An analysis of technological improvements of a financial department in a drinking water plant in Myanmar, South East Asia

CARDIFF METROPOLITAN UNIVERSITY

ST20109826: AUNG MYO THET (APRIL 24, 2017)
Abstract

In this dissertation project, the small businesses in the developing countries, impact of using management software in small business, the benefit of using a database in the software and financial management in developing countries are discussed in the literature review. This dissertation is researched for the financial department in a drinking water plant in developing country, Myanmar, also known as Burma, South East Asia.

To develop a software for the technological improvements of the financial department, this research is discussed about the software development methodologies. Then, the requirements are analysed and one of the methodologies is selected for this project and this dissertation will follow this technology to develop the software. As the method, this software development is designed to meet the customer requirements, implemented by following the design, tested by black box method, evaluated and recommended by the developer.

Finally, in the conclusion, it is described with the problems the developer has faced and how the developer solved it. It contains the different experience between questioning in the factory and the expected result. This conclusion is mentioned about the technical experience and it obtained the aims of this dissertation. Moreover, the personal aims are also met with the researcher’s objects. The sources for this research are all referenced to the Harvard style.
Acknowledgements

Firstly, I am delighted to be an international student in Cardiff Metropolitan university. And I would like to thank for my parents: U Moe Tun and Daw Theingi Khin Htwe although they are separated. They supported me to live and study smoothly here without getting any part time jobs. Special thanks to my English teacher, Mr Irin David Evans, for pushing me to Cardiff and for checking my mistakes. Big thanks to Cardiff mum, my landlady, Mrs Debbie Coleman for cooking lunch and dinner while I am studying for this research. Other thanks to my supervisor Dr Imtiaz Khan for helping me and the support that he guided me how to structure the dissertation and how to point out the facts while he was in the Easter holidays through the mail.

I declare that this software engineering dissertation project which is titled "An analysis of technological improvements in a financial department in a drinking water plant in Myanmar, South East Asia" is 100 percent my own work. I have never submitted this dissertation for any other degree or for any other purposes.

Supervisor Name:  

Signature:

Candidate Name:  

Signature:

Student Number:  

BSc (Hons) Software Engineering Dissertation Project (2016-2017)
Table of Contents

1. Introduction ......................................................................................................................... 7
  1.1 Aim of this research ........................................................................................................ 7
  1.2 Primary Objectives ........................................................................................................ 8
  1.3 Personal Objectives ....................................................................................................... 8

2. Literature Review .................................................................................................................. 9
  2.1 Introduction .................................................................................................................... 9
  2.2 Small Businesses in Developing Countries .................................................................... 9
  2.3 The Importance of Technology in Financial Management in Small Businesses in Developing Countries .................................................................................................................. 10
    2.3.1 Financial Planning, Monitoring and Controlling ...................................................... 10
    2.3.2 The Importance of Project Management for Small Business ................................. 11
    2.3.3 The Importance of Financial Management for Small Business ............................. 11
    2.3.4 Benefit of Using Databases in the Software .......................................................... 12
    2.3.5 Cloud Computing .................................................................................................. 12
  2.4 The Weakness of Paper-Based System .......................................................................... 12
  2.5 Conclusion ...................................................................................................................... 13

3. Software Development Life Cycle (SDLC) ........................................................................... 14
  3.1 Introduction .................................................................................................................... 14
  3.2 Requirement Phase ........................................................................................................ 14
  3.3 Design Phase .................................................................................................................. 15
  3.4 Development or coding Phase ....................................................................................... 15
  3.5 Testing Phase ................................................................................................................ 15
    3.5.1 Unit testing ............................................................................................................ 15
    3.5.2 Integration testing ................................................................................................ 15
    3.5.3 System testing ...................................................................................................... 16
    3.5.4 Acceptance Testing ............................................................................................... 16
  3.6 Evaluation and Maintenance Phase .............................................................................. 17

4. Collecting Information and the Functions of Financial Department of Water Factory .......................................................... 18
  4.1 Company Background ................................................................................................. 18
  4.2 Description of the people I interviewed ...................................................................... 18
  4.3 Interviewee 1 .............................................................................................................. 18
  4.4 Interviewee 2 .............................................................................................................. 20
  4.5 Interviewee 3 .............................................................................................................. 22
  4.6 Summary of this Interview .......................................................................................... 24

5. Analysis and Suitable Methodology for Financial Department ........................................... 26
  5.1 Introduction .................................................................................................................... 26
  5.2 Software Development Methodologies ......................................................................... 26
  5.3 Waterfall Model .......................................................................................................... 27
    5.3.1 Advantages of the Waterfall Model ..................................................................... 27
    5.3.2 Disadvantages of the Waterfall Model ................................................................. 28
  5.4 Spiral Model ................................................................................................................ 28
5.4.1 Advantages of Spiral Model ................................................................. 29
5.4.2 Disadvantages of the Spiral Model ..................................................... 29
5.5 Rapid Application Development (RAD) .................................................. 29
5.5.1 Advantages of RAD ........................................................................... 30
5.5.2 Disadvantages of RAD ....................................................................... 30
5.6 The Suitable Methodology for the Project .............................................. 31

6. Design for the Project .............................................................................. 32
  6.1 Introduction .......................................................................................... 32
  6.2 Requirements for implementing a software application ....................... 32
  6.3 Choosing of a real-world object from the customer requirements ........ 32
    6.3.1 Agent Table .................................................................................... 33
    6.3.2 Employee Table ............................................................................ 33
    6.3.3 Income Table ................................................................................ 33
    6.3.4 Salary Table .................................................................................. 33
  6.4 Functions of the financial department .................................................. 34
  6.5 Data Flow Design .................................................................................. 34
  6.6 Hand Drawn Design ............................................................................ 35
    6.6.1 The Main Form ............................................................................. 35
    6.6.2 The Register Agent Form ............................................................... 35
    6.6.3 The Check Agent Form ................................................................ 36
    6.6.4 The Employee Register Form ......................................................... 37
    6.6.5 The Employee Information Form ................................................... 37
    6.6.6 The Salary Form ............................................................................ 38
    6.6.7 The Income Record Form .............................................................. 39

7. Implementation of the Project ................................................................. 40
  7.1 Introduction .......................................................................................... 40
  7.2 Class Diagram ....................................................................................... 40
  7.3 Use Case Diagram ................................................................................ 40
  7.4 Sequence Diagram ............................................................................... 41
  7.5 User Manual for the Financial Department ......................................... 41
    7.5.1 Main Board or Main Form ............................................................. 42
    7.5.2 Agent Registration Form ............................................................... 43
    7.5.3 Check Agent Form ....................................................................... 44
    7.5.4 Employee Registration Form ......................................................... 45
    7.5.5 Employee Information Form ......................................................... 46
    7.5.6 Salary Calculator Form ................................................................. 48
    7.5.7 Income Record Form .................................................................... 49

8. Testing of the Project ............................................................................... 50
  8.1 Introduction .......................................................................................... 50
  8.2 White Box or Structural Testing .......................................................... 50
  8.3 Black Box or Functional Testing ......................................................... 50
  8.4 Test Plan for Developed Software ....................................................... 51
    8.4.1 Test Plan for Main Form ............................................................... 51
    8.4.2 Test Plan for Agent Register Form ................................................ 53
8.4.3 Test Plan for Employee Register Form .............................................. 56
8.4.4 Test Plan for Check Agent Form ....................................................... 59
8.4.5 Test Plan for Employee Information Form ....................................... 62
8.4.6 Test Plan for Salary Calculator Form .................................................. 63
8.4.7 Test Plan for Income Record Form ..................................................... 67

9. Evaluation, Recommendation and Maintenance of the Developed Software ................................................................. 70
   9.1 Introduction .......................................................................................... 70
   9.2 Evaluation for this project .................................................................. 70
   9.3 Recommendation and Maintenance for this project .......................... 70

10. Conclusion ............................................................................................... 72
   10.1 Experience in questioning ................................................................. 72
   10.2 Improvement of Language Skill ......................................................... 72
   10.3 Technical Experience ........................................................................ 72

11. Reference .................................................................................................. 75
1. Introduction

Society has witnessed plenty of revolutions caused by innovations dating as far back as the invention of the wheel or as recent as the light bulb, but modern technology, 21st-century technology, has an extra dimension – speed. The pace of change that today's technological revolution is setting, accelerating the entire planet into a faster, more automated rhythm, has shifted from the gentle introduction of new devices and snappy software to a full-on assault course that possibly only those involved in technology will be able to face.

The developed world is leading the way and the developing world seems to be scrambling to keep up. This disparity of technological knowhow is surely impeding developing countries from transitioning quicker and for economic sectors such as the SME sector from flourishing and becoming more competitive. Perhaps the international institutions could help stimulate growth in developing economies by investing heavily in technology. This study proposes to explore the improvements of a financial department of a drinking-water plant in the developing economy of the Republic of the Union of Myanmar in South East Asia.

With the evolution of the technology, most businesses are applying information systems and using software to solve their problems and to save their precious time and money. These businesses are likely to feel the effects of the positive impacts of using a specifically designed software. Many businesses in Myanmar, however, are not using any such systems and therefore find it more difficult to identify problems, such as low profitability, and elaborate a plan to overcome this. Therefore, the role of using the software in the developing sector is crucial to the evolution and survival of all business.

For the purposes of this study, a medium-sized enterprise in Yangon, the commercial and economic centre of Myanmar, has been selected.

1.1 Aim of this research

Myanmar, also known as Burma, is one of the least-developed countries in Southeast Asia. Most businesses simply do not know how to use software or are even aware of what the impact of using the software in the companies would be. The aim of this dissertation is to investigate an enterprise called AMT - a purified drinking water plant in Myanmar. After completing the research and designing the appropriate software, it can be tested by applying it to the financial department of the plant. The project would demonstrate how technology can improve profit percentage for example and allow finance staff to work more efficiently and effectively in the future. A final improvement would be the saving of natural resources that a paper-based system demands.
At present, the enterprise in question is using paper to record every transaction, a very time-consuming and laborious system, which is then stored in cupboards. This is not safe for many reasons. For example, the information could be stolen, or in the event of the fire, the entire information could be lost. Examining the possible impacts of technology on a medium-sized enterprise in a developing economy is the fundamental idea of this dissertation.

1.2 Primary Objectives
The first objective is to introduce technology to enterprises in developing countries so that their economies can begin to grow at a quicker pace and will eventually be as technologically advanced as first world countries. Another objective is to review the problems of the factory and choose a suitable software structure to both addresses the issues and solve them. This investigation would take place within the finance department, identifying the work processes. In turn, this could ensure that costs are kept to a minimum. A more long-term objective would be for the enterprise to expand as a result of the system so that the business itself depended on the system instead of a person. Eventually, even if the founder of the business is not present, the business will run itself. A further objective is to create a smarter workplace, contributing towards an improved work-life balance for the employees simply by replacing the old paper-based system with a computerised one. This would save both time and money. Finally, with the introduction of technology into the SME sector, perhaps this could stimulate the development of E-Government in Myanmar.

1.3 Personal Objectives
The overall personal objective is to complete an academic research paper. This would enable me to design specific software for practical use, get actual project experience and prove my fortitude while working under pressure. This challenge should also improve my communication skills and it gives me the opportunity to practice in the hope of perfecting my programming abilities. By researching new techniques and being aware of the difficulties and opportunities in the area, I hope in the long term to stimulate growth in the SME sector in Southeast Asia, bringing higher salaries and a better standard of living to the citizens of Myanmar.
2. Literature Review

2.1 Introduction

For the purpose of this study, prior research was done on small- and medium-sized enterprises (SMEs) in developing countries, the importance of technology in financial management in small businesses in developing countries, for example, Myanmar, the benefit of using databases in software, the cloud computing and the weakness of paper-based systems.

2.2 Small Businesses in Developing Countries

There are 195 countries in the world. They can normally be divided into two types of country. They are developed and developing countries. Among them, Myanmar is one of the developing countries.

According to the Myanmar Government organisation, Central Department of Small and Medium Enterprises Development (2015), there are 68.24% small business, 18.93% medium-sized business and 12.83% of large business operating in Myanmar. The German Institute for Development Evaluation (2015) stated that “more than 90% of the Myanmar economy is composed of private companies, particularly small and medium enterprises (SMEs)”.

The UK government (2016) declared that “small businesses accounted for 99.3% of all private sector businesses at the start of 2016 and 99.9% were small or medium-sized (SMEs) in the UK. The total employment in SMEs was 15.7 million; 60% of all private sector employment in the UK. The combined annual turnover of SMEs was £1.8 trillion, 47% of all private sector turnover in the UK”.

By analysing the above facts, it becomes evident that both developed and developing countries rely on SMEs for their economy.

The Government of the Netherlands indicated that “economic growth is an important weapon in the fight against poverty and hunger in developing countries. The government therefore encourages these countries to develop their private sectors. For example, by helping entrepreneurs build their knowledge and skills”. As they said, developing countries need to develop a growing economy. Therefore, these countries need to encourage small businesses. They may give some loans from the government to the small business for development. On the other hand, small businesses can promote themselves. For example, applying technologies in the business may stimulate the development of the business profit. Consequently, it can be support the economic growth in developing countries.
2.3 The Importance of Technology in Financial Management in Small Businesses in Developing Countries

Applying technology in business is one of the tools that can be used to improve the business profit, to produce a quality product, and to build an excellent partnership with customers. Small businesses are carefully working with their customers in order to provide the best service and/or product and earn fidelity from their customers. Technology positively has the capacity to develop small businesses. Although human beings may create everything, small businesses will rely on the technology to operate the business functions such as financial management.

Financial management is critically important to develop the quality of business sectors. Running a business directly depends on the functions of the financial department. Sponaugle (2014) expressed that “there are so many variables to consider that enable the operation to run smoothly”.

2.3.1 Financial Planning, Monitoring and Controlling

Financial planning is a management strategy usually employed by financial administrators to steer the functions of finance without the interference of an owner or the shareholders. However, some managers may sometimes be actively involved on the Board to support the financial management. Financial planning can contain setting the budget for running the business, paying the employees’ salary and making investments. Before starting a business, the plan should be set into objectives with these budgets. This plan should then be evaluated to prove that the plan is efficiently supporting and doubtlessly working towards business growth within a fixed period of time (usually 12 months). A business without a suitable plan may not meet the business requirements. Planning may also include a decision of investing money into the important or correct place, for example, extending the business area. (Sponaugle, 2014)

Financial monitoring can perhaps be one of the most important functions of financial management. It may comprise a combination of producing and analysing financial reports which depend on the functions of management. They are payable and receivable reports, income reports, currently advantages and disadvantages reports, working investment reports, and the cashflow reports. By the use of close monitoring and analysing these reports, the positive impacts of business become developing and growing not only in the financial department but also the whole business. (Sponaugle, 2014)

Financial control is usually one of the jobs of policy makers and financial administrators. By drafting policies and procedures, they can support the business and block mismanagement of
money. These policies may include how the business manages the budget overall, how to record the income and the outflows, what method of financial reporting is accepted. There are numerous methods in accounting to detect the income, for example, these methods will contain how revenue is eventually reported and how taxes, therefore, are controlled such as the repayment, the recovery cost, the achievements and the proportion of completion. (Sponaugle, 2014)

2.3.2 The Importance of Project Management for Small Business

Dey (2016) stated that “project management should be implemented in small as well as large organizations. It does not matter how small a business is, but project management certainly ensures that tasks get done quickly and efficiently”. Project management can be a necessary technique for every business because businesses should achieve the objectives of the business within the limited time, human and budget resources. These objectives will define the value of the business and they will promote the business stock, and the skill of the employees. Meeting the limit of the goals is a challenge for small enterprises because they have the limited resources rather than large businesses. There, choosing the correct tools for the project is also important for small businesses because they have to face the unfixed and complicated business surroundings. However, a lack resources is still one of the barriers for small business. (Dey, 2016)

2.3.3 The Importance of Financial Management for Small Business

Financial management is also essential for small business as part of the project management in small enterprises. One of the benefits of financial management allows the business owner to control the business, for example, checking the income on the daily processes. Businesses need to be controlled under the law in developing countries, especially Myanmar, regulation is still underdeveloped. The business owners can only access their balance information but cannot transfer money from one place to another through the Internet because, for example, the banks from Myanmar do not allow the transference of money online, although the banking sector is now beginning to develop this. In financial management, the functions from the management offer the opportunities of updating from their documents or books into electrical storage. The main point of financial management is to operate the system regularly after storing and retrieving the data for the business. (Inform Accounts, 2013) Any business doing this in a developing country at the moment will be in a strong position to take advantage of the transition period and be ready for the moment when banks and financial services start.
functioning like in the developed world.

2.3.4 Benefit of Using Databases in the Software

A Database is an electronic storage of collected data which is important for many purposes. The developer can speedily store the data into it, retrieve or explore the data that are already stored, update the data and delete each datum or all of them from the database. A database may contain many tables which represent different aspects such as customer, employee, product, etc.

In the 21st century, the database is necessary for businesses which are running with technical support. The database can help to manage for improving employee records, the inventory records, and the customers’ records.Nibusinessinfo (2017) expressed that “if customers are the lifeblood of your business, then a good customer relationship management (CRM) database should be at the heart of your growth plans. Full-featured CRM databases are usually powerful enough to store and process everything, from customer contact details, interaction history and accounts, to new prospects, leads and business opportunities”.

2.3.5 Cloud Computing

The Cloud is a place where data can be stored enormously. It is similar to a database. However, the Cloud does not need physical space such as hard drives. For example, people can easily see the Cloud but they cannot grasp the Cloud. Cloud providers will service the customer to sell the space and they will monitor them so that information is never lost. Cloud computing means that users can upload or store many files and they can download or retrieve them again even if they are not in the working area (BBC, 2012). However, the user needs an Internet connection. Although it is powerful for business storage, it is not compatible for small businesses in the developing countries with poor internet connection.

2.4 The Weakness of Paper-Based System

This paragraph will describe the disadvantages of using paper-based management. In more modern times, information has been growing in businesses. If a business is using a paper-based system, an employee needs to file the papers and find them quickly and efficiently. These tasks become a risk for the management of collecting information. However, some businesses in developing countries are still using paper-based management system. The bigger the information, the more the company needs space to save them. Therefore, the company premises could be full of documents. And these documents could be damaged by
extreme weather events such as cyclones, flooding, etc. and of course in the event of fire. Furthermore, the papers may be stolen or seen by someone who is not authorized. Using paper is ineffective in transferring information from one person or place to another. It may be easy to transfer if this transportation is not copious. Otherwise, if the company needs to send numerous papers, it becomes difficult. From the point of view of cost, using paper is one of the negatives for financial management. For example, the price of 1TB portable external hard drive is around 50 pounds on Amazon and the price of a cupboard may be equal 50 pounds. Though the prices are equal, the capacity of collecting data on a hard drive is significantly higher than the cupboard. And the last problem is for editing documents. For example, if the user wants to change something from the document, the user needs to copy the original first. It is wasting time and money. (Loci Solutions, 2016)

Subsequently, electronic document management systems, applying database, has become very popular in the modern businesses world and needs to be extended to the developing world.

2.5 Conclusion

After analysing the above the information of small businesses in developing countries, we can realise how important the role of small business to all countries, why businesses need technology, how to apply the technology into businesses, how important the financial department is in business, and how important project and financial management is in SME sectors. Consequently, small businesses in developing countries require financial management for the development of business systems, the improvement of business and the production of quality goods and services.
3. Software Development Life Cycle (SDLC)

3.1 Introduction
The software is probably popular in the modern age because of the demands of business. Software Development Life Cycle is an essential procedure to ensure the excellence of necessary software. It consists of a series of phases for the development and management of a software. There are typically five phases in SDLC. It begins with the analysis and gathering of user requirements and ends with the evaluation or deployment. The requirement is then translated into design and code and followed by the design phase of the project. Testing and maintenance are performed at the end.

3.2 Requirement Phase
This phase is fundamental for the project. It is also a critical step for the project to succeed. Business requirements and customer needs are collected in this phase. The interviewer needs to collect the expectations of the customers in detail and document them. This phase is a repetitive step with much transmission between customers and team members. The following three techniques can be applied to collect the customer requirements:

- to classify and collect stakeholder requirements after surveying and questioning the customers;
- to construct many use cases to define each action which the customer will participate in the new system;
- some examples to demonstrate to the customer what the final product will look like.

In this gathering phase, it is essential to take great care of the customers, ascertaining what their requirements are, and thus being able to design a successful outcome for the project.
3.3 Design Phase
The development team leads in this phase. This body of staff can consist of designers, engineers and lead developers. The requirements of technical design will be prepared from the specification of the identified customer needs which were analysed in the previous stage. This design will help to specify hardware, software, security processes, and system requirements as well as give support to define the overall system design. The requirements of business and the technical necessity will be used to elaborate details such as the tables to add into the database. Moreover, new transactions may be defined if the customer needs other requirements.

3.4 Development or coding Phase
This is probably the longest phase of SDLC. After receiving the documentation of the system design, the work of actual coding can be started. The development team is the main role in this phase because it is the stage to implement the code for the project. The developer will need to report to the business analysts how much of the work has been done or what process is being done at the current time. It is a noticeable point for the developer because some changes may be required by the customer. Therefore, the developer should be tolerant, flexible, and willing to re-create. After finishing this coding phase, the product is ready for testing.

3.5 Testing Phase
After developing the code, the product is tested to ensure the quality of the software, to meet the customer’s requirements, to reduce the cost of developing the program and to ensure that the product can actually solve the business issues in question. In fact, it measures the quality of the software the developer has written. The following testing steps apply in this phase.

3.5.1 Unit testing
This testing works at the lowest level. The fundamental work of unit testing is to take the smallest piece of testable software. Every unit will be tested separately before combining them into modules. It checks the basic unit of software which is often referred to as component, unit, or module.

3.5.2 Integration testing
This is the next rational step following the unit testing. This testing classifies the problems which appear when a combination of units has occurred. By the use of a testing proposal that you need to test every unit, some errors may unexpectedly be found during the research.
This testing can decrease the number of negative possibilities at analysis level. There are three approaches for the integration testing. They are the top-down approach, bottom-up approach and umbrella approach. Firstly, the top-down approach needs the highest-level units to be integrated and tested first. High-level logic and data flow are permitted by this approach for immediately checking in the process and this manages to reduce the need for drivers. Nevertheless, the disadvantage of the top-down approach in integration testing is the situation of low support for the release of restricted functions. Secondly, the bottom-up approach is different from the top-down approach. It needs the lowest-level units to be integrated and tested first. Nevertheless, this approach has a similar weakness to the top-down approach. It also keeps low support for the release of restricted functions. Finally, the umbrella approach tests the efficiency of the data and the paths of control flow. The inputs for the tasks will be integrated into the bottom-up design and the outputs for each task will also be integrated into the top-down pattern. The advantage of the umbrella approach has the advantage of the above approaches, giving support for the release of restricted functions. The disadvantage of this approach, however, is systematically lower than the other two approaches.

3.5.3 System testing
This is the testing step where a completed and integrated software is tested. It performs between integration testing and acceptance testing. This testing is created on the requirement specification of the system. The aim of this testing is to assess the action of compiling the system. Security, maintainability and reliability will be checked in this testing.

3.5.4 Acceptance Testing
This is the final level of the testing plan. After system testing, the completion of creating software is realised. This testing will work in the situation where the system is handed over from the developer to the customer or user. The main purpose of this testing is to gain the confidence of the customer by presenting a workable and reliable system for the user. Internal acceptance testing is checked by the people, the product manager or customer support, who are the team members of this project. External acceptance testing is checked by the people who are not the team members of this project. Customer acceptance testing and user acceptance testing are two types of external acceptance testing. Customer acceptance testing is checked by the people who want to develop a software to meet their requirements. User acceptance testing, also known as beta testing, is checked by the people who are the end users of the software.
3.6 Evaluation and Maintenance Phase
This is one of the most important phases of SDLC and is performed after the testing phase. This phase can be a combination of evaluation and maintenance for the project. Evaluation can measure how the developed software meets the user requirements and reaches the primary objectives of the project. The evaluation phase aims to improve the experience of the lessons learned in the current project. There are too many standards to measure the project. It depends not only on the customer requirements but also the conditions of the developed software such as ease of use, speed, reliability, robustness etc. When the user starts the software, the actual problems will appear. The system needs to be solved as soon as possible. This phase involves taking care of the developed software. It can involve upgrading the software, improving the system's performance, increasing security, correcting the problems and repairing anything that may be broken. Monitoring of the performance of the software is one of the duties of the maintenance phase.
4. Collecting Information and the Functions of Financial Department of Water Factory

4.1 Company Background
The name of this company I will research is A.M.T purified drinking water company limited. It is a medium-sized enterprise with more than 50 employees. This company was founded 10 years ago and it is situated in Number 1805, Pyi Htaung Su Road, 5 Quarter, Dagon Myothit (East), Yangon, Myanmar (Burma). The activity of this company is the purification, bottling and distribution of drinking water. In the modern day, companies are the system-based businesses. However, the financial department of this factory is using a paper-based system to collect the daily information and calculating with a human-based system. Therefore, this business needs to change from the document system to a computerised system.

4.2 Description of the people I interviewed
U Moe Tun, the owner of this water factory, is currently 50 years old. He studied only to grade 9 in secondary education. Although he doesn’t have a degree, he is an educated man. He worked as a technician in Malaysia for 4 years. He knows how to handle more than 50 employees and how to manage all the functions of this factory.
Most of the employees are the general workers and they are also not educated people, including the drivers in this factory. They have a language barrier to interview. Therefore, I questioned them using the Burmese Language and I translated all the questions and answers into English for this interview and research.
This section is to collect the information of the workers in the water factory in Myanmar. I went to the factory and have asked the following questions to the interviewees. I have interviewed 20 people among the workers and got their answers and feedback. However, I will mention only three of them for this dissertation.

4.3 Interviewee 1
Background questions
1. Name of the organisation
   A.M.T Purified Drinking Water
2. Job role
   Driver
3. Working experience (Yes/No)
   Yes
4. Overall feeling
Overall questions
1. How many years have you been here?
   7 years and a half
2. Do you think you will be here next one or two years?
   I think so.
3. In your experience, have you lost any document in this department?
   As a said, I am a driver and I do not work in the department but I have lost some receipts from the agent I have delivered.
4. If yes, how did you cover it?
   It is a little complicated. I phone the agents again. I think they are disturbed in their free time by my phone and I need to pay the calling charge. Sometimes, they do not like this condition.
5. In your experience, have you noticed any change in this department?
   No answer.
6. If yes, how do you want to change?
   No answer.
7. Are you using any special software for the financial department?
   No answer.
8. Do you want to work smartly?
   Yes, I do.
9. Are you happy with the document works?
   No, I am not.
10. Do you think it should be changed?
    Yes, I do.
11. How many hours do you work a day?
    Normally, 8 hours. If overtime, around 10 hours
12. Do you want to save time?
    Of course, I do.

Discuss department's needs
1. What is your department's requirements concern with the working process?
   No answer.
2. Did you discuss the requirements with your supervisor?
Discuss the problem
1. Do you have any other problem while you are working?
   Lost receipt, wasting time while they are calculating the amount
2. What is your main problem?
   I am an employee who earns money with the percentage of the orders. Thus, I need to save time.
3. Did you solve it?
   No, I did not.
4. Could you please explain me the functions of the financial department?
   No answer.

Discuss the change
1. Do you want to participate when the department is changed from the paperwork to the computerised system?
   Yes, I do. I am thinking about this for a long time.
2. Could you please give me any idea to change this department?
   I want the purchasing system with the electric payments.
3. Do you have any feedback for this interview?
   No, thanks. I am happy to answer you and I hope our factory will become better.

4.4 Interviewee 2
Background questions
1. Name of the organisation
   A.M.T Purified Drinking Water
2. Job role
   Auditor
3. Working experience (Yes/No)
   Yes
4. Overall feeling
   Great

Overall questions
1. How many years have you been here?
5 years and a half

2. Do you think you will be here next one or two years?
   Of course, I will.

3. In your experience, have you lost any document in this department?
   Yes, I have lost some documents and some are bitten by the mice.

4. If yes, how did you cover it?
   I just keep in the cupboard but I think it is not a real solution. If the owner wants to
   know the information last month, I need to find it because there are too many documents in
   the cupboard. We normally record both the daily inventory and the daily information of each
   employee.

5. In your experience, have you noticed any change in this department?
   Yes, I have.

6. If yes, how do you want to change?
   This department should change from document works to computerised works.

7. Are you using any special software for the financial department?
   Nothing. In the department, the electronic devices are a calculator and a phone.

8. Do you want to work smartly?
   Yes, I do.

9. Are you happy with the document works?
   Absolutely not.

10. Do you think it should be changed?
    Yes, I do.

11. How many hours do you work a day?
    Exactly 8 hours.

12. Do you want to save time?
    Yes, I do. It is the best question I have ever asked by myself.

Discuss department's needs

1. What is your department's requirements concern with the working process?
   I think we need a desktop computer to process the financial works.

2. Did you discuss the requirements with your supervisor?
   No, I did not.

Discuss the problem

1. Do you have any other problem while you are working?
Lost documents, waste time, sometimes wrong the calculation

2. What is your main problem?
   I want to cover all of them.

3. Did you solve it?
   No, I did not.

4. Could you please explain me the functions of the financial department?
   - record each employee’s information
   - record daily income
   - calculate and pay for percentage employee
   - in the payday of a month, calculate and pay for monthly employee
   - record the orders

Discuss the change
1. Do you want to participate when the department is changed from the paperwork to the computerised system?
   Yes, I do. I really want to participate in this change. What can I do for it?

2. Could you please give me any idea to change this department?
   I have no idea but I want my job to be smarter.

3. Do you have any feedback for this interview?
   I hope this research will help us.

4.5 Interviewee 3

Background questions
1. Name of the organisation
   A.M.T Purified Drinking Water

2. Job role
   The owner

3. Working experience (Yes/No)
   Yes

4. Overall feeling
   I have not satisfied yet.

Overall questions
1. How many years have you been here?
   10 years and a half
2. Do you think you will be here next one or two years?
   No answer.
3. In your experience, have you lost any document in this department?
   No answer.
4. If yes, how did you cover it?
   No answer.
5. In your experience, have you noticed any change in this department?
   No answer.
6. If yes, how do you want to change?
   No answer.
7. Are you using any special software for the financial department?
   No, we are not.
8. Do you want to work smartly?
   Yes, we do.
9. Are you happy with the document works?
   No, we are not.
10. Do you think it should be changed?
    Yes, we do.
11. How many hours do you work a day?
   It depends on the orders.
12. Do you want to save time?
   Absolutely yes.

Discuss department's needs
1. What is your department's requirements concern with the working process?
   Firstly, we have many orders daily. We need to collect them and inform the drivers. Then, we need to save this information to pay salary and calculate the monthly or yearly profit. Now, we are using the document for collecting the information. It has many risks such as losing document or information and wasting time.
2. Did you discuss the requirements with your supervisor?
   No answer.

Discuss the problem
1. Do you have any other problem while you are working?
   There are too many issues in this factory, especially, in the financial department.
2. What is your main problem?
   As I said, it is in the financial department. We need to save time, to get more profit, and to service the customers.
3. Did you solve it?
   No, I did not.
4. Could you please explain me the functions of the financial department?
   - collect orders and give to the drivers
   - pay salary
   - collect the employees’ information

Discuss the change
1. Do you want to participate when the department is changed from the paperwork to the computerised system?
   I really happy to participate.
2. Could you please give me any idea to change this department?
   I want to change my department from document to computerised system. And I want my business to be the best.
3. Do you have any feedback for this interview?
   I have no idea. I am very pleased to answer your questions. If you have any further questions, please feel free to come to my factory and collect any information. Thank you very much.

4.6 Summary of this Interview
After interviewing the owner and the employees, the following requirements for the financial department have been identified.

The audit needs to register the agents or customers. After registering them into the database, the audit can retrieve the data of the agents whenever the audit needs their information. The audit will also register the employees and record their daily attendance. The owner can also check the information of the employees. Since the software is recording the employees’ attendance, the audit can calculate the salary of every employee from the factory. The audit has to record the daily orders to calculate the daily income. Furthermore, the owner can check profit daily, monthly or yearly. The driver wants to use electronic payment to purchase the orders. It is a good idea but Myanmar is a developing country and electronic payment is not a popular method at the moment in Myanmar. Consequently, the following functions for the financial department will be implemented.
• Register the employees’ information of attendance and retrieve it when needed
• Register the customers’ information and retrieve it when needed
• Record the daily orders and calculate the daily income
• Calculate and pay the salary for the employees
• Check income and profit
5. Analysis and Suitable Methodology for Financial Department

5.1 Introduction
This section is crucial for choosing a suitable software development method for the project. The software development life cycle above serves as a basis for this selection. Firstly, this section will describe and define many real-world methods in current use, before outlining the advantages and disadvantages of each method. Finally, the reasons behind choosing and for what purposes will be examined. Having explored the options, one suitable methodology will be chosen for this project and the reasons for choosing it given.

There are three popular methodologies for creating software. They are Waterfall, Spiral and RAD (Rapid Application Development). They will be investigated and evaluated for this project in this section.

5.2 Software Development Methodologies
Software Development Methodologies are frameworks to structure software, to control the procedure of development and to get a qualify application for meeting customer requirements. After describing the methodologies, it can help to select a suitable method for this research. It can help to reduce the risk management and to provide an adequate process without wasting time.

The first stage involves establishing clear customer needs and requirements. Following this, there is a process to analyse the information from the customer and elaborate a design. After this process, the developers will start to write the code. This is called the development step. Then, the testing stage is reached where the code the developers have written is put on trial. After testing, the final stage of evaluation is implemented to evaluate the results from the feedback of customers and the end-users for the software in order to carry out software maintenance.
5.3 Waterfall Model

The Waterfall model is established as the earliest method in software development methodologies. The different names of this method are one-shot or once-through. It has a series of steps which are performed from top to bottom (Hughes and Cotterell, 2002, p. 63). It normally has five phases: customer requirements, design, implementation, verification or testing, and Maintenance or Evaluation, see figure 5.3.1. These activities are carried out in sequence from one stage to another. Each phase produces one output and this output will become input for the next phase (Bell, 2005, p.291). For example, the program design stage will output the coding stage and this stage will perform as a new phase to produce the testing phase.

The principles of the waterfall model
- It is a sequence of activities.
- Each phase will be defined systematically.
- Each phase produces a specific output.
- Each output becomes the input for the next phase.
- The perfection of each phase can be tested. (Bell, 2005, p. 292)

5.3.1 Advantages of the Waterfall Model
This model is easy to understand and easy to use. The benefit of waterfall development is that it allows for dividing into departments and controlling them. Every stage of development can be set with a time limitation. A well-structured technical design is already written before
the coding phase. It is easier for the new developer to change aspects within the development phase and it can help during the maintenance stage, too. As soon as the requirements have been approved, the total budget of the project can be correctly estimated. (Hughey, 2009)

5.3.2 Disadvantages of the Waterfall Model

The main risk of the waterfall model is the lack of changing requirements. If the customer wants to change something after collecting the requirements, this method cannot allow its procedure to be interrupted. It also does not allow a change of scope. It is difficult to cover for the unexpected risks. Bell (2015, p. 293) described as "When preparing a meal if you find that some ingredient is missing when you get to the stage of cooking the vegetables, you need to go back to the shopping stage". Therefore, the project manager needs to ensure the success each project in the design stage.

5.4 Spiral Model

The distinctive feature of the spiral model is the analysis of risk at many stages during a software development project (Bell, 2005, pp. 297-298). Bell (2005, p. 299) stated that "if meeting user requirements is identified as a potential problem, then the decision might be taken to carry out some prototyping to clarify the users’ needs”.

It normally contains four parts which are performing as the iterative process model, see in figure 5.3.2. They are planning or gathering customer requirement, risk analysis, development and verification or testing.

Bell (2005, p. 301) summarised that “the spiral model consists of a series of cycles. Each
cycle consists of a series of steps. At every cycle, any risks to the successful progress of the project are assessed. Then an appropriate method is selected in order to minimise that risk. Thus, the spiral model is essentially a cautious and robust approach to development”.

5.4.1 Advantages of Spiral Model

By the use of a high amount of risk management, the spiral model can be used to avoid the risk of the system or the project collapsing. As each spiral has a series of steps, there are no arguments between the previous requirements and the design for the development. Therefore, customers can add any additional requirements if they want to make any changes. In the development part, it can be split into smaller sections and it firstly develops the riskier elements with risk management evaluation. It can be a hugely powerful tool for large projects. By the use of strong documentation, the user can see the running outcome at the earlier stages of SDLC.

5.4.2 Disadvantages of the Spiral Model

This model performs too slowly so the customer needs to wait patiently to get the final product and somebody can leave from the development team (Bell, 2005, p. 298). A further pitfall is that project members cannot estimate the project deadline in the first stage. Thus, time management may be difficult. Finally, as this model uses a looping method, it has too many stages and uses a large amount of documentation (Gurendo, 2015). As the customers can add any requirements anytime, the designers may not be sure what they really need for the project. It is not suitable for smaller projects.

5.5 Rapid Application Development (RAD)

RAD is a model framework for the development of a project. This model uses a minimal plan
and providing the team is competent, it takes only a short period of time to create a product. Furthermore, the architecture of the project has to be well defined. McQuaid expressed that “One of the goals of RAD is to provide an updated ‘look and feel’ of the evolving product and to allow the client to have hands-on contact with the product as soon as possible”. According to Hambling (2000), "The ability of an information system to evolve to meet new requirements is one of the key quality characteristics, but systems must also be capable of rapid evolution if they are to deliver real value to the business community in this volatile business environment. Systems that cannot evolve rapidly offer little or no support to their users, who in turn become less responsive to their environment and consequently incur increased risk of failure in the marketplace".

5.5.1 Advantages of RAD
The changing requirements are similar to those of the Spiral Model. Therefore, the customers can add any requirements if they wish. By the use of a short development period, the customer can feel satisfied to see their product quickly and it is very compatible for smaller projects. It has a measurable progress. Powell-Morse (2016) expressed that "With frequent iterations, components, and prototypes coming down the pipe, progress on the overall project, as well as lesser segments, can be easily measured and evaluated to maintain schedules and budgets". It also has a simple adaptability: the developers from the team can shape the code without breaking. (Powell-Morse, 2016)

5.5.2 Disadvantages of RAD
It totally depends on having a strong team that can classify the customer requirements. Therefore, this method will need a skilful designer, developer, project manager, etc. It will not work for larger projects because of the relatively short development time. It demands frequent user interacting: "Gaining user insight and feedback early and often is certainly a benefit from a design perspective, but this double-edged sword requires that the team be both willing and able to communicate with users on a much more frequent basis, in comparison to a typical waterfall development method". (Powell-Morse, 2016)
5.6 The Suitable Methodology for the Project

After reviewing the above three methods for SDLC, all three have elements that could be beneficial to the project. The following table therefore clearly illustrates which method fits the project requirements best.

<table>
<thead>
<tr>
<th></th>
<th>Waterfall</th>
<th>Spiral</th>
<th>RAD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirement</strong></td>
<td>Cannot change</td>
<td>Can change</td>
<td>Can change</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>Difficult to measure the progress</td>
<td>Cannot predict and slow</td>
<td>Can predict and quick</td>
</tr>
<tr>
<td><strong>Design</strong></td>
<td>Highly structured</td>
<td>For risk management</td>
<td>Skilful designer</td>
</tr>
<tr>
<td><strong>Business</strong></td>
<td>Smaller</td>
<td>Larger</td>
<td>Smaller</td>
</tr>
<tr>
<td><strong>User review</strong></td>
<td>Till the end</td>
<td>Early</td>
<td>Early</td>
</tr>
</tbody>
</table>

After analysing the above table, RAD has emerged the most suitable model for the project rather than the waterfall or spiral methods.

Due to the quick development period, RAD will take around two or three months to accomplish the project. In the project scope, the objectives of the business are highly defined and limited, the project then is adequate with RAD. If the data is not complex and already exists for the project, it is then appropriate for this project. If the project team members are few yet skilful, RAD should be the choice for this project. If the technique requirements such as programming language, databases, response time, etc. are reasonable to create a software, RAD would be one of the best models for this project. SQA (2007) stated that “If project or development decisions can be made by a small number of people who are available and, preferably co-located, then it is suitable for RAD”. When the technical structure is well-defined and key points are tested in place, RAD is excellent to apply in this project. (SQA, 2007)
6. Design for the Project

6.1 Introduction
The design is essential for an application and should take into consideration the demand of user requirements and the use of technical requirements. It is a critical part of the development process for the project. Software design is one of the phases of SDLC. It is the process of choosing the requirements needed for developing in this project such as programming language, databases, etc., objects to convert as the variables for the coding, functions for the financial process and user interface (UI) to satisfy the users. This section will present how to structure the application that will be applied in the financial department in the question of this project. It will involve requirements for the project, choosing of the real-world objects to store into the database, the functions for processing, data flow design and a hand drawn the design for UI.

6.2 Requirements for implementing a software application
The software will need a programming language which is object-oriented, a database to store and retrieve the data and an IDE (Integrated Development Environment) that the programming language can run and support the UI for the project. For the programming language, Java has been chosen because it is an object-oriented language and it has many powerful IDEs. It can run on any platform and it is easy to implement a software application. As the requirement for IDE, NetBeans was chosen because it is also powerful and a reasonable platform for programmers. Moreover, it already has Java DB database and the programmer can create or add any database to implement the software. By choosing NetBeans therefore, we gain the advantage of having a built-in database to store and retrieve the user data.

6.3 Choosing of a real-world object from the customer requirements
This process is designed to select the objects that the program will definitely need. The customer is expected to add many objects for the purpose of the project. For example, the address is essential for both the information of the agent and employee. However, age may be required for the employee but not for the agent of the water factory. Before the database is created, the data needs to be collated and column names given in the tables. In this project, four tables to collect data are needed. Therefore, the database is named "Finance.db" and the tables are Agent, Employee, Income and Salary respectively.
6.3.1 Agent Table
This table will need five columns to save and retrieve data for the agent of water factory.
- ID – this is used as an agent identifier
- Name – this is used as the agent’s name
- NRC – this will represent the primary key for the agent
- Phone – to contact the agent
- Address – to contact the agent

6.3.2 Employee Table
This table is similar to the agent table. In this table, the user can store, retrieve and check the information of the employee.
- Name – this will be used for the employee’s name
- Phone – to contact the employee
- NRC – this will represent the primary key for the employee
- Address – this will be used as information of employee

6.3.3 Income Table
This table will be created for the information of daily income. In this table, the user can save and check the daily income for the factory.
- ID – this will be used as an order identifier
- Agent – this will be used to identify the agent of the driver
- Price – the amount of product from the factory
- Quantity – the number of products that the agent bought
- Total – the multiplied amount of price and quantity
- Date – this will be used as information of order date

6.3.4 Salary Table
This table will be created for the information of paying salary. In this table, the user can store and check the information of the salary of each employee.
- Name – this will be used as the employee’s name
- Amount – this will show the amount of daily payment for each employee
- Day – this records the total days that the employee worked in a month
- Total – this calculates the total amount of salary multiplied by amount and day
- Date – this is used as information of salary paid date
6.4 Functions of the financial department
As I mentioned in 4.6, the program will need the following functions for the financial department of water factory.

- Register the agent’s information
- Check the agent’s information
- Register the employee’s information
- Check the employee’s information
- Calculate the salary for the employees
- Check the salary record
- Record the daily income
- Check the daily income record

6.5 Data Flow Design
According to Bell (2005), data flow design is a method to carry out the structural design of software. It will present how the data flows through the processing stages. Before moving on to the next step, the data may transform in every step. When this data flow design is used for the software design, these steps will perform as software functions. (Sommerville, 2008) The following figure represents the data flow design for this project. In this illustration, how the user can save any data into the database and check or retrieve data from the database through the program can be seen, see the figure 6.1.
6.6 Hand Drawn Design
The above hand drawn design is also important for the design of the UI. It will show how to connect to each other and how to present the information as the UI to the user. Though there are many software applications that facilitate drawing the UI as a draft, hand drawing is still the best one to create UI.

6.6.1 The Main Form
On the main form, the main idea for this form is to connect with other forms. The user can see and click on nine buttons: six buttons are to connect with another form and one is to terminate the program, see figure 6.2. The remaining two buttons perform the functions of retrieving information from the database. One will work to check the salary record and one is to retrieve the income record.

![Figure 6.2](image)

6.6.2 The Register Agent Form
This hand draw design for registering an agent will perform to add an agent to the database after the user enters the information such as the ID, Name, NRC, Phone and Address. The first button is to return to the previous form (Main Form). The second button is to register the agent's information into the database. The third button is to terminate the program from the user view, see all in figure 6.3.
6.6.3 The Check Agent Form

This form aims to check the agent’s information which is already saved into the database after entering the information such as ID and NRC. There are two options for checking the information. The first is to check all agents from the factory and another one is to check only an agent. This agent’s ID and NRC need to be the same with the inputs that the user entered. The last button is to go back to the previous form, see in figure 6.4.
6.6.4 The Employee Register Form
This hand drawn design for registering an employee will perform to add an employee to the database after the user enters the information such as the Name, NRC, Phone and Address. The first button is to go back to the previous form (Main Form). The second button is to register the employee's information into the database. The third button is to terminate the program from the user view, see all in figure 6.5.

![Register Employee form](image)

Figure 6.5

6.6.5 The Employee Information Form
This form aims to check the employee’s information which is already saved into the database after entering the information such as Name and NRC. There are two options for checking information. The first is to check all employees from the factory and another one is to check only an employee. This employee’s Name and NRC need to be the same as the inputs that the user entered. The last button is to go back to the previous form, see figure 6.6.
6.6.6 The Salary Form

This hand drawn design for salary payment purposes to calculate each employee's salary after entering the information such as name, the amount per day, day per month. Here, the total is automatically calculated and shown after clicking button number 1. It will then save this information into the database. Button 2 is to go back to the previous form and button 3 allow the user to quit the program from the user view, see figure 6.7.
6.6.7 The Income Record Form

This hand drawn design for income record purposes to calculate for each income of each order after entering the information such as ID, agent's name, the price for the goods, the number of goods. Here, the total is automatically calculated and shown after clicking button number 1. It will then record each order income into the database. Button 2 is to go back to the previous form and button 3 allows the user to quit the program from the user view, see figure 6.8.

![Income Form Diagram]

Figure 6.8
7. Implementation of the Project

7.1 Introduction
Implementation is one of the phases of SDLC. It ensures that the longest time possible is taken in all phases to implement and elaborate an excellent program. During this phase, the program may be changed as a result of customer needs. If the questioner or interviewer accepts the requirements from the customer, then there is no change in this phase. This section will present how to transform the user requirements into the software. Firstly, this section will support the UML diagrams such as a class diagram, a use case diagram and a sequence diagram for creating the program. After the process of implementation of coding, the user manual will be described for software users. These UML diagrams are necessary to implement the software. These UML diagrams are not detailed yet.

7.2 Class Diagram
A class diagram is used for describing the object and information from the user requirements to the developer. It will contain many classes, classifiers, variables, attributes, operations, etc. Microsoft (2017) stated that “its classes and relationships can be implemented in many ways, such as database tables, XML nodes, or compositions of software objects”.

7.3 Use Case Diagram
This is similar to a data flow design. A use case diagram focuses on describing the user requirements. It will show the relationships between actors and requirements. The use case diagram can help to understand, discuss and communicate the customers’ needs. (Microsoft, 2017)
7.4 Sequence Diagram
A sequence diagram is used to represent the sequence of notes between actors, actions and objects from the project. Microsoft (2017) mentioned about its process, "it shows the flow of control from one participant to another. Use sequence diagrams to visualise instances and events, instead of classes and methods. More than one instance of the same type can appear on the diagram. More than one occurrence of the same message can also appear".

7.5 User Manual for the Financial Department
After developing a software, the developer normally creates a user manual how to use this application. This user manual will contain many instructions and the user can easily apply in the business. Therefore, the following instructions and the titles are presenting to ensure the ease of use for the users in the financial department.
7.5.1 Main Board or Main Form
When the user clicks the application, he or she sees the first form of this software. There are nine buttons on this form. They are Add Agent Button, Add Employee Button, Check Agent Button, Check Employee Button, Pay Salary Button, Check Salary Button, Record Income Button, Check Income Button and Exit Button.

1. The Add Agent button is for registering the agents for the factory into the database.
2. The Add Employee button is to add the employees for this industry into the database.
3. When the user clicks the Check Agent button, the user can see the agent information from the database.
4. If the user clicks the Check Employee button, the user can get the information of the employees from the database of financial department.
5. The Pay Salary button will perform the calculation of each employee from the factory and it will save the calculated information into the database.
6. By clicking the Check Salary button, the user can see detailed information of the paid salary.
7. If the user wants to record the information of daily income, the user needs to click the Record Income button.
8. The Check Income button can show detailed information of income for the factory.
9. The last button, the Exit, can terminate the application.

![Figure 7.1](image-url)
7.5.2 Agent Registration Form

When the user clicks the Add Agent button from the main form, this form will appear. There are three buttons on this form to manipulate it. When the user clicks the Back button, the user will reach the previous form. When the user clicks the Exit button, the program will be terminated. After entering the correct information, see figure 7.3, this form will register the agents into the database. Then, it will alert with a user information message which reads "Saved", see figure 7.4.

NOTE***
The user needs to enter only figures into the ID and Phone categories and only characters into the Name category. Either figures or characters may be entered into the NRC and Address categories. Otherwise, the program cannot run because of number format exception.
7.5.3 Check Agent Form
When the user clicks the Check Agent button from the main form, this form will appear. There are three buttons on this form to manipulate it. When the user clicks the Back button, the user will reach the previous form. This form will execute to find the information of the agents from the database. If the user clicks the Check All button from figure 7.5, it will alert the message box of information of the agents, see figure 7.6. If erroneous ID and NRC figures or characters are entered into the textboxes, see figure 7.7, it will warn with the message box that reads "We cannot find!", see figure 7.8., when the user clicks the Check button. After entering the correct ID and NRC into the textboxes, see figure 7.9, it will show detailed information about the agent who the user finds in the database, see figure 7.10.

![Figure 7.5](image1)

![Figure 7.6](image2)

![Figure 7.7](image3)

![Figure 7.8](image4)
7.5.4 Employee Registration Form
When the user clicks the Add Employee button from the main form, this form will appear. There are three buttons on this form to manipulate it. When the user clicks the Back button, the user will reach the previous form. When the user clicks the Exit button, the program will be terminated. This form will register the information of the employees for the factory into the database. There are three buttons on this form to manipulate it. When correct information is entered into the textboxes, see figure 7.11, it will be saved. When the user clicks the Submit button, it will then alert with a message which reads "Saved", see 7.12.

NOTE***
The user needs to enter only figures into the ID and Phone categories and only characters into the Name category. Either figures or characters may be entered into the NRC and Address categories. Otherwise, the program cannot run because of number format exception.
7.5.5 Employee Information Form

When the user clicks the Check Employee button from the main form, this form will be shown. There are three buttons on this form to manipulate it. When the user clicks the Back button, the user will reach the previous form. When the user wants to check all employees from the database, the user can click Check All button from figure 7.13. Then, a message box that contains all the detailed information of the employees from the factory will appear, see in figure 7.14.

If an erroneous Name and NRC is entered into the textboxes, see figure 7.15, it will warn with a message box that reads “We cannot find!”, see figure 7.16, when the user clicks the Check button. After filling the correct Name and NRC into the textboxes, see figure 7.17, it will show detailed information of the employee the user is searching in the database, see 7.18.
7.5.6 Salary Calculator Form

When the user clicks the Pay Salary button from the main form, this form will appear. There are three buttons on this form to manipulate it. When the user clicks the Back button, the user will reach the previous form. When the user clicks the Exit button, the program will be terminated.

After entering the information into the text boxes, see in figure 7.19, the user can calculate the salary of each employee by clicking the Calculate button, see in figure 7.20. It will then be calculated and shown beside the total. This information will also be saved into the database and a message box that reads "Saved" will alert the user about what he or she has done.

After calculating and saving the information of paid salary, the user can check the salary on the database. The user needs to click Check Salary button from the main form. It will then alert the user with a message box that involves detailed information of paid salary.
7.5.7 Income Record Form
When the user clicks the Record Income button from the main form, this form will appear. There are three buttons on this form to manipulate it. When the user clicks the Back button, the user will reach the previous form. When the user clicks the Exit button, the program will be terminated.

After entering the information into the text boxes, see in figure 7.24, the user can calculate the income of each order by clicking the Calculate button, see in figure 7.25. It will then be calculated and shown beside the total. This information will also be saved into the database and a message box that reads "Saved" will alert to know the user about what he or she has done.

After calculating and saving the information of each income, the user can check the income on the database. The user needs to click Check Income button from the main form. The user will be alerted by a message box that involves detailed information of each income.

![Figure 7.24](image1)
![Figure 7.25](image2)
![Figure 7.26](image3)

![Figure 7.27](image4)
![Figure 7.28](image5)
8. Testing of the Project

8.1 Introduction
Testing is one of the phases of SDLC. The developer or tester needs to apply it after developing software. According to Bell (2005, p. 267), every software needs to test and testing aims to produce fault-free software. This section will present two types of testing approaches. They are white box testing and black box testing. It will then test the software which is developed for this project and it will display the test plan.

8.2 White Box or Structural Testing
White box testing is a type of box testing of software. It is also known as the structural testing. It will test the internal coding and the system's infrastructure. This white box testing will execute every statement from the program during the testing time. According to Bell (2005), "this is equivalent to ensuring that every path through the program is executed at some time during testing. This includes null paths, so and if statement without an else has two paths and every loop has two paths". For example, Hamflett (2009-20110 stated that "there are two possible paths through the code. If A equals one or if A does not equal one. In order to trigger the two paths, the programmer would need to run two tests", see in the figure 8.1.1.

8.3 Black Box or Functional Testing
Black box testing is a type of box testing of software. As it applies only the functions of the program, it is also known as the functional testing, see in the figure 8.2.1. Therefore, it is about testing the functionality of the code. It will test all possible input data or functions. Once the program runs with the inputs, it will then see what occurs (Bell, 2005). This black box testing does not need to understand how the code works. Hamflett (2009, 2011) expressed that "we are not interested in how the black box performed the result. We are not interested in what paths the code is following. What we are interested in is if the functionality of the code is correct".

Figure 8.1.1

Two possible paths through this code
8.4 Test Plan for Developed Software
The developed is quite smaller than others so it does not need to much testing for this project. However, the following test plans are in each form. This testing will grasp outputs after filling the user inputs. Therefore, it can be the block box testing and it focuses on functional tests of inputs and outputs of the system. It does not need to have knowledge of internal software and it tests all possible data.

8.4.1 Test Plan for Main Form

![Figure 8.1](image)

<table>
<thead>
<tr>
<th>No</th>
<th>Input</th>
<th>Expected result</th>
<th>Actual result</th>
<th>Pass or Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Click add agent button</td>
<td>Appear agent register form</td>
<td>Appear agent register form</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>Click add</td>
<td>Appear agent register form</td>
<td>Appear nothing or appear</td>
<td>Fail</td>
</tr>
<tr>
<td>Step</td>
<td>Action</td>
<td>Expected Output 1</td>
<td>Expected Output 2</td>
<td>Result</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>--------</td>
</tr>
<tr>
<td>3</td>
<td>Click add employee button</td>
<td>Appear employee register form</td>
<td>Appear employee register form</td>
<td>Pass</td>
</tr>
<tr>
<td>4</td>
<td>Click add employee button</td>
<td>Appear employee register form</td>
<td>Appear nothing or appear another form</td>
<td>Fail</td>
</tr>
<tr>
<td>5</td>
<td>Click check agent button</td>
<td>Appear check agent form</td>
<td>Appear check agent form</td>
<td>Pass</td>
</tr>
<tr>
<td>6</td>
<td>Click check agent button</td>
<td>Appear check agent form</td>
<td>Appear nothing or appear another form</td>
<td>Fail</td>
</tr>
<tr>
<td>7</td>
<td>Click check employee button</td>
<td>Appear employee information form</td>
<td>Appear employee information form</td>
<td>Pass</td>
</tr>
<tr>
<td>8</td>
<td>Click check employee button</td>
<td>Appear employee information form</td>
<td>Appear nothing or appear another form</td>
<td>Fail</td>
</tr>
<tr>
<td>9</td>
<td>Click pay salary button</td>
<td>Appear salary calculator form</td>
<td>Appear salary calculator form</td>
<td>Pass</td>
</tr>
<tr>
<td>10</td>
<td>Click pay salary button</td>
<td>Appear salary calculator form</td>
<td>Appear nothing or appear another form</td>
<td>Fail</td>
</tr>
<tr>
<td>11</td>
<td>Click check salary button</td>
<td>Appear message that shows each salary record</td>
<td>Appear message that shows each salary record</td>
<td>Pass</td>
</tr>
<tr>
<td>12</td>
<td>Click check salary button</td>
<td>Appear message that shows each salary record</td>
<td>Appear nothing or it terminates the system</td>
<td>Fail</td>
</tr>
<tr>
<td>13</td>
<td>Click record income</td>
<td>Appear income record form</td>
<td>Appear income record form</td>
<td>Pass</td>
</tr>
</tbody>
</table>
### 8.4.2 Test Plan for Agent Register Form

<table>
<thead>
<tr>
<th>No</th>
<th>Input</th>
<th>Expected result</th>
<th>Actual result</th>
<th>Pass or Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input</td>
<td>Error message</td>
<td>Please fill the information!</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>Nothing</td>
<td>Error message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>3</td>
<td>Wrong input like String into ID or phone</td>
<td>Error Message</td>
<td>Your input does not meet the requirements. Please check your information again!</td>
<td>Pass</td>
</tr>
<tr>
<td>4</td>
<td>Wrong input like String into ID or phone</td>
<td>Error Message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
</tbody>
</table>
When the user clicks the register button without filling any data into the ID or name or NRC or phone or address, or all, the following error message that contains "Please fill the information!" will appear to the user, see in figure 8.2 and 8.3.
When the user clicks the register button after filling the wrong data type into the ID or phone, the following error message that contains "Your input does not meet the requirements. Please check your information again!" will appear to the user, see in figure 8.4 and 8.5.

When the user clicks the register button after filling zero and less than zero into the ID and phone number textboxes, the following error message that contains "Please check your id or phone! They must be greater than 0!" will appear to the user, see in figure 8.6 and 8.7.
8.4.3 Test Plan for Employee Register Form

<table>
<thead>
<tr>
<th>No</th>
<th>Input</th>
<th>Expected result</th>
<th>Actual result</th>
<th>Pass or Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input Nothing</td>
<td>Error message</td>
<td>Please fill the information!</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>Input Nothing</td>
<td>Error message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>3</td>
<td>Wrong input like String into phone number</td>
<td>Error Message</td>
<td>Your input does not meet the requirements. Please check your information again!</td>
<td>Pass</td>
</tr>
<tr>
<td>4</td>
<td>Wrong input like String into phone number</td>
<td>Error Message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>5</td>
<td>Right input</td>
<td>Saved information into the database and output message</td>
<td>Saved message</td>
<td>Pass</td>
</tr>
<tr>
<td>6</td>
<td>Right input</td>
<td>Saved information into the database and output message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>Step</td>
<td>Action</td>
<td>Expected Response</td>
<td>Actual Response</td>
<td>Outcome</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>-------------------</td>
<td>-----------------</td>
<td>---------</td>
</tr>
<tr>
<td>7</td>
<td>Less than or equal zero into Phone</td>
<td>Error message</td>
<td>Please check your phone! It must be greater than 0.</td>
<td>Pass</td>
</tr>
<tr>
<td>8</td>
<td>Less than or equal zero into Phone</td>
<td>Error message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>9</td>
<td>Click back button</td>
<td>Go to previous page or main page</td>
<td>Go to previous page or main page</td>
<td>Pass</td>
</tr>
<tr>
<td>10</td>
<td>Click back button</td>
<td>Go to previous page or main page</td>
<td>It does not go and terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>11</td>
<td>Click exit button</td>
<td>Terminate system</td>
<td>Terminate system</td>
<td>Pass</td>
</tr>
<tr>
<td>12</td>
<td>Click exit button</td>
<td>Terminate system</td>
<td>It does not terminate system or goes another form</td>
<td>Fail</td>
</tr>
<tr>
<td>13</td>
<td>Click submit button</td>
<td>Saved message</td>
<td>Saved message</td>
<td>Pass</td>
</tr>
<tr>
<td>14</td>
<td>Click submit button</td>
<td>Saved message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
</tbody>
</table>

When the user clicks the submit button without filling any data into the name, phone number or NRC or address, or all, the following error message that contains "Please fill the information!" will appear to the user, see in figure 8.8 and 8.9.
When the user clicks the submit button after filling the wrong data type into the phone number, the following error message that contains “Your input does not meet the requirements. Please check your information again!” will appear to the user, see in the figure 8.10 and 8.11.

When the user clicks the submit button after filling zero and less than zero into the phone number, the system will also display an error message with the same content, reminding the user to check their information again.
number, the following error message that contains “Please check your phone! It must be greater than 0!” will appear to the user, see in the figure 8.12 and 8.13.

![Employee Register](image1.png)  
![Message](image2.png)

Figure 8.12  
Figure 8.13

### 8.4.4 Test Plan for Check Agent Form

<table>
<thead>
<tr>
<th>No</th>
<th>Input</th>
<th>Expected result</th>
<th>Actual result</th>
<th>Pass or Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input Nothing</td>
<td>Error message</td>
<td>Please fill the information!</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>Input Nothing</td>
<td>Error message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>3</td>
<td>Wrong input like String into ID</td>
<td>Error Message</td>
<td>Your input does not meet the requirements. Please check your information again!</td>
<td>Pass</td>
</tr>
<tr>
<td>4</td>
<td>Wrong input like String into ID</td>
<td>Error Message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>5</td>
<td>Right input</td>
<td>Saved information into the database and output</td>
<td>Saved message</td>
<td>Pass</td>
</tr>
<tr>
<td>Step</td>
<td>Action</td>
<td>Expected Output</td>
<td>Actual Output</td>
<td>Result</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>-----------------</td>
<td>---------------</td>
<td>--------</td>
</tr>
<tr>
<td>6</td>
<td>Right input</td>
<td>Saved information into the database and output message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>7</td>
<td>Less than or equal zero into ID</td>
<td>Error message</td>
<td>Please check your ID! It must be greater than 0.</td>
<td>Pass</td>
</tr>
<tr>
<td>8</td>
<td>Less than or equal zero into ID</td>
<td>Error message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>9</td>
<td>Click check all button</td>
<td>Show message that contains all agent</td>
<td>Show message that contains all agent</td>
<td>Pass</td>
</tr>
<tr>
<td>10</td>
<td>Click check all button</td>
<td>Show message that contains all agent</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>11</td>
<td>Click check button</td>
<td>Show message that contains an agent</td>
<td>Show message that contains an agent</td>
<td>Pass</td>
</tr>
<tr>
<td>12</td>
<td>Click check button</td>
<td>Show message that contains an agent</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>13</td>
<td>Click back button</td>
<td>Go to previous page or main page</td>
<td>Go to previous page or main page</td>
<td>Pass</td>
</tr>
<tr>
<td>14</td>
<td>Click back button</td>
<td>Go to previous page or main page</td>
<td>It does not go and terminates system</td>
<td>Fail</td>
</tr>
</tbody>
</table>

When the user clicks the check button without filling any data into either ID or NRC, or both, the following error message that contains “Please fill the information!” will appear to the user, see in the figure 8.14 and 8.15.
When the user clicks the check button after filling the wrong data type into the id, the following error message that contains “Your input does not meet the requirements. Please check your information again!” will appear to the user, see in the figure 8.16 and 8.17.

![Figure 8.14](image1.png)  ![Figure 8.15](image2.png)

When the user clicks the check button after filling zero and less than zero into the id, the following error message that contains “Please check your ID! It must be greater than 0!” will appear to the user, see in the figure 8.18 and 8.19.

![Figure 8.16](image3.png)  ![Figure 8.17](image4.png)
8.4.5 Test Plan for Employee Information Form

<table>
<thead>
<tr>
<th>No</th>
<th>Input</th>
<th>Expected result</th>
<th>Actual result</th>
<th>Pass or Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input Nothing</td>
<td>Error message</td>
<td>Please fill the information!</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>Input Nothing</td>
<td>Error message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>3</td>
<td>Right input</td>
<td>Saved information into the database and output message</td>
<td>Saved message</td>
<td>Pass</td>
</tr>
<tr>
<td>4</td>
<td>Right input</td>
<td>Saved information into the database and output message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>5</td>
<td>Click check all button</td>
<td>Show message that contains all employee</td>
<td>Show message that contains all employee</td>
<td>Pass</td>
</tr>
<tr>
<td>6</td>
<td>Click check all button</td>
<td>Show message that contains all employee</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>7</td>
<td>Click check button</td>
<td>Show message that contains an employee</td>
<td>Show message that contains an employee</td>
<td>Pass</td>
</tr>
<tr>
<td>8</td>
<td>Click check button</td>
<td>Show message that contains an employee</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>9</td>
<td>Click back</td>
<td>Go to previous page or main</td>
<td>Go to previous page or main</td>
<td>Pass</td>
</tr>
</tbody>
</table>
When the user clicks the check button without filling any data into either name or NRC textbox, or both, the following error message that contains “Please fill the information!” will appear to the user, see in the figure 8.20 and 8.21.

![Employee Information](image_url) ![Message](image_url)

**Figure 8.20**  
**Figure 8.21**

### 8.4.6 Test Plan for Salary Calculator Form

<table>
<thead>
<tr>
<th>No</th>
<th>Input</th>
<th>Expected result</th>
<th>Actual result</th>
<th>Pass or Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input Nothing</td>
<td>Error message</td>
<td>Please fill the information!</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>Input Nothing</td>
<td>Error message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>3</td>
<td>Wrong input like String into amount per day or day per month</td>
<td>Error Message</td>
<td>Your input does not meet the requirements. Please check your information again!</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Expected Output</td>
<td>Actual Output</td>
<td>Result</td>
</tr>
<tr>
<td>---</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>4</td>
<td>Wrong input like String into amount per day or day per month</td>
<td>Error Message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>5</td>
<td>Right input</td>
<td>Saved information into the database and output message</td>
<td>Saved message</td>
<td>Pass</td>
</tr>
<tr>
<td>6</td>
<td>Right input</td>
<td>Saved information into the database and output message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>7</td>
<td>Less than or equal zero into amt/day or day/mth or greater than 32 into day/mth</td>
<td>Error message</td>
<td>Please check your amount or day! They must be greater than 0 and day must be less than 32.</td>
<td>Pass</td>
</tr>
<tr>
<td>8</td>
<td>Less than or equal zero into amt/day or day/mth or greater than 32 into day/mth</td>
<td>Error message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>9</td>
<td>Click back button</td>
<td>Go to previous page or main page</td>
<td>Go to previous page or main page</td>
<td>Pass</td>
</tr>
<tr>
<td>10</td>
<td>Click back button</td>
<td>Go to previous page or main page</td>
<td>It does not go and terminates system</td>
<td>Fail</td>
</tr>
</tbody>
</table>
When the user clicks the calculate button without filling any data into name or amount per day or day per month or all, the following error message that contains "Please fill the information!" will appear to the user, see in the figure 8.22 and 8.23.

![Figure 8.22](image1.png)  ![Figure 8.23](image2.png)

When the user clicks the calculate button after filling the wrong data type into either the amount per day or day per month or both, the following error message that contains "Your input does not meet the requirements. Please check your information again!" will appear to the user, see in the figure 8.24 and 8.25.
When the user clicks the calculate button after filling zero and less than zero into either the amount per day or day per month or both, the following error message that contains "Please check your amount or day! They must be greater than 0 and day must be less than 32!" will appear to the user, see in the figure 8.26 and 8.27.
### 8.4.7 Test Plan for Income Record Form

<table>
<thead>
<tr>
<th>No</th>
<th>Input</th>
<th>Expected result</th>
<th>Actual result</th>
<th>Pass or Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input Nothing</td>
<td>Error message</td>
<td>Please fill the information!</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>Input Nothing</td>
<td>Error message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>3</td>
<td>Wrong input like String into id or price or quantity</td>
<td>Error Message</td>
<td>Your input does not meet the requirements. Please check your information again!</td>
<td>Pass</td>
</tr>
<tr>
<td>4</td>
<td>Wrong input like String into id or price or quantity</td>
<td>Error Message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>5</td>
<td>Right input</td>
<td>Saved information into the database and output message</td>
<td>Saved message</td>
<td>Pass</td>
</tr>
<tr>
<td>6</td>
<td>Right input</td>
<td>Saved information into the database and output message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>7</td>
<td>Less than or equal zero into id or price or quantity</td>
<td>Error message</td>
<td>Please check your id or price or quantity! They must be greater than 0.</td>
<td>Pass</td>
</tr>
<tr>
<td>8</td>
<td>Less than or equal zero into id or price or quantity</td>
<td>Error message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td>9</td>
<td>Click back button</td>
<td>Go to previous page or main page</td>
<td>Go to previous page or main page</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>10</strong></td>
<td>Click back button</td>
<td>Go to previous page or main page</td>
<td>It does not go and terminates system</td>
<td>Fail</td>
</tr>
<tr>
<td><strong>11</strong></td>
<td>Click exit button</td>
<td>Terminate system</td>
<td>Terminate system</td>
<td>Pass</td>
</tr>
<tr>
<td><strong>12</strong></td>
<td>Click exit button</td>
<td>Terminate system</td>
<td>It does not terminate system or goes another form</td>
<td>Fail</td>
</tr>
<tr>
<td><strong>13</strong></td>
<td>Click calculate button</td>
<td>Saved message</td>
<td>Saved message</td>
<td>Pass</td>
</tr>
<tr>
<td><strong>14</strong></td>
<td>Click calculate button</td>
<td>Saved message</td>
<td>Appear nothing and it terminates system</td>
<td>Fail</td>
</tr>
</tbody>
</table>

When the user clicks the calculate button without filling any data into id or agent or price or quantity, or all, the following error message that contains “Please fill the information!” will appear to the user, see in the figure 8.28 and 8.29.

![Figure 2.28](image1)

![Figure 2.29](image2)

When the user clicks the calculate button after filling the wrong data type into the id or price or quantity, or all, the following error message that contains “Your input does not meet the requirements. Please check your information again!” will appear to the user, see in the figure
Figure 8.30 and 8.31.

When the user clicks the calculate button after filling zero and less than zero into id or price or quantity, or all, the following error message that contains “Please check your price or quantity! They must be greater than 0.” will appear to the user, see in the figure 8.32 and 8.33.

9.1 Introduction

Evaluation is key to the success of any project because it allows us to identify areas that need modification or correction and enables us to make recommendations and to elaborate a maintenance schedule.

9.2 Evaluation for this project

This developed software is quite powerful for small and medium businesses in developing countries. After using the software, they should notice the impact of applying technologies in the business. They will have more free time within the working period. This means they can dedicate time to have other ideas or do other tasks in the freed-up time or they may simply have a deserved break. Moreover, the owner can check the daily income from the software. The audit does not need to collect the documents of daily records to calculate a daily income. The owner can predict the profit by comparing the daily income records from the database. The audit does not need any documents for collecting the information from the factory. It means they can reduce the cost for the documents and it is covered from the stealing information, the loss of documents or in the event of fire.

By the use of software, the overall business will increase in some area. In the financial department of the water factory, the audit does not need to find documents and does not need to calculate the daily income. The audit can use only Microsoft Excel, which is a powerful tool for financial records. However, Excel is limited to accounting and cannot perform the later requirements, for example, electronic payments or the online purchasing system. Finally, they become smarter workers. This water factory business is a business which is applying technology and a business that saves time and reduces the need of paper documents.

9.3 Recommendation and Maintenance for this project

This section is one of the roles for the extending requirements in the financial department of the water factory. In this recommendation section, it will present the recommendation for the customer requirements and the recommendation of the developed software that will contain how to use the system safely and how to make a routine backup of the system.

According to the driver (see the first interviewee in 4.3 of this dissertation), he wants to use a purchase system with electronic payment. It is one of the risks for him. It is currently difficult to develop a software because the online payment is not popular in developing
countries such as Myanmar. In fact, the banks in Myanmar do not offer the facility of purchasing through electrical payment. I realised that the driver does not want to count the money and bring it back with him. According to the owner (see the third interviewee in 4.5), he wants to check the profit or income monthly and yearly. However, the developed software can only check for the daily income. Therefore, I need to create and develop these functions in a later version of the software for this factory.

In the developed software, the user does not need to install it. It can run on any platform, for example, Linux, Microsoft Windows and Apple Mac etc. The user needs to have secure access to the information of this factory. The user or the audit should not allow anyone else to use the computer because someone may delete the database file from this file or someone may input some query into the database by using this software. For this problem, in the later version, I can create an authentication for the user who must be either the owner or the auditor from the finance department.

Moreover, the user needs to make a backup every week or month for this database. I strongly recommend a backup file should be made every week. The user needs to be careful for the database file which is “Finance.db” in the dist file because it stores all the information of the factory such as agent information, employee information and daily income record. The user should buy extra space such as external hard disks or memory sticks (memory sticks can now store much data) or use the cloud service. Then, the user should create a backup file and save the database or the software into it. The most important one is the database. Even if the software has broken down by someone’s fault, it can be solved but if the database is lost, no one will be able to retrieve it. Subsequently, it is important to be aware not to permanently delete the database file from the software.
10. Conclusion
The study period at Cardiff Metropolitan University in the U.K. has been a very productive time both personally and academically. Living and studying in another country has been a very enriching experience that I shall never forget and researching and developing this dissertation has played a large part of that because it means I have crossed a big hurdle in my career. The fact that the project also looks at a methodology that could be successfully implemented within the SME sector in Southeast Asia is also a rewarding aspect. During the research period, there was considerable pressure to complete this dissertation. I have highlighted my points of experience as a conclusion of this dissertation.

10.1 Experience in questioning
I went to Myanmar during the December academic break to collect the information for the dissertation. Although I managed to meet many willing participants to interview about improvements to their factory, some were very late for the meetings and some people did not turn up at all. The interview process was quite stressful because participants found some of the questions difficult to understand even though I translated the questions into their native language. To overcome this, I avoided any technical jargon during the interview. After questioning about 20 people in one week, I collated the information.

10.2 Improvement of Language Skill
Writing a dissertation involves extensive reading and academic writing skills. Being an international student, I lack the mother-tongue advantages that British students have. After writing this dissertation, I have gained the confidence to read and write. As a software engineer, I prefer practical assessments to reports and only read essential books. However, for the purpose of this study, I needed to read many books, plenty of articles and the online websites. The final writing up of this dissertation has provided the opportunity to practice my academic writing skills.

10.3 Technical Experience
I chose my dissertation topic because it offered the possibility to create an application. There are many reasons for this choice. Firstly, the experience allowed me to polish my coding skill. Secondly, I was able to learn a new language and a new type of application. Finally, I would like to elaborate a special kind of software as a result of doing this research. In the past, I have normally done university projects without having any management. However, I have learned about Software Development Life Cycle (SDLC) in this research.
After studying the SDLC, I realised that I need to manage the project step by step. I understood that gathering customer needs and information clearly, designing before starting the code, implementing the design, testing the code, and evaluating and maintaining the produced software are all essential to obtain a quality software and to achieve or meet the user requirements.

For the software project, I need to write a program. Therefore, I needed to choose a programming language and platform to on which to write this language. I chose Java Programming Language and NetBeans IDE. This decision allowed me to get up to date with Java Language. As this research is for a financial department, I have gained much specific experience in writing a program for desktop applications.

While I was implementing the software, I firstly chose Java DB for the embedded database in NetBeans. Nevertheless, my database did not work in run time because I needed to add a client-derby jar file into the libraries. After sorting this problem out, I could not find how to import the jar file into the embedded database. Although I found it, it did not work again in other computers or platforms. I realised the reason for this was an error in the file path. Then, I tried to use another database which was SQLite. I did not need to embed the database and I could use the relative path. It also did not work in the program run time because of the relative path. Finally, I found the reason where I need to add the SQLite database. I needed to build and run to get the jar file and add that database which was being used in the program into the dist file.

To sum up, I have introduced the technique to the relevant people. This means that now they have understood how to use the software in the business and because of the special software designed for the financial department in the water factory they can begin to feel that technology in developing countries is beginning to catch up with the developed world. I have analysed the problems of the factory and chosen a suitable methodology to solve these problems. Now, they can cover most of the problems such as retrieving the information of salary and income. After reducing the cost in the financial department, they can invest in other areas. On the other hand, the profit of this factory can be increased. After changing from the document based work to a computerised system, their work became smarter than they were in the past. And the employee can be changed anytime because it the business now depends on the system and not on people. With the use of the software system, the finance department is running in the short time for calculating and saving information. Finally, doing this dissertation is a win-win situation for both the factory and myself because they have experienced many positive impacts of using a computerised system and I have developed through the experience such as researching, writing, reading and coding.
The demonstration of the software of this project is in the following link.
https://www.youtube.com/watch?v=bu9RRVyl1LA&feature=youtu.be
11. Reference


