A PRIVACY-PRESERVING SMART METERING SYSTEM

NYAM MONG LI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS TESIS

JUDUL: A PRIVACY-PRESERVING SMART METERING SYSTEM

SESI PENGAJIAN: 2015/2016

Saya NYAM MONG LI

mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ** Sila tandakan (/)

_______ SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

_______ TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

___/___ TIDAK TERHAD

Nyam
(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat tetap: LOT 152,
JALAN JERAM,
TAMAN SRI MANIR,
21200 KUALA TERENGGANU,
TERENGGANU.

EN. MOHD RIZUAN BIN BAHARON

Tarikh: 24/8/2016

Tarikh: 24/8/2016

CATATAN: * Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana Muda (PSM)

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

A PRIVACY-PRESERVING SMART METERING SYSTEM

NYAM MONG LI

This report is submitted in partial fulfilment of the requirements for the Bachelor of Computer Science (Computer Security)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERITI TEKNIKAL MALAYSIA MELAKA
2016

DECLARATION

I hereby declare that this project report entitled

A PRIVACY-PRESERVING SMART METERING SYSTEM

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : Nyam Date: 24|8|2016

(NYAM MONG LI)

I hereby declare that I have read this project report and found this project report is sufficient in term of scope and quality for the award of Bachelor of Computer Science (Computer Security) With Honours.

SUPERVISOR : Date: 24/8/2016

(MR. MOHD RIZUAN BIN BAHARON)

DEDICATION

Special thanks to my parents who supported and believed in me from the beginning of project until the end. Furthermore, I would like to dedicate this project to my supervisor, Mr. Mohd Rizuan Bin Baharon who has mentally and physically gave me a guidance to complete this project. Not to forget to thank all my friends who are willing to lend their hands to help me whenever I face problems in this project.

ACKNOWLEDGEMENTS

I would like to express my highest gratitude to my supervisor, Mr. Mohd Rizuan Bin Baharon for his immense interest in my topic of final year project, for providing me with materials and links that I could not possibly have discovered on my own, for his kind words and suggestions. Sir, words can never be enough to thank your kindness. It is a great honour to work under his supervision.

I am hugely indebted to other lecturers who are willing to spend their time to reply to my emails and guide me in implementing the system. Their precious kind advices help me to finish my final year project on time.

I would like to express my deepest thanks and sincere appreciation to my parents and friends for being constant source of motivation, kind endless help, generous advice and support until I completed my final year project.

ABSTRACT

Smart metering system is gaining more popularity as such a system improve energy consumption by the user. Through such a system, the users can monitor their daily energy usage. Nevertheless, such a system discloses user information to the processor, thus breach the privacy of the users. In order to prevent such a privacy breach, we propose a system which implements homomorphic encryption properties. The system is coded using Java language since it is a multiplatform compatible programming language. The methodology used is a waterfall model, which is a sequential manner that must be proceeds from one phase to next phase. Besides that, it emphasizes on documentation as well as source code. In my system, RSA (Rivest-Shamir-Adleman) encryption is used only for initial processing to secure the key for sharing purposes. The homomorphic encryption allows specific type of computations such as addition and multiplication on cipher-text without decryption. The purpose of this system is to provide security on data and protects content of message by helping it secure from unintended audiences although it is being processed by an aggregator. Hence, such a scheme prevents aggregator to disclose users' information to outsiders so as to guarantee users' privacy, confidentiality and integrity. Furthermore, this scheme prevents an attacker from tampering and altering the data since the data is in encrypted form.

ABSTRAK

Sistem meter pintar semakin mendapat perhatian kerana sistem ini dapat menambah baik penggunaan tenaga dalam kalangan pengguna. Melalaui sistem ini, pengguna dapat memantau penggunaan tenaga elektrik mereka. Walau bagaimanapun, sistem ini akan mendedahkan maklumat pengguna kepada pemproses, hal ini akan melanggar privasi pengguna. Untuk mencegah pelanggaran privasi, sebuah sistem yang mengguna ciri-ciri penyulitan homomorphic dicadangkan. Sistem ini ditulis dalam bahasa Java kerana Java merupakan satu bahasa pengaturcaraan yang boleh disokong oleh pelbagai platform. Metodologi yang digunakan ialah model *waterfall* di mana ia merupakan cara yang berurutan dari satu fasa ke fasa lain. Selain itu, ia memberi penekanan kepada dokumentasi dan sumber kod. Dalam sistem saya, penyulitan RSA (Rivest-Shamir-Adleman) digunakan sebagai permulaan pemprosesan untuk melindungi kunci atas tujuan perkongsian. Penyulitan *homomorphic* membenarkan pengiraan yang tertentu seperti penjumlahan dan pendaraban di atas tulisan rahsia tanpa penyahsulitan. Tujuan sistem ini ialah untuk memberikan keselamatan atas data dan melindungi kandungan mesej daripada orang luar yang tidak ada kaitan walaupun data tersebut diproses oleh pemproses. Oleh itu, skema ini dapat menghalang pemproses untuk mendedahkan maklumat pengguna kepada orang luar dan menjamin privasi pengguna, kerahsiaan dan integriti. Tambahan pula, skema ini mampu mencegah penyerang daripada menggangu dan mengubahsuai data kerana data telah disulitkan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ATTACHMENTS

CHAPTER I

INTRODUCTION

1.1 Introduction

A Privacy-preserving Smart Metering System (AP-SMS) using a homomorphic and RSA encryption scheme is proposed to replace the current system which is inefficient and to protect the privacy of smart grid users. In this project, data authority will generate both public and private keys using RSA encryption scheme. The public key will be given to an energy provider while the private key will be sent to a smart meter. The value of random prime *P* which is the secret key of the homomorphic encryption scheme is generated by the energy provider. Then, the energy provider encrypts *P* value using the public key that has been sent by the data authority. The smart meter uses the private key to decrypt the value of *P*.

Once the user entered their personal information and the hourly time in which they want to know the energy usage (*Wh*), the smart meter will encrypt the energy usage. Encrypted energy usage then will be sent to the aggregator to process the encrypted data. Since the energy generator that will decide the rate of *Wh*, the rate of *Wh* will be sent to the aggregator in an encrypted form.

The homomorphic encryption scheme is used by the aggregator to allow specific types of computations such as additions and multiplications to be carried out on cipher text and obtains an encrypted result. It allows the aggregator to compute arbitrary functions over encrypted data without the decryption key. Finally, the result of encrypted data is sent to the energy provider. The energy provider uses the prime *P* to decrypt the encrypted result. As a result, the energy provider is able to display the hourly bill to the user.

1.2 Problem Statement

Table 1.1: Summary of Problem Statement

No.	Problem Statement
1	Sending an hourly meter reading from a smart meter to energy provider in plaintext form attracts an attacker to get the information of the energy usage by the users.

1.3 Project Question

Table 1.2: Summary of Project Question

No.	Project Question
1	How can an RSA encryption scheme be used for protecting the key that has been shared between smart meter and energy provider?
2	How can a homomorphic encryption scheme guarantee user privacy, authentication and aggregated data integrity in smart grids?

3	How a smart meter cans eliminates the need for monthly inspections of meters by energy provider?

1.4 Project Objective

Table 1.3: Summary of Project Objective

No.	Project Objective
1	To protect a key shared by smart meter and energy provider using RSA algorithms.
2	To prevent an attacker from tampering and altering the data using a homomorphic encryption scheme.
3	To ensure users' information did not disclose to outsiders.
4	To allow energy provider automates hourly billing which can be monitored by the user, so as to reduce the energy consumption.

1.5 Project Scope

1) Data Authority

- Publishes both public and private keys.
- Public key will be sent to the energy provider.
- Private key will be sent to a smart meter.

2) Energy provider

- Energy provider generates a random prime *P* which is to be used in the homomorphic encryption scheme.
- Uses a public key to encrypt *P* and sent the encrypted *P* to the smart meter.
- Energy provider uses *P* and encrypts the rate of kWh, it sent the encrypted rate to the aggregator.
- Generates hourly billing to the user.

3) User

- User enters personal information and time hourly in which they want to know the energy usage (kWh).
- User sent the information in plain text to the smart meter.

4) Smart meter

- Smart meter uses the private key to decrypt encrypted prime *P*.
- Smart meter uses prime *P* and calculates the energy usage of a user in an encrypted form, the encrypted data is sent to the aggregator.

5) Aggregator

- Aggregator processes an encrypted data and sent it to the energy provider.

1.6 Project Contribution

Table 1.4: Summary of Project Contribution

No.	Project Contribution
1	Proposed an encryption scheme for smart metering system using RSA and homomorphic encryption scheme.
2	The billing data will not be stolen by a hacker or an unauthorized person

	because of this scheme will strengthen the security aspect to access the system.
3	Users always keeping in track of their hourly energy usage, thus less energy is consumed when the real-time meter readings are provided to the customers.

1.7 Thesis Organization

This section gives a summary of each chapter presented in this report which is introduction, literature review, project methodology, analysis and design, implementation and testing.

Chapter 1: Introduction

This chapter identifies the needs of users in smart metering system. This chapter also gathers all of related information from other works about the current and proposed system.

Chapter 2: Literature Review

This chapter discusses some methodologies that are being used in other research works which related to our smart metering system project. This chapter also discusses about the software and hardware that have been used in other research works which are related to this project.

Chapter 3: Project Methodology

This chapter reviews the project methodology and how it would be carried out. It describes each stage of the selected methodology and describe the activities that will be done in every stage and relate it with this project.

Chapter 4: Analysis and Design

This chapter defines the results of the analysis of the preliminary design and the result of the detailed design.

Chapter 5: Implementation

This chapter describes the activity involved in the implementation phase and what is the expected output after this phase is completed.

Chapter 6: Testing

This chapter describes the activity involved in the implementation phase in the project. It should consist of graphical results using the collected data from the implementation phase.

Chapter 7: Project Conclusion

This chapter summarizes the project by stating its objectives and describe how the objective can be achieved by integrating the information that have been reported in implementation and testing phase.

1.8 Conclusion

In a nutshell, hopefully this system will work well and efficient to help the smart meter user knows their hourly energy usage and billing. There are several types of cryptographic approaches that can be used to develop the system. Compared to other cryptographic approaches, RSA and homomorphic encryption scheme are more outstanding due to it guarantees user privacy, authentication and aggregated data integrity in smart grids. Hence, it can prevent the data from being tampered and altered by an attacker.

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

This chapter discusses some literature reviews based on the title of smart metering system. The domain of user privacy is discussed in this project. The chapter also discusses on algorithms of RSA and homomorphic encryption scheme, the explanation on both schemes is also provided. There will be a list of keywords that will be implemented in this project. Discussions on the existing project methodologies, techniques, parameters and software will be further discussed and compared to highlight the differences from one to another. Moreover, the justification of the selected technique is also provided in this project.

2.2 Related Work/ Previous Work

This section discusses about domain that is related to my project with given explanations and the several terms that being used for my project.

2.2.1 Domain

From both a technical and a legal perspective studies, smart meters can cause damage to user privacy. Aggregating individual metering data at a range of levels or by anonymizing high-frequency metering data through the use of pseudonymous identities is one of the suggestions of protecting user privacy.

Furthermore, a multiparty computation protocol could be used to allow multiple smart meters in a neighbourhood to compute a partial aggregation of their data without disclosing their individual measurements. Such a scheme that has been suggested by Garcia et al. implemented Paillier's additive homomorphic encryption.

However, a metering data aggregation scheme using Paillier homomorphic encryption is vulnerable to active attacks by intermediate malicious meters as the reason behind it is that granting the task of verification to each meter rather than to be collection unit although using aggregated data verification mechanism using a hash tree (Hur et al. 2015).

2.2.2 Keywords

1) **Homomorphic encryption:** Method of performing calculations on encrypted information without decrypting it, when decrypted, obtains the same results on the plaintext based on the operations performed.

2) **RSA:** Use two different but mathematically linked keys, one public and one private. The sender uses public key to encrypt the message, the receiver uses private key to decrypt the message.

3) **Plaintext:** It is a message that can be read clearly by human before encryption and after decryption.

4) **Ciphertext:** Result of encryption performed on plaintext using an algorithm, called a cipher.

5) **Key:** A parameter used to encrypt and decrypt the message.

6) **Encryption:** Process of converting the plaintext into ciphertext by encoding message that only allows authorized parties to read it.

7) **Decryption:** Process of converting the encrypted text back to plaintext which can be read by human.

8) **Smart meter:** An electronic device that keep track of the consumption of energy in 30 minutes or each hour, user will know the amount of billing and energy usage for the past hours.

9) **Symmetric Cryptosystem:** Using the same key for both the encryption and decryption process.

2.3 Critical Review of Current Problem and Justification

This section will do some comparisons on techniques, parameters and software that have been implemented in the existing works. Based on such comparisons, we have proposed a solution that will be discussed in next section.

2.3.1 Existing Techniques

There is several cryptography techniques can be used in developing cryptography application.

1) Homomorphic Crytography

According to Josep Domingo-Ferrer (2012), homomorphic encryption schemes can be described as “encryption transformations mapping a set of operations on cleartext to another set of operations on ciphertext.” Rivest, Adleman and Dertouzos were the one who first proposed the existence of a fully homomorphic scheme with the introduction of the RSA public encryption scheme.

A combination of homomorphic operators with every possible process can be executed on the cipher text without decrypting it is known as a fully homomorphic scheme. It shows the same result as the operators executed on the plaintext after decrypted .The ability to compute on ciphertext offers huge possibilities for privacy applications because most homomorphic algorithms are based on highly complex mathematical operations. Hence, it is computationally more complex than non-homomorphic asymmetric or symmetric algorithms(Zirm & Niedermeier n.d.)

Naehrig et al. have argued that a polynomial addition will take shorter time which is 1 ms and very cheap compare to a polynomial multiplication modulo which takes 11 ms. A polynomial modular multiplication should have some improvements on it to ensure it will have major effects on the efficiency of the homomorphic scheme (Naehrig et al. 2013).

In other work, Hur et al. have added that the weakness of this encryption scheme is malleable. Based on this scheme, an attacker could produce another ciphertext which decrypt to another meaningful plaintext in the same domain as original plaintext when it is given the ciphertext and public key. This can indirectly cause the smart meter to generate false and fake data, leading to inaccurate aggregation results (Hur et al. 2015).

2) Public Key Cryptosystem (Pailler, RSA and ELGamal)

Paillier cryptosystem is an additive homomorphic property and is on the basis of “decisional composite residuosity assumption (DCRA). Thus, it has various applications, for example, e-voting systems, threshold schemes, etc (V.Parmar et al. 2014).

On the other hand, RSA is a public key algorithm and is the exponential homomorphic encryption. The scheme’s difficulty is depending on factoring large integer on the security. It can be used for multiplicative property application such as to secure internet, banking and credit card transaction (V.Parmar et al. 2014).

Furthermore, ElGamal encryption algorithm is a public key algorithm and it is multiplicative homomorphic. It can be used in the application of Hybrid systems (V.Parmar et al. 2014).

Key size plays an important role to increase security. When there is an increasing bits of a key in key size, it is difficult to be susceptible to attack by brute-force attack. However, increasing bit slows down the cryptosystem. Therefore, it is crucial when choosing a right key size for each key.

Private key bit sizes are used according to NIST recommendation. Private key size of 1024 bit for RSA, 160 bits for ElGamal and Paillier was used for experimental purpose. This is because RSA provides same amount of security on 1024 bit key size as provided by Elgamal and Paillier on 160 bit.

The evaluation parameters for the above three encryption and decryption algorithms is encryption time, decryption time, throughput, encrypted file size and decrypted file size. RSA showed better performance over ElGamal and Paillier in encryption time whereas ElGamal had better performance over RSA and Paillier in decryption time. In addition, in terms of throughput, RSA has better throughput over ElGamal and Paillier in encryption process. On the other hand, Elgamal showed better throughput compare to RSA and Paillier in decryption process.

Among all of the algorithms, Paillier showed worst result in encrypted file size due to its encrypted file size increased exponentially with increase of input file size. In terms of parameters used in this experiment, it can be concluded that performance of RSA is better over ElGamal and Paillier (Farah et al. 2012).

3) Data Aggregation

He et al. and Li et al. have proposed the integrity preserving data aggregation schemes which is called as iPDA and EEHA using the concept of data slicing and assembling for wireless sensor networks. Three steps have been proposed by the authors. Firstly, by using the popular LEACH algorithm to construct an aggregation tree. Secondly, segmenting or slicing the data. Lastly, merging the pieces of the data at the aggregator and sending the merged data to a sink node. iPDA uses multiple aggregation trees by sending more than one copy of data to the destination to provide better integrity level (Alamatsaz et al. 2014).

2.3.2 Parameter/attributes

1) Private Key for homomorphic cryptography

For both encryption and decryption of secret message, symmetric cryptography uses only one key. There are two types of symmetric algorithms which are stream ciphers and block ciphers. Stream ciphers are used to encrypt each digit separately while a number of bits known as a block will be encrypted by block ciphers. It is depending on the algorithm used to determine the size of the block (Filip 2013).

Most schemes prefer to use symmetric key due to its low computing power required, the low computing delay and the low storage is needed due to the reduced key length which has typically from 128 to 160 bits (Alohali et al. 2015). But, the secret key is easily breakable in two ways which are using brute force on the key and discovering the key during initial key agreement. We can prevent such attacks by making sure the key is sufficient long (Filip 2013).

2) Private/Public key using public key cryptography

These algorithms use public key and private key as key pairing, so it is known as a public key cryptography (Filip 2013). The size of public keys and signatures in X.509 certificates (RSA_SHA256) is 1024 bit and 2048 bit (Nsiah et al. 2015). Those keys are related such that one key encrypts, and only the complementary counterpart can decrypt.

These algorithms provide protection with more computational cost than symmetric key schemes. Thus, such a scheme is not efficient as it requires more computational time. However, the advantage of using this algorithm is that it has longer life span because it needs more time to break the key value (Filip 2013).

2.3.3 Software

1) NetBeans IDE

This software is implemented in a Java program running on multiple laptops with Core 2 CPU. The computational power is important for the simulation because it shows an embedded system in a smart meter can meet the performance need for a real-world deployment of the proposed method(Hur et al. 2015).

2) **Matlab**

Omijeh has proposed smart metering system that was modelled into the system using the Matlab. Results obtained were very satisfactory(Omijeh 2013). Koay et al (2003) in their work have designed and implanted a Bluetooth energy meter which several meters can communicate wirelessly with a Master PC and work in close proximity. The distance coverage in this system is an issue because the Bluetooth technology only can work effectively at a close range(Omijeh 2013).

3) **OMNeT++**

OMNeT++ is written in C++ and is an open source for event simulation package. It has been designed to enable large scale simulation with hierarchichal and cuctomizable modules.But, due to its modularity, each module can be easily reused in any other module which leads to the development of diverse framework on top of simulation environment. (Robin n.d.)

2.4 Proposed Solution/ Further Project

2.4.1 Justification of Selected Techniques

1) **RSA**

RSA was proposed by Ron Rivest, Adi Shamir and Leonard Adleman who published their algorithm in 1977. In this algorithm, there are two different keys to be used which are a public key and a private key. In this cryptosystem, a public key is made public and can be accessed by anyone to encrypt message while a private key must be kept secretly and confidentially. The key is used to decrypt the ciphertext. RSA is based on the factoring problem which is a level of difficulty in factoring the product of two large prime numbers (Zirm & Niedermeier n.d.).

2) Homomorphic encryption

This encryption scheme allows one to calculate arbitrary functions over encrypted data without any decryption key. It converts data into encrypted data, then analysed it as it is in the original form. Most homomorphic algorithms are based on highly complex mathematical operations and are therefore computationally more complex than non-homomorphic asymmetric or symmetric algorithms but the ability to do operations on a ciphertext offers huge possibilities for privacy applications (Zirm & Niedermeier n.d.).

2.5 Conclusion

In a conclusion, we have reviewed some of the previous works that have been completed by other researchers. In developing a smart metering system, RSA and homomorphic encryption scheme have been selected and our approach on proposing a solution to the smart metering system will be further explain in chapter 3.

CHAPTER III

PROJECT METHODOLOGY

3.1 Introduction

Methodology is a system of methods used in a particular area of study or activity in which its purpose is to build an efficient and quality system that is easy to manage and is valuable to the organizations. Normally, methodology is used for investigating the concept of focal points. Thus, the methodology can be reused for many times and implemented it again in other system or application. In this proposed system, the methodology that is going to be used in this project is a waterfall model. The model is a sequential manner in which it must be proceeded from one phase to the next phase. It emphasizes on documentation as well as source code. There is a review at the end of each phase to determine whether each phase has completed and make decision either to allow the system to proceed or terminate.

3.2 Methodology

1) Waterfall methodology

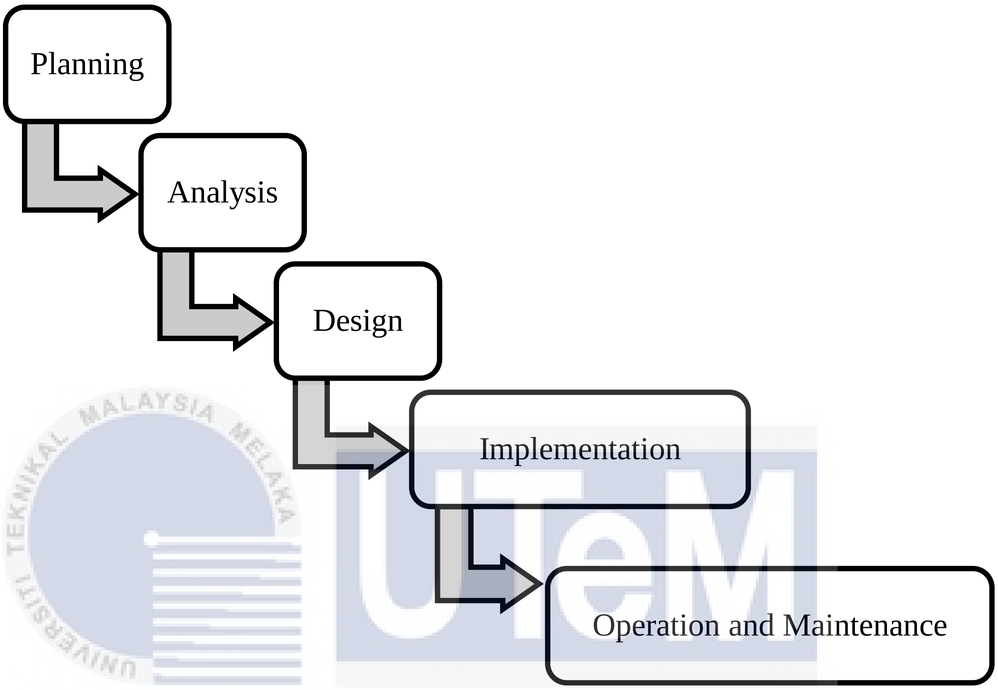


Figure 3.1: Waterfall Model Methodology

The waterfall model is shown as in Figure 3.1. A waterfall model has different function and tasks to be conducted on every phase of this methodology as described in Table 3.1.

Table 3.1: Description of each phase in a waterfall model

Phase	Explanation	Task(s)
Planning	Planning is conducted for the proposed system in the earliest stage.	-Identify the needs of existing smart metering system.

		-Gather some related information from journals and internet.
Analysis	During the planning phase, analyze the information that has been collected.	-Study and analyze the information collected. -Determine the weaknesses of the existing system. -List out the possible algorithms that can be implemented in the new system to improve the existing system.
Design	Using the hardware and software for this phase, then design the system interfaces that fulfill the requirements.	-Design interface. -Design coding.
Implementation	Installed the production system and train the users.	-Install the new system. -Provide training and guides for the users to use the new system. -Post-implementation review summary. -Distribute the user documentation.
Operation and Maintenance	Maintain the new system after implementation	-Maintain the new system by fixing the errors immediately once it

	phase is completed.	occurred. -To improve the system, monitor it and do the maintenance regularly.

3.3 Project Milestones

It is important to see the progress of the project that is doing and determine whether the project is on schedule to ensure it is able to meet the deadline of project development at the end. Thus, milestone is introduced in this project because it is a significant event marked on a timeline and can be used to monitor the progress of project to see how well the project is doing. Moreover, it is very helpful for planning and scheduling project. It is often create to ensure the project is complete towards the end of duration for developing a project. Project milestone is shown at Table 3.2.

Table 3.2: Project Milestone for PSM I and PSM II

No	Month/ Week Task Name	March			April					May				Jun		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Deciding the proposal title with the supervisor.	▨														
2	Preparing and submitting the		▨													

	proposal.		▨													
3	Preparing Literature Reviewing Report and Project Methodology Report.			▨												
4	Preparing Security Analysis Report (Analysis Phase).			▨ M1												
5	Preparing Security Report (Design Phase).				▨	▨ M2										
6	Preparing Security Implementation Reporting (Implementation Phase).						▨	▨ M3								
7	Planning and Testing (Testing Phase).								▨	▨	▨					
8	Preparing Security Testing Report.											▨	▨ M4			

9	Preparing Final Report.	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨		
10	Preparing Individual Log Book.	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨ M 5		
11	Final Presentation.														▨	▨

Planned milestones:

M1: Completion of Analysis phase

M2: Completion of Design phase

M3: Completion of Implementation phase

M4: Completion of testing phase

M5: Submission of Final Report and Individual Log Book

3.4 Conclusion

This chapter has described which methodology to be used in the proposed system. In this project, the system will be developed based on PPDIOO model throughout the entire project. Besides that, this chapter has also described the flow of the project by using milestone to coordinate and keep track of all the different tasks of the project. Such a technique is proven helpful in making sure that nothing falls behind schedule or off the radar entirely.

CHAPTER IV

ANALYSIS AND DESIGN

4.1 Introduction

This chapter discusses some weaknesses of current systems and the proposed solution to improve such weaknesses. To explain the flow and functions of the current systems, Microsoft Visio 2007 has been chosen as a tool to draw the flow chart of the system. Furthermore, this chapter describes some features of the system such as interface design, process diagrams, pseudo code and other documentations.

4.2 Problem Analysis

Problem analysis is a process to analyse problems that occur in the current system.

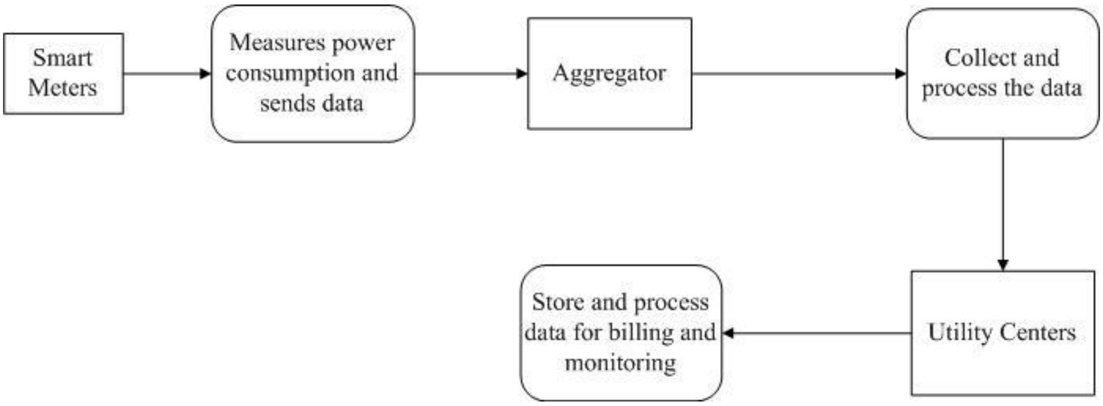

Smart Meters
Measures power consumption and sends data
Aggregator
Collect and process the data
Utility Centers
Store and process data for billing and monitoring

Figure 4.1: Overall flow of the current system

Figure 4.1 shows a data flow diagram. In the current metering system, a physical meter is easy to be compromised as the attacker can access its on-board memory through hacking, thus revealing diagnostic ports and other network interfaces. Examples of tools used by hackers are the SecureState like "Termineter" and the InGuardian such as "OptiGuard". Their purpose is to provide functionality for the C12.18 and C12.19 ANSI (American National Standards Institute) communication protocols. The C12.19 protocol enables for viewing of meter table data such as meter identity, operating mode and configuration mode.

On the other hand, the system is also vulnerable to brute force attacks that could discover the encryption key, for stealing the data. The attacker can gain access to physical frames of the network to ensure correct algorithms are used to decrypt the data. Side channel attack is another form of attack which exploits some aspects of a physical system that employs a data encryption algorithm.

4.3 Requirement analysis

4.3.1 Data Requirement

There are two data input requirements, which are plain text of message to be encrypted and keys that are used for the encryption process in the system.

1) **Text message to be encrypted**– In plain text format (including alphabets, numbers and symbols).

2) **Keys** – The keys that are used in RSA encryption scheme such as public key and private key for key sharing of common key used in Homomorphic encryption scheme. The common key is used to encrypt the plain text message.

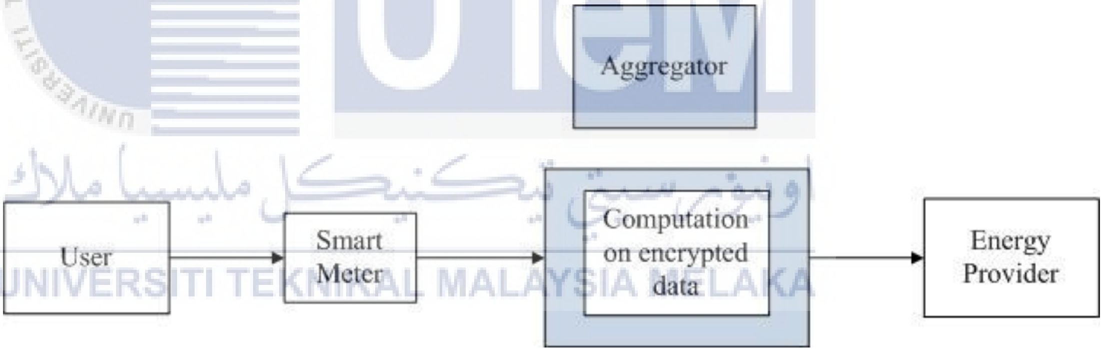


Figure 4.2: Flow chart

Figure 4.2 shows the flow chart of the process to allow user to enter the plain text message into user form in order to get the hourly energy usage. In this process, the smart meter reads the hourly energy usage for all electric appliances that have been used by user. Then, those data is sent to the aggregator to be computed in an encrypted form using operations such as addition and multiplication. The encrypted data then is sent to the energy provider to be decrypted to retrieve the total hourly energy usage and total hourly payment. The billing then is sent to the user for monitoring purposes.

4.3.2 Functional Requirement

Functional requirement captures the intended behaviour and functions of the system. This section discusses the process of system records, computes and transmits data. Figure 4.3 shows how the process in the smart metering system.

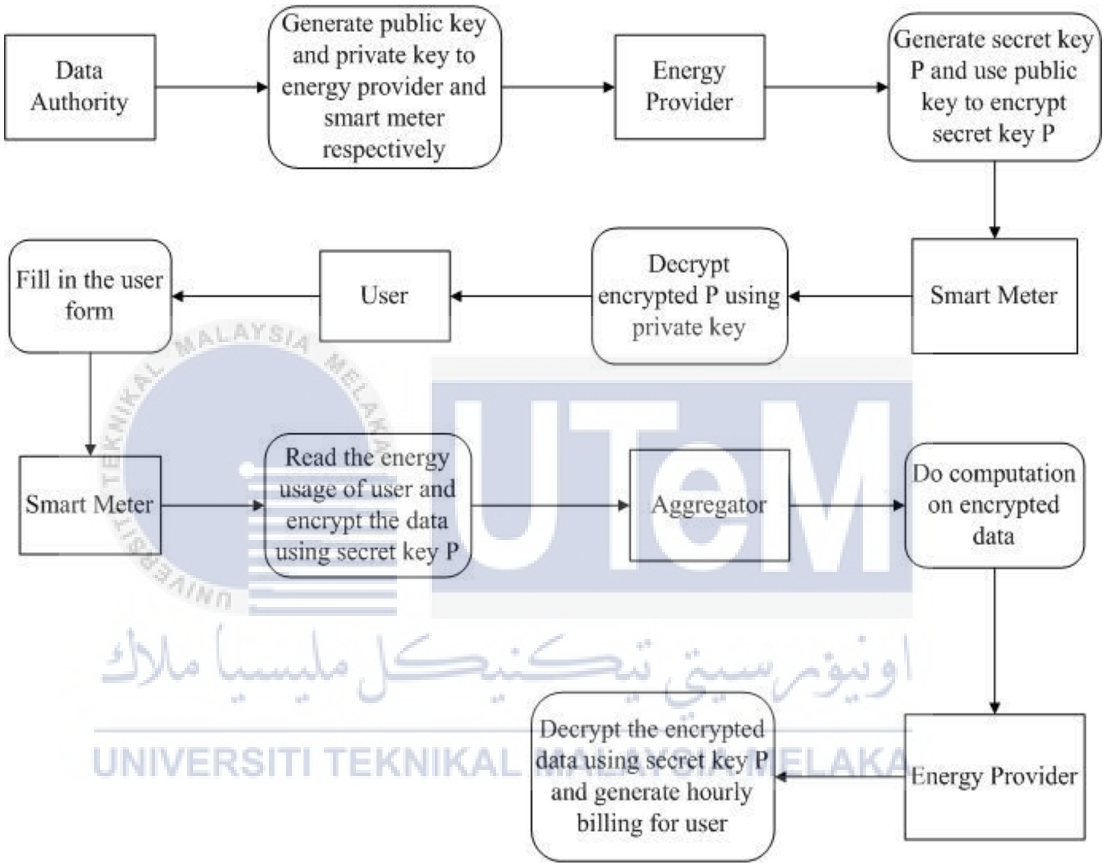

Data Authority
Generate public key and private key to energy provider and smart meter respectively
Energy Provider
Generate secret key P and use public key to encrypt secret key P
Smart Meter
Decrypt encrypted P using private key
User
Fill in the user form
Smart Meter
Read the energy usage of user and encrypt the data using secret key P
Aggregator
Do computation on encrypted data
Energy Provider
Decrypt the encrypted data using secret key P and generate hourly billing for user

Figure 4.3: Data flow diagram

4.3.3 Non-Functional Requirement

Software quality attributes such as performance, usability, efficiency and security-level requirements must be highly considered in a well-developed system. A non-functional requirement is a requirement that specifies how well the system performs its intended functions. It can be used to judge operation of system instead of specific function or behaviour.

The non-functional requirements for this smart metering system with RSA and homomorphic encryption scheme are shown in the Table 4.1.

Table 4.1: Non-functional requirements with description

Non-functional requirements	**Description**
Simple User Interface	The graphic user interface design of the system is simple and user-friendly. It allows the user to use the system smoothly and easily. Although the buttons labelled with descriptions according to their specific functions, the system should provide some guidance to the user when training them to use the system.
Error handling	When user enter wrong input to the system, the system will auto-detect and display meaningful error messages to keep them be informed about invalid input.
Performance	It takes a short time to process the input and generate output of the system, the system can be considered work fast.
Reliability	It is important to ensure that the decrypted data will get back the original value of plain text after undergo encryption and decryption process. The result must be correct and free from errors.
Usability	User does not need any knowledge when using the system. First time user will not encounter any problems when using the system because it is simple and easy to use.
Portability	The system size is small and it is convenience to

	save it in USB thumb drive and use it at anywhere.
Safety	Safety is assured when using the system. It will not create any damage or harm the user while the user is installing or running the application.

4.3.4 Other Requirements

To develop the system, there are some requirements on software and hardware.

4.3.4.1 Software Requirement

To ensure the project will be successfully developed, it is necessary for us to choose the most suitable software to be used during system development process. The list of software is shown as in the Table 4.2.

Table 4.2: List of software

Software	Description
Microsoft Window 8	Window 8 is an operating system that has been frequently used by current market system.
Microsoft Office Word 2010	It is a popular and trusted word processor tool. It is used in documentation of project such as processes, resources, methodologies and analysis outcome of the project.
Microsoft Office Visio 2007	It is a software diagramming program that uses vector graphic to create diagrams such as data flow diagrams.

SQLite	It acts as a database server for us to store the data inside the table of database and retrieve the value from it during the development of the system.
Eclipse IDE for Java Developers	It is a Java-based open source platform that allows us to code the system using Java programming language. It does a great job for programming applications in Windows platforms.

4.3.4.2 Hardware Requirement

There are some hardware used in this system and the requirements are listed in Table 4.3.

Table 4.3: List of hardware used

Hardware	Minimum Configuration
Processor	Intel Core i5-3337U Processor
Installed Memory (RAM)	4GB DDR3 RAM
Type of System	Genuine Windows 8
Monitor	Any Monitor
Keyboard and Mouse	Any Keyboard and Mouse
Network Card	Any Network Card
Printer	Printing Report

4.4 High-Level Design

High-level design describes the current architecture and project development that appear as a right model for coding. The architecture diagram gives an outline of an entire system. The initial step during application architecture phase is to study and examine all conditions for the proposed system. Further step is to determine the structure which can meet the requirements. High level design acts as a reference manual for how the modules interact at a high level, it covers the database architecture, application architecture (layers), application flow (navigation) and data flow architecture.

Below is the list of high level design aspects that should be included:

1) the implemented graphical user interface
2) the interfaces of software and hardware
3) the requirements for performance
4) the project architecture and design features

4.4.1 System Architecture

System architecture defines the fundamental and unifying system structure in terms of system elements, interfaces, processes, constraints and behaviours. Figure 4.4 shows the architecture of the system.

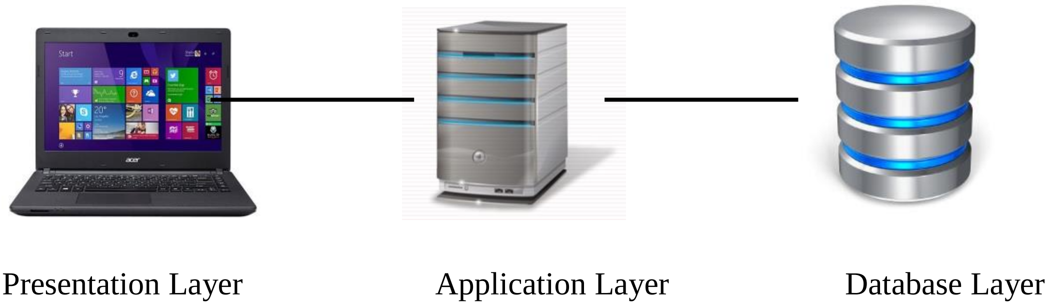


Figure 4.4: System Architecture

According to Figure 4.4, the architecture of the system consists of three layers which are presentation layer, application layer and database layer. A graphical user interface is for user interaction with the system to choose which function of button to run at the presentation layer. At the application layer, it processes the user input and undergoes the process of encryption and decryption, then return the correct value of output to the user. The database layer that has been used in this system is SQLite which is a self-contained, zero-configuration and transactional SQL database engine.

4.4.2 User Interface Design

User interface design (UI) concentrates on what users may need to do and ensuring that the interface has elements that are easy to access, understand and use for those actions. The concepts come from interaction design, visual design and information architecture.

4.4.2.1 Navigation Design

Navigation design plays an important role for the users as a road map for them to gain access to the system effortlessly. Besides that, it aids the user to understand the system flow clearly and accurately. Thus, the navigation components must be easy to ensure the users without requiring scrolling down to navigate further into the system. It should deliver a friendly using experience to the users.

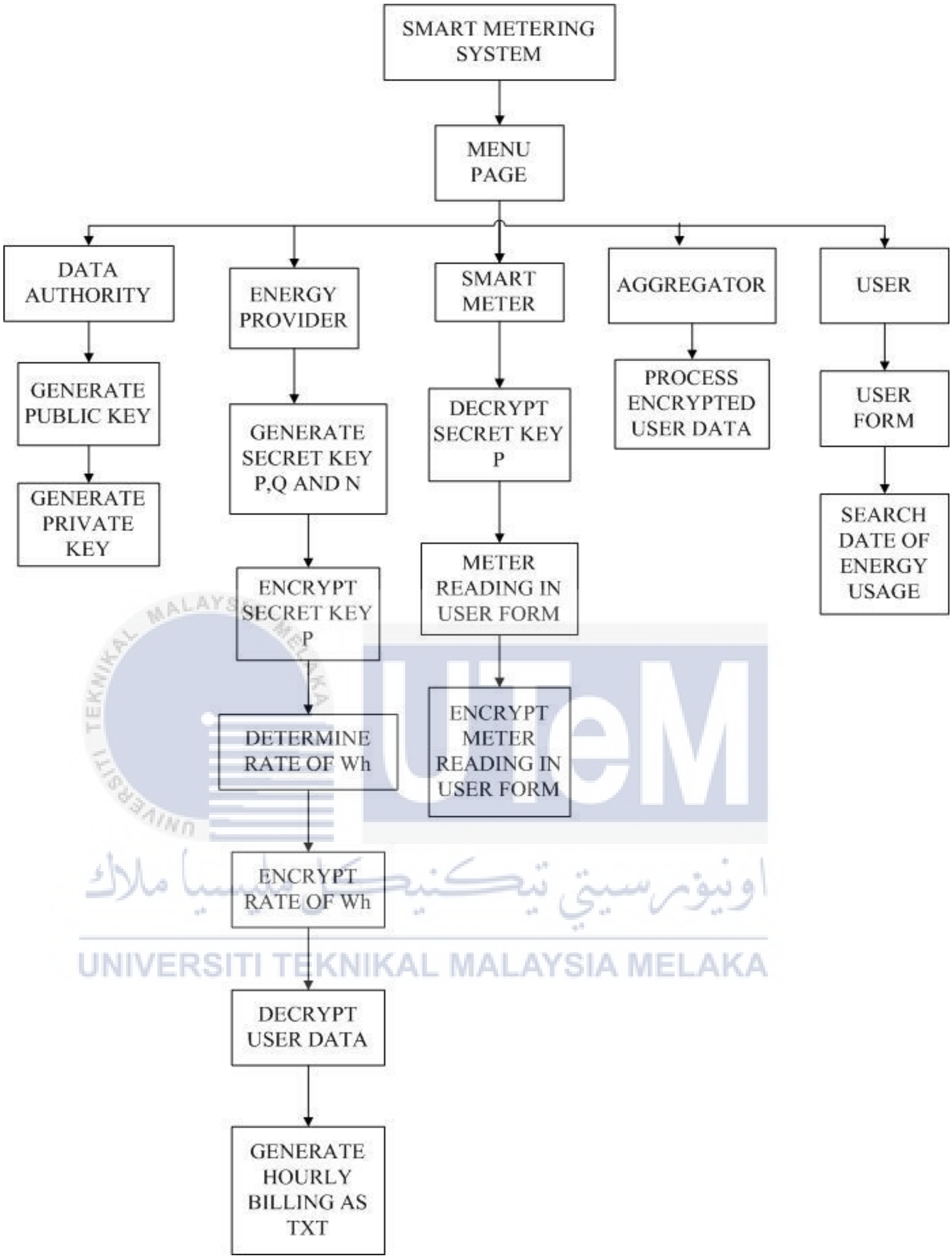


Figure 4. 5: Navigation Design

4.4.2.2 Input Design

There are several component designs implemented in this system, which are JTextField , JComboBox and JButton for confirmation. The designs are illustrated as below.

1) JTextField

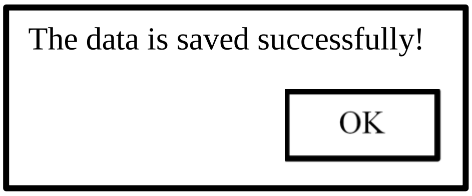
2) JComboBox

3) JButton for confirmation

4.4.2.3 Output Design

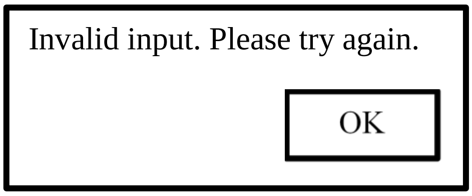
Output design allows the user to know the result by displaying components. The output designs are shown as below.

a) Successful message



The data is saved successfully!

b) Error message



Invalid input. Please try again.

4.4.3 Database Design

Database design is the process of coming out with a cleared and detailed data model of database. It must have all the required logical and physical design options. To create a database, physical storage parameters are required to produce a design which fulfilled the data definition language. A good database design will make us gain access to latest and correct information, it is necessary to help us achieve our objectives when dealing with a database.

4.4.3.1 Conceptual Database Design

Conceptual database design involves creating a conceptual schema for collected information at database using a high level conceptual data model. It gives us a clear solution to convert from high-level model to relational model. Conceptual schema is a long-lasting description of database conditions. The design is as shown in Figure 4.6.

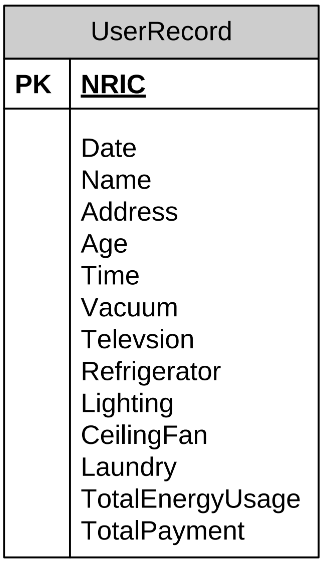


Figure 4.6: Entity-Relationship Diagram

4.4.3.2 Logical Database Design

To translate the conceptual design to logical design, the selected database management system is SQLite database software. The main purpose of logical database design is to produce a well-structured and arranged table that properly reflect the project system. To describe the Figure 4.6 above and the script of query, data dictionary is used and shown in Table 4.4.

Table 4.4: Data Dictionary

4.4.3.2.1 UserRecord

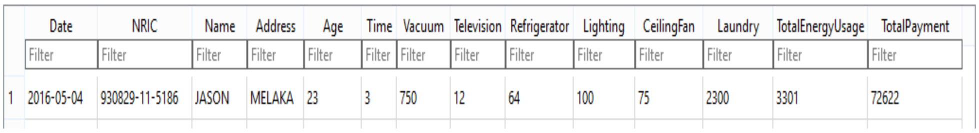
Attribute	Data Type	Description	PK/FK	References
Date	varchar2(10)	Date for user to search		
NRIC	varchar2(14)	User’s name	PK	
Name	varchar2(50)	User’s name		

Address	varchar2(50)	User’s address		
Age	varchar2(2)	User’s age		
Time	varchar2(2)	Time for user to search		
Vacuum	varchar2(10)	User’s energy usage on vacuum		
Television	varchar2(10)	User’s energy usage on television		
Refrigerator	varchar2(10)	User’s energy usage on refrigerator		
Lighting	varchar2(10)	User’s energy usage on lighting		
CeilingFan	varchar2(10)	User’s energy usage on ceiling fan		
Laundry	varchar2(10)	User’s energy usage on laundry		
TotalEnergyUsage	varchar2(10)	Total energy usage of users on the appliances		
TotalPayment	varchar2(10)	Total payment of users on the appliances		

4.4.3.2.2 Query Design

1) Retrieve payment information on 2016-05-04 and NRIC on 930829-11-5186

```
SELECT *
FROM UserRecord
WHERE Date LIKE ‘%2016-05-04%’ AND NRIC = ‘930829-11-5186’;
```



	Date	NRIC	Name	Address	Age	Time	Vacuum	Television	Refrigerator	Lighting	CeilingFan	Laundry	TotalEnergyUsage	TotalPayment
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	2016-05-04	930829-11-5186	JASON	MELAKA	23	3	750	12	64	100	75	2300	3301	72622

4.5 Detailed Design

Detailed design is the final design activity before implementing the system. It shows the flow of the whole system design and how the system works. Although it is still an abstraction compared to source code, it must be comprehensive to verify that translation to source is accurate mapping rather than rough draft of interpretation.

4.5.1 Software Design

Software design is a process to transform requirements into appropriate form which assists us during software coding and implementation. It discusses in depth for every component and operation coded in the system that interacts with one another. Pseudo code will describe about input/output parameter and the way algorithm processes them.

Interface name : Smart Metering System using RSA and homomorphic encryption scheme GUI

Responsibility : Main and self-contained interface for message encryption, calculation of encrypted data and decryption of encrypted data

Methods : User enter the date of energy usage and National Registration Identity Card (NRIC), the results will display for hourly total energy usage and hourly billing of payment

Input parameter :

(For Message Encryption)

- message: encrypt plain text message using RSA and homomorphic encryption scheme
- key for RSA: public key (used to encrypt key sharing of secret keyP)
- key for randomization: fixed value and random value
- key for homomorphic encryption: public key *N*

Output parameter:

(For Message Decryption)

- message: decrypt encrypted message using randomization and homomorphic decryption scheme
- key for RSA: private key (used to decrypt secret key *P*)
- key for randomization decryption: fixed value
- key for homomorphic decryption: secret key *P*

Pseudo code:

```
1.0 Start application;
    1.1 Click public key button of data authority
        1.1.1 Generate value for public key and small exponent $e$
    1.2 Click private key button of data authority
        1.2.1 Generate value for private key
    1.3 Click button secret key $P$,$Q$ and public key $N$ of energy provider
```

1.3.1 Generate value for secret key *P*,*Q* and public key *N*

1.4 Click button encrypt secret key *P* of energy provider

1.4.1 Generate encrypted secret key *P* using RSA public key of data authority

1.5 Click button decrypt secret key *P* of smart meter

1.5.1 Generate decrypted secret key *P* using RSA private key of data authority

1.6 Click button determine rate of *Wh* of energy provider

1.6.1 Generate rate of payment (*cents*) for every unit of *Wh*

1.7 Click button encrypt rate of *Wh* of energy provider

1.7.1 Generate encrypted rate of *Wh* using randomization and homomorphic encryption scheme

1.8 Click button user form of user

1.8.1 User fills in the personal information, data and time of energy usage to be searched

1.9 Click button meter reading in user form of smart meter

1.9.1 Display meter reading for electric appliances that has been used by user for the past hour

2.0 Click button encrypt meter reading in user form of smart meter

2.0.1 Generate encrypted meter reading for electric appliances that has been used by user for the past hour using randomization and homomorphic encryption scheme

2.1 Click button process encrypted user data of aggregator

2.1.1 Aggregator will do computation on encrypted data such as addition and multiplication

2.2 Click button decrypt user data of energy provider

2.2.1 Energy provider will decrypt the encrypted data that sent from aggregator, plain text of user's personal information and meter reading value is displayed.

2.3 Click button generate hourly billing as TXT of energy provider

2.3.1 A txt file will be generated and showed lists of records that have been used by all users for the past hours.

2.4 Click button search date of energy usage of user

2.4.1 A list of records that have been used by the user that differentiate by NRIC and date of energy usage for the past hours will be displayed.

2.5 End of application.

4.5.2 Physical Database Design

Physical database design is translating the logical description of data into the technical specifications for storing and retrieving data. It is very important to create a design for storing data that will provide sufficient performance and guaranteed database integrity, security and recoverability. It shows the script of create table in SQLite at below section:

4.5.2.1 UserRecord

```
create table "UserRecord"
("Date" varchar2(10) not null enable,
 "NRIC" varchar2(14) ,
 “Name” varchar2(50) not null enable,
 “Address” varchar2(50) not null enable,
 “Age” varchar2(2) not null enable,
 “Time” varchar2(2) not null enable,
  “Vacuum” varchar2(10) not null enable,
  “Television” varchar2(10) not null enable,
  “Refrigerator” varchar2(10) not null enable,
  “Lighting” varchar2(10) not null enable,
  “CeilingFan” varchar2(10) not null enable,
  “Laundry” varchar2(10) not null enable,
  “TotalEnergyUsage” varchar2(10) not null enable,
  “TotalPayment” varchar2(10) not null enable,
   primary key (“NRIC”) enable);
```

4.6 Conclusion

This chapter explained the system design to meet the requirements and needs in the design phase. It has successfully described the system design and that can be used as input to system development in next phase.

CHAPTER V

IMPLEMENTATION

5.1 Introduction

In the implementation phase, our project executes its plan, idea, design and algorithm through the development and construction of the real project outcome. We enhance the existing source codes to be used in developing our system. Furthermore, we implement the results of design phase to build our system based on the documents from design phase and analysis phase. Though we emphasize on the quality and performance of the system, some improvements on the flexibility and requirement changes are still allowed to the system. This phase also explains about the method for setting up the system software environment and the procedures for its configurations.

5.2 Software Development Environment Setup

This sub-section describes a deployment diagram to visualise a hardware topology of A Privacy-preserving Smart Metering System (AP-SMS). The diagram explains the deployed system hardware component topology and the software components. In addition, the diagram is often used to depict the rigid development view of system. Thus, it contains nodes and their relationships in the deployment diagram. A smart deployment diagram has the ability to manage the parameters of the system being developed such as performance, scalability, maintainability and portability.

Eclipse software has been chosen to code the system using Java programming language. Since Eclipse is an integrated development environment (IDE) that frequent used in computer programming, it is necessary to setup a good development environment for it to be function efficiently. To run Java application in Eclipse, a Java virtual machine (JVM) is needed to allow a computer to run Java program. For this project, Java Development Kit (JDK) 1.6 is installed in Window 8. Then, the smart metering system will use Java Runtime (JRE) library JAR files from JDK distribution to compile, debug and run the Java program. Eclipse supports JDK 1.6 which is a built-in Java compiler. Figure 5.1 shows the deployment diagram.

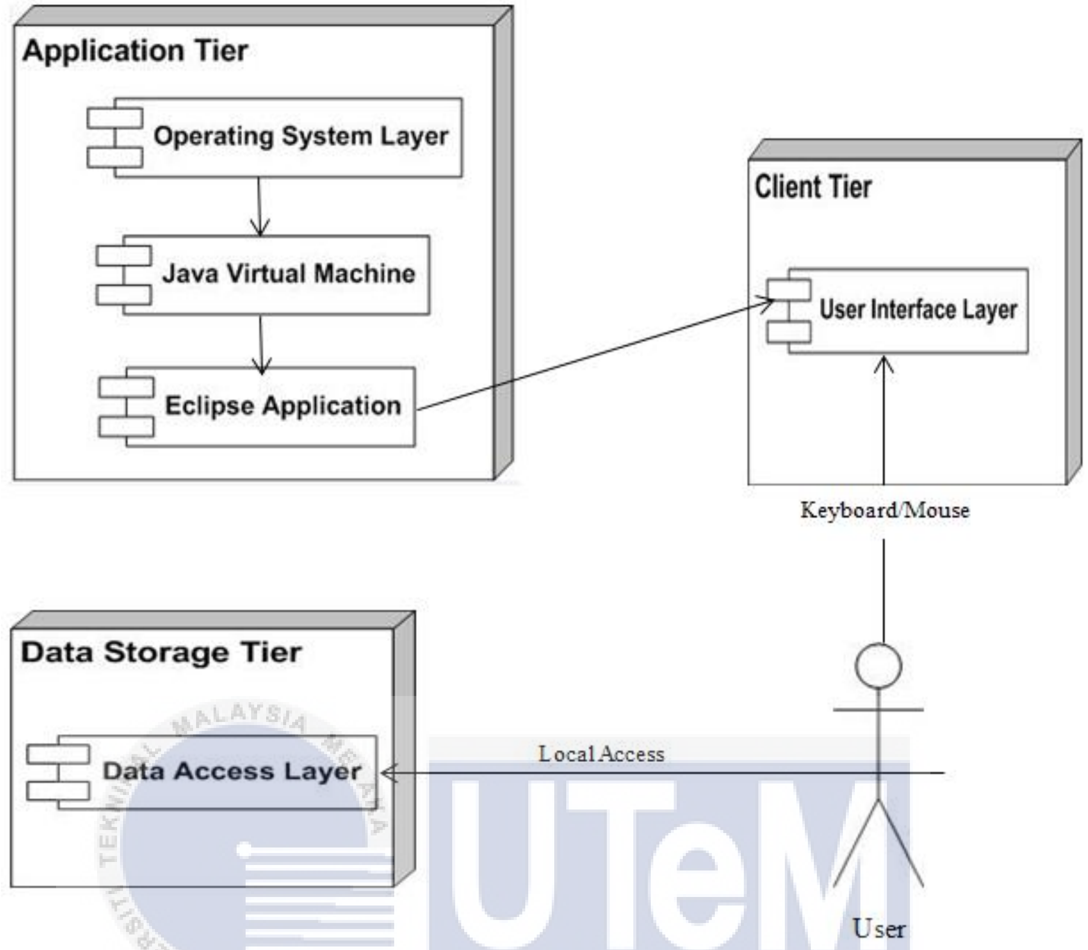


Figure 5.1: Deployment Diagram

Figure 5.2 illustrates the software architecture for AP-SMS. The figure describes the relationship between system controller and SQLite. The information of AP-SMS that has been written with Java programming language, it will be store in SQLite database to be retrieved for further usage and vice versa.

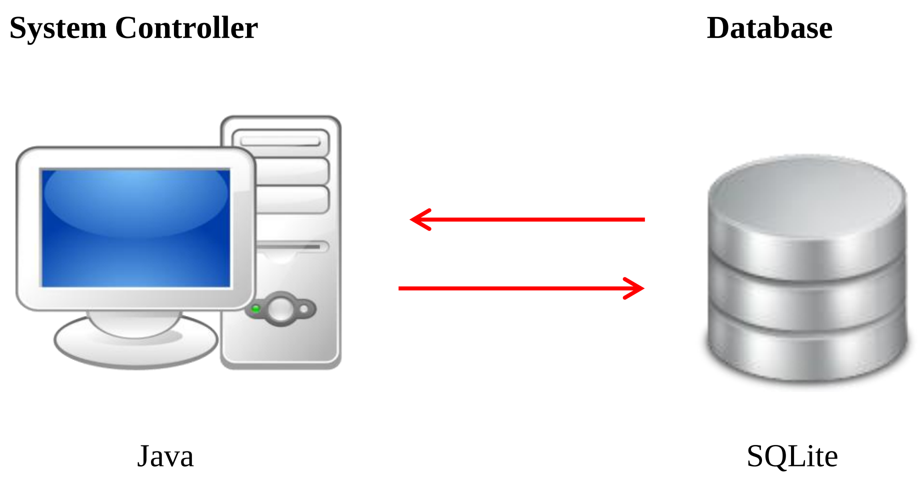


Figure 5.2: Software Architecture of AP-SMS

5.3 Software Configuration Management

Many organizations often used software configuration management (SCM) which is a software engineering discipline that contains standard techniques and processes to operate the changes in the software products. Its role is to identify individual elements, keep track of the changes through trusted version selection, control and baseline. It ensures the integrity of software is not being degraded or destroyed by changes made to current work products. Steps for SCM include:

1) identifying the components and related work products that are probably to change;
2) controlling work products by establishing and maintaining relationships among them;
3) monitoring and keeping track of the change requests for items of configuration;
4) managing changes in the configuration items that have been implemented;
5) performing configuration auditing and report writing on the changes that have been made.

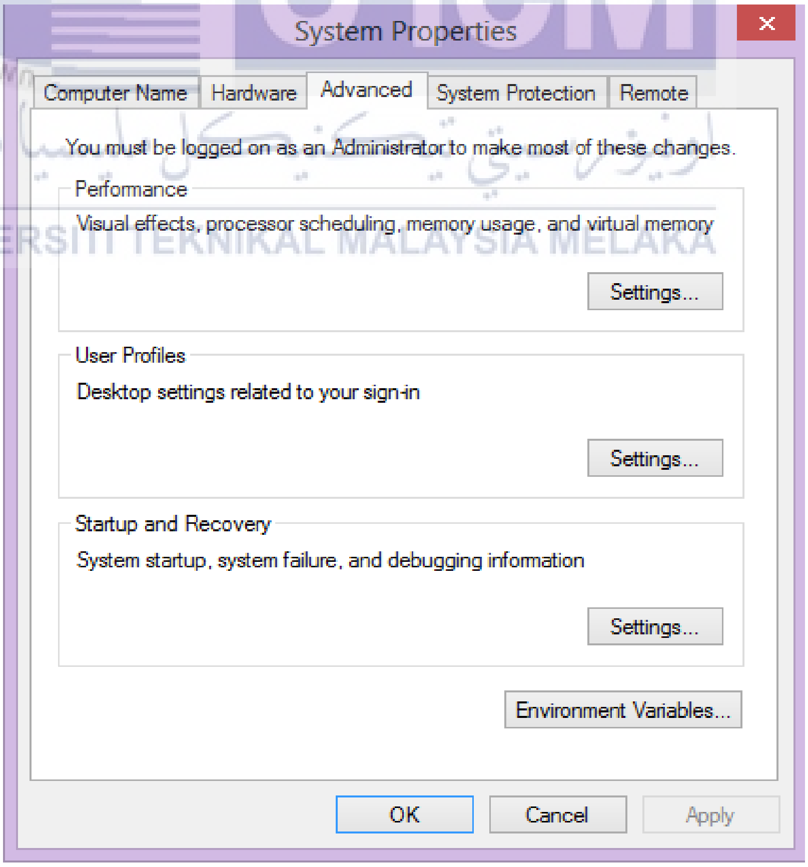
5.3.1 Configuration Environment Setup

JDK is required for development of Java application. In order to ensure correct JDK has been installed and located under a general folder such as C:\Program Files\Java, some Java technologies will be installed in the common folder. Then, environment variables have to be set to point towards them.

Below shows the configuration steps for Java environment setup in Window 8:

1. In Windows, open **Control Panel**, choose **System and Security**, and choose **System.**

2. Click on the **System protection**, and then choose one of the tabs which is **Advanced** system setting properties. Figure 5.3 shows system properties.



System Properties

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles

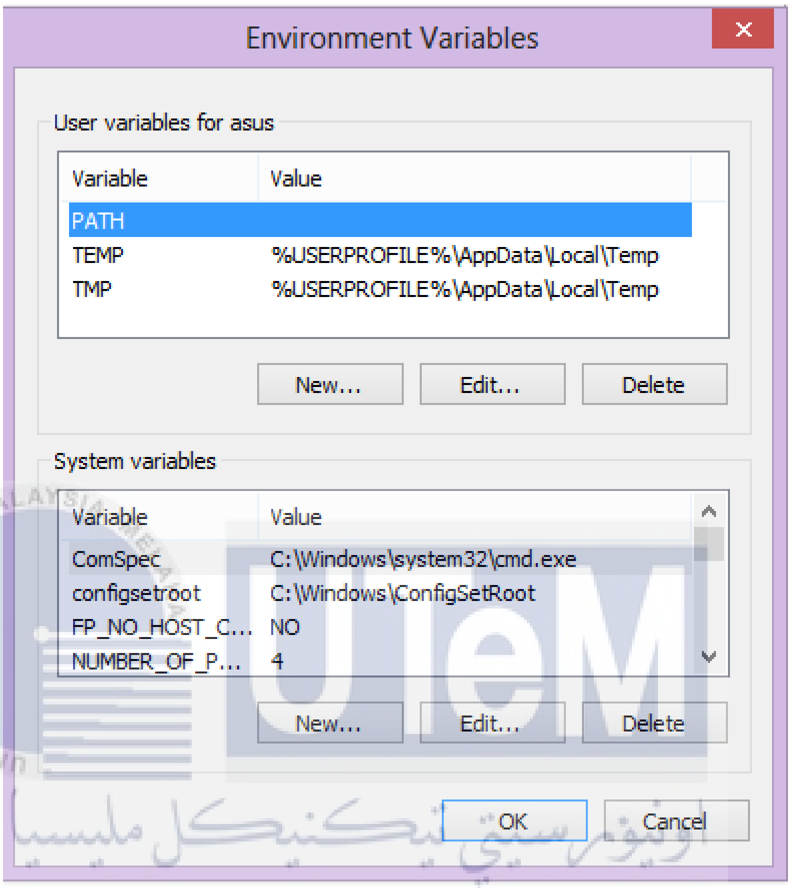
Desktop settings related to your sign-in

Startup and Recovery

System startup, system failure, and debugging information

Figure 5.3: System Properties

3. Click **Environmental Variables** from the **Advanced** tab. From the user variables list, find the **PATH** variable. Figure 5.4 illustrates environment variables.

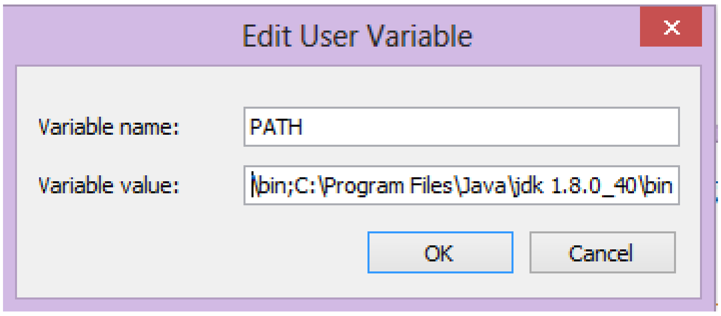


Variable	Value
PATH	
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
configsetroot	C:\Windows\ConfigSetRoot
FP_NO_HOST_C...	NO
NUMBER_OF_P...	4

Figure 5.4: Environment Variables

4. Click **Edit** and add the directory pointing to which the JDK resides. For example, C:\Program Files\Java\jdk 1.8.0_40\bin. Figure 5.5 shows edit user variable.



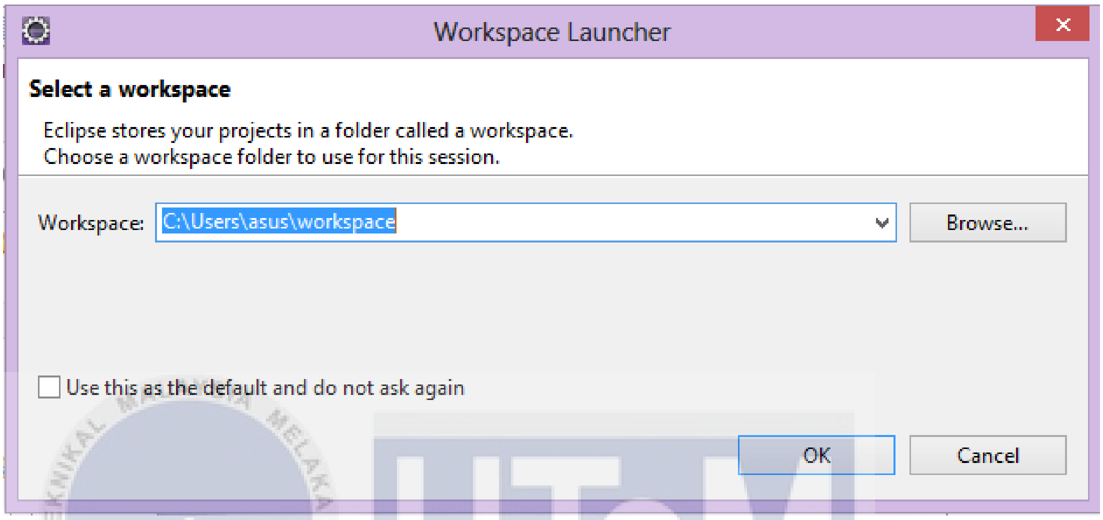
Variable name: PATH

Variable value: \bin;C:\Program Files\Java\jdk 1.8.0_40\bin

Figure 5.5: Edit User Variable

5. Click **OK**. The configuration for development environment is completed.

Every Eclipse projects are setup in a workspace. To change the current workspace, click **Browse** and select **File** from the menu bar. Lastly, choose **Switch Workspace** and click **OK**. Figure 5.6 illustrates edit Eclipse workspace.



Workspace Launcher

Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\Users\asus\workspace

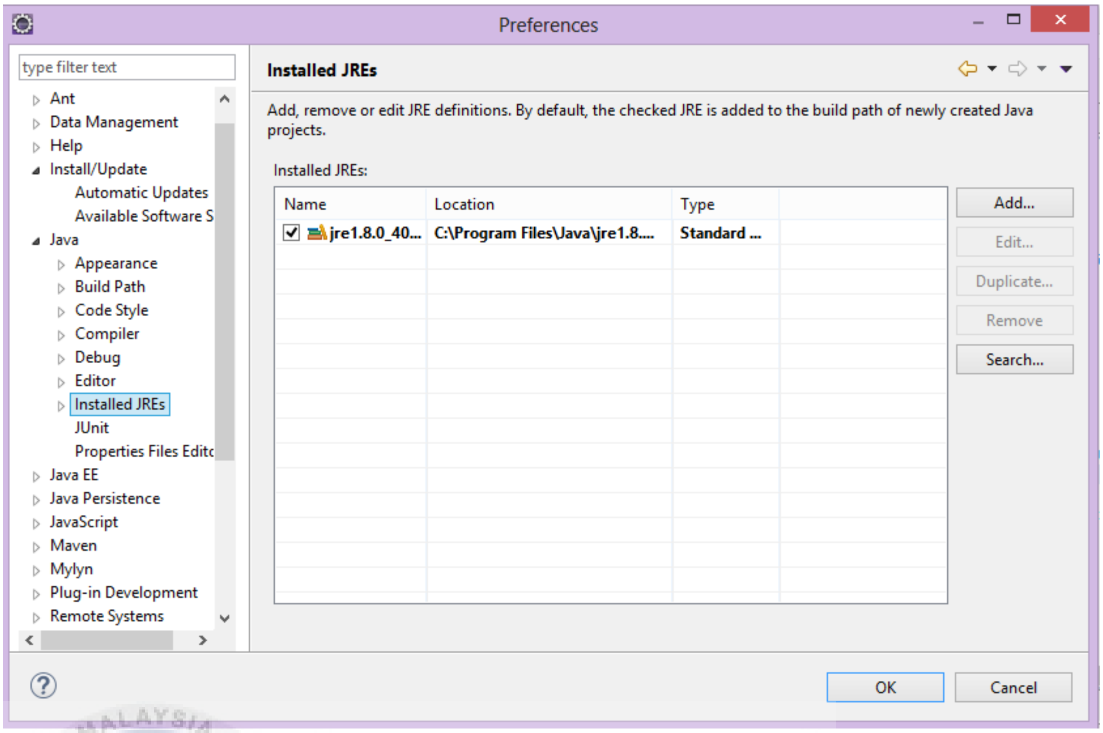
☐ Use this as the default and do not ask again

Figure 5.6: Edit Eclipse workspace

To compile and run the Java programs, use the latest version of Eclipse.

1. Select **Windows** from the menu bar, choose **Preferences**.

2. Go to **Java**-> **Installed JRE** subpane. Then, set the default installed JRE to **JRE1.8.0_40**. Figure 5.7 illustrates preferences for installed JREs.



Installed JREs

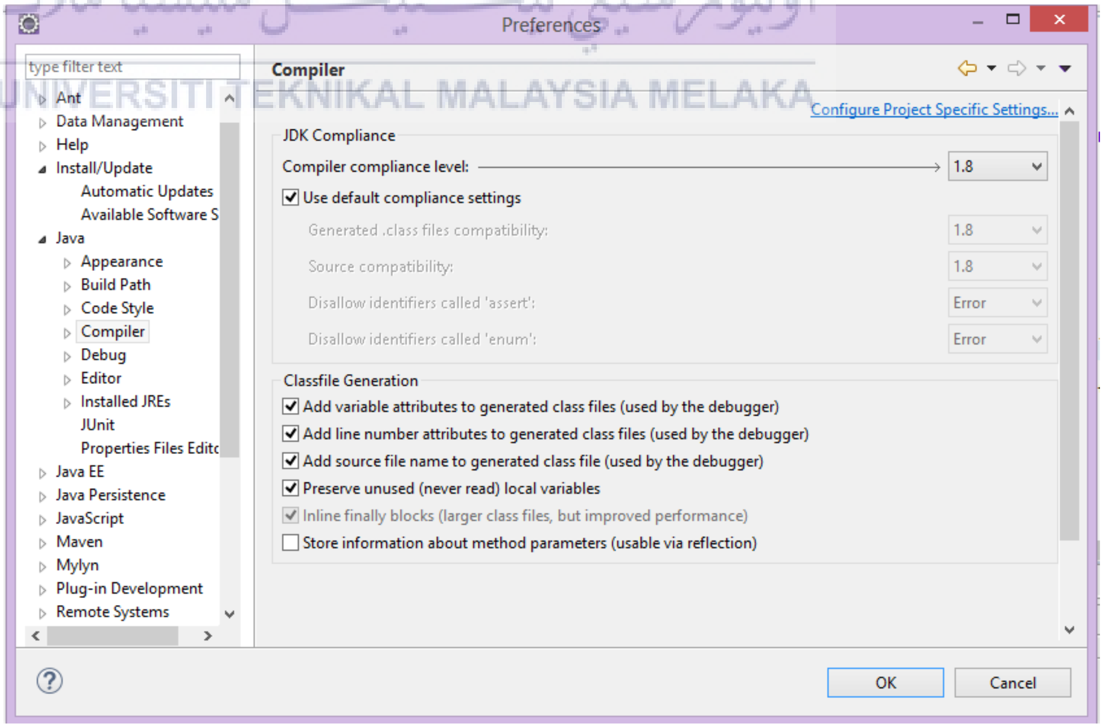
Add, remove or edit JRE definitions. By default, the checked JRE is added to the build path of newly created Java projects.

Installed JREs:

Name	Location	Type
☑ jre1.8.0_40...	C:\Program Files\Java\jre1.8....	Standard ...

Figure 5.7: Preferences for installed JREs

3. Go to **Java -> Compiler -> JDK Compiler** compliance level. Set the Java compiler to **Java 1.8**. Figure 5.8 shows preferences for JDK Compliance.



Compiler

JDK Compliance

Compiler compliance level: 1.8

☑ Use default compliance settings

Generated .class files compatibility: 1.8

Source compatibility: 1.8

Disallow identifiers called 'assert': Error

Disallow identifiers called 'enum': Error

Classfile Generation

☑ Add variable attributes to generated class files (used by the debugger)

☑ Add line number attributes to generated class files (used by the debugger)

☑ Add source file name to generated class file (used by the debugger)

☑ Preserve unused (never read) local variables

☑ Inline finally blocks (larger class files, but improved performance)

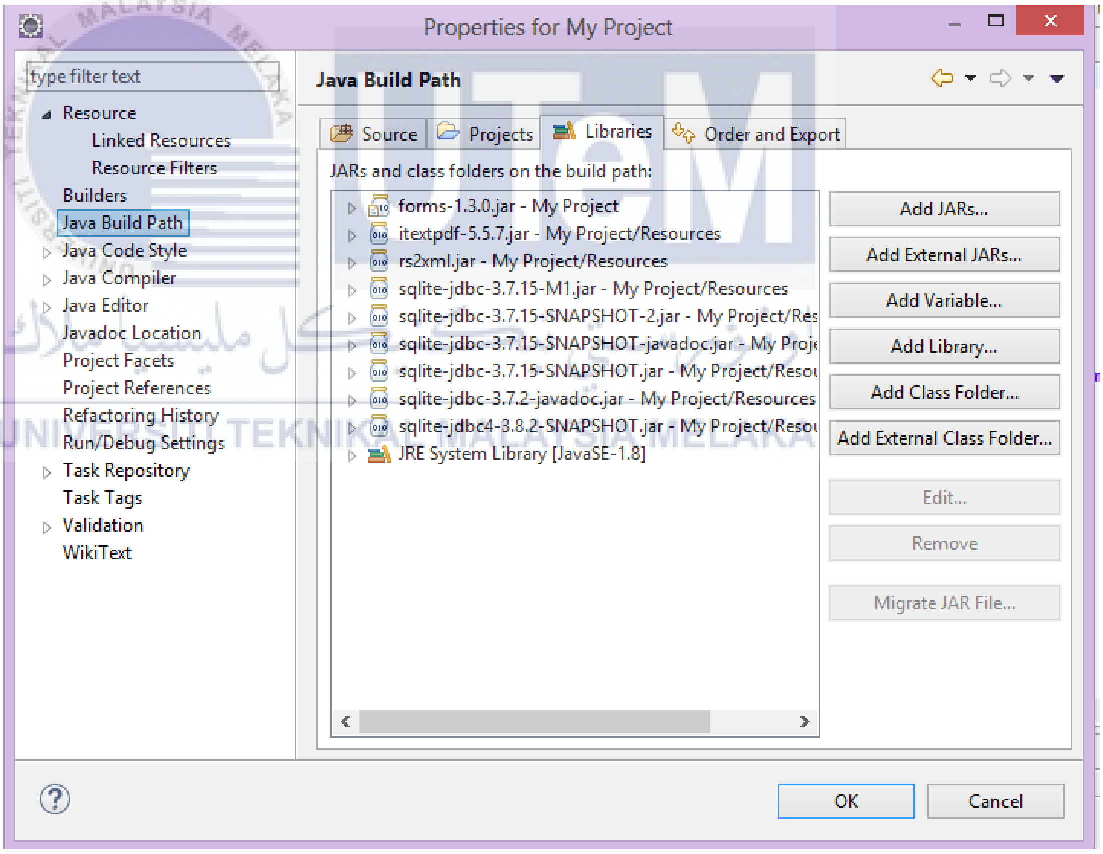
☐ Store information about method parameters (usable via reflection)

Figure 5.8: Preferences for JDK Compliance

4. Click **OK** to accept the changes that apply on it.

To ensure the Eclipse application is able to communicate with external data sources such as SQLite database, this can be done using Connection Navigator to set up and manage database connections in JDeveloper.

1. Right click on the Java project, then select **Properties**. Go to **Java Build Path->Libraries**, click **Add JARs** to add and load SQLite JDBC Driver JAR file in **Libraries**. Figure 5.9 illustrates jar file sqlite-jdbc-3.7.15.jar has been added to Java build path.



JARs and class folders on the build path:

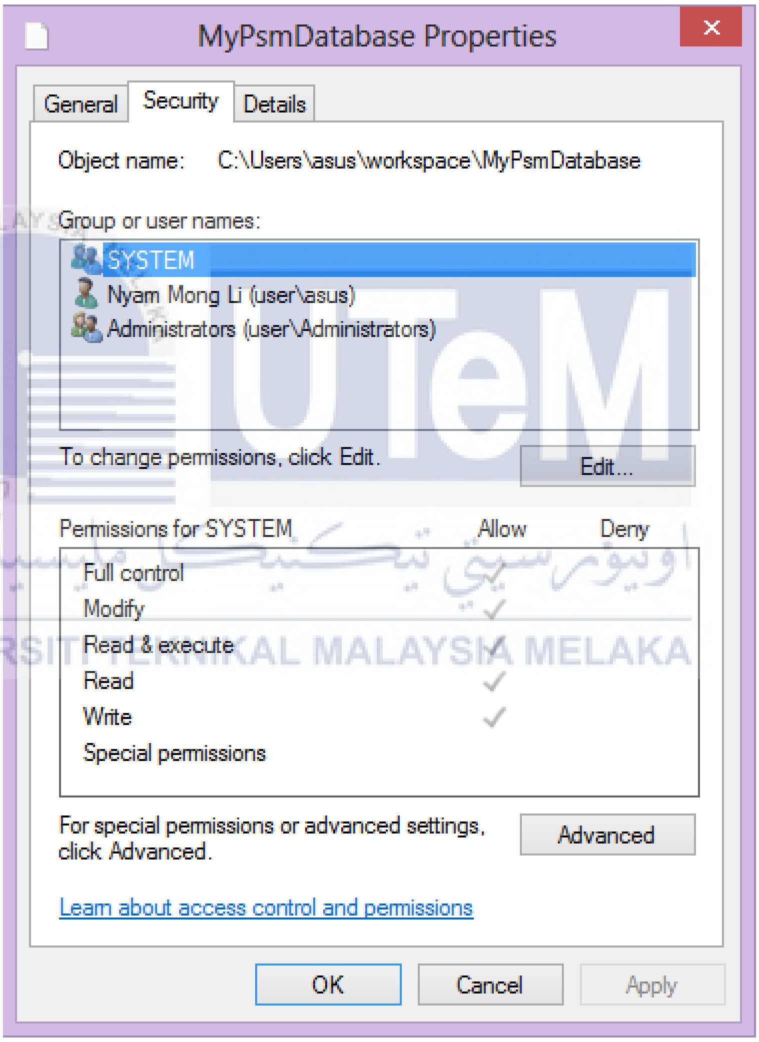
- forms-1.3.0.jar - My Project
- itextpdf-5.5.7.jar - My Project/Resources
- rs2xml.jar - My Project/Resources
- sqlite-jdbc-3.7.15-M1.jar - My Project/Resources
- sqlite-jdbc-3.7.15-SNAPSHOT-2.jar - My Project/Res
- sqlite-jdbc-3.7.15-SNAPSHOT-javadoc.jar - My Proje
- sqlite-jdbc-3.7.15-SNAPSHOT.jar - My Project/Resou
- sqlite-jdbc-3.7.2-javadoc.jar - My Project/Resources
- sqlite-jdbc4-3.8.2-SNAPSHOT.jar - My Project/Resou
- JRE System Library [JavaSE-1.8]

Figure 5.9: Add jar file sqlite-jdbc-3.7.15.jar in Java Build Path

2. To find the location of database, go to **workspace** and find the name of the database which has been stored in the file. In my project, I named my database as **MyPsmDatabase** .

3. Right click on the selected database, and then click **properties**.

4. Click on the **Security** tab to read the location of **MyPsmDatabase** database. Figure 5.10 illustrates MyPsmDatabase properties.



Object name: C:\Users\asus\workspace\MyPsmDatabase

Group or user names:

SYSTEM
Nyam Mong Li (user\asus)
Administrators (user\Administrators)

To change permissions, click Edit.

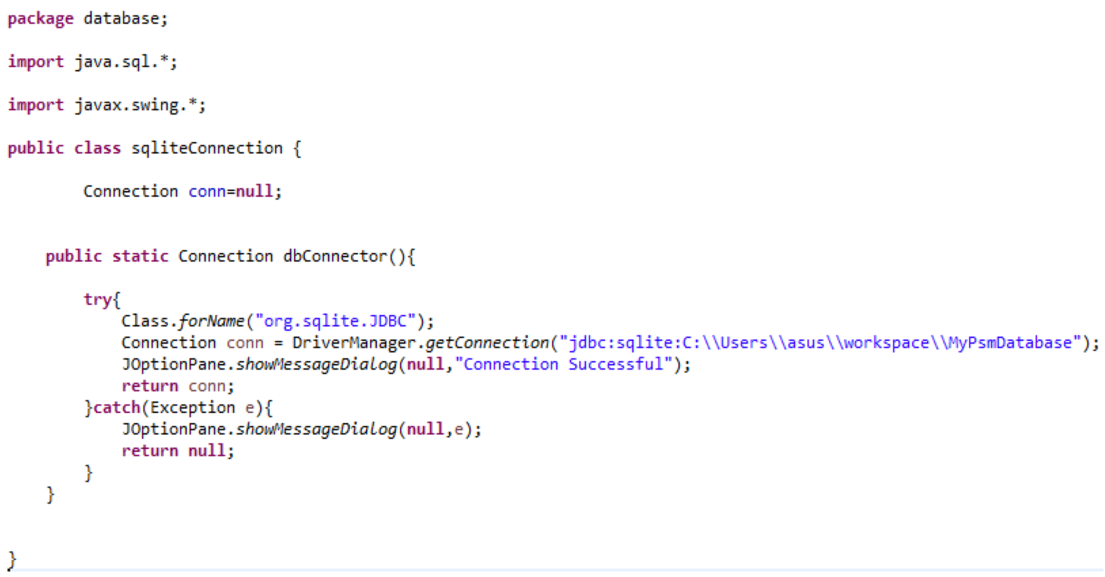
Permissions for SYSTEM	Allow	Deny
Full control	✓	
Modify	✓	
Read & execute	✓	
Read	✓	
Write	✓	
Special permissions		

For special permissions or advanced settings, click Advanced.

Learn about access control and permissions

Figure 5.10: MyPsmDatabase Properties

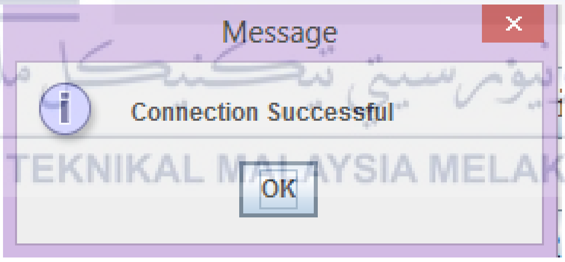
5. Copy the address of object name and add “\” on every single slash. Below is the code for how Java programs connect to the database. Figure 5.11 shows code for connection between Eclipse and SQLite database.



```
package database;

import java.sql.*;

import javax.swing.*;

public class sqliteConnection {

        Connection conn=null;


    public static Connection dbConnector(){

        try{
            Class.forName("org.sqlite.JDBC");
            Connection conn = DriverManager.getConnection("jdbc:sqlite:C:\\Users\\asus\\workspace\\MyPsmDatabase");
            JOptionPane.showMessageDialog(null,"Connection Successful");
            return conn;
        }catch(Exception e){
            JOptionPane.showMessageDialog(null,e);
            return null;
        }
    }


}
```

Figure 5.11: Code for connection between Eclipse and SQLite database

6. Compile and run the code, it shows that Eclipse connecting to SQLite database successfully. Figure 5.12 illustrates connection is successful for database connection.



Connection Successful

Figure 5.12: Connection is successful for database connection

5.3.2 Version Control Procedure

Version control is any kind of practice or technologies that can be used for tracking and providing control changes to the source code files in Java programs. It is important for us to locate and fix the bugs to discover the presence of bugs in a version by retrieving and running different versions of software. Besides that, multiple versions of the common software can be deployed, users would like to compare the current version with the previous version of some software. Version control can keep track of every change that has been done and to reverse the changes that applied on it previously when there is a need. It is useful for maintaining documentation and files of configuration to have a record of each software version, person who is in charge of the changes and comparison for the performance of particular versions.

Every single piece of information such as design drafts, source code and documentation that could provide changes is versioned. They should be kept in the common repository tree for easy reference.

For this project, Table 5.1 shows the version updates of AP-SMS.

Table 5.1: Version of AP-SMS

Version	**Modifier**	**Change/Improvement**
1.0	Nyam Mong Li	Initial version
2.0	Nyam Mong Li	Delete randomization function

AP-SMS 1.0 is the first version that means to be final product for early phases of development process. It is labelled as version 1.0 because it is a new developed system. Version 2.0 has been released when realized there is a need to do some improvement changes and spaces on it. Therefore, it leads to development changes. To develop a latest version of the system, previous version of the system has been reviewed and used it for developing a new feature or bug-fixed software. If there are some unresolvable problems or errors occurring in the latest version, old version can be used and replaced the latest version. Under version control procedures, the completed source code and documentation should be kept unchanged for the old version. The final product can be released after going through the review and acceptance evaluation test.

5.4 Implementation Status

Implementation status is used to monitor duration for each activity to be completed. The information such as details of programming language, platform, environment and duration of each module are recorded in schedule form. There are five main modules for AP-SMS which are data authority module, energy provider module, smart meter module, aggregator module and user module. Table 5.2 describes about implementation status for each of these five modules.

Table 5.2: Description of implementation status for every module

No	Module Name	Description	Duration (days)	Start Date	End Date
1	Data Authority	To generate public key and private key of RSA	10	17/03/2016	27/03/2016
2	Energy Provider	To generate secret key *P*,*Q* and *N* of homomorphic encryption. Then, encrypt secret key *P* and sent it to smart meter. Besides that, it determines and encrypts rate of *Wh*, then sent it to aggregator to do	20	28/03/2016	17/04/2016

		some calculation on the encrypted data. It generates billing report to view the results.			
3	Smart Meter	To decrypt secret key *P*. Furthermore, it reads meter usage of user and encrypt meter reading, then sent it to aggregator.	15	18/04/2016	02/05/2016
4	Aggregator	It does calculation such as addition and multiplication on encrypted data.	20	03/05/2016	22/05/2016
5	User	User fills in the user form to know the energy usage for particular date. Furthermore, user can search for previous date to view hourly energy usage and rate of hourly payment.	10	23/05/2016	02/06/2016

5.5 Conclusion

This chapter has discussed the software development environment setup and software configuration management for implementation phase. Those setup and configuration management ensure the production of the system to be more stable and in steady state. User training, user documentation and installed production system are the output for this phase. The next chapter discusses the testing phase, where defects and damages in a system can be identified.

CHAPTER VI

TESTING

6.1 Introduction

Testing is carried out in the system after the system has implemented successfully. The purpose of testing is to find imperfection in the system and came out with the solutions of fixing the errors. Furthermore, quality of the system has been reviewed to ensure there is an improvement on the system quality. Thus, testers can choose to use a test script to perform the test and verify the results to determine whether the application is functioning well and fulfilled the documentation of requirement analysis phase. There are some strategies to use for this testing phase which will be discussed in this chapter.

6.2 Test Plan

Test plan is a document that describes in details for the methods, testing scope and tools chosen to test software or an application. The aim of carrying out a test plan is to verify the scope and objectives of the system that had been discussed before the development of the system. Furthermore, it creates awareness about the importance of validating the system. It also provides a way for developers or programmers to decide correct resources when developing a system.

6.2.1 Test Organization

This subsection is used to discuss the person who is responsible to carry out the testing. There are divided into three groups which are developer, tester and colleagues. Each of the group has different tasks to be done. The developer will find defects during testing process and records it in a document including test results and test dataset. Tester and other colleagues play a great role in this testing process because they able to identify the problems that are previously undetected. In addition, they will monitor performance of the system and ensure it is an effective system. Table 6.1 shows responsibility of persons who are involved in testing.

Table 6.1: Responsibility of persons who are involved in testing

Person involved	Role/Responsibility	Name
Developer	Compile and run the system to discover errors. Write documentation for system progress during program development. To allow people understand how the code works, developer will write	NYAM MONG LI

	comments in the source code as coded instructions.	
Tester	Follow the requirements when doing testing on the system	ENCIK RIZUAN ENCIK NOR AZMAN
Colleagues	Use the system and give feedbacks about the system. The colleagues also monitor the system to know the weakness and strengths of the system.	CHAN SOOK YIE CHEN WAI KEAT SEE YUAN YIN WONG JUN RONG

6.2.2 Test Environment

For this project, testing environment is divided into two types which are hardware and software. The testing can be performed on the laptop of developer as well as on other computers with Eclipse and Java been installed. Before testing is carried out by the testers, developer will show a demonstration to give them a clearer look on how to use the system. During testing, all images will be stored in removable disk or local disk as a proof. Table 6.2 indicates the conditions for the hardware and software to conduct the testing environment.

Table 6.2: Requirements of Hardware and Software

Environment Category	Tools	Description
Hardware	Operating System	Window 8
	CPU	Intel Core i3
	RAM Memory	8.0 GB of RAM
	Hard Disk Space	500 GB
	Input Device	Mouse and Keypad
Software	Eclipse	Java language
	Database	SQLite

6.2.3 Test Schedule

This section discussed about the duration taken for each activity to be conducted on this test and the result is recorded.

Table 6.3: Test Schedule of AP-SMS

Test Activity	Description	Test Cycle (times)	Duration (Days)	Start Date	End Date
Unit Testing	To define AP-SMS's unit requirements.	5	2	15/05/2016	17/05/2016
Integration Testing	To ensure each module	15	7	18/05/2016	24/05/2016

	of AP-SMS meet its accuracy.				
System Testing	To verify the performance of the system can be acceptable by the user.	10	4	24/05/2016	28/05/2016
Acceptance Testing	To examine the system fulfill the requirements.	15	5	29/05/2016	03/06/2016

6.3 Test Strategy

For this project development, three methods had been chosen to carry out the test. The selected methods are white box testing, black box testing and bottom up testing. White box testing is an approach to test internal structures of a system for testing the software. Thus, the tester must have some programming knowledge or logic to recognize defective or invalid code. The person who has the ability to discover the data input type for testing the system is developer. Besides that, developer has the rights to delete unnecessary code that are causing the system to be crash.

Black box testing is a way to check the functionality and requirements of a system without inspecting its internal structures. It can be carried out by colleagues to test the system because no code structure or internal logic is needed for this type of testing method. They only need to have knowledge on how the system should work and behave in response to data input or specific action that has been taken.

Bottom up testing is performed from the lowest level of components and proceeding to the highest level of components in a system. It is because larger system is formed by linking together all the small base elements. In the end, the system will grow in completeness and complexity.

6.3.1 Classes of tests

1. **Functionality Test**

   This test is carried out by inserting data to the system and verifying the output to check for the features of system. It is a testing to ensure system works well according to specification and performs what is intended for.

2. **Usability Test**

   This test is conducted by asking the user to gain access to the system. If the user has trouble to access the system, user interface design of the system should be improved. A good designed interface with buttons is very important because it gives an impression that the system is easy to be use and user friendly.

3. **Security Test**

   It is a checklist of tasks used to examine whether the system is safe and secure to be used. Defect and loophole in a system can be identified to ensure the system achieve its goals of confidentiality and integrity in protecting the data.

4. **Stress Test**

This testing is used to determine whether the system is robust enough and can be withstanding beyond of a normal operation. The purpose of this test is to check the system is stable to be used although the system faces the problem of insufficient memory space or hard disk.

6.4 Test Design

Test description and test data had been carried out on AP-SMS, results of each test has been recorded and displayed in the table respectively.

6.4.1 Test Description

Identification of module and availability of test cases had been tested. Table 6.4 until Table 6.8 describes the results of each module for their test description.

Table 6.4: Data Authority Test Description

Test ID	**Action**	**Expected Result/Output**	**Actual Result (OK/NOT OK)**
SMS_DA001	Click public key button	An application window will be pop up, it shows the new generated public key.	OK
SMS_DA002	Click private key button	An application window will be pop up, it displays the new generated private key.	OK

Table 6.5: Energy Provider Test Description

Test ID	**Action**	**Expected Result/Output**	**Actual Result (OK/ NOT OK)**
SMS_EP001	Click generate prime *P*, prime *Q* and *N* button	An application window will be pop up, it shows the new generated prime *P*, prime *Q* and *N*.	OK
SMS_EP002	Click encrypt *P* button	An application window will be pop up, it displays value of prime *P* in encrypted form.	OK
SMS_EP003	Click rate of *Wh* button	An application window will be pop up, it shows rate for every unit of *Wh* in *cents*.	OK
SMS_EP004	Click encrypt rate of *Wh* button	An application window will be pop up, it displays value of *Wh* in encrypted form.	OK
SMS_EP005	Click decrypt user data button	An application window will be pop up, it displays value of user data that has been decrypted which is	OK

		in plaintext form.	
SMS_EP006	Click generate hourly billing as TXT	A notepad document will be pop up, it displays billing report in TXT form.	OK

Table 6.6: User Test Description

Test ID	**Action**	**Expected Result/Output**	**Actual Result (OK/NOT OK)**
SMS_USR001	Click user form button	A form will be pop out which asked the user to enter their personal information into the form.	OK
SMS_USR002	All fields are empty	Alert box pop up and ask user to fill up the fields.	OK
SMS_USR003	Click clear button	The field will be empty and combo box will back to its initial index.	OK
SMS_USR004	Insert wrong or invalid information	Error message pop up and ask user re-enter into the field again.	OK

SMS_USR005	Data insert correctly and saved and accepted by system	The particular data saved successfully into the database, new data will be displayed.	OK
SMS_USR006	Click search date button	Enter the correct date into the field, and then enter national registration identification card (NRIC) number into the field. The result will be shown out.	OK

Table 6.7: Smart Meter Test Description

Test ID	**Action**	**Expected Result/Output**	**Actual Result** **(OK/NOT OK)**
SMS_MTR001	Click decrypt *P* button	An application window will be pop up, it displays value of *P* that has been decrypted which is in plaintext form.	OK
SMS_MTR002	Click meter reading in user	An application window will be	OK

	form button	pop up, it shows personal information of user and consumption of hourly meter reading which is in plaintext form.	
SMS_MTR003	Click encrypt meter reading in user form button	An application window will be pop up, it shows personal information of user and consumption of hourly meter reading which is in cipher text form.	OK

Table 6.8: Aggregator Test Description

Test ID	Action	Expected Result/Output	Actual Result (OK/NOT OK)
SMS_AGR001	Click process encrypted user data button	An application window will be pop up, it shows the calculation such as addition and multiplication on encrypted data.	OK

6.4.2 Test Data

To determine the performance and efficiency of the system, test data can be used to examine each modules of AP-SMS. Unit test, system test and integration test has been used to test data, they are identified from test cases respectively.

6.5 Test Results and Analysis

This topic is used to display the results that had been conducted on each module, the result can be passed or failed. The result has been screenshot and recorded. Table 6.9 shows the outcome of each test.

Table 6.9: Test Results

Test Case Identification	**Tester Identification**	**Result (PASSED/FAILED)**
AP-SMS_DA001	OK	PASSED
AP-SMS_DA002	OK	PASSED
AP-SMS_EP001	OK	PASSED
AP-SMS_EP002	OK	PASSED
AP-SMS_EP003	OK	PASSED
AP-SMS_EP004	OK	PASSED
AP-SMS_EP005	OK	PASSED
AP-SMS_EP006	OK	PASSED
AP-SMS_USR001	OK	PASSED
AP-SMS_USR002	OK	PASSED

AP-SMS_USR003	OK	PASSED
AP-SMS_USR004	OK	PASSED
AP-SMS_USR005	OK	PASSED
AP-SMS_USR006	OK	PASSED
AP-SMS_MTR001	OK	PASSED
AP-SMS_MTR002	OK	PASSED
AP-SMS_MTR003	OK	PASSED
AP-SMS_AGR001	OK	PASSED

Table 6.9 shows that all test cases had been passed as expected. Therefore, the test cases have been screenshot during the testing as proof.

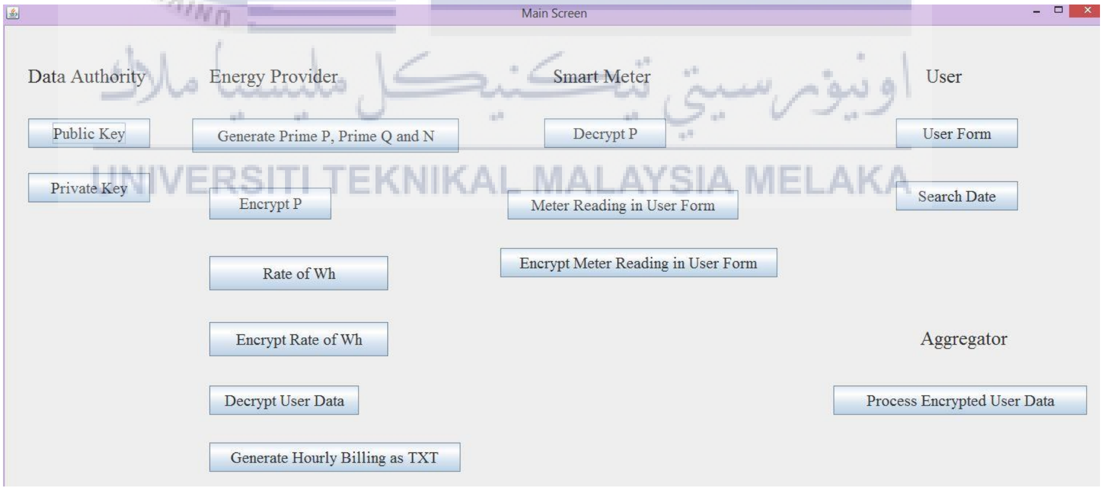
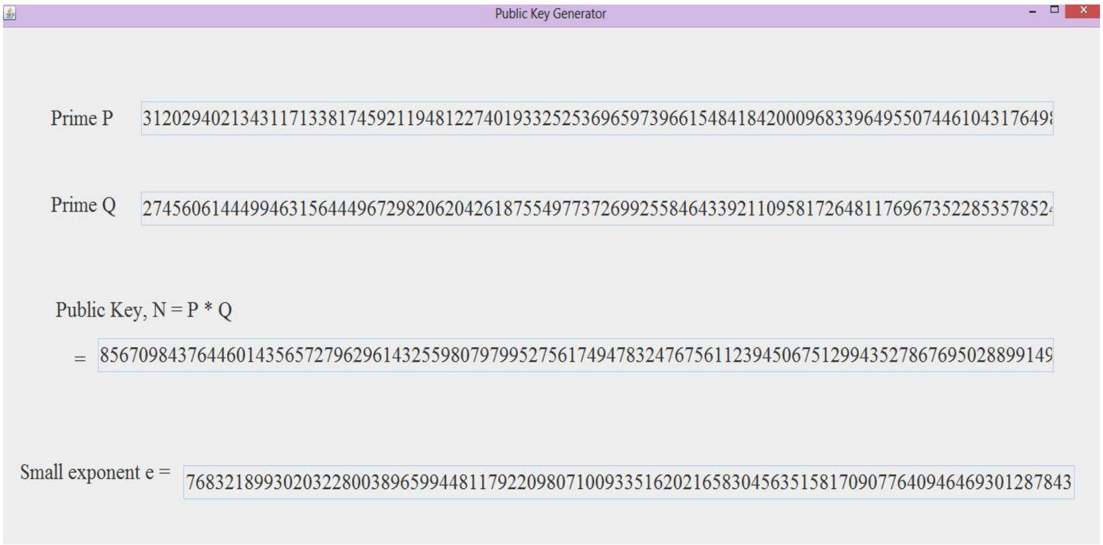


Figure 6.1: Main interface of the system. Click Public Key button. (AP-SMS_DA001)



Prime P 312029402134311713381745921194812274019332525369659739661548418420009683396495507446104317649

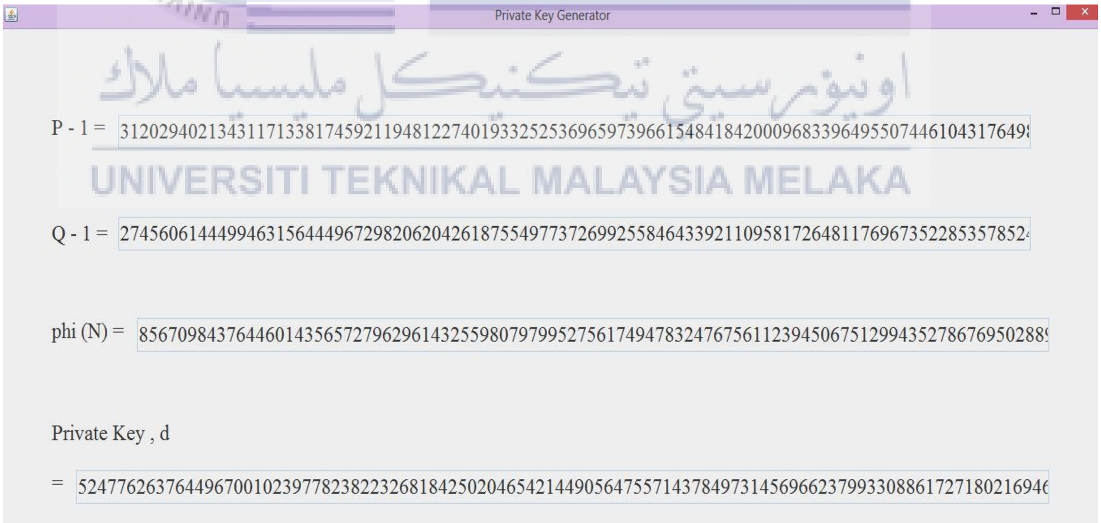
Prime Q 274560614449946315644496729820620426187554977372699255846433921109581726481176967352285357852

Public Key, N = P * Q

= 8567098437644601435657279629614325598079799527561749478324767561123945067512994352786769502889914

Small exponent e = 7683218993020322800389659944811792209807100933516202165830456351581709077640946469301287843

Figure 6.2: A new public key has been generated by data authority. (AP-SMS_DA001)



P - 1 = 312029402134311713381745921194812274019332525369659739661548418420009683396495507446104317649

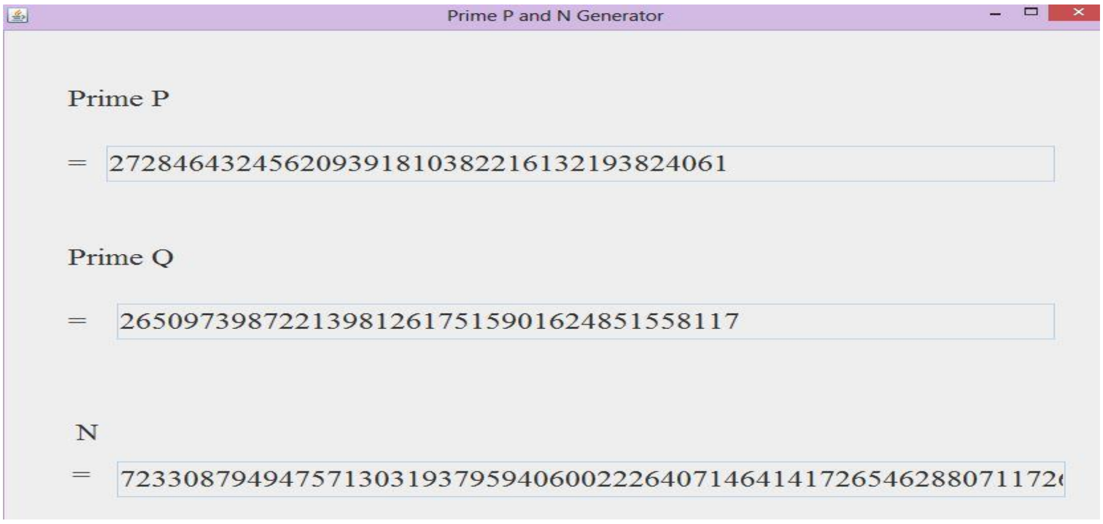
Q - 1 = 274560614449946315644496729820620426187554977372699255846433921109581726481176967352285357852

phi (N) = 856709843764460143565727962961432559807979952756174947832476756112394506751299435278676950288

Private Key , d

= 524776263764496700102397782382232681842502046542144905647557143784973145696623799330886172718021694

Figure 6.3: Click Private Key button. A new private key has been generated by data authority.(AP-SMS_DA002)



Prime P

= 272846432456209391810382216132193824061

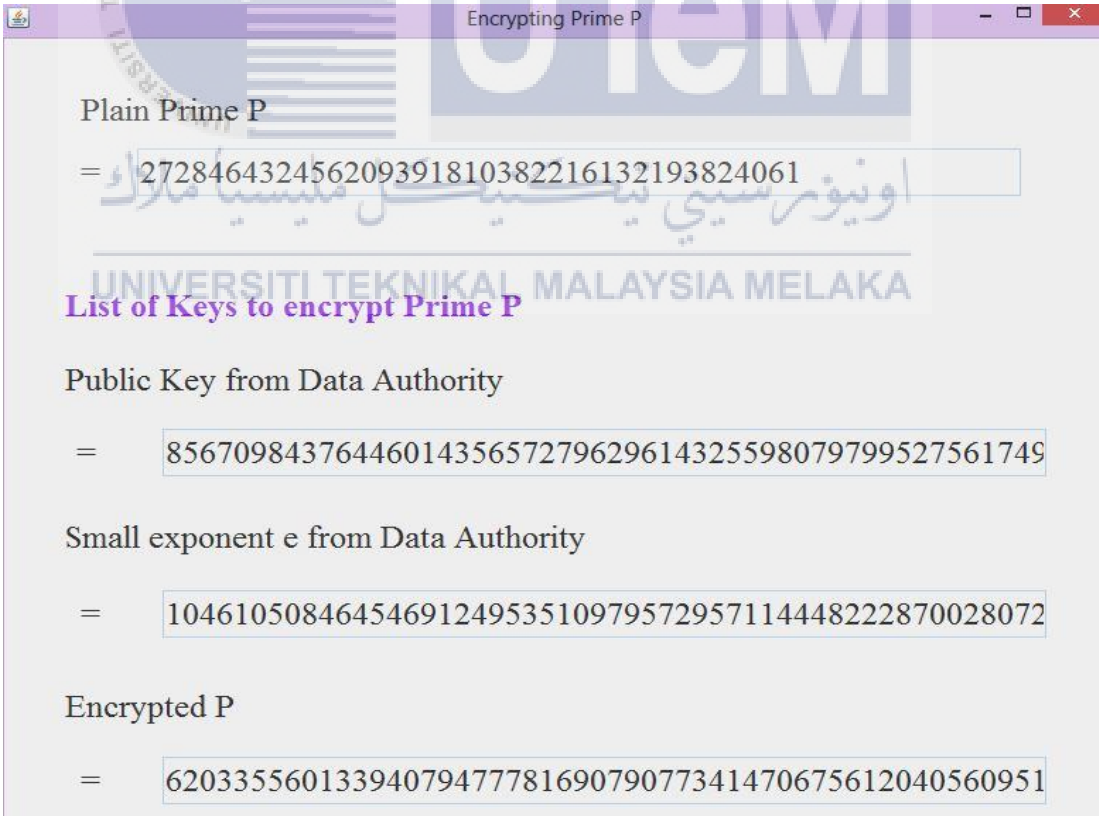
Prime Q

= 265097398722139812617515901624851558117

N

= 72330879494757130319379594060022264071464141726546288071172

Figure 6.4: Click Generate Prime P, Prime Q and N button. A new prime P, prime Q and N have been generated by energy provider. (AP-SMS_EP001)



Plain Prime P

= 272846432456209391810382216132193824061

List of Keys to encrypt Prime P

Public Key from Data Authority

= 8567098437644601435657279629614325598079799527561749

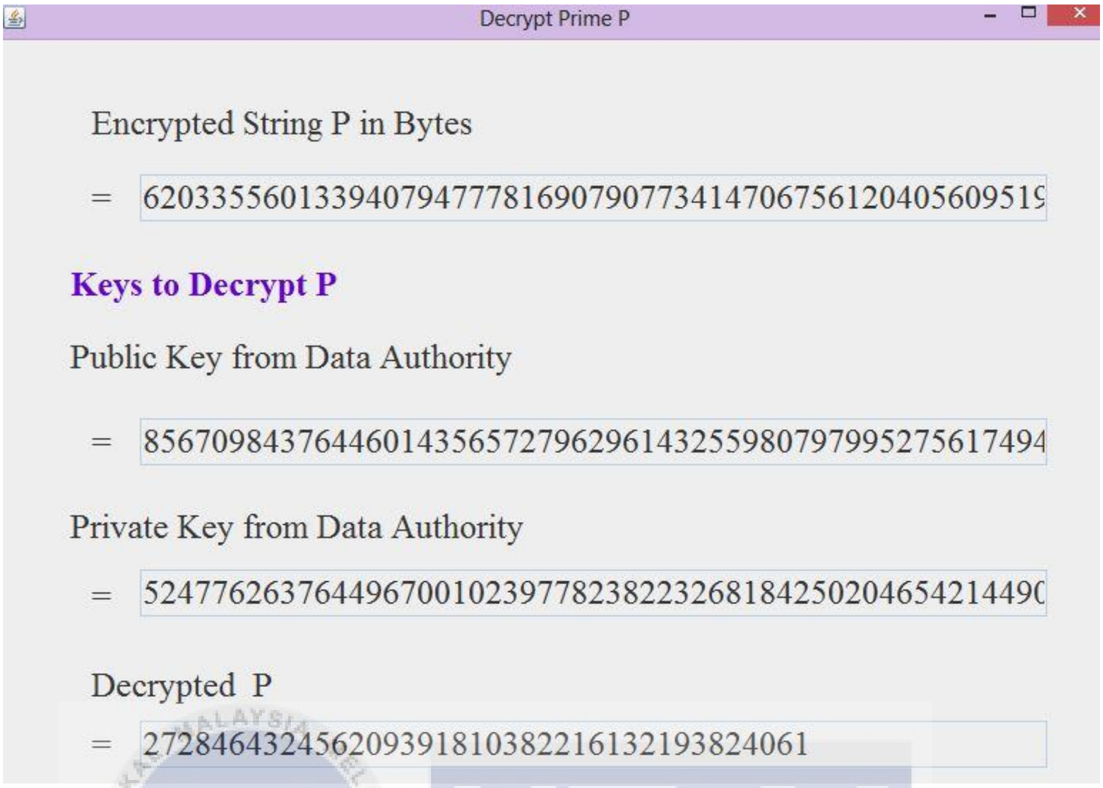
Small exponent e from Data Authority

= 1046105084645469124953510979572957114448222870028072

Encrypted P

= 6203355601339407947778169079077341470675612040560951

Figure 6.5: Click Encrypt P button. Energy provider uses public key and small exponent e to encrypt prime P. (AP-SMS_EP002)



Encrypted String P in Bytes

= 62033556013394079477781690790773414706756120405609519

Keys to Decrypt P

Public Key from Data Authority

= 856709843764460143565727962961432559807979952756174944

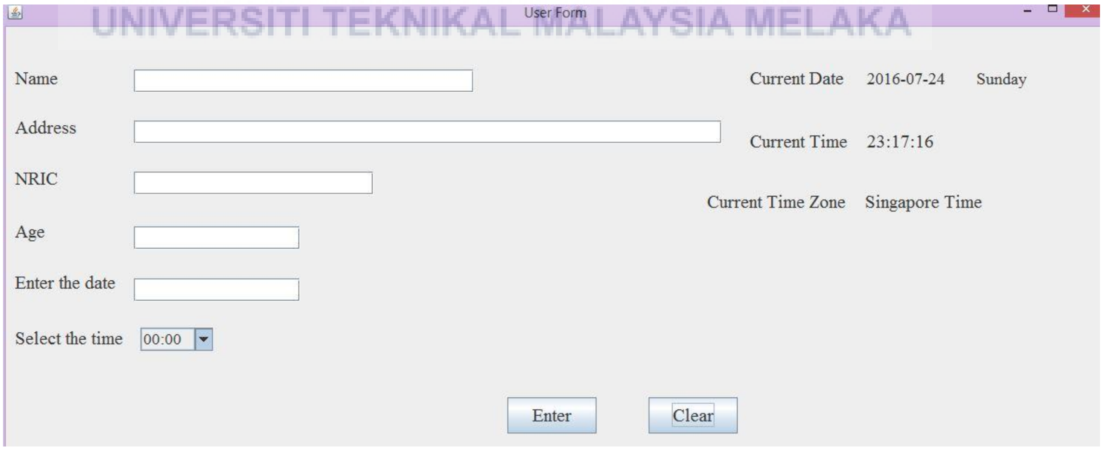
Private Key from Data Authority

= 52477626376449670010239778238223268184250204654214490

Decrypted P

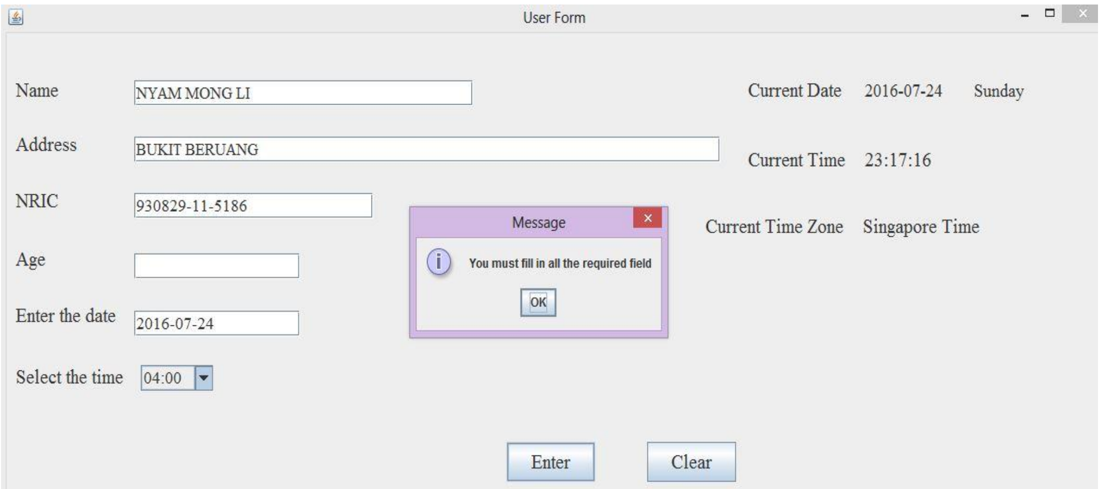
= 272846432456209391810382216132193824061

Figure 6.6: Click Decrypt P button. Smart meter uses public key and private key to get decrypted P. (AP-SMS_MTR001)



Name		Current Date	2016-07-24	Sunday
Address		Current Time	23:17:16	
NRIC		Current Time Zone	Singapore Time	
Age				
Enter the date				
Select the time	00:00			

Figure 6.7: Click User Form button. A user form will appear. Click Clear button will cause all the fields to be empty. (AP-SMS_USR001 & AP-SMS_USR003)



Name NYAM MONG LI

Address BUKIT BERUANG

NRIC 930829-11-5186

Age

Enter the date 2016-07-24

Select the time 04:00

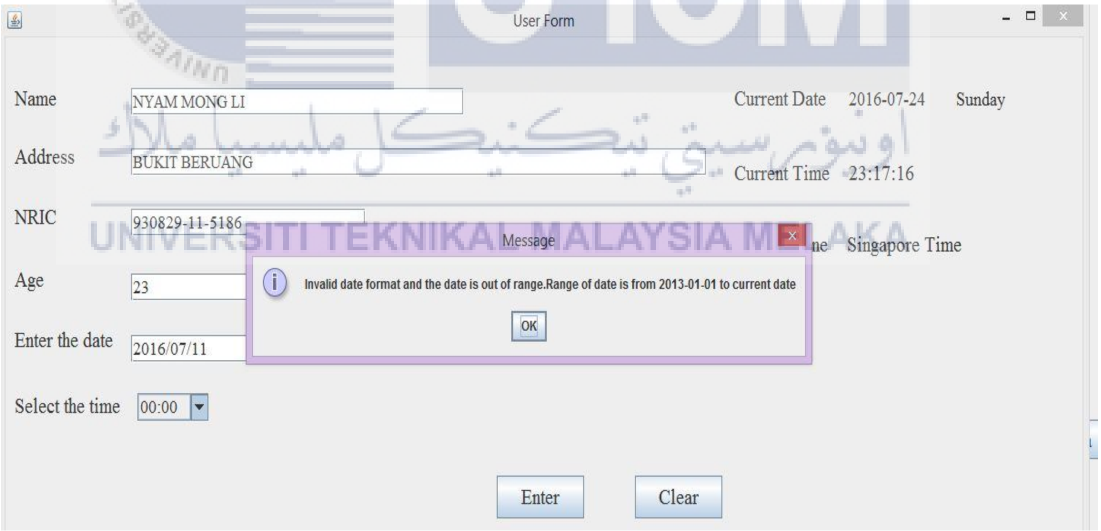
Current Date 2016-07-24 Sunday

Current Time 23:17:16

Current Time Zone Singapore Time

You must fill in all the required field

Figure 6.8: An error message pop up which require user to fill up all the empty fields. (AP-SMS_USR002)



Name NYAM MONG LI

Address BUKIT BERUANG

NRIC 930829-11-5186

Age 23

Enter the date 2016/07/11

Select the time 00:00

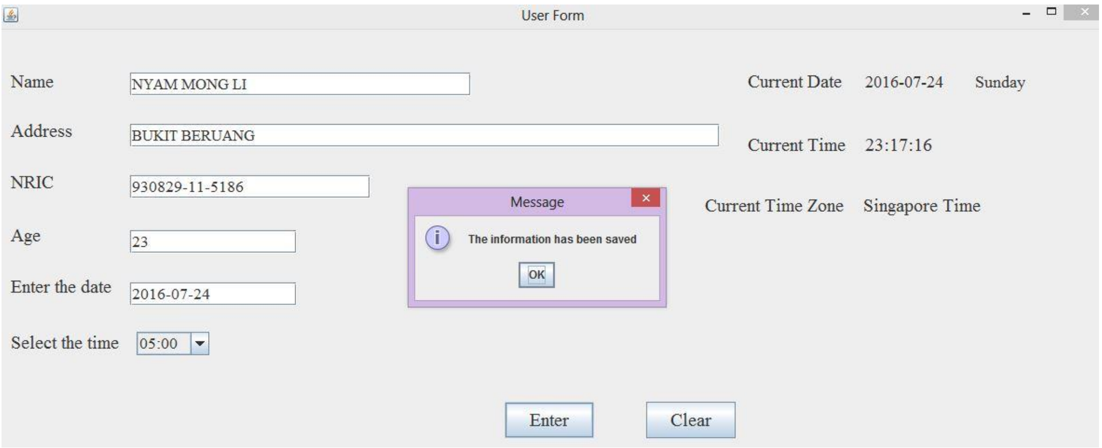
Current Date 2016-07-24 Sunday

Current Time 23:17:16

Singapore Time

Invalid date format and the date is out of range.Range of date is from 2013-01-01 to current date

Figure 6.9: Error message pop up when user inserts invalid information into the fields. (AP-SMS_USR004)



User Form

Name: NYAM MONG LI

Address: BUKIT BERUANG

NRIC: 930829-11-5186

Age: 23

Enter the date: 2016-07-24

Select the time: 05:00

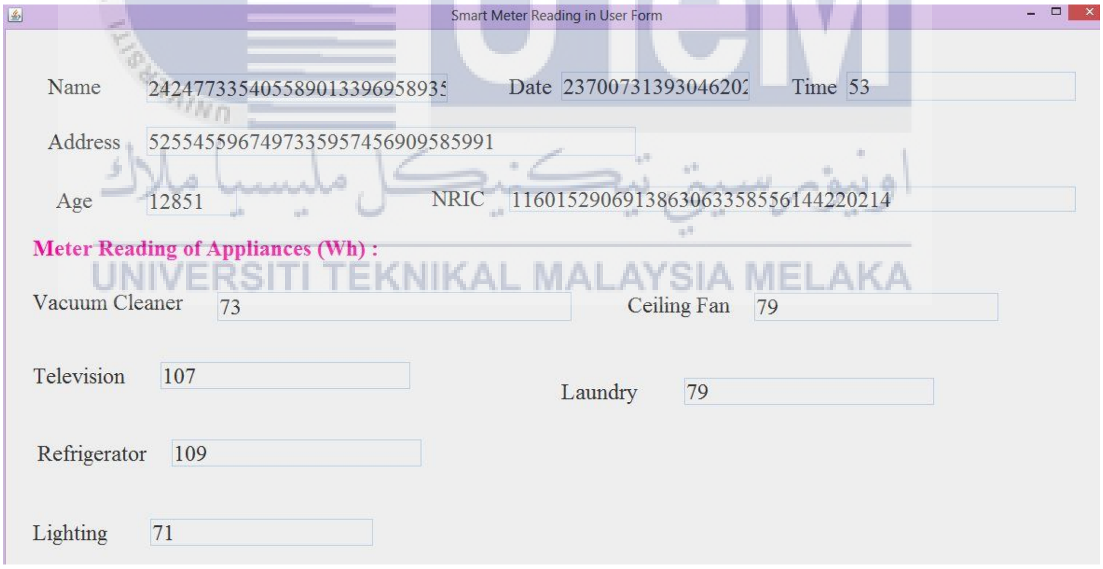
Current Date: 2016-07-24 Sunday

Current Time: 23:17:16

Current Time Zone: Singapore Time

Message: The information has been saved

Figure 6.10: Click Enter button. A useful message pop up which shows the information has been saved. (AP-SMS_USR005)



Smart Meter Reading in User Form

Name: 24247733540558901339695893

Date: 237007313930462O

Time: 53

Address: 5255455967497335957456909585991

Age: 12851

NRIC: 1160152906913863063358556144220214

Meter Reading of Appliances (Wh) :

Vacuum Cleaner: 73

Ceiling Fan: 79

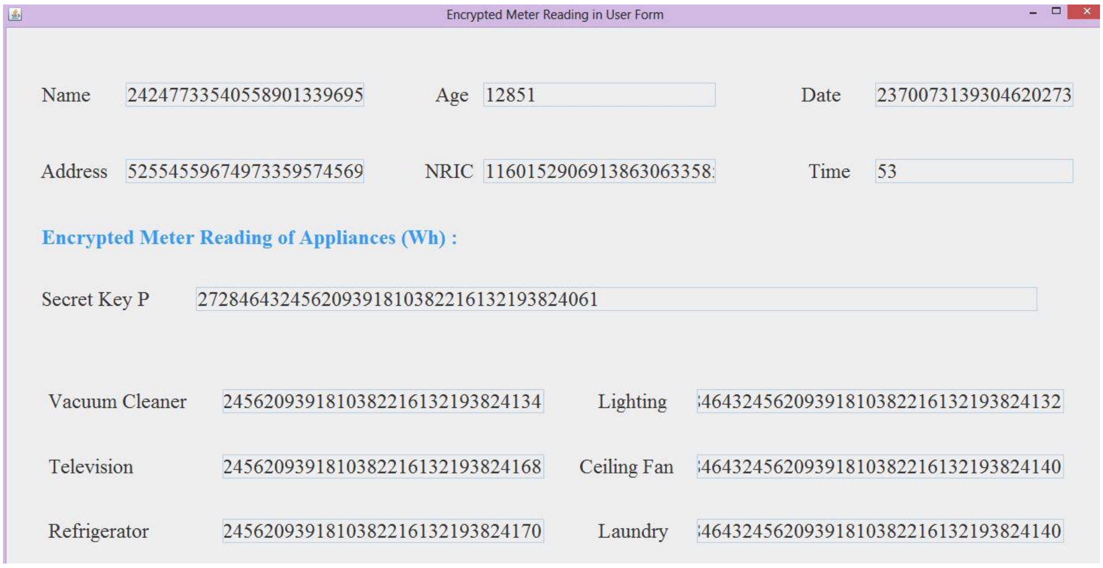
Television: 107

Laundry: 79

Refrigerator: 109

Lighting: 71

Figure 6.11: Click Meter Reading in User Form button. An application window which shows the hourly meter reading of appliances that has been used by user. (AP-SMS_MTR002)



Encrypted Meter Reading in User Form

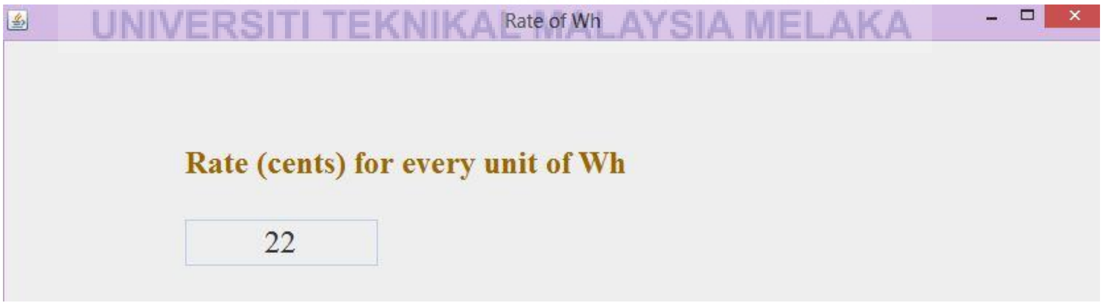
Name	24247733540558901339695	Age	12851	Date	2370073139304620273
Address	52554559674973359574569	NRIC	1160152906913863063358	Time	53

Encrypted Meter Reading of Appliances (Wh) :

Secret Key P 272846432456209391810382216132193824061

Vacuum Cleaner	2456209391810382216132193824134	Lighting	46432456209391810382216132193824132
Television	2456209391810382216132193824168	Ceiling Fan	46432456209391810382216132193824140
Refrigerator	2456209391810382216132193824170	Laundry	46432456209391810382216132193824140

Figure 6.12: Click Encrypt Meter Reading in User Form button. An application window which shows the encrypted meter reading of appliances that has been used by user. (AP-SMS_MTR003)

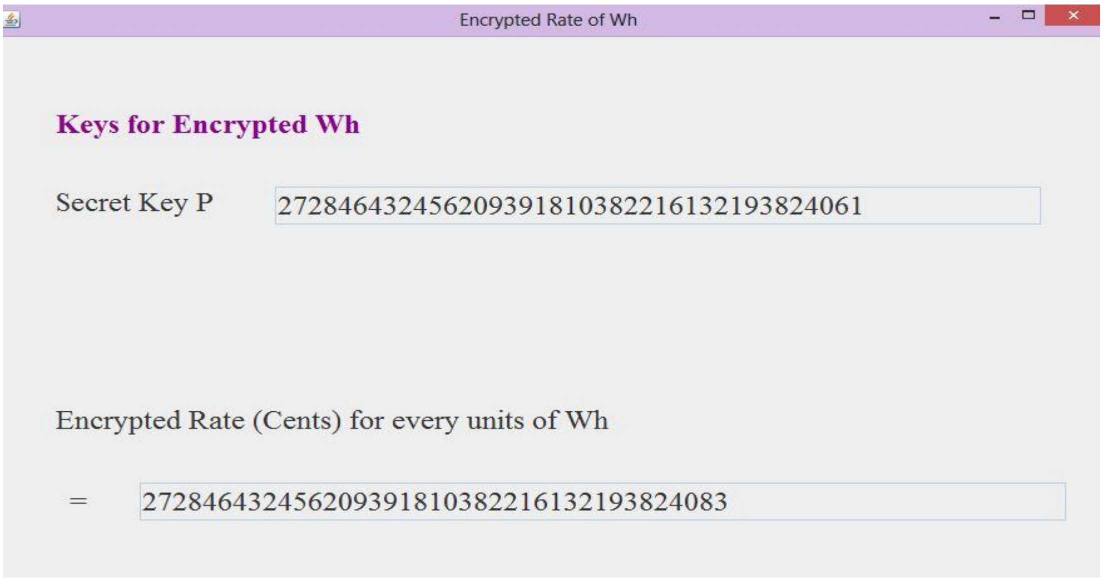


Rate of Wh

Rate (cents) for every unit of Wh

22

Figure 6.13: Click Rate of Wh button. An application window will appear which shows rate for every unit of Wh in cents. (AP-SMS_EP003)



Encrypted Rate of Wh

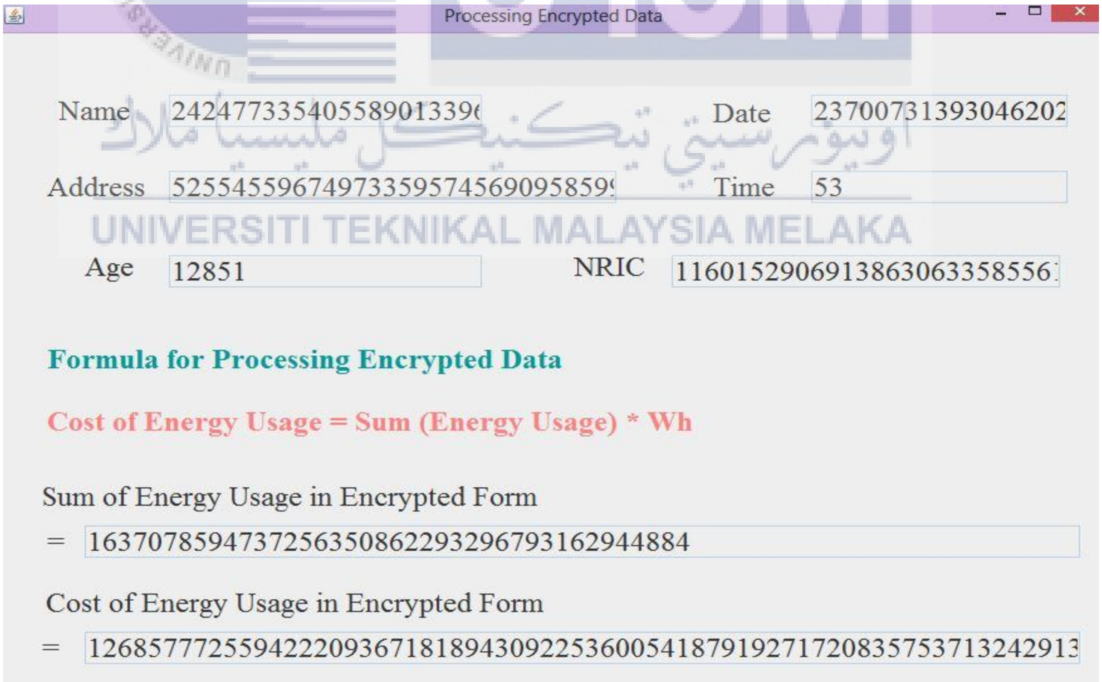
Keys for Encrypted Wh

Secret Key P 272846432456209391810382216132193824061

Encrypted Rate (Cents) for every units of Wh

= 272846432456209391810382216132193824083

Figure 6.14: Click Encrypt Rate of Wh button. Encrypted rate of Wh will show in application window. (AP-SMS_EP004)



Processing Encrypted Data

Name 242477335405589013396

Date 2370073139304620 2

Address 525545596749733595745690958599

Time 53

Age 12851

NRIC 11601529069138630633585561

Formula for Processing Encrypted Data

Cost of Energy Usage = Sum (Energy Usage) * Wh

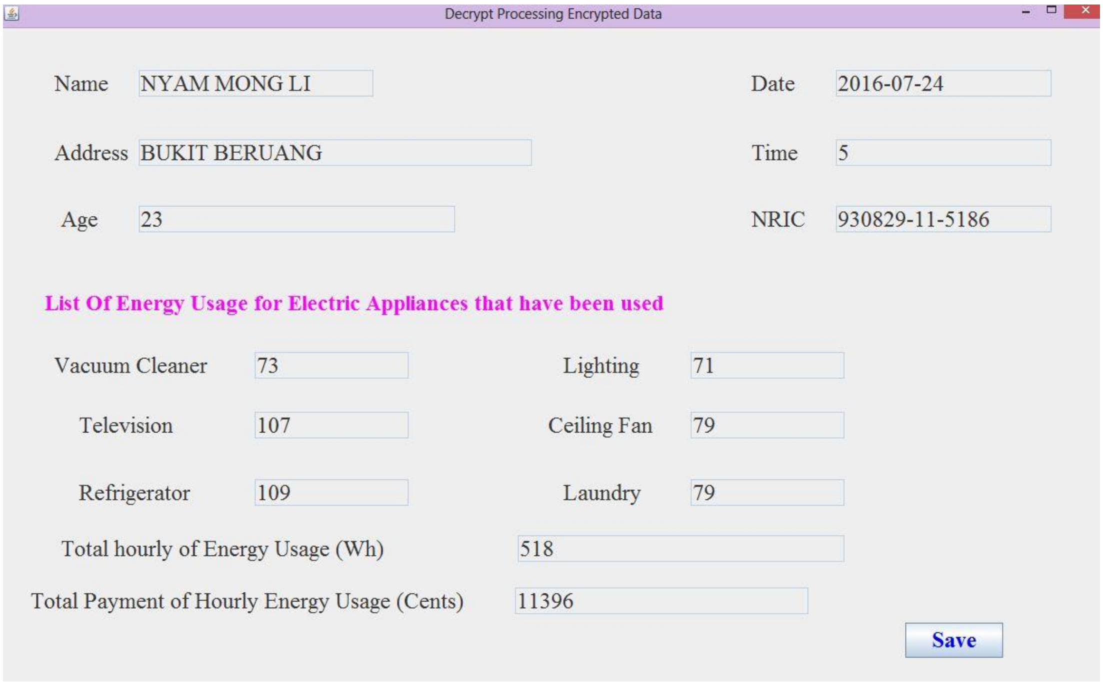
Sum of Energy Usage in Encrypted Form

= 16370785947372563508622932967931629448 84

Cost of Energy Usage in Encrypted Form

= 12685777255942220936718189430922536005418791927172083575371324291 3

Figure 6.15: Click Process Encrypted User Data button. It shows the encrypted form for the sum and cost of energy usage in an application window. (AP-SMS_AGR001)



Name NYAM MONG LI

Date 2016-07-24

Address BUKIT BERUANG

Time 5

Age 23

NRIC 930829-11-5186

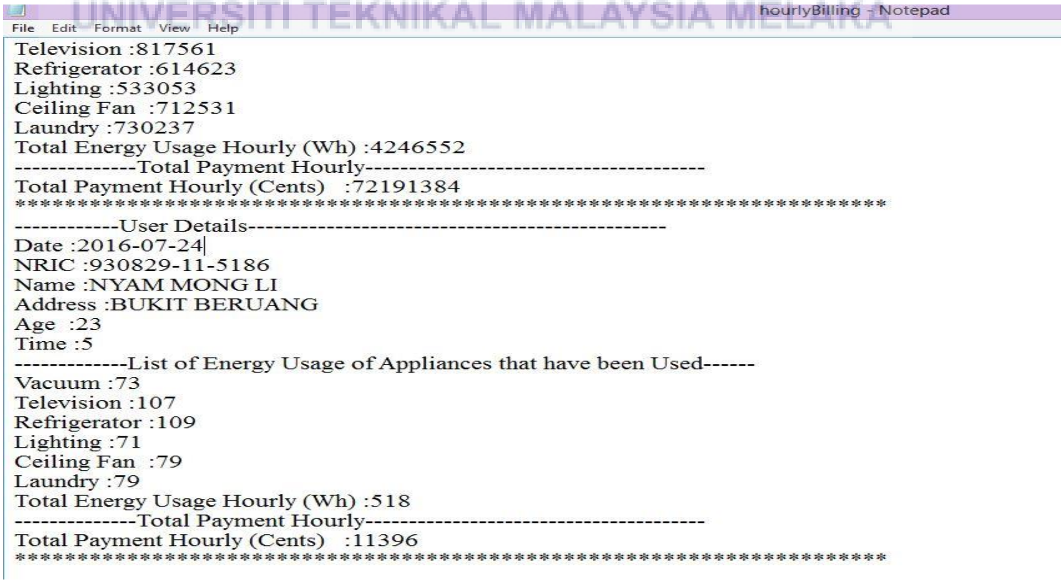
List Of Energy Usage for Electric Appliances that have been used

Vacuum Cleaner	73	Lighting	71
Television	107	Ceiling Fan	79
Refrigerator	109	Laundry	79

Total hourly of Energy Usage (Wh) 518

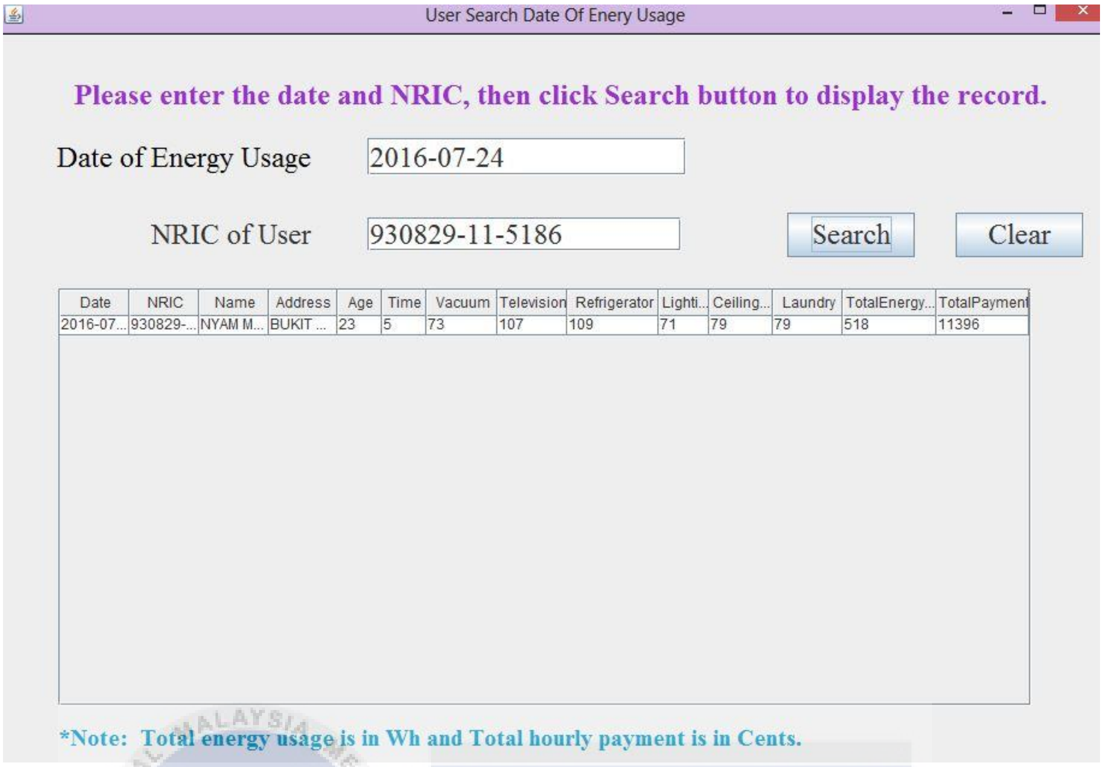
Total Payment of Hourly Energy Usage (Cents) 11396

Figure 6.16: Click Decrypt User Data button. It shows the decrypted value of energy usage for electric appliances and total payment of hourly energy usage in the plain text form. Click Save button to save the information into database. (AP-SMS_EP005)



```
Television :817561
Refrigerator :614623
Lighting :533053
Ceiling Fan  :712531
Laundry :730237
Total Energy Usage Hourly (Wh) :4246552
--------------Total Payment Hourly----------------------------------------
Total Payment Hourly (Cents)   :72191384
******************************************************************
------------User Details-----------------------------------------------
Date :2016-07-24
NRIC :930829-11-5186
Name :NYAM MONG LI
Address :BUKIT BERUANG
Age  :23
Time :5
-------------List of Energy Usage of Appliances that have been Used------
Vacuum :73
Television :107
Refrigerator :109
Lighting :71
Ceiling Fan  :79
Laundry :79
Total Energy Usage Hourly (Wh) :518
--------------Total Payment Hourly----------------------------------------
Total Payment Hourly (Cents)   :11396
******************************************************************
```

Figure 6.17: Click Generate Hourly Billing as TXT. The record can be found in notepad. (AP-SMS_EP006)



Please enter the date and NRIC, then click Search button to display the record.

Date of Energy Usage: 2016-07-24

NRIC of User: 930829-11-5186

Date	NRIC	Name	Address	Age	Time	Vacuum	Television	Refrigerator	Lighti...	Ceiling...	Laundry	TotalEnergy...	TotalPayment
2016-07...	930829-...	NYAM M...	BUKIT ...	23	5	73	107	109	71	79	79	518	11396

*Note: Total energy usage is in Wh and Total hourly payment is in Cents.

Figure 6.18: Click Search Date button. Enter all the fields in the application window. Then click Search button. The result will be displayed. (AP-SMS_USR006)

6.6 Conclusion

To conclude, the system has been examined clearly from the developer, testers and other colleagues. The application has been carefully checked, it is observed that the system fulfilled all the requirements as discussed in Chapter III Analysis. It was a successful test because all designed test cases gave positive test results and does not have any errors found on it. Thus, this project will be carry to next level before the system is ready to be use.

CHAPTER VII

CONCLUSION

7.1 Introduction

This is the final phase for completing the project. This chapter discusses the weaknesses and strengths of the system. In addition, it also mentions about the project's limitation and some propose solutions to improve the system.

7.2 Project Summarization

The objectives of the project are to ensure the encrypted data can be calculated without decryption to prevent aggregator to disclose users' information. It also allows the users to reduce energy consumption. This project has been successfully implemented. This can be seen through the developed system that has met the objectives and requirements of the project. In this project, RSA algorithm has been used for initial processing. The RSA algorithm helps to protect a key shared within a smart meter and an energy provider. The homomorphic encryption scheme has been introduced into the system to allow specific types of computations such as additions and multiplications to be carried out on encrypted data without decryption. It provides security on data even though the encrypted data is being processed by an aggregator. By using our proposed scheme, the decryption of processed ciphertext data produces the same results as operating data in plaintext form.

7.2.1 Strengths

There are some strengths found in the system in contrast to current system. This can be seen by all the data in the system has been encrypted when it transfer from one party to another. Although the cipher-text is visible to people, it is unreadable due to plain-text data has been converted to secret code. This can prevent attacker from alter and change the plaintext because the data is in cipher-text form. Besides that, the system implements homomorphic encryption algorithm which allow aggregator that acts as a third party between a smart meter and an energy provider to do calculation on the ciphertext without decryption key. Thus, aggregator could not disclose user information to other people because the data is in cipher-text form. Since an energy provider is the party that generates key for homomorphic encryption and distribute the key to smart meter for encryption purpose, other parties such as data authority, user and aggregator could not decrypt the cipher-text to obtain the original plain-text. Only a smart meter and an energy provider could decrypt the cipher-text because these two parties possess the key prime *P*. The system guarantees the data has been protected to achieve confidential and integrity. It keeps the user safe that protects users' privacy.

7.2.2 Weaknesses

Conversely, the system has its observed weaknesses from users' feedback. The size of bits for original plain-text must be less than the size of bits for key prime *P* for homomorphic encryption in order to get back correct result after decryption. Since homomorphic encryption is a type of symmetric key encryption, the smart meter has to find out a method to distribute the key to energy provider for sharing data purpose. Since we can find out a secure way to transfer key to that particular party, we might not need to encrypt the data at the beginning. Besides that, we have to distribute the key to the trusted party such as an energy provider to prevent other parties from misuse the key to decrypt the cipher-text. It would create more harm and damage consequences if the key has been compromised by other parties.

7.3 Project Contribution

The final product for this project with its title name is A Privacy-preserving Smart Metering System bring benefits for people to communicate safely between two parties when third parties is involved in it by using homomorphic encryption scheme. Although the cipher-text is visible by other parties, they still could not know the exact content of data in plain text form. It provides security features to the people. RSA is only part of the system because it is used for key sharing between a smart meter and an energy provider. To develop this project, it has been coded into a useful system to allow students and lecturers who have the passion to improve this system. Besides that, all the information has been documented during development of this system. Lecturers and students can refer to the documentation for further explanation on the system.

7.4 Project Limitation

RSA algorithm has been used for key sharing purpose from a smart meter and an energy provider. Due to its size for RSA keys which is public key and private key are much larger compare to key size for homomorphic encryption scheme, RSA algorithm will generate big numbers that increase the processing time .The system has been slow down and causing delay in time to generate public and private key respectively. In addition, the size of metering data in this system is much smaller compare to real system which has larger size for its metering data.

7.5 Future Works

The system that has been developed indeed can produce correct decrypted result after going through encryption process that uses homomorphic encryption.

Homomorphic encryption is using 128-bit for its key prime P in my system. In future work, longer key length for homomorphic encryption can be used. Longer key length does increase security because it can counter some kinds of security attack such as brute force attack. It increases the complexity of brute force iteration of the key space and factoring. Research should be done on shorten the time for RSA encryption and decryption by simplifying its algorithm. It can decrease the time taken and amount of system resources used for calculation purpose to generate large prime numbers.

7.6 Conclusion

This system provides a high secured approach by combining RSA and homomorphic encryption algorithms that can prevent aggregator which is located between a smart meter and an energy provider to disclose to outsider about the users' information. Homomorphic encryption allows specific types of computations such as addition and multiplication on the cipher-text without decryption. This can ensure confidentiality and integrity of the data. With the reference and information from internet resources, reference books, friends and lecturers, the system has met its objectives and requirements that have been written in Chapter 1 Introduction. In conclusion, this system provides some advantages to people and contributes to society for future working environment.

REFERENCES

Alamatsaz, N. et al., 2014. AgSec: Secure and efficient CDMA-based aggregation for smart metering systems. *2014 IEEE 11th Consumer Communications and Networking Conference, CCNC 2014*, pp.489–494.

Alohali, B. et al., 2015. A Survey on Cryptography Key Management Schemes for Smart Grid. , 3(3), pp.27–39.

Farah, S. et al., 2012. An experimental study on Performance Evaluation of Asymmetric Encryption Algorithms. *Recent advaces in information science*, 8, pp.121–124.

Filip, Š., 2013. New Cryptographic Key Management for Smart Grid. , (May).

Hur, J., Koo, D. & Shin, Y., 2015. Privacy-Preserving Smart Metering with Authentication in a Smart Grid. *Applied Sciences*, 5(4), pp.1503–1527. Available at: http://www.mdpi.com/2076-3417/5/4/1503.

Nsiah, A.K. et al., 2015. Embedded TLS 1 . 2 Implementation for Smart Metering & Smart Grid Applications. , 13(4), pp.373–378.

Omijeh, B.O., 2013. Design and Simulation of Single Phase Intelligent Prepaid Energy Meter . , 4(1), pp.17–30.

Robin, D., - Towards Automated Fault Management in Smart Grid.

V.Parmar, P. et al., 2014. Survey of Various Homomorphic Encryption algorithms and Schemes. *International Journal of Computer Applications*, 91(8), pp.26–32.

Zirm, M. & Niedermeier, M., The Future of Homomorphic Cryptography in Smart Grid Applications.

Zhang, R. et al., 2013. Verifiable privacy-preserving aggregation in people-centric urban sensing systems. *IEEE Journal on Selected Areas in Communications*, 31(9), p.268-278.

Zeadally, S. et al., Towards Privacy Protection in Smart Grid. , p.1-25.

Thoma, C., Cui, T. & Franchetti, F., 2012. Secure Multiparty Computation Based Privacy Preserving Smart Metering System. *44th North American Power Symposium (NAPS)*, p.1-6.

Thoma, C., Cui, T. & Franchetti, F., 2013. Privacy preserving smart metering system based retail level electricity market. In *IEEE Power and Energy Society General Meeting*.

Selga, J.M. et al., 2014. Smart Grid ICT Research Lines out of the European Project INTEGRIS. *Network Protocols and Algorithms*, 6(2), p.93.

Naab, B.J. & Loibl, A., 2014. Privacy Strategies in Smar t Metering. *Network Architectures and Services*, (August), p.107-113.

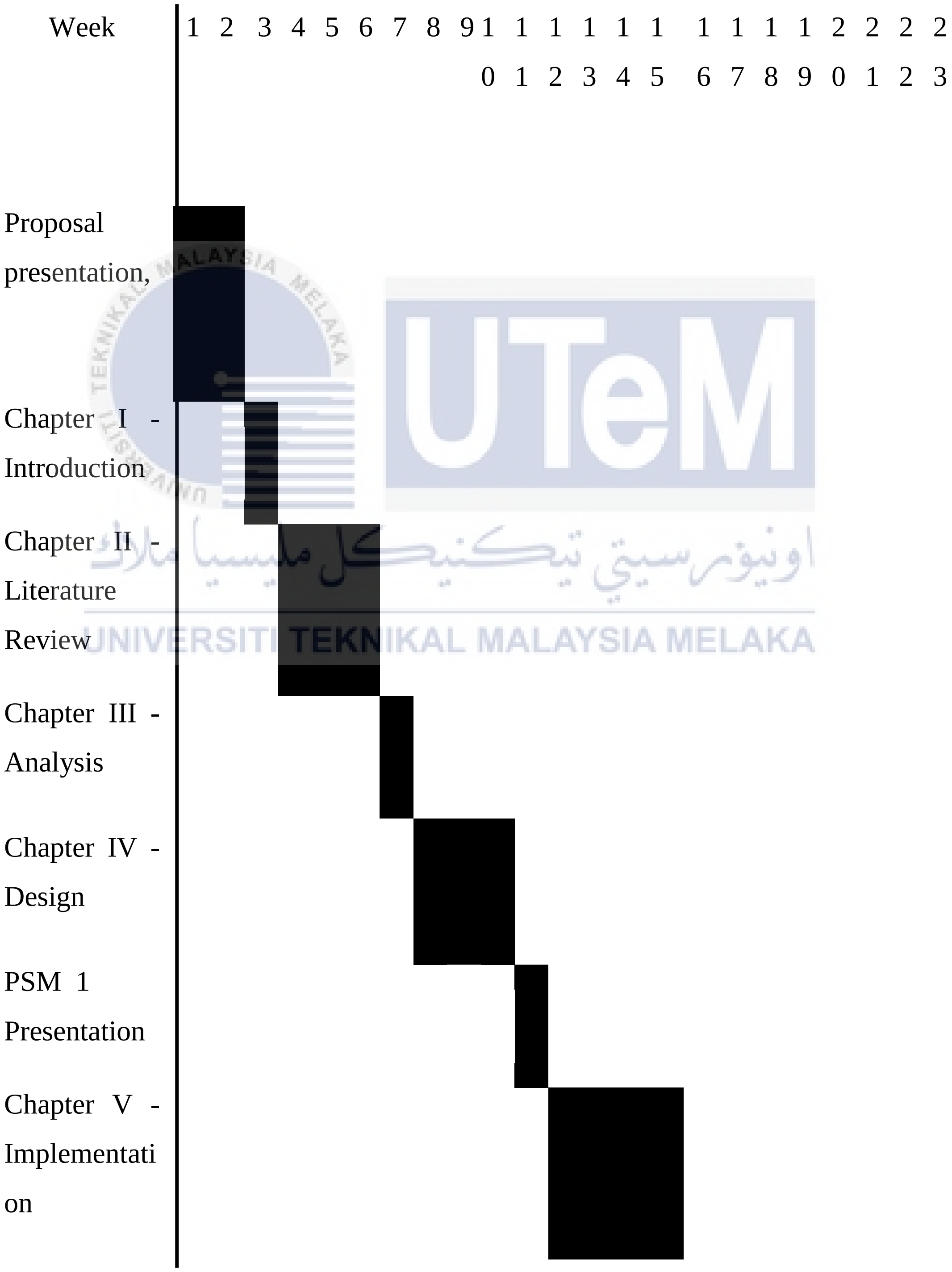
Ellenki, S.K., Reddy, S. & Ch, G.S., 2014. An Advanced Smart Energy Metering System for Developing Countries. , 2(1), p.242-258.

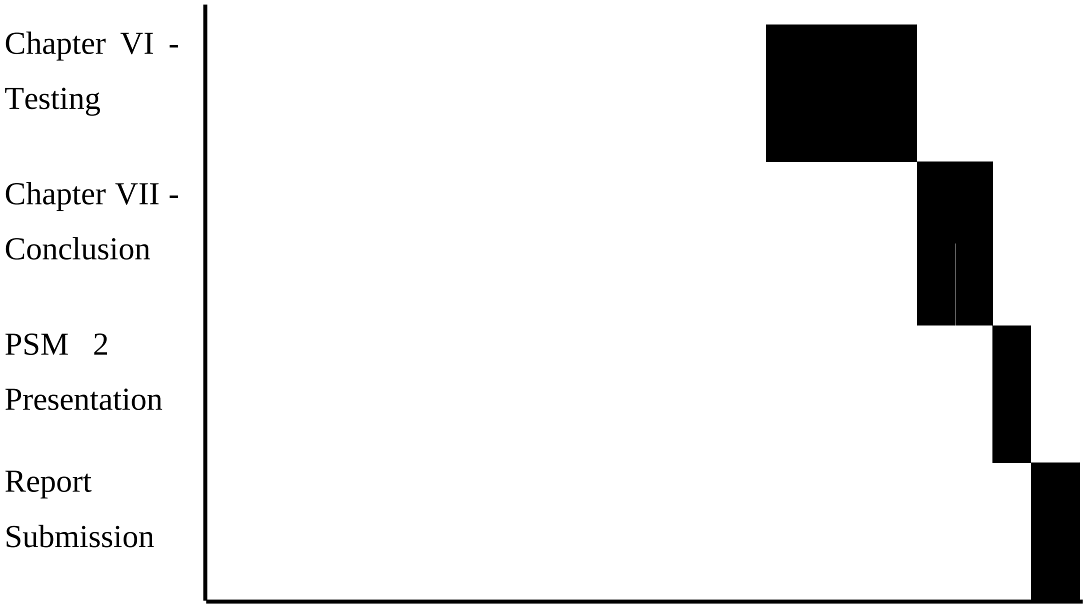
Carlos Lopez, 2015. Smart Grid Cyber Security: An Overview of Threats and Countermeasures. *Journal of Energy and Power Engineering*, 9(7), p.632-647.

Bhatia, R.K. & Bodade, V., 2014. Smart Grid Security and Privacy : Challenges , Literature Survey and Issues. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(1), p.702-706.

APPENDIX

1.1 Gantt Chart




Chapter VI - Testing
Chapter VII - Conclusion
PSM 2 Presentation
Report Submission

1.2 Homomorphic Encryption Scheme

This scheme consists of key generation, data encryption, data evaluation and data decryption.

1.2.1 Key Generation Algorithm

```
private BigInteger keyP;

private int bitlengthHomomorphic =128;

Random r = new Random();
keyP = BigInteger.probablePrime(bitlengthHomomorphic, r);
```

1.2.2 Data Encryption Algorithm

```
private BigInteger ceilingfan;
private BigInteger refrigerator;
private BigInteger encryptedCeilingFan;
private BigInteger encryptedRefrigerator;
private BigInteger keyQ;
private BigInteger keyN;
Random Ea = new Random();
private int bitlengthElectricAppliances = 6;

keyQ = BigInteger.probablePrime(bitlengthHomomorphic, r);
keyN = keyP.multiply(keyQ);


ceilingfan =
BigInteger.probablePrime(bitlengthElectricAppliances, Ea);

refrigerator =
BigInteger.probablePrime(bitlengthElectricAppliances, Ea);
```

```
encryptedCeilingFan = (ceilingfan.add (keyP)).remainder(keyN);

encryptedRefrigerator =(refrigerator.add (keyP)).remainder
(keyN);
```

1.2.3 Data Evaluation Algorithm

1) **Addition**

```
private BigInteger SumofEncryptedAppliancesValue;
private BigInteger  FinalAdditionOfEncryptedAppliances;

SumofEncryptedAppliancesValue =
encryptedCeilingFan.add(encryptedRefrigerator);


FinalAdditionOfEncryptedAppliances =
SumofEncryptedAppliancesValue.remainder(keyN);
```

2) **Multiplication**

```
private BigInteger
MultiplicationOfEncryptedApplicancesandEncryptedkWh;
private BigInteger encryptedkWh;
private BigInteger
FinalMultiplicationOfEncryptedApplicancesandEncryptedkWh;

MultiplicationOfEncryptedApplicancesandEncryptedkWh =
SumofEncryptedAppliancesValue.multiply (encryptedkWh) ;

FinalMultiplicationOfEncryptedApplicancesandEncryptedkWh
=
MultiplicationOfEncryptedApplicancesandEncryptedkWh.rema
inder(keyN);
```

1.2.4 Data Decryption Algorithm

```
private BigInteger decryptedEnergyUsage;

private BigInteger  decryptedTotalPayment;

decryptedEnergyUsage =
FinalAdditionOfEncryptedAppliances.remainder(keyP);

decryptedTotalPayment =
FinalMultiplicationOfEncryptedApplicancesandEncryptedkWh.remai
nder(keyP);
```

1.3 User Manual

Step 1: Double click to open the application file.

Step 2: Click Public Key button.

Step 3: Click Private Key button.

Step 4: Click Generate Prime *P*, Prime *Q* and *N* button.

Step 5: Click Encrypt Sk (*P*) button.

Step 6: Click Decrypt *P* button.

Step 7: Click Rate of *Wh* button.

Step 8: Click Encrypt Rate of *Wh* button.

Step 9: Click User Form button.

Step 10: Click Meter Reading in User Form button.

Step 11: Click Encrypt Meter Reading in User Form button.

Step 12: Click Process Encrypted User Data button.

Step 13: Click Decrypt User Data button.

Step 14: Click Generate Hourly Billing as TXT.

Step 15: Click Search Date button.