iSe-NFC FERRY: INTELLIGENT SECURED FERRY BOARDING SYSTEM USING NFC INTEGRATED WITH RFID TAG FOR MALAYSIA'S TOURISM

NUR HIDAYAH BINTI RAZALI

BORANG PENGESAHAN STATUS

JUDUL: iSe-NFC FERRY: INTELLIGENT SECURED FERRY BOARDING SYSTEM USING NFC INTEGRATED WITH RFID TAG FOR MALAYSIA'S TOURISM

SESI PENGAJIAN: 2016/2017

Saya NUR HIDAYAH BINTI RAZALI

mengaku membenarkan tesis Projek Sarjana Muda 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	

_____	_____
(TANDATANGAN PENULIS)	(TANDATANGAN PENYELIA)
Alamat tetap: Batu 19 ½,	NURUL AZMA BINTI ZAKARIA
Kampung Kelubi,	NAMA PENYELIA
77000 Jasin, Melaka	
Tarikh: 22/8/2017	Tarikh: 22/8/2017

iSe-NFC FERRY: INTELLIGENT SECURED FERRY BOARDING SYSTEM USING NFC INTEGRATED WITH RFID TAG FOR MALAYSIA'S TOURISM

NUR HIDAYAH BINTI RAZALI

This report in partial fulfilment of the requirement for the

Bachelor of Computer Science (Computer Security)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2017

DECLARATION

I hereby declare that this project report entitled

iSe-NFC Ferry: INTELLIGENT SECURED FERRY BOARDING SYSTEM USING NFC INTEGRATED WITH RFID TAG FOR MALAYSIA'S TOURISM

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

STUDENT : _____ _____ Date : __22/8/2017__

(NUR HIDAYAH BINTI RAZALI)

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

SUPERVISOR : _____ _____ Date : __22/8/2017__

(NURUL AZMA BINTI ZAKARIA)

DEDICATION

To my beloved parents thank you for their endless support and always stay behind me all the time. They support me mentally, physically and financially.

To my lovely supervisor for encouraging, motivating and believing in me throughout the development of this project.

ACKNOWLEDGEMENTS

Assalamualaikum Warahmatllahi Wabarakatuh and greetings

Thanks to God for the consent, iSe-NFC Ferry: Intelligent Secured Ferry Boarding System using NFC integrated with RFID tag for Malaysia's Tourism has ran smoothly and successfully.

I would like to express my sincere gratitude to my PSM supervisor, DR. Nurul Azma Binti Zakaria for accepting me under her supervision for the Projek Sarjana Muda. She guide me, motivate me and give support throughout the entire process in accomplishing the project. Without her help, I would be clueless to accomplish the project.

I would like to thanks to both of my parents, Razali Bin Juhari@Johari and Mariah Binti Adom for always motivating me and support me to keep on completing this project. I also would like to express my gratitude to my friends who help me completing this system.

My fellow friends who also went through the Projek Sarjana Muda (PSM) with me. Their help and support always brings me forward, also not to forget the lecturers and the staff in UTeM especially FTMK for giving me endless knowledges.

ABSTRACT

In this era, Radio-frequency Identification(RFID) is an electronic tagging technology that use electromagnectic fields to transfer data has evolved with the emerging of NFC technology. Near-field Communication(NFC) which is a set of communication protocols that enables two electronics devices that containing NFC tag by bringing them close to establish communication. NFC is more efficient and more secure because NFC application is used within small radius. In Malaysia, services and process for ferry's passenger entry is not systematic and disorganized. Problem that can be found is long queue and the process to get the tickets is complicated at entry of the terminal. In addition, there is improper passenger's record keeping for administrator to monitor passenger access. Thus, passenger cannot be track and identified when ferry had technical problem. The objective of this project are to design entry system interfaces that can use NFC and RFID technology, to improve passenger record management by RFID tagging and to track passenger by using NFC-enabled devices with RFID card (tag). The methodology used in the project is Waterfall model integrated with Bell-LaPadula model that is used to protect user's data access. Six steps are involved, first is the project planning which is to develop a big view of the project and defined it goals. Project analysis is the phase where new requirement for this project is explained in detail. The Bell-LaPadula model is implemented for security in project design phase. Installation and configuration involves in project implementation. All units developed in implementation phase are tested in project testing. In project maintenance phase, implementation of the project need to be ensure in order to fulfil all requirement statements. Project contributions are it can reduce time in getting the tickets and reduce human intervention. Passenger's record management can be improved as the data can be retrieved and passenger's access can be monitored. Hence, passenger can be tracked and identified when there is future problem occurred. The result of the system is it can improve Malaysia's tourism.

ABSTRAK

Dalam era ini, *Radio-frequency Identification* (RFID) adalah teknologi tag elektronik yang menggunakan medan electromagnetik untuk memindahkan data telah berkembang, dengan pembangunan pesat teknologi NFC. *Near-field Communication* (NFC) yang merupakan satu set protokol komunikasi yang membolehkan dua peranti elektronik yang mengandungi tag NFC dengan membawa mereka berhampiran untuk mewujudkan komunikasi. NFC adalah lebih cekap dan lebih selamat kerana aplikasi NFC digunakan dalam radius kecil. Di Malaysia, perkhidmatan dan proses membawa masuk penumpang feri tidak sistematik dan tidak teratur. Masalah didapati adalah barisan yang panjang dan proses untuk mendapatkan tiket adalah rumit di pintu masuk terminal. Di samping itu, terdapat penyimpanan rekod penumpang yang tidak teratur, menyukarkan pentadbir memantau akses penumpang. Oleh itu, penumpang tidak boleh dikesan dan dikenal pasti apabila feri mempunyai masalah teknikal. Objektif projek ini adalah untuk merekabentuk sistem kemasukan yang boleh menggunakan teknologi NFC dan RFID, untuk meningkatkan pengurusan rekod penumpang dengan tag RFID dan untuk mengesan penumpang dengan menggunakan peranti NFC yang dibolehkan dengan gelang tangan RFID (tag). Kaedah yang digunakan dalam projek ini adalah model *Waterfall* dan model *Bell-LaPadula* digunakan untuk melindungi akses data pengguna. Enam langkah terlibat, pertama adalah perancangan projek membangunkan gambaran besar projek dan takrifan objektif. Analisis projek adalah fasa keperluan projek ini dijelaskan secara terperinci. Model *Bell-LaPadula* dilaksanakan untuk keselamatan dalam fasa reka bentuk projek. Pemasangan dan konfigurasi terlibat dalam pelaksanaan projek. Semua unit dibangunkan dalam fasa pelaksanaan diuji dalam ujian projek. Dalam fasa penyelenggaraan projek, pelaksanaan projek perlu dipastikan memenuhi semua keperluan projek. Sumbangan projek adalah ia boleh mengurangkan masa dalam mendapatkan tiket dan mengurangkan campur tangan manusia. Pengurusan rekod penumpang boleh dipertingkatkan dimana data boleh diambil dan akses penumpang boleh dipantau. Oleh itu, penumpang boleh dikesan dan dikenal pasti apabila ada masalah berlaku. Hasil daripada sistem ini adalah ia boleh meningkatkan pelancongan Malaysia.

TABLE OF CONTENTS

LIST OF TABLE

TABLE **TITTLE** **PAGE**

LIST OF FIGURE

LIST OF ABBREVIATIONS

RFID	- Radio-frequency identification
NFC	- Near-field communication
PC	- Personal Computer
ERD	- Entity relationship Interface
UID	- Unique Identifier

CHAPTER I

INTRODUCTION

1.1 Project Background

Physical security is protection of personnel, hardware, software, networks and data from physical actions and events that could cause serious damages to agency or institution. The protection include protection from disasters, burglars, theft and terrorism. In mitigating the damages there are a lot of protections measures and access control that can be implemented such as RFID, biometrics, detection sensor and etc. In this system, the technology that will be used is NFC with integrated RFID tag (RFID card). NFC is the evolution of RFID.

Radio-frequency identification (RFID) is a technology on automatic identification and data collection. RFID also provide automatic identification and real time data acquisition that do not require human resources. Nowadays, RFID has been used in wide variety of areas such as animal identification, transportation payment tool and also tagging and tracking for luggage in airport. RFID provide a lot benefits in today's transportation that can increases the flow of passengers, and vehicles, reduces number of cash payments and also reduces ports congestion.

In this era, Radio-frequency Identification(RFID) has evolved. The revolution is the emerging of NFC technology. NFC is ‘Near-field communication’ which is a set of communication protocols that enables two electronics devices that containing NFC tag or RFID tag by bringing them close to establish communication. NFC is more efficient rather than other RFID technology and more secure because NFC application is used within small radius. NFC usually benificials for credit cards as alternatives in terms of security. Moreover, NFC usually used for transfering data such as pictures, videos, documents and etc between two NFC-enabled devices.

Here in Malaysia, NFC and RFID implementation is low when it comes to transportation areas. There are a lot of problems arise when peak hours. Sometimes, people get stranded at the terminal when there is technical problem to the transport or when there is natural disaster happen. There are a lots of problems such as pickpockets, people get free ride, stealing, unauthorized access and etc.

For this system, it will focus more on marine transportation which is ferry’spassenger supervision system. Example ferry embarkation and disembarkation current flow in Mersing, Malaysia is first passenger have to get ticket from the agent. Second, passenger need to register the ticket at counter. Third, fill up names on another counter. Fourth, pay the fees for Jabatan Laut at another counter. Fifth, pay fees for Taman Laut at another counter and lastly, que get boarding pass which the que is very long. The process have a lot human interventions and the process is complicated. The current system may have low security because people tend to get free ride. Bad people for example thug or drug dealers easier for them to pass through the entrance because of the inefficient security of access controls.

1.2 Problem Statement (PS)

In Malaysia, services and process for ferry's passenger entry is not systematic and disorganized. There are a lot of negative comments from the tourist about the ferries' service in Malaysia. So, it is a shame for Malaysia as popular tourism country. Problem that can be found is long queue and the process to get the tickets is complicated at entry of the terminal. Moreover, there is problem when people tends to cut queue. Thus, it takes time to get tickets and need a lot of human interventions. Second, the problem is the passenger's record keeping management. There is no proper passenger's record keeping that can allow the administrator to monitor passenger access. Hence, passenger cannot be track and identified when ferry had technical problem.

Table 1.1: Problem Statement

PS	Problem Statement
PS_1	Long queue and the process to get the tickets is complicated at entry of the terminal. Second, there is no proper passenger's record keeping that can allow the administrator to monitor passenger access. Passenger cannot be track and identified when ferry had technical problem.

1.3 Project Question (PQ)

Table 1.2: Project Question

PS	PQ	Project Question
PS_1	PQ_1	How to study about entry system interfaces that can use NFC and RFID technology: NFC reader and RFID tag?
	PQ_2	How to study about record management that can be enhance by RFID tagging?
	PQ_3	How to track passenger by using NFC-enabled device and RFID tag?

1.4 Project Objective (PO)

Objectives are recommendations and solutions for the problem statement. The project is built to improve ferries' services for Malaysia's tourism. The project objectives (PO) are based on the project question (PQ).

Table 1.3: Project Objectives

PS	PO	Project Objectives
PS_1	PO_1	To design entry system interfaces that can use NFC and RFID technology.
	PO_2	To improve passenger record management by RFID tagging.
	PO_3	To track passenger by using NFC-enabled device and RFID tag.

1.5 Project Scope

For this system, scope that will be covered are:

i. Focused on water transportation which is ferry, small range ferry that is used for tourism.
ii. The system will use portable device NFC reader/writer will be NFC-enabled device.
iii. The system can be used at different points, where there will be entry and exit process involved.
iv. RFID card will acts as tag, so user will wear it as double confirmation for identification.
v. This system enabled for tracking, it can be used to track people using timestamp that tally with time passenger's arrival.

1.6 Project Contribution (PC)

This project was built to improve the current ferries' services in boarding system for Malaysia's tourism. New methods should be implemented with advanced technology. With aid of NFC and RFID the current issues can be solved. Table 1.4 below is the project contribution after implementation of NFC and RFID.

Table 1.4: Project Contribution

PS	PQ	PO	PC	Project Contribution
PS1	PQ1	PO1	PC1	Improve the entry system interfaces that can use NFC and RFID technology.
	PQ2	PO2	PC2	Improve passenger record management by using RFID tag.
	PQ3	PO3	PC3	Passenger can be tracked by using NFC-enabled device and RFID tag.

1.7 Expected Output

The expected result from the iSe-NFC Ferry: Intelligent Secured Ferry Boarding System using NFC integrated with RFID tag for Malaysia's Tourism is problem of long queue can be decrease and the process to get the tickets can be more easier. Thus, it can reduce the time in getting the tickets and reduce human intervention. Moreover, passenger's record management is improved as the data can be retrieved and passenger's access can be monitored. When there is data inspection from the authority, data can be retrieved. Hence, passenger can be tracked and identified when there is future problem occurred. The authority can tracked and identified how many passenger are there in order to provide boats. In addition, thug or drug dealers that want to use the ferry to do their crimes can be arrested. So, hopefully this system can improve Malaysia's tourism.

1.8 Project Organization

In this PSM report there will be seven chapters which are Chapter I: Introduction, Chapter II: Literature Review, Chapter III: Methodology, Chapter IV: Design, Chapter V: Implementation, Chapter V1: Testing and lastly Chapter VII: Conclusion.

Chapter I: Introduction

In this chapter, there will be background of the project, problem statements, project question, project objectives, project scope, project contribution, expected output from the project and thesis organization.

Chapter II: Literature Review

For this chapter, it will discuss on related works or previous works, explanation and details of project, also analysis of the current problems that supported with reading materials and conference papers. All the topics from the papers and problems will be analyzed to justify and make more understanding of the project.

Chapter III: Methodology

In this chapter, it will focus on methodology of the project, steps and phase in the making of the project. It will also consist project planning which are project schedule and milestones to achieve goals of the project for the given time.

Chapter IV: Design

In this chapter, software and hardware will be introduced to be used in implementing the project. It also consist environment setup, architecture network design, experimental design and software design.

Chapter V: Implementation

For this chapter, it will discuss on implemented project methodology by showing the implementation and installation of hardware and software. Also, sample outputs will be covered as well.

Chapter VI: Testing

This chapter will test and validate the implemented project. It also will explain methods in testing and analyzing collected data. It will discuss and determine on the results and analyze the problem that occur during development of the project.

Chapter VII: Conclusion

As for this chapter, it will summarized and give an overall picture of limitations, contributions and future works. Also it will compiles all chapters and state contribution of the projects for future works.

1.9 Conclusion

Generally, this chapter portray the strategy that will be performed to develop this project. Improvements can be done from the current problems and this project can satisfy all the objectives. Also, by taking the scope of the project, it can help in completing the project development and enhanced ferries' services. With advanced technology, ferries' services can be improved and more effective to passengers that use the services.

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

In this chapter will explain on review of previous researchers and current discourse on particular problem that has been identified by other researcher. The particular problem that has been identified is ferry boarding system. The current low services of ferry boarding system such as security and access control can affect the quality of the system. Hence, it may lead to other incidents.

Security is safety or protection from attackers or enemies or people that will do harm intentionally or accidentally is the objective of security. Multi-layers of security is a must to have a successful organization to protect its network sturcture, operations, physical infrastructure, people, communications, and valuable information (Whitman & Mattord, 2014). While physical security is protection of personnel, physical items such as hardware, software, network infrastructures, objects and data from unauthorizes access and misuse that could cause serious damage to organizations or institutions, for example, terrorrism, natural disasters and vandalism (Whitman & Mattord, 2014).

Access controls definition is to ensure only authorized individuals have access to information and prevent modification of information by unauthorized users. The purpose of access controls is to preserve internal and external consistency of data (Peltier, 2013). There are many physical security controls that can be used for access controls such as biometric, NFC, RFID and motion detector and etc. In this project we will focus on NFC and RFID that implement on ferries' services; ferry boarding system.

The current services and process for ferry's passenger entry in Malaysia is not systematic and disorganized. There are a lot of negative comments from the tourist about the ferries' service in Malaysia. Thus, reputation of Malaysia as popular tourism country will decrease. Problem that can be found is long queue and the process to get the tickets is complicated at entry of the terminal. Moreover, there is problem when people tends to cut queue. Thus, it takes time to get tickets and need a lot of human interventions. Another problem is the passenger's record keeping management. There is no proper passenger's record keeping that can allow the administrator to monitor passenger access. Hence, passenger cannot be track and identified when ferry have technical problem.

Solution for the current problems, this project is developed to improve the current problems using new technologies to overcome all problems that stated earlier. This project will make process to get tickets will be less and human intervention will be reduced. Record management for passenger's information will be more manageable. Thus, passenger can be tracked by RFID card when undesirable incidents happen.

In this chapter, in-depth discussion about literature review on ferry boarding system will explain in subsequent section. Figure 2.1 below shows the structure chapter that will briefly disscuss.

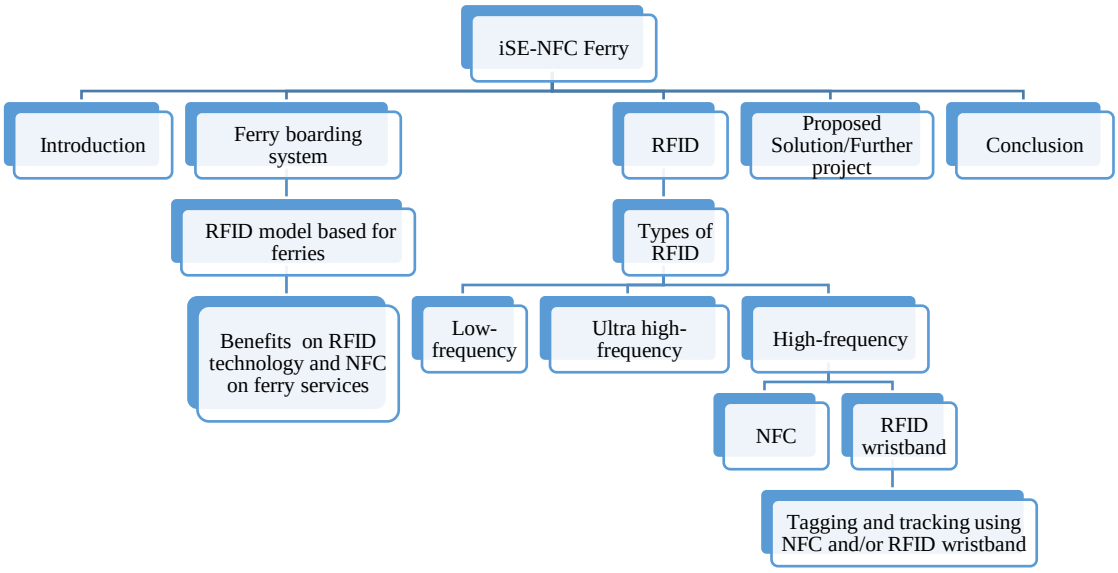


Figure 2.1: An overview of literature review

2.2 Ferry boarding system

This section will explain about ferry boarding system in Malaysia. The ferry boarding system is currently ineffective and not systematic which can lead to many problems; which are:

- Passenger consume more time in getting the tickets and require a lot of human interventions.
- There is no proper record management for passenger's data.
- Passenger cannot be tracked if there is accident happen because of no proper record management.

Ferry boarding system should be upgrade and more digitalize as it can give condusive services to the passenger, also it can ease the the organizations in managing the passenger's access. Technology that will be use to solve this matter is by using NFC and RFID card.

2.2.1 RFID access control model based for ferry

Ferry boarding system is the embarkation and disembarkation of passenger and/or vehicles into the ferry from one terminal to the other ferry terminal. From figure 2.2 below, according to Mišković et al., (2016) it demonstrates RFID system for boarding system for passengers and vehicles on ferry where RFID tags will be registered in and store information in the database. It is to control passenger for access control and prevent them from getting free ride. Passenger need to check that they board on the correct ferry by checking the port destination. It is to prevent passenger got misunderstanding in boarding the wrong ferry. RFID tag can differentiate automatically between passenger and vehicles. In the respective diagram, RFID system will calculate current capacity of vehicles and passenger on the ferry. From this RFID system, maritime agency and captain of the ship have real-time access and communnication to ensure business policies at bay.

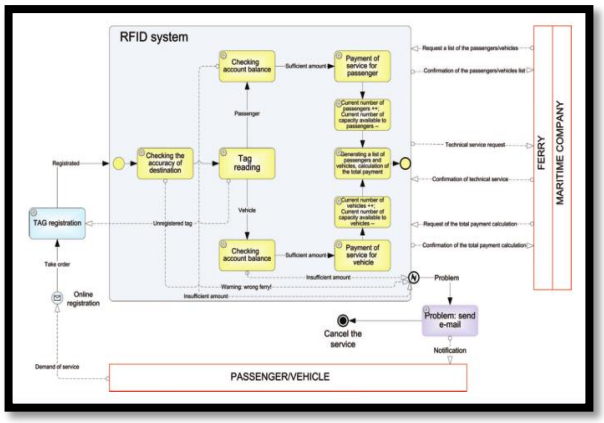


Figure 2.2: Example access control model using RFID technology (Mišković et al., 2016)

2.2.2 Benefits on RFID technology and NFC on ferry services

With advanced technologies, quality of ferry services will increase. These are the benefits on RFID technology and NFC on ferry services:

- The used of RFID and NFC technology can reduce the time of queing at the counter for passenger and prevent passenger got into the wrong ferry. Thus, it will prevent passenger form getting free ride for security issue. This will ensure the optimum operation and efficiency in eliminates the possiblity of fraud.

- It can differentiate and identify between passenger and vehicles where it can calculate the capacity of the ferry. This help the maritime company in getting real-time monitoring the access of passenger and vehicles at the ferry. It ensuring the business policies at bay.

- It help the system in keeping record of the user properly and efficiency.

- It also help in security drill in tracking passenger if there is a possibility of accident happen.

- With RFID and NFC technology it allow cash free services, means passenger can keep track of the amount of the fund paid at the beginning of the boarding process, so passenger do not have to pay for the services provided on the ferry.

2.3 Radio Frequency Identification (RFID)

Radio frequency identification is a technology based on wireless detection of electromagnetic signals. RFID have three main components which are tag, a reader and a middleware. Transponder is RFID tag programmed electronically with information. According to Mišković et al. (2016), based on the simplified RFID model system, RFID tag located inside a object such as RFID wristband. It will irradiated with radio waves emitted by the reader. Then, RFID tag convert the received signal into electronic energy. The tag send its information about its identified object to a RFID reader. RFID reader converts the information received into digital format and transmit it to central computer. Middleware at between reader and processing unit to manage the information flow. Computer receives and processess information and stores them in database. Then, data transfer between reader and central computer through communication interface. Life will be easier, productive and efficient when RFID is implemented. Figure 2.3 below shows basic model and coressponding software of a RFID system(Mišković et al., 2016).

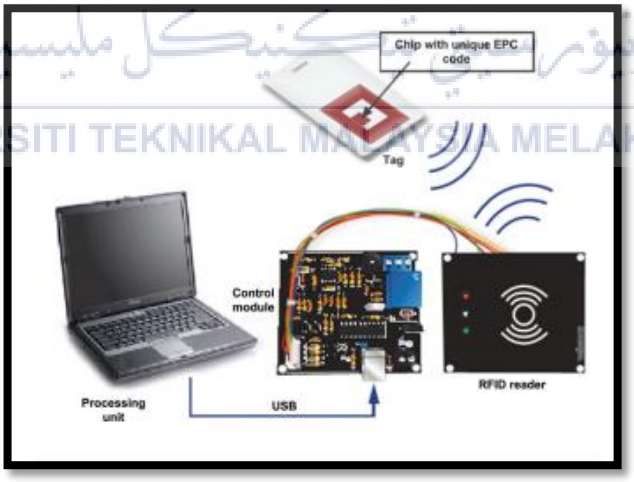


Figure 2.3: Basic model and corresponding software of a RFID system for registration of embarkation(Mišković et al., 2016)

2.3.1 Types of RFID tags

There are three types of RFID tags which are low-frequency, high-frequency and ultra-high frequency. RFID available in passive mode and active mode. Passive mode do not have power supply, so it need wave radiation from RFID reader as it cannot transmit signal. While active mode have power supply and it can transmit signal. Tags that is in passive mode is low cost that active tags. Nowadays, passive RFID usually used for security, identification, tagging, tracking and access control. It is suitable to implement passive RFID tags for this project.

Low-frequency RFID

Frequencies used are 125 and 134.2 kHz for lower-frequency RFID. Lower-frequency RFID can penetrates water and body tissue, so it usually implement on animal tracking. Resistant to metal surrounding but not resistant to electrical noise. Transponders for low-frequency RFID are more expensive than high-frequency transponders. It have lower data transfer rate and cannot simultaneously read multiple transponders (Rfid, 2012). Figure 2.4 below are the example of low-frequency RFID tags(CNRFID, n.d.).

Figure 2.4: Low-frequency RFID tags(CNRFID, n.d.)

Ultra high-frequency RFID

Ultra high-frequency have range 300 MHz to 3 GHz which is ideal for supply chain because it can reads at longer distances. It have higher data rates intransfering data that high-frequency RFID and have longer read distances (10 m) but have poor performance with liquids and metals where lower-frequency and high-frequency can works in liquids and metals surrounding. Ultra high-frequency tags easily embedded into solid non-metallic items. Figure 2.5 below show example of ultra hihgh-frequency RFID tags(CNRFID, n.d.)

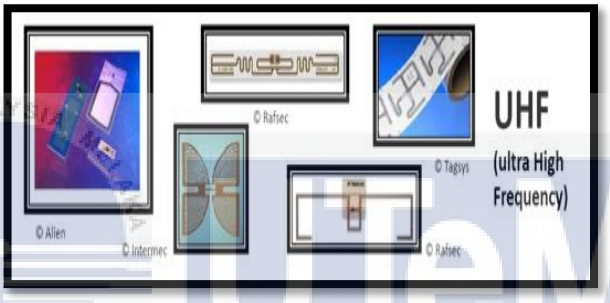


Figure 2.5: Example of ultra high-frequency RFID tag(CNRFID, n.d.)

High-frequency RFID

Utilization of high-frequency RFID globally used and it operates at 13.56 MHz such as NFC that have the same frequency. It is low cost for tags comparing to low-frequency and the tag is small. It can send signal through water and body tissue. With its size that is small, so it can easily embedded and printed, also written on any labels. It have higher rate in data transfering than low-frequency and can read more than 50 tags simultaneously at once. High-frequency usually used for security system such as access control with integration of biometrics. Besides that, it capable in identifying and tracking for items. According to Rfid (2012), Octopus card widely use by people in Hong Kong for public transit system because high-frequency RFID is suitable for contact-less smart cards. It is suitable solution for lower cost identification and have ability to read multiple tags application. Figure 2.6 below show example of high-frequency RFID tags(CNRFID, n.d.).

Figure 2.6: Example of high-frequency RFID tag(CNRFID, n.d.).

2.3.2 Near-Field Communication

Near-field Communication (NFC) is an advanced wireless technology of RFID in short range of four to ten centimeters for communication iniatially developed by Sony and Philips. NFC receiver recieve the data through magnetic inductive computing if lies in close and could generates a radio frequency in 13.56 MHz band in limited short range communication. NFC permits a data between 106 and 424 kbps and it is fully compatible with RFID tags in high frequency (HF) band, according to (Riekki et al., 2013) . As it communicate through 13.56 MHz, the two devices will generate the transmission in active mode, it allow transfering of data more flexible and more secure. During transmission energy is coupled between transmitter and receiver to form air-core transformer as in wireless communication. Figure 2.7 below show on magnetic induction(Riekki et al., 2013).

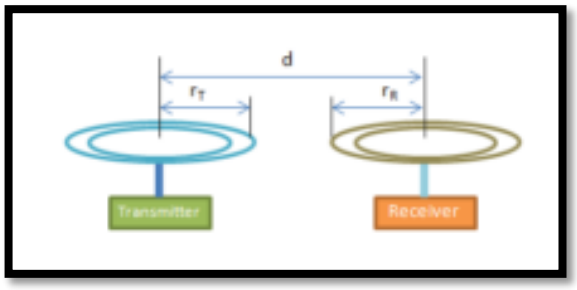


Figure 2.7: Magnetic induction(Riekki et al., 2013).

According to Jain & Dahiya (2015), it earned internationally acredited standard ISO/IEC 21481 which meant in the future it become known technology all over the world and will have various application.

Modes of communication in NFC

NFC devices consist NFC tag, NFC mobile phone or NFC-enabled device and backend server. It operates in two modes, passive mode and active mode. NFC tag act as a passive device that contains information that other devices can read but cannot read information by itself. NFC tag will store data and read by NFC enabled devices. Active devices on the other hand, it can read and send information. NFC tag embedded in items such as Point-of-Sale (POS), smart posters and etc. It have small chip hidden behind a sticker with NFC logo in order to make users aware of its existence and it also contain small data such as contact number, valuable information and etc. NFC mobile phone or NFC-enabled device is embedded with NFC chips within it to read NFC tags. Then NFC mobile phone or NFC-enabled device will communicate with backend server. Communication need to be secure as the service provided can be various(Timalsina et al., 2012). In this project, NFC tag will act as passive mode. Figure 2.8 below show NFC works in active and passive mode(Timalsina et al., 2012).

Figure 2.8: NFC works in active and passive mode (Timalsina et al., 2012).

NFC modes of operation

NFC communicate in three types of operation, which are peer-to-peer mode, reader/writer mode and card emulation mode(Alsaedi, 2015).

- Peer-to-peer mode

Peer-to-peer mode allow two devices communicate with each other in

transfering information. It is usually used for device pairing, networking and etc. Figure 2.9 below show peer-to-peer mode operation(RFIP Blog, 2017).

Figure 2.9: Peer-to-peer mode(RFIP Blog, 2017)

- Reader/writer mode

As for reader/writer mode, a NFC device read data from one NFC tag and will write data on detected NFC tag at a time. In this mode, data will be transfer between the device. Example of this mode is advertisement posters. A person taps his or her NFC-enabled device such as smartphone at the NFC tag, he or she can access the link provided by the poster and can read the information about the advertisement. Figure 2.10 below show reader/writer mode operation(RFIP Blog, 2017).

Figure 2.10: Reader/writer operation(RFIP Blog, 2017)

- Card emulation mode

The last type of operation is card emulation mode. In this mode, NFC-enabled phone act as contactless smart card that widely known as debit cards, identity cards and etc. This can reduce the percentage of user carrying contactless smartcard. Recently, the used of this mode is raising as it can integrate with signifficant application. Example of signifficant application is payment application. However, in this project reader/writer mode will be implemented. Figure 2.11 below show card emulation mode operation(RFIP Blog, 2017).

Figure 2.11: Card emulation mode(RFIP Blog, 2017).

Advantages of NFC

NFC uses lower consumption of energy and it have shorter range in transmitting data up. NFC is magnetic inductive based coupling that makes the connectivity more faster. NFC is low cost and compatible with RFID. NFC also provide fast transaction which is suitable for public transportation, tourism and etc. To make NFC device to interact with each other, it need to be pairing in three mode (Patil et al., 2014). These are the reason to choose NFC as the main development for this project to enhance ferry services. Below is the table 2.1 show comparison between NFC, Bluetooth and IrDA(Patil et al., 2014).

Table 2.1:Differences between NFC, Bluetooth and IrDA(Patil et al., 2014)

	NFC	Benefits of NFC	Bluetooth	IrDA
Network Type	Point-to-Point	Easy Setup, pairing = bringing close	Point-to-Point Multipoint	Point-to-Point
Range	<0.1 m	Safe, suitable for crowded areas	10m	1m
Speed	424kbps		721kbps	115kbps
Set-up time	<0.1s	Fast transaction e.g. for public transport	6s	0.5s
Modes	Active-active, active-passive	Reader mode and card like mode	No	No
Compatible with RFID	Yes	Can work with existing infrastructure	No	No
Costs	Low	Affordable for most devices	Moderate	Low

Comparison NFC and RFID

Both RFID and NFC can operates at 13.56 MHz, NFC interact in two way communication while RFID in one way communication. It is true that NFC have limited communication than RFID that have distance up to 1 m. That makes NFC more secure that RFID. In providing flexibility and intelligent technology NFC available for mobile phones that have NFC-enabled (Chopade et al., 2016). However, in this project there will be integration between NFC technology and RFID card. Table 2.2 below shows comparison between RFID and NFC.

Table 2.2: Comparison RFID and NFC(Chopade et al., 2016).

RFID	NFC
Frequency range 13.56 MHz (High freq) and 902-928 MHz ultra-high frequency.	Operates at frequency 13.56 MHz.
One way communication.	Two way communication.
Can be used for communication between devices at a distance upto 1m.	Limited to close proximity communication (10 cm).
Tags can be scanned simultaneously	Tags cannot be scanned simultaneously.
Are not available in mobile phones	These are available in mobile phones.

NFC application in daily routine

The usage of NFC is increasing, that is implemented in many application for daily routine to provide flexibility, reduce time consume of services or process, reduce human intervention and make life more easier, also playful.

NFC based patient identification by using NFC wristband

NFC become important in modern hospital information system in improving care quality(Köstinger et al., 2013). Identification of patient crucial for clinical process. The patient identification can be identify based on three areas, which are the admission time, bedside during round and treatments during stay at the hospital. The system used is openEHR which is a ward round system by using NFC based integrated with NFC wristband that patient need to wear them. During ward round, physicians will identify patients with NFC-enabled phone. OpenEHR was developed using Android SDK version 4.0 to ensure compability for all smartphones. Figure 2.12 below show Patient identification workflow(Köstinger et al., 2013).

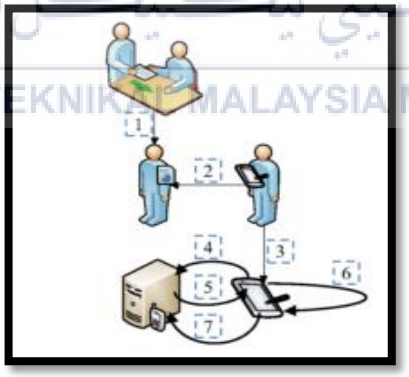


Figure 2.12: Patient identification workflow(Köstinger et al., 2013).

From the workflow, first, patient will wear NFC wristband that have information on patient's EHR ID. Next, physician will read NFC wristband by tapping the NFC-enabled phone to NFC wristband. NFC tag information read by NFC-enabled phone and NFC health application will be launched to read EHR ID. It will request EHR from backend server.

Card emulation mode for payment

As NFC can operate in card emulation mode, there are many NFC payment ecosystem has been introduced in the world. Example of NFC payment ecosystems are Apple Pay, Samsung Pay, Google Wallet and many more. The most successfully contactless scheme that NFC provide is public transportation. For ecosystem transition it consist three types NFC-payments. Physical SE-based payments, HCE-based payments and 'direct' payments are the ecosystem transition, but direct payments do not involve financial payments(Penttilä et al., 2016). Figure 2.13 below, is the payment model that visualize the detailed payment model (Penttilä et al., 2016).

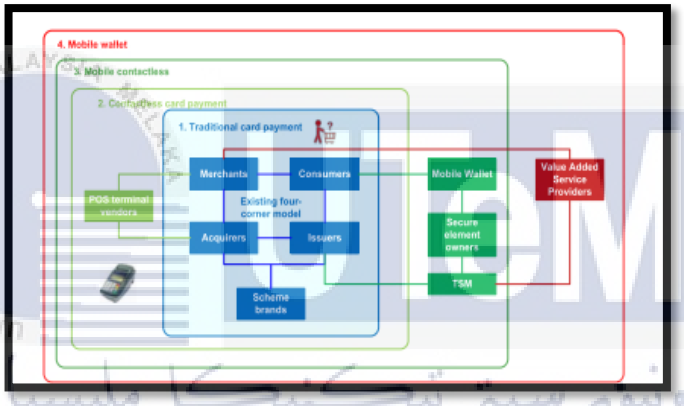


Figure 2.13: Traditional payment to mobile pay(Penttilä et al., 2016).

NFC nametags for playful learning

Other application that widely used NFC based is in education. According to Riekki et al., (2013) by touching nametags with NFC phones can be a playful approach in education in Finland. It is implemented for kindergarten student as a new medium for learning. An application is developed for three-to-five-year-old children. The application works when children NFC-enabled phone is tapped to the NFC tags, then it will give feedback after it tapped. The NFC-enabled phone used is Nokia 6131 NFC phone. Nokia 6131 NFC have sufficient capabilities and it fits to children's hands.

Kindergarten's staff will attach children's nametags on furnitures, also they equip children's nametags with star icons that have NFC tags to identify tag locations. After the children tapped it to the star icon, a poster of an animal will display at the phone and it instruct student to the application. It consist two modes of learning: exploring mode and practicing mode. Exploring mode will be "Fox Game" that display a hand holding phone that tapped to star icon. Meanwhile, practicing mode is "Rabbit Game" that display a name and if there is no nametag is tapped within a period, the phone will say "Great" and show name. It will say "Please try again later" if wrong nametag is tapped. Figure 2.14 below shows, NFC phone tapped to star icon(Riekki et al., 2013).

Figure 2.14:NFC tag is placed behind the star icon(Riekki et al., 2013)

2.3.3 RFID wristband

RFID wristband is a high-frequency RFID tag. In this project it act as a tag that operates in 13.56 MHz which is high frequency, same as NFC technology. It is a suitable integration between RFID and NFC technology. Moreover, RFID wristband can act as security wristband that control identification and tracking of passenger/user in the ferry services. RFID chip is attached inside wristband to store information. Figure 2.15 below is a RFID wristband.

Figure 2.15: RFID wristband

Tagging and tracking with NFC and/or RFID wristband

According to Mišković et al.,(2016) Royal Carribean company used RFID technology for a cruise ship. Passengers had to buy a waterproof RFID wristband before boarding that performed manually by staff using handheld RFID reader and all passenger luggage is tagged with UHF RFID tag. RFID system will connecting to an application. In this cruise ship system, RFID wristband will identify user's age automatically in serving alcoholic beverages. All the service free will deducted from the funds paid to the wristband, so it is cash-free service. For safety, it enabling for identification and tracking passenger when there is security drills. Figure 2.16 below shows the usage of RFID wristband(Mišković et al., 2016).

Figure 2.16: Usage of RFID wristband(Mišković et al., 2016)

There is another study on locating dementia patients using NFC wristband it is to prevent dementia patients from getting accidents when going outdoors (Huang et al., 2015). When outdoors, an alert will be sent out when there is unusual activity is detected also, there is an integration with Global Positioning System (GPS) in detecting lost patients, so it can pin point their locations. They used encrypted chip embed inside the wristband where concerned passerby can use their NFC-enabled phone to read and extract the information from the NFC wristband. Help from integration of GPS and internet technology they can automatically get notify if the dementia patients is the member of a high-risk group and if a report need to be made. Then, an alert will securely transmitted through HTTPS to call for help if dementia patients is a member of high-risk group. Figure 2.17 below explaining the steps in reporting process(Huang et al., 2015).

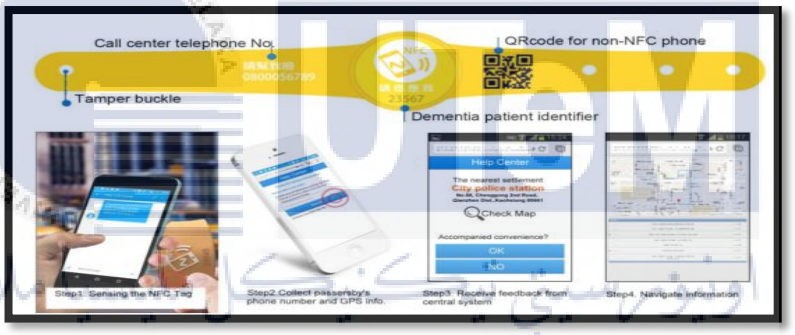


Figure 2.17: Steps in reporting process(Huang et al., 2015).

2.4 Proposed Solution/Further project

Figure 2.18 show the overview of further project for iSe-NFC Ferry.

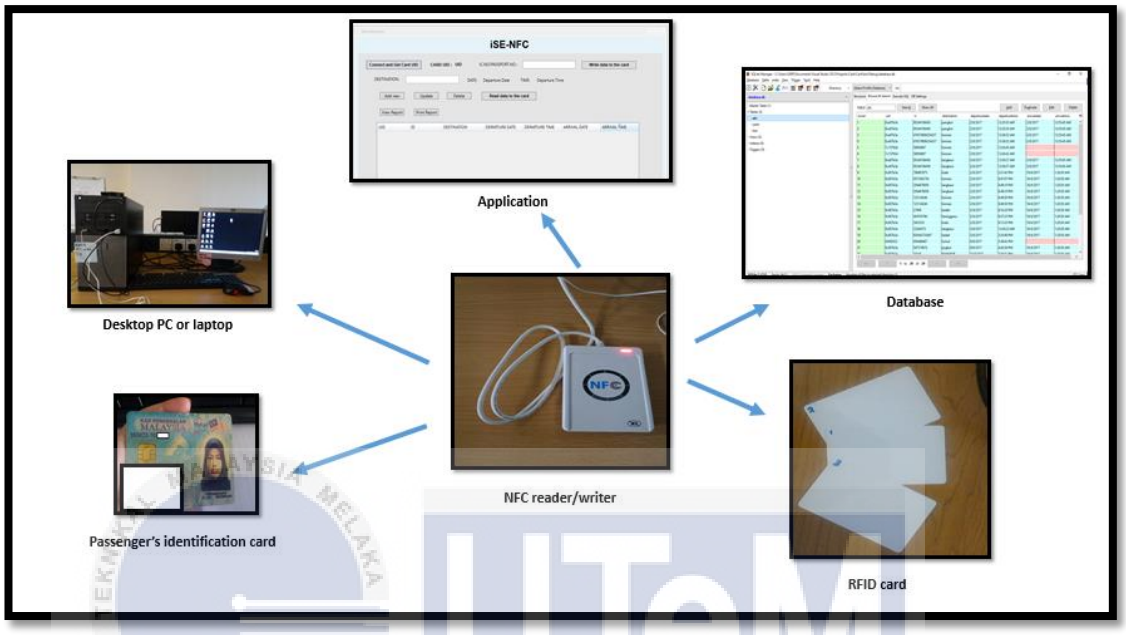


Figure 2.18: Further project for iSe-NFC Ferry

Based on the previous studied paper had been discusses, for further project is to develop iSe-NFC Ferry:Intelligent Secured Ferry Boarding System using NFC integrated with RFID tag for Malaysia's Tourism that can improve Malaysia ferry's services for tourism. For this project there will be no payment transaction occur but focussing on passenger identification, tagging and tracking. Also there will be an advanced technology use, which is NFC that integrate with RFID card. User/passenger will keep RFID card given by staff at the counter after he/she give their identification card for the staff at the counter to insert their credentials in the iSe-NFC Ferry application. NFC reader/writer will be attached at the personal computer (pc) or laptop and their information will be uploaded to the database through iSe-NFC Ferry application. This project will reduce human intervention and at same time can improve record management of passenger access. Thus, it can track people when there is possibility unexpected accident happen.

2.5 Conclusion

In this chapter consist study of previous research such as methods, techniques, technologies and hardware that being used from the previous research. All the research sources are from books, online journals and previous final year project research. Lastly for the outcome from the previous study, a futher project/solution is proposed. The proposed solution is explained briefly with an overview of the project. For next chapter, proposed solution will be explained with details within the stages in methodology.

CHAPTER III

METHODOLOGY

3.1 Introduction

In chapter three, which is methodology states methods that this project use and explains activity for every stages based on the project flow. This chapter is about project tools and requirements for simulation testing and implementation. All tools and requirements for project have been decided in choosing the best framework for the project to run smoothly and to secure the system of this project.

3.2 Project Methodology

The project methodology is the guidance to make sure the project can be handled in right manner with accurate flows. Waterfall methodology have been decided as the best methodology for System Development Life Cycle (SDLC). The methodology has been called Waterfall methodology because of it sequential design process. It is simple and easy to understand. Also, it is basically used for small project and have no certain requirements. In conclusion, this methodology is chosen as it is very suitable for this project because it is simple and easy to understand and use. For some reason, this methodology works well for smaller projects and security features

can be added. This project consist six phases as shown in Figure 3.1 is about activity involve in each phases of the Waterfall model.

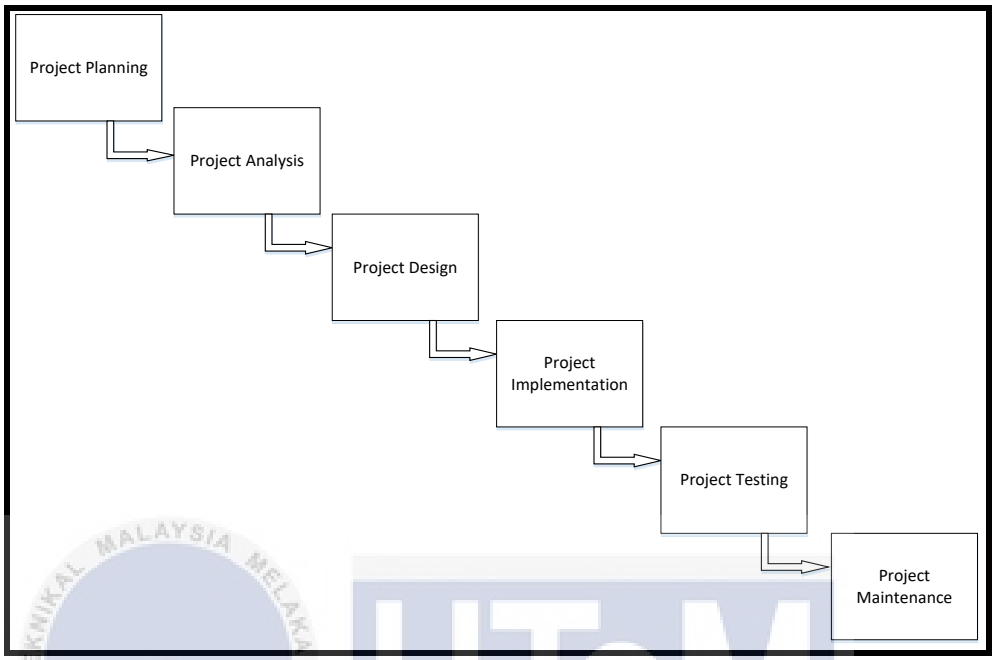


Figure 3.1: Phases in Waterfall Methodology.

Each process creates product or deliverables. Every conscious decision is made to continue the project implementations or to drop it at every phases. The products or deliverables act as milestones for progress and in continuing the project's viability.

3.2.1 Project Planning

For phase 1, which is Project planning. The purpose of this phase is to develop a big view of the project and defined it goals. This phase starts with review on the current situation of the passenger boarding system at ferry terminal. Then, identify the current technology used for the passenger entry for boarding system. Next, determine the suitable software and hardware to run the project.

1. Hardware
    - RFID card
    - NFC reader and writer

- Laptop/Desktop PC

2. Software
    - Microsoft Visual Studio 2012
    - SQLite Manager
    - ADO.Net
    - Finisar SQLite
    - C# language

3.2.2 Project Analysis

In this phase, new requirement for this project is explained in detail. It is crucial to identify and analyze the current problem. Besides that, target user is identified in this phase which is focus on ferry's passenger, staff whom in charge at ticket counter and administrator. Next, all software and hardware need to be considered in this chapter. Passengers' review and previous researches/works are used as the techniques to gather data for this phase.

3.2.3 Project Design

Implementation of Bell-LaPadula model for security in this project design is involved. Bell-LaPadula model implied in this project related to access right for difference user. No read up implies no user such as a staff can read, view and tamper data that is confidential, secret and top secret. A staff only can may read, view and tamper unclassified data. Only staff who have the authority of the data can read, view and tamper based on the privilege of top secret, secret, confidential and unclassified. Figure 3.2 below shows Bell-LaPadula model.

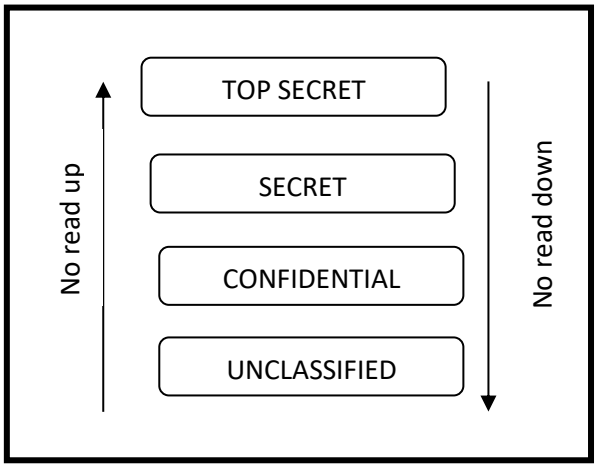


Figure 3.2 Bell-LaPadula model

There are three users which are administrator, staff and passenger. Data that need to be secured is administrator's data, staff's data and passenger's data. Administrator can read and write staff's data, where staff can write passenger's data and for passenger, they only can read their data. Figure 3.3 below show the level security property of every user.

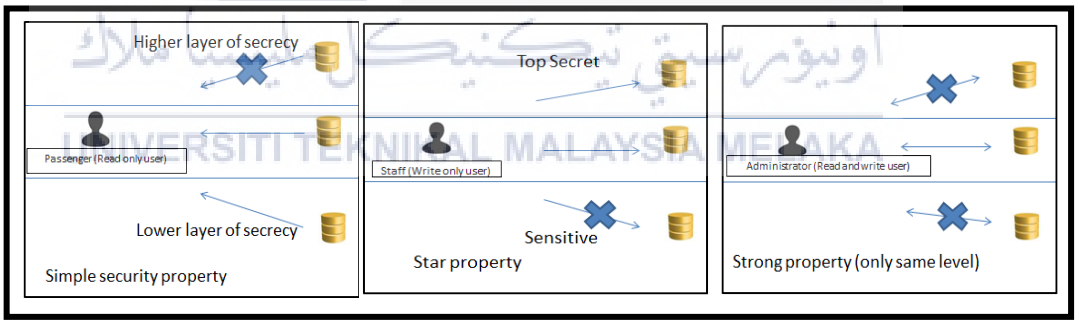


Figure 3.3 Level security property of every user

3.2.4 Project Implementation

Installation and configuration involve in this phase. In developing the system, it involves utilization of NFC technology integrated with RFID card conducts entirely. System interface is designed and developed using C# langage. The data collected from the process of inserting passenger identification card credentials into the iSe-NFC application before it stores in the database. The database server is SQLite Manager.

3.2.5 Project Testing

All units developed in implementation phase is tested in this phase. Post integration of entire system is tested to find any errors and failures. This phase can test tagging process, record keeping process and tracking process. Tagging process involves when RFID card is tapped with NFC reader. Record keeping process involves when passenger data is stored and matched with data in the record. Tracking process includes when timestamp of passenger embark are matched in the system and its tally with the time the passenger arrive. There are two flow processes which are inflow and outflow process. Inflow process is the process of boarding of passenger from the entry location. Process start when passenger gives their identification card to the staff in charge at the counter. Then the identification card credentials are inserted in the iSe-NFC Ferry application at the counter. It identify passenger and it records the relevant details of the passenger in the database. Passenger record is identified in the system. RFID card is tapped on NFC reader/writer to insert relevant passenger credentials in it and RFID card is given to the passenger. Outflow process is the process when passenger is arrived at the ferry terminal on the exit location. Upon the arrival of passenger, passenger must tap their given RFID card with NFC reader/writer at the exit location. Then, if the data is matched with the record, passenger can proceed with returning the RFID card at the counter.

Figure 3.4 below show the passenger entry testing process and Figure 3.5 show passenger exit testing process of the system.

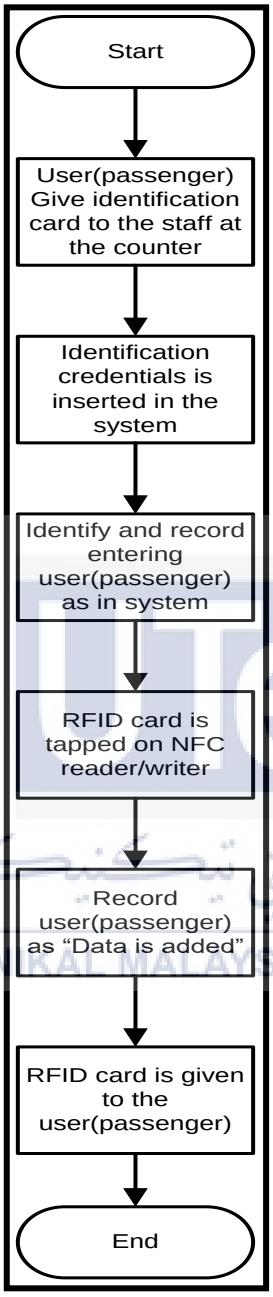


Figure 3.4: Passenger entry testing process

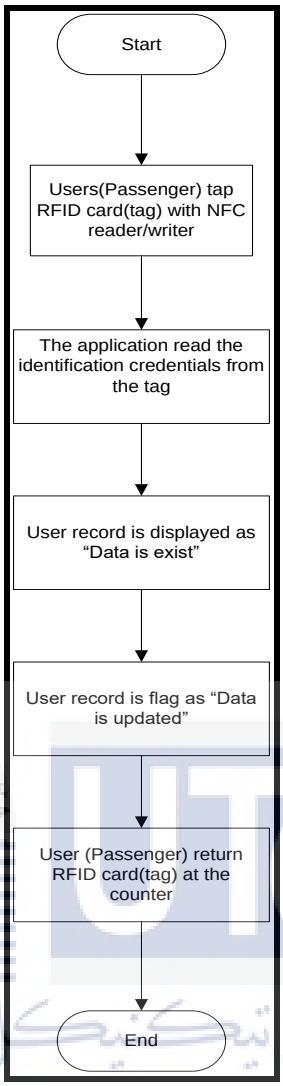


Figure 3.5: Passenger exit testing process

3.2.6 Maintenance

In this last phase, developers need to ensure that implementation of this project is in order to fulfil all requirements statements. This phase deals with changes that need to be made if there is any modification should be made to improve this system. This phase shows data collected to define effectiveness of the tagging, tracking and record keeping. Result of this project is concluded after the simulation is completed. All findings and results of this project are documented and concluded in a report.

3.3 Project Schedule and Milestones

For this project, Gantt chart is used to illustrate activities display against time. It shows the schedule of start time and end time for each activities of the project. It is to make sure the project is developed efficiently and accordingly, and all the project goals can be achieved The Gantt chart is designed according to weeks, from week one until last week which is final presentation. The Gantt chart is aided with milestones to keep track of the project development.

Table 3.1: Gantt chart

Week / Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Identify Problem and Define Objective															
Study and Research															
Design the Project															
Develop the Project															
Testing															
Maintenance the Project															
Document All the Project															

Table 3.2: Milestones

Week	Activity
1 13 Feb – 19 Feb **Meeting 1**	Proposal PSM: Discussion & Submission using PSM Online System Proposal assessment & verification
2 20 Feb -26 Feb	Proposal Correction/Improvement (Chapter 1)
3 27 Feb – 5 Mac **Meeting 2**	Chapter 1 (System Development Begins)
4 6 Mac – 12 Mac	Chapter 1 Chapter 2
5 13 Mac – 19 Mac	Chapter 2
6 20 Mac – 26 Mac **Meeting 3**	Chapter 2 Chapter 3
7 27 Mac – 2 April	Chapter 3 Chapter 4
8 3 April – 9 April	**MID SEMESTER BREAK**
9 10 April – 16 April	Chapter 4 Project Demo
10 17 April – 23 April **Meeting 4**	Chapter 4 Project Demo

11 24 April – 30 April **Demonstration**	Project Demo
12 1 May – 7 May	Project Demo and PSM1 Report
13 8 May -14 May **Meeting 5**	Project Demo and PSM1 Report Presentation schedule
14 15 May – 21 May	Project Demo and PSM1 Report
15 22 May – 28 May **Final Presentation**	**FINAL PRESENTATION & PROJECT DEMO**

3.4 Conclusion

In conclusion, project methodology, project schedule and milestones are crucial in completing the project. Project methodology illustrated the waterfall model which was used in this project. This model is an easy and suitable model to implement for the development of the project. All the phases in project methodology are important in order to run project smoothly and accordingly. Project schedule and milestones became guideline to work efficiently according to the time and assisted in achieving all the project expectation within the time. In the next chapter, analysis and design of the system will be discussed.

CHAPTER IV

ANALYSIS & DESIGN

4.1 Introduction

Discussion about the design of this project will explained in this chapter. This chapter focus on analysing software requirements, hardware requirements and the flow of project to have a better understanding in analysis stage. Any related design will explain in detail to carry out the project smoothly.

4.2 Problem Analysis

Current problem of ferry's passenger entry in Malaysia

The services and process for ferry's passenger entry (Passenger boarding system for ferry) in Malaysia currently not systematic and disorganized. In which, there is a lot of human intervention. Therefore there are many problem arise. The problem that arise is long que and the process to get the tickets is complicated at entry of the terminal. In addition, there is no proper passenger's record keeping that can allow the administrator to monitor passenger access. Lastly, as there is no proper record keeping for passenger. When future problem occur, when the ferry had technical problem such as sinking or passenger missing, passenger cannot be track and

identified. Figure 4.1 is the simplified problem of ferry's passenger entry in Malaysia.

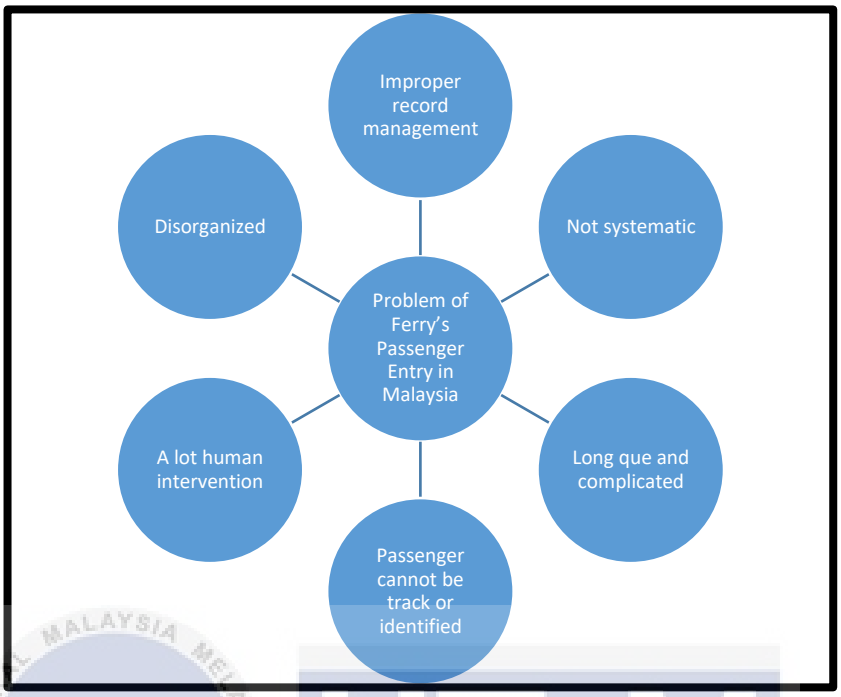


Figure 4.1 Simplified problem for current ferry's passenger entry in Malaysia

New proposed ferry's passenger entry in Malaysia

This project is developed to enhance and increase the effectiveness of current entry's passenger entry in Malaysia. It require two process; inflow proccess and outflow process This system start with passenger give their identification card to the staff at the counter. Staff will insert passenger's credentials at the iSe-NFC Ferry application. A staff at the counter have to access application interface to add and save information. NFC reader/writer is attached to the desktop pc or laptop. RFID card is tapped with NFC reader/writer to insert identification credentials in the RFID card. Then the information is saved and stored in the database. After that RFID card is given to passenger. Passenger can be tracked with timestamp, passenger's destination and also with passenger identification number that temporary stored in RFID card. Figure 4.2 show the inflow process.

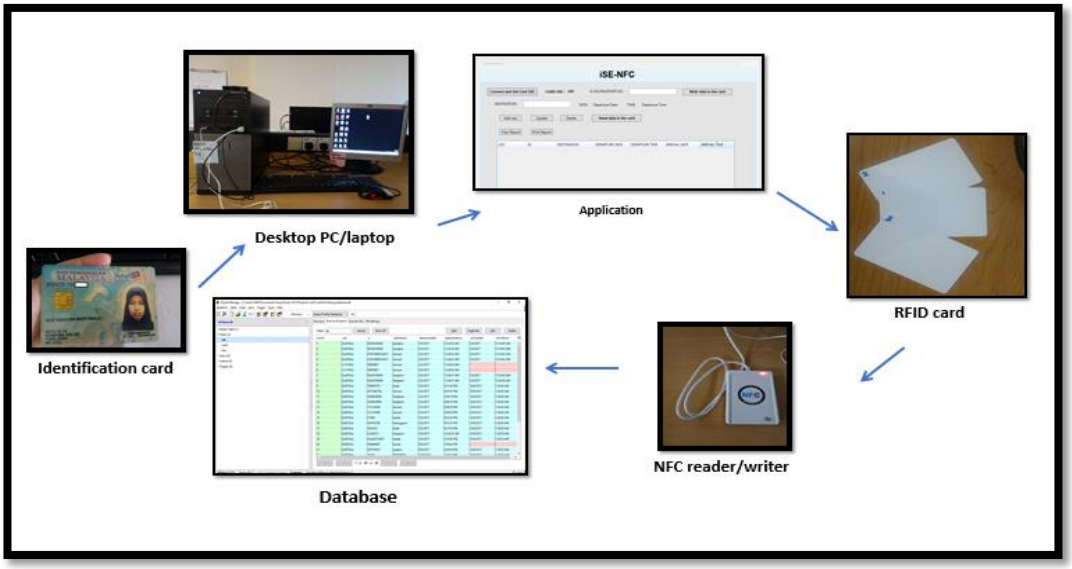


Figure 4.2: Inflow process

Upon arrival, passenger have to tapped their RFID card with NFC reader/writer. The reader is attached to the desktop pc. Then a staff at the counter access the iSe-NFC Ferry-application interface to verify the information. Next, the verfied information is saved in the database. Figure 4.3 show the outflow process.

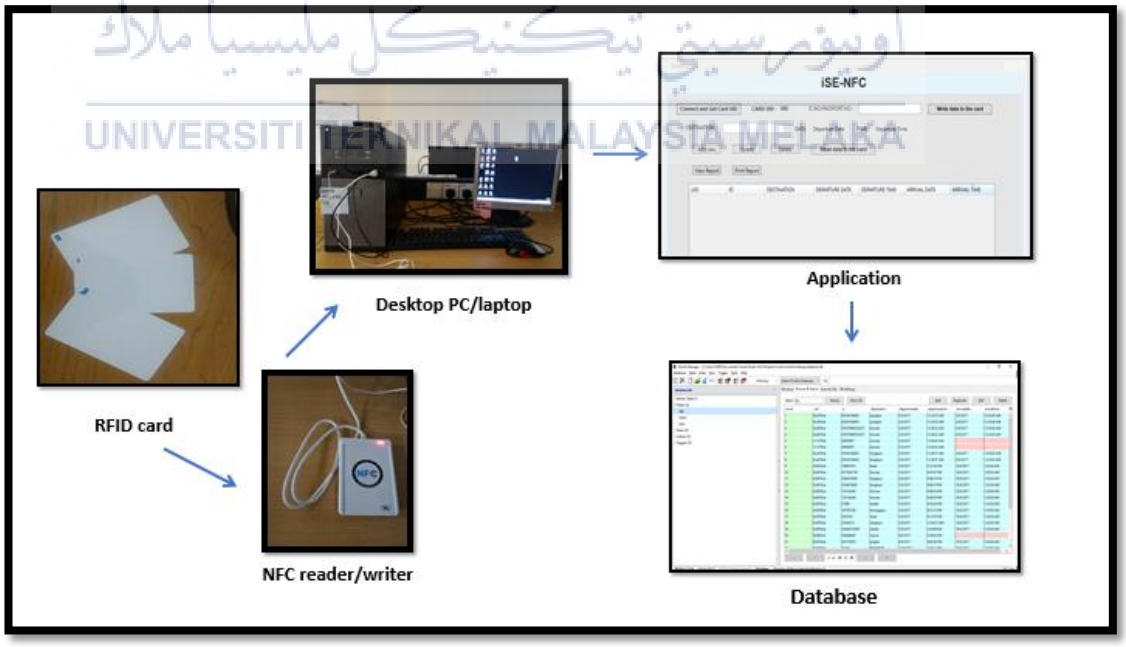


Figure 4.3: Outflow process

4.3 Requirement Analysis

This section explained about requirements that used in project that include NFC process, inflow process and outflow process.

4.3.1 Data Requirement

The input of the project is passenger's identification card, timestamp, RFID card and data dictionary. At the beginning of the process, passenger gives their identification card to the staff at the counter and the staff will insert passenger's information at iSe-NFC Ferry application. Identification card's information is the input of passenger's information. Timestamp act as the indicator for duration of ferry's journey from one destination to the other. RFID card is given to the passenger and passenger will keep them temporary. There is information such as passenger identification number stored in the RFID card's chip to track and identified passenger. Data dictionary is the list and type of data stored in database.

4.3.2 Functional Requirement

This section explains on functional requirement of the project that includes NFC process, inflow process and outflow process.

In this project, Figure 4.4 below show how NFC reader/writer is used and how the information is process and stored. First, a staff at the counter insert passenger's information at the iSe-NFC Ferry application. RFID card is tapped to NFC reader/writer to store identification number and timestamp temporarily in the RFID card. LED light at the NFC reader/writer turn yellow and a sound is produced indicates information is read or written. The information appeared at the interfaces to show the details of boarded passenger. After a staff verifies and save the information, the information will store in the database.

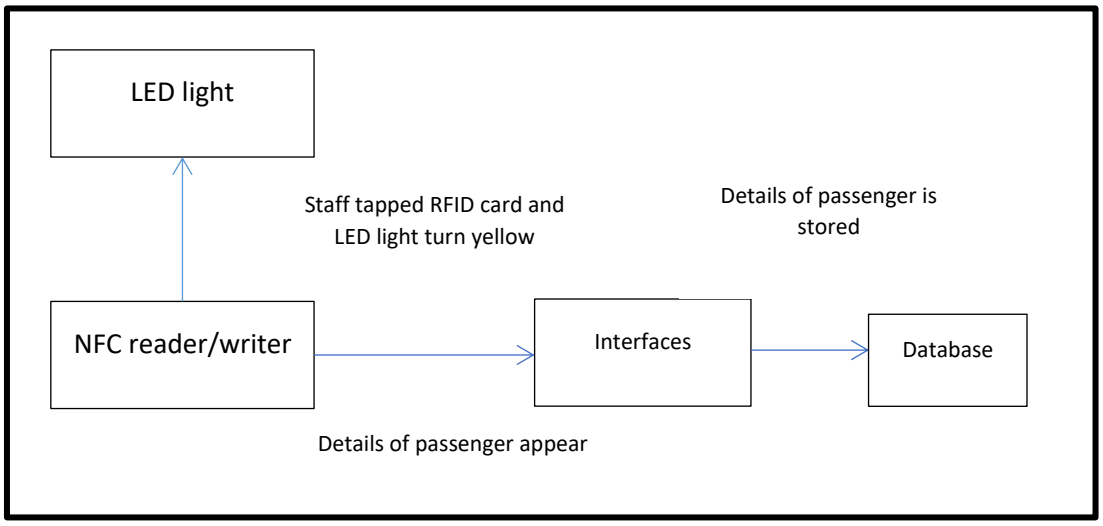


Figure 4.4: NFC tagging process diagram

Figure 4.5 below show the inflow process for passenger entry. Passenger give the identification card and staff at the counter insert passenger's information at the interface. Identification card act as input of details and RFID card act as a tag. RFID card is tapped to the NFC reader/writer. The details will notified as in system on interfaces before the details save and store in database. RFID card is given to passenger and passenger keep the RFID card. RFID card act as ticket to enter the ferry. After that, the details is saved and stored in database.

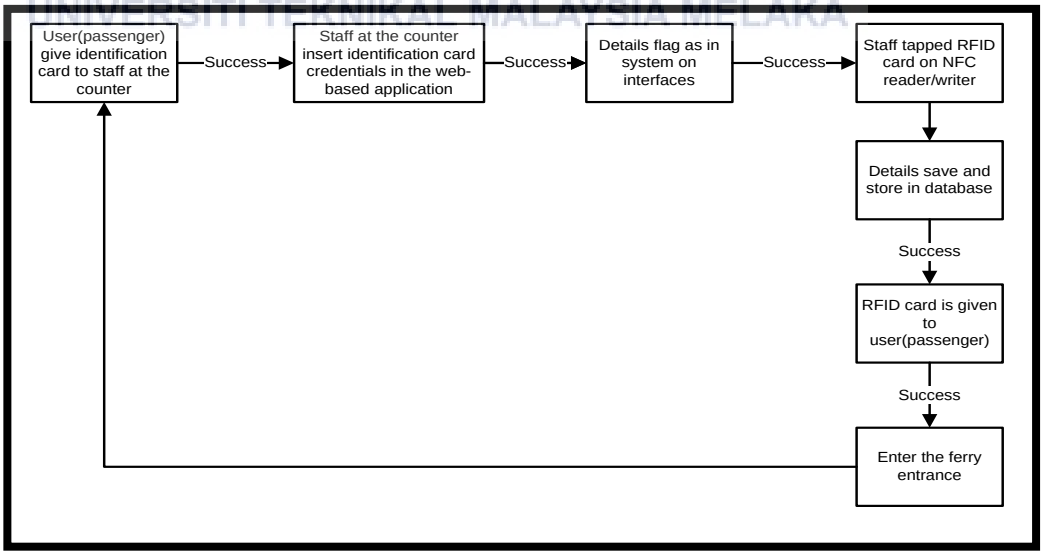


Figure 4.5 Inflow process for passenger entry

Figure 4.6 show the outflow process for passenger exit. Upon the arrival. Passenger exit the ferry and go to the counter to tap the RFID card on NFC reader/writer. Details will notified as “Data is exist” in system on the interface. Next, details is updated before save and stored in the database. After the process is succeeded, passenger may retrieve the RFID card and leave the ferry terminal.

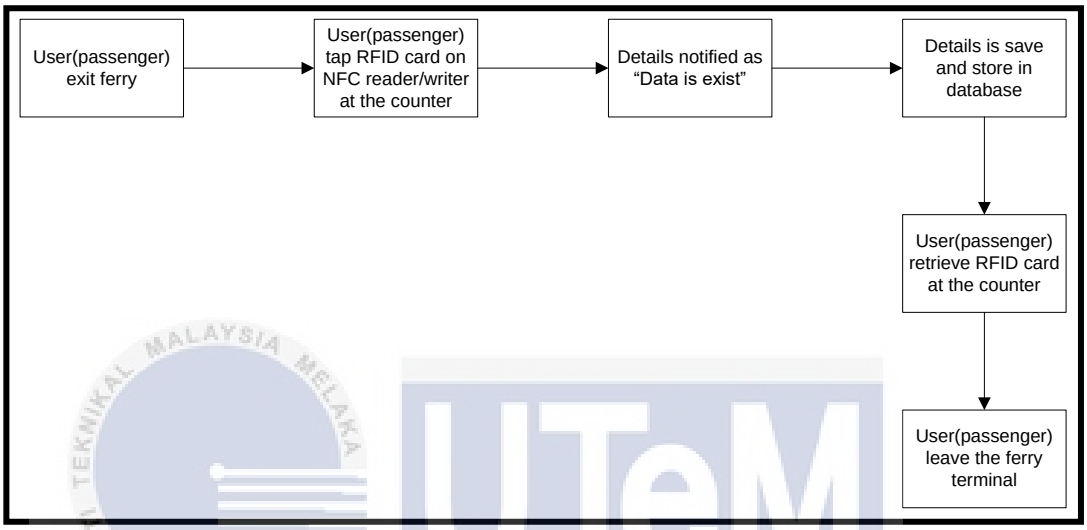


Figure 4.6 Outflow process for passenger exit

4.3.3 Other Requirement

Software Requirement

a) C# language

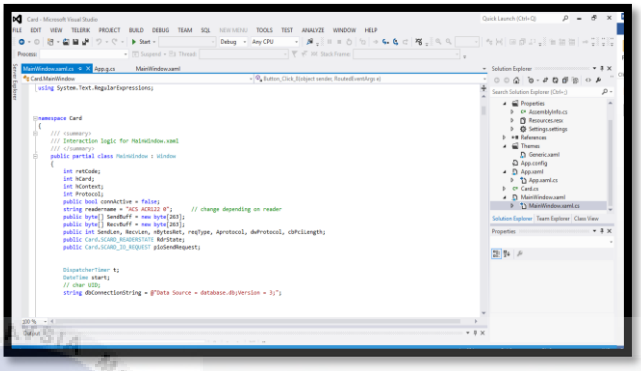


Figure 4.7: C# language

C# language is used as the programming language to develop a iSe-NFC Ferry application. C# language is a multi-paradigm programming language encompassing strong typing, functional and object-oriented programming disciplines. iSe-NFC Ferry application will integrate with database to insert data, record and retrieve data from database.

b) SQLite database

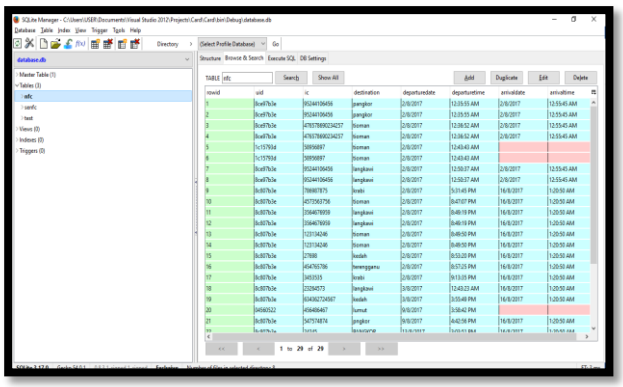


Figure 4.8: SQLite Database

SQLite will be the database for this project. SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite reads and rite directly to ordinary disk files. SQLite database support tables, indices, triggers and views contained in disk file. It is popular choice for application format.

c) Finisar SQLite

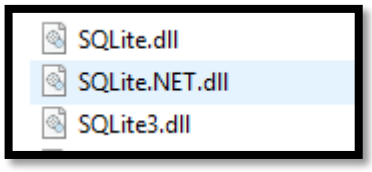


Figure 4.9: Finisar SQLite

Finisar SQLite is an ADO.NET Data Provider for accessing SQLite-Database using NET-Framework. SQLite is a small C library that implements a self-contained, embeddable, zero-configuration SQL database engine. It is an easy way to use SQLite-Database in C#.

d) ADO.Net

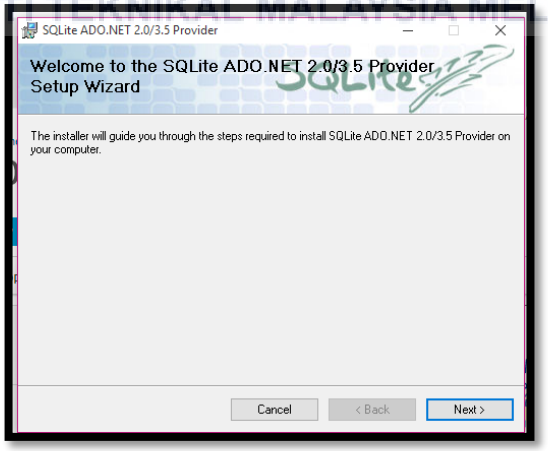


Figure 4.10: ADO.Net

ADO.Net separates data access from data manipulation into discrete components that can be use separately. It includes .Net Framework data providers for connecting to a database, executing commands and retrieve results. ADO.Net classes are found in System.Data.dll. ADO.Net is suggested for accessing data in .Net applications.

e) SQLite Manager

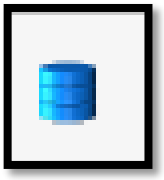


Figure 4.11: SQLite Manager as database server

SQLite Manager is manage any SQLite database in computer. It manage tables, indexes, views and triggers. It provides facility to execute any sql query. It also export tables/views/database in csv/xml/sql format. Import from csv/xml/sql. Furthermore, it is easy to use.

f) Microsoft Visual Studio 2012

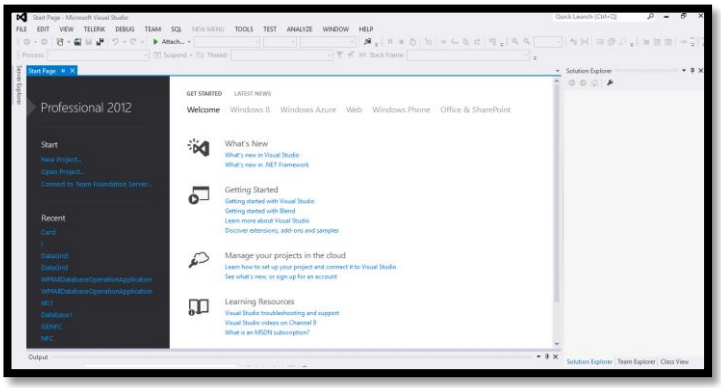


Figure 4.12: Microsoft Visual Studio 2012

Microsoft Visual Studio 2012 is an integrated development envronment (IDE) from Microsot. It uses platforms such as Windows API, Windows Forms, Windows Presentation Foundation and Microsoft Silverlight. It includes code editor. It also support built-in languages includes C#, VB.Net and C++.

Hardware Requirement

a) Desktop pc

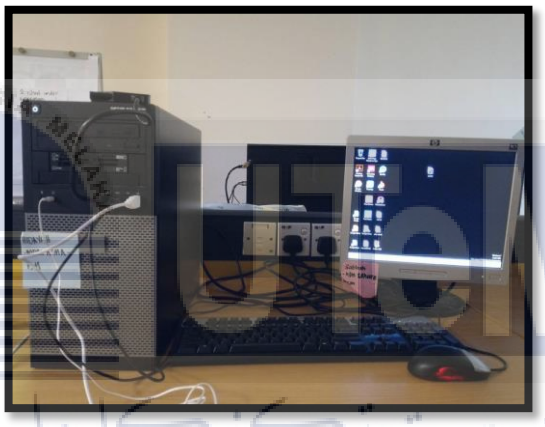


Figure 4.13: Desktop pc

Desktop pc or laptop is used in this project. It is utilize to do all the documentation and all the implementation. All the software are installed in this pc desktop. Table 4.1 below show the properties.

Table 4.1: Desktop pc properties

No.	System Configuration	Specification
1	Operating System	Windows 10 Home
2	Hardware	Processor: Intel(R) Core(TM) i5-3470 CPU @ 3.20 GHz RAM: 4.00 GB Hard disk space: 1TB
3	Software	SQLite ADO.Net SQLite Manager iSe-NFC Ferry application Microsoft Visual Studio 2012

b) ACR NFC reader/writer

Figure 4.14: NFC reader/writer

NFC reader/writer is used to read, write and encode NFC tags. NFC reader/writer will be attach to the pc desktop to connect with the iSe-NFC Ferry application. RFID card will tag to the NFC reader/writer. Table 4.2 show the properties of NFC reader/writer.

Table 4.2: NFC reader/writer properties

No.	System Configuration	Specification
1	Product name	ACS ACR 122
2	Hardware	P/N: ACR 122U-A9 Power:5VDC, 200mA S/N: RR171-281658
3	Manufactured by	Advanced Card Systems Ltd. Hong Kong

c) Identification card

Figure 4.15: Passenger identification card

Passenger identification card acts as input. Identification card credentials will be insert in the iSe-NFC application and stored in the database. Table 4.3 below show the properties of the identification card.

Table 4.3: Identification card properties

No.	System Configuration	Specification
1	Product name	Identification card
2	Hardware	Technologies supported: ISO/IEC 14443-3 (Type A) compatible Memory size: 1kB
3	Manufactured by	Infineon Technologies AG

d) RFID card

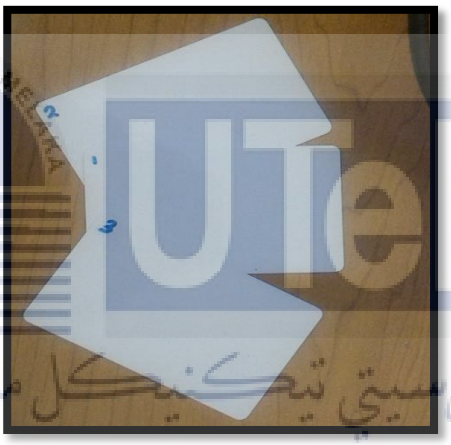


Figure 4.16: RFID card

RFID card is used as a RFID tag for the passenger. It is a passive RFID card. Information such as tracking passenger's identification number will temporarily stored in the RFID card. The information inside RFID card can only detected when RFID card is tapped with NFC reader/writer. Table 4.4 show properties of RFID card.

Table 4.4: RFID card properties

No.	System Configuration	Specification
1	Product name	MFOUL1101DUx
2	Hardware	Type: MIFARE Ultralight EV1 (MFOUL 11) Technologies supported: • ISO/IEC 14443-3 (Type A) compatible Memory size: 48 bytes Maximum transceive length: 253 bytes Default maximum transcieve time-out: 618 ms
3	Manufactured by	NXP Semiconductors

4.4 High-level Design

In this section, it provide an overview of entire system. System architecture, user interface design and conceptual database design will be discussed to show this system will run.

4.4.1 System Architecture

In this section, system architecture is design to define the system in a certain area. Figure 4.16 show the system architecture design. NFC reader/writer is the center of the system. NFC reader/writer will read, write and encode RFID card. NFC reader/writer will attach to the desktop pc or laptop. Staff at the counter will insert passenger's credentials from the identification card through the iSe-NFC Ferry application before the data is stored in the database.

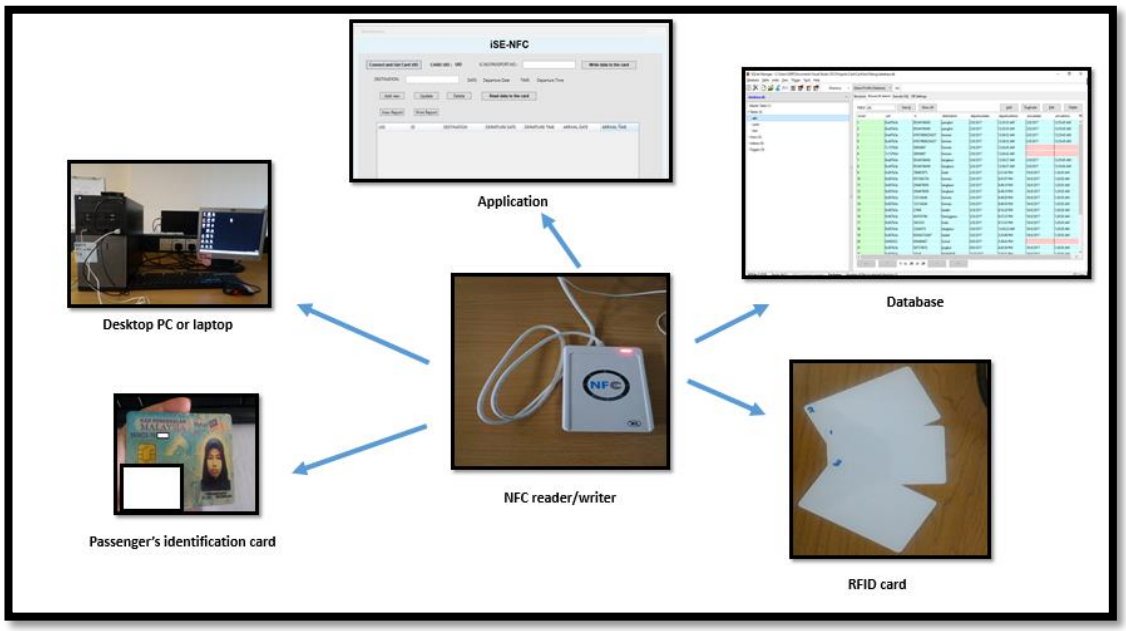


Figure 4.17: Design of system architecture

4.4.2 Interface Design

This section about interface design for this project. The interface allow user to connect with reader, get card UID, read and write card. Also insert, update, delete, search and view data. It display insert, update and retrieve data. The Figures below will show the interface of the system.

- **Passenger data interface**

Passenger data interface is an interface where staff add and modify data of passenger. Figure 4.18 below show passenger data interface.

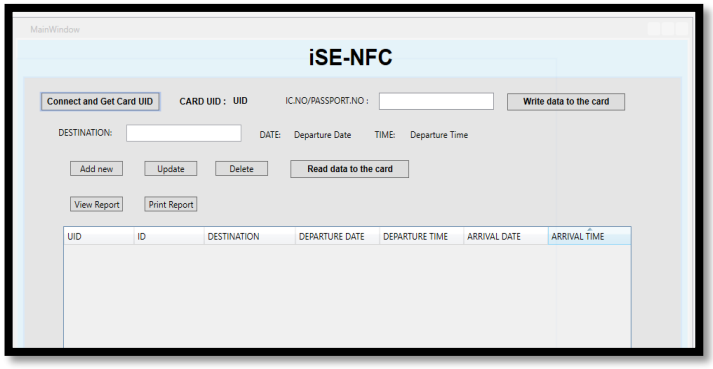
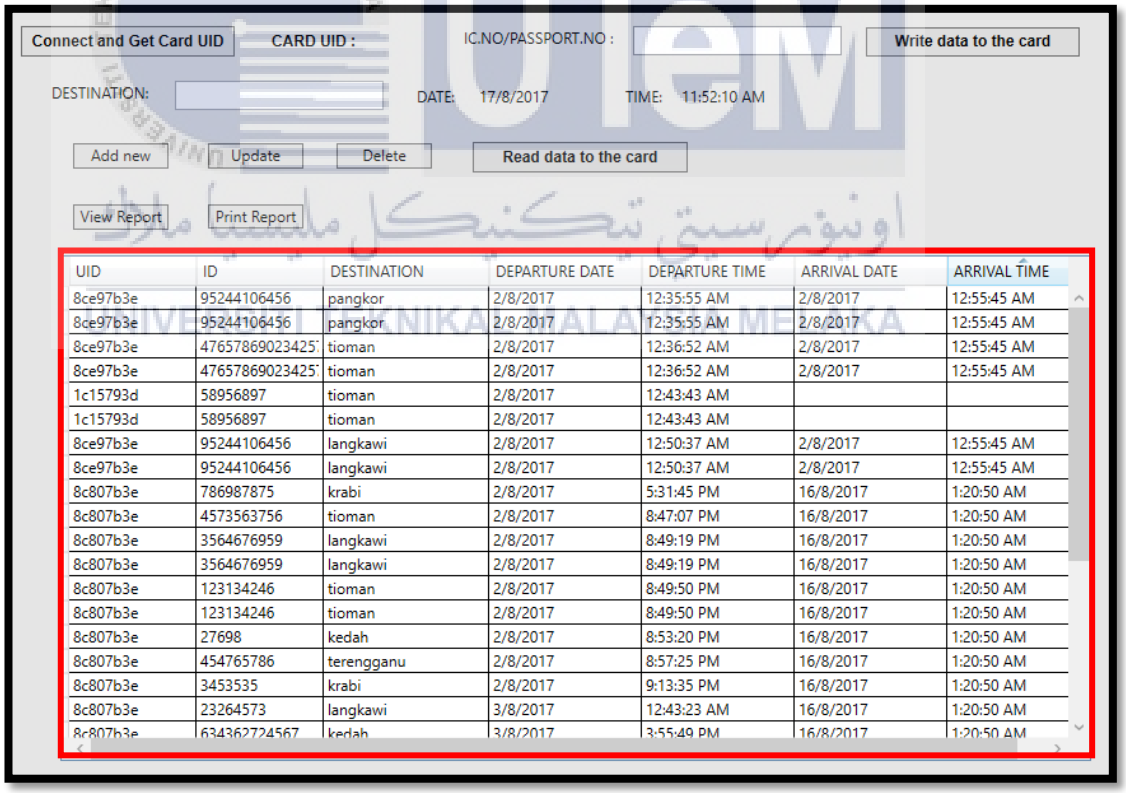


Figure 4.18: Passenger data interface

- **Result interface**

Result interface will display data that has been tag to the RFID card. Figure 4.19 show the data display in DataGrid shown in red box.



UID	ID	DESTINATION	DEPARTURE DATE	DEPARTURE TIME	ARRIVAL DATE	ARRIVAL TIME
8ce97b3e	95244106456	pangkor	2/8/2017	12:35:55 AM	2/8/2017	12:55:45 AM
8ce97b3e	95244106456	pangkor	2/8/2017	12:35:55 AM	2/8/2017	12:55:45 AM
8ce97b3e	47657869023425	tioman	2/8/2017	12:36:52 AM	2/8/2017	12:55:45 AM
8ce97b3e	47657869023425	tioman	2/8/2017	12:36:52 AM	2/8/2017	12:55:45 AM
1c15793d	58956897	tioman	2/8/2017	12:43:43 AM		
1c15793d	58956897	tioman	2/8/2017	12:43:43 AM		
8ce97b3e	95244106456	langkawi	2/8/2017	12:50:37 AM	2/8/2017	12:55:45 AM
8ce97b3e	95244106456	langkawi	2/8/2017	12:50:37 AM	2/8/2017	12:55:45 AM
8c807b3e	786987875	krabi	2/8/2017	5:31:45 PM	16/8/2017	1:20:50 AM
8c807b3e	4573563756	tioman	2/8/2017	8:47:07 PM	16/8/2017	1:20:50 AM
8c807b3e	3564676959	langkawi	2/8/2017	8:49:19 PM	16/8/2017	1:20:50 AM
8c807b3e	3564676959	langkawi	2/8/2017	8:49:19 PM	16/8/2017	1:20:50 AM
8c807b3e	123134246	tioman	2/8/2017	8:49:50 PM	16/8/2017	1:20:50 AM
8c807b3e	123134246	tioman	2/8/2017	8:49:50 PM	16/8/2017	1:20:50 AM
8c807b3e	27698	kedah	2/8/2017	8:53:20 PM	16/8/2017	1:20:50 AM
8c807b3e	454765786	terengganu	2/8/2017	8:57:25 PM	16/8/2017	1:20:50 AM
8c807b3e	3453535	krabi	2/8/2017	9:13:35 PM	16/8/2017	1:20:50 AM
8c807b3e	23264573	langkawi	3/8/2017	12:43:23 AM	16/8/2017	1:20:50 AM
8c807b3e	634362724567	kedah	3/8/2017	3:55:49 PM	16/8/2017	1:20:50 AM

Figure 4.19: Result interface

4.4.3 Conceptual Database Design

In this section display Figure 4.20, it show Entity Relationship Diagram (ERD) that has been develop for project system. Those table in ERD contain attributes and type of data and it normality. There are five table which are Staff, Admin, Passenger, Tracking and Route. Table staff contain staff's information and table for admin contain information of admin. Table for passenger contain information passenger. While table of tracking and route contain information of the tracking id for passenger and destination of the ferry. Table 4.5 - Table 4.11 show the data dictionary of the database.

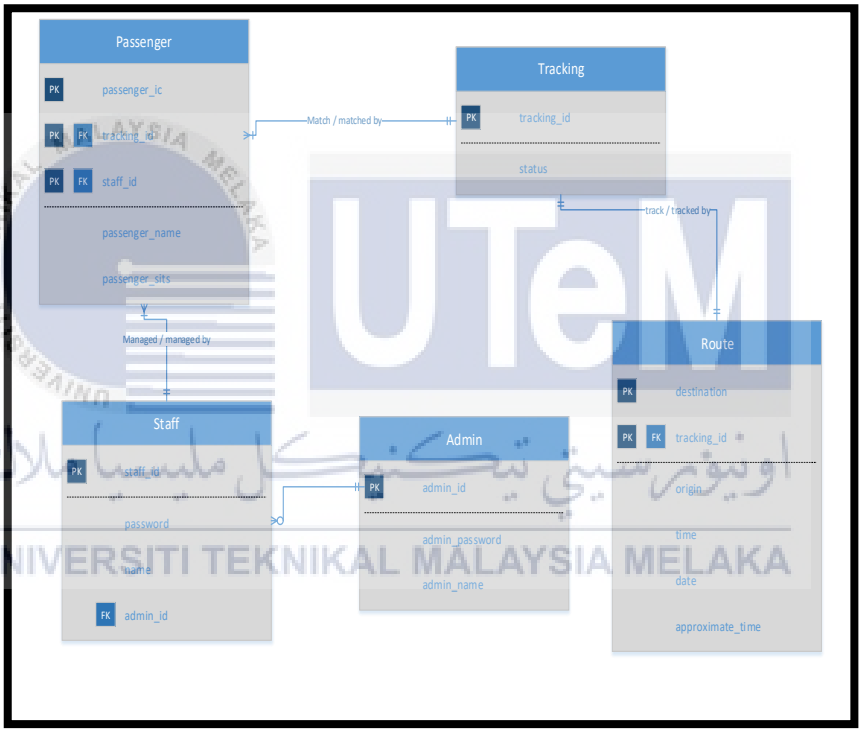


Figure 4.20: ERD of project system

Data Dictionary

Table 4.5: Admin data dictionary

No	Attribute Name	Data Type	Length	Primary Key (PK) / Foreign Key (FK)	Definition and example
1	Admin_id	Varchar	15	PK	Auto increment number
2	Admin_name	Varchar	30		Admin name (admin123)
3	Admin_password	Varchar	10		Password admin will be the security for acces to system (admin123)

Table 4.6: Staff data dictionary

No	Attribute Name	Data Type	Length	Primary Key (PK) / Foreign Key (FK)	Definition and example
1	Staff_id	Varchar	10	PK	Auto increment number
2	Name	Varchar	30		Staff name (staff123)
3	Password	Varchar	10		Password staff will be the security for acces to system (admin123)
4	Phone	Varchar	10		Phone number with 10 digit (019-1234873)

Table 4.7: Passenger data dictionary

No	Attribute Name	Data Type	Length	Primary Key (PK) / Foreign Key (FK)	Definition and example
1	Passenger_ic	Varchar	10	PK	Input identification number (950423-12-8989)
2	Passenger_name	Varchar	30		Passenger name (Dalilah Ahmad)
3	Passenger_sits	Varchar	10		Passenger sits act as data to track and identified passenger (A01)

Table 4.8: Roles data dictionary

No	Attribute Name	Data Type	Length	Primary Key (PK) / Foreign Key (FK)	Definition and example
1	Roles_id	Int	2	PK	Roles id differentiate between admin and staff (A11)
2	Position	Varchar	10		Position of admin or staff (Admin)

Table 4.9: Login data dictionary

No	Attribute Name	Data Type	Length	Primary Key (PK) / Foreign Key (FK)	Definition and example
1	Username	Varchar	10	PK	Username of Admin and staff (admin123)
2	Password	Varchar	10		Password of admin or staff (@dmin123)

Table 4.10: Route data dictionary

No	Attribute Name	Data Type	Length	Primary Key (PK) / Foreign Key (FK)	Definition and example
1	Destination	Varchar	20		Name of second point/terminal
2	Origin	Varchar	20		Name of first point/terminal
3	Approximate_time	Timestamp		PK	Approximate time passenger/ferry arrived
4	Date	Date			
5	Time	Time			

Table 4.11: Tracking data dictionary

No	Attribute Name	Data Type	Length	Primary Key (PK) / Foreign Key (FK)	Definition and example
1	Tracking_id	Varchar	10	PK	Tracking id of passenger (P01)
2	Status	Char	10		Status of passenger (Arrived)

4.5 Project and system Flow Chart

This section explains on the flow chart of the project and the system. Figure 4.21 below show the overall process how the project works.

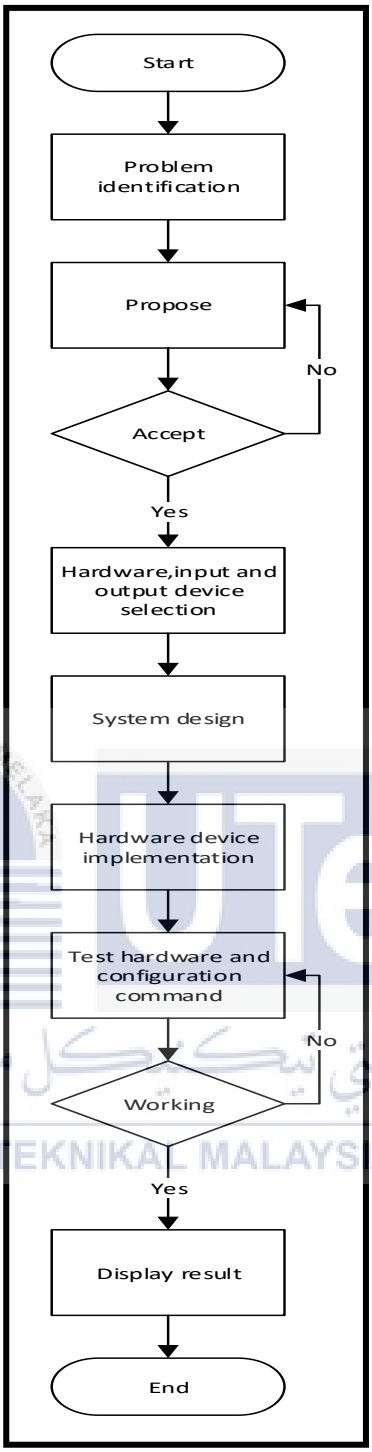


Figure 4.21: Flowchart of project process

4.5.1 System Flowchart

This system have to process for passenger, passenger entry process and passenger exit process. Figure 4.22 show the passenger entry process and Figure 4.23 show the passenger exit process.

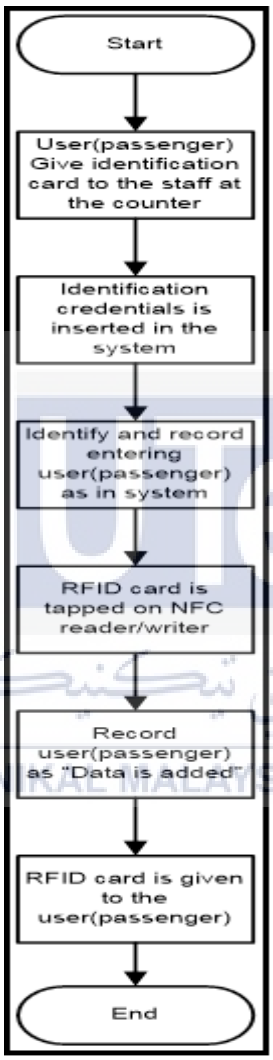


Figure 4.22: Flowchart passenger entry process

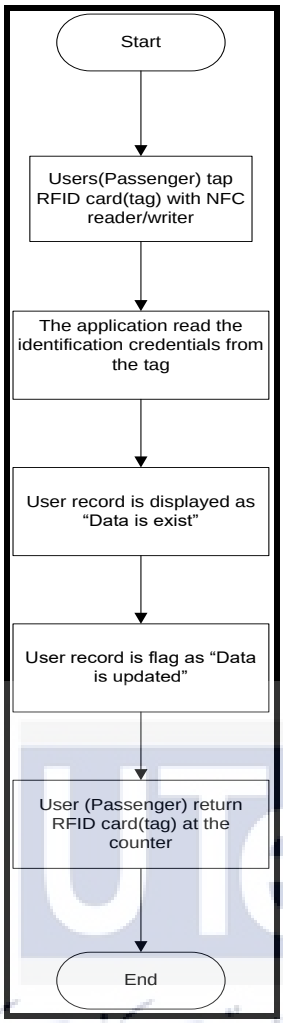


Figure 4.23: Flowchart passenger exit process

4.6 Conclusion

This chapter explains on analysis and design of the project system. It includes analysis part which are problem analysis and requirement analysis in developing the system. At this point, this chapter include software and hardware requirements as a part for implementation and testing. Detail design is show how the system is working and which data will store in the database. This chapter will be refers for system implementation chapter to develop real system.

CHAPTER V

IMPLEMENTATION

5.1 Introduction

Implementation of the project describes in details for this chapter. Configuration management and environment setup will be listed and explained to make the project run smoothly. NFC reader/writer is attached to the desktop pc and tagging process is conducted. Software and hardware installation will be explained as well.

5.2 Project Environment Development Setup

This section describes the overall development environment of the project that covers the following sub-topic:

i. NFC reader/writer Environment Setup

The NFC reader/writer environment setup is to discover on how the NFC reader/writer being setup from the beginning until the project complete. The description about what software has been installed and configured will be explained in more details. Configurations need to be done for tag RFID card.

ii. SQLite and ADO.Net server Database Environment Setup

This section describes about SQLite and ADO.Net server database on setup. The description is about the database that has been installed for this project.

iii. iSe-NFC Ferry Application Environment Setup

This sub-topic is about application setup. It explains on which programming language is used and it explains about software that has been installed to code the programming language.

5.2.1 NFC reader/writer Environment Setup

NFC reader/writer is the main platform of this project. The configuration needs to be done to tag RFID card. This section will describe on hardware and software requirements as below and Table 5.1 show the features of NFC reader/writer.

a) Hardware requirements
   - Desktop pc/laptop
   - RFID card

b) Software requirements
   - Microsoft Visual Studio 2012 application

Table 5.2: System Configuration (NFC reader/writer)

No.	System Configuration	Specification
1	Product name	ACS ACR 122
2	Hardware	P/N: ACR 122U-A9 Power:5VDC, 200mA S/N: RR171-281658
3	Manufactured by	Advanced Card Systems Ltd. Hong Kong

5.2.2 SQLite, ADO.Net and SQLite Manager (Database) Environment Setup

SQLite is an in-process library that implements a self-contained, server less, zero-configuration, transactional SQL database engine. ADO.Net is a data access technology from the Microsoft .NET Framework that provides communication between relational and non-relational systems through a common set of component. The configurations need to be done to setup a database. Table 5.2 below is the hardware requirement:

Table 5.3: System Configuration (Desktop pc)

No.	System Configuration	Specification
1	Operating System	Windows 10 Home
2	Hardware	Processor: Intel(R) Core(TM) i5-7200u CPU @ 3.20 GHz RAM: 4.00 GB Hard disk space: 1TB
3	Software	SQLite ADO.Net SQLite Manager iSe-NFC Ferry application Microsoft Visual Studio 2012

5.2.3 iSe-NFC Ferry Application Environment Setup

This section explains on iSe-NFC Ferry application setup. It is to integrate NFC reader/writer software integrate with the application. Table 5.3 below is the software requirement and programming language used:

Table 5.4: System Configuration (iSe-NFC Ferry application)

No.	System Configuration	Specification
1	Operating System	Windows 10 Home
2	Hardware	Processor: Intel(R) Core(TM) i5-7200u CPU @ 3.20 GHz RAM: 4.00 GB Hard disk space: 1TB
3	Software	Visual Studio 2012
4	Language	C#

5.3 Project Configuration Management

This section explains the overall configuration management for the project which covers sub-topic below:

5.3.1 NFC reader/writer Configuration

NFC reader/writer configuration is to explain step of the setup. The configuration will include as listed below:

5.3.1.1 NFC Reader/Writer application installation.

Below are the steps of installations:

1. Attach NFC reader/writer to the laptop/pc. Figure 5.1 below show NFC reader/writer setup.

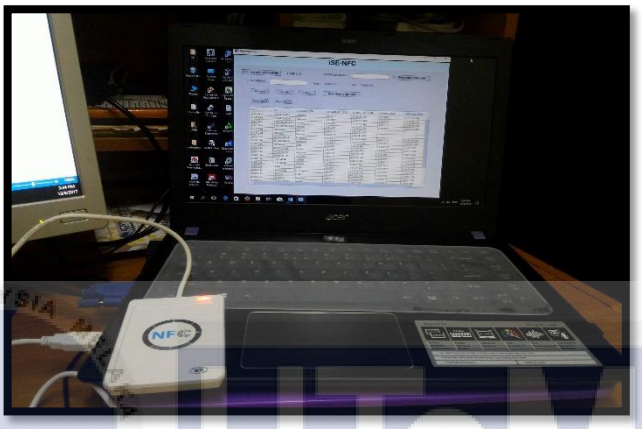


Figure 5.1: NFC reader/writer setup

2. LED on the reader will be RED colour indicates there is no any of RFID tag present. Figure 5.2 show the colour of LED.

Figure 5.2: Red colour LED

3. LED turn to YELLOW colour indicates the present of RFID tag and there will be a sound produce from the laptop/pc. Figure 5.3 show the colour of LED.

Figure 5.3: Yellow colour LED

5.3.2 SQLite, ADO.Net and SQLite Manager (Database) Configuration

This section is about installation of SQLite, ADO.Net and SQLite Manager. This configuration will include as listed below:

5.3.2.1 SQLite installation

Below are the steps of installations:

1. Download SQLITE software from this link: http://adodotnetsqlite.sourceforge.net/downloads/

2. Click at the red link:SQLite.NET.0.21 x86_dll.zip(x86). Figure 5.4 is the SQLite.NET page.

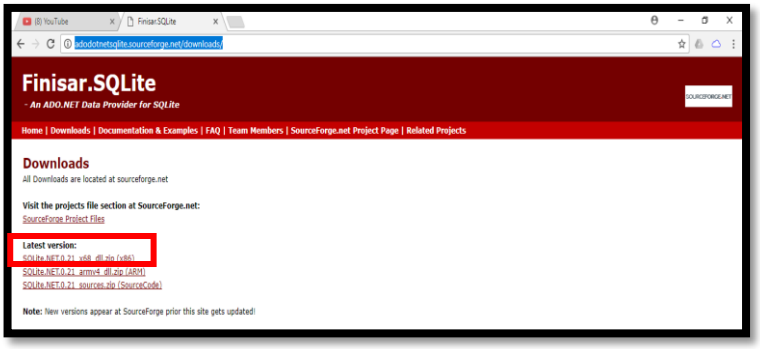


Figure 5.4: SQLite.NET download page

3. After the .zip file downloaded, extract the SQLite3.dll, SQLite.NET.dll and SQLite.dll to the projects file. Figure 5.5 below are the .dll files.

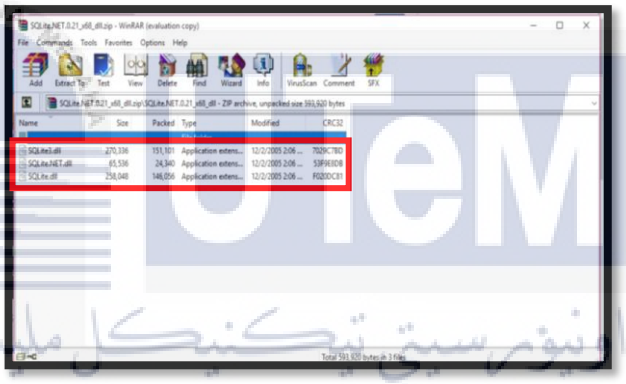


Figure 5.5: SQLite3.dll, SQLite.NET.dll and SQLite.dll files.

4. Put the .dll files in the Projects>Card>bin>Debug directory. Figure 5.6 are .dll files that are extracted.

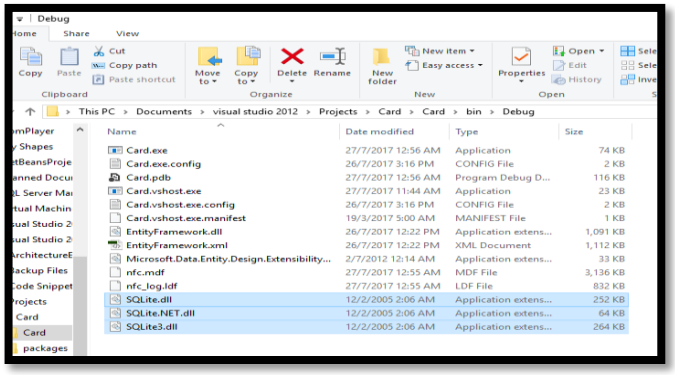


Figure 5.6: SQLite3.dll, SQLite.NET.dll and SQLite.dll files extracted.

5. Open the project and go to Card at solution explore. Click Add Reference. Figure 5.7 is where to add reference.

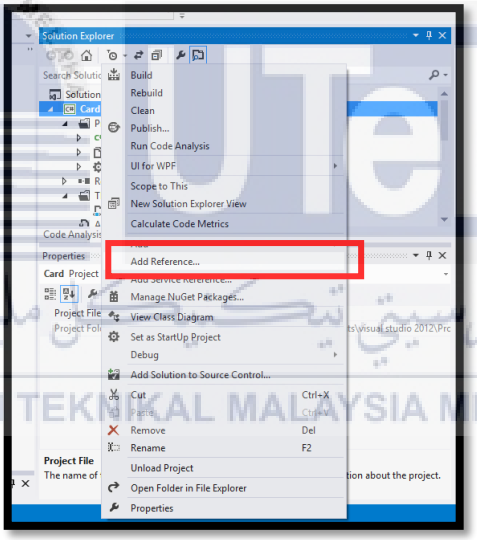


Figure 5.7: Where to Add Reference

6. Click Browse as refer to Figure 5.8.

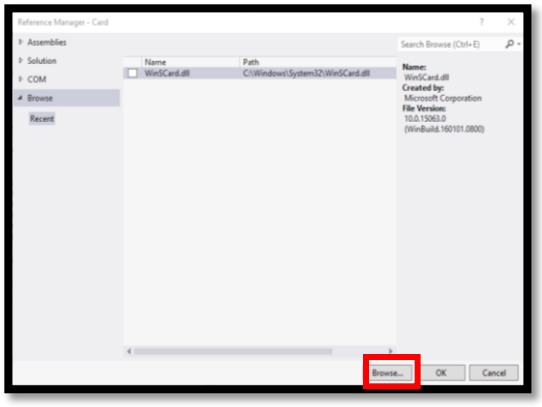


Figure 5.8: Browse to add SQLite reference

7. Choose SQLiteNET.dll and click Add as refer to Figure 5.9.

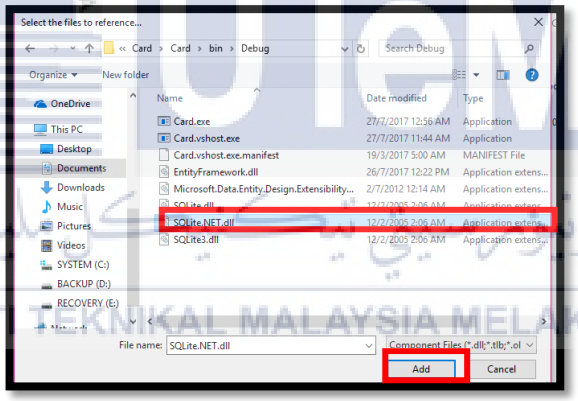


Figure 5.9: Choose SQLite

8. Right click at SQLiteNET.dll and click OK as refer to Figure 5.10.

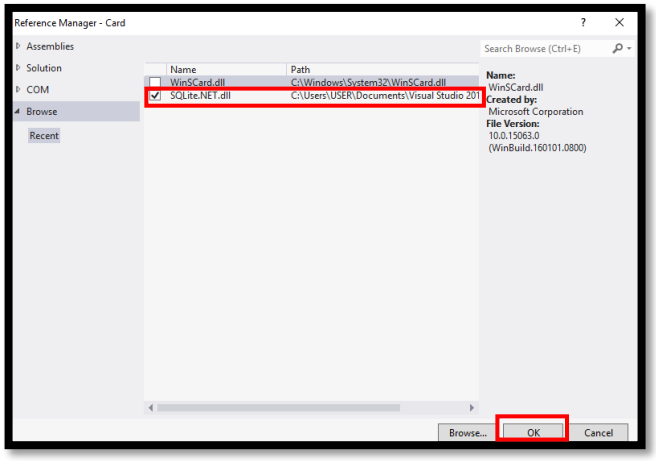


Figure 5.10: Choose SQLiteNET.dll

5.3.2.2 ADO.Net. Installation

Below are the steps of installations:

1. Download ADO.Net software from this link:
   https://sourceforge.net/projects/sqlite-dotnet2/

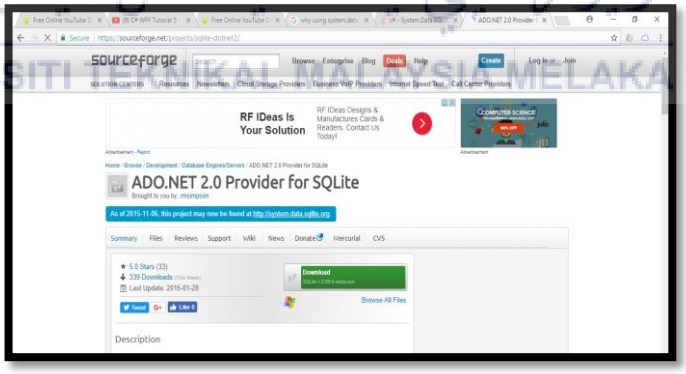


Figure 5.11: ADO.Net software download page

2. After the SQLite-1.0.66.0-set.exe downloaded. Click at the .exe file to start the installation. Click Next on every configuration. Figure 5.12 until Figure 5.14 are the steps.

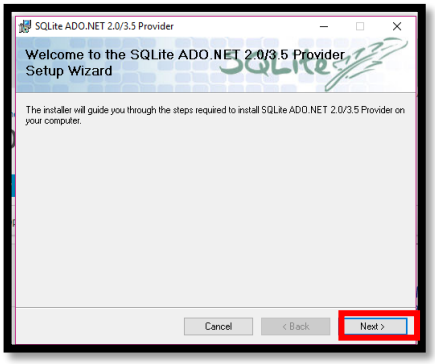


Figure 5.12: ADO.Net installation 1

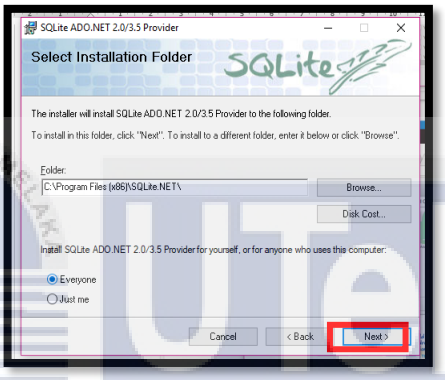


Figure 5.13: ADO.Net installation 2

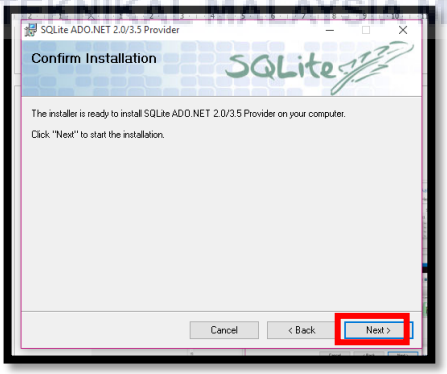


Figure 5.14: ADO.NeT installation 3

3. After installation finished, go to the project and Add Reference. Click at System.Data.SQLite and click OK.

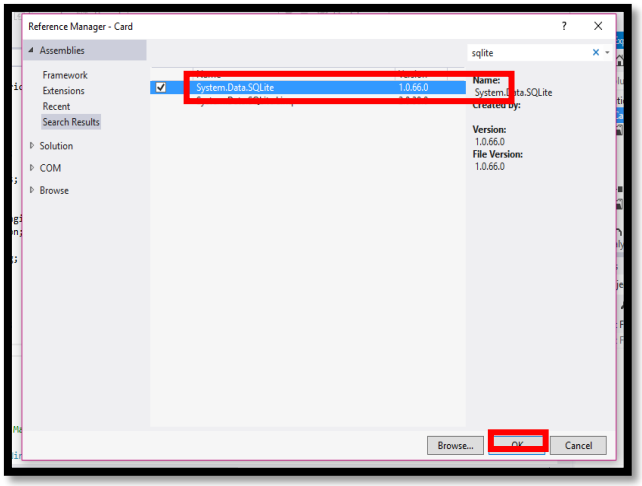


Figure 5.15: System.Data.SQLite

5.3.2.3 SQLite Manager on Firefox

Below are the steps of installations:

1. Download SQLite Manager on Firefox as plugin at this link:

   https://addons.mozilla.org/en-US/firefox/addon/sqlite-manager/

2. Click at Add to Firefox to download.

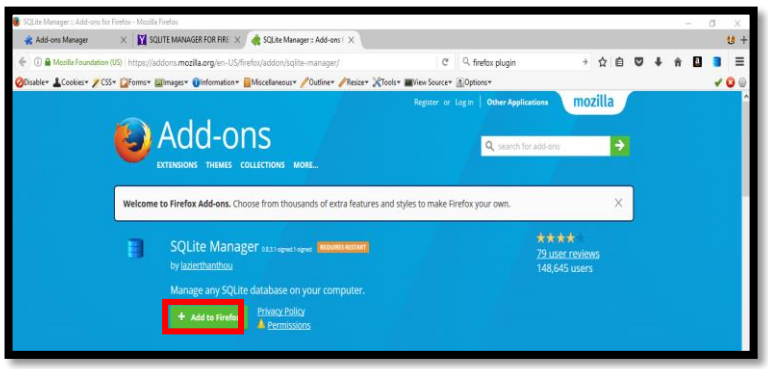


Figure 5.16: SQLite manager download website

3. Go to the Add-ons Manager and enable the SQLite Manager.

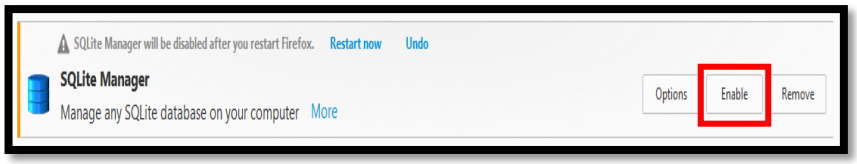


Figure 5.17: Enabling SQLite Manager

4. Open SQLite Manager after installation and start create “database.db” and “nfc”table.

5. Create “database.db” and click OK.

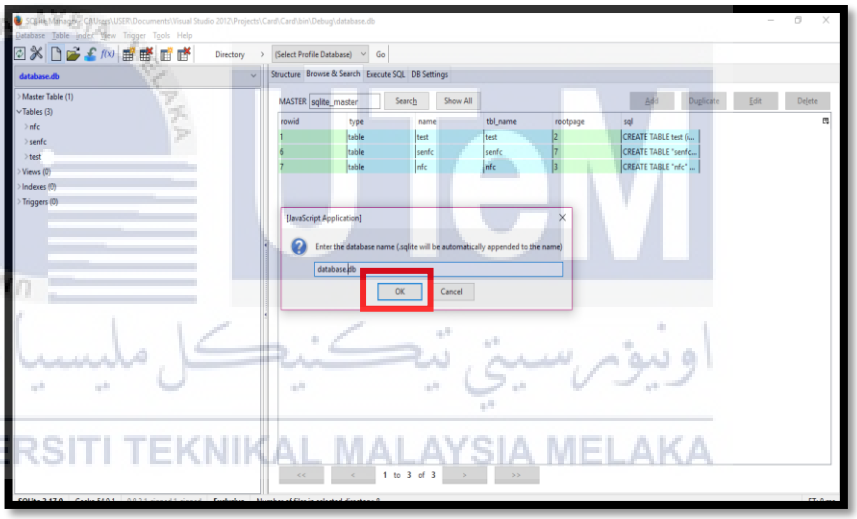


Figure 5.18: Create database.db

6. Create “nfc” table by click at Tables and choose Create Table.

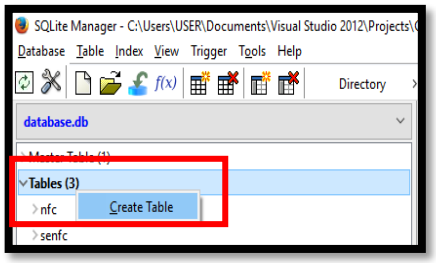


Figure 5.19: Create table

7. Fill in all the database attributes and types. Then, Click OK.

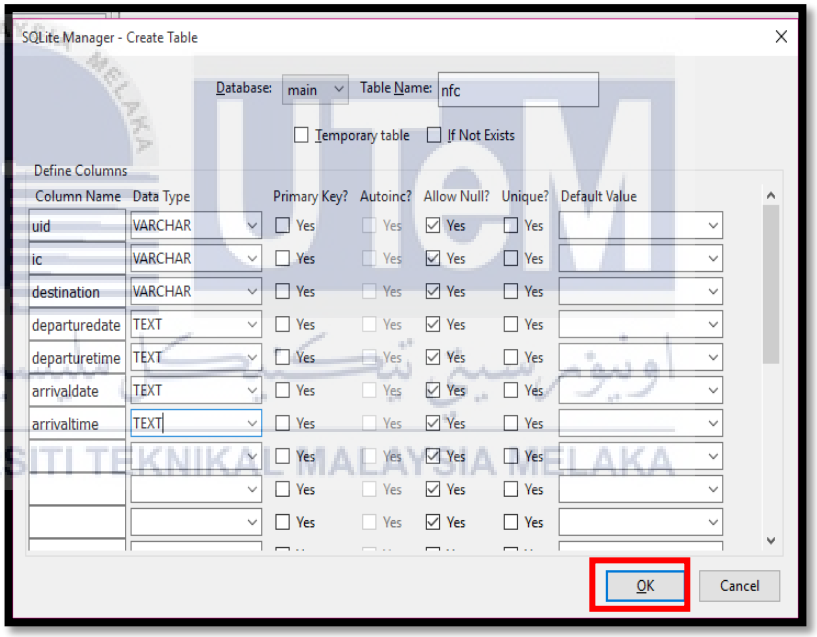


Figure 5.20: "nfc" table

5.3.2.4 Run database command

Below are the steps of installations:

1. Run database command in the Card project at Microsoft Visual Studio 2012 to connect the application with the database.

2. Code below is the code for connection string code:

```
string dbConnectionString = @"Data Source = database.db;Version = 3;";
```

Figure 5.21: Database connection string

3. Code below is the example code to retrieve back data in the database and to connect database with the application:

```
SQLiteConnection sqliteCon = new SQLiteConnection(dbConnectionString);
sqliteCon.Open();
string Query = "select uid,ic,destination,departuredate,departuretime,arrivaldate,arrivaltime from nfc";
SQLiteCommand createCommand = new SQLiteCommand(Query, sqliteCon);
createCommand.ExecuteNonQuery();
```

Figure 5.22: String Query

4. From Figure 5.22, String Query can be varies based on the function.

5.3.3 iSe-NFC Ferry Application Configuration

This section explains on configuration of iSe-NFC Ferry application. This configurations will include as listed below. Below are the steps of installations:

1. Open Microsoft Visual Studio 2012 and click New Project in red box. It is to make new C# language project.

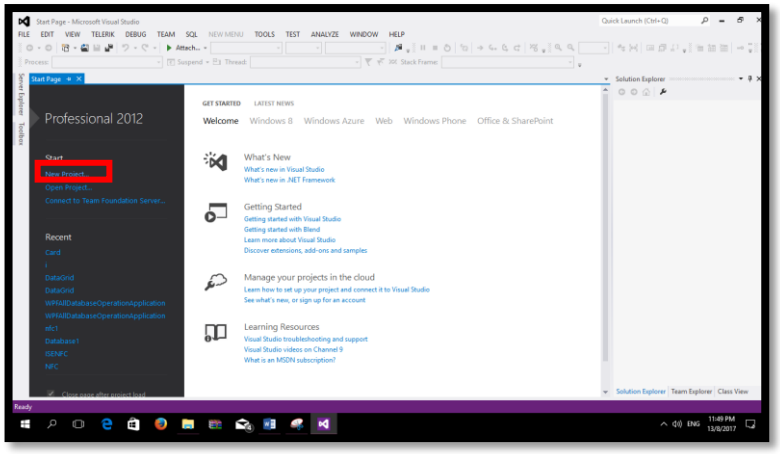


Figure 5.23: New project in Microsoft Visual Studio 2012

2. Choose WPF Application and name it Card. Click OK. WPF Application is a graphical subsystem by Microsoft for rendering user interfaces in Windows-based applications.

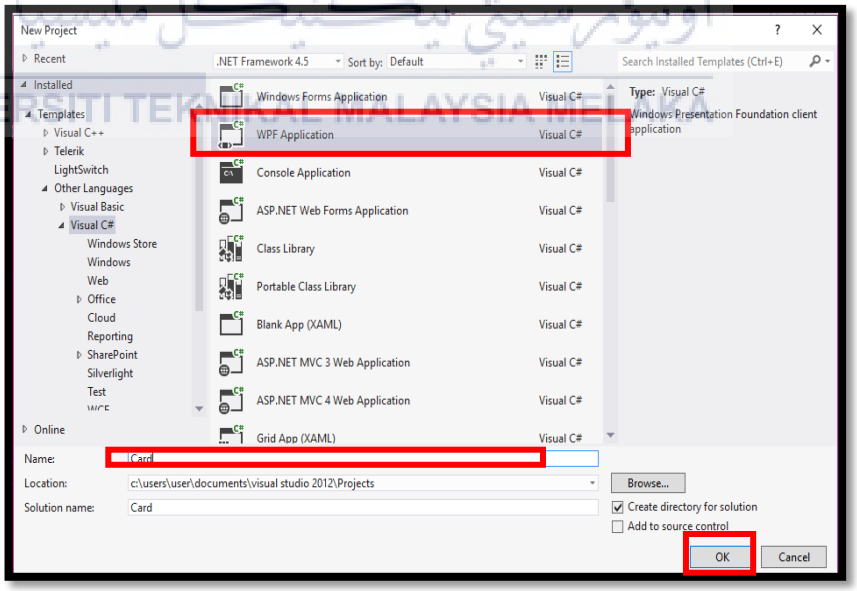


Figure 5.24: Card.cs is created

3. Drag and drop Buttons, Text Blocks, Text Boxes and DataGrid from Toolbox. Arrange he interface.

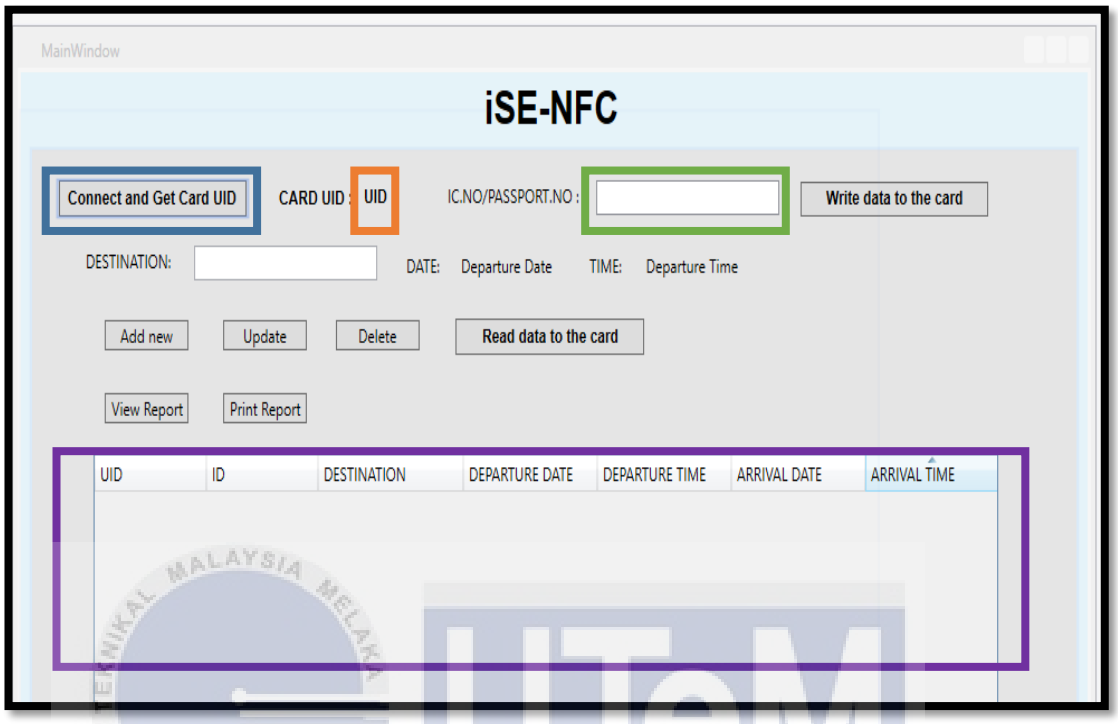


Figure 5.25: Interface of iSe-NFC

From above Figure 5.25, there are colour boxes that indicates the properties of every label. Below are the properties.

	Blue indicates Button. A user can click at the button to perform its functionalities. Connect and Get Card UID, Write data to the card, Read data to the card, Add new, Update, Delete, View report and Print report are Button.
	Orange indicates Text Blocks. A user cannot click at the Text Block to perform its functionalities. The Text Blocks will automatically display information. CARD UID, UID, IC.NO/PASSPORT, DESTINATION, DATE, Departure Date, TIME, and Departure time are Text Blocks.
	Green indicates Text Boxes. A user can enter data in the Text Box to perform its functionalities. IC.NO/PASSPORT Text Box and DESTINATION Text Box tare Text Boxes.

	Purple indicates DataGrid. It display data that is added and updated from the database.

5.3.3.1 Connect and Get Card UID function button.

It is the Connect function to connect to the NFC reader/writer with the application. Get Card UID function is for to scan the UID of every RFID cards or RFID tags. Below are the steps of installations:

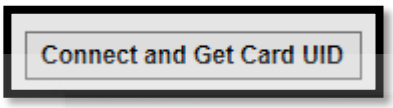


Figure 5.26: Connect and Get Card UID button

1. Click at the button and code the scripting below:

```
//connect and get UID button
private void Button_Click_1(object sender, RoutedEventArgs e)
{

    if (connectCard())
    {
        string cardUID = getcardUID();
        textBlock1.Text = cardUID; //displaying on text block
    }
}
```

Figure 5.27: Command for connect and get card UID

From the Figure 5.27 above, string **cardUID = getcardUID();** and **textBlock1.Text = cardUID;** is the commands to display the card UID on the Text Block. Every card have its own UID. When the RFID tag is tapped on the NFC reader, it can scan the any RFID tag's UID.

5.3.3.2 Write data to the card.

It is the function to write IC.NO/PASSPORT number into the RFID card or RFID tag. Below are the steps of installations:

Figure 5.28: Write data button

1. Click at the button and code the scripting below:

```
//Write UID button
private void Button_Click_3(object sender, RoutedEventArgs e)
{
    if (connectCard())// establish connection to the card: you've declared this from previous post
    {
        submitText(textBox1.Text, "5"); // 5 - is the block we are writing data on the card

        MainWindowClose();
    }
}
```

Figure 5.29: Write data command

From Figure 5.29 above, **submitText(textBox1.Text, "5");** is the command to write data on the card or any RFID tag.

5.3.3.3 Read data to the card.

It is the function to read IC.NO/PASSPORT number that is written to the RFID card or RFID tag. It will scan the RFID tag that is written. Below are the steps of installations:

Figure 5.30: Read data button

1. Click at the button and code the scripting below:

```
//READ CARD UID BUTTON
private void Button_Click_2(object sender, RoutedEventArgs e)
{
    string text = verifyCard("5"); // 5 - is the block we are reading
    textBox1.Text = text.ToString();

    //to display database values in datagrid
    SQLiteConnection sqliteCon = new SQLiteConnection(dbConnectionString);
    //Open connection to database
    try
    {
        sqliteCon.Open();
        string Query = "select uid,ic,destination,departuredate,departuretime,arrivaldate,arrivaltime from nfc where  ic ='" + this.textBox1.Text + "'";
        SQLiteCommand createCommand = new SQLiteCommand(Query, sqliteCon);
        createCommand.ExecuteNonQuery();
        //SQLiteDataReader dr = createCommand.ExecuteReader();
        SQLiteDataAdapter dataAdp = new SQLiteDataAdapter(createCommand);
        DataTable dt = new DataTable("nfc");
        dataAdp.Fill(dt);
        passengerinfo.ItemsSource = dt.DefaultView;
        dataAdp.Update(dt);
        sqliteCon.Close();
        MessageBox.Show("DATA IS EXIST");


    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
```

Figure 5.31: Read data command

From the above Figure 5.31, **string text = verifyCard("5");** and **textBlock1.Text = text.ToString();** are the commands for to read the written data from the card or RFID tag.

5.3.3.4 Add new button.

This function is to add new data into the database. Below are the steps of installations:

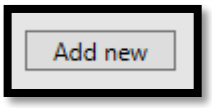


Figure 5.32: Add data button

1. Click at the button and code the scripting below:

```
sqliteCon.Open();
string Query = "insert into nfc (uid,ic,destination,departuredate,departuretime) values ('" + this.textBlock1.Text + "','" + this.textBox1.Text + "','" + th
SQLiteCommand createCommand = new SQLiteCommand(Query, sqliteCon);
createCommand.Parameters.AddWithValue("@uid", textBlock1.Text);
createCommand.Parameters.AddWithValue("@ic", textBox1.Text);
createCommand.Parameters.AddWithValue("@destination", textBox2.Text);
createCommand.Parameters.AddWithValue("@departuredate", DepartureDate.Content);
createCommand.Parameters.AddWithValue("@departuretime", DepartureTime.Text);
createCommand.ExecuteNonQuery();
// SQLiteDataReader dr = createCommand.ExecuteReader();
MessageBox.Show("DATA IS ADDED");
sqliteCon.Close();
displaydata();
ClearData();
```

Figure 5.33: Add/insert data command

From the above Figure 5.33, **string Query = "insert into nfc (uid,ic,destination,department,departuredate,departuretime) values ("+this.textBlock1.Text+","+this.textBox1.Text+");** is the command to insert data into the table "nfc" at the database.db.

5.3.3.5 Update button.

This function is to update any changes to the existed data in the database. Below are the steps of installations:

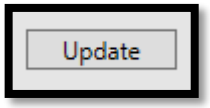


Figure 5.34: Update data button

1. Click at the button and code the scripting below:

```
string Query = "update nfc set uid ='" + this.textBlock1.Text + "',
arrivaldate='" + this.DepartureDate.Content + "', arrivaltime='" +
this.DepartureTime.Text + "' where uid ='" + this.textBlock1.Text + "'
";
```

Figure 5.35: Update data command

From Figure 5.35 above, is the command for update button. It is to update the arrival date and arrival time of passenger that arrive to the ferry terminal.

5.3.3.6 Delete button.

This function is to delete or remove any unwanted data from database. Below are the steps of installations:

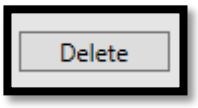


Figure 5.36: Delete data button

1. Click at the button and code the scripting below:

```
string Query = "delete from nfc where ic='" + this.textBox1.Text + "'";
```

Figure 5.37: Delete data command

From Figure 5.37 above, is th command to delete data that is added or update from the database. Command of **"delete from nfc where ic='"** + **this.textBox1.Text** + **"'";** means ic attributes act as the platform delete function to delete the data.

5.3.3.7 Print Report button.

This function is to print data that is displayed from DataGrid. Below are the steps of installations:

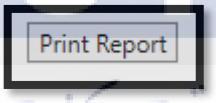


Figure 5.38: Print report button

1. Click at the button and code the scripting below:

```
System.Windows.Controls.PrintDialog Printdlg = new System.Windows.Controls.PrintDialog();
if ((bool)Printdlg.ShowDialog().GetValueOrDefault())
{
  Size pageSize = new Size(Printdlg.PrintableAreaWidth, Printdlg.PrintableAreaHeight);
  //sizing of the element.
  passengerinfo.Measure(pageSize);
   passengerinfo.Arrange(new Rect(5, 5, pageSize.Width, pageSize.Height));
   Printdlg.PrintVisual(passengerinfo, Title);

}
```

Figure 5.39: Print report command

Figure 5.39 above shows, command to print the DataGrid. **Printdlg.PrintVisual(passengerinfo, Title);** is the command to print the DataGrid and save it into a .pdf file. Name for DataGrid is passengerinfo.

5.3.3.8 View Report button.

This function is to refresh data from database to DataGrid. Below are the steps of installations:

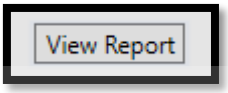


Figure 5.40: View report button

1. Click at the button and code the scripting below:

```
string Query = "select uid,ic,destination,departuredate,departuretime,arrivaldate,arrivaltime from nfc";
```


Figure 5.41: View report command

From the Figure 5.41 above, it is the command to select all the that is added or updated from the database to be display in the DataGrid.

5.3.3.9 Automatic current date and time.

This function is to display automatic current date and time that is tally with the time and date of laptop/pc. Below are the steps of installations:

Figure 5.42: Current data and time

2. Code for current date and time:

3. Code this at public partial class MainWindow:Window:

```
DispatcherTimer t;
DateTime start;
```

Figure 5.43: Public partial class

4. Code at public MainWindow():

```
public MainWindow()
{
    this.InitializeComponent();
    SelectDevice();
    establishContext();
    t = new DispatcherTimer(new TimeSpan(0, 0, 0, 0, 50), DispatcherPriority.Background
    t_Tick, Dispatcher.CurrentDispatcher); t.IsEnabled = true;
    start = DateTime.Now;
    displaydata();
    ClearData();


}
```


Figure 5.44: Initialization command in MainWindow();

5. Code for timerTick:

```
// to set DATE AND TIME (TIMESTAMP)
private void t_Tick(object sender, EventArgs e)
{
    DepartureDate.Content = DateTime.Now.ToShortDateString();
    DepartureTime.Text = DateTime.Now.ToLongTimeString();
    t.Start();
}
```

Figure 5.45: Current date and time command

From Figure 5.45 above, **DepartureDate.Content =** DateTime**.Now.ToShortDateString();** is the command for current date. **DepartureTime.Text =** DateTime**.Now.ToLongTimeString();** is the command is the command for current time. **t.Start();** is the command for the time and date to start counting.

5.3.3.10 Display data to DataGrid.

This function is used to display data from database. Below are the steps of installations:

UID	ID	DESTINATION	DEPARTURE DATE	DEPARTURE TIME	ARRIVAL DATE	ARRIVAL TIME
1c15793d	58956897	tioman	2/8/2017	12:43:43 AM		
8ce97b3e	95244106456	langkawi	2/8/2017	12:50:37 AM	2/8/2017	12:55:45 AM
8ce97b3e	95244106456	langkawi	2/8/2017	12:50:37 AM	2/8/2017	12:55:45 AM
8c807b3e	786987875	krabi	2/8/2017	5:31:45 PM	13/8/2017	3:10:57 PM
8c807b3e	4573563756	tioman	2/8/2017	8:47:07 PM	13/8/2017	3:10:57 PM
8c807b3e	3564676959	langkawi	2/8/2017	8:49:19 PM	13/8/2017	3:10:57 PM
8c807b3e	3564676959	langkawi	2/8/2017	8:49:19 PM	13/8/2017	3:10:57 PM
8c807b3e	123134246	tioman	2/8/2017	8:49:50 PM	13/8/2017	3:10:57 PM
8c807b3e	123134246	tioman	2/8/2017	8:49:50 PM	13/8/2017	3:10:57 PM
8c807b3e	27698	kedah	2/8/2017	8:53:20 PM	13/8/2017	3:10:57 PM
8c807b3e	454765786	terengganu	2/8/2017	8:57:25 PM	13/8/2017	3:10:57 PM
8c807b3e	3453535	krabi	2/8/2017	9:13:35 PM	13/8/2017	3:10:57 PM
8c807b3e	23264573	langkawi	3/8/2017	12:43:23 AM	13/8/2017	3:10:57 PM
8c807b3e	634362724567	kedah	3/8/2017	3:55:49 PM	13/8/2017	3:10:57 PM
04560522	456486467	lumut	9/8/2017	3:58:42 PM		
8c807b3e	547574874	pngkor	9/8/2017	4:42:56 PM	13/8/2017	3:10:57 PM
8c807b3e	24245	PANGKOR	13/8/2017	3:03:51 PM	13/8/2017	3:10:57 PM
8c807b3e	2342642	TIOMAN	13/8/2017	3:10:48 PM	13/8/2017	3:10:57 PM

Figure 5.46: Data in DataGrid

1. In XAML, it need to be code to name the header and binding the data from the database. Below is the code:

```
<DataGridTextColumn Header="UID"  Binding="{Binding uid}"  Width="100" ClipboardContentBinding="{x:Null}
<DataGridTextColumn Header="ID" Binding="{Binding ic}" Width="100" ClipboardContentBinding="{x:Null}" Is
<DataGridTextColumn Header="DESTINATION" Binding="{Binding destination}" Width="130" ClipboardContentBin
<DataGridTextColumn Header="DEPARTURE DATE"  Binding="{Binding departuredate, StringFormat=dd/mm/yyyy}"
<DataGridTextColumn  Header="DEPARTURE TIME" Binding="{Binding departuretime, StringFormat=HH:MM:SS}"  W
<DataGridTextColumn  Header="ARRIVAL DATE" Binding="{Binding arrivaldate, StringFormat=dd/mm/yyyy}" Widt
<DataGridTextColumn Header="ARRIVAL TIME" Binding="{Binding arrivaltime, StringFormat=HH:MM:SS}" Width="
```

Figure 5.47: XAML code for DataGrid

From Figure 5.47 above, **Header** =”**UID**” is the command for header name in the DataGrid. **Binding** = “{**Binding uid**}” is the command to bind the DataGrid with the attributes from the database.

2. Choose DataGrid at Toolbox and code:

```
public MainWindow()
{
    this.InitializeComponent();
    SelectDevice();
    establishContext();
    t = new DispatcherTimer(new TimeSpan(0, 0, 0, 0, 50), DispatcherPriority.Background,
    t_Tick, Dispatcher.CurrentDispatcher); t.IsEnabled = true;
    start = DateTime.Now;
    displaydata();
    ClearData();


}
```

Figure 5.48: displaydata(); initialization command

From Figure 5.48 above, **displaydata();** is the command to initialize the private void for **displaydata();**.

3. Command in the red box is the command to display data from the database to be display in DataGrid.

```
//Display data on datagrid
private void displaydata()
{
    SQLiteConnection sqliteCon = new SQLiteConnection(dbConnectionString);
    sqliteCon.Open();
    string Query = "select uid,ic,destination,departuredate,departuretime,arrivaldate,arrivaltime from nfc";
    SQLiteCommand createCommand = new SQLiteCommand(Query, sqliteCon);
    createCommand.ExecuteNonQuery();
    //SQLiteDataReader dr = createCommand.ExecuteReader();
    SQLiteDataAdapter dataAdp = new SQLiteDataAdapter(createCommand);
    DataTable dt = new DataTable("nfc");
    dataAdp.Fill(dt);
    passengerinfo.ItemsSource = dt.DefaultView;
    dataAdp.Update(dt);
    sqliteCon.Close();
}
```

Figure 5.49: Display in DataGrid in command

From Figure 5.49 above, the command in the red box indicates the data from database table which is “nfc” to be display into the DataGrid.

4. In every button code displaydata(); to display data added, update or delete from the database to the DataGrid. Figure 5.50 below is the example code for Add New button:

```
// add button
private void Button_Click_4(object sender, RoutedEventArgs e)
{
    SQLiteConnection sqliteCon = new SQLiteConnection(dbConnectionString);
    //Open connection to database
    try
    {
        sqliteCon.Open();
        string Query = "insert into nfc (uid,ic,destination,departuredate,departuretime) values ('" + this.text
        SQLiteCommand createCommand = new SQLiteCommand(Query, sqliteCon);
        createCommand.Parameters.AddWithValue("@uid", textBlock1.Text);
        createCommand.Parameters.AddWithValue("@ic", textBox1.Text);
        createCommand.Parameters.AddWithValue("@destination", textBox2.Text);
        createCommand.Parameters.AddWithValue("@departuredate", DepartureDate.Content);
        createCommand.Parameters.AddWithValue("@departuretime", DepartureTime.Text);
        createCommand.ExecuteNonQuery();
        // SQLiteDataReader dr = createCommand.ExecuteReader();
        MessageBox.Show("DATA IS ADDED");
        sqliteCon.Close();
        displaydata();
        ClearData();

    }
    catch (Exception ex)
```

Figure 5.50: Example code for Add new button

5.3.3.11 ClearData() function.

It is to empty the Text Blocks and Text Boxes after data is added, update or delete. Below are the steps of installations:

1. Below is the code for ClearData(); :

```
public MainWindow()
{
    this.InitializeComponent();
    SelectDevice();
    establishContext();
    t = new DispatcherTimer(new TimeSpan(0, 0, 0, 0, 50), DispatcherPriority.Background,
    t_Tick, Dispatcher.CurrentDispatcher); t.IsEnabled = true;
    start = DateTime.Now;
    displaydata();
    ClearData();

}
```

Figure 5.51: ClearData(); initialization command

```
//CLEAR DATA
private void ClearData()
{
    textBlock1.Text = "";
    textBox1.Text = "";
    textBox2.Text ="";
    DepartureDate.Content = "";
    DepartureTime.Text = "";


}
```

Figure 5.52: Command for clear data

Figure 5.52 above, shows the command for every Text Boxes and Text Blocks will empty after added function, update function, delete function, and read to the card function is applied.

2. In every added function, update function, delete function, and read to the card function button need to code ClearData(); . Figure 5.53 below is the example of code for delete function:

```
//delete button
private void Button_Click_5(object sender, RoutedEventArgs e)
{
    SQLiteConnection sqliteCon = new SQLiteConnection(dbConnectionString);
    //Open connection to database
    try
    {
        sqliteCon.Open();
        string Query = "delete from nfc where ic='" + this.textBox1.Text + "'";
        SQLiteCommand createCommand = new SQLiteCommand(Query, sqliteCon);
        createCommand.Parameters.AddWithValue("@ic", textBox1.Text);
        createCommand.ExecuteNonQuery();
        // SQLiteDataReader dr = createCommand.ExecuteReader();
        MessageBox.Show("DATA IS DELETED");
        sqliteCon.Close();
        displaydata();
        ClearData();

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }

}
```

Figure 5.53: Example code

5.4 Summary of project directory.

This section describe the project directory of the iSe-NFC Ferry application. The file and folder used in this project will show in Figure 5.56. In / folder there are Card.sln and Card.v11.suo. The folder of this project is named Card. Card.sln is the C# scripting code where a developer open it to edit the functionalities of the code in Microsoft Visual Studio 2012. In Card folder there are C# project code where all the .xaml and .cs files are stored. The main WPF application, MainWindows.xaml.cs and Card.cs are stored in Card folder. In Card/bin/debug are C# project.exe file which is Card.exe and database.db file are stored. These are the main files for the project.

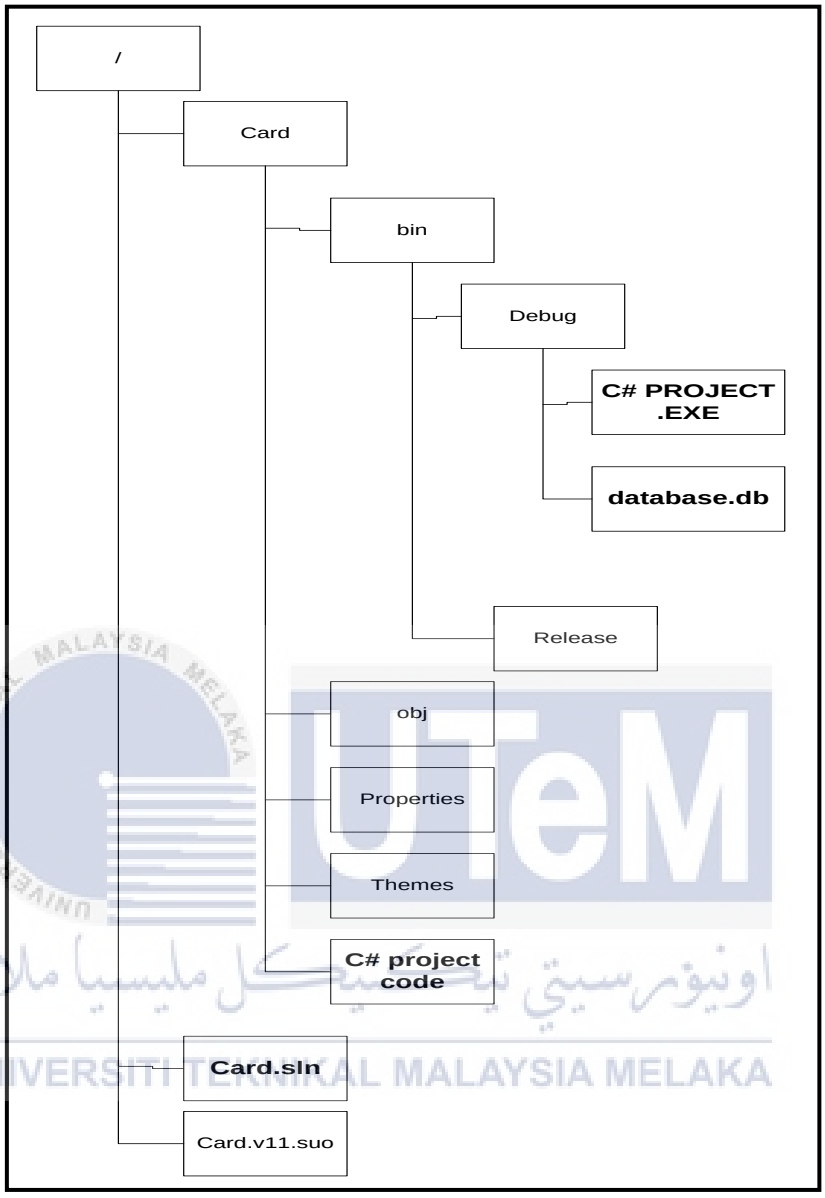


Figure 5.4: Summary of project directory

5.5 Implementation Status

This section explain about the status of development for each component or configuration.

Table 5.5: Implementation status

No	Component	Description	Duration of Complete
1	NFC reader/writer Configuration	• Process to integrate between NFC reader/writer with laptop/pc	1 minute
2	SQLite, ADO.Net and SQLite Manager Configuration	• Process to integrate between SQLite database with iSe-NFC Ferry application and make database connection to the application. • Downloading and installation process is showed.	4 days
3	iSe-NFC Ferry Application Configuration	• Process to integrate between iSe-NFC Ferry application with NFC reader/writer and SQLite database.	2 weeks

5.6 Conclusion

The implementation phase explains on how the project works. The environment setup of the project and it explains all the process involved. All the collected information in this phase will be test on the next process which are testing and analysis.

CHAPTER VI

TESTING

6.1 Introduction

This chapter explains on the testing of the project. The process of testing is an efficient way to find out the effectiveness on the NFC reader as the main component to control the access of ferry passenger at the entrance. The testing has been conducted multiple times to ensure the NFC reader is detected and functioning. The process of the system is a passenger have to give their identification card to the person who is in charge at the counter by fill in the identification number and the name in the form. Then, the person at the counter will tap RFID tag at the NFC reader to validate the passenger information. Based on the previous phase, this chapter will include test plan, test strategy, test design, result and analysis.

6.2 Test Plan

This part explains on the basis of the testing for the system. It includes testing activities and scope of the system.

6.2.1 Test Organization

Developer will test the system in this phase. It shows how the system works from the beginning until the end of the phase. This include the demonstration of the full system. It will also test the functionalities of every function in the interface.

6.2.2 Test Environment

The interface for user details is configured in this design phase. Thus, users able to view and use the functions provided at the interface via screen of the laptop or pc. The purpose of the test environment is to provide a structure for the testing activities. In this task, the test environment requirements are established and reviewed before implementation.

6.2.3 Test Schedule

This phase describes how the testing are carries out by the developer in a period of time. During this process, errors and problems in system will be fix in implementation phase. Errors and problems are solved after it fixed. This process is continuous until the system is successfully built.

6.3 Test Strategy

White-box testing is used for testing strategy. The white-box testing is a testing techniques that examines the program structure and devices test data from the program code. In general testing phase involves various functionality testing which cover tagging system and track ferry passenger where it give access control at the ferry terminal. Card.cs solution file is the main C# application for the system. The developer runs the file to execute the system interface at Microsoft Visual Studio 2012. A RFID tag is put on the NFC reader and make sure the LED light of the reader is yellow and there is sound produced from the reader. Then, developer write and read the RFID tag to perform the tagging and tracking process. Figure 6.1 and 6.2 show the both inflow and outflow of project flow.

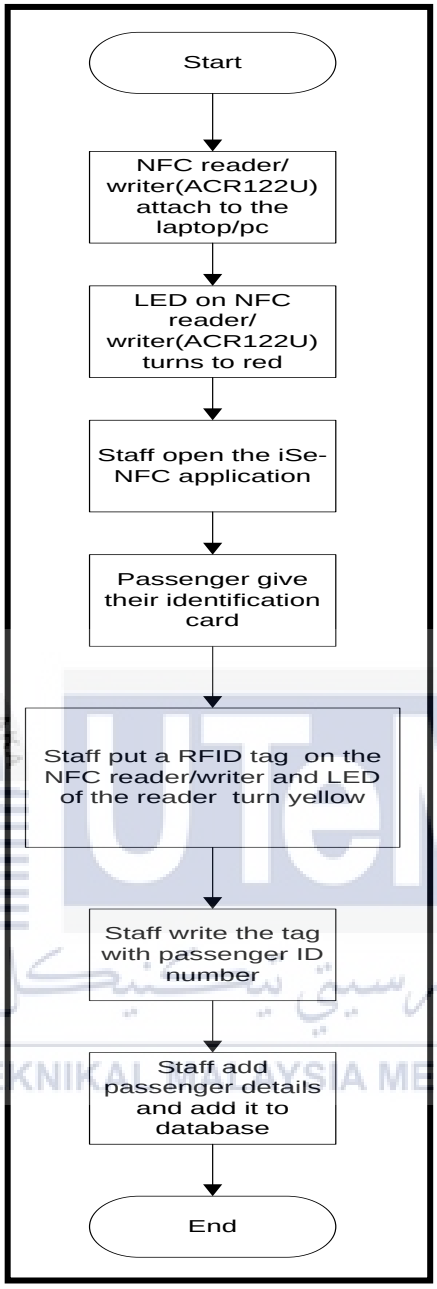


Figure 6.1 Inflow project flow

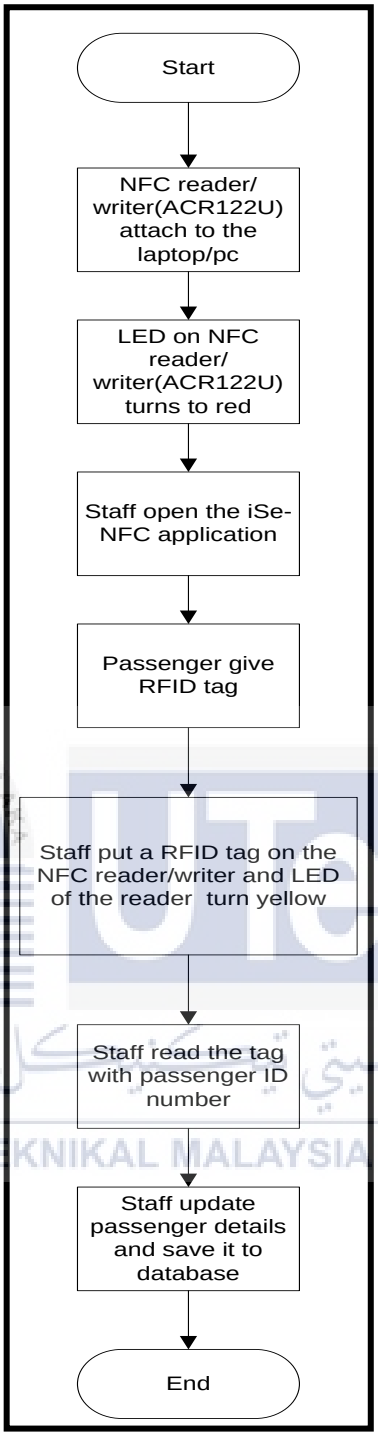


Figure 6.2 Outflow Project flow

6.4 Test Design

Test design is where developer designs the testing cases for the system. Each test case specifies the step-by-step procedures in order to validate the system.

6.4.1 Test Description

Test cases are identified with the expected result for each module. This test includes separate parts of the system such as NFC reader/writer, the interface, database and overall system. All test description are shown in the following tables.

Table 6.1: NFC reader/writer (ACR122U)

Test	NFC reader/writer (ACR122U) test
Test purpose	To test integration between NFC reader/writer with laptop/pc.
Test environment	NFC reader/writer must be attach to the laptop/pc.
Test setup	Attach NFC reader/writer to the laptop/pc.
Expected result	i. The LED turn red because without RFID card on the reader. ii. The LED turn yellow with present of RFID tag. The output can be refer to section 6.5.1.

Table 6.2: SQLite database

Test	SQLite database test
Test purpose	To test integration between SQLite database with iSe-NFC Ferry application.
Test environment	SQLite must be installing. Installation procedure as stated in section 5.3.2.1 until section 5.3.2.4.
Test setup	<u>Finisar.SQLite software</u> i. Download Finisar.SQLite software at Sourceforge. ii. Extract the zip file into the C:\Users\USER\Documents\Visual Studio 2012\Projects\Card\Card\bin\Debug. iii. Open Card project at Microsoft Visual Studio 2012. iv. Add reference SQLiteNET.dll on server explorer section. <u>ADO.NET 2.0 Provider for SQLite</u> i. Download ADO.NET 2.0 Provider for SQLite software at Sourceforge. ii. Install the software. iii. Open Card project at Microsoft Visual Studio 2012. iv. Add reference System.Data.SQLite on server explorer section. <u>SQLite Manager on Firefox</u> i. Download SQLite Manager on Firefox as plugin. ii. Install the SQLite Manager. iii. Open SQLite Manager after installation and start create “database.db” and “nfc”table. <u>Run database command</u> i. Run database command in the Card project at Microsoft Visual Studio 2012. The command are shown in section 5.3.2.4.
Expected output	There is data added and updated into the database. Section 6.5.2 is the output.

Table 6.3: iSe-NFC application

Test	iSe-NFC Ferry application test
Test purpose	To test integration between iSe-NFC Ferry application with NFC reader/writer and SQLite database.
Test environment	iSe-NFC Ferry application is scripted and executed in Card.cs file at Microsoft Visual Studio 2012. The scripting process as stated in section 5.3.3.1 until 5.3.3.11.
Test setup	i. Open Microsoft Visual Studio 2012 and click New Project. ii. Choose WPF Application and name it Card. iii. Drag and drop Buttons, Text Blocks, Text Boxes and DataGrid from Toolbox. Arrange the interface. iv. Run the commands for every function button.
Expected output	The data will be added and updated into the database. It also will display in the DataGrid. Section 6.5.3 is the output.

6.5 Result and Analysis

6.5.1 Test NFC reader/writer

The NFC reader/Writer test is related to Table 6.1, the NFC reader/writer need to been tested separately in order to ensure the correct functionality. To start the test, NFC reader/writer must be attach to the pc/laptop as referred to section 5.3.1.

Figure 6.3: Red LED colour

Based on Figure 6.3, when reader is attached to the pc/laptop. LED on the reader will be RED colour indicates there is no any of RFID tag present.

Figure 6.4: Yellow colour LED

Based on the Figure 6.4, when the reader is attached to the pc/laptop. LED turn to YELLOW colour indicates the present of RFID tag and there will be a sound produce from the laptop/pc.

6.5.2 Test SQLite, ADO.Net and SQLite Manager (Database)

The Table 6.2 show the description about the database testing. In order to test the database, “nfc” table is created in database.db file. Full installation can be refer to section 5.3.2. Figure 5.21 and Figure 5.22 are the commands to establish a connection between database and the application. Figure 6.5 below, is the output testing for the database. All data are displayed in the database. Thus, the testing is success.

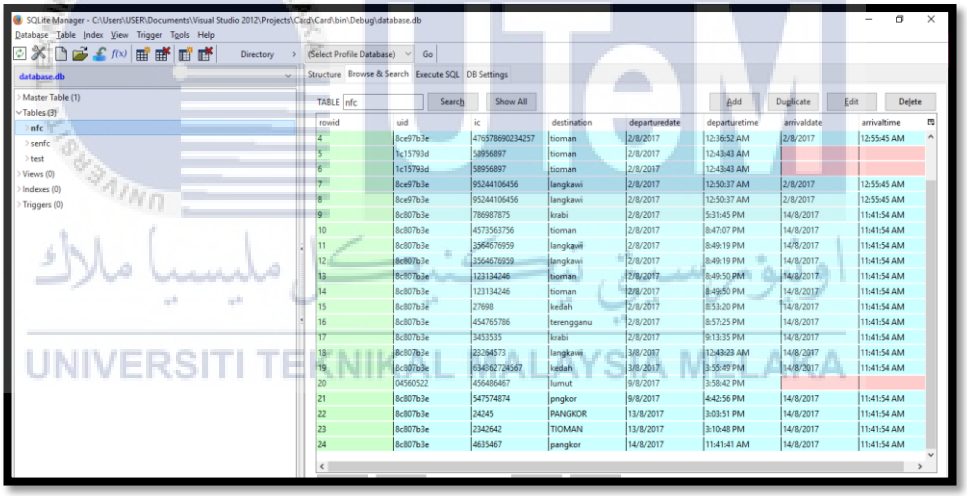


Figure 6.5: Database testing is success

6.5.3 Test iSe-NFC Ferry Application

All the test are related to Table 6.2, but these test is done by its different functionality. Section 6.5.3.1 until section 6.5.3.11 are the test for different functionalities in the interface.

6.5.3 Connect and Get Card UID function button.

The Connect and Get Card UID function button test is related to Table 6.3 which is the function to connect to the NFC reader/writer with the application. Get Card UID function is for to scan the UID of every RFID cards or RFID tags. The coding commands are referred to section 5.3.3.1. Figure 6.6 below is the output from the scripting. In the red box show the card UID that is displayed when a user click at the Connect and Get Card UID button.

Figure 6.6: Card UID output

6.5.3.2 Write data to the card.

The Write data to the card function button test is related to Table 6.3 which is the function to write IC.NO/PASSPORT number into the RFID card or RFID tag. The coding commands are referred to section 5.3.3.2. Figure 6.7 below is the output where the data is written to the card is success.

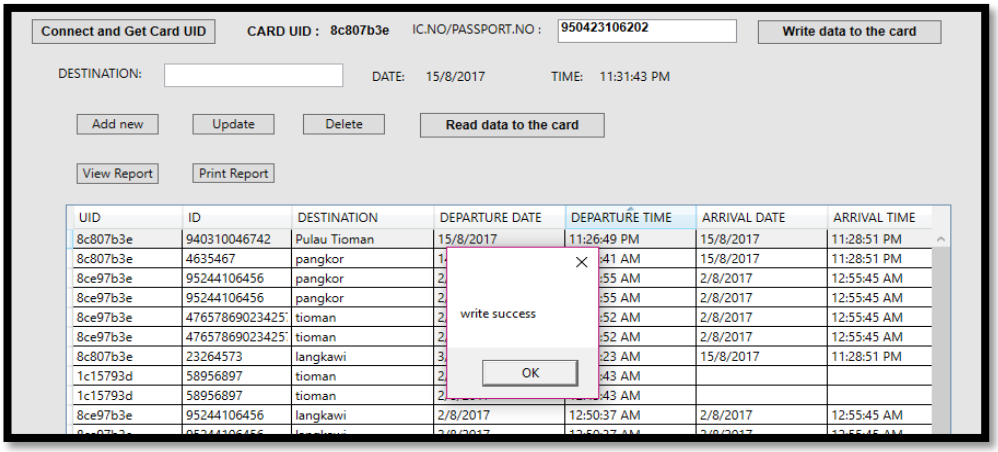


Figure 6.7: Write data is success

6.5.3.3 Read data to the card.

The Read data to the card function button test is related to Table 6.3 which is the function to read IC.NO/PASSPORT number that is written to the RFID card or RFID tag. The coding commands are referred to section 5.3.3.3. Figure 6.8 below is the output where the data is read to the card is success. The message box display data is exist and in the red box data is displayed in the DataGrid.

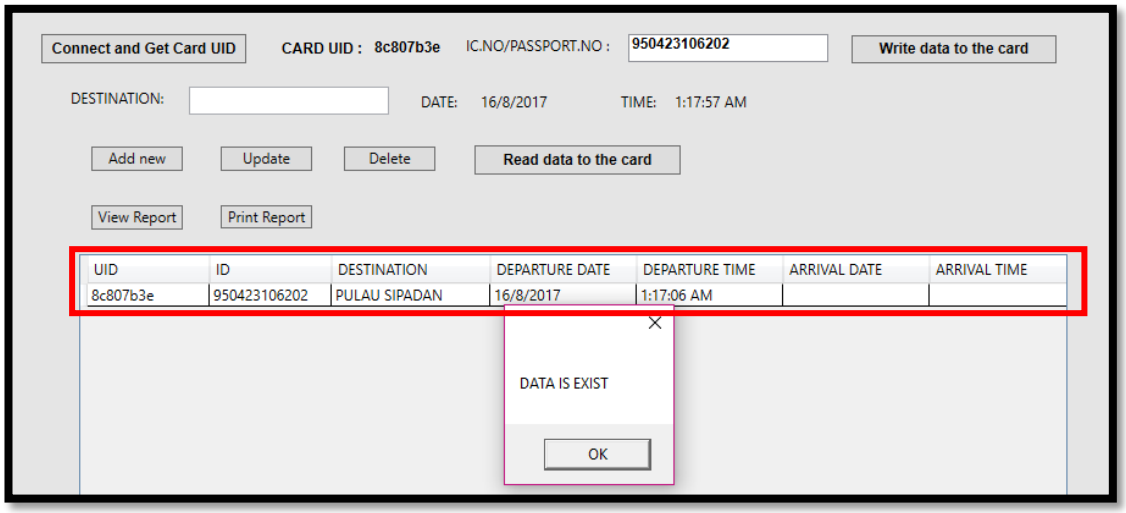


Figure 6.8: Read data output is success

6.5.3.4 Add new button.

The Add new function button test is related to Table 6.3 which is is to add new data into the database. The coding commands are referred to section 5.3.3.4. Figure 6.9 below is the output where the data is added is success. Figure 6.9 show the message box display data is added and there is new entry display in the DataGrid in Figure 6.10.

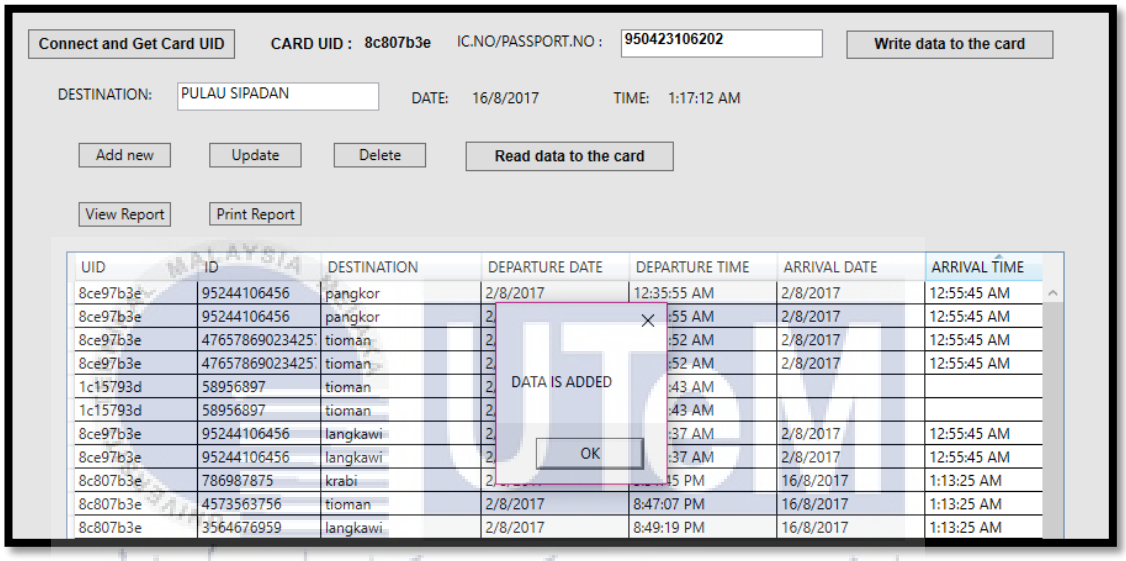


Figure 6.9: Data is added success

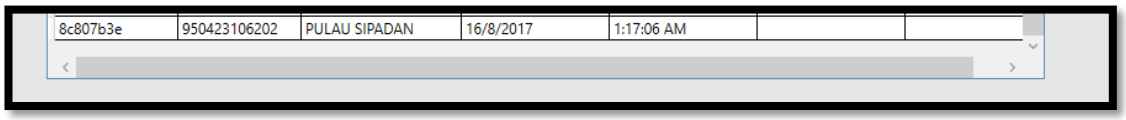


Figure 6.10: New entry is added in DataGrid

6.5.3.5 Update button.

The Update function button test is related to Table 6.3 is to update any changes to the existed data in the database. The coding commands are referred to section 5.3.3.5. Figure 6.11 below is the output where the data is updated is success. Figure 6.11 show the message box display data is updated and there is new updated entry display in the DataGrid in Figure 6.12.

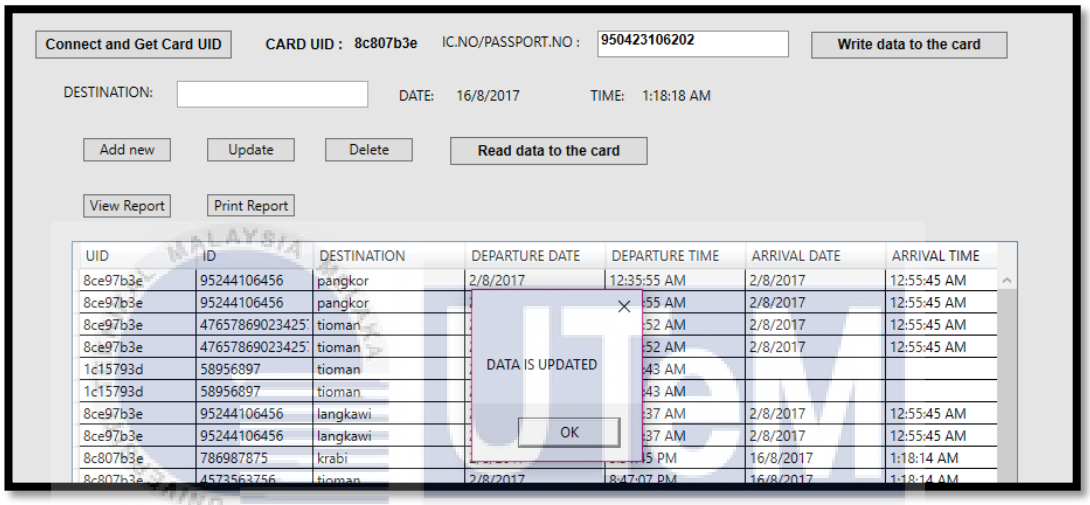


Figure 6.11: Data is updated displayed in message box

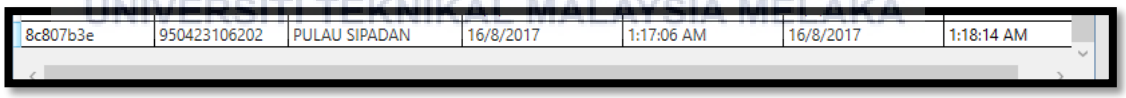


Figure 6.12: New updated data displayed in DataGrid

6.5.3.6 Delete button.

The Delete function button test is related to Table 6.3 which is to delete or remove any unwanted data from database. The coding commands are referred to section 5.3.3.6. Figure 6.13 and Figure 6.14 below are the output where the data is deleted is success. Figure 6.13 show the message box display data is deleted and there is no deleted data is displayed in the DataGrid in Figure 6.14.

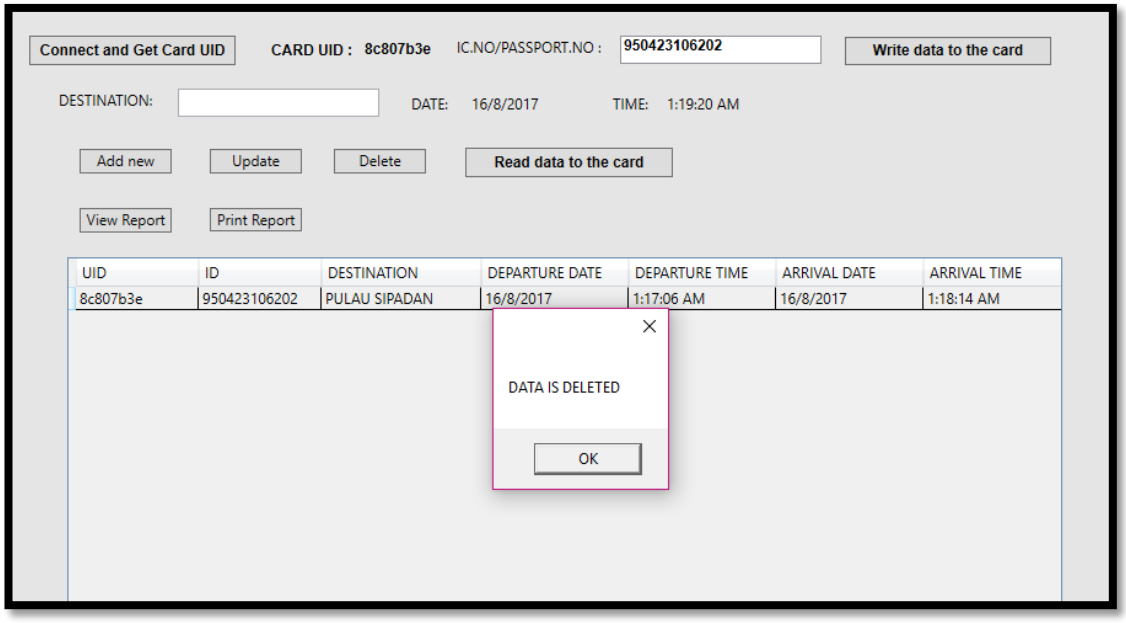
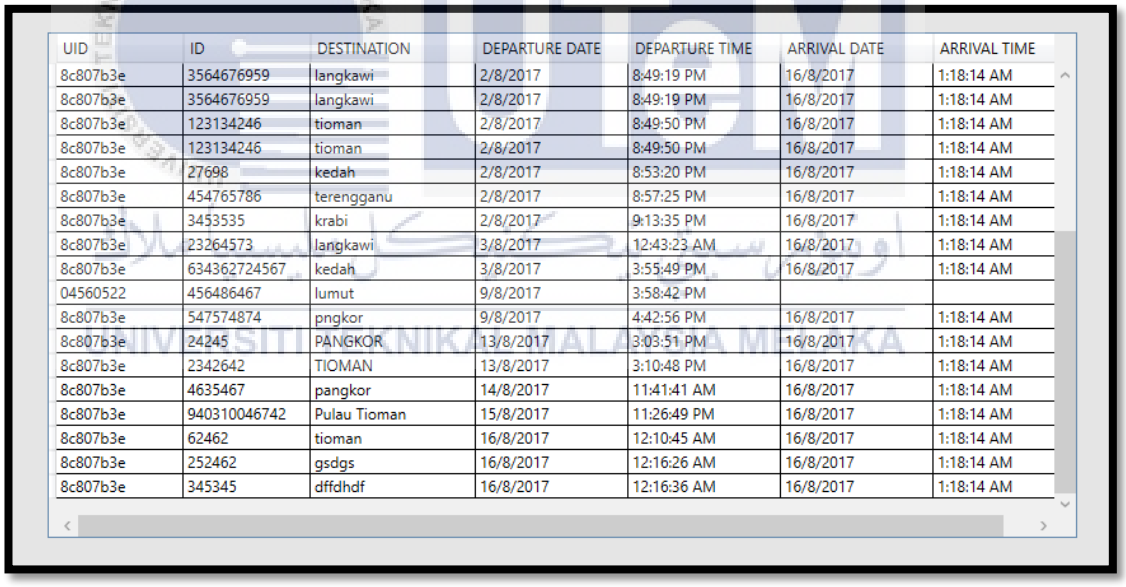


Figure 6.13: Data is deleted output



UID	ID	DESTINATION	DEPARTURE DATE	DEPARTURE TIME	ARRIVAL DATE	ARRIVAL TIME
8c807b3e	3564676959	langkawi	2/8/2017	8:49:19 PM	16/8/2017	1:18:14 AM
8c807b3e	3564676959	langkawi	2/8/2017	8:49:19 PM	16/8/2017	1:18:14 AM
8c807b3e	123134246	tioman	2/8/2017	8:49:50 PM	16/8/2017	1:18:14 AM
8c807b3e	123134246	tioman	2/8/2017	8:49:50 PM	16/8/2017	1:18:14 AM
8c807b3e	27698	kedah	2/8/2017	8:53:20 PM	16/8/2017	1:18:14 AM
8c807b3e	454765786	terengganu	2/8/2017	8:57:25 PM	16/8/2017	1:18:14 AM
8c807b3e	3453535	krabi	2/8/2017	9:13:35 PM	16/8/2017	1:18:14 AM
8c807b3e	23264573	langkawi	3/8/2017	12:43:23 AM	16/8/2017	1:18:14 AM
8c807b3e	634362724567	kedah	3/8/2017	3:55:49 PM	16/8/2017	1:18:14 AM
04560522	456486467	lumut	9/8/2017	3:58:42 PM		
8c807b3e	547574874	pngkor	9/8/2017	4:42:56 PM	16/8/2017	1:18:14 AM
8c807b3e	24245	PANGKOR	13/8/2017	3:03:51 PM	16/8/2017	1:18:14 AM
8c807b3e	2342642	TIOMAN	13/8/2017	3:10:48 PM	16/8/2017	1:18:14 AM
8c807b3e	4635467	pangkor	14/8/2017	11:41:41 AM	16/8/2017	1:18:14 AM
8c807b3e	940310046742	Pulau Tioman	15/8/2017	11:26:49 PM	16/8/2017	1:18:14 AM
8c807b3e	62462	tioman	16/8/2017	12:10:45 AM	16/8/2017	1:18:14 AM
8c807b3e	252462	gsdgs	16/8/2017	12:16:26 AM	16/8/2017	1:18:14 AM
8c807b3e	345345	dffdhdf	16/8/2017	12:16:36 AM	16/8/2017	1:18:14 AM

Figure 6.14: There no data that is deleted displayed in DataGrid

6.5.3.7 Print Report button.

The Print report function button test is related to table 6.3 which is to print data displayed in DataGrid. The coding commands are referred to section 5.3.3.7. Figure 6.15, Figure 6.16 and Figure 6.17 below are the output where the output for report print is success. Figure 6.15 show the message box of print preferences and save the printed report preferences in Figure 6.16. As for Figure 6.17 is the output of printed report that is saved in .pdf file.

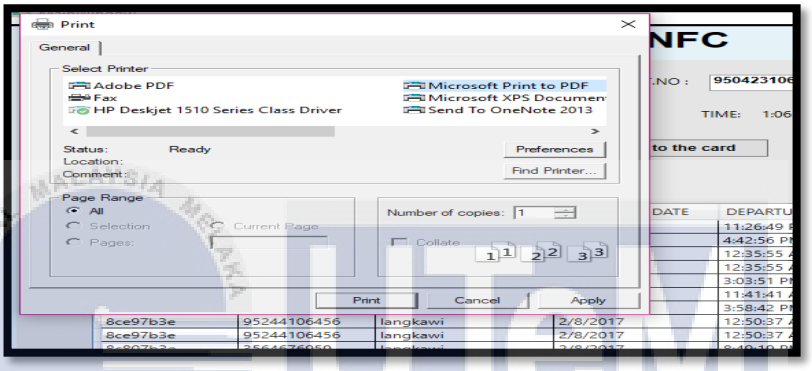


Figure 6.3: Printing preferences

From Figure 6.15 above, Microsoft Print to PDF is the option to print the report in .pdf file.

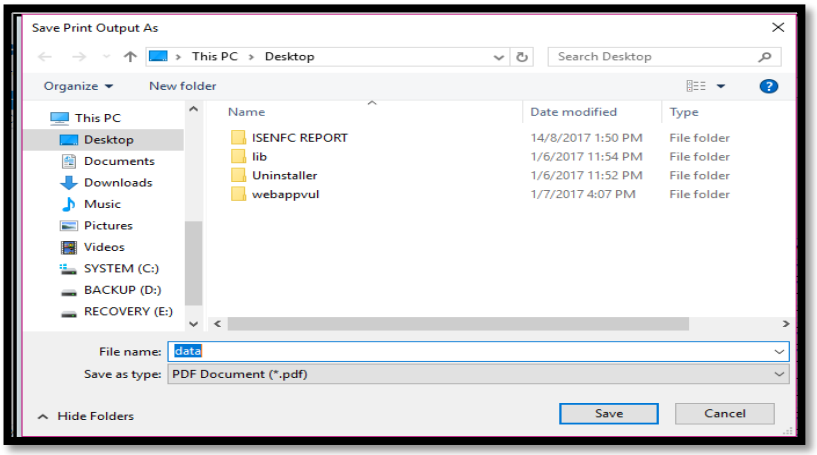


Figure 6.16: Save as .pdf file

From Figure 6.16 above, the report is save as data in .pdf file.

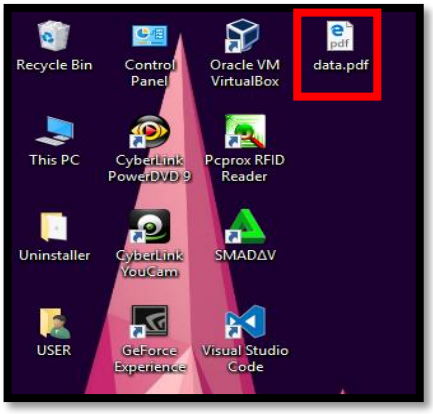


Figure 6.17: Output of printed report in .pdf

From Figure 6.17 above, show the output of the printed report.

6.5.3.8 View Report button.

The View report function button test is related to table 6.3 which is to refresh data from database to DataGrid. The coding commands are referred to section 5.3.3.8. Figure 6.18 below is the output where the output for view report is success. Figure 6.18 show all the data displayed in DataGrid from the database.

UID	ID	DESTINATION	DEPARTURE DATE	DEPARTURE TIME	ARRIVAL DATE	ARRIVAL TIME
8c807b3e	23264573	langkawi	3/8/2017	12:43:23 AM	16/8/2017	1:03:20 AM
04560522	456486467	lumut	9/8/2017	3:58:42 PM		
8ce97b3e	95244106456	pangkor	2/8/2017	12:35:55 AM	2/8/2017	12:55:45 AM
8ce97b3e	95244106456	pangkor	2/8/2017	12:35:55 AM	2/8/2017	12:55:45 AM
8c807b3e	24245	PANGKOR	13/8/2017	3:03:51 PM	16/8/2017	1:03:20 AM
8c807b3e	4635467	pangkor	14/8/2017	11:41:41 AM	16/8/2017	1:03:20 AM
8c807b3e	547574874	pngkor	9/8/2017	4:42:56 PM	16/8/2017	1:03:20 AM
8c807b3e	940310046742	Pulau Tioman	15/8/2017	11:26:49 PM	16/8/2017	1:03:20 AM
8c807b3e	454765786	terengganu	2/8/2017	8:57:25 PM	16/8/2017	1:03:20 AM
8ce97b3e	476578690234257	tioman	2/8/2017	12:36:52 AM	2/8/2017	12:55:45 AM
8ce97b3e	476578690234257	tioman	2/8/2017	12:36:52 AM	2/8/2017	12:55:45 AM
1c15793d	58956897	tioman	2/8/2017	12:43:43 AM		
1c15793d	58956897	tioman	2/8/2017	12:43:43 AM		
8c807b3e	4573563756	tioman	2/8/2017	8:47:07 PM	16/8/2017	1:03:20 AM
8c807b3e	123134246	tioman	2/8/2017	8:49:50 PM	16/8/2017	1:03:20 AM
8c807b3e	123134246	tioman	2/8/2017	8:49:50 PM	16/8/2017	1:03:20 AM
8c807b3e	2342642	TIOMAN	13/8/2017	3:10:48 PM	16/8/2017	1:03:20 AM
8c807b3e	62462	tioman	16/8/2017	12:10:45 AM	16/8/2017	1:03:20 AM

Figure 6.18: Output of View report

6.5.3.9 Automatic current date and time.

The automatic current date and time function test is related to table 6.3 which is to display automatic current date and time that is tally with the time and date of laptop/pc. The coding commands are referred to section 5.3.3.9. Figure 6.19 below is the output where the date and time is success to display in the interface is success. Figure 6.20 show the date and time at pc/laptop is tally with date and time at the interface.

Figure 6.19: Date and time displayed in the interface

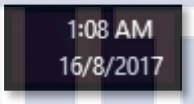
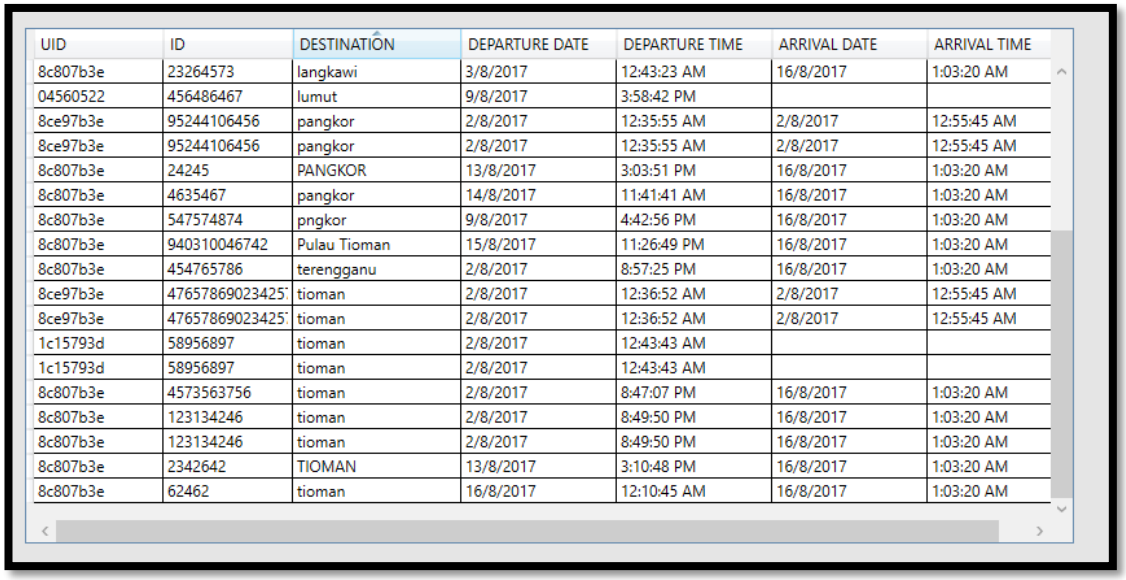


Figure 6.20: Date and time at pc/laptop

6.5.3.10 Display data to DataGrid.

The display data to DataGrid function test is related to table 6.3 which is to display data from database. The coding commands are referred to section 5.3.3.10. Figure 6.21 below is the output where data from database is display in the DataGrid is success.

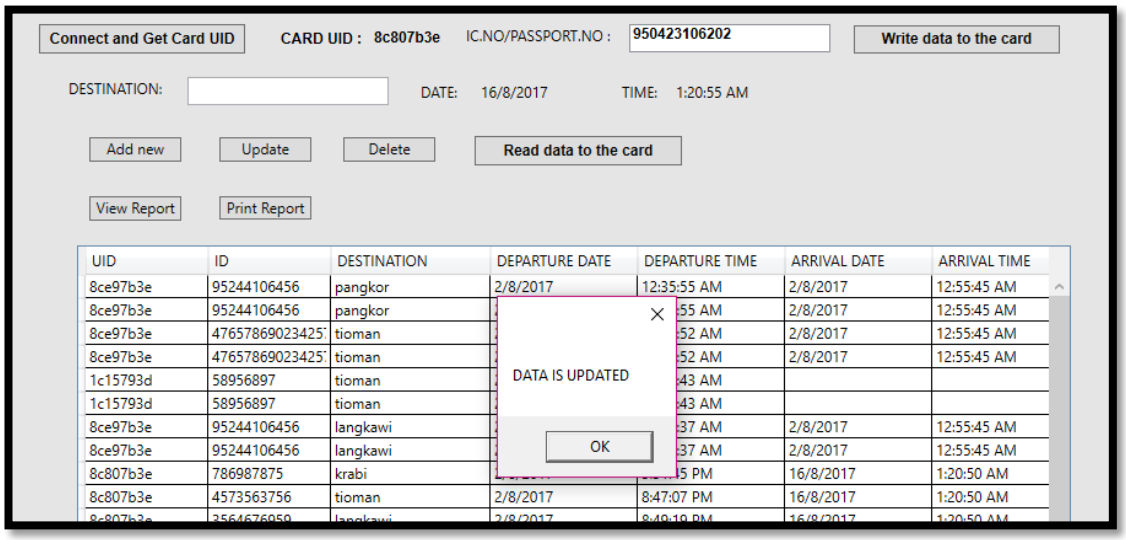


UID	ID	DESTINATION	DEPARTURE DATE	DEPARTURE TIME	ARRIVAL DATE	ARRIVAL TIME
8c807b3e	23264573	langkawi	3/8/2017	12:43:23 AM	16/8/2017	1:03:20 AM
04560522	456486467	lumut	9/8/2017	3:58:42 PM		
8ce97b3e	95244106456	pangkor	2/8/2017	12:35:55 AM	2/8/2017	12:55:45 AM
8ce97b3e	95244106456	pangkor	2/8/2017	12:35:55 AM	2/8/2017	12:55:45 AM
8c807b3e	24245	PANGKOR	13/8/2017	3:03:51 PM	16/8/2017	1:03:20 AM
8c807b3e	4635467	pangkor	14/8/2017	11:41:41 AM	16/8/2017	1:03:20 AM
8c807b3e	547574874	pngkor	9/8/2017	4:42:56 PM	16/8/2017	1:03:20 AM
8c807b3e	940310046742	Pulau Tioman	15/8/2017	11:26:49 PM	16/8/2017	1:03:20 AM
8c807b3e	454765786	terengganu	2/8/2017	8:57:25 PM	16/8/2017	1:03:20 AM
8ce97b3e	476578690234257	tioman	2/8/2017	12:36:52 AM	2/8/2017	12:55:45 AM
8ce97b3e	476578690234257	tioman	2/8/2017	12:36:52 AM	2/8/2017	12:55:45 AM
1c15793d	58956897	tioman	2/8/2017	12:43:43 AM		
1c15793d	58956897	tioman	2/8/2017	12:43:43 AM		
8c807b3e	4573563756	tioman	2/8/2017	8:47:07 PM	16/8/2017	1:03:20 AM
8c807b3e	123134246	tioman	2/8/2017	8:49:50 PM	16/8/2017	1:03:20 AM
8c807b3e	123134246	tioman	2/8/2017	8:49:50 PM	16/8/2017	1:03:20 AM
8c807b3e	2342642	TIOMAN	13/8/2017	3:10:48 PM	16/8/2017	1:03:20 AM
8c807b3e	62462	tioman	16/8/2017	12:10:45 AM	16/8/2017	1:03:20 AM

Figure 6.21: Displayed data in the DataGrid

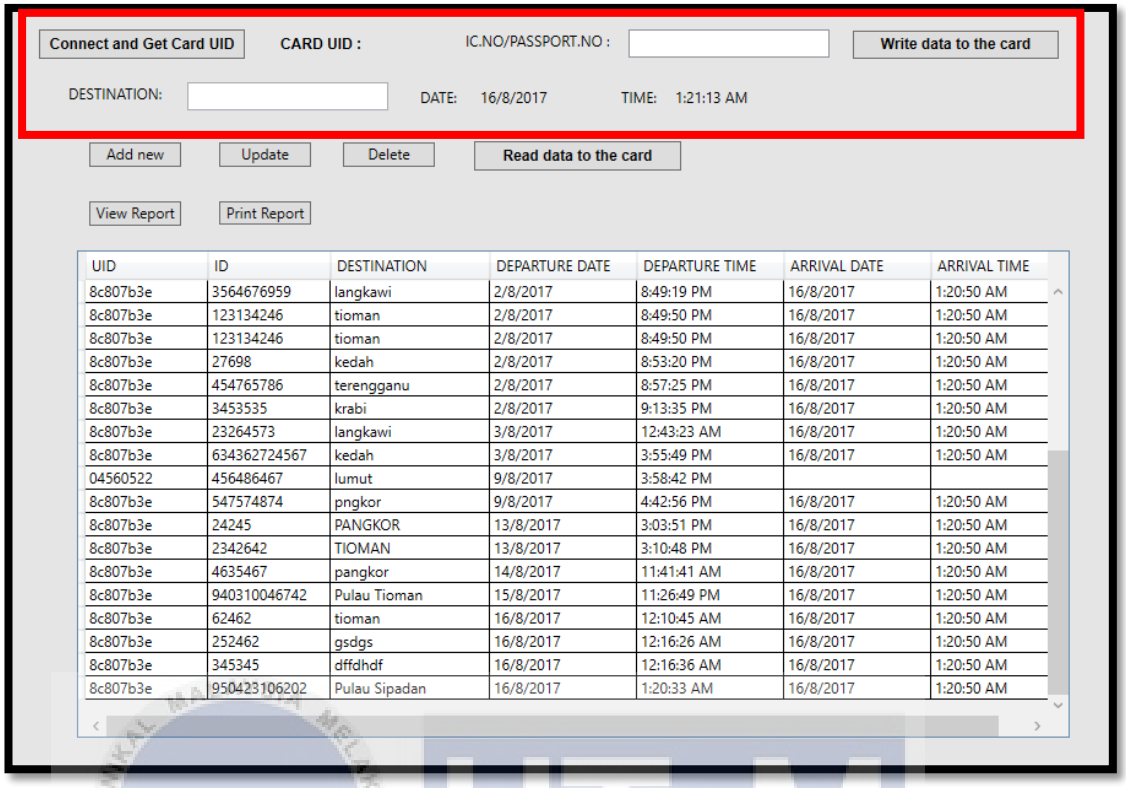
6.5.3.11 ClearData() function.

The ClearData() function test is related to table 6.3 which is to empty the Text Blocks and Text Boxes after data is added, update or delete. The coding commands are referred to section 5.3.3.11. Figure 6.22 and Figure 6.23 below are the output where the function is success in testing. Figure 6.23 show in red box, that the Text Blocks and Text Boxes are empty after data is updated.



UID	ID	DESTINATION	DEPARTURE DATE	DEPARTURE TIME	ARRIVAL DATE	ARRIVAL TIME
8ce97b3e	95244106456	pangkor	2/8/2017	12:35:55 AM	2/8/2017	12:55:45 AM
8ce97b3e	95244106456	pangkor		:55 AM	2/8/2017	12:55:45 AM
8ce97b3e	476578690234257	tioman		:52 AM	2/8/2017	12:55:45 AM
8ce97b3e	476578690234257	tioman		:52 AM	2/8/2017	12:55:45 AM
1c15793d	58956897	tioman		:43 AM		
1c15793d	58956897	tioman		:43 AM		
8ce97b3e	95244106456	langkawi		:37 AM	2/8/2017	12:55:45 AM
8ce97b3e	95244106456	langkawi		:37 AM	2/8/2017	12:55:45 AM
8c807b3e	786987875	krabi		:5 PM	16/8/2017	1:20:50 AM
8c807b3e	4573563756	tioman	2/8/2017	8:47:07 PM	16/8/2017	1:20:50 AM

Figure 6.22: Output for update data



UID	ID	DESTINATION	DEPARTURE DATE	DEPARTURE TIME	ARRIVAL DATE	ARRIVAL TIME
8c807b3e	3564676959	langkawi	2/8/2017	8:49:19 PM	16/8/2017	1:20:50 AM
8c807b3e	123134246	tioman	2/8/2017	8:49:50 PM	16/8/2017	1:20:50 AM
8c807b3e	123134246	tioman	2/8/2017	8:49:50 PM	16/8/2017	1:20:50 AM
8c807b3e	27698	kedah	2/8/2017	8:53:20 PM	16/8/2017	1:20:50 AM
8c807b3e	454765786	terengganu	2/8/2017	8:57:25 PM	16/8/2017	1:20:50 AM
8c807b3e	3453535	krabi	2/8/2017	9:13:35 PM	16/8/2017	1:20:50 AM
8c807b3e	23264573	langkawi	3/8/2017	12:43:23 AM	16/8/2017	1:20:50 AM
8c807b3e	634362724567	kedah	3/8/2017	3:55:49 PM	16/8/2017	1:20:50 AM
04560522	456486467	lumut	9/8/2017	3:58:42 PM		
8c807b3e	547574874	pngkor	9/8/2017	4:42:56 PM	16/8/2017	1:20:50 AM
8c807b3e	24245	PANGKOR	13/8/2017	3:03:51 PM	16/8/2017	1:20:50 AM
8c807b3e	2342642	TIOMAN	13/8/2017	3:10:48 PM	16/8/2017	1:20:50 AM
8c807b3e	4635467	pangkor	14/8/2017	11:41:41 AM	16/8/2017	1:20:50 AM
8c807b3e	940310046742	Pulau Tioman	15/8/2017	11:26:49 PM	16/8/2017	1:20:50 AM
8c807b3e	62462	tioman	16/8/2017	12:10:45 AM	16/8/2017	1:20:50 AM
8c807b3e	252462	gsdgs	16/8/2017	12:16:26 AM	16/8/2017	1:20:50 AM
8c807b3e	345345	dffdhdf	16/8/2017	12:16:36 AM	16/8/2017	1:20:50 AM
8c807b3e	950423106202	Pulau Sipadan	16/8/2017	1:20:33 AM	16/8/2017	1:20:50 AM

Figure 6.23: Text Blocks and Text Boxes are empty

6.6 Conclusion

This testing phase is conducted to test the functionality of NFC reader/writer, connection between SQLite database with the application and the application interface. In the next chapter, conclusion of the project will be discussed. It includes project summarization, limitation and future works.

CHAPTER VII

CONCLUSION

7.1 Introduction

This chapter explains about conclusion of the project. It explain about overall information of this project generally. The explanation include contribution and limitation of the project, future works for this project and also on how the project can be enhanced and improved by others.

7.2 Project Summarization

7.2.1 Project Objective

The objective of this project that has been identified is listed below:

i. To design entry system interfaces that can use NFC and RFID technology. All background study has been done in literature review phase where all related details are referred such as ferry boarding system is the embarkation and disembarkation of passenger and/or vehicles into the ferry from one terminal to the other ferry terminal by Mišković et al., (2016) and tagging and tracking. It give ideas on how to develop the project from background

study. Thus, “iSe-NFC Ferry: Intelligent Secured Ferry Boarding System using NFC integrated with RFID tag for Malaysia’s Tourism” can be done.

ii. To improve passenger record management by RFID tagging. All the RFID tag have its own unique ID where it can improve the record management by save it in the database. Thus, the record management will be efficient.

iii. To track passenger by using NFC-enabled device and RFID tag. Passenger will be track with the timestamp when the RFID tag is tapped with NFC-enabled device. The outcome will show that the application that connected with NFC-enabled device validates the functions of the system. The details of the tag unique ID, passenger identification number, destination and time in and out of passenger.

7.2.2 Project Strength and Weakness

In generally, the project has a lot of strength and weakness. The strength of this project is the cost of the NFC-enable device (reader) is affordable and it is a commercialize build in reader. Also, RFID tag is waterproof and easy to keep. The weakness for this project is RFID tag is not active RFID tag and it is have limited storage in saving information.

The NFC-enable device (reader) is commercialize build in reader which is easy to plug in to the laptop or pc. The price is affordable which is RM 150 – RM 250. Features of the NFC reader is, it have LED light and buffer sound where it act as indicator if there is a card is tapped on it. It also have a build in USB cable that allow user to attach it to the laptop or pc. RFID tag is waterproof and easy to keep. Price for RFID tag is affordable and it can be recycle.

As for the weakness, the RFID tag that are used for the project is passive tag and it have low storage to save information which is 48 bytes of data only, that can be save. The RFID tag is passive tag which is only it can be tapped and detect when it is near to the NFC reader.

7.3 Project Contribution

This project was built to improve the current ferries' services in boarding system for Malaysia's tourism. New methods should be implemented with advanced technology. With aid of NFC and RFID the current issues can be solved.

It can improve the entry system interfaces that can use NFC and RFID technology. The entry system is inefficient before this, where passenger have to wait and had a long que to get tickets. Also the process to get tickets is complicated. With this system the que process will be a lot more efficient and effective.

Moreover, it can improve passenger record management by using RFID tag. Passenger will give their identification card to the staff at the counter. Then, staff will key in identification number. RFID tag will tap at the NFC-enabled device. After RFID card is tapped, the RFID tag is given to the passenger as a tag and the confirmation of the passenger to board the ferry. The data will be save in the database for safe keeping.

Passenger can be tracked by using NFC-enabled device and RFID tag. As stated just now, RFID tag will be the tag and as a confirmation of the passenger in the ferry. The time and the destination is recorded as a proved of tracking process. When passenger arrive at the destination, passenger need to retrieve the RFID tag at the counter. These process will record the arrival time and date, means it indicates passenger is safely arrived at their destination. Also, it enable the authority to see the record in and out access of passenger.

7.4 Project Limitation

The limitation of this project is passenger need to give their identification card to the staff, where the card should be tapped directly on the NFC-enabled device. But this system do not cater the encryption of the identification card as it is confidential. Other than that, there is no centralize database or cloud storage. Also, this system do not cater the live physical tracking using GPS. This system also do not cater ticket payment. The RFID tag is not compatible if it is already written by other commercialize software.

7.5 Future Works

This project can be upgraded for a better performance and function. The future works that can be considered are:

i. Identification card can be tap directly with NFC-enabled device. The identification card could not tap directly because this system do not cater the encryption of the chip in the card. Suggestion for future works, this project should be partner with Jabatan Pendaftaran Negara. The identification card need to be cater for developer can develop system or application that enable identification to tap directly on the NFC-enabled device.

ii. Establish a centralize database or cloud storage to enhance the safe keeping of the record. This will make the system more useful as it can give more storage on record keeping. Suggestion for future work is, a developer of the system need to make a centralize database or cloud storage.

iii. Add a live physical tracking integrated with GPS or Google Map. This features will enhance on the tracking process where passenger can be monitor effectively. Suggestion for future works, developer need to add live physical tracking integrated with GPS or Google Map.

iv. Current type of RFID tag need to be replace with active RFID tag. Currently, passive RFID tag is utilize. Thus, it have limitation for detecting and live physical tracking. The suggestion is, current RFID tag need to be replace with active RFID tag.

v. Replace RFID card with other RFID tag that have extra bytes to save information. RFID card only have 48 bytes in order to save information. For suggestion, use Mifare PLUS card (RFID tag) which have extra bytes and more secure encryption to save information.

vi. Add and enable the offline application to online application. Currently, the application is offline, thus it is difficult for data synchronization. For suggestion, developer need to add and enable online features to the system.

vii. Add ticket payment function. This system do not cater ticket payment function. Thus, for improvement, developer need to add ticket payment sorting by age and price of destination.

viii. Add and enable format RFID tag functionality. It is because the written RFID tag that is already written by commercialize software cannot be scan. As for suggestion, find a scripting on how to format written tag that is already written by commercialize software.

7.6 Conclusion

In conclusion, this project successfully meets the objective that has been identified earlier in this project. Hopefully, iSe-NFC Ferry: Intelligent Secured Ferry Boarding System using NFC integrated with RFID tag for Malaysia's Tourism can enhance Malaysian Tourism that focus on ferry services. With technology, it also can provide secure ferry services to all tourist.

REFERENCES

Alsaedi, S. (2015). Nfc-enabled smartphone application for theme park safety and control, (March).

Chopade, P. R., Deshmukh, P., Kamble, K., & Nazarkar, D. (2016). NFC Based Health Care System, *3*(3), 414–419.

CNRFID. (n.d.). RFID frequency ranges. *French National RFID Center*. Retrieved from http://www.centrenational-rfid.com/rfid-frequency-ranges-article-16-gb-ruid-202.html

Huang, J. C. S., Lin, Y. T., Yu, J. K. L., Liu, K., & Kuo, Y. H. (2015). A wearable NFC wristband to locate dementia patients through a participatory sensing system. *Proceedings - 2015 IEEE International Conference on Healthcare Informatics, ICHI 2015*, 208–212. https://doi.org/10.1109/ICHI.2015.32

Jain, G., & Dahiya, S. (2015). NFC: Advantages, Limits and Future Scope. *International Journal on Cybernetics & Informatics*, *4*(4), 1–12. https://doi.org/10.5121/ijci.2015.4401

Köstinger, H., Gobber, M., Grechenig, T., Tappeiner, B., & Schramm, W. (2013). Developing a NFC based patient identification and ward round system for mobile devices using the android platform. *IEEE EMBS Special Topic Conference on Point-of-Care (POC) Healthcare Technologies: Synergy Towards Better Global Healthcare, PHT 2013*, 176–179. https://doi.org/10.1109/PHT.2013.6461313

Mišković, T., Ristov, P., & Markić, Z. (2016). RFID Control System for the Embarkation/Disembarkation of Passengers and Vehicles on/from Ferries. *Transactions on Maritime Science*, *5*(2), 161–171. https://doi.org/10.7225/toms.v05.n02.007

Patil, V., Varma, N., Vinchurkar, S., & Patil, B. (2014). NFC based health monitoring and controlling system. *2014 IEEE Global Conference on Wireless Computing & Networking (GCWCN)*, 133–137. https://doi.org/10.1109/GCWCN.2014.7030864

Penttilä, M., Siira, E., & Tihinen, M. (2016). Mobile Payment Ecosystems in Transition. *International Journal of Scientific and Technical Research in Engineering (IJSTRE)*, *1*(6).

Rfid, P. (2012). Understanding RFID (Radio Frequency Identification), (November), 1–19.

RFIP Blog, T. (2017). The RFIP Blog _ News from RFIP Ltd, expert consultants in RFID technology. wordpress. Retrieved from https://rfipblog.wordpress.com/

Riekki, J., Cortés, M., Hytönen, M., Sánchez, I., & Korkeamäki, R. L. (2013). Touching nametags with NFC phones: A playful approach to learning to read. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7775 LNCS*, 228–242. https://doi.org/10.1007/978-3-642-37919-2_14

Timalsina, S. K., Bhusal, R., & Moh, S. (2012). NFC and its application to mobile payment: Overview and comparison. *Proceedings - ICIDT 2012, 8th International Conference on Information Science and Digital Content Technology*, *1*(1), 203–206.

Peltier, T. R. (2013). *Information Security Fundamentals, Second Edition.* Boca Raton: CRC Press.

Whitman, M. E., & Mattord, H. J. (2014). *Principles of Information Security.* Boston: Cengage Learning.

APPENDIX

(USER MANUAL)

Source code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Windows.Threading;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Data.Sql;
using System.Data.SqlServerCe;
using System.Data.SQLite;
using System.Text.RegularExpressions;



namespace Card
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        int retCode;
        int hCard;
        int hContext;
        int Protocol;
        public bool connActive = false;
        string readername = "ACS ACR122 0";      // change
depending on reader
        public byte[] SendBuff = new byte[263];
        public byte[] RecvBuff = new byte[263];
        public int SendLen, RecvLen, nBytesRet, reqType,
Aprotocol, dwProtocol, cbPciLength;
        public Card.SCARD_READERSTATE RdrState;
        public Card.SCARD_IO_REQUEST pioSendRequest;


        DispatcherTimer t;
        DateTime start;
        // char UID;
        string dbConnectionString = @"Data Source =
database.db;Version = 3;";



        public MainWindow()
        {
```

```
            this.InitializeComponent();
            SelectDevice();
            establishContext();
            t = new DispatcherTimer(new TimeSpan(0, 0, 0, 0, 50),
DispatcherPriority.Background,
            t_Tick, Dispatcher.CurrentDispatcher); t.IsEnabled =
true;
            start = DateTime.Now;
            displaydata();
            ClearData();


        }



        //Display data on datagrid
        private void displaydata()
        {
            SQLiteConnection sqliteCon = new
SQLiteConnection(dbConnectionString);
            sqliteCon.Open();
            string Query = "select
uid,ic,destination,departuredate,departuretime,arrivaldate,arriva
ltime from nfc";
            SQLiteCommand createCommand = new
SQLiteCommand(Query, sqliteCon);
            createCommand.ExecuteNonQuery();
            //SQLiteDataReader dr =
createCommand.ExecuteReader();
            SQLiteDataAdapter dataAdp = new
SQLiteDataAdapter(createCommand);
            DataTable dt = new DataTable("nfc");
            dataAdp.Fill(dt);
            passengerinfo.ItemsSource = dt.DefaultView;
            dataAdp.Update(dt);
            sqliteCon.Close();
        }

        //CLEAR DATA
        private void ClearData()
        {

            textBlock1.Text = "";
            textBox1.Text = "";
            textBox2.Text ="";
            DepartureDate.Content = "";
            DepartureTime.Text = "";


        }


        // to set DATE AND TIME (TIMESTAMP)
        private void t_Tick(object sender, EventArgs e)
        {
```

```
            DepartureDate.Content =
DateTime.Now.ToShortDateString();
            DepartureTime.Text = DateTime.Now.ToLongTimeString();
            t.Start();
        }


        private void Window_Loaded(object sender, RoutedEventArgs
e)
        {
            SQLiteConnection sqliteCon = new
SQLiteConnection(dbConnectionString);
            //Open connection to database
            try
            {
                sqliteCon.Open();
                string Query = "select
uid,ic,destination,departuredate,departuretime,arrivaldate,arriva
ltime from nfc";
                SQLiteCommand createCommand = new
SQLiteCommand(Query, sqliteCon);
                createCommand.ExecuteNonQuery();
                //SQLiteDataReader dr =
createCommand.ExecuteReader();
                SQLiteDataAdapter dataAdp = new
SQLiteDataAdapter(createCommand);
                DataTable dt = new DataTable("nfc");

                dataAdp.Fill(dt);
                passengerinfo.ItemsSource = dt.DefaultView;
                dataAdp.Update(dt);
                MessageBox.Show("DATA IS EXIST");
                sqliteCon.Close();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }

        }

        public void SelectDevice()
        {
            List<string> availableReaders = this.ListReaders();
            this.RdrState = new Card.SCARD_READERSTATE();
            readername =
availableReaders[0].ToString();//selecting first device
            this.RdrState.RdrName = readername;
        }




        public List<string> ListReaders()
        {
            int ReaderCount = 0;
            List<string> AvailableReaderList = new
List<string>();
```

```
            //Make sure a context has been established before
            //retrieving the list of smartcard readers.
            retCode = Card.SCardListReaders(hContext, null, null,
ref ReaderCount);
            if (retCode != Card.SCARD_S_SUCCESS)
            {
                MessageBox.Show(Card.GetScardErrMsg(retCode));
                //connActive = false;
            }

            byte[] ReadersList = new byte[ReaderCount];

            //Get the list of reader present again but this time
add sReaderGroup, retData as 2rd & 3rd parameter respectively.
            retCode = Card.SCardListReaders(hContext, null,
ReadersList, ref ReaderCount);
            if (retCode != Card.SCARD_S_SUCCESS)
            {
                MessageBox.Show(Card.GetScardErrMsg(retCode));
            }

            string rName = "";
            int indx = 0;
            if (ReaderCount > 0)
            {
                // Convert reader buffer to string
                while (ReadersList[indx] != 0)
                {

                    while (ReadersList[indx] != 0)
                    {
                        rName = rName + (char)ReadersList[indx];
                        indx = indx + 1;
                    }

                    //Add reader name to list
                    AvailableReaderList.Add(rName);
                    rName = "";
                    indx = indx + 1;

                }
            }
            return AvailableReaderList;

        }

        internal void establishContext()
        {
            retCode =
Card.SCardEstablishContext(Card.SCARD_SCOPE_SYSTEM, 0, 0, ref
hContext);
            if (retCode != Card.SCARD_S_SUCCESS)
            {
                MessageBox.Show("Check your device and please
restart again", "Reader not connected", MessageBoxButton.OK,
MessageBoxImage.Warning);
                connActive = false;
```

```
                return;
            }
        }

        //connect and get UID button
        private void Button_Click_1(object sender,
RoutedEventArgs e)
        {

            if (connectCard())
            {
                string cardUID = getcardUID();
                textBlock1.Text = cardUID; //displaying on text
block
            }
        }

        public bool connectCard()
        {
            connActive = true;

            retCode = Card.SCardConnect(hContext, readername,
Card.SCARD_SHARE_SHARED,
                        Card.SCARD_PROTOCOL_T0 |
Card.SCARD_PROTOCOL_T1, ref hCard, ref Protocol);

            if (retCode != Card.SCARD_S_SUCCESS)
            {
                MessageBox.Show(Card.GetScardErrMsg(retCode),
"Card not available", MessageBoxButton.OK,
MessageBoxImage.Error);
                connActive = false;
                return false;
            }
            return true;
        }

        private string getcardUID()//only for mifare 1k cards
        {
            string cardUID = "";
            byte[] receivedUID = new byte[256];
            Card.SCARD_IO_REQUEST request = new
Card.SCARD_IO_REQUEST();
            request.dwProtocol = Card.SCARD_PROTOCOL_T1;
            request.cbPciLength =
System.Runtime.InteropServices.Marshal.SizeOf(typeof(Card.SCARD_I
O_REQUEST));
            byte[] sendBytes = new byte[] { 0xFF, 0xCA, 0x00,
0x00, 0x00 }; //get UID command      for Mifare cards
            int outBytes = receivedUID.Length;
            int status = Card.SCardTransmit(hCard, ref request,
ref sendBytes[0], sendBytes.Length, ref request, ref
receivedUID[0], ref outBytes);

            if (status != Card.SCARD_S_SUCCESS)
            {
                cardUID = "Error";
            }
```

```
            else
            {
                cardUID =
BitConverter.ToString(receivedUID.Take(4).ToArray()).Replace("-",
string.Empty).ToLower();
            }

            return cardUID;
        }

        //READ CARD UID BUTTON
        private void Button_Click_2(object sender,
RoutedEventArgs e)
        {

            string text = verifyCard("5"); // 5 - is the block we
are reading
            textBox1.Text = text.ToString();


            //to display database values in datagrid
            SQLiteConnection sqliteCon = new
SQLiteConnection(dbConnectionString);
            //Open connection to database
            try
            {
                sqliteCon.Open();
                string Query = "select
uid,ic,destination,departuredate,departuretime,arrivaldate,arriva
ltime from nfc where  ic ='" + this.textBox1.Text + "'";
                SQLiteCommand createCommand = new
SQLiteCommand(Query, sqliteCon);
                createCommand.ExecuteNonQuery();
                //SQLiteDataReader dr =
createCommand.ExecuteReader();
                SQLiteDataAdapter dataAdp = new
SQLiteDataAdapter(createCommand);
                DataTable dt = new DataTable("nfc");
                dataAdp.Fill(dt);
                passengerinfo.ItemsSource = dt.DefaultView;
                dataAdp.Update(dt);
                sqliteCon.Close();

                MessageBox.Show("DATA IS EXIST");

            }

            catch (Exception ex)

            {
               MessageBox.Show(ex.Message);
            }

        }



        public string verifyCard(String Block)
```

```
        {
            string value = "";
            if (connectCard())
            {
                value = readBlock(Block);
            }

            value = value.Split(new char[] { '\0' }, 2,
StringSplitOptions.None)[0].ToString();
            return value;
        }

        public string readBlock(String Block)
        {
            string tmpStr = "";
            int indx;

            if (authenticateBlock(Block))
            {
                ClearBuffers();
                SendBuff[0] = 0xFF; // CLA
                SendBuff[1] = 0xB0;// INS
                SendBuff[2] = 0x00;// P1
                SendBuff[3] = (byte)int.Parse(Block);// P2 :
Block No.
                SendBuff[4] = (byte)int.Parse("16");// Le

                SendLen = 5;
                RecvLen = SendBuff[4] + 2;

                retCode = SendAPDUandDisplay(2);

                if (retCode == -200)
                {
                    return "outofrangeexception";
                }

                if (retCode == -202)
                {
                    return "BytesNotAcceptable";
                }

                if (retCode != Card.SCARD_S_SUCCESS)
                {
                    return "FailRead";
                }

                // Display data in text format
                for (indx = 0; indx <= RecvLen - 1; indx++)
                {
                    tmpStr = tmpStr +
Convert.ToChar(RecvBuff[indx]);
                }

                return (tmpStr);
            }
            else
            {
```

```
                return "FailAuthentication";
            }
        }

        //Write UID button
        private void Button_Click_3(object sender,
RoutedEventArgs e)
        {
            if (connectCard())// establish connection to the
card: you've declared this from previous post
            {
                submitText(textBox1.Text, "5"); // 5 - is the
block we are writing data on the card

                MainWindowClose();
            }
        }

        //submit data method
        public void submitText(String Text, String Block)
        {

            String tmpStr = Text;
            int indx;
            if (authenticateBlock(Block))
            {
                ClearBuffers();
                SendBuff[0] = 0xFF;
// CLA
                SendBuff[1] = 0xD6;
// INS
                SendBuff[2] = 0x00;
// P1
                SendBuff[3] = (byte)int.Parse(Block);
// P2 : Starting Block No.
                SendBuff[4] = (byte)int.Parse("16");
// P3 : Data length

                for (indx = 0; indx <= (tmpStr).Length - 1;
indx++)
                {
                    SendBuff[indx + 5] = (byte)tmpStr[indx];
                }
                SendLen = SendBuff[4] + 5;
                RecvLen = 0x02;

                retCode = SendAPDUandDisplay(2);

                if (retCode != Card.SCARD_S_SUCCESS)
                {
                    MessageBox.Show("fail write");

                }
                else
                {
                    MessageBox.Show("write success");
                }
            }
```

```
            else
            {
                MessageBox.Show("FailAuthentication");
            }
        }

        // block authentication
        private bool authenticateBlock(String block)
        {
            ClearBuffers();
            SendBuff[0] = 0xFF;                                 // CLA
            SendBuff[2] = 0x00;                                 // P1:
same for all source types
            SendBuff[1] = 0x86;                                 // INS:
for stored key input
            SendBuff[3] = 0x00;                                 // P2 :
Memory location;  P2: for stored key input
            SendBuff[4] = 0x05;                                 // P3:
for stored key input
            SendBuff[5] = 0x01;                                 // Byte
1: version number
            SendBuff[6] = 0x00;                                 // Byte 2
            SendBuff[7] = (byte)int.Parse(block);               // Byte
3: sectore no. for stored key input
            SendBuff[8] = 0x60;                                 // Byte 4
: Key A for stored key input
            SendBuff[9] = (byte)int.Parse("1");                 // Byte 5
: Session key for non-volatile memory

            SendLen = 0x0A;
            RecvLen = 0x02;

            retCode = SendAPDUandDisplay(0);

            if (retCode != Card.SCARD_S_SUCCESS)
            {
                //MessageBox.Show("FAIL Authentication!");
                return false;
            }

            return true;
        }

        // clear memory buffers
        private void ClearBuffers()
        {
            long indx;

            for (indx = 0; indx <= 262; indx++)
            {
                RecvBuff[indx] = 0;
                SendBuff[indx] = 0;
            }
        }

        // send application protocol data unit : communication
unit between a smart card reader and a smart card
        private int SendAPDUandDisplay(int reqType)
```

```
        {
            int indx;
            string tmpStr = "";

            pioSendRequest.dwProtocol = Aprotocol;
            pioSendRequest.cbPciLength = 8;

            //Display Apdu In
            for (indx = 0; indx <= SendLen - 1; indx++)
            {
                tmpStr = tmpStr + " " + string.Format("{0:X2}",
SendBuff[indx]);
            }

            retCode = Card.SCardTransmit(hCard, ref
pioSendRequest, ref SendBuff[0],
                                         SendLen, ref pioSendRequest, ref
RecvBuff[0], ref RecvLen);

            if (retCode != Card.SCARD_S_SUCCESS)
            {
                return retCode;
            }

            else
            {
                try
                {
                    tmpStr = "";
                    switch (reqType)
                    {
                        case 0:
                            for (indx = (RecvLen - 2); indx <=
(RecvLen - 1); indx++)
                            {
                                tmpStr = tmpStr + " " +
string.Format("{0:X2}", RecvBuff[indx]);
                            }

                            if ((tmpStr).Trim() != "90 00")
                            {
                                //MessageBox.Show("Return bytes
are not acceptable.");
                                return -202;
                            }

                            break;

                        case 1:

                            for (indx = (RecvLen - 2); indx <=
(RecvLen - 1); indx++)
                            {
                                tmpStr = tmpStr +
string.Format("{0:X2}", RecvBuff[indx]);
                            }

                            if (tmpStr.Trim() != "90 00")
```

```
                                {
                                    tmpStr = tmpStr + " " +
string.Format("{0:X2}", RecvBuff[indx]);
                                }

                                else
                                {
                                    tmpStr = "ATR : ";
                                    for (indx = 0; indx <= (RecvLen -
3); indx++)
                                    {
                                        tmpStr = tmpStr + " " +
string.Format("{0:X2}", RecvBuff[indx]);
                                    }
                                }

                                break;

                            case 2:

                                for (indx = 0; indx <= (RecvLen - 1);
indx++)
                                {
                                    tmpStr = tmpStr + " " +
string.Format("{0:X2}", RecvBuff[indx]);
                                }

                                break;
                        }
                    }
                    catch (IndexOutOfRangeException)
                    {
                        return -200;
                    }
                }
                return retCode;
            }

            //disconnect card reader connection

            public void MainWindowClose()
            {
                if (connActive)
                {
                    retCode = Card.SCardDisconnect(hCard,
Card.SCARD_UNPOWER_CARD);
                }
                //retCode = Card.SCardReleaseContext(hCard);
            }

            private void textBox1_TextChanged_1(object sender,
TextChangedEventArgs e)
            {

            }

            // add button
```

```
        private void Button_Click_4(object sender,
RoutedEventArgs e)
        {
            SQLiteConnection sqliteCon = new
SQLiteConnection(dbConnectionString);
            //Open connection to database
            try
            {
                sqliteCon.Open();
                string Query = "insert into nfc
(uid,ic,destination,departuredate,departuretime) values ('" +
this.textBlock1.Text + "','" + this.textBox1.Text + "','" +
this.textBox2.Text + "', '" + this.DepartureDate.Content + "','"
+ this.DepartureTime.Text + "');";
                SQLiteCommand createCommand = new
SQLiteCommand(Query, sqliteCon);
                createCommand.Parameters.AddWithValue("@uid",
textBlock1.Text);
                createCommand.Parameters.AddWithValue("@ic",
textBox1.Text);

createCommand.Parameters.AddWithValue("@destination",
textBox2.Text);

createCommand.Parameters.AddWithValue("@departuredate",
DepartureDate.Content);

createCommand.Parameters.AddWithValue("@departuretime",
DepartureTime.Text);
                createCommand.ExecuteNonQuery();
                // SQLiteDataReader dr =
createCommand.ExecuteReader();
                MessageBox.Show("DATA IS ADDED");
                sqliteCon.Close();
                displaydata();
                ClearData();

            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }

        }

        //delete button
        private void Button_Click_5(object sender,
RoutedEventArgs e)
        {
            SQLiteConnection sqliteCon = new
SQLiteConnection(dbConnectionString);
            //Open connection to database
            try
            {
                sqliteCon.Open();
                string Query = "delete from nfc where ic='" +
this.textBox1.Text + "'";
```

```
                SQLiteCommand createCommand = new
SQLiteCommand(Query, sqliteCon);
                createCommand.Parameters.AddWithValue("@ic",
textBox1.Text);
                createCommand.ExecuteNonQuery();
                // SQLiteDataReader dr =
createCommand.ExecuteReader();
                MessageBox.Show("DATA IS DELETED");
                sqliteCon.Close();
                displaydata();
                ClearData();

            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }

        }



        private void DataGrid_SelectionChanged_1(object sender,
SelectionChangedEventArgs e)
        {

        }

        private void DataGrid_SelectionChanged_2(object sender,
SelectionChangedEventArgs e)
        {

        }
        private void TextBox_TextChanged_1(object sender,
TextChangedEventArgs e)
        {

        }

        //UPDATE BUTTON
        private void Button_Click_7(object sender,
RoutedEventArgs e)
        {
            SQLiteConnection sqliteCon = new
SQLiteConnection(dbConnectionString);
            //Open connection to database
            try
            {
                sqliteCon.Open();
                string Query = "update nfc set uid ='" +
this.textBlock1.Text + "', arrivaldate='" +
this.DepartureDate.Content + "', arrivaltime='" +
this.DepartureTime.Text + "' where uid ='" + this.textBlock1.Text
+ "' ";
```

```
                SQLiteCommand createCommand = new
SQLiteCommand(Query, sqliteCon);
                createCommand.Parameters.AddWithValue("@uid",
textBlock1.Text);
                createCommand.Parameters.AddWithValue("@ic",
textBox1.Text);

createCommand.Parameters.AddWithValue("@destination",
textBox2.Text);

createCommand.Parameters.AddWithValue("@departuredate",
DepartureDate.Content);

createCommand.Parameters.AddWithValue("@departuretime",
DepartureTime.Text);
                createCommand.ExecuteNonQuery();
                // SQLiteDataReader dr =
createCommand.ExecuteReader();
                sqliteCon.Close();
                displaydata();
                MessageBox.Show("DATA IS UPDATED");
                ClearData();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }

        }
        //view report button
        private void Button_Click_6(object sender,
RoutedEventArgs e)
        {
            SQLiteConnection sqliteCon = new
SQLiteConnection(dbConnectionString);
            //Open connection to database
            try
            {
                sqliteCon.Open();
                string Query = "select
uid,ic,destination,departuredate,departuretime,arrivaldate,arriva
ltime from nfc";
                SQLiteCommand createCommand = new
SQLiteCommand(Query, sqliteCon);
                createCommand.ExecuteNonQuery();
                //SQLiteDataReader dr =
createCommand.ExecuteReader();
                SQLiteDataAdapter dataAdp = new
SQLiteDataAdapter(createCommand);
                DataTable dt = new DataTable("nfc");
                dataAdp.Fill(dt);
                passengerinfo.ItemsSource = dt.DefaultView;
                dataAdp.Update(dt);
                sqliteCon.Close();
            }
            catch (Exception ex)
            {
```

```
                MessageBox.Show(ex.Message);
            }
        }

        //print report button
        private void Button_Click_8(object sender,
RoutedEventArgs e)
        {
            System.Windows.Controls.PrintDialog Printdlg = new
System.Windows.Controls.PrintDialog();
            if ((bool)Printdlg.ShowDialog().GetValueOrDefault())
            {
              Size pageSize = new
Size(Printdlg.PrintableAreaWidth, Printdlg.PrintableAreaHeight);
              //sizing of the element.
              passengerinfo.Measure(pageSize);
               passengerinfo.Arrange(new Rect(5, 5,
pageSize.Width, pageSize.Height));
               Printdlg.PrintVisual(passengerinfo, Title);

            }
        }
    }
}
```