

**REMOTE MONITORING AND CONTROL OF A RECONFIGURABLE SYSTEM
USING A WEB INTERFACE**

ONALENNA WHITNEY KEHOSITSE

Submitted in fulfilment of the requirements for the degree:

MASTER OF ENGINEERING: ELECTRICAL

in the

Faculty of Engineering and Information Technology

Department of Electrical, Electronic and Computer Engineering

at the

Central University of Technology, Free State

Supervisor: Prof. HJ Vermaak, PhD

2016

Declaration of Independent Work

I, ONALENNA WHITNEY KEHOSITSE, identity number [REDACTED], and student number [REDACTED] hereby declare that this research project which has been submitted to the Central University of Technology for Degree Master of Engineering: Electrical, is my own independent work; and complies with the Code of Academy Integrity, as well as relevant policies, procedures, rules and regulations of the Central University of Technology; and has not been submitted by any person in fulfilment of the requirement for the attainment of any qualification.

.....

SIGNATURE OF STUDENT

.....

DATE

Acknowledgement

I would like to thank GOD almighty whom through him everything is possible, Prof HJ Vermaak who gave me the opportunity and supervised me throughout the research, my mom who sacrificed a lot for me to be where I am, my best friend TJ Matsemela and all those who supported me. Lastly I would like to thank The Central University of Technology for their facilities and financial support.

Abstract

The assembly system at the Research Group in Evolvable Manufacturing System (RGEMS) is designed to assemble products part by part and also ensuring good quality products. This system uses different kinds of industrial devices that are integrated in a way that the devices are dependent on each other in order to work properly on a production cycle. The disadvantage of systems of this nature is that if one of the devices malfunctions and is not attended to, the production line will get affected, which will cost time and money. In order to ensure that malfunctioning of devices is acknowledged in time, the system devices must be monitored while busy in progress; this is normally done by a system operator who uses a computer and checks if there is any change in system data using a Supervisory Control and Data Acquisition (SCADA) system.

This is, however, inconvenient in that the system operator can only monitor and control the system when connected to the system's network. Today we live in a wireless world, where anything can be accessible from anywhere in the world; internet and web technologies made all this possible. It was therefore proposed to design a system that can monitor and control the assembly system from a remote location. This system uses OLE Process Control (OPC) technology and web technology in order to read and write data to the system from the OPC server through the internet. This data is presented on a web user interface where it is viewed as analogue or digital data. The user only needs the internet browser installed to his/her computer or smart phone and also to be authorized by the system administrator in order to be able to monitor and control the system.

Table of Contents

Declaration of Independent Work	I
Acknowledgement	II
Table of Contents.....	IV
List of figures.....	IX
List of Tables	XIII
List of Abbreviations	XIV
1. CHAPTER I.....	1
1.1 PURPOSE OF THIS STUDY	2
1.2 PROBLEM STATEMENT	2
1.3 HYPOTHESIS	2
1.4 WEB-BASED REMOTE MONITORING AND CONTROL SYSTEM	3
1.5 METHODOLOGY	4
1.6 PROJECT OUTCOME	7
1.7 OUTLINE OF THESIS.....	7
2. CHAPTER II.....	10
2.1 INTRODUCTION.....	10
2.2 DISTRIBUTED WEB APPLICATION.....	10
2.3 TYPES OF APPLICATIONS.....	12
2.3.1 Single-tier Application	12
2.3.2 Two-tier Application	13
2.3.3 Multi-Tier	14
2.4 ELEMENTS OF A DISTRIBUTED APPLICATION.....	15

2.4.1	Presentation Tier (User and Presentation Tier)	16
2.4.2	Middle Layer (Business Logic Tier)	17
2.4.3	Lower Layer (Data Tier).....	18
2.5	OPC TECHNOLOGY IN AUTOMATED CONTROL SYSTEMS.....	18
2.5.1	OPC SPECIFICATIONS.....	21
2.6	REVIEW ON THE DESIGN OF THE CURRENT WEB-BASED RMCS	22
2.7	OPC XML-DA	25
2.7.1	SOAP Protocol	27
2.8	MODEL OF WEB INTEGRATION	27
2.8.1	Benefits of Web Integration	29
2.9	WEB SERVICES	30
2.9.1	Implementation of Web services	31
2.9.1.1	Data Management	32
2.9.1.2	Server Management.....	32
2.9.1.3	Address Space Management	33
2.9.2	Limitations of Web services	34
2.10	OPC SECURITY ISSUE	35
2.11	CONCLUSION.....	36
3.	CHAPTER III.....	38
3.1	INTRODUCTION.....	38
3.2	SYSTEM DESCRIPTION.....	38
3.3	SYSTEM OPERATION.....	40
3.4	THE SYSTEM DEVELOPMENT LIFE CYCLE.....	42

3.5	SYSTEM ANALYSIS AND REQUIREMENTS	43
3.5.1	Specification of Remote Monitoring and Control Systems.....	43
I	Step 1: Monitoring or Control?	44
II	Step 2: What type of remote monitoring system is required?	45
III	Step 3: What are the benefits of making this system remote monitored?	45
IV	Step 4: What type of remote control system is required?.....	46
V	Step 5: What are the benefits of making this system remote controlled?.....	46
VI	Step 6: Specification of use - cases and exchanged information.....	47
3.6	System Design	49
3.6.1	System Architecture	49
3.6.2	Advantages obtained using a 4-tier system	51
3.6.3	Web Server (ASP.NET Web Server)	51
3.6.4	Log in.....	52
3.6.5	Monitoring Function.....	52
3.6.6	Control Function.....	53
3.6.7	Intelligent user support	53
3.6.8	Database Server (MySQL Server).....	53
3.6.9	Control Server (OPC Server).....	54
3.7	DETAILED SYSTEM DESIGN.....	55
3.7.1	Login Module	55
3.7.2	Monitoring Module.....	58
3.7.3	Control Module.....	61
3.8	CONCLUSION.....	64

4. CHAPTER IV	65
4.1 INTRODUCTION.....	65
4.2 DESIGNING OBJECTIVE.....	65
4.2.1 System Implementation	65
4.3 USER ACCESS MANAGEMENT SYSTEM.....	71
4.3.1 User Registration System	77
4.3.2 User Management System	80
4.3.3 Login System.....	83
4.4 MONITORING SYSTEM.....	85
4.4.1 Monitored Data	86
4.5 HISTORICAL DATA ARCHIVING SYSTEM	103
4.6 CONTROL SYSTEM.....	106
4.6.1 Start Function.....	108
4.6.2 Stop Function	109
4.7 CONCLUSION.....	111
5. CHAPTER V.....	112
5.1 INTRODUCTION.....	112
5.2 ACCESS CONTROL.....	112
5.2.1 Login Page.....	113
5.2.2 User Registration Page	114
5.2.3 User Management Page	118
5.3 MONITORING SYSTEM PAGE	120
5.3.1 Alarms and Events System.....	123

5.4	CONTROL SYSTEM PAGE	131
5.5	SYSTEM DEPLOYMENT	134
5.6	SATISFYING THE HYPOTHESIS	135
6.	CHAPTER VI	137
6.1	INTRODUCTION.....	137
6.2	PROJECT SUMMARY	137
6.2.1	System Functionality	138
6.2.2	Remote Monitoring includes:	138
6.2.3	Remote Control includes:	138
6.2.4	System Operator Functionality	138
6.2.5	RMCS Security.....	139
6.3	CONTRIBUTION OF THE STUDY	140
I.	Manufacturing Industry.....	141
II.	Mining Industry.....	142
6.4	FUTURE STUDIES.....	143
	APPENDIX I: REFERENCES	144

List of figures

Figure 1.1: Client\Server model.....	5
Figure 2.1: User Interface, Business Rules, Presentation Rules and Database Resides Separately	15
Figure 2.2: OPC Connectivity.....	20
Figure 2.3: Xiaofeng Lee Design model	24
Figure 2.4: Ju Chen Design Model	25
Figure 2.5: Generic Architecture of web integration for process monitoring systems	29
Figure 2.6: The aspect of web service implementation.....	34
Figure 3.1: RGEMS OPC Configuration	39
Figure 3.2: The sub-system of the assembling system.....	40
Figure 3.3: Production cycle of the system.....	41
Figure 3.4: Software Development Life Cycle	42
Figure 3.5: ULM Case Diagram for Modelling the System.....	48
Figure 3.6: The System Overview	50
Figure 3.7: Login activity diagram	56
Figure 3.8: UML class diagram for Login Control	58
Figure 3.9: UML activity diagram for System monitor	60
Figure 3.10: System Monitor class diagram	61
Figure 3.11: System Control activity diagram	62
Figure 3.12: System Control Class diagram	63
Figure 4.1: OPC System.NET service running on a local machine	66
Figure 4.2 OPC System window for configuring tags	67
Figure 4.3: OPC System Service Control	68
Figure 4.4: OPC System.NET OPC server browser	69

Figure 4.5 OPC System service configuration window	70
Figure 4.6: User Access Management System Architectural Design	71
Figure 4.7: Connection string for connecting to the database.....	74
Figure 4.8: User Access Management System Database Structure	76
Figure 4.9: User registration method in C#.....	78
Figure 4.10: Unique password generation method using C#	79
Figure 4.11: Username creation method using C#.....	80
Figure 4.12: Grid view filled with list of registered users	81
Figure 4.13: Activate edit on grid when row editing event is invoked	81
Figure 4.14: Perform system update on row update event.....	82
Figure 4.15: Perform system delete when row delete event is invoked.....	83
Figure 4.16: Authenticate and authorize user on button click event.....	84
Figure 4.17: Architectural design of a monitoring and control system.....	85
Figure 4.18: OPC Web Control Textbox Properties for system status	86
Figure 4.19: Browsing tags on a local host.....	88
Figure 4.20: System Status tag configuration	89
Figure 4.21: System control panel	90
Figure 4.22: Current consumed tag configuration	92
Figure 4.23: OPC Web Control Textbox Properties for current consumed	93
Figure 4.24: Current consumed alarm configuration	94
Figure 4.25: Browsing tags on a local host.....	95
Figure 4.26: Three-phase induction motors used to drive the conveyor belt.....	96
Figure 4.27: Arithmetic calculation to determine the state of the system.....	97
Figure 4.28: Proximity Sensor used for counting number of products produced.	98
Figure 4.29 OPC Web Control Textbox Properties for number of units produced	99
Figure 4.30: Calculating power consumed	100

Figure 4.31: OPC Web Control Textbox Properties for power consumed	101
Figure 4.32: Architectural design for historical data archiving system	103
Figure 4.33: OPC System data logging configuration window	104
Figure 4.34: SQL Server Management Studio user access window	105
Figure 4.35: Data logged in the database	106
Figure 4.36: Control system structure.....	107
Figure 4.37: Human Machine Interface	108
Figure 4.38: OPC web control start button properties	109
Figure 4.39: OPC web control stop button properties.....	110
Figure 5.1: RMCS Login Page.....	113
Figure 5.2: Incorrect credentials error.....	114
Figure 5.3: RMCS User registration page.....	115
Figure 5.4: RMCS required field error message	116
Figure 5.5: RMCS creates username and password.....	116
Figure 5.6: Null record error message	117
Figure 5.7: RMCS Change password page	118
Figure 5.8: RMCS User management page (Edit Mode).....	119
Figure 5.9: RMCS user management page	120
Figure 5.10: RMCS monitoring page (When the system is running)	121
Figure 5.11: RMCS monitoring page (When the system is on halt).....	122
Figure 5.12: Alarm management system (Alarm active).....	124
Figure 5.13: Alarm management system (Alarm acknowledged).....	125
Figure 5.14: Load, Current and Temperature data trends in real-time.....	127
Figure 5.15: Load, Current and Temperature historical data trends	129
Figure 5.16: Data archived in a csv file	130
Figure 5.17: Data logged in the database.....	131

Figure 5.18: RMCS control page indicating that the system is running 132

Figure 5.19: RMCS Control indicating that the system has stopped 133

Figure 5.20: IIS server 134

Figure 5.21: Adding a web site to the IIS server..... 135

List of Tables

Table 2.1: Service Attributes of each Application Element.....	12
Table 4.1: Motor Specifications.....	91

List of Abbreviations

AES	Advanced Encryption Standard
AS	Assembly System
ASP	Active Server Page
CMMS	Computerized Maintenance Management System
COM	Component Object Model
CPU	Central Processing Unit
DA	Data Access
DCOM	Distributed Component Object Model
EAM	Enterprise Asset Management
GUI	Graphic User Interface
HMI	Human Machine Interface
HTML	Hyper-Text Mark-up Language
HTTP	Hypertext Transfer Protocol
IBM	International Business Machine
ID	Identity
IP	Internet Protocol
LAN	Local Area Network
MES	Manufacturing Execution System

MTU	Monitoring Terminal Unit
OPC	Object Linking and Embedding for Process Control
OPC XML_DA	Object Linking and Embedding (OLE) Process Control Extended Mark-up Language Data Access
PLC	Programmable Logic Device
PHP	Hypertext Pre-processor
RDBMS	Relational Database Management System
RGEMS	Research Group in Evolvable Manufacturing Systems
RMCS	Remote Monitoring and Control System
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SDLC	Software Development Life Cycle
SOAP	Simple Object Access Protocol
UML	Unified Modelling Language
URL	Uniform Resource Locator
WAN	Wide Area Network
WSDL	Web Service Definition Language
XML	eXtended Mark-up Language

1. CHAPTER I

Introduction

Web technologies are proliferating through plant automation systems and enabling managers to review production and control data from anywhere they can access the Internet [1]. Web-based plant monitoring can be used to obtain production data, such as how much each plant is producing; what devices are getting produced; what the supply levels are; and what orders have been filed. Web technologies allow plant operators to monitor and control their system and to diagnose problems as they occur [2].

The Research Group in Evolvable Manufacturing Systems (RGEMS) located within the Department of Electrical, Electronic and Computer Engineering aims to design a system that will allow remote monitoring and control of manufacturing and assembly systems. The combination of technologies such as ASP.NET, JavaScript and HTML and widespread Internet connectivity now provides possibilities for a solution to this research problem [3]. Using any standard Internet browser, user(s) can monitor real-time plant performance conditions from anywhere in the world. This remote monitoring technology allows plant engineers to monitor the conditions of the plant facilities from remote locations. If a problem occurs, operators can view the real-time conditions of the affected plant area, diagnose the problem remotely, and provide advice to the on-site plant engineers, thereby reducing costly plant failures and breakdowns [4].

1.1 Purpose of this Study

The aim of this study is to develop a web interface system in order to monitor and control an assembly system at RGEMS remotely.

1.2 Problem Statement

The monitoring and control system that is only accessible through a local area network (LAN) has the following disadvantages:

- LAN is a private network that links computers at a single location. This means that the monitoring and control system can only be accessed locally by the user within the network. For you to connect to the localized systems, you need to be within its network range; thus, not allowing users to connect to the system if not within the network.
- If the file server goes down, the entire network may go down.
- Special client/server software must be installed.
- A larger network becomes difficult to manage.

1.3 Hypothesis

A system that uses a web interface in order to control and monitor process data through the Internet and that has the following advantages:

- The internet links millions of networks with over a billion computers connected at any given time. Systems that are monitored or controlled through the internet are convenient, flexible and inexpensive because the internet infrastructure has already been built.

- Web browser as a tool for commissioning, service and visualization. No proprietary software tools or runtime license necessary.
- Smart phones can be used for control and monitoring.

1.4 Web-based Remote Monitoring and Control System

Web-based Remote Monitoring and Control Systems (RMCS) make use of Internet and Hypertext Transfer Protocol (HTTP) and other Web technologies as a communication layer of the system. It also uses development tools, framework, platforms and computer languages used by regular Internet applications as development application. Web-based RMCS uses the Internet to transfer data between the system devices or operators and the system. This will reduce the cost of installation of the SCADA network if compared with installing a dedicated network for it. It also uses Internet browsers programs such as Mozilla Firefox, Netscape Navigator or Microsoft Internet Explorer as Graphical User Interface (GUI) for the operators [5]. This would give all the benefits of browser-based systems, such as simplifying the installation process on the client side of the RMCS and also enable users to access the system using a wide range of platforms, as browsers are now available in almost all of the modern operating systems.

Web-based RMCS have many advantages, including: [5]

- Using the client/server n-tier platforms and development tools to develop Web-based RMCS will cut the development cost and time to minimum.
- Using the infrastructure of the Internet or the corporate intranet will cut the deployment cost to the minimum.
- Increase distance, data sharing and data provision for monitoring and control systems.

- Enabling collaboration between skilled plant managers situated in geographically diverse locations.
- Enabling the business to relocate the physical location of plant management staff easily in response to business needs.
- For education and researching purposes, the risk involved in a real laboratory may be avoided by doing dangerous experiments remotely.

1.5 Methodology

The most sensible approach for accessing, controlling and monitoring a plant system via the Internet is to use a browser like Netscape or Internet Explorer. Web-based systems are based on a client/server model (as seen Figure 1.1) which is used primarily for computer network communication. This model may include multiple servers and multiple clients in which one server can handle multiple requests from different clients and also one client can send multiple requests to different servers at any time. The software development of this system is divided into two parts, the client side and the server side. Whilst the client side interacts with users and executes the control task, the server side implements the data acquisition program to achieve the control task.

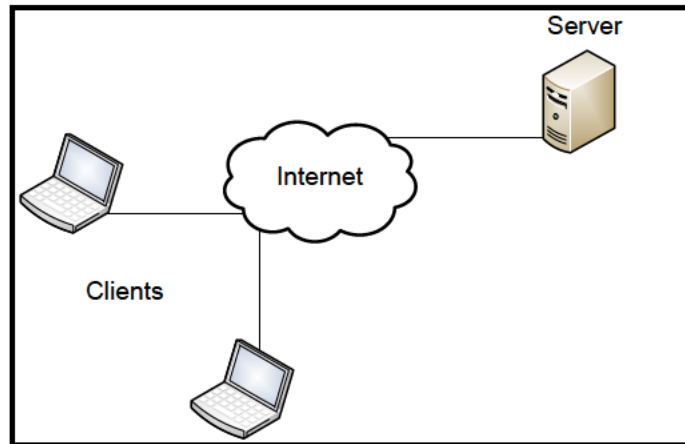


Figure 1.1: Client/Server model

The objective of the development of this web-based user interface is to enable the user to see more swiftly what is happening in the assembly system (process plant) and to provide a more effective problem-solving environment outside the central control room. According to the project operational goal, there are two types of functions that need to be executed; the process monitoring function and the process control function. These two functions are implemented in two pages, the monitoring page and control page. The monitoring page will give a dynamic image that consists of graphic information providing the essential information of the current system status. The control page will give a user a privilege to adjust data if needed. Unlike a normal web page, the dynamic page image is regularly generated by the server according to the system status, sent to the client and it is automatically refreshed after a certain period of time. This provides the client with real-time information about the system.

In order to implement the above-mentioned functions, the server push mechanism is employed. Its basic principle is that the information-sending action is based upon information changes monitored

by the server, rather than the client request. This not only speeds up the client information updating, but also reduces server loading. These functions will be implemented by integrating Active Server Pages (ASP).NET technology and OLE for Process Control (OPC) technology. ASP.NET is a web application framework developed by Microsoft to allow programmers to develop dynamic websites, web applications and web services. This framework supports all .NET languages such as C#, Java, C++ and Visual Basic. The preferred programming languages for developing this system are Java and C# because of its simplicity and because of experience in working with them. OPC is a software interface standard developed by the OPC foundation to allow Windows programs to communicate with industrial hardware devices [6]. OPC XML-DA is a specification preferred in this project as it uses the eXtended Mark-up Language to transfer plant data through the internet.

Because of the vulnerability of the internet to virus and other malicious attacks, several security mechanisms are used to enhance the system security, identity (ID) authentication and allowance of only one user to access the system at any time and real time data encryption.

- ID authentication and authorization: The authentication is accomplished by verifying the ID and the password. When a user asks to login, the system will compare inputted ID and password with user information stored in the database to check the user's identity. Authorization is accomplished by verifying that the authenticated user has permission to access particular resources.
- Allowance of only one user to access the system at any time. In order to avoid the system from being controlled by more than one user at any time, the system will provide a mechanism that will allow only one user at any time to have control function.

- Real-time data encryption. The existing data encryption algorithms are not appropriate for an internet-based real-time system as they take too long to encrypt and decrypt data. The combination of the Advanced Encryption Standard (AES) and secure sockets layer is proposed.

1.6 Project Outcome

The main objective of this project is to develop software for Web-based RCMS. Other objectives are as follows:

- Provide remote system operator with a dynamic, user-friendly Graphic User Interface (GUI).
- Ensure high-quality system security and integrity.
- Provide secure remote control of the system.
- Provide remote monitoring of the system.
- Alarm acknowledgement and reset.
- Request data and reports.

1.7 Outline of Thesis

The thesis is structured in to six chapters as follows:

Chapter I: Introduction

In this chapter, the background of the project is reviewed and following components are discussed:

- Aim of the project - the purpose of this study is reviewed.

- Problem Statement - the existing problem, which is the main reason this study is conducted, is explained.
- Hypothetical Result - the expected result of the study is discussed
- Methodology- the proposed method of implementing the project is explained.

Chapter II: Literature Review

In this chapter, the relevant concepts related to the existing problem are defined, namely Web technology, OPC technology, Information technology and web securities.

Chapter III: Methods and Techniques

In this chapter, the concepts and techniques regarding the methodology that was followed to obtain an implementation of a remote monitoring and control system is explained. Unified Modelling Language is presented as the modelling language used in this research study.

Chapter IV: Design and Implementation

In this chapter, the implementation of the proposed system is divided into two designs - the architectural design and the comprehensive design - and each design is described.

Chapter V: Results

In this chapter, the output of the project implemented is reviewed. The operation of the system is explained.

Chapter 6: Conclusion

In this chapter, the major findings, problems encountered and contributions from the research is summarized. Some future work in this field is also pointed out.

2. CHAPTER II

Literature Review

2.1 Introduction

In Web-based Remote Monitoring and Control Systems (RMCS), the Master Terminal Unit (MTU) is implemented as a distributed web application. For that reason, it is essential to introduce the basic concepts of this type of applications. These concepts are illustrated here in this chapter, which includes Web technology and OPC technology. System security is also discussed within this chapter.

2.2 Distributed Web Application

Nowadays, web technologies are gaining increased importance in automation and control systems. However, the choice of web technologies depends on the use cases in the application environment. Especially in automation and control systems, the data can be read not only from many different field systems and devices, but also from different OLE for Process Control (OPC) Servers. Current OPC Client might be able to read simple data from OPC Server, but there are some problems to get structured data and to exchange structured information between collaborating applications.

Therefore, OPC Foundation has defined interfaces like OPC XML-DA (OPC XML Data Access) and OPC Complex Data that aim to solve those problems. The OPC XML-DA can facilitate the exchange of plant data across the internet and upwards into the enterprise domain [6].

A Web application is an application that can be accessed by the users through a Web browser or specialized user agent. The browser creates Hypertext Transfer Protocol (HTTP) requests for specific Uniform Resource Locators (URLs) that map to the resources on a Web server. The server renders and returns HTML pages to the client, which the browser can display. The core of a Web application is its server-side logic [7].

An application is a computer program that solves a particular problem or related set of problems [5]. A simple application runs in a single process space and often loads in utility, or helper, functions through dynamic-link libraries, which helps the application to achieve its task. A typical application that interacts with a user consists of three elements: presentation, application logic and data services.

Each of these elements (or services) has its own attributes as shown in the Table 2.1 [8]. Presentation, also known as the user interface (UI), focuses on interacting with the user. Application logic, or business rules, perform calculations and determine the flow of the application. Business rules are constraints, usually self-imposed, that companies or organizations use to help them operate in their particular business environment - essentially they encompass those practices and policies that define an organization's behaviour.

Business rules often define a baseline for application requirements and provide guidance to the developer. In practical terms, these business rules are goals that developers strive to meet for their applications. Data service manages information by storing data and providing data-related

functionality. For example, a MySQL running on a Linux Server computer would be a data service [8].

Table 2.1: Service Attributes of each Application Element

Service Type	Service Attribute
Presentation	Presentation of information and functionality, navigation, and protection of user interface consistency and integrity.
Application Logic	Shared business policies, generation of business information from data, and protection of business integrity.
Data Services	Definition of data, storage and retrieval of persistent data, and protection of data integrity.

2.3 Types of Applications

According to the combining or separation of the application elements in logical layers, the application could be classified according to single tier, two tier and multiple tiers [5]:

2.3.1 Single-tier Application

In a single-tier application, only one layer supports the presentation, application and data service. Only one application or application element processes all three of these services. The data itself can be physically stored in any location, such as on a server. However, the functionality for accessing data is part of the application [8].

2.3.2 Two-tier Application

Two-tier, or standard client/server applications, group presentation and application logic components on the client machine and access a shared data source using a network connection. In a two-tier application, the user interface and business rules are a single layer that runs on the client computer. Separate applications, such as MySQL or Oracle database servers, provide the data services. Client/server applications are often two-tier applications, such as in a Java application that calls a MySQL stored procedure to provide data to the application. The Java application is one layer and the MySQL data service is another layer [8]. The classical two-tier architectures have brought efficiencies to business, but there are also some limitations:

- *Monolithic client applications* - Two-tier applications tend to have monolithic client-side components, which prevent incremental improvements (upgrades and bug fixes) to the application.
- *Difficult to scale* - Application scaling is poor because of the limited number of database connections available to clients. Connection requests beyond this limit are simply rejected.
- *Difficult to maintain* - It is hard to maintain client-side application logic because it has to be deployed to every client. Any change in the logic must be redistributed to all clients.
- *Compromised confidentiality* - Application logic on the client potentially exposes business rules to users.
- *Difficult to use broadly* - It is difficult to use two-tier application logic broadly, because applications are bound to specific database systems and table formats.
- *Tightly bound to data source* - The client is often configured for a particular database, so moving data to a different database is more difficult.

- *Poor network performance* - A network runs inefficiently because of the amount of raw data that is transferred across it. Much of the database processing is not localized.

2.3.3 Multi-Tier

Tiers represent a logical concept. The three tiers are generally described as user (first), business (second or middle), and data (third); however, there can be more than three tiers in a multi-tier application. Because of this fact, multi-tier applications are sometimes referred to as n-tier applications where n is any number greater than or equal to three. A service is a unit of application logic that implements operations, functions, or transformations that are applied to objects [8].

In multi-tier architectures, presentation, application logic, and data elements are conceptually separated. These tiers do not necessarily correspond to physical locations on the network. For example, all three tiers may exist on only two machines or they may be deployed on five [8]. Presentation components manage user interaction and request application services by calling middle-tier components. Application components perform business logic and make requests to databases [8]. With multi-tier applications, the client provides only one layer: the user interface. The business rules are performed on a system between the user interface and the data storage system. This allows the presentation services, user interface, business rules, and database to reside separately, as illustrated in Figure 2.1 [8].

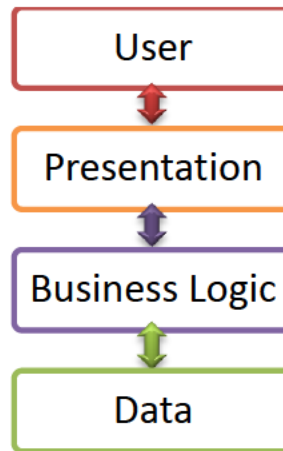


Figure 2.1: User Interface, Business Rules, Presentation Rules and Database Resides Separately

2.4 Elements of a Distributed Application

Applications could be viewed as being separated into presentation, business rules, and data services, and each application could be built as a set of features or services that are used to fill consumer requests. When an application is modelled as a collection of discrete services, the application's features and functionality could be packaged for reuse-shared among multiple applications, and distributed across network boundaries [9].

Three-tier architectures are often called server-centric, because they uniquely enable application components to run on middle-tier servers, independent of both the presentation interface and database implementation. The independence of application logic from presentation and data offers many benefits [9] :

- *Centralized components:* Components could be centralized for easy development, maintenance, and deployment.

- *Load balancing*: Application components could be spread across multiple servers, allowing for better scalability.
- *More efficient data access*: Database connection limitation problem is minimized since the database now sees only the application component, not all of its clients. Also, database connections and drivers are not required on the client. Database connections in two-tier applications are acquired early and held; in three-tier applications, they are acquired late and released.
- *Improved security*: Developers can secure middle-tier application components centrally by using a common infrastructure. Developers can grant or deny access on a component-by-component basis, simplifying administration.
- *Simplified access to external resources*: Multi-tier application simplifies access to external resources, such as mainframe applications and other databases.

2.4.1 Presentation Tier (User and Presentation Tier)

The Presentation tier of a distributed Web application is usually implemented using Hypertext Mark-up Language (HTML) because it is a standard language of all Web browsers [8].

HTML is a text-based mark-up language. It is a simple language for formatting documents that are displayed in a Web browser. The primary task of the browser is to render documents according to the HTML tags they contain and display them on the monitor. HTML pages provide interaction with user in two ways; allowing the user to jump from page to page through hyperlinks. The other way allows the user to send data to the Web server using the HTML Forms.

The Web browser does not know how the server processes these data. It only sends the data using the HTTP request and gets a response back from the server. The browser renders the response as a normal HTML document. It does not care how the server had generated it [8]. From the Web browser perspective, it only sends the HTTP request for the Web server. The request could be a name of an HTML document or a name of a server side application with some data sent by the user using HTML Forms. The Web browser then expects an HTTP response. The response should be an HTML page which is rendered by the browser [8]

2.4.2 Middle Layer (Business Logic Tier)

In distributed Web applications, the middle tier is responsible for [5]:

- Receiving the requests from the user presentation layer.
- According to the request and application logic, the middle tier performs specific tasks; for example, read/update a database, send an e-mail or consume a Web service.
- Format the response in a human-readable fashion using HTML and send it back to the user.

In this sense, it should be noted that the middle tier in fact generates the presentation layer [8].

There are many languages and platforms that could be used to implement the middle tier. One of the most obvious choices is the Hypertext Pre-processor (PHP). PHP is an open-source, server-side, Web-scripting language that is compatible with all the major Web servers (most notably Apache). PHP makes it possible to embed code fragments in normal HTML pages—code that is interpreted as the pages that are served up to users. PHP also serves as a “glue” language written by and for Web developers [5]. When a PHP script executes, it does not interact directly with the

browser; only the final product of the PHP script, which usually is an HTML document, is dealt with by the requesting browser. If a browser were sent an unprocessed PHP script, the browser would attempt to render the PHP script as regular HTML. Browsers cannot execute PHP scripts [5].

2.4.3 Lower Layer (Data Tier)

This layer refers to the components that manage an application's internal data. These data are typically under the direct control of a Relational Database Management System (RDBMS) like MySQL or Oracle [5].

2.5 OPC Technology in Automated Control Systems

In the automation industry, OLE for Process Control (OPC) has established itself as an industry standard in the last few years and provides a sophisticated, innovative infrastructure of the underlying information and data model [10]. OLE for Process Control is an OPC foundation standard, which is used for communicating among numerous data sources on the factory floor, or a database in the control room. This standard provides a common way for an application to access data within the plant from any data source like a device. OPC has gained increased popularity and trust in the automation industry because of its reliability and interoperability.

OPC was designed to provide a common bridge for Windows-based software applications and process control hardware. Standards define consistent methods of accessing field data from plant floor devices. This method remains the same regardless of the type and source of data. An OPC

Server for one hardware device provides the same methods for an OPC Client to access its data as any and every other OPC Server for that same and any other hardware device. The aim was to reduce the amount of duplicated effort required from hardware manufacturers and their software partners, and from the Supervisory Control and Data Acquisition (SCADA) and other Human Machine Interface (HMI) producers in order to interface the two. Once a hardware manufacturer had developed their OPC driver for the new hardware device, their work was done to allow any 'top end' to access their device, and once the SCADA producer had developed their OPC Client, their work was done to allow access to any hardware, existing or yet to be created, with an OPC compliant server.

OPC interoperability means the ability of computer systems to run application programs from different vendors and to interact with each other across the local or wide area network regardless of their physical architecture and operating system, as shown in Figure 2.2. Automation industries today cannot begin to think of remote monitoring and control of systems without the use of OPC [8, 9, 10, 11]; OPC technologies adopt the client/server architecture, where client is dedicated to reading and writing data of the system through the server.

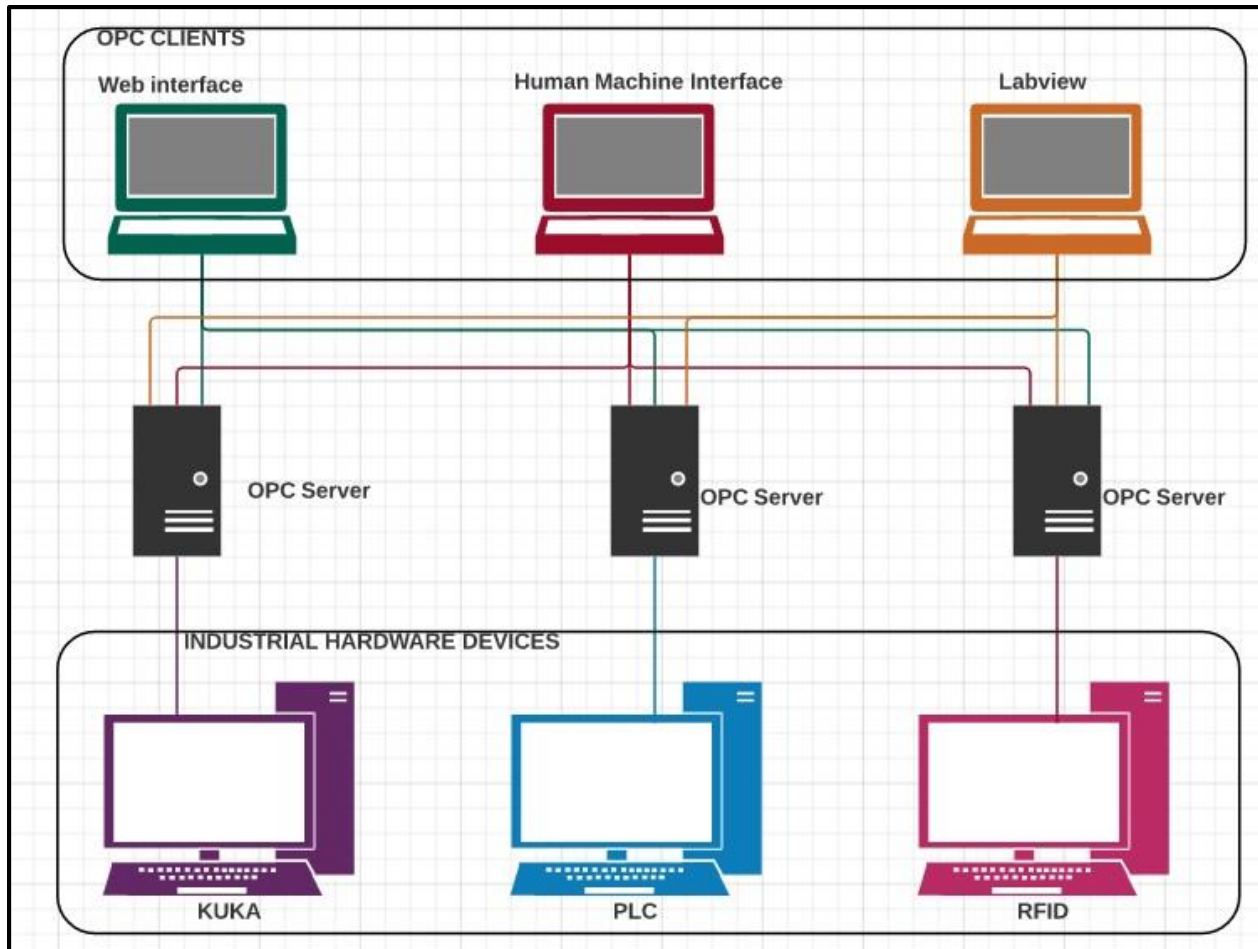


Figure 2.2: OPC Connectivity

OPC is designed to allow client applications access to plant floor data in a consistent manner. With wide industry acceptance OPC provides many benefits [10]:

- OPC solution minimizes device drivers. Hardware manufacturers only have to make one set of software components for customers to utilize in their applications.
- Software developers will not have to rewrite drivers because of feature changes or additions in a new hardware release.

- Customers will have more choices with which to develop World Class integrated manufacturing systems.
- OPC drivers are readily available.
- OPC reduces device loads significantly.
- Implementation time and all costs are drastically reduced.

2.5.1 OPC Specifications

The OPC Data Access (DA) is based on DCOM, which is the foundation of Object Linking and Embedding (OLE), later named ActiveX and the XML DA specification is based on the web service standards, XML, Simple Object Access Protocol (SOAP) and Web Service Description Language (WSDL) and standardizes the SOAP messages exchanged between the client and server [7].

At present, devices are based on OPC DA server in most enterprises. Redevelopment of OPC XML-DA server is not realistic, which is bound to result in a waste of resources and development difficulties are also relatively large [12]. An OPC server based on COM/DCOM is currently running well and would not immediately be eliminated by an OPC XML-DA server. Since the OPC DA specification is based on COM/DCOM technology it has some limitations, which are as follows [11]:

- It is only for Windows platforms.
- DCOM can be used for the application over the internet, but firewall authentication is not easy to resolve.

Data exchange between devices on the plant floor and enterprises application is an issue that needs to be handled. The OPC XML-DA specification uses the web services standards in order to exchange data, which has the following advantages [11]:

- Web Services are platform-independent and language-independent, since they use standard XML languages. This means that my client program can be programmed in C++ and running under Windows, while the Web Service is programmed in Java and running under Linux [13].
- Most Web Services use HTTP for transmitting messages (such as the service request and response). This is a major advantage if you want to build an Internet-scale application, since most of the Internet's proxies and firewalls will not mess with HTTP traffic, unlike Common Object Request Broker Architecture (CORBA), which usually has trouble with firewalls) [13].
- Web services use text-based protocol that all applications can understand. Firewalls do not block text information, so the popular way to represent data on the Internet is XML [14].
- Web services reduce licensing costs.
- Web services do not rely on special protocols.

2.6 Review on the Design of the current Web-based RMCS

There are different ways in which data can be accessed from the automation and control system using OPC protocol. The utility of OPC has now reached a point where automation without OPC is unthinkable. It provides a mechanism to provide data from a data source and delivers the data to any client application in a standard way. OPC is based on the DCOM/COM component-object programming model developed by Microsoft in which software is divided into smaller independent

units (the objects). Web-based RMCS uses the internet to transfer data to the Remote Terminal Unit (RTU) and the Monitoring Terminal Unit (MTU) and /or between the operator's workstation and the MTU [15]. This will reduce the cost of the installation of a SCADA network if compared with installing a dedicated network [11].

Many researchers worldwide tried to design and to implement an approach to access an OPC DA server through the internet to realize a web-based RMCS. DCOM is suitable for LANs where there are less interruptions and noise, but when used through the Internet there will be some limitations related to its nature. For this reason, researchers tried to use Information Technology (IT) services to achieve their goals. In the following sections, an overview of the work in this area is presented.

Most previous research used DCOM for communication with the OPC DA server through LANs. For example, Xiaofeng Lee uses communication between the OPC DA client and the OPC DA server, and then they transfer the OPC DA client data in XML format to be able to access this through the Internet with an XML-DA client that communicates to the XML server (Web server) to get data, as shown in Figure 2.3 [16] .

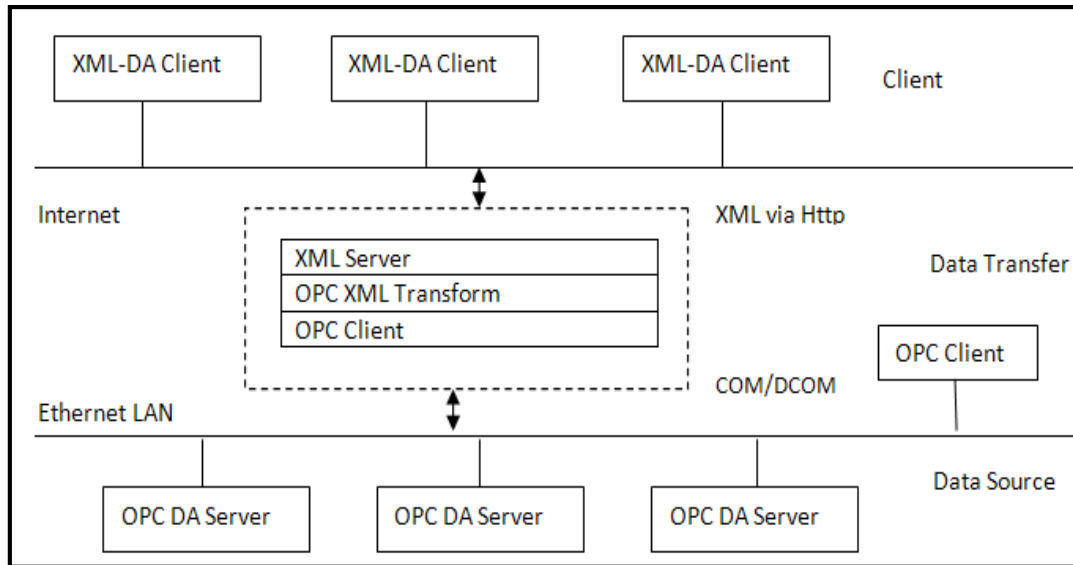


Figure 2.3: Xiaofeng Lee Design model

Ju Chen also uses DCOM to enable C# server script to access OPC DA through the LAN and because the OPC DA client is a .NET client, they use an OPC .NET wrapper to make the transformation from .NET to COM and COM to .NET as shown in Figure 2.4 [16] [17]. From this discussion, it can be concluded that the only way to communicate directly to the OPC DA server is DCOM (or COM if the client and server are on the same machine).

Unfortunately, using DCOM through the Internet is avoided for many reasons, such as [16]:

- DCOM is a Windows-dependent platform.
- Difficult to configure and cannot be used for Internet communication.
- Has very long and non-configurable timeouts.

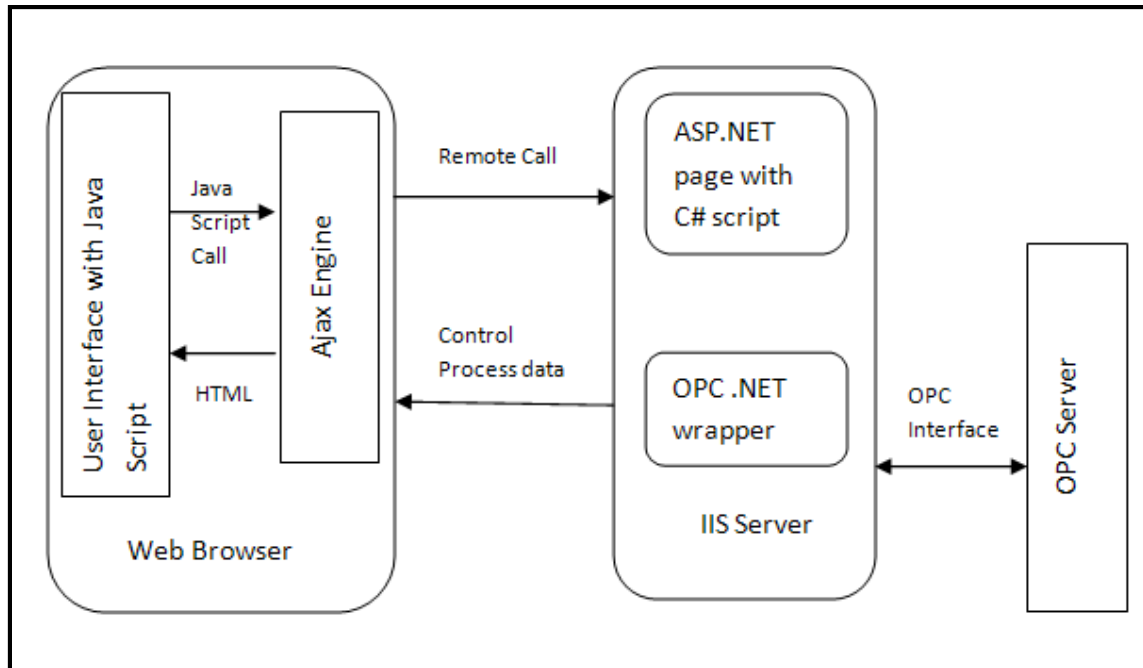


Figure 2.4: Ju Chen Design Model

2.7 OPC XML-DA

XML is a platform-independent, which is an important feature to achieve interoperability between different applications, which are running on different platforms. This is the reason that forced the OPC FOUNDATION to release the OPC XML-DA specifications to allow the XML applications to access OPC data in a standard way. The other advantage of OPC XML-DA is the simple administration as it is based on SOAP and XML. On the other hand, it has some disadvantages - such as [11]:

- Not suitable for transferring large data volumes.
- XML technology is generally slower than COM.
- The interaction parameters coded using XML, which leads to an overhead.
- An OPC XML-DA Service is stateless.

According to [6] the OPC XML-DA is a standard Web Service interface for reading and writing data from and to plant floor automation systems. The OPC XML-DA data model is based on OPC Items that are named and organized in a hierarchy. Each item stores a single value. It is defined with the combination of two strings: item path and item name, the “item path” identifying a namespace in which the “item name” is unique. A set of dynamically retrievable properties is associated with every item containing its metadata including human readable description, access rights, a timestamp, change rate, engineering unit and data type. The possible data types are simple type, enumeration or arrays.

Operations for accessing item values are Read and Write. Both allow accessing several items with a single call. Each item’s path and name is contained in requests to these operations. To optimize periodic reads of the same set of items, a subscription mechanism is provided. The set of items is subscribed by calling the operation *Subscribe* and then periodically polled using “*SubscriptionPolledRefresh*”. The OPC operation *Browse* and “*GetProperties*” are used to query which OPC Items are available, and retrieve values of the properties [6].

Browse allows querying an OPC Item’s immediate successors including filtering and can return property values of the items found. Property values can also be retrieved with “*GetProperties*”. The operation “*GetStatus*” is used to retrieve the status of the OPC Server. The OPC XML-DA provides better connectivity and interoperability for production management and enterprise applications such as Manufacturing Execution System (MES), Enterprise Resource Planning (ERP), Computerized Maintenance Management System (CMMS), Enterprise Asset Management (EAM) and plant optimization that need to access plant floor data [6].

Many applications are running on non-Microsoft computer platforms that do not have built-in support for the COM/DCOM interfaces used with the OPC DA. The OPC XML-DA is complementary with products based on the existing OPC DA specification. The OPC XML-DA was specifically designed to allow existing OPC DA COM based products to be wrapped by the OPC XML-DA interface and in effect support both interface from the same OPC Server. Any group can develop a generic OPC XML-DA wrapper to internet-enable existing OPC DA Servers, allowing them to publish plant floor data to the Web.

2.7.1 SOAP Protocol

Simple Object Access Protocol (SOAP) is a simple and lightweight XML-based protocol to let applications exchange structured and typed information over the Web. It consists of three parts: an envelope that defines an overall framework for describing what is in a message; who should deal with it and whether it is optional or mandatory; a set of encoding rules defines a serialization mechanism for exchanging instances of application-defined data types, and a SOAP Remote Procedure Call (RPC) representation defines a convention for representing remote procedure calls and responses. The SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages [6].

2.8 Model of Web Integration

Web integration allows users to access data and functions, via an internet browser, of an application hosted on a web server [18]. Web-based interface(s) are used to deliver multiple

applications remotely. This can reduce cost and provide users with a browser-based working environment. A general architecture of web integration consists of three layers, as shown in Figure 2.5 adapted from [19]. The lower layer provides information from automation devices to the controller level of the automation system.

The upper layer (user) is based on standard IT technologies, such as a client/server model using a web server as data source and web browser as clients. The middle layer (business logic) contains functionalities and business logic and performs as an application gateway between the upper client and the lower automation systems. The lower layer refers to the components that manage an application's internal data. The web server can be used to assign information from the automation and control systems to object models that can be accessed via the Components Object Model/Distributed Components Object (COM/DCOM) Model between the web applications, because data have different types of semantic meaning [6].

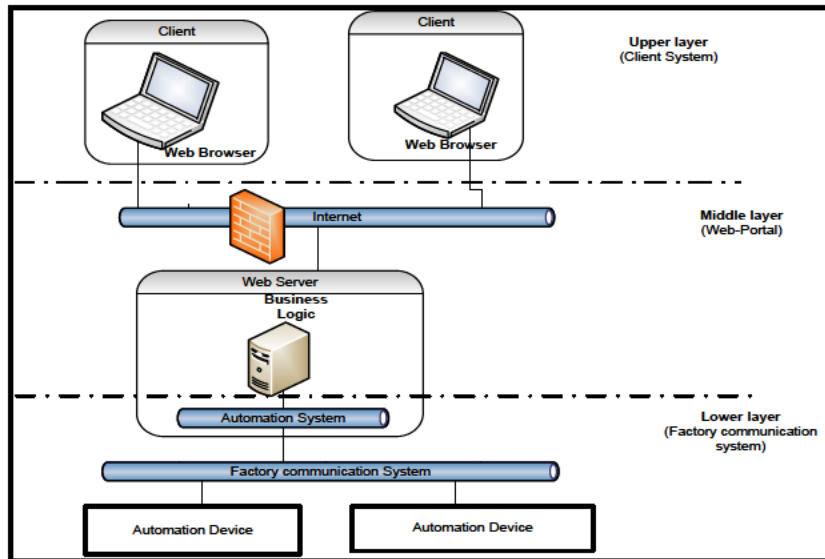


Figure 2.5: Generic Architecture of web integration for process monitoring systems

Using web technologies for slow control systems, it means integrating a multitude of different technologies. In Figure 2.5, the XML-DA services are implemented in the middle layer to provide web services for the client to invoke.

2.8.1 Benefits of Web Integration

Web integration has a number of benefits compared to other traditional integration approaches [20]:

- *Lower cost:* Using web integration, the effort required becomes much less expensive compared to traditional integration. There are numerous reasons, including lower skills for developers, no change to existing applications, and no infrastructure changes to network.
- *Non-Intrusive:* Web Integration is done non-intrusively, thereby lowering the risk and impact of the entire integration project. Because there are no architectural changes

required, it is often easier to justify cross-enterprise projects. This benefit extends even further to those external applications where the user interface is the only available option for integration.

- *Faster Development:* Since the browser interface is well understood by both the end user and the programmer, application design becomes much easier and less prone to error.
- *Faster overall integrations:* Even complex Web Integration projects can be completed in weeks rather than months. Companies can gain competitive advantages by leveraging their existing enterprise applications more quickly than their competitors.
- *Lower skill requirements:* A traditional integration project requires highly skilled development staff. In-depth knowledge of the applications and application integration techniques is required. With Web Integration technologies, development personnel with basic programming experience and web-application knowledge can do a superior job. The need for high-skilled expensive programmers can be obviated.
- *Potentially lower risk:* Web Integration allows for shorter and more cost-effective implementation cycles. The initial integration can often be up and running quickly and further integration can be accomplished once results from the initial integrations have been proven. This allows companies to try out new business opportunities at lower risk than using traditional methods.

2.9 Web services

Any organization today works with multiple vendors, suppliers, contractors and other entities. Each of these entities would have developed their own software systems based on Microsoft technologies or on Sun Microsystems or on International Business Machines (IBM) technologies.

Each of these software systems would have been developed over a period of time with hundreds of thousands of rands investments. It will be almost impossible for any of them to change their systems for compatibility.

Communicating amongst all these entities without affecting their existence is made possible by web services [14]. A Web service is a method of communication between two electronic devices over the web (internet) [21]. Web services communicate using the World Wide Web (WWW). They rely on the standard Internet protocols HTTP and SOAP which are present in every system. Web services transport their messages using HTTP, which means these messages are transmitted over port 80, an open port for web server firewalls. Web service messages are transmitted as SOAP-formatted messages. SOAP messages are in XML format, meaning they are simply text, and not binary data. All the systems (every computer in the company) provide native support for web. And all computers have inherent support for XML. Any application written in any language running on any platform can process XML data [14].

2.9.1 Implementation of Web services

According to J.Y Vun Van Tan [22] in order to design and implement a web service, firstly a module is presented, which is the OPCXML-DA (Data Access) module; it is used to manage the complex data, server, and address space (e.g., the OPC Groups and OPC Items, etc.) and is based on the OPC XML-DA Specifications [22]. The OPC XML-DA module is fully aggregated by three classes such as Data Management, Server Management, and Address Space Management. When a client wants to achieve data or events from the hardware I/O devices on the plant floor or from

OPC Servers, it should first know the address of its server, and then a connection between the client and server is established including the information of server, endpoints (i.e., physical objects, OPC Servers, software objects, etc.), session, and secure channel. When this connection is created successfully, the client can monitor and acquire data or events from the plant floor or through OPC Servers, and can also write data to hardware I/O devices on the plant floor.

2.9.1.1 Data Management

The data management class manages a data buffer. It consists of some functions to refresh data, the value and timestamp of OPC items and to check availability of data. This class uses the OPC Complex Data class that provides the methods to represent and convert the complex data from the hardware I/O devices. Complex data types are defined as dictionaries in Web Service Description Language (WSDL) files [22]. Hence, each OPC item is defined as an OPC Complex data item.

2.9.1.2 Server Management

The server management class contains several functions to accept new connections, manage existing connections, manage connected clients, etc. It initializes field bus information, gets all parameters, and handles client requests. When a request is received, it will connect to an OPC Server and responds to the OPC Clients regarding all kinds of requests such as browse address space, add/delete item groups, add/delete items, and read/write, data subscription, data refresh, etc. [10].

- **Status** (parameters—GetStatus and GetStatusResponse): Allows the clients to query the current status of the server.

- **Read** (parameters—read and Read-Response): Allows the clients to read the values from items.
- **Write** (parameters—write and WriteResponse): Allows the clients to write the values to items.
- **Subscription** (parameters—SubscriptionRequest and SubscriptionResponse): Allows the clients to tell the server to monitor changes on a set of item values.
- **Request and StreamStopRequest**: Allows the clients to start receiving the server stream with any data changes.
- **Subscription Cancel** (parameters—StreamSubscriptionCancelRequest and StreamSubscriptionCancelResponse): Allows the clients to tell the server to stop streaming item values for changes.
- **Browse address space** (parameters—BrowseRequest and BrowseResponse): Allows the clients to identify what items exist in the server.

2.9.1.3 *Address Space Management*

The address space management class performs basic functions of the OPC XML-DA, such as add/delete item groups, add/delete Items, read/write and subscription. [23]. The Web Service allows the OPC XML-DA Clients to read and decode any type of data from the OPC Servers or measurement and control systems (i.e., hardware I/O devices). Web Server runs in multi-task (i.e., multi-processor and multi-threaded) pre-emptive system and it gives a firm support for all HTTP (Hypertext Transfer Protocol) versions. In order to design and implement an efficient Web Service for supporting several kinds of OPC Clients that read and decode any type of data from the hardware I/O devices and perform high speed data exchange in DCS (Distributed Control System),

the aspects of Web Service implementations can be represented as shown in Figure 2.6 adapted from [24]. These aspects indicate that the proposed system supports not only browser-based clients, but also application-based clients.

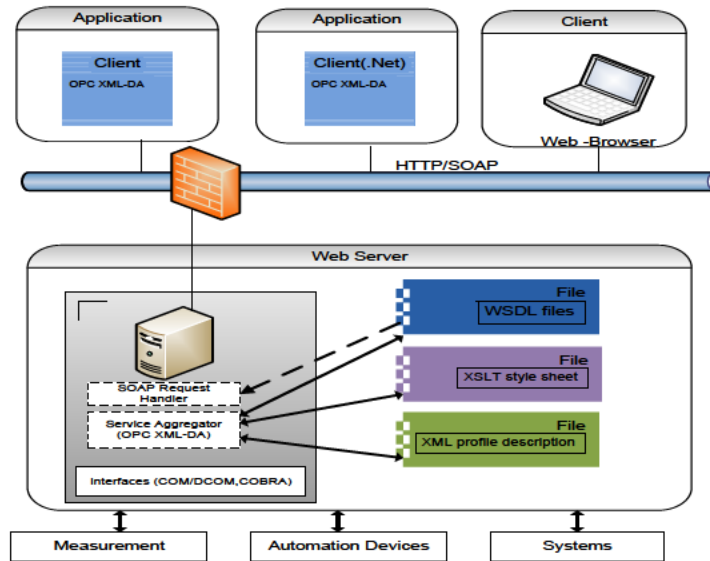


Figure 2.6: The aspect of web service implementation

Communication between the OPC Clients and Web Service should provide some mechanisms to monitor and control the hardware I/O devices, sending web pages to clients and dealing with the submissions of the clients [24]. This Web Service should also provide security interface for remote clients.

2.9.2 Limitations of Web services

- Overhead - web services use XML, so transmitting all data in XML is not efficient as using proprietary binary code. A critical real-time application never uses web services.

- Lack of versatility - Currently, Web Services are not very versatile, since they only allow for some very basic forms of service invocation. CORBA, for example, offers programmers several supporting services (such as persistency, notifications, lifecycle management and transactions). Fortunately, there are many emerging Web services specifications (including WSRF) that are helping to make Web services more and more versatile [13] .

2.10 OPC Security Issue

Security is becoming an important issue in the automation industry due to the collaboration of business networks and process networks [9]. Devices that are connected via such communication systems and Programmable Logic Controllers (PLCs) can be easily programmed via a communication network. Data and events from industrial systems are sent to operator systems and enterprise systems via the Ethernet and Internet. The problems with malicious network attacks will be carried into the process area, and viruses and worms will corrupt the system from PLC to field devices. For the classic OPC standards (e.g., OPC Data Access), an OPC server might implement one of three levels of security [5]:

- Disabled security - no security.
- DCOM security - launch and access permissions to OPC servers are limited to selected clients.
- OPC security - the OPC server serves as a reference monitor to control access to vendor specific security objects exposed by the OPC server. The OPC security specification covers only server/object access control, but is not concerned with confidentiality and integrity during transmission. However, additional security settings as well as defined in the OPC security

specification have been implemented in only a few products on the market [18]. The OPC UA standard has a scalable security concept based on W3C standards and includes user authentication, digital signatures, and encryption for the exchange messages [5] [18]. Implementation, evaluation, and performance depending on this standard are now a challenge for researchers and developers.

2.11 Conclusion

A number of conclusions have been considered based on the literature survey and the following are considered to be the most important ones:

1. The MTU of a Web-based RMCS is implemented as a distributed web application. This means that in order to develop the proposed system, it must be implemented as distributed web application.
2. The utility of OPC has now reached a point where automation without OPC is unthinkable. Currently, most of the automation system companies have adopted the OPC protocol as their communication protocol because of its interoperability.
3. Real-time data monitoring control over the internet it is still an issue due to the web load time. Although there are a few researchers who have attempted to solve the issue by using mechanisms such as data compression and web refresh, they did not pay attention to the Central Processing Unit (CPU) load of the web server and the network bandwidth; until now this still remains an issue which should be addressed when working with RMCS.
4. To avoid compromising the integrity and the security of the RMCS, strict security measures should be undertaken.

The next Chapter discusses the software modelling of the proposed system.

3. CHAPTER III

Methods and Techniques

3.1 Introduction

In this chapter, the concepts and techniques that influenced the methods used to implement a remote monitoring and control system are explained. A brief description of the assembly system within the Research Group in Evolvable Manufacturing Systems (RGEMS) is presented. Unified Modelling Language (UML) is used to model this software product.

3.2 System Description

The assembly system at RGEMS aims at assembling a product part by part, ensuring that good quality products are produced and cost effectiveness is maintained. This system is composed of different industrial devices which are a put together in order to form an efficient automation system. These devices communicate with each other, exchanging data using the Object Linking and Embedding OLE Process Control (OPC) protocol and TCP/IP protocol. The OPC protocol creates a platform for different industrial devices from different manufactures to communicate or exchange data without the need of special driver software being installed [25]. This protocol uses client/server communications, which are the OPC client and OPC server. The OPC server is used to store data of the device connected to the system in real-time, allowing the OPC client to read the data of the system by connecting to it. OPC configuration for the system at RGEMS is seen in Figure 3.1. According to [26] the goal that had to be met concerning OPC at RGEMS was:

- Compatibility

- Security
- Simplicity.

As seen from Figure 3.1 the OPC configuration in the production environment is constructed of components from various software vendors and uses many different technologies [26].

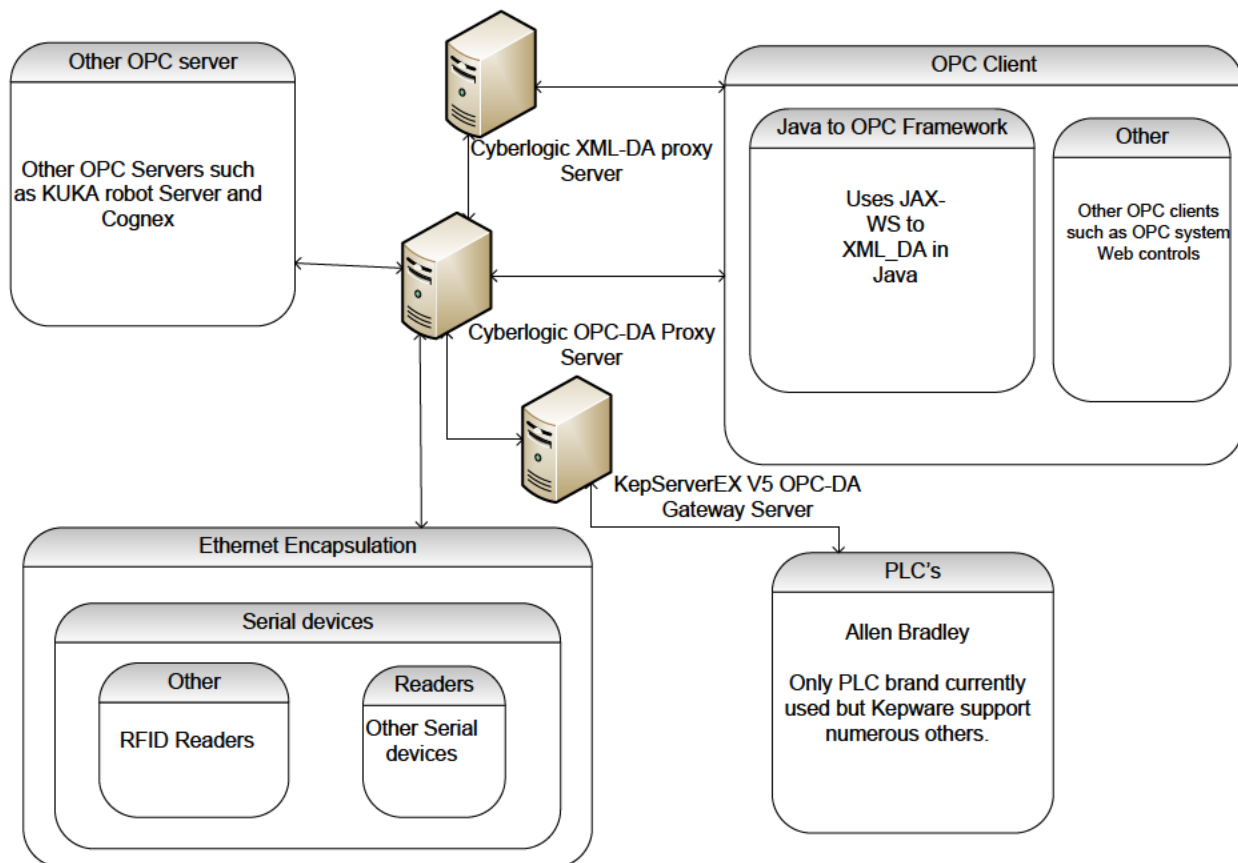


Figure 3.1: RGEMS OPC Configuration

The assembly system at RGEMS is divided into different sub-systems and each sub-system is responsible for a certain task contributing to the full production cycle, as seen in Figure 3.2. The system is structured as follows:

- *Feeder System* – This system is used for feeding the production line with products every time there is a need for product assembly. This system is controlled by the Programmable Logic Controller (PLC).
- *Gantry Robot System* – This system is rail driven and used for picking and/or packing objects. This system is controlled by the PLC.
- *Object Sensing* – This system is responsible for object detection. The system uses industrial sensors for detection and is controlled by the PLC.
- *Machine Vision System* – This system is also known as a quality control system. This is where the object is scrutinized for quality purposes. The system uses smart industrial cameras that are accompanied by a lighting device for high quality object vision.
- *Automated Guided Vehicle* – This system is used for product transportation after the product has been assembled. This system is controlled by microcontrollers.



Figure 3.2: The sub-system of the assembling system

3.3 System Operation

The system operates in such a way that the sub-systems are dependent on each other in order for the system to operate in full cycle. Figure 3.3 describes the workflow of the system operation. The system first gets fed by the parts that need to be assembled to get a product; after that, they are

transported by the conveyor belt to the next sub-system called object sensing; after the object is sensed, a signal is sent to the other subsystem called the gantry system where the next phase of assembly takes place. After that is completed, it is sent for quality control; after quality is verified, it is then transported for packing by the Automated Guided Vehicle.

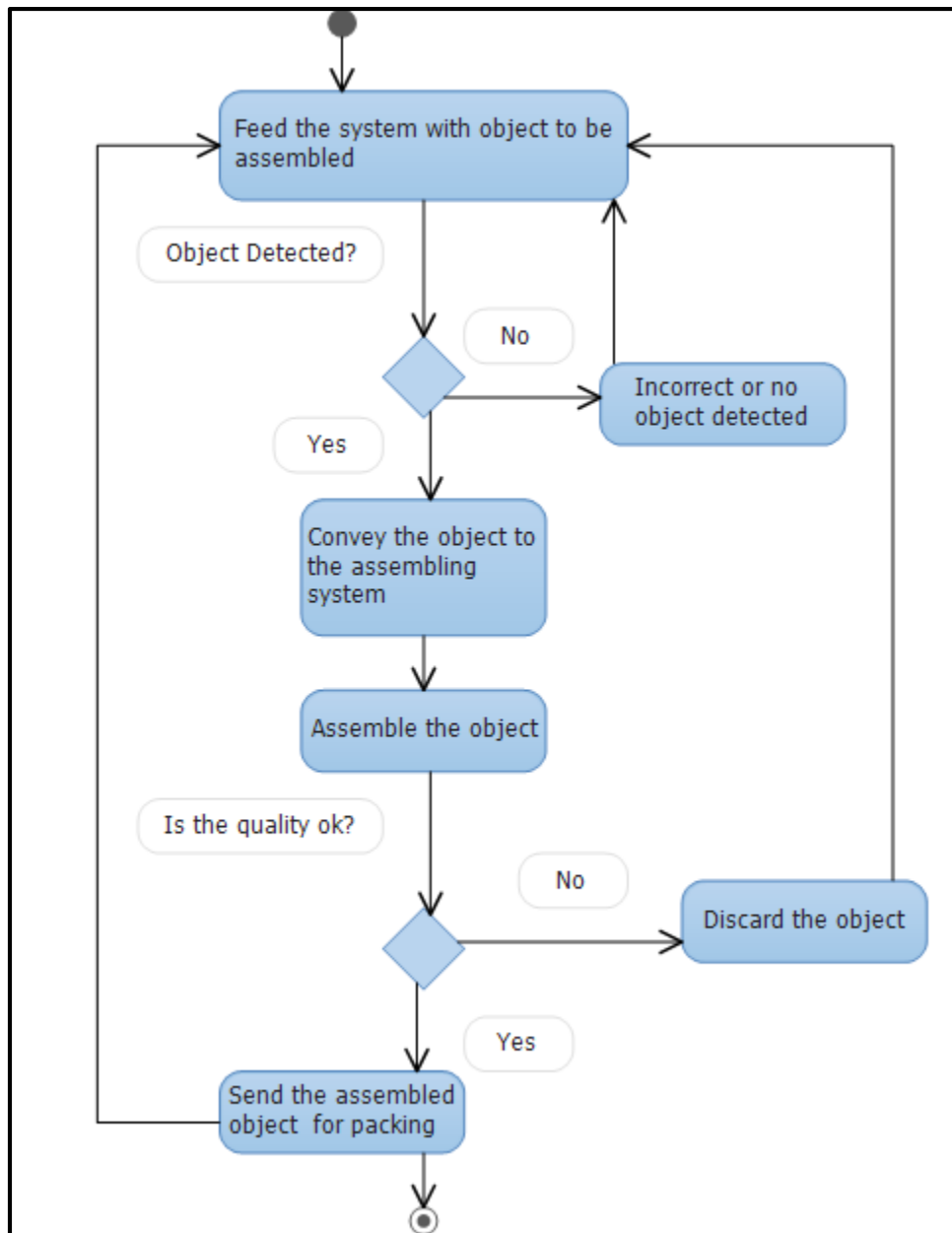


Figure 3.3: Production cycle of the system

3.4 The System Development Life Cycle

A software development process, also known as a Software Development Life Cycle (SDLC), is a structure imposed on the development of a software product [27]. There are several models for this development process that will be used in the implementation and design of this project; each one of these models describe a variety of tasks or activities that take place during the development. The process was chosen to make the implementation and design of the software product much easier, by dividing it into smaller tasks and tackling them one at a time. The first task that was carried out according to the proposed SDLC, is the System Analysis and Requirement followed by Design, Implementation, Integration and Testing, Installation and Acceptance - as seen in Figure 3.4.

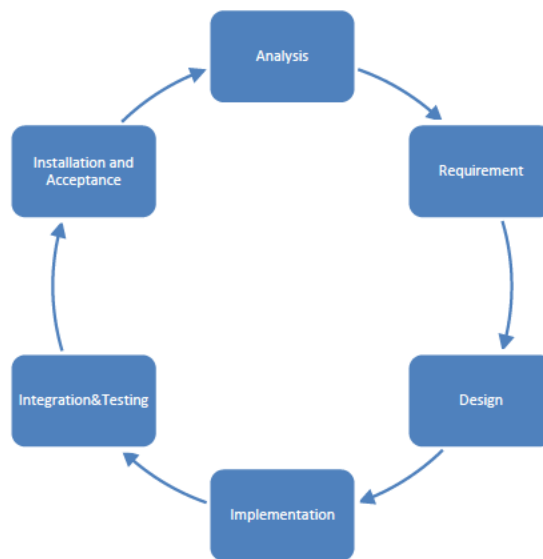


Figure 3.4: Software Development Life Cycle

3.5 System Analysis and Requirements

The first step in developing any application is performing an analysis to define its purpose, scope, and objective. At this stage, the composition of the system had to be analysed and visualised and as much information about the specifications of the system as possible had to be gathered.

Information gathered about the system had to answer the following questions:

- Where will the system be used?
- What are the components needed to build the system?
- How will the system work?
- Who will use it?

Answering the questions involved the UML use case diagrams to refine the project specifications and understand its objectives. The UML specification was designed to describe and document the analysis and designs for the systems that use object-oriented language such as Java. The main construct UML has for describing what an application will accomplish is the “use case”. A use case is a description of something a system does at request of or in response to an action by one of its actors. An actor is the user type or external system serviced or affected by the use case. Although the word actor has connotation of being an actual person, a UML actor can be external system or an organization type or role.

3.5.1 Specification of Remote Monitoring and Control Systems

The specifications of the Remote Monitoring and Control System (RMCS) are organized into a sequence of steps. Each step approaches the system under a different perspective and increases the level of detail of the system specification. In each step, a set of questions must be answered,

analysing a different view of the remote system to be developed. The steps are organized in questions in order to guide and facilitate its applicability.

I Step 1: Monitoring or Control?

The first step consists of defining the purpose of the remote access system. The first point to be defined is whether the system will have only monitoring functions or also control facilities as well. The answers given at this point may change due to limitations imposed by the characteristics of the assembly system at RGEMS and the available resources. In a remote access system, the information collected from the local manufacturing system is available in real-time in the remote destination. A remote access system is justifiable only when data collected is also processed and used in real-time in the destination. When the real-time data availability is not the issue, a simple solution is to store the data locally and use conventional tools for sharing databases through the internet. The first questions to be answered are:

- What are the advantages of making information about the remote system available in real-time?
- What kind of decision can be taken based on the available data in the remote destination?

The purpose of making the information of the remote system available is to provide the user with the facilities necessary to understand the system behaviour, propose supervisory control strategies and test them. The user must be able to monitor the system behaviour and the execution of events in each module. This is clearly the case when developing a remote monitoring and control system, as the remote must interface with the local manufacturing system.

II Step 2: What type of remote monitoring system is required?

The type of remote monitoring system that is required has to provide the following:

- *A tool that is easy for operators to use:* This refers to the functionality of the system; this system should be user-friendly.
- *A “read to system” privilege for the system operator:* The system operator should be able read data from the manufacturing system in real-time with an automatic refresh.
- *Error Detection:* The system should be able to make use of an alarm control and detect if there is any error on the system and display the error message on the web interface until the system operator acknowledges it.

III Step 3: What are the benefits of making this system remote monitored?

The benefits of a remote monitoring system through the internet are as follows [28]:

- *Web browser accessibility:* The monitoring system will reside in a web server so it will be accessible anywhere and anytime from a standard web browser; there is no need to install the software application on your PC.
- *Cost Reduction:* In case there is a system failure, this can be noticed and be dealt with in time. Acknowledging the failure on time means production will not stop for a long time and also not much damage will be done to the devices of the system, which will curb expenses.
- *Maximize system performance or identify system deficiency:* By monitoring the system, failure can be identified before it causes more problems and this will maximize the performance of the system.

- *Understand the performance of the system through interval data:* By using controls such as trends and alarms, the operator can monitor the performance of the system easily.
- *Just in time maintenance:* Operators are able to identify a problem in time and can call maintenance in time.
- *Early warnings of possible breakdowns:* The operator is able to see if the system is about to have a breakdown this might be a minor deficiency that may lead to a possible breakdown.
- *Real-time production reports with direct data from machines:* real-time data monitoring can be achieved without having to add any new hardware on the system.

IV Step 4: What type of remote control system is required?

The type of remote control system that is required has to provide the following:

- *A tool that is easy for operators to use:* This refers to the functionality of the system; the system should be user-friendly and not complex.
- *An option of emergency shutdown control to the operator:* In case there is a system malfunction, the system operator should be able to use the emergency shutdown control.
- *A write to system privilege:* The authorized system operator should be able to run and stop the system from a web interface without any difficulties.
- *System security and Integrity:* The system should only give the write permission to authorized operators to avoid malicious users.

V Step 5: What are the benefits of making this system remote controlled?

The benefits of a remote control system through the internet are:

- *Flexibility of the system operator:* The authorized operator does not have to be physically at the system every time he/she wants to control it.
- *Web browser accessibility:* The system operator can control the system from anywhere and at any time.

VI Step 6: Specification of use - cases and exchanged information

In order to determine the data to be transmitted between the remote and the local system, the first question to look at is:

- What are the system use-cases and who are their actors?

There are two actors, which are the local user and the remote user. The system can operate in one of the following modes: automatic, step-by-step and testing. Figure 3.5 details the use-case for the system.

This first question is documented using the Use-Case Diagram of the ULM seen in Figure 3.5. Among the points to be analysed when specifying the use-cases is the maintenance of history records, and databases, which can be in the local system or in a remote system. Once the use-cases have been specified, the next step is to make a list of the data needed on the remote processing point to take decision specified in Step 1. The second question of Step 6 is:

- What information is exchanged between local and remote systems?

Answer: The data exchanged between local and remote system; it can be real-time data of the system retrieved from an OPC server and archived data retrieved from a database server.

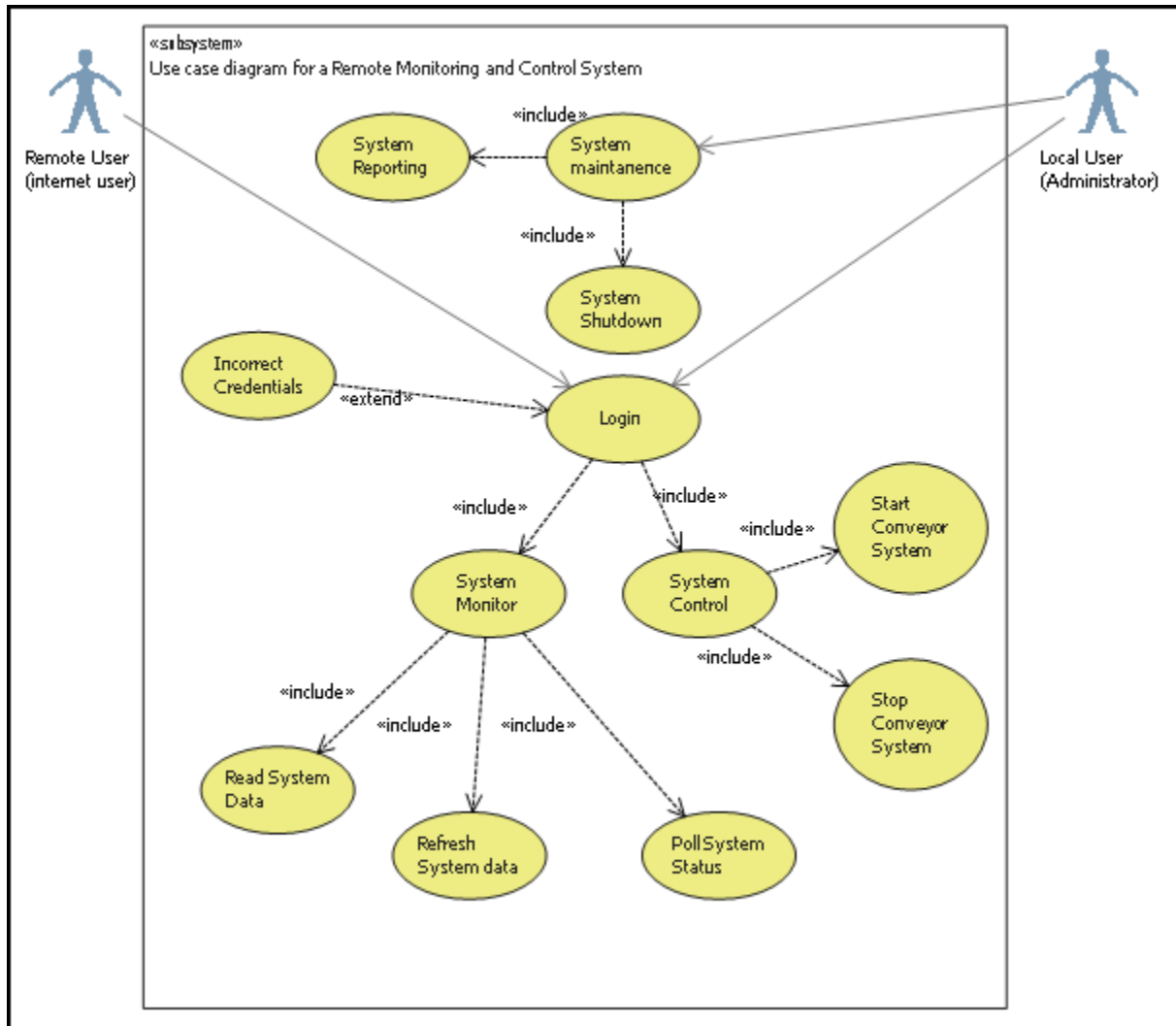


Figure 3.5: ULM Case Diagram for Modelling the System

The first step in use-case analysis is to identify the actors [29]. An UML actor is the user type or external system serviced or affected by the use-case; it can also be an organization type or role [29]. In this case, the actors are: Internet user (system operator) and Administrator (system maintainer).

The local user has priority over the remote user. Although a remote user can always connect to the system and monitor it, he/she can only control the system when local user (administrator) gives

permission. The local user can also remove the permission from a remote user. The information to be transmitted from the local manufacturing system to the remote user is:

- If the remote user is connected or not, if he/she is controlling the system or not.
- The current state of the system.
- A real-time video to understand the system operation and detect problems.

The information to be transmitted from the remote user to local manufacturing system is:

- A request to connect or disconnect from the system.
- The current mode of control chosen by the remote user.
- Commands to start and interrupt the sequence of operation in the automatic mode, perform another step in the step-by-step mode, and perform each operation in the testing mode.

3.6 *System Design*

The classical design phase consists of three activities: architectural design, detailed design, and design testing [27]. During this phase, the project specification undergoes two consecutive design processes; the architectural design and the detailed design. In the architectural design, the project as a whole is broken down into components called modules and each module is designed in detail.

3.6.1 System Architecture

The Web-enabled remote monitoring and control system is designed using a four-tier architecture shown in Figure 3.6 [30] . It consists of 4 sub-systems, which are as follows:

- *Client (Web-user interface)* – The client requests data from the server(s) connected to the assembly system, and displays the requested data in a web interface.
- *Database Server (MySQL Database Server)* - The database server manages all the data of the assembly system.
- *Control Server (OPC Server)* - The control server delivers the request from the client to the assembly system.
- *Web Server (ASP.NET Web Server)* - The web server manages communication among database server, control server, and client. The client can monitor and control the plant on the web using this web server.

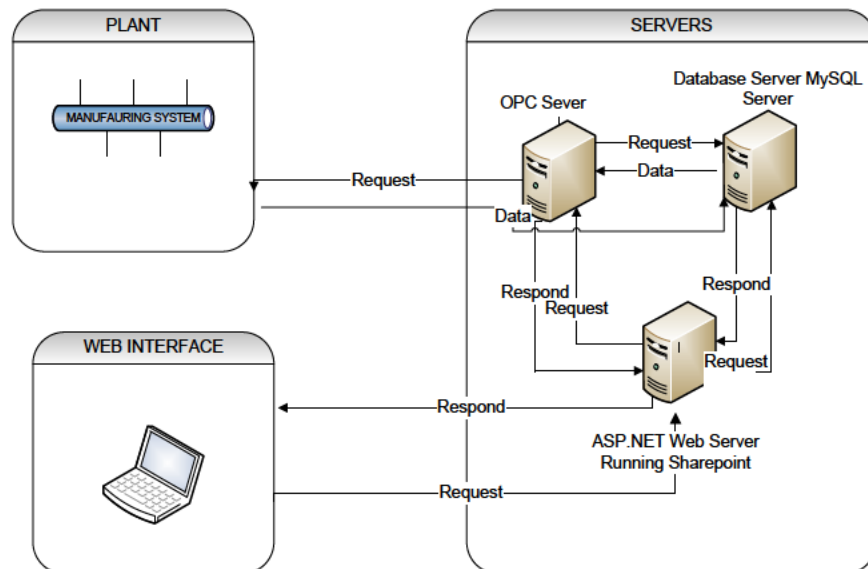


Figure 3.6: The System Overview

3.6.2 Advantages obtained using a 4-tier system

- Improvement of system stability – The server decentralizes the loads into three servers; the system stability can be improved [31].
- Reduction of network traffic – The system eliminates a possible network bottleneck by distributing the network traffic between the servers. Therefore, the network traffic of each system will be reduced and the probability of network congestion will be reduced [31].
- Improvement of security - The control server blocks direct connections from clients to the plant [31].

3.6.3 Web Server (ASP.NET Web Server)

The main function of the web server is to manage the communication between clients and remote monitoring/control system [31]. In order to provide information required by the clients, the web is connected to the database server; it is also connected to the control server so as to transfer control signals from clients to the plant. The proposed web server will use the ASP. Net technology and will meet the following requirements:

- The web server will display the real-time data of the plant requested by clients.
- The web server will have an efficient graphic user interface.

This web server for remote monitoring and controlling of the assembly system has the following basic features:

- Log in
- Monitoring Function
- Control Function

3.6.4 Log in

The clients of the remote monitoring and control system are divided into 3 classes of security and the access is limited according to the classes. The classes are as follows:

- *Administrator* - The administrator reads, deletes, updates and backups data.
- *Registered user* - The registered user reads all data (monitor, order product).
- *Guest* - The guest reads public data, log in not needed.

3.6.5 Monitoring Function

Connected to the data server, the web server shows the current state of the plant to the client in real-time on a web-user interface. Only the registered user and administrator will be able to monitor the plant; guests will only be allowed to access public data.

The web server has the following monitoring function:

- It shows the current information requested by clients as systems, or as functions, indicator, alarm and status window, by text or graph.
- It shows the trend of specific data requested by the client.
- It shows the updated time of the data to client.
- It controls the amount of information provided for client.

It provides the essential information with which the client will recognise the state of the plant because the web browser has a spatial limitation.

3.6.6 Control Function

The web server delivers the control signal from client to the control server (OPC server). The client can access real-time data from the control server. This control function is only authorized to the administrator (plant engineer), which gives him/her the authority to update, delete and control the analogue and digital values of the plant. The control function has two basic features:

- Control icon - Used for on/off control, analogue control and digital control.
- Control support - Used to confirm and validate control signals, to check that a control signal is correctly transferred to the control server.

3.6.7 Intelligent user support

It may be difficult for web clients to identify correctly the performance of the plant because the client should monitor and control the plant through the web [31]. Therefore, there are functions that support the client as follows:

- To diagnose abnormal state and suggest appropriate procedure.
- To validate the control signal from the clients.
- The takeover between automation and manual operation.

3.6.8 Database Server (MySQL Server)

The Database server is used to store data like products available for order, archive data of the plant and the username and password of users for ordering and access purposes. Only the administrator

is allowed to access this database giving him/her the authority to update, delete, and add data. The Database server is required to perform the following tasks:

- Data management - To minimize loss of data while maximizing data integrity.
- Real-time extraction of plant data - To facilitate immediate transfer of user actions.
- Security level management - Each user will have different database access permissions. If all users are able to change the data stored in the database freely, the data stored in the database will not be reliable. Furthermore, if someone who is not allowed succeeds in corrupting the database, the stability of the whole system will be compromised.
- Transfer of requested data to the web server - the user connects to the web page and requests data. Therefore, the database server links with web server. Backing up data - Data backup will be necessary to protect the data from unexpected situations that could lead to data loss or damage.

3.6.9 Control Server (OPC Server)

There are some issues in a network-based system, such as the security issue, a network disturbance issue, a real-time performance issue and so on. The control server will be used in order to cope with these issues [32]. The control server plays a role of an agent between the web server and the plant. Users will be connected with a plant through the control server, which can remove unsuitable control requests for the current situation of a plant.

The control server proposed in this case uses the OPC as seen in Figure 3.6, which are now referred to as an open connectivity via open standard protocol.

3.7 Detailed System Design

There are three main functions that this system will serve; login, control, and monitor. During the detailed design phase, each module brought out by the architectural design is now represented in UML class diagrams and UML activity diagrams in order to elaborate more about the system functionalities. The UML activity diagrams are used to describe the dynamic aspects of this system as seen in Figure 3.7. The UML class diagrams are used to visualize the static view of the system and to construct the executable code for forward and reverse engineering of the system as seen in Figure 3.8. The UML activity and class diagrams serve the following purposes [29]:

- Help construct the executable system by using forward and reverse engineering.
- Assist when investigating requirements at a later stage.
- Provide high level of understanding the system's functionalities.
- Model workflow by using activities.
- Describing the static view of the system.
- Showing the collaboration among the elements of the static view.
- Describing the functionalities performed by the system.
- Help construct the software application using object-oriented languages.

3.7.1 Login Module

The first requirement of the system is that it should have a controlled accessibility to avoid having the system being tampered with. Figure 3.7 models the workflow of the system login by using activities. These activities are used in order to understand the system functionality thoroughly and also understand the workflow of the system. During the architectural phase, the system accessibility was divided into three classes, which are guest, administrator, and registered user.

Each user is given a privilege to monitor the system but only the registered users and the administrator are allowed to control the system.

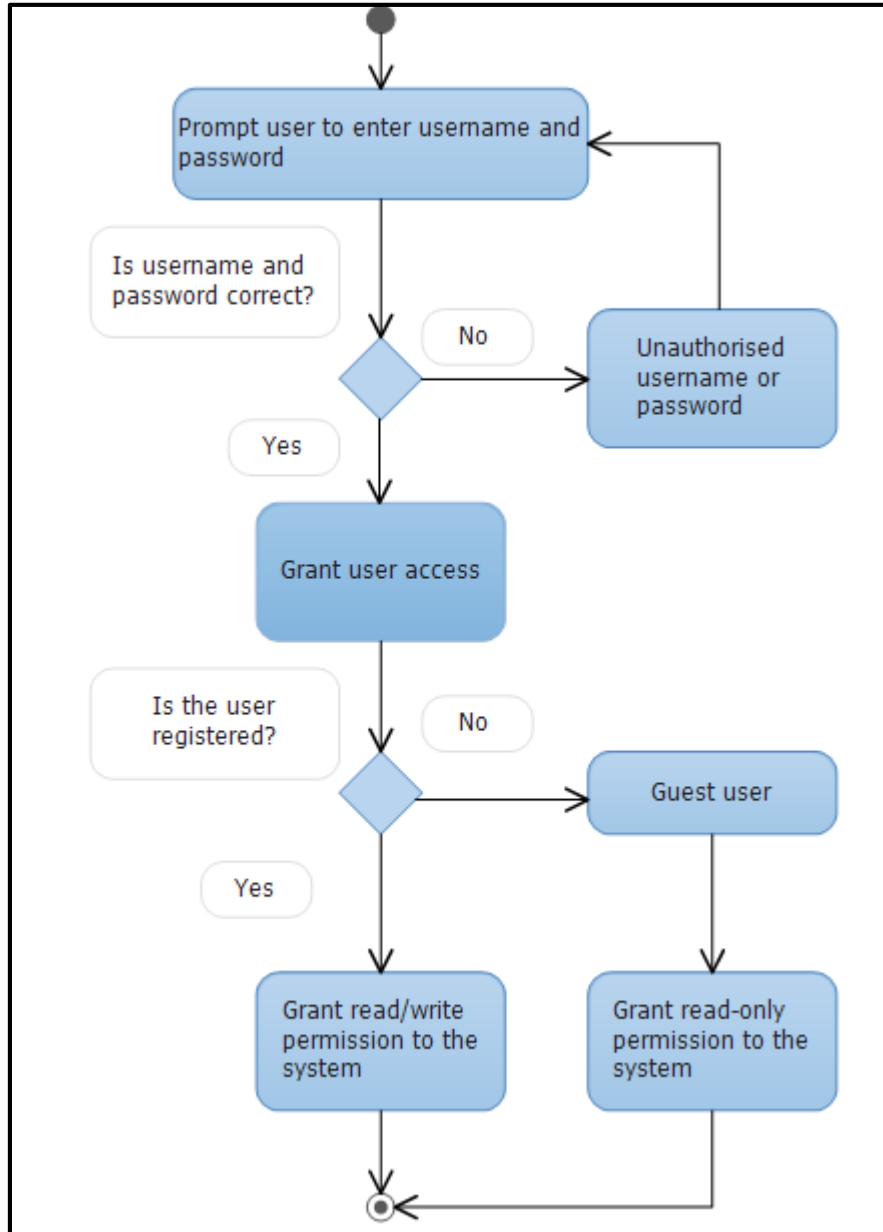


Figure 3.7: Login activity diagram

The user logs in to the system by entering the username and password; the system validates the login, and if successful, the system grants the registered user with write and read permission. If not successful, the system prevents entry. The UML class diagrams seen in Figure 3.8, Figure 3.10 and Figure 3.12 describe functionalities performed by the system. These diagrams are used to construct the proposed software application using object oriented programming. From Figure 3.8 it is noticed that system login has been divided into three classes of user access which are, guest, administrator and registered user. These three classes inherit objects from the class user, which means that without the user class these three classes would not exist. The class login is the main class which connects to the individual user account and uses the database as a referral to validate them.

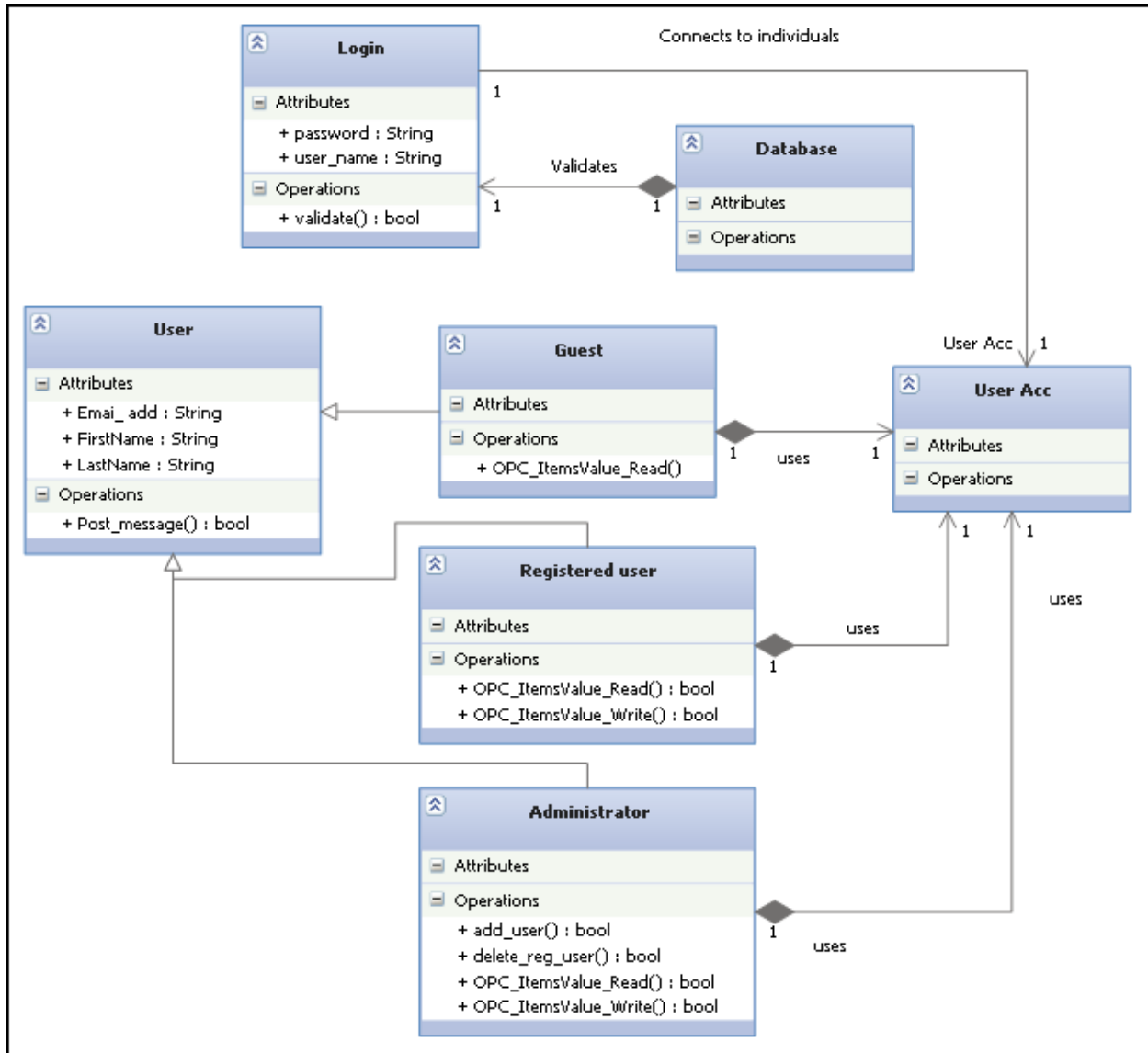


Figure 3.8: UML class diagram for Login Control

3.7.2 Monitoring Module

The monitoring module describes the design of the remote monitoring system in detail, using the UML class activity and class diagrams. Figure 3.9 demonstrates the workflow of the monitoring system and Figure 3.10 demonstrates the functionality of each activity performed by the system.

The client initiates the connection to the server by sending a request for connection and the server responds by accepting connection; the server then waits for another request from the client.

When the client sends any request that requires authentication, the server will first confirm if the client is authenticated; if not, the server will deny access and wait for another server request; but if the client is authenticated, the server will allow access to the OPC server with the read permission, allowing the user to monitor system data and status. This data is then displayed in a

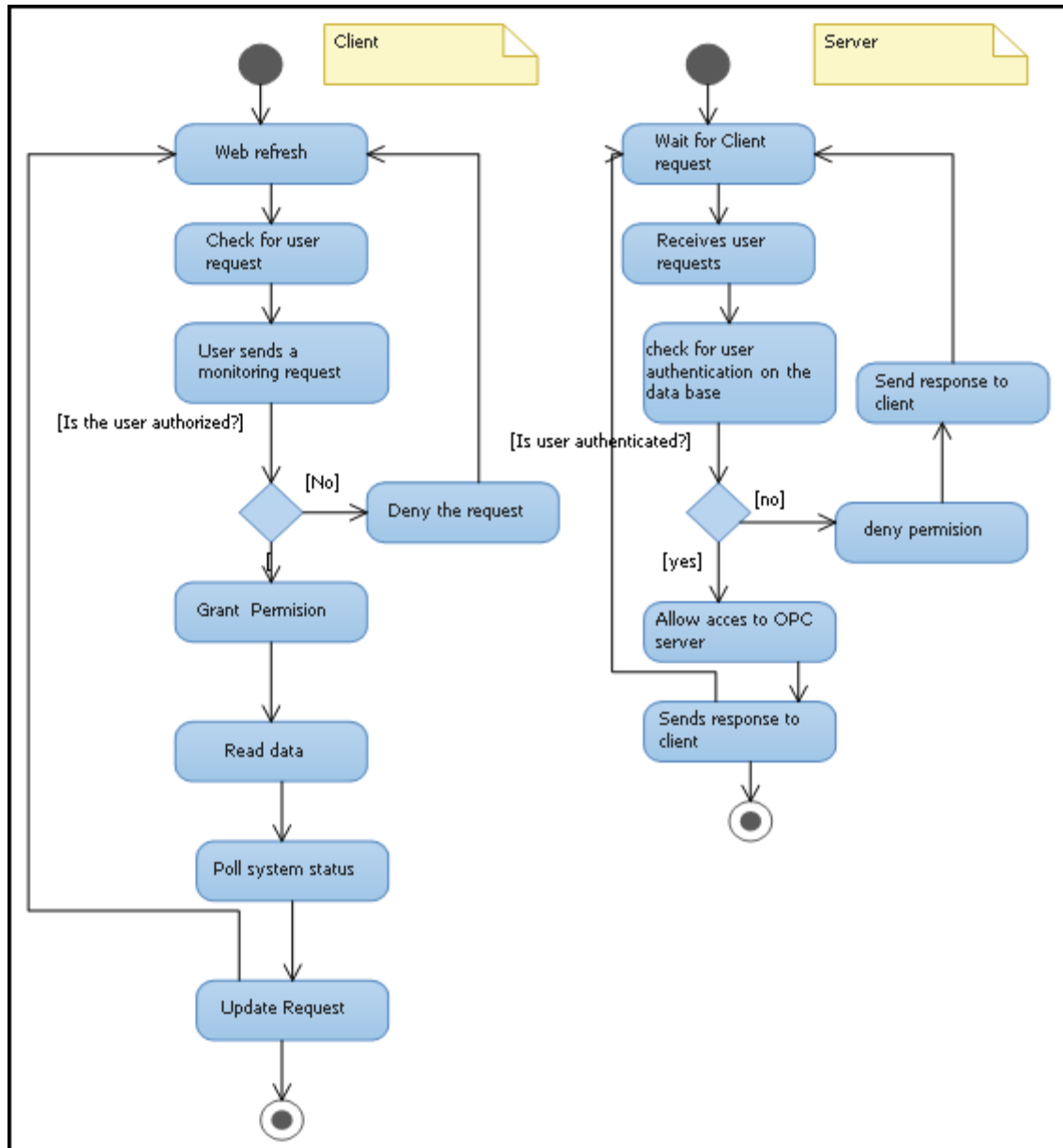


Figure 3.9: UML activity diagram for System monitor

web-user interface. After few a seconds, the web page is refreshed in order to achieve as much real-time data as possible.

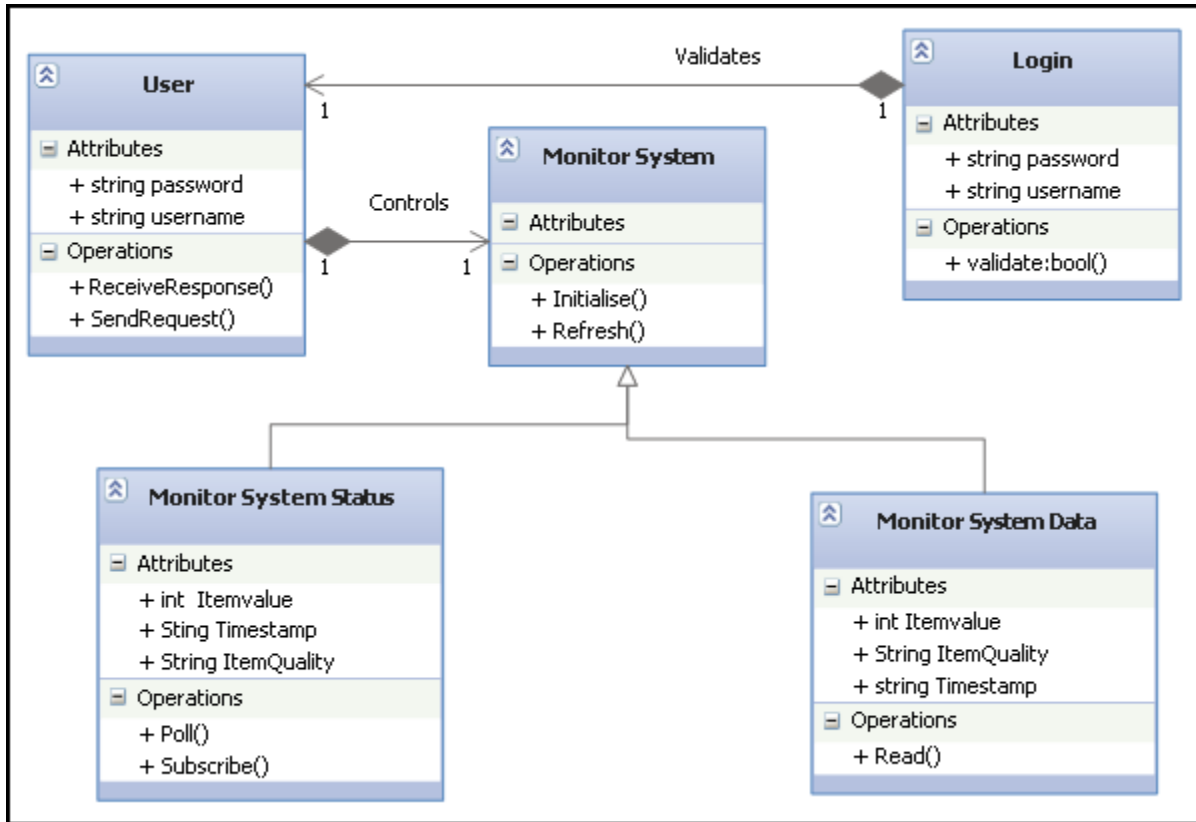


Figure 3.10: System Monitor class diagram

3.7.3 Control Module

The control module describes the remote control system in detail by using the UML activity and class diagram. Figure 3.1 represents the workflow of the control system; these workflows describe the activities that will take place during the performance of the system. Figure 3.12 describes the functionality of each activity that will take place during system performance.

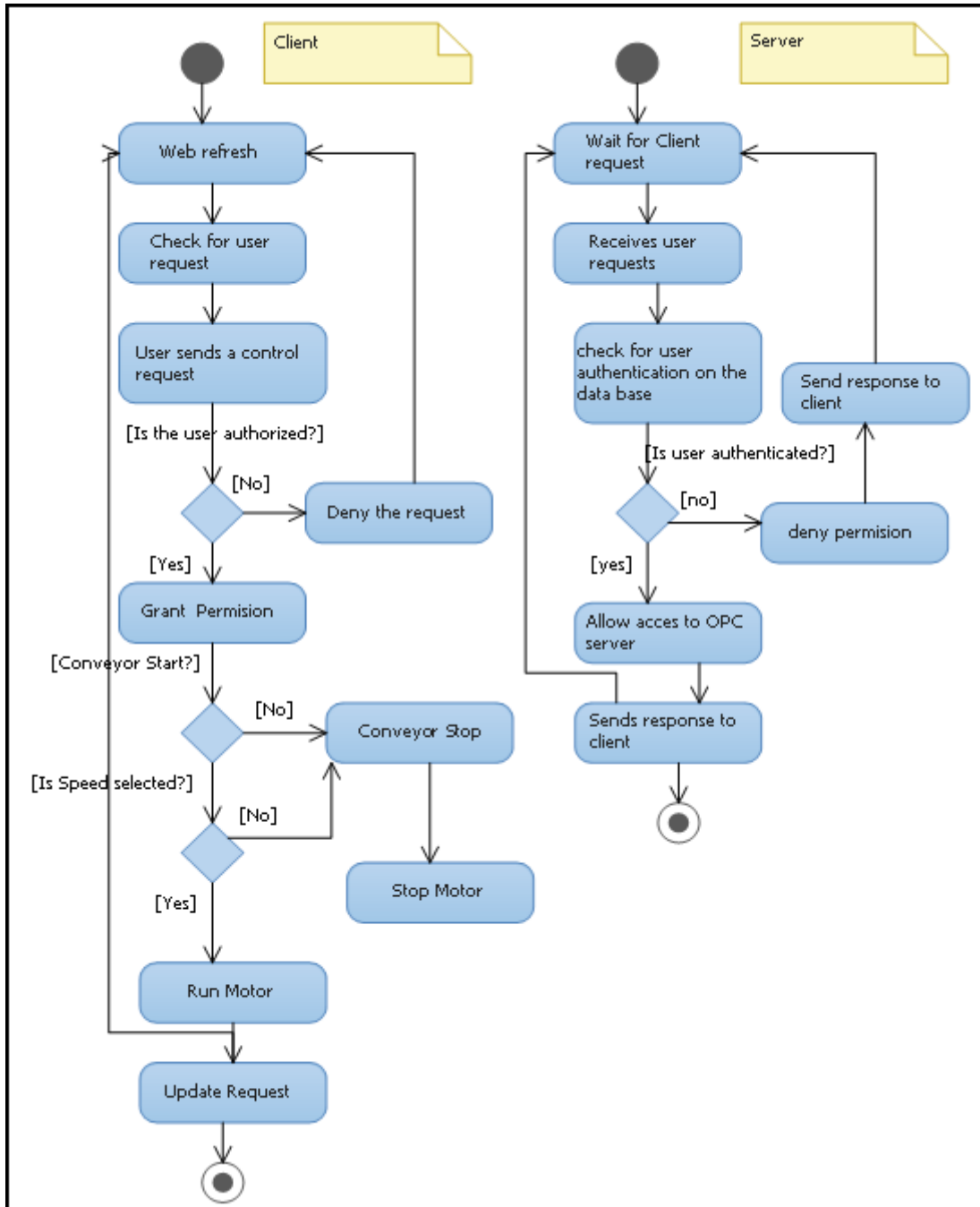


Figure 3.11: System Control activity diagram

The client will initiate connection to the server; the server accepts the connection and waits for user requests. When the server receives a request, it will first check if the client is authenticated;

if not, the client is denied request; if authenticated, the user is then allowed to access the OPC server with the write permission granted. The web interface will give the user an option of running the conveyor belt at different speeds or stop the conveyor belt.

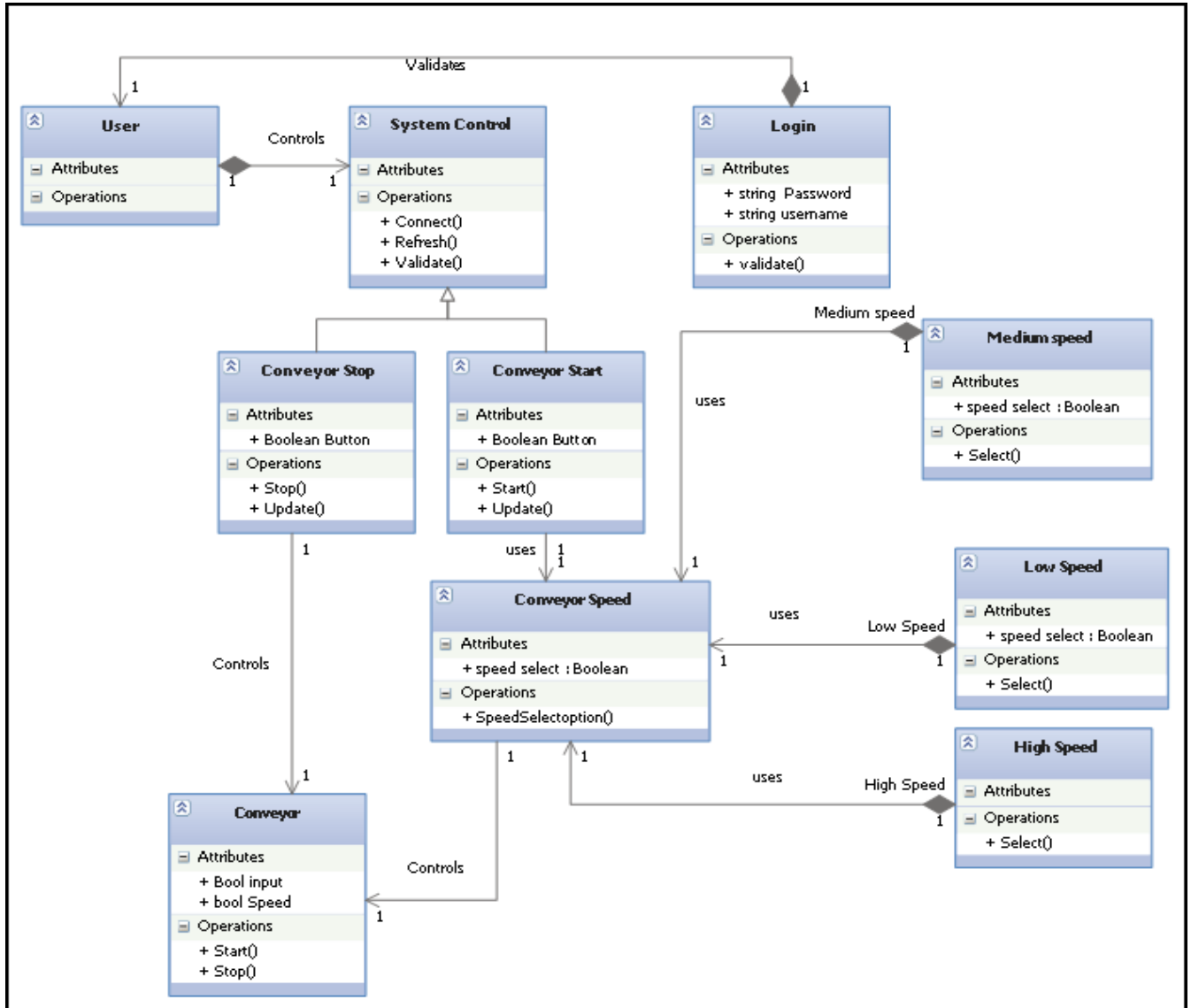


Figure 3.12: System Control Class diagram

3.8 Conclusion

Industry experience demonstrated that attempting to engage in the production of software without a well-defined, well-communicated and relevant SDLC is a recipe for disaster. It was then important for the software project proposed to go through a SDLC. The specification of a Remote Monitoring and Control System has been refined by using the UML case, class and activity diagrams. The reason for the use-case diagrams was to gather the requirements of the system and identify the external and internal factors influencing the system. After the system requirement had been refined, the next thing to do was to model those requirements into activities in order to have a clear vision of what the system is supposed to do.

The UML activity diagram was used to describe the workflow of the system behaviour. The activities in this workflow describe what actions will be performed by the system and which conditions will trigger these actions. The actions performed by this system are then used to create classes using the UML class diagram. Each class represents the different aspects of this system and each connection represents the relationship of this aspects.

The main aim of this chapter was to explore and refine the system requirements intensively before systems implementation is executed. In that way we can be sure of who will use the system, how the system will work, what components will be needed to implement the system and where the system will be used? The next chapter will explain the implementation of the system, describing the components and methods used.

4. CHAPTER IV

Design and Implementation

4.1 Introduction

In this chapter, the design and implementation of the RMCS is explained. User-interface components are described, and the method used to implement the system is discussed.

4.2 Designing Objective

The main objective of designing the RMCS web application is to provide the functions of real-time data monitoring, error alarming and remote controlling to system operator at RGEMS. The design of the system takes into accounts the capacity of the security and easy to operate user friendly interface.

4.2.1 System Implementation

The RMCS web application is implemented by integrating the OPC System.NET service and ASP.Net technology. OPC Systems.NET is an OPC Foundation laboratory certified OPC data access specification product [33]. It implements technology to provide cross platform HMI SCADA software for enterprise SCADA with support for mobile HMI and mobile SCADA solutions [33]. OPC Systems.NET is comprised of several .NET assemblies used to accomplish real-time communications to OPC Servers, OPC Clients, Visual Studio.NET applications, Microsoft Excel, and database engines like SQL Server, Oracle, MySQL, and Access. The central

communications service eliminates the need for DCOM by implementing Windows Communication Foundation communications over adjustable TCP port 58724. The real-time database provides data for Human Machine Interface, trending, alarming, data logging, recipe management, alarm notification; for standard Windows applications, Windows Presentation Foundation applications, Web-based applications, and Windows mobile pocket PC applications. Because all .NET components are fully managed and communications are implemented using .NET you can deploy Smart Client or Web applications for communications over the internet with roll-based security defined using Configure-Security and Configure-Users on the OPC Systems Service data source [33].

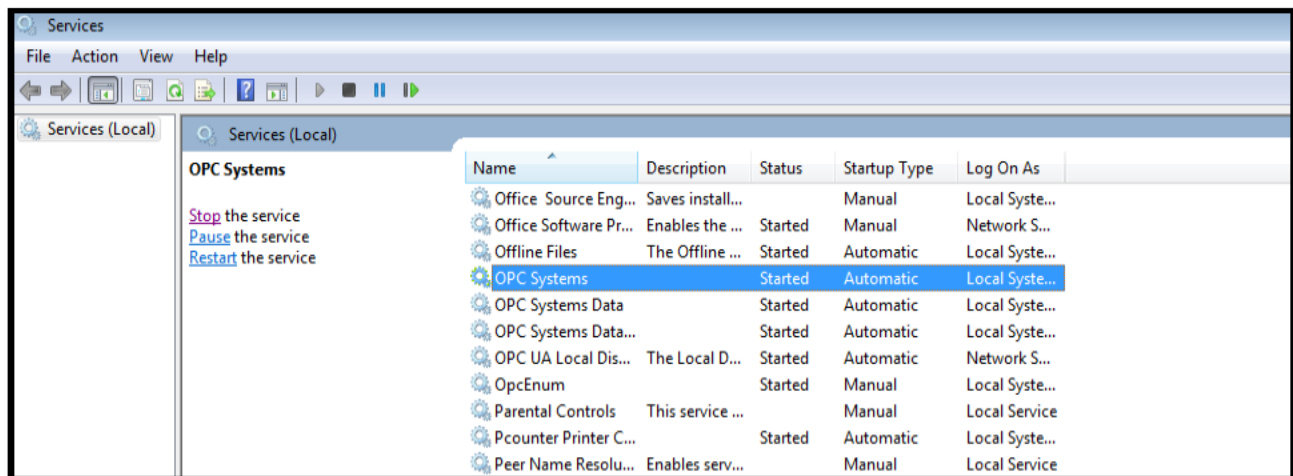


Figure 4.1: OPC System.NET service running on a local machine

The OPC System.NET service is installed locally on the machine and it allows us to connect to both remote and local OPC servers. After installation, we are able to control the operation of the OPC System.NET service by either using Windows Services as seen in Figure 4.1 or OPC System Service Control seen in Figure 4.3. This service can be started, stopped and refreshed.

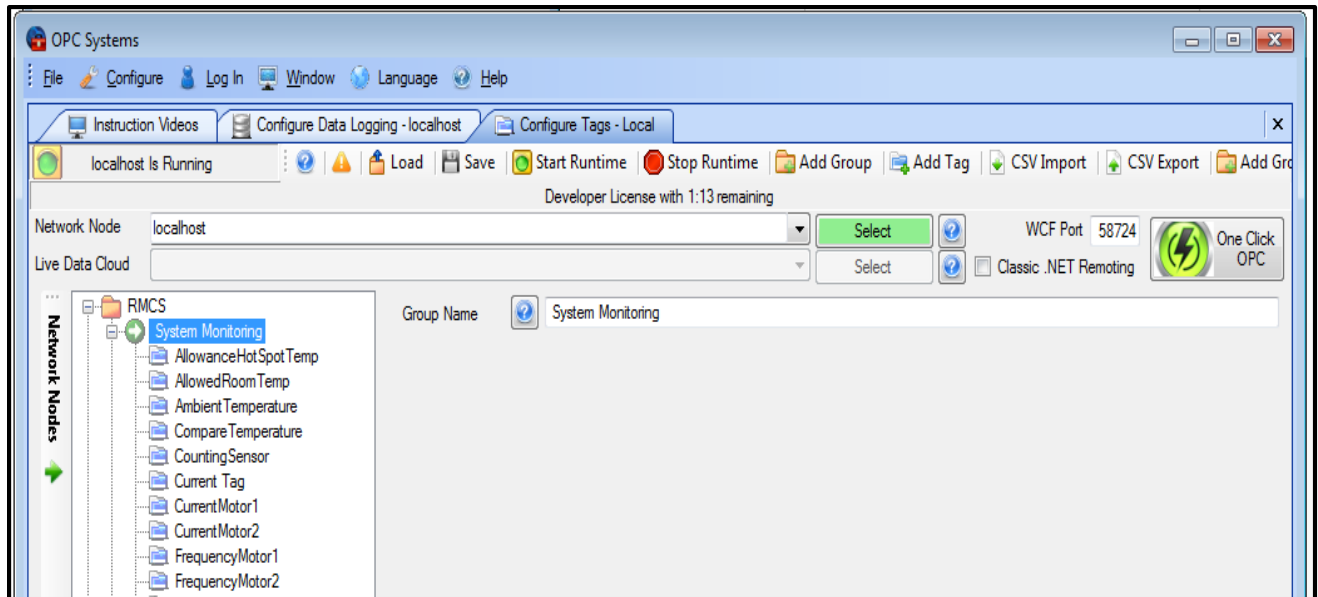


Figure 4.2 OPC System window for configuring tags

There are a number of configurations that an OPC System service offers that is available for use, as seen in Figure 4.5. In this project, only the following five configurations are to be employed:

- *Tags Configuration* - By using this configuration, one could add or delete single tags or group of tags from different OPC servers both locally and remotely. To enable access to access data tags from different OPC servers as seen in Figure 4.4, these tags were organised into groups with descriptive names (as seen in Figure 4.2) so as to avoid confusion among the large number of tags.

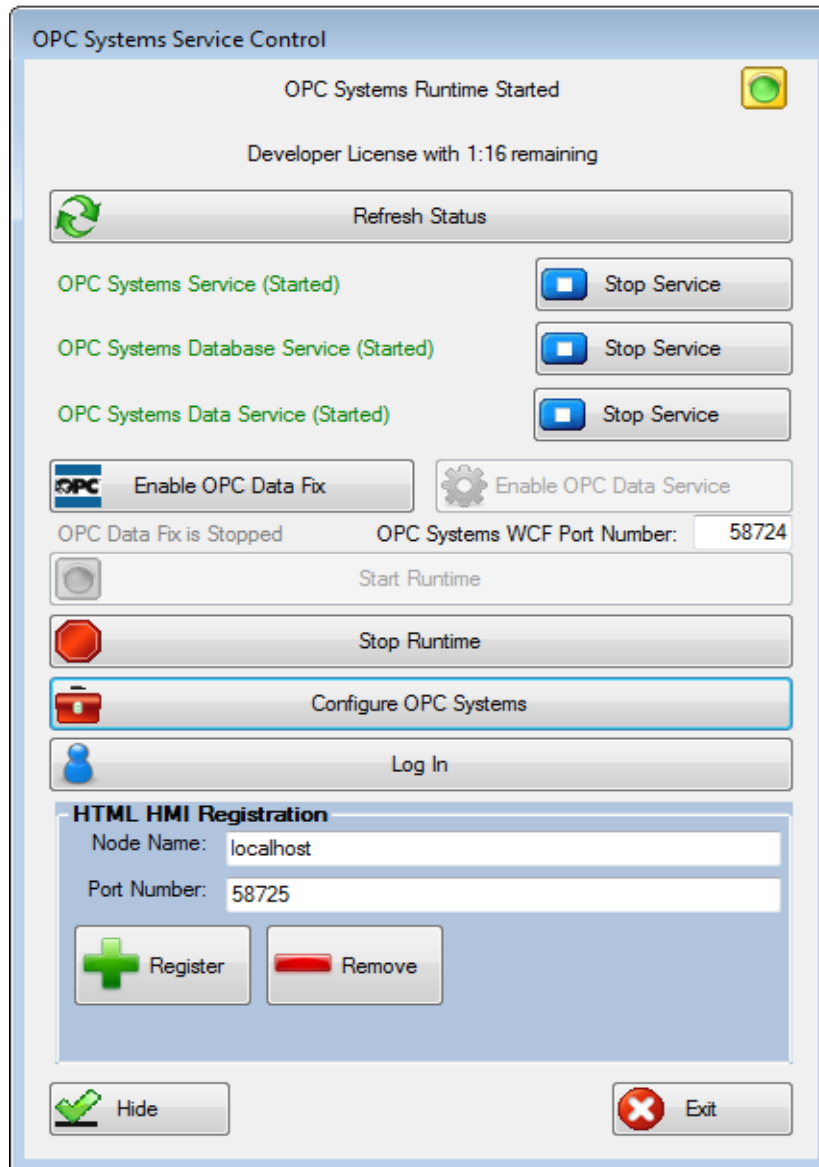


Figure 4.3: OPC System Service Control

- *Data logging Configuration* - In order to keep track of the trends of the system data, a log for every occurrence of data change in the database is needed; this configuration is used to do just that.

- *Alarm logging Configuration* - This configuration is used in this project to log the occurrence of each alarm with its level of priority and the date and time at which it occurred.
- *Security Configuration* - This configuration is used to control what users can do and not do on the system.
- *Users* - This configuration is used to add single users or groups of users who are authorised to use the system.

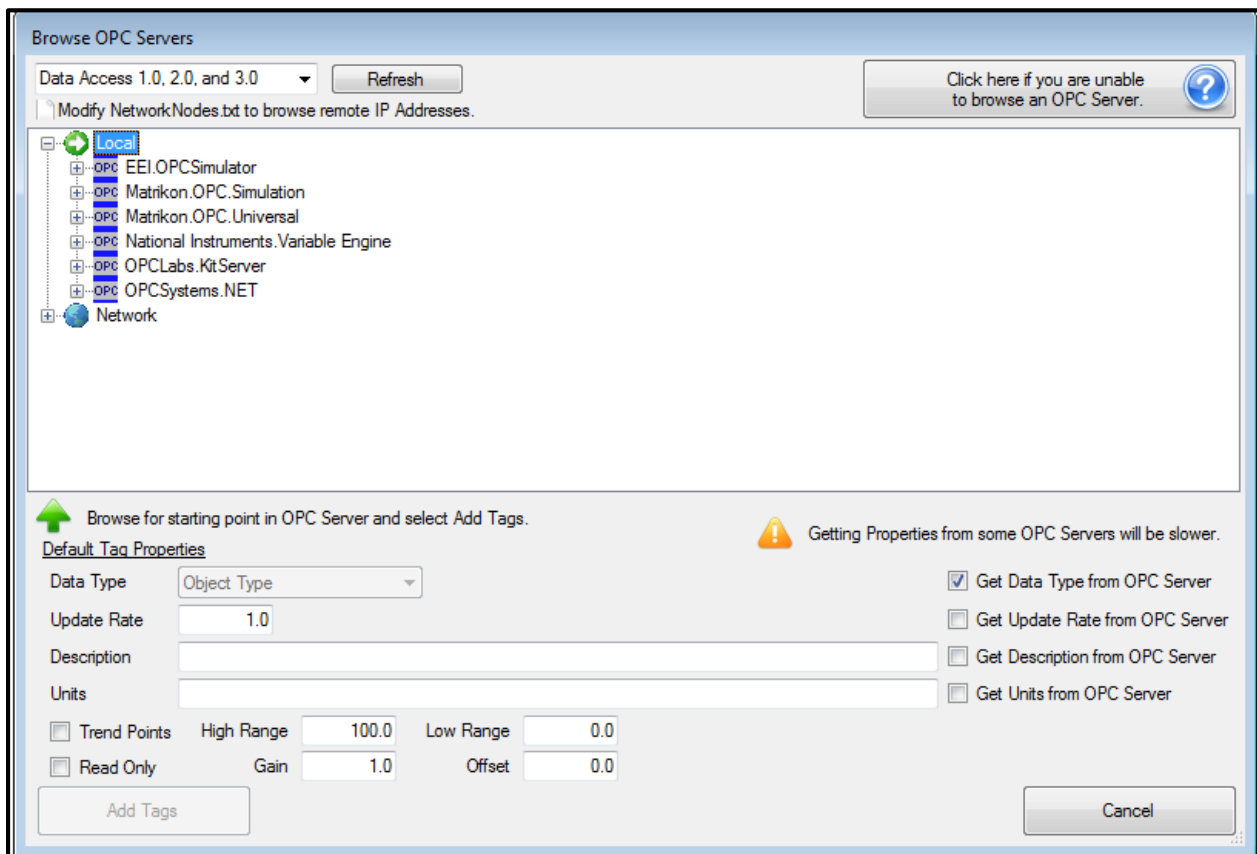


Figure 4.4: OPC System.NET OPC server browser

The OPC System.NET also offers eighteen OPC Web Controls that can be integrated with ASP.NET and .NET Framework 3.5 in order to read and write data from and to the OPC server using a web application.

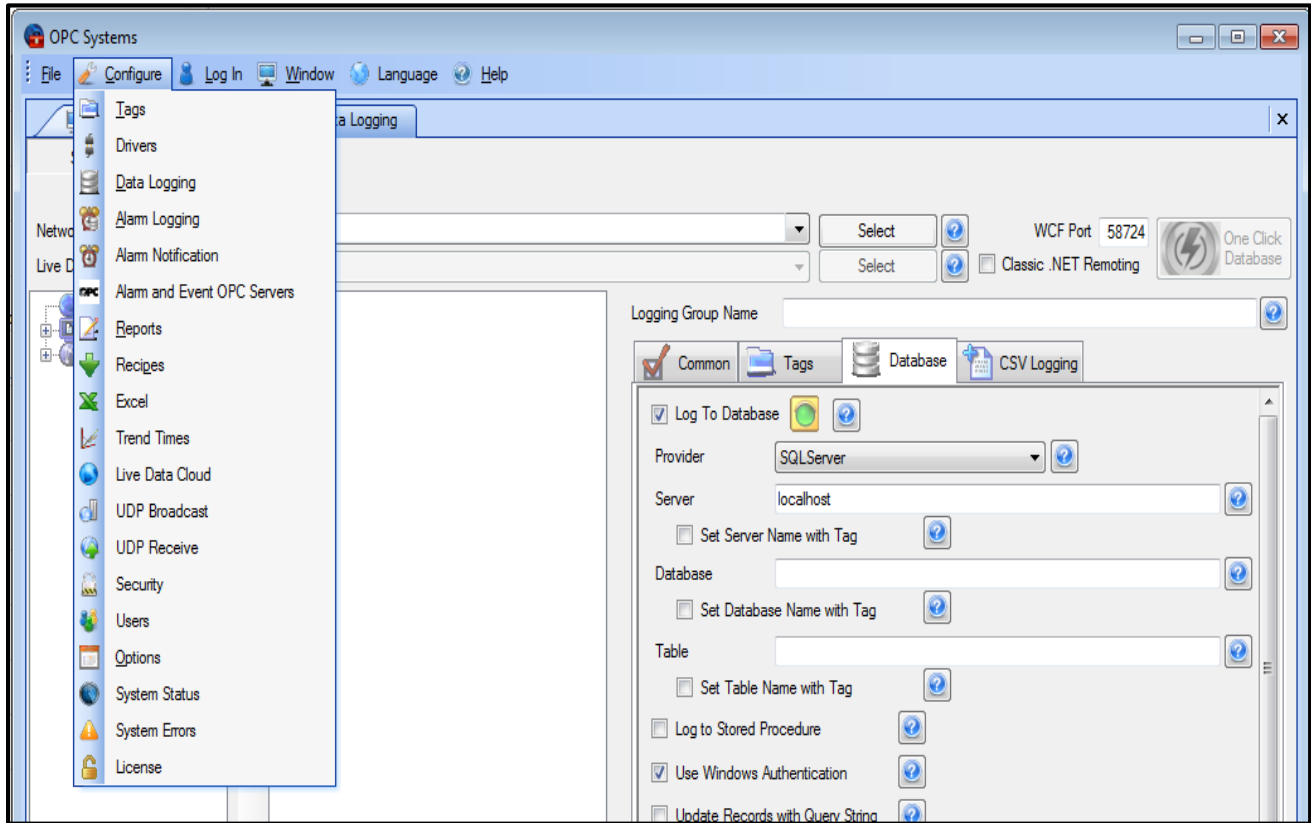


Figure 4.5 OPC System service configuration window

This is achieved by installing the OPC Web Control data link libraries in to Visual Studio; after doing so, one can drag and drop the control to the web from and edit accordingly.

4.3 User Access Management System

The user access management system of the RMCS is designed for using three-tier architectural designs, as seen in Figure 4.6. It is comprised of the client, ASP.NET web server and SQL database server. The proposed RMCS web application resides and run on the ASP.NET web server. The client will connect to the ASP.NET web server and request access to RMCS by using a web browser; a web server then connects to the SQL database server to validate the user by using a connection string from the web application, as seen in Figure 4.7.

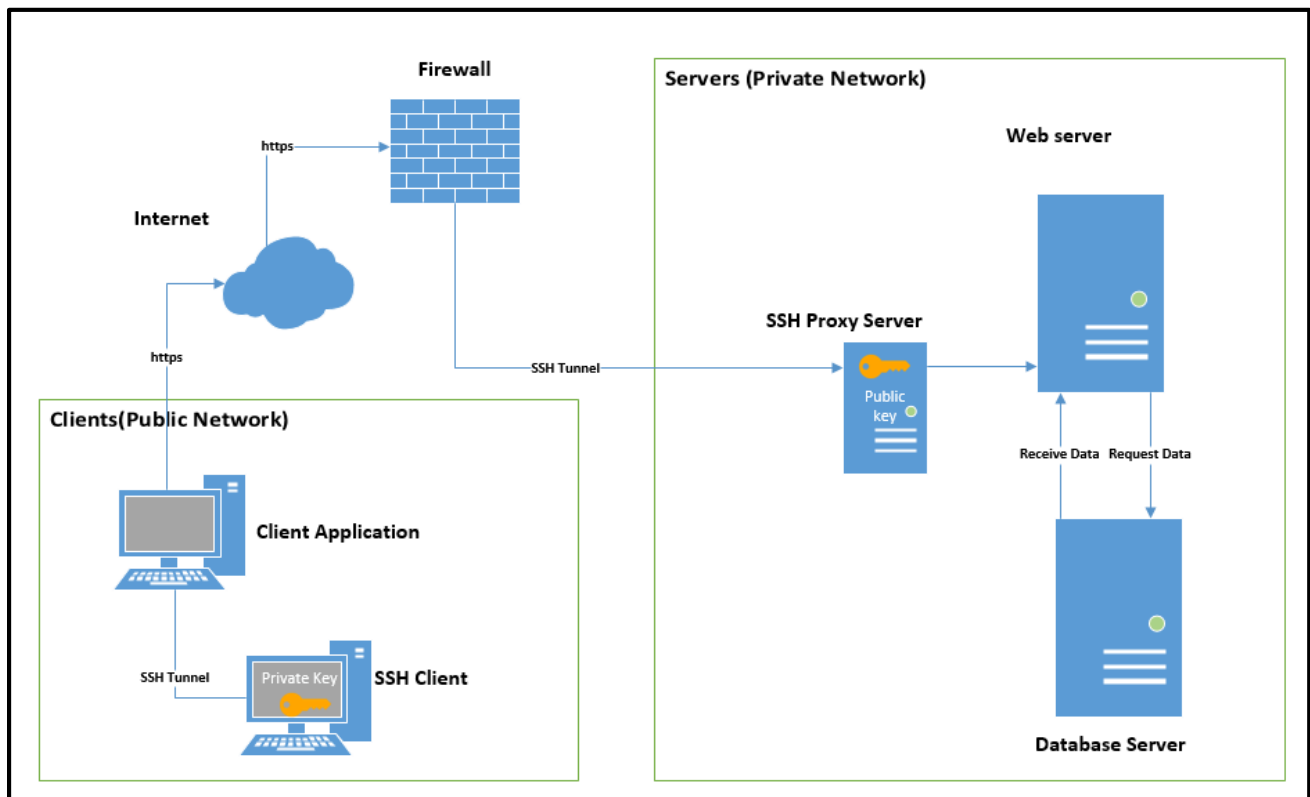


Figure 4.6: User Access Management System Architectural Design

It is almost impossible for a web system to be anti-hackable. The hacker has a full time job of trying to get into a network and/or server. It is difficult to fight the unknown exploit coming down

the pipe tomorrow or next week but there are few securities measures that was implemented to make it almost impossible for hackers to gain access to the RMCS system. The security measures are as follows:

- **Security Socket Layer (SSL)** - is the standard security technology for establishing an encrypted link between a web server and a browser. This link ensures that all data passed between the web server and browsers remain private and integral [34]. The SSL is used inside the firewall, from the load balancer to the application, from the application to the cache and SSL for Application Program Interface (API) calls. The largest key size is used as it makes it harder to break the encryption. RMCS uses the https protocol to request data from the server as it is securing the communication between the client and server by encrypting it in order to make it unreadable to the hacker.
- **User Data Security**- all data that can be used by hackers to uniquely identify a user is encrypted. Hackers usually use that data to filter their hashing and cracking attacks to data for what they identify as high profile user. RMCS uses the https protocol to request data from the server as seen in Figure 4.6, this protocol secures the communication between the client and server by encrypting it in order to make it unreadable to the hacker.
- **Server Access Security**- Another security measure that we used is to protect the system data from being breached is limiting access to the servers. This is done by:
 - a) Disabling all access from outside to the application server.
 - b) Disabling remote root login - When root login is enabled the root user by default has access to all commands and files, we don't want to make it easier for the hackers that is why we are disabling the remote root login.

- c) Use of Secure Socket Shell (SSH) to access the server- SSH is a network protocol that provides administrators with a secure way to access a remote computer. It provides strong authentication and secure encrypted data communications between client and server connecting over an insecure network such as the Internet [35] . SSH uses public key cryptography to authenticate the remote computer and allow the remote computer to authenticate the user, if necessary as seen in Figure 4.6. It will be difficult for a hacker to gain access remotely to the server unless they have the keys.
- a) Use Secure Copy Protocol (SCP) to move files to and from the server. The SCP is a means of transferring files between local and remote host or remote host to remote host [36]. The traffic between the hosts is encrypted so it will not be easy for the hacker to eavesdrop.
- **Authentication-** lastly the security measure that is implemented is authentication. User passwords are uniquely generated using hashing.

I Client

In order for user(s) to access the RMCS web application, which runs and resides on the ASP.NET web server, they need to use any device that can run a web browser; these devices are often called client(s) because of their capability to communicate with the server. The client in this case would be a user's smart phone, laptop, tablet, personal computer, etc. The user(s) can use any of these devices to connect to the RMCS web application. The client is one of the components that contribute to architecture of the user access management system.

II ASP.NET Web Server

The RMCS web application runs on the ASP.NET Web server with a unique IP address, HTTP port and URL name. The unique IP address, port and URL name is to make sure that the web application is uniquely identified and on the Internet, since web application cannot share the same address. The role of the ASP.NET web server in this project is to allow the RMCS web application to be accessible through the internet; without it our web application will not be accessible through the internet. The ASP.NET web server connects to the database server, as seen in Figure 4.6, using the “web. Config” file of the RMCS web application. This XML-formatted .config file used .Net Framework version 4.0 to configure the connection from our web application to the database server. The connection string is shown in Figure 4.7; this connection string specifies the name of the database, SQL server and the credentials used to connect to the server which the web server can use to connect to the SQL server.

```
<configuration>
  <connectionStrings>
    <add name="RMCS" connectionString="Data Source=.\SQLEXPRESS;Initial Catalog=CEDatabase;Integrated Security=True" providerName="System.Data.SqlClient"/>
  </connectionStrings>
</configuration>
```

Figure 4.7: Connection string for connecting to the database

III SQL Database Server

The database server stores information about the user, which includes the username and the password. The structure of the database is as shown in Figure 4.8; there are two tables that will be used, which are UserReg and Credentials. UserReg has the columns ID, PersoStudNo, Name, Surname Role, and RegDate and is used to store brief identification data of user(s).

- a) ID - it is used to number the items on the table.
- b) PersoStudNo - This column is of a type integer and is created to store the unique number that a user can be identified with; this unique number can be a student number or a personnel number. This number is a primary key to table UserReg.
- c) Name - This column is of a type string and stores the first name of the user.
- d) Surname – This column is of a type string stores the surname of the user.
- e) Role - This column is of a type string and stores the role that the user has on the system e.g. administrator, guest user or registered user.
- f) RegDate - This column is of type datetime and stores the date when the user was created or modified.

The table Credential has only four columns namely PersoStudNo, Username, Password, RegDate and it is used to store user(s)'s credentials; that is, a username(s) and password(s).

- a) *PersoStudNo*- This column is of a type integer a foreign key on table Credentials and is used to store the unique number that a user can be identified with; this unique number can be a student number or a personnel number.
- b) *Username*- This column is of type string and is used to store the user's username.
- c) *Password* – This column is of type variable binary and is used to store the user's password in a series of bytes.

The user password should only be known by the user and the user alone, so one needs to encrypt the password before saving it to the database. In order for us to ensure that we store an encrypted password, we will use a data field of type variable binary (varbinary). The encrypted password

will be saved as a series of bytes that a human mind cannot decode. These steps were followed to achieve insertion of encrypted data into the database:

1. Took user plain-text input, which is the password entered on the textbox.
2. Encrypt it.
3. Convert it from string into a byte array.
4. Issue the INSERT/UPDATE query.

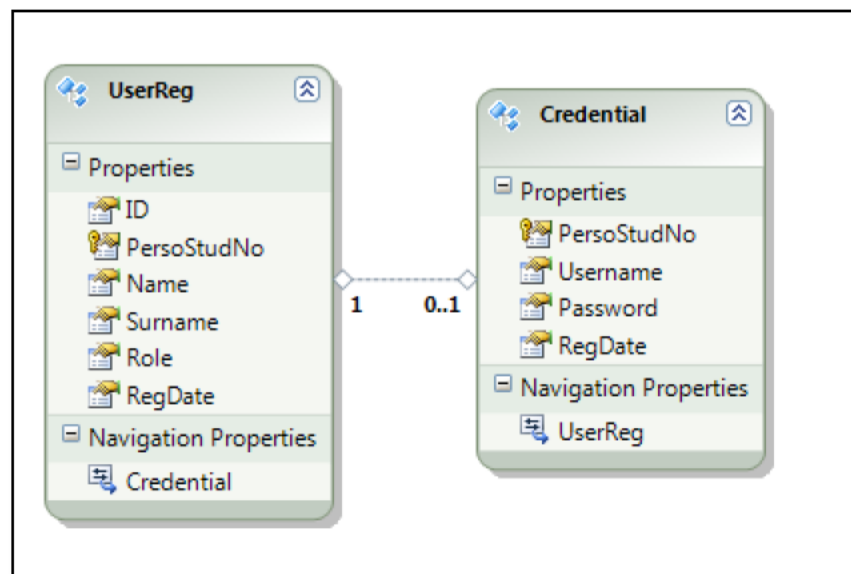


Figure 4.8: User Access Management System Database Structure

These steps would be repeated every time one wants to insert/update data in the database. By doing this, one makes it impossible for malicious hackers to use user passwords to gain access to the system.

4.3.1 User Registration System

The user registration system was implemented in order to register every user who will be using the RMCS web application. By registering users, we can identify who the user is, keep log of what he/she did on the system and when was it done - we will know exactly who stopped and started the system and when. It is essential to identify users because whenever we have a problem on the system, we can go back to our logs and check who was logged in when the system had started to malfunction and what operations the user had performed. This way is better because the person who logged on at that time might have a clue of what he/she did that might have caused the system to malfunction; we do not have to struggle finding the root of the malfunction not even knowing which operation was performed by who?

```

protected void btnRegister_Click(object sender, EventArgs e)
{
    if (txtName.Text.Length > 0 && txtPersoStudNo.Text.Length>0 &&
txtSurname.Text.Length > 0)
    {
        //Query insert to database
        SqlCommand cmd = new SqlCommand("insert INTO UserReg
(PersoStudNo,Name,Surname,Role,RegDate) VALUES (" + txtPersoStudNo.Text + "','" +
txtName.Text + "','" + txtSurname.Text + "','" + cboRole.SelectedValue + "','" +
DateTime.Now + "')", connection);
        //Randomly generate password with 8 characters and 1 non-aplhanumeric
charater
        string generatePassword =
System.Web.Security.Membership.GeneratePassword(8,1);
        string password = generatePassword;
        // Create a random instatnce
        Random rand = new Random();
        //Pick a random number from 0 to 100
        int number = rand.Next(100);
        string first = txtName.Text;
        string second = txtSurname.Text;
        //get the first string of the word
        string getfirst = first.Substring(0, 1);
        //create username by combining the first string of the name and surname
        string username = getfirst + second;

        //Query insert to database
        SqlCommand cmd1 = new SqlCommand("insert INTO Credentials
(PersoStudNo,Username>Password,RegDate) VALUES (" + txtPersoStudNo.Text + "','" + username
+ "','" + password + "','" + DateTime.Now + "')", connection);
        cmd1.ExecuteNonQuery(); //Execute second SQL command

        cmd.ExecuteNonQuery();// Execute first sql command
        //clear the database
        txtName.Text = "";
        txtPersoStudNo.Text = "";
        txtSurname.Text = "";
        // redirect page and store username and password
        Response.Redirect("~/Feedback.aspx?myusername=" + username +
"&mypassword=" + password);
    }
    else
    {
        lblAllfields.Text = "Please fill all the fields !!";
        lblAllfields.ForeColor = System.Drawing.Color.Red;
    }
}
}

```

Figure 4.9: User registration method in C#

Every user who will need to use the RMCS will only be given access to do so by the administrator. The reason for allowing only the administrator to register users, is because even though this system will be running on the public internet, it is a private system and it should only be accessed by a collective number of people and preferably an RGEMS member who is also working on the

system. By limiting people who can access the system, we avoid compromising the system functionality and integrity as the internet is vulnerable to malicious hackers.

The administrator will need a student number or personnel number, and the name and surname of the person requesting access to the system in order to register the user. The registering system will generate the password using Microsoft System Web Security; it is a namespace developed by Microsoft and it comes with different classes and methods that serve different purposes when working with web security. The class that was adopted in our project is the “Membership” class. This class has the method “Generate Password”, and this method accepts two parameters, the first parameter being the length of the password and the second parameter being the number of non-alphanumeric character as seen in Figure 4.10.

```
//Randomly generate password with 8 characters and 1 non-aplhanumeric charater  
string generatePassword = System.Web.Security.Membership.GeneratePassword(8,1);  
string password = generatePassword;
```

Figure 4.10: Unique password generation method using C#

The method “Generate Password’ was used to create a unique user’s password. This was achieved by combining eight randomly picked letters and one non-alphanumeric character. The generated password will serve as a temporary password that can be changed by the user whenever they want to and as many times as they need to. The username is created by combining the user’s first name’s initial and surname as seen in Figure 4.11. The username does not have any length restriction; it will just depend on how long the user’s surname is.

```
string first = txtName.Text;
string second = txtSurname.Text;
//get the first string of the word
string getfirst = first.Substring(0, 1);
//create username by combining the first string of the name and surname
string username = getfirst + second;
```

Figure 4.11: Username creation method using C#

4.3.2 User Management System

A user management system was implemented in order to manage the authentication and authorization of users. The authentication and authorization is done by the administrator. The user management system is a sub-system that is used by the administrator in order to control a user's access to the system. The administrator will control the user's access by either deleting the user or changing the role of the user.

Changing Role - The administrator can decide to change the role of the user depending on what he/she wants the user to have access to and what he/she does not want the user to have access to. The administrator has three options for roles, namely administrator, user and guest user. If the user is needed to have full access, the right administrator role is given; if the user only needs access to the system with read – only the right guest role is given; and if the user needs access to the system with read and write, the right user role is granted.


```
public void FillGrid()
{
    SqlDataAdapter adapt = new SqlDataAdapter("Select * From UserReg", connection);

    DataSet set = new DataSet();
    adapt.Fill(set, "UserReg");
    if (set.Tables["UserReg"].Rows.Count > 0)
    {
        grdManageUsers.DataSource = set;
        grdManageUsers.DataBind();
    }
}
```

Figure 4.12: Grid view filled with list of registered users

In order to make the user management system user-friendly and easy to manage, the ASP.NET grid view control is employed. This control is similar to a table as it is also composed of rows and columns. Firstly, a list of registered users was selected from the database and filled into a grid view. This was done by using C# SQL select statement as seen in Figure 4.12. Secondly, one needs to be able to change the roles of the users when necessary, so the grid view needed to be editable, as seen in Figure 4.13. Thirdly and lastly, one needs to make sure that after changing or modifying users' data, the database is being updated with the new data.

```
protected void grdManageUsers_RowEditing(object sender, GridViewEditEventArgs e)
{
    grdManageUsers.EditIndex = e.NewEditIndex;
    FillGrid();
}
```

Figure 4.13: Activate edit on grid when row editing event is invoked

The database will be updated by using a C# SQL update statement to update users' data on the database when event "Row Updating" of "grid view" is invoked, as seen in Figure 4.14.

```
protected void grdManageUsers_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    TextBox txtNo =
    (TextBox)grdManageUsers.Rows[e.RowIndex].FindControl("txtNo");
    TextBox txtUserID =
    (TextBox)grdManageUsers.Rows[e.RowIndex].FindControl("txtUserID");
    TextBox txtName =
    (TextBox)grdManageUsers.Rows[e.RowIndex].FindControl("txtName");
    TextBox txtSurname =
    (TextBox)grdManageUsers.Rows[e.RowIndex].FindControl("txtSurname");
    DropDownList cboRole =
    (DropDownList)grdManageUsers.Rows[e.RowIndex].FindControl("cboRole");
    SqlCommand cmd = new SqlCommand("Update UserReg Set
    PersoStudNo="+txtUserID.Text+",
    Name='"+txtName.Text+"',Surname='"+txtSurname.Text+"',Role='"+cboRole.SelectedValue+"',
    RegDate='"+DateTime.Now+"' WHERE ID="+txtNo.Text+" ", connection);
    cmd.ExecuteNonQuery();
    grdManageUsers.EditIndex = -1;
    FillGrid();
}
```

Figure 4.14: Perform system update on row update event

Delete User - Only the administrator has the right to delete the user. By deleting the user, this means that the user's record will be permanently deleted and the deleted user will not be able to log in to the system. The system deletes the data using the method seen in Figure 4.15.

```
protected void grdManageUsers_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    Label lblUserID =
(Label)grdManageUsers.Rows[e.RowIndex].FindControl("lblUserID");
    SqlCommand delete = new SqlCommand("Delete From UserReg Where
UserReg.PersoStudNo=" + lblUserID.Text + " ", connection);
    delete.ExecuteNonQuery();
    FillGrid();
}
```

Figure 4.15: Perform system delete when row delete event is invoked

4.3.3 Login System

The login system was implemented in order to authenticate and authorize users on the system. We used a C# SQL select statement to validate if the record using the username and password provided by the user exists on the database; if the record does not exist, the system knows that the user is not authenticated and it will deny him/her access.

```

protected void btnLogin_Click(object sender, EventArgs e)
{
    SqlDataAdapter adapt = new SqlDataAdapter("Select Username,Password,Role From
dbo.Credentials INNER JOIN dbo.UserReg ON dbo.Credentials.PersoStudNo =
dbo.UserReg.PersoStudNo where Username='"+txtUsername.Text+"' AND
Password='"+txtPassword.Text+"'", connection);
    DataTable set = new DataTable();
    adapt.Fill(set);
    if(set.Rows.Count>0)
    {
        foreach (DataRow row in set.Rows)
        {
            string role = row["Role"].ToString();
            switch (role)
            {
                case "Administrator":
Response.Redirect("~/RegisterUser.aspx?myusername="+txtUsername.Text+"");
                break;
                case "User":
                Response.Redirect("~/ControlSystem.aspx?myusername=" +
txtUsername.Text + "");
                break;
                case "UserGuest":
                Response.Redirect("~/Monitoring.aspx?myusername=" +
txtUsername.Text + "");
                break;
            }
        }
    }
    else
    {
        lblErrorMessage.Text= "Incorrect Username or Password";
        lblErrorMessage.ForeColor= System.Drawing.Color.Red;
    }
}
}

```

Figure 4.16: Authenticate and authorize user on button click event

The system does not only authenticate the user's username and password, but also authorizes what the user can do on the system by checking which role the users have on the system and provides them rights to the system according to those roles. In Figure 4.16 it is a code snippet that explains the method used to authenticate and authorize the user.

4.4 Monitoring System

The monitoring system was implemented to allow authenticated and authorized user to monitor the state of the assembly system from a remote location using a smart phone, table, Personal Computer (PC), laptop, etc.

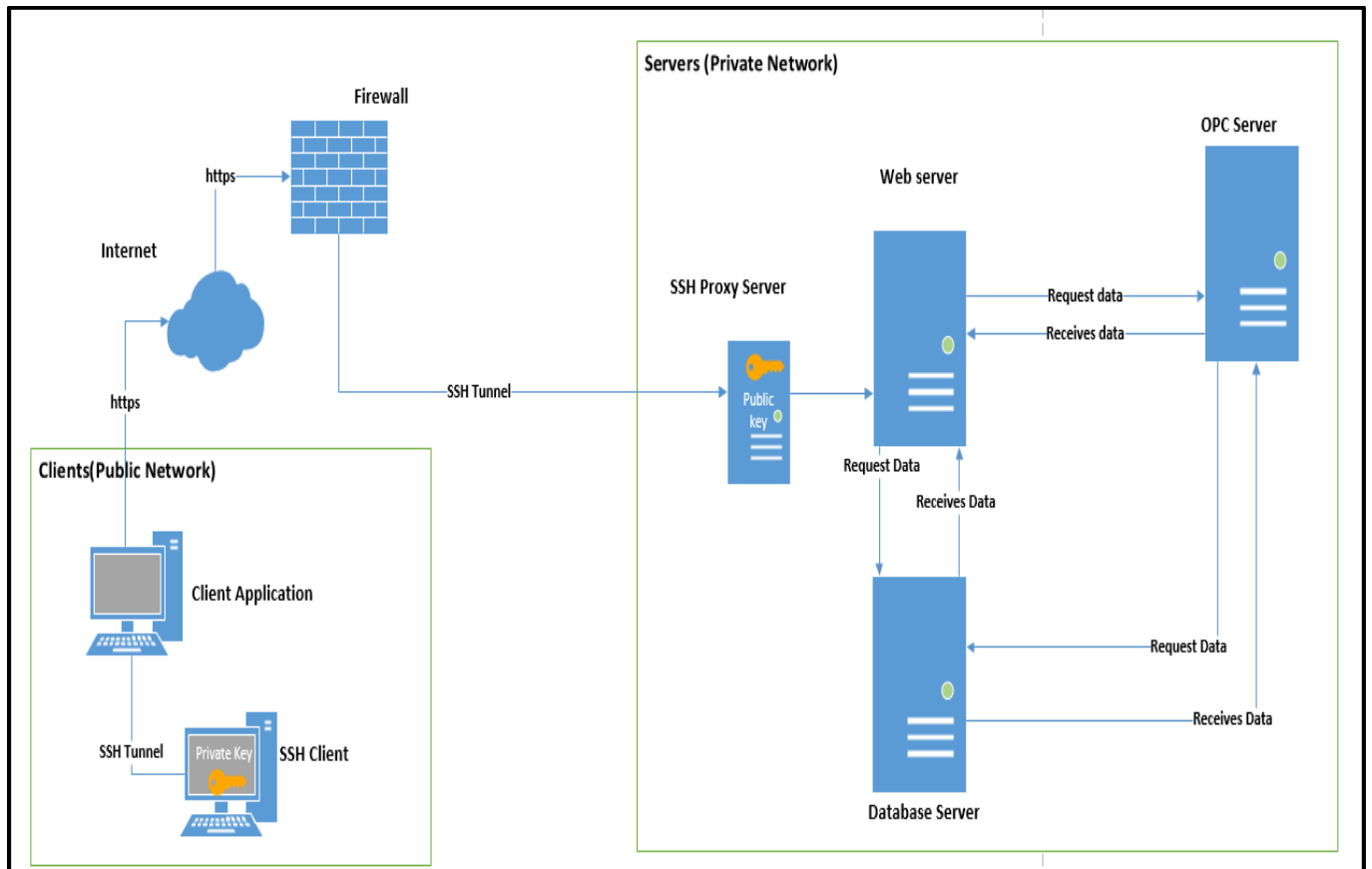


Figure 4.17: Architectural design of a monitoring and control system

The important elements of the system that need to be monitored are current consumed by each motor driving the entire system, system sensors, power consumed, state of the entire system (running or halted) and units produced. The monitoring system is designed using a three-tier

architectural design, which is similar to the architecture used by user access system; but instead of SQL database server, an OPC server is used - in this case as seen in Figure 4.17.

4.4.1 Monitored Data

a) *System State* - Before the system operator can do anything on the system, he/she needs to know if the system is running or on halt - especially when operating the system remotely. The system can stop running because of a power cut or emergency shutdown caused by a system malfunction; we would not know if not physically present at the assembly system. If it happens that the above-mentioned occurs, it will create an alarm that will be sent to RMCS web application to alert the remote system operator about the state of the system.

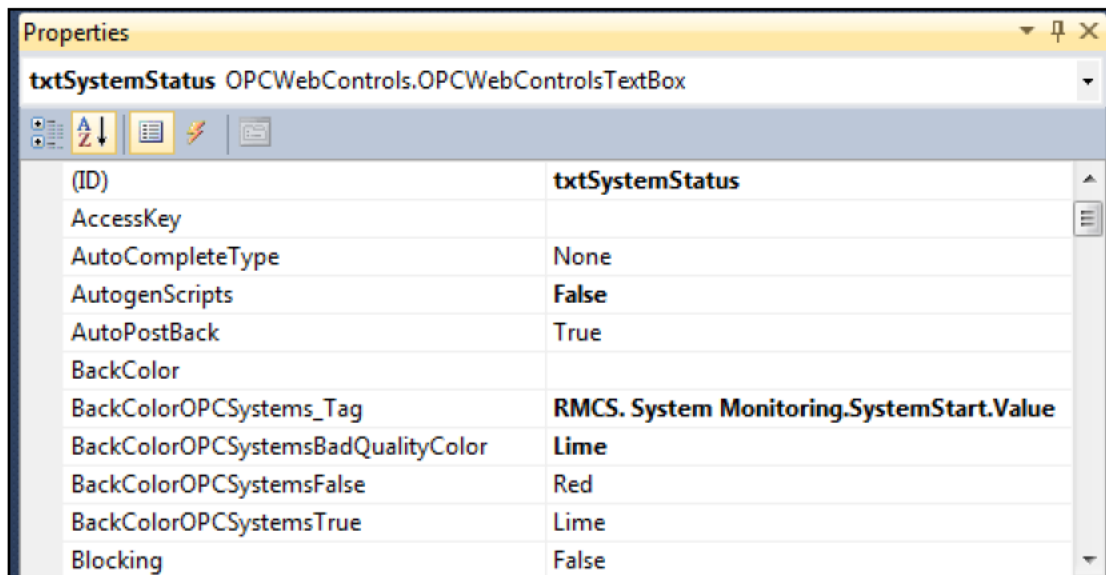


Figure 4.18: OPC Web Control Textbox Properties for system status

The system state is represented by the tag “SystemStart” of data type Boolean as seen in Figure 4.20. If the system is running the “SystemStart” state will be logic “1” and if the system has stopped

the state will be logic “0”. When the system operator is physically at the assembly system, he can use the green switch to start the system (i.e. set the “SystemStart” tag to logic “0”) and red switch to stop the system (i.e. set the “SystemStart” tag to logic “1”) as seen Figure 4.21.

To allow the system operator to monitor the state of the system in real-time from a web application, the integration of ASP.NET technology and OPC System.NET service is employed. The OPC System.Net service is responsible for allowing the connection between the web application and the local or remote OPC server. We used OPC web control of type textbox as an LED to indicate if the system is “ON” or “OFF” by changing its background colour. We have referenced the property “BackColorOPCSytems_Tag” to the value of the “SystemStart” tag as seen in Figure 4.18 so that the back colour of the textbox would indicate the state of the system.

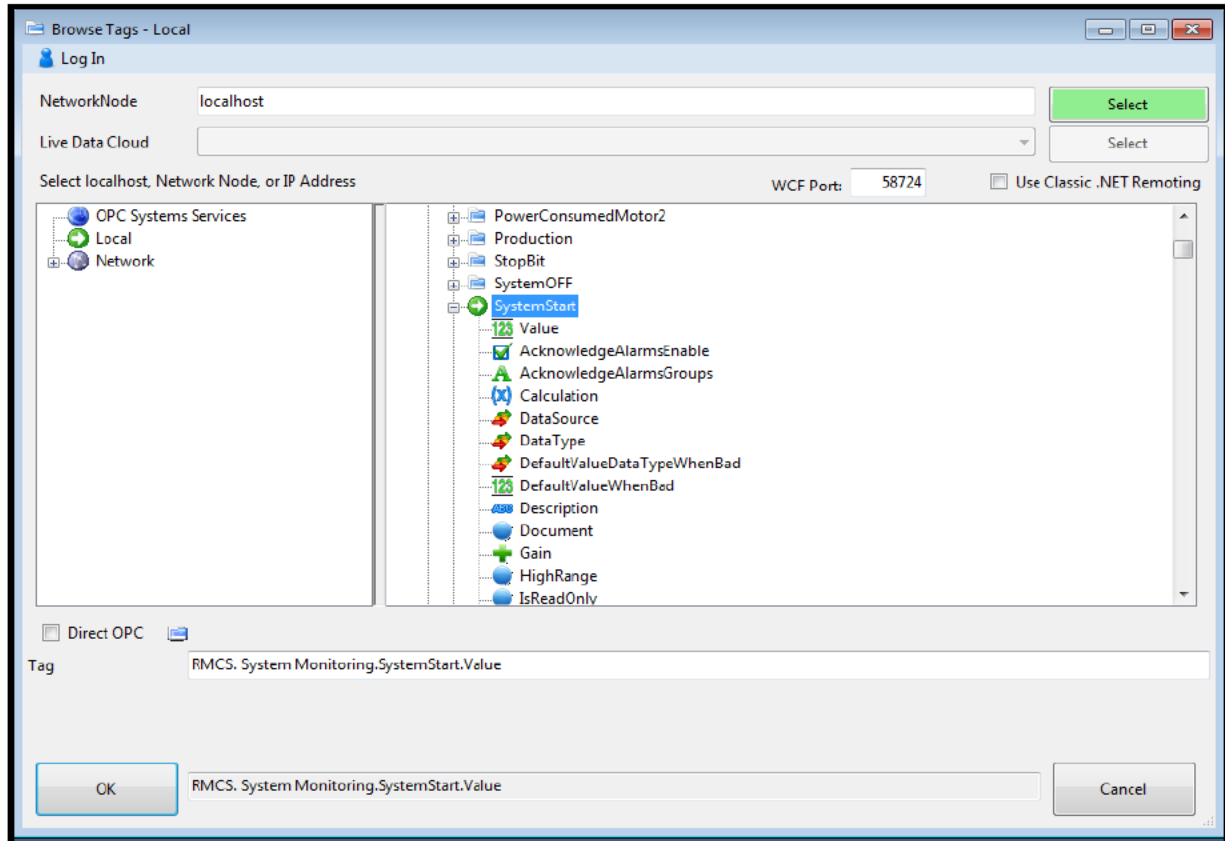


Figure 4.19: Browsing tags on a local host

When the “SystemStart” tag value is true (i.e. logic “1”) the back colour of the textbox will return lime indicating the system is running and when the value is false (i.e. logic “0”); the back colour of the textbox will return red indicating that the system has stopped.

An OPC web control of type web refresh is also used to refresh only the OPC web controls on the page - not the entire page - as refreshing the whole page would take up more time than just refreshing some part of the web page.

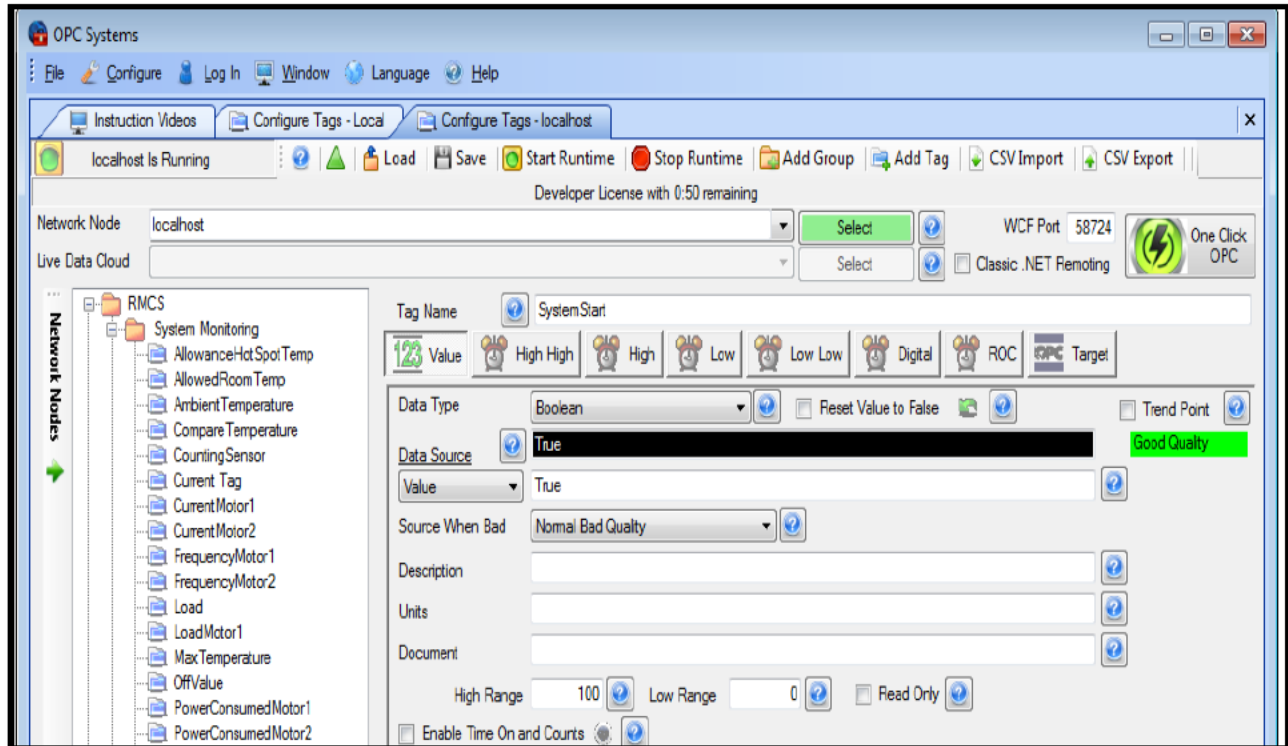


Figure 4.20: System Status tag configuration

The refresh rate is set to 100ms so the controls on the page will be refreshed every 100ms. This means that every 100ms second the OPC System.NET will fetch the new state of “SystemStart” tag and display it on the web page; this will go on continuously until the web page is closed.



Figure 4.21: System control panel

b) ***Current Consumed*** -The assembly system uses a motor as seen in Figure 4.26 in order to drive the conveyor belt and transport products from one assembly unit to another until finish. These motors use variable frequency speed drives that can drive motors from a low speed to its maximum. It is very important to monitor over-current of motors, because if a motor draws more current than its ratings, which can happen when the motor is overloaded with the task that is not properly sized for, this will greatly impact the operation of the motor. Every induction motor comes with a nameplate on it as seen in Figure 4.26. In this nameplates there are information about the specifications of this motor, which can be seen in Table 4.1.

Table 4.1: Motor Specifications

Type of Connection	Current	Speed	Voltage	Power	Insulation Class	cos ϕ
Δ	0.8A	50/60Hz	200/220V	0.09/0.10KW	F (155°C)	0.70
Y	0.5A	50/60Hz	346/400V	0.09/0.10KW	F (155°C)	0.70

The information provided in Table 4.1 explains that this motor can consume a maximum of 0.8 Amps or 0.5 Amps, depending on how it is connected to power supply. The connection type used for our motor is delta (Δ), so the maximum current this motor can consume with this connection is 0.8 Amps. One needs to monitor the current that is consumed by this motor so that it does not exceed 0.8 Amps. The motor operating temperature is determined by the National Electrical Manufacturer Association (NEMA).

NEMA has defined the temperature rise for electric motor in motors and generators, NEMA standard MG 1-1998 [37]. The insulation temperature class as seen in Table 4.1 is based on the overall temperature; for example, with this research we used Class F with windings system rated at 155°C. The normal maximum ambient temperature per NEMA is 40°C [38]. The temperature rise limit for Class F winding would be estimated at 115°C (155-40). This means that the motor temperature should not rise above 115°C when operational or non-operational as this would cause damage to the motor.

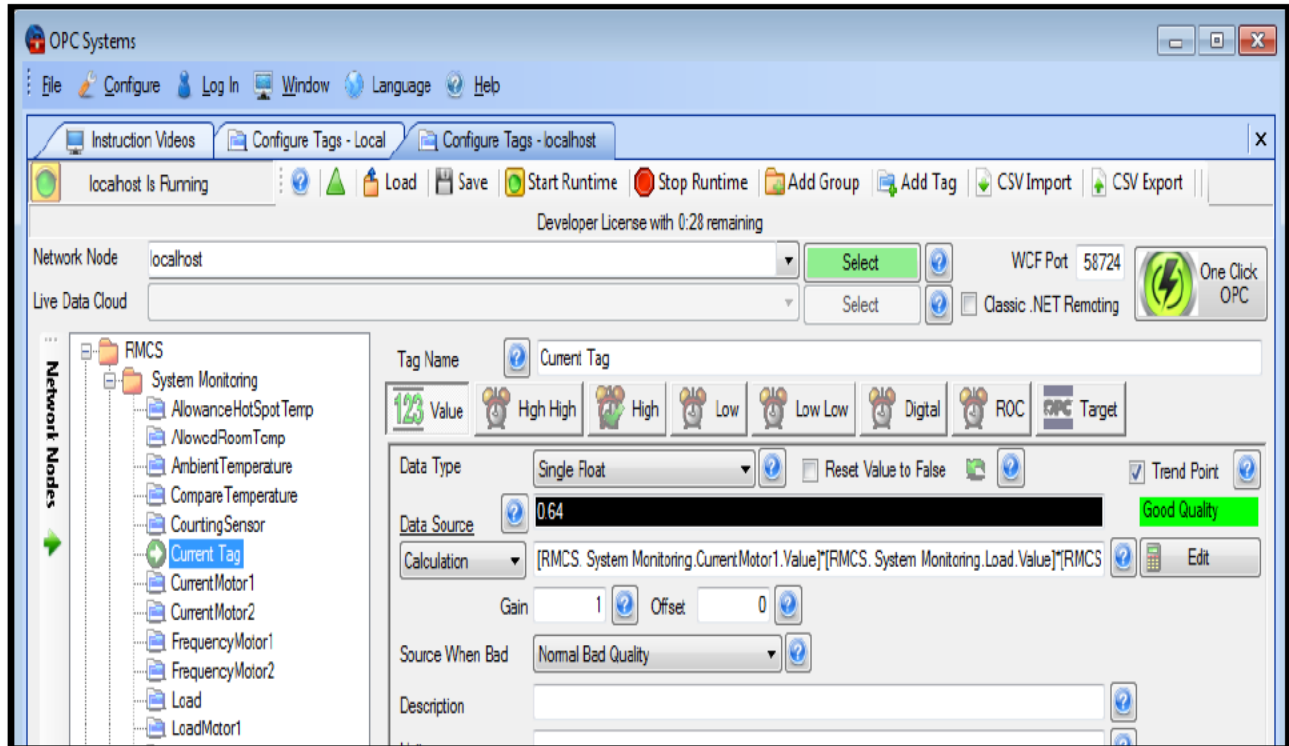


Figure 4.22: Current consumed tag configuration

To read the current drawn by each motor in the system, we connect to the OPC server using OPC System.NET service and read the current (A) tag. After reading the current drawn by each motor, this value is then compared with the rating of each motor; if this value exceeds the motor rating the system monitor will be alarmed.

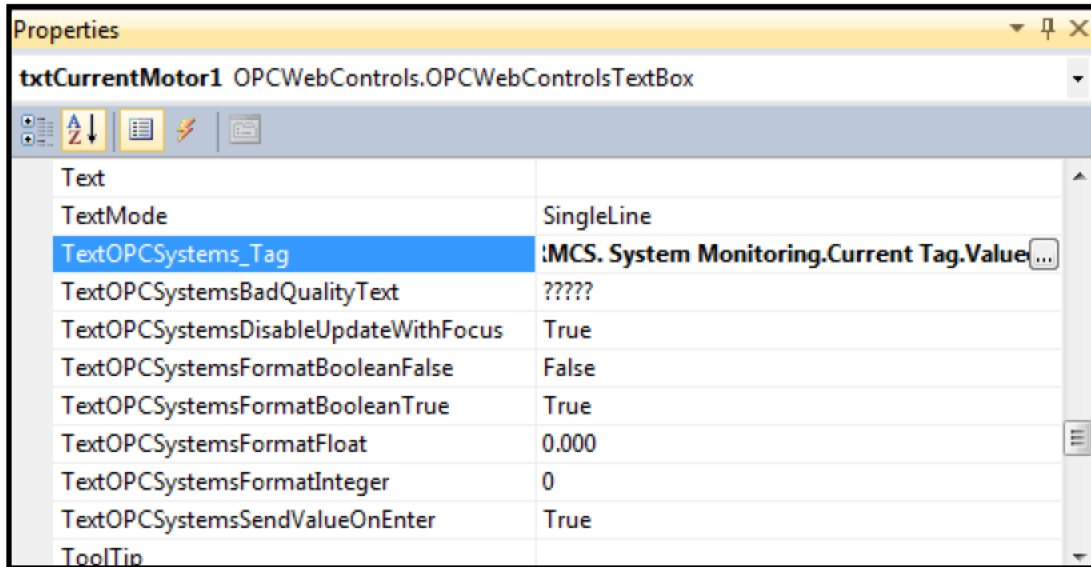


Figure 4.23: OPC Web Control Textbox Properties for current consumed

The first step is to configure tag for current consumed in the OPCSystem.NET server; this tag currently resides in a remote OPC server so we need to save it to our OPCSystem.NET server so that we can access it using ASP.NET technology. By using the configuration option “Tag Configuration” of the OPC System service, as mentioned before, we configured the Current consumed tag to be of group “System Monitoring” as seen in Figure 4.22 and to update the value of current every 100ms. So 100ms the OPC System services will go and fetch the updated data from this remote OPC server and display it on the tag configuration window. In order for the system to alert the system operator that the current consumed by motors has exceed 0.8Amps, the alarm function is enabled (as seen in Figure 5.12), which will be activated as soon as the current consumed value goes up to 0.81Amps.

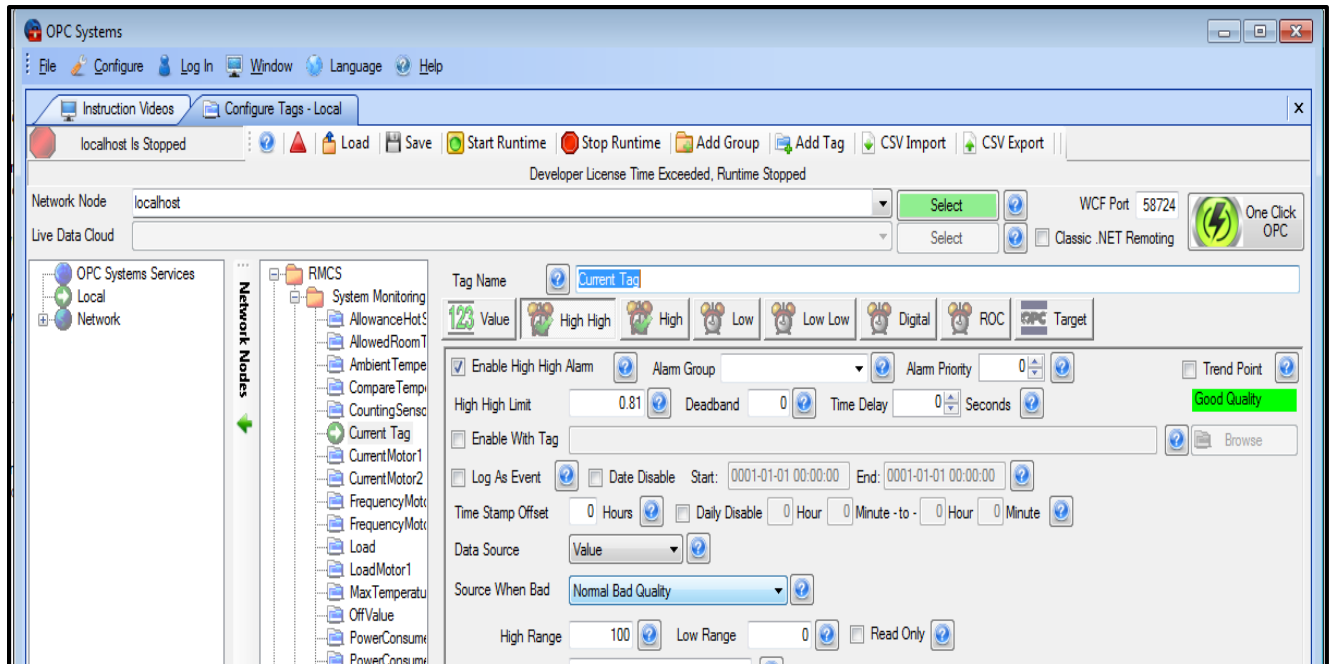


Figure 4.24: Current consumed alarm configuration

The OPC System.NET service can also be used with ASP.NET to fetch data and display it on a web application. In order to monitor the current consumed by the motors using a web interface, an OPC web control of type textbox is used and its property “TextOPCSys_tems_Tag” (seen in Figure 4.23) referenced to the Current tag value in the OPCSystem.NET server (as seen in Figure 4.25). The OPC web refresh control is also used and the refresh rate set to 100ms so that 100ms when the data change in the OPCSystem.NET server it can also be seen on the web application. For every half second, updated data about the current consumed by each on the assembling system will be displayed on a textbox.

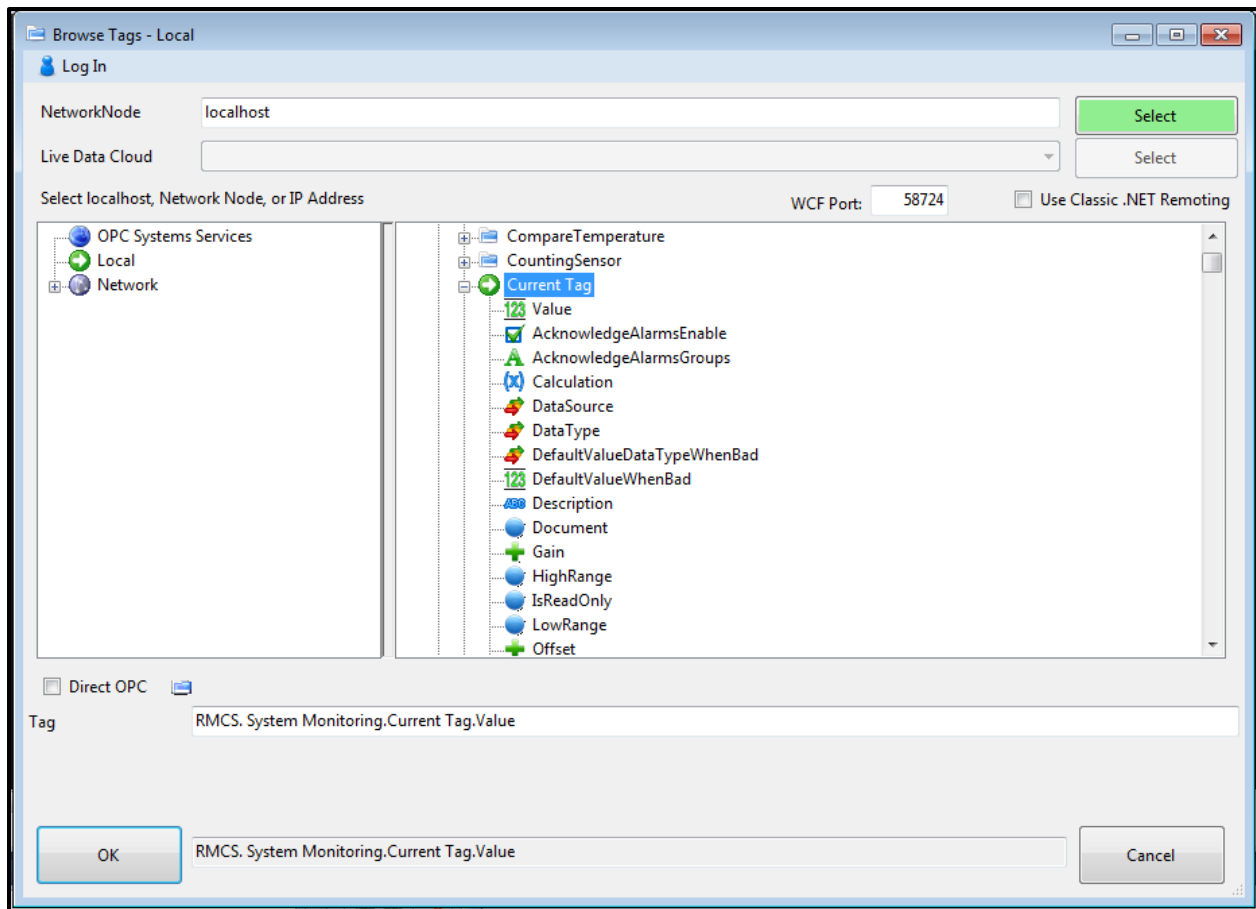


Figure 4.25: Browsing tags on a local host

However, whenever the system is switched off, the current consumed value should be returned to zero since we are only monitoring current consumed by motor only when the system is running. The state of the system is represented by a tag “SystemStart” with a value of “1” and “0” or true and false respectively. The true state of the tag “SystemStart” means that the system is running and false state means it has stopped.

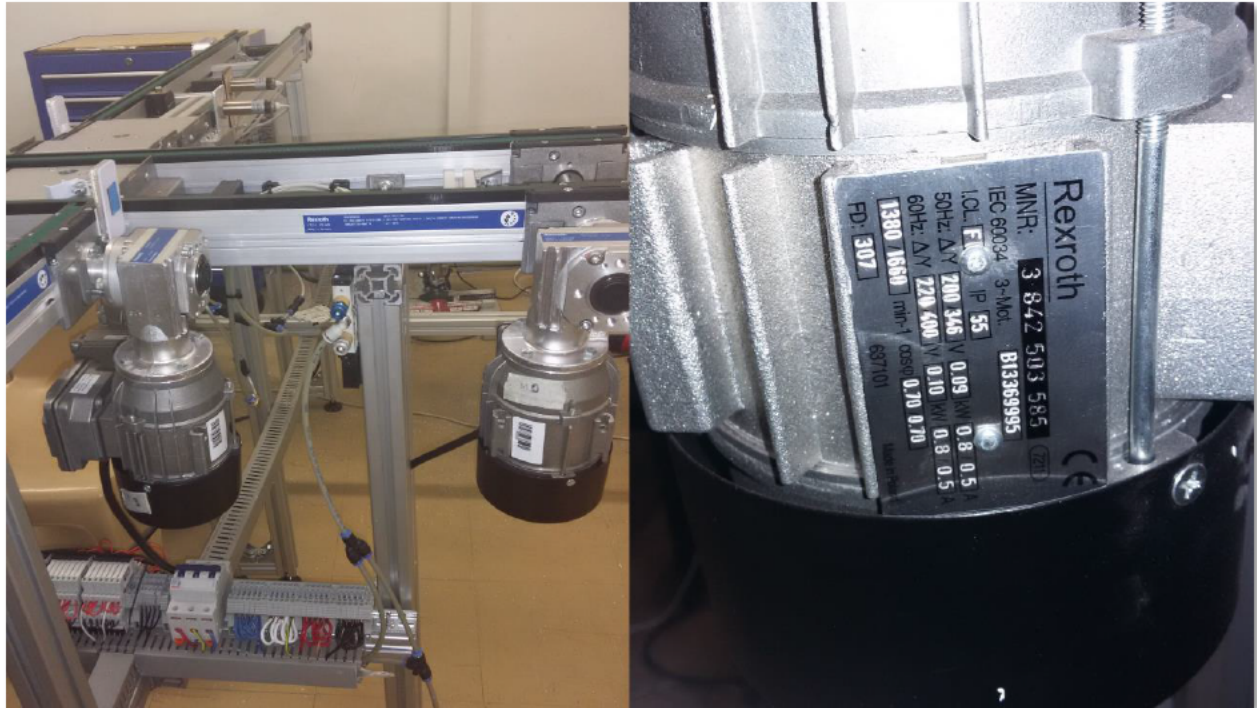


Figure 4.26: Three-phase induction motors used to drive the conveyor belt

In order to make sure that the “SystemStart” tag returns a zero when the system is not running, an arithmetic calculation was performed by multiplying the current value of the tag with the value of “CurrentMotor1” tag as seen in Figure 4.27. The result of this will return zero when system has stopped (logic “0”) or return the value of current consumed when the system is running (logic “1”).

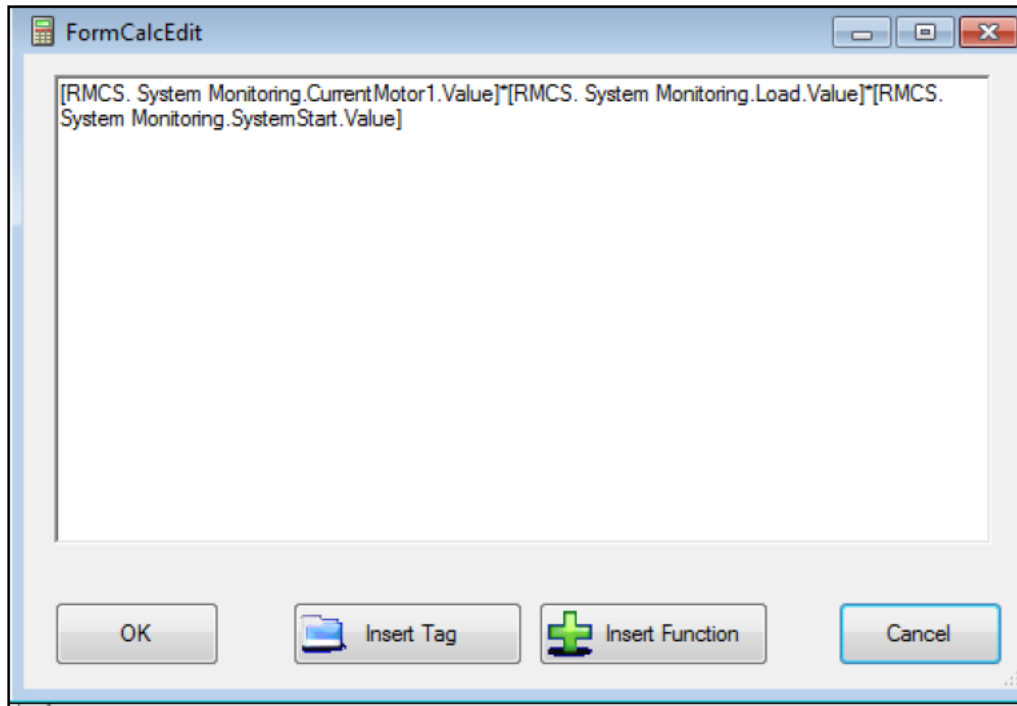


Figure 4.27: Arithmetic calculation to determine the state of the system

c) **Units Produced** – The number of units produced by the system needs to be monitored to check if there is any change in production from day to day. The proximity sensor is responsible for counting the number of completed products; this sensor is programmed and controlled by the PLC. For every new count, the total count is updated; the “new count” is the count for that current day and the “total count” is the total count of units the assembly has produced since its operation.

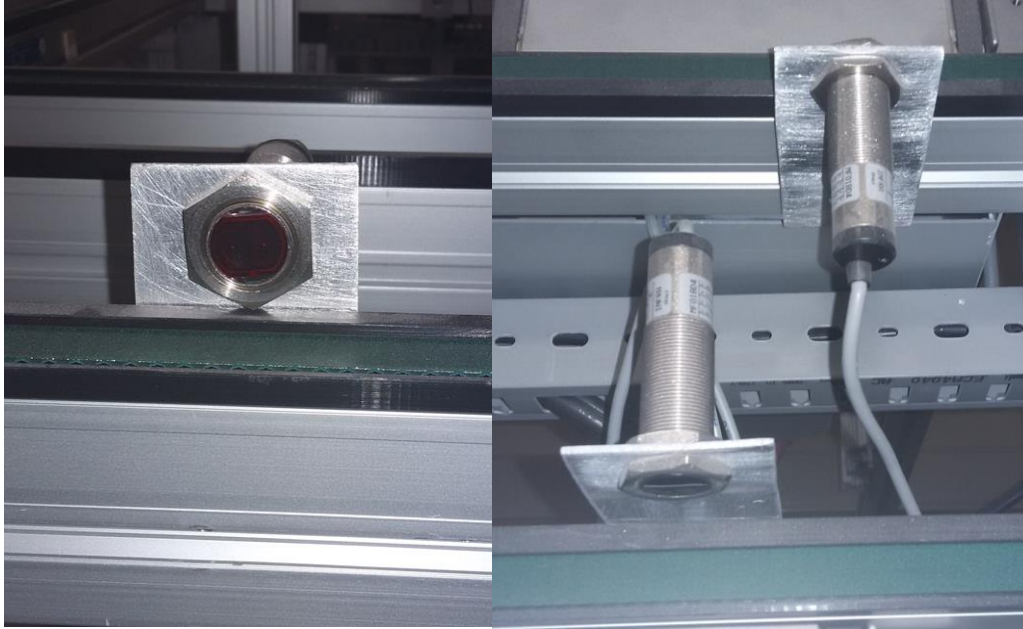


Figure 4.28: Proximity Sensor used for counting number of products produced.

The “total count” and the “new count” values are saved as tags in the remote OPC server; whenever the count of units is updated, it will also be updated in the OPC server. To read, refresh and display this data in real-time in a web application, the OPC System.NET service and ASP.NET web application is used. Firstly, the tag “Production” was configured using the same method as when creating the current consumed tag; the update rate is also kept at half a second.

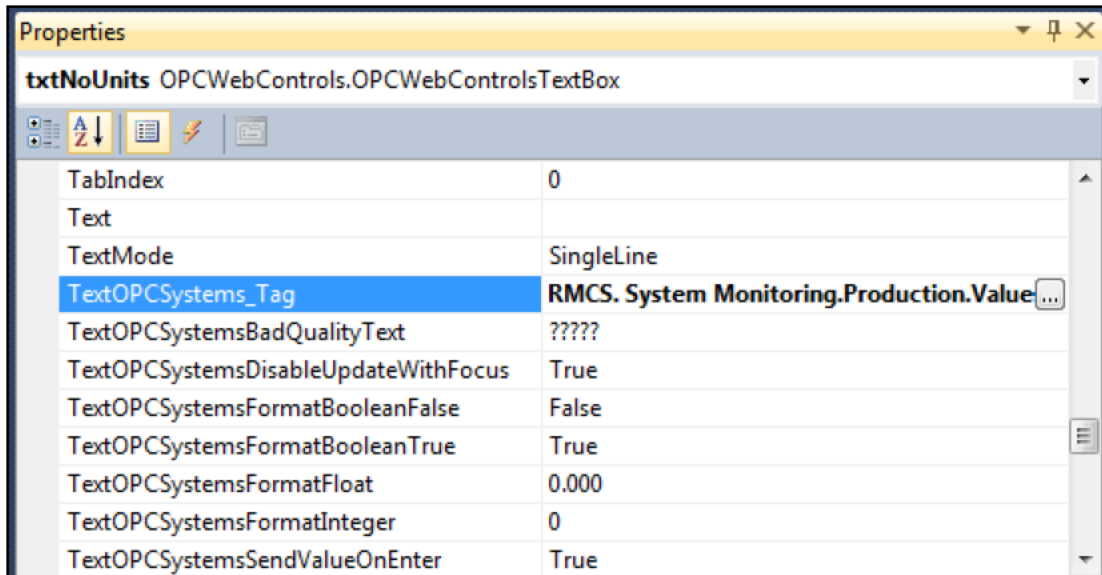


Figure 4.29 OPC Web Control Textbox Properties for number of units produced

The OPC Textbox web control is used to display the value of number of units produced and its “TextOPCSys..._Tag” property is referenced to the “Production” tag value, as seen in Figure 4.29. The web refresh control is also added on the web form and its refresh rate is set to half a second; this is to allow the page to refresh every half second on a continuously as long the web application is running. On web load, the value of the tag “Production” will be populated on the textbox and it will be refreshed every half second in order to obtain real-time data as much as possible. To make sure that the “Production” tag value returns zero only when system is not running the same method used for current consumed is also used here. The ‘Production’ tag value is multiplied by the “SystemStart” tag value. The “Production” value returned will be zero if the “SystemStart” value is zero or return the current “Production” if otherwise.

d) **Power Consumed** – It is crucial to monitor the power consumed by the system to check how much it will cost us to run the system every day. We need to check how much power is consumed to produce a certain number of products as it will not make business sense to consume so much energy only to produce one unit that when sold could not even cover the energy cost incurred in producing it. In addition to that, South Africa is current experiencing an energy crisis, so it is critical that we do not waste energy unnecessarily.

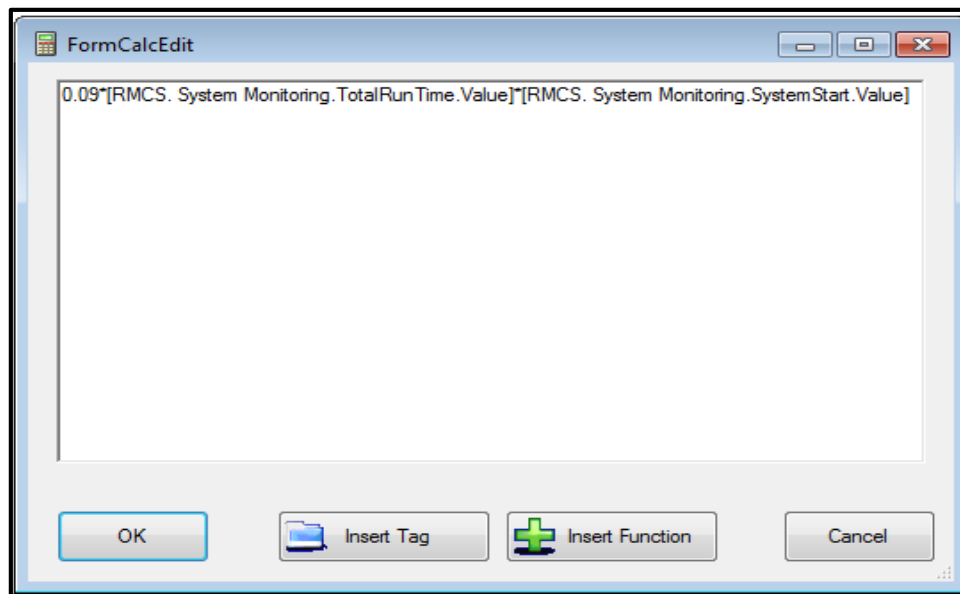


Figure 4.30: Calculating power consumed

To determine how much power, the system has consumed, one could concentrate only on motors and calculate how much power each motor is consuming in an hour and how many units are produced during that hour. According to the specifications given by the motor manufacturer as seen in (Table 4.1) this motors consumes at total power of 0.09KW when running at a frequency of 50Hz and it also consumes 0.10KW when running at a frequency of 60Hz.

If our motors are constantly running at a frequency of 50Hz for a full hour, then the power consumed for that is 0.09KW; if it happens that within that hour our motor changed to a different frequency, calculations for power consumed during that hour will include both frequencies and the time each frequency has occurred. The method used to calculate the power consumed by each motor is as seen in Figure 4.30. Before one can perform any calculations, the data tags are to be configured in the OPC System server. The data tags configured are as follows - the speed, power and time tags. Before the OPC System.NET service can perform calculations, it needs to check first how many hours the motor has been running and at what frequency the motor was running. After getting all this information, the service can then calculate the power consumed by the motors running - for certain hours and running at a particular frequency. The faster the motor is running and the longer it runs will result in more power consumption as compared to when the motor is running slower and for a short period of time.

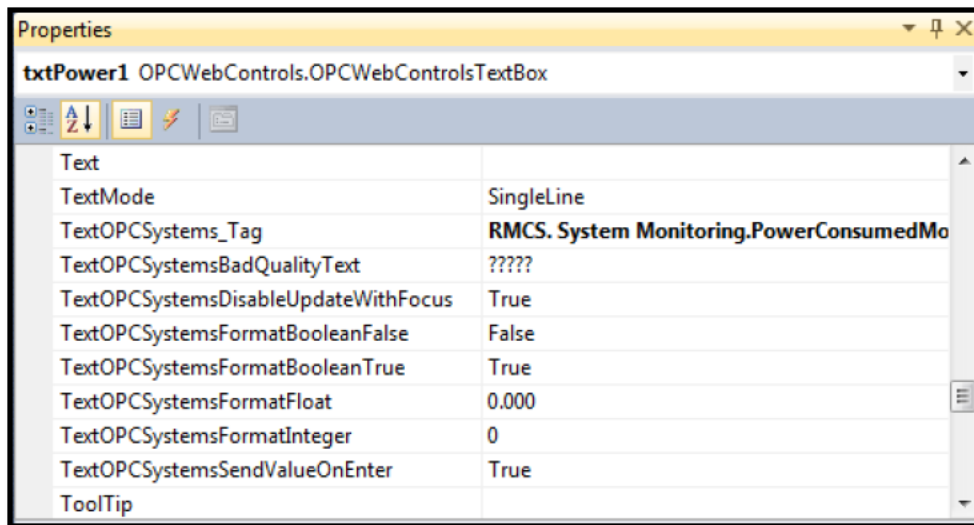


Figure 4.31: OPC Web Control Textbox Properties for power consumed

In order to monitor the power consumed by motors using a web application, the combination of OPC Web controls, OPC System.NET service and ASP.NET technology was used. The OPC Web control provides one with web controls that are OPC compliant and that can be programmed using ASP.NET technology. The OPC System.NET service allows our ASP.NET web application to connect to local or remote OPC servers.

The OPC web control that is used to monitor power consumed is of type textbox. This textbox is used to display the result of the calculated power consumed. The tag “PowerConsumedMotor” is configured in the OPC System.NET server to hold the value of power consumed by motors. In order to read the value of the “PowerConsumedMotor” tag and display it on the textbox, the “TextOPCSystems_Tag” property of the textbox was referenced to the value of “PowerConsumedMotor” tag - as seen in Figure 4.31. When the web application load on the browser, the textbox will be populated with the value of the “PowerConsumedMotor” tag, but this data will not be updated as the “PowerConsumedMotor” value updates in the OPC System.NET server. To allow this value to update on the web application as it updates on the OPC System.NET server, an OPC web control of type web refresh is used with an update rate of 100ms, so every 100ms seconds the OPC System.NET service will fetch the new update value of “PowerConsumedMotor” from the OPC System.NET server and display it on the textbox. The web refresh control will refresh the textbox continuously until the web application is closed.

e) **System Sensors** - The state of the sensors used by the system needs to be monitored around production cycle, because if not functional, this could result in faulty production or no production at all. If it happens that one of the sensors is not functioning properly, an alarm will be activated

to inform the system operator about the faulty sensor. These sensors could be the ones deciding what the assembly system should do, how it should do it and when it should do it. The on and off state of sensors should be monitored to make sure that the system operates correctly.

4.5 Historical Data Archiving System

The historical data archiving system's purpose is to log the behaviour of the system data as the system is in operation. This data is kept in the SQL database so that we can use it later for troubleshooting and analysing data trends.

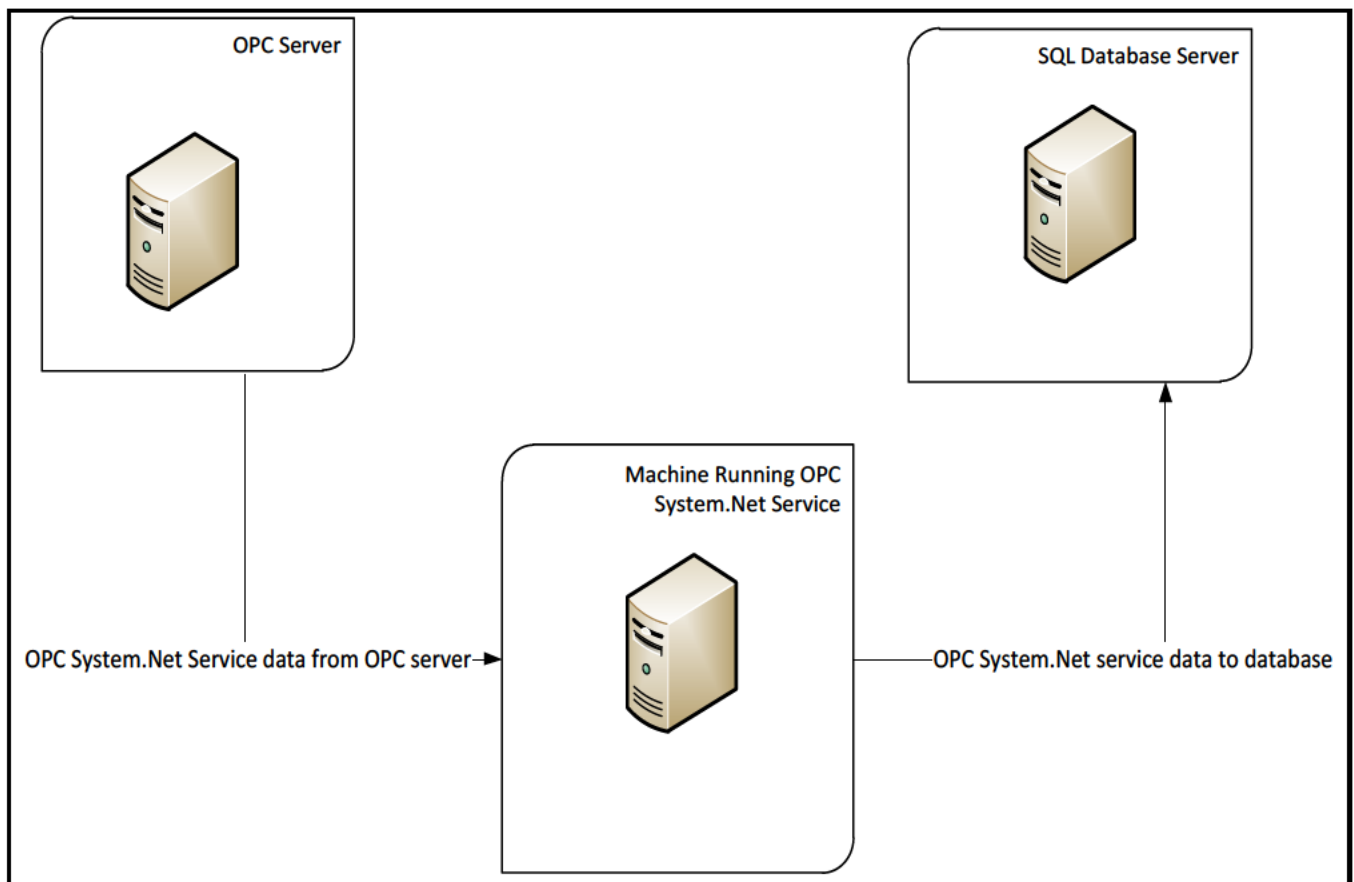


Figure 4.32: Architectural design for historical data archiving system

The OPC System.Net service running on a local machine gets the data from the local and the remote OPC server and saves it in the SQL database server, as demonstrated in Figure 4.32. To enable the service to archive the system data from the OPC server to SQL database server, a data logging configuration window was used, as seen in Figure 4.33 . The service gives one options on different kinds of database server that you can log our data into; for example, MSAccess, Oracle, ODBC and SQLServer. For this research project, SQLServer was chosen as the database server because of its reliability and integrity, but mostly because of the excellent skills available for operating and managing this server.

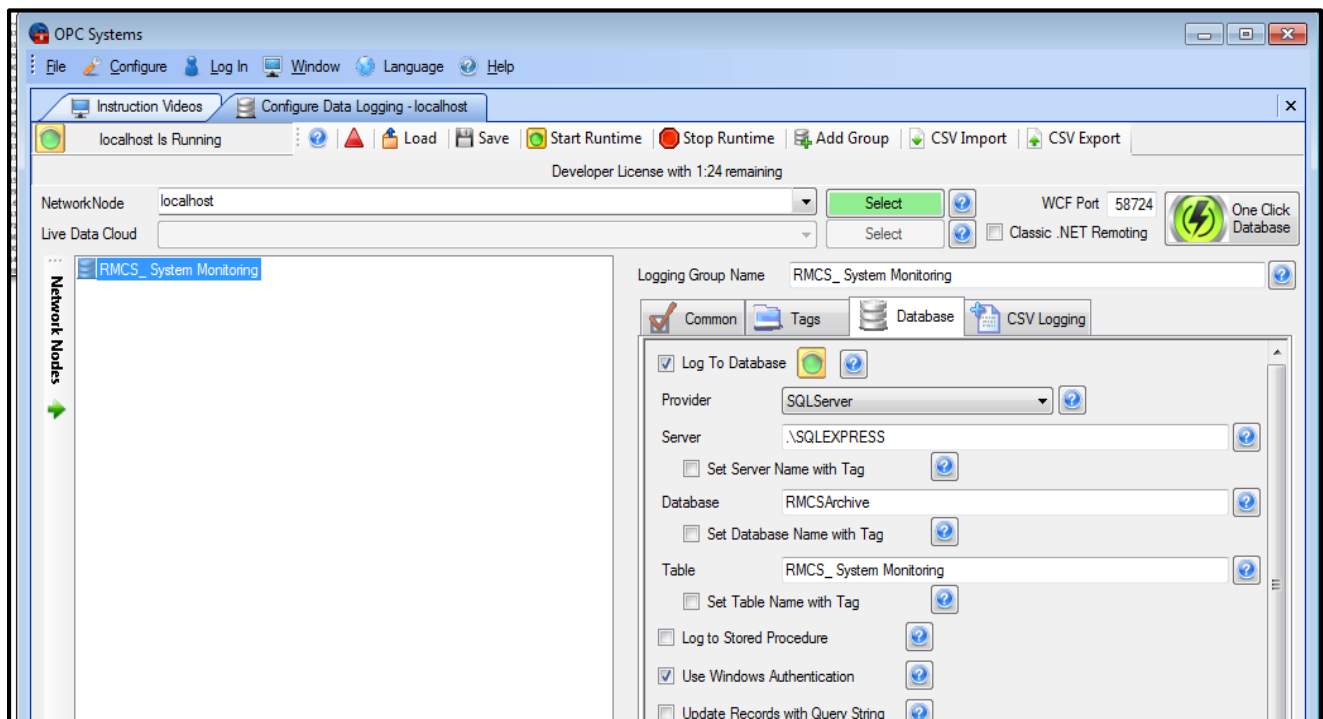


Figure 4.33: OPC System data logging configuration window

The service will automatically create the database and a database table for us with the name of our choice as seen in Figure 4.33. This database can be accessed using SQL Server Management Studio

application or Visual Studio. When the database is accessed using the SQL Server Management Studio, the user's credentials are requested in order to access the database server as seen in Figure 4.34. This is to ensure only authorized users can access the database, thus ensuring data security and improving data integrity.

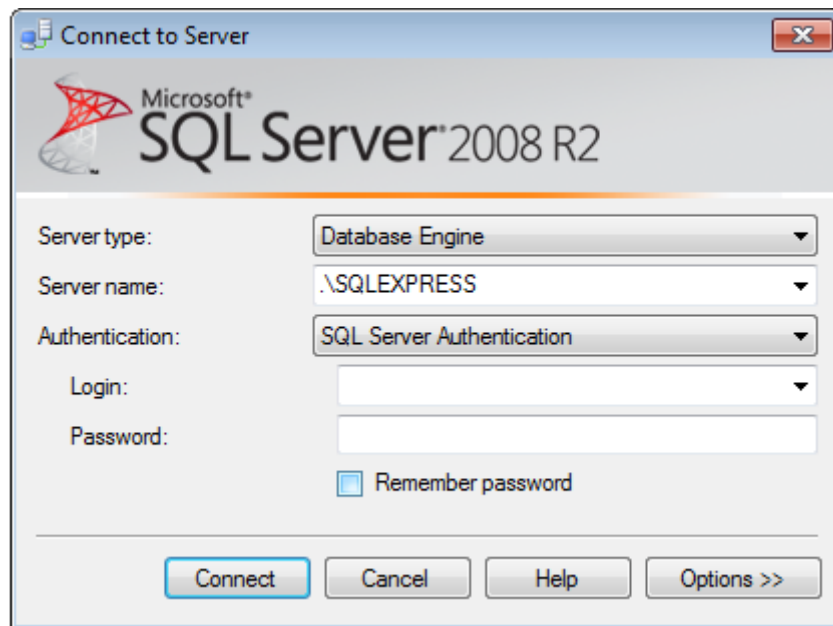


Figure 4.34: SQL Server Management Studio user access window

Only authorised users will be able to access the “RMCSArchive” database as seen in Figure 4.35. The “RMCSArchive” database has table “RMCS _System _Monitoring”, which keeps the record of the system data behaviour. Each record has time and date indicating exactly when change of data happened.

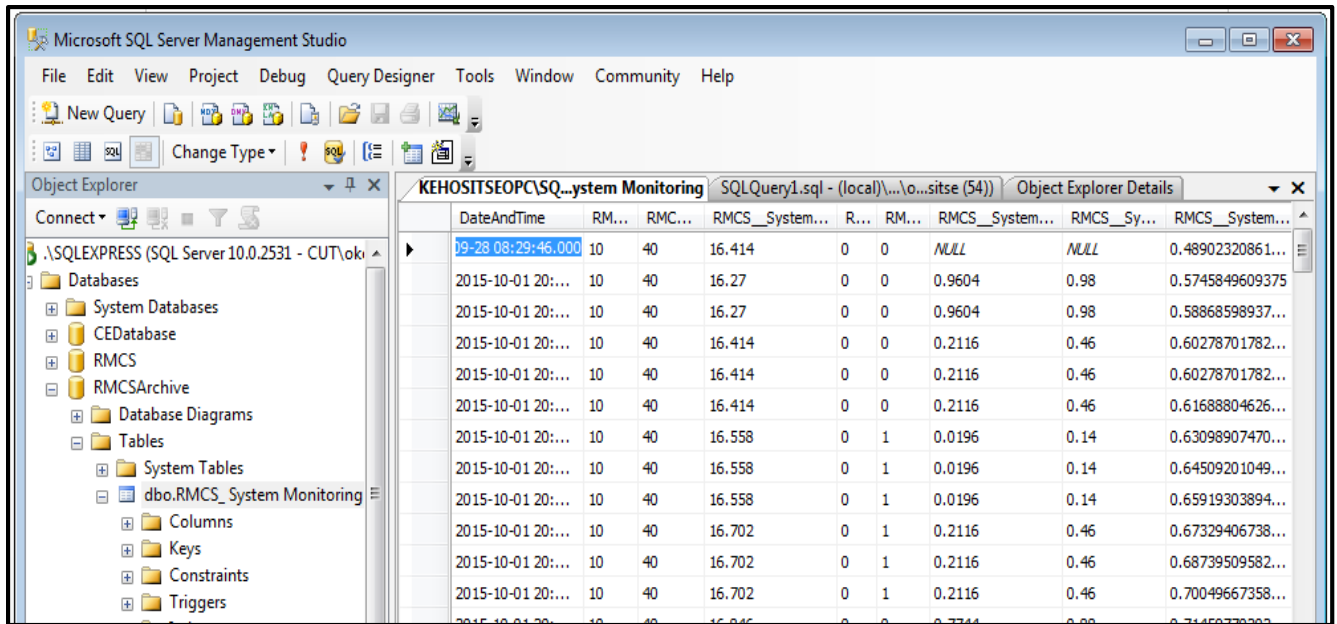


Figure 4.35: Data logged in the database

4.6 Control System

The control system uses a three-tier architectural design as seen in Figure 4.17. This architecture is comprised of a client, web server and OPC server. The system has only two control options: start and stop. By starting system, the system connects to OPC server using OPC System.NET service to write value “1” to the “SystemStart” tag in the OPC server and by stopping the system a value of “0” is written to the “SystemStart” tag. The structure block of the controlled system is shown in Figure 4.36. The controlled object consists of an asynchronous three-phase motor. An encoder is fixed to the axis of the three-phase motor with the purpose of measuring the speed and the position of the motor axis. Block D represents a frequency inverter with implemented Proportional Integral Derivative (PID) regulation of the motor speed. The frequency inverter is connected to the PLC via the PROFIBUS communication network. PLC is connected to the

computer (OPC server on the network), via PROFINET network. Another computer is connected to the server with the role of a client.

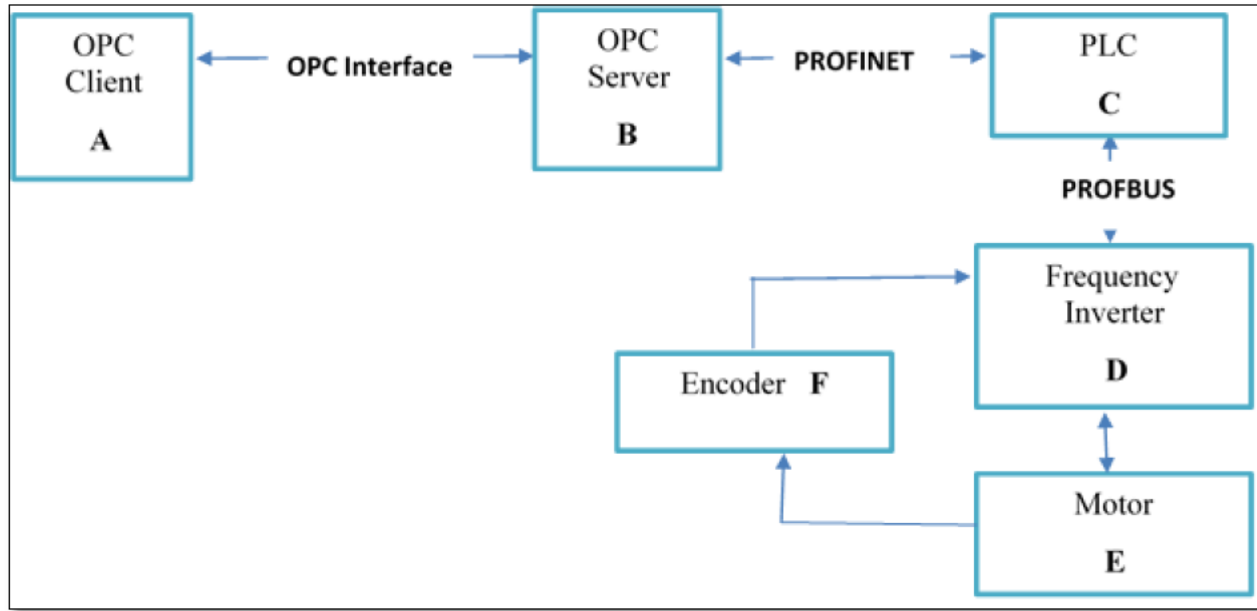


Figure 4.36: Control system structure

The desired speed input is done in the graphic panel (HMI), as seen in Figure 4.37, which is connected to OPC server via PROFINET. Apart from the momentary speed of the motor, it is possible to keep track of several other important parameters on the HMI (the motor voltage, temperature of the motor and etc.). If the desired speed is assigned on the HMI, then it is sent to the server via the local network. The server redirects it to the PLC while the PLC sends that same information to the frequency inverter, which tracks the difference between the current and the desired speed, and using that difference - generates the controlling signal for the motor, all by using the PID control.



Figure 4.37: Human Machine Interface

4.6.1 Start Function

The start function is implemented by combining the OPC System.NET service, OPC web controls and the ASP.NET technology. The OPC web control of type button is used in order to set the value of the “SystemStart” tag from false to true on event click. This was done by setting the property “SetValueOPCSysDescreteValue” of button “btnStart” to true as seen in Figure 4.38. The “SetValueOPCSys_Tag” property of button “btnStart” is referenced to the “SystemStart” tag value as seen in Figure 4.38, to indicate to the OPC System.NET service which tag value on the OPC server to set when the button “btnStart” is clicked.

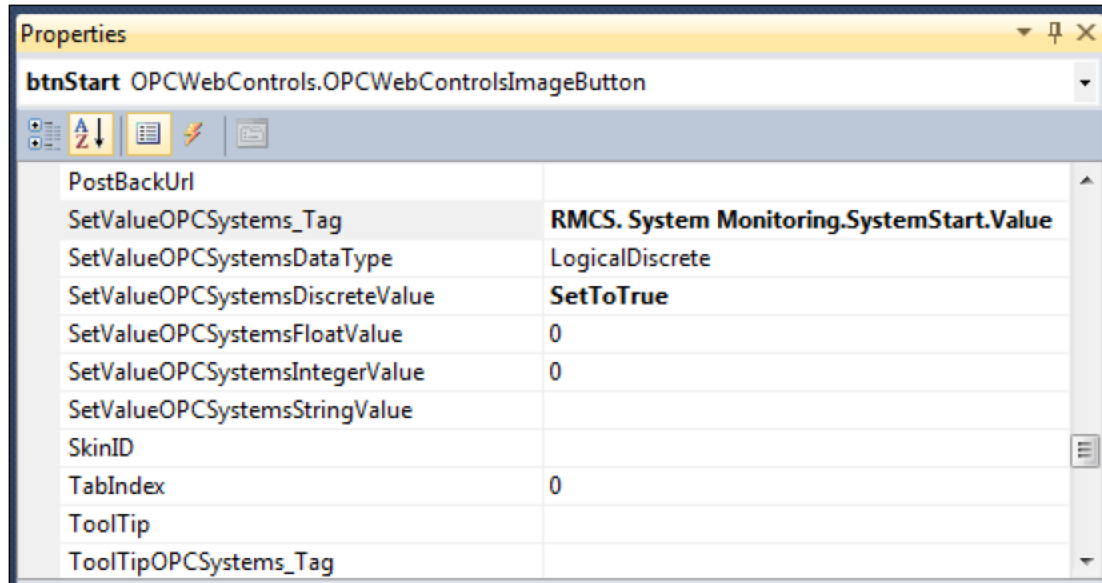


Figure 4.38: OPC web control start button properties

The OPC web control of type web refresh is also used to refresh the web application so that we can get and set the most current value of the “SystemStart” tag. The refresh rate of the web refresh control is set to 100ms. When the web application loads the most current value of the “SystemStart” tag will load and after 100ms the “SystemStart” tag will be new value if any change occurred. On button “btnStart” click event, the “SystemStart” tag will be set, and its value will be updated within 100ms in the OPC System.Net server, which means the system will start running.

4.6.2 Stop Function

The start function is implemented by combining the OPC System.NET service, OPC web controls and the ASP.NET technology. The OPC web control of type button is used in order to set the value of the “SystemStart” tag from true to false on event click. This was done by setting the property “SetValueOPCSysytemsDescreteValue” of button “btnStop” to false, as seen in

Figure 4.39. The “SetValueOPCSytems_Tag” property of button “btnStop” is referenced to the “SystemStart” tag value, as seen in Figure 4.38, to indicate to the OPC System.NET service which tag value on the OPC server to clear when the button “btnStop” is clicked.

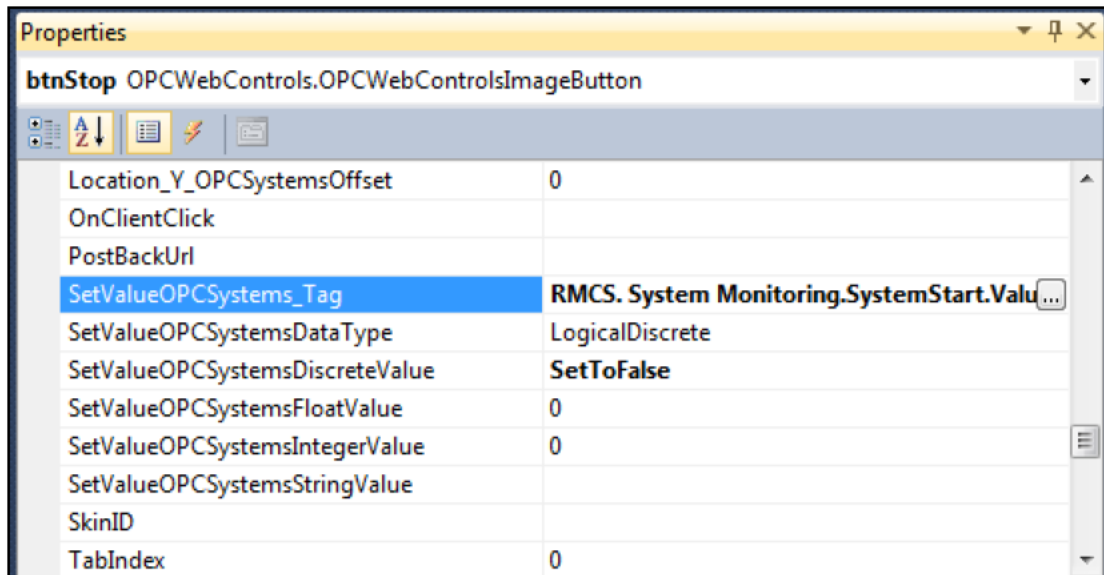


Figure 4.39: OPC web control stop button properties

The OPC web control of type web refresh is also used to refresh the web application so that we can get and set the most current value of the “SystemStart” tag. The refresh rate of the web refresh control is set to 100ms. When the web application load, the most current value of the “SystemStart” tag will load and after 100ms the “SystemStart” tag will be new value if any change occurred. On button “btnStop’ click event the “SystemStart” tag will be cleared, and its value will be updated within 100ms in the OPC System.Net server, which means the system will stop running.

4.7 Conclusion

In this chapter, the design and implementation of a Remote Monitoring and Control System of an assembly system at RGEMS were discussed. The architectural design of the system is developed using Microsoft Visio 2010 and the implementation of the system is developed using the combination of ASP.NET technology and OPC technology.

The RMCS can be used with real-time supervision and control of the induction motor and other devices working with it. The motor parameters that the systems monitors are the ambient temperature, the motor temperature, frequency, voltage and current. These parameters are monitored while the motors are running and functional. We can remotely start these motors with the RMCS and we also have an option of selecting the speed at which the motors can run at. The supervision of how many units are produced by the assembly system, is also performed by the RMCS.

5. CHAPTER V

Results

5.1 Introduction

In this chapter, the operation of the system is explained using visual pictures so that it can be understood how the systems operate in reality and not only conceptually. We are going to explain how each and every part of the project work as a system. The user access to the system is discussed, and the system control and monitoring through an RMCS web application is also explained.

5.2 Access Control

In order to ensure that system data integrity and reliability are not affected by malicious user(s), a user access control system that will control who can use the system and who cannot use the system needed to be developed. The access control system has four sub-systems that will make sure that only authenticated and authorized users will be able to access the system. These sub-systems are as follows: login page, user registration page, user management page and change password page.

- a) *Login page* - the login page makes sure that the user(s) does not use the system without identifying themselves.
- b) *User registration page* - this page makes sure that the user is registered before he/she can use the system.
- c) *Change password page* – this page makes sure that only registered users can change their password using the RMCS web application password creation rule.

- d) *User management* - this page makes sure that only administrators can delete the users' accounts or modify their role on the system.

5.2.1 Login Page

Accessing the system is controlled by the login system, as seen in Figure 5.1. This is the start page of the RMCS web application and no user can access the system without going through the login page.

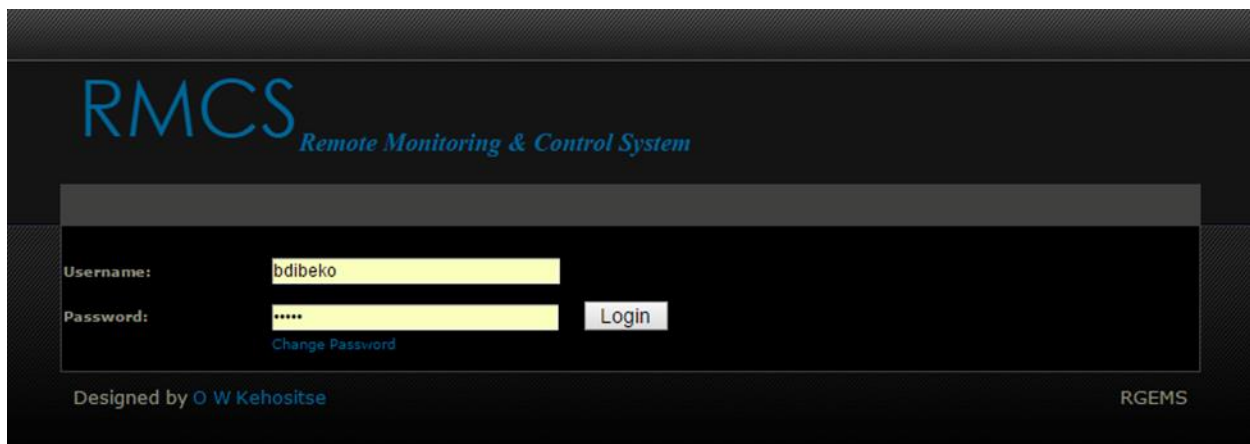
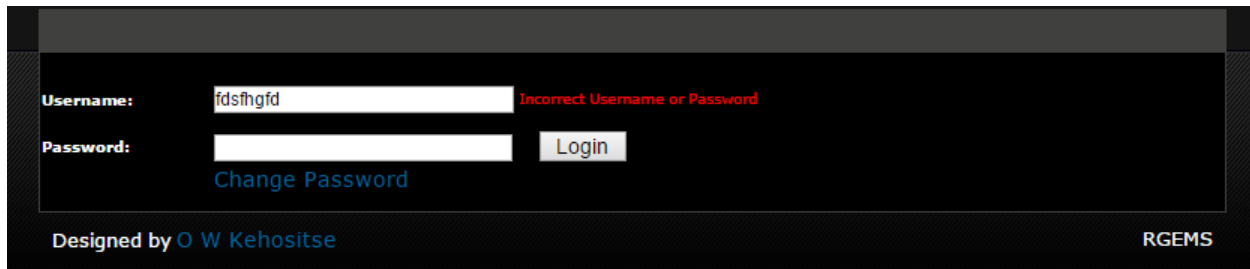


Figure 5.1: RMCS Login Page

The login page will request the user to provide his/her credentials and these credentials will be validated against the one in the database. If the user's credentials are not valid, an error message will appear as seen in Figure 5.2 and the user will not be granted access to the system. In case user(s) have forgotten their password, they can let the administrator know so that the administrator can delete their current account and create a new one for them.



The image shows a login form with a dark background. The 'Username:' field contains the text 'fdsfhgfd'. To the right of this field, the error message 'Incorrect Username or Password' is displayed in red. Below the 'Username:' field is the 'Password:' field, which is empty. To the right of the 'Password:' field is a 'Login' button. Below the 'Password:' field is a link that says 'Change Password' in blue text. At the bottom left of the form, it says 'Designed by O W Kehositse' and at the bottom right, it says 'RGEMS'.

Figure 5.2: Incorrect credentials error

5.2.2 User Registration Page

When registering the user, the administrator needs to fill all the fields required on the registration form as seen in Figure 5.3 before submitting the form. If all fields are not filled before submitting the registration form, the system will throw out a required field error message as seen in Figure 5.4 requesting the user to fill the fields before submitting the form.

RMCS
Remote Monitoring & Control System

Sign out

Monitor System Control System Register User Manage Users Change Password

Personnel/Student No:

Name:

Surname:

Role: Administrator

Register

Designed by O W Kehositse RGEMS

Figure 5.3: RMCS User registration page

On the contrary, if all fields are filled with all the required field type, the system will register the user and generate a username and a temporary password as seen in Figure 5.5. The user(s) can use the password for as long as registered, but they also have an option of changing their password if they find the one generated for them too complex to remember. User(s) can do this by visiting the change password page as seen in Figure 5.7. Users can only change their password after they have been registered on the system.

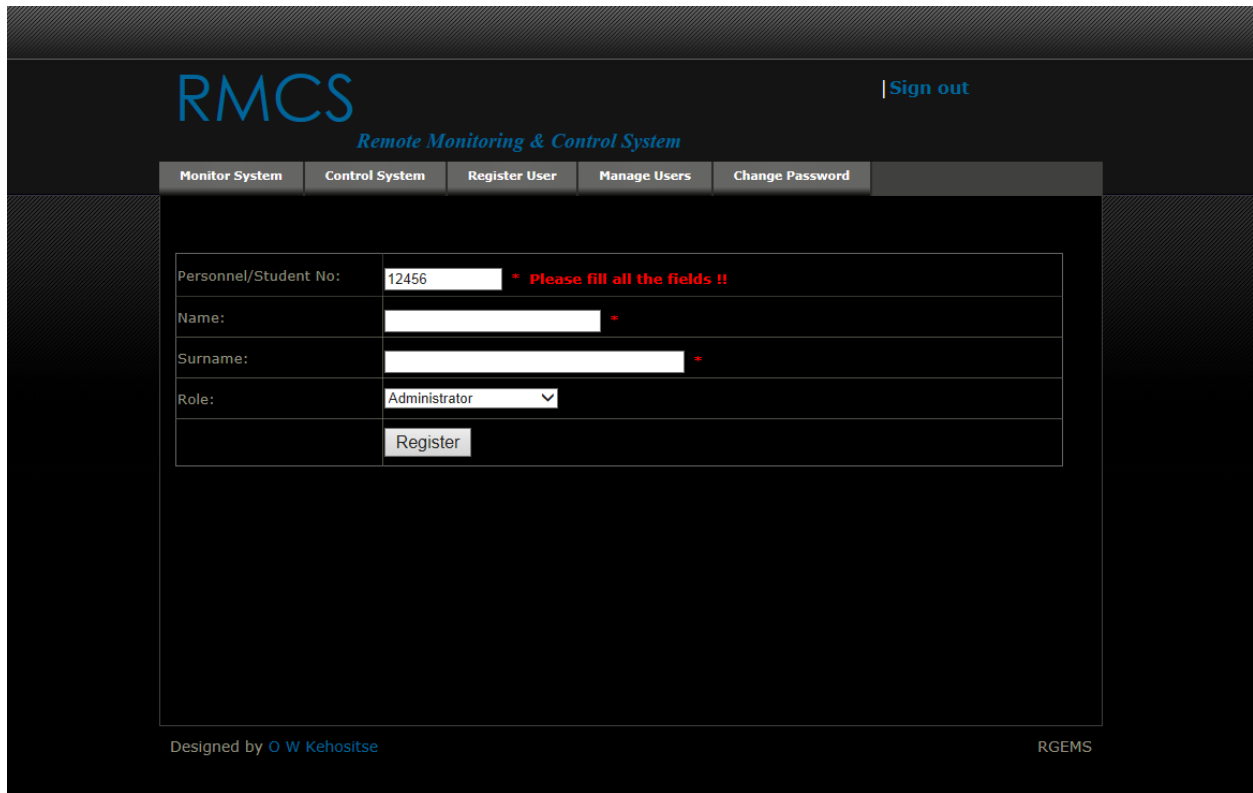


Figure 5.4: RMCS required field error message

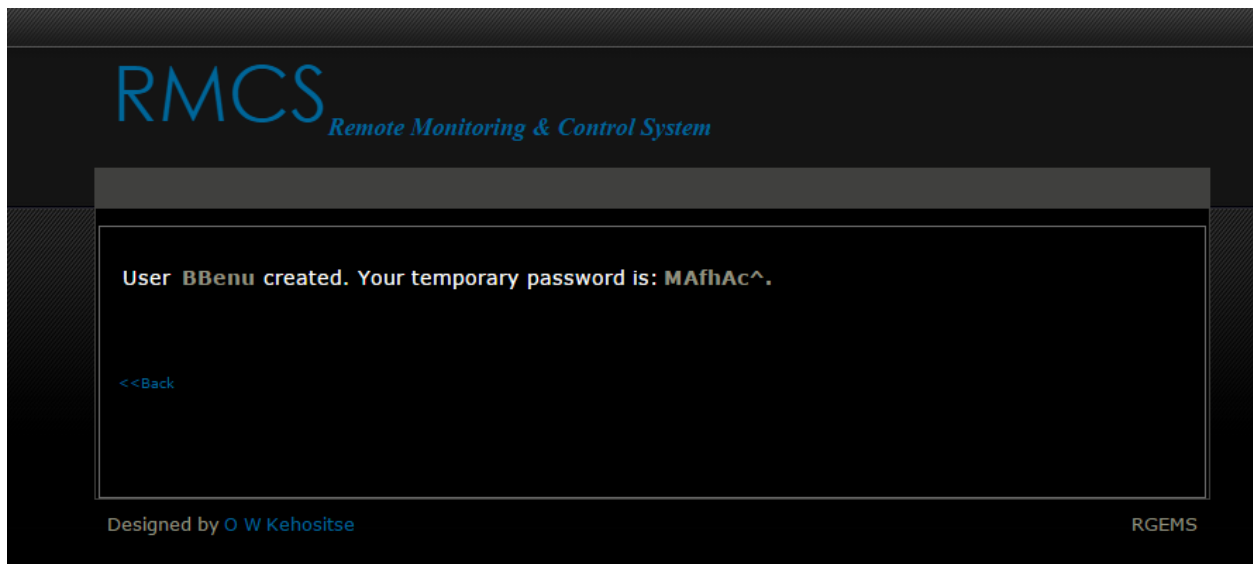


Figure 5.5: RMCS creates username and password

The user needs to know their temporary or old password and their personnel or student number in order for them to create a new password of their choice. Before the system allows a user to create a new password, it first validates if the data entered in the “Personnel/Student No” text field is of a type integer if true; then it checks if that personnel or student number exists in the database; if it does not exist null record error will be thrown out, as seen in Figure 5.6 not allowing the user to change the password. If all the fields of the change password form are not filled, the system will not allow the form to be submitted.

Personnel/Student No: * User with ID :125636 does not exist

Old password *

New Password: *

Retype New Password: *

Designed by O W Kehositse RGEMS

Figure 5.6: Null record error message

The RMCS web application password creation rule states that the password must be eight characters long with one special character. If the password does not conform to this rule, a new password cannot be created until the new password meets the requirements of RMCS web application password creation rule. The new password must also match with the retyped new password; if it does not match, the user(s) will not be allowed to submit the change password form.

RMCS
Remote Monitoring & Control System

Personnel/Student No:	<input type="text"/>	*
Old password	<input type="text"/>	*
New Password:	<input type="text"/>	*
Confirm New Password:	<input type="text"/>	*
	<input type="button" value="Change"/>	

Designed by O W Kehositse

RGEMS

Figure 5.7: RMCS Change password page

5.2.3 User Management Page

The user management page displays a list of all registered users in a table as seen in Figure 5.8. The table lists the user(s) with their ID, name, surname and role that they have on the system. The administrator can edit this table however he/she likes; he/she can also change the role of the user, modify the name, surname, and ID if it was entered incorrectly. After any data has been modified, the update button as seen on Figure 5.8 needs to be clicked to ensure that the new modified data is saved on the database. If the administrator has any change of mind about the modification of data, a cancel button can be pressed.

RMCS | Sign out
Remote Monitoring & Control System

Monitor System | Control System | Register User | Manage Users | Change Password

		User ID	Name	Surname	Role
Update		44544	onsa	vader	Administrator
Cancel					Administrator
Edit	Delete	121345	koko	seleke	User
Edit	Delete	134545	howza	motho	Guest_User
Edit	Delete	213695	kefi	styles	User
Edit	Delete	252434	kukla	kusta	Administrator

1 2 3 4 5 6

Designed by O W Kehositse | RGEMS

Figure 5.8: RMCS User management page (Edit Mode)

The administrator can also delete the user in case the user can no longer be allowed to access the system or the user is not an active member of RGEMS. To delete, the administrator can just press the button delete and the confirmation of deletion message will pop asking the administrator if he/she is sure about his/her action before continuing to delete.

RMCS
Remote Monitoring & Control System

| Sign out

Monitor System Control System Register User Manage Users Change Password

		User ID	Name	Surname	Role
Edit	Delete	13425	onalenna	kehositse	Administrator
Edit	Delete	13426	Jacob	Kruning	User
Edit	Delete	14256	Thabo	Bihi	User
Edit	Delete	15426	Sabata	Seekoei	Guest_User

Designed by O W Kehositse

RGEMS

Figure 5.9: RMCS user management page

5.3 Monitoring System Page

The monitoring page monitors the important parameters of the induction motor - such as current consumed, power consumed and temperature. The monitoring page also monitors the number of products produced on that day and the functionality of the sensors in the system. On the top-left corner of the RMCS Monitoring page, there are three buttons, namely, Monitor, Alarms and Trends as seen in Figure 5.10, which will be discussed in the next section. Just below these buttons there are two text fields captioned Room Temperature and Maximum Allowed Room Temperature. The Room Temperature text field displays the current reading of the room temperature and the Maximum Allowed Room Temperature text field displays the allowable maximum temperature in the room. This data is displayed in real-time.

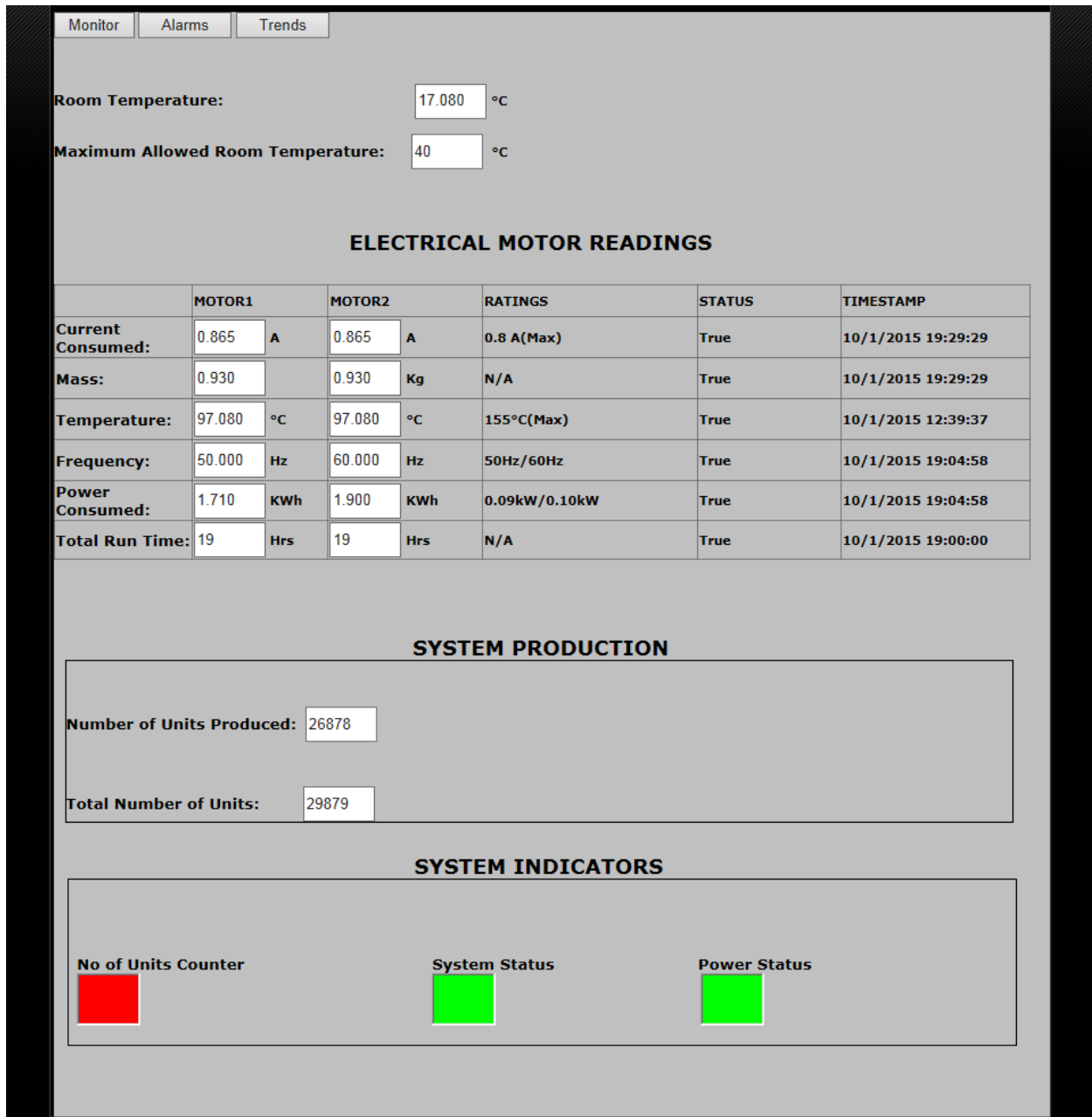


Figure 5.10: RMCS monitoring page (When the system is running)

Below the Maximum Allowed temperature field is a table with the readings of the induction motor in real-time. The parameters that are monitored are current consumed, temperature, frequency, power consumed, mass and total run time of the motor. These readings are found in column

MOTOR1 and MOTOR2. The column **TIMESTAMP** display the timestamp of each of the parameters. The column **STATUS** displays the status of the parameters and column **RATINGS** display the ratings of the motor according to its nameplate seen in Figure 4.26.

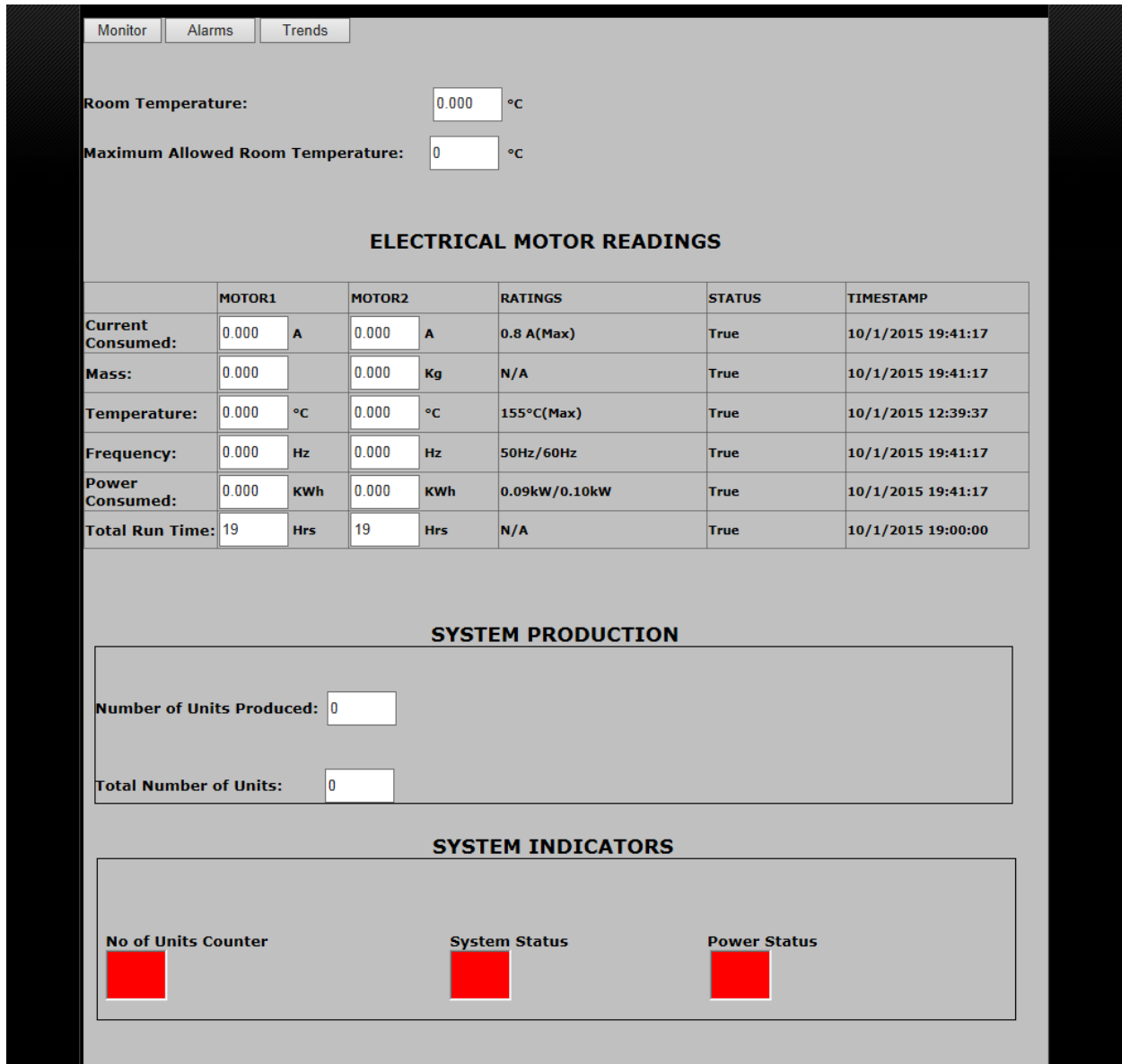


Figure 5.11: RMCS monitoring page (When the system is on halt)

Just below table ELECTRICAL MOTOR READINGS there is a field-set with the name SYSTEM PRODUCTION that contains the text field “Number of Units Produced” and “Total number of units”. The “Number of Units Produced” text field displays the number of units the assembly system has produced in a day and the Total Number of Units displays total number of units the assembly has produced since its existence.

Below SYSTEM production there is the field-set SYSTEM INDICATORS. This field-set contains three indicators which are “No of Units Counter”, “System Status” and “Power Status”. The “No of Units Counter” indicators represent the sensor that is used to count the number units produced by the assembly system; it will from red to green once it has sensed a finished product. The “System Status” indicator indicate the state of the system to be on /running when it is green and when it is red it shows that the system is not running. The “Power Status” indicator indicates if the system is powered or not; the indicator will turn red if system not powered and turn green when system is powered.

When the power is switched off and the system is not running, the monitoring page will return all zeros on its text field as indication that the system is not running as seen in Figure 5.11. The state of the SYSTEM INDICATORS will also change to red to indicate that the power has been shut down. Real-time values of the motor will not be able to be read since the motor will be switched off. Only historical values will be read.

5.3.1 Alarms and Events System

The alarm and events system of the assembly system logs the following types of information:

- *General information* -The current consumed by motors has exceeded its ratings, motor temperature has exceeded its maximum or the ambient temperature is too high.
- *State change update* – The motor states change, like start and stop.
- *Warning information* – Component-wise information change to a warning level, like temperature: motor temperature is about to reach maximum level.
- *Critical information* – System shutdown, power failure and so on.

The screenshot shows a web-based alarm management system. At the top, there are navigation tabs: 'Monitor System', 'Control System', 'Register User', 'Manage Users', and 'Change Password'. Below these, there are sub-tabs: 'Monitor', 'Alarms', and 'Trends'. The 'Alarms' tab is active. Underneath, there are buttons for 'Ack All', 'RealTim' (highlighted in green), and 'History'. The main area displays a table of active alarms with the following data:

DateTime	NetworkNode	Text	Type	AlarmValue
2015-10-01 7:30:50 PM	localhost	Temperature is too High	High High	101.881999969482
2015-10-01 7:30:47 PM	localhost	Current is too High	High	0.810000002384186
2015-10-01 7:30:47 PM	localhost	Load is too High	High High	0.9
2015-10-01 7:14:29 PM	localhost	localhost in demo evaluation	System	0
2015-10-01 6:03:50 PM	localhost	Alarms OPC Alarm Demo Mode	System	0

Figure 5.12: Alarm management system (Alarm active)

The user(s) or a system operator will be alarmed as seen in Figure 5.12 requesting the system to be shut down immediately if any device is not operating accordingly or if it has exceeded its specified ratings. The state of each alarm is colour coded; if an alarm is highlighted in red as seen in Figure 5.12, it indicates that the alarm is active and it needs attention; if purple, it indicates that the alarm is inactive; and if yellow, it indicates that the alarm has been acknowledged.

The screenshot shows the RMCS interface with a navigation menu at the top. The main content area displays an 'Alarms' tab with a table of active and acknowledged alarms. The table has five columns: DateTime, NetworkNode, Text, Type, and AlarmValue. The first row is yellow, the second is red, and the third is yellow.

DateTime	NetworkNode	Text	Type	AlarmValue
2015-10-01 7:38:29 PM	localhost	Load is too High	High High	0.59
2015-10-01 7:14:29 PM	localhost	localhost in demo evaluation	System	0
2015-10-01 6:03:50 PM	localhost	Alarms OPC Alarm Demo Mode	System	0

Figure 5.13: Alarm management system (Alarm acknowledged)

The alarm control seen in Figure 5.12 provides the system operator with real-time alarms and historical alarms; each alarm has a date and time of when it has occurred, and the cause of the alarm is also revealed in this control. Whenever an alarm event occurs, a system operator will have to acknowledge the alarm and if the alarm is successfully acknowledged, the acknowledged alarm will be highlighted in yellow - as seen in Figure 5.13.

A flashing red colour is used to capture this system operator attention of new alarms that are activated. Once the alarm has been acknowledged, the colour will change to a solid yellow and still be noticeable. The return to normal condition, which means the alarm condition no longer exists, the solid colour purple is used. From my experience, I have found it best to experiment with colours to see how things look, and determine whether or not the information can be seen clearly. With the right combination of colours, one can see at a glance the status of all equipment standing some distance away from the screen. Once the colour standards have been selected, then everyone knows what colours indicate before approaching the screen to see the details.

5.2.4 Trends and History System

Looking back to previous data to discover trends is an invaluable tool in helping improve our process and troubleshoot potential issues. That is why the high-performance industrial historian functionality is included in the system. A trend graph is a visual representation of past and current activity [39]. It provides an effective way of displaying process data. It builds a picture over time of how a variable (such as temperature, current or load) is changing or how a device or process is

performing. We can monitor current activity as it happens as seen in Figure 5.14 and scroll back through time to view the trend history as seen in Figure 5.15.

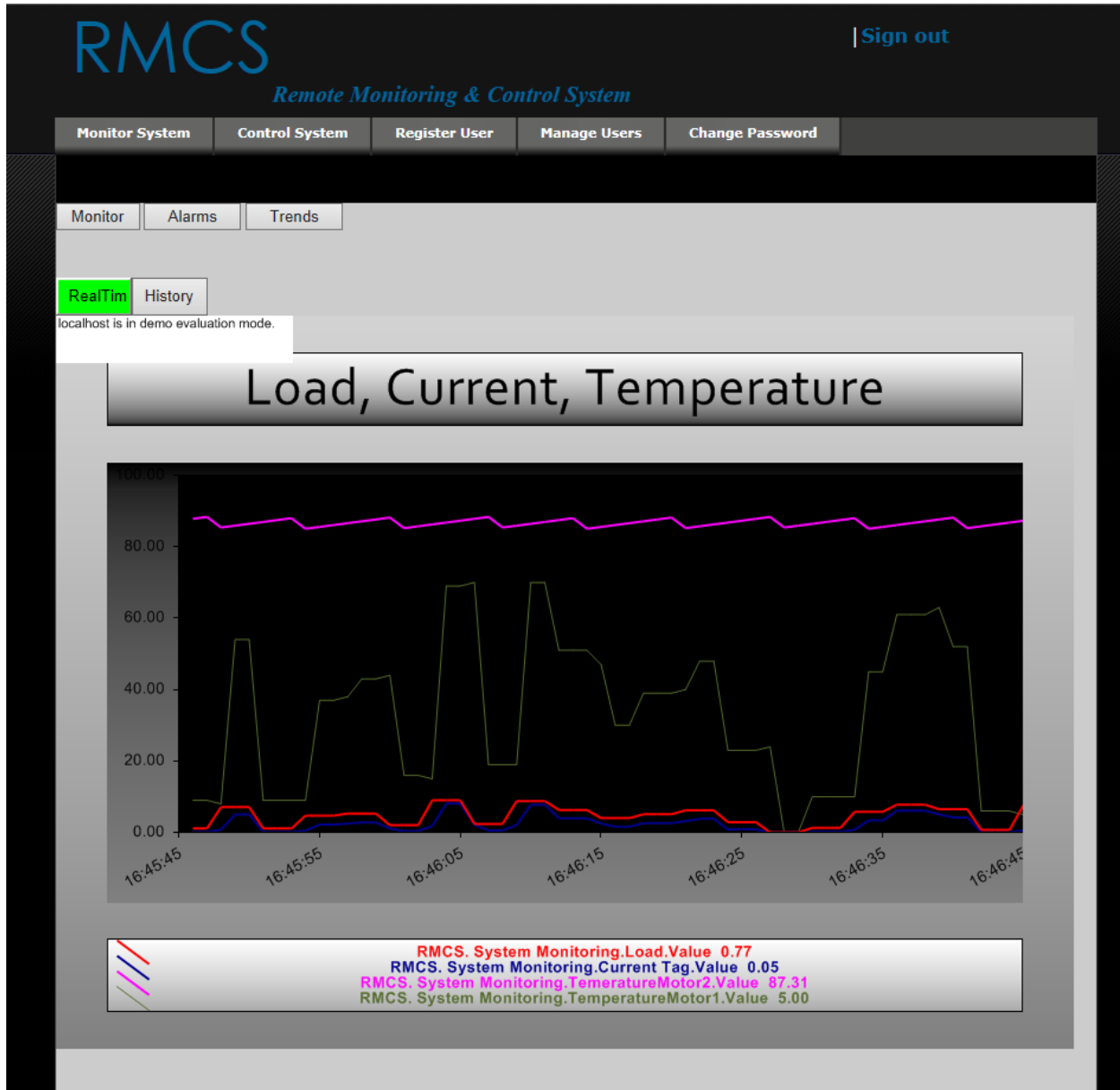


Figure 5.14: Load, Current and Temperature data trends in real-time

As values of variables changes over time, or as events happen, the graph moves across the page, as seen in Figure 5.14. The latest values are always displayed. We can scroll back using the button “History” as seen in Figure 5.15 to display past values of the variable (or process). Historical data collection continues even when the display is not active. We can switch between the pages without affecting the trend graphs. Trend data acquisition and storage of data (in trend history files) continue even when the display is not active. To store trend historical data, the SQL database server and the Comma Separated Value (CSV) file is used, as seen in Figure 5.17 and Figure 5.16 respectively. The system logs process data in the CSV file and saves the file on the local computer.

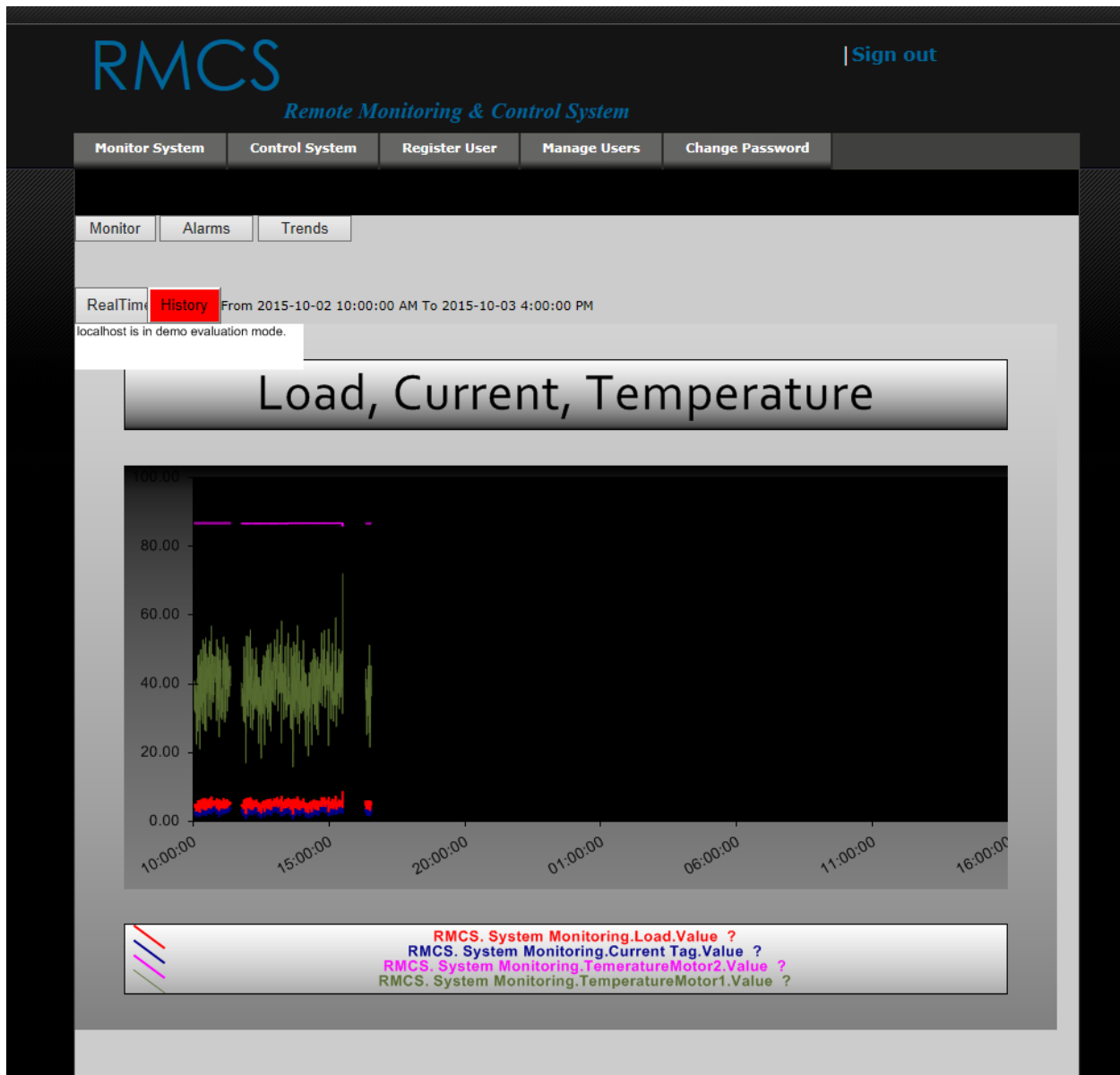


Figure 5.15: Load, Current and Temperature historical data trends

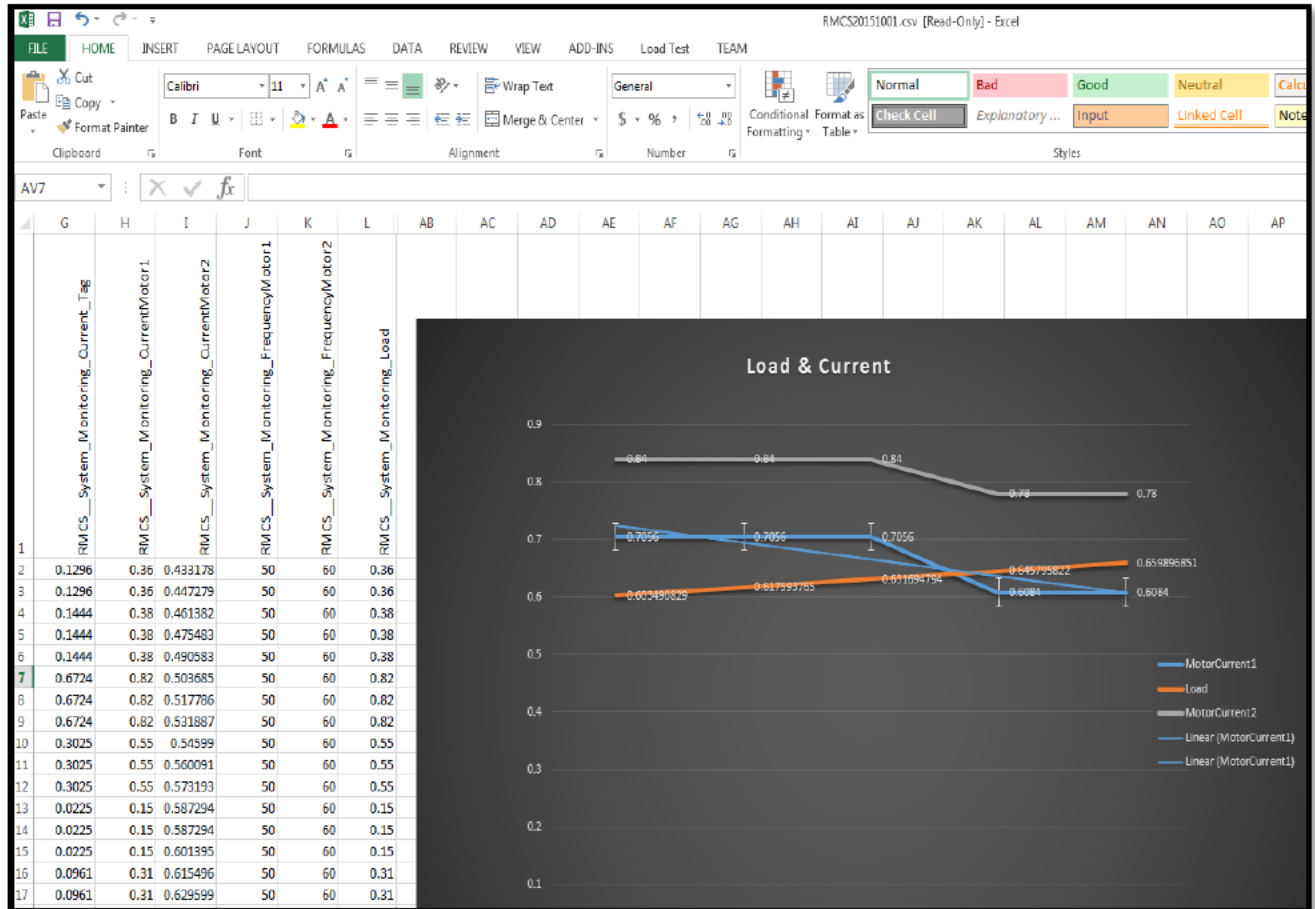


Figure 5.16: Data archived in a csv file

The SQL database server (historical database) is used to store data values with time and date stamps so that the information can be retrieved by tag name, process area, and alarm group and so on.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'dbo.RMCS_System Monitoring'. The main window displays a query result table with the following data:

DateAndTime	RM...	RMC...	RMCS_System...	R...	RM...	RMCS_System...	RMCS_Sy...	RMCS_System...
19-28 08:29:46.000	10	40	16.414	0	0	NULL	NULL	0.48902320861...
2015-10-01 20:...	10	40	16.27	0	0	0.9604	0.98	0.5745849609375
2015-10-01 20:...	10	40	16.27	0	0	0.9604	0.98	0.58868598937...
2015-10-01 20:...	10	40	16.414	0	0	0.2116	0.46	0.60278701782...
2015-10-01 20:...	10	40	16.414	0	0	0.2116	0.46	0.60278701782...
2015-10-01 20:...	10	40	16.414	0	0	0.2116	0.46	0.61688804626...
2015-10-01 20:...	10	40	16.558	0	1	0.0196	0.14	0.63098907470...
2015-10-01 20:...	10	40	16.558	0	1	0.0196	0.14	0.64509201049...
2015-10-01 20:...	10	40	16.558	0	1	0.0196	0.14	0.65919303894...
2015-10-01 20:...	10	40	16.702	0	1	0.2116	0.46	0.67329406738...
2015-10-01 20:...	10	40	16.702	0	1	0.2116	0.46	0.68739509582...

Figure 5.17: Data logged in the database

5.4 Control System Page

The control system page is used to control the operation of the assembling system by using two buttons, the start and stop. When the system operator wants to start the motor, he/she must first select the speed he wants to run the motor at and then press the start button. The motor will start and the green Light Emitting Diode (LED) will switch on indicating that the system is running, as seen in Figure 5.18.

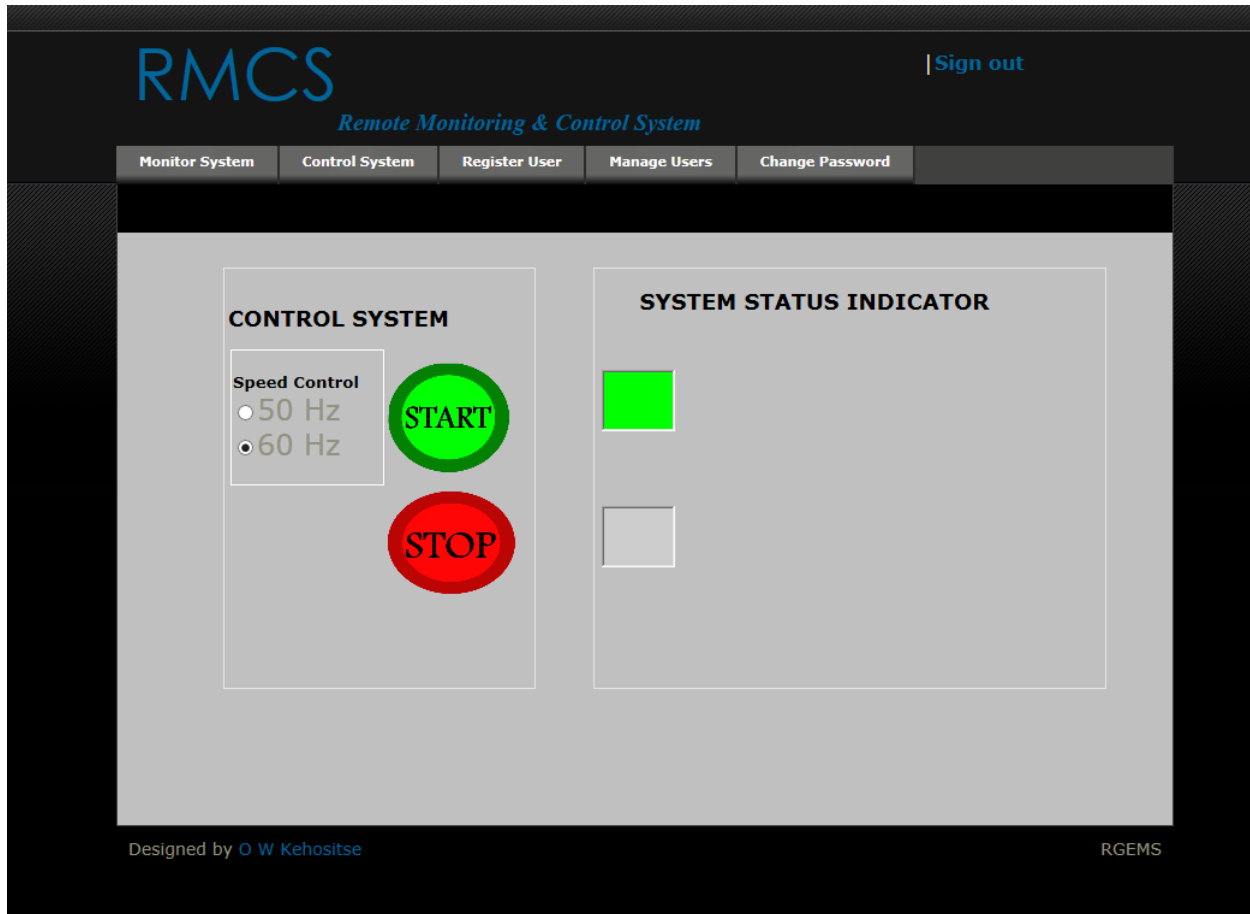


Figure 5.18: RMCS control page indicating that the system is running

The system operator can stop the motor from running by just pressing on the stop button and the motor will be stopped. The control system page will indicate to the system operator that the system has stopped switching on the red LED as seen in Figure 5.19.

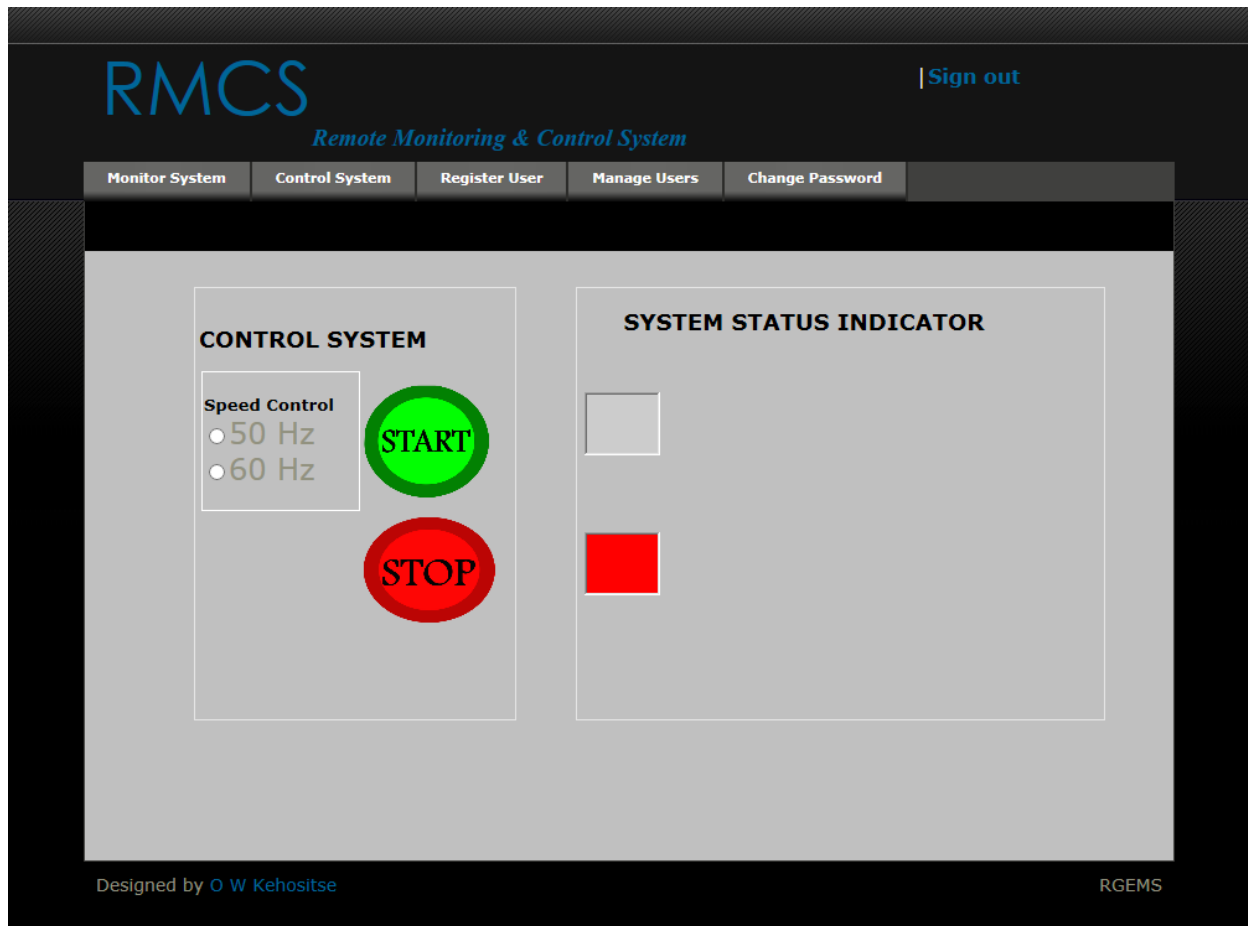


Figure 5.19: RMCS Control indicating that the system has stopped

The system operator has the option of running the motor in two different speeds 50Hz and 60Hz. If the system operator wants the motor to run slower, he/she will select the 50Hz option; if he/she want the motor to run a little faster, he/she will select the 60Hz option.

5.5 System Deployment

In order for the RMCS web application to be accessible through the internet, it must be deployed to the Internet Information Service (IIS) server (i.e. web server) as seen Figure 5.20. The IIS server that our application is deployed to is of version IIS 7. The IIS (web server) allows web applications to be accessed through the internet using a browser. The RMCS is added to the IIS server as web site using the form as seen in Figure 5.21.

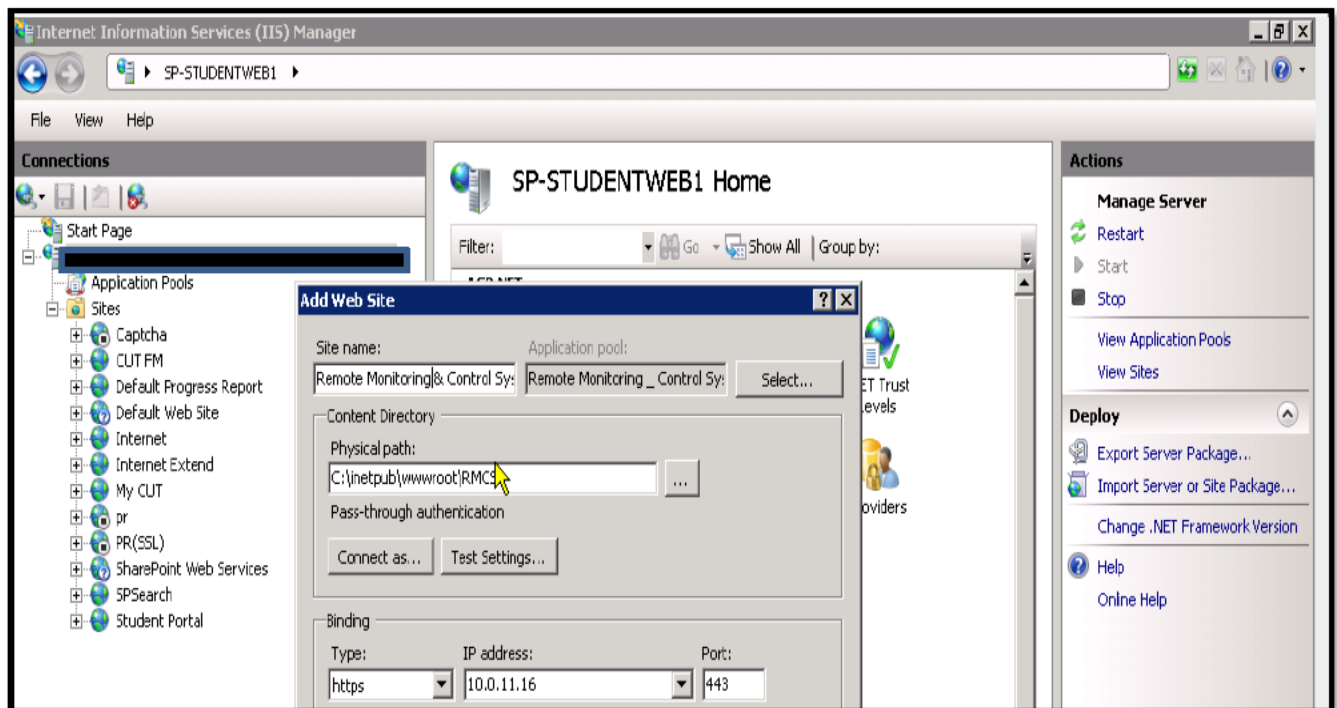


Figure 5.20: IIS server

The binding type is “https” to make the web application more secure. We want the communication between the server and client to be secure, so we used a signed SSL certificate.

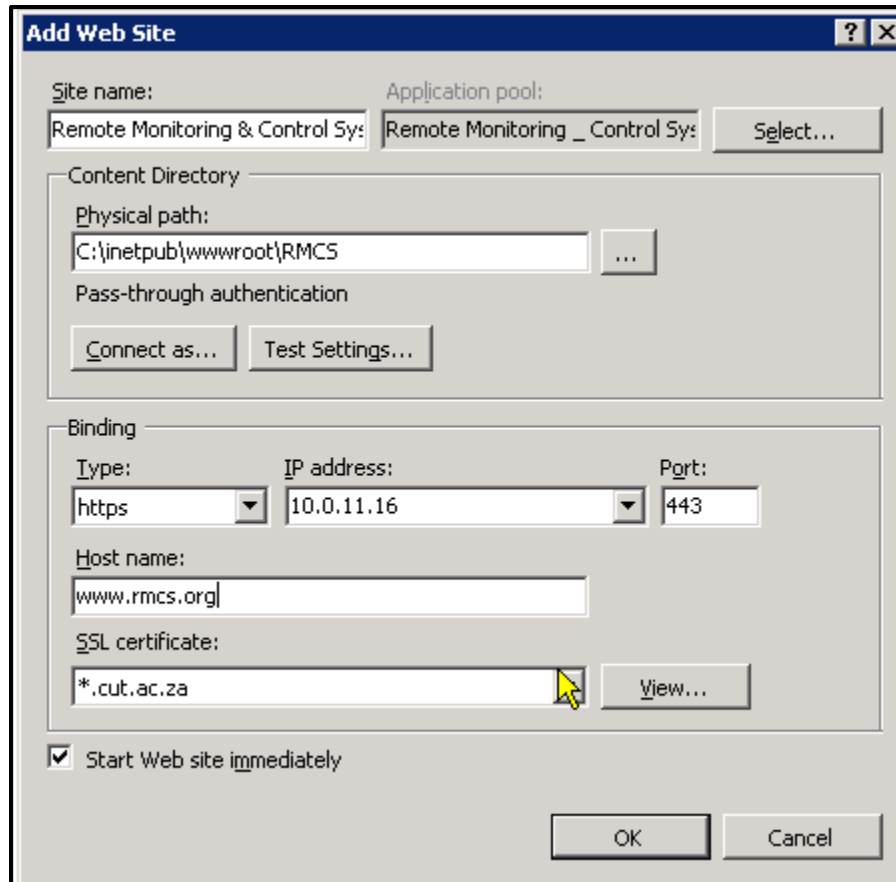


Figure 5.21: Adding a web site to the IIS server

5.6 Satisfying the hypothesis

On completion of the RMCS, all objectives that were set out to be achieved have been achieved.

The system can perform the following:

- Allows real-time remote monitoring of assembly system devices such as motors and sensors while the assembly system is in operation.
- Remotely start and stop the assembly system.
- Activate an alarm whenever there is a fault in the system that needs to be acknowledged.
- Provide historical information about the performance of the system.

- Provide trends of how the process is performing.

6. CHAPTER VI

Conclusion

6.1 Introduction

In this chapter, the significance of the study and its contribution to the manufacturing and mining industry are explained. Suggestions to make the system more advanced are also explained. Conclusions on remote monitoring and control of an automated assembly system using a web interface is discussed.

6.2 Project Summary

This system applies OPC technology into industrial Ethernet in engineering practice; completes the information technology in the production process of an assembly system; and realizes operations for the assembly system, which includes that it would monitor the equipment's (i.e. system devices) running state, and execute online fault diagnosis and the failure prediction function. On completion of the project, all fault information is displayed on the screen, which replaces the HMI display – preventing the operators from missing major fault alarm message prompting. Online operation of the system improves the level of their overall health management and production efficiency. Using the OPC technology as a field control device solves the problem of incompatible hardware devices in industrial control systems, compensates for the lack of fieldbus technology and achieves certain economic benefits. The OPC server technology in the application of this system could provide a reference for other project in engineering practice.

6.2.1 System Functionality

The Remote Monitoring and Control System is capable of performing the following:

- *Data acquisition and display* - The RMCS can acquire data from the OPC server and display it on the web interface. It can also write analogue and digital process data to the OPC server.
- *Alarms and Events* – The RMCS can record important process value changes and system operator’s actions. Whenever there is any process value that goes over its limit, the alarm will be activated.
- *History and Database* - The RMCS keeps all records of the process values.

6.2.2 Remote Monitoring includes:

- *Current State* - Monitor the current state of the process data, both analogue and digital.
- *Alarms and Events* - Record important changes of process value and operator actions.
- *Trends and History* - Keep record of the processed value.

6.2.3 Remote Control includes:

- *Operator control of the process* - The system operator can control the process by means of a web interface; the operator can, for example, pre-set a reference value for the controls or start a motor.

6.2.4 System Operator Functionality

The operations performed through the RMCS by the system operator are as follows:

- Logging on and off the system using passwords and user names.
- Effecting remote control for various equipment such as start and stop.
- Invoking process displays to view the operations throughout the system.
- Changing set point parameters, with appropriate security allowance.
- Viewing historical trend displays and transferring data to other files for exporting.
- Viewing real-time trends displays.
- Viewing the current alarm summary to identify alarm conditions requiring attention.
- Viewing the alarm/event summary to view the chronological series of events.

The system operator can effect control over the motors and other devices, as well as invoke displays which show the current and historical information about any aspect of the assembly system.

6.2.5 RMCS Security

The objective of RMCS is to allow operation by authorized users via an information network while excluding unauthorized operation. The internet-based system requires security measures to improve reliability and system integrity. The security measures that were used are as follows:

- **User Authentication:** This was accomplished by the means of combining of user ID's and encrypted passwords.
- **Access Log:** All events are logged for the purpose of tracking and investigating to determine who performed an operation at a particular time and whether the operation was authorized.

- Instruction restriction control: Establish sufficient access restriction and command filtering to prevent unauthorized or wrong operation.

During the implementation of the RMCS application, a number of conclusions have been reached based on the practical result obtained from the implemented system and the following are the most important ones:

- The implemented system comprises cost-effective solutions as compared with other approaches to build such systems. A basic PC or a smart mobile phone could serve as Human Machine Interface to the system. The use of open source even led to lower cost of the licensing for the development tools and OPC servers.
- Because this system is a browser-based application, there is no need for special drivers to be installed to one's computer or cell phone and it can be accessed from anywhere in the world through the internet.
- Because of the use of standard-based security implementation, the system is very secure. The Security Socket Layer has provided a high level of privacy and data integrity. Moreover, the authentication and authorization of the system are designed to be very strong.

6.3 Contribution of the Study

The findings of this study will contribute greatly to benefit manufacturing, mining industries and other.

I. Manufacturing Industry

Manufacturing in South Africa is dominated by the following industries: Agri-processing, Automotive, Chemicals, ICT and Electronics, Metals, Textiles, clothing and Footwear [40]. Each of these industries has a fair contribution to the South Africa's gross domestic product (GDP). Vendors in process automation and manufacturing face a unique set of challenges when it comes to supporting their systems as every minute of the system downtime can slow down or even halt a plant operation - resulting in some cases in millions of rands lost in revenue. These are the things that the RMCS application developed can do for vendors in process automation and manufacturing:

- Maximise system uptime by proactively preventing problems using its early fault-prediction function.
- Deliver real-time diagnostic data through the web.
- Reduce field service cost by enabling secure remote access.

For example, Coca-Cola Company is known to be the largest manufacturer of soft-drinks in the entire world. Because of the high demand for Coca-Cola soft drinks, the Coca-Cola bottling plant produces 4 000 (330ml) cans per minute [41] to meet this demand, and this is just for a small plant in a town or city. In the case of equipment failure, the bottling plant will go into production halt and Coca-Cola will lose 4 000 (330ml) cans worth of production for every minute the plant is halted.

II. Mining Industry

The traditional way of remote monitoring and control in mining is using PLC's and SCADA systems. These traditional technologies were designed for a world of central command and control, tailored for the production environment not necessarily suited to the new distributed world of business imperatives such as: increased plant availability; reduced environmental impact; increased green credentials; reduced operating costs; and reduced manpower resources.

This brings into focus current initiatives in the mining industry such as environmental compliance, optimised asset management and energy efficiency, all over widening geographic areas. What is required, is the application of a new breed of technology overlaying the traditional production systems that is capable of reaching further to acquire more data and sending it directly to those in the organisation that require it, all in real time. The findings of this study address the technology barriers that the conventional remote monitoring and control in the mining industry face. Because the RMCS application developed is web-based, it provides a convenient way of monitoring and controlling PLC's and other devices from a large geographical area. The RMCS can increase the plant availability by using its early fault-prediction function to give notifications in real-time if there is a fault that is about to happen, and in that way the fault can be dealt with before it happens; by doing that, plant reliability will be enhanced. The RMCS application will not introduce any new operational cost to the current system, because it's web-based, and the internet infrastructure is already in place, so there are no additional components that need to be installed to operate it. In addition, email and SMS alerts can be configured to provide immediate notification if any limits or rules are breached.

6.4 Future Studies

The possibilities of future studies are:

- A Remote Monitoring and Control System (RMCS) that can send alarm notifications to the system operator's cell phone and emails.
- RMCS that can create reports daily and automatically send them through email to appropriate people.

Appendix I: References

- [1] A. Ahad, C. Zaifeng and L. Jay, "Web-Enabled Platform for Distributed and Dynamic Decision Making," *The International Journal of Advanced Manufacturing Technology*, vol. 11, no. 12, pp. 1260-1270, October 2008.
- [2] D. Blanc, "Web-Based Distributed System for TOF Experimental Cooling Plant," 05 March 2001. [Online]. Available: <http://www.dgasser.com>. [Accessed August 2010].
- [3] E. Croke, B. Raleigh and D. Donoboe, "Remote Plant Manufacturing using Wireless Technology," *Irish Engineers Journal*, vol. 57, no. 10, pp. 46-49, December 2003.
- [4] L. Dewey, "Remote Machinery Condition Monitoring using Wireless Technology and Internet," 06 07 2003. [Online]. Available: <http://www.realityweb.com>. [Accessed August 2010].
- [5] A. Y. K. Al-Obaidy, "Design and Implementation of Web Based SCADA system," 2004.
- [6] V. V. Tan, D. S. Yoo and M. J. Yi, "Design and Implementation of Web Service by Using OPC XML-DA and OPC Complex Data for Automation and Control Systems," *Computer Engineering Journal*, 2006.
- [7] Microsoft, "Library," Microsoft, 1 January 1998. [Online]. Available: <http://msdn.microsoft.com/en-us/library/ee658099.aspx>. [Accessed 15 November 2013].

- [8] V. V. Tan, D. S. Yoo and M. J. Yi, "Designing and Developing a Modern Distributed Data System," *Journal of Research and Practice in Information Technology*, pp. 243-261, 2010.
- [9] A. M. M. Hosny A Abbas, "Efficient Web based Monitoring and Control System," *ICAS- The Seventh Internatioal Conference on Automatic and Autonomous system*, pp. 18-23, 2011.
- [10] "Web Based Monitoring and Supervisory control," *Track1:Automatic Control*, pp. 7-13, 24 October 2007.
- [11] H. A. Mohamed and A. M. Abbas, "Review on the Design of Web Based SCADA Systems Based," *International Journal Of Computer Networks (IJCN), Volume (2) : Issue (6)*, pp. 266-277, 2011.
- [12] opcfoundation.org, "About :What is OPC?," OPC foundation, 1 January 1996. [Online]. Available: <https://www.opcfoundation.org>. [Accessed 22 November 2010].
- [13] Globus, "Tutorial : Multiple HTML Chapter1," [Online]. Available: <http://gdp.globus.org/gt4-tutorial/multiplehtml/ch01s02.html>. [Accessed 22 April 2012].
- [14] VKInfotek, "Webservice: Why Create a Web Service?," VKInfotek Inc., 18 February 2012. [Online]. Available: <http://www.vkinfotek.com/webservice/whycreatewebservice.aspx>. [Accessed 20 April 2012].
- [15] J. Chen and Y. Tu, "Design and Implementation of a Web-Based Oil Delivery Remote Monitoring and Controlling System," *IEEE*, pp. 192-197, 2004.
- [16] L. H. W Xuetao, "Design of a Remote Monitoring System Based on OPC Techniques," 2009.

- [17] D. C. Ashmore, *The J2EE Architect's Handbook*, B. McGowran, Ed., DVT Press, 2004.
- [18] I. toolbox, "Wiki: Category Toolbox for IT Groups," IT toolbox, 12 January 1998. [Online]. Available: http://it.toolbox.com/wiki/index.php/CategoryToolbox_for_IT_Groups. [Accessed 12 August 2010].
- [19] F. M. Marques, R. A. Castro, P. E. Miyagi and E. Villani, "Remote Monitoring and Control of Manufacturing System," *Information Technology for Balanced Manufacturing Systems*, vol. 220, no. 1, pp. 369-376, November 2006.
- [20] RedOak, "Red Oak Software: Web Integrator," 5 January 1999. [Online]. Available: http://www.redoaksw.com/products/webclipper/RedOak_WebIntegrationOverview.pdf. [Accessed 22 February 2013].
- [21] Wikipedia, "Wiki: Web Service," Wikipedia, 2 April 2006. [Online]. Available: http://en.wikipedia.org/wiki/Web_service. [Accessed 27 April 2012].
- [22] V. V. Tan, D. S. Yoo and M. J. Yi, "Efficient Web Service Based Data Exchange for Control and Monitoring Systems," *International Journal of Information Technology*, vol. 14, no. 1, pp. 12-28, 2008.
- [23] H. A. Abbas and A. M. Mohamed, "Review on the Design of Web Based SCADA Systems Based on OPC DA Protocol," *International Journal Of Computer Networks*, vol. 2, no. 6.

- [24] R. Li and X. Xiao, "Application Research of Embedded Web," in *Proceedings of the International Symposium on Computer Science and Computational Technology (ISCST 2009)*, Huangshan, 2009.
- [25] Institute-OPC-Training, "opcti," OPCTI, 01 January 2007. [Online]. Available: <http://www.opcti.com>. [Accessed 05 March 2012].
- [26] D. Weppenaar, "Intellegence Maintanance Management in Reconfigurable Manufacturing Enviroment using Multi-Agent Systems," Bloemfontein, 2010.
- [27] S. R. Schach, *Object Oriented and Classical Software Engineering*, New York: McGraw Hill, 2005.
- [28] T. Angerame, "Real Time Online Performance Monitoring of Didtrict Chiller Plants".
- [29] Tutorialspoint, "ULM: ULM Activity Diagram," Tutorialspoint, 22 February 1999. [Online]. Available: http://www.tutorialspoint.com/uml/uml_activity_diagram.htm. [Accessed 14 October 2013].
- [30] RGEMS, "Research Group in Evolvable Manufacturing Systems," RGEMS, [Online]. Available: www.rgems.co.za. [Accessed 5 July 2010].
- [31] S. Lee, J. H. Kim and P. H. Seung, "Conceptual Design of Remote Monitoring and Control System for Nuclear Power Plant," vol. vol 35, no. 3, 2003.

- [32] J. Chunguo, Y. Wang and X. Song, "Development of an OPC Server for Remote Monitoring and Control," in *The Tenth International Conference on Electronic Measurement & Instruments*, Sheyang, 2011.
- [33] OPCSystems, "Open Automation Software: OPC HMI AND SCADA SOFTWARE FOR .NET & WEB APPLICATIONS," Open Automation Software, 22 January 1993. [Online]. Available: <https://www.opcsystems.com/>. [Accessed 26 10 2015].
- [34] info.ssl, "Home: What is SSL?," 2 March 2011. [Online]. Available: <http://info.ssl.com/article.aspx?id=10241>. [Accessed 21 June 2016].
- [35] Search-Security, "Network Security: Secure Shell," 21 January 2000. [Online]. Available: <http://searchsecurity.techtarget.com/definition/Secure-Shell>. [Accessed 21 6 2016].
- [36] Techopedia, "Home: Dictionary: Tags: Networking," 1 June 2005. [Online]. Available: <https://www.techopedia.com/definition/26142/secure-copy>. [Accessed 21 June 2016].
- [37] T. Bishop, "Understanding Motor Temperature Rise Limits," *EASA*, 2003.
- [38] EASA, "The Electro Mechanical Authority," 22 November 2003. [Online]. Available: www.easa.com. [Accessed 23 October 2015].
- [39] Electric-Schneider, "Vijeo Historian Technical Overview: A powerful plant-wide reporting tool that delivers business critical data for real-time decision support," Schneider Electric Industries SAS, France, 2011.

[40] southafrica.info, "Economy Sectors: Manufacturing," South Africa, 1 January 1999. [Online].

Available:

http://www.southafrica.info/business/economy/sectors/manufacturing.htm#.VsMsd_194dU.

[Accessed 16 February 2016].

[41] Coca-Cola, "Coca-Cola Bottling Plant, Wakefield, United Kingdom," Coca Cola, [Online].

Available: <http://www.water-technology.net/projects/cocacolabottling/>. [Accessed 15 February 2016].