameter("@CreatedDate", SqlDbType.DateTime);
                CreatedDate.Value = DateTime.Now;
                cmd.Parameters.Add(CreatedDate);
                con.Open();
                int result = cmd.ExecuteNonQuery();
                con.Close();
                if (result > 0)
                    Label15.Text = "File saved to database";
                //GridView3.DataBind();
                GridView4.DataBind();
            }
        }

        protected void GridView3_SelectedIndexChanged1(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Redirect("HomePage.aspx");

        }

        public void GetInformation()
        {
            // This section of code is the gateway to have the student be granted access to the process control platform
            // as outlined each students has their own details in the field of (Prac Name,Prac Date ,Prac Start Time and
            // Prac End Time ,This Method retrieves these attributes according to the student number they are linked to
            //  from the  database of the website.
            DateTime currentSystem = DateTime.Now;
            String PracDay1, p2, StartTime1, EndTime1;
            SqlCommand cmd = new SqlCommand();
            cmd.Connection = new SqlConnection(strCon);
            cmd.Connection.Open();
            cmd.CommandText = "select * from adminisd where StudentName='" + Session["New"].ToString() + "'";
            SqlDataReader reader = cmd.ExecuteReader();
            while (reader.Read())
            {

                Label2.Text = reader["StudentName"].ToString();
                Label4.Text = reader["PRAC1"].ToString();
```



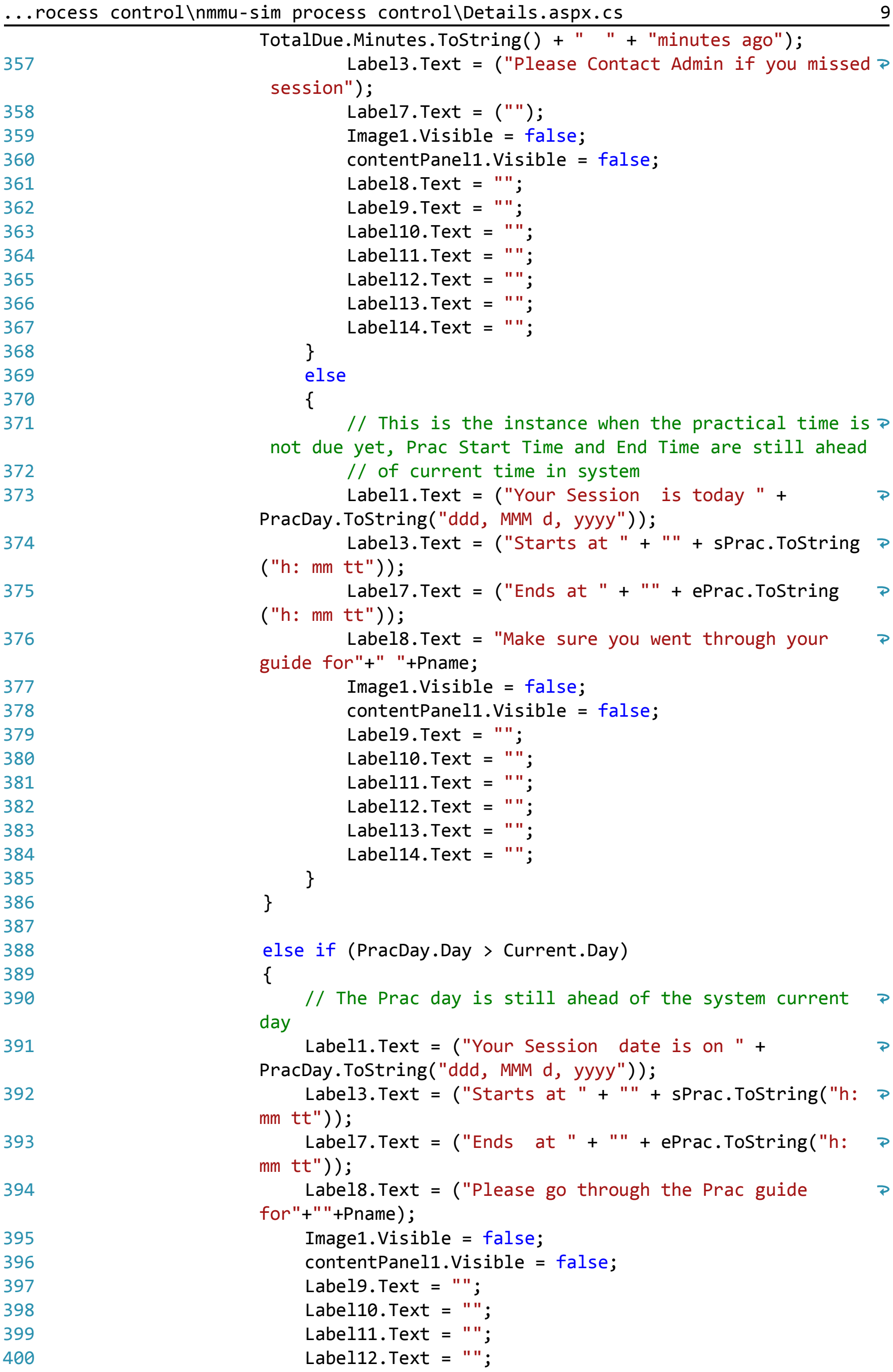
...rocess control\nmmu-sim process control\Details.aspx.cs 7

```
                Label5.Text = reader["PRAC2"].ToString();
                Label6.Text = reader["Fmark"].ToString();
                //labels 4,5 and 6 display the student marks on the page at all times
                PracDay1 = reader["PracDate"].ToString();
                p2 = reader["PracDate"].ToString();
                StartTime1 = reader["StartTime"].ToString();
                EndTime1 = reader["EndTime"].ToString();

                SessionActivator(PracDay1, StartTime1, EndTime1, reader["PracName"].ToString());
                // Session acivator is a sub-method called within GetInformation Method
                //The method takes in four variables used to give user access to process control
                // platform, A timing session algorithm has been developed to ensure that the user
                // is always aware of time that they are allowed to access the SCADA
            }
            reader.Close();
            cmd.Connection.Close();
        }

        protected void SessionActivator(string day1, string start1, string end1, String Pname)
        {
            // This developed algorithm  goes through a number of checks before accessing the SCADA platform and these checks include:
            // 1. Values for Date , Start time ,End Time and Prac Name must be not be blank or null ,if so no practical has been assigned
            //     and user is advised to communicate with administrator to request for a practical to be assigned for them
            // 2. Upon having attributes stated in 1, algorithm goes to check on Prac Date , in this instance the Prac Date is compared
            //    to the current date on the system that the user is working on, so for time there are three instances which are past ,
            //    present and future. If current date is ahead of Prac Date then user is informed on Prac Date they are set for Prac
            //    ,Prac Name which they can use to check and download the Prac Guide with that Prac Name, also with the Start and END time
            // 3. If current date is the same date as the assigned date then algorithm goes to check and compare the system current time with
            //    the Prac Start time and End Time , if current time is ahead of Start and End time, then user has no access to SCADA ,access is
            //    granted only when Start time is equivalent to current time ,behind current time and also current time must be behind Prac End
            //    time. The iframe is active in this instance  and the SCADA IP address is loaded in the iframe, and a timeout panel is active
            //    in this instance whereby user is informed the time left for their session. When the session times out panel is automatically
            //    deactivated  for the user and this is to give a cahnce to
```



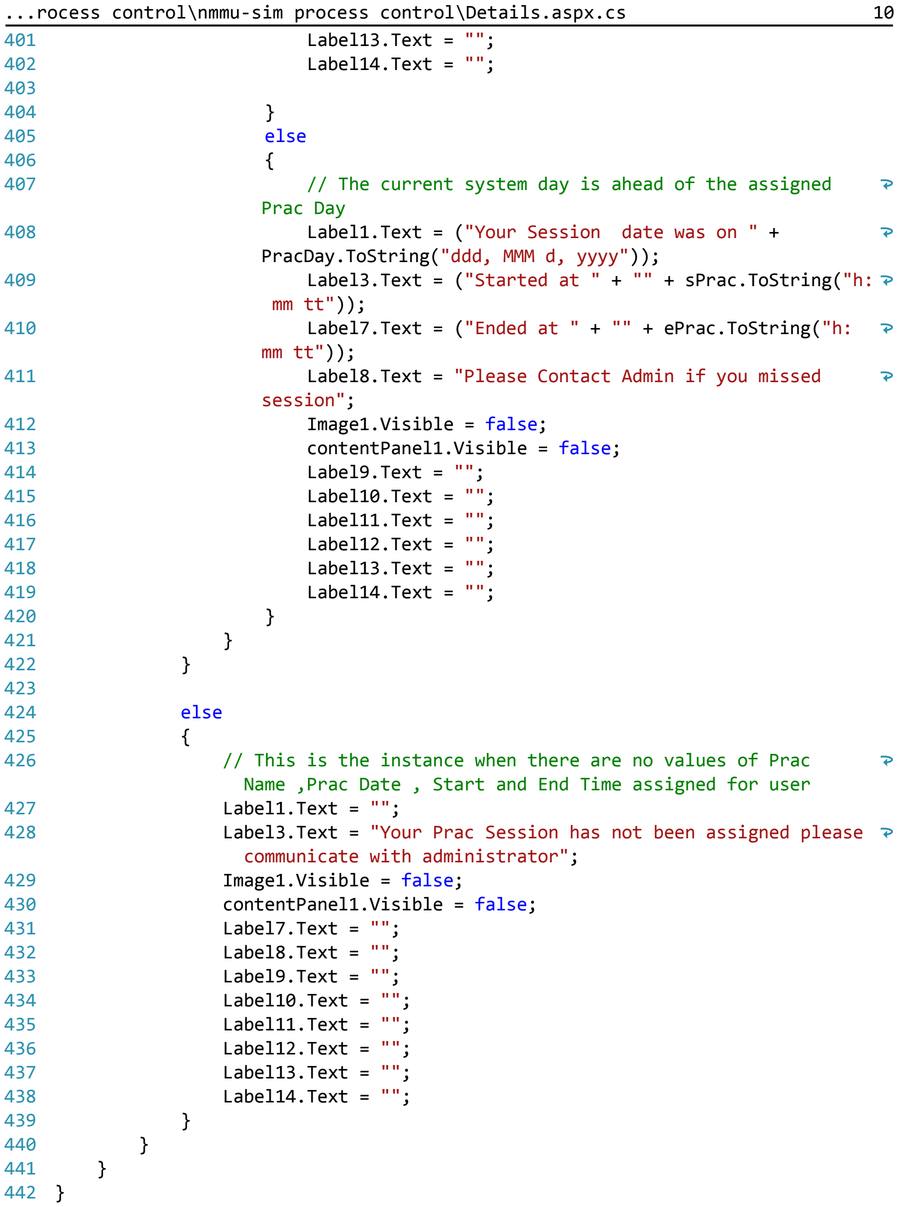
...rocess control\nmmu-sim process control\Details.aspx.cs 8

```
                 the next user.

            if (day1 != "" && start1!="" && end1 != ""&&Pname!="")
                {
                // The instance whereby Prac date,Start time,End time and Prac Name are checked if they have been set in the database
                // In this instance the values are not null
                DateTime day = Convert.ToDateTime(day1);
                DateTime start = Convert.ToDateTime(start1);
                DateTime end = Convert.ToDateTime(end1);
                var sPrac = Convert.ToDateTime(start);
                var ePrac = Convert.ToDateTime(end);
                DateTime Current = DateTime.Now;
                DateTime CurrentTime = default(DateTime).Add(Current.TimeOfDay);
                DateTime PracDay = day;
                DateTime PracStartTime = default(DateTime).Add(start.TimeOfDay);
                DateTime PracEndTime = default(DateTime).Add(end.TimeOfDay);
                DateTime OverDue = PracDay.Add(timeOfDay);
               {
                   if (PracDay.Day == Current.Day)
                    {
                        if (PracStartTime.TimeOfDay <=Current.TimeOfDay && Current.TimeOfDay<=PracEndTime.TimeOfDay)
                        {
                            // Prac session is loaded and SCADA is available to user  ,the only instance iframe is visble and active
                            Label1.Text = (PracLeft.Hours.ToString() + " " + "hours" + " " + PracLeft.Minutes.ToString() + " " + "minutes" + " " + PracLeft.Seconds.ToString() + " " + "seconds left");
                            Label3.Text = "Good day to you were assigned to";
                            Label7.Text = Pname + "  " + "as your practical which is currently in progress.";
                            Image1.Visible = false;
                            contentPanel1.Visible = true;
                            Label8.Text = "";
                            Label9.Text = "";
                            Label10.Text = "";
                            Label11.Text = "";
                            Label12.Text = "";
                            Label13.Text = "";
                            Label14.Text = "";
                        }
                        else if (PracStartTime.TimeOfDay <= Current.TimeOfDay && Current.TimeOfDay >= PracEndTime.TimeOfDay)
                        {
                            // This is the instance when the Prac Session times are over and user is informed that their session is over
                            // , if session was missed user may inform the administrators to have another prac session allocated
                            Label1.Text = ("Your Session expired out" + "    " + TotalDue.Days.ToString() + "  " + "days" + "  " + TotalDue.Hours.ToString() + " " + "hours" + "  " +
```



...rocess control\nmmu-sim process control\Details.aspx.cs 9

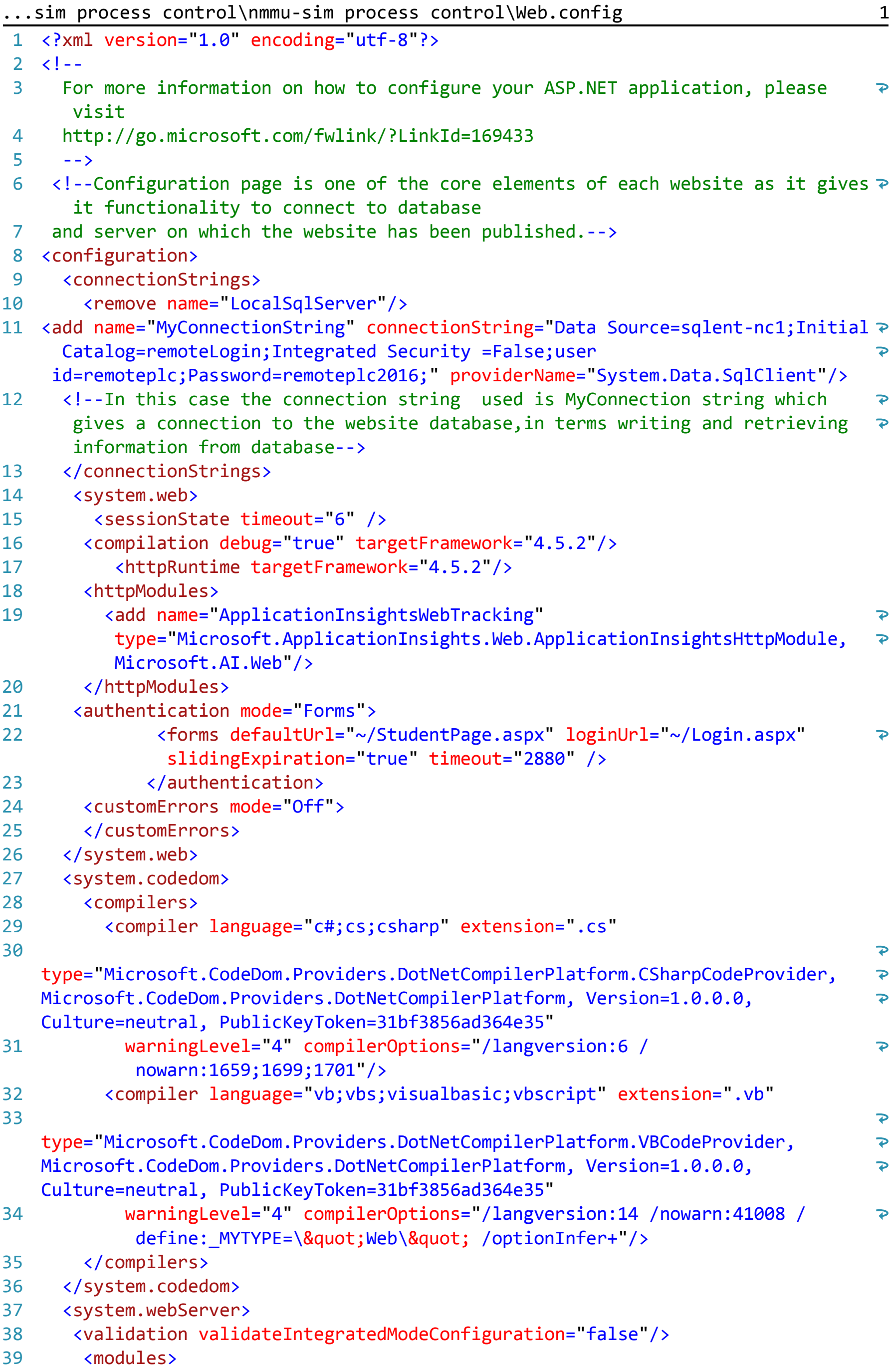
```
TotalDue.Minutes.ToString() + "  " + "minutes ago");
            Label3.Text = ("Please Contact Admin if you missed session");
            Label7.Text = ("");
            Image1.Visible = false;
            contentPanel1.Visible = false;
            Label8.Text = "";
            Label9.Text = "";
            Label10.Text = "";
            Label11.Text = "";
            Label12.Text = "";
            Label13.Text = "";
            Label14.Text = "";
        }
        else
        {
            // This is the instance when the practical time is not due yet, Prac Start Time and End Time are still ahead
            // of current time in system
            Label1.Text = ("Your Session  is today " + PracDay.ToString("ddd, MMM d, yyyy"));
            Label3.Text = ("Starts at " + "" + sPrac.ToString("h: mm tt"));
            Label7.Text = ("Ends at " + "" + ePrac.ToString("h: mm tt"));
            Label8.Text = "Make sure you went through your guide for"+" "+Pname;
            Image1.Visible = false;
            contentPanel1.Visible = false;
            Label9.Text = "";
            Label10.Text = "";
            Label11.Text = "";
            Label12.Text = "";
            Label13.Text = "";
            Label14.Text = "";
        }
    }

    else if (PracDay.Day > Current.Day)
    {
        // The Prac day is still ahead of the system current day
        Label1.Text = ("Your Session  date is on " + PracDay.ToString("ddd, MMM d, yyyy"));
        Label3.Text = ("Starts at " + "" + sPrac.ToString("h: mm tt"));
        Label7.Text = ("Ends  at " + "" + ePrac.ToString("h: mm tt"));
        Label8.Text = ("Please go through the Prac guide for"+""+Pname);
        Image1.Visible = false;
        contentPanel1.Visible = false;
        Label9.Text = "";
        Label10.Text = "";
        Label11.Text = "";
        Label12.Text = "";
```



...rocess control\nmmu-sim process control\Details.aspx.cs 10

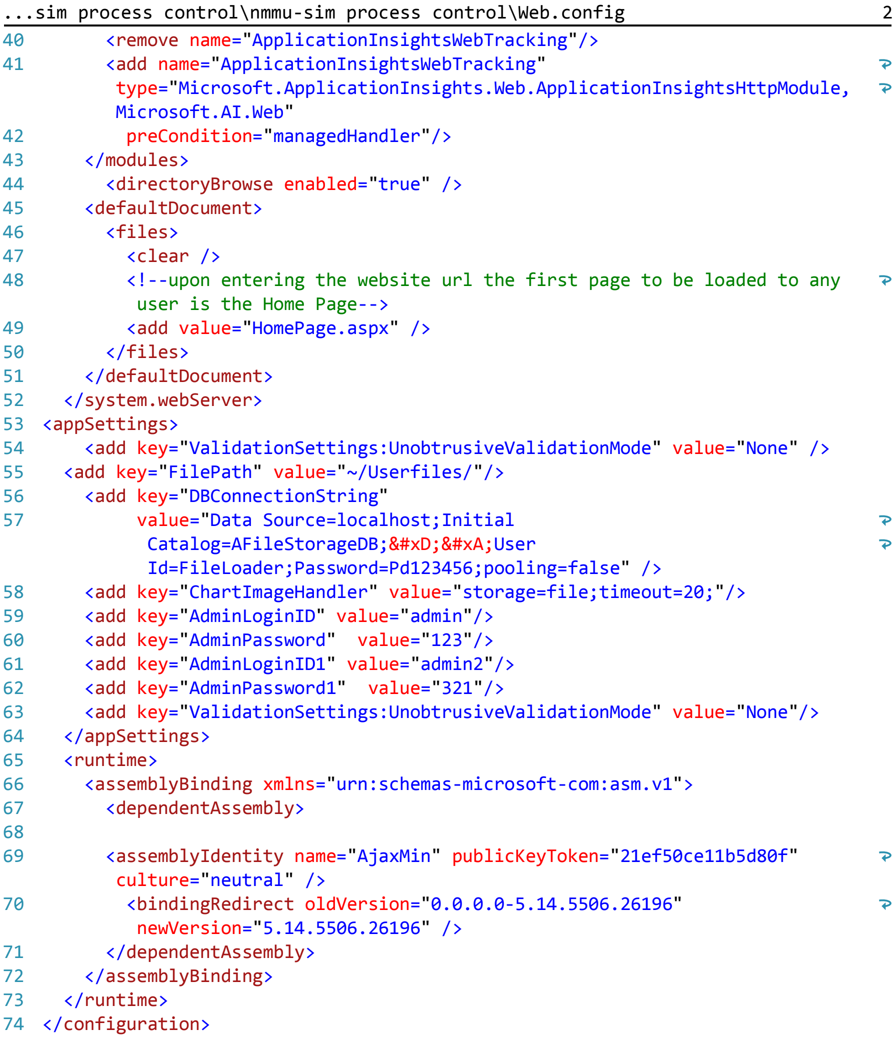
```
                        Label13.Text = "";
                        Label14.Text = "";

                    }
                    else
                    {
                        // The current system day is ahead of the assigned Prac Day
                        Label1.Text = ("Your Session  date was on " + PracDay.ToString("ddd, MMM d, yyyy"));
                        Label3.Text = ("Started at " + "" + sPrac.ToString("h: mm tt"));
                        Label7.Text = ("Ended at " + "" + ePrac.ToString("h:mm tt"));
                        Label8.Text = "Please Contact Admin if you missed session";
                        Image1.Visible = false;
                        contentPanel1.Visible = false;
                        Label9.Text = "";
                        Label10.Text = "";
                        Label11.Text = "";
                        Label12.Text = "";
                        Label13.Text = "";
                        Label14.Text = "";
                    }
                }
            }

            else
            {
                // This is the instance when there are no values of Prac Name ,Prac Date , Start and End Time assigned for user
                Label1.Text = "";
                Label3.Text = "Your Prac Session has not been assigned please communicate with administrator";
                Image1.Visible = false;
                contentPanel1.Visible = false;
                Label7.Text = "";
                Label8.Text = "";
                Label9.Text = "";
                Label10.Text = "";
                Label11.Text = "";
                Label12.Text = "";
                Label13.Text = "";
                Label14.Text = "";
            }
        }
    }
}
```

7.3.19 Website Configuration : HTML Mark-Up



...sim process control\nmmu-sim process control\Web.config 1

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
  -->
 <!--Configuration page is one of the core elements of each website as it gives it functionality to connect to database
 and server on which the website has been published.-->
<configuration>
  <connectionStrings>
    <remove name="LocalSqlServer"/>
<add name="MyConnectionString" connectionString="Data Source=sqlent-nc1;Initial Catalog=remoteLogin;Integrated Security =False;user id=remoteplc;Password=remoteplc2016;" providerName="System.Data.SqlClient"/>
  <!--In this case the connection string  used is MyConnection string which gives a connection to the website database,in terms writing and retrieving information from database-->
  </connectionStrings>
   <system.web>
     <sessionState timeout="6" />
    <compilation debug="true" targetFramework="4.5.2"/>
       <httpRuntime targetFramework="4.5.2"/>
    <httpModules>
      <add name="ApplicationInsightsWebTracking" type="Microsoft.ApplicationInsights.Web.ApplicationInsightsHttpModule, Microsoft.AI.Web"/>
    </httpModules>
   <authentication mode="Forms">
           <forms defaultUrl="~/StudentPage.aspx" loginUrl="~/Login.aspx" slidingExpiration="true" timeout="2880" />
          </authentication>
    <customErrors mode="Off">
    </customErrors>
  </system.web>
  <system.codedom>
    <compilers>
      <compiler language="c#;cs;csharp" extension=".cs"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider, Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
        warningLevel="4" compilerOptions="/langversion:6 /nowarn:1659;1699;1701"/>
      <compiler language="vb;vbs;visualbasic;vbscript" extension=".vb"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider, Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
        warningLevel="4" compilerOptions="/langversion:14 /nowarn:41008 /define:_MYTYPE=\&quot;Web\&quot; /optionInfer+"/>
    </compilers>
  </system.codedom>
  <system.webServer>
   <validation validateIntegratedModeConfiguration="false"/>
    <modules>
```



...sim process control\nmmu-sim process control\Web.config 2

```
      <remove name="ApplicationInsightsWebTracking"/>
      <add name="ApplicationInsightsWebTracking" type="Microsoft.ApplicationInsights.Web.ApplicationInsightsHttpModule, Microsoft.AI.Web"
        preCondition="managedHandler"/>
    </modules>
      <directoryBrowse enabled="true" />
    <defaultDocument>
      <files>
        <clear />
        <!--upon entering the website url the first page to be loaded to any user is the Home Page-->
        <add value="HomePage.aspx" />
      </files>
    </defaultDocument>
  </system.webServer>
<appSettings>
    <add key="ValidationSettings:UnobtrusiveValidationMode" value="None" />
  <add key="FilePath" value="~/Userfiles/"/>
    <add key="DBConnectionString"
         value="Data Source=localhost;Initial Catalog=AFileStorageDB;&#xD;&#xA;User Id=FileLoader;Password=Pd123456;pooling=false" />
    <add key="ChartImageHandler" value="storage=file;timeout=20;"/>
    <add key="AdminLoginID" value="admin"/>
    <add key="AdminPassword"  value="123"/>
    <add key="AdminLoginID1" value="admin2"/>
    <add key="AdminPassword1"  value="321"/>
    <add key="ValidationSettings:UnobtrusiveValidationMode" value="None"/>
  </appSettings>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>

      <assemblyIdentity name="AjaxMin" publicKeyToken="21ef50ce11b5d80f" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-5.14.5506.26196" newVersion="5.14.5506.26196" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```