# E-WALLET USING NFC MOBILE APPLICATION DEVELOPMENT

BY

CHONG YAO EN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) COMMUNICATIONS AND NETWORKING

Faculty of Information and Communication Technology

(Kampar Campus)

MAY 2023

**UNIVERSITI TUNKU ABDUL RAHMAN**

# REPORT STATUS DECLARATION FORM

**Title**: E-WALLET USING NFC
MOBILE APPLICATION DEVELOPMENT

**Academic Session**: May 2023

I CHONG YAO EN
**(CAPITAL LETTER)**

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

chong

(Author's signature)

(Supervisor's signature)

**Address**:
24, Jalan Yap Tau Sah
86000 Kluang
Johor

TAN LYK YIN
Supervisor's name

**Date**: 14 September 2023

**Date**: 15 Sep 2023

| Universiti Tunku Abdul Rahman | | | |
|---|---|---|---|
| Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis** | | | |
| Form Number: **FM-IAD-004** | Rev No.: **0** | Effective Date: **21 JUNE 2011** | Page No.: **1 of 1** |

**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

**UNIVERSITI TUNKU ABDUL RAHMAN**

Date: 14 September 2023

**SUBMISSION OF FINAL YEAR PROJECT**

It is hereby certified that CHONG YAO EN (ID No: 21ACB00289 ) has completed this final year project entitled " E-WALLET USING NFC MOBILE DEVELOPMENT " under the supervision of MS TAN LYK YIN (Supervisor) from the Department of Computer and Communication, Faculty of Information and Communication Technology .

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

chong

CHONG YAO EN

# DECLARATION OF ORIGINALITY

I declare that this report entitled “**METHODOLOGY, CONCEPT AND DESIGN OF A 2-MICRON CMOS DIGITAL BASED TEACHING CHIP USING FULL-CUSTOM DESIGN STYLE**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : chong

Name : CHONG YAO EN

Date : 14 September 2023

# ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Ms Tan Lyk Yin who has given me this bright opportunity to engage in mobile application development. It is my first step to establishing a career in mobile application development and the NFC communication development field. A million thanks to you.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

# ABSTRACT

There are many payment methods that exist in this advanced technology era, so payment is no longer limited to cash payment, card payment and so on. It could be easy to be replaced by a technology payment method.  People also no longer require bringing along their payment card or cash to make any transaction since a new technology NFC can be utilised to develop this activity.

E-wallet is popular to be used to make payment among Malaysian, but there are still existing some weaknesses, such as payment processing time, authentication in terms of payment, and balance reload when urgent outside. These are the weaknesses going to be solved. In this proposed development, the NFC HCE module is important in further improving the e-wallet, in terms of payment duration, payment security, and fund transfer.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| *NFC* | Near Field Communication |
| *QR* | Quick Response |
| *POS* | Point-of-Sales |
| *PIN* | Personal Identification Number |
| *CVV* | Card Verification Value |
| *P2P* | Peer to Peer |
| *SE* | Secure Element |
| *HCE* | Host-based Card Emulation |
| *CPU* | Central Processing Unit |
| *AID* | Application ID |
| *SDLC* | Software Development Life Cycle |
| *RAD* | Rapid Application Development |
| *APK* | Android Package Kit |
| *PAN* | Personal Area Network |

# CHAPTER 1

# Introduction

## 1.1 Background Information

Traditionally, cash was the main form of payment, and it is offline payment. With the advancement of technology, payment methods are being enhanced to include contactless payments to make transactions more convenient for people, this can be proved in the [16], as the usage of non-cash payment has increased gradually over a consecutive year. Contactless payment provides a secure and convenient way to make payments without physically exchanging cash. This payment method involves the use of various technologies, including debit or credit cards, smart cards, Quick Response (QR) codes, as well as Near Field Communication (NFC) [17].

When it comes to making purchases using plastic cards, there is a method known as payWave that involves waving a debit or credit card over a terminal at the Point of Sale (POS). It is also available for online payment on an online shipping platform by providing their card information to checkout [15].

In terms of QR code payment, it is only widely used in mobile e-wallets (electronic wallets) [17]. Currently, the most widely used e-wallets in Malaysia are Touch ‘N Go, GrabPay, and so on. According to [20], there are steps to use the e-wallet to perform a transaction. Firstly, the users initiate the transaction by opening their preferred e-wallet, scanning the QR code displayed by the merchant, and then inputting the specified amount that must be paid to the merchant.

In [18], the existence of e-wallets is to cut down on the probability of fraudulent activity involving debit or credit cards whilst simultaneously providing users with more conveniences, such as cashless payment methods and eliminating the need to give a change to the payer as proven in [19].

## 1.2 Problem Statement and Motivation

The existing mechanism of contactless payment using an e-wallet, which requires a user to scan a merchant's QR code to carry out the contactless payment, is more than adequate for most customers. E-wallets offer a faster and more convenient way of making payments than traditional methods. However, there are some issues that users need to be aware of.

Firstly, while the scanning of the merchant's QR code through the system is faster, there is a possibility of an error in the QR code scanning. There are some problems stated in [26]. One general issue is the QR code is blurred printed, this can make the QR code reader decrypt the QR code information in confusion impressive scan. In the end, this kind of error requires the user to rescan the QR code, which leads the transaction processing to result in slow down.

Secondly, payment security is a concern when making payments through an e-wallet system. Before processing to payment transfer, the users need to enter their Personal Identification Number (PIN) to verify whether the current transaction is being conducted by the e-wallet owner. In detail, anyone nearby could potentially observe it and get the information, this allows them to steal the user's smartphone and then make any transactions.

Other than that, the existing development which is the Google Wallet can perform payment transactions quickly payment without unlocking the mobile screen [30]. This is not going to request any authentication then the payment can be done with or without the awareness of the user. So, anyone who gets a mobile phone with the installed Google Wallet, the person can make any payment with the device he or she gets.

Lastly, there is a similar issue to the problem mentioned above, which is reloading the e-wallet issue that poses a risk to privacy. In some cases, the users might need to perform an instant reload as they have an insufficient balance or must use other payment methods to complete the transaction. At this moment, the reload method requires users to enter sensitive information such as bank account ID and password, or credit card information that includes the Card Verification Value (CVV) code. The important information could be disclosed to people around the user and lead to shoulder surfing.

Therefore, the problems of the digital wallet can be summarised as follows:

- **Transaction Processing Time in E-wallet**
- **Security in Terms of Making Payment**
- **Privacy in Performing Reload when Instant Reload Is Needed**

The motivation is more focused on convenience, fast, and some of the e-wallet system's privacy aspects by emphasising the abovementioned issues.

## 1.3 Project Objectives

- **To develop a mobile application that utilises NFC card emulator technology**

  The primary goal of this project is to improve the speed at which payments are processed. Therefore, the main objective is to take the functionality of the NFC host-based card emulation technique to emulate a card and complete the transaction. This somehow reduces the time of the entire payment process as well as the errors to occur during the payment process.

- **To require fingerprint authentication before processing transaction**

  In current practice, there are a lot of authentication methods that can be utilised in various aspects. It includes one-time authentication, biometric authentication, and so on. As for now, most of the e-wallets mainly make use of PIN as a method of verifying the user. Fingerprint scanning as one of the biometric authentications will be involved in this project because this can take advantage to avoid authentication challenges. This includes static challenges such as shoulder surfing to get the PIN.

- **To allow a user to perform money transfers using NFC technology**

  In some situations, such as when the user is attempting to pay for public transportation with an NFC-based e-wallet, but their balance is insufficient, it may not be possible for the user to reload the balance of their e-wallet at this time. Alternatively, the user may be unable to ask others for assistance in making the payment. Hence, users can carry out fund transfers by using NFC from one person who is close to them. Consequently, this feature is helpful in the event of an emergency.

## 1.4 Project Scope and Direction

This project is an Android-based e-wallet mobile application and utilises the Java programming language to complete the whole development. At the end of this project, one of the goals is to create a mobile application that functions as an e-wallet to enable users to swiftly complete their transactions or fund transfers via NFC technology. The transactions may include payments for transportation, purchases, and other types of business.

Other than that, based on the weakness of the NFC card emulation mode to develop a further security layer to protect against the e-wallet payment system to avoid unwanted fraud in the system.

In addition, this application consists of two roles: the first is that of a client or consumer, and the second will be a merchant position. The user is given different functions to be used based on their role.

## 1.5 Contributions

Compared to those existing e-wallet applications in Malaysia, most of the applications that have been researched throughout the research period did not come with the NFC functionality to perform the transaction processing. The use of NFC technology could provide a more advanced and secure method to optimize the transaction. Hence, both the user and the merchant are required to use the smartphone device with NFC-enabled otherwise the development could not apply to them. However, this could not be an issue as most of the smartphone devices are NFC-enabled just not so popular to those users in Malaysia.

Firstly, this project mainly benefits the user’s side even though the system involves two parties: merchant and user. Importantly, this project will enable users to use NFC technology to speed up the process of the transaction or fund transfer. This is followed by enhancing more convenient ways to decrease the risk of revealing their information while making payment as well as urgently instant reload.

In terms of the merchant, the merchant is no longer required to hire a Point-of-Sales (POS) machine to accept the payWave payment method. This can be done by applying the use of merchant-side applications.

## 1.6 Report Organization

This report is divided into seven chapters, each of which will discuss different details and processes in terms of research and development.

The first chapter is to brief introduction to the evolution of payment methods. Based on the evolution, to come out some of the problems to be solved in this development directly point out that the solutions are the objectives of this proposed application.

Following the second chapter is the literature review which discusses the issues, strengths, and weaknesses of the existing applications and technologies.

Next, Chapter 3 will focus on the methodology while Chapter 4 introduces this project's system design.

Furthermore, Chapter 5 details the documentation about the system implementation, it can brief the reader on how to implement the proposed system from the beginning.

The second last chapter, which is Chapter 6, is critical in the system evaluation, it aims to test whether the proposed system archives the objective scope.

Lastly, chapter 7 concludes this project.

# Chapter 2

# Literature Review

## 2.1 Mobile Operating System

In current practice, there are a lot of mobile operating systems (OS) that exist. However, there are only three operating systems mainly dominant the market in worldwide.  It includes Android, iOS, and Harmony operating systems [33].

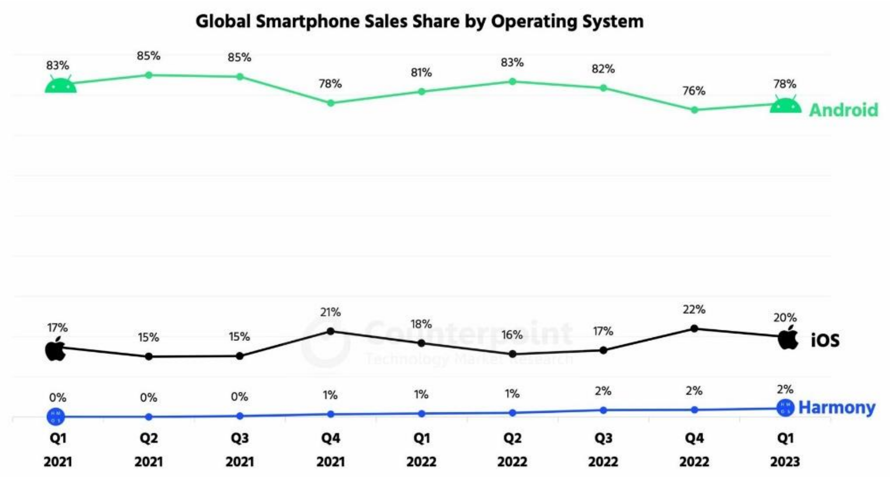


Figure 2.1.1.1 Global Smartphone Sales Share by Operating Systems [33]

The figure above shows how each mobile operating system dominates the mobile market. In the order of the dominant, the first occupant OS is Android, followed by iOS, and lastly Harmony OS from Huawei.

As analysis for quartet 1 of 2023:

**Table 2.1 The Occupancy of OS**

| Operating System | Occupancy (%) |
|---|---|
| Android | 78 |
| iOS | 20 |
| Harmony | 2 |

Most users intend to choose the Android and iOS mobile operating systems as their mobile phones, so the following sections will be discussed in detail.

### 2.1.1 Android

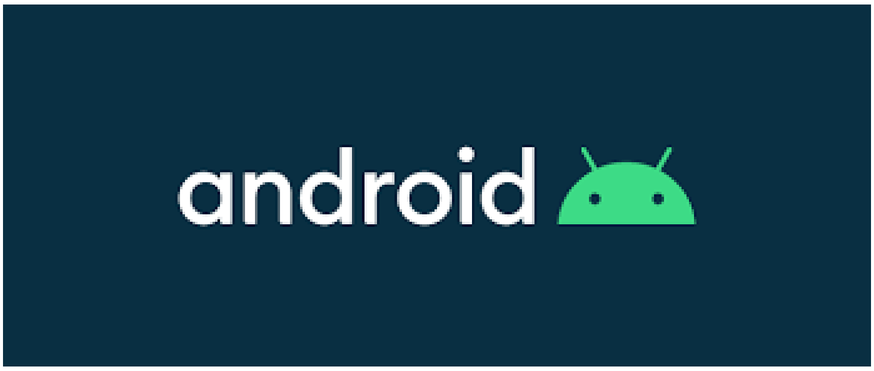


Figure 2.1.1.2 Android Logo

In [31], Android is an Operating System (OS) that is a platform developed by Android Inc in 2003. It is having a bottom layer based on a Linux kernel and other open-source software in the development of the framework. Other than that, it contains a huge of native libraries such as SQLite, OpenGL, and others on top of the Linux kernel.

Initially, Google primarily designed it for smartphone devices which a device that implements a touch screen as a kind of input from a user to interact with the OS [32]. After that, Android has also been developed for many platforms such as smart watches, television, tables, and cars.

Since there are quite a lot of native libraries and it is based on open-source as well it allows developers to develop applications as quickly as possible by using the native libraries.

There are some strengths and weaknesses summaries from [33] and [35].

**Strengths**

1. **Open-source Platform for Customization**

   Android is an open-source platform that allows developers to build a customization application as it is under Apache's license. However, this is impossible to do in iOS because the iOS application developer must strictly obey the rules set by the company when they need to code some specific development for a specific platform.

2. **Supports 3rd Party Widget**

   Android platform users are allowed to download and install 3rd party widgets and make them available on the home screen. This makes them quickly access the content or features of the corresponding application without launching the application into the foreground.

3. **Fast Improvement and Update**

   Android has a huge community of developers to develop the Android platform. Other than this, Android consists of a large number of users, it provides the opportunity that the user to give feedback about their experience on the system so that the development can be further improved as soon as possible.

4. **Supports Multiple Application Running Simultaneously**

   Android can perform multitasking. It allows users to execute multiple applications at the same time. Either running an application in the background or the user can split the screen to run the application with a certain Windows size.

## Weaknesses

### 1. Security on Customized Application

Since Android is open-source, any developer gets the chance to develop any application with malicious or hacking intentions. An Android user is allowed to easily install an application that they do not aware of. Eventually, the hacker gets into the device with their developed application.

### 2. A lot of Advertisements

Android can just simply publish to the Google Play store to their user so that the developer can post some advertisements on the application to gain extra revenue and cause the performance of the OS to slow down.

### 3. Lower Performance on Low-Specification Devices

Usually, Android OS is huge and must consume much storage on the mobile device. Hence, those lower specification devices with less storage or low hardware specifications will run slow the application. Or else, it totally cannot run well at all.

### 4. High Battery Consumption

Android allows multitasking so that a lot of processes can run in the background. This consumes a lot of battery power and decreases the usage time for the user.

The following table summarises the strengths and weaknesses of Android.

**Table 2.1.1.1 Strengths and Weaknesses of Android**

| Strengths | Weaknesses |
|---|---|
| Open-source Platform for Customization | Security on Customized Application |
| Supports 3rd Party Widgets | A lot of Advertisements |
| Fast Improvement and Update | Lower Performance on Low-Specification Devices |
| Supports Multiple Applications Running Simultaneously | High Battery Consumption |

### 2.1.2 iOS

Figure 2.1.2.1 iOS Logo

iOS was first developed by Apple. The OS of iOS is written in Objective-C language. As the second dominant in worldwide mobile device share, the reason that is chosen by most users is iOS are more focused on the user's privacy and security of the system.

Below are the strengths and weaknesses of iOS [36].

**Strengths**

1. **Strong Security**

   Based on security concerns, the iPhone strictly does not allow the user to install the application downloaded from third-party applications. Only applications from AppStore are allowed to be installed. Hence, this can be aimed to decrease the risk of virus attacks or data stealing.

2. **The App Store**

   Every application to be published into the AppStore must go through an investigation to check whether achieves the requirements or the rules set by the Apple company and then only be published to the AppStore.

3. **Hardware and Software**

   Apple is developing the integration tightly with Apple's hardware and software. This aim is to optimize the performance between hardware and software in iPhone devices so that their devices are not too dependent on powerful specs.

## Weaknesses

1. **High Cost**

   The price of the devices is quite high. In other words, their products are very expensive, so the consumers might not be able to afford their products. This is the reason why they cannot have high dominance in the mobile share market.

2. **Close-source Development**

   Unlike Android, iOS does not allow users to customize iOS themselves rather it provides limited customization to the user.

   Fully develop rights to those hired developers by Apple.

3. **Limited NFC HCE**

   NFC HCE is an advanced communication technology. It can be simply developed on Android. However, Apple does not provide the public SDK for iOS developers [37]. So, this project could not be fully developed on an iOS device.

The following table summarises the strengths and weaknesses of iOS.

**Table 2.1.2.1 Strengths and Weaknesses of iOS**

| Strengths | Weaknesses |
|---|---|
| Strong Security in terms of application | High cost of devices |
| App Store provides trusted Applications that allow users to install | Close-source development for programmers on active duty |
| Develop hardware and software tightly | Limited NFC HCE development |

### 2.1.3 Summary

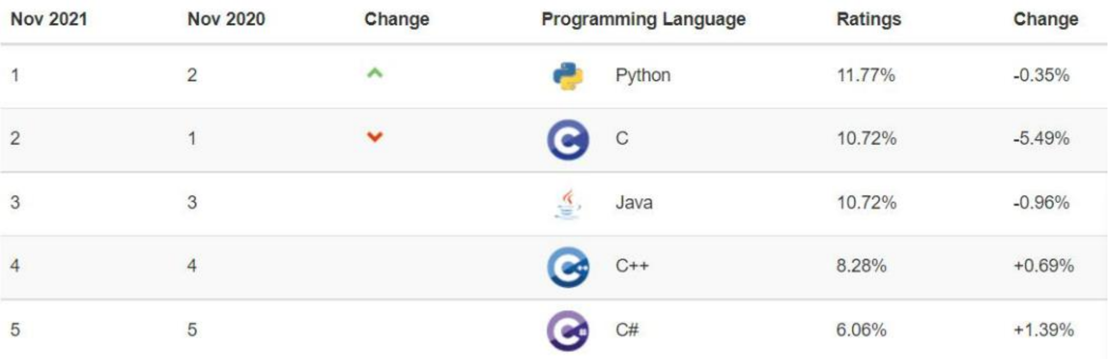
**Table 2.1.3.1 Comparison between Android and iOS [33] – [39]**

| | **Android** | **iOS** |
|---|---|---|
| **OS Platform** | Open-source and Linux-based | Proprietary by Apple |
| **Prior Programming Language** | • Java<br>• Kotlin | • Swift<br>• Objective-C |
| **Customization** | Yes | Limited Customization |
| **Price** | Depends on Specs | Generally high |
| **Application Install Package** | • Google Play Store<br>• Third party | • App Store |
| **App Store** | Provides various apps (either official or customization apps) | Provides quality and secure app |
| **System/Software Update** | Depends on manufacturers | Timely update |
| **Security** | Open source, cause more vulnerabilities | Main on security and privacy |
| **Access File System** | √ | X |
| **NFC HCE Customization Development** | √ | X |

The more focusing mobile operating system is Android, as this project aims to develop the NFC HCE technology to the system, and this feature is only allowed to develop in Android platform.

## 2.2 Mobile Application Programming Language

There are a lot of programming languages that can be used for mobile application development. For instance, Java, Objective C, and others [47]. The programming languages are used to be based on the platform on which the developers are going to develop the system.



| Nov 2021 | Nov 2020 | Change | Programming Language | Ratings | Change |
|---|---|---|---|---|---|
| 1 | 2 | ⌃ | Python | 11.77% | -0.35% |
| 2 | 1 | ⌄ | C | 10.72% | -5.49% |
| 3 | 3 | | Java | 10.72% | -0.96% |
| 4 | 4 | | C++ | 8.28% | +0.69% |
| 5 | 5 | | C# | 6.06% | +1.39% |

Figure 2.2.1.1 Mobile App Development Programming Language Rating [47]

The figure above shows that the top 5 programming languages were used to build mobile applications between 2020 and 2021 [47]. The popular programming language to be used is Python. It is because it is implemented pre-defined library, including Kivy and BeeWare. Other than that, it is capable of building cross-platform systems, including for iOS and Android [47].

Programming languages can be categorised into crossover or non-crossover-platform. For the crossover-platform example, the web application’s platform is HTML5, so the application is built through the browser [47].

The following section will discuss each of the crossover and non-crossover programming languages that are intended to be used in this development.

### 2.2.1 Java

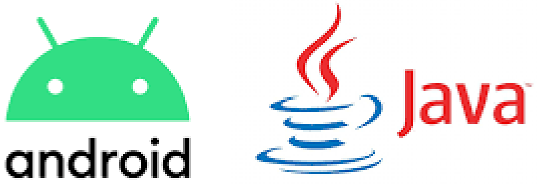


Figure 2.2.1.2 Android Java

According to [40], Java is an object-oriented programming language introduced in 1995. The use of Java in Android application development is ascribed to its ease of programming, compatibility, and extensive developer usage.

**<u>Strengths</u>**

1. **Platform-Independent**

   Java has the capability that write once and run anywhere. This is because Java is implemented using the Java Virtual Machine (JVM) while the Dalvik Virtual Machine (DVM) is the specialised virtual machine for Android. Unlike other programming languages, Java does not require code recompilation [40]-[41].

2. **Community Support**

   Java is widely utilized on a variety of devices. It has long been the preferred language among developers working on mobile application development. Hence, it grew a larger developer community than any other programming language around the world [41].

3. **Object-Oriented Programming (OOP) Language**

   OOP allows Java to be implemented easily. It can maintain large pieces of code by dividing them into several smaller modules. There are a lot of benefits to the use of OOP, including code reuse, encapsulation, and scalable [42].

## Weaknesses

1. **Memory Consumption**

   Java applications tend to consume a greater amount of memory compared to applications developed in languages like C or C++. This can pose challenges for devices with limited resources or situations where optimizing memory usage is of utmost importance.

2. **Verbose Coding**

   Java requires a lot of code to achieve even if it is a simple program. Therefore, Java developers have to remember those complex syntax, throughout this problem, it increased the complexity of development and slower the speed of development.

3. **Performance Overhead**

   Java runs on JVM, which introduces performance overhead compared to lower-level languages such as C/C++. This is because they need the interpreter to convert the source code into the machine language then will slow down the performance.

**Table 2.2.1.1 Strengths and Weaknesses of Java Android**

| Strengths | Weaknesses |
|---|---|
| Platform-Independent | Much Memory Consumption |
| Community of Developers Support | Verbose and Complex Coding |
| OOP-based Programming Language | Performance Overhead |

### 2.2.2 React Native

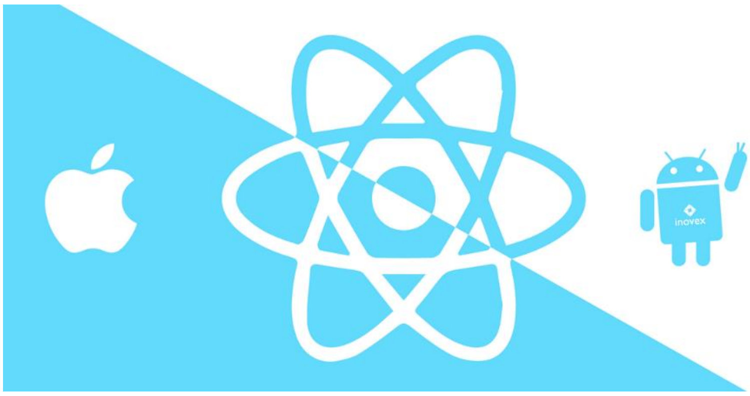


Figure 2.2.2.1 React Native

React Native is a JavaScript-based programming language. It aims to develop applications for mobile devices natively, such as iOS and Android [45]. It is mainly introduced to build mobile application user interfaces that utilise Facebook’s JavaScript library.

**Strengths**

1. **Crossover-Platform**

   One of the characterises of the React Native programming language, it allows React Native programmers to write source code once, and then apply it on multiple platforms. This makes developers save time and increase their efficiency in development. For example, compile React Native source then can apply to Android or iOS mobile devices.

2. **Code Reuse**

   React Native allows developers to reuse their codebase between different platforms, such as mobile, web, and others. Other than that, it allows for code reuse within the development to improve efficiency.

## Weaknesses

1. **Still Relatively Young Technology**

   React Native was only introduced to support iOS devices in March 2015 while Android was supported in September 2015. Both platforms are quite new to this programming language compared to Java or Switch programming languages but are quite familiar to developers.

2. **Unsupported Features**

   Since React Native are new to Android and iOS, some features on iOS still are not supported, they are still under discovery by developers. If the feature does not exist, the developer has to implement support for the API. This can be the module integration, such as Java source code integrated into the React Native module.

**Table 2.2.2.1 Strengths and Weaknes React Native**

| Strengths | Weaknesses |
|---|---|
| Crossover-Platform Development | Still Relatively Young Technology |
| Code Reuse and Efficient Development | Rely on third parties to support those unsupported features |

### 2.2.3 Summary

**Table 2.2.3.1 Programming Languages Comparison**

| | **Java (Android)** | **React Native** |
|---|---|---|
| **Language** | Java | Java Script |
| **Platform** | Android | iOS, Android |
| **Development Speed** | Longer development time | Faster development |
| **Third-party Libraries** | Huge libraries | Growing ecosystem |
| **Native API** | Full access | Limited access |
| **Community Support** | Yes | Limited |
| **Debugging** | Easier with tools | Easier for JavaScript developers |

In this project, the chosen programming language is Java, since the Java programming language is having full access to the native API library.

## 2.3 Electronic Payment Methods

### 2.3.1 Card Payment

The popular innovation of contactless payment was first seen by plastic cards (credit or debit cards). However, there is a drawback to this payment method in terms of the merchant (payer). This can be demonstrated in [4] which revealed that a retail clerk had stolen credit card information from 1300 customers during their transactions. This incident highlights the risk of information leakage during contactless transactions, despite the convenience of faster payment compared to traditional cash transactions.

**Weakness**

In terms of the payer, the payer passes their card to the retail clerk to complete their purchase, then the retail clerk gets the chance to record the card information and use their card without any authorization as the news mentioned above.

### 2.3.2 E-wallet

Based on the findings of a study in [5], e-wallets are not commonly utilised in Malaysia. Nevertheless, the adoption of e-wallets has significantly surged during the COVID-19 pandemic as the Malaysian government has been promoting their use and has also provided incentives for those who opt for this payment method.

People have begun to consider using e-wallets as a credible alternative to their preferred method of payment. This makes it less likely that the card payment weakness will happen, where the QR code payment method stands as a study case in this report.

#### 2.3.2.1 QR Code Payment

Generally, there is still a weakness for merchants (payee). The news reported in [21] that the criminals had stolen about 13 million dollars involving the fraudulent QR codes. This can be done by pasting criminals' QR codes to cover the original printed QR codes to illicitly obtain money.

Other than that, there is a possibility as a first problem statement that an error will occur with the QR code of a merchant while the customer scans the QR code. To solve this problem, the e-wallet business has offered a new solution that involves inverting the previous payment method and permitting the retailer to scan the QR code of the customer to complete the transaction. This can be done using the same style as one of the official websites for an e-wallet, which can be found in [8]. This website provides two ways to complete the payment, one of which has already been discussed, and the other will focus on dynamic QR codes.

**Strength**

- **One-Time Validity**

The user's QR code with a dynamic type of code, like a one-time password, and must be scanned by the merchant. Based on the research paper [9], the dynamic QR code has a few advantages, one of which is good security. Since a dynamic QR code is valid for a short time, it can no longer be used in the future, even if someone has obtained it. This is true even if no transactions took place before the code expired.

**Weakness**

- **Short Validity Period**

However, there is a restriction that was mentioned in the FAQs section of the Grab website [10]. Where it indicated that the consumer's QR code could not be scanned. This is because the consumer's e-wallet only supports dynamic QR codes with a 45-second validity period. Then, the customer tries to make the payment but fails since they somehow are unaware that it has expired. Eventually, the cashier must request the consumer to obtain a fresh dynamic QR code by using re-entering the interface to finish the transaction.

### 2.3.2.2 NFC Payment

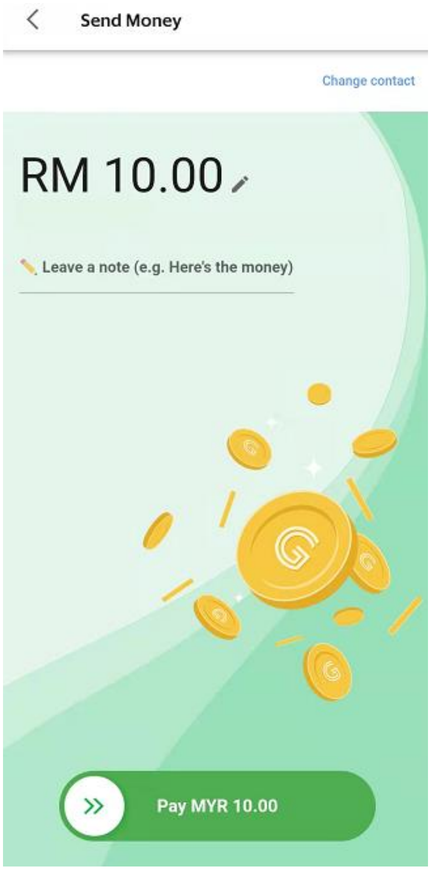
This payment method involved two objects which are NFC card reader or terminal, while another one will be NFC card or NFC-enabled device. The payment process is just finished by using one tap in short, then the transaction will be processed accordingly. The details will be discussed in NFC technology section.

## 2.4 Authentication on the e-Wallet Owner

### 2.4.1 PIN Authentication

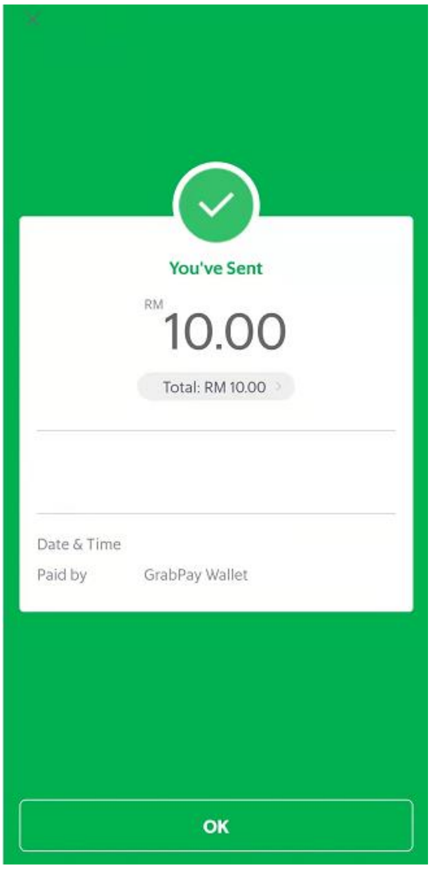
Based on the traditional method of contactless payment, the customer will simply have to wave their credit card to complete the transaction. If they lose their card and it is stolen by others, there is a possibility that it will become a credit card fraud. Then the person may use the stolen card to perform any transaction as claimed in [11]. However, there is an e-wallet which is involved in the same issue, which is named GrabPay.

The two figures below show that there is no request for any verification to confirm whether the current person is the e-wallet owner to perform the transaction. The user just needs to slide the slider after the amount has been input, as shown in Figure 2.4.1. After the slider is slid to the right, the system will switch the current scene into a successful interface as shown in Figure 2.4.2.



RM 10.00

Leave a note (e.g. Here's the money)



You've Sent

RM 10.00

Total: RM 10.00

Date & Time

Paid by GrabPay Wallet

Figure 2.4.1 GrabPay Transfer Interface

Figure 2.4.2 GrabPay Fund Transfer Done

In [12], a few potential resolutions to the problem described above were presented. To begin with, the verification problem is essential to improving the safety of the wallet, and most of the currently available wallets have integrated at least some of the identification protocols into their e-wallet systems, such as a password or a PIN. The reason for this is that the more security mechanisms that are implemented in the system, the more difficult it is to become a victim of theft.

Verification adds a layer to protect funds and prevents thieves from using a victim's e-wallet without authentication. The burglar can only guess the PIN by combining numbers if they steal the smartphone.

**Weaknesses**

- **Brute-Force Attack**

The thief may even unintentionally guess the PIN. Since the PIN is based on *k* digits specified by the system, the thief can only calculate the number combination $10^k$ times to get the only one correct PIN like using a brute force attack. However, there is a restriction in the current e-wallet, which only allows a few attempts, or else the e-wallet will be suspended.

- **Shoulder-Surfing Attack**

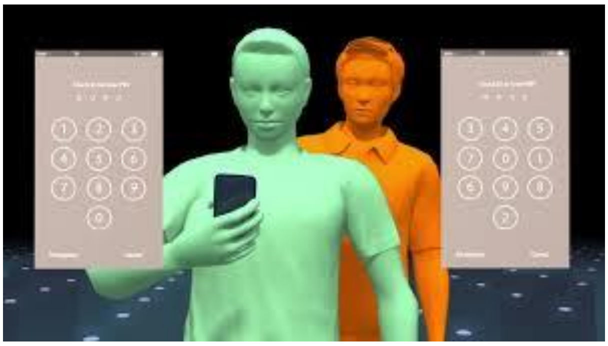


Figure 2.4.3 Shoulder-Surfing Attack

However, there is still a problem that can occur if a person who has the intention of obtaining the PIN of the e-wallet owner can engage in shoulder-surfing and so on. If the thief was eventually able to get the PIN and then stole the user's smartphone, he or she could easily use the e-wallet balance to pay for anything.

**Recommendation**

### 2.4.2 Biometrics Authentication

In [12], it was mentioned that an effective technique to increase the security of e-wallets is to include biometric authentication in the e-wallet systems. This was done as an enhancer to tackle the problem that was described above. To discuss one of the mechanisms, if the user's fingerprint is found to match any record in the smartphone, for instance, the system will grant the user permission to carry out the transaction. If not, the user will not be granted permission to perform any more transactions.

On the other hand, the function of the fingerprint scanner is not implemented in the current e-wallet systems. If PIN authentication is going to be replaced by fingerprint authentication, the security will be increased by a protected level of the e-wallet systems. Even if the burglar stole the device, he would not be granted any permission to use any balance as his fingerprint did not match any record.

According to the advantages of the paragraph above, biometric authentication will be implemented into this proposed development in terms of a payment system.

## 2.5 NFC Technology

Figure 2.5.1 NFC Logo

In terms of NFC, it is a well-known technology that can exchange data over a short distance of roughly 4 cm almost the same as contactless payment, as has been claimed in [1]. In [2], NFC is a technology that requires two devices that support NFC to communicate or exchange data with one another although the data rate that can be transmitted with NFC is limited to 424 kbit/s at most.

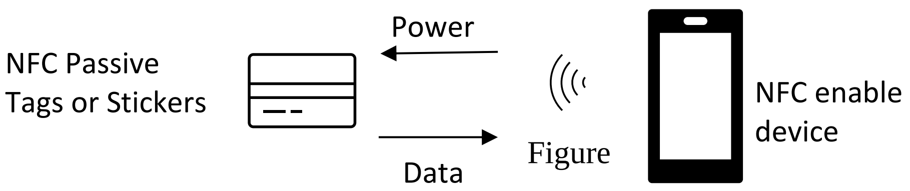
In addition to the functions of NFC described above, NFC can emulate a credit or debit card to process transactions immediately [3].

Since NFC technology already exists, there is no reason not to implement it onto e-wallets to enhance the same capabilities as a payWave payment mechanism.

In [2], a supported NFC device is capable of implementing three main modes of operation:

### 2.5.1 Reader/Writer Mode

This mode enables data transfer for the NFC-enabled smartphone application. In other words, it also enables passive NFC tags or stickers to be read or written by an NFC-enabled device. The NFC reader will read the data from the NFC tags or stickers when it is in reader mode. In writer mode, the NFC-enabled device will write some data onto the tags or stickers.



2.5.2 NFC Read/Write Mode

### 2.5.2 Peer-to-Peer Mode (P2P Mode)

P2P mode is carried out by two NFC-enabled active devices able to communicate with one another to share data. Messages, documents, and other types of data can be exchanged between two NFC-enabled devices. Android Beam is an example of an application that adopts this mechanism.

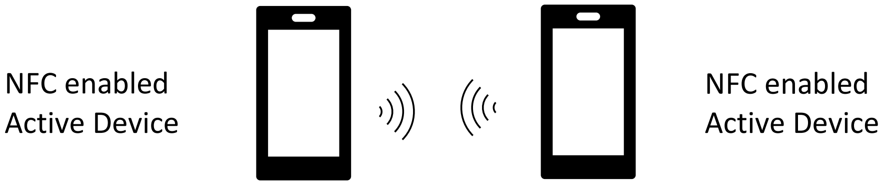


Figure 2.5.3 NFC P2P Mode

### 2.5.3 Card Emulation Mode (CE Mode)

An NFC-enabled device also supports emulating a card or a tag as a card emulation mode. Before adapting to this technology, people will have to carry a lot of cards. After the card emulation is released, all the cards such as payment cards, transportation cards, door cards, and all other cards can be stored in a wallet. This mechanism converts the physical plastic card into a virtual card, then people are only having their phone with the virtual card holder instead of carrying a lot of cards.

There are two main methods to carry out the card emulation as follows:

Figure 2.5.4 Card Emulation Mode

### (a) Secure Element (SE)

For card emulation utilising SE, hardware that represents a chip inside a smartphone is needed. When one of the applications intends to employ a simulated contactless card to make a payment, the application must first store the card information in the SE before proceeding with the transaction. The SE must be close to the NFC reader, then both devices can only proceed to the next process. Then, the NFC controller inside the smartphone will instruct the NFC reader to perform communication with the SE. The application will not be involved in the communication, but it will eventually get a notification from SE when the transaction is finished.

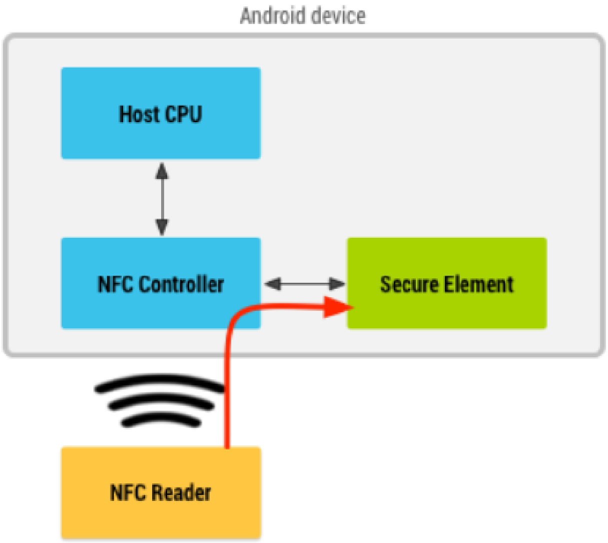


Figure 2.5.5 NFC Communicate with Secure Element

### (b) Host-based Card Emulation (HCE)

HCE is a method that may be used in place of SE to simulate a contactless card or tag. The HCE immediately allowed the NFC reader to interact with the host Central Processing Unit (CPU), which handled the procedure, whereas SE emulation used the SE chip to execute the communication.

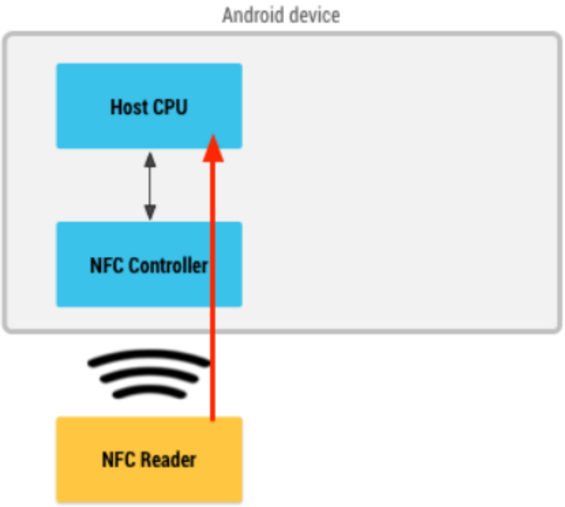


Figure 2.5.6 NFC Communicate with Host CPU

**<u>Strengths</u> [43]**

1. **Virtualize Entity Card**

   As the capability of the NFC HCE, it can emulate the card with certain information to integrate them into the application. This helps to store all the card information in the application instead of carrying all the physical cards.

2. **Flexible / Scalable**

   NFC HCE developers have more flexibility in implementing mobile payment applications or systems. They are allowed to build their payment applications or integrate other NFC-based services into the mobile applications. This provides a customized user experience designed application.

**3. Eliminates the Need for Network Connectivity**

NFC HCE allows the user to perform the payment without a network connection, this was done by the Google Wallet [14]. The details will be discussed in the following section.

**Weaknesses [2], [46]**

**1. HCE Runs Transparently in the Background**

NFC HCE comes with an architecture in Android which allows the HCE to run in the background without any awareness of the user or the help of the user interface. This may provide convenience to the user that does not require to launch the application to initialize the NFC HCE but somehow the user will be unaware that their application will execute the transaction that had been done by others as long as the NFC module is running mode.

**2. Limited Range**

Any mode of the NFC, it has a relatively maximum range of communication capability, it is typically around 4 cm. This can be considered as shortest-range communication compared to Personal Area Network (PAN) devices communication.

**3. Security Concern**

Even though NFC can be only done in a short range, it is still possible to encounter potential risks such as eavesdropping, data corruption, interception attacks and others.

**Table 2.5.1 Strengths and Weaknesses of NFC HCE**

| Strengths | Weaknesses |
|---|---|
| HCE virtualize physical card | HCE runs in the background |
| Scalable HCE mobile application | Limited communication range |
| Eliminate the need for network connectivity | Security concern |

### 2.5.4 Communication Process of NFC Card Emulation

Since the card emulation will be utilised in this project, the communication process must be researched in depth.

According to [6], the Application ID (AID) must be matched to determine whether the application was appropriately selected on the user's side before executing the request. The AID represents the name of the application that intends to emulate a card. After the AID is found based on the NFC reader request, the host CPU will pass the request to the corresponding application to send back the response data to the NFC reader. Only the "OK" command is stated in the received response data, and then both the NFC reader and NFC card emulator can continue to do the data exchange between them. Otherwise, the request will fail.

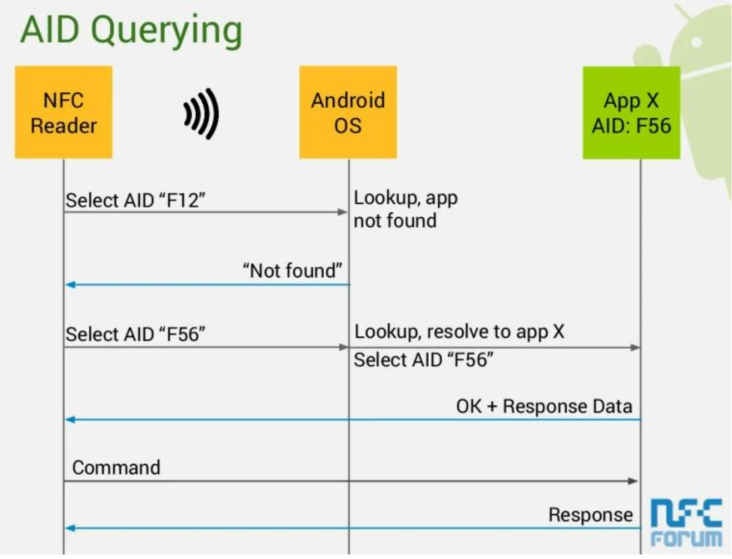


Figure 2.5.7 NFC Responding Process

### 2.5.5 Summary

**Table 2.5.2 Summary of NFC Technology**

| **Mode** | **Definition** | |
|---|---|---|
| **Reader / Writer** | Communication Objects | • NFC passive tag<br>• NFC card<br>• NFC sticker |
| | Features | The NFC-enabled device acts as a reader(/writer) to read(/write) from(/into) the destination object. |
| **P2P** | Communication Objects | • NFC-enabled device |
| | Features | This mode allows the exchange of data, including documents and messages, between two NFC-enabled devices. |
| **CE** | Communication Objects | • NFC-enabled device<br>• NFC card reader |
| | Features | The NFC-enabled device emulates a card with sufficient information to be exchanged. This allows the device to act as a virtual card that can be read by an NFC card reader.<br>• **Secure Element Mode (SE Mode):** This mode utilizes a secure element chip to store the information for card emulation. The secure element chip acts as a third-party intermediary between the NFC-enabled device and the NFC reader.<br>• **Host-based Card Emulation: (HCE Mode):** In this mode, the NFC reader directly communicates with the host application on the NFC-enabled device. This eliminates the need for a separate secure element chip. |

## 2.6 Existing Systems / Mobile Applications

### 2.6.1 Google Pay Wallet

Figure 2.6.1.1 Google Pay Wallet Logo

According to the author of [13], Google is allegedly working on developing the Google Wallet with an NFC function so that it can be used to make payments. MasterCard, Visa, and a variety of other payment services are among those that can be used to add Google Wallet can be used to make payments in traditional brick-and-mortar stores that support Google Wallet payment. If the user saves their card information within Google Wallet, it can perform the functions of a convenient wallet, freeing them from the need to always carry their physical card around with them.

To be able to make any kind of payment with Google Wallet, such as transportation or in-store payment, the user must initially proceed to tap their phone against an NFC card reader to finish the transaction.

At this time, the card information will be stored in SE by Google Wallet to let the NFC controller route the NFC reader to communicate with SE. After that, the payment credentials will be transferred to the NFC card reader. At the same point, the retailer will acquire the confirmation on the POS, which will print the receipt. In the meantime, the Google Wallet application on the user's side will prompt a notification to notify the user that the transaction in question has been completed.

According to the authors of [14], NFC payments may only be made offline, which means that they must be carried out in person. This implies that NFC payments must be conducted physically. This action will involve three parties, the payer, the payee, and the payment service provider.

Google Pay has implemented a quick response in payment without any further action if the NFC module is being turned on and near the NFC reader. This is a conclusion for [22], and this fast payment is only available for a smaller amount which is up to RM250 in Malaysia [23]. Noteworthily, this quick payment also can be done without the screen on. In other words, even when your mobile phone’s screen is off, if your NFC module is working near the NFC reader, the payment still can be made.

With enhanced protection by Google has been written in [23], they added an optional layer that lets the user set up how many times locked transactions are and whether to unlock the device to complete the payment.

In details of the payment process, after the payment application is running on the user (payee) side, the application will request a token from the service provider which is the server. Then, the payer taps against the POS, the token will be transmitted to the POS, and the POS will transfer the payment info and the token to the server to complete the transaction. In the token, it contains some information, such as user ID, secret key, and others.

The illustration of the payment process is shown in Figure 2.6.1.2.

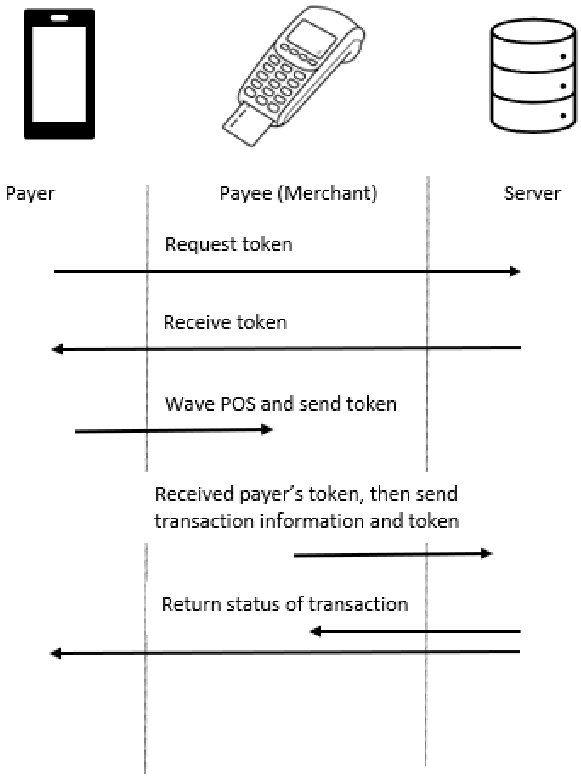

Payer
Payee (Merchant)
Server
Request token
Receive token
Wave POS and send token
Received payer's token, then send transaction information and token
Return status of transaction

Figure 2.6.1.2 Google Wallet Payment Process

### 2.6.2 Touch ‘n Go e-Wallet

Figure 2.6.2.1 Touch ‘n Go e-Wallet Logo

Touch n’ Go e-wallet generally supports those functions that are included inside other e-wallets, including fund transfer, utility bill payment, and other services [48]. Different from other e-wallets, the Touch n’ Go e-wallet does include the merchant account service inside the system. Other than this, the system allows the merchant to scan the user’s dynamic QR code with their valid merchant account.

Since the COVID-19 epidemic quickly spread over the world in 2019, this created the chance for the world to implement a new transformation and renovation for financial technology that shifted physical payment to digital payment [49]. The Malaysian government did implement a stimulus programme, that aims to speed up the usage of the e-wallet cashless payment in Malaysia.

### 2.6.3 Comparison between Existing Systems and Proposed Development

The following table is based on the system experience to compare the systems.

**Table 2.6.3.1 Systems Comparison**

| **Features** | **Google Wallet (Google Pay)** | **Touch n' Go e-Wallet** | **Propose System** |
|---|---|---|---|
| Merchant Applicable | No. Only use terminal POS to receive payment | Yes. A merchant account is required | Yes. Require the use in Merchant side application |
| Payment using HCE | Yes | No | Yes |
| Authentication for Payment Process | No. Given that the NFC module is on | Yes. PIN / Facial Scanning | Yes. Fingerprint / PIN |
| Offline Transaction | Yes | No | Yes |
| Payment History | No | Yes | Yes |

Throughout the systems, there are two strengths in the proposed system among the systems:

- To perform fund transfers between two users by using NFC if urgent
- Secure protection and convenience in terms of payment.

# CHAPTER 3

# Proposed Method/Approach

## 3.1 Methodology

Based on [27], it is necessary to implement a system development methodology while developing an information system. A mindset that the Software Development Life Cycle (SDLC) then comes in. It is a framework which defines the tasks that should be done in each of the development processes. It could help in achieving the project objectives, eventually fulfilling the requirements.

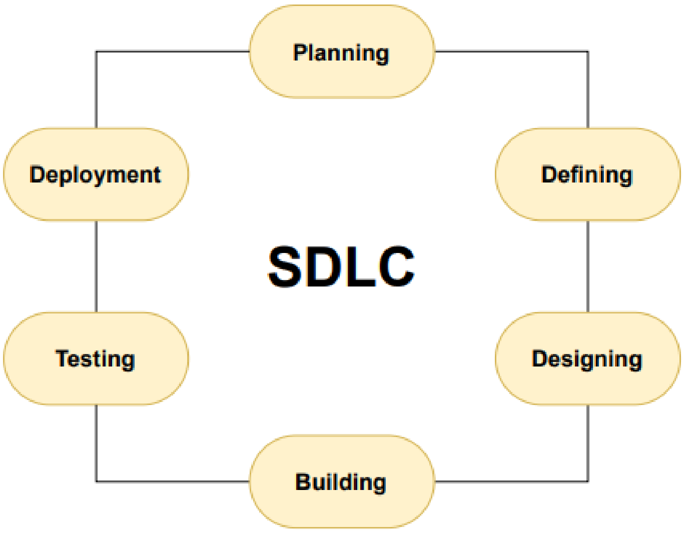


Figure 3.1.1 System Development Life Cycle

Generally, these are the phases as shown in Figure 3.1.1 that will be involved in most of the software development life cycle model. Those models always surround the phases in SDLC to define their phases and then be adopted in software and system development. There are several models in SDLC including the Waterfall model, System Development Cycle, and others. Then, the two stated models which may be appropriate to this project will be briefly discussed in the following section.

### 3.1.1 Waterfall Model

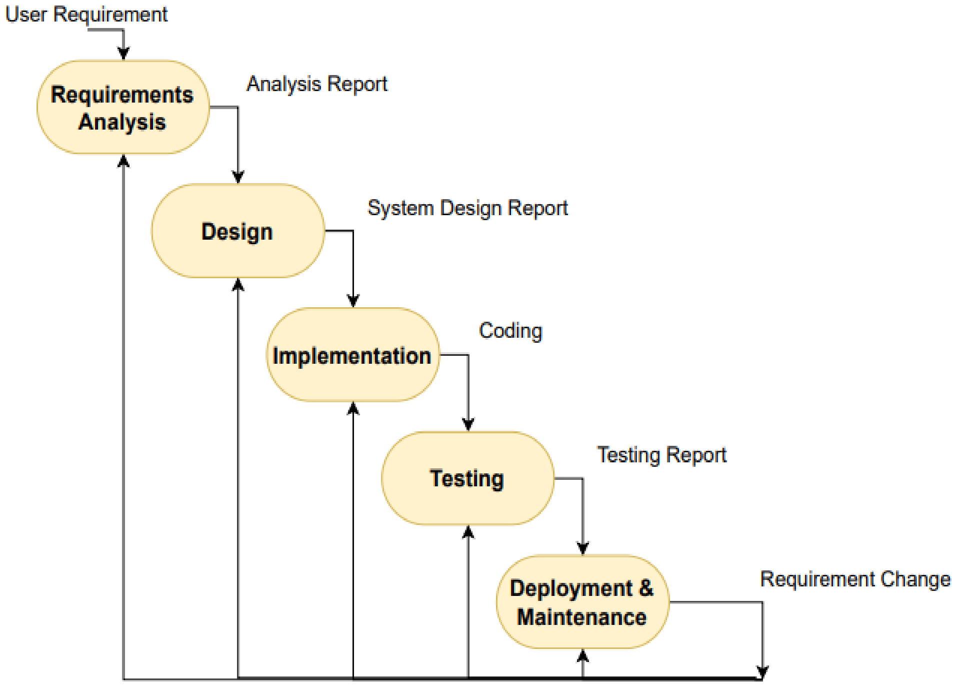


Figure 3.1.2 Waterfall Model

The waterfall model is classified as a linear sequential type of model, illustrated in Figure 3.2. The waterfall model was the earliest development model, which continues to play an essential role in software development by providing the basis for software development.

The procedure is to accept input from the previous activity for the work object of this current activity. It is important to have the necessary input from the previous activity. Somehow the input includes the required information and relevant details to carry out the corresponding task.

At the same time, assess the activity's implementation; If successful, move on to the next action; otherwise, return to the previous task, or even earlier task if necessary.

However, there are a lot of disadvantages stated in [29]. One of them is the project scope cannot be changed or any modification after the product is output. In conclusion for this model, this model is worthless for projects that wish to be flexible.

### 3.1.2 Rapid Application Development (RAD) Model

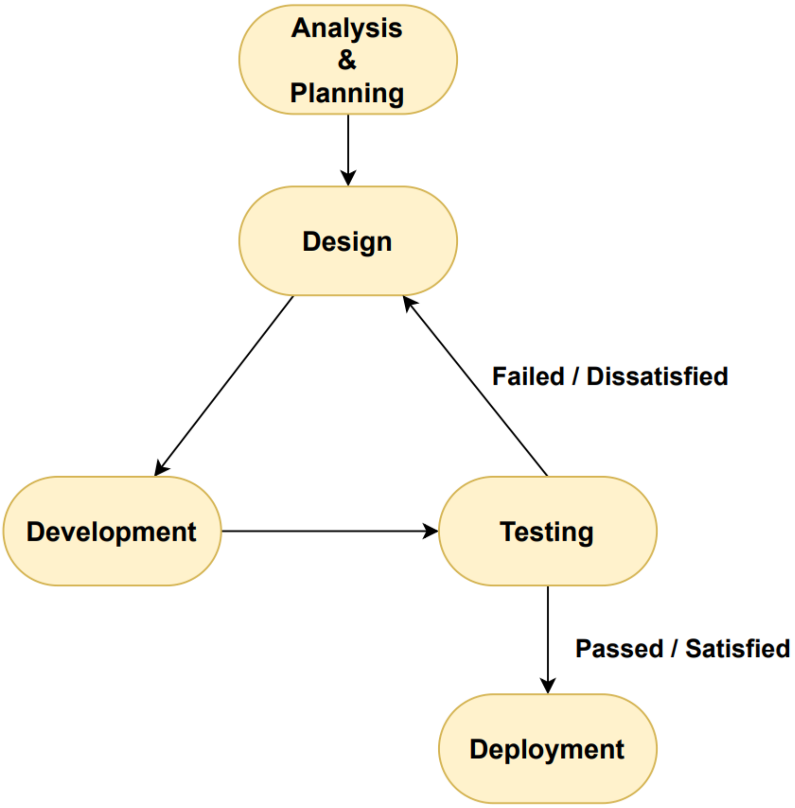


Figure 3.1.3 Rapid Application Development (RAD) Model

The focus of the RAD model is to gather user requirements and reuse the prototypes to be further developed and then apply them to the product. This model is based on prototyping and iterative development.

A prototype is a workable model that is functional and equivalent to part of the released product. There is no minimal or specific planning involved, which can enable the developer to handle the changes in the development process well and favour product delivery as soon as possible.

One of the benefits of adopting this model is the development that can be modularized feasibly.

### 3.1.3 Selected Model [29]

Among those suitable models that have been discussed, the RAD model is chosen to develop the system in this project. The main reason to choose the RAD model is not only because the RAD model is acceptable to deliver the system part by part, but also suitable for short development duration.

According to Figure 3.3, the system development of this project will be subdivided into several phases. There are phases of analysis and planning, design, development, testing, and implementation.

- **Analysis and Planning Phase**

During this phase, it is required to do a system investigation before continuing. It is essential to analyse what the problem is that already exists in the current system and to determine whether the existing problem affects the users. After that, a solution should be proposed that would work to fix the problem if the problem affected users. Next, the project scope is planned carefully to prevent development from progressing out of scope.

- **Design Phase**

The user interface should take precedence in this system, and it should be designed with the “ease of use” principle in mind as part of the user experiment design (UXD). This is done to guarantee that the user interface of the system is intuitive and is having the ease to use for the end user. It is also important to think about the development environment, the programming language, and the service to build the system.

- **Development Phase**

This stage focuses mainly on the development of new components or the fixing of problematic modules that already exist. In addition, always make sure that the development is within the intended scope.

- **Testing Phase**

Each component will be evaluated after the system development is completed. This step's goal is to confirm whether the component satisfies the scope's requirements. Otherwise, it will be necessary to go back to the design phase if any unexpected results are produced. The cycle will end until the output satisfies the criteria.

- **Implementation Phase**

Although each module has been finished, the modules still need to be combined to construct a new system. Implementing the system is the last step.

## 3.2 System Requirements

To implement the proposed system, there are some requirements for the system device. The specifications for the device are as follows:

**Table 3.2.1 Minimum Device Specifications Requirements**

| Description | Specifications |
|---|---|
| **Platform** | Android |
| **Operating System Version** | Android 4.4 |
| **NFC HCE Module** | Supported |

### 3.2.1 Hardware Components

There are three hardware components involved in this project development. The laptop is utilised for source code development and user interface design. The two Android devices are applied for testing and deploying purposes.

Figure 3.2.1.1 Laptop

**Table 3.2.1.1 Specifications of Laptop**

| Description | Specifications |
|---|---|
| **Model** | Acer Aspire A515-56 |
| **Processor** | Intel Core i5-1135G7 |
| **Operating System** | Windows 11 |
| **Memory** | 8GB DDR4 RAM |
| **Storage** | 512 NVMe SSD |

Figure 3.2.1.2 The First Mobile Device (MI 9)

**Table 3.2.1.2 Specifications of the First Mobile Device**

| **Description** | **Specifications** |
|---|---|
| **Model** | Mi 9 |
| **Processor** | Snapdragon 855 |
| **Operating System** | Android 10 |
| **Memory** | 6GB |
| **Storage** | 128GB |
| **NFC Connectivity** | Yes |

Figure 3.2.1.3 The Second Mobile Device (MI 13)

**Table 3.2.1.3 Specifications of Second Mobile Device**

| Description | Specifications |
|---|---|
| **Model** | Mi 13 |
| **Processor** | Snapdragon 8 Gen 2 |
| **Operating System** | Android 13 |
| **Memory** | 12GB |
| **Storage** | 256GB |
| **NFC Connectivity** | Yes |

### 3.2.2 Software Components

**Table 3.2.2.1 Software Involvement**

| Tools | Specification |
|---|---|
| Programming Language | 1. Java<br>2. XML |
| Application Platform | 1. Android Studio<br>2. Firebase Database |

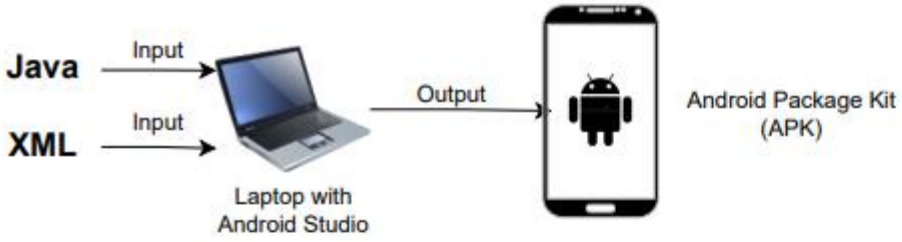


Figure 3.2.2.1 Diagram of Producing APK

The APK will be produced by utilising the programming languages which are Java and XML. Both the source codes will be compiled by the Android Studio compiler, and eventually output the install package.

### 3.2.2.1 Android Studio

Figure 3.2.2.2 Android Studio

Android Studio is a platform application that is an Integrated Development Environment (IDE) specifically for Android application development. It provides huge and plenty of native built-in tools that help developers develop the whole application efficiently and quickly. Importantly, Android Studio comes with a built-in Android OS emulator to make developers debug programs as quickly as possible.

### 3.2.2.2 Google Firebase

Figure 3.2.2.3 Google Firebase

Google Firebase provides a wide range of tools and services that increase the speed of programmers to develop and manage their applications more efficiently and quickly. Firebase provides a unified platform for backend services, such as user authentication, database, et al.

## 3.3 Functional Requirement

**Table 3.3.1 Function Requirement for both Merchant and Client Roles**

| Functions | Particular |
|---|---|
| User Login | A function that allows the user to log in to the mobile application. |
| NFC Login | A function that allows the user to log in using NFC technology with one tap. |
| User Registration | A function that allows the user to register. |
| Password Recovery | A function that allows the user to reset the login password. |
| Reload/Withdrawal | A function that allows the user to balance reload/withdraw. |
| Transactions Searching | A function that allows the user to view or filter transaction histories by specified duration. |
| Change Language | A function that allows the user to select the language to be applied in the application. |
| Log Out | A function that allows the user to log out from the system. |
| PIN Registration | A function that allows the client to register their payment PIN if they do not have any PIN in the database. |
| PIN Modification | A function that allows the client to modify their payment PIN. |
| Authentication | A function that allows the client to authenticate themselves using biometric or PIN authentication. |

Table 3.3.2 Function Requirement for Merchant Role

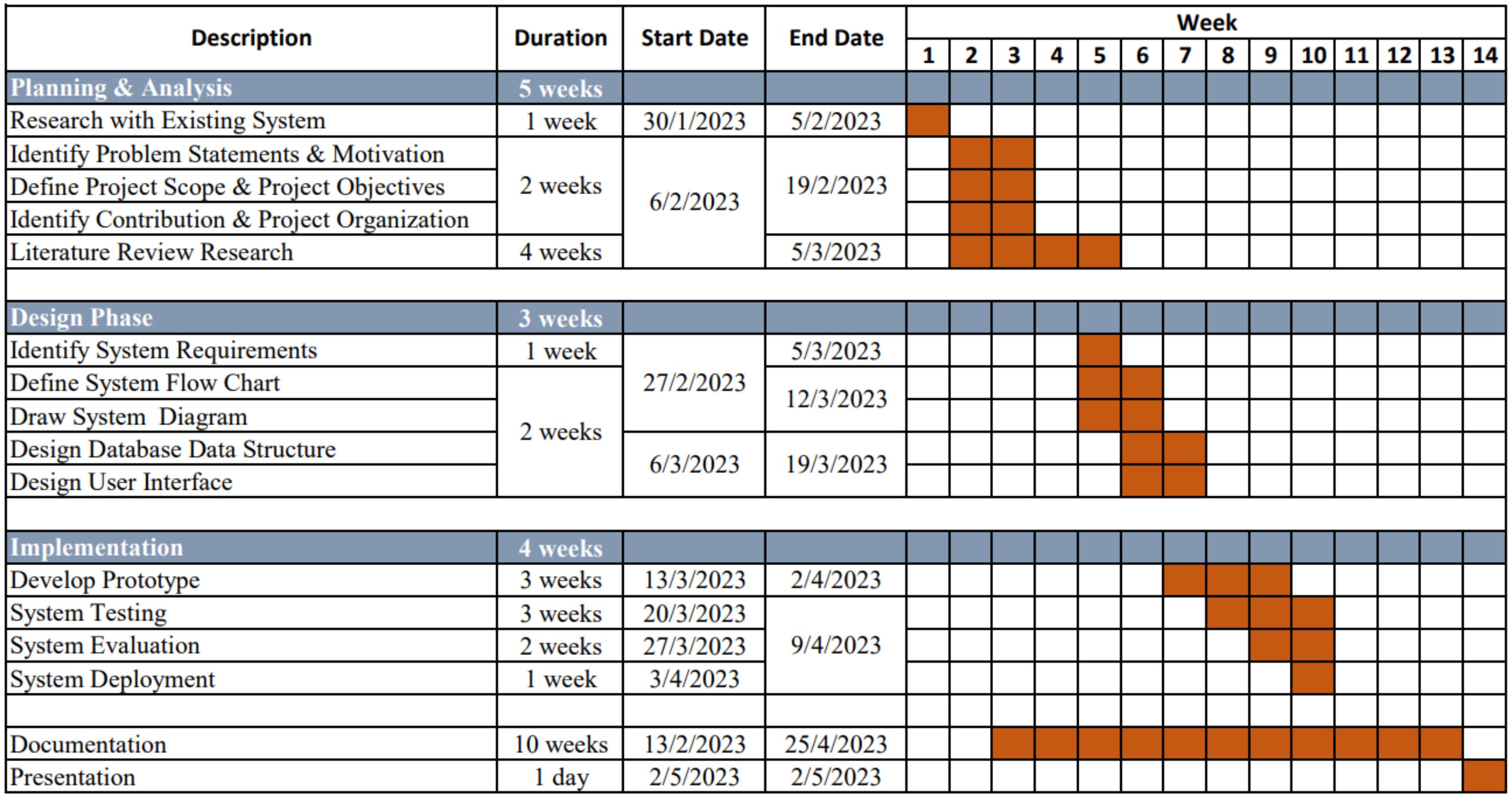
| Functions | Particular |
|---|---|
| Receiving Amount Input | A function that allows the merchant to key in the amount that intend to receive from their consumer or client. |
| Amount Short Cut Key Function | A function that allows the merchant to select the amount button instead of keying in the amount. |
| NFC Payment Receive | A function that allows the merchant to receive the amount through NFC technology. |
| Amount Key Customization | A function that allows the merchant to customize the amount shortcut key buttons. |

Table 3.3.3 Function Requirement for Client Role

| Functions | Particular |
|---|---|
| NFC Payment | A function that allows the client to make a payment using NFC technology. |
| NFC Funds Transfer | A function that allows the client to perform fund transfers to another client using NFC technology. |

## 3.4 Project Milestone

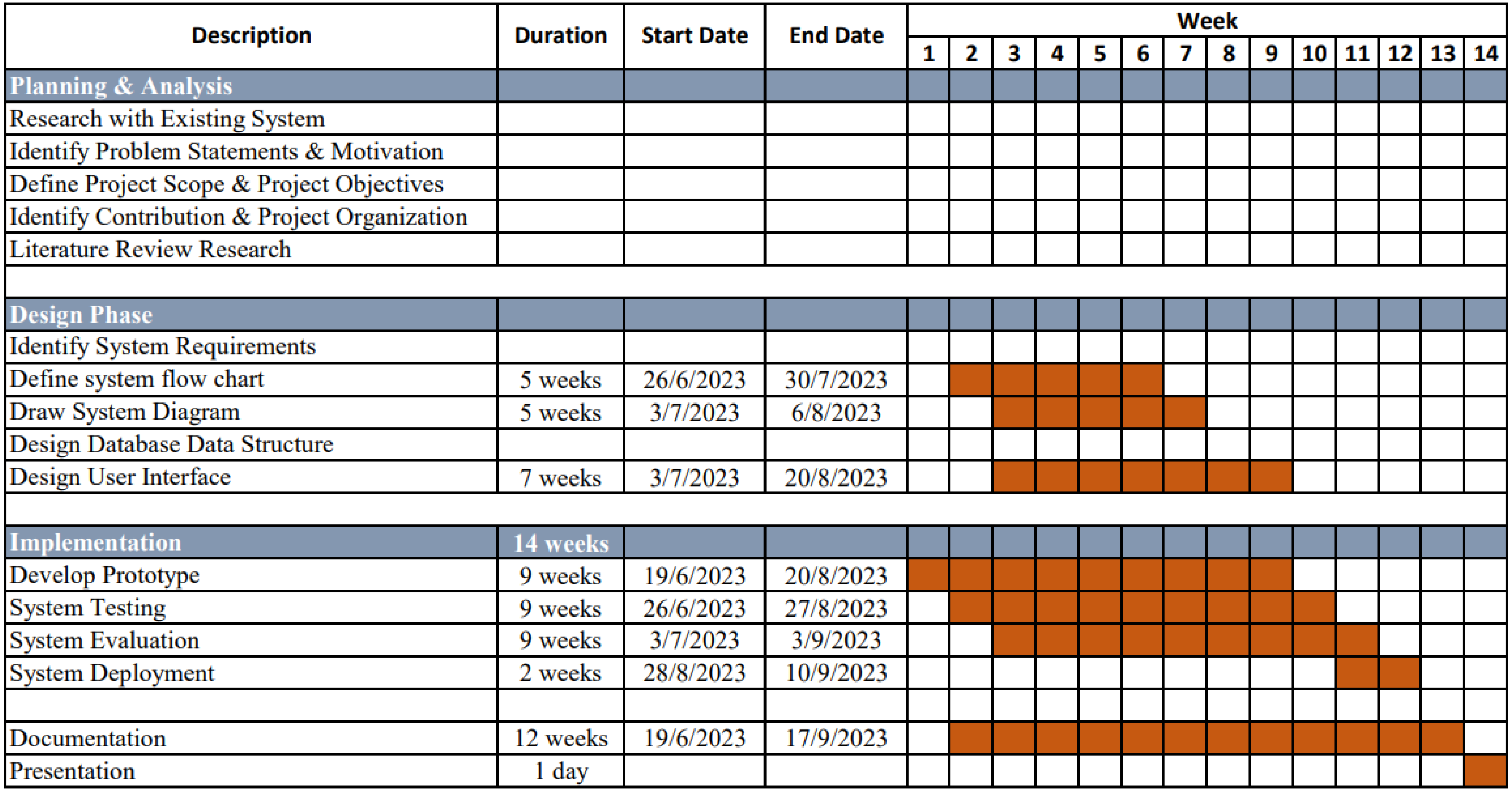
### 3.4.1 Project I Timeline



| Description | Duration | Start Date | End Date | Week | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| **Planning & Analysis** | **5 weeks** | | | | | | | | | | | | | | | | |
| Research with Existing System | 1 week | 30/1/2023 | 5/2/2023 | 🟧 | | | | | | | | | | | | | |
| Identify Problem Statements & Motivation | 2 weeks | 6/2/2023 | 19/2/2023 | | 🟧 | 🟧 | | | | | | | | | | | |
| Define Project Scope & Project Objectives | | | | | 🟧 | 🟧 | | | | | | | | | | | |
| Identify Contribution & Project Organization | | | | | 🟧 | 🟧 | | | | | | | | | | | |
| Literature Review Research | 4 weeks | | 5/3/2023 | | 🟧 | 🟧 | 🟧 | 🟧 | | | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| **Design Phase** | **3 weeks** | | | | | | | | | | | | | | | | |
| Identify System Requirements | 1 week | 27/2/2023 | 5/3/2023 | | | | | 🟧 | | | | | | | | | |
| Define System Flow Chart | 2 weeks | | 12/3/2023 | | | | | 🟧 | 🟧 | | | | | | | | |
| Draw System  Diagram | | | | | | | | 🟧 | 🟧 | | | | | | | | |
| Design Database Data Structure | | 6/3/2023 | 19/3/2023 | | | | | | 🟧 | 🟧 | | | | | | | |
| Design User Interface | | | | | | | | | 🟧 | 🟧 | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| **Implementation** | **4 weeks** | | | | | | | | | | | | | | | | |
| Develop Prototype | 3 weeks | 13/3/2023 | 2/4/2023 | | | | | | | 🟧 | 🟧 | 🟧 | | | | | |
| System Testing | 3 weeks | 20/3/2023 | 9/4/2023 | | | | | | | | 🟧 | 🟧 | 🟧 | | | | |
| System Evaluation | 2 weeks | 27/3/2023 | | | | | | | | | | 🟧 | 🟧 | | | | |
| System Deployment | 1 week | 3/4/2023 | | | | | | | | | | | 🟧 | | | | |
| | | | | | | | | | | | | | | | | | |
| Documentation | 10 weeks | 13/2/2023 | 25/4/2023 | | | 🟧 | 🟧 | 🟧 | 🟧 | 🟧 | 🟧 | 🟧 | 🟧 | 🟧 | 🟧 | 🟧 | |
| Presentation | 1 day | 2/5/2023 | 2/5/2023 | | | | | | | | | | | | | | 🟧 |

Figure 3.4.1.1 Project I Timeline

### 3.4.2 Project II Timeline



| Description | Duration | Start Date | End Date | Week | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| **Planning & Analysis** | | | | | | | | | | | | | | | | | |
| Research with Existing System | | | | | | | | | | | | | | | | | |
| Identify Problem Statements & Motivation | | | | | | | | | | | | | | | | | |
| Define Project Scope & Project Objectives | | | | | | | | | | | | | | | | | |
| Identify Contribution & Project Organization | | | | | | | | | | | | | | | | | |
| Literature Review Research | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| **Design Phase** | | | | | | | | | | | | | | | | | |
| Identify System Requirements | | | | | | | | | | | | | | | | | |
| Define system flow chart | 5 weeks | 26/6/2023 | 30/7/2023 | | ■ | ■ | ■ | ■ | ■ | | | | | | | | |
| Draw System Diagram | 5 weeks | 3/7/2023 | 6/8/2023 | | | ■ | ■ | ■ | ■ | ■ | | | | | | | |
| Design Database Data Structure | | | | | | | | | | | | | | | | | |
| Design User Interface | 7 weeks | 3/7/2023 | 20/8/2023 | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | | | | |
| | | | | | | | | | | | | | | | | | |
| **Implementation** | **14 weeks** | | | | | | | | | | | | | | | | |
| Develop Prototype | 9 weeks | 19/6/2023 | 20/8/2023 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | | | | |
| System Testing | 9 weeks | 26/6/2023 | 27/8/2023 | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | | | |
| System Evaluation | 9 weeks | 3/7/2023 | 3/9/2023 | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | | |
| System Deployment | 2 weeks | 28/8/2023 | 10/9/2023 | | | | | | | | | | | ■ | ■ | | |
| | | | | | | | | | | | | | | | | | |
| Documentation | 12 weeks | 19/6/2023 | 17/9/2023 | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | |
| Presentation | 1 day | | | | | | | | | | | | | | | | ■ |

Figure 3.4.2.1 Project II Timeline

# CHAPTER 4

# System Design

## 4.1 System Architecture

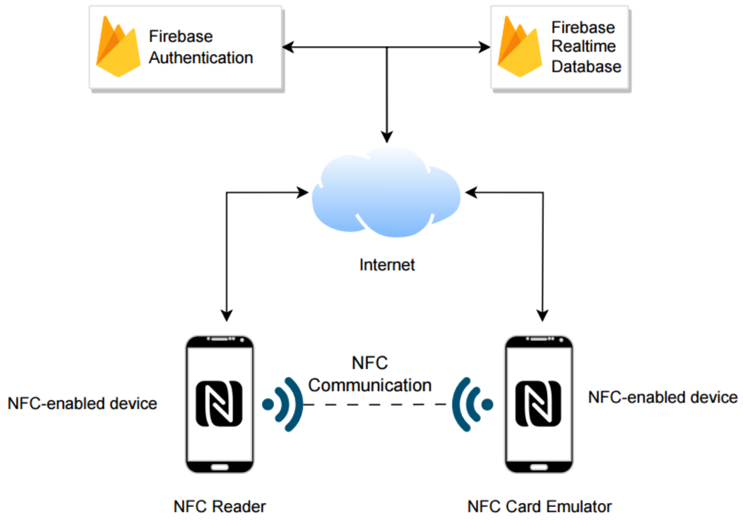


Figure 4.1.1 Diagram of the System Connection

The figure above shows how the system is connected. First, there are two NFC-enabled devices will be involved in this development. One of the devices will act as an NFC reader while another NFC-enabled device will act as an NFC card emulator.

When the moment the client wishes to make a payment or perform a fund transfer, the client will be required to turn on the NFC module to let the mobile phone emulate a card then wait for the NFC reader to be near and to perform communication between them. The NFC module of the NFC reader side must also be turned on.

During the process of the payment, the NFC reader will retrieve the information of the client by the data given by the card-emulating device. Then the reader side will do the accounting accordingly such as getting the funds from the client side and then updating the server so that the process is done.

## 4.2 Use Case Diagram

The use case diagram is used to describe the overview of the system that can be interacted with and function from the view of the users.

In this project, an unauthenticated user does not register an account for this system, so an unauthenticated user only has access to sign up, login and language-changing use cases.

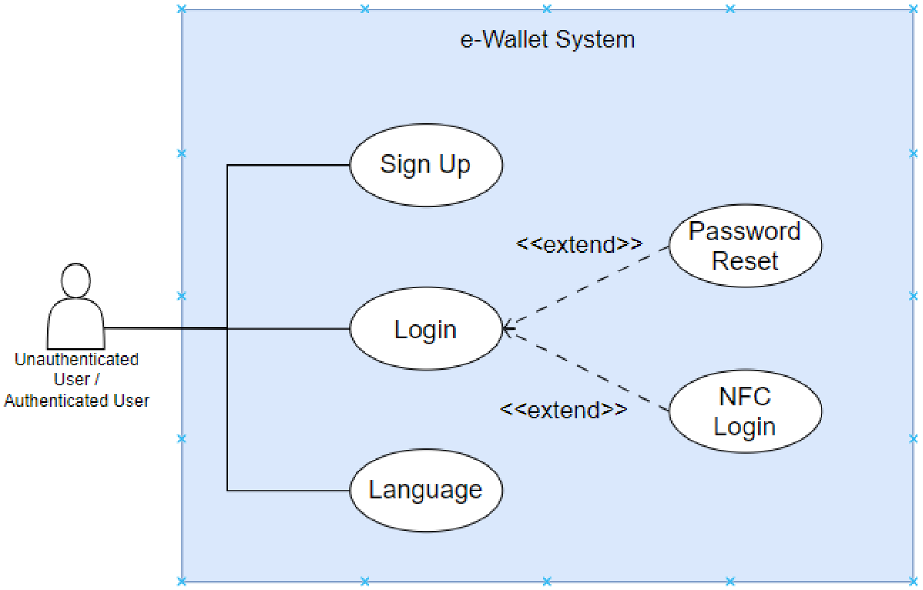


Figure 4.2.1 Use Case Diagram

**Table 4.2.1 Use Case Description**

| Module | Description |
|---|---|
| Sign Up | Allows users to create an account. |
| Login | Allows users to log in to the system. This can be extended to the Password Reset to allow the unauthenticated user to reset their account password. And also, it can be extended to the NFC Login which allows the user to transfer their account from an old device to a new device. |
| Language Changing | Allows users to change the system displaying language. |

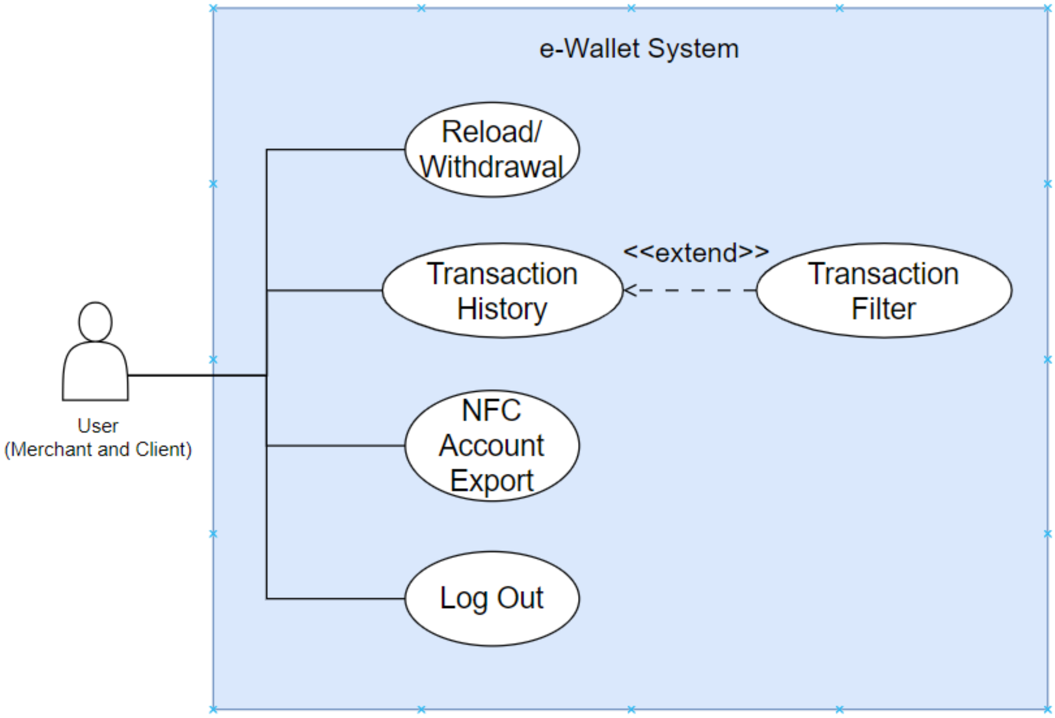


Figure 4.2.2 Use Case Diagram

The figure above includes the modules that are implemented in both roles (Merchant and Client). With these use cases, it is for logged-in users, including the Merchant and Client roles.

**Table 4.2.2 Use Case Description**

| Module | Description |
|---|---|
| Reload / Withdrawal | Allows the users to perform balance reload or withdrawal when it is necessary. |
| Transaction History | Allows the users to trace back what transaction has been made. This can be extended to filter the transaction for a specific range to check. |
| NFC Account Export | Allows the users to transfer their account to a new device instead of keying in their email and password in their new device again. |
| Log Out | Allows the users to log out from the system. |

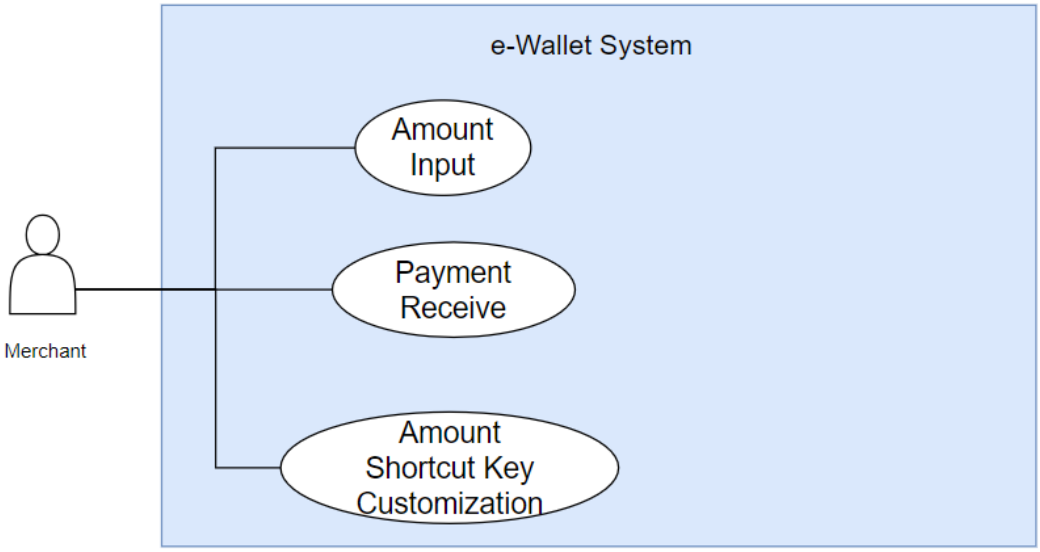


Figure 4.2.3 Use Case Diagram

The use case diagram above is mainly focused on the merchant side if the user account is the merchant role, and log-in using the merchant application.

**Table 4.2.3 Use Case Description**

| Module | Description |
|---|---|
| Amount Input | Allows the merchant to enter the amount that is to be received from their customer. |
| Payment Receive | Allows the merchant to perform payment receive using NFC technology. |
| Amount Shortcut Key Customization | Allows the merchant to use the customized shortcut key to enter the amount that is to be received from their customer. The merchant can always update their amount shortcut key. |

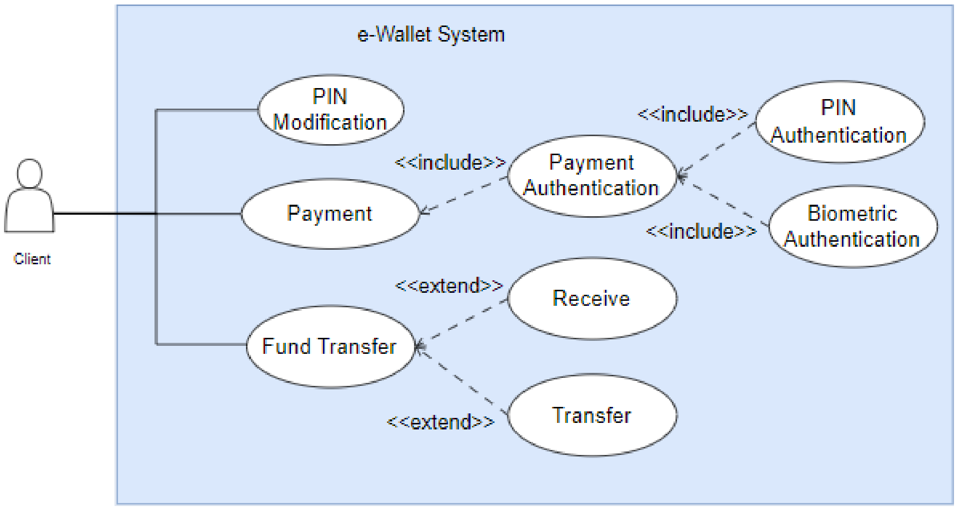


Figure 4.2.4 Use Case Diagram

The diagram above mainly describes the use case for client-side only, if the user account is the client role, and log-in using the client application.

**Table 4.2.4 Use Case Description**

| **Module** | **Description** |
|---|---|
| PIN Modification | Allows the client to modify their payment PIN as needed. |
| Payment | Allows the client to make a payment using NFC HCE technology. This includes payment authentication to authenticate the payment, and it can use the PIN or biometric to authenticate. |
| Fund Transfer | Allows the client to perform fund transfers using NFC HCE technology. This can be extended to fund receive or fund transfer. |

## 4.3 System Flow

### 4.3.1 Account Login/NFC Login Activities

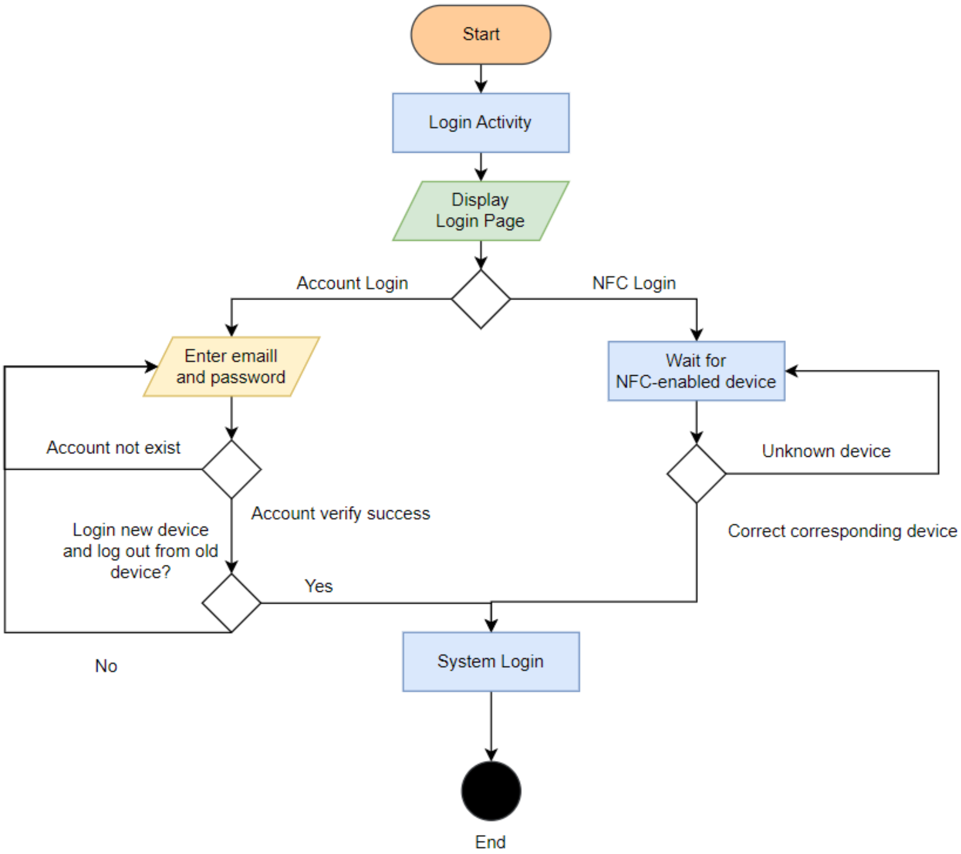


Figure 4.3.1 Login Activity Diagram

The figure above shows the flow of the system logging in. There are two methods to allow the user to perform system login. The first method is to manually input their email and password. Only the valid account will be granted access and authorization to use the application. With this login method, the system will check whether the current login device whether same as the last login device, if not the same, the system will be required to replace the last login device ID, log out from the old device and log in to the new device.

The other method is using NFC technology to transfer their old account from the old device to the new device, this does not involve any validation since the validation had been done from the side of the old device. Then, retrieve the information through the NFC and log in to the system accordingly.

### 4.3.2 Registration Activity

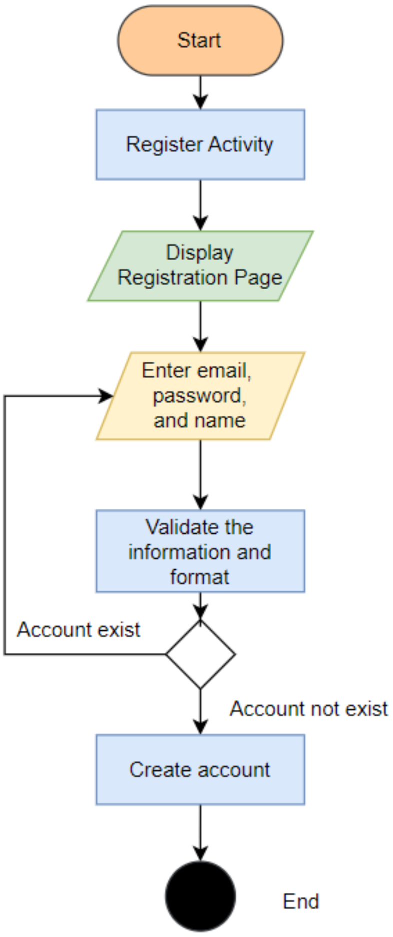

Start
Register Activity
Display Registration Page
Enter email, password, and name
Validate the information and format
Account exist
Account not exist
Create account
End

Figure 4.3.2 Registration Activity Diagram

For the registration module, the user is required to key in their valid email, password and name. Then, the format of the email and password will be validated. It includes the email format whether valid, and the length of the password whether more than 6 digits to create an account. Otherwise, the system will display an error message to notify the user where the error is.
After that, the system will check the user database, if the account exists, the registration will fail and toast a message to the user.

Only valid and the email does not exist in the database will register successfully.

### 4.3.3 Forgot Password Activity

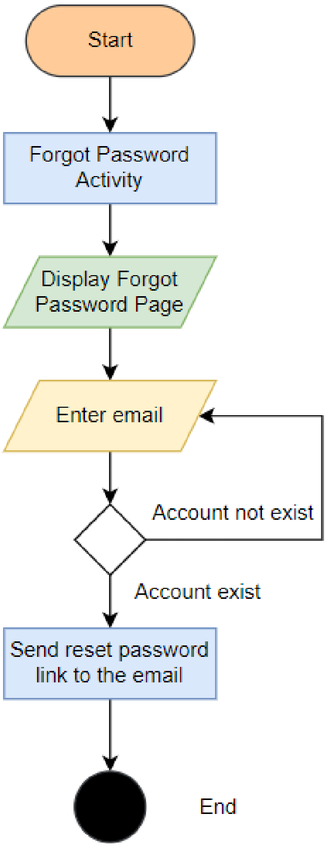


Figure 4.3.3 Forgot Password Activity Diagram

For the forgot password module, the user is allowed to reset their account password. First of all, the user is required to enter their email. Only the account existing in the database, the server will just send the reset password link to the user’s mailbox. So that the user can click the link and reset their password accordingly.

Otherwise, the system will prompt an error message to the user that implies the account does not exist in the user database.

### 4.3.4 Reload/Withdraw Activities

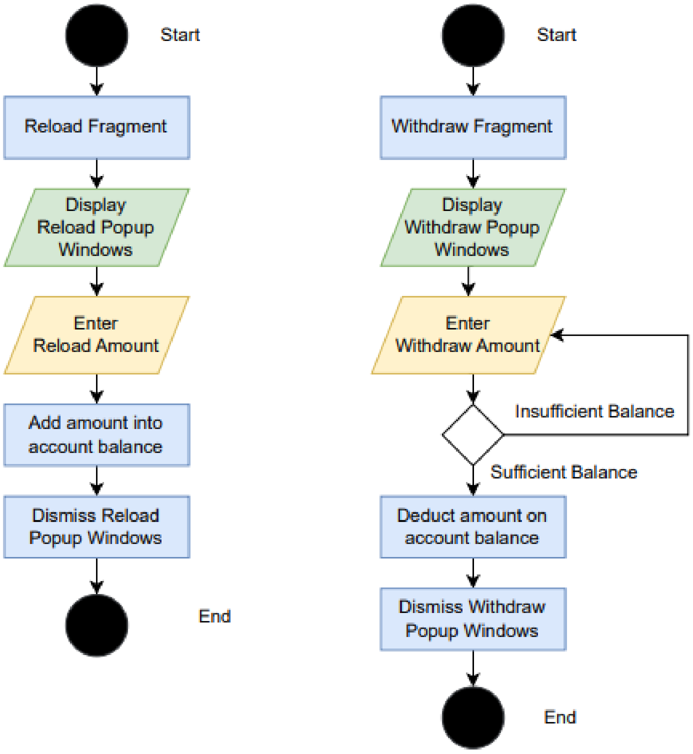

Start
Reload Fragment
Display Reload Popup Windows
Enter Reload Amount
Add amount into account balance
Dismiss Reload Popup Windows
End
Start
Withdraw Fragment
Display Withdraw Popup Windows
Enter Withdraw Amount
Insufficient Balance
Sufficient Balance
Deduct amount on account balance
Dismiss Withdraw Popup Windows
End

Figure 4.3.4 Reload/Withdraw Activity Diagram

The diagram above illustrates the flow of balance reloads or withdrawals. When the user would like to perform the related action, the corresponding window will be prompted. A certain amount will be required to be input by the system, then the system will act accordingly.

In terms of reload, the user can top up the balance while the terms of withdrawal, the user can withdraw the amount from their balance into their bank account when the user does not want to keep too much balance in the system.

In the withdrawal process, the withdrawal amount is required to be less or equal to the balance, otherwise, failure to withdraw the amount.

### 4.3.5 Transaction Activity

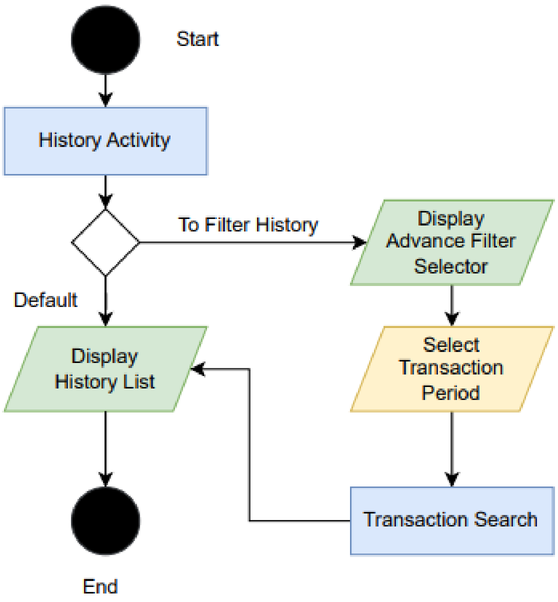


Figure 4.3.5 Transaction Activity Diagram

Figure 4.3.5 shows the flow chart of the transaction module. It first displays the default history list, which is all the histories to be displayed. Otherwise, the user can select the range to filter the transaction the user wants to be displayed. Then the list will be based on the filtered list to display the transaction history.

### 4.3.6 Language Changing Activity

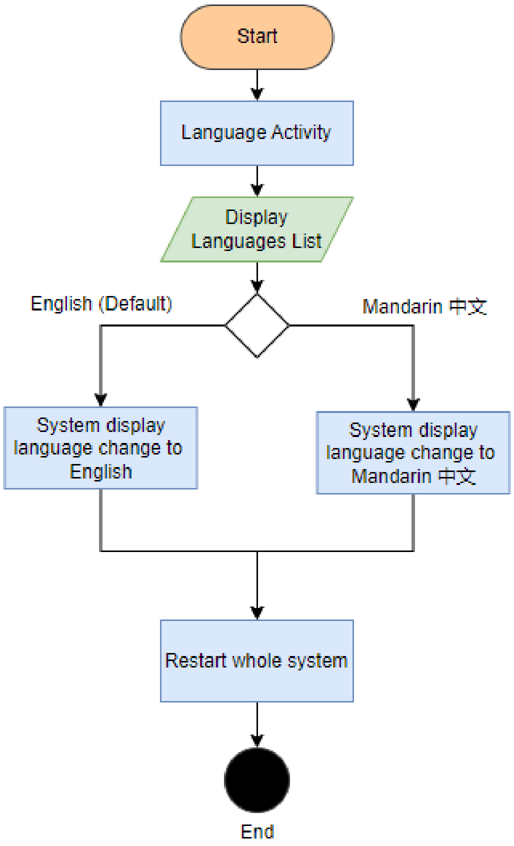


Figure 4.3.6 Language Changing Activity Diagram

The figure above is mainly describing the language application process. Generally, there are three available languages to be selected, including Mandarin and English. Once the user selects the language that intends to be used in the system, the system will automatically restart and display the selected language as the system user interface.

### 4.3.7 Log Out Activity

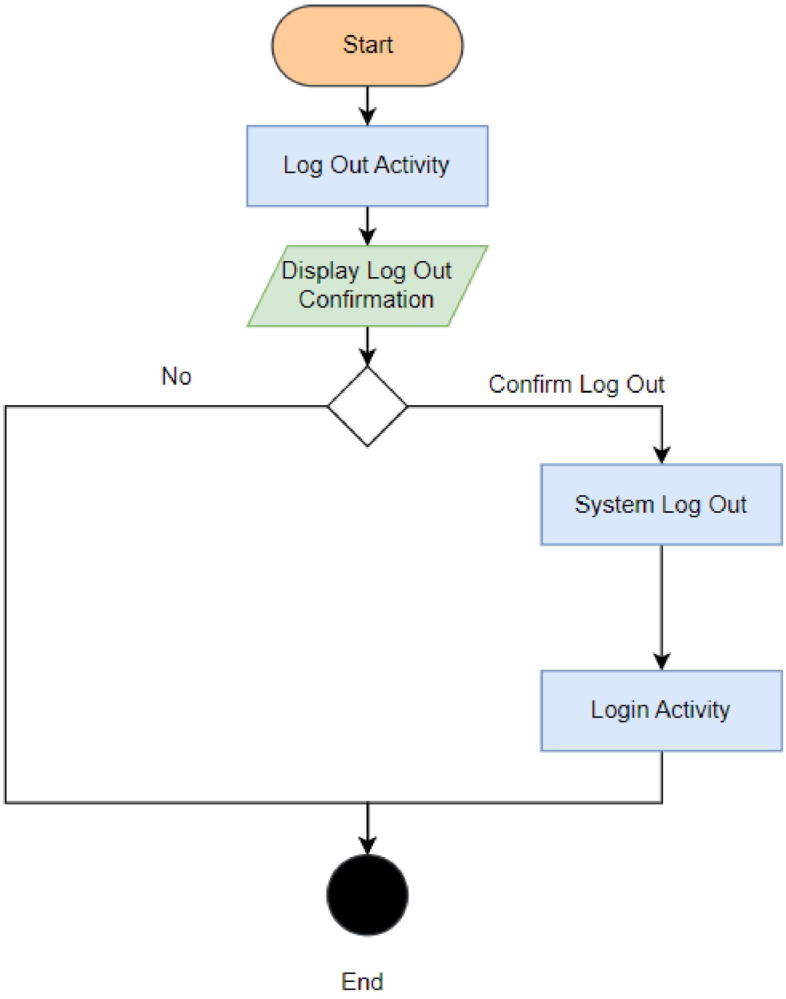


Figure 4.3.7 Log Out Activity Diagram

This is the diagram for the log-out activity. Once the user clicks the “Log Out” button, the user will be asked to confirm whether to log out. If yes, the system will log the user out, and redirect back to the login interface.

### 4.3.8 PIN Registration Activity

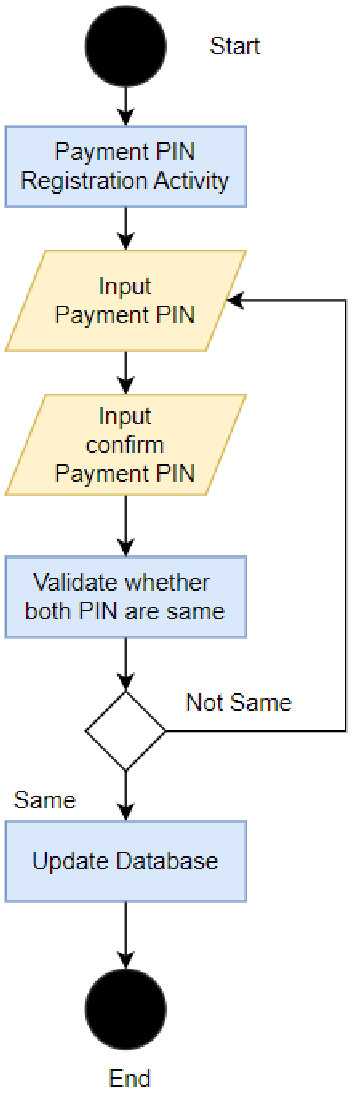

Start
Payment PIN Registration Activity
Input Payment PIN
Input confirm Payment PIN
Validate whether both PIN are same
Not Same
Same
Update Database
End

Figure 4.3.8 PIN Registration Activity Diagram

The figure above illustrates the process of PIN registration. The PIN is needed for authentication usage such as before the payment process. This activity only involves when they do not have any PIN in the database and they would like to proceed to make a payment or withdrawal.

First, the system will request the client to key in a PIN and then key in again to confirm the PIN. If both PINs are the same, the system will keep them in the database. Otherwise, the system will keep requiring the client to key in both PINs until both PINs are matched.

### 4.3.9 PIN Modification Activity

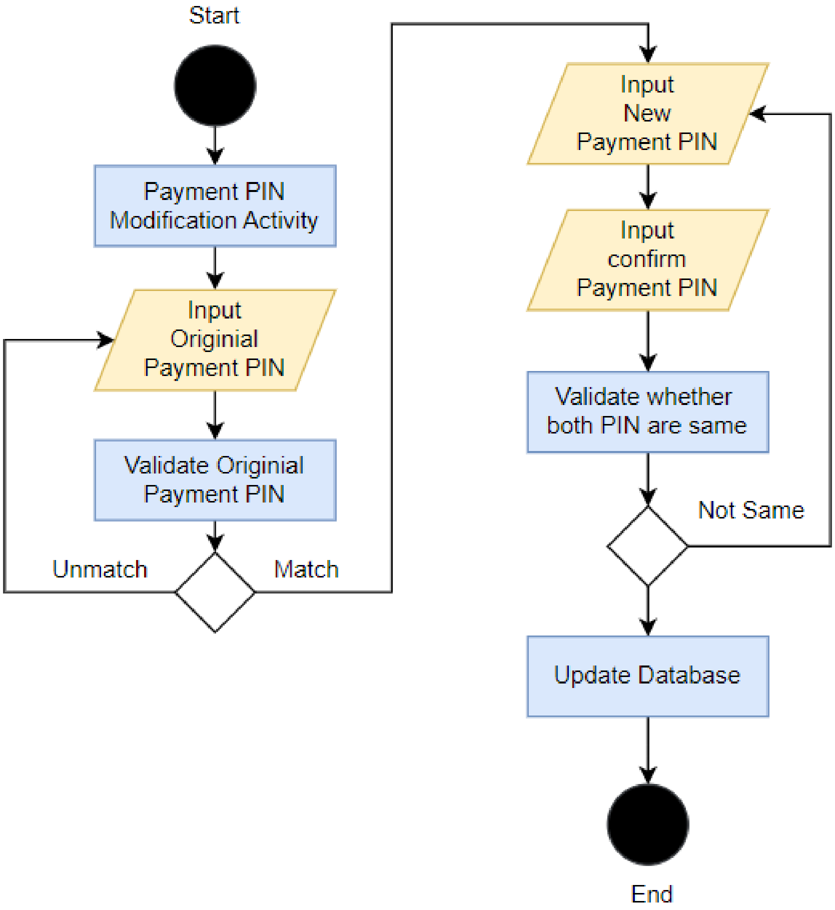

Start
Payment PIN Modification Activity
Input Originial Payment PIN
Validate Originial Payment PIN
Unmatch
Match
Input New Payment PIN
Input confirm Payment PIN
Validate whether both PIN are same
Not Same
Update Database
End

Figure 4.3.9 PIN Modification Activity Diagram

The diagram above is to show the process of PIN modification activity. Compared to the PIN registration diagram, this activity is just adding one more process which is the verification of the current PIN.

Only when the original PIN is corrected, the client will be allowed to modify or change their PIN to a new PIN. Otherwise, the modification is not allowed at all.

### 4.3.10 Receiving Amount Input Activity

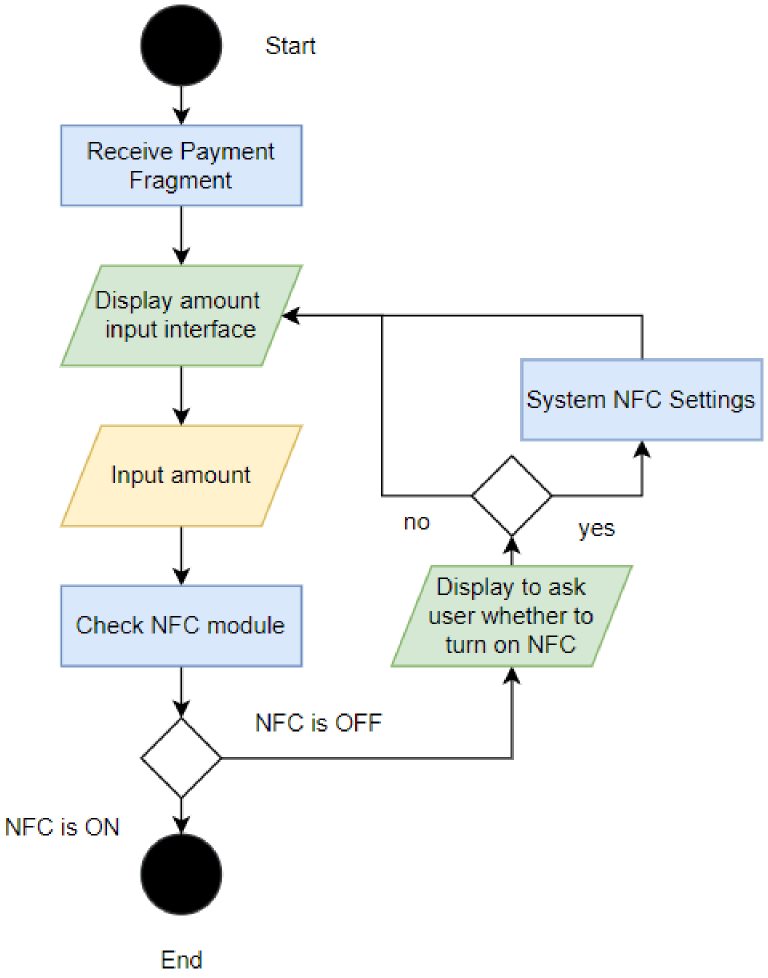


Figure 4.3.10 Diagram of Receiving Amount Input for Merchant

The diagram above is to describe the payment amount intended to be received by the merchant that needs to be input. First, the interface comes out, and then the merchant has to input the amount to be received from their client. Or else, they are able to click the amount shortcut key to select the amount.

After the amount is checked that it is valid, the system will check whether the NFC module is currently running. If no, the system will ask the user whether to turn on the NFC module, and the user will be redirected to the system NFC Settings interface to enable it.

### 4.3.11 NFC Payment Receive Activity

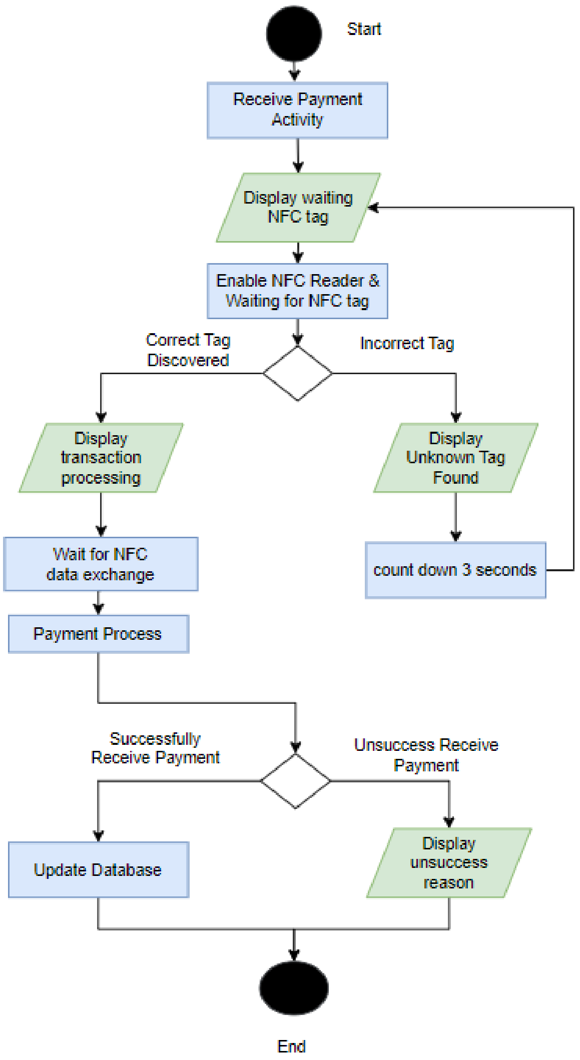

Start
Receive Payment Activity
Display waiting NFC tag
Enable NFC Reader & Waiting for NFC tag
Correct Tag Discovered
Incorrect Tag
Display transaction processing
Display Unknown Tag Found
Wait for NFC data exchange
count down 3 seconds
Payment Process
Successfully Receive Payment
Unsuccess Receive Payment
Update Database
Display unsuccess reason
End

Figure 4.3.11 NFC Payment Receive Activity Diagram for Merchant

The diagram above is to describe the payment-receiving process for the merchant role. Once the payment amount is confirmed, the merchant will be redirected to the payment receive activity. At this moment, the system will enable the NFC reader mode and wait for the corresponding NFC tag.

The NFC tag stands for the NFC HCE card to be read in this system. If the tag is not what the system wants, it will result in payment being unsuccessful, and then the system will keep waiting for the valid card until the payment is done. In terms of the successful payment, the merchant will receive the amount as well as an update to the history database.

### 4.3.12 Amount Key Customization Activity

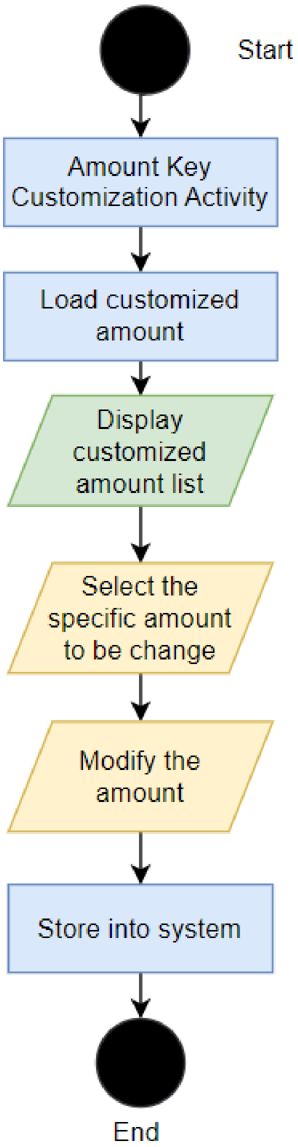

Start
Amount Key Customization Activity
Load customized amount
Display customized amount list
Select the specific amount to be change
Modify the amount
Store into system
End

Figure 4.3.12 Amount Key Customization Activity Diagram for Merchant

The diagram above describes the process of customization on the amount shortcut key for merchants while they input the amount that is to be received from a client. When the merchant enters this interface, there are five amounts are able to be customized, then the merchant can select one of them to modify.

Once the modification is finished, the customization will be stored in the mobile system, then only the merchant enters the amount of input activity, and the customization will be loaded from the system again.

### 4.3.13 Client NFC Payment Activity

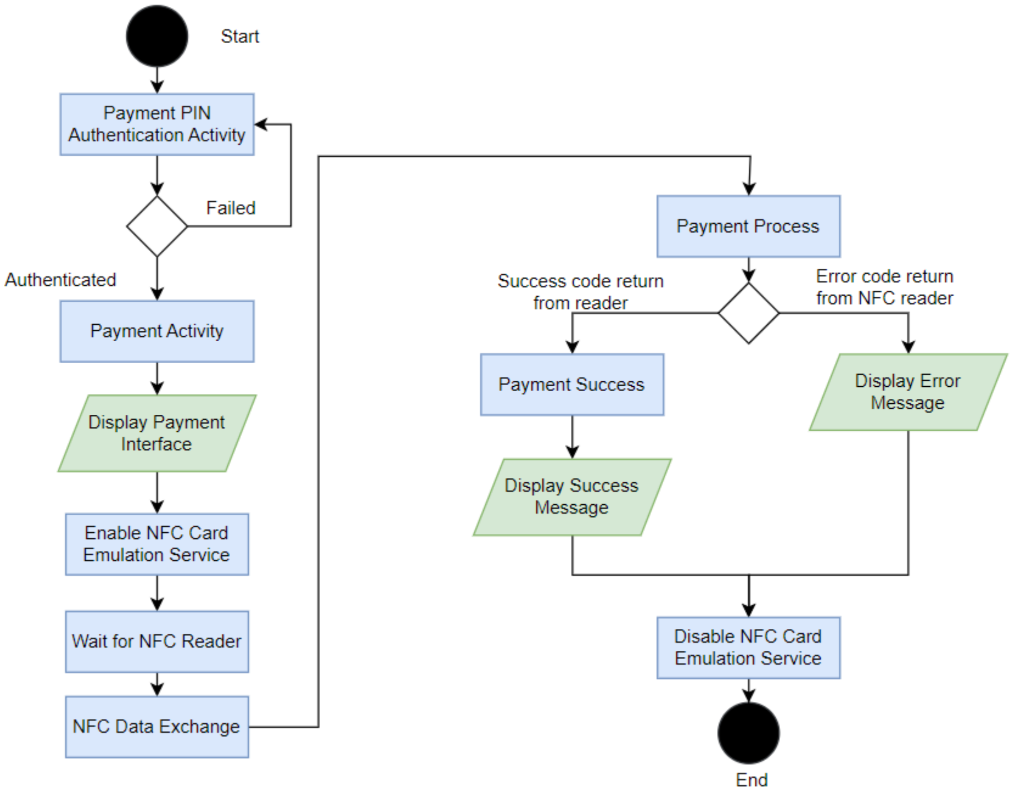

Start
Payment PIN Authentication Activity
Failed
Authenticated
Payment Activity
Display Payment Interface
Enable NFC Card Emulation Service
Wait for NFC Reader
NFC Data Exchange
Payment Process
Success code return from reader
Error code return from NFC reader
Payment Success
Display Error Message
Display Success Message
Disable NFC Card Emulation Service
End

Figure 4.3.13 Client Payment Process Diagram

Figure 4.3.13 shows the diagram for the payment process. After the user intends to make a payment, the user needs to key in their payment PIN to authenticate the payment activity to process a transaction. Then, the system will start enabling the card emulation service.

After that, the user interface will notify the user to let the mobile approach the NFC reader to carry out the payment transaction. Then wait for the status code returned by the NFC card reader. There are two status codes to be returned by the reader, which are the success code and the failure code. The system will be based on the status code to display the message in the user interface.

In the end, the system will stop the host-based card emulation service.

### 4.3.14 Client NFC Fund Transfer Activity

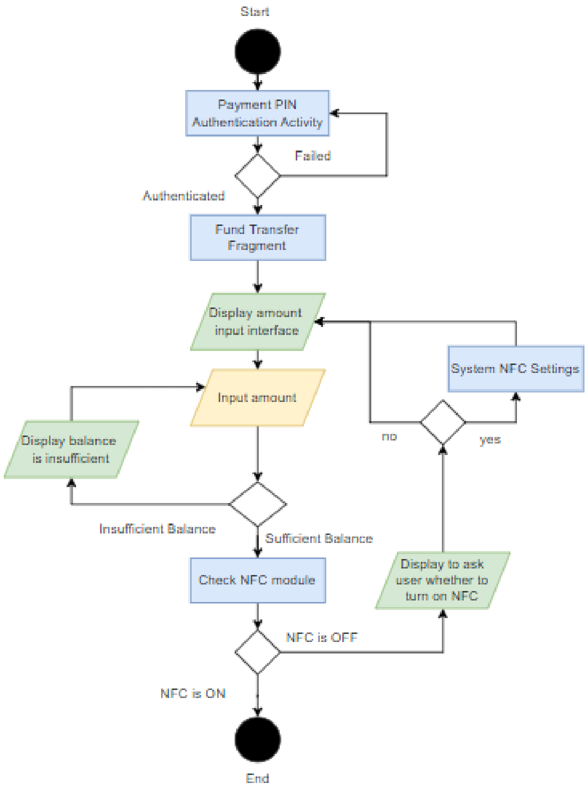


Figure 4.3.14 Fund Transfer Activity Diagram

First of all, the user is required to authenticate for entering the fund transfer fragment, this is to authenticate themselves to process the transfer. After that, the user has to input the amount to be transferred to another user, and the amount cannot be more than their e-wallet balance. Before entering the fund transfer NFC activity, the system will check whether the NFC module is available and turn it on in the current situation. Only the conditions are fulfilled and then the system will redirect to the fund transfer activity.

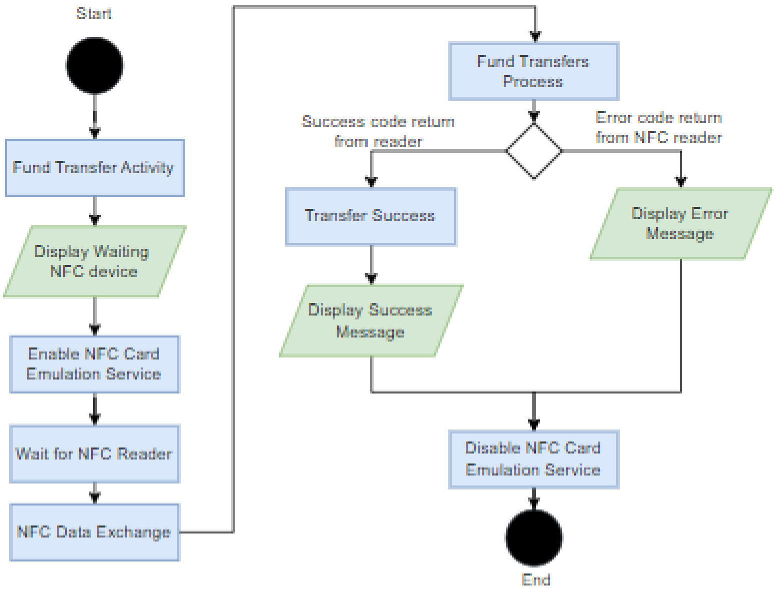

Start
Fund Transfer Activity
Display Waiting NFC device
Enable NFC Card Emulation Service
Wait for NFC Reader
NFC Data Exchange
Fund Transfers Process
Success code return from reader
Error code return from NFC reader
Transfer Success
Display Error Message
Display Success Message
Disable NFC Card Emulation Service
End

Figure 4.3.15 Fund Transfer NFC Process

The figure above shows the process of fund transfer. Once all the authentication is done and the NFC module is running, then the system will start enabling the card emulation service. After that, the user interface will notify the user to tap their NFC-enabled mobile device to approach to the NFC card reader (or NFC-enabled mobile device in card reader mode) for NFC data exchange purposes.

When the corresponding NFC card reader is connected, this system will wait for the status code to be returned by the NFC card reader. There are two types of status codes to be returned by the card reader, which are the success code and the failure code.

The system will be based on the returned status code display-related messages in the user interface. Once all the actions are done, the system will immediately stop the NFC HCE emulation service.

### 4.3.15 NFC Fund Transfer Receiving Activity

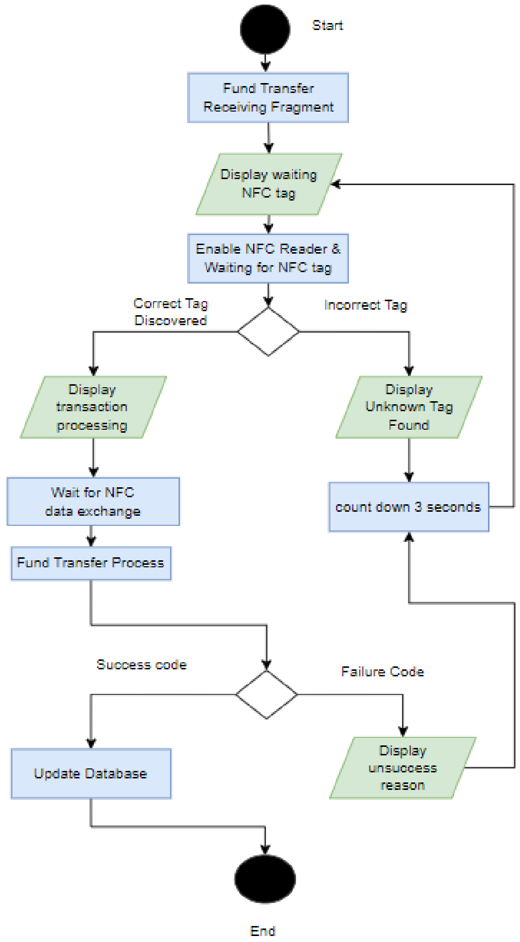


Figure 4.3.16 Fund Transfers Receiving Activity Diagram

The diagram above is to describe the fund transfer receiving process. When the user selects to receive a fund transfer, the user interface will first display that currently is waiting for an NFC-enabled device or tag to perform the NFC data exchange.

If it is not the corresponding NFC-enabled device or tag is found, the system will display an error message and back to the initial UI to wait for the corresponding NFC-enabled device or tag.

Once the correct NFC-enabled device or tag is found, the system NFC card reader will perform data exchange to get the fund transfers. This usually will get the success result code, as the exception handling is carried out before entering the transfer activity.

## 4.4 Database Design

The following figures show the database design and the entity relationship diagram to show how the database has a relationship with each other.

Then, the table 4.4.2 and 4.4.3 explain each field of the column that is needed for what purpose.

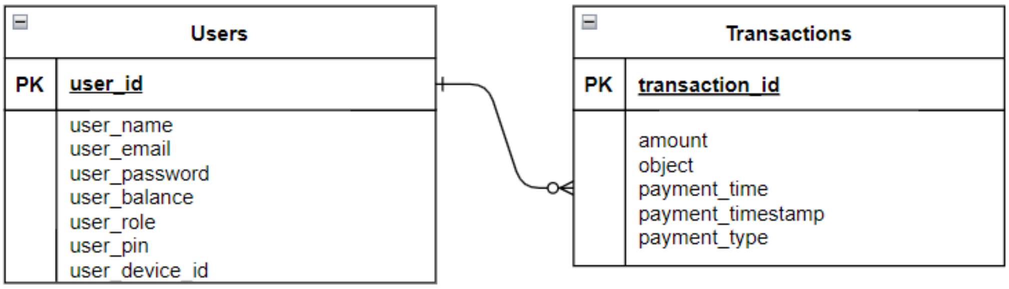


Figure 4.4.1 Entity Relationship Diagram

**Table 4.4.1 Entity Description**

| Entity | Description |
|---|---|
| Users | To store all the user's personal details |
| Transactions | To store all the transaction details |

**Table 4.4.2 Data Dictionary of Users**

| Field Name | Data Type | Description |
|---|---|---|
| user_id (PK) | String | User Identification |
| user_name | String | User name. |
| user_password | String | User account password. |
| user_balance | String | User account balance. |
| user_role | String | User’s role. |
| user_pin | String | Client’s payment PIN |
| user_device_id | String | The account is linked device’s ID. |

**Table 4.4.3 Data Dictionary of Transactions**

| Field Name | Data Type | Description |
|---|---|---|
| transaction_id | String | Transaction Identification |
| amount | Double | Transaction Amount |
| object | String | Transaction Payee(/Payer) |
| payment_time | String | Transaction date and time |
| payment_timestamp | Long | The timestamp of the transaction is created. |
| payment_type | String | The type of the transaction. |

## 4.5 Concluding Remarks

This chapter defines the architecture of the proposed system, which illustrates how the system connects to each other, including end-users, databases and others.

This is followed by the use case diagrams that show different roles or types of identification and how they interact with the system. Based on their role what they can do in the system.

Then, the activity diagram describes the flow of the process and how it works from the beginning until the end of the activity.

Lastly, the database design is how the database looks like. Other than that, it shows the entity relationship between each database.

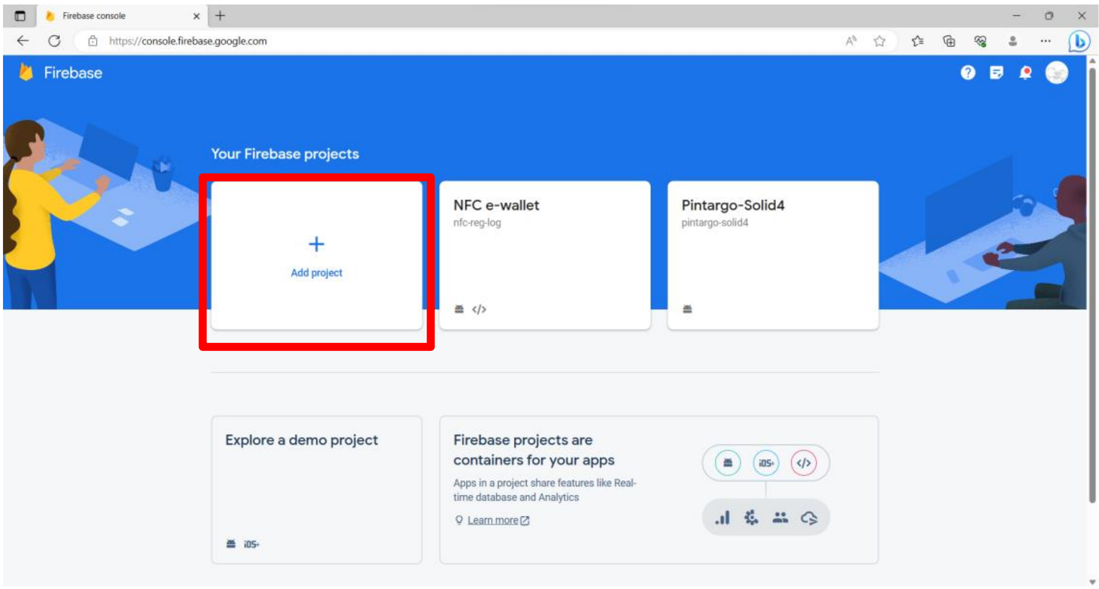
# Chapter 5

# System Implementation

## 5.1 Software Setup (Installation)

### 5.1.1 Firebase

Google implements Firebase as a platform that allows developers to develop their applications or systems quickly without worrying about the backend [44]. Google as a platform provider offers a Backend-as-a-Service (Baas) that includes a lot of features, such as cloud messaging, machine learning and so on. In this project, Authentication and real-time database will be utilised in the development.



Firebase projects are containers for your apps

Apps in a project share features like Real-time database and Analytics

Figure 5.1.1.1 Adding a Firebase Project

To set up a Firebase project, proceed to the Firebase website to add a Firebase project after account login.

Specify the name of the Firebase project, then press the button "*Continue*" to create the Firebase project.

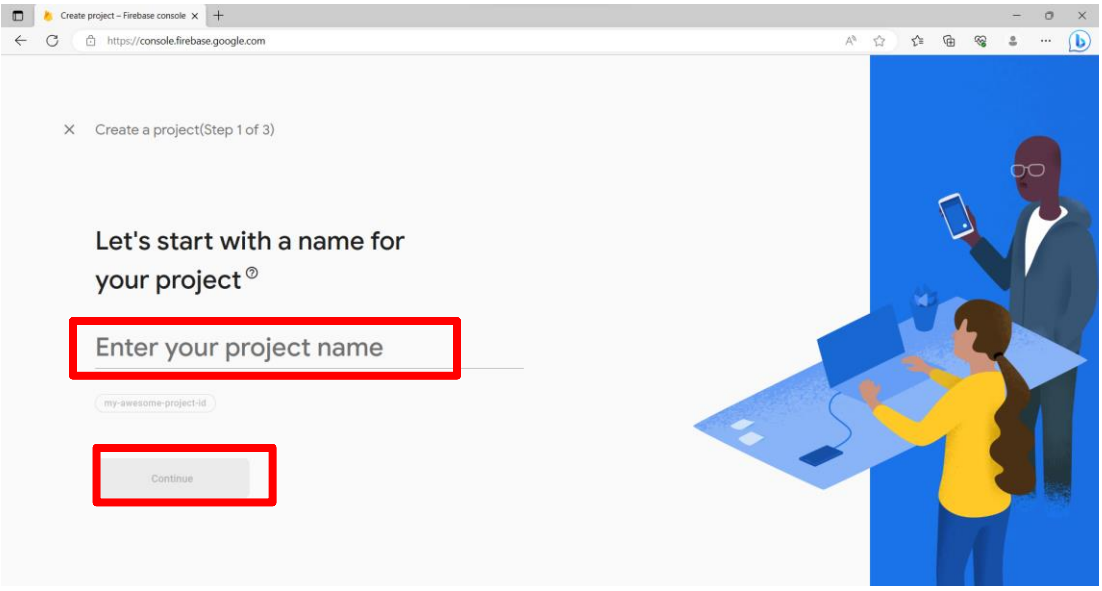

Create project – Firebase console
https://console.firebase.google.com
Create a project(Step 1 of 3)
Let's start with a name for your project
Enter your project name
my-awesome-project-id
Continue

Figure 5.1.1.2 Creating Firebase Project with Specified Name

After the Firebase project is successfully created, click "*Build*" in the drop-down menu > "*Authentication*" to set up the authentication with this project.

In the Sign-in option, select the "*Email/Password*" as the preferred sign-in method.

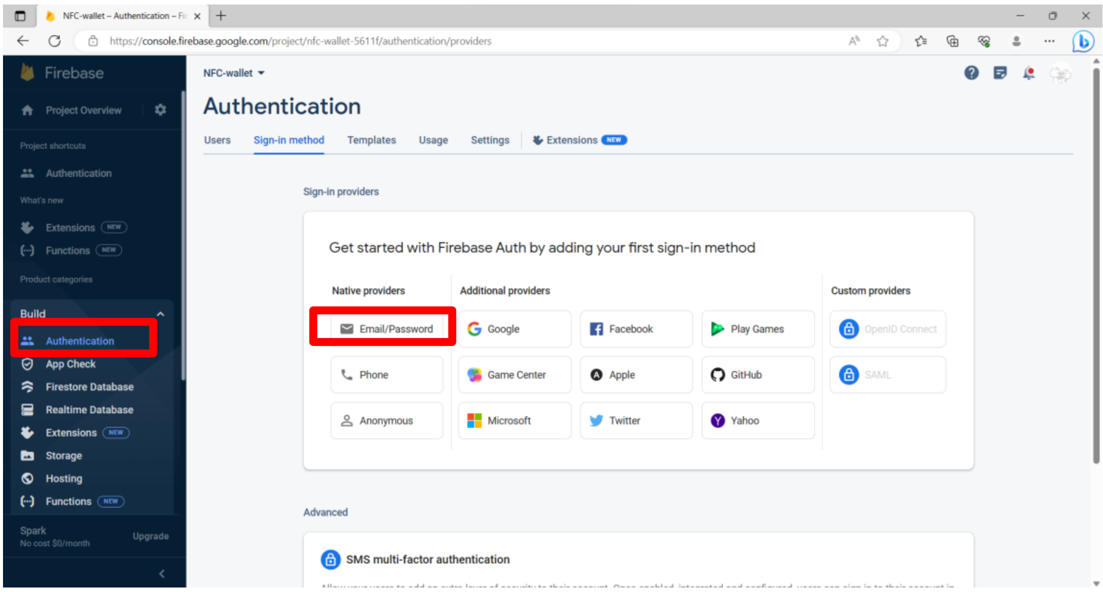
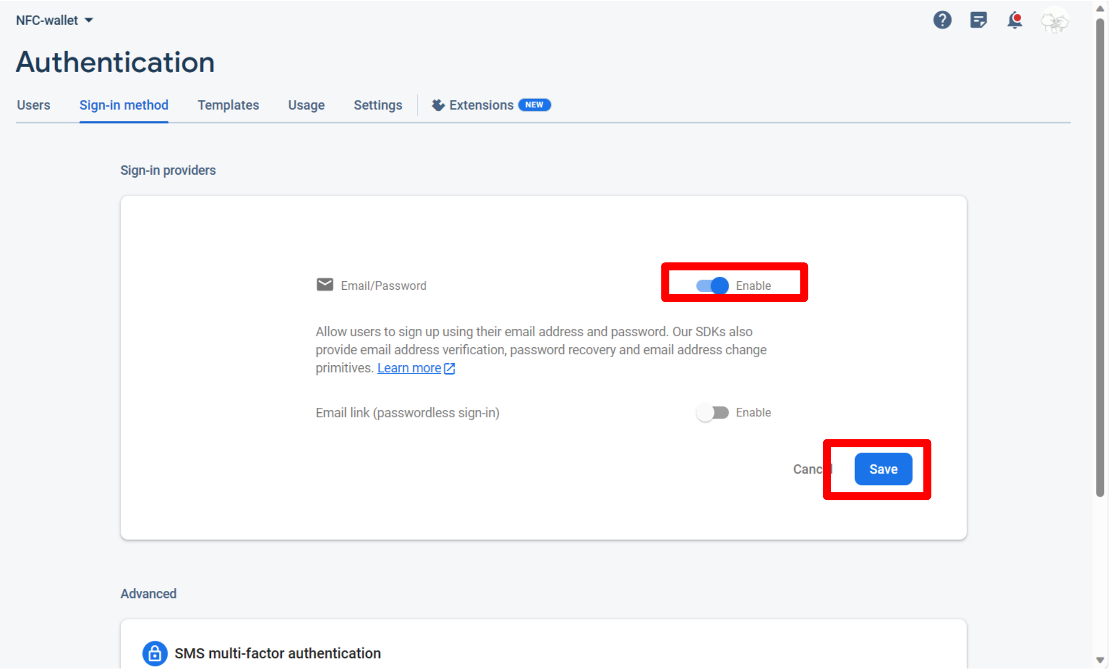


Figure 5.1.1.3 Authentication Set Up

Then, select the "*Sign-in method*" at the top, and enable this sign-in method then save.



Allow users to sign up using their email address and password. Our SDKs also provide email address verification, password recovery and email address change primitives. Learn more

Figure 5.1.1.4 Authentication Method Enable

Another feature which is Firebase Realtime Database is used to synchronize the user application or system data in the cloud, so the user no longer has to use the local storage to store the data.

To set up Firebase Realtime Database, click *"Realtime Database"* under the *"Build"* drop-down menu. Then, click on the button *"Create Database"* to set up the database.
After that, based on preference decide the operation mode of the database.

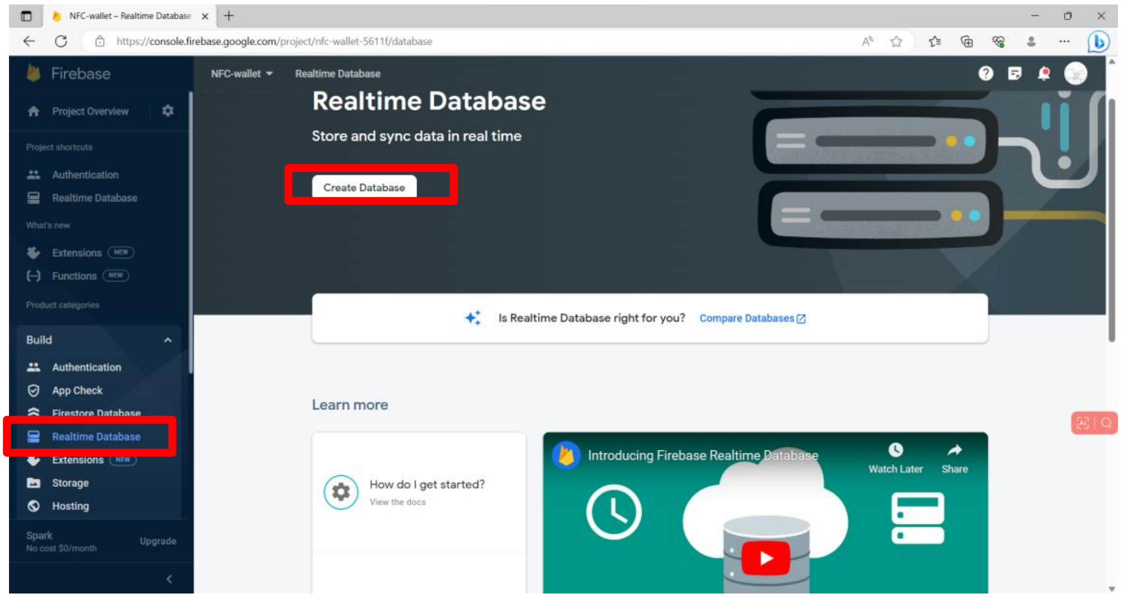
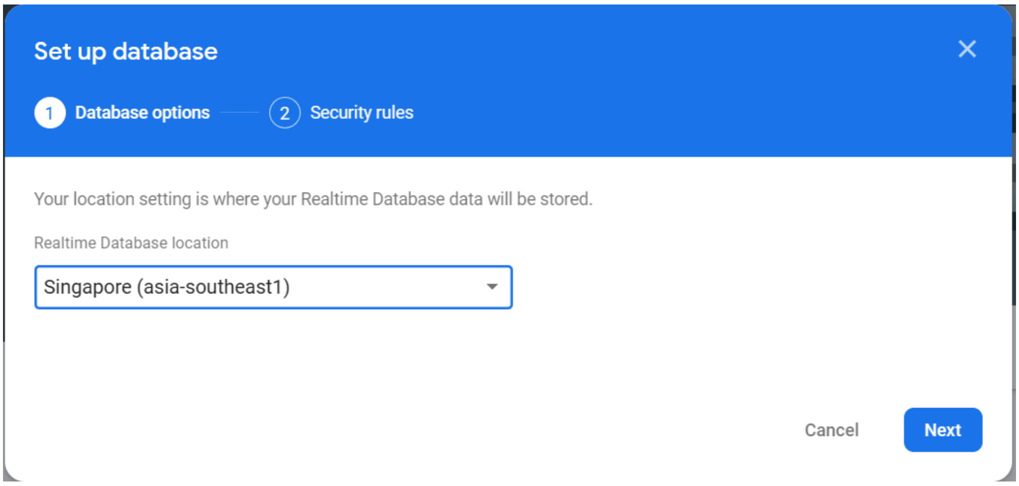


Figure 5.1.1.5 Firebase Realtime Database Set-Up

Based on the current location, select the nearest database location. Then, click “*Next*”.

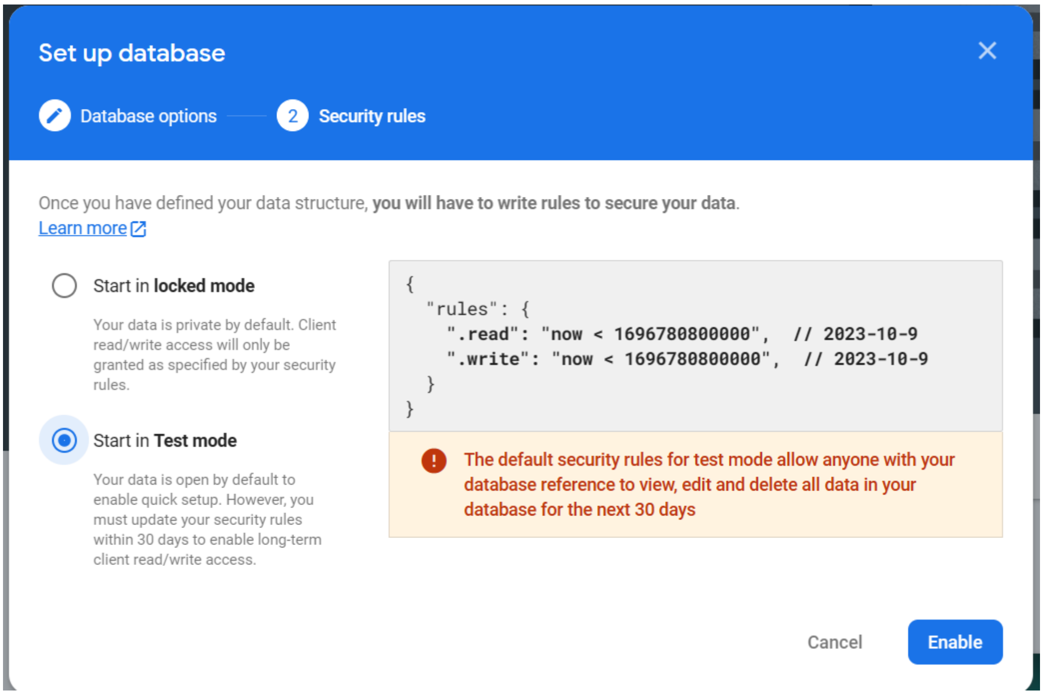


Your location setting is where your Realtime Database data will be stored.

Realtime Database location: Singapore (asia-southeast1)

Figure 5.1.1.6 Database Cloud Location Selection

Select “*Start in **Test mode***” which allows the Firebase database insertion and click “*Enable*”.



Once you have defined your data structure, **you will have to write rules to secure your data.**

○ Start in **locked mode**

Your data is private by default. Client read/write access will only be granted as specified by your security rules.

● Start in **Test mode**

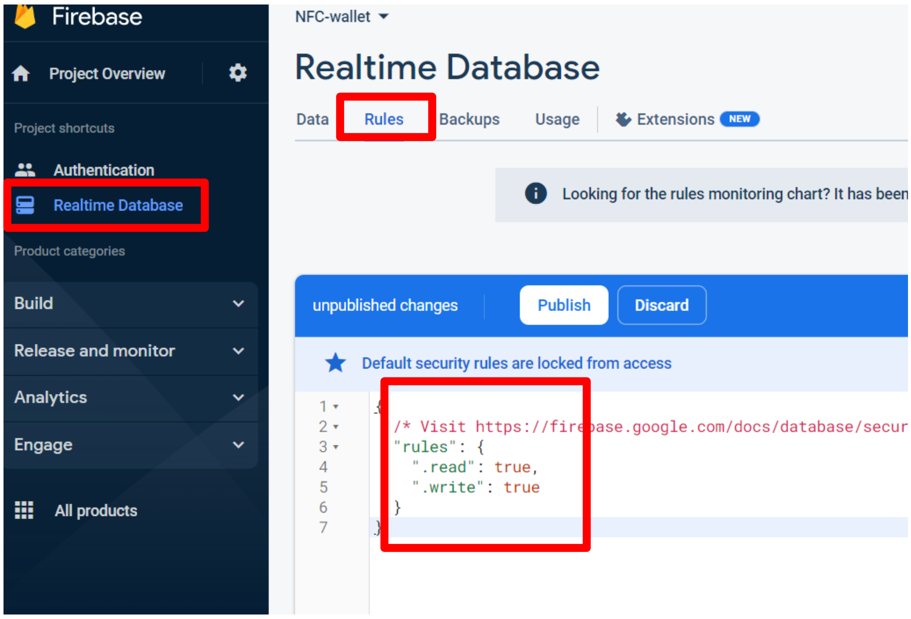
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
{
  "rules": {
    ".read": "now < 1696780800000",  // 2023-10-9
    ".write": "now < 1696780800000",  // 2023-10-9
  }
}
```

The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days

Figure 5.1.1.7 Database Security Rules Settings

Go to the “*Realtime Database*” > “*Rules*” and change the rules as follows. This aims to lifetime real-time database usage.



```
{
  /* Visit https://firebase.google.com/docs/database/secur
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

Figure 5.1.1.8 Firebase Realtime Database Rules Configuration

Click the “*Project Overview*” and select this project, then select the Android icon to link the app to this Firebase project.

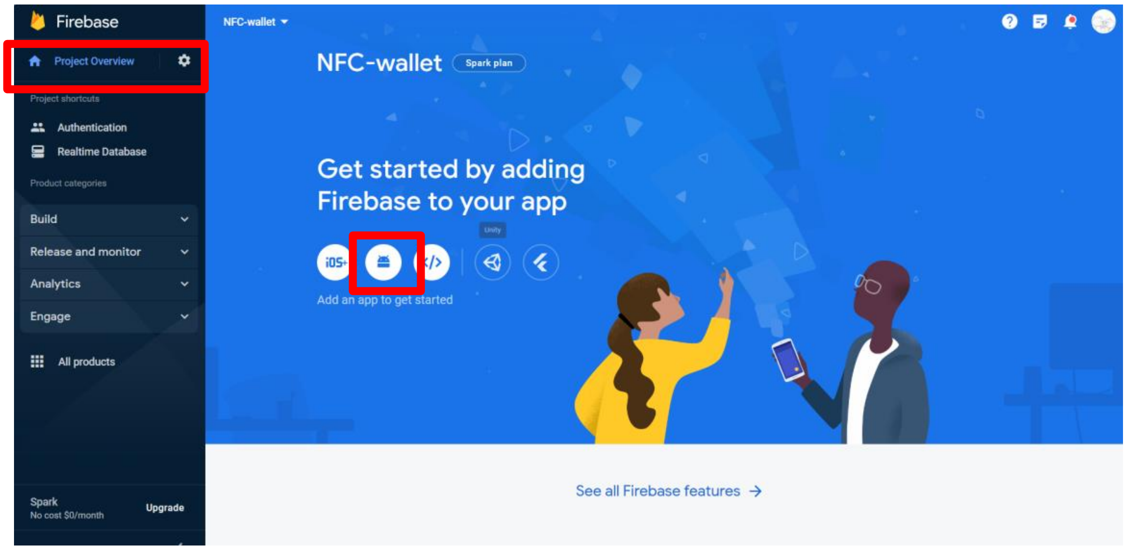
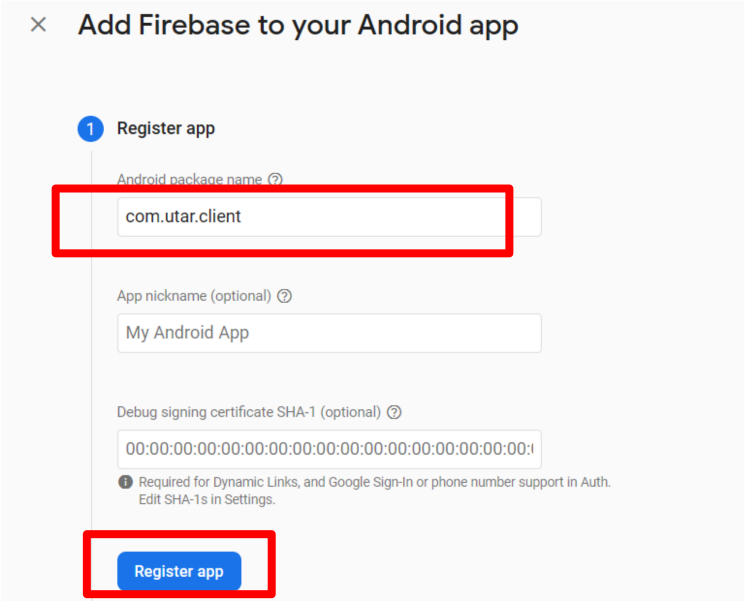


Figure 5.1.1.9 App Linking Configuration

In this interface, just key in "*com.utar.client*" and click "Register app" for the app registration in this Firebase project.



Add Firebase to your Android app

1 Register app

Android package name: com.utar.client

App nickname (optional): My Android App

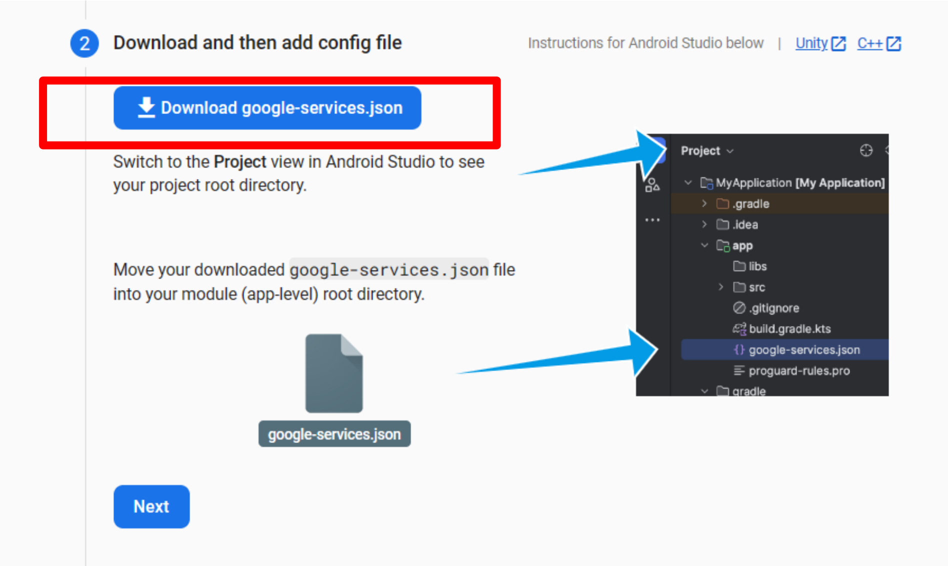
Debug signing certificate SHA-1 (optional): 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:0

Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

Figure 5.1.1.10 Firebase Project App Registration

After the app registration, the configuration file is required to be downloaded. This is needed after the Android Studio project is created.

Do the same steps again for the merchant app. Instead of keying in "*com.utar.client*", now turn to key in "*com.utar.merchant*" and repeat the step from Figure 5.1.1.9.



2 Download and then add config file

Switch to the **Project** view in Android Studio to see your project root directory.

Move your downloaded `google-services.json` file into your module (app-level) root directory.

Figure 5.1.1.11 Firebase Configuration File Download

### 5.1.2 Android Studio

In this section, this ensures the Android Studio has been installed, and there are two NFC-enabled Android devices ready.

Followed by downloading the source code.

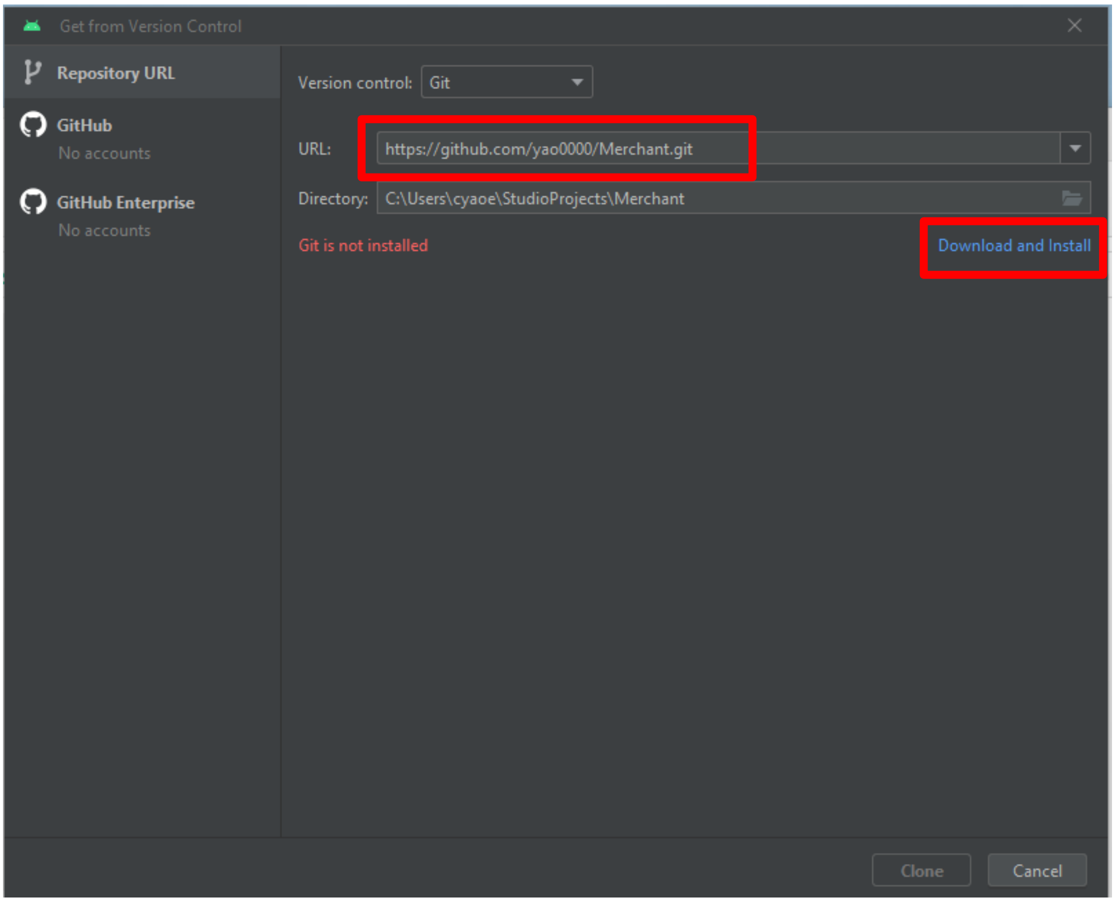
First of all, import the source code into the Android Studio project using "Get from Version Control". This will require the GitHub login and Git to clone the project.

In the URL field, paste the source code address as follows:

Merchant Application Source Code Available: https://github.com/yao0000/Merchant.git

Client Application Source Code Available: https://github.com/yao0000/Client.git

Then, click "*Clone*".



Version control: Git

URL: https://github.com/yao0000/Merchant.git

Directory: C:\Users\cyaoe\StudioProjects\Merchant

Git is not installed

Figure 5.1.2.1 Git Installation

The project will be cloned in this moment and wait until it finishes.

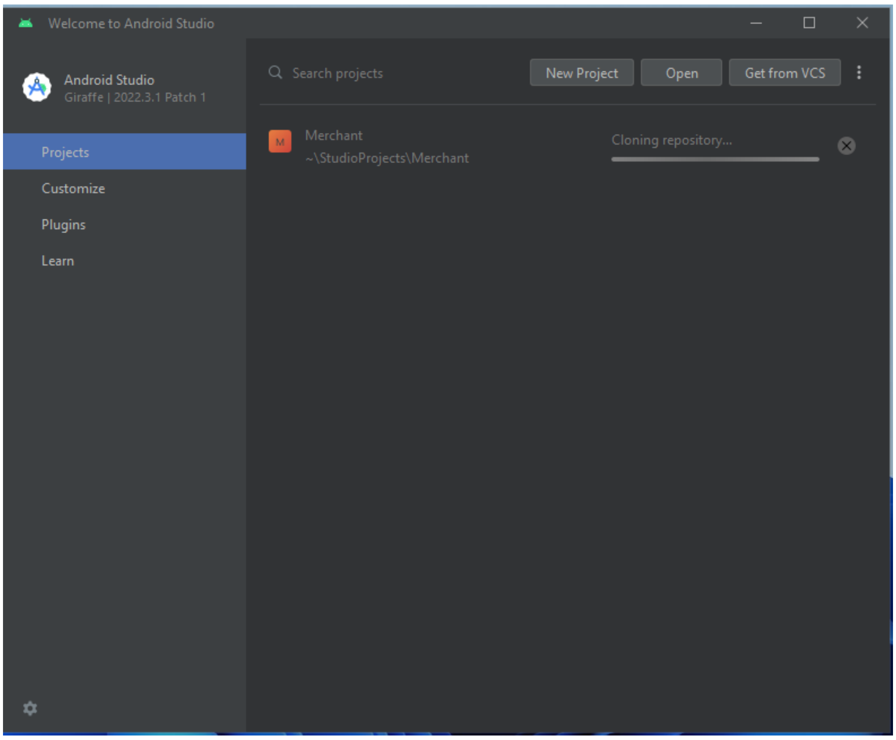


Figure 5.1.2.2 Cloning Repository

After the cloning process is done, click the "*Android*" on the top left side, then select "*Project*".

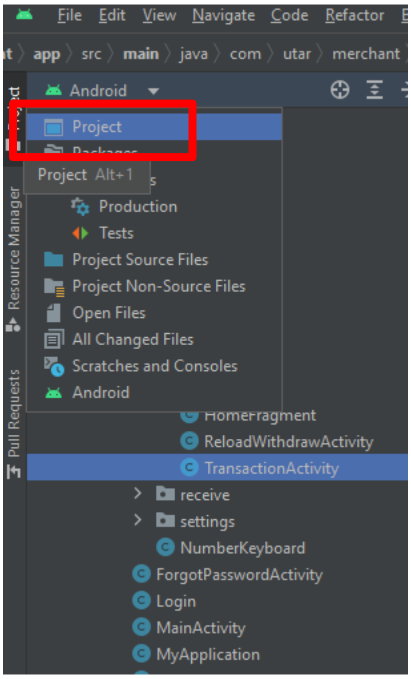
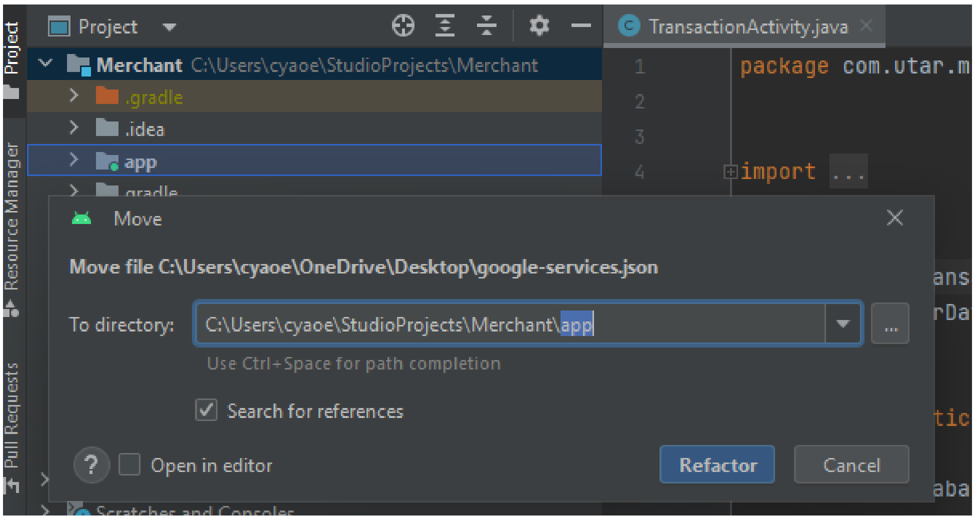


Figure 5.1.2.3 Project Configuration

Copy the Google Firebase configuration file downloaded in Figure 5.1.1.11 and paste it into the app directory.



Move

Move file C:\Users\cyaoe\OneDrive\Desktop\google-services.json

To directory: C:\Users\cyaoe\StudioProjects\Merchant\app

Use Ctrl+Space for path completion

☑ Search for references

□ Open in editor

Figure 5.1.2.4 Firebase Configuration File Locate

All the configurations are finished until here, the application is ready to be built.

## 5.2 Implementation Details

This section will mainly discuss the implementation of the NFC module since the resource is quite limited to be researched. This includes the details on how to implement the NFC, attached with the partial source code.

### 5.2.1 NFC Module Integration

The NFC module is the most important component of this proposed development. NFC comes with a lot of modes and functionalities, such as payments, data transfer, or access control. In this project, it is utilized in payment, receive payment, fund transfer and NFC login functions. Since it has the capability to emulate a card service and becomes an NFC card reader to perform the data exchange. Therefore, this section outlines the steps and the implementation method to integrate the whole system with the functionality of NFC.

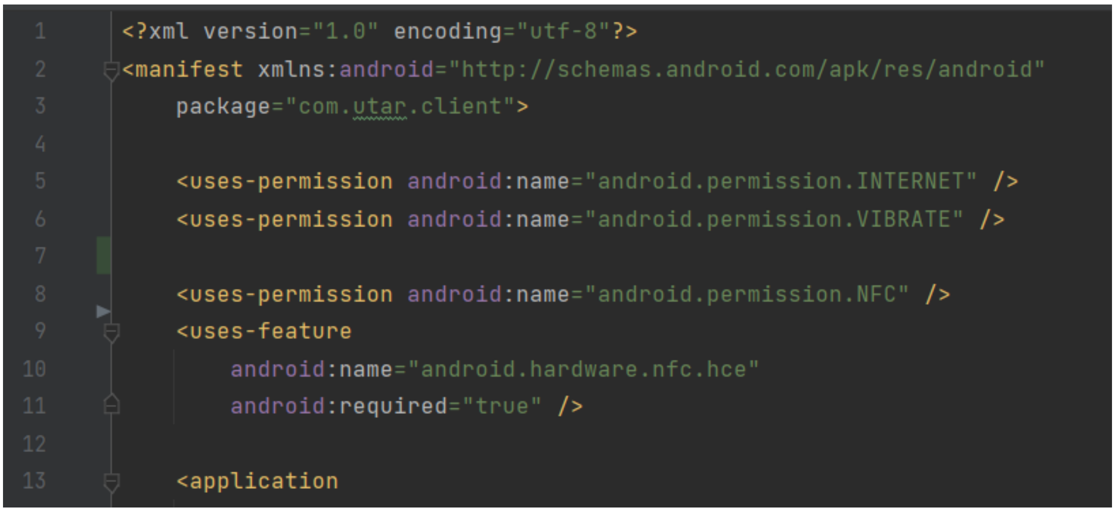
#### 5.2.1.1 Hardware

The mobile device must have a built-in NFC module. This NFC module will be used throughout the development and the software testing.

### 5.2.1.2 Permission

To implement the NFC module, some permissions are necessary to be requested. Only the user grants permission to use the NFC module, as well as the NFC HCE function.

In Android, permission is required to be requested inside the AndroidManifest.xml file. The following figure explains how to request NFC permission.



```
1   <?xml version="1.0" encoding="utf-8"?>
2   <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3       package="com.utar.client">
4
5       <uses-permission android:name="android.permission.INTERNET" />
6       <uses-permission android:name="android.permission.VIBRATE" />
7
8       <uses-permission android:name="android.permission.NFC" />
9       <uses-feature
10          android:name="android.hardware.nfc.hce"
11          android:required="true" />
12
13      <application
```

Figure 5.2.1.2 Permission in AndroidManifest.xml

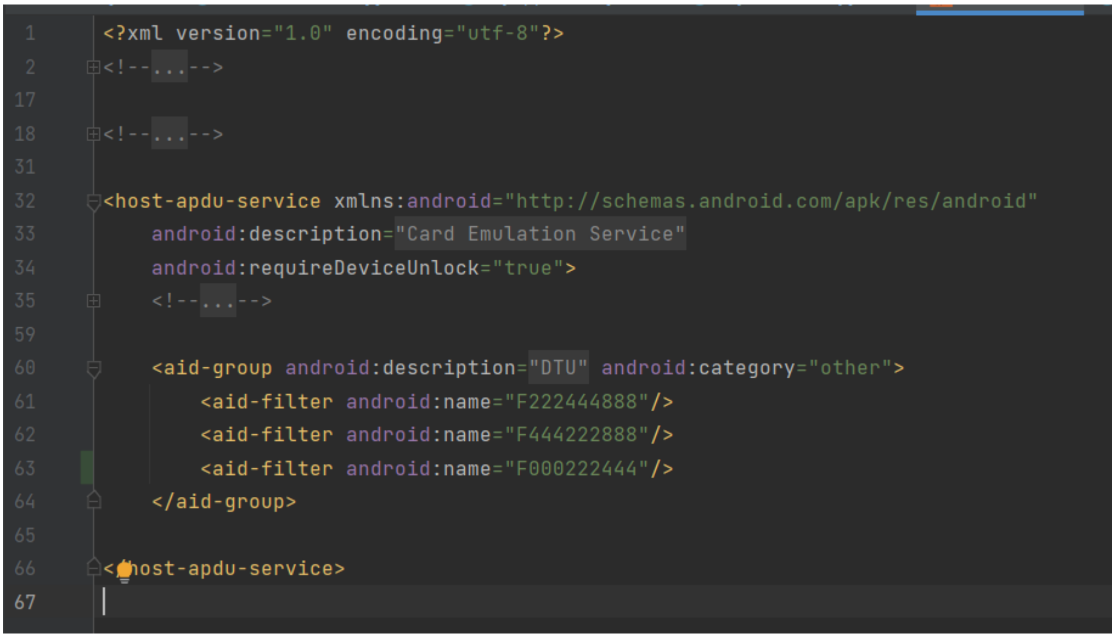
Line 8 is to request the NFC module that implements the NFC reader mode.

Lines 9 to 11 is to request the NFC HCE function permission to emulate the card service.

### 5.2.1.3 AID List Declaration

The AID list declaration is used to specify which applications or services should handle NFC transactions when a specific NFC card or NFC-enabled device with a binding NFC service application is detected.

AID lists are commonly used in contactless smart car systems, mobile payments and other NFC applications. The following figure shows how to declare the AID list in Android. To strengthen the main point, AID is used for the card emulator service.

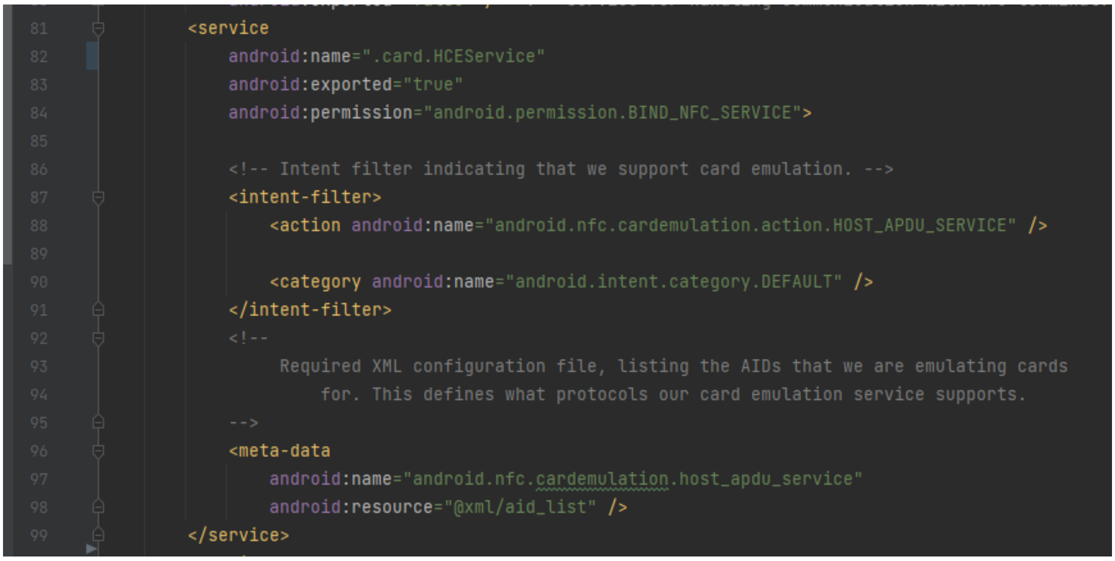


```
1   <?xml version="1.0" encoding="utf-8"?>
2   <!--...-->
17
18  <!--...-->
31
32  <host-apdu-service xmlns:android="http://schemas.android.com/apk/res/android"
33      android:description="Card Emulation Service"
34      android:requireDeviceUnlock="true">
35      <!--...-->
59
60      <aid-group android:description="DTU" android:category="other">
61          <aid-filter android:name="F222444888"/>
62          <aid-filter android:name="F444222888"/>
63          <aid-filter android:name="F000222444"/>
64      </aid-group>
65
66  </host-apdu-service>
67
```

Figure 5.2.1.3 AID Lists Declaration in aid_list.xml

Firstly, this declaration file has to be declared under the “res” directory and then create a directory named “xml”. In this directory, create the aid_list.xml to declare the AID list.

In lines 61 to 63, the AID to be used needs to be declared within the <*aid-group*> tag. Otherwise, the NFC reader will fail to communicate the application that is designed to connect to the corresponding NFC reader.



```
<service
    android:name=".card.HCEService"
    android:exported="true"
    android:permission="android.permission.BIND_NFC_SERVICE">

    <!-- Intent filter indicating that we support card emulation. -->
    <intent-filter>
        <action android:name="android.nfc.cardemulation.action.HOST_APDU_SERVICE" />

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
    <!--
         Required XML configuration file, listing the AIDs that we are emulating cards
             for. This defines what protocols our card emulation service supports.
    -->
    <meta-data
        android:name="android.nfc.cardemulation.host_apdu_service"
        android:resource="@xml/aid_list" />
</service>
```

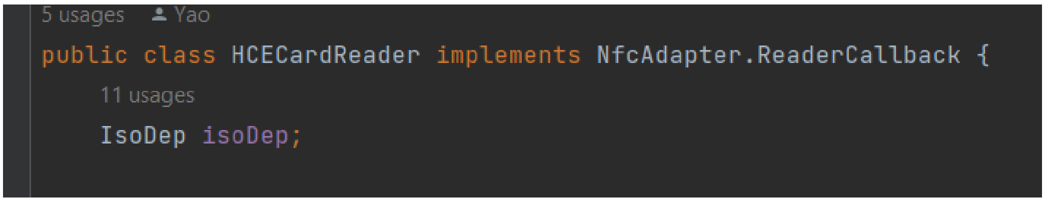
Figure 5.2.1.4 HCE Service Declaration in AndroidManifest.xml

Other than the AID declaration, the NFC HCE services must be also declared inside the AndroidManifest.xml. This is because NFC HCE is a background-based service, and it should be automatically started once the application is launched. Since the service has to bind with the NFC feature, so the declaration is compulsory in AndroidManifest.xml.

### 5.2.1.4 Source Code Implementation

This section will explain how the source code implements the NFC reader mode and NFC HCE mode.

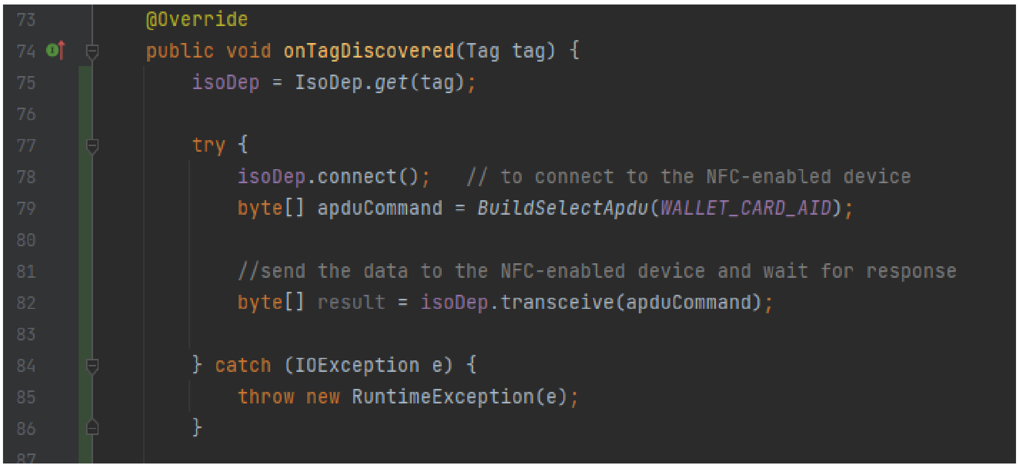
- **NFC Reader**



```
5 usages  Yao
public class HCECardReader implements NfcAdapter.ReaderCallback {
    11 usages
    IsoDep isoDep;
```

Figure 5.2.1.5 Reader Java Class Implements Interface

First of all, the NFC reader java class must implement **NfcAdapter.ReaderCallback** interface, and override the method from the interface which is the onTagDiscovered () method. This method is used to handle while a new tag is found, and then to perform the related response and communication to the object.

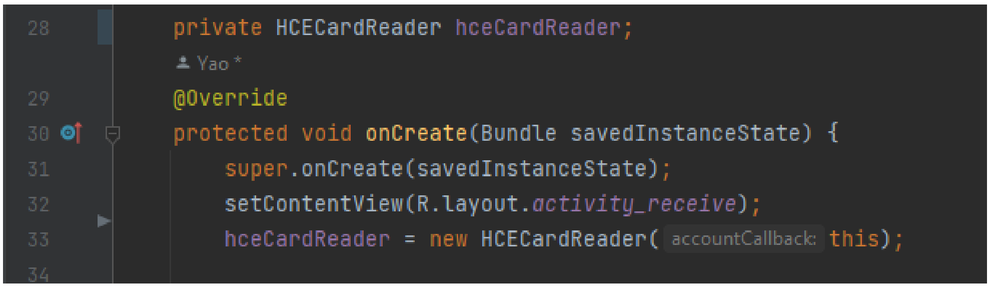


```
@Override
public void onTagDiscovered(Tag tag) {
    isoDep = IsoDep.get(tag);

    try {
        isoDep.connect();   // to connect to the NFC-enabled device
        byte[] apduCommand = BuildSelectApdu(WALLET_CARD_AID);

        //send the data to the NFC-enabled device and wait for response
        byte[] result = isoDep.transceive(apduCommand);

    } catch (IOException e) {
        throw new RuntimeException(e);
    }
```

Figure 5.2.1.6 Partial Source Code of Reader Callback Interface Method Override

Figure 5.2.1.6 shows an example of how to communicate with an NFC-enabled device or an NFC tag.

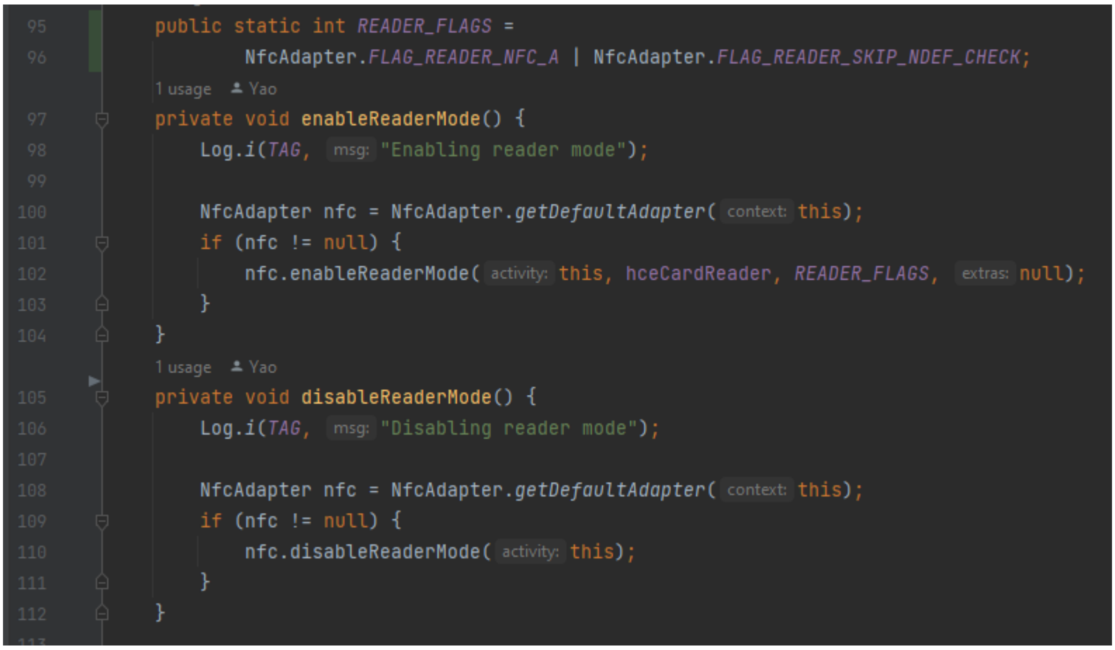
First of all, get the tag object, then use the class to connect to the tag object. Then build an APDU packet send it to the tag and wait for the response using the transceive() method.



```
    private HCECardReader hceCardReader;
    Yao *
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_receive);
        hceCardReader = new HCECardReader( accountCallback: this);
```

Figure 5.2.1.7 Partial Source Code of NFC Reader Mode Activity

Inside the activity source code of implementing the NFC card reader mode is required to declare an instance of the NFC card reader class for enable or disable the NFC card reader mode purposes.



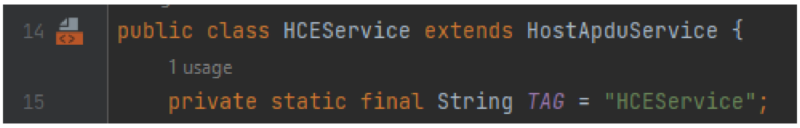
```
    public static int READER_FLAGS =
            NfcAdapter.FLAG_READER_NFC_A | NfcAdapter.FLAG_READER_SKIP_NDEF_CHECK;
    1 usage  Yao
    private void enableReaderMode() {
        Log.i(TAG,  msg: "Enabling reader mode");

        NfcAdapter nfc = NfcAdapter.getDefaultAdapter( context: this);
        if (nfc != null) {
            nfc.enableReaderMode( activity: this, hceCardReader, READER_FLAGS,  extras: null);
        }
    }
    1 usage  Yao
    private void disableReaderMode() {
        Log.i(TAG,  msg: "Disabling reader mode");

        NfcAdapter nfc = NfcAdapter.getDefaultAdapter( context: this);
        if (nfc != null) {
            nfc.disableReaderMode( activity: this);
        }
    }
```

Figure 5.2.1.8 Partial Source Code of NFC Reader Mode Enable and Disable

In the activity, the NFC card reader mode must be enabled or disabled using the method inside the NfcAdapter object to perform. Those are disableReaderMode() and enableReaderMode().

The NFC data exchange only can be performed once the NFC reader mode is enabled.

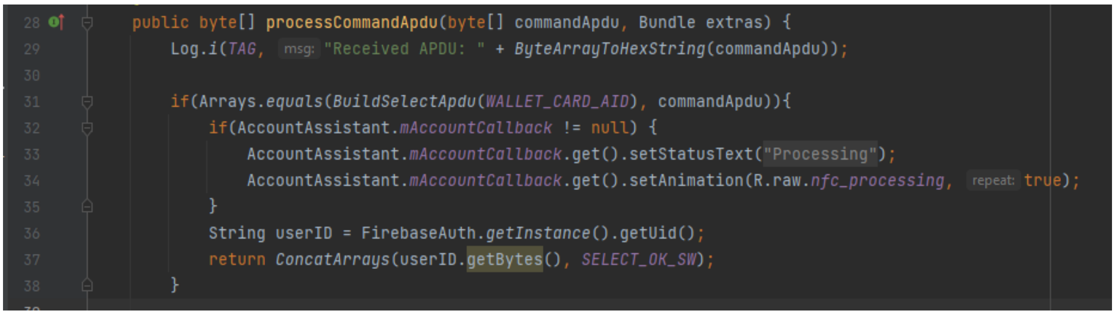
- **NFC HCE**



```
public class HCEService extends HostApduService {
    1 usage
    private static final String TAG = "HCEService";
```

Figure 5.2.1.9 HCE Service Class

To allow a Java class to implement HCE service, the class must extend the HostApduService class, and then override the abstract method which is processCommandApdu().



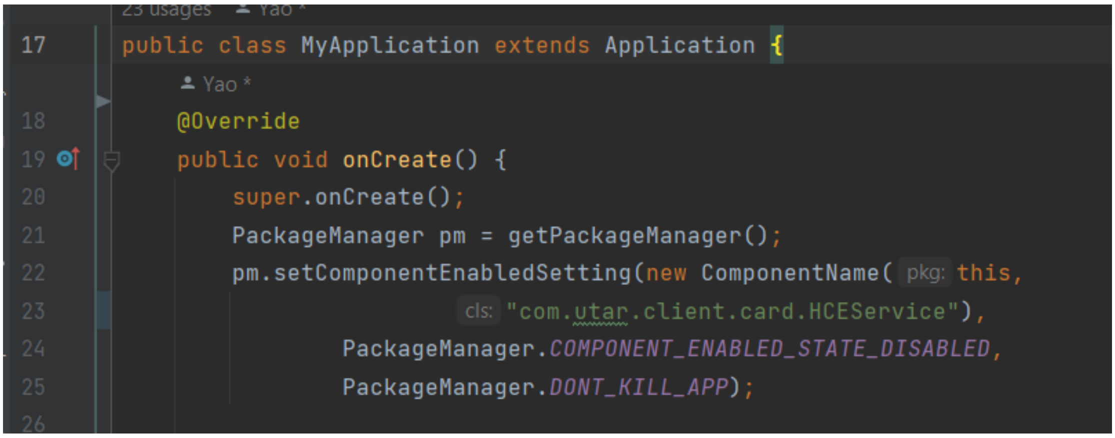
```
public byte[] processCommandApdu(byte[] commandApdu, Bundle extras) {
    Log.i(TAG, msg: "Received APDU: " + ByteArrayToHexString(commandApdu));

    if(Arrays.equals(BuildSelectApdu(WALLET_CARD_AID), commandApdu)){
        if(AccountAssistant.mAccountCallback != null) {
            AccountAssistant.mAccountCallback.get().setStatusText("Processing");
            AccountAssistant.mAccountCallback.get().setAnimation(R.raw.nfc_processing, repeat: true);
        }
        String userID = FirebaseAuth.getInstance().getUid();
        return ConcatArrays(userID.getBytes(), SELECT_OK_SW);
    }
```

Figure 5.2.1.10 Partial Source Code of the processCommandApdu()

Once this NFC HCE-enabled device approaches an NFC card reader, the HCE device will detect and select the corresponding application using the AIDs to perform data exchange. So, the processCommandApdu() will be called by the system.

### 5.2.1.5 NFC HCE Service Binding

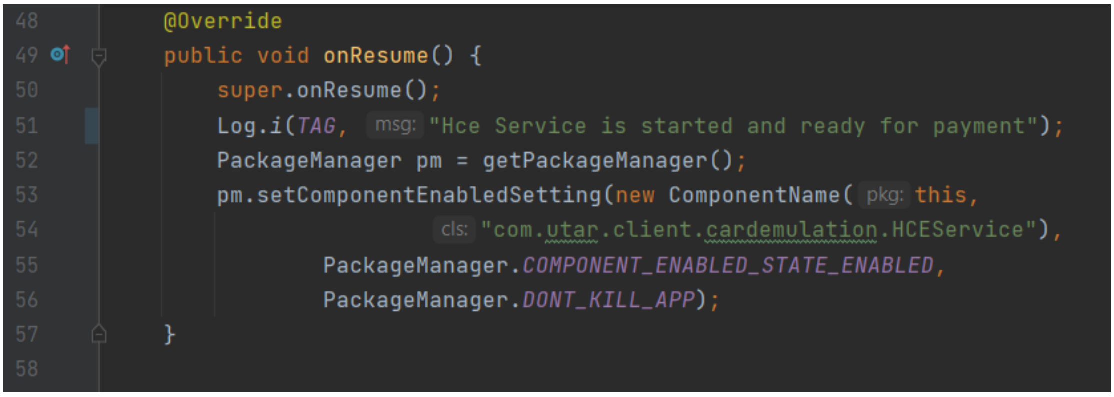
As mentioned in the weakness of the NFC HCE in the Chapter 2 literature review, there is a method to 5.2.1.11figure out the problem.



```
23 usages   Yao *
public class MyApplication extends Application {
        Yao *
        @Override
        public void onCreate() {
            super.onCreate();
            PackageManager pm = getPackageManager();
            pm.setComponentEnabledSetting(new ComponentName( pkg: this,
                                  cls: "com.utar.client.card.HCEService"),
                    PackageManager.COMPONENT_ENABLED_STATE_DISABLED,
                    PackageManager.DONT_KILL_APP);
```

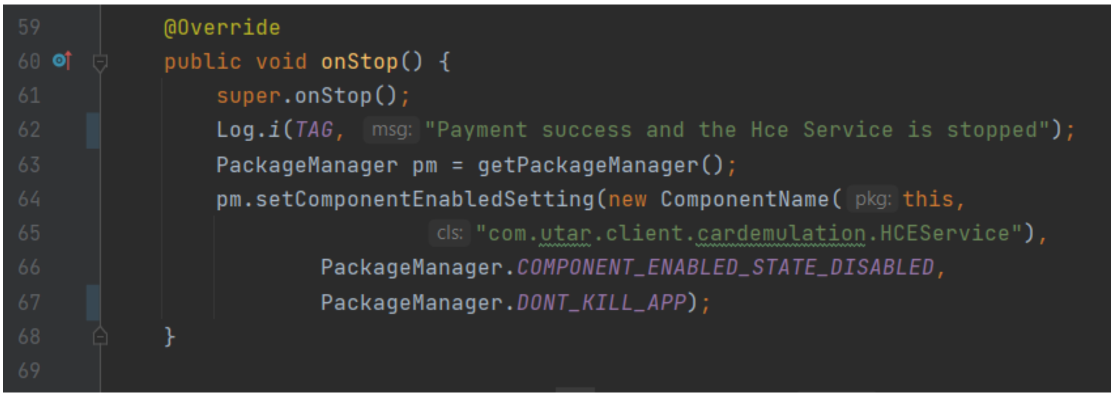
Figure 5.2.1.11 Partial Source Code of Initialization on Unbinding NFC HCE Service

The figure above shows showing to unbind the NFC service to this current application. Once the application is started up by the system or the user, the NFC service will be stopped at the entry point of this application. This can prevent unaware fraud from occurring when the application is taken up by the Android host system.



```
    @Override
    public void onResume() {
        super.onResume();
        Log.i(TAG, msg: "Hce Service is started and ready for payment");
        PackageManager pm = getPackageManager();
        pm.setComponentEnabledSetting(new ComponentName( pkg: this,
                        cls: "com.utar.client.cardemulation.HCEService"),
                PackageManager.COMPONENT_ENABLED_STATE_ENABLED,
                PackageManager.DONT_KILL_APP);
    }
```

Figure 5.2.1.12 Partial Source Code of Binding NFC Card Emulation Service Before NFC Payment



```
    @Override
    public void onStop() {
        super.onStop();
        Log.i(TAG, msg: "Payment success and the Hce Service is stopped");
        PackageManager pm = getPackageManager();
        pm.setComponentEnabledSetting(new ComponentName( pkg: this,
                        cls: "com.utar.client.cardemulation.HCEService"),
                PackageManager.COMPONENT_ENABLED_STATE_DISABLED,
                PackageManager.DONT_KILL_APP);
    }
```

Figure 5.2.1.13 Partial Source Code of Unbinding NFC Card Emulation Service After Payment is Done

Figure 5.2.1.12 and Figure 5.2.1.13 show that the NFC service will be disabled or enabled from the application. The two partial source codes above aim to prevent the NFC transaction from being processed when the activity is not running in the foreground currently.

## 5.3 System Operation

### 5.3.1 User Login Module

The user interface will be divided into three sections, where the activity is to be based on availability for both roles, for the merchant role, and for the client role.

**User Interface for both Roles (Merchant and Client)**

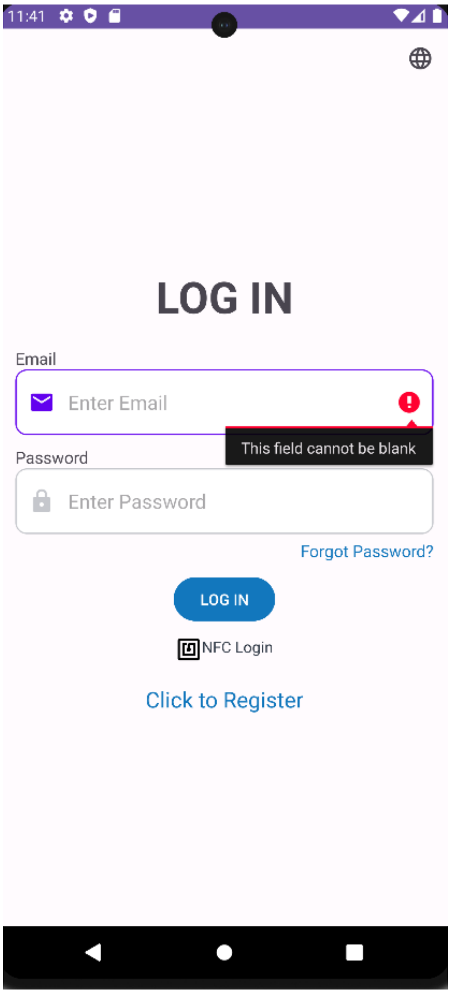


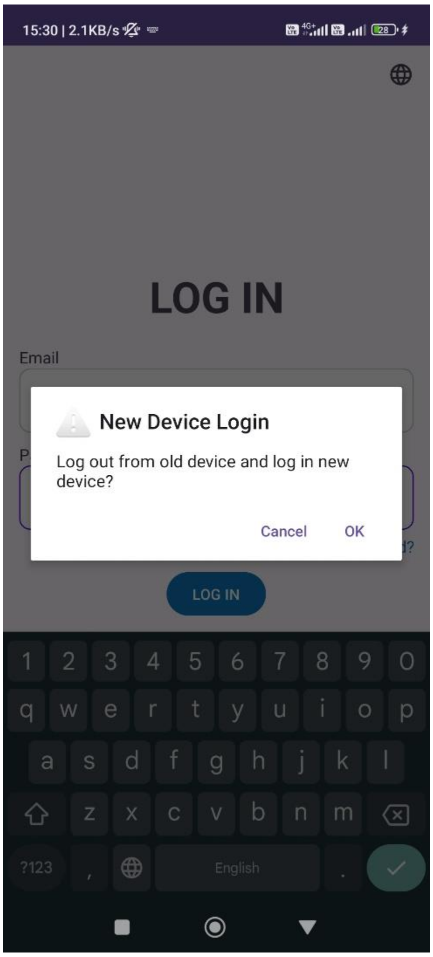
Figure 5.3.1 Log-in Interface

When the moment users launch the application, the login interface shown in Figure 5.3.1 will be first displayed. The user enters a valid email and password to authenticate and get access to the application. If the user with an existing email but forgot his/her password, they can reset their password using the "*Forgot Password?*" to redirect them to the forgot password interface to do the related activity.

Other than email and password login, the unauthenticated user can select to use the "*NFC Login*" to log in to the system.
There is a "*Click to Register*" keyword to let the new user who would like to create a new account access this application.

With exception handling, when there is any field has not filled, the corresponding field will prompt an error message.

This login interface will be only displayed when users never log in or if they did log in but have not logged out from this application. In other words, it is only displayed when the account authentication is not granted.



New Device Login

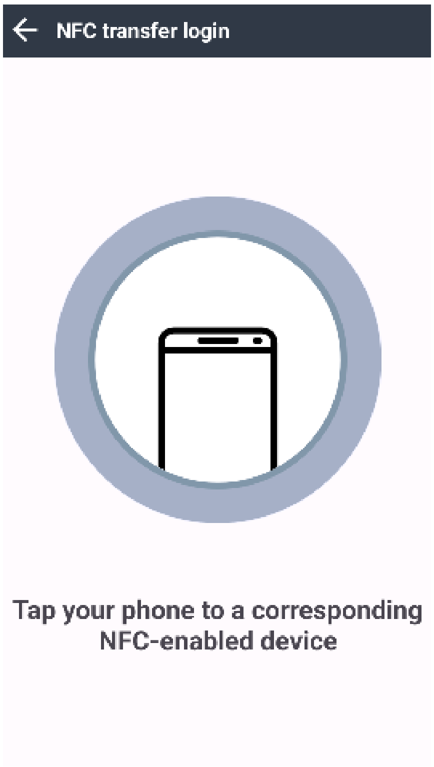
Log out from old device and log in new device?

Figure 5.3.2 New Device Login Confirmation

As one of the important for the system, the account is only bound to one device. Once a new device login the account, the account will be automatically logged out from the old device.

This will be done if the user selects “OK” with the confirmation window in Figure 5.3.2.

### 5.3.2 NFC Login Module



← NFC transfer login

Tap your phone to a corresponding NFC-enabled device

Figure 5.3.3 NFC Login

Figure 5.3.3 shows the NFC login interface, it allows the authenticated user to transfer their account from an old device to a new device. This makes the user convenient in account transfer.

### 5.3.3 Registration Module

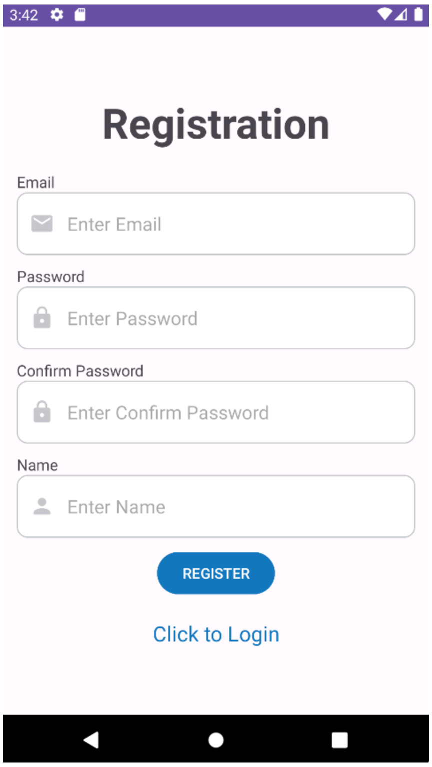


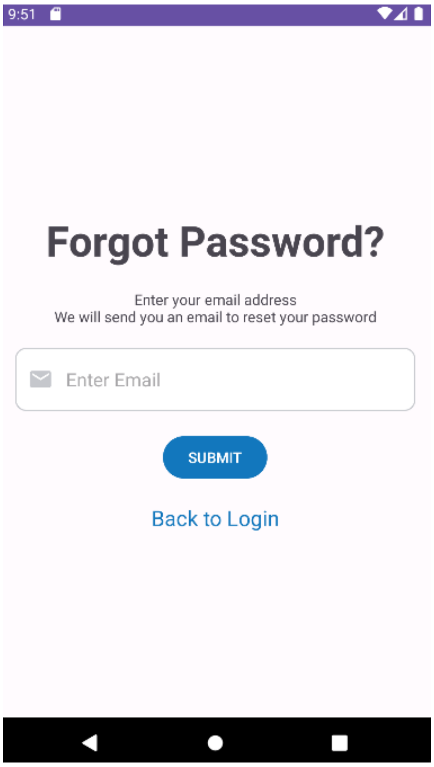
Figure 5.3.4 Registration Interface

The layout above shows the interface that lets the user register an account to gain access to this application.

In this registration interface, the user is required to enter the email, password, confirmed password, and name. Once all the information is entered, the user needs to press the "*REGISTER*" button to continue. Only the valid account, such as the account does not exist in the database or both the password and confirmed password are matched, the account is then only successfully registered. Otherwise, an error message will be displayed with the corresponding field.

In this interface, if the user intends to log in account, they can click the blue colour message text which shows "*Click to Login*" and proceed to the login interface.

### 5.3.4 Forgot Password Module



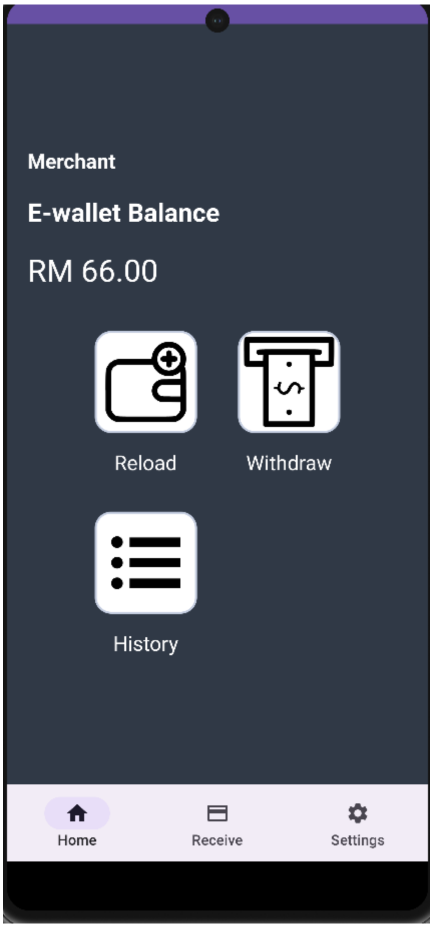
# Forgot Password?

Enter your email address
We will send you an email to reset your password

Figure 5.3.5 Forgot Password Interface

The figure above shows the interface that allows user to reset their password. It requires the user to enter a valid email address, and then click the *"SUBMIT"* button. After that, the user needs to check their mailbox and follow the instructions given to reset their password.

### 5.3.5 Home Page Module

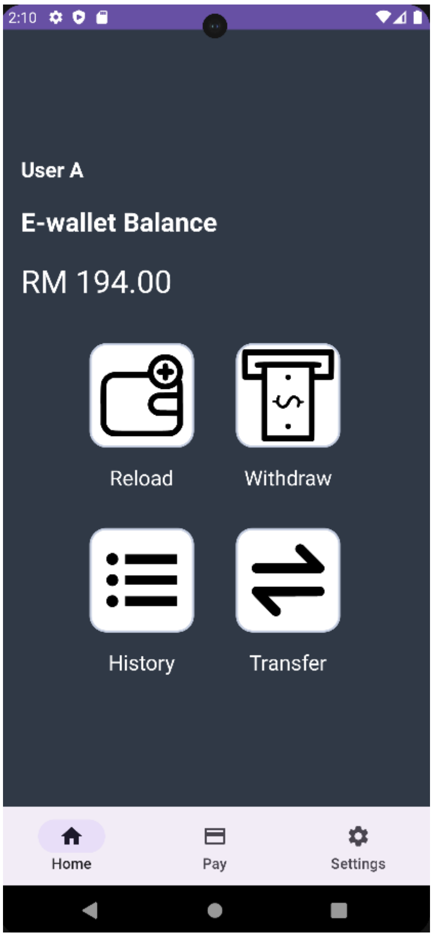


Merchant

E-wallet Balance

RM 66.00

Figure 5.3.6 Merchant Main Page



User A

E-wallet Balance

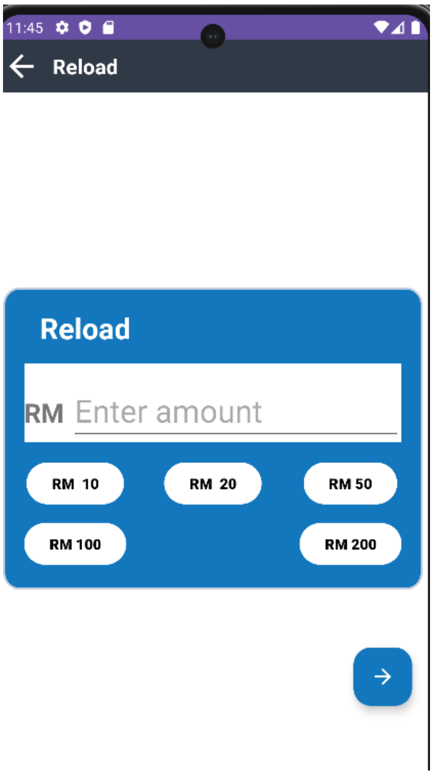
RM 194.00

Figure 5.3.7 Client Main Page

Both figures 5.3.6 and figure 5.3.7 show the home page based on the role of their account. It included the name of the user, e-wallet balance, and corresponding activity directories.

Both have the same functions to reload, balance withdraw, history trace back, and settings fragment. However, the merchant-side application mainly focuses on having the Payment Receive function while the client-side with the Payment function with one more fund transfer function.

### 5.3.6 Reload/Withdrawal Module



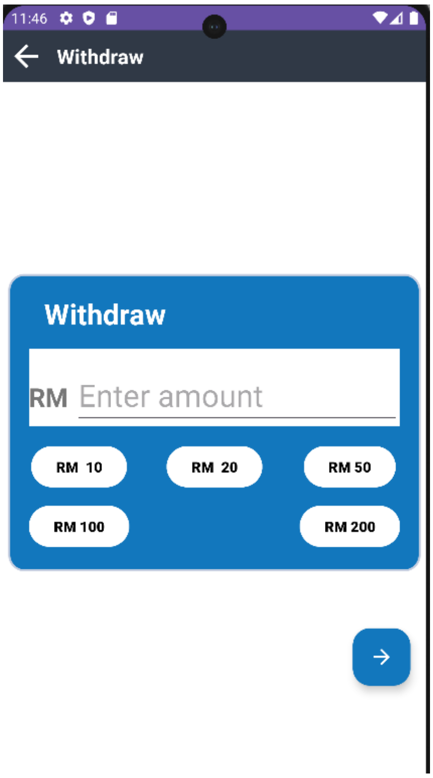


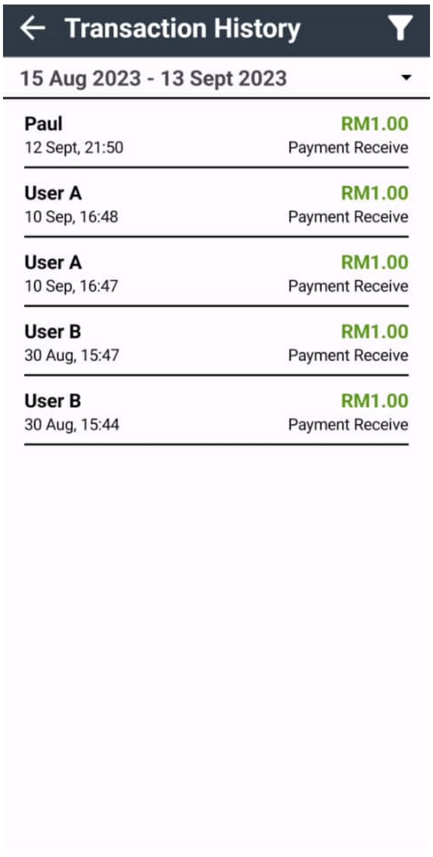
Figure 5.3.8 Reload Interface 5.3.9 Withdrawal Interface

Both figures above allow the user the perform the balance reloads or withdrawals.

User can use the shortcut key to complete their balance reload or withdrawal function.

The reload function is to top up the amount into the balance while the withdraw is to withdraw the amount from the account balance.

### 5.3.7 Transaction Module



Transaction History

15 Aug 2023 - 13 Sept 2023

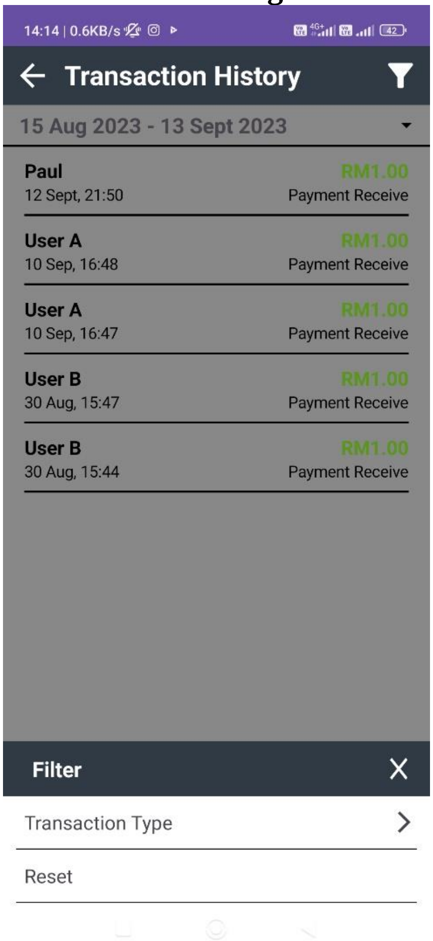
| | |
|---|---|
| **Paul**<br>12 Sept, 21:50 | **RM1.00**<br>Payment Receive |
| **User A**<br>10 Sep, 16:48 | **RM1.00**<br>Payment Receive |
| **User A**<br>10 Sep, 16:47 | **RM1.00**<br>Payment Receive |
| **User B**<br>30 Aug, 15:47 | **RM1.00**<br>Payment Receive |
| **User B**<br>30 Aug, 15:44 | **RM1.00**<br>Payment Receive |

Figure 5.3.10 Transaction History Interface

This is the interface that allows the user to view the transaction history. All the records will be in descending order, which means the recent record will be displayed at the top, and the oldest record will be the last. The displayed transaction records are only available with the last 30 days' transactions with default mode.

Other than that, the user is allowed to filter the transaction records. This can be functioned with the top section indicated.

### 5.3.8 Transaction Searching Module



14:14 | 0.6KB/s

Transaction History

15 Aug 2023 - 13 Sept 2023

Paul RM1.00
12 Sept, 21:50 Payment Receive

User A RM1.00
10 Sep, 16:48 Payment Receive

User A RM1.00
10 Sep, 16:47 Payment Receive

User B RM1.00
30 Aug, 15:47 Payment Receive

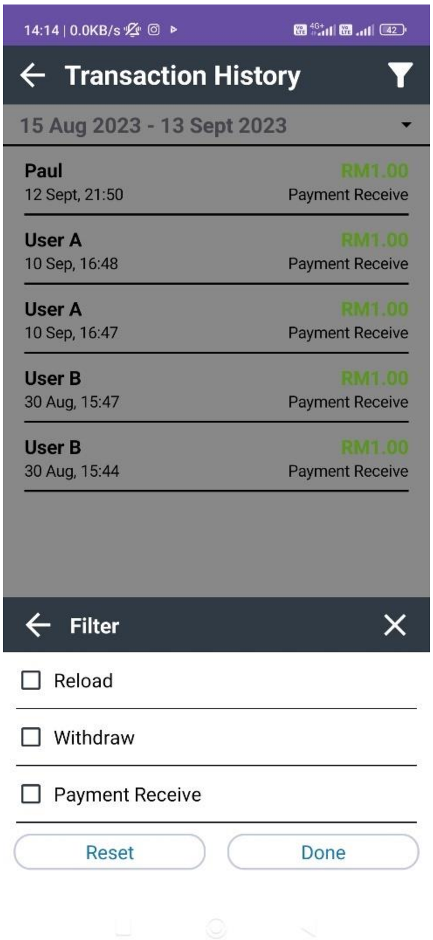
User B RM1.00
30 Aug, 15:44 Payment Receive

Filter

Transaction Type

Reset

Figure 5.3.11 Filter Type Selection



14:14 | 0.0KB/s

Transaction History

15 Aug 2023 - 13 Sept 2023

Paul RM1.00
12 Sept, 21:50 Payment Receive

User A RM1.00
10 Sep, 16:48 Payment Receive

User A RM1.00
10 Sep, 16:47 Payment Receive

User B RM1.00
30 Aug, 15:47 Payment Receive

User B RM1.00
30 Aug, 15:44 Payment Receive

Filter

□ Reload

□ Withdraw

□ Payment Receive

Reset Done

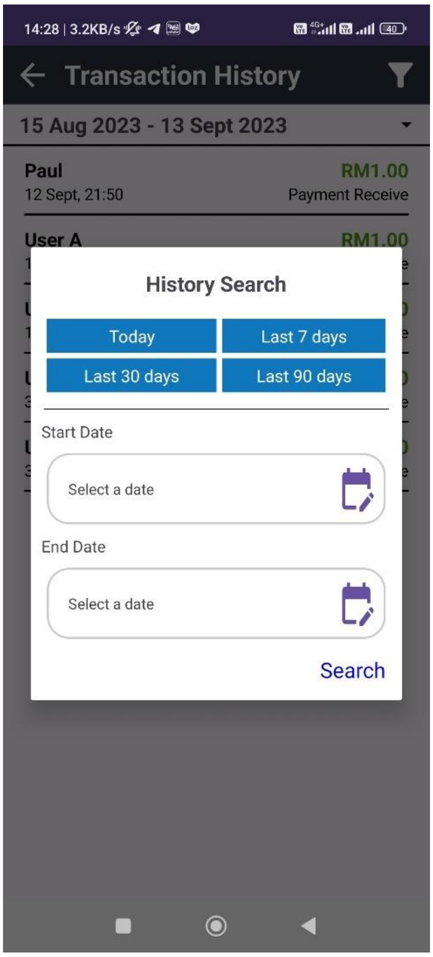
Figure 5.3.12 Transaction Type Filtering

Both figures above are used to perform the advanced searching function.

In Figure 5.3.11, there are two options. The first option allows the user can select the specific type of transaction to filter the transaction record they want. The options in Figure 5.3.12 can be multiple-selected.

The following option is “Reset” in Figure 5.3.11 which allows the user to back to the default transaction record display with the last 30 days’ transactions.

After the selection is done, these prompted windows will disappear and the list will be refreshed.



Transaction History

15 Aug 2023 - 13 Sept 2023

Paul RM1.00
12 Sept, 21:50 Payment Receive

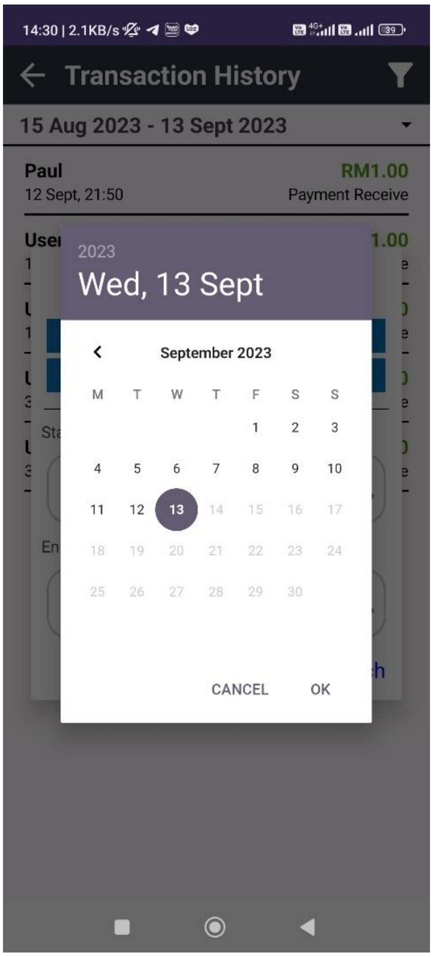
User A RM1.00

History Search

Start Date: Select a date

End Date: Select a date

Figure 5.3.13 Specify Date Transaction Searching Window



Transaction History

15 Aug 2023 - 13 Sept 2023

Paul RM1.00
12 Sept, 21:50 Payment Receive

2023
Wed, 13 Sept

September 2023

| M | T | W | T | F | S | S |
|---|---|---|---|---|---|---|
|  |  |  |  | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 | 30 |  |

Figure 5.3.14 Date Picker Floating Window

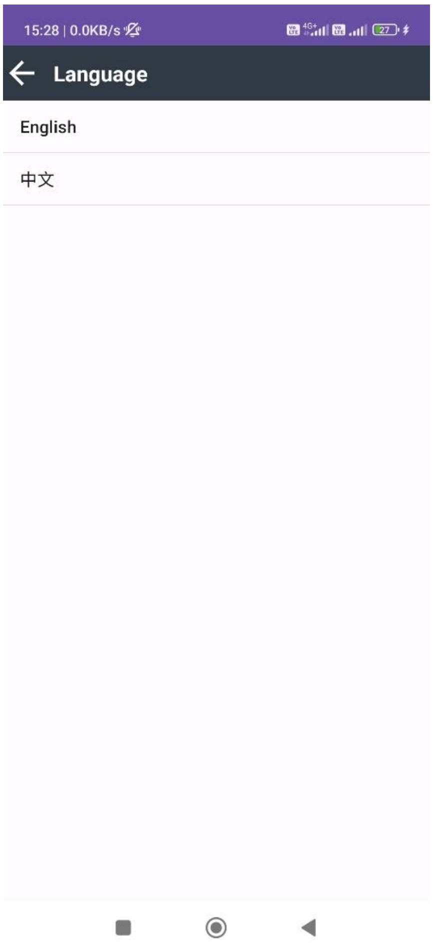
The prompted window shown in Figure 5.3.13 can be triggered by the date range in transaction history activity.

There are four shortcut buttons that quickly specify the range of the date, including only search for today's transaction records, last 7 days, last 30 days, or last 90 days.

Other than that, the user can manually specify the date range with the help of the date picker.

Finally, the user clicks the “*Search*” button to view the transaction within the range that has been set.

### 5.3.9 Language Changing Module



15:28 | 0.0KB/s

Language

English

中文

Figure 5.3.15 Display Language Changing Interface

This system is not restricted to only English language display, there is also Mandarin available to be selected as system displaying language.

Once the language is selected, the system will be restarted and based on the selected language initialize the user interface.

### 5.3.10 PIN Authentication Module

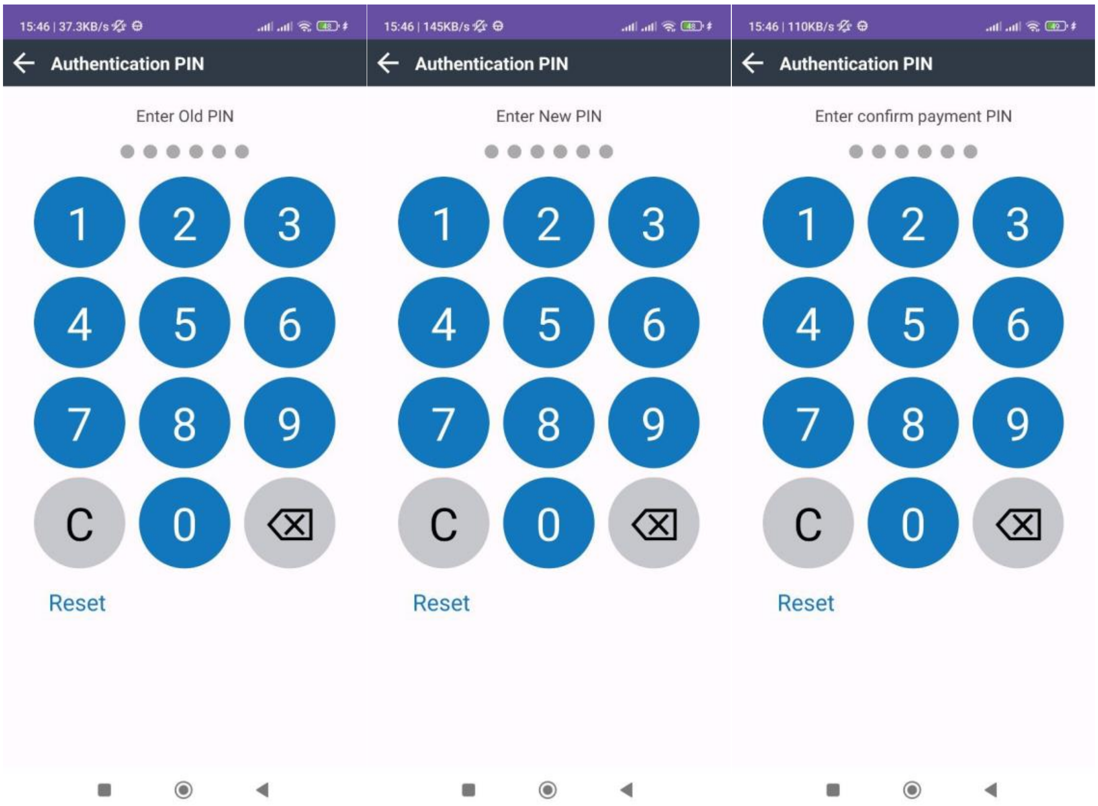


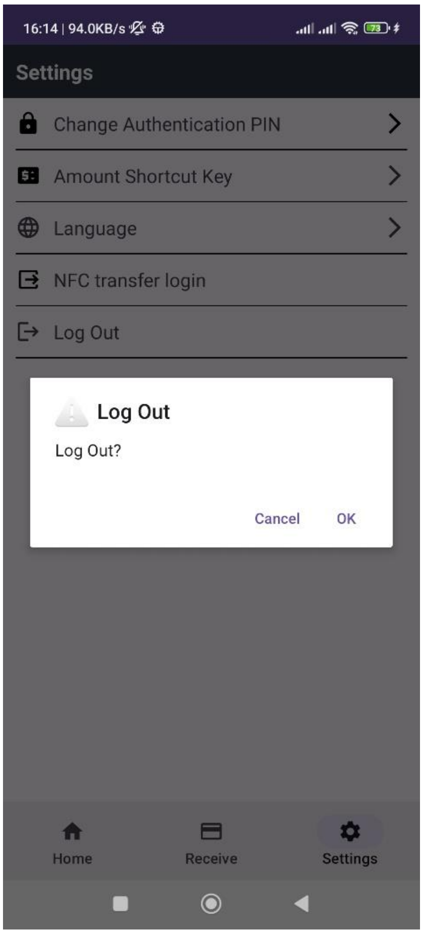
Figure 5.3.16 Authentication PIN Modification Interface

The user is allowed to modify their authentication PIN at any time. Generally, the user is required to enter their original PIN or old PIN for verification purposes. If the user fails to verify them, the authentication PIN is not allowed to modify.

If they are able to authenticate themselves, the system will allow them to change the authentication PIN with twice confirmation.

The authentication will be utilized where the merchant wishes to withdraw their balance, while the client performs payment, and fund transfer.

### 5.3.11 Logout Module



Log Out

Log Out?

Figure 5.3.17 Logout Confirmation

In the setting fragment, there is a logout available function to let the user log out from the system. Once it is confirmed to log out from the system, the user will be redirected to the login interface for login purposes.

## For Merchant Role

### 5.3.12 Merchant Payment Receiving Module

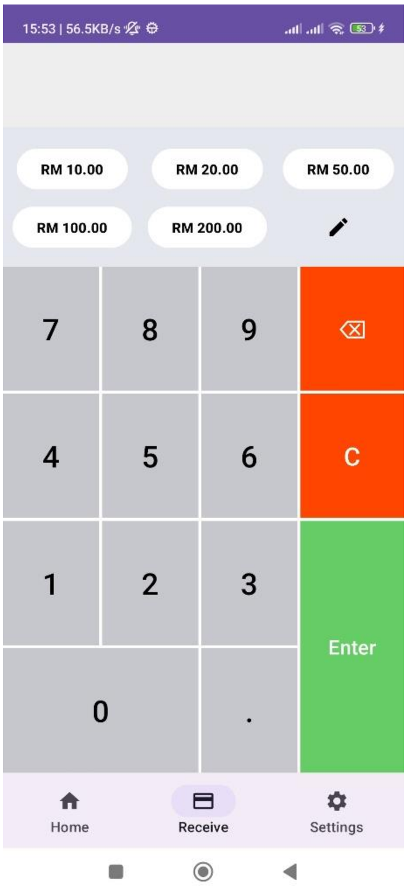


Figure 5.3.18 Payment Amount Input Interface

The figure above shows the interface that allows the merchant to enter the amount which the merchant would like to receive from the client. The merchant can use the amount shortcut key instead of keying the payment amount manually.

Once the merchant completes inputting the amount, the merchant needs to press the *"Enter"* button to redirect to the payment receive interface.

To enter the receiving interface, the NFC module is required to be turned on and available. Otherwise, the merchant will fail to enter the payment receive interface and do not get the function to receive payment.

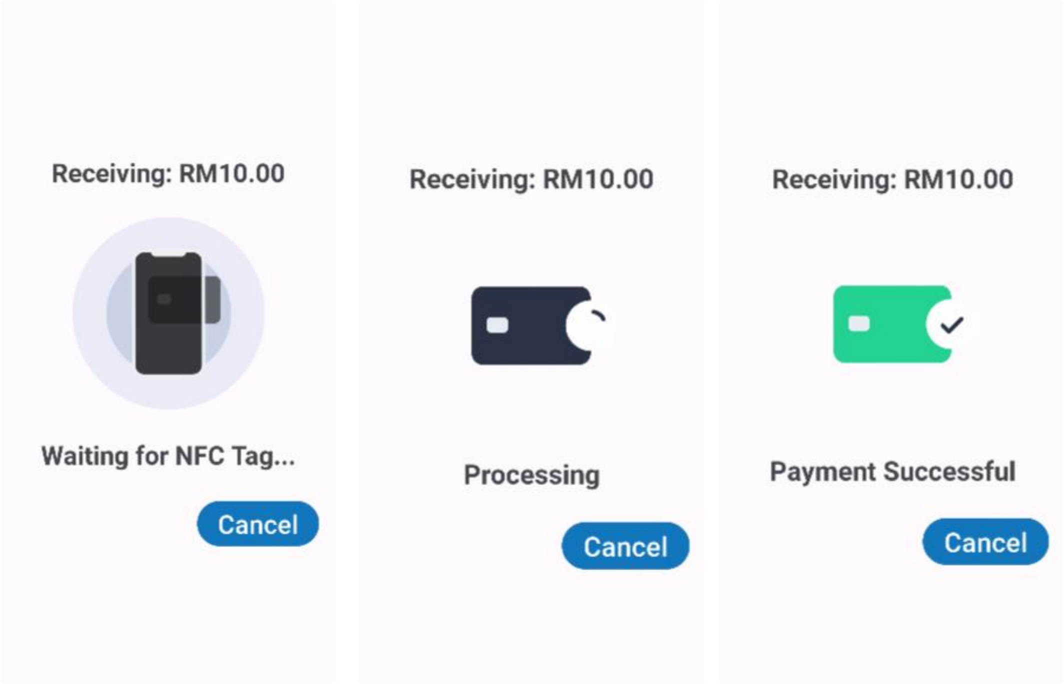
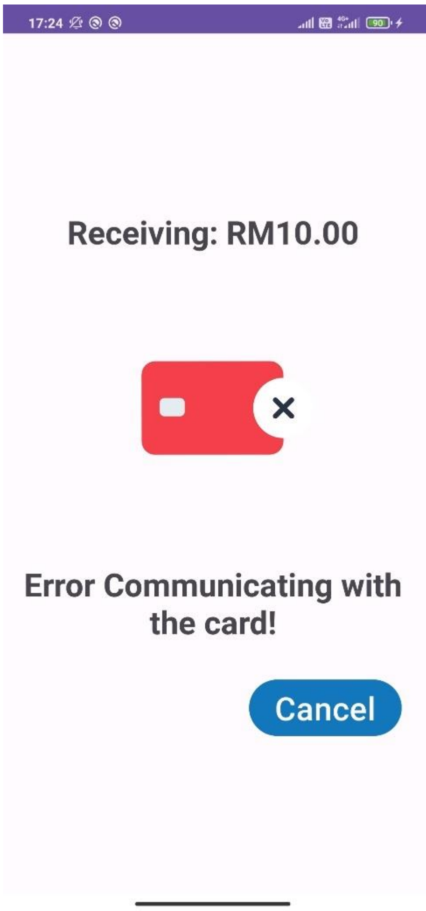


Figure 5.3.19 Merchant Payment Receiving Process

All figures above show the receiving payment process on the merchant side. First, after the receiving amount is entered in the interface shown in Figure 5.3.18 which is confirmed by pressing the "Enter" button, the system then only redirects the user to the waiting NFC interface (as shown in the first step of Figure 5.3.19).

It will wait until the correct NFC card is near and communicate with it. Once the correct card approaches this reader mode, the interface will notify the user the system is currently processing the payment request.

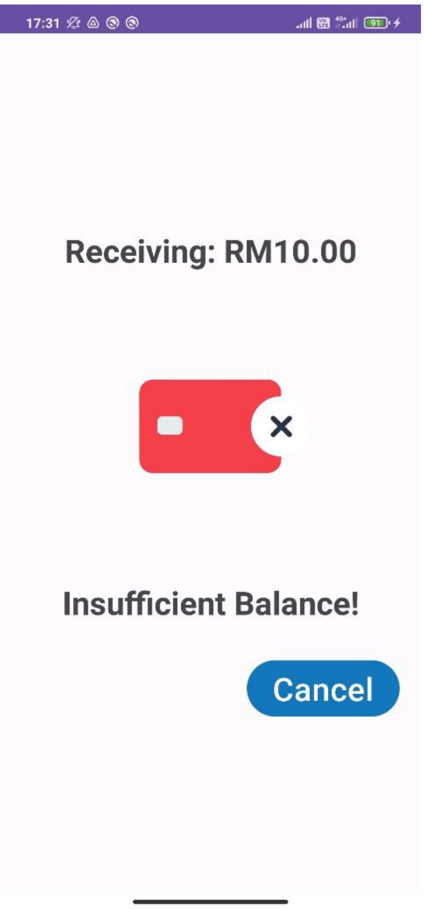
Once the payment is successful, the success notification will be given, it looks exactly like the last step of Figure 5.3.19.



Receiving: RM10.00

Error Communicating with the card!

Figure 5.3.20 Unknown Tag is Communicated Interface



Receiving: RM10.00

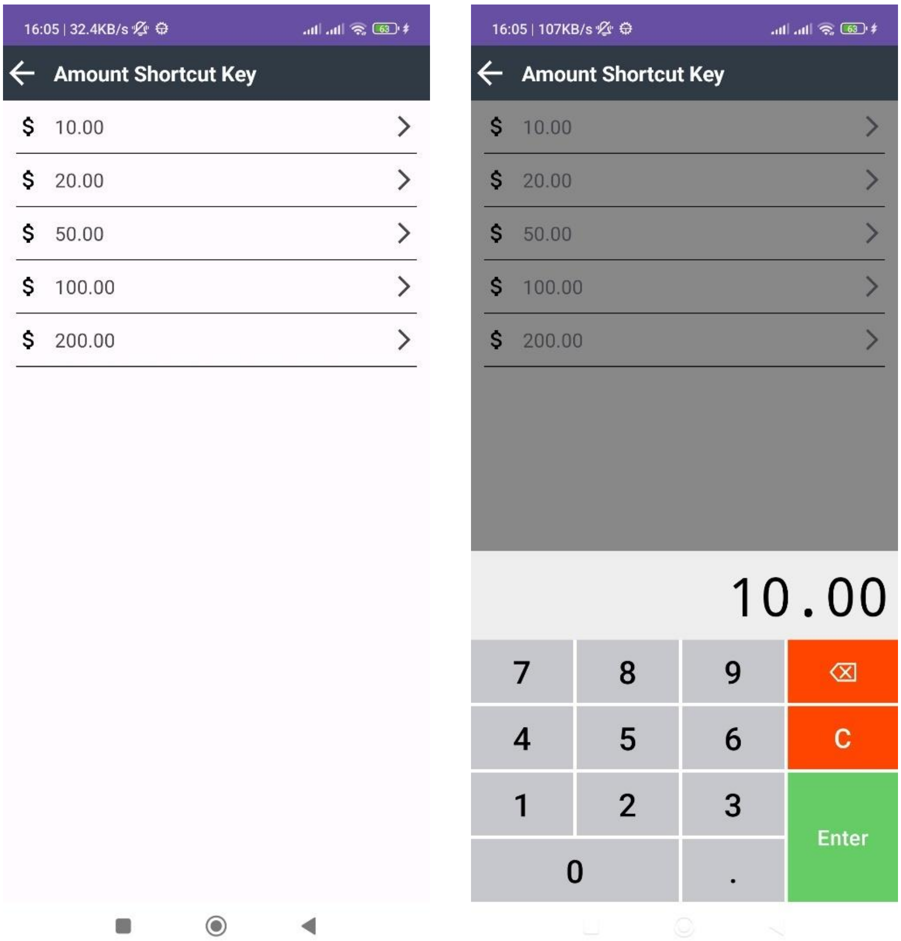
Insufficient Balance!

Figure 5.3.21 Client with Insufficient Balance Interface

Figure 5.3.20 shows that when the reader of the system communicates with an invalid HCE device or the NFC card then the response is given.

Figure 5.3.21 occurs when the client is having a low balance, so it returns the error message to the interface.

### 5.3.13 Amount Shortcut Key Module



Amount Shortcut Key

$ 10.00

$ 20.00

$ 50.00

$ 100.00

$ 200.00

10.00

Figure 5.3.22 Amount Shortcut Key Modification

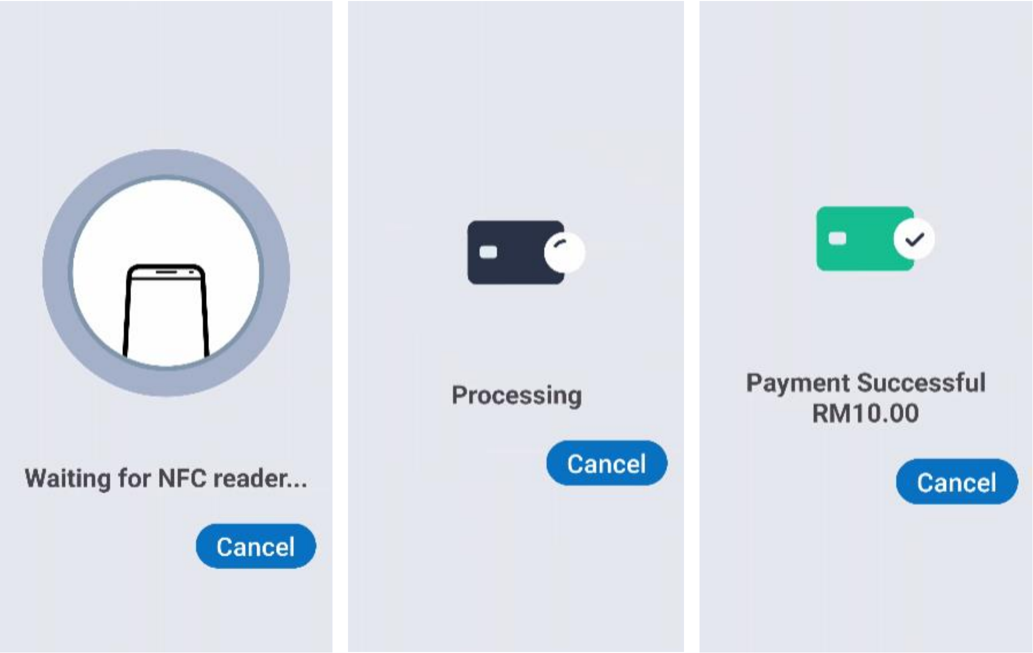
Figure 5.3.18 Payment Amount Input interface comes with the amount shortcut keys that allow the merchant to select the amount instead of keying the amount manually. So that the amount shortcut keys are able to be modified on the merchant's own.

Figure 5.3.22 allows the merchant to modify them accordingly. First of all, it will display all five amounts, then the merchant can select which one to be modified. Then, it will prompt a window with a keyboard to let the merchant modify.

The modification will reflect the amount of shortcut keys in Figure 5.3.18.

**For Client Role**

### 5.3.14 Client NFC Payment Module



Waiting for NFC reader...

Processing

Payment Successful
RM10.00

Figure 5.3.23 Client NFC HCE Payment Process

All figures above show the payment process on the client side.

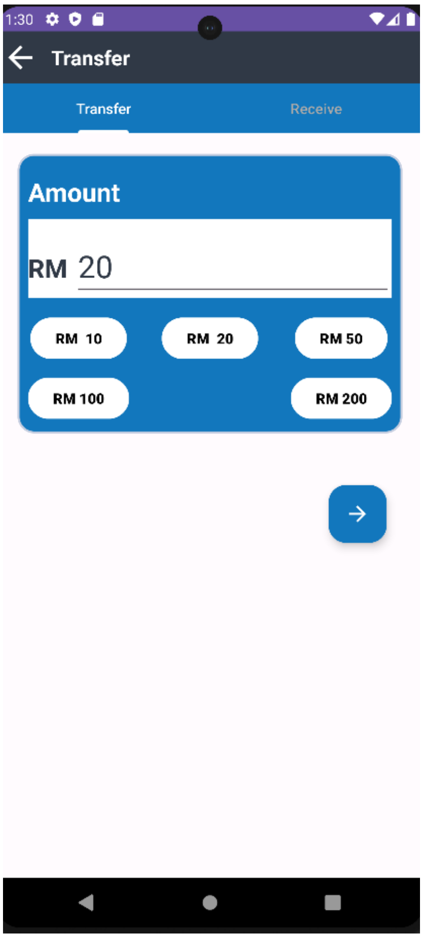
After the user clicks the *"Pay"* button on the home page, the system will show that it is waiting for the NFC reader to perform communication as shown in the first step.

Once an NFC reader is detected, the interface will be displayed as shown in the second step while the system processes the payment.

Once the payment is successful, the interface will notify the user as well as the amount the user has paid.

Otherwise, it will show the error payment with insufficient balance as shown in Figure 5.3.21 without displaying the amount.

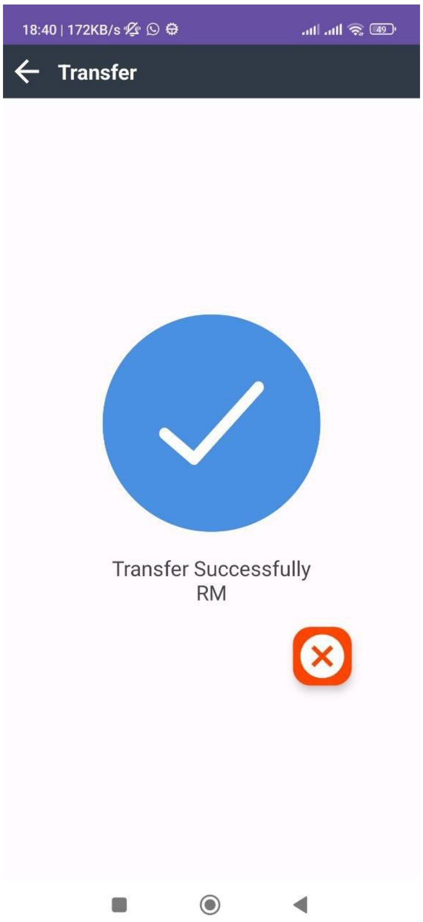
### 5.3.15 Client NFC Fund Transfer Module



Amount

RM 20

Figure 5.3.24 Client Fund Transfer Amount Input Interface



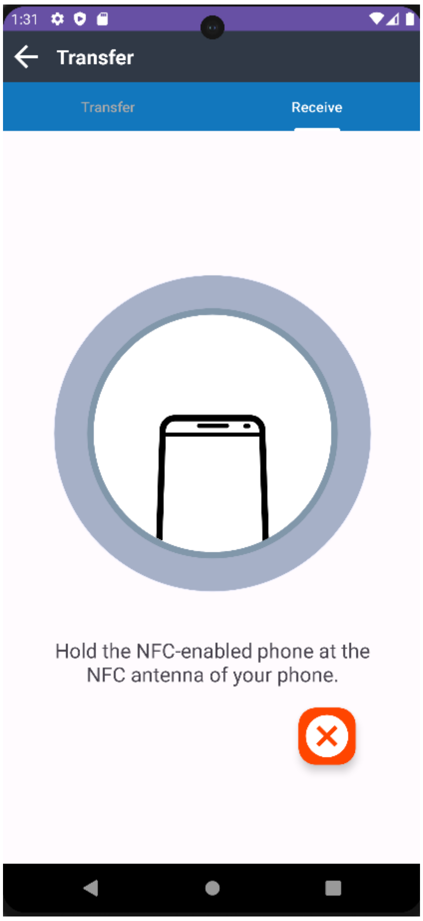
Transfer Successfully
RM

Figure 5.3.25 Fund Transfer Successfully

As Figure 5.3.24 shows, the client is able to perform fund transfers using NFC technology. There are five buttons that are the shortcut key for the client instead of keying the amount.

By clicking the enter, the system will bring the user to the authentication page to enter the authentication PIN to authenticate the transfer.

Once the transfer is successful, the interface will be displayed as Figure 5.3.25.



1:31

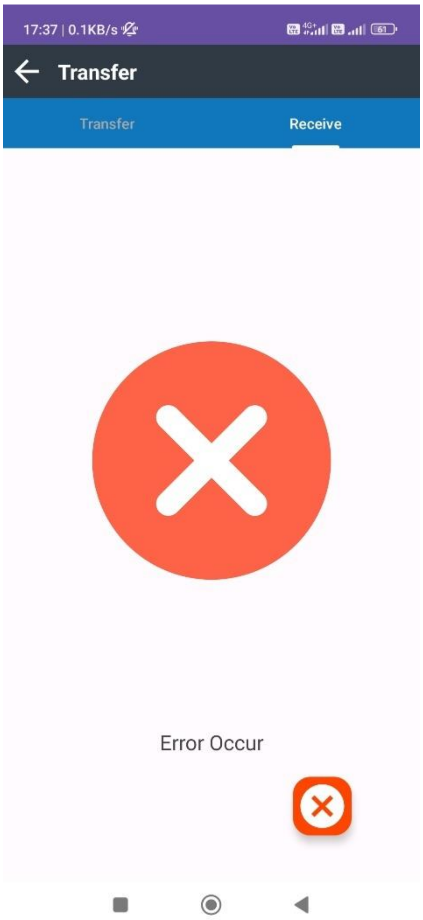
Transfer

Hold the NFC-enabled phone at the NFC antenna of your phone.

Figure 5.3.26 Client Receive Fund Transfers Process

The figure above shows the interface to receive the fund transfers from the other client. This requires another side of the client to launch the corresponding activity as shown in Figure 5.3.24.

Then, another side client must get ready and let two NFC-enabled mobile devices with just one tap to perform communication and data exchange. Once the transfer is done, it will display the success message as shown in Figure 5.3.25.



Error Occur

Figure 5.3.27 Error Occurs when Fund Transfer

When the fund transfer encounters any error, the user interface will display the same as in Figure 5.3.27, including an unknown tag found, communication disrupted, and others.

# CHAPTER 6

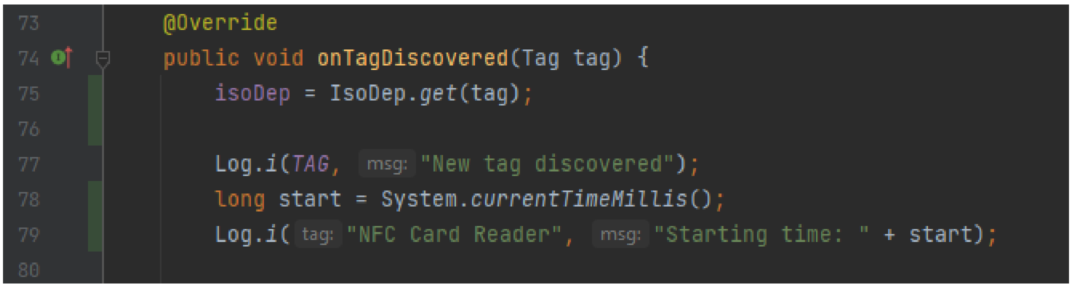
# System Evaluation and Discussion

## 6.1 System Testing and Performance Metrics

System testing and evaluation will be carried out in this section. The testing is based on reliability under various conditions, which is the response or the action to come out of the expected output.

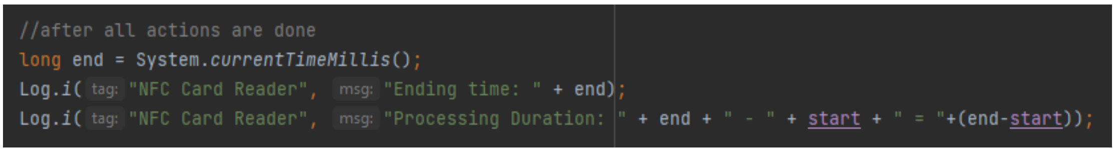
### 6.1.1 NFC Processing Time Analysis

NFC Processing Time Analysis focuses on the delay or latency of how long the processing will be consumed. The NFC responsiveness is a critical part of this project, as one of the sub-objectives mentioned that the NFC technology can speed up the payment process.



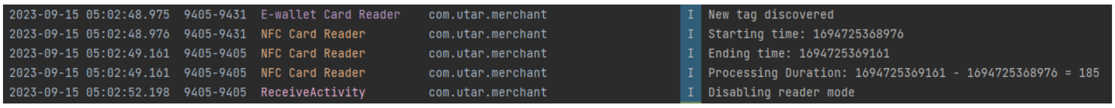
```
    @Override
    public void onTagDiscovered(Tag tag) {
        isoDep = IsoDep.get(tag);

        Log.i(TAG, msg: "New tag discovered");
        long start = System.currentTimeMillis();
        Log.i( tag: "NFC Card Reader", msg: "Starting time: " + start);

```

Figure 6.1.1.1 Start Time Estimation



```
//after all actions are done
long end = System.currentTimeMillis();
Log.i( tag: "NFC Card Reader", msg: "Ending time: " + end);
Log.i( tag: "NFC Card Reader", msg: "Processing Duration: " + end + " - " + start + " = "+(end-start));
```

Figure 6.1.1.2 End Time Estimation



```
2023-09-15 05:02:48.975  9405-9431  E-wallet Card Reader    com.utar.merchant                 I  New tag discovered
2023-09-15 05:02:48.976  9405-9431  NFC Card Reader         com.utar.merchant                 I  Starting time: 1694725368976
2023-09-15 05:02:49.161  9405-9405  NFC Card Reader         com.utar.merchant                 I  Ending time: 1694725369161
2023-09-15 05:02:49.161  9405-9405  NFC Card Reader         com.utar.merchant                 I  Processing Duration: 1694725369161 - 1694725368976 = 185
2023-09-15 05:02:52.198  9405-9405  ReceiveActivity         com.utar.merchant                 I  Disabling reader mode
```

Figure 6.1.1.3 Logcat for Reading the Time

Figures 6.1.1.1 to Figure 6.1.1.3 are used to record the start time of the NFC card communicating with another NFC-enabled device, and the end time as well. Based on the start time until the whole NFC card reader process is ended which is the end time, to calculate the whole NFC card reader payment processing duration.

Table 6.1.1.1 shows the 10 times testing on the NFC payment processing duration as follows:

**Table 6.1.1.1 Duration Analysis of NFC Payment Process**

| No. | Start Time (System Millis) | End Time (System Millis) | Duration (ms) |
|---|---|---|---|
| 1 | 1694725368976 | 1694725369161 | 185 |
| 2 | 1694727800358 | 1694727800519 | 161 |
| 3 | 1694727880023 | 1694727880196 | 173 |
| 4 | 1694727910757 | 1694727910965 | 208 |
| 5 | 1694727944111 | 1694727944276 | 165 |
| 6 | 1694727997038 | 1694727997214 | 176 |
| 7 | 1694728117212 | 1694728117416 | 204 |
| 8 | 1694728141433 | 1694728141749 | 316 |
| 9 | 1694728162504 | 1694728162705 | 201 |
| 10 | 1694728181660 | 1694728181842 | 182 |

Throughout the 10 times testing, the whole payment processing duration would not be greater than 1 second which is 1000 milliseconds.

## 6.2 System Testing Setup and Result

**Table 6.2.1 Login Test Case**

| | | | | |
|---|---|---|---|---|
| **Case Name** | Login | | **ID** | 1 |
| **Description** | The system requires information to log in from a user.<br>Require Information:<br>- Email<br>- Password | | | |
| **Applied Role** | • Merchant<br>• Client | | | |
| | | | | |
| | **Particular** | **Expected Output** | **Status** | |
| **Usual Case** | Enter sufficient and valid information the system requires, then press the login button. | Success in login and redirecting to the home page. | 10 | /10 |
| | | | | |
| **Exceptional Case** | Enter an invalid format email address. | Failure to log in with a specific error message displayed on the corresponding field. | 10 | /10 |
| | Enter a password with less than 6 characters. | | 10 | /10 |
| | Lack of required information input. | | 10 | /10 |
| | Valid information, but the account does not exist. | Toast an error message. | 10 | /10 |
| | Valid account, but apply in the incorrect application. (Client log-in account in Merchant side application.) | | 10 | /10 |

**Table 6.2.2 Registration Test Case**

| | | | | |
|---|---|---|---|---|
| **Case Name** | Registration | | **ID** | 2 |
| **Description** | The system requires information to register an account from a user.<br>Require Information:<br>- Email<br>- Password & Confirm Password<br>- Name | | | |
| **Applied Role** | • Merchant<br>• Client | | | |
| | | | | |
| | **Particular** | **Expected Output** | **Status** | |
| **Usual Case** | Enter sufficient and eligible information that the system requires, then press the register button. | Success to create a new account and redirect to the login page. | 10 | /10 |
| | | | | |
| **Exceptional Case** | Enter an invalid format email address. | Failure to create a new account with a specific error message displayed on the corresponding field. | 10 | /10 |
| | Enter a password with less than 6 characters. | | 10 | /10 |
| | Mismatch password and confirm password. | | 10 | /10 |
| | Lack of required information input. | | 10 | /10 |

**Table 6.2.3 Forgot Password Test Case**

| | | | | |
|---|---|---|---|---|
| **Case Name** | Forgot Password | | **ID** | 3 |
| **Description** | The system requires information to reset the account password.<br>Require Information:<br>- Email | | | |
| **Applied Role** | • Merchant<br>• Client | | | |
| | | | | |
| | **Particular** | **Expected Output** | **Status** | |
| **Usual Case** | Enter sufficient and valid information the system requires, then press the button. | Successfully send the password reset link to the mailbox and redirect to the home page. | 10 | /10 |
| | | | | |
| **Exceptional Case** | Enter an invalid format email address. | Failure to reset with an error display on the corresponding field. | 10 | /10 |
| | Lack of required information input. | | 10 | /10 |
| | Account does not exist. | Toast an error message. | 10 | /10 |

**Table 6.2.4 Reload Test Case**

| | | | | |
|---|---|---|---|---|
| **Case Name** | Reload | | **ID** | 4 |
| **Description** | The function that allows the user to perform top-up on their balance | | | |
| **Applied Role** | • Merchant<br>• Client | | | |
| | | | | |
| | **Particular** | **Expected Output** | **Status** | |
| **Usual Case** | Enter the amount then top-up to the e-wallet balance | Successfully reload and add the amount to the balance. | 10 | /10 |
| | | | | |
| **Exceptional Case** | Lack of required information input. | Failure to reload, then an error display on the amount field. | 10 | /10 |

Table 6.2.5 Withdrawal Test Case

| | | | | |
|---|---|---|---|---|
| **Case Name** | Withdrawal | | **ID** | 5 |
| **Description** | The function that allows the user to withdraw from their balance | | | |
| **Applied Role** | • Merchant<br>• Client | | | |
| | | | | |
| | **Particular** | **Expected Output** | **Status** | |
| **Usual Case** | Enter the amount to withdraw from the wallet balance and authentication is verified. | Successfully withdraw from the e-wallet. | 10 | /10 |
| | | | | |
| **Exceptional Case** | Enter the amount that is insufficient to be withdrawn. | Failure to withdraw, then an error display on the amount field. | 10 | /10 |
| | Enter a password with less than 6 characters. | | 10 | /10 |
| | Lack of information input. | | 10 | /10 |

Table 6.2.6 Transaction Searching Test Case

| | | | | |
|---|---|---|---|---|
| **Case Name** | Transaction Searching | | **ID** | 6 |
| **Description** | The function that allows the user to view and search their transaction records. | | | |
| **Applied Role** | • Merchant<br>• Client | | | |
| | | | | |
| | **Particular** | **Expected Output** | **Status** | |
| **Usual Case** | Check Transaction histories had been made. | 30 days recent records will be displayed. | 10 | /10 |
| | | | | |
| **Exceptional Case** | No default record is found. | Failure to display, then a message display is prompted. | 10 | /10 |
| | No specified record is found. | | 10 | /10 |
| | Lack of information input. | | 10 | /10 |

**Table 6.2.7 Change Language Test Case**

| | | | | |
|---|---|---|---|---|
| **Case Name** | Transaction Searching | | **ID** | 7 |
| **Description** | The function that allows the user to change the displaying language of the system. | | | |
| **Applied Role** | • Merchant<br>• Client | | | |
| | | | | |
| | **Particular** | **Expected Output** | **Status** | |
| **Usual Case** | Change the language that is to be displayed. | System restart, and based on the selected language to display in the user interface. | 10 | /10 |

**Table 6.2.8 Log Out Test Case**

| | | | | |
|---|---|---|---|---|
| **Case Name** | Transaction Searching | | **ID** | 8 |
| **Description** | The function that allows the user to log out from the system. | | | |
| **Applied Role** | • Merchant<br>• Client | | | |
| | | | | |
| | **Particular** | **Expected Output** | **Status** | |
| **Usual Case** | Click log-out in the Settings interface, and log out from the system. | Log out successfully and redirect to the login interface. | 10 | /10 |

**Table 6.2.9 PIN Registration Test Case**

| | | | | |
|---|---|---|---|---|
| **Case Name** | PIN Registration | | **ID** | 9 |
| **Description** | The function allows the user to register the PIN for authentication purposes. | | | |
| **Applied Role** | • Merchant<br>• Client | | | |
| | | | | |
| | **Particular** | **Expected Output** | **Status** | |
| **Usual Case** | Enter PIN and confirm PIN are the same. | The PIN is successfully registered. | 10 | /10 |
| | | | | |
| **Exceptional Case** | Both PIN and confirm PIN are entered inconsistently. | Display the PIN is mismatched. | 10 | /10 |

**Table 6.2.10 PIN Modification Test Case**

| | | | | |
|---|---|---|---|---|
| **Case Name** | PIN Modification | | **ID** | 10 |
| **Description** | The function allows the user to modify the PIN for authentication purposes. | | | |
| **Applied Role** | • Merchant<br>• Client | | | |
| | | | | |
| | **Particular** | **Expected Output** | **Status** | |
| **Usual Case** | Enter PIN and confirm PIN are the same. | The PIN is successfully modified. | 10 | /10 |
| | | | | |
| **Exceptional Case** | Both PIN and confirm PIN are entered inconsistently. | Display the PIN is mismatched. | 10 | /10 |

**Table 6.2.11 PIN Authentication Test Case**

| | | | | |
|---|---|---|---|---|
| **Case Name** | PIN Modification | | **ID** | 11 |
| **Description** | The function allows the user to authenticate themselves using PIN or biometrics for authentication purposes. | | | |
| **Applied Role** | • Merchant<br>• Client | | | |
| | | | | |
| | **Particular** | **Expected Output** | **Status** | |
| **Usual Case** | Enter PIN and confirm PIN are the same. | The PIN is successfully verified. | 10 | /10 |
| | Valid biometrics. | The biometric is successfully verified. | 10 | /10 |
| | | | | |
| **Exceptional Case** | Both PIN and confirm PIN are entered inconsistently. | Display the PIN is mismatched. | 10 | /10 |

**Table 6.2.12 Receiving Amount Input Test Case**

| | | | | |
|---|---|---|---|---|
| **Case Name** | Receiving Amount Input | | **ID** | 12 |
| **Description** | The function allows the merchant to key in the amount that should be paid by the customer. | | | |
| **Applied Role** | • Merchant | | | |
| | | | | |
| | **Particular** | **Expected Output** | **Status** | |
| **Usual Case** | Key in the correct amount and the confirm button is clicked. | Redirect to receiving activity and enable NFC card reader mode. | 10 | /10 |
| | | | | |
| **Exceptional Case** | The NFC module is off. | Prompt windows to ask the user whether to redirect to the settings interface to turn on the NFC module. | 10 | /10 |
| | Lack of information input. | Toast a message to notify the user to input a minimum amount. | 10 | /10 |

**Table 6.2.13 NFC Payment Receive Test Case**

| | | | | |
|---|---|---|---|---|
| **Case Name** | NFC Payment Receive | | **ID** | 13 |
| **Description** | The function allows the merchant to receive the amount that should be paid by the customer. | | | |
| **Applied Role** | • Merchant | | | |
| | | | | |
| | **Particular** | **Expected Output** | **Status** | |
| **Usual Case** | Key in the correct amount and the confirm button is clicked. | Redirect to receiving activity and enable NFC card reader mode. | 10 | /10 |
| | | | | |
| **Exceptional Case** | The NFC module is off. | Prompt windows to ask the user whether to redirect to the settings interface to turn on the NFC module. | 10 | /10 |
| | Lack of information input. | Toast a message to notify the user to input a minimum amount. | 10 | /10 |

**Table 6.2.14 NFC Payment Test Case**

| | | | | |
|---|---|---|---|---|
| **Case Name** | NFC Payment | | **ID** | 14 |
| **Description** | The function allows the client to make payments using NFC HCE technology. | | | |
| **Applied Role** | - Client | | | |
| | | | | |
| | **Particular** | **Expected Output** | **Status** | |
| **Usual Case** | Perform payment transactions with the corresponding NFC card reader. | The payment is successful. | 10 | /10 |
| | | | | |
| **Exceptional Case** | Insufficient balance. | Display error | 10 | /10 |
| | An incorrect NFC card reader is approached. | The user interface remains the same. | 10 | /10 |

**Table 6.2.15 NFC Fund Transfer Test Case**

| | | | | |
|---|---|---|---|---|
| **Case Name** | NFC Fund Transfer | | **ID** | 15 |
| **Description** | The function allows the client to make fund transfers using NFC technology. | | | |
| **Applied Role** | - Client | | | |
| | | | | |
| | **Particular** | **Expected Output** | **Status** | |
| **Usual Case** | A correct NFC application is approached. | The transfer is successful. | 10 | /10 |
| | | | | |
| **Exceptional Case** | Insufficient balance in transfer out role. | Display error with the amount input field. | 10 | /10 |
| | An incorrect NFC card reader is approached. | The user interface remains the same. | 10 | /10 |

## 6.3 Project Challenges

There are several difficult issues that have been encountered during the development.

- **NFC technology resource**

  The resource is limited, there is difficulty in learning the working principle of the NFC host-based card emulation technology.

- **React Native module integration due to its version.**

  React Native is a newly released programming language and have never learned it before. When the moment comes into the progress to integrate the Android Java NFC module into the React Native module, it results in failure integration.

  The issue is caused by the version of the Java module and the version of React Native module, which then makes the Java NFC Application Programming Interface (API) have difficulty functioning well.

  After debugging for several days, the whole development is moved to Java programming language from React Native.

- **NFC Mode Implementation**

  It is difficult to implement the NFC reader mode and NFC HCE mode within a mobile application, as the NFC HCE is a background-based service, sometimes the mobile host will be confused about what service will be the priority to be called.

## 6.4 Project Evaluation

Project evaluation is a process to measure the proposed system and whether it achieves the project objectives.

The first objective is to improve the payment processing speed using NFC card emulator technology. This is achieved by using NFC HCE to emulate a card and complete transactions to reduce payment time.

The second object is requiring fingerprint authentication before processing transactions. It is needed for strong authentication methods, as fingerprint scanning will be incorporated as a biometric authentication method to enhance security in terms of NFC HCE and prevent issues like PIN theft.

The last objective is to allow users to perform money transfers using NFC technology. This feature is designed to address situations where users have insufficient e-wallet balance, so another user can use NFC for fund transfers in emergencies, such as when they cannot reload their e-wallet at the moment.

In this project, the three objectives have been fulfilled the contributions that can benefit the public in terms of e-wallet payment.

# CHAPTER 7

# Conclusion

## 7.1 Conclusion

The payment methods could come out with better innovations to resolve the weaknesses or issues that exist with the current popular payment methods. This research has been done in Chapter 2 Literature Review. Therefore, a better payment method which is NFC to be introduced in this proposed development, the system requirements and system design have been outlined in Chapter 4 accordingly. Chapter 5 illustrates the implementation of the proposed system to have better prototyping. Lastly, Chapter 6 performs software testing for this prototyping of the proposed system on how the performance of the application runs, such as the expected output, the efficiency and others.

In conclusion, NFC technology is newly coming out to the world. It comes with a lot of advantages to the technology communication among the end devices but at least two NFC-enabled devices.

There are two mobile devices to be involved to perform the functionalities in this project. One of them mainly develops the NFC technology function which emulates a card to make payment instead of using card payment, cash payment, and QR code payment. Another one acts as a reader to read information from the emulated card.

This proposed development as an example, is developed to speed up the payment processing time and decrease the risk of revealing sensitive information. Based on these outcomes to solve the issues that always happen in our daily life.

## 7.2 Recommendation

This project successfully achieved all the objectives. The main objective of this project is to allow the user to use the NFC technology to process the payment. In the end, it can speed up the payment process while decreasing the risk of being attacked by the man in the middle.

There is a limitation to this project, which is the system application must be used within an NFC-enabled OS device. Otherwise, the payment and the NFC card reader cannot be implemented at all.

Since this project is still not completed, there are still some recommendations to be added to the proposed system future. Firstly, the NFC technology is only available in the payment system, it would be great with additional payment methods such as QR code payment if the NFC is not available.

The next recommendation is to integrate the daily routine activity into the application. This can refer to the Touch n' Go e-wallet that includes those bill services and others. This purposely can attract users to use this proposed system in terms of convenience.

# REFERENCES

[1] Google, "Near Field Communication Overview," https://developer.android.com/guide/topics/connectivity/nfc (Accessed Aug. 17, 2022).

[2] A. Rahul, G. Krishnan, U. Krishnan, S. Rao, "Near Field Communication (NFC) Technology: A survey," *International Journal on Cybernetics & Informatics*, Vol. 4, pp. 133-144, Apr. 2015. [Online]. Available: https://www.researchgate.net/publication/276534674_Near_Field_Communication_NFC_Technology_A_Survey [Accessed Aug. 17, 2022].

[3] O. Charles, et al., "An Overview of Near Field Communication (NFC)," *International Journal of Scientific & Engineering Research*, Vol. 9, Dec. 2018. [Online]. Available: https://www.ijser.org/researchpaper/An-Overview-of-Near-Field-Communication-NFC.pdf [Accessed Aug. 17, 2022]

[4] J. Yeung and Y. Wakatsuki, "Cashier Arrested in Japan for Allegedly Using 'Photographic Memory' to Steal Credit Card Information of 1,300 Customers," Sept. 2019. [Online]. Available: https://edition.cnn.com/2019/09/10/asia/japan-memory-credit-card-intl-hnk-scli/index.html?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+rss%2Fcnn_latest+%28RSS%3A+CNN+-+Most+Recent%29 [Accessed Aug. 20, 2022].

[5] Oppotus, "Malaysian E-Wallet Usage as We Move Towards Recovery [2021]," Nov. 2021. [Online]. Available: https://www.oppotus.com/malaysian-e-wallet-usage-towards-recovery-2021/ [Accessed Aug. 20, 2022].

[6] Google, "Host-based Card Emulation Overview," Mar. 2022. [Online]. Available: https://developer.android.com/guide/topics/connectivity/nfc/hce [Accessed Sept. 2, 2022].

[7] Y. Zhu and S. H. Li, "A Hangzhou Story: The Development of China's Mobile Payment Revolution," Mar. 2018. [Online]. Available: https://lkyspp.nus.edu.sg/docs/default-source/case-studies/a-hangzhou-story.pdf?sfvrsn=2bb6690a_2 [Accessed Sept. 3, 2022].

[8] Touch n' Go Sdn. Bhd., "How Do I Pay with the Touch 'n Go eWallet using QR Code at Merchant's outlets?" [Online]. Available: https://support.tngdigital.com.my/hc/en-my/articles/360035649734-How-do-I-pay-with-the-Touch-n-Go-eWallet-using-QR-Code-at-Merchant-s-outlets- [Accessed: Sept. 3, 2022].

[9] B. Hu and Y. Zhou, "Research on Quickpass Payment Terminal Application System Based on Dynamic QR Code," *Journal of Physics: Conference Series*, vol. 1168, March 2019. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1168/3/032059/pdf [Accessed Sept. 3, 2022].

[10] Grab, "How to Scan My Customer's GrabPay QR Code,". [Online]. Available: https://help.grab.com/merchant/en-sg/360041516312-What-is-consumer-GrabPay-QR-

code#:~:text=What%20happens%20if%20I'm,the%20codes%20after%2045%20seconds. [Accessed Sept. 3, 2022].

[11] T. Budhram, “Lost, Stolen or Skimmed: Overcoming Credit Card Fraud in South Africa,” *South African Crime Quarterly,* Mar. 2016. [Online]. Available: https://www.researchgate.net/publication/301275602_Lost_stolen_or_skimmed_Overcoming_credit_card_fraud_in_South_Africa [Accessed Sept. 4, 2022].

[12] T. Ebringer, P. Thorne and Y. Zheng, “Parasitic Authentication to Protect your e-wallet,” *Computer,* vol. 33, no. 10, pp. 54-60, Oct. 2000. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=876293 [Accessed Sept. 4, 2022].

[13] O. Ghag, “A Comprehensive Study of Google Wallet as an NFC Application,” International Journal of Computer Applications, vol. 58, Nov. 2012. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.252.6928&rep=rep1&type=pdf [Accessed Sept. 4, 2022]

[14] X. Bai, et al., “Picking Up My Tab: Understanding and Mitigating Synchronized Toke Lifting and Spending in Mobile Payment,” 26th USENIX Security Symposium, Aug. 2017. [Online]. Available: https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-bai.pdf [Accessed Sept. 4, 2022].

[15] R. Lake, “How Do Credit Cards Work?” Jan. 2023. [Online]. Available: https://www.investopedia.com/how-do-credit-cards-work-5025119#:~:text=your%20credit%20limit.-,How%20Credit%20Cards%20Work,network%20to%20process%20the%20transaction. [Accessed Apr. 10, 2023].

[16] A. B. Amir, “Payment Systems in Malaysia: Recent Developments and Issues,” ADBI Working Paper Series, no. 151, Sept. 2009. [Online]. Available: https://www.adb.org/sites/default/files/publication/156006/adbi-wp151.pdf [Accessed Feb. 16, 2023].

[17] R. H. A. Witjaksono, et al., “Quick Response Code Acceptance on Digital Wallet Mobile Applications in Indonesia,” *2021 International Conference on Advanced Computer Science and Information Systems (ICACSIS),* Depok, Indonesia, 2021, pp. 1-6, doi: 10.1109/ICACSIS53237.2021.9631354. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9631354 [Accessed Feb 21, 2023].

[18] S. Nseir, N. Hirzallah and M. Aqel, “A Secure Mobile Payment System Using QR Code,” *2013 5th International Conference on Computer Science and Information Technology,* Amman, Jordan, 2013, pp. 111-114, doi: 10.1109/CSIT.2013.6588767. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6588767 [Accessed Feb 21, 2023].

[19] N. Kumari and J. Khanna, “Cashless Payment: A Behaviourial Change to Economic Growth,” *Qualitative and Quantitative Research Review*, vol. 2, Issue 2, 2017. [Online]. Available: https://nfct.co.uk/wp-content/uploads/journal/published_paper/volume-2/issue-2/LS0q4m3F.pdf [Accessed Feb. 21, 2023].

[20] “A quick guide to QR code payments,”. Available: https://www.checkout.com/blog/post/a-quick-guide-to-qr-code-payments [Accessed Mar. 1, 2023]

[21] E. Xiao, “Thieves Are Pickpocketing Wallet Apps in China,” Mar. 9, 2017. [Online]. Available: https://www.techinasia.com/fake-qr-code-scams-china [Accessed Mar. 2, 2023]

[22] R. Lee, “PSA: Google Pay Can Be Used without Unlocking Your Phone,” Nov. 15, 2022. [Online]. Available: https://soyacincau.com/2022/11/15/google-pay-lockscreen-unlock-smartphone-security/ [Accessed Mar. 3, 2023]

[23] Google, “Set Up Screen Lock for Tap to Pay Transactions,”. Available: https://support.google.com/wallet/answer/12059519?hl=en#:~:text=No%20unlock%20needed%20for%20smaller%20payments&text=You%20can%20only%20make%20a,minutes%20after%20you%20unlock%20it. [Accessed Mar. 3, 2023]

[24] Thales, “Comprehensive Guide to Authentication Technologies and Methods,” Mar. 2020. [Online]. Available: https://www.net-ctrl.com/wp-content/uploads/2020/12/Comprehensive_Guide_to_Authentication_WP_v2.pdf [Accessed Mar. 4, 2023]

[25] “A Guide to Authentication Factors: Knowledge, Possession, and Inherence,” Mar. 30, 2022. Available: https://q5id.com/blog/a-guide-to-authentication-factors-knowledge-possession-and-inherence [Accessed Mar. 4, 2023]

[26] Vall, “10 QR Code Scanning Problems and How to Fix Them, “Apr. 2, 2023. [Online]. Available: https://www.qrcode-tiger.com/qr-code-scanning-problems [Accessed Apr. 10, 2023]

[27] F. Tawfiq, “Software System Development Life Cycle, “June 4, 2020. [Online]. Available: https://www.researchgate.net/publication/341883828_SDLC_system_development_life_cycle [Accessed Apr. 11, 2023]

[28] “The Top 7 SDLC Methodologies, “Dec. 19, 2021. [Online]. Available: https://www.michaelpage.com.au/advice/career-advice/productivity-and-performance/top-7-sdlc-methodologies [Accessed Apr. 11, 2023]

[29] Y. T. W. Tiky, “Software Development Life Cycle, “The Hong Kong University of Science and Technology. [Online]. Available: https://www.cse.ust.hk/~rossiter/independent_studies_projects/software_development/software_development_report.pdf [Accessed Apr5. 11, 2023]

[30] R. Lee, “PSA: Google Pay Can Be Used without Unlocking Your Phone, “Nov. 15, 2022. [Online]. Available: https://soyacincau.com/2022/11/15/google-pay-lockscreen-unlock-smartphone-security/ [Accessed Apr. 12, 2023]

[31] R. Uttarwar, et al., “A Literature Review on Android -A Mobile Operating System,” Sept. 14, 2021. [Online]. Available: https://www.researchgate.net/publication/354576500_A_Literature_Review_on_Android_-A_Mobile_Operating_system [Accessed May. 25, 2023].

[32] N. K. Ekanayake, “Android Operating System,” May. 20, 2018. [Online]. Available: https://www.researchgate.net/publication/325257105_Android_Operating_System [Accessed May. 25, 2023].

[33] “Advantages and Disadvantages of Google Android,” Apr. 2, 2021. [Online]. Available: https://www.digitalaptech.com/advantages-and-disadvantages-of-google-android/ [Accessed May. 26, 2023].

[34] “iOS vs Android Quarterly Market Share,” May. 18, 2023. [Online]. Available: https://www.counterpointresearch.com/global-smartphone-os-market-share/ [Accessed May. 26, 2023].

[35] “Strengths and Weaknesses of Android Based Applications,” Sept. 9, 2022. [Online]. Available: https://distinguished.io/blog/strengths-weaknesses-android-OS [Accessed May. 26, 2023].

[36] S. Tiwari, “13 Advantages and Disadvantages of iOS,” Mar. 17, 2021. [Online]. Available: https://honestproscons.com/13-advantages-and-disadvantages-of-ios/ [Accessed May. 26, 2023].

[37] Miker Works, “Your App can Open Doors,” n.d. [Online]. Available: https://miker.works/english/blog/your-app-can-open-doors/ [Accessed May. 26, 2023].

[38] “E. Cervantes, “10 Things iOS does Better than Android,” Android Authority, May. 13, 2023. [Online]. Available: https://www.androidauthority.com/ios-vs-android-1068950/ [Accessed May. 28, 2023].

[39] S. Goldman, “iPhone vs. Android 2023: How the Best Phones Compare and 4 New Features from Each,”, May. 26, 2023. [Online]. Available: https://history-computer.com/iphone-vs-android-2023-how-the-best-phones-compare-and-4-new-features-from-each/ [Accessed May. 28, 2023].

[40] D. Levi and E. Kizzy, “A Comparative Review of Mobile Application Development Frameworks: Kotlin Vs Java,” vol. 6, Sept. 21, 2022. [Online]. Available: https://www.researchgate.net/publication/363721478_A_Comparative_Review_of_Mobile_Application_Development_Frameworks_Kotlin_Vs_Java [Accessed May. 28, 2023].

[41] Data Flair, "Pros and Cons of Java | Advantages and Disadvantages of Java," n.d. [Online]. Available: https://data-flair.training/blogs/pros-and-cons-of-java/ [Accessed May. 28, 2023].

[42] A.S. Gillis, "Object-Oriented Programming (OOP)," Jul, 2021. [Online]. Available: https://www.techtarget.com/searchapparchitecture/definition/object-oriented-programming-OOP [Accessed May. 28, 2023].

[43] M. Comviva, "What is Host Card Emulation (HCE)," Jul. 13, 2015. [Online]. Available: https://www.comviva.com/blog/references/what-is-hce/#:~:text=Benefits%20of%20The%20HCE%3A,the%20card%20without%20network%20operators [Accessed Jun. 3, 2023].

[44] H. Ozsahan, "What is Firebase Basically?" *What is Firebase? Learn The Basics from A to Z,* May. 15, 2023. [Online]. Available: https://www.resmo.com/blog/what-is-firebase#:~:text=Firebase%20features%20include%20data%20storage,on%20creating%20excellent%20user%20experiences. [Accessed Jun. 4, 2023].

[45] B. Eisenman, "Learning React Native: Building Native Mobile Apps with JavaScript," Dec. 01, 2015. [Online]. Available: https://books.google.com.my/books?hl=en&lr=&id=274fCwAAQBAJ&oi=fnd&pg=PR2&dq=react+native&ots=tGxmfFj8m_&sig=w3GrsimAMy5jlsRrMpnIHZ8aHQs&redir_esc=y#v=onepage&q=react%20native&f=false [Accessed Jun. 13, 2023].

[46] M. Cavallari, L. Adami, F. Tornieri, "Organisational Aspects and Anatomy of an Attack on NFC/HCE Mobile Payment Systems," *ICEIS (2),* Apr. 2015, pp. 685-700. [Online]. Available: https://pdfs.semanticscholar.org/84e9/828e8f16cb5b9a970940ce1687656954c58d.pdf [Accessed Jun. 29, 2023].

[47] K. Michal, "Top Most Popular Programming Languages for Mobile App Development," Jun. 28, 2023. [Online]. Available: https://fireart.studio/blog/top-most-popular-programming-languages-for-mobile-app-development/ [Accessed Aug. 26, 2023].

[48] A. Hussain, E. Mkpojiogu, et al., "An Instrumental Assessment of Touch 'n Go eWallet Mobile App," *International Journal of Interactive Mobile Technologies (iJIM),* Mar. 30, 2021. [Online]. Available: https://www.researchgate.net/publication/350488039_An_Instrumental_Assessment_of_Touch'n_Go_eWallet_Mobile_App [Accessed Aug. 29, 2023].

[49] V. S. Woon Chong, J. M. S. Lam and Y. J. Too, "Empirical Study on Intention to Use of Touch 'n Go e-Wallet among Millennial in Malaysia during Covid-19 Endemic," *2022 International Conference on Digital Transformation and Intelligence (ICDI), Kuching, Sarawak, Malaysia,* 2022, pp 156-160, doi: 10.1109/ICDI57181.2022.10007416.

# APPENDIX

## FINAL YEAR PROJECT WEEKLY REPORT

*(Project I)*

| | |
|---|---|
| **Trimester, Year: Trimester 1, Year 3** | **Study week no.: 2** |
| **Student Name & ID: Chong Yao En 21ACB00289** | |
| **Supervisor: Ms Tan Lyk Yin** | |
| **Project Title: E-wallet using NFC Mobile Application Development** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- React Native programming language learning.
- The research and review on the technology of NFC.
- The draft version of the User Interface of the merchant-side application.

**2. WORK TO BE DONE**

- To start developing the UI (login and Registration modules).
- To establish the connection between Firebase authentication and mobile application.
- To draft a report on Chapter 1 Introduction.

**3. PROBLEMS ENCOUNTERED**

- The resource of NFC technology is limited to research.

**4. SELF EVALUATION OF THE PROGRESS**

- Consume a lot of time learning React Native programming language.

Supervisor's signature

chong

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project I)*

| | |
|---|---|
| **Trimester, Year: Trimester 1, Year 3** | **Study week no.: 4** |
| **Student Name & ID: Chong Yao En 21ACB00289** | |
| **Supervisor: Ms Tan Lyk Yin** | |
| **Project Title: E-wallet using NFC Mobile Application Development** | |

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- The design of UI (Login and Registration modules).
- Firebase authentication.
- Chapter 1 Introduction of the report.

## 2. WORK TO BE DONE

- To develop a forgot password interface.
- To develop the card discovery on NFC card reader mode.
- To integrate the Java NFC card reader source code into React Native.
- To draft chapter 2 of the report.

## 3. PROBLEMS ENCOUNTERED

N/A

## 4. SELF EVALUATION OF THE PROGRESS

- Progress is completed in time.

chong

Supervisor's signature

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project I)*

| | |
|---|---|
| **Trimester, Year: Trimester 1, Year 3** | **Study week no.: 6** |
| **Student Name & ID: Chong Yao En 21ACB00289** | |
| **Supervisor: Ms Tan Lyk Yin** | |
| **Project Title: E-wallet using NFC Mobile Application Development** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Forgot password interface.
- The investigation of the card discovery on NFC card reader mode.
- Chapter 2 of the report.

**2. WORK TO BE DONE**

- To start over the user interface development using Java programming language.
- User Interfaces:
  - History
  - Reload/Withdraw
  - Client Payment
  - Merchant Receive
- Firebase Realtime Database.

**3. PROBLEMS ENCOUNTERED**

- Failure to integrate the Java NFC card reader mode source code into React Native module.

**4. SELF EVALUATION OF THE PROGRESS**

- Consume much time to debug the issue of integration on Java into React Native module.

Supervisor's signature

chong

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project I)*

| | |
|---|---|
| **Trimester, Year: Trimester 1, Year 3** | **Study week no.: 8** |
| **Student Name & ID: Chong Yao En 21ACB00289** | |
| **Supervisor: Ms Tan Lyk Yin** | |
| **Project Title: E-wallet using NFC Mobile Application Development** | |

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- The basic user interfaces.
- Firebase Realtime Database.

## 2. WORK TO BE DONE

- To finalise the report.
- To embed NFC HCE reader into the Merchant application.
- To embed NFC HCE into the Client application.
- User Interface:
  - Client Payment / Merchant Payment Receive

## 3. PROBLEMS ENCOUNTERED

- Animation for the payment and receive interface difficult to be embedded.

## 4. SELF EVALUATION OF THE PROGRESS

- All arranged schedules can be completed on time.

Supervisor's signature

chong

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project I)*

| | |
|---|---|
| **Trimester, Year: Trimester 1, Year 3** | **Study week no.: 10** |
| **Student Name & ID: Chong Yao En 21ACB00289** | |
| **Supervisor: Ms Tan Lyk Yin** | |
| **Project Title: E-wallet using NFC Mobile Application Development** | |

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- The development of NFC HCE reader mode into the Merchant application.
- The development of NFC HCE card emulation mode into the Client application.
- Payment/Receive Interfaces.

## 2. WORK TO BE DONE

- To finalise the FYP1 report.

## 3. PROBLEMS ENCOUNTERED

- Not familiar with the NFC in Java API.
- New to NFC technology, having difficult making both devices communicate through NFC.

## 4. SELF EVALUATION OF THE PROGRESS

- All arranged schedules can be completed on time.

Supervisor's signature

chong

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project I)*

| | |
|---|---|
| **Trimester, Year: Trimester 1, Year 3** | **Study week no.: 12** |
| **Student Name & ID: Chong Yao En 21ACB00289** | |
| **Supervisor: Ms Tan Lyk Yin** | |
| **Project Title: E-wallet using NFC Mobile Application Development** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- FYP1 report.

**2. WORK TO BE DONE**

N/A

**3. PROBLEMS ENCOUNTERED**

N/A

**4. SELF EVALUATION OF THE PROGRESS**

- All arranged schedules can be completed on time.

Supervisor's signature

chong

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Trimester 2, Year 3** | **Study week no.: 2** |
| **Student Name & ID: Chong Yao En 21ACB00289** | |
| **Supervisor: Ms Tan Lyk Yin** | |
| **Project Title: E-wallet using NFC Mobile Application Development** | |

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Achieve the second objective (Biometric authentication).
- Authentication Interface.
- Transfer Interfaces in Client application.

## 2. WORK TO BE DONE

- Transfer out and receive through NFC technology. (Third objective)
- To revise FYP1 report.

## 3. PROBLEMS ENCOUNTERED

N/A

## 4. SELF EVALUATION OF THE PROGRESS

- All arranged schedules can be completed on time.

______________________________
Supervisor's signature

chong
______________________________
Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Trimester 2, Year 3** | **Study week no.: 4** |
| **Student Name & ID: Chong Yao En 21ACB00289** | |
| **Supervisor: Ms Tan Lyk Yin** | |
| **Project Title: E-wallet using NFC Mobile Application Development** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Chapter 1 to Chapter 3 of the report.
- Achieved third objective.

**2. WORK TO BE DONE**

- Report: Chapter 4
- Client-side system (Objective 2) – to implement authentication payment system.
- To draft a merchant-side system of the activity diagrams.

**3. PROBLEMS ENCOUNTERED**

No.

**4. SELF EVALUATION OF THE PROGRESS**

- All arranged schedules can be completed on time.

Supervisor's signature

chong

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Trimester 2, Year 3** | **Study week no.: 6** |
| **Student Name & ID: Chong Yao En 21ACB00289** | |
| **Supervisor: Ms Tan Lyk Yin** | |
| **Project Title: E-wallet using NFC Mobile Application Development** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Draft of the Merchant-side system activity diagram.
- The 2nd objective.
- Report – Chapter 3, 4 (draft)

**2. WORK TO BE DONE**

- 3rd objective – to allow the user to make fund transfers using NFC technology.

**3. PROBLEMS ENCOUNTERED**

- The NFC reader side could be confused with reader mode or card emulator mode.

**4. SELF EVALUATION OF THE PROGRESS**

- All arranged schedules can be completed on time.

Supervisor's signature

chong

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Trimester 2, Year 3** | **Study week no.: 8** |
| **Student Name & ID: Chong Yao En 21ACB00289** | |
| **Supervisor: Ms Tan Lyk Yin** | |
| **Project Title: E-wallet using NFC Mobile Application Development** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- 3rd objective development.
- Debug with the NFC mode confusing.

**2. WORK TO BE DONE**

- To design the amount shortcut key for merchant amount input activity.
- To draft report – Chapters 4,5,6

**3. PROBLEMS ENCOUNTERED**

No.

**4. SELF EVALUATION OF THE PROGRESS**

- All arranged schedules can be completed on time.

Supervisor's signature

chong

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Trimester 2, Year 3** | **Study week no.: 10** |
| **Student Name & ID: Chong Yao En 21ACB00289** | |
| **Supervisor: Ms Tan Lyk Yin** | |
| **Project Title: E-wallet using NFC Mobile Application Development** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- To design the amount shortcut key for merchant amount input activity.

**2. WORK TO BE DONE**

- To develop an NFC transfer login function
- To develop multiple languages
- To draft report – Chapters 4,5,6

**3. PROBLEMS ENCOUNTERED**

No.

**4. SELF EVALUATION OF THE PROGRESS**

- All arranged schedules can be completed on time.

Supervisor's signature

chong

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

| | |
|---|---|
| **Trimester, Year: Trimester 2, Year 3** | **Study week no.: 12** |
| **Student Name & ID: Chong Yao En 21ACB00289** | |
| **Supervisor: Ms Tan Lyk Yin** | |
| **Project Title: E-wallet using NFC Mobile Application Development** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Report – Chapters 4, 5, 6
- To develop an NFC transfer login function
- To develop multiple languages

**2. WORK TO BE DONE**

- Finalize Report.

**3. PROBLEMS ENCOUNTERED**

No.

**4. SELF EVALUATION OF THE PROGRESS**

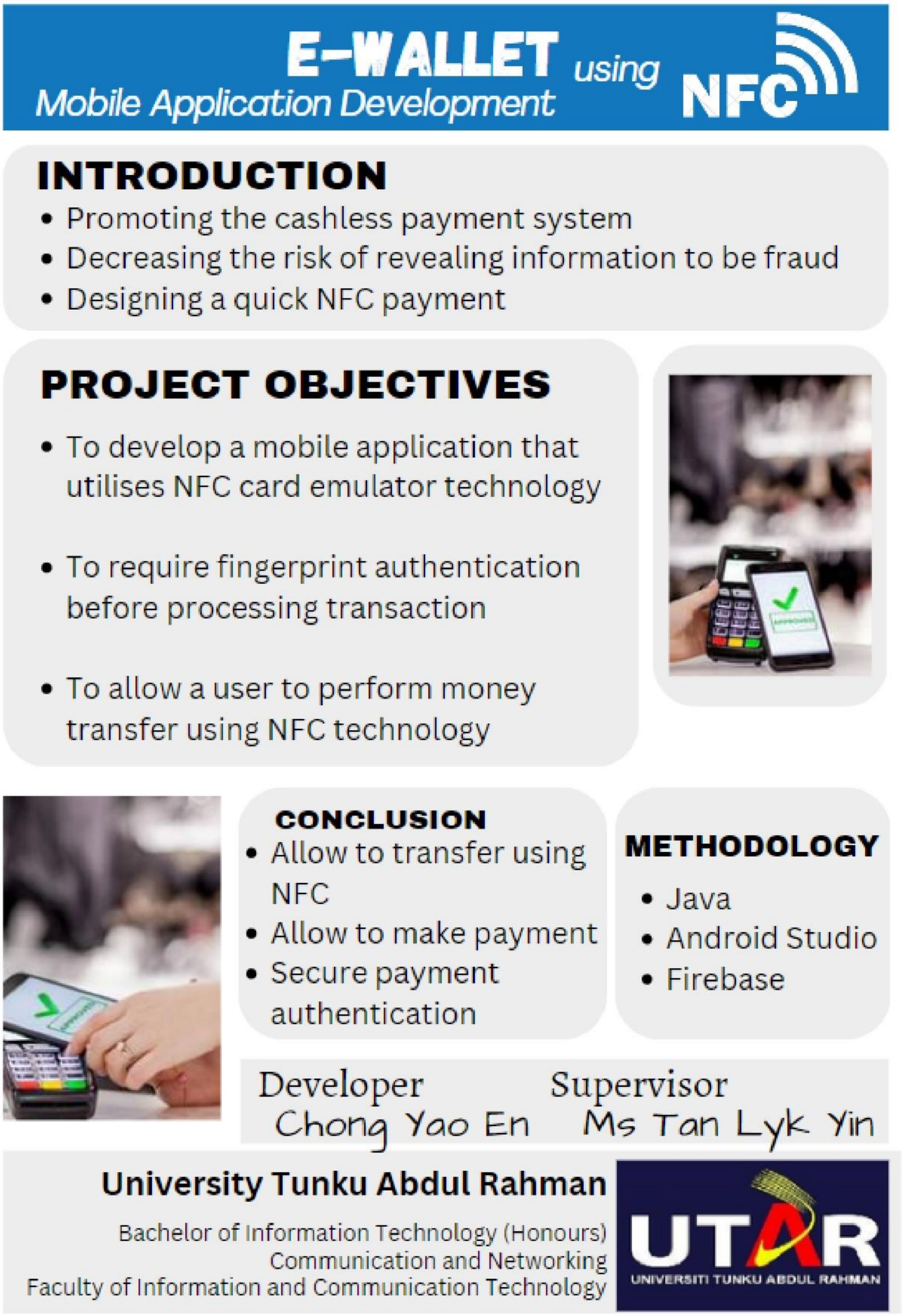
- All arranged schedules can be completed on time.

Supervisor's signature

chong

Student's signature

# POSTER



## E-WALLET using NFC

*Mobile Application Development*

### INTRODUCTION

- Promoting the cashless payment system
- Decreasing the risk of revealing information to be fraud
- Designing a quick NFC payment

### PROJECT OBJECTIVES

- To develop a mobile application that utilises NFC card emulator technology
- To require fingerprint authentication before processing transaction
- To allow a user to perform money transfer using NFC technology

### CONCLUSION

- Allow to transfer using NFC
- Allow to make payment
- Secure payment authentication

### METHODOLOGY

- Java
- Android Studio
- Firebase

Developer
Chong Yao En

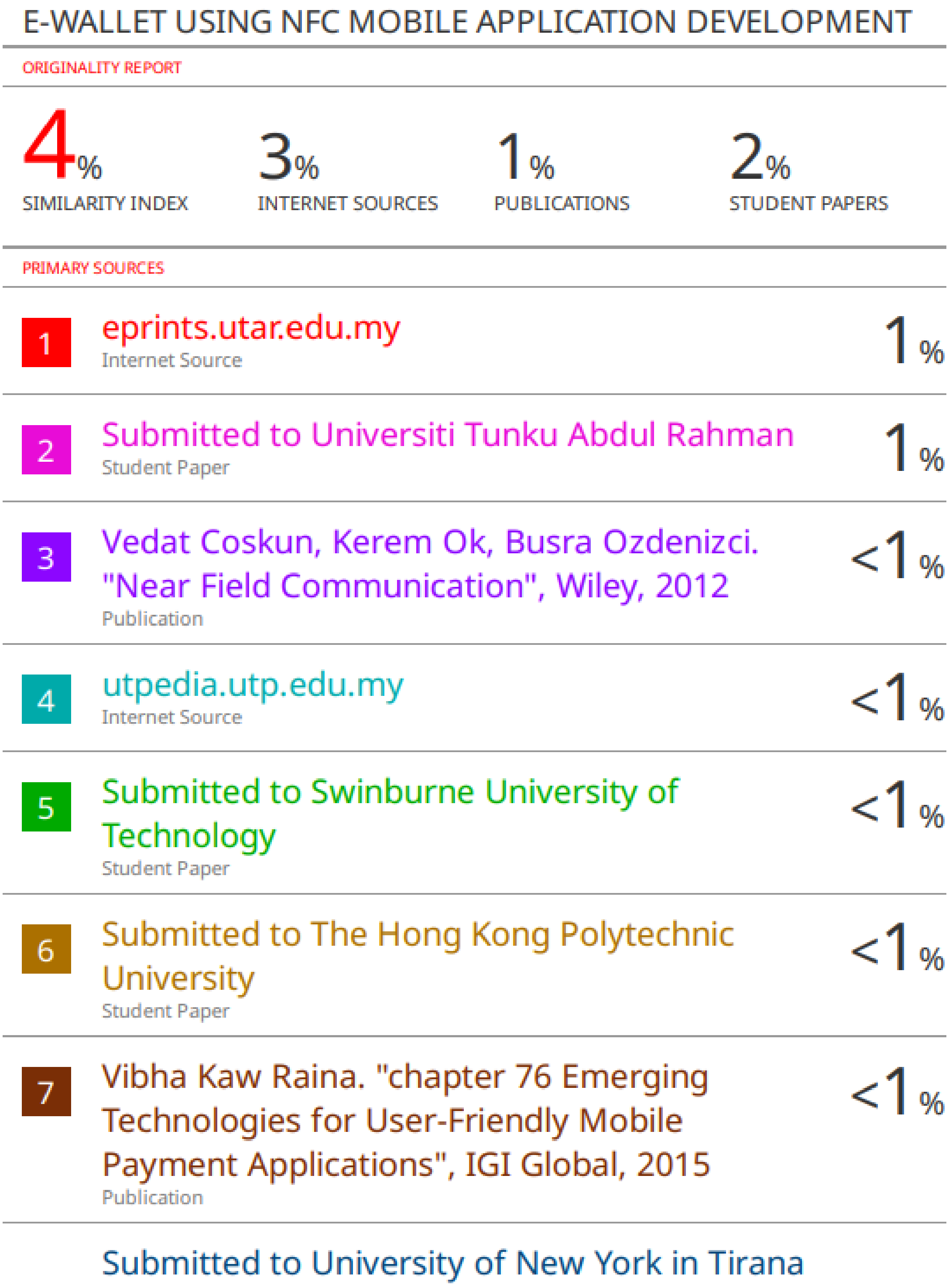
Supervisor
Ms Tan Lyk Yin

**University Tunku Abdul Rahman**

Bachelor of Information Technology (Honours)
Communication and Networking
Faculty of Information and Communication Technology

UTAR
UNIVERSITI TUNKU ABDUL RAHMAN

## PLAGIARISM CHECK RESULT



E-WALLET USING NFC MOBILE APPLICATION DEVELOPMENT

ORIGINALITY REPORT

| 4% | 3% | 1% | 2% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

|  |  |  |
|---|---|---|
| 1 | eprints.utar.edu.my<br>Internet Source | 1% |
| 2 | Submitted to Universiti Tunku Abdul Rahman<br>Student Paper | 1% |
| 3 | Vedat Coskun, Kerem Ok, Busra Ozdenizci. "Near Field Communication", Wiley, 2012<br>Publication | <1% |
| 4 | utpedia.utp.edu.my<br>Internet Source | <1% |
| 5 | Submitted to Swinburne University of Technology<br>Student Paper | <1% |
| 6 | Submitted to The Hong Kong Polytechnic University<br>Student Paper | <1% |
| 7 | Vibha Kaw Raina. "chapter 76 Emerging Technologies for User-Friendly Mobile Payment Applications", IGI Global, 2015<br>Publication | <1% |
|  | Submitted to University of New York in Tirana |  |

PLAGIARISM CHECK RESULT

| | | | |
|---|---|---|---|
| **Universiti Tunku Abdul Rahman** | | | |
| **Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)** | | | |
| Form Number: FM-IAD-005 | Rev No.: 0 | Effective Date: 01/10/2013 | Page No.: 1of 1 |

## FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

| | |
|---|---|
| **Full Name(s) of Candidate(s)** | Chong Yao En |
| **ID Number(s)** | 21ACB00289 |
| **Programme / Course** | CN – Communication and Networking |
| **Title of Final Year Project** | E-wallet using NFC Mobile Application Development |

| **Similarity** | **Supervisor's Comments**<br>**(Compulsory if parameters of originality exceeds the limits approved by UTAR)** |
|---|---|
| **Overall similarity index:** 4 **%**<br>**Similarity by source**<br>Internet Sources: 3 %<br>Publications: 1 %<br>Student Papers: 2 % | |
| **Number of individual sources listed** of more than 3% similarity: 0 | |
| **Parameters of originality required and limits approved by UTAR are as Follows:**<br>**(i) Overall similarity index is 20% and below, and**<br>**(ii) Matching of individual sources listed must be less than 3% each, and**<br>**(iii) Matching texts in continuous block must not exceed 8 words**<br>***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.*** | |

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

***Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.***

______________________________ ______________________________

Signature of Supervisor Signature of Co-Supervisor

Name: TAN LYK YIN Name: __________________________

Date: 15 SEP 2023 Date: ___________________________

# UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

## CHECKLIST FOR FYP2 THESIS SUBMISSION

| | |
|---|---|
| Student Id | 21ACB00289 |
| Student Name | Chong Yao En |
| Supervisor Name | Ms. Tan Lyk Yin |

| TICK (√) | **DOCUMENT ITEMS**<br>Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item. |
|---|---|
| √ | Title Page |
| √ | Signed Report Status Declaration Form |
| √ | Signed FYP Thesis Submission Form |
| √ | Signed form of the Declaration of Originality |
| √ | Acknowledgement |
| √ | Abstract |
| √ | Table of Contents |
| √ | List of Figures (if applicable) |
| √ | List of Tables (if applicable) |
| | List of Symbols (if applicable) |
| √ | List of Abbreviations (if applicable) |
| √ | Chapters / Content |
| √ | Bibliography (or References) |
| √ | All references in bibliography are cited in the thesis, especially in the chapter of literature review |
| √ | Appendices (if applicable) |
| √ | Weekly Log |
| √ | Poster |
| √ | Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005) |
| √ | I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report. |

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

chong

(Signature of Student)

Date: 14 / 09 / 2023