



The American University in Cairo

School of Sciences and Engineering

**“USER FRIENDLY PROGRESS REPORTING SYSTEM
FOR CONSTRUCTION PROJECTS”**

A thesis submitted to

The School of Sciences and Engineering,

Construction Engineering Department

In partial fulfillment of the requirements for the degree of

Master of Science in Construction Engineering

By

MOHAMED F.A. ELGINDI

B.Sc. Civil Engineering

Under the Supervision of

DR. SAMER EZELDIN

Professor and Chairman

Construction Engineering

May 2017

To My Mother
Nabila Mohamed Ali ELHEMALY

My Father
Fathi Amin ELGINDI

And

My Brother
Ahmed Fathi ELGINDI

ACKNOWLEDGEMENT

Praise and thanks be to Allah with the blessings of Whom the good deeds are fulfilled.

I wish to express my deepest gratitude and appreciation to my advisor Dr. Samer Ezeldin for his inspiration, encouragement, continuous support, and excellent guidance for these years of my life. Learning and working with him has been one of the most rewarding experiences at The American University in Cairo.

I also would like to extend my sincere thanks and appreciation to my committee members, Dr. Ahmed Waly, Dr. Osama El-Husseiny and Dr. Ezeldin Yazid, who offered guidance and support during the development of this work.

A large thank you is to my Friends and Colleagues at the American University in Cairo, for their help and valuable discussions during my studies.

Osama Mohamed is gratefully acknowledged for his great support towards this study.

Last, but not least, an affectionate thank you to my parents, for their continues love, endless support and prayer throughout my life. Thank you for always believing in me. Many thanks are also to all the members of my family for their encouragement and support.

ABSTRACT

Construction is a dynamic and fragmented industry which has many interacting stakeholders' represented by either individuals or organizations who are involved in a common project for a certain period of time. The simple form for the stakeholders consists of owner, consultant and contractor. As such, construction industry is depending on appointing a consultant, by the owner, to carry out the design of the project and then assigning a contractor, which is responsible to allocate a team to build and give them the sufficient tool to communicate and report. Each party shall formulate its organization chart for such project and assign a project manager on top of hierarchy of this chart. For all involved stakeholders, managing construction projects requires controlling its running schedule and cost comparing to the planned ones. In order to do that, project managers for each stakeholders need periodical reports that represent the status of the project to predict any potential problems and consider taking the necessary measures on the proper time to maintain the project course and budget. Most problems in the construction industry are referred to communication and reporting problems. The links between the different stakeholders have to be built on a robust communication and reporting system that allows to acquire precise, simple and accurate information. Lack in communication and reporting system, shall result in having a misleading information which consequently lead to wrong decisions.

This research tackles the consequences of poor communication and reporting systems and provide some tips to have an effective one. Followed by, describing the current used reporting techniques where their suitability for use is assessed. Furthermore, a user friendly reporting technique is presented where the used data in

developing the technique was acquired from literature study and structured interviews with 29 professional expert in construction industry. The proposed technique aims to integrate the communication process to allow project managers to acquire a simple overview about the projects progress and obstacles. The suggested interface is pointing to achieve the balance between simplicity, accuracy and effectiveness with a reasonable cost. Through simple tables and graphs, it gives a quick overview for the managers to understand the status of their projects for each activity and the assigned resources, evaluate the progress through the key performance indicators, predict the problems at early stage and appraise the project's cash flow to act in a proactive mode not in a reactive one. The proposed interface depends on acquiring on-spot information from each of the project team members about the progress and incurred costs for all project activities through user friendly interface which is designated with its simplicity. Formerly, the acquired information for the progress shall be translated to comprehensive progress report which allows the project manager to understand the status of the project. The offered technique is suitable to be used in any of stakeholders firms as long as it has projects enclose planned activities associated with costs. However, the contractor perspective is considered in this study.

KEYWORDS

Construction communications, information flow, reporting techniques, key performance indicators (KPI's), Earned Value, Actual cost, physical progress

TABLE OF CONTENTS

ACKNOWLEDGEMENT	iii
ABSTRACT	iv
KEYWORDS	v
TABLE OF CONTENTS	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
CHAPTER 1: INTRODUCTION	1
1.1 Background	1
1.2 Statement of the Problem	1
1.3 Objectives.....	2
1.4 Scope of work	3
1.5 Research Methodology	3
1.6 Thesis Organization	5
CHAPTER 2: LITERATURE REVIEW	7
2.1 Introduction	7
2.2 Consequences of poor communication and reporting system.....	9
2.2.1 Case Studies	13
2.2.1.1 Metronet case study (Burr, 2009).....	13
2.2.1.2 A Study for inefficient information flow system in two Chinese's construction sites (Xu and Luo, 2014)	14
2.3 Solutions to Have an Effective Reporting System	16
2.4 The Available Reporting Techniques	21
2.4.1 El-Omari and Moselhi's Model (2010).....	21
2.4.2 Shiau and Wang's Model (2002).....	24
2.4.3 Corewox's Model	25

2.4.4	Bransetter’s Model (2015)	26
2.4.5	Other Commercial Softwares	28
2.5	Gap in Literature and Assessment of the Available Reporting Techniques.....	28
CHAPTER 3: MODEL DEVELOPMENT		30
3.1	Introduction	30
3.2	Background and Methodology	31
3.2.1	Background.....	31
3.2.2	Proposed Reporting System Methodology	31
3.2.3	Model Considerations	32
3.3	Data Collection: Literature Study and Interviews with Professionals.....	33
3.3.1	Literature Study for the Progress Report Contents	33
3.3.2	Interviews with Professionals	34
3.3.2.1	Sampling	34
3.3.2.2	Interview Sessions.....	35
3.3.2.3	Interview Results.....	37
3.3.2.4	Interviews Findings.....	41
3.4	Model Development Criteria	42
3.4.1	INPUT Module	44
3.4.2	ANALYSIS AND OUTPUT Module	49
3.4.2.1	Cover page.....	49
3.4.2.2	Dashboard Outputs.....	51
CHAPTER 4: MODEL VERIFICATION AND VALIDATION		57
4.1	Model Verification.....	57
4.1.1	Model Verification Methodology	57
4.1.2	Model Verification Results	58
4.1.2.1	Number of activities.....	58

4.1.2.2	Number of Projects	58
4.1.2.3	Number of resources	59
4.1.2.4	Allocation of resources on projects	59
4.1.2.5	Projects Duration Period	60
4.1.2.6	SPI and CPI	61
4.1.2.7	Number of invoices per project	61
4.1.2.8	Model Verification Results Discussion	62
4.2	Model Validation	63
4.2.1	Introduction.....	63
4.2.2	Case Study	63
4.2.2.1	Project Information	63
4.2.2.2	As-built Monthly Report Information.....	64
4.2.2.3	User Friendly Reporting Model Application.....	67
4.2.2.4	Results Comparison	70
4.2.2.4.1	Spent time.....	70
4.2.2.4.2	Planned Activities Progress.....	71
4.2.2.4.3	Key Performance Indicators.....	72
4.2.3	Discussion with Professionals	73
4.2.4	Model Validation Results Discussion	76
CHAPTER 5: SUMMARY, CONCLUSIONS AND RECOMMENDATION		
.....	77
5.1	Research Summary	77
5.2	Research Conclusion	79
5.3	Recommendation for Future Research	82
REFERENCES		84
APPENDIX A: INTERVIEW WITH PROFESSIONALS		87
A.1	Interview Survey.....	87

A.2 Interview Survey Results	90
APPENDIX B: KEY PERFORMANCE INDICATORS.....	92
B.1 Abstract.....	92
B.2 Purpose of the Earned value	92
B.3 Definitions and Equations	93
B.4 Earned Value Curves	95
APPENDIX C: PROGRAMMING CODE	96
APPENDIX D: VALIDATION CASE STUDY- GENERATED MONTHLY REPORT	124

LIST OF TABLES

Table 1: A Study for ineffeiceint information flow system in two Chinese’s construction sites (Xu and Luo, 2014)	15
Table 2: List of interviews and interview type	36
Table 3: Interview Results - Progress Report Contents	40
Table 4: Model - Planned VS. Actual Progress Table with the used formulas	51
Table 5: Model - Key Performance Indicators for each activity and the used formulas	54
Table 6: Model - Look ahead Activities and the used formulas.....	56
Table 7: Model - Invoices and cash flow fields and used formulas	56
Table 8: Model Verification - Number of Activities.....	58
Table 9: Model Verification - Number of Projects	59
Table 10: Model Verification - Number of Resources.....	59
Table 11: Model Verification - Allocation of resources on projects.....	60
Table 12: Model Verification – Maximum Projects Duration.....	60
Table 13: Model Verification - Key Performance Indicators	61
Table 14: Model Verification - Number of Invoices.....	62
Table 15: Model Limits.....	62
Table 16: Validation Case Study, Activities within period and KPI’s provided in the as-built monthly report	66
Table 17: Validation Case Study, Invoices Details extracted from the as-built monthly report.....	66
Table 18: Spent Time Comparison.....	70
Table 19: Activities Progress Comparison	71
Table 20: KPI's Comparison	73

LIST OF FIGURES

Figure 1: Highly effective communicators versus minimally effective ones (PMI, 2013)	7
Figure 2: Projects Risks (Monterroso, 2013).....	8
Figure 3: Projects Failure Reasons (Monterroso, 2013)	10
Figure 4: Construction Industry Groups (Furst, 2012).....	13
Figure 5: Wasted time / Overall Recorded Man-hours due to inefficient information flow system (Xu and Luo, 2014).....	15
Figure 6: El-Omari and Moselhi (2010) Proposed System Architecture	22
Figure 7: Activity Form (El-Omari and Moselhi, 2010).....	23
Figure 8: Shiau and Wang (2002) proposed system structure	25
Figure 9: Bransetter's Weekly Reporting Spreadsheet (2015).....	26
Figure 10: Bransetter's Automated Reporting Printing (2015).....	27
Figure 11: Research Methodology Stages	32
Figure 12: Specimen Distribution - Firms Type.....	35
Figure 13: Specimen Distribution - Years of Experience	35
Figure 14: Specimen Distribution - Qualifications	36
Figure 15: Interview Results - Reliable tools to measure the project progress	38
Figure 16: Interview Results – an automated reporting techniques in firms.....	38
Figure 17: Interview Results - Do you think that the automated reporting systems are useful?	39
Figure 18: Interview Results - Progress Report Contents.....	41
Figure 19: Model Flowchart	43
Figure 20: Login Information Window	45
Figure 21: Assigned Activities to Each User	45
Figure 22: Updating Each Activity Status	46
Figure 23: Tasks Search, Export and Print	47
Figure 24: Importing Updated or Revised Programme	48
Figure 25: Data Entry History	48
Figure 26: Progress Report Cover Sheet.....	50
Figure 27: Progress Report in Table Format	52
Figure 28: Progress Report Dashboard	53

Figure 29: Reporting Model Application Steps	67
Figure 30: Model Validation Case Study - Model SPI	68
Figure 31: Model Validation Case Study - SPI vs CPI.....	69
Figure 32: Model Validation Case Study - Cash In	69
Figure 33: Spent Time Comparison	71
Figure 34: Activities Progress Comparison	72
Figure 35: KPI's Comparison.....	73
Figure 36: Earned Value Curve (Czarnigowska, 2008).....	95

CHAPTER 1: INTRODUCTION

1.1 Background

Construction projects necessitate years of planning, design, and construction. Activities pertained to this extent require interacting between different stakeholders which involve a group of experienced experts, as well as a set of management tools and frameworks. Construction Projects have a huge volume of complex information that entails to frequently reporting to project managers to allow them taking the adequate measures to accomplish project's objectives.

1.2 Statement of the Problem

Construction is a fragmented industry that has various stakeholders who shall be interacting through project's construction phases. The most simple form of these stockholders consists of three parties; owner, consultant and contractor represented by project manager for each party on the top of each project organization chart. Construction projects are vulnerable to failure as a result for loosing good communication and reporting systems for the different stakeholders. If any of stakeholder's project managers has limited information about the status of the project, wrong decisions might be taken which endanger the projects and escalate risks. It also could hinder the productivity of works due to an unorganized and huge amount of

information that make it difficult to be followed. Furthermore, the communication between the stakeholders shall be adversely affected as well, resulting in isolating the stakeholder's management than the actual status of the project. The reason behind the limited information for the project manager might be referred to the lack in communication and reporting system which hinder the smooth transfer for information from his subordinates precisely on the right time. As a result, the project manager shall lose the clear vision for his project which is impeding him to take the appropriate measures to maintain project's course and budget. It also obstructing the project manager to communicate the progress to his managers and other stockholders in an effective manner which will eventually endanger the project and increase the probability and severity of the risks.

1.3 Objectives

The main objectives of this research are as follows:

1. To develop a comprehensive, simple and accurate reporting technique that could help project managers to follow the progress of the works, the incurred costs and obstacles that their projects are facing.
2. In order to achieve the objective mentioned hereinabove, other sub-objectives are raised which tackles the following inquiries:
 - a. What are the consequences for inefficient communication and reporting system on construction project?
 - b. May poor communication and reporting lead to failure of project?
 - c. What are the characteristics of a good reporting system?

1.4 Scope of work

The scope of work in this research is to develop a user friendly reporting interface based on the knowledge acquired from literature study and structured interviews with 29 professional expert in construction industry. The developed technique aims at integrating the communication process to allow project managers to acquire simple overview of the project progress and obstacles. The proposed technique aims to integrate the communication process to allow project managers to acquire a simple overview about the projects progress and obstacles

The suggested interface tries to achieve the balance between simplicity, accuracy and effectiveness. Through simple tables and graphs, it gives an overview for the managers to understand the status of their projects for each activity and the assigned resources, evaluates the progress through the key performance indicators, predicts the problems at early stage and appraise the project's cash flow to act in a proactive mode not in a reactive one. The proposed interface can be used in any type of firms (i.e. owners, investors, consulting or contracting firms) as long as it has projects enclosed with activities associated with costs. However, the contractors' perspective is adopted during developing the model.

1.5 Research Methodology

In pursuing the objectives of this research, the work is divided into three phases:

Phase 1: Knowledge Acquisition and Analysis

- a. Study the consequences of poor communication and reporting in construction projects: This study is conducted by performing a literature review, including presenting two case studies to examine the impacts for having communication problems.

- b. Providing recommendations from literature to have an effective reporting system.
- c. Studying the available used reporting systems in the literature and pointing out the gap on each.
- d. Knowledge acquisition through literature study and direct interviews with 29 professionals in the construction industry to collect data about the optimum content for the progress reporting system;

Phase 2: Model Development

Based on the information collected from the previous phase, a user friendly progress reporting model is developed in two stages:

- a. Establishing a database and inputs interface using C# programming language;
- b. Developing a reporting interface using Microsoft Excel software.

Phase 3: Model Verification, Validation and Analysis

- a. Model verification measures were adapted using ad-hoc technique that depends on assuming fictitious cases to examine the model limits and boundaries.
- b. Model validation using an actual case study to demonstrate the model features and to examine the simplicity, accuracy and efficiency of the suggested model. Followed by, a discussion with eight professional experts is held to examine the developed system features and drawbacks.

1.6 Thesis Organization

This dissertation is organized into 5 chapters; this section summarizes the contents of each chapter:

Chapter 1 - Introduction: The statement of the problem of this study is detailed, followed by the research objectives, scope of work and used methodology.

Chapter 2 - Literature Review: the consequences for poor communication and reporting system are tackled through literature study, followed by two case studies which are presented to elaborate the importance of communication and reporting system. Then, some tips to have an effective reporting system are provided. Finally, a literature study for the available reporting techniques to overcome the consequences for poor communication and reporting system is offered. Gaps in literature are also included in this chapter.

Chapter 3 – Model Development: this chapter tackles the research scope and objective through presenting an innovative user friendly technique for reporting system. The developed model is based on collecting data about the optimum progress report contents through literature study and direct interviews with 29 professional expert.

Chapter 4 – Model Verification and Validation: Verifying the developed model to test its limits and validating the outcomes generated from the newly developed interface on an actual case study to examine the effectiveness, accuracy and simplicity of the proposed system. Followed by, a discussion with eight professional experts is arranged to examine the developed system features and drawbacks.

Chapter 5 – Summary, Conclusion and Recommendation: summary and conclusion of this study is presented. Recommendations for future research are highlighted in this chapter.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Construction industry has been emerging since the industrial revolution in the early years of the twentieth century. Projects are becoming larger and more complicated. This shall consequently require a strong communication and reporting system. In fact, many of the problems that occur on construction projects are caused by poor and ineffective communication (TenStep, 2016).

According to PMI's statistics in 2013, 80 percent of projects that meet original objectives in the organizations contain a high performance communication system versus 52 percent in the organizations which have a lower performance communicators. Highly effective communicators are also more likely to deliver projects on time (71 percent versus 37 percent) and within budget (76 percent versus 48 percent) (PMI,2013).

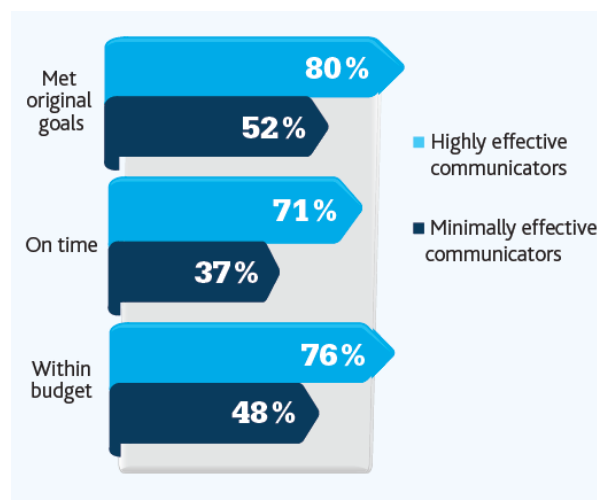


Figure 1: Highly effective communicators versus minimally effective ones (PMI, 2013)

A good communication and reporting system is described as “a rare commodity” according to PMI’s report (2013). Sometimes, project’s team members are unable to transfer a full and precise image for the actual project status to the project stakeholders whether external or internal.

Monterroso (2013) pointed out that “Companies risk \$135 million for every \$1 billion spent on a project and new research indicates that \$75 million of that \$135 million (56 percent) is put at risk by ineffective communications, indicating a critical need for organizations to address communications deficiencies at the enterprise level”.

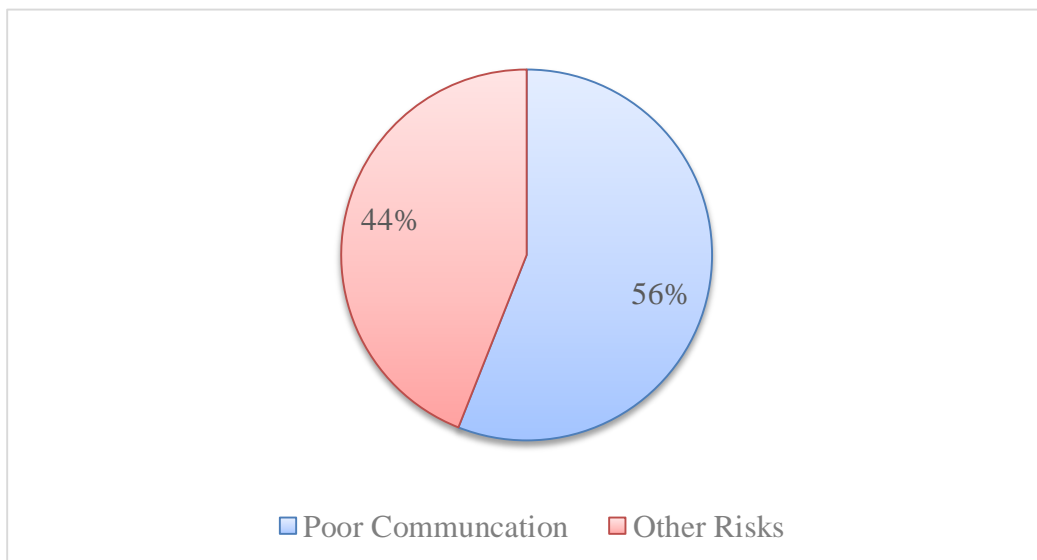


Figure 2: Projects Risks (Monterroso, 2013)

Since the literature review for reporting systems in the construction industry is relatively large, this chapter focuses particularly on four main areas that support the context of this research and can be summarized as follows:

1. Define the consequences resulted by adapting an ineffective communication and reporting system. This shall be illustrating through literature study and two case studies to illustrate the consequences.
2. Provide some solutions to have a good and effective reporting system
3. Literature study for the used reporting techniques to overcome the consequences for poor communication and reporting system.
4. Present gap in literature and assessment for the used reporting techniques.

2.2 Consequences of poor communication and reporting system

Monterroso (2013) refers to a recent study, conducted by the Project Management Institute (PMI), which concludes that ineffective and poor communication flow has a negative impact on construction project. Poor communication and reporting system shall increase risk impacts and probabilities. Using traditional communication tools such as email and spreadsheets without a strong communication strategy designed to store and circulate project information on time, shall result in throwing millions of dollars “out the window”. Monterroso (2013) concludes that one third of the time, the primary contributor to project failure is as a result of poor communications and more than half the time, ineffective communication shall result in terrific negative impact on project success.

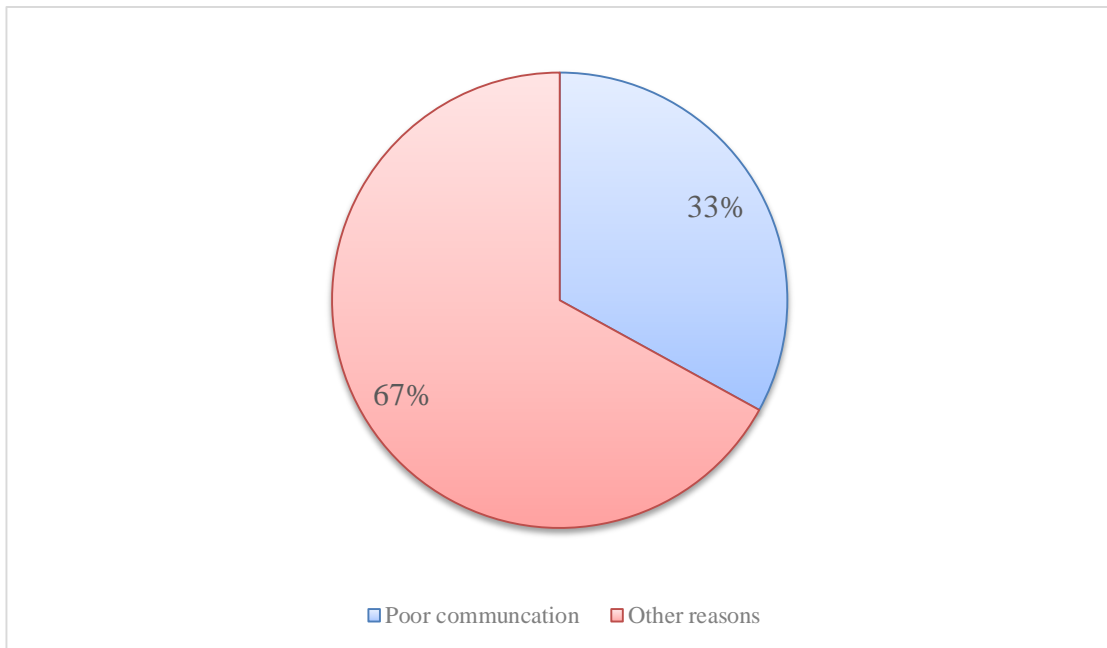


Figure 3: Projects Failure Reasons (Monterroso, 2013)

TenStep, (2016) states that poor communication can lead to the following trouble areas:

1. **Differences in expectations.** Inaccurate reporting system shall make stakeholders to think that the project is going well and will be finished on the planned time and budget. This might confuse them to take right decisions on the right time and make it difficult to achieve their expectation.
2. **People are surprised.** If people are not kept informed as to what is going on, they will be surprised when changes occur.
3. **No one knows what the status of the project is.**
4. **People are impacted by the project at the last-minute.** This will give them no room to take appropriate decisions to retake the project into its

original course. All stakeholders shall realize the status of the project after the fact.

5. Team members don't know what is expected of them.

In United Kingdom, the Office of Government Commerce (OGC) 2005 classify the causes of projects failure as follows (Morgan, 2010):

1. Lack of clear links between the project and the organization's key strategic priorities, lack of clear senior management, lack of engagement with stakeholders
2. Evaluation of proposals driven by initial price rather than long-term value of money
3. Lack of effective project team integration between Owners, the supplier and the supply chain
4. Lack of skills and lack of effective approach to project management and risk management
5. Lack of understandings and contact at senior levels in the organization
6. Too little attention to breaking development and implementation into manageable steps.

It is thought by Morgan that these causes shall result in creating ineffective reporting system that will lead to poor project management eventually (Morgan, 2010).

Garemo et al. (2015) present an explanation for unsuccessful projects. They conclude that one common factor; lack of adequate controls results in a weak risk-analysis and insufficient timely reporting on progress relative to budgets and timelines. The data used to report on project progress are typically outdated and not aligned with

the true progress of the project as they generally rely on payments to contractors rather than on actual work performed.

Abbasi et al. (2014) summarize the main reasons for projects' failure as follows:

1. Lack of senior management
2. Unclear project objectives
3. Unexpected and uncontrolled delays
4. Gaps in communication
5. Lack of visibility of all projects

According to Abbasi et al. (2014), two out of five reasons are directly linked to ineffective communication system. The authors describe the poor in communication as the key common factor for projects' failure as it has direct impact on the other four factors. Most of project's team use email to communicate in their tasks, which is kept, in each individual email box. As such, any new team member joins the project, is not able to get a full history of the project as no centralized data storing or reporting systems are available. The authors also mention that the management teams in projects are mostly having unclear overview on their project due to weak reporting system which affects their decisions to maintain the project on track.

Furst in 2012 classifies the construction industry into three essential groups; the final product which is the project, the project's team and finally the used tools that allow the project's team to manage, control and report their progress. Without integrating the last two groups (i.e. the assigned team and tools), the end product (the first group) might suffer from delays, budget difficulties, quality constrains and finally the possibility of incidents, injuries, or losses. The author concludes that insufficient productivity

measuring system “Metrics” is a common defective practice in construction companies which is usually lead to unintended outcomes. He recommends that the senior management shall make sure that the received progress reports contain the sufficient elements to perceive any potential risks that might drive project’s team to take unnecessary and unevaluated actions which have bad influence on project’s budget, cost or timeline.

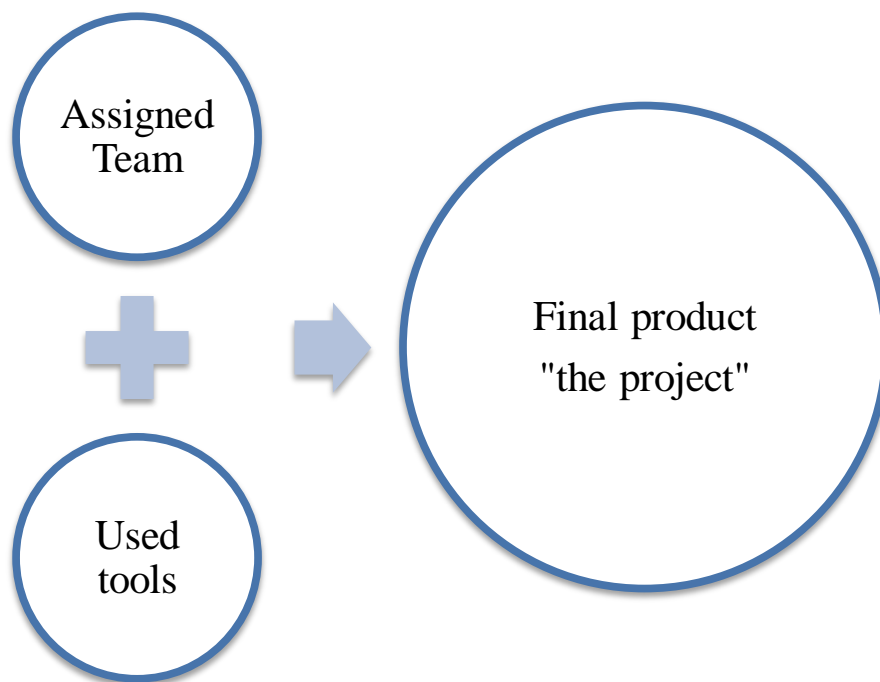


Figure 4: Construction Industry Groups (Furst, 2012)

2.2.1 Case Studies

2.2.1.1 *Metronet case study (Burr, 2009)*

Metronet BCV and Metronet SSL companies were awarded a 30 years Public Private Partnership (PPP) contract by Underground Limited (London Underground) to rehabilitate London Underground’s infrastructure. In July 2007, Metronet BCV and Metronet SSL companies went to administration as a result of their failure to meet their financial obligations. As such, Underground Limited (London Underground) was

forced to buy 95 percent of Metronet's outstanding debt obligations, with total amount of £1.7 billion, in February 2008 (Burr, 2009).

Metronet's failure was attributed to its poor management and lack of communication. The executive management was changed many times with no solid reporting system in place which hindered a smooth handing over process. They had five main shareholders that were acting as Metronet's suppliers with different roles. These suppliers had better access to cost information than the management. With insufficient reporting system, Metronet's was unable to monitor the project's costs which created an immense problem in the cash flow. They could not claim their invoices with evidence of performed works on time due to misleading and inaccurate information they had (Burr, 2009).

2.2.1.2 A Study for inefficient information flow system in two Chinese's construction sites (Xu and Luo, 2014)

Xu and Luo (2014) classify the inefficient communication system and information flow in construction projects as a main reason for decreasing the productivity. Information - which is considered as one of the seven important flows on site according to their classification, the other flows are equipment, manpower, materials, location, external environment and previous works - is vulnerable to be wasted in the absence of an effective information flow system. These wasted information shall negatively affect the productivity and result in wasting time of the project. It also usually a reason for reworks, changes, suspensions, delays for project's activities (Xu and Luo, 2014). The authors conducted a study on two construction sites in China of which they observed that 60% of total recorded man-hours is wasted due reworks, changes, suspensions and delays. As shown in Table 1 and Figure 5, out of

32,022 recorded manhours, 19,072 hours are wasted as a result of ineffeicent information flow system.

Table 1: A Study for ineffeicent information flow system in two Chinese's construction sites (Xu and Luo, 2014)

	Reworks	Changes	Suspensions	Delays	Total
Frequencies	23	2	9	35	
Recorded Man-hours (min)	18,630	2,940	8,094	2,358	32,022
Wasted Time as a result of inefficient information flow system (min)	10,650	2,100	5,094	1,228	19,072
Wasted time / overall Recorded Man-hours	33%	7%	16%	4%	60%

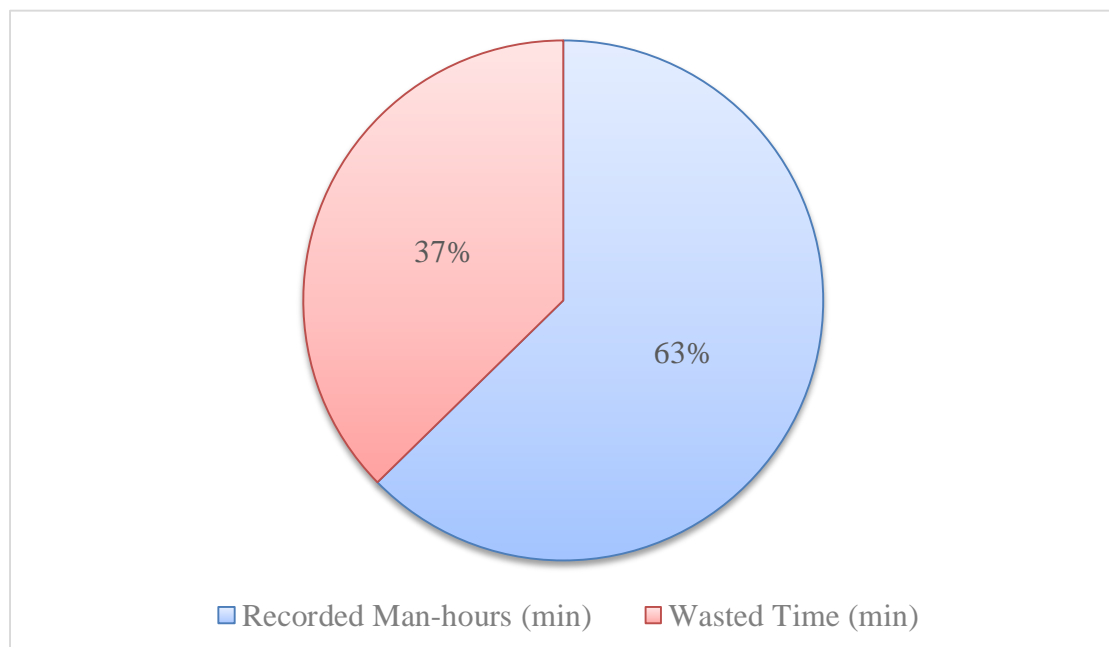


Figure 5: Wasted time / Overall Recorded Man-hours due to inefficient information flow system (Xu and Luo, 2014)

2.3 Solutions to Have an Effective Reporting System

Hoezen, et al. (2006) provide four reasons to answer why construction projects are in need for an effective reporting system as follows:

1. improvement in the communication shall reduce the risk of being failure
2. more open communication at all levels could lead to innovations
3. Communication improvements would positively influence the quality as perceived by all stakeholders involved
4. Leads for better decision making

Monterroso (2013) recommends the following strategies to be adapted by any firm to improve the communication and reporting systems:

1. Close the communications gap around business benefits.
2. Tailor communications to different stakeholder groups.
3. Acknowledge the value of project management, including project management communications.
4. Use standardized project communications practices, and use them effectively.

Any integrated communication and reporting solution shall provide:

1. Reports to identify any deliverables behind schedule
2. Dashboards with real time data such as performance indicators or early identification of potential bottlenecks
3. Official and auditable records of vendor information submitted via secure system transmittals
4. An accurate project history to share best practices and lessons learned for future projects (Monterroso, 2013)

From TenStep (2016) point of view, the key for good communicating is to “keep the receiver as the focal point not the sender”. Try to think about what the receiver needs and the information that will be most helpful to them. If you are creating a status report, put in all the information necessary for the reader to understand the true status of the project, including accomplishments, issues, risks, scope changes, etc. They recommend that the communicator shall focus on:

1. work proactively and not surprise the receiver
2. provide clear and precise information
3. be brief and succinct

Amstrong and Gilge (2010) describe a robust reporting system by having:

1. Transparency and accuracy which shall demonstrate:
 - a. Variances
 - b. Discrepancies
2. Clearly defined objectives and purposes to give the receipts of the report to have
 - a. Take proper decisions
 - b. Ease comprehensive
 - c. Ability to satisfy regulatory requirements

The authors also divide the components of any projects reporting systems as follows:

1. Project dashboard includes the important project key performance indicators
2. Summary management report
3. Summary cost report

4. Risk report
5. Milestones and schedules (Amstrong and Gilge, 2010)

The PMBOK classifies project communications management as one out of ten construction project management knowledge areas that should be well addressed to achieve project's goals. The PMBOK categorizes the performance reporting as one out of four sub-areas which is shown-up during the control process. They define the performance reporting as "collecting and disseminating performance information includes status reporting, progress measurement and forecasting". Any performance report shall compose of the following sections (PMBOK, 2013):

1. Analysis of past performance,
2. Current status of risks and issues,
3. Work completed during the period,
4. Work to be completed next,
5. Summary of changes approved in the period, and
6. Other relevant information which must be reviewed and discussed.

King (2017) concludes that any of the progress reports shall presents the following:

1. problem areas,
2. anticipated delays and their anticipated impacts on the schedule,
3. recommended corrective actions and their intended impacts, and
4. The effects of changes on the schedule of other contractors.

The author also emphasizes that the progress reports shall be audited from time to time to determine if it contains the sufficient type of information, check for the

accuracy of the included data and the schedule of issuance. These audit shall be based on comparing the issued report against the correspondences, meetings minutes, schedule updates, different issued requesters such as requests for information, requests for approval, inspection requests and supervisors different logs. If the audit concludes that the progress reports contents do not match with the other documents, it shows that poor communication management is existing and should be rectified (King, 2017).

Allen and Alexander (2013) have grouped the most important data that shall be well documented in any of construction reports. They classify the project information to achieve three primary goals; monitor project's progress, satisfy contractor requirements and track productivity and profitability. In the authors' opinion, a robust construction report should contains:

1. Schedule updates
2. Missed deadlines and the reason behind it.
3. Processes taking longer than planned
4. Any unusual events
5. Notes explaining hot issues including:
 - a. What happened
 - b. Time and cost impacts (if any)
 - c. Additional scope or staffing
6. Productivity documentation using earned value technique which compare planned versus actual progress and costs.

Redmond (2014), summarizes any strong progress report in three sections:

1. **Overall:** which the reader shall check the project overview health includes:

- a. Project's name.
- b. Project identification number if your company uses a tool to store projects.
- c. The overall project health (red yellow green)
- d. Percentage of The planned / actual completion
- e. The number of days behind or ahead from the schedule
- f. The number of blocking issues

2. **Milestones:** to check the schedule which shall be defined by:

- a. The milestone name
- b. The percent complete of the milestone
- c. The planned start
- d. The planned finish
- e. The actual start
- f. The actual finish

3. **Issues:** To see brief details about each issue so to take the right decision containing:

- a. Ticket number (if any)
- b. Issue name: this should be brief
- c. Date and time reported
- d. Priority or severity of the issue

- e. Owner of the issue
- f. Actions to resolve it
- g. Expected time for resolution (Redmond, 2014).

2.4 The Available Reporting Techniques

2.4.1 El-Omari and Moselhi's Model (2010)

In 2010, El-Omari and Moselhi propose a model for controlling construction projects through an IT platform which use automated data acquisition from construction sites using some technologies such as bar coding, Radio Frequency Identification (RFID) 3D laser scanning, photogrammetry, multimedia, and pen-based computers. A relation between planning and scheduling software system and the proposed model is presented to measure the progress.

El-Omari and Moselhi's model is based on dividing the data acquisition into two groups; group (1) for input data which are needed to produce deliverables such as labor man-hours, equipment usage and material consumption, and group (2) for output data which describe the quantities for completed works. The authors develop a Microsoft Access database as shown in Figure 6 where the acquired data from site is stored in centralized database of which interaction with scheduling software system is established. Then the user, using user interface, responds to some queries as shown in Figure 7. Then the outputs such as reports, digital photo images, videotapes, and voice comments are generated automatically (El-Omari and Moselhi, 2010).

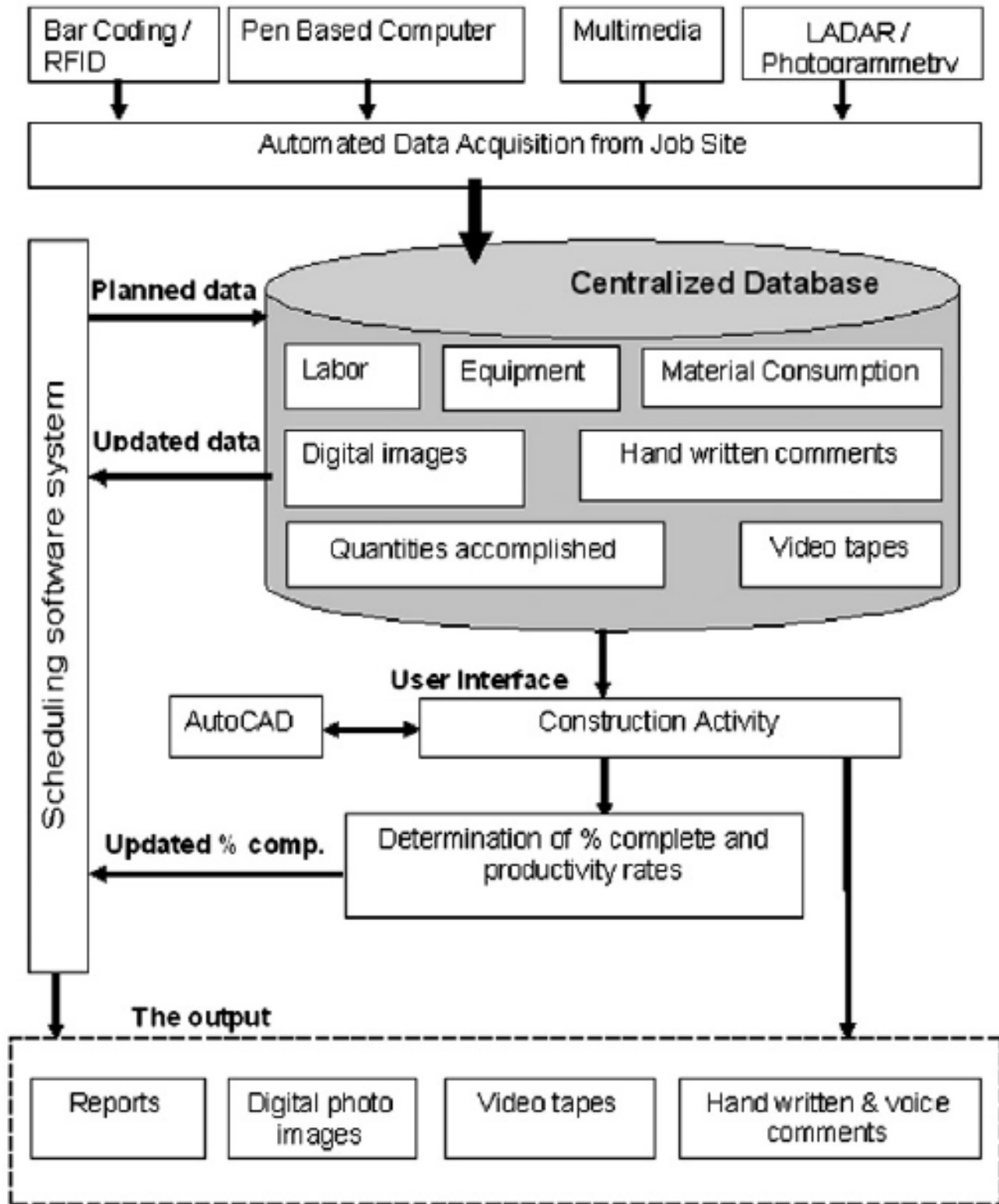


Figure 6: El-Omari and Moselhi (2010) Proposed System Architecture

The screenshot displays a software window titled "Activity" with a light beige background and a blue border. The window contains several data entry fields and buttons. At the top, there are three main fields: "Project ID" (value: 1), "Activity ID" (value: 65), and "Description" (value: Excavation & shoring). Below these are fields for "Activity Duration" (70), "Projec Last Update" (empty), "Sub-Contractor" (button), "Planned Start Date" (12/13/2006), "Planned Finish Date" (4/3/2007), "Resource Time Sheet" (button), "Actual Start Date" (12/23/2006), "Actual Finish Date" (empty), "Material" (button), "TotalQuantity" (52900), "QuantityToDate" (850), "ACWP" (\$16,247.50), "Percent Complete" (2.00%), "Percent Complete Template" (empty), "EV" (\$0.00), "BCWS" (empty), "BAC" (\$371,000.00), "EAC" (\$0.00), "Photos" (button), "Videos" (button), "Notes" (button), and "Drawings" (button). At the bottom left, there is a "Record:" label with navigation icons and the text "64 of 150".

Figure 7: Activity Form (El-Omari and Moselhi, 2010)

The proposed model by El-Omari and Moselhi (2010) shows a user friendly interface which is easy to be used. However, it is mainly based on quantities to calculate the progress. However, some activities cannot be measured by the quantities such as the engineering activities and preparation works for construction activities. Additionally, the proposed system is time consuming as it requires from the user to update around 12 fields. Finally, the system accuracy is doubtful, as some of the updating fields are related to the earned value measurements which could be easily modified to serve the user's favor.

2.4.2 Shiao and Wang's Model (2002)

Shiao and Wang (2002) proposed another model for daily report. Their suggested tool is aiming to calculate the cash flow effectively by transfer construction data to headquarter through world-wide-web site. As shown in Figure 8, the model is divided into 8 modules as follows:

1. Project's basic data for building includes all items, which are categorized into three main groups; large, medium and small items.
2. A budget module which is responsible for generate project budget
3. Purchasing and contracting module to monitor the subcontractors and executing parties
4. Fund auditing module that is responsible to alert if any of project items exceed the budget.
5. Daily report module of which site records such as engineering information, quantities, manpower, used equipment and material are updated on daily basis.
6. Estimation and pricing module that develop the accomplished quantities prices.
7. Design modification control module that records items and quantifies of every design change.
8. Project accounting module to control the finished items costs, income, and expenditure.

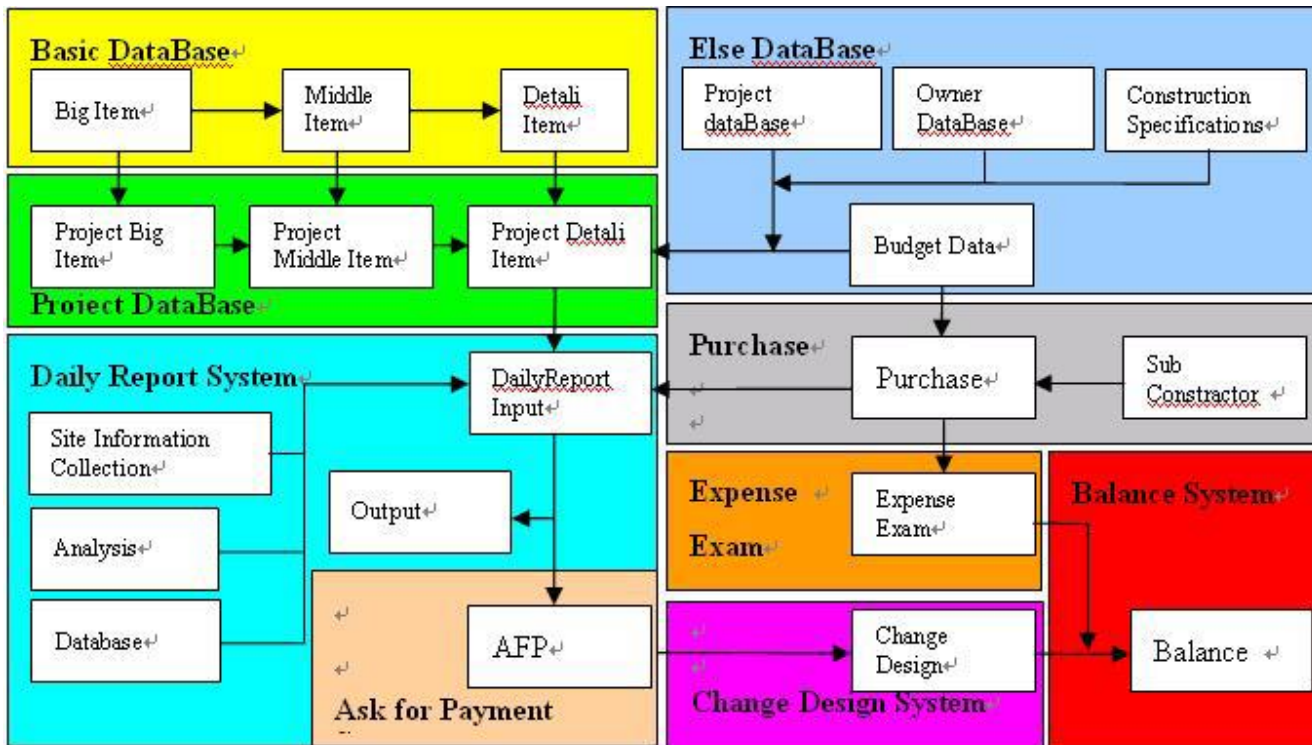


Figure 8: Shiao and Wang (2002) proposed system structure

The proposed model by Shiao and Wang in 2002 is designated by its simplicity. However, it is obvious that the provided system is looking for the project from financial perspective only neglecting the schedule progress.

2.4.3 Corewox's Model

A software's development company called Corewox provides a project information management solution entitled "Corewox Project Information Control software" which automates project management processes and regulate projects' information and processes, Corewox system captures all data related to individual communication activities and provides the information via reports and dashboards appropriately for different stakeholders over the entire project lifecycle. In the developers of Corewox system's opinion, the communicated information to the correct

people throughout the project means more than sending an email to a project team member with a recent update (Monterroso, 2013).

The Coreworx Project Information Control software shall give his advantage through concentrating on the following:

1. Document control
2. Design Review and Approval
3. Management Reporting and Dashboards
4. Transmittals and Vendor Portal
5. Document Viewer and Markup
6. Integration

On contraire, the provided technique tackles the reporting issue from documents control perspective mainly.

2.4.4 Bransetter's Model (2015)

In 2015, Bransetter recommends to have a spread sheet as shown in Figure 9 to automate the weekly reporting process. His solution is based on dividing the report into two groups; projects' goals and key initiatives.

Status	Goal/Initiative	Metric/Owner	Progress	Comments / Notes
	Goals	Goal Metric	Actual	
●	Maintain an NPS score of over 50%	50%	63%	Continue to be ahead of plan. Next survey goes out in two weeks.
●	Provide Training to 100,000 users	100000	40000	Behind vs. plan for the year but launching 3 initiatives this month that should help us catch up to plan over the next 3 months.
●	Reduce response time by 30%	30%	43%	Response times continue to improve primarily do to increased team size, specialization, team training initiatives, and improved email routing system.
	Key Initiatives	Owner		
●	Implement new phone and routing system	Jon Johanson	●	We've selected the new system and are working with a vendor on creating a project plan in Smartsheet to schedule the implementation.
●	Hire 8 additional representatives	Sarah Smithson	●	6 new folks added to the team and onboarded. 2 additional offers pending that we expect to be accepted.
●	Launch Webinar program: 1 per week	Bill Bower	●	Launched program and provided 1 webinar per week. In order to get to 1 per week we are solidifying a list of guest speakers and topics with Marketing.
●	Implement staff training program	Kathy Knightley	●	Created new 6 of 12 planned modules and built them into our LMS system. Soft launched the 6 existing modules on new hires. Feedback positive but lots of good suggested edits that we are implementing.

Figure 9: Bransetter's Weekly Reporting Spreadsheet (2015)

In goals section, fields such as the title, weights, actual completion percentages and remarks of goals shall be filled.

While in the key initiatives section, the individual tasks to achieve the goals are presenting, owner of each initiative, actual progress through using symbols, a summary for the achieved progress.

This spreadsheet shall be installed at each of projects' individuals which is linked together on a one server. The administrator of the spreadsheet shall automatically collect the information from each individual and send the report through an email automatically after selecting the required period and recurrence as shown in Figure 10.

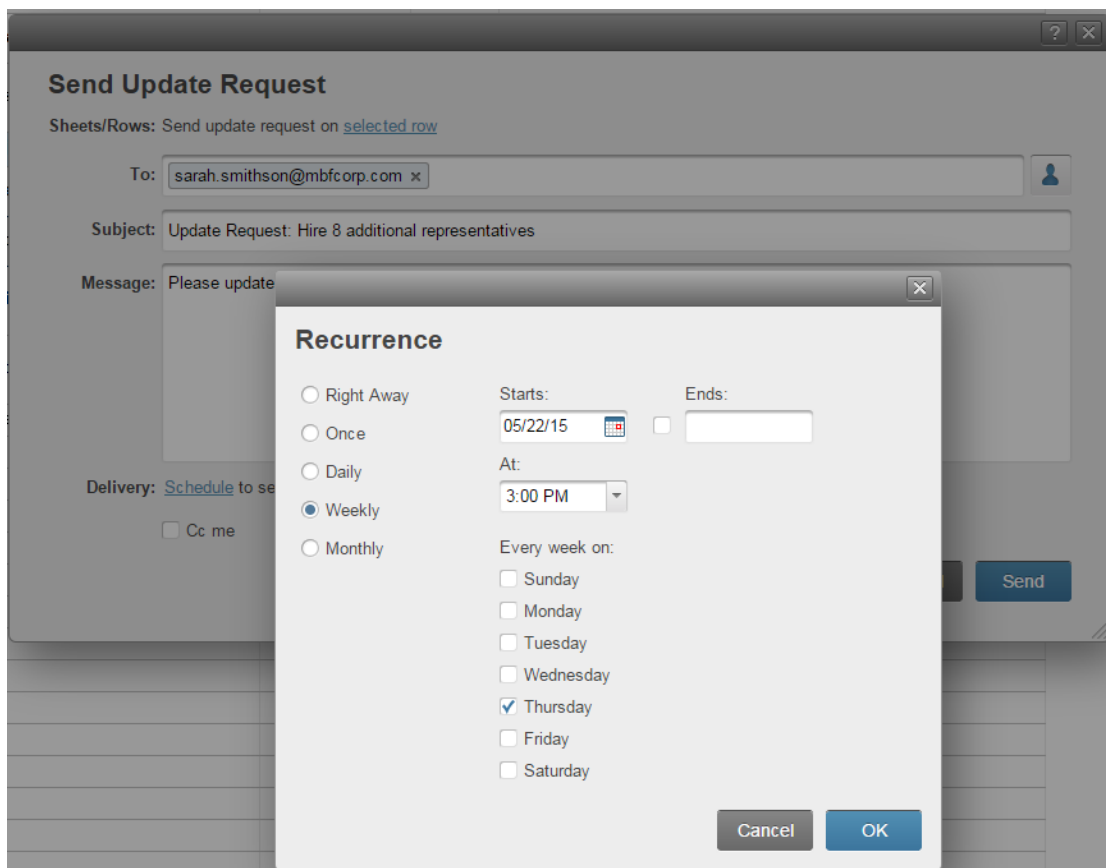


Figure 10: Bransetter's Automated Reporting Printing (2015)

The proposed system by Bransetter (2015) is labeled with its ease to use and its cost is reasonable comparing to other commercial softwares mentioned hereinafter. However, it mainly focuses on the physical progress and neglects any information related to the incurred costs or key performance indicators.

2.4.5 Other Commercial Softwares

Various commercial softwares are developed to tackle the progress reporting issue such as Primavera Contract Management (previously known by Primavera Expedition) and SNC-Lavalin PM+. According to the interviewed construction industry professionals (refer to the Chapter 3), most of commercial softwares are proven to be effective and accurate while its high cost value is a noteworthy drawback for such softwares. Nevertheless, most of it is also lacking simplicity and requires an intensive training courses to be acquainted with its interface.

2.5 Gap in Literature and Assessment of the Available Reporting Techniques

The construction industry is challenged to deliver projects within the planned budget on the baseline time schedules and with the required quality. After reviewing the literature for the commonly used reporting systems, there is an evident gap in the effectiveness, accuracy, simplicity and cost for the available techniques. As such, reporting model is required to address the following aspects which are mostly lacking in the literature:

1. Simplicity: most of the presented interfaces are lacking simplicity. It depends on one person only (the one who is responsible for generation the reports) who requires long time to acquire the needed information

from each team member to produce a progress report. In addition, all other commercial softwares require an intensive training courses to be acquainted with its interface.

2. Accuracy: since the periods between collection of data is depending on reports frequency, the collected data might be vulnerable to lack of accuracy as a result of forgetting some data. Depending on one member to collect the date from all team members which are usually do not have much time to provide this member with accurate information shall result in producing inaccurate report.
3. Effectiveness: most of the presented models are lacking producing key performance indicators which are effective technique to evaluate the actual status of the project. In contrast, other interfaces that produce KPI's allow the users to update it manually which might be updated according to users favor and not presenting the real indicators.
4. Cost: the price for the most of commercially used softwares is relatively expensive.

CHAPTER 3: MODEL DEVELOPMENT

3.1 Introduction

Successful firms that normally complete projects on time, on budget and within goals have a budget to develop a strong communications and reporting strategy which is nearly twice as their lower competitors which usually have major problems in delivering their projects on time, on budget, and within goals. (Monterroso, 2013).

Using basic tools such as spreadsheets to track project information, especially in case of large projects, is subject to human error, difficult to keep updated with large document volumes and nearly impossible to track and audit for updates. This statement is referred to the deficiency of the spreadsheets to be consistent and integrated. The manual nature of spreadsheets lacks the controls that can be provided with automated tools that ensure all information is captured, reviewed, and approved as mandated by project management (Monterroso, 2013).

Schroeppel in PMI report (2013) concludes that “If you want to achieve anything, you need to reach out to people, tell them what you’re doing, listen to what they need and adjust your goals as necessary”.

This chapter proposes a user friendly reporting technique which is aiming to integrate the communication process and allow project managers to have a simple overview about the projects progress and obstacles.

3.2 Background and Methodology

3.2.1 Background

The focus of this study is to develop a comprehensive, simple and accurate reporting technique that could help project managers to follow and control the progress of the works, the incurred costs and obstacles that their projects are facing. In order to do so, data collection from each of project's team is necessary to track and control projects' activities. As shown in the literature review, a considerable studies have been carried out to automate data collection and generate automated reporting system. The proposed reporting technique shall adapt the same concept to collect the data from each individual in a simple and time efficient routine to generate automated progress reports.

3.2.2 Proposed Reporting System Methodology

The work on this research is divided into three stages as shown in Figure 11. First, the data collection stage where the optimum progress report contents were figured out from the literature. Followed by direct interviews with 29 of industry professionals are conducted to rank these contents and examine the most sufficient ones to be presented in the progress report. Second, based on the information collected from the previous stage, user interface reporting model is developed depending on two modules: INPUT Module (Database and Interface phase); ANALYSIS AND OUTPUT module (Reporting output phase). Third, verification for the developed model using fictitious

and random cases to examine the model limits and boundaries is conducted. Then, a validation process took place for the proposed reporting model using an actual case study to demonstrate the model capabilities, effectiveness and limitations.

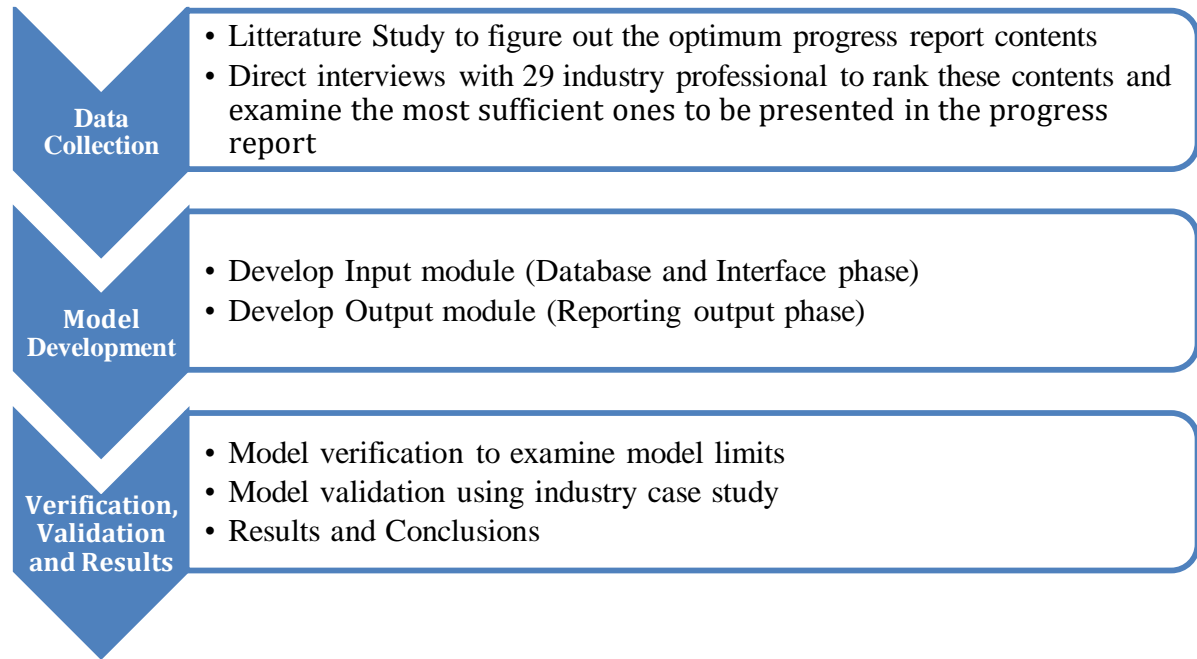


Figure 11: Research Methodology Stages

3.2.3 Model Considerations

The considerations of the suggested model are as follows:

1. Each activity in the project should be assigned to a single person called “task owner”. For example, if we have a project that consists of 600 activities and the project team contains 10 members. As such, all 600 activities should be assigned to the 10 members (i.e each member will be responsible for more than one activity). However, no single activity should be assigned to two or more members; each activity should be assigned to one out of the 10 team members. Each task owner shall be granted the privilege to update the status of his activities.

2. The physical progress and incurred cost for each activity shall be subject to the judgment of its assigned resource.
3. Project invoices should be treated as activities similar to the construction activities and it should be assigned to a “task owner” as well.

More limitations related to the model analysis were generated after verifying the developed model as mentioned in section 4.1. (Model Verification).

3.3 Data Collection: Literature Study and Interviews with Professionals

3.3.1 Literature Study for the Progress Report Contents

As described in the aforementioned chapter, any robust progress report shall tackle the following sections:

1. Report information
2. Project's Information
3. Executive Summary
4. Progress Pictures
5. S curve and cash flow analysis
6. Variation Orders Summary
7. Claims
8. Risks and Area of concerns
9. Key performance indicators
10. Planned activities in the coming periods
11. HSE Records
12. Activities percentage of completion
13. QA / QC Records

14. Logs (transmittals, Request for Approvals (RFA's), Request for information (RFI))

These contents are ranked in the following section using direct interviews with professionals to figure out the sufficient contents for any progress report.

3.3.2 Interviews with Professionals

3.3.2.1 Sampling

After collecting various contents for the progress reports from literature, this stage aims to rank these contents and identify the sufficient contents in any progress report. Face-to-face and telephone structured interviews were conducted with a number of 29 interviewees from the construction industry working in 11 reputable and multinational firms in Egypt. The specimen of interviewees included professionals from three perspectives; owners, consultants and contractors with percentages of 34.5%, 34.5%, 31% respectively as shown in Figure 12. The majority of the specimen (64%) has an average years of experiences vary from 10 to 20 years as mentioned in Figure 13. Figure 14 shows that 78% of the interviewees have a bachelor degree in engineering while the Ph.D and Master of Science holders are represented by 7% and 15% respectively. The used population size is determined using the same percentage between number of tier 1 contractors in Egypt (5) to the total number of the contractors registered in the Egyptian Engineering Syndicate (36,000) and the percentage of number of population (29) to the total registered engineers in the Egyptian Engineering Syndicate (530,000) (Egyptian Engineering Syndicate, 2016). The interviews took place in January 2017. At the beginning of the interview, the interviewees were reassured of the confidentiality of their identities and their organizations. After the

consent of the interviewees, a transcript of their responses was written down. For this research, all interviewees have direct work experience and exposure in the phases of engineering, tendering and construction.

3.3.2.2 Interview Sessions

The interview sessions were about 15 minutes with each interviewee and were conducted on different dates, depending on the available timeslot from each interviewee. The interview type was structured with predetermined list of questions and limited set of responses. At the beginning of the interview, the research topic and its significance were presented to each interviewee. Each interviewee was provided with survey questions. The questions were intended to discuss one area of the topic; however, spaces for informal discussion were given at the end of the interview in order to obtain more information. A list of the used questions in the interviews is documented in **Appendix A**.

Table 2 presents the list of interviewees, their positions, qualifications, years of experience, the type of firm they work for and interview type.

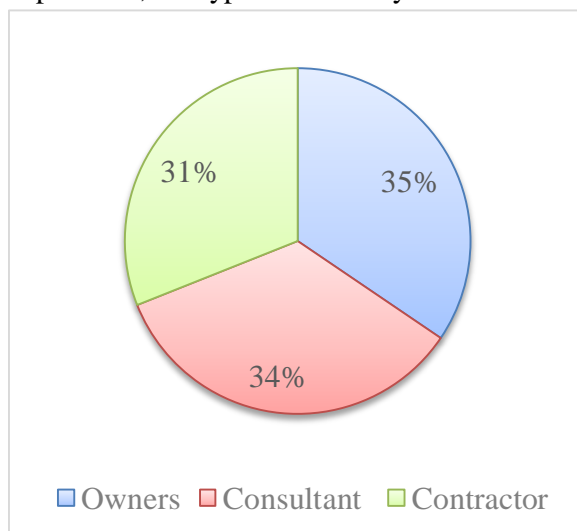


Figure 12: Specimen Distribution - Firms Type

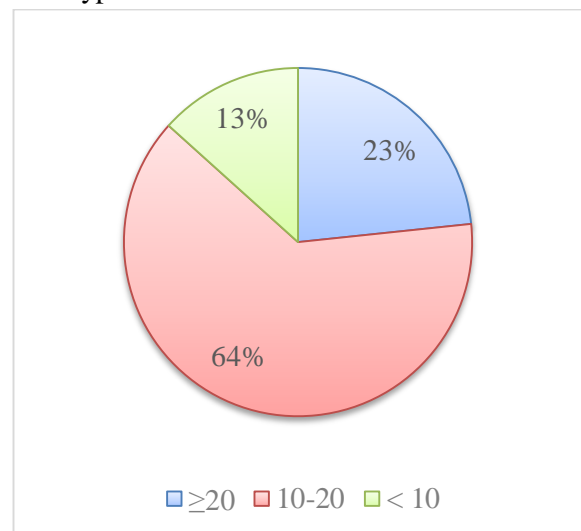


Figure 13: Specimen Distribution - Years of Experience

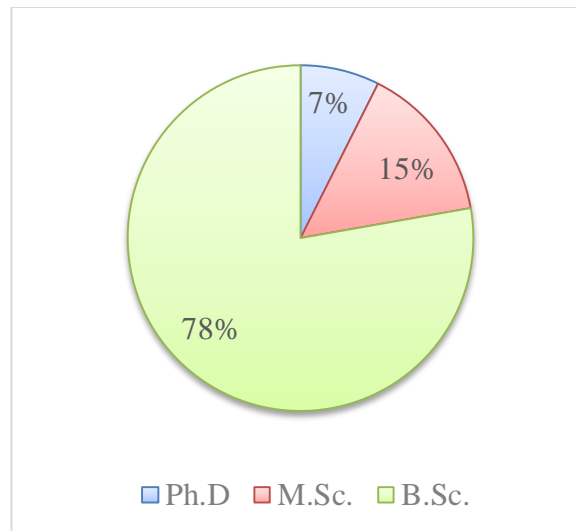


Figure 14: Specimen Distribution - Qualifications

Table 2: List of interviews and interview type

Interviewee No.	Current Position	Current Firm Type	Qualifications	Years of Experience				Interview / Questionnaire Type
				Owner	Consultant	Contractor	Total	
01	Construction Manager	Owner	B.Sc.	7	0	6	13	Face-to-Face
02	Project Manager	Owner	B.Sc.	10	0	0	10	Face-to-Face
03	Sr VP. / Operation Manager	Owner	B.Sc.	16	0	0	16	Telephone
04	Engineering Manager	Owner	B.Sc.	15	0	2	17	Telephone
05	Engineering Manager	Owner	B.Sc.	12	0	1	13	Face-to-Face
06	Project Manager	Contractor	B.Sc.	0	0	13	13	Face-to-Face
07	Project Manager	Owner	B.Sc.	13	0	2	15	Telephone
08	Project Engineer	Owner	B.Sc.	9	0	0	9	Telephone
09	Project Manager	Owner	B.Sc.	15	1	0	16	Face-to-Face
10	General Manager	Consultant	B.Sc.	0	13	10	23	Face-to-Face
11	Projects Manager	Consultant	B.Sc.	0	12	8	20	Telephone
12	Engineering Manager	Consultant	Ph.D	0	15	0	15	Face-to-Face
13	Project Manager	Consultant	M.Sc.	0	11	9	20	Face-to-Face
14	Project Manager	Consultant	B.Sc.	0	8	3	11	Telephone
15	Project Manager	Consultant	M.Sc.	0	11	0	11	Face-to-Face
16	Engineering Manager	Consultant	Ph.D	0	14	9	23	Telephone
17	Procurement Manager	Owner	B.Sc.	15	0	0	15	Telephone

Interviewee No.	Current Position	Current Firm Type	Qualifications	Years of Experience				Interview / Questionnaire Type
				Owner	Consultant	Contractor	Total	
18	CEO	Contractor	B.Sc.	0	0	25	25	Telephone
19	Projects Manager	Contractor	B.Sc.	0	1	20	21	Face-to-Face
20	Project Manager	Contractor	B.Sc.	0	0	15	15	Telephone
21	Project Manager	Contractor	B.Sc.	0	0	14	14	Telephone
22	CEO	Contractor	B.Sc.	0	0	19	19	Face-to-Face
23	Project Manager	Contractor	B.Sc.	1	0	14	15	Face-to-Face
24	Projects Engineer	Consultant	B.Sc.	0	4	0	4	Telephone
25	Projects Control Engineer	Consultant	B.Sc.	0	6	0	6	Telephone
26	Projects Control Manager	Consultant	M.Sc.	0	13	5	18	Face-to-Face
27	Engineering Manager	Owner	B.Sc.	18	0	2	20	Telephone
28	Technical Office Manager	Contractor	B.Sc.	0	0	17	17	Face-to-Face
29	Projects Control Manager	Contractor	M.Sc.	0	4	12	16	Telephone

3.3.2.3 Interview Results

The responses were documented by hand in a transcript during the interview.

Refer to **Appendix A** for the interview results. The following is a summary of the concluded results from the interviews.

Progress Reporting Effect on the Construction

1. 48% of the interviews agree that frequent progress reports is a reliable tool to examine the project progress. While other tools such as direct meetings, conferences, phone calls and site tours are coming in the following orders to the progress reports with percentages of 31%, 13% and 8% respectively as shown in Figure 15.

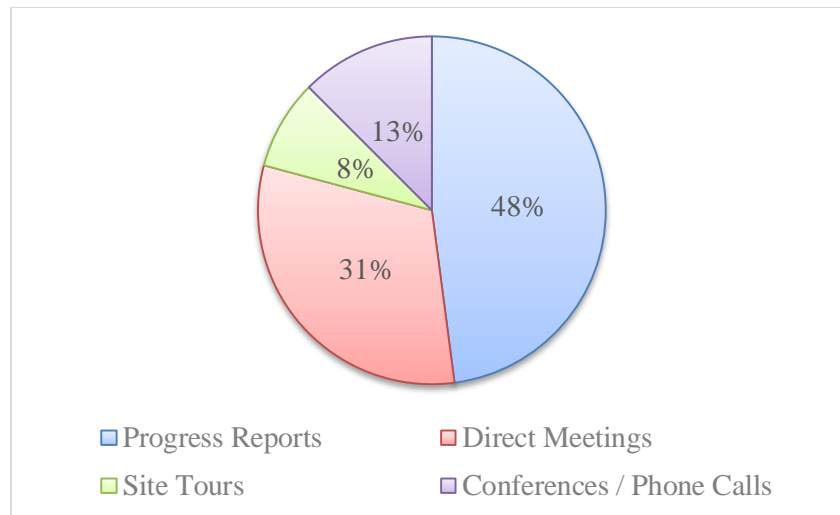


Figure 15: Interview Results - Reliable tools to measure the project progress

- It is worth to mention that 38% of the interviews worked before in firms that has an automated reporting system while 62% of the responders did not (refer to Figure 16).

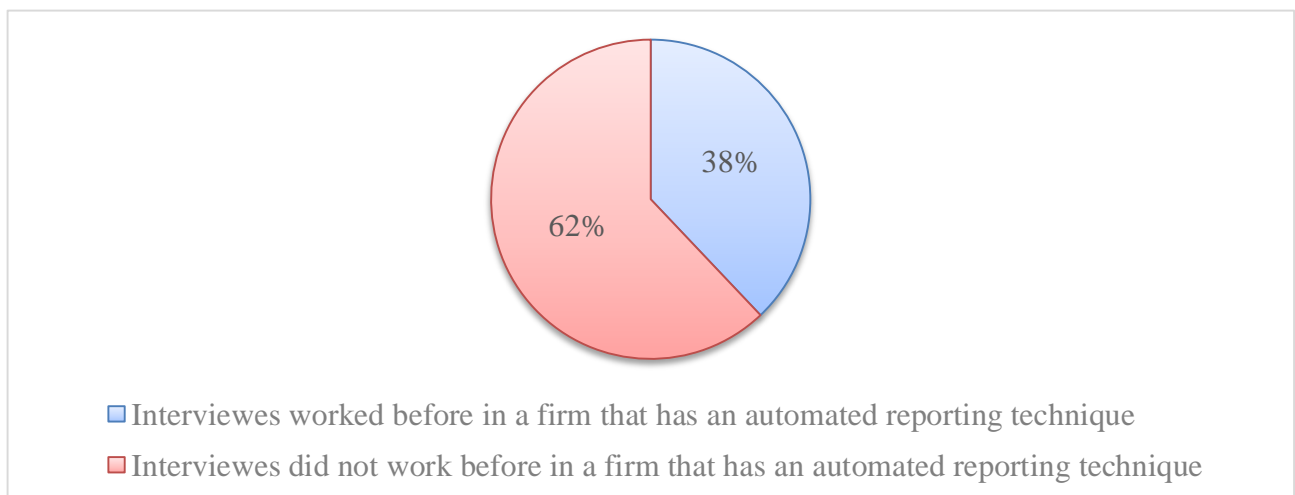


Figure 16: Interview Results – an automated reporting techniques in firms

- As shown in Figure 17, 48% of the responders agreed that having an automated reporting technique shall definitely assist in tracing the progress report while 45% are not confident of this argument claiming that it depends on the characteristics of the used technique itself. In other words,

if the used technique is complicated in the inputs and has difficulties to comprehend the outputs, then the used system is useless. The reason behind their answer is because they had worked in firms that use reporting commercial softwares which are complicated to be used and needs long time of training. Thus, they ignore using it because of its complexity.

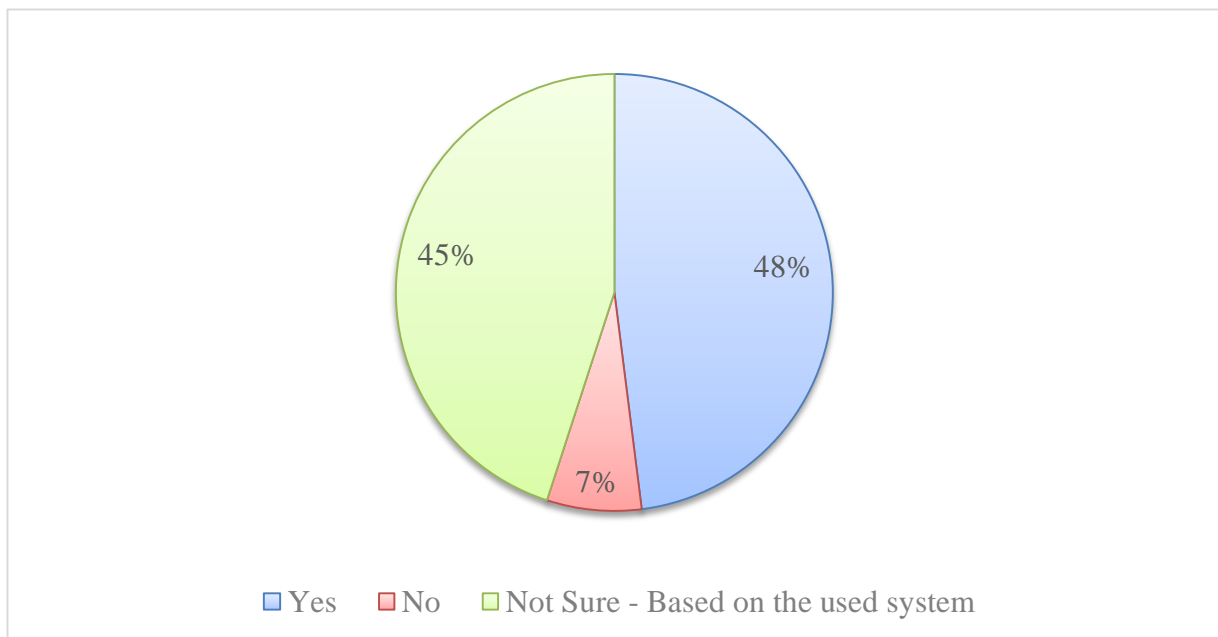


Figure 17: Interview Results - Do you think that the automated reporting systems are useful?

Progress Reports Content

Table 3 and Figure 18 show the results for the progress report contents. Out of 29 responder's answers, the most contents that any of progress reports should have are as follows: Report information, Progress Pictures, S curve and cash flow analysis, Project's Information, Variation Orders Summary, Claims, Risks and Area of concerns, Executive Summary, Key performance indicators using Earned Value Measurement

Technique, Planned activities in the coming periods, HSE Records and Planned activities in the coming periods.

Table 3: Interview Results - Progress Report Contents

Progress report content	Answers out of 29 responders
Report information	29
Progress Pictures	29
S curve and cash flow analysis	28
Project's Information	27
Variation Orders Summary	27
Claims	27
Risks and Area of concerns	27
Executive Summary	26
Key performance indicators using Earned Value Measurement Technique	25
Planned activities in the coming periods	18
HSE Records	16
Planned Activities percentage of completion	15
QA / QC Records	14
Logs such as transmittals, Requests For Approval (RFA's), Requests for Information (RFI's)	12
Others	3

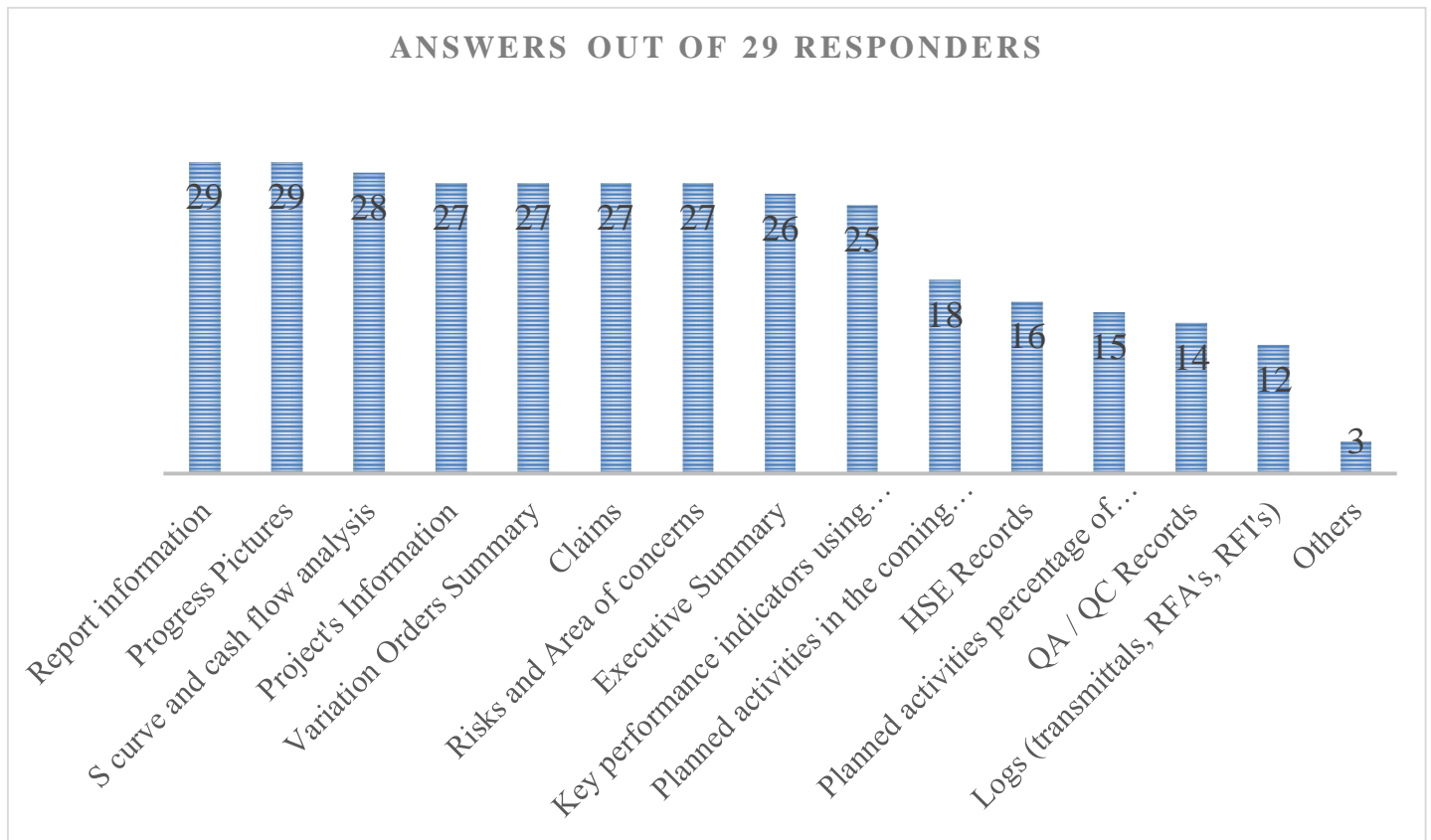


Figure 18: Interview Results - Progress Report Contents

3.3.2.4 Interviews Findings

The interview results provided a global image on the used reporting practices in construction. Accordingly, the significance of this research was formulated based on the interview results and findings. Further to the results, the highest ranked contents which depend on quantitative data, as shown in Table 3 and Figure 18, are considered in developing the reporting model. However, other contents are excluded since it considered as qualitative data that require more manual contribution than automated inputs such as the variation orders, claims, HSE and QA/QC records and logs. Accordingly, the following contents are considered in building the reporting model:

1. Project information and report data
2. Planned activities percentage of completion
3. Key performance indicators using Earned Value Measurement Technique
4. Planned activities in the coming periods
5. Project Cash flow analysis
6. Progress Pictures

3.4 Model Development Criteria

Based on the literature review and the conducted interviews; the researcher proceeds with the model development. The proposed model adopts the flowchart mentioned in Figure 19 which is divided into two module; **INPUT module** (Database and Interface phase) and **ANALYSIS AND OUTPUT module** (Reporting output phase). The developed model is suitable to be run on Microsoft Windows platform. In addition, Microsoft Excel and a setup for an “EXE” file which was built using C# programming language is required to launch the model.

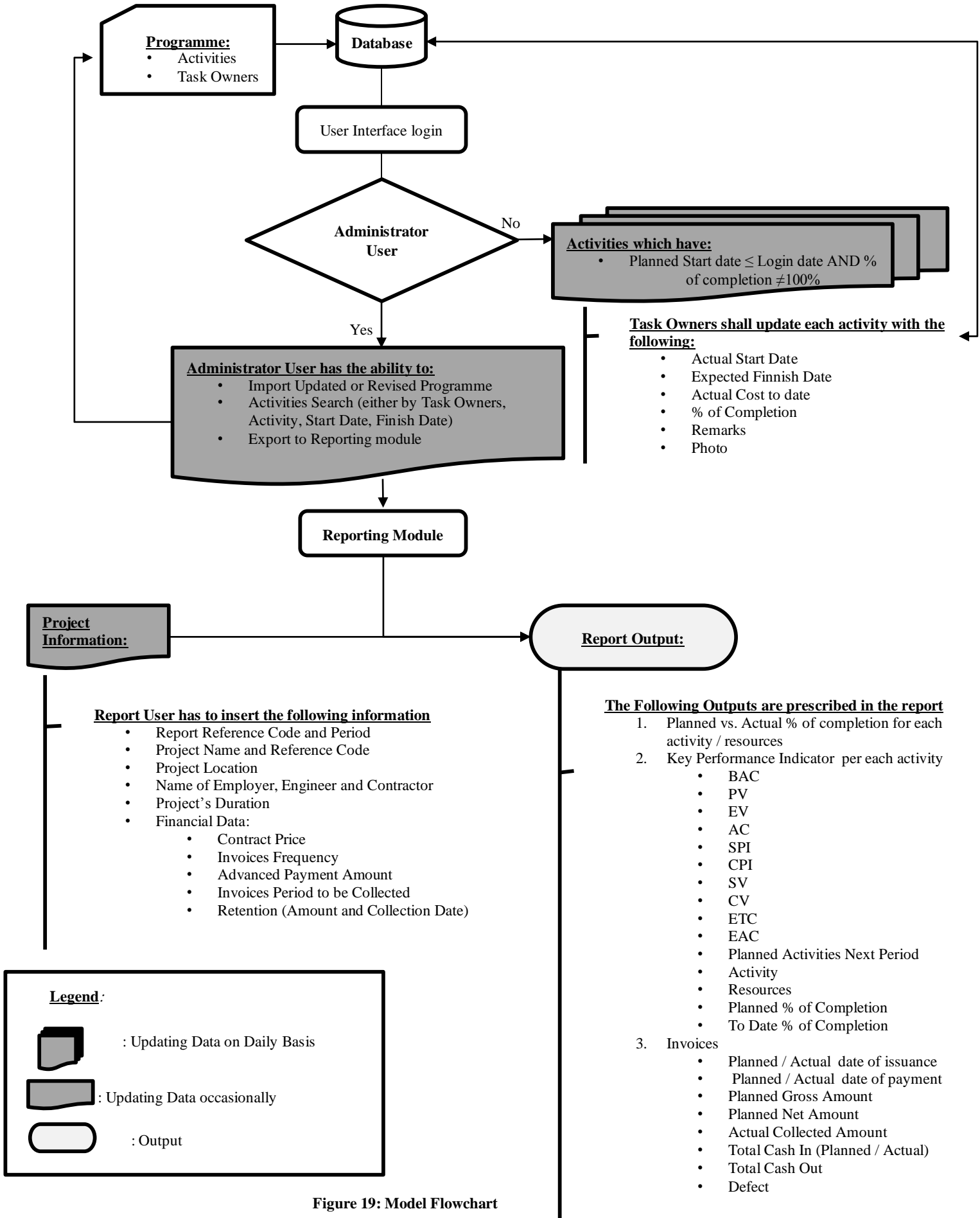


Figure 19: Model Flowchart

3.4.1 INPUT Module

In the INPUT module, an interface software was developed using C# programming language which shall be installed on each of project individuals' computers and is linked together to one server. Refer to **Appendix C** for the programming code. The following steps were considered in developing this model:

1. Prepare a database including all project activities on condition that each activity shall be assigned to a member in the project's team, called "task owner". For example, if we have a project that consists of 600 activities and the project team contains 10 members. As such, all 600 activities should be assigned to the 10 members (i.e each member will be responsible for more than one activity). However, no single activity should be assigned to two or more members; each activity should be assigned to one out of the ten team members. The initial database could be imported from an individual planning software's such as Primavera P6 or Microsoft project.
2. Interface users are divided into two groups based on their access privileges as follows:
 - a. Administrator user who is defined as a user that has an access to surf all activities, can search, filter and export activities details to either the database or to the analysis and output module.
 - b. User: Project's team member (i.e. Task Owner) who has an access to write an update for the activities that were assigned to him.

- c. Each user shall have a username and password to login to predetermined projects included in the interface as shown in Figure 20.

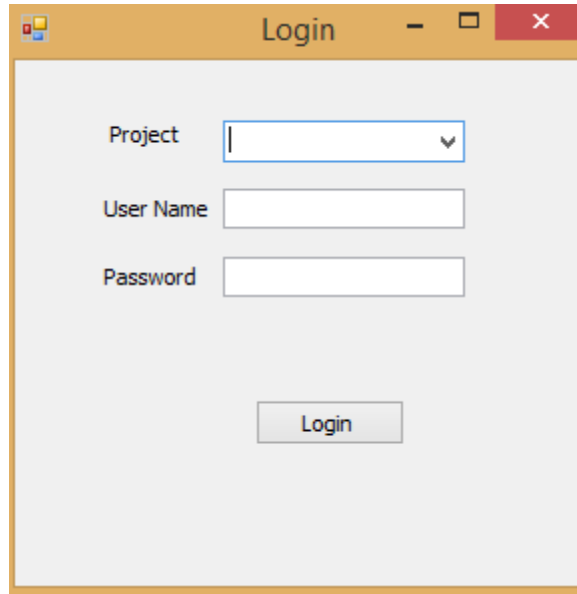
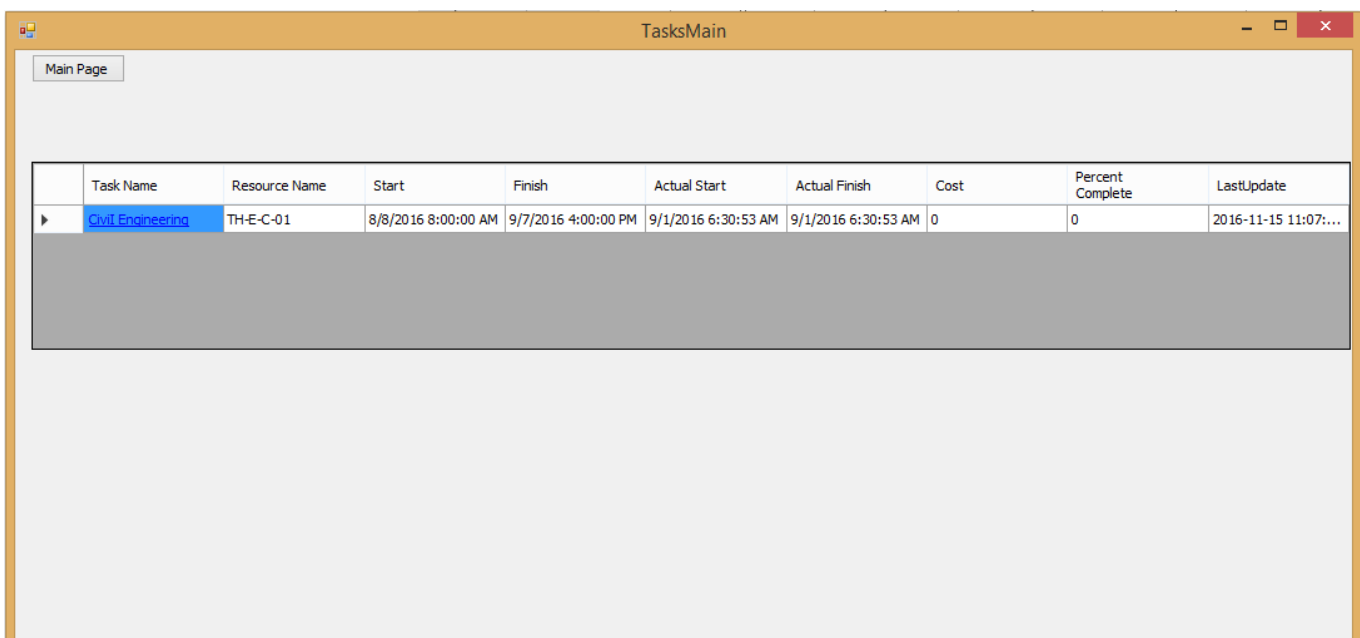


Figure 20: Login Information Window

3. On daily basis, each user shall access the interface. The related activities that were assigned to him which have a start date less than or equal to the login date and percentage of completion is not equal to 100% shall be shown up (refer to Figure 21).



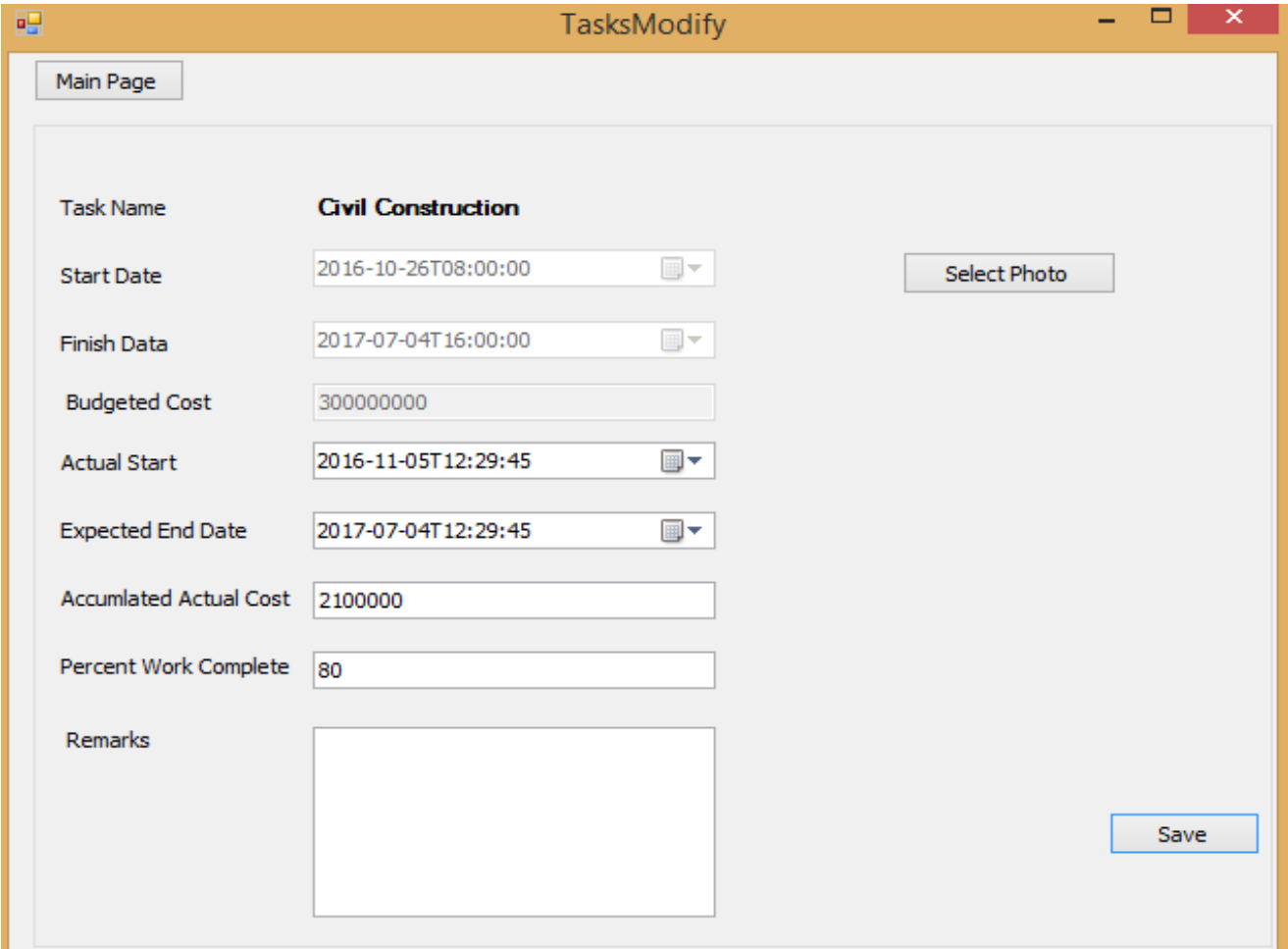
	Task Name	Resource Name	Start	Finish	Actual Start	Actual Finish	Cost	Percent Complete	LastUpdate
▶	Civil Engineering	TH-E-C-01	8/8/2016 8:00:00 AM	9/7/2016 4:00:00 PM	9/1/2016 6:30:53 AM	9/1/2016 6:30:53 AM	0	0	2016-11-15 11:07:...

Figure 21: Assigned Activities to Each User

4. As shown in Figure 22, the user shall write an update each activity as follows:

- a. Actual Start Date
- b. Expected Finnish Date
- c. Accumulated actual cost to date
- d. Percentage of completion
- e. Remarks
- f. Photo

This step shall be updated easily on frequently basis as it will not consume long time for each user. The inputs shall update the activity information on the database.



The screenshot shows a web application window titled "TasksModify". At the top left, there is a "Main Page" button. The main content area contains a form with the following fields and values:

Task Name	Civil Construction	
Start Date	2016-10-26T08:00:00	<input type="button" value="Select Photo"/>
Finish Data	2017-07-04T16:00:00	
Budgeted Cost	300000000	
Actual Start	2016-11-05T12:29:45	
Expected End Date	2017-07-04T12:29:45	
Accumlated Actual Cost	2100000	
Percent Work Complete	80	
Remarks	<input type="text"/>	<input type="button" value="Save"/>

Figure 22: Updating Each Activity Status

It is worth to mention that Project invoices should be treated as activities similar to the construction activities and it should be assigned to a user who will be responsible to update its progress as well according to the following procedure:

- a. Actual start date shall be the date of issuance of the invoice,
 - b. Accumulated Actual Cost is the actual invoice amount.
 - c. Percent of Completion is the collected amounts.
5. The Administrator user has the ability to do the following tasks in the interface (refer to Figure 23 and Figure 24):
- a. Import Updated, Revised or new programme to the database.
This privilege shall be used occasionally.
 - b. Activities Search (either by Task Owners, Activity, Start Date, Finish Date). He can also check each activity history of data entry by the task owner as mentioned in Figure 25.
 - c. Export to Analysis and Output module

Total Cost: 88

Task Name	Resource Name	Start	Finish	Actual Start	Actual Finish	Cost	Percent Complete
Civil Engineering	TH-E-C-01	8/8/2016 8:00:00 AM	9/7/2016 4:00:00 PM	9/1/2016 6:30:53 AM	9/1/2016 6:30:53 AM	0	0
Arch Engineering	TH-E-A-01	8/8/2016 8:00:00 AM	9/6/2016 4:00:00 PM			3	0
Mechanical Engineering	TH-E-M-01	8/8/2016 8:00:00 AM	9/7/2016 4:00:00 PM			3	0
Electrical Engineering	TH-E-E-01	8/8/2016 8:00:00 AM	9/7/2016 4:00:00 PM			3	0
Procurement						3	
Arch Procurement		9/7/2016 8:00:00 AM	10/18/2016 4:00:00 PM			3	0

Figure 23: Tasks Search, Export and Print

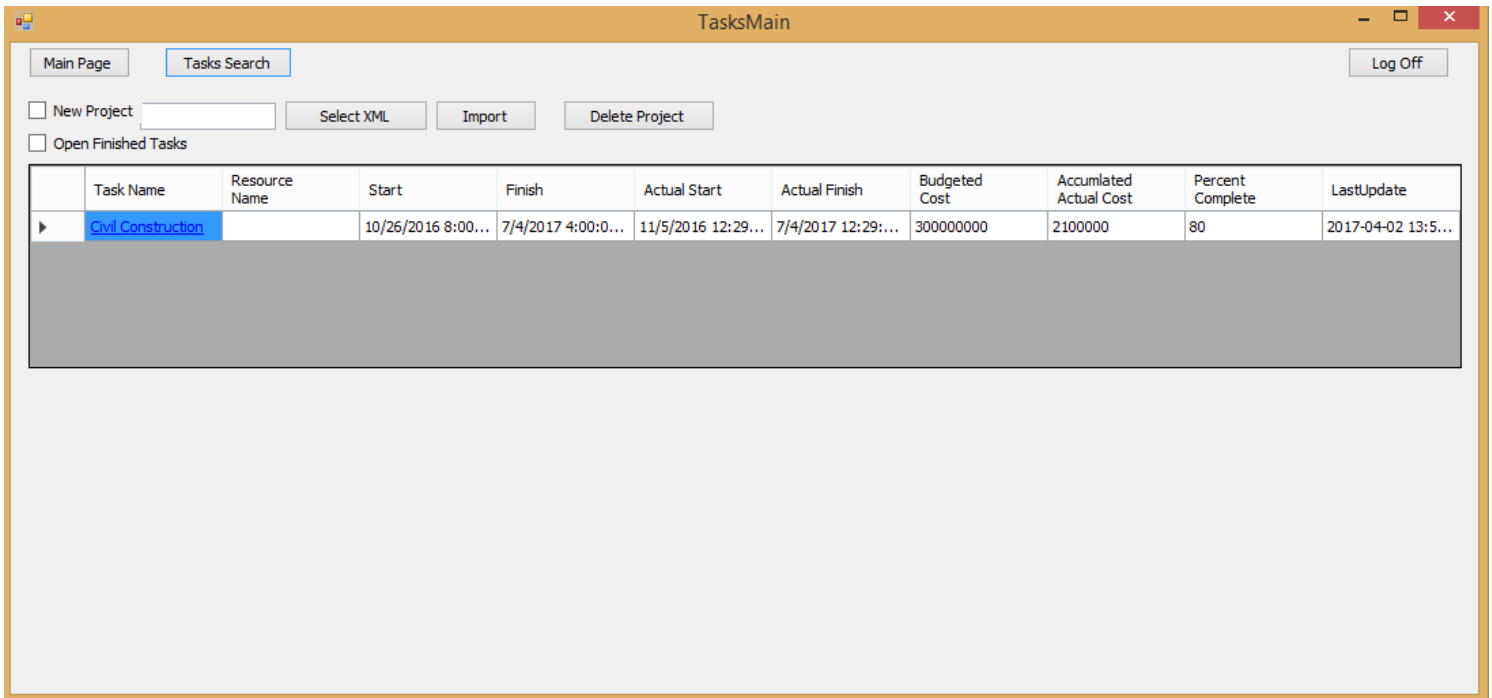


Figure 24: Importing Updated or Revised Programme

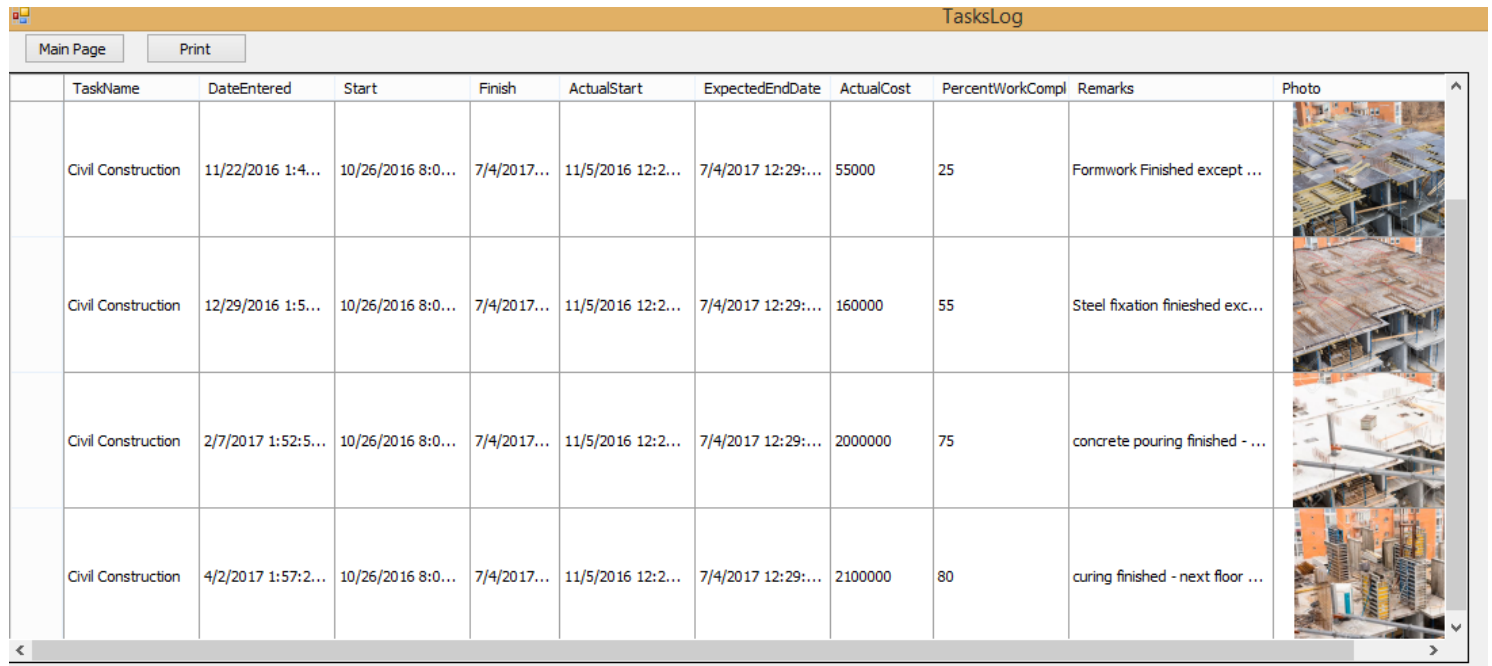


Figure 25: Data Entry History

3.4.2 ANALYSIS AND OUTPUT Module

In the ANALYSIS AND OUTPUT module, a report shall be generated with the following contents:

1. Cover Page
2. Dashboard contains brief tables and graphs about the following:
 - a. Planned versus Actual % of completion for each activity / resources
 - b. Key Performance Indicator for each activity
 - c. Planned Activities Next Period
 - d. Invoices and cash flow

3.4.2.1 Cover page

As shown in Figure 26, the report cover page contains information about the project and the subject report such as:

1. Report Reference Code and Period
2. Project Name and Reference Code
3. Project Location
4. Name of Employer, Engineer and Contractor
5. Project's description and Duration
6. Financial Data:
 - a. Contract Price
 - b. Invoices Frequency
 - c. Advanced Payment Amount
 - d. Invoices Period to be Collected
 - e. Retention (Amount and Collection Date)

Most of these data shall be updated occasionally.



American University in Cairo
 School of Science and Engineering
 www.aucegypt.edu

Tel: +20 (2) 2615.1000
 Fax: +20 (2) 2795 7562

Progress Report

Reference

From *October 15, 2016*

To *November 15, 2016*

Project Code

Project Name

Employer

Contractor

Commencement Date

Project Duration

Contract Price \$ *100,000,000.00*

Invoices Frequency *Monthly*

Retention *10%*

Project Location

Project Name

Engineer

Contractor

Expected Completion Date

Advance Payment *20%*

Invoices period to be Collected *1 month*

Retention Collect *1 year*

Project Description

Figure 26: Progress Report Cover Sheet

3.4.2.2 Dashboard Outputs

Using the exported data from the INPUT Module, executive outputs dashboard using tables and graphs shall be generated including the following sections (refer to Figure 27 and Figure 28):

1. Planned vs. Actual percent of completion for each activity / resources

In this section, comparison between the planned activities within the period of the progress report versus the actual progress. To ensure showing only the activities within the period of the report, the following if statement was adapted:

$$\text{if } \left\{ \begin{array}{l} \text{report start date} \geq \text{activity start date and actual percent of completion} \neq 100\% \\ \text{or} \\ \text{report start date} \leq \text{activity start date} \leq \text{report finish date} \end{array} \right.$$

\therefore activity within period

Table 4 shows the generated table with the used formulas.

Table 4: Model - Planned VS. Actual Progress Table with the used formulas

Task Title	Assigned Resources	Planned % of completion	Actual % of completion	Status	Remarks
		$\begin{array}{l} \text{if report finish date} \\ \geq \text{activity finish date} \\ \rightarrow 100\%; \\ \\ \text{otherwise} \\ = \frac{\text{report finish date} - \text{activity start date}}{\text{activity finish date} - \text{activity start date}} \end{array}$	Input from the assigned resource	$\begin{array}{l} \text{Actual \% of completion} \\ - \text{Planned \% of Completion} \end{array}$	

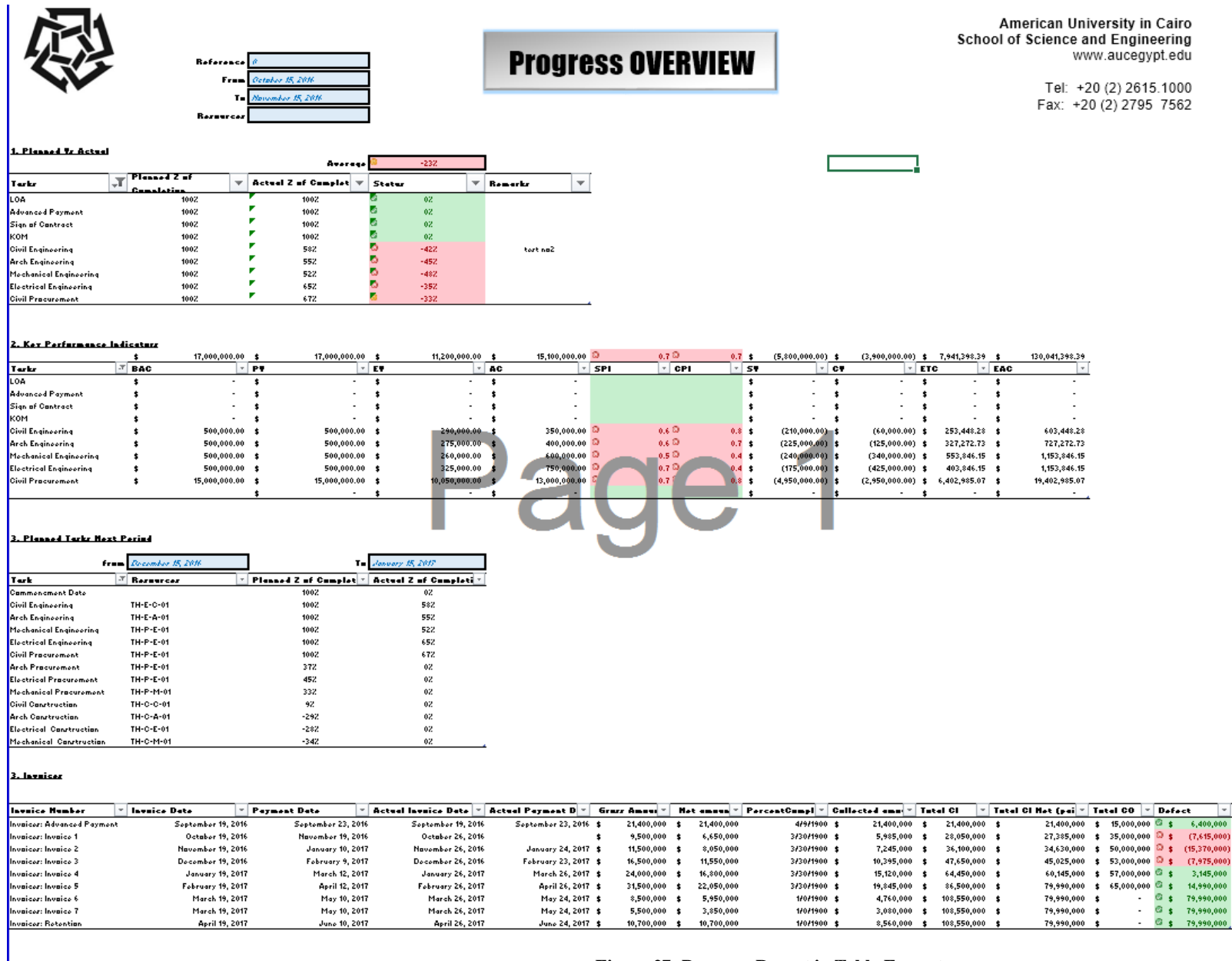
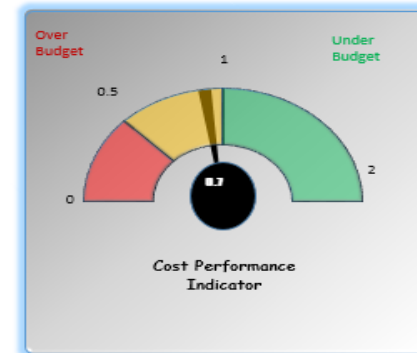
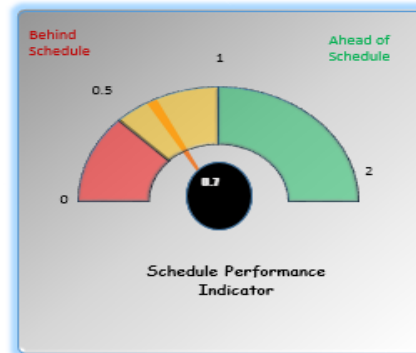
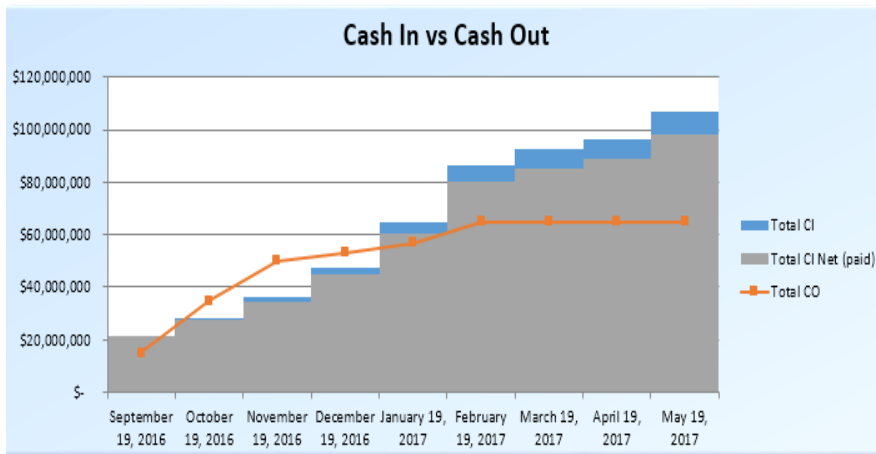
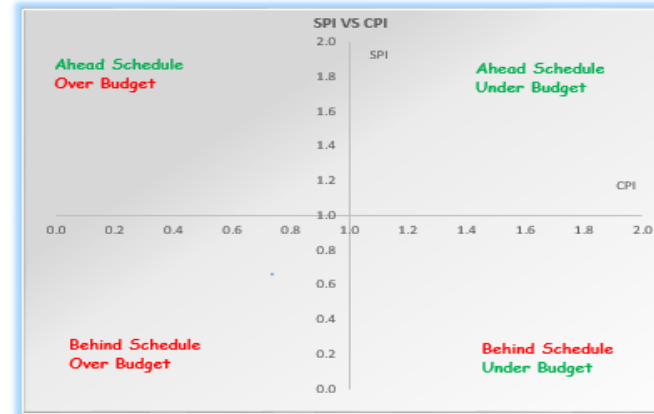
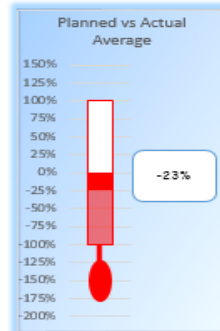
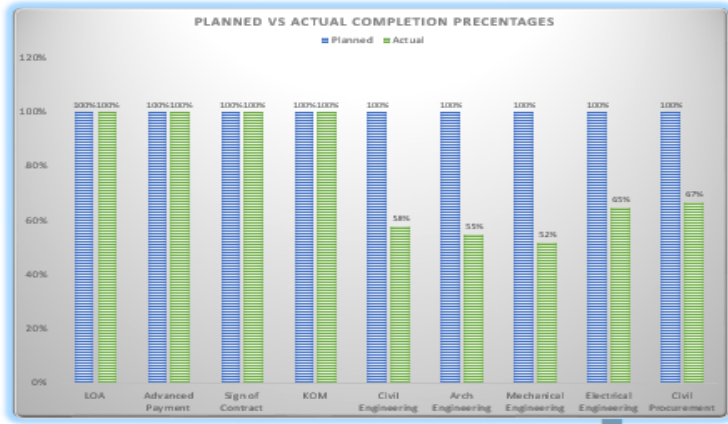


Figure 27: Progress Report in Table Format



Progress Dashboard



2. Key Performance Indicators for each activity

The following indicators were included in the report:

- a. Budget at Completion (BAC)
- b. Planned Value (PV)
- c. Earned Value (EV)
- d. Actual Cost (AC)
- e. Schedule Performance Indicator (SPI)
- f. Cost Performance Indicator (CPI)
- g. Schedule Variance (SV)
- h. Cost Variance (CV)
- i. Estimate to Complete (ETC)
- j. Estimate Actual Cost (EAC)

Appendix B describes more details about used key performance indicators.

To examine if the activity should be included in the report, the following if statement was considered.

if report finish date \geq activity start \therefore activity should be included

Table 5 shows the generated table with the used formulas.

Table 5: Model - Key Performance Indicators for each activity and the used formulas

Field	Used Formula
Task title	
Assigned Resource	
BAC	Imported from the programme

PV	$\min \left\{ \begin{array}{l} BAC \\ BAC \times \frac{\text{report finish date} - \text{activity start date}}{\text{activity finish date} - \text{activity start date}} \end{array} \right.$
EV	$BAC \times \text{Actual \% of Completion}$
AC	Input from the assigned resource
SV	$EV - PV$
CV	$EV - AC$
SPI	EV/PV
CPI	EV/AC

3. Planned Activities Next Period

In this section, an overview for the look ahead activities within the next period was presented. Each activity shall be provided with the planned percentage of completion at the end of the next period and up to date actual percentage of completion. To ensure showing only the activities within the next period, the following if statement was considered:

$$\text{if } \left\{ \begin{array}{l} (\text{report coming period start date} \geq \text{activity start} \ \& \ \text{actual \% of completion} \neq 100\%) \\ \text{or} \\ \text{report coming period start date} \leq \text{activity start} \leq \text{report coming period finish date} \end{array} \right.$$

\therefore activity within period

Table 6 shows the generated table with the used formulas.

Table 6: Model - Look ahead Activities and the used formulas

Task Title	Assigned Resources	Planned % of completion	Actual % of completion
		<p><i>if next coming period finish date</i> \geq <i>activity finish date</i> $\rightarrow 100\%$;</p> <p><i>otherwise</i> $= \frac{\textit{report coming period finish date} - \textit{activity start date}}{\textit{activity finish date} - \textit{activity start date}}$</p>	Latest input from the assigned resource

4. Invoices

In this section an overview about the project invoices and cash flow was provided. Table 7 shows the generated table with the used formula for each field.

Table 7: Model - Invoices and cash flow fields and used formulas

Field	Used Formula
Invoice number	
Planned Invoice Date	From the imported programme
Planned Payment Date	From the imported programme
Actual Invoice Date	Input in the model from the assigned resource for the invoicing
Actual Payment Date	Input in the model from the assigned resource for the invoicing
Gross Amount	From the imported programme
Net Amount	From the imported programme
Collected Amount %	Input in the model from the assigned resource for the invoicing
Total Cash In	Accumulative sum for the collected amounts
Total Cash out	Accumulative sum for the actual costs incurred by all resources
Deficit	Difference between the total cash out and cash in

CHAPTER 4: MODEL VERIFICATION AND VALIDATION

This chapter presents the used verifications and validations measures that were considered to examine the following:

1. The limitation of the proposed model using the verification measures;
2. Examine the accuracy of the model outputs via the validation stage

4.1 Model Verification

4.1.1 Model Verification Methodology

The objective for verifying the model is to test the boundary of the model. After applying the verification measures, the limitations of the model shall be concluded. The verification measures shall be based on ad-hoc technique that assume fictitious cases to examine the limits and boundaries for:

1. number of activities per project
2. number of projects
3. number of resources
4. allocation of resources on projects
5. Projects duration periods
6. SPI and CPI
7. Number of invoices per projects

4.1.2 Model Verification Results

4.1.2.1 Number of activities

6 cases were considered to test the maximum number of activities that the model could bear. As shown in Table 8, it is concluded that 600 activities per project is the limitation for the model. However, if activities exceed this limit, the generated output sheet has to be adjusted manually to accommodate the activities.

Table 8: Model Verification - Number of Activities

Case No	Number of activities	INPUT Module (Interface)		ANALYSIS AND OUTPUT Module		Final Result
		Result	Remarks	Result	Remarks	
1	200	Passed		Pass		Passed
2	400	Passed		Pass		Passed
3	600	Passed		Pass		Passed
4	700	Pass	exporting to output module spent 10 minutes	Failed	Manual adjustment for the output file shall be requested	Failed
5	1000	Pass	exporting to output module spent 13 minutes	Failed	manual adjustment for the output file shall be requested	Failed
6	1300	Pass	exporting to output module spent 16 minutes	Failed	manual adjustment for the output file shall be requested	Failed

4.1.2.2 Number of Projects

To test the limits for number of projects, 5 cases were considered. It is worth to mention that the tests addressed only the input module since the output module are not a function in the number of projects as it is generated per each project individually. Thus, the output modules is excluded from testing the number of projects. Since the INPUT Module is based on a database, the maximum number of projects is the maximum size for the database which is 140 terabytes which is unreachable. However, the tested cases reached 40 projects without any errors as shown in Table 9.

Table 9: Model Verification - Number of Projects

Case No	Number of projects	INPUT Module		ANALYSIS AND OUTPUT Module		Final Result
		Result	Remarks	Result	Remarks	
7	5	Passed			N/A	Passed
8	10	Passed			N/A	Passed
9	16	Passed			N/A	Passed
10	25	Passed			N/A	Passed
11	40	Passed			N/A	Passed
Since the INPUT Module is based on a database, the maximum number of resources is the maximum size for the database which is 140 terabytes which is unreachable.						

4.1.2.3 Number of resources

Since each activity shall be assigned to one resource only as explained in section 3.4.1 , the maximum number of resources shall be correlated to the maximum number of activities per project which are 600 which is unreachable number in construction projects. Nevertheless, 4 cases were tested as shown in Table 10. The model succeed to reach 60 different resources in one project which is reasonably accepted.

Table 10: Model Verification - Number of Resources

Case No	Number of resources	INPUT Module		ANALYSIS AND OUTPUT Module		Final Result
		Result	Remarks	Result	Remarks	
12	15	Passed		Passed		Passed
13	25	Passed		Passed		Passed
14	30	Passed		Passed		Passed
15	60	Passed		Passed		Passed

4.1.2.4 Allocation of resources on projects

This test is aiming to check if common resources could be allocated on different projects. Since the INPUT Module is based on a database, the maximum number of common resources assigned to projects is the maximum size for the database which is 140 terabytes which is unreachable. However, 4 cases were considered as shown in

Table 11. The results show all inserted resources could be allocated to all imported projects without any errors. However, as an opportunity for future development is raised which is to appraise each resource productivity in multiple projects that he is assigned to.

Table 11: Model Verification - Allocation of resources on projects

Case No	Number of common resources	INPUT Module		ANALYSIS AND OUTPUT Module		Final Result
		Result	Remarks	Result	Remarks	
16	15	Passed		Passed		Passed
17	25	Passed		Passed		Passed
18	30	Passed		Passed		Passed
19	60	Passed		Passed		Passed
Since the INPUT Module is based on a database, the maximum number of common resources assigned to projects is the maximum size for the database which is 140 terabytes which is unreachable.						

4.1.2.5 Projects Duration Period

To test the maximum period that the model could endure, 5 cases were studied with maximum 7 years construction period which is very rare figure to be exceed in construction industry. The results shown in Table 12 prove that the model could bear up to 7 years without any errors.

Table 12: Model Verification – Maximum Projects Duration

Case No	Projects Period in months	INPUT Module		ANALYSIS AND OUTPUT Module		Final Result
		Result	Remarks	Result	Remarks	
20	10	Passed		Passed		Passed
21	24	Passed		Passed		Passed
22	35	Passed		Passed		Passed
23	49	Passed		Passed		Passed
24	60	Passed		Passed		Passed
25	72	Passed		Passed		Passed
26	84	Passed		Passed		Passed

However, the projects that has long construction periods exceed 3 years, shall consequently have number of activities exceed the maximum number of activities as described in 4.1.2.1.

4.1.2.6 SPI and CPI

This test is aiming to check the allowable limits for the key performance indicators (KPI) presented by schedule performance indicator (SPI) and cost performance indicator (CPI). 5 cases were considered as shown in Table 13. The results demonstrate that maximum 5 for the SPI and CPI is the limits for the model which is relatively high value. However, if values exceed this limit, the generated output charts has to be adjusted manually to accommodate these value. The input module is excluded from this test as the KPI's are generated only in the output module.

Table 13: Model Verification - Key Performance Indicators

Case No	CPI / SPI Value	INPUT Module		ANALYSIS AND OUTPUT Module		Final Result
		Result	Remarks	Result	Remarks	
20	0.2		N/A	Passed		Passed
21	0.6		N/A	Passed		Passed
22	1		N/A	Passed		Passed
23	2		N/A	Passed		Passed
24	5		N/A	Passed		Passed

4.1.2.7 Number of invoices per project

In order to test the maximum number of invoices that the model could stand without any errors, 5 cases were considered. The results shown in Table 14, presents that 30 invoices is the maximum number of invoices that the model could bear. However, if number of invoices exceed this limit, manual adjustment for the output sheet is required. It is worth to mention that the following constrains have to be considered in the program before importing it to the INPUT Module:

1. Each invoice has to be entered in the programme as an individual activity.

2. The name of invoice activities should include “Invoice” in its titles even though for the advanced payment and retention invoices.

Table 14: Model Verification - Number of Invoices

Case No	Number of invoices	INPUT Module		ANALYSIS AND OUTPUT Module		Final Result
		Result	Remarks	Result	Remarks	
25	12	Passed		Passed		Passed
26	24	Passed		Passed		Passed
27	30	Passed		Passed		Passed
28	36	Passed		Failed	Manual adjustment for the output sheet shall be requested	Failed
29	42	Passed		Failed	manual adjustment for the output sheet shall be requested	Failed

4.1.2.8 Model Verification Results Discussion

Following the 29th tested cases mention in the aforesaid sections, the limits of the proposed model is mentioned in Table 15.

Table 15: Model Limits

Type	Model Limit
Number of activities per project	600
Number of projects	40 projects is the maximum that was tested and passed. It is thought that this number could not be easily exceeded.
Number of resources	60 resource is the maximum that was tested and passed. It is thought that this number could not be easily exceeded.
Allocation of resources on projects	
Projects duration periods	7 years is the maximum that was tested and passed.
SPI and CPI	5
Number of invoices per projects	30

4.2 Model Validation

4.2.1 Introduction

The validation process shall be divided into two phases as follows:

1. A case study project in Cairo, Egypt that is used in validation of the newly developed reporting system presented in this research and discussed in the previous chapter. As-Built data from the case study project for a given period was used to compare the outcomes produced from the execution of the model; including as-built update program for the activities and physical progress, monthly report for the same period of the updated program, KPI's and S curve. In order to test model capabilities and features, the originator of the as-built monthly was selected to run the developed model. To provide a basis for comparison, the as-built KPI's were compared to the KPI's generated from the model, (2) the as-built cash flow was compared to the outputs of model. A comparison between the results was then discussed.
2. Discussion is conducted with the originator of the as-built report mentioned hereinabove as well as further discussions about the developed model are arranged with 8 out of the 29 professional experts, mentioned in section 3.3.2, to examine the features and drawbacks of the proposed reporting technique.

4.2.2 Case Study

4.2.2.1 Project Information

The subject project is four industrial waste water treatment plants with a capacity of 450 m³ / day per each. The Owner was awarded a Design, Build and Operate (DBO) contract to one of the reputable company in the field of water and waste water treatment field in Egypt. The plants shall be constructed in four different factories and

the contract price is 4,000,000 USD (1,000,000 USD per each plant). The planned construction schedule is eight months starting from January 2017. The case study is focused on the construction of one plant only where the data obtained from the site are as follows:

1. An update for one of the plants the programme generated using Premavira P6 software issued on March 31st, 2017 including activities dates, physical progress and KPI's.
2. Monthly report issued with the aforesaid programme.

4.2.2.2 As-built Monthly Report Information

Table 16 and Table 17 summarize the construction information extracted from the programme and monthly report. This information is then used in executing the reporting model, then a comparison between the as-built and the model is conducted. The actual cost incurred by the contractor was not prevailed in the provided monthly report. Thus, any figures related to the actual cost shall be excluded from the validation. Discussion with the responsible person to issue the monthly report was held. It took from him three working days in collecting the data and eight working days to generate the report.

The as-built monthly report has the following sections:

- ***Introduction including Scope of works and narrative projects***
- ***Executive Summary includes the following***
 - Status
 - Sub-contractors information
 - Risks

- **Works progressed/completed during report period**
 - shop/ as-built drawing status
 - Request for Information's Log
 - Request for Approval's Log
- **Procurement /submittal status**
- **Time Schedule**
 - Narrative for the critical Path Analysis
 - Schedule updated (in primavera software)
 - one month look ahead
- **photos with description**
- **Risks and Area of concerns**
- **Commercial**
 - Payment status
 - cash flow forecast
 - Variation Order's
 - Claims
- **QA/QC Records**
 - Inspection Test Plan -ITP list
 - Inspection Request
 - Material inspection Request
 - Nonconformance Records -NCR
- **HSE Records**
 - Site Safety Statistics (incidents, KPI,...)
 - Training Hours

Table 16: Validation Case Study, Activities within period and KPI's provided in the as-built monthly report

Activities within period of report	Planned progress	Physical progress	Deficit	Budget (USD)	AC	PV (USD)	EV (USD)	SV (USD)	SPI	CPI
Engineering	96.05%	94.52%	-1.5%	82,000	No information were provided	78,761	77,506	-1,255	0.98	N/A
Procurement and Fabrication	54.31%	59.81%	+ 5.5%	477,000		259,059	285,294	26,235	1.10	N/A
Civil and Arch. Construction Works	33.43%	33.14%	-0.3%	205,000		68,532	67,937	595	0.99	N/A
Total project	32%	39%	+ 7%	1,000,000		812,703	861,474	48,772	1.21	N/A

Table 17: Validation Case Study, Invoices Details extracted from the as-built monthly report

Invoice Amount	Invoice Amount (USD)	Accumulative (USD)	Planned Payment Date	Actual Payment Date
Advanced Payment (25%)	250,000	250,000	January 31 st , 2017	January 19 th , 2017
Invoice 1	42,857	292,857	March 15 th , 2017	March 30 th , 2017
Invoice 2	50,000	342,857	April 30 th , 2017	N/A
Invoice 3	485,714	828,571	May 31 st , 2017	N/A
Invoice 4	35,714	864,285	June 29 th , 2017	N/A
Invoice 5	17,142	881,428	July 31 st , 2017	N/A
Invoice 6	18,571	900,000	August 31 st , 2017	N/A
Retention (10%)	100,000	1,000,000	August 31 st , 2018	N/A

4.2.2.3 User Friendly Reporting Model Application

The validation of the reporting model was applied on a number of steps as shown in Figure 29. The first step starts by loading the corresponding programme for the same period. Then the model administrator shall be provided with the sufficient access to update the activities within the same period using the provided data in the as-built report. Followed by, the administrator shall generate the report. The last step is to compare the results generated from the model report with the as-built in term of KPI's and cash flows. To examine the software capabilities in an effective manner, the validation steps was conducted by the same person who issue the as-built report mentioned hereinabove.

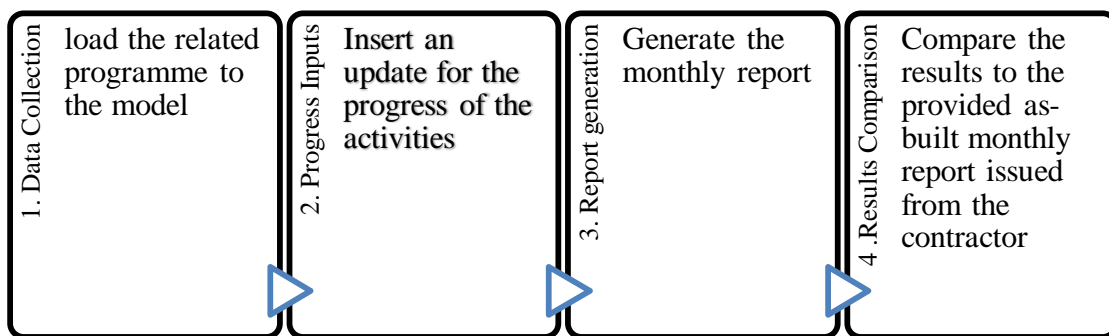


Figure 29: Reporting Model Application Steps

Data Collection

In this step, the related programme was loaded to the model.

Progress Inputs

The progress updated mentioned in the as-built monthly report was updated in the model.

Report Generation

Then the monthly report was generated from the model. The spent time to generate the monthly report using the model was 2 working days considering that the time for collecting data was excluded as it was based on the provided data from the as-built monthly report.

Figure 30, Figure 31 and Figure 32 show some of dashboard graphs. Refer to **Appendix D** for the monthly report generated from the model.

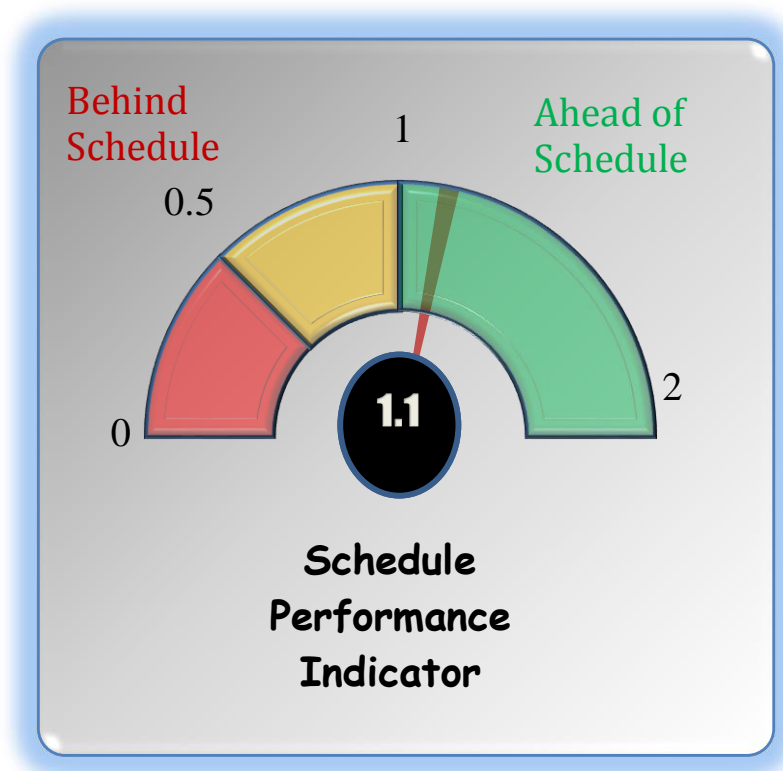


Figure 30: Model Validation Case Study - Model SPI

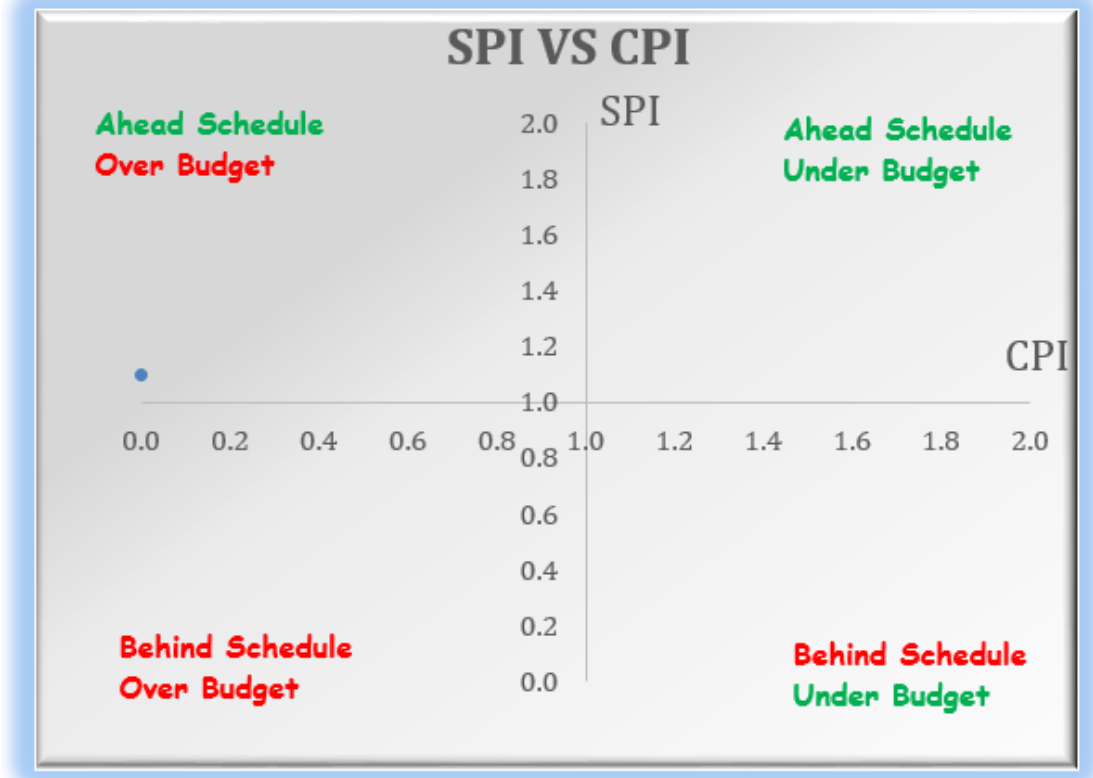


Figure 31: Model Validation Case Study - SPI vs CPI

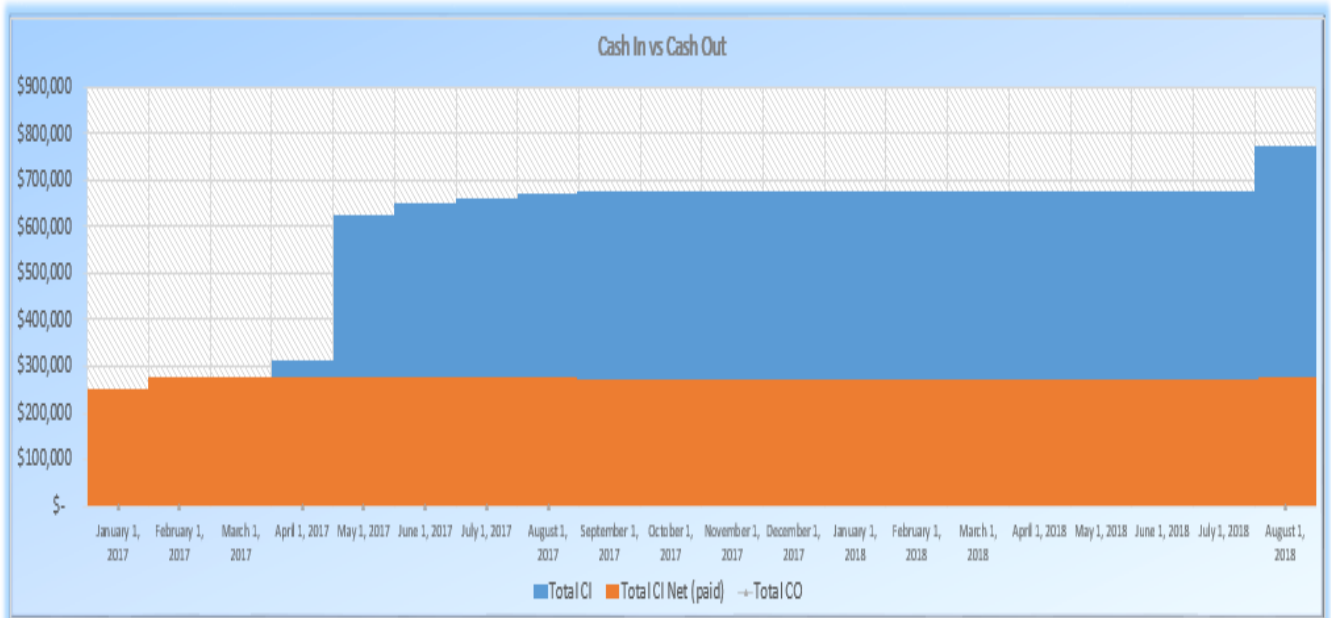


Figure 32: Model Validation Case Study - Cash In

4.2.2.4 Results Comparison

4.2.2.4.1 Spent time

The spent time to generate the monthly report was reduced with a percentage of 66% as shown in Table 18 and Figure 33. It is worth to mention that the data collection is excluded from the comparison as the model was based on the collected data from the as-built report. Moreover, since the as-built report has more sections than the model report such as variation orders, claims, risks, HSE and QA/QC, the time to prepare such sections was also excluded for precise comparison.

Table 18: Spent Time Comparison

	As-built monthly report	User friendly reporting model	Comment
Data collection	3 days	N/A as the model was based on the collected data	
Report generation for the missed sections in the model	2 days	N/A	
Report generation for the common sections in both reports	6 days	2 days	66% reduction

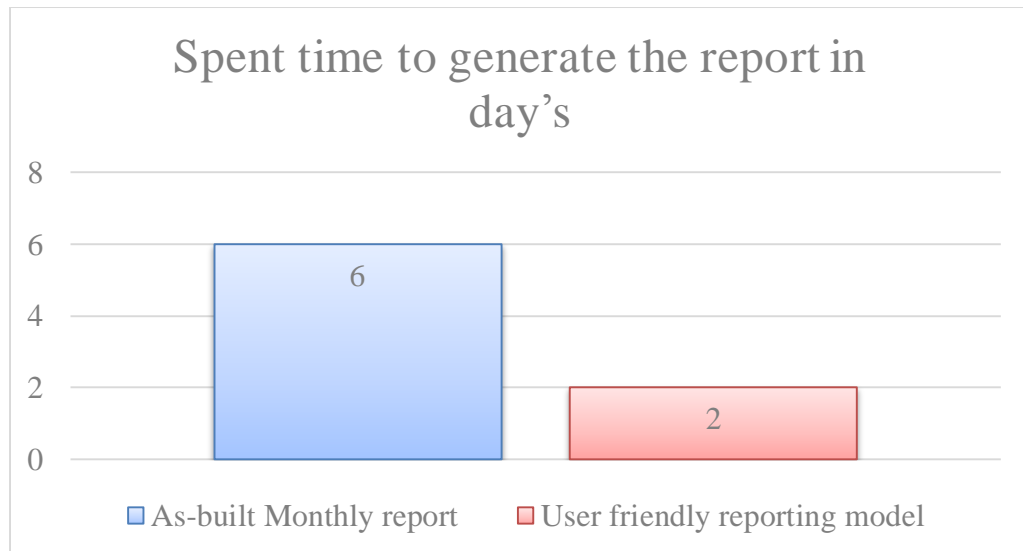


Figure 33: Spent Time Comparison

4.2.2.4.2 Planned Activities Progress

Table 19 and Figure 34 show comparison between the planned and actual progress of the activities stated in the as-built monthly report and the ones generated from the model. The differences are negligible and acceptable.

Table 19: Activities Progress Comparison

Activities within period of report	Planned progress			Actual progress		
	As-built monthly report	Model	Difference	As-built monthly report	Model	Difference
Engineering	96.05%	100%	4%	94.52%	94%	-0.5%
Procurement and Fabrication	54.31%	55%	0.7%	59.81%	60%	0.2%
Civil and Arch. Construction Works	33.43%	34%	0.6%	33.14%	32%	-0.9%

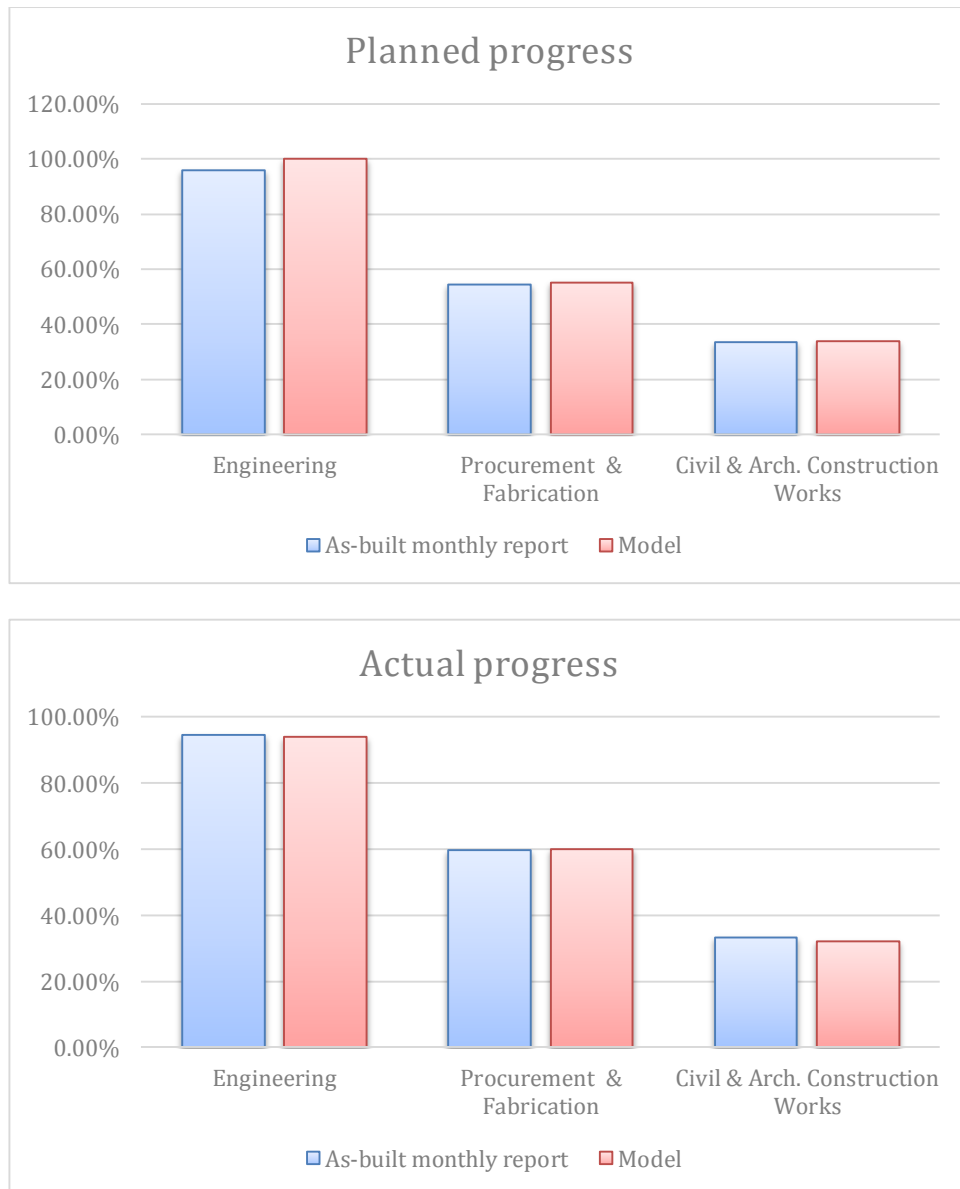


Figure 34: Activities Progress Comparison

4.2.2.4.3 Key Performance Indicators

Table 20 and Figure 35 show comparison between the KPI's stated in the as-built monthly report and the ones generated from the model. The results has a variance plus or minus ten percent which is acceptable.

Table 20: KPI's Comparison

Activities	PV (USD)			EV (USD)			SPI		
	As-built monthly report	Model	Difference	As-built monthly report	Model	Difference	As-built monthly report	Model	Difference
Engineering	78,761	82,022	4%	77,506	77,265	0%	0.98	0.9	-8%
Procurement and Fabrication	259,059	263,698	2%	285,294	284,345	0%	1.1	1.1	0%
Civil and Arch. Construction Works	68,532	70,211	2%	67,937	65,816	-3%	0.99	0.9	-9%
Total project	406,351	415,931	2%	430,737	427,426	-1%	1.2	1.1	-9%

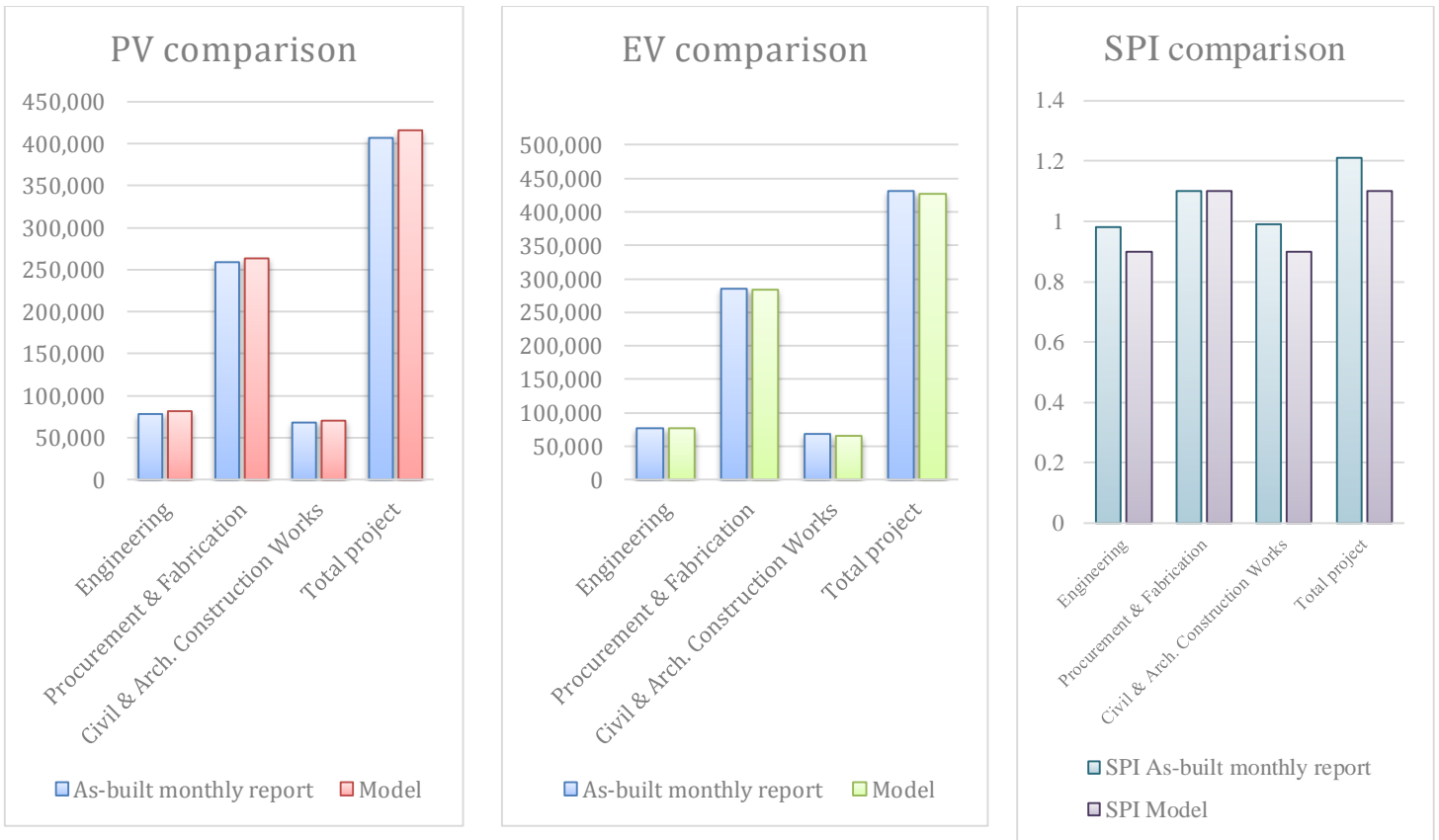


Figure 35: KPI's Comparison

4.2.3 Discussion with Professionals

A discussion with the originator of the as-built report mentioned in the previous case study, who is the same person who test the model, was conducted. He confirmed that

the model presents a simple and feasible way to collect the data and generate an accurate progress reports. He finally concluded that this model is an effective reporting approach.

Out of the 29 professional experts mentioned in section 3.3.2 , 8 professionals were selected randomly to attend a seminar held on May 08th, 2017 by the researcher to present the developed model and get their feedback about it. The seminar took around 20 minutes. The received feedback could be summarized as follows:

1. The developed model proves its simplicity as 100% of the attendees confirmed that the used system is simple and easy to be used. They also confirmed the argument that updating the progress will not take more than 10 minutes from the users every day.
2. The effectiveness of the model was validated as 75% of the attendees confirmed that the using the model shall ease the information flow and help in predicting the problems at an early stage. While the remaining 25% argue that the developed model is missing essential sections in the generated report such as claims, variations and risks which shall affect negatively the information flow.
3. All attendees could not confirm the accuracy of the model as they claiming that they have to test it themselves to validate it. The researcher depends on the case study results mentioned in 4.2.2.4 to prove the accuracy of the developed model.
4. One of the attendees comment on the cost of the proposed model claiming the model price is of value issue that needs to be tackled. If its price is reasonable comparing to other commercial softwares, it will add an edge for developed system considering its simplicity and effectiveness. The

researcher confirms this argument as the developed model is based on a simple programming coding. Thus, the model cost shall be rational.

5. Some drawbacks and recommendations for future research are presented by the attendees as follows:

- Activities on the critical path should be highlighted in the generated report. This will help the project managers to assess the SPI in more accurate manner. For example, if the SPI of an activity is 0.9 this means that the activity is behind the schedule according to the earned value method. But if this activity is not on the critical path and although this activity is behind schedule, its impact on the whole project shall not be tangible compared to the critical activities. The researcher considers this drawback as an opportunity for improvement in the future research.
- The used earned value technique has drawback in the procurement items as it will change the SPI and CPI suddenly. The procurement activities are considered as binary activities (i.e. 0 / 1 activity) and usually it has high planned cost values. For example, the long lead items shall be an enormous portion of the project cost; by this, as long as the long lead items are not delivered to site yet, the SPI and CPI will be adversely affected. While, when it arrives to the site, the CPI and SPI will instantaneously change. To overcome come this drawback, the researcher recommends to divide any procurement activity into 5 activities; purchase ordering, manufacturing, shipping, custom clearance and delivery on site activity. By this, the total value of the items will be distributed over

the 5 activities and SPI and CPI will be presented in more accurate manner.

4.2.4 Model Validation Results Discussion

After arrange discussion with 8 professional experts and after implement the comparison with the case study, although the proposed model has minor variances compared to the as-built monthly report, these variances were expected as a result of having different approaches in calculations and judgment of the as-built monthly report originators. However, the differences are negligible and the proposed model provide additional data and better presentation than the as-built report in terms of activities progress within the periods, look ahead activities and KPI's. The time to generate the monthly report is reduced by 66% using the proposed model.

On the other hand, the proposed model is missing the following sections which are included in the as-built monthly report and recommended by some professionals such as:

- Variation Orders
- Claims
- Risks
- QA/ QC records statistics
- HSE statistics

Additionally, an opportunity for improvement in future research is raised by highlighting the critical activities within the generated report.

As a conclusion, the provided model is proven as a simple and reliable tool to generate progress report in a reasonable time with accepted accuracy. However, it missed some sections that need to be included in the report.

CHAPTER 5: SUMMARY, CONCLUSIONS AND RECOMMENDATION

5.1 Research Summary

Construction industry is always provoked to convey the projects within its planned budget, time plan and the desired quality. Since construction is classified as a fragmented industry which has various and interacting stakeholders, communication is one of the challenges that any construction project is continuously facing. Construction projects are vulnerable to failure as a result for loosing good communication and reporting systems for all stakeholders.

On the off chance that any of stakeholder's management has limited information about the status of the project, wrong choices may be taken which jeopardize the project's success and heighten risks. It additionally could thwart the team members and affect their productivity due to a disorderly and enormous measure of data that make it hard to be followed.

More simple and accurate communication and reporting techniques can ease the construction process and assist projects managers to facilitate the decision making process, measuring the productivity and anticipating any contingencies that are

translated to extra costs and time. The current practices of communication and reporting techniques are lacking for simplicity versus the accuracy. The more accurate techniques are generated through complicated processes which takes long time in the preparation. While the simple techniques are fronting with the lacking in the precision.

This research presented an approach in generating communication and reporting model that combine the precision with the simplicity. The newly developed algorithm was based on the acquired knowledge from literature study and conducting direct structured interviews with 29 professional experts in construction field. The developed model is suitable to be run on Microsoft Windows platform. In addition, Microsoft Excel and a setup for an “EXE” file which was built using C# programming language is necessary to launch the model. The proposed model is divided into two modules; the INPUT Module which consist of a user interface reporting model that gather simple information about projects’ activities progress per each assigned resources, the ANALYSIS AND OUTPUT MODULE where simple report is generated through analyze the provided information from the INPUT Module and translate it to dashboards including some tables and graphs. The input module is depending on updating 6 fields only for each activity by its assigned resource. The required fields are as follows: actual start date, expected finish date, incurred actual costs, percent of completion, remarks and pictures for the progress. Updating such fields shall not consume much time from the assigned resource to update it on a daily basis. Using the updates, the output report shall consists from four tables and graphs; comparison for the planned versus actual activities progress, key performance indicators for the total project and each activity as well, look ahead for the activities that should took place in the coming period and finally the cash flow analysis.

The model was tested using a pilot project that explain the working principle of the model supported by screenshots from the inputs and outputs. Followed by testing the model's limits and boundaries by applying 31 fictitious cases to examine model parameters. Subsequently, the validation of the model was conducted on two stages; (1) a case study project was used to validate the outcomes from the reporting model. The case study was for an industrial waste water plant project, (2) a discussion with 8 out of 29 professional experts was held to present the developed model. The results from the validation stages demonstrate the simplicity, accuracy and efficiency of the proposed model. The model achieved to reduce the report preparation time by 66%. Thus, the model could assist in cutting the costs and accelerating collecting information for project's activities. Moreover, comparing the results from the key performance indicators with the as-built report show the high accuracy of the model as it has minimal variance than the as-built information. On contraire, the developed model misses some qualitative sections such as variation orders, claims, and risks.

5.2 Research Conclusion

- Communication and reporting measures is an of value issue that need to be precisely tackled due to its high influence on project's objectives.
- The characteristic for good progress report, as mentioned in the literature and 29 direct interviews with professional experts, is to be simple, comprehensive, short, and accurate. It should give a whole idea about the achieved progress and look ahead prediction for the project status in the coming periods as well as the incurred costs and the problems that the project is facing.

- There are numerous attempts in literature to use automated modeling for the generation of the progress reports. The mentioned practices relies either on complicated processes that are designated as a time consumable or on simple procedure that lacks the required accuracy.

Based on the current developments, this research makes a number of contributions:

- This research presents a new reporting technique that achieved the balance between simplicity, accuracy and effectiveness for the flow of information at a reasonable cost. The model has been tested, verified and validated by external users who confirm that the four concepts are fulfilled as follows:
 - Simplicity:
 - The results of using the proposed interface show that the used technique could be easily performed on a daily basis as it is not time consumable. The simplicity of the interface and the minimal required data (i.e. 6 fields only), shall allow each user to update his related activities in less than 10 minutes. Consequently, the report shall be generated automatically.
 - Accuracy:
 - The accuracy of the proposed model is tested by comparing the results produced from the model with the case study's as-built report highlight the robust features of the model as the generated report using the model was almost equal to the as-built figures.

- Effectiveness:
 - The proposed interface achieved to balance between preserving working time and effectiveness. Through simple inputs and comprehensive outputs presented by tables and graphs, it gives an overview for the managers to understand the status of their projects for each activity and its assigned resources, evaluate the progress through the key performance indicators, predict and mitigate the problems at early stage and appraise the project's cash flow to act in a proactive mode not in a reactive one.
 - The model has demonstrated significant productivity improvements in terms of the number of man-hours required for generating the progress report and enhance the transfer of information between projects' team as well. This will allow contractor to cut-off the incurred overhead costs.
 - The proposed model are missing some sections that need to be included in future such as variation orders, claims, risks, HSE and QA/ QC records statistics. However, since most of these sections are based on qualitative data and require more manually entered data rather than automated data, it could be attached manually to the output report.
- Model's Cost:
 - The proposed model is based on a simple programming coding. Thus, the model's cost is rational comparing to other commercial softwares

which give an edge to the proposed model tacking into consideration its simplicity, accuracy and effectiveness.

- As a conclusion, the provided model is suitable for any firms as long as it has a planned activities associated with costs. The model is able to generate a simple, reliable and effective progress report in a reasonable time with accepted accuracy at a reasonable price.

5.3 Recommendation for Future Research

Despite the capabilities and benefits of the developed reporting system, it has limitations that could be improved through further research as follows:

- More development in the generated report to include summary for variations, claims, HSE and QA statistics occurred in the project.
- The ability to export the updated inputs from each activity to any scheduling software shall reduce the spent time to generate an updated programme.
- Developing a mobile application for the suggested model connected to a clouded server, shall definitely increase the simplicity and efficiency of the model.
- Minimizing the updating fields shall also ease model using. This might be achieved for example by removing the actual incurred costs which was assumed to be updated by each resource. It could be replaced by an

automatic calculation through multiplying the activity budget by the updated percent of completion.

- More development to tackle each resource productivity who is assigned to multiple projects shall strengthen the model.
- Highlighting the critical activities within the generated progress report shall enforce the model.

REFERENCES

- Abbasi, N., Wajid, I., Iqbal, Z., & Zafar, F. (2014). Project Failure Case Studies and Suggestion. *International Journal of Computer Applications*, 86(6), 34-49.
- Allen, D., & Alexander, B. (2013). Data Collection for Construction Projects. *Construction Advisory Report*, 1-6.
- Amstrong, G., & Gilge, C. (2010). Effective Reporting for Construction Projects: Increasing the Likelihood of Project Success. *KPMG Global Energy Institute*.
- Branstetter, S. (2015). *Killer Time Savers: Automate Weekly Status Reports* . Retrieved from Smartsheets Blog: <https://www.smartsheet.com/blog/killer-time-savers-automate-weekly-status-reports> (accessed on 2016, August 06).
- Burr, T. (2009). *The Failure of Metronet*. London: National Audit Office: Department for Transport.
- Buyse, P., & Vandenbussche, T. (2010). *Performance analysis of Earned Value Management in the construction industry*. UNIVERSITEIT GENT.
- Czarnigowska, A. (2008). Earned value method as a tool for project control. *Budownictwo i Architektura* 3, 15-32.
- El-Omari, S., & Moselhi, O. (2010). Automation in Construction. *Elsevier B.V.*, 20, 699-705.
- Furst, P. (2012). *Ineffective Construction Management Practices and Their Impact on Project Safety*. Retrieved from International Risk Management Institute, Inc:

- <https://www.irmi.com/articles/expert-commentary/ineffective-construction-management-practices> (accessed on 2016, July 07).
- Garemo, N., Matzinger, S., & Palter, R. (2015). *Megaprojects: The good, the bad, and the better*. Retrieved from McKinsey & Company Home Website: <http://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/megaprojects-the-good-the-bad-and-the-better> (accessed on 2016, August 10).
- Hoezen, M., Reymen, I., & Dewulf, G. (2006). The problem of communication in construction. *University of Twente*.
- King, T. (2017). *Assessment of Problems Associated With Poor Project Management Performance*. Colorado: Long International.
- Monterroso, Y. (2013). *PMI Study Reveals Poor Communication Leads to Project Failure One Third of the Time*. Retrieved from PMI Study Reveals Poor Communication Leads to Project Failure One Third of the Time: <http://www.coreworx.com/pmi-study-reveals-poor-communication-leads-to-project-failure-one-third-of-the-time/> (accessed on 2016, June 15).
- Morgan, A. (2010). Does poor project governance cause delays? *Does poor project governance cause delays?* United Kingdom: PricewaterhouseCoopers LLP.
- PMBOK. (2013). *A guide to the Project Management Body of Knowledge (PMBOK Guide)*. Project Management Institute.
- PMI. (2013). Communication: The Message Is Clear . *Project Management Institute, Inc.* .
- Redmond, R. (2014). How to Report Status on a Project. *Project Smart*. Retrieved from How to Report Status on a Project.

- Shiau, Y.-C., & Wang, W.-C. (2002). Daily Report Module for Construction Management Information System. *Chung-Hua University, Department of Construction Engineering, Taiwan, 603-608.*
- Syndicate, T. E. (2016). *The Egyptian Engineering Syndicate Webpage*. Retrieved from <http://eea.org.eg/> (accessed on 2016, June 1).
- TenStep, I. (2016). *Projectkickstart Homepae*. Retrieved from Poor project communication will cause many projects to end unsuccessfully: <https://www.projectkickstart.com/downloads/tips11-poor-project-communication.cfm> (accessed on 2016, September 15).
- Xu, S., & Luo, H. (2014). The Information-related Time Loss on Construction Sites: A Case Study on Two Sites. *International Journal of Advanced Robotic Systems*.

APPENDIX A: INTERVIEW WITH PROFESSIONALS

A.1 Interview Survey

Survey for using “USER FRIENDLY PROGRESS REPORTING SYSTEM IN CONSTRUCTION PROJECTS”

Thank you for agreeing to participate in this survey which is part in partial fulfillment of the requirements for the degree of Master of Science in Construction Engineering in the American University in Cairo. This survey is designed by Mohamed ELGINDI. It should only take 10 to 15 minutes. Please be assured that all answers and information provided by you shall remain strictly confidential.

Interviewee Information

Number					
Current Position				Date	
Current Company Type	<input type="checkbox"/> Client	<input type="checkbox"/> Engineer	<input type="checkbox"/> Contractor		
Years of Experience	<input type="checkbox"/> Client	<input type="checkbox"/> Engineer	<input type="checkbox"/> Contractor	<input type="checkbox"/> Total	
Qualification	<input type="checkbox"/> Ph.D	<input type="checkbox"/> M.Sc.	<input type="checkbox"/> B.Sc.	<input type="checkbox"/> Other	
Interview / Questionnaire Type	<input type="checkbox"/> Face to Face	<input type="checkbox"/> Telephone	<input type="checkbox"/> Other		

Questions

1. Please arrange the best tools in your opinion to measure the project progress?

	1: Very Important Tool	2: Important Tool	3: Not Important Tool
Progress Reports	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comments</i>			
Direct Meetings	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comments</i>			
Site Tours	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comments</i>			
Conferences / Phone Calls	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comments</i>			
Others	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comments</i>			

2. Did you worked before in a firm that has an automated reporting technique?

	1: Yes	2: No
	<input type="checkbox"/>	<input type="checkbox"/>

If yes, please specify the used technique

3. In your opinion, do you think that having an automated reporting technique might be useful in tracing the progress?

	1: Yes	2: No	3: Not Sure
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comment

4. Please check the progress report contents from your point of view.

	1: Yes	0: No
Project's Information	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comment</i>		
Report information	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comment</i>		
Executive Summary	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comment</i>		
S curve and cash flow analysis	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comment</i>		
Planned activities percentage of completion	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comment</i>		
Key performance indicators using Earned Value Measurement Technique	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comment</i>		
Planned activities in the coming periods	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comment</i>		
Logs (transmittals, RFA's, RFI's)	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comment</i>		
Variation Orders Summary	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comment</i>		
Claims	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comment</i>		
QA / QC Records	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comment</i>		
HSE Records	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comment</i>		
Risks and Area of concerns	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comment</i>		
Progress Pictures	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comment</i>		
Others	<input type="checkbox"/>	<input type="checkbox"/>
<i>Comment</i>		

Open Discussion

ADDITIONAL COMMENTS

Appendix A: Interview With professionals

A.2 Interview Survey Results

APPENDIX B: KEY PERFORMANCE INDICATORS

B.1 Abstract

In this research, the used technique to develop the key performance indicator is the earned value. Earned value is a common project's management approach to control the schedule and budget of the project. It depends on a concluding a few simple rates that allow the project's manager to predict the most likely final results and effect. It gives a better understandings for the actual status of the project and extrapolate the current trend of the project to predict the end result of the project. This will allow the projects manager to take the necessary measures to rectify the trend of the project whenever needed (Czarnigowska, 2008).

B.2 Purpose of the Earned value

Earned value method was developed to answer the following questions:

1. Is the project behind or ahead schedule?
2. Is the project over or under budget?
3. What is the estimate and budget at the completion of the project?
4. What is the estimated finish date of the project?

B.3 Definitions and Equations

The earned value technique terminologies are as follows: (Buyse and Vandebussche, 2010)

Budget Actual Cost (BAC): the total cost of the project or activity as it was budgeted at the start of the project

Planned Value (PV): is the value that was planned to have been spent according to the baseline plan at a certain point in time.

Actual Cost (AC): is the value that was spent for all work which is executed at a certain point in time.

Earned Value (EV): is the monetary value of the activities that are finished at a certain point in time. The EV equals the BAC multiplied by the percentage completed (PC) at a certain point in time ($EV = BAC \times PC$)

Schedule Variance (SV): one of the performance measure which represent the difference between the current status of the project and its baseline, in schedule terms. ($SV = EV - PV$). If SV has positive value, it means the project is ahead of schedule. On contraire, if SV has a negative value, it means the project is behind schedule while if it equals to 1, it means that the project is on schedule.

Cost Variance (CV): another performance measure which represent the difference between the current status of the project and its baseline, in monetary terms. ($CV =$

$EV - AC$). If CV has positive value, it means the project is under budget. On contraire, if CV has a negative value, it means the project is over budget while if it equals to 1, it means that the project's budget is as planned.

Schedule Performance Indicator (SPI): one of performance measures which shows whether the project is performing on schedule or not. It is generated using the following formula:

$$SPI = \frac{EV}{PV} \quad \text{where; } \begin{cases} SPI > 1 ; \text{ project is ahead of scheule} \\ SPI < 1 ; \text{ project is behind scheule} \\ SPI = 1 ; \text{ project is on scheule} \end{cases}$$

Cost Performance Indicator (CPI): another performance measures which expresses the cost efficiency of the executed work. It is generated using the following formula:

$$CPI = \frac{EV}{AC} \quad \text{where; } \begin{cases} CPI > 1 ; \text{ project is under budget} \\ CPI < 1 ; \text{ project is over budget} \\ CPI = 1 ; \text{ project is on budget} \end{cases}$$

Estimate to Completion (ETC): is an estimate of the cost for the remaining works. It is calculated using the following equation:

$$ETC = \text{Performance factor} \times (BAC - EV)$$

; where the Performance factor equal to $\left\{ \begin{array}{l} 1 : \text{optimistic result} \\ 1/CPI : \text{most likely result} \\ 1 \times SPI/CPI : \text{pessemistic result} \end{array} \right.$

Estimate Actual Cost (EAC): is an estimate to the whole spent cost for the project

$$(EAC = ETC + AC)$$

B.4 Earned Value Curves

Figure 36 shows the earned value curves in relation to cost and time. The original planned S curve is shown by green and the actual cost is appeared by red while the earned value curve is presented by blue. At any given time, the differences between the three curves provides the variances either schedule variance or the cost variance.

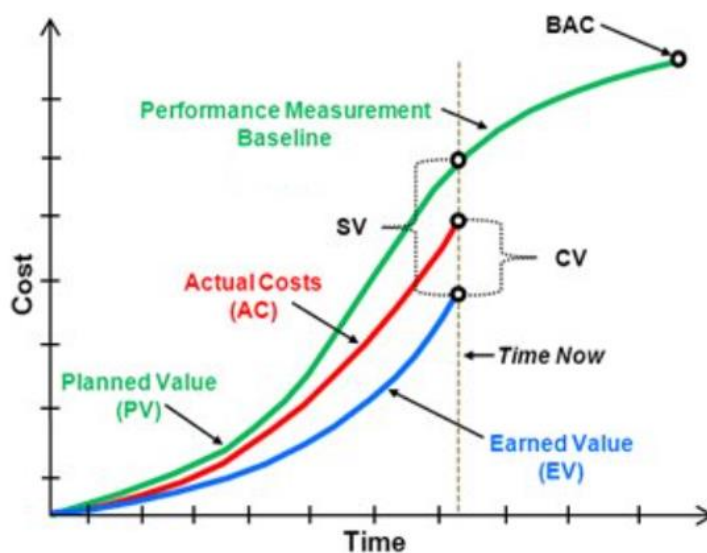


Figure 36: Earned Value Curve (Czarnigowska, 2008)

APPENDIX C: PROGRAMMING CODE

Login

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SQLite;
using System.Xml.Linq;

namespace Constructions
{
    public partial class Login : Form
    {
        public SQLiteConnection sql_con;
        public SQLiteCommand sql_cmd;
        SQLiteDataReader sql_datareader;
        public class ComboboxItem
        {
            public string Text { get; set; }
            public object Value { get; set; }

            public override string ToString()
            {
                return Text;
            }
        }
        public Login()
        {
            InitializeComponent();
            Extension.SetMainConnection(ref sql_con);
            SQLiteCommand GetTasks = new SQLiteCommand("Select ProjectID,Name As ProjectName From
Project", sql_con);
            try
            {
                sql_con.Open();
                sql_datareader = GetTasks.ExecuteReader();
                while (sql_datareader.Read() // Read() returns true if there is still a result line to read
                {
                    ComboboxItem item = new ComboboxItem();
                    item.Text = sql_datareader["ProjectName"].ToString();
                    item.Value = sql_datareader["ProjectID"].ToString();
                    ddlProjects.Items.Add(item);
                    ddlProjects.SelectedIndex = ddlProjects.Items.Count - 1;
                }
                sql_con.Close();
            }
            catch (Exception ex)
            {
                throw new Exception(ex.Message);
            }
        }

        private void btnLogin_Click(object sender, EventArgs e)
        {
            Extension.SetMainConnection(ref sql_con);
            SQLiteCommand GetUserLogin = new SQLiteCommand("Select
ResourceUID,ResourceInitials,IsAdministrator From UserLogins Where UserName=@UserName And
Password=@Password", sql_con);
            GetUserLogin.Parameters.Add(new SQLiteParameter("@UserName", txtUserName.Text));
            GetUserLogin.Parameters.Add(new SQLiteParameter("@Password", txtPassword.Text));
            try

```

```

    {
        sql_con.Open();
        sql_datareader = GetUserLogin.ExecuteReader();
        if (sql_datareader.HasRows)
        {
            while (sql_datareader.Read()) // Read() returns true if there is still a result line to read
            {
                MyGlobals.UserName = txtUserName.Text;
                MyGlobals.Password = txtPassword.Text;
                MyGlobals.ProjectID =
                Int32.Parse(((Constructions.Login.ComboboxItem)(ddlProjects.SelectedItem)).Value.ToString());
                MyGlobals.ProjectName =
                ((Constructions.Login.ComboboxItem)(ddlProjects.SelectedItem)).Text.ToString();
                MyGlobals.ResourceUID = Int32.Parse(sql_datareader["ResourceUID"].ToString());
                MyGlobals.ResourceInitials = sql_datareader["ResourceInitials"].ToString();
                MyGlobals.IsAdministrator = Boolean.Parse(sql_datareader["IsAdministrator"].ToString());
                TasksMain TaskMain = new TasksMain();
                TaskMain.Show();
                this.Hide();
            }
        }
        else
        {
            lblErrorMessage.Text = "Login Failed!";
        }
        sql_con.Close();
    }
    catch (Exception ex)
    {
        lblErrorMessage.Text = ex.Message;
    }
}
}
}

```

Tasks Search

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SQLite;
using DGVPrinterHelper;
using Microsoft.Office.Interop;
using System.IO;
using System.Drawing.Imaging;
using System.Reflection;

namespace Constructions
{
    public partial class TasksSearch : Form
    {
        public SQLiteConnection sql_con;
        public SQLiteCommand sql_cmd;
        public SQLiteDataReader sql_datareader;
        public class ComboboxItem
        {
            public string Text { get; set; }
            public object Value { get; set; }
        }
    }
}

```

```

public override string ToString()
{
    return Text;
}
}
public TasksSearch()
{
    InitializeComponent();
    dtpStartFrom.Format = DateTimePickerFormat.Custom;
    dtpStartFrom.CustomFormat = " ";
    dtpStartTo.Format = DateTimePickerFormat.Custom;
    dtpStartTo.CustomFormat = " ";
    dtpFinishFrom.Format = DateTimePickerFormat.Custom;
    dtpFinishFrom.CustomFormat = " ";
    dtpFinishTo.Format = DateTimePickerFormat.Custom;
    dtpFinishTo.CustomFormat = " ";
    Extension.SetMainConnection(ref sql_con);
    SQLiteCommand GetTasks = new SQLiteCommand("Select UID,Name As ResourceName From
Resources", sql_con);
    try
    {
        sql_con.Open();
        sql_datareader = GetTasks.ExecuteReader();
        while (sql_datareader.Read() // Read() returns true if there is still a result line to read
        {
            ComboboxItem item = new ComboboxItem();
            item.Text = sql_datareader["ResourceName"].ToString();
            item.Value = sql_datareader["UID"].ToString();
            cbResources.Items.Add(item);
        }
        sql_con.Close();
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
}

private void dgvTasks_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    int TaskUID = Int32.Parse(dgvTasks.Rows[e.RowIndex].Cells["UID"].Value.ToString());
    TasksLog TaskLog = new TasksLog(Int32.Parse(TaskUID.ToString()));
    TaskLog.Show();
    this.Hide();
}

private void btnSearch_Click(object sender, EventArgs e)
{
    string strQuery = "Select ID,UID,Name As TaskName,(Select Name From Resources Where UID=(Select
Max(ResourceUID) From Assignments Where Assignments.TaskUID=Tasks.UID And ProjectID=" +
MyGlobals.ProjectID + ") As
ResourceName,Start,Finish,ActualStart,ActualFinish,FixedCostAccrual,FixedCost,PercentComplete From Tasks
Where ProjectID=" + MyGlobals.ProjectID + " ";
    if (cbResources.SelectedItem != null)
    {
        if (Extension.IsNumeric(((ComboboxItem)cbResources.SelectedItem).Value))
            strQuery += " And UID in (Select TaskUID From Assignments Where ResourceUID=" +
((ComboboxItem)cbResources.SelectedItem).Value + ")";
    }
    if (txtTaskName.Text.Trim() != "")
        strQuery += " And Name Like %" + txtTaskName.Text + "%";
    if (dtpStartFrom.Text.Trim() != "")
        strQuery += " And Start >= " + dtpStartFrom.Text + """;
    if (dtpStartTo.Text.Trim() != "")
        strQuery += " And Start <= " + dtpStartTo.Text + """;
    if (dtpFinishFrom.Text.Trim() != "")

```

```

    strQuery += " And Finish >= " + dtpFinishFrom.Text + """;
if (dtpFinishTo.Text.Trim() != "")
    strQuery += " And Finish <= " + dtpFinishTo.Text + """;
Extension.SetMainConnection(ref sql_con);
SQLiteCommand GetTasks = new SQLiteCommand(strQuery, sql_con);
try
{
    sql_con.Open();
    sql_datareader = GetTasks.ExecuteReader();
    DataTable dt = new DataTable();
    dt.Clear();
    dt.Columns.Add("UID");
    dt.Columns.Add("TaskName");
    dt.Columns.Add("ResourceName");
    dt.Columns.Add("Start");
    dt.Columns.Add("Finish");
    dt.Columns.Add("ActualStart");
    dt.Columns.Add("ActualFinish");
    dt.Columns.Add("FixedCost");
    dt.Columns.Add("FixedCostAccrual");
    dt.Columns.Add("PercentComplete");
    double CostsSum = 0;
    while (sql_datareader.Read()) // Read() returns true if there is still a result line to read
    {
        DataRow _ravi = dt.NewRow();
        _ravi["UID"] = sql_datareader["UID"];
        _ravi["TaskName"] = sql_datareader["TaskName"];
        _ravi["ResourceName"] = sql_datareader["ResourceName"];
        _ravi["Start"] = sql_datareader["Start"];
        _ravi["Finish"] = sql_datareader["Finish"];
        _ravi["ActualStart"] = sql_datareader["ActualStart"];
        _ravi["ActualFinish"] = sql_datareader["ActualFinish"];
        _ravi["FixedCost"] = sql_datareader["FixedCost"];
        _ravi["FixedCostAccrual"] = sql_datareader["FixedCostAccrual"];
        _ravi["PercentComplete"] = sql_datareader["PercentComplete"];
        dt.Rows.Add(_ravi);
        if (Extension.IsNumeric(sql_datareader["FixedCostAccrual"].ToString()))
            CostsSum += Double.Parse(sql_datareader["FixedCostAccrual"].ToString());
    }
    dgvTasks.AutoGenerateColumns = false;
    dgvTasks.DataSource = dt;
    lblSumForCosts.Text = "Total Cost: " + CostsSum.ToString();
    sql_con.Close();
}
catch (Exception ex)
{
    throw new Exception(ex.Message);
}

private void dtpStartFrom_ValueChanged(object sender, EventArgs e)
{
    dtpStartFrom.Format = DateTimePickerFormat.Custom;
    dtpStartFrom.CustomFormat = "yyyy-MM-ddTHH:mm:ss";
}

private void dtpStartTo_ValueChanged(object sender, EventArgs e)
{
    dtpStartTo.Format = DateTimePickerFormat.Custom;
    dtpStartTo.CustomFormat = "yyyy-MM-ddTHH:mm:ss";
}

private void dtpFinishFrom_ValueChanged(object sender, EventArgs e)
{
    dtpFinishFrom.Format = DateTimePickerFormat.Custom;
    dtpFinishFrom.CustomFormat = "yyyy-MM-ddTHH:mm:ss";
}

```



```

}

private void dtpFinishTo_ValueChanged(object sender, EventArgs e)
{
    dtpFinishTo.Format = DateTimePickerFormat.Custom;
    dtpFinishTo.CustomFormat = "yyyy-MM-ddTHH:mm:ss";
}

private void btnPrint_Click(object sender, EventArgs e)
{
    //printDocument1.Print();
    DGVPrinter printer = new DGVPrinter();

    printer.Title = "Tasks";

    //printer.SubTitle = "An Easy to Use DataGridView Printing Object";

    printer.SubTitleFormatFlags = StringFormatFlags.LineLimit |
        StringFormatFlags.NoClip;

    printer.PageNumbers = true;

    printer.PageNumberInHeader = false;

    printer.PorportionalColumns = true;

    printer.HeaderCellAlignment = StringAlignment.Near;
    printer.Footer = "";
    printer.FooterSpacing = 15;
    printer.PrintDataGridView(dgvTasks);
}

private void printDocument1_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
{
    Bitmap bm = new Bitmap(this.dgvTasks.Width, this.dgvTasks.Height);
    this.dgvTasks.DrawToBitmap(bm, new Rectangle(0, 0, this.dgvTasks.Width, this.dgvTasks.Height));
    e.Graphics.DrawImage(bm, 0, 0);
}

private void btnMainPage_Click(object sender, EventArgs e)
{
    TasksMain TaskMain = new TasksMain();
    TaskMain.Show();
    this.Hide();
}

private void btnExportTasks_Click(object sender, EventArgs e)
{
    string strQuery = "Select Name As TaskName,(Select Name From Resources Where UID=(Select
Max(ResourceUID) From Assignments Where Assignments.TaskUID=Tasks.UID)) As
ResourceName,Start,Finish,ActualStart,ActualFinish,FixedCostAccrual,PercentComplete From Tasks Where
ProjectID=" + MyGlobals.ProjectID + " ";
    if (cbResources.SelectedItem != null)
    {
        if (Extension.IsNumeric(((ComboBoxItem)cbResources.SelectedItem).Value))
            strQuery += " And UID in (Select TaskUID From Assignments Where ResourceUID=" +
                ((ComboBoxItem)cbResources.SelectedItem).Value + ")";
    }
    if (txtTaskName.Text.Trim() != "")
        strQuery += " And Name Like '%" + txtTaskName.Text + "%'";
    if (dtpStartFrom.Text.Trim() != "")
        strQuery += " And Start >=" + dtpStartFrom.Text + " ";
    if (dtpStartTo.Text.Trim() != "")
        strQuery += " And Start <=" + dtpStartTo.Text + " ";
}

```

```

if (dtpFinishFrom.Text.Trim() != "")
    strQuery += " And Finish >= " + dtpFinishFrom.Text + """;
if (dtpFinishTo.Text.Trim() != "")
    strQuery += " And Finish <= " + dtpFinishTo.Text + """;
Extension.SetMainConnection(ref sql_con);
SQLiteCommand GetTasks = new SQLiteCommand(strQuery, sql_con);
try
{
    sql_con.Open();
    sql_datareader = GetTasks.ExecuteReader();
    DataTable dt = new DataTable();
    dt.Load(sql_datareader);
    ExportToExcel(dt);
    sql_con.Close();
}
catch (Exception ex)
{
    throw new Exception(ex.Message);
}
}

private void btnExportTasksDetails_Click(object sender, EventArgs e)
{
    string strQuery = "Select (Select Name From Tasks Where UID=TasksLog.TaskUID) As
TaskName,Start,Finish,ActualStart,ExpectedEndDate,ActualCost,PercentWorkComplete,DateEntered,Remarks,Ph
oto From TasksLog Where ProjectID=" + MyGlobals.ProjectID + " And TaskUID in (Select UID From Tasks
Where ProjectID=" + MyGlobals.ProjectID + " ";
    if (cbResources.SelectedItem != null)
    {
        if (Extension.IsNumeric(((ComboBoxItem)cbResources.SelectedItem).Value))
            strQuery += " And UID in (Select TaskUID From Assignments Where ResourceUID=" +
(((ComboBoxItem)cbResources.SelectedItem).Value + ")";
    }
    if (txtTaskName.Text.Trim() != "")
        strQuery += " And Name Like '%" + txtTaskName.Text + "%'";
    if (dtpStartFrom.Text.Trim() != "")
        strQuery += " And Start >= " + dtpStartFrom.Text + """;
    if (dtpStartTo.Text.Trim() != "")
        strQuery += " And Start <= " + dtpStartTo.Text + """;
    if (dtpFinishFrom.Text.Trim() != "")
        strQuery += " And Finish >= " + dtpFinishFrom.Text + """;
    if (dtpFinishTo.Text.Trim() != "")
        strQuery += " And Finish <= " + dtpFinishTo.Text + """;
    strQuery += ") order by TaskUID";
    Extension.SetMainConnection(ref sql_con);
    SQLiteCommand GetTasks = new SQLiteCommand(strQuery, sql_con);
    try
    {
        sql_con.Open();
        sql_datareader = GetTasks.ExecuteReader();
        DataTable dt = new DataTable();
        dt.Load(sql_datareader);
        ExportToExcel(dt);
        sql_con.Close();
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
}

private void ExportToExcel(DataTable dt)
{
    int colIndex = 1;
    int rowIndex = 1;
    Microsoft.Office.Interop.Excel.Application xlApp;
    Microsoft.Office.Interop.Excel.Workbook xlWorkBook;

```

```

Microsoft.Office.Interop.Excel.Worksheet xlWorkSheet;
object misValue = System.Reflection.Missing.Value;
xlApp = new Microsoft.Office.Interop.Excel.Application();
xlApp = new Microsoft.Office.Interop.Excel.Application();
xlWorkBook = xlApp.Workbooks.Add(misValue);
xlWorkSheet = (Microsoft.Office.Interop.Excel.Worksheet)xlWorkBook.Worksheets.get_Item(1);
xlWorkSheet.Name = "test1";
colIndex = 1;
foreach (DataColumn dc in dt.Columns)
{
    xlWorkSheet.Cells[1, colIndex] = dc.ColumnName;
    colIndex = colIndex + 1;
}
foreach (DataRow dr in dt.Rows)
{
    colIndex = 0;
    foreach (DataColumn dc in dt.Columns)
    {
        colIndex = colIndex + 1;
        if (dc.ColumnName != "Photo")
        {
            xlWorkSheet.Cells[rowIndex + 1, colIndex] = dr[dc.ColumnName];
        }
        else if (dr[dc.ColumnName] != DBNull.Value)
        {
            Microsoft.Office.Interop.Excel.Range oRange =
(Microsoft.Office.Interop.Excel.Range)xlWorkSheet.Cells[rowIndex, colIndex];
            float Left = (float)((double)oRange.Left);
            float Top = (float)((double)oRange.Top);
            const float ImageSize = 32;
            byte[] bitmap = (Byte[])dr[dc.ColumnName];

            using (Image image = Image.FromStream(new MemoryStream(bitmap)))
            {
                image.Save("output.jpg", ImageFormat.Jpeg); // Or Png
            }
            string assemblyPath = Path.GetDirectoryName(Assembly.GetEntryAssembly().Location);
            string FileName = Path.Combine(assemblyPath, "output.jpg");
            xlWorkSheet.Shapes.AddPicture(FileName, Microsoft.Office.Core.MsoTriState.msoFalse,
Microsoft.Office.Core.MsoTriState.msoCTrue, Left, Top, ImageSize, ImageSize);
            oRange.RowHeight = ImageSize + 2;
        }
    }
    rowIndex = rowIndex + 1;
}

xlApp.Visible = true;
System.Runtime.InteropServices.Marshal.ReleaseComObject(xlWorkSheet);
System.Runtime.InteropServices.Marshal.ReleaseComObject(xlWorkBook);
System.Runtime.InteropServices.Marshal.ReleaseComObject(xlApp);
}
private void ExportToExistingExcel(DataTable dt, int SheetIndex, string SheetName, ref
Microsoft.Office.Interop.Excel.Workbook xlWorkBook)
{
    int colIndex = 1;
    int rowIndex = 1;
    Microsoft.Office.Interop.Excel.Worksheet xlWorkSheet;
    object misValue = System.Reflection.Missing.Value;
    //xlWorkBook = xlApp.Workbooks.Add(misValue);

    xlWorkSheet =
(Microsoft.Office.Interop.Excel.Worksheet)xlWorkBook.Worksheets.get_Item(SheetIndex);
    xlWorkSheet.Name = SheetName;
    colIndex = 1;
    foreach (DataColumn dc in dt.Columns)
    {

```

```

xlWorksheet.Cells[1, colIndex] = dc.ColumnName;
colIndex = colIndex + 1;
}
foreach (DataRow dr in dt.Rows)
{
    colIndex = 0;
    foreach (DataColumn dc in dt.Columns)
    {
        colIndex = colIndex + 1;
        if (dc.ColumnName != "Photo")
        {
            xlWorksheet.Cells[rowIndex + 1, colIndex] = dr[dc.ColumnName];
        }
        else if (dr[dc.ColumnName] != DBNull.Value)
        {
            Microsoft.Office.Interop.Excel.Range oRange =
            (Microsoft.Office.Interop.Excel.Range)xlWorksheet.Cells[rowIndex, colIndex];
            float Left = (float)((double)oRange.Left);
            float Top = (float)((double)oRange.Top);
            const float ImageSize = 32;
            byte[] bitmap = (Byte[])dr[dc.ColumnName];

            using (Image image = Image.FromStream(new MemoryStream(bitmap)))
            {
                image.Save("output.jpg", ImageFormat.Jpeg); // Or Png
            }
            string assemblyPath = Path.GetDirectoryName(Assembly.GetEntryAssembly().Location);
            string FileName = Path.Combine(assemblyPath, "output.jpg");
            xlWorksheet.Shapes.AddPicture(FileName, Microsoft.Office.Core.MsoTriState.msoFalse,
            Microsoft.Office.Core.MsoTriState.msoCTrue, Left, Top, ImageSize, ImageSize);
            oRange.RowHeight = ImageSize + 2;
        }
    }
    rowIndex = rowIndex + 1;
}
}
private DataTable GetTasks()
{
    string strQuery = "Select Name As TaskName,(Select Name From Resources Where UID=(Select
    Max(ResourceUID) From Assignments Where Assignments.TaskUID=Tasks.UID And ProjectID=" +
    MyGlobals.ProjectID + ")) As
    ResourceName,Start,Finish,ActualStart,ActualFinish,FixedCost,FixedCostAccrual,PercentComplete,(Select Name
    From Project P Where P.ProjectID=Tasks.ProjectID) As ProjectName From Tasks Where ProjectID=" +
    MyGlobals.ProjectID + " ";
    if (cbResources.SelectedItem != null)
    {
        if (Extension.IsNumeric(((ComboBoxItem)(cbResources.SelectedItem)).Value))
            strQuery += " And UID in (Select TaskUID From Assignments Where ResourceUID=" +
            ((ComboBoxItem)(cbResources.SelectedItem)).Value + ")";
    }
    if (txtTaskName.Text.Trim() != "")
        strQuery += " And Name Like '%" + txtTaskName.Text + "%'";
    if (dtpStartFrom.Text.Trim() != "")
        strQuery += " And Start >= '" + dtpStartFrom.Text + "'";
    if (dtpStartTo.Text.Trim() != "")
        strQuery += " And Start <= '" + dtpStartTo.Text + "'";
    if (dtpFinishFrom.Text.Trim() != "")
        strQuery += " And Finish >= '" + dtpFinishFrom.Text + "'";
    if (dtpFinishTo.Text.Trim() != "")
        strQuery += " And Finish <= '" + dtpFinishTo.Text + "'";
    Extension.SetMainConnection(ref sql_con);
    SQLiteCommand GetTasks = new SQLiteCommand(strQuery, sql_con);
    try
    {
        sql_con.Open();
        sql_datareader = GetTasks.ExecuteReader();
    }
}

```

```

        DataTable dt = new DataTable();
        dt.Load(sql_datareader);
        sql_con.Close();
        return dt;
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
}
private DataTable GetTasksDetails()
{
    string strQuery = "Select (Select Name From Tasks Where UID=TasksLog.TaskUID And ProjectID=" +
MyGlobals.ProjectID + ") As
TaskName,Start,Finish,ActualStart,ExpectedEndDate,ActualCost,PercentWorkComplete,DateEntered,Remarks,Ph
oto From TasksLog Where ProjectID=" + MyGlobals.ProjectID + " And TaskUID in (Select UID From Tasks
Where ProjectID=" + MyGlobals.ProjectID + " ";
    if (cbResources.SelectedItem != null)
    {
        if (Extension.IsNumeric(((ComboBoxItem)cbResources.SelectedItem).Value))
            strQuery += " And UID in (Select TaskUID From Assignments Where ResourceUID=" +
((ComboBoxItem)cbResources.SelectedItem).Value + ")";
    }
    if (txtTaskName.Text.Trim() != "")
        strQuery += " And Name Like '%" + txtTaskName.Text + "%'";
    if (dtpStartFrom.Text.Trim() != "")
        strQuery += " And Start >=" + dtpStartFrom.Text + """;
    if (dtpStartTo.Text.Trim() != "")
        strQuery += " And Start <=" + dtpStartTo.Text + """;
    if (dtpFinishFrom.Text.Trim() != "")
        strQuery += " And Finish >=" + dtpFinishFrom.Text + """;
    if (dtpFinishTo.Text.Trim() != "")
        strQuery += " And Finish <=" + dtpFinishTo.Text + """;
    strQuery += ") order by TaskUID";
    Extension.SetMainConnection(ref sql_con);
    SQLiteCommand GetTasks = new SQLiteCommand(strQuery, sql_con);
    try
    {
        sql_con.Open();
        sql_datareader = GetTasks.ExecuteReader();
        DataTable dt = new DataTable();
        dt.Load(sql_datareader);
        sql_con.Close();
        return dt;
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
}
private void btnExport_Click(object sender, EventArgs e)
{
    DataTable dtTasks = new DataTable();
    dtTasks = GetTasks();
    DataTable dtTasksDetails = new DataTable();
    dtTasksDetails = GetTasksDetails();
    Microsoft.Office.Interop.Excel.Application xlApp;
    xlApp = new Microsoft.Office.Interop.Excel.Application();
    Microsoft.Office.Interop.Excel.Workbook xlWorkbook;
    xlWorkbook =
xlApp.Workbooks.Open(System.Configuration.ConfigurationSettings.AppSettings["OriginalExcelFile"].ToString(
));
    ExportToExistingExcel(dtTasks, 1, "Imported Sheet 1", ref xlWorkbook);
    ExportToExistingExcel(dtTasksDetails, 2, "Imported Detailed", ref xlWorkbook);
    xlWorkbook.SaveAs(System.Configuration.ConfigurationSettings.AppSettings["ProductionExcelFile"].ToString()
.Replace("Production",MyGlobals.ProjectName + "-" + DateTime.Now.ToString("ddMMyyyyHHmm")));
}

```

```

        xlWorkbook.Close();
        MessageBox.Show("Export Done!");
    }
}
}

```

Tasks Main

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SQLite;
using System.Xml.Linq;

namespace Constructions
{
    public partial class TasksMain : Form
    {
        public SQLiteConnection sql_con;
        public SQLiteCommand sql_cmd;
        public SQLiteDataReader sql_datareader;
        public string FilePath;
        public TasksMain()
        {
            InitializeComponent();
        }

        private void TasksMain_Load(object sender, EventArgs e)
        {
            bool IsAdministrator = Aggregates.getAggregateBoolean("Select IsAdministrator From UserLogins Where ResourceInitials=" + MyGlobals.ResourceInitials.ToString() + "");
            btnTasksSearch.Visible = IsAdministrator;
            txtXMLFilePath.Visible = IsAdministrator;
            btnSelectXMLFile.Visible = IsAdministrator;
            btnImport.Visible = IsAdministrator;
            cbOpenFinishedTasks.Visible = IsAdministrator;
            LoadTasks(false);
        }

        private void dgvTasks_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
            string TaskName = "";
            if (e.ColumnIndex == 0)
            {
                TaskName = dgvTasks.Rows[e.RowIndex].Cells[e.ColumnIndex].Value.ToString();
                int TaskUID = Int32.Parse(dgvTasks.Rows[e.RowIndex].Cells["UID"].Value.ToString());
                TasksModify TaskModify = new TasksModify(Int32.Parse(TaskUID.ToString()));
                TaskModify.Show();
                this.Hide();
            }
        }

        private void btnTasksSearch_Click(object sender, EventArgs e)
        {
            TasksSearch TaskSearch = new TasksSearch();
            TaskSearch.Show();
            this.Hide();
        }

        private void btnSelectXMLFile_Click(object sender, EventArgs e)
        {
            try

```

```

    {
        OpenFileDialog dlg = new OpenFileDialog();
        dlg.InitialDirectory = @"C:\\";
        dlg.Title = "Select Image File";
        dlg.Filter = "All Files|*.*|XML|*.xml";
        dlg.FilterIndex = 2;
        if (dlg.ShowDialog() == DialogResult.OK)
        {
            FilePath = dlg.FileName;
            txtXMLFilePath.Text = dlg.FileName;
        }
    }
}
catch
{
}
}

private void btnImport_Click(object sender, EventArgs e)
{
    int ProjectID = MyGlobals.ProjectID;
    if (cbNewProject.Checked)
    {
        ProjectID = LoadAndImportXMLProjectData();
    }
    LoadAndImportXMLTasksData(ProjectID);
    LoadAndImportXMLResourcesData();
    LoadAndImportXMLAssignmentsData(ProjectID);
    MessageBox.Show("Imports Done!");
}

public void LoadAndImportXMLTasksData(int ProjectID)
{
    // Query the data and write out a subset of contacts
    XDocument xml = XDocument.Load(FilePath);
    XNamespace ns2 =
XNamespace.Get(System.Configuration.ConfigurationSettings.AppSettings["XNamespaceName"].ToString());
    var Tasks = from data in xml.Root.Descendants(ns2 + "Task")
    select new
    {
        UID = (String)data.Element(ns2 + "UID"),
        ID = (String)data.Element(ns2 + "ID"),
        OutlineNumber = (String)data.Element(ns2 + "OutlineNumber"),
        OutlineLevel = (String)data.Element(ns2 + "OutlineLevel"),
        Priority = (String)data.Element(ns2 + "Priority"),
        Summary = (String)data.Element(ns2 + "Summary"),
        IsSubproject = (String)data.Element(ns2 + "IsSubproject"),
        ExternalTask = (String)data.Element(ns2 + "ExternalTask"),
        Name = (String)data.Element(ns2 + "Name"),
        Type = (String)data.Element(ns2 + "Type"),
        IsNull = (String)data.Element(ns2 + "IsNull"),
        WBS = (String)data.Element(ns2 + "WBS"),
        FixedCostAccrual = (String)data.Element(ns2 + "FixedCostAccrual"),
        ConstraintType = (String)data.Element(ns2 + "ConstraintType"),
        Start = (String)data.Element(ns2 + "Start"),
        Finish = (String)data.Element(ns2 + "Finish"),
        Duration = (String)data.Element(ns2 + "Duration"),
        DurationFormat = (String)data.Element(ns2 + "DurationFormat"),
        Work = (String)data.Element(ns2 + "Work"),
        EffortDriven = (String)data.Element(ns2 + "EffortDriven"),
        Milestone = (String)data.Element(ns2 + "Milestone"),
        FixedCost = (String)data.Element(ns2 + "FixedCost"),
        PercentComplete = (String)data.Element(ns2 + "PercentComplete"),
        ActualStart = (String)data.Element(ns2 + "ActualStart"),
        ActualFinish = (String)data.Element(ns2 + "ActualFinish"),
        ActualDuration = (String)data.Element(ns2 + "ActualDuration"),
        ActualWork = (String)data.Element(ns2 + "ActualWork"),
        RemainingDuration = (String)data.Element(ns2 + "RemainingDuration"),
    }
}

```

```

CalendarUID = (String)data.Element(ns2 + "CalendarUID"),
ConstraintDate = (String)data.Element(ns2 + "ConstraintDate"),
IgnoreResourceCalendar = (String)data.Element(ns2 + "IgnoreResourceCalendar"),
PhysicalPercentComplete = (String)data.Element(ns2 + "PhysicalPercentComplete"),
};
foreach (var Task in Tasks)
{
    if (string.IsNullOrEmpty(Task.Name) == false)
    {
        int TaskUID = (Int32)Aggregates.getAggregate("Select UID From Tasks Where UID=" + Task.UID +
" Or Name=" + Task.Name + "");
        if (TaskUID == 0)
        {
            ImportTasks("Insert", ProjectID, Task.UID, Task.ID, Task.OutlineNumber, Task.OutlineLevel,
Task.Priority, Task.Summary, Task.IsSubproject, Task.ExternalTask, Task.Name, Task.Type, Task.IsNull,
Task.WBS, Task.FixedCostAccrual, Task.ConstraintType, Task.Start, Task.Finish, Task.Duration,
Task.DurationFormat, Task.Work, Task.EffortDriven, Task.Milestone, Task.FixedCost, Task.PercentComplete,
Task.ActualStart, Task.ActualFinish, Task.ActualDuration, Task.ActualWork, Task.RemainingDuration,
Task.CalendarUID, Task.ConstraintDate, Task.IgnoreResourceCalendar, Task.PhysicalPercentComplete);
        }
        else
        {
            ImportTasks("Update", ProjectID, Task.UID, Task.ID, Task.OutlineNumber, Task.OutlineLevel,
Task.Priority, Task.Summary, Task.IsSubproject, Task.ExternalTask, Task.Name, Task.Type, Task.IsNull,
Task.WBS, Task.FixedCostAccrual, Task.ConstraintType, Task.Start, Task.Finish, Task.Duration,
Task.DurationFormat, Task.Work, Task.EffortDriven, Task.Milestone, Task.FixedCost, Task.PercentComplete,
Task.ActualStart, Task.ActualFinish, Task.ActualDuration, Task.ActualWork, Task.RemainingDuration,
Task.CalendarUID, Task.ConstraintDate, Task.IgnoreResourceCalendar, Task.PhysicalPercentComplete);
        }
    }
}
}
public void ImportTasks(String InsertOrUpdate, int ProjectID, String UID, String ID, String OutlineNumber,
String OutlineLevel, String Priority, String Summary, String IsSubproject, String ExternalTask, String Name,
String Type, String IsNull, String WBS, String FixedCostAccrual, String ConstraintType, String Start, String
Finish, String Duration, String DurationFormat, String Work, String EffortDriven, String Milestone, String
FixedCost, String PercentComplete, String ActualStart, String ActualFinish, String ActualDuration, String
ActualWork, String RemainingDuration, String CalendarUID, String ConstraintDate, String
IgnoreResourceCalendar, String PhysicalPercentComplete)
{
    Extension.SetMainConnection(ref sql_con);
    SQLiteCommand ImportTasksSQL;
    if (InsertOrUpdate == "Insert" || cbNewProject.Checked)
        ImportTasksSQL = new SQLiteCommand("INSERT INTO Tasks
(ProjectID,UID,ID,OutlineNumber,OutlineLevel,Priority,Summary,IsSubproject,ExternalTask,Name,Type,[IsNull
],WBS,FixedCostAccrual,ConstraintType,Start,Finish,Duration,DurationFormat,Work,EffortDriven,Milestone,Fix
edCost,PercentComplete,ActualStart,ActualFinish,ActualDuration,ActualWork,RemainingDuration,CalendarUID,
ConstraintDate,IgnoreResourceCalendar,PhysicalPercentComplete) VALUES
(@ProjectID,@UID,@ID,@OutlineNumber,@OutlineLevel,@Priority,@Summary,@IsSubproject,@ExternalTas
k,@Name,@Type,@IsNull,@WBS,@FixedCostAccrual,@ConstraintType,@Start,@Finish,@Duration
nFormat,@Work,@EffortDriven,@Milestone,@FixedCost,@PercentComplete,@ActualStart,@ActualFinish,@Ac
tualDuration,@ActualWork,@RemainingDuration,@CalendarUID,@ConstraintDate,@IgnoreResourceCalendar,
@PhysicalPercentComplete)", sql_con);
    else
        ImportTasksSQL = new SQLiteCommand("UPDATE Tasks Set ProjectID=@ProjectID,
ID=@ID,OutlineNumber=@OutlineNumber,OutlineLevel=@OutlineLevel,Priority=@Priority,Summary=@Sum
mary,IsSubproject=@IsSubproject,ExternalTask=@ExternalTask,Name=@Name,Type=@Type,[IsNull]=@IsNull
,WBS=@WBS,FixedCostAccrual=@FixedCostAccrual,ConstraintType=@ConstraintType,Start=@Start,Finish=
@Finish,Duration=@Duration,DurationFormat=@DurationFormat,Work=@Work,EffortDriven=@EffortDriven,
Milestone=@Milestone,FixedCost=@FixedCost,PercentComplete=@PercentComplete,ActualStart=@ActualStart,
ActualFinish=@ActualFinish,ActualDuration=@ActualDuration,ActualWork=@ActualWork,RemainingDuration
=@RemainingDuration,CalendarUID=@CalendarUID,ConstraintDate=@ConstraintDate,IgnoreResourceCalendar
=@IgnoreResourceCalendar,PhysicalPercentComplete=@PhysicalPercentComplete Where UID=@UID",
sql_con);
}

```



```

ImportTasksSQL.Parameters.Add(new SQLiteParameter("@ProjectID", ProjectID));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@UID", UID));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@ID", ID));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@OutlineNumber", OutlineNumber));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@OutlineLevel", OutlineLevel));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@Priority", Priority));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@Summary", Summary));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@IsSubproject", IsSubproject));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@ExternalTask", ExternalTask));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@Name", Name));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@Type", Type));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@IsNull", IsNull));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@WBS", WBS));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@FixedCostAccrual", FixedCostAccrual));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@ConstraintType", ConstraintType));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@Start", Start));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@Finish", Finish));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@Duration", Duration));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@DurationFormat", DurationFormat));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@Work", Work));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@EffortDriven", EffortDriven));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@Milestone", Milestone));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@FixedCost", FixedCost));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@PercentComplete", PercentComplete));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@ActualStart", ActualStart));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@ActualFinish", ActualFinish));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@ActualDuration", ActualDuration));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@ActualWork", ActualWork));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@RemainingDuration", RemainingDuration));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@CalendarUID", CalendarUID));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@ConstraintDate", ConstraintDate));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@IgnoreResourceCalendar",
IgnoreResourceCalendar));
ImportTasksSQL.Parameters.Add(new SQLiteParameter("@PhysicalPercentComplete",
PhysicalPercentComplete));
try
{
    sql_con.Open();
    ImportTasksSQL.ExecuteNonQuery();
    sql_con.Close();
}
catch (Exception ex)
{
    throw new Exception(ex.Message);
}
}
public int LoadAndImportXMLProjectData()
{
    int ProjectID = 0;
    // Query the data and write out a subset of contacts
    XDocument xml = XDocument.Load(FilePath);
    XNamespace ns2 =
XNamespace.Get(System.Configuration.ConfigurationSettings.AppSettings["XNamespaceName"].ToString());
    var Projects = from data in xml.Descendants(ns2 + "Project")
select new
{
    Name = (String)data.Element(ns2 + "Name"),
    Title = (String)data.Element(ns2 + "Title"),
    ScheduleFromStart = (String)data.Element(ns2 + "ScheduleFromStart"),
    StartDate = (String)data.Element(ns2 + "StartDate"),
};
foreach (var Project in Projects)
{
    if (string.IsNullOrEmpty(Project.Name) == false)
    {

```

```

        ProjectID=ImportProject("Insert", 0, Project.Name, Project.Title, Project.ScheduleFromStart,
Project.StartDate);
    }
}
return ProjectID;
}
public int ImportProject(String InsertOrUpdate, int ProjectID, String Name, String Title, String
ScheduleFromStart, String StartDate)
{
    Extension.SetMainConnection(ref sql_con);
    SQLiteCommand ImportProjectsSQL;
    if (InsertOrUpdate == "Insert")
        ImportProjectsSQL = new SQLiteCommand("INSERT INTO Project
(Name,Title,ScheduleFromStart,StartDate) VALUES (@Name,@ Title,@ ScheduleFromStart,@StartDate)",
sql_con);
    else
        ImportProjectsSQL = new SQLiteCommand("UPDATE Project Set
Name=@Name,Title=@ Title,ScheduleFromStart=@ ScheduleFromStart,StartDate=@StartDate Where
UID=@ UID", sql_con);

    ImportProjectsSQL.Parameters.Add(new SQLiteParameter("@ProjectID", ProjectID));
    ImportProjectsSQL.Parameters.Add(new SQLiteParameter("@Name", Name));
    ImportProjectsSQL.Parameters.Add(new SQLiteParameter("@Title", Title));
    ImportProjectsSQL.Parameters.Add(new SQLiteParameter("@ScheduleFromStart", ScheduleFromStart));
    ImportProjectsSQL.Parameters.Add(new SQLiteParameter("@StartDate", StartDate));
    try
    {
        sql_con.Open();
        ImportProjectsSQL.ExecuteNonQuery();
        if (ProjectID == 0)
        {
            ImportProjectsSQL = new SQLiteCommand("SELECT last_insert_rowid()",sql_con);
            ProjectID = Int32.Parse(ImportProjectsSQL.ExecuteScalar().ToString());
        }
        sql_con.Close();
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
    return ProjectID;
}
public void LoadAndImportXMLResourcesData()
{
    // Query the data and write out a subset of contacts
    XDocument xml = XDocument.Load(FilePath);
    XNamespace ns2 =
XNamespace.Get(System.Configuration.ConfigurationSettings.AppSettings["XNamespaceName"].ToString());
    var Resources = from data in xml.Root.Descendants(ns2 + "Resource")
select new
    {
        UID = (String)data.Element(ns2 + "UID"),
        ID = (String)data.Element(ns2 + "ID"),
        Name = (String)data.Element(ns2 + "Name"),
        Type = (String)data.Element(ns2 + "Type"),
        Initials = (String)data.Element(ns2 + "Initials"),
        Group = (String)data.Element(ns2 + "Group"),
        EmailAddress = (String)data.Element(ns2 + "EmailAddress"),
        MaxUnits = (String)data.Element(ns2 + "MaxUnits"),
        CalendarUID = (String)data.Element(ns2 + "CalendarUID"),
        IsEnterprise = (String)data.Element(ns2 + "IsEnterprise"),
    };
    foreach (var Resource in Resources)
    {
        if (string.IsNullOrEmpty(Resource.Name) == false)
        {

```

```

        int ResourceUID = (Int32)Aggregates.getAggregate("Select UID From Resources Where UID=" +
Resource.UID + " Or Name=" + Resource.Name + "");
        if (ResourceUID == 0)
        {
            ImportResources("Insert", Resource.UID, Resource.ID, Resource.Name, Resource.Type,
Resource.Initials, Resource.Group, Resource.EmailAddress, Resource.MaxUnits, Resource.CalendarUID,
Resource.IsEnterprise);
        }
        else
        {
            ImportResources("Update", Resource.UID, Resource.ID, Resource.Name, Resource.Type,
Resource.Initials, Resource.Group, Resource.EmailAddress, Resource.MaxUnits, Resource.CalendarUID,
Resource.IsEnterprise);
        }
    }
}

public void ImportResources(String InsertOrUpdate, String UID, String ID, String Name, String Type, String
Initials, String Group, String EmailAddress, String MaxUnits, String CalendarUID, String IsEnterprise)
{
    Extension.SetMainConnection(ref sql_con);
    SQLiteCommand ImportResourcesSQL;
    if (InsertOrUpdate == "Insert")
        ImportResourcesSQL = new SQLiteCommand("INSERT INTO Resources
(UID,ID,Name,Type,Initials,[Group],EmailAddress,MaxUnits,CalendarUID,IsEnterprise) VALUES
(@UID,@ID,@Name,@Type,@Initials,@Group,@EmailAddress,@MaxUnits,@CalendarUID,@IsEnterprise)",
sql_con);
    else
        ImportResourcesSQL = new SQLiteCommand("UPDATE Resources Set
UID=@UID,ID=@ID,Name=@Name,Type=@Type,Initials=@Initials,[Group]=@Group,EmailAddress=@Email
Address,MaxUnits=@MaxUnits,CalendarUID=@CalendarUID,IsEnterprise=@IsEnterprise Where UID=@UID",
sql_con);

    ImportResourcesSQL.Parameters.Add(new SQLiteParameter("@UID", UID));
    ImportResourcesSQL.Parameters.Add(new SQLiteParameter("@ID", ID));
    ImportResourcesSQL.Parameters.Add(new SQLiteParameter("@Name", Name));
    ImportResourcesSQL.Parameters.Add(new SQLiteParameter("@Type", Type));
    ImportResourcesSQL.Parameters.Add(new SQLiteParameter("@Initials", Initials));
    ImportResourcesSQL.Parameters.Add(new SQLiteParameter("@Group", Group));
    ImportResourcesSQL.Parameters.Add(new SQLiteParameter("@EmailAddress", EmailAddress));
    ImportResourcesSQL.Parameters.Add(new SQLiteParameter("@MaxUnits", MaxUnits));
    ImportResourcesSQL.Parameters.Add(new SQLiteParameter("@CalendarUID", CalendarUID));
    ImportResourcesSQL.Parameters.Add(new SQLiteParameter("@IsEnterprise", IsEnterprise));

    try
    {
        sql_con.Open();
        ImportResourcesSQL.ExecuteNonQuery();
        sql_con.Close();
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
}

public void LoadAndImportXMLAssignmentsData(int ProjectID)
{
    // Query the data and write out a subset of contacts
    XDocument xml = XDocument.Load(FilePath);
    XNamespace ns2 =
XNamespace.Get(System.Configuration.ConfigurationSettings.AppSettings["XNamespaceName"].ToString());
    var Assignments = from data in xml.Root.Descendants(ns2 + "Assignment")
select new
{
    UID = (String)data.Element(ns2 + "UID"),
    TaskUID = (String)data.Element(ns2 + "TaskUID"),

```

```

ResourceUID = (String)data.Element(ns2 + "ResourceUID"),
PercentWorkComplete = (String)data.Element(ns2 + "PercentWorkComplete"),
ActualCost = (String)data.Element(ns2 + "ActualCost"),
ActualOvertimeCost = (String)data.Element(ns2 + "ActualOvertimeCost"),
ActualOvertimeWork = (String)data.Element(ns2 + "ActualOvertimeWork"),
ActualStart = (String)data.Element(ns2 + "ActualStart"),
ActualWork = (String)data.Element(ns2 + "ActualWork"),
CostRateTable = (String)data.Element(ns2 + "CostRateTable"),
Delay = (String)data.Element(ns2 + "Delay"),
Finish = (String)data.Element(ns2 + "Finish"),
OvertimeWork = (String)data.Element(ns2 + "OvertimeWork"),
RegularWork = (String)data.Element(ns2 + "RegularWork"),
RemainingCost = (String)data.Element(ns2 + "RemainingCost"),
RemainingWork = (String)data.Element(ns2 + "RemainingWork"),
Start = (String)data.Element(ns2 + "Start"),
Units = (String)data.Element(ns2 + "Units"),
Work = (String)data.Element(ns2 + "Work"),
WorkContour = (String)data.Element(ns2 + "WorkContour"),
};
foreach (var Assignment in Assignments)
{
    if (Extension.IsNumeric(Assignment.UID))
    {
        if (Int32.Parse(Assignment.UID) > 0)
        {
            int AssignmentUID = (Int32)Aggregates.getAggregate("Select UID From Assignments Where
UID=" + Assignment.UID + " Or (TaskUID=" + Assignment.TaskUID + " And ResourceUID=" +
Assignment.ResourceUID + ")");
            if (AssignmentUID == 0)
            {
                ImportAssignments("Insert", ProjectID, Assignment.UID, Assignment.TaskUID,
Assignment.ResourceUID, Assignment.PercentWorkComplete, Assignment.ActualCost,
Assignment.ActualOvertimeCost, Assignment.ActualOvertimeWork, Assignment.ActualStart,
Assignment.ActualWork, Assignment.CostRateTable, Assignment.Delay, Assignment.Finish,
Assignment.OvertimeWork, Assignment.RegularWork, Assignment.RemainingCost,
Assignment.RemainingWork, Assignment.Start, Assignment.Units, Assignment.Work,
Assignment.WorkContour);
            }
            else
            {
                ImportAssignments("Update", ProjectID, Assignment.UID, Assignment.TaskUID,
Assignment.ResourceUID, Assignment.PercentWorkComplete, Assignment.ActualCost,
Assignment.ActualOvertimeCost, Assignment.ActualOvertimeWork, Assignment.ActualStart,
Assignment.ActualWork, Assignment.CostRateTable, Assignment.Delay, Assignment.Finish,
Assignment.OvertimeWork, Assignment.RegularWork, Assignment.RemainingCost,
Assignment.RemainingWork, Assignment.Start, Assignment.Units, Assignment.Work,
Assignment.WorkContour);
            }
        }
    }
}

public void ImportAssignments(String InsertOrUpdate, int ProjectID, String UID, String TaskUID, String
ResourceUID, String PercentWorkComplete, String ActualCost, String ActualOvertimeCost, String
ActualOvertimeWork, String ActualStart, String ActualWork, String CostRateTable, String Delay, String Finish,
String OvertimeWork, String RegularWork, String RemainingCost, String RemainingWork, String Start, String
Units, String Work, String WorkContour)
{
    Extension.SetMainConnection(ref sql_con);
    SQLiteCommand ImportAssignmentsSQL;
    if (InsertOrUpdate == "Insert" || cbNewProject.Checked)
        ImportAssignmentsSQL = new SQLiteCommand("INSERT INTO Assignments
(ProjectID,UID,TaskUID,ResourceUID,PercentWorkComplete,ActualCost,ActualOvertimeCost,ActualOvertime
Work,ActualStart,ActualWork,CostRateTable,Delay,Finish,OvertimeWork,RegularWork,RemainingCost,Remaini
ngWork,Start,Units,Work,WorkContour) VALUES
(@ProjectID,@UID,@TaskUID,@ResourceUID,@PercentWorkComplete,@ActualCost,@ActualOvertimeCost,

```

```

@ActualOvertimeWork,@ActualStart,@ActualWork,@CostRateTable,@Delay,@Finish,@OvertimeWork,@Reg
ularWork,@RemainingCost,@RemainingWork,@Start,@Units,@Work,@WorkContour)", sql_con);
    else
        ImportAssignmentsSQL = new SQLiteCommand("UPDATE Assignments Set
ProjectID=@ProjectID,UID=@UID,TaskUID=@TaskUID,ResourceUID=@ResourceUID,PercentWorkComplete
=@PercentWorkComplete,ActualCost=@ActualCost,ActualOvertimeCost=@ActualOvertimeCost,ActualOvertim
eWork=@ActualOvertimeWork,ActualStart=@ActualStart,ActualWork=@ActualWork,CostRateTable=@CostRa
teTable,Delay=@Delay,Finish=@Finish,OvertimeWork=@OvertimeWork,RegularWork=@RegularWork,Remain
ingCost=@RemainingCost,RemainingWork=@RemainingWork,Start=@Start,Units=@Units,Work=@Work,Wor
kContour=@WorkContour Where UID=@UID", sql_con);

        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@ProjectID", ProjectID));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@UID", UID));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@TaskUID", TaskUID));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@ResourceUID", ResourceUID));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@PercentWorkComplete",
PercentWorkComplete));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@ActualCost", ActualCost));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@ActualOvertimeCost",
ActualOvertimeCost));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@ActualOvertimeWork",
ActualOvertimeWork));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@ActualStart", ActualStart));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@ActualWork", ActualWork));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@CostRateTable", CostRateTable));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@Delay", Delay));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@Finish", Finish));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@OvertimeWork", OvertimeWork));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@RegularWork", RegularWork));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@RemainingCost", RemainingCost));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@RemainingWork", RemainingWork));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@Start", Start));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@Units", Units));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@Work", Work));
        ImportAssignmentsSQL.Parameters.Add(new SQLiteParameter("@WorkContour", WorkContour));

    try
    {
        sql_con.Open();
        ImportAssignmentsSQL.ExecuteNonQuery();
        sql_con.Close();
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
}

private void cbOpenFinishedTasks_CheckedChanged(object sender, EventArgs e)
{
    LoadTasks(cbOpenFinishedTasks.Checked);
}

private void LoadTasks(bool IsCpmpleted)
{
    Extension.SetMainConnection(ref sql_con);
    string strCommand = "Select ID,UID,Name As TaskName,(Select Name From Resources Where
UID=(Select Max(ResourceUID) From Assignments Where Assignments.TaskUID=Tasks.UID And ProjectID="
+ MyGlobals.ProjectID + ")) As
ResourceName,Start,Finish,ActualStart,ActualFinish,FixedCost,FixedCostAccrual,PercentComplete,(Select
max(DateEntered) From TasksLog Where TasksLog.TaskUID=Tasks.UID) as LastUpdate From Tasks Where
ProjectID=" + MyGlobals.ProjectID.ToString() + " ";
    if (IsCpmpleted)
    {
        strCommand += " And PercentComplete=100 And Tasks.UID IN (Select TaskUID From
Assignments) ";
    }
}

```

```

else
{
    strCommand += " And PercentComplete<100 And date('now') >= Start And Tasks.UID IN (Select
TaskUID From Assignments Where ResourceUID=" + MyGlobals.ResourceUID + ") ";
}
SQLiteCommand GetTasks = new SQLiteCommand(strCommand , sql_con);
try
{
    sql_con.Open();
    sql_datareader = GetTasks.ExecuteReader();
    DataTable dt = new DataTable();
    dt.Clear();
    dt.Columns.Add("UID");
    dt.Columns.Add("TaskName");
    dt.Columns.Add("ResourceName");
    dt.Columns.Add("Start");
    dt.Columns.Add("Finish");
    dt.Columns.Add("ActualStart");
    dt.Columns.Add("ActualFinish");
    dt.Columns.Add("FixedCost");
    dt.Columns.Add("FixedCostAccrual");
    dt.Columns.Add("PercentComplete");
    dt.Columns.Add("LastUpdate");
    while (sql_datareader.Read() // Read() returns true if there is still a result line to read
    {
        DataRow _ravi = dt.NewRow();
        _ravi["UID"] = sql_datareader["UID"];
        _ravi["TaskName"] = sql_datareader["TaskName"];
        _ravi["ResourceName"] = sql_datareader["ResourceName"];
        _ravi["Start"] = sql_datareader["Start"];
        _ravi["Finish"] = sql_datareader["Finish"];
        _ravi["ActualStart"] = sql_datareader["ActualStart"];
        _ravi["ActualFinish"] = sql_datareader["ActualFinish"];
        _ravi["FixedCost"] = sql_datareader["FixedCost"];
        _ravi["FixedCostAccrual"] = sql_datareader["FixedCostAccrual"];
        _ravi["PercentComplete"] = sql_datareader["PercentComplete"];
        _ravi["LastUpdate"] = sql_datareader["LastUpdate"];
        dt.Rows.Add(_ravi);
    }
    dgvTasks.AutoGenerateColumns = false;
    dgvTasks.DataSource = dt;
    sql_con.Close();
}
catch (Exception ex)
{
    throw new Exception(ex.Message);
}
}

private void btnMainPage_Click(object sender, EventArgs e)
{
    TasksMain TaskMain = new TasksMain();
    TaskMain.Show();
    this.Hide();
}

private void btnLogOff_Click(object sender, EventArgs e)
{
    Login Login = new Login();
    Login.Show();
    this.Hide();
}

private void button1_Click(object sender, EventArgs e)
{
    ProjectDelete ProjectDelete = new ProjectDelete();

```

```

        ProjectDelete.Show();
        this.Hide();
    }
}
}

```

Tasks Modify

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SQLite;
using System.IO;

namespace Constructions
{
    public partial class TasksModify : Form
    {
        //private DBHelper helper = null;
        private string fileLocation = string.Empty;
        private bool isSuccess = false;
        private int maxImageSize = 2097152;

        public SQLiteConnection sql_con;
        public SQLiteCommand sql_cmd;
        SQLiteDataReader sql_datareader;
        public string FilePath;
        public int TaskUID;
        public int ResourceUID;
        public int AssignmentUID;
        public TasksModify(int iTaskUID)
        {
            InitializeComponent();
            TaskUID = iTaskUID;
            dtpStart.Format = DateTimePickerFormat.Custom;
            dtpStart.CustomFormat = "yyyy-MM-ddTHH:mm:ss";
            dtpFinish.Format = DateTimePickerFormat.Custom;
            dtpFinish.CustomFormat = "yyyy-MM-ddTHH:mm:ss";
            dtpActualStart.Format = DateTimePickerFormat.Custom;
            dtpActualStart.CustomFormat = "yyyy-MM-ddTHH:mm:ss";
            dtpExpectedEndDate.Format = DateTimePickerFormat.Custom;
            dtpExpectedEndDate.CustomFormat = "yyyy-MM-ddTHH:mm:ss";
        }

        private void TasksModify_Load(object sender, EventArgs e)
        {
            Extension.SetMainConnection(ref sql_con);
            SQLiteCommand GetTasks = new SQLiteCommand("Select ID,UID,Name As
TaskName,Start,Finish,ActualStart,ActualFinish,FixedCost,FixedCostAccrual,PercentComplete,(Select UID From
Assignments Where Assignments.TaskUID=Tasks.UID) As AssignmentUID From Tasks Where
Tasks.ProjectID=@ProjectID And Tasks.UID=@TaskUID", sql_con);
            GetTasks.Parameters.Add(new SQLiteParameter("@TaskUID", TaskUID));
            GetTasks.Parameters.Add(new SQLiteParameter("@ProjectID", MyGlobals.ProjectID));
            try
            {
                sql_con.Open();
                sql_datareader = GetTasks.ExecuteReader();
                while (sql_datareader.Read()) // Read() returns true if there is still a result line to read
                {
                    AssignmentUID = Int32.Parse(sql_datareader["AssignmentUID"].ToString());
                }
            }
        }
    }
}

```

```

        lblTaskName.Text = sql_datareader["TaskName"].ToString();
        dtpStart.Text = sql_datareader["Start"].ToString();
        dtpFinish.Text = sql_datareader["Finish"].ToString();
        dtpActualStart.Text = sql_datareader["ActualStart"].ToString();
        dtpExpectedEndDate.Text = sql_datareader["ActualFinish"].ToString();
        txtActualCost.Text = sql_datareader["FixedCostAccrual"].ToString();
        txtBudgetedCost.Text = sql_datareader["FixedCost"].ToString();
        txtPercentWorkComplete.Text = sql_datareader["PercentComplete"].ToString();
    }
    sql_con.Close();
}
catch (Exception ex)
{
    throw new Exception(ex.Message);
}
}

private void btnSave_Click(object sender, EventArgs e)
{
    Extension.SetMainConnection(ref sql_con);
    SQLiteCommand UpdateAssignments = new SQLiteCommand("Update Assignments Set
TaskUID=@TaskUID,ResourceUID=@ResourceUID,Start=@Start,Finish=@Finish,ActualStart=@ActualStart,Ex
pectedEndDate=@ExpectedEndDate,ActualCost=@ActualCost,PercentWorkComplete=@PercentWorkComplete
Where ProjectID=@ProjectID And UID=@AssignmentUID", sql_con);
    UpdateAssignments.Parameters.Add(new SQLiteParameter("@AssignmentUID", AssignmentUID));
    UpdateAssignments.Parameters.Add(new SQLiteParameter("@ProjectID", MyGlobals.ProjectID));
    UpdateAssignments.Parameters.Add(new SQLiteParameter("@TaskUID", TaskUID));
    UpdateAssignments.Parameters.Add(new SQLiteParameter("@ResourceUID", ResourceUID));
    UpdateAssignments.Parameters.Add(new SQLiteParameter("@Start", dtpStart.Text));
    UpdateAssignments.Parameters.Add(new SQLiteParameter("@Finish", dtpFinish.Text));
    UpdateAssignments.Parameters.Add(new SQLiteParameter("@ActualStart", dtpActualStart.Text));
    UpdateAssignments.Parameters.Add(new SQLiteParameter("@ExpectedEndDate",
dtpExpectedEndDate.Text));
    UpdateAssignments.Parameters.Add(new SQLiteParameter("@ActualCost", txtActualCost.Text));
    UpdateAssignments.Parameters.Add(new SQLiteParameter("@PercentWorkComplete",
txtPercentWorkComplete.Text));

    SQLiteCommand UpdateTasks = new SQLiteCommand("Update Tasks Set
Start=@Start,Finish=@Finish,ActualStart=@ActualStart,ActualFinish=@ExpectedEndDate,FixedCostAccrual=@
ActualCost,PercentComplete=@PercentWorkComplete Where ProjectID=@ProjectID And UID=@TaskUID",
sql_con);
    UpdateTasks.Parameters.Add(new SQLiteParameter("@TaskUID", TaskUID));
    UpdateTasks.Parameters.Add(new SQLiteParameter("@ResourceUID", ResourceUID));
    UpdateTasks.Parameters.Add(new SQLiteParameter("@ProjectID", MyGlobals.ProjectID));
    UpdateTasks.Parameters.Add(new SQLiteParameter("@Start", dtpStart.Text));
    UpdateTasks.Parameters.Add(new SQLiteParameter("@Finish", dtpFinish.Text));
    UpdateTasks.Parameters.Add(new SQLiteParameter("@ActualStart", dtpActualStart.Text));
    UpdateTasks.Parameters.Add(new SQLiteParameter("@ExpectedEndDate", dtpExpectedEndDate.Text));
    UpdateTasks.Parameters.Add(new SQLiteParameter("@ActualCost", txtActualCost.Text));
    UpdateTasks.Parameters.Add(new SQLiteParameter("@PercentWorkComplete",
txtPercentWorkComplete.Text));

    SQLiteCommand InsertTasksLog = new SQLiteCommand("INSERT INTO TasksLog
(TaskUID,ResourceUID,ProjectID,Start,Finish,ActualStart,ExpectedEndDate,ActualCost,PercentWorkComplete,
Remarks,DateEntered,Photo) VALUES
(@TaskUID,@ResourceUID,@ProjectID,@Start,@Finish,@ActualStart,@ExpectedEndDate,@ActualCost,@Per
centWorkComplete,@Remarks,@DateEntered,@Photo)", sql_con);
    InsertTasksLog.Parameters.Add(new SQLiteParameter("@TaskUID", TaskUID));
    InsertTasksLog.Parameters.Add(new SQLiteParameter("@ResourceUID", ResourceUID));
    InsertTasksLog.Parameters.Add(new SQLiteParameter("@ProjectID", MyGlobals.ProjectID));
    InsertTasksLog.Parameters.Add(new SQLiteParameter("@Start", dtpStart.Text));
    InsertTasksLog.Parameters.Add(new SQLiteParameter("@Finish", dtpFinish.Text));
    InsertTasksLog.Parameters.Add(new SQLiteParameter("@ActualStart", dtpActualStart.Text));
    InsertTasksLog.Parameters.Add(new SQLiteParameter("@ExpectedEndDate",
dtpExpectedEndDate.Text));

```



```

        InsertTasksLog.Parameters.Add(new SQLiteParameter("@ActualCost", txtActualCost.Text));
        InsertTasksLog.Parameters.Add(new SQLiteParameter("@PercentWorkComplete",
txtPercentWorkComplete.Text));
        InsertTasksLog.Parameters.Add(new SQLiteParameter("@Remarks", txtRemarks.Text));
        InsertTasksLog.Parameters.Add(new SQLiteParameter("@DateEntered", DateTime.Now));
        if (pictureBox1.Image != null)
        {
            MemoryStream ms = new MemoryStream();
            pictureBox1.Image.Save(ms, pictureBox1.Image.RawFormat);
            byte[] a = ms.GetBuffer();
            ms.Close();
            InsertTasksLog.Parameters.Add(new SQLiteParameter("@Photo", a));
        }
        else
        {
            InsertTasksLog.Parameters.Add(new SQLiteParameter("@Photo", null));
        }
        try
        {
            sql_con.Open();
            InsertTasksLog.ExecuteNonQuery();
            UpdateTasks.ExecuteNonQuery();
            UpdateAssignments.ExecuteNonQuery();
            sql_con.Close();
            TasksMain TaskMain = new TasksMain();
            TaskMain.Show();
            this.Hide();
        }
        catch (Exception ex)
        {
            throw new Exception(ex.Message);
        }
    }

    private void gbTaskModify_Enter(object sender, EventArgs e)
    {
    }

    private string FileLocation
    {
        get { return fileLocation; }
        set
        {
            fileLocation = value;
        }
    }

    public Boolean GetSuccess()
    {
        return isSuccess;
    }

    private void btnSelectPhoto_Click(object sender, EventArgs e)
    {
        try
        {
            OpenFileDialog dlg = new OpenFileDialog();
            dlg.InitialDirectory = @"C:\\";
            dlg.Title = "Select Image File";
            dlg.Filter = "All Files|*.*|JPEGs|.jpg|BitMaps|.bmp|GIFs|.gif";
            dlg.FilterIndex = 2;
            if (dlg.ShowDialog() == DialogResult.OK)
            {
                pictureBox1.Image = Image.FromFile(dlg.FileName);
                pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
                pictureBox1.BorderStyle = BorderStyle.Fixed3D;
            }
        }
    }

```

```

        FilePath = dlg.FileName;
    }
}
catch
{
}
}

private void btnMainPage_Click(object sender, EventArgs e)
{
    TasksMain TaskMain = new TasksMain();
    TaskMain.Show();
    this.Hide();
}
}
}

```

Tasks Log

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SQLite;
using System.IO;
using DGVPrinterHelper;

namespace Constructions
{
    public partial class TasksLog : Form
    {
        public SQLiteConnection sql_con;
        public SQLiteCommand sql_cmd;
        SQLiteDataReader sql_datareader;
        public int TaskUID;
        public TasksLog(int iTaskUID)
        {
            InitializeComponent();
            TaskUID = iTaskUID;
            string strQuery = "Select TasksLogID,(Select Name From Tasks Where ProjectID=@ProjectID And
UID=@TaskUID) As
TaskName,Start,Finish,ActualStart,ExpectedEndDate,ActualCost,PercentWorkComplete,Remarks,Photo,DateEnte
red From TasksLog Where ProjectID=@ProjectID And TaskUID=@TaskUID ";

            Extension.SetMainConnection(ref sql_con);
            SQLiteCommand GetTasks = new SQLiteCommand(strQuery, sql_con);
            GetTasks.Parameters.Add(new SQLiteParameter("@TaskUID", TaskUID));
            GetTasks.Parameters.Add(new SQLiteParameter("@ProjectID", MyGlobals.ProjectID));
            try
            {
                sql_con.Open();
                sql_datareader = GetTasks.ExecuteReader();

                ////////////
                DataGridViewTextBoxColumn TasksLogID = new DataGridViewTextBoxColumn();
                TasksLogID.HeaderText = "TasksLogID";
                TasksLogID.Name = "TasksLogID";
                DataGridViewTextBoxColumn TaskName = new DataGridViewTextBoxColumn();
                TaskName.HeaderText = "TaskName";
                TaskName.Name = "TaskName";
                DataGridViewTextBoxColumn DateEntered = new DataGridViewTextBoxColumn();
            }
        }
    }
}

```

```

DateEntered.HeaderText = "DateEntered";
DateEntered.Name = "DateEntered";
DataGridViewTextBoxColumn Start = new DataGridViewTextBoxColumn();
Start.HeaderText = "Start";
Start.Name = "Start";
DataGridViewTextBoxColumn Finish = new DataGridViewTextBoxColumn();
Finish.HeaderText = "Finish";
Finish.Name = "Finish";
DataGridViewTextBoxColumn ActualStart = new DataGridViewTextBoxColumn();
ActualStart.HeaderText = "ActualStart";
ActualStart.Name = "ActualStart";
DataGridViewTextBoxColumn ExpectedEndDate = new DataGridViewTextBoxColumn();
ExpectedEndDate.HeaderText = "ExpectedEndDate";
ExpectedEndDate.Name = "ExpectedEndDate";
DataGridViewTextBoxColumn ActualCost = new DataGridViewTextBoxColumn();
ActualCost.HeaderText = "ActualCost";
ActualCost.Name = "ActualCost";
DataGridViewTextBoxColumn PercentWorkComplete = new DataGridViewTextBoxColumn();
PercentWorkComplete.HeaderText = "PercentWorkComplete";
PercentWorkComplete.Name = "PercentWorkComplete";
DataGridViewTextBoxColumn Remarks = new DataGridViewTextBoxColumn();
Remarks.HeaderText = "Remarks";
Remarks.Name = "Remarks";
DataGridViewImageColumn Photo = new DataGridViewImageColumn();
Photo.HeaderText = "Photo";
Photo.Name = "Photo";
Photo.ImageLayout = DataGridViewImageCellLayout.Zoom;

//dgvTasksLog.Columns.Add(TasksLogID);
dgvTasksLog.Columns.Add(TaskName);
dgvTasksLog.Columns.Add(DateEntered);
dgvTasksLog.Columns.Add(Start);
dgvTasksLog.Columns.Add(Finish);
dgvTasksLog.Columns.Add(ActualStart);
dgvTasksLog.Columns.Add(ExpectedEndDate);
dgvTasksLog.Columns.Add(ActualCost);
dgvTasksLog.Columns.Add(PercentWorkComplete);
dgvTasksLog.Columns.Add(Remarks);
dgvTasksLog.Columns.Add(Photo);

//////////

while (sql_datareader.Read()) // Read() returns true if there is still a result line to read
{
    ////////////
    Image img = null;
    if (sql_datareader["Photo"] != System.DBNull.Value)
    {
        img = GetPhoto((byte[])(sql_datareader["Photo"]));
    }
    int rowId = dgvTasksLog.Rows.Add();
    // Grab the new row!
    DataGridViewRow row = dgvTasksLog.Rows[rowId];
    //row.Cells["TasksLogID"].Value = sql_datareader["TasksLogID"];
    row.Cells["TaskName"].Value = sql_datareader["TaskName"];
    row.Cells["DateEntered"].Value = sql_datareader["DateEntered"];
    row.Cells["Start"].Value = sql_datareader["Start"];
    row.Cells["Finish"].Value = sql_datareader["Finish"];
    row.Cells["ActualStart"].Value = sql_datareader["ActualStart"];
    row.Cells["ExpectedEndDate"].Value = sql_datareader["ExpectedEndDate"];
    row.Cells["ActualCost"].Value = sql_datareader["ActualCost"];
    row.Cells["PercentWorkComplete"].Value = sql_datareader["PercentWorkComplete"];
    row.Cells["Remarks"].Value = sql_datareader["Remarks"];
    row.Cells["Photo"].Value = img;
    row.Height = 100;
}

```

```

        //dgvTasksLog.Rows.Add(row);
        ////////////////
    }
    sql_con.Close();
}
catch (Exception ex)
{
    //throw new Exception(ex.Message);
}
}
private Image GetPhoto(byte[] img)
{
    byte[] a = (byte[])(img);
    MemoryStream ms = new MemoryStream(a);
    return Image.FromStream(ms);
}

private void btnMainPage_Click(object sender, EventArgs e)
{
    TasksMain TaskMain = new TasksMain();
    TaskMain.Show();
    this.Hide();
}

private void btnPrint_Click(object sender, EventArgs e)
{
    //printDocument1.Print();
    DGVPrinter printer = new DGVPrinter();

    printer.Title = "Tasks";

    //printer.SubTitle = "An Easy to Use DataGridView Printing Object";

    printer.SubTitleFormatFlags = StringFormatFlags.LineLimit |

        StringFormatFlags.NoClip;

    printer.PageNumbers = true;

    printer.PageNumberInHeader = false;

    printer.PorportionalColumns = true;

    printer.HeaderCellAlignment = StringAlignment.Near;
    printer.Footer = "";
    printer.FooterSpacing = 15;
    printer.PrintDataGridView(dgvTasksLog);
}
}
}

```

Tasks Report

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SQLite;

namespace Constructions
{

```

```

public partial class TasksReport : Form
{
    public SQLiteConnection sql_con;
    public SQLiteCommand sql_cmd;
    SQLiteDataReader sql_datareader;
    public class ComboboxItem
    {
        public string Text { get; set; }
        public object Value { get; set; }

        public override string ToString()
        {
            return Text;
        }
    }
    public int ResourceUID;
    public TasksReport(int iResourceUID)
    {
        InitializeComponent();
        ResourceUID = iResourceUID;
        dtpStartFrom.Format = DateTimePickerFormat.Custom;
        dtpStartFrom.CustomFormat = " ";
        dtpStartTo.Format = DateTimePickerFormat.Custom;
        dtpStartTo.CustomFormat = " ";
        dtpFinishFrom.Format = DateTimePickerFormat.Custom;
        dtpFinishFrom.CustomFormat = " ";
        dtpFinishTo.Format = DateTimePickerFormat.Custom;
        dtpFinishTo.CustomFormat = " ";
        Extension.SetMainConnection(ref sql_con);
        SQLiteCommand GetTasks = new SQLiteCommand("Select UID,Name As ResourceName From
Resources", sql_con);
        try
        {
            sql_con.Open();
            sql_datareader = GetTasks.ExecuteReader();
            while (sql_datareader.Read() // Read() returns true if there is still a result line to read
            {
                ComboboxItem item = new ComboboxItem();
                item.Text = sql_datareader["ResourceName"].ToString();
                item.Value = sql_datareader["UID"].ToString();
                cbResources.Items.Add(item);
            }
            sql_con.Close();
        }
        catch (Exception ex)
        {
            throw new Exception(ex.Message);
        }
    }

    private void TasksReport_Load(object sender, EventArgs e)
    {
        this.rvTasks.RefreshReport();
    }
}
}

```

Project Delete

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;

```

```

using System.Text;
using System.Windows.Forms;
using System.Data.SQLite;
using System.Xml.Linq;

namespace Constructions
{
    public partial class ProjectDelete : Form
    {
        public SQLiteConnection sql_con;
        public SQLiteCommand sql_cmd;
        SQLiteDataReader sql_datareader;
        public class ComboboxItem
        {
            public string Text { get; set; }
            public object Value { get; set; }

            public override string ToString()
            {
                return Text;
            }
        }
        public ProjectDelete()
        {
            InitializeComponent();
            Extension.SetMainConnection(ref sql_con);
            SQLiteCommand GetTasks = new SQLiteCommand("Select ProjectID,Name As ProjectName From
Project", sql_con);
            try
            {
                sql_con.Open();
                sql_datareader = GetTasks.ExecuteReader();
                while (sql_datareader.Read() // Read() returns true if there is still a result line to read
                {
                    ComboboxItem item = new ComboboxItem();
                    item.Text = sql_datareader["ProjectName"].ToString();
                    item.Value = sql_datareader["ProjectID"].ToString();
                    ddlProjects.Items.Add(item);
                    ddlProjects.SelectedIndex = ddlProjects.Items.Count - 1;
                }
                sql_con.Close();
            }
            catch (Exception ex)
            {
                throw new Exception(ex.Message);
            }
        }

        private void btnDelete_Click(object sender, EventArgs e)
        {
            var confirmResult = MessageBox.Show("Are you sure to delete this Project ???",
                "Confirm Delete!!",
                MessageBoxButtons.YesNo);
            if (confirmResult == DialogResult.Yes)
            {
                Aggregates.Aggregate("Delete From Tasks Where ProjectID=" +
((Constructions.ProjectDelete.ComboboxItem)(ddlProjects.SelectedItem)).Value.ToString());
                Aggregates.Aggregate("Delete From TasksLog Where ProjectID=" +
((Constructions.ProjectDelete.ComboboxItem)(ddlProjects.SelectedItem)).Value.ToString());
                Aggregates.Aggregate("Delete From Assignments Where ProjectID=" +
((Constructions.ProjectDelete.ComboboxItem)(ddlProjects.SelectedItem)).Value.ToString());
                Aggregates.Aggregate("Delete From Project Where ProjectID=" +
((Constructions.ProjectDelete.ComboboxItem)(ddlProjects.SelectedItem)).Value.ToString());
                Login Login = new Login();
                Login.Show();
                this.Hide();
            }
        }
    }
}

```

```
    }
    else
    {
        // If 'No', do something here.
    }
}

private void btnMainPage_Click(object sender, EventArgs e)
{
    TasksMain TaskMain = new TasksMain();
    TaskMain.Show();
    this.Hide();
}
}
```

**APPENDIX D: VALIDATION CASE STUDY- GENERATED
MONTHLY REPORT**



Reference

From

To

Resources

Progress OVERVIEW

American University in Cairo
 School of Science and Engineering
 www.aucegypt.edu

Tel: +20 (2) 2615.1000
 Fax: +20 (2) 2795 7562

1. Planned Vs Actual

Average

Tasks	Planned % of Completion	Actual % of Completion	Status	Remarks
Engineering	100%	94%	⚠	-6%
Layout drawings	100%	94%	⚠	-6%
Structural Drawings	100%	94%	⚠	-6%
Foundation drawings (sections, detail	100%	95%	⚠	-5%
RC walls and columns,shop drawings	100%	95%	⚠	-5%
super structure drawings as slabs	100%	95%	⚠	-5%
P&ID drawings	100%	93%	⚠	-7%
Piping Arrangement drawings	100%	94%	⚠	-6%
Shop Drawings (Basket Screen , MBBF	100%	93%	⚠	-7%
Cable Routing	100%	95%	⚠	-5%
Equipment Submittal	100%	94%	⚠	-6%
Internal & External Lighting	100%	93%	⚠	-7%
Cable Routing	100%	95%	⚠	-5%
SLD + Wiring Diagram	100%	95%	⚠	-5%
Load List sheet	100%	94%	⚠	-6%
Earthing Calculation	100%	94%	⚠	-6%
PROCUREMENT	55%	60%	✓	4%
DAF System	100%	59%	✗	-41%
pH Adjustment Dosing Set	100%	58%	✗	-42%
Air Blowers	100%	61%	✗	-39%
Sludge Transfer Pumps	100%	61%	✗	-39%
Sludge conditioning Dosing Set	100%	60%	✗	-40%
Filter Press Feed Pump	100%	60%	✗	-40%
Filter Press	100%	58%	✗	-42%
Lifting Station Pumps	100%	61%	✗	-39%
Buffer Tank Feed Pump	100%	59%	✗	-41%
GOS System	100%	61%	✗	-39%
MBBR Media	100%	59%	✗	-41%
Air Compressor	100%	60%	✗	-40%
Control panel	100%	60%	✗	-40%
Internal & External lighting	100%	93%	⚠	-7%
Buffer Tank Set	100%	61%	✗	-39%
Piping	100%	58%	✗	-42%
Basket Screens	49%	61%	✓	12%
Balancing Pumps	49%	58%	✓	9%
GOS System	100%	61%	✗	-39%
pH Adjustment Dosing Set	100%	58%	✗	-42%
DAF System	100%	59%	✗	-41%
pH Adjustment Dosing Set	100%	58%	✗	-42%
Nutrient Dosing Set	51%	58%	✓	7%
Air Blowers	100%	61%	✗	-39%
Air Diffuser	68%	61%	⚠	-7%
MBBR Media	100%	59%	✗	-41%
Sludge conditioning Dosing Set	100%	60%	✗	-40%
Filter Press Feed Pump	100%	60%	✗	-40%
Air Compressor	100%	60%	✗	-40%
Interconnecting Piping Works	51%	60%	✓	9%
Control Panel	100%	60%	✗	-40%
Filter Press	100%	58%	✗	-42%
Internal & External lighting	100%	93%	⚠	-7%
Ultrasonic level transmitters	51%	60%	✓	9%
Sludge Transfer Pumps	100%	61%	✗	-39%
Lifting Station Pumps	100%	61%	✗	-39%
CIVIL	34%	32%	⚠	-2%
Excavation	100%	100%	✓	0%
Soil replacement	100%	100%	✓	0%
Ground floor columns	100%	100%	✓	0%
Casting	100%	100%	✓	0%
PC	100%	100%	✓	0%
Masonry under slab on grade works	87%	100%	✓	13%
Steel fixing	100%	100%	✓	0%
Shuttering	100%	100%	✓	0%



Progress OVERVIEW

Reference	0
From	February 2, 2017
To	March 31, 2017
Resources	

Tel: +20 (2) 2615.1000
 Fax: +20 (2) 2795 7562

2. Key Performance Indicators

Tasks	BAC	PV	EV	AC	SPI	CPI	#DIV/O!	SV	CV	ETC	EAC
Down Payment /Commencement	\$ -	\$ -	\$ -	\$ -	-	1.1	#DIV/O!	\$ -	\$ -	\$ -	\$ -
Site Handing Over To IETOS	\$ -	\$ -	\$ -	\$ -	-			\$ -	\$ -	\$ -	\$ -
Kick Off Meeting	\$ -	\$ -	\$ -	\$ -	-			\$ -	\$ -	\$ -	\$ -
Letter Of Award	\$ -	\$ -	\$ -	\$ -	-			\$ -	\$ -	\$ -	\$ -
Contract Signature	\$ -	\$ -	\$ -	\$ -	-			\$ -	\$ -	\$ -	\$ -
Mobilization	\$ -	\$ -	\$ -	\$ -	-			\$ -	\$ -	\$ -	\$ -
Engineering	\$ 82,022.14	\$ 82,022.14	\$ 77,264.86	\$ -	0.9	#DIV/O!	\$ (4,757.28)	\$ 77,264.86	#DIV/O!	#DIV/O!	#DIV/O!
Layout drawings	\$ 5,468.14	\$ 5,468.14	\$ 5,140.05	\$ -	0.9	#DIV/O!	\$ (328.09)	\$ 5,140.05	#DIV/O!	#DIV/O!	#DIV/O!
Structural Drawings	\$ 5,468.14	\$ 5,468.14	\$ 5,140.05	\$ -	0.9	#DIV/O!	\$ (328.09)	\$ 5,140.05	#DIV/O!	#DIV/O!	#DIV/O!
Foundation drawings (sections, d	\$ 5,468.14	\$ 5,468.14	\$ 5,194.74	\$ -	1.0	#DIV/O!	\$ (273.41)	\$ 5,194.74	#DIV/O!	#DIV/O!	#DIV/O!
RC walls and columns,shop drawin	\$ 5,468.14	\$ 5,468.14	\$ 5,194.74	\$ -	1.0	#DIV/O!	\$ (273.41)	\$ 5,194.74	#DIV/O!	#DIV/O!	#DIV/O!
super structure drawings as slab	\$ 5,468.14	\$ 5,468.14	\$ 5,194.74	\$ -	1.0	#DIV/O!	\$ (273.41)	\$ 5,194.74	#DIV/O!	#DIV/O!	#DIV/O!
P&ID drawings	\$ 5,468.14	\$ 5,468.14	\$ 5,085.37	\$ -	0.9	#DIV/O!	\$ (382.77)	\$ 5,085.37	#DIV/O!	#DIV/O!	#DIV/O!
Piping Arrangement drawings	\$ 5,468.14	\$ 5,468.14	\$ 5,140.05	\$ -	0.9	#DIV/O!	\$ (328.09)	\$ 5,140.05	#DIV/O!	#DIV/O!	#DIV/O!
Shop Drawings (Basket Screen ,	\$ 5,468.14	\$ 5,468.14	\$ 5,085.37	\$ -	0.9	#DIV/O!	\$ (382.77)	\$ 5,085.37	#DIV/O!	#DIV/O!	#DIV/O!
Cable Routing	\$ 5,468.14	\$ 5,468.14	\$ 5,194.74	\$ -	1.0	#DIV/O!	\$ (273.41)	\$ 5,194.74	#DIV/O!	#DIV/O!	#DIV/O!
Equipment Submittal	\$ 5,468.14	\$ 5,468.14	\$ 5,140.05	\$ -	0.9	#DIV/O!	\$ (328.09)	\$ 5,140.05	#DIV/O!	#DIV/O!	#DIV/O!
Cable Routing	\$ 5,468.14	\$ 5,468.14	\$ 5,194.74	\$ -	1.0	#DIV/O!	\$ (273.41)	\$ 5,194.74	#DIV/O!	#DIV/O!	#DIV/O!
SLD + Wiring Diagram	\$ 5,468.14	\$ 5,468.14	\$ 5,194.74	\$ -	1.0	#DIV/O!	\$ (273.41)	\$ 5,194.74	#DIV/O!	#DIV/O!	#DIV/O!
Load List sheet	\$ 5,468.14	\$ 5,468.14	\$ 5,140.05	\$ -	0.9	#DIV/O!	\$ (328.09)	\$ 5,140.05	#DIV/O!	#DIV/O!	#DIV/O!
Earthing Calculation	\$ 5,468.14	\$ 5,468.14	\$ 5,140.05	\$ -	0.9	#DIV/O!	\$ (328.09)	\$ 5,140.05	#DIV/O!	#DIV/O!	#DIV/O!
PROCUREMENT	\$ 477,000.00	\$ 263,697.79	\$ 284,345.00	\$ -	1.1	#DIV/O!	\$ 20,647.21	\$ 284,345.00	#DIV/O!	#DIV/O!	#DIV/O!
pH Adjustment Dosing Set	\$ 17,857.14	\$ 17,857.14	\$ 10,357.14	\$ -	0.6	#DIV/O!	\$ (7,500.00)	\$ 10,357.14	#DIV/O!	#DIV/O!	#DIV/O!
Air Blowers	\$ 20,428.57	\$ 20,428.57	\$ 12,461.43	\$ -	0.6	#DIV/O!	\$ (7,967.14)	\$ 12,461.43	#DIV/O!	#DIV/O!	#DIV/O!
Sludge Transfer Pumps	\$ 28,571.43	\$ 28,571.43	\$ 17,428.57	\$ -	0.6	#DIV/O!	\$ (11,142.86)	\$ 17,428.57	#DIV/O!	#DIV/O!	#DIV/O!
Sludge conditioning Dosing Set	\$ 18,571.43	\$ 18,571.43	\$ 11,142.86	\$ -	0.6	#DIV/O!	\$ (7,428.57)	\$ 11,142.86	#DIV/O!	#DIV/O!	#DIV/O!
Filter Press Feed Pump	\$ 20,000.00	\$ 20,000.00	\$ 12,000.00	\$ -	0.6	#DIV/O!	\$ (8,000.00)	\$ 12,000.00	#DIV/O!	#DIV/O!	#DIV/O!
Filter Press	\$ 24,285.71	\$ 24,285.71	\$ 14,085.71	\$ -	0.6	#DIV/O!	\$ (10,200.00)	\$ 14,085.71	#DIV/O!	#DIV/O!	#DIV/O!
Buffer Tank Feed Pump	\$ 20,714.29	\$ 20,714.29	\$ 12,221.43	\$ -	0.6	#DIV/O!	\$ (8,492.86)	\$ 12,221.43	#DIV/O!	#DIV/O!	#DIV/O!
GOS System	\$ 20,000.00	\$ 20,000.00	\$ 12,200.00	\$ -	0.6	#DIV/O!	\$ (7,800.00)	\$ 12,200.00	#DIV/O!	#DIV/O!	#DIV/O!
MBBR Media	\$ 15,714.29	\$ 15,714.29	\$ 9,271.43	\$ -	0.6	#DIV/O!	\$ (6,442.86)	\$ 9,271.43	#DIV/O!	#DIV/O!	#DIV/O!
Air Compressor	\$ 18,571.43	\$ 18,571.43	\$ 11,142.86	\$ -	0.6	#DIV/O!	\$ (7,428.57)	\$ 11,142.86	#DIV/O!	#DIV/O!	#DIV/O!
Control panel	\$ 17,142.86	\$ 17,142.86	\$ 10,285.71	\$ -	0.6	#DIV/O!	\$ (6,857.14)	\$ 10,285.71	#DIV/O!	#DIV/O!	#DIV/O!
Buffer Tank Set	\$ 6,428.57	\$ 6,428.57	\$ 3,921.43	\$ -	0.6	#DIV/O!	\$ (2,507.14)	\$ 3,921.43	#DIV/O!	#DIV/O!	#DIV/O!
Piping	\$ 10,000.00	\$ 10,000.00	\$ 5,800.00	\$ -	0.6	#DIV/O!	\$ (4,200.00)	\$ 5,800.00	#DIV/O!	#DIV/O!	#DIV/O!
Basket Screens	\$ 4,285.71	\$ 2,088.53	\$ 2,614.29	\$ -	1.3	#DIV/O!	\$ 525.75	\$ 2,614.29	#DIV/O!	#DIV/O!	#DIV/O!
Balancing Pumps	\$ 17,142.86	\$ 8,354.12	\$ 9,942.86	\$ -	1.2	#DIV/O!	\$ 1,588.73	\$ 9,942.86	#DIV/O!	#DIV/O!	#DIV/O!
GOS System	\$ 20,000.00	\$ 20,000.00	\$ 12,200.00	\$ -	0.6	#DIV/O!	\$ (7,800.00)	\$ 12,200.00	#DIV/O!	#DIV/O!	#DIV/O!
DAF System	\$ 21,142.86	\$ 21,142.86	\$ 12,474.29	\$ -	0.6	#DIV/O!	\$ (8,668.57)	\$ 12,474.29	#DIV/O!	#DIV/O!	#DIV/O!
pH Adjustment Dosing Set	\$ 17,857.14	\$ 17,857.14	\$ 10,357.14	\$ -	0.6	#DIV/O!	\$ (7,500.00)	\$ 10,357.14	#DIV/O!	#DIV/O!	#DIV/O!
Nutrient Dosing Set	\$ 8,571.43	\$ 4,361.34	\$ 4,971.43	\$ -	1.1	#DIV/O!	\$ 610.08	\$ 4,971.43	#DIV/O!	#DIV/O!	#DIV/O!
Air Blowers	\$ 20,428.57	\$ 20,428.57	\$ 12,461.43	\$ -	0.6	#DIV/O!	\$ (7,967.14)	\$ 12,461.43	#DIV/O!	#DIV/O!	#DIV/O!
Air Diffuser	\$ 11,428.57	\$ 7,723.21	\$ 6,971.43	\$ -	0.9	#DIV/O!	\$ (751.79)	\$ 6,971.43	#DIV/O!	#DIV/O!	#DIV/O!
MBBR Media	\$ 15,714.29	\$ 15,714.29	\$ 9,271.43	\$ -	0.6	#DIV/O!	\$ (6,442.86)	\$ 9,271.43	#DIV/O!	#DIV/O!	#DIV/O!
Filter Press Feed Pump	\$ 20,000.00	\$ 20,000.00	\$ 12,000.00	\$ -	0.6	#DIV/O!	\$ (8,000.00)	\$ 12,000.00	#DIV/O!	#DIV/O!	#DIV/O!
Air Compressor	\$ 18,571.43	\$ 18,571.43	\$ 11,142.86	\$ -	0.6	#DIV/O!	\$ (7,428.57)	\$ 11,142.86	#DIV/O!	#DIV/O!	#DIV/O!
Interconnecting Piping Works	\$ 14,285.71	\$ 7,268.91	\$ 8,571.43	\$ -	1.2	#DIV/O!	\$ 1,302.52	\$ 8,571.43	#DIV/O!	#DIV/O!	#DIV/O!
Control Panel	\$ 17,142.86	\$ 17,142.86	\$ 10,285.71	\$ -	0.6	#DIV/O!	\$ (6,857.14)	\$ 10,285.71	#DIV/O!	#DIV/O!	#DIV/O!
Filter Press	\$ 24,285.71	\$ 24,285.71	\$ 14,085.71	\$ -	0.6	#DIV/O!	\$ (10,200.00)	\$ 14,085.71	#DIV/O!	#DIV/O!	#DIV/O!
Internal & External lighting	\$ 5,468.14	\$ 5,468.14	\$ 5,085.37	\$ -	0.9	#DIV/O!	\$ (382.77)	\$ 5,085.37	#DIV/O!	#DIV/O!	#DIV/O!
Sludge Transfer Pumps	\$ 28,571.43	\$ 28,571.43	\$ 17,428.57	\$ -	0.6	#DIV/O!	\$ (11,142.86)	\$ 17,428.57	#DIV/O!	#DIV/O!	#DIV/O!
Lifting Station Pumps	\$ 14,285.71	\$ 14,285.71	\$ 8,714.29	\$ -	0.6	#DIV/O!	\$ (5,571.43)	\$ 8,714.29	#DIV/O!	#DIV/O!	#DIV/O!



Progress OVERVIEW

Reference	0
From	February 2, 2017
To	March 31, 2017
Resources	

Tel: +20 (2) 2615.1000
 Fax: +20 (2) 2795 7562

	\$	205,000.00	\$	70,211.42	\$	65,815.79	\$	-	0.9	#DIV/0!	\$	(4,395.63)	\$	65,815.79	#DIV/0!	#DIV/0!
CIVIL	\$	205,000.00	\$	70,211.42	\$	65,815.79	\$	-	0.9	#DIV/0!	\$	(4,395.63)	\$	65,815.79	#DIV/0!	#DIV/0!
Excavation	\$	6,000.00	\$	6,000.00	\$	6,000.00	\$	-	1.0	#DIV/0!	\$	-	\$	6,000.00	#DIV/0!	#DIV/0!
Soil replacement	\$	5,285.71	\$	5,285.71	\$	5,285.71	\$	-	1.0	#DIV/0!	\$	-	\$	5,285.71	#DIV/0!	#DIV/0!
Casting	\$	16,428.57	\$	16,428.57	\$	16,428.57	\$	-	1.0	#DIV/0!	\$	-	\$	16,428.57	#DIV/0!	#DIV/0!
PC	\$	2,857.14	\$	2,857.14	\$	2,857.14	\$	-	1.0	#DIV/0!	\$	-	\$	2,857.14	#DIV/0!	#DIV/0!
Steel fixing	\$	2,714.29	\$	2,714.29	\$	2,714.29	\$	-	1.0	#DIV/0!	\$	-	\$	2,714.29	#DIV/0!	#DIV/0!
Shuttering	\$	3,571.43	\$	3,571.43	\$	3,571.43	\$	-	1.0	#DIV/0!	\$	-	\$	3,571.43	#DIV/0!	#DIV/0!
Excavation	\$	6,000.00	\$	6,000.00	\$	6,000.00	\$	-	1.0	#DIV/0!	\$	-	\$	6,000.00	#DIV/0!	#DIV/0!
Soil replacement	\$	5,285.71	\$	5,285.71	\$	5,285.71	\$	-	1.0	#DIV/0!	\$	-	\$	5,285.71	#DIV/0!	#DIV/0!

3. Planned Tasks: Next Period

Task	Resources	Planned % of Completion	Actual % of Completion
Building license		100%	0%
Engineering	TH-P-M-01	100%	94%
Layout drawings	DESIGN ENGINEER	100%	94%
Structural Drawings	DESIGN ENGINEER	100%	94%
Foundation drawings (sections, d	DESIGN ENGINEER	100%	95%
RC walls and columns shop drawin	DESIGN ENGINEER	100%	95%
super structure drawings as slab	DESIGN ENGINEER	100%	95%
P&ID drawings	DESIGN ENGINEER	100%	93%
Piping Arrangement drawings	DESIGN ENGINEER	100%	94%
Shop Drawings (Basket Screen,	DESIGN ENGINEER	100%	93%
Cable Routing	DESIGN ENGINEER	100%	95%
Equipment Submittal	DESIGN ENGINEER	100%	94%
Internal & External Lighting	DESIGN ENGINEER	100%	93%
Cable Routing	DESIGN ENGINEER	100%	95%
SLD + Wiring Diagram	DESIGN ENGINEER	100%	95%
Load List sheet	DESIGN ENGINEER	100%	94%
Earthing Calculation	DESIGN ENGINEER	100%	94%
PROCUREMENT		78%	60%
DAF System	PROC. & FAB. ENGINEERS	100%	59%
pH Adjustment Dosing Set	PROC. & FAB. ENGINEERS	100%	58%
Air Blowers	PROC. & FAB. ENGINEERS	100%	61%
Sludge Transfer Pumps	PROC. & FAB. ENGINEERS	100%	61%
Sludge conditioning Dosing Set	PROC. & FAB. ENGINEERS	100%	60%
Filter Press Feed Pump	PROC. & FAB. ENGINEERS	100%	60%
Filter Press	PROC. & FAB. ENGINEERS	100%	58%
Lifting Station Pumps		100%	61%
Buffer Tank Feed Pump	PROC. & FAB. ENGINEERS	100%	59%
GOS System	PROC. & FAB. ENGINEERS	100%	61%
MBBR Media	PROC. & FAB. ENGINEERS	100%	59%
Air Compressor	PROC. & FAB. ENGINEERS	100%	60%
Control panel	PROC. & FAB. ENGINEERS	100%	60%
Internal & External lighting	DESIGN ENGINEER	100%	93%
Buffer Tank Set	PROC. & FAB. ENGINEERS	100%	61%
Piping		100%	58%
Basket Screens	PROC. & FAB. ENGINEERS	75%	61%
Balancing Pumps	PROC. & FAB. ENGINEERS	75%	58%
GOS System	PROC. & FAB. ENGINEERS	100%	61%
pH Adjustment Dosing Set	PROC. & FAB. ENGINEERS	100%	58%
DAF System	PROC. & FAB. ENGINEERS	100%	59%
pH Adjustment Dosing Set	PROC. & FAB. ENGINEERS	100%	58%
Nutrient Dosing Set	PROC. & FAB. ENGINEERS	78%	58%



Reference	0
From	February 2, 2017
To	March 31, 2017
Resources	

Progress OVERVIEW

American University in Cairo
School of Science and Engineering
 www.aucegypt.edu

Tel: +20 (2) 2615.1000
 Fax: +20 (2) 2795 7562

Air Blowers	PROC. & FAB. ENGINEERS	100%	61%
Air Diffuser	PROC. & FAB. ENGINEERS	100%	61%
MBBR Media	PROC. & FAB. ENGINEERS	100%	59%
Sludge conditioning Dosing Set	PROC. & FAB. ENGINEERS	100%	60%
Filter Press Feed Pump	PROC. & FAB. ENGINEERS	100%	60%
Air Compressor	PROC. & FAB. ENGINEERS	100%	60%
Interconnecting Piping Works	PROC. & FAB. ENGINEERS	78%	60%
Control Panel	PROC. & FAB. ENGINEERS	100%	60%
Filter Press	PROC. & FAB. ENGINEERS	100%	58%
Internal & External lighting	DESIGN ENGINEER	100%	93%
Ultrasonic level transmitters	PROC. & FAB. ENGINEERS	78%	60%
Sludge Transfer Pumps	PROC. & FAB. ENGINEERS	100%	61%
Lifting Station Pumps		100%	61%
Deep clean waste injection	SITE ENGINEER	43%	0%
Pumps	SITE ENGINEER	23%	0%
Control Panel	PROC. & FAB. ENGINEERS	100%	60%
Inter connecting Cables	SITE ENGINEER	15%	0%
CIVIL		54%	32%
Ground floor slab works	Carpenter	60%	30%
Insulation works	REPAIRING WORKS TEAM	100%	90%
Slab on grade works	Steel fixer	100%	0%
Doors / Stairs		100%	0%
Plastering and Finishing		100%	0%
Casting	Concrete forman	100%	100%
PC		100%	100%
Steel fixing	Steel fixer	100%	100%
Shuttering	Carpenter	100%	100%

3. Invoices

Invoice Number	Invoice Date	Payment Date	Actual Invoice Date	Actual Payment Date	Gross Amount	Net amount	PercentComplete	Collected amount	Total CI	Total CI Net (paid)	Total CO	Defect
Invoices: Advanced Payment	January 16, 2017	January 31, 2017	January 16, 2017	January 19, 2017	\$ 250,000	\$ 250,000		1 \$ 250,000.00	\$ 250,000	\$ 250,000	\$ -	✔ \$ 250,000
Invoices: Invoice 1	February 28, 2017	March 15, 2017	March 1, 2017	March 30, 2017	\$ 42,857	\$ 27,857		1 \$ 27,857.05	\$ 277,857	\$ 277,857	\$ -	✔ \$ 277,857
Invoices: Invoice 2	April 15, 2017	April 30, 2017	January 0, 1900	January 0, 1900	\$ 50,000	\$ 32,500		0 \$ -	\$ 310,357	\$ 277,857	\$ -	✔ \$ 277,857
Invoices: Invoice 3	May 16, 2017	May 31, 2017	January 0, 1900	January 0, 1900	\$ 485,714	\$ 315,714		0 \$ -	\$ 626,071	\$ 277,857	\$ -	✔ \$ 277,857
Invoices: Invoice 4	June 14, 2017	June 29, 2017	January 0, 1900	January 0, 1900	\$ 35,714	\$ 23,214		0 \$ -	\$ 649,285	\$ 277,857	\$ -	✔ \$ 277,857
Invoices: Invoice 5	July 16, 2017	July 31, 2017	January 0, 1900	January 0, 1900	\$ 17,142	\$ 11,142		0 \$ -	\$ 660,428	\$ 277,857	\$ -	✔ \$ 277,857
Invoices: Invoice 6	August 16, 2017	August 31, 2017	January 0, 1900	January 0, 1900	\$ 18,571	\$ 12,071		0 \$ -	\$ 672,499	\$ 277,857	\$ -	✔ \$ 277,857
Invoices: Retention	August 16, 2018	August 31, 2018	January 0, 1900	January 0, 1900	\$ 100,000	\$ 100,000		0 \$ -	\$ 772,499	\$ 277,857	\$ -	✔ \$ 277,857



Progress Dashboard

