DIGITAL STUDENT ID CARD USING NFC TECHNOLOGY

By

Eng Mun Shuen

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2022

DIGITAL STUDENT ID CARD USING NFC TECHNOLOGY

By

Eng Mun Shuen

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2022

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Digital Student ID Card Using NFC Technology

Academic Session: January 2022

I Eng Mun Shuen

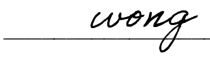
(CAPITAL LETTER)

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.

(Author's signature)

Verified by,



(Supervisor's signature)

Address:
1,Jalan Sri Klebang 5
Bandar Baru Sri Klebang
Chemor Perak,31200

Wong Pei Voon
Supervisor's name

Date: 22/04/2022

Date: 22/04/2022

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 19 / 04/ 2022

SUBMISSION OF FINAL YEAR PROJECT

It is hereby certified that ***Eng Mun Shuen*** (ID No: ***18ACB03271***) has completed this final year project entitled "DIGITAL STUDENT ID CARD USING NFC TECHNOLOGY" under the supervision of Ts Dr Wong Pei Voon (Supervisor) from the Department of Digital Economy Technology, 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.

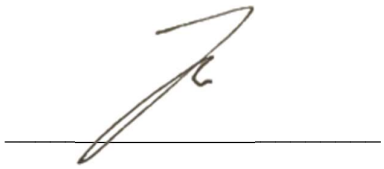
Yours truly,

(*ENG MUN SHUEN*)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**DIGITAL STUDENT ID CARD USING NFC TECHNOLOGY**" 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 : 

Name : Eng Mun Shuen

Date : 19-04-2022

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Ts Dr Wong Pei Voon who has given me this bright opportunity to develop a mobile application project. It is my first step to establish a career in mobile application 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

Most of the education institution, student and lecturer are given an identity card that could identify themselves. It is important to protect the personal information data on the card from identity thief in this era of big data. Besides, traditional Identity Card management is inefficient. Education institution which still using traditional ID card will waste a lot of resources on managing the cards. Not to mention that the ID card could not be reused, cause a lot of harm to the environment. Student ID card are designed to perform multiple functions. One of the functions is to take class attendance. Some institution uses different types of attendance taking method and these methods have their disadvantages.

This mobile android application project is designed to replace the traditional ID card. This application will give users a more secured authorized identity in the education institution area by registering each device's unique Android ID. Besides, this application provides attendance management function for admin and attendance taking function for student and lecturer. NFC technology will be implemented to become the attendance taking method. NFC technology is not only safe, but also could utilize the attendance taking process. Most importantly, it could increase the attendance rate and improve the attendance data accuracy. This application will be developed using Android Studio IDE and Firebase cloud database.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API	Application Programming Interface
CRUD	Create, Read, Update, Delete
GPS	Global Positioning System
ID	Identity
NFC	Near Field Communication
PVC	Polyvinyl chloride
QR	Quick Response
RAD	Rapid Application Development
RFID	Radio-Frequency Identification
SDK	Software Development Kit
SIM	Subscriber Identity Module
SQL	Structured Query Language
SDLC	System Development Life Cycle
Wi-Fi	Wireless Fidelity

CHAPTER 1

Introduction

This proposal is aiming to replace the traditional student card with digital mobile student card. There are a few problems with the traditional physical student card.

1.1 Problem Statement

i) Physical card can be easily mislaid

Student must bring along their identity student card with them when they are inside the campus area. Student ID card is used to prove student identity and they will be punished for forgetting their student card. However, there are many cases where student forget to bring student card with them. According to Sumathi Reddy, people misplace their belongings in an average less than 10 times daily and spending more than 10 minutes to search for the missing item [1]. These items could be keys, identity cards, mobile devices or another item that is small.

Generally, student ID card are only usable in school area and student could use the card to purchase food, borrow books from library or access building. Yet not all student's identity card from different school has all these features. Student tends to forget their student card when their student card is not important, and they have no issue to access school area and attending class. Besides, whenever student come back from school, they will take off their student ID card because these cards are unusable outside the school area. Then, students will face the situation to search for their misplaced ID cards tomorrow.

ii) Traditional card is unsecured

There is printed student's information on every student card. The general main elements of student identity card are printed picture, printed personal information, printed education institution information, student identifying number and so on. Besides, some advanced identity card has magnetic stripe, bard code, integrated circuit chip or RFID tag that could store student's personal information. However, this information are not carefully protected. First, information inside the magnetic strip is encoded according to international standards. People could decode and obtain the information by using magnetic strip equipment that can be easily purchased. Student personal photo could also lead to security issue. In some cases, leaking of student information could lead to

criminal cases. Identity thief could use the student information on the card such as picture to create some fake video or photo with your face. One of the threats of faking the video or photo could lead to serious image-abused and cyberbullying cases among the society [2]. Other than that, fake student card also has the security risk. People who printed a fake student card on PVC or plastic card could access the building anytime. Any educational institution does not welcome unauthorize visitors.

iii) Inefficient physical card management system

Traditional card is not easy to manage, and it will consume a lot of resources and time. To create a new student ID card, administration must purchase PVC cards and print student information on each of it. Whenever student want to update their student information, their student ID card might also need to be replaced with a new card. Administration must replace the old identity card and remake an updated card for the student. Besides, replacing ID card must go through the identity verification process and it will usually take times for the admin to go through the process. Usually, it might take hours or days to process the card replacement process.

1.2 Motivation

Mobile identity card also be defined as a form of digital identity that could represent a person's credentials through the digital devices, mobile network. [3]. According to an interview article written by Salim in 2016, the International Student Identity Card Organization hope to replace traditional physical student ID card with mobile student card. The reason behind this statement is the organization believe that mobile student card would bring a lot of benefits to students and might improve student learning experience in school [4]. There are motivations in solving the problem of traditional student card.

The first problem of using physical student card is student might forget to bring along their student card with them and they could be punished with that. Replacing physical card with mobile application can solve this problem. Nowadays people will bring their mobile phone with them all the time. People use mobile phone to perform different type of daily operation such as communication, learning, entertaining and so on. Compare to student card that are usually only usable within school area, student would not easily forget their mobile phone that could be use inside or outside school area. Therefore, replacing traditional student ID card with mobile phone could ensure student are capable to have a evidence of authority in their phone.

Protect student personal information will be the second motivation of this project. It is important to keep the student information confidential and secured. Identity thief could use this information to commit crime and danger the student. Mobile smart phone is one of the solution to solve this problem because smart phone could provide a lot of security protection features compare to physical card. There could be a lot of security layer inside a smart phone. For example, student could set a password, pattern, or pin on the smart phone. Besides that, the data inside the mobile application can be encrypted uniquely and it is not that easy to decode the encrypted data compare to magnetic strip which the data is encoded according to international standards. Moreover, there are also a lot of security features such as two-factor authentication, multi-factor authentication, biometric authentication, behavioural authentication and so on.

Utilizing the student identity card management will be the third motivation. It is important to have more efficient and effective system and reduce the time and effort to

manage the physical student ID card. This problem could be solved by replacing physical student card with mobile application. Student do not have to worry their student ID card will be lost since all the student information is stored permanently inside the cloud database. The cost and time to replace and update the student identity will be shorten.

Other than that, there are a lot of features from traditional physical card that could be also operate using mobile application such as student could use student card to take attendance. This mobile application also could perform the attendance taking features too. Using mobile application to manage attendance is more efficient. Some education institutions are still taking student attendance by signing on a list of students or using physical student card to take attendance. One of the main problems with these attendance taking methods is student faking attendance. People will sign for their friends who are absent. Besides, it is also difficult to manage the attendance record when lecturer or teacher record the attendance manually on paper. Then they must update the attendance record one by one into the database manually. To utilize the attendance management system for both student and teacher, mobile application will have a much better performance in handling the system.

1.3 Project Objectives

i) Enhance of the security to protect student information in mobile application

All student and lecturer information is protected inside the cloud database. Unauthorized user could not perform log in operation using another student and lecturer account into their device. Each user's Android - ID is unique, and it will be bind with each user's account once they registered an account for this application.

ii) Utilize the student attendance system by using NFC technology

Using NFC to take attendance could speed up the attendance record process. Student just tap their mobile device on the NFC tag and the attendance will be updated inside the device database. Other than that, the lecturer could collect and record the attendance automatically after students tap their phone on the NFC tag. It utilizes lecturer time instead of dealing with the student attendance data. One of the advantage of using NFC technology is simplicity and convinience. Comparing to other data storing method such as QR code and bar code, the cost is cheaper.

iii) To improve the attendance rate of student

Every student must appear physically in front of the NFC tag to record attendance and each student account is binding with only one registered mobile device. Therefore, there will be a reduce number of fake attendance and the attendance rate will be increase. Student could not sign the attendance for their friends and the accuracy of the attendance data will be improved.

1.4 Project Scope

i) Verified user identity using mobile application

Student and lecturer identity could be verified through this application. When user register with their identical student or lecturer ID into this mobile application, the Android Identity code from user's device will be retrieved and store on the database. Each registered student will be bind with one and only mobile device. Meaning that each student and lecturer could not log in into different mobile device after they registered themselves.

ii) Attendance taking method using NFC

Before student take the attendance using NFC chip on their phone, lecturer could write the unique class code inside an NFC tag. Whenever student attend to a class, student tap their phone on the NFC tag that prepared by lecturer and the attendance will be updated and recorded. Besides, teacher could retrieve and collect all student's attendance record from the cloud database using their mobile phone.

iii) Student Attendance management system

Student, lecturer, and the administrator could always view and check the attendance record on the application. All the attendance record is updated and stored inside the cloud database. They could view the attendance with their mobile phone. Administrator for this application could perform create, update and read operation on the attendance system and student information database. All these operations could be performed on mobile device and the modification will be updated on the cloud server.

iv) Can write message on the NFC tag

Every user could perform write and read message on the NFC tag. User could leave any short message into the NFC tag for notice purpose. Other than that, the code that encoded into the NFC tag can be also used to perform attendance taking function and people counter function.

1.5 Impact, significance and contribution

i) Attendance rate will be improved

Using NFC technology as attendance taking method will improve the attendance rate. This is because student could not fake their attendance. Student must present to the in front of the NFC tag and tap their phone on the NFC tag to take attendance. Besides, each student could only bind one smart phone to the mobile application. Students are not allowed to register same mobile device and log in into the multiple account in more than 1 devices. Therefore, student must physically attend to the class to take attendance. When the attendance rate is improved and accurate, it could help an education institution to have a deeper understand on the student condition based on the attendance rate. Using the attendance data could also help the education institution to design and some plans and strategy to improve student. According to GreatSchools Staff in 2011, higher attendance rate also mean students will have higher changes to succeed in their academic [5]. Having more excellent student also will boost the ranking of the education institution and become more successful.

ii) Stronger security protection for student

Replacing traditional student card with the mobile application could secure student. Besides, using NFC technology to take attendance is more secure compare to QR code. It is proven by Deliyannis, NFC perform better in security because the user must get close to the NFC tag or NFC device to perform data transaction operation [6]. On the other hand, QR code is not secure compare to NFC because user could access to the URL encoded into the QR code easily by using phone camera. It is very important to protect student private information and it could cause a lot of impact. Criminal might use the leaked student information to perform criminal activity such as using student social identity number to register for a credit account. Besides, student private and confidential information such as address could also attract the thief and perform any criminal activity. It is necessary to stop using traditional physical student card and replaced it with the mobile student card which could provide more security protection on student information. Compare to traditional student card that student information is printed on the card, student information is protected inside the mobile application using password protection, biometrics protection, and data encryption.

iii) Save environment

Replacing the traditional student card could save the environment. Traditional student card is usually made of plastic or PVC card and it will take 450 years to decompose. Besides, the traditional student card has the expire date. The card will be useless after the expire date. Since it takes hundreds of years to decompose, it is harmful to environment. Replacing the physical card into bit and bytes could save the environment. Moreover, it is cheaper to producing some bits and byte data that represent a student identity inside mobile application rather than make a physical card. The cost of producing a physical card include manufacturing cost, serving cost, transportation cost and so on. When student change their name or any personal information, they need to change the personal information on their campus card by changing a new card. By using this mobile application, the admin can modify and edit the student information easily and student do not have to change card. Other than that, the NFC tag could be reuse multiple time.

iv) Improve productivity

Replacing the traditional student card and implement the attendance system into the mobile application could utilize the student attendance and student information management system. Lecturer could have more time to focus on delivering the lesson to students. Lecturer could get rid of the human error when updating the student attendance. The attendance accuracy will be improved when the data is updated automatically using mobile device. Other than that, lecturer have more time to deal with their research, planning teaching method, preparing teaching materials and other duties of lecturer. Therefore, lecturer could improve the working productivity and bring more value for the University or other education institution.

1.6 Report Organization

There are 7 chapters in this report which are Introduction, Literature Review, System Methodology/Approach, System Design, System Implementation, System Evaluation and Discussion and Conclusion.

Firstly, chapter 1 will introduce problem statement, motivation, project objectives, project scope, impact significance and contribution. In this chapter, an overall introduction to the purpose and background before developing the application and report.

Secondly, a introduction on hardware platform, firmware, Operating system, database, programming language and Integrated development environment that involving in this project will be written. Then, several similar applications will be reviewed and study.

Chapter 3 consists of system design diagram such as System architecture diagram, use case diagram and description, and activity diagram. The overall design of the system component and end-user interaction with the system will be shown in this chapter.

Chapter 4 consists of system block diagram, system components, system flowchart. The application development steps will be showed here.

Chapter 5 consists of system implementation where it showed the how the system operates.

Chapter 6 consists of system evaluation and discussion. System will be going through testing phases to ensure the quality of application could be maintained. A several scenarios of test case will be tested, and the test result will be showed. Then, project challenges and evaluation of objectives will be written in the same chapter.

Chapter 7 consists of conclusion and recommendation of the project. This chapter will conclude the whole development process and mentioning some potential that can be develop from this application.

CHAPTER 2
Literature Reviews

In this literature review, there will be 8 mobile application that will be reviewed. These 8 applications are mainly used by campus student. These applications are designed to replace some of the functionalities of a physical campus card.

2.1 Review of Technologies

2.1.1 Hardware Platform

Android (Operating System) devices

In 2005, Andy Rubin, Nick Sears, Chris White and Rich Miner founded Android company. At the same year, this start up company was owned by Google. In 2007, Open handset Alliance (OHA) was formed to boost up the techology and invantion of mobile phone devices and improve the user mobile experience.

Android OS mobile device market in March of 2022 is 71.7% followed by iOS which has 275%. Therefore, it is easier to obtain a Android OS device in the market as there are more than 70% of the phone in market is Android based.

Near Field Communication

NFC chip is one of the main technology involve in this project. NFC also known as Near Field Communication. NFC is mainly used to build up a wireless connection between two interface where the distance is limited within 10cm [7]. The NFC I/O will control the flow of data between devices. NFC are usually support three types of operation modes. Reader/writer mode, peer-topeer mode and card emulation mode. There is also active mode and passive mode in NFC. Active NFC device will generate the magnetic field by itself and usually this device has their own electrical power supply. On the other hand, passive device does not generate the magnetic field itself [8]. Read/writer mode allow the NFC smart phone to read or write the information or URL on another passive NFC tag [9]. The NFC enabled smart phone will generate the RD field and activate the passive NFC tag. The callback model in Android NFC stack support reader/writer mode and this mode involve one active host device while the other side is passive device

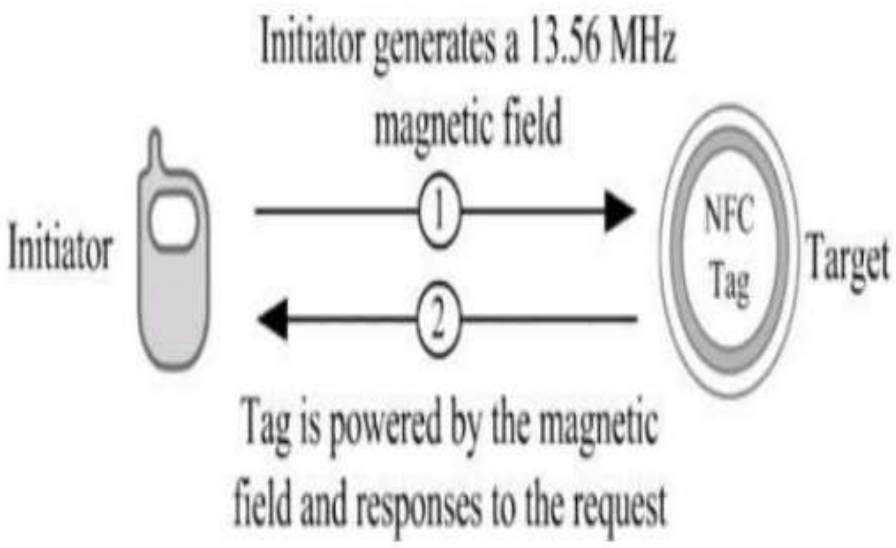


Figure 2.1.1 NFC magnetic field from initiator and Tag

NFC tag

NFC tag that will be using in this project is NFC Ntag 213. The specification of NTAG213 is as follow:

- Could store up to 180 bytes of data
- Could encode URL link up to 132 characters
- Could encode text string up to 130 characters
- Could store the data not less than 9 years
- Could perform read and write operation not less than 100,000 times
- Provided password security function

Work perfectly with all mobile smartphone that supported with NFC

2.1.2 Firmware/OS

Android (Operating System)

Based on developer prespecktive, Android OS has a more open development environment as developers could have more access to the hardware, file system and so on. It is a huge differences compare to iOS development environment. Therefore, the cost and budget of development is less.

Android API requirement

Since this project required NFC write and read function, Android API level 10 which the code name is Gingerbread is required to use to develop the application. Besides, developers would need to have an NFC available android device to perform testing and debugging process.

2.1.3 Database

It is important to have a place to organize the information data needed by the project. A proper database could help developer to manage and access the structured data easily. This project also requires a lot of data information. For instance, the data of student, lecturer, class, subject, attendance and so on. These data will be needed for later use.
Firebase

It is a platform that developers could develop their mobile and web project. Google owned this platform in 2014. Firebase provide a lot of service for developers such as authentication, cloud “Firestore” database, realtime database, file storage, hosting, functions and machine learning services. It is a non-SQL database which will provide developers more flexibility to develop and manage the database. Other than that, the data will be transfer and synced across the device and Firebase in real time. Client will receive the latest data immediately.
User could perform register and login function by using the authentication features introduced by firebase. Email and password that retrieved from users’ input will be saved into firebase authentication database automatically. Student, lecturer, classes, subject and attendance data are store in the cloud firestore database. Other than that, student and lecturer photo could also be saved in firebase file storage database.

Since developers would not need to query SQL statement to perform any database transaction, developers could get rid of complicated SQL query.

2.1.4 Programming Language

Java programming language is firstly published to public in 1996. It is a object-oriented programming language and could be use to develop application in multiple platform.

Integrated development environment (IDE)

Android studio will be chosen as the main IDE to develop the application. Android studio is Android official IDE. It provided all the necessary tools for developer to code, debug and test the project.

Android Studio allow developer to have the flexibility to build the system using Gradle-based build system. It provides binary libraries and variant support, and it will ease the developers to process configuration and customization.

Drag and drop features in Android Studio will reduce the effort to develop the user interface. Therefore, develops could have more options and flexibility to design the user interface.

2.2 Review of the Existing Systems/Applications

2.2.1 Hi-Hive, MSU Mobile, UM Touch

Hi-Hive Community application is introduced by SILVERLAKE company. It is a software that mainly focus on building up a community ecosystem. In this community ecosystem, user could communicate with other users and just like social media. User could select and join different community and communicate with each other within the same community. University Tunku Abdul Rahman (UTAR) is one of the communities inside the Hi-Hive. When student from UTAR had successfully register into the UTAR community through Hi-Hive application, they can take the attendance during the class by using QR code scanning. This function successfully replaced one of the functions that provided by physical campus card.

Figure 2.2.1 Logo of Hi-Hive application

University of Malaya introduced their mobile application, UM Touch. This mobile application provides features such as use as mobile credentials, campus announcement, events timetable, cashless payment, academic calendar, course registration, QR code attendance and so on. This mobile application is available in Android and IOS smartphone. This application not only for student and staff of University Malaya, but it is also opened to public visitors too.

Figure 2.2.2 Logo of UM Touch application

Management & Science University in Malaysia also introduced the mobile application, MSU Mobile. This mobile application can also be used to perform QR attendance, P2P Transfer, QR payment and mobile credit Top-up. Other than that, it can also record down extra-curricular events attendance or other attendance reporting.

Figure 2.2.3 Logo of MSU Mobile application

UTAR student used Hi-Hive to take their attendance by scanning location QR code or attendance QR code (Figure 2.4). These QR code will be generated and encrypted by Hi-Hive system. Before these QR codes are distributed to students, lecturer must activate it. Other than that, lecturer could also modify and the attendance details list before distributing to student. Lecturer can access to the server of SILVERLAKE and check on the attendance event and attendance report. Student could also check on their attendance on QR Class Attendance Record (Figure 2.6).

Figure 2.2.4 Location QR code and attendance QR code

Figure 2.2.5 Inactivate attendance QR code

Figure 2.2.6 Attendance record

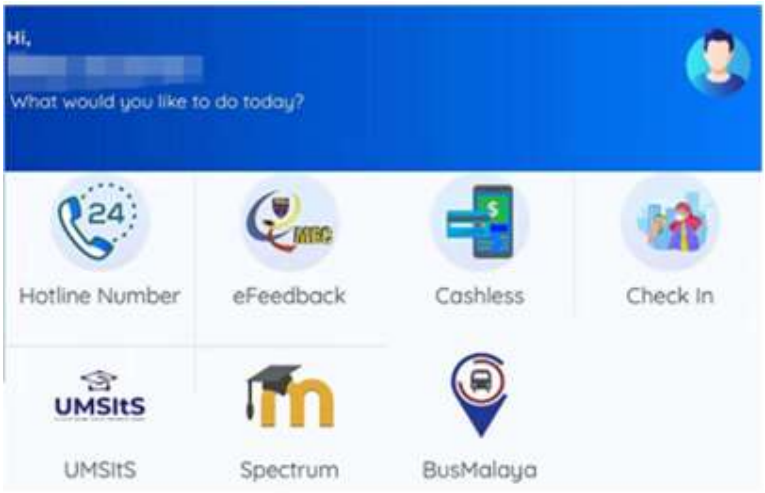


Figure 2.2.7 UM touch interface

Figure 2.2.8 UM touch QR code attendance

Benefit of using QR code attendance

Using QR code technology to take attendance is convenient, fast and easy. All student has to do is just open their camera and scan on the QR code. Student attendace will be recorded and uploaded to database server. Lecturer and staff can collect all the student attendance data and make report effortlessly without having to key in all the student attendance into the database server one by one. Most of the attendance taking method in UTAR is old and inefficient. Lecturer must distribute the attendance list for student to check on their attendance or even call out the name to take attendance. This method is inefficient because the attendance list is in paper form and it can be easily lost or broken when this list is passing around students. Sometimes it can be more than 100 students in one class and it is very time consuming for all student to take the attendance. Therefore, QR code attendance could be the solution of this problem. Besides that, it is also very cost effective and cheap compare to other attendance taking method such as print out a student attendance list which will waste a lot of paper. QR code is popular and user friendly since the cost to implement the system is very low [10].Not to mention traditional attendance taking method is extremely time consuming. QR code attendance save the time of student, lecturer and administrator staffs. Moreover, QR code can be very flexible and a lot of information could be encrypted into the QR code. For example, Hi-Hive attendance system is used the dynamic QR code attendance and this type of QR code is very flexible. Admin can adjust the details or data of the QR code anytime. Furthermore, the Dynamic QR code is trackable and the location QR record can also work with GPS and keep the location of the student when they scanned the QR code.

Limitation of QR code attendance

There are some limitations of this application on the QR technology. First, this QR class attendance is using Dynamic QR code. Inside dynamic QR code, there is a small URL that encoded inside the QR code will redirect user to the final URL destination. User will be redirect two times to the final URL when user scan the attendance QR code. When student could not access the first small URL that is encoded inside the QR code, user could not redirect to the destination. By doing this way, it will increase the complexity of the data transfer from student mobile application to the SILVERLAKE server since the data has to redirect two times.

Moreover, QR code could be unsecure. First, user do not know what information or URL that encode inside the QR code. The QR code can be printed on a paper and it can be easily replaced by anyone else without other people knowing. User also could not identify the authenticity of the QR code. Some people might encode some URL or program that will steal user information inside the QR code. Moreover, using QR code to take attendance can be slow when there was some internet connection issue. It is because user will be redirect to an URL to take the attendance and it require a stable internet connection. When someone internet connection is bad, it will take a long time to take the attendance. Other from that, user could not scan the QR code when the camera is too far from the code. The angle, light and the distance will affect the QR code decode process. The location QR code has its own limitation too. Student can fake the GPS location by using third party application.

2.2.2 TARCApp

Figure 2.2.9 Logo of TARCApp application

Tunku Abdul Rahman University College (TARUC) also introduced their own application for student and staff, TARCApp that can perform the same online attendance method. The ways of TARC student take their attendance was student connect to the TARUC Wifi and enter the code of the attendance given by the lecturer during the class. Other than that, it also has the features such as providing news, virtual tour, online library, checking on offered programmes, student academic timetable and result and so on. There are more than 10,000 Tunk Abdul Rahman University College student downloaded and install it in Android and IOS device.

Advantages of using Wi-Fi attendance

The advantages of using Wi-Fi attendance can detect the mac address of the mobile device and complete the authentication process. Beside that, all smart phone in the world have Wi-Fi connection features. Which means that if students have a smart phone, he/she can attend the class by using this technology. It is very cost efficient and secured at the same time. Compare to other attendance taking method such as taking attendance through QR code and NFC, some of the smart phone do not have the ability to scan QR code on their camera or even some of the people do not have NFC chip inside their smartphone. Using Wi-Fi to take attendance can be implemented to the public and accepted by public much easier.

2.2.2 Limitation of using Wi-Fi attendance

The limitation of this attendance taking method is student can go anywhere else in the TARUC that provide Wi-Fi connection and sign in their attendance although they are not in the class. This is proven by Ong Chong Keat who is currently studying Mechanical Engineering in TARUC, he said that many student skip class because of this attendance taking method, they can sign in their attendance by connecting to the library Wi-Fi or any Wi-Fi access point in campus area and enter the attendance code provided by teacher even they are not inside the class. It proved that this system does not track the location of Wi-fi access point. Besides that, student can also log in into different account to take their attendance with one smart phone. It is also proved that the anyone could access to the account. This will cause a serious security problem because student private identity information could be stolen by identity thief.

2.2.3 CBORD Mobile ID

Figure 2.2.10 Logo of CBORD Mobile ID application

CBORD Mobile ID is introduced by The CBORD Group, Inc. This application is used for replaced the ID card and it perform functions like building access, attendance marking, contactless payment in dining area or make payment at vending machine. Kenyon College, West Virginia University, Carnegie Mellon University and many more universities is subscribing this service. After user install this application, they are required to connect to their school server. After they have successfully connected, student is required to enter their campus credentials to gain the authentication of this application. After that, this application will be able to access to user campus account.

To access to a door, user can open the application and their GPS location, the surrounding door will be appeared based on the location of the user. After that user choose and select the access number and tap GO to request the approve from the server to access the door or building. Then, the door will be opened automatically if the user is given approval to access the building. If the door is only support pin pad, user is required to enter pin number that are randomly generate to you every time through this application.

The attendance system of Mobile ID use Bluetooth and WI-FI technology. Lecturer and student mobile device will connect with each other and that is the way the attendance will be recorded. Google Nearby is used to take attendance. Ultrasonic sound, Bluetooth and Wi-Fi access point are the key element of this features. First, ultrasonic sound wave that produced from your phone by Google Nearby will be

detected by other devices microphone. Besides, the Wi-fi connection will match the nearby Wi-Fi access point to determent the current location of student and there will be a token broadcasted through the Bluetooth and other devices within the same class could be able to detect it. Therefore, there is no way a student could skip the class and fake the attendance.

Limitation

The limitation of this application is it heavily depends on the internet availability. When there is no internet access around them, they could not take the attendance. Once user do not have the internet connectivity, Mobile ID could not determine the whether the device is located at which Wi-Fi access point. Student also could not turn on the location services when there is no internet connection.

Another weakness of this application is user must select the specific door to access it by searching through a long list of door number. Besides that, it heavily depends on the GPS service which will locate the nearest gate around you. This technology is used to prevent some people from accessing and remote the door if the person is not within the reader. However, there are many factors that will affect the accuracy of GPS signal. GPS is not accurate when there is a satellite signal blockage which cause by the building and there is something blocking the signal from an open sky [11]. Besides that, student who is in indoor or underground will have the problem of inaccurate GPS. There will be some mis-leading GPS signal when the satellite signal is reflected off walls or building. Besides, it will usually take 5 seconds to unlock the door because a long backbend process is needed to process in order to open the door.

Furthermore, when user want to borrow book or purchases food in cafeteria, user must show the bar code on the app. Bar code reader could not scan the bar code when the bar code is directly under the sunlight. Sunlight will introduce some effect that make barcode reader will read the light reflected by the bar code. Therefore, bar code reader could not see the bar code when the environemt light frequency affect the light frequency that come from the scanner. User could not scan the bar code to access the door, purchases food or borrow books when user is under the sun from a clear sky.

2.2.4Transact eAccount

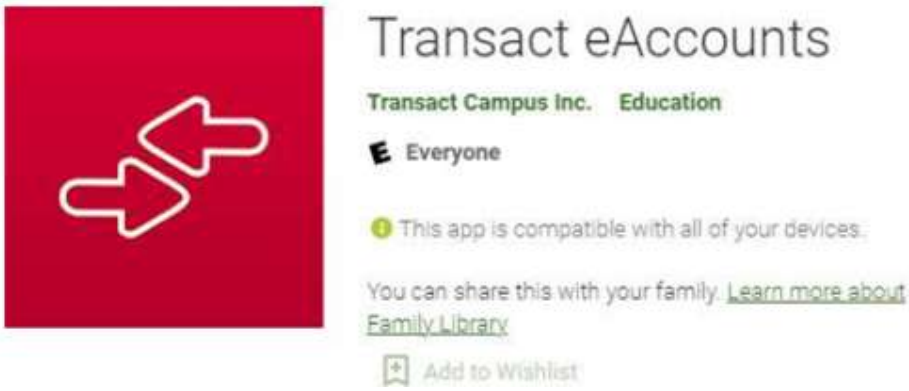


Figure 2.2.11 Logo of Transact eAccounts application

TransacteAccount is introduced by Transact Campus Inc. Transact eAccount is another software made for campus student. User could add their campus card into this application and make sure that the university has subscribe to their service. After Duke student successfully verified themselves in Transact eAccount, then student can add their campus card into the Apple Wallet or Google Pay.

Advantage

In 5th Jun 2018, Apple announced that Apple Wallet in iPhone and Apple watch supports student campus card or ID cards with the release of iOS 12 and WatchOS 5 which the patch will be released at 17th September 2018. Duke University is one of the University that selected in integrate the campus card into Apple Wallet. Duke student can replace their DukeCard with apple wallet which provide the same functions. Duke student can access building, make purchase in cafeteria, vending machines, laundry services and other functions that can be done with DukeCard. NFC chip inside Iphone and Apple watch play an important role in completing this function. Before, student in Duke university can integrate their campus card into their phone, student is require gaining access and permission from the Transact eAccounts. When university subscribe to Transact Campus Inc to use the mobile credentials service, each student or staff inside the university information will be store inside the cloud.

Limitation

However, there might be some limitation when education institution decides to use third-party services to manage student activities. Using third party service has the security trust issue. This is because all the student behavior information on the application could use as a useful message for the application provider. This information is known as metadata. Metadata is the useful information that can be obtained from another set of data set. For example, the click through rate of the application, time duration on the application, number of payment transaction per day, bounce rate and so on. Some people could analysis these data and make some business marketing decision. This is an illegal activity if education institution does not notice that third party is using this information for business use. Most of the university in Malaysia student information will also be protected by the Personal Data Protection Act 2010 (PDPA) and university will follow the law and guidance to handle student information. However, this is not included metadata. In United State of America, a lot of online educational services will collect metadata of student based on the student interaction with the mobile application. Since student are one the customers of these third-party application, therefore, there will be a high risk in exposing customers behaviour to the third-party company.

2.2.5 My-UU application

This is the official application is for Utrecht University launched on 2016. Inside the application, student could use this application to as a digital student card to prove that student is enrolled at Utrecht University.

Figure 2.2.12 Logo of MyUU application

This application will fully replace the physical student card start from 1 June 2017. In this student card, there will shown academic year, study programme, name, date of birth, student id and student passport photo. Student must log in into the application by using the student id. Based on the benefits they stated on the website, My-UU application is innovative, sustainable, save, convenient. It is more eco-friendly because it reduces the number of physical cards that made from plastic or paper. Besides that, this student ID card will not be stolen or lost from your mobile phone if student information is still on the university server.

One of the features of this application is student can always use this application even there is no internet access. The only requirement to use application in offline condition is student must log in into the application once and student must ensure themselves is enrolled into the university. Student could use this application to view and manage their timetable by themselves. Besides that, student can also check their academic result, their course timetable, campus map, from this application. Student could check their exam timetable and weekly schedule.

Limitation

However, there are some weakness of this application compare to other digital campus id. When the internet connection is lost, all the timetable will not be seen unless user screen captures it and save it into their phone. Compare to other Transact eAccount application that support NFC features, it provides more function and more secure compare to My-UU application. Moreover, any student could log in into the application multiple times and this could leak the private information of a student.

2.2.6 UiTM iStudent

Universiti Teknologi Mara (UTM) also introduced their own mobile application which is only targeted for UTM students. The mobile application is UiTM iStudent.

Figure 2.2.13 Logo of UiTM iStudent

This application provides basic student information and it can be used as a virtual student ID card. Other than that, it also has the student class timetable and their academic result.

Limitation

This application also do not has any security protection on it and other user could easily steal the information of the student. When a user knows another student's portal ID credential, anyone can log in into the system and steal your information. On the other hand, there is also another application that perform almost the same function with UiTM iStudent, UiTM Digital Campus.

UiTM Digital Campus also perform the could be use for mobile identification for students, staff and visitors of UTM. Same as the above, it does not perform much features and user can easily log in into the system.

2.3 Summary of the problem

Problem 1: QR code security issue & durability

People can always modify the QR code or replace the QR code into fake QR code since some of the QR code is only print on a paper.

Solution

Replace QR code with NFC technology that is more secure. This is because the NFC needed specific chip to be operate. NFC chip will normally be protected with a case and it is inside the smart phone. It is difficult to steal and exchange the chip which is protected. However, QR code normally be print on a paper and it is very easy to exchange the original QR code to another fake QR code. Moreover, using NFC chip not only more durable compare to QR code that printed on paper, but it is very easy to encode message into the NFC chip. The device that will be used in this project is protected well with the case.

Problem 2: Incomplete student information protection

Application that only rely on the student campus id and password is not secure enough

Solution

When user install and register the application, the device's Android ID will be taken and register inside the cloud server. This will prevent other people to install and access your information on their device. NFC technology is also used to protect and secure the information of students. According to GSMArena, using NFC is more secure than using Bluetooth connection because of the short-range communication. Student need to put their NFC smart phone device close to the NFC reader or another NFC device to start a communication. The distance should be maintained at least 10cm. Therefore, student do not need to worry their information will be stolen from another people from a far distance. Therefore, NFC is very suitable to be used in public area such as attendance taking or building access.

Problem 3: Ineffective QR code, Wi-fi attendance

Student could take the attendance even though they are not inside the class. QR code attendance and Wi-Fi attendance can be sign by other people or sign at another place.

Solution

Binding the student smart phone device with the application will make sure there is no "attendance buddies" who sign attendance for another person or student sign the attendance from the outside of the class. NFC attendance system will require student to sign in and sign out before and after the class. Besides that, there will be time limit for the attendance. Using NFC technology to take attendance is so much secure and efficient compare to QR code attendance. QR code can be easily replaced since it is extremely easy to produce a fake QR code.

Table 2.3 Mobile application attendance method

	QR code	Wi-fi	NFC	Bluetooth
Hi-Hive	Yes	No	No	No
MSU Mobile	Yes	No	No	No
UM Touch	Yes	No	No	No
TARCApp	No	Yes	No	No
CBORD Mobile ID	No	Yes	No	Yes
Transact eAccounts	No	No	Yes	No

CHAPTER 3
System Methodology/Approach

3.1 System Design Diagram

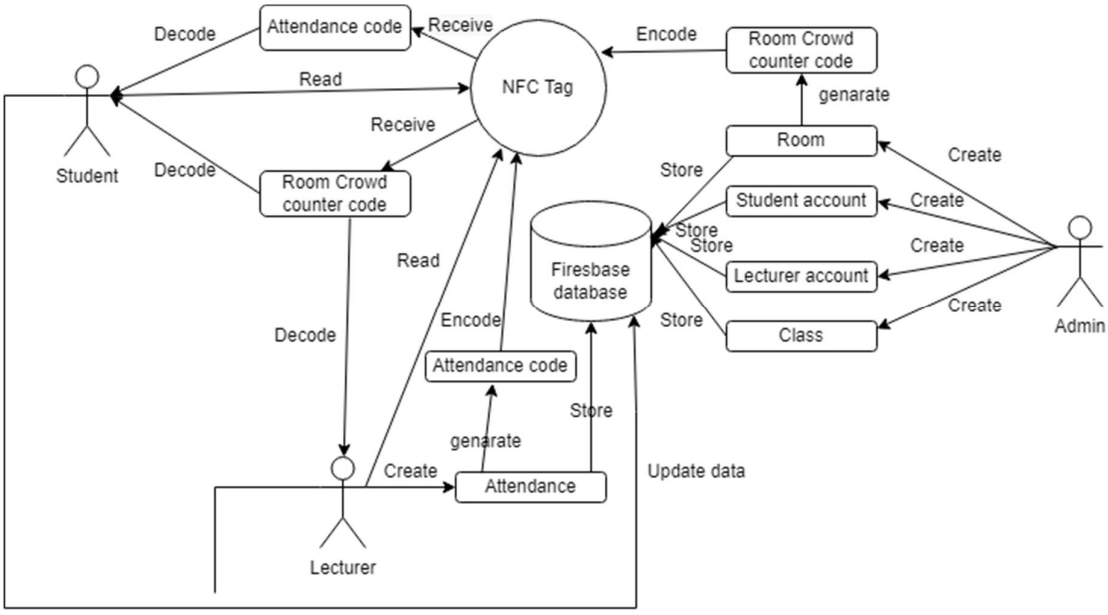


Figure 3.1.1 System Architecture Diagram

The system architecture diagram shows the relationship of all the elements involved in the system. It provides a model that describe the concept of the system. The main actor of this systems are Student, Lecturer and Admin. Besides, the Firebase database will be the data storing center for this system. Besides, NFC tag will also be involving in this system as it helps to store attendance code and room crowd counter code.

Firstly, the end user will be using this system through the mobile device that had already install the application. End user will interact with NFC tag and Firebase database through the application in their mobile phones. User will input data from the UI of the mobile application and processed with the business logic within the application. Then system will perform data transfer with the logic result with the database and NFC tag.

3.2 Use Case

The diagram below shows the use case diagram of mobile student ID card with NFC technology. Three actors using this application which are student, lecturer and admin.

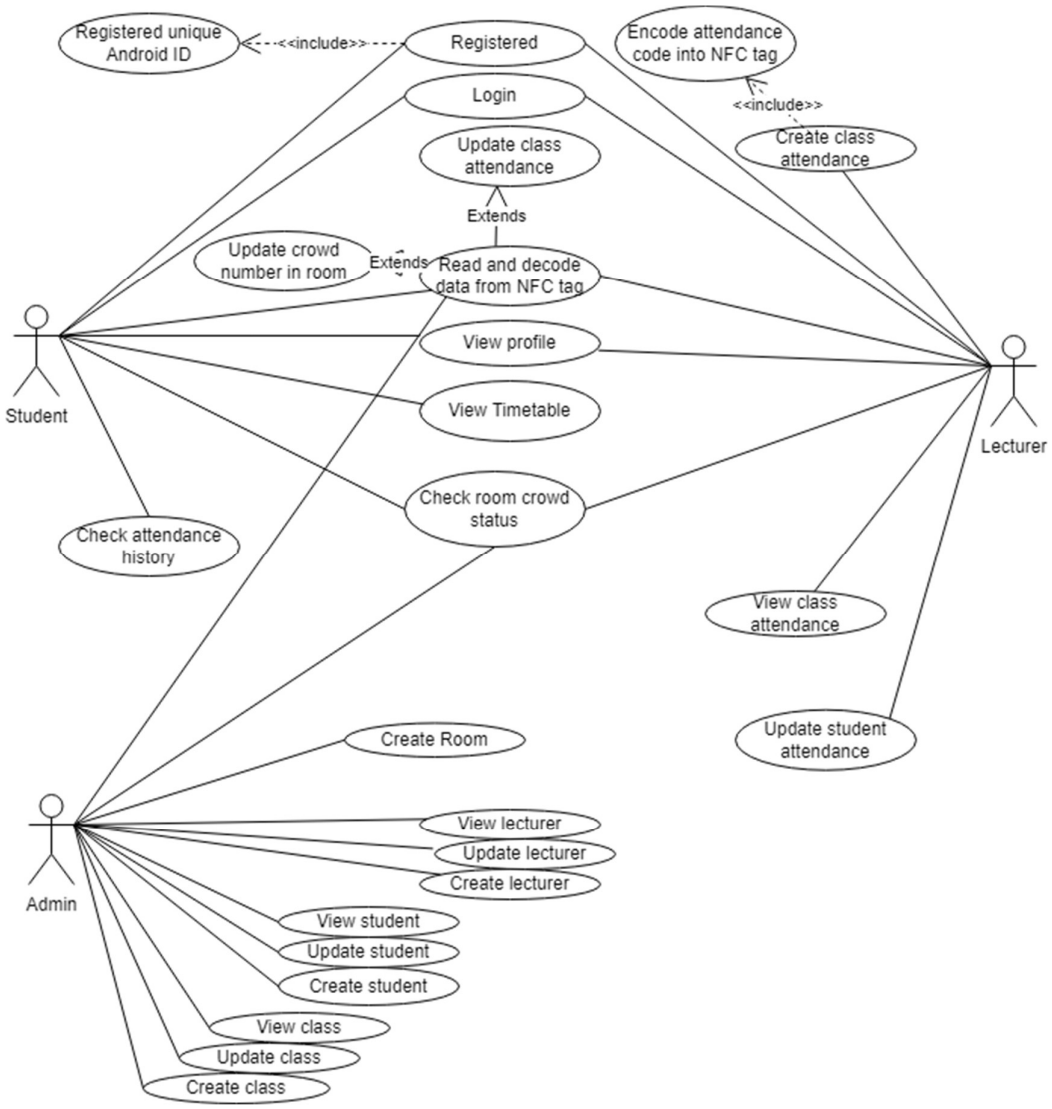


Figure 3.2.1 Use case diagram

Student and lecturer could perform login, register, register unique device android-decode NDEF data from NFC Tag, update crowd number, view profile, view room crowd status. Based on the diagram, it showed that the student could check their attendance history, take attendance using NFC tag, view their student information and view timetable. On the other hand, lecturer could perform create class attendance by

encoding attendance code into NFC tag. Lecturer also could update the student attendance and view their class attendance.

Admin who has given more authorities in managing this system will have more functions. Admin could the four basic operation which is create, read, and update on lecturer, student, class record in Firestore database. Oher than that, admin is authorized to create a new room that could perform Room Crowd Counting function. Admin could have the authorities to check the current Room Crowd status. Different actor could perform different functionalities in this application

3.3 Use case description

Table 3.3.1 Register account use case

Use Case ID	UC001	Version	1.0
Use Case	Register account		
Purpose	To registered user using their ID number for an account in this application		
Actor	Lecturer, Student		
Trigger	Clicks on “Register” button		
Precondition	-		
Normal flow of events	1. User enters the ID number, email, password. 2. Clicks on register button. 3. System perform verification on the device Android ID and the input ID number 4. System sent an update to database server 5. Redirect user to the main screen.		
Alternate Flows	3a. System displays an error message if the Android ID has been registered. 3b. System displays an error message if the ID number has been registered in other mobile device		
Author	Eng Mun Shuen		

Table 3.3.2 Login use case

Use Case ID	UC002	Version	1.0
Use Case	Login		
Purpose	Allow authenticated user to login into their account.		
Actor	Lecturer, Student		
Trigger	Clicks on “Login” button		
Precondition	-		
Normal flow of events	1. User enters the email, password. 2. Clicks on login button. 3. System perform verification on the device Android ID and the input ID number 4. System authenticate user 5. Redirect user to the main screen.		
Alternate Flows	3a. Display an error message if the user’s Android ID is not match with the registered Android ID under the user’s account. 4a. Displays an error message if database failed to authenticate user.		
Author	Eng Mun Shuen		

Table 3.3.3 Create student/ lecturer use case

Use Case ID	UC003	Version	1.0
Use Case	Create student/ lecturer		
Purpose	Create each student and lecturer a new record of their identity information for this application		
Actor	Admin		
Trigger	Click on “Create student” or “Create lecturer” button		
Precondition	-		
Normal flow of events	1. User enters the student/lecturer ID, First name, Last name, Faculty 2. Users insert profile picture of the student/lecturer to system 3. Users click on create button 4. System validates the input data 5. System connects to respective document in Firestore database 6. System adds the student/lecturer information to the database 7. System displays success message		
Alternate Flows	4a Display an error message if input is invalid 6a Display an error message if system failed to add new record to database		
Author	Eng Mun Shuen		

Table 3.3.4 Create class use case

Use Case ID	UC004	Version	1.0
Use Case	Create class		
Purpose	Create a class that recorded data of students, lecturer, class day, class time, class subject		
Actor	Admin		
Trigger	Click on create button		
Precondition	-		
Normal flow of events	1. User input class details such as class ID, lecturer ID, class day, class time, class subject and classroom 2. User click on next button 3. System redirect user to select student 4. User select students ID 5. System validate the input value 6. System connect to the respective document in Firestore database. 7. System create the new class information in the database 8. System display success message		
Alternate Flows	5a. Display an error message if input is invalid 6a Display an error message if system failed to create a new record in database		
Author	Eng Mun Shuen		

Table 3.3.5 View class use case

Use Case ID	UC005	Version	1.0
Use Case	View class		
Purpose	User could view the details of the created class		
Actor	Admin		
Trigger	Click on view button		
Precondition	-		
Normal flow of events	1. User input class ID wanted to be view 2. System validate the input value 3. System validate the existence of the ID in database 4. User click on next button 5. System connect to respective document in Firestore database 6. System display the class detail information		
Alternate Flows	2a Display an error message if input is invalid 3a. Display an error message if ID is not existed in database 4a Display an error message if system fail to connect to database		
Author	Eng Mun Shuen		

Table 3.3.6 Update class use case

Use Case ID	UC006	Version	1.0
Use Case	Update class		
Purpose	User want to update the class with the latest information		
Actor	Admin		
Trigger	Click on update button		
Precondition	-		
Normal flow of events	1. System connect to respective document in Firestore database 2. System display out the class list 3. User click the any class on the list 4. System redirect user to update the class 5. User input the information wanted to be updated 6. User click on update button 7. System update the new class information in the database		
Alternate Flows	1a. Display an error message if system fail to connect to database 7a Display an error message if system failed to update a new record in database		
Author	Eng Mun Shuen		

Table 3.3.7 View student / lecturer use case

Use Case ID	UC007	Version	1.0
Use Case	View student / lecturer		
Purpose	User could view the details information of the created student / lecturer		
Actor	Admin		
Trigger	Click on view button		
Precondition	-		
Normal flow of events	1. User input class ID wanted to be view 2. System validate the input value 3. System validate the existence of the ID in database 4. User click on next button 5. System connect to respective document in Firestore database 6. System display the student / lecturer detail information		
Alternate Flows	2a Display an error message if input is invalid 3a. Display an error message if ID is not existed in database 4a Display an error message if system fail to connect to database		
Author	Eng Mun Shuen		

Table 3.3.8 Update student / lecturer use case

Use Case ID	UC008	Version	1.0
Use Case	Update student / lecturer		
Purpose	User want to update the student / lecturer personal identity with the latest information		
Actor	Admin		
Trigger	Click on Update button		
Precondition	-		
Normal flow of events	1. System connect to respective document in Firestore database 2. System display out the student / lecturer list 3. User click the any ID on the list 4. System redirect user to update the student / lecturer 5. User input the information wanted to be updated 6. User click on update button 7. System update the new student / lecturer information in the database		
Alternate Flows	1a. Display an error message if system fail to connect to database 7a Display an error message if system failed to update a new record in database		
Author	Eng Mun Shuen		

Table 3.3.9 Create Room use case

Use Case ID	UC009	Version	1.0
Use Case	Create Room		
Purpose	Create a room record that could count the crowd number inside the room		
Actor	Admin		
Trigger	Click on create button		
Precondition	-		
Normal flow of events	1. User input the room details : Room ID number and maximum number of crowd 2. System validate the input data 3. System connect to the Firestore database to store the data 4. System create a new document record on the firestore database 5. System return a success message		
Alternate Flows	2a Display an error message if input is invalid 3a Display an error message if system fail to connect to database 4a Display an error message if system failed to create a new record in database		
Author	Eng Mun Shuen		

Table 3.3.10 Read and decode data from NFC tag use case

Use Case ID	UC010	Version	1.0
Use Case	Read and decode data from NFC tag		
Purpose	User could decode the raw message from NFC tag to readable string value		
Actor	Lecturer, Student, Admin		
Trigger	Approach mobile device to NFC tag		
Precondition	-		
Normal flow of events	1. User tap their phone on the NFC tag 2. System set up a connection with the NFC tag 3. System receive a raw message from the NFC tag 4. System convert raw message into String value		
Alternate Flows	2a Display an error message when NFC failed to connect to NFC tag 4a. Display an error message if raw message failed to convert into raw message		
Author	Eng Mun Shuen		

Table 3.3.11 Update class attendance use case

Use Case ID	UC011	Version	1.0
Use Case	Update class attendance		
Purpose	User could update user's class attendance		
Actor	student		
Trigger	Approach mobile device to NFC tag		
Precondition	Read and decode data from NFC tag		
Normal flow of events	1. System validate the value 2. System set up a connection with the Firestore database 3. System update the user class attendance status 4. System display success message		
Alternate Flows	1a. Display an error message if value is not in class attendance code format 2a Display an error message if system fail to connect to database 3a Display an error message if system failed to update a new record in database		
Author	Eng Mun Shuen		

Table 3.3.12 Update Crowd number in room use case

Use Case ID	UC012	Version	1.0
Use Case	Update Crowd number in room		
Purpose	User update the current number of people inside the registered room		
Actor	Student. Lecturer, Admin		
Trigger	Approach mobile device to NFC tag		
Precondition	Read and decode data from NFC tag		
Normal flow of events	1. System validate the value 2. System set up a connection with the Firestore database 3. System get the room current number of people 4. System update the current number of people inside the room 5. System display success message		
Alternate Flows	1a. Display an error message if value is not in class attendance code format 2a Display an error message if system fail to connect to database 3a Display an error message if system failed to update a new record in database		
Author	Eng Mun Shuen		

Table 3.3.13 View Profile use case

Use Case ID	UC013	Version	1.0
Use Case	View Profile		
Purpose	User could view their identity information		
Actor	Student / Lecturer		
Trigger	Click on view profile button		
Precondition	-		
Normal flow of events	1. User click on view profile 2. System connet to Firebase database 3. System get user's identity information based on android ID 4. System display the identity information		
Alternate Flows	2a Display an error message if system fail to connect to database		
Author	Eng Mun Shuen		

Table 3.3.14 View timetable use case

Use Case ID	UC014	Version	1.0
Use Case	View timetable		
Purpose	User view the classes they taken		
Actor	Student		
Trigger	Click on view timetable		
Precondition	-		
Normal flow of events	1. User click on view timetable 2. System connect to Firestore database 3. System get the classes taken by user into a list 3. System display the class taken by users		
Alternate Flows	2a Display an error message if system fail to connect to database 3a Display an error message if system failed to get the classes taken by user from database		
Author	Eng Mun Shuen		

Table 3.3.15 Check attendance history use case

Use Case ID	UC015	Version	1.0
Use Case	Check attendance history		
Purpose	User check their class attendance		
Actor	student		
Trigger	Click on check button		
Precondition	View timetable		
Normal flow of events	1. User click on class they wanted to view 2. System connect to the Firestore database 3. System get the all the attendance history of user 3. System display the attendance record into a list view		
Alternate Flows	2a Display an error message if system fail to connect to database 3a Display an error message if system failed to get the all the attendance history of user		
Author	Eng Mun Shuen		

Table 3.3.16 Check room crowd status use case

Use Case ID	UC016	Version	1.0
Use Case	Check room crowd status		
Purpose	User could view the current number of crowd inside a room		
Actor	Student, Lecturer, Admin		
Trigger	Click on check button		
Precondition	-		
Normal flow of events	1. User click on view room button 2. System connect to Firestore Database 3. System get a list of registered room with their information 4. System display the list of room into a listview 5. User select a room to view 6. System get the room ID that user want to view 7. System redirect user to new activity that shows room information 8. System display the room information		
Alternate Flows	2a Display an error message if system fail to connect to database 3a Display an error message if system failed to get a list of registered room with their information		
Author	Eng Mun Shuen		

Table 3.3.17 Create class attendance use case

Use Case ID	UC017	Version	1.0
Use Case	Create class attendance		
Purpose	User create an attendance record for the class on specific date and encode the attendance code into the NFC Tag		
Actor	Lecturer		
Trigger	Click on create button		
Precondition	-		
Normal flow of events	1. User select the class and attendance date. 2. System build an connection with the Firestore Database 3. System create a new class attendance record in the database 4. System display success message		
Alternate Flows	2a Display an error message if system fail to connect to database 3a Display an error message if system failed to create a new record in database		
Author	Eng Mun Shuen		

Table 3.3.18 Encode attendance code in NFC tag use case

Use Case ID	UC018	Version	1.0
Use Case	Encode attendance code in NFC tag		
Purpose	User encode unique class attendance code into NFC tag for student to take their attendance		
Actor	Lecturer		
Trigger	Click on Create button		
Precondition	Create class attendance		
Normal flow of events	1. User tap their phone on the NFC tag 2. System set up a connection with the NFC tag 3. System assign unique class attendance code by user into string value 4. System convert the string value into the NDEF message 5. System encode the NDEF message into NFC Tag		
Alternate Flows	2a Display an error message when NFC failed to connect to NFC tag 4a. Display an error message if string value failed to convert into raw NDEF message		
Author	Eng Mun Shuen		

Table 3.3.19 View class attendance use case

Use Case ID	UC019	Version	1.0
Use Case	View class attendance		
Purpose	User want to view all the student attendance within a class		
Actor	lecturer		
Trigger	Click on View button		
Precondition	-		
Normal flow of events	1. User select class ID and attendance date 2. User click on check button 3. System validate the input value from user 4. System redirect user to student attendance list view activity 5. System connect to Firestore Database 6. System get the student attendance record from database 7. System display all student attendance status and time		
Alternate Flows	2a. Display an error message if input is invalid 5a Display an error message if system fail to connect to database 6a Display an error message if system failed to get the student attendance record from database		
Author	Eng Mun Shuen		

Table 3.3.20 Update student attendance use case

Use Case ID	UC020	Version	1.0
Use Case	Update student attendance		
Purpose	User could update attendance of each student in the class		
Actor	Lecturer		
Trigger	Click on Update button		
Precondition	View class attendance		
Normal flow of events	1. System list the student attendance status to a recycle list view 2. User update student attendance status by clicking on the menu on selected student row 3. System connet to Firestore database 4. System update the attendance status		
Alternate Flows	3a Display an error message if system fail to connect to database 4a Display an error message if system failed to update the attendance status		
Author	Eng Mun Shuen		

3.4 Activity Diagram

Use case: Create new student/lecturer

Actor: Admin

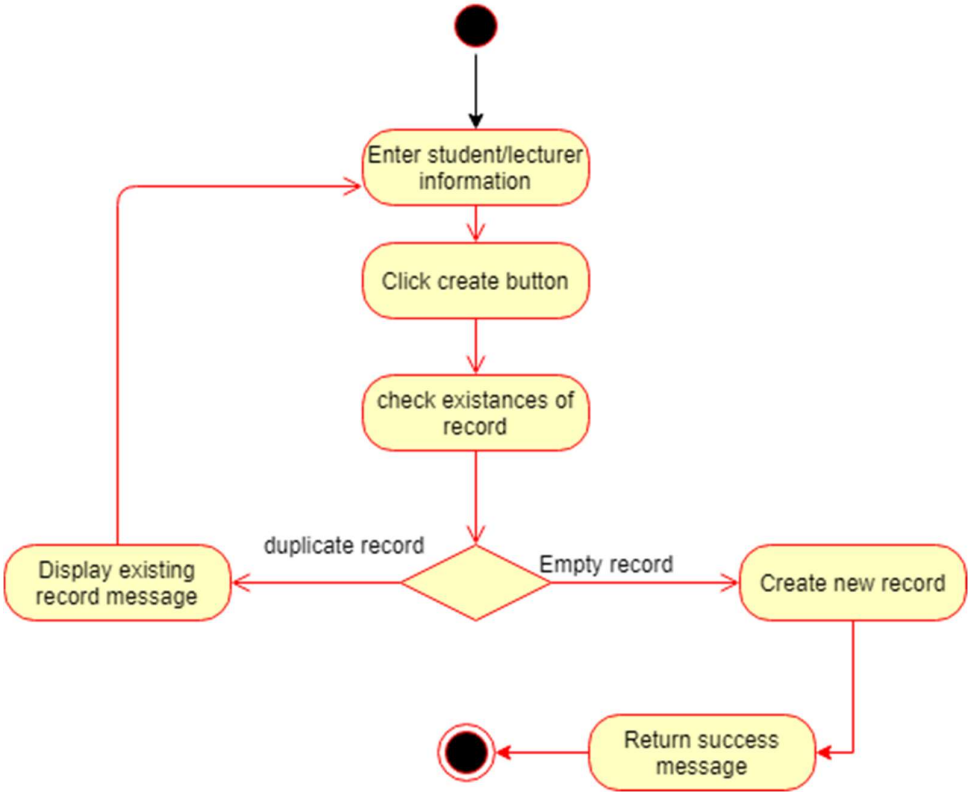


Figure 3.4.1 Create student/lecturer activity diagram

Description

Before student and lecturer could use this application, their identity must be existed into the database. This is because this application is only for the student and lecturer of the education institutions and do not open for outsiders. Therefore, admin must create each student and lecturer a record that consists of their information such as name, faculty and unique identity number in the database. Only student and lecturer that is verified and existed in this database could perform registration.

Admin will enter each user's personal information before creating a new record for the user. When admin click on "Create" button, system will check the existence of the record to prevent duplicate record. If there is a existing record, system will display a message informing admin the record is existed. If there is no record for the user, the user information will be created inside the database. Then, a success message will be responded to admin.

Use case: Register

Actor: Student / lecturer

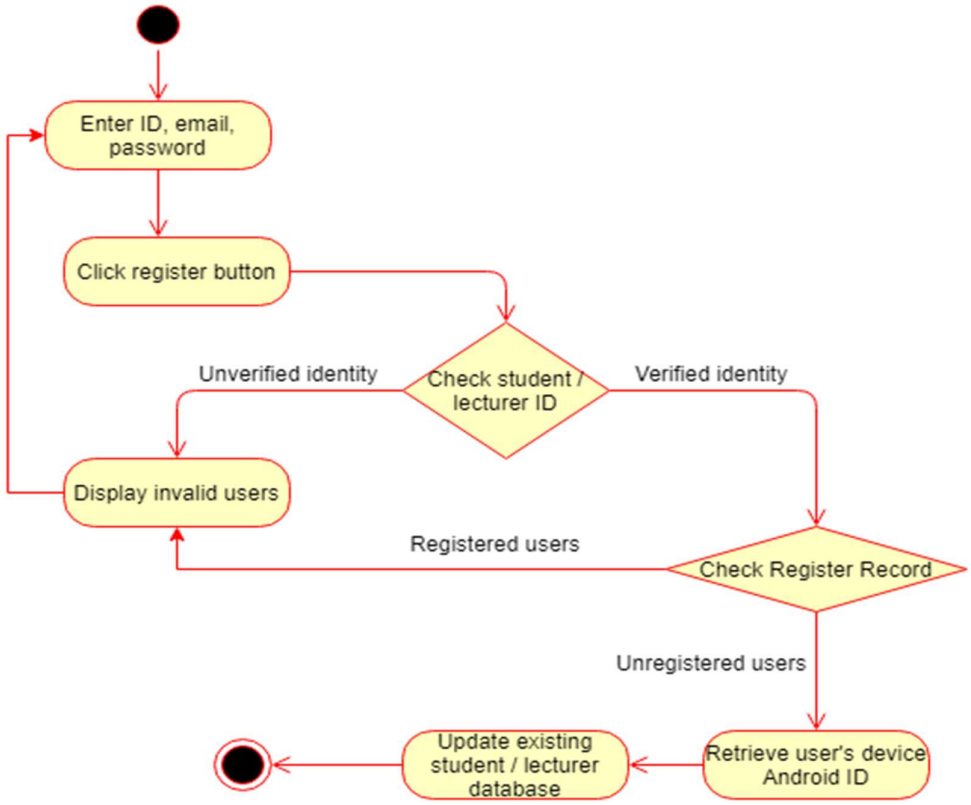


Figure 3.4.2 register activity diagram

Description

The first activity user encounter when user open this application will be the register activity page. This register activity will register user's unique device Android ID for security purpose. User could not log in into thus application without going through this step. User must enter their identity number that given by the education institution along with the email and a password to register themselves. After user click on the register button, the system will verify the identity number by comparing to the respective database. If the identity number is not founded in database, it will return error message to user and return back to previous activity. If user's identity number is verified, then system will check on the register history. If user had registered before, it would return the error message to user. Else, the system will retrieve user's device Android ID. Then, system will update the existing user's database that created by admin with the email and unique Android ID.

Use case: login

Actor: Student, Lecturer

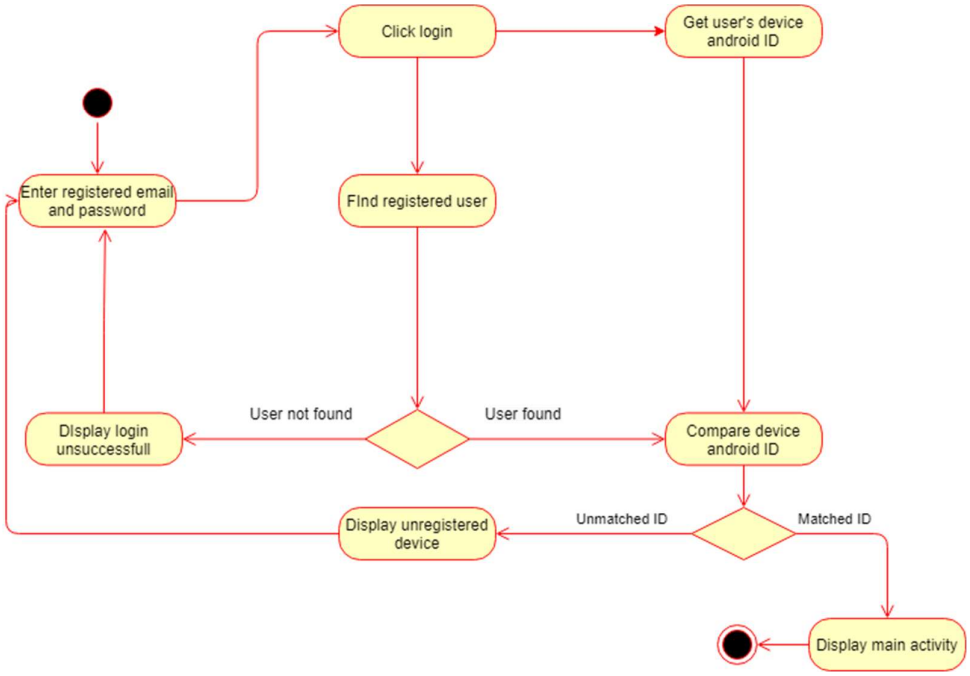


Figure 3.4.3 login activity diagram

Description

After student and lecturer had registered themselves, they could perform login function to enter the application. Before that, users are required to input the registered email and password for verification purpose. Then user will enter "Login" function. System will perform identity verification process to verify the users. If there is no registered record for the user, user will fail to login. If user had registered themselves, then it will verify the user's device's Android ID that was registered and recorded into the database. If the Android ID that retrieve from the current user's device is no matching with the recorded Android ID, user will fail to login into the application. On the other hand, if the device android ID is matched, login is successful.

Use case: Attendance taking and decode data from NFC tag

Actor: Student

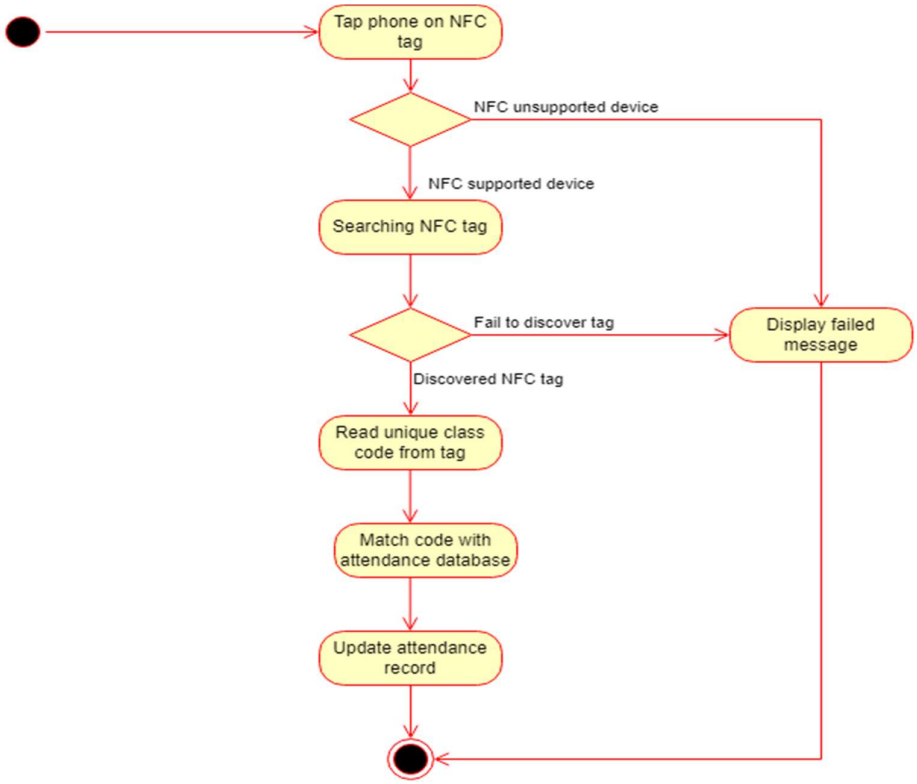


Figure 3.4.4 Attendance taking and decode data from NFC tag activity diagram

Description

When student want to take their attendance, firstly, they should tap their phone on the NFC tag. System will check on the existence of NFC adapter. If there is adapter for the NFC, the device is supported with NFC and could perform the next activity which is searching for the NFC tag. If the system failed to discover the tag, error message will be prompt out and the activity will be ended. Or else, the system will perform read function from the NFC tag if tag is successfully discovered. After retrieved the unique class code that is encoded by lecturer from the NFC tag, it will match with the class code from Firestore database. If the result is matched, it means that user has taken the correct class code and the attendance record will be updated on the database server

Use case: Create attendance and encode attendance code into NFC tag

Actor: Lecturer

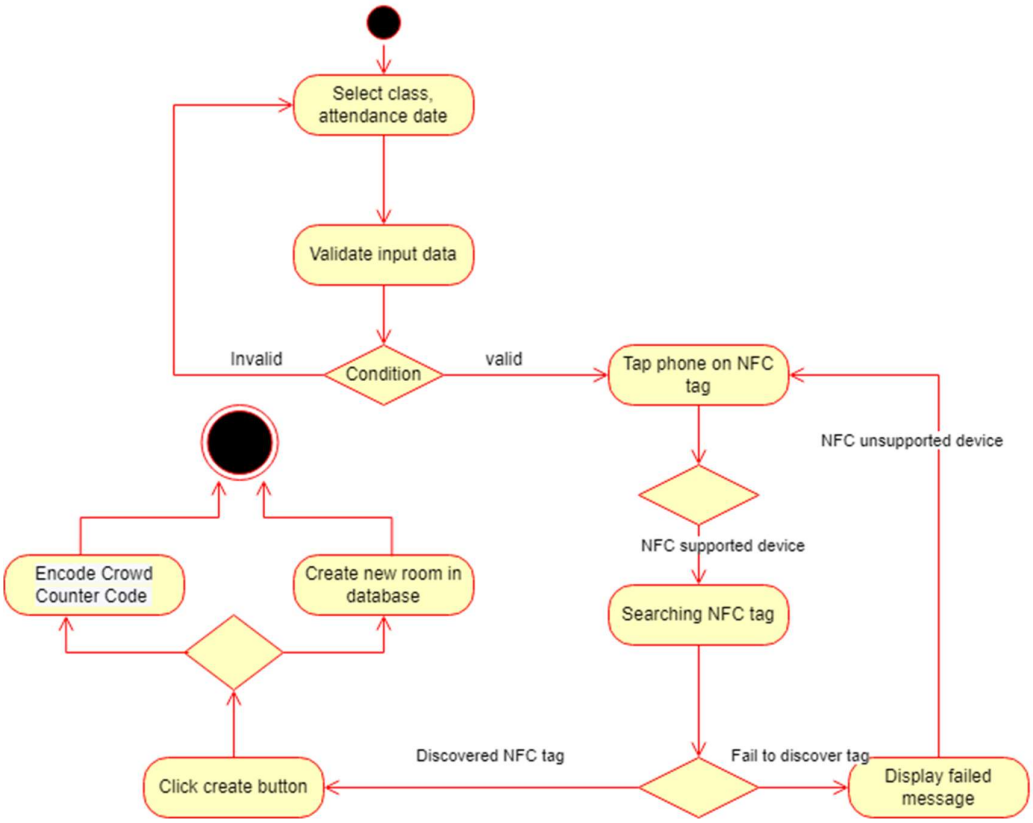


Figure 3.4.5 Create attendance and encode attendance code into NFC tag activity diagram

Description

Before student could read the unique class code inside the NFC tag and take their attendance, lecturer must encode the class attendance code into the NFC tag. Firstly, lecturer will select the class and attendance time. After the class and time are selected, the class code that represent the class will going to be written into the tag. After lecturer click on the “Write” button, the system will check the existence of NFC adapter in the device. If the adapter is existed, system will proceed to the next activity which is searching for the NFC tag. If the NFC tag is found, the class code will be encoded into the NFC tag in binary format. At the same time, system will create a new attendance record for each student that is taking this class. After the binary code is successfully encoded inside the NFC tag and attendance record successfully created in database, it will display the success message.

Use case: Update existing student/lecturer

Actor: Admin

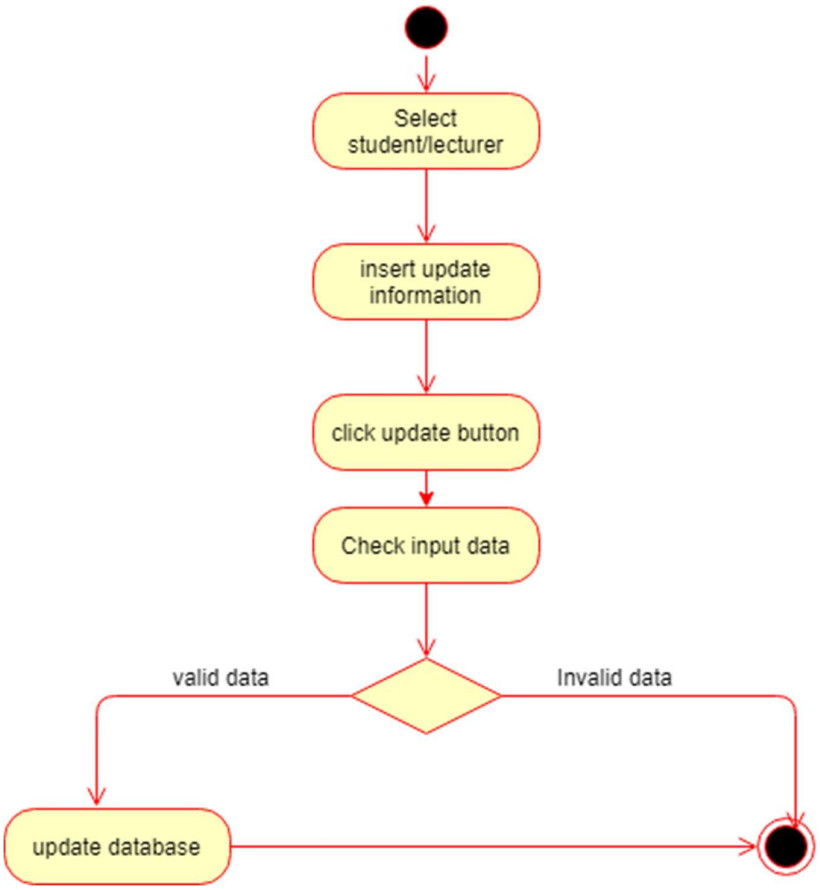


Figure 3.4.6 Update existing student/lecturer activity diagram

Description

Before admin update the student or lecturer information, the list of student/lecturer will be listed out. After admin select the targeted user, admin will insert the new data that wanted to be updated. Then, the inserted data will be validated after admin click on the "Update" button. If the format of the data is valid, the data will be sent to the database and update it

Use case: Create class

Actor: Admin

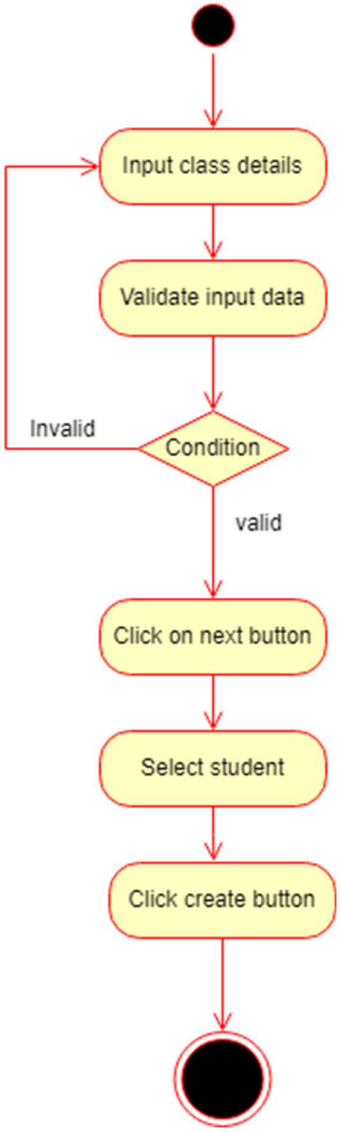


Figure 3.4.7 Create class activity diagram

Description

After the database contain lecturer, student, and subject record, admin could create new classes. Firstly, admin must input the lecturer, student and subject ID and click “Check button” to check the existence of these ID. If these ID are existed, then admin will input all the class details such as class day, start time, end time and location. After that, admin

will click on the “Create” button to validate the input. If all the inputs data are valid, a new class record will be created in the database.

Use case: Create room

Actor: Admin

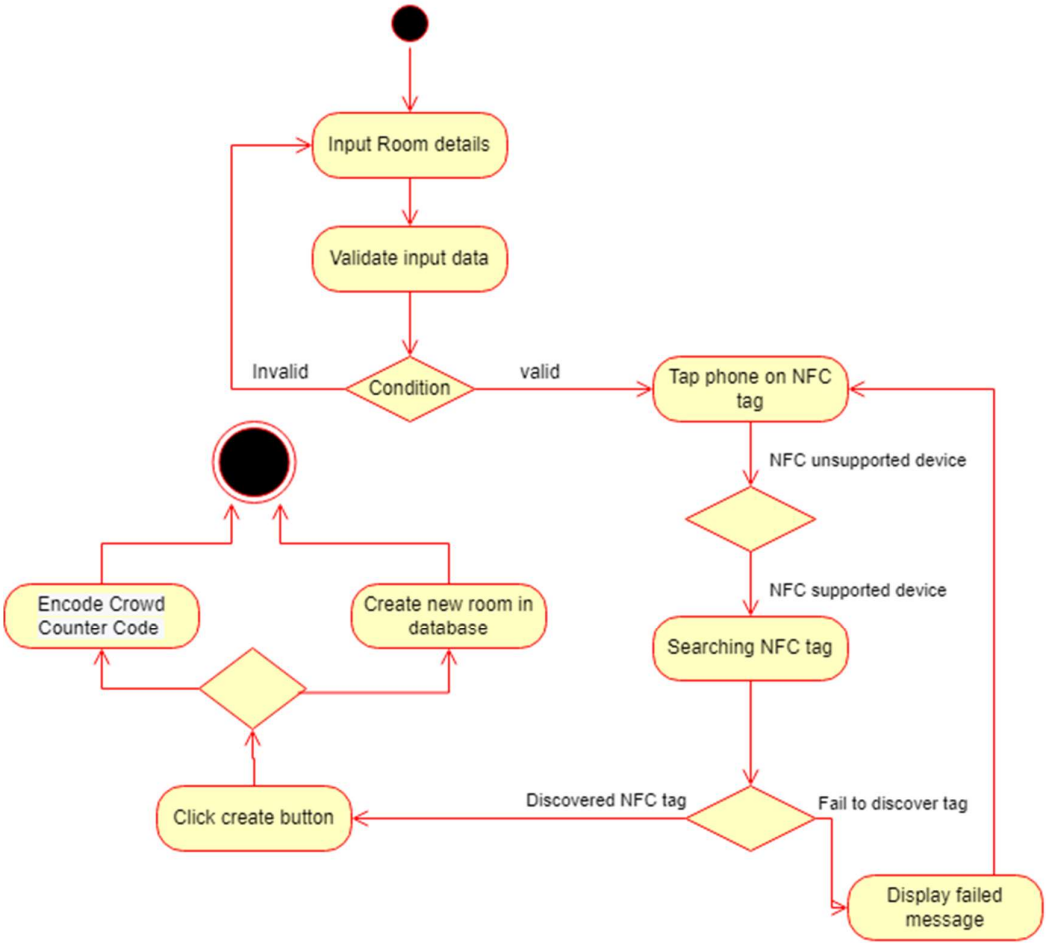


Figure 3.4.8 Create room activity diagram

Description

People counter function is one of the functions that could count and limit the number of people within a place. Before that, admin should write on a NFC tag that could count the number of incoming users. A unique counter code that combining “rcc” with the room ID number will be written on the NFC tag. At the same time, a new database record will be created using with the room name and maximum number of people.

Use case: Update class

Actor: Admin

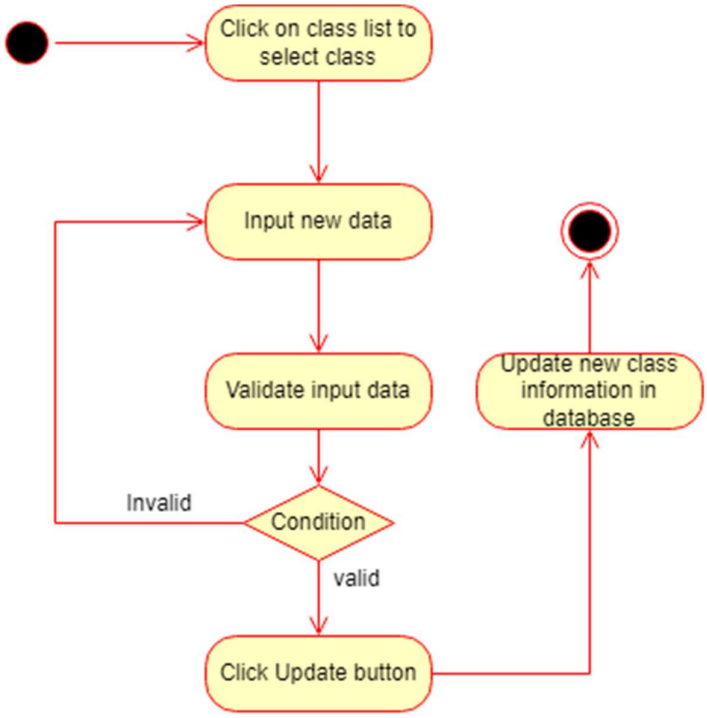


Figure 3.4.9 Update class

Description

Admin user will click on a list of class ID that are retrieved from firestore database. Then, system will list out all the saved data of the class. Admin could update each attribute of the class with the latest information. System will validate the data input by Admin at the same time. Then, system will perform an update method to update the updated class attributes to respective document in Firestore database.

Use case: Update crowd number in room and decode data from NFC tag

Actor: Admin, Lecturer, Student

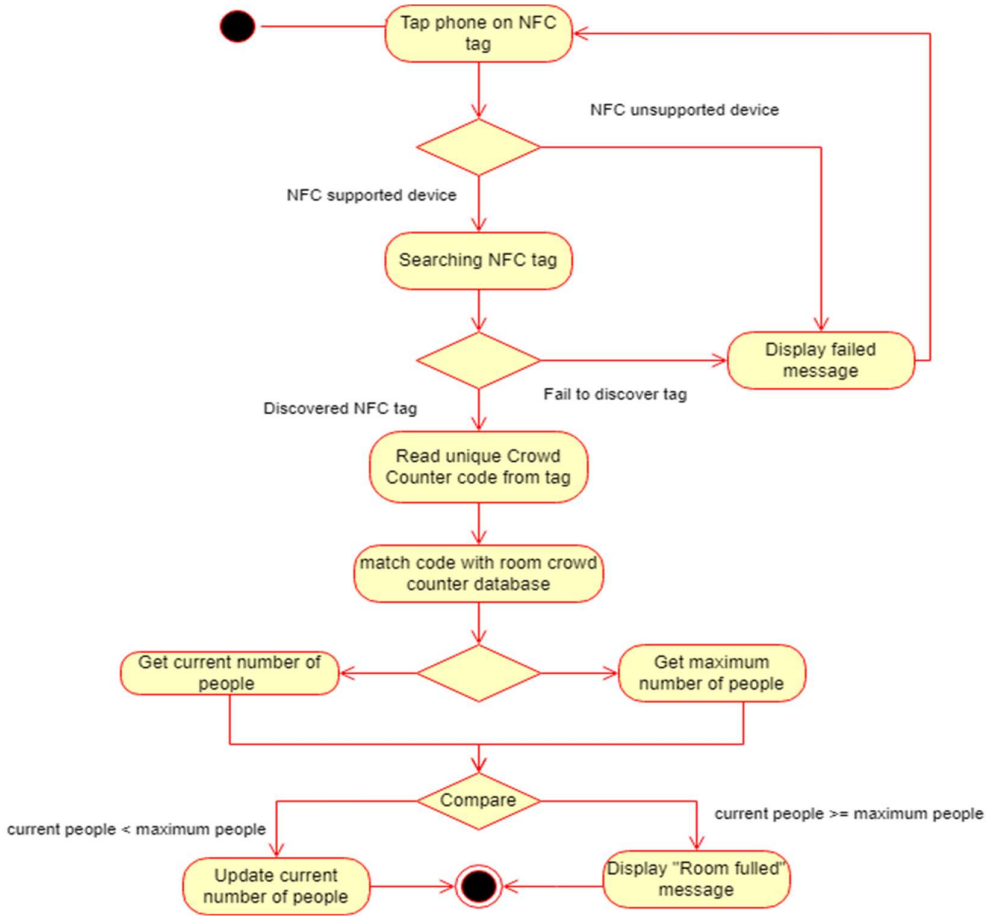


Figure 3.4.10 Update crowd number in room and decode data from NFC tag

Description

This function is provided to all of the role user. Firstly, user will tap their phone on the NFC tag. System will be searching for NFC tag actively and try to connect to the NFC tag. After the Tag is discovered and connected, system will retrieve the raw NDEF data from the NFC tag which encoded by Admin and match the data with the Room Crowd Counter unique code. The code is in format of : “rcc”+class ID. After system validated the unique code from NFC tag and it passed, then it will get the current status of the room. If the room status is full, then user could not check into the room. On the other hand, user could check in into the room if the room is not full. After that, an update of current number of crowds in the room will be saved on respective firebase database.

CHAPTER 4

System Design

4.1 System Block Diagram

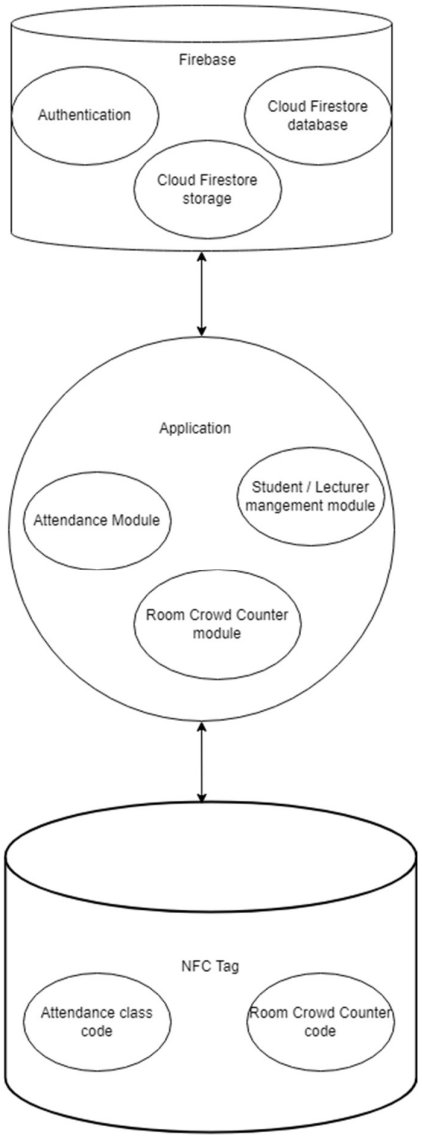


Figure 4.1 System Block Diagram

It shows the system block diagram of the system. It can be simplified into three section which are Firebase, Application and NFC tag. Firstly, user will access to the system through the UI of the application. Inside the application, there will be three main modules as the foundation of the application which are attendance module, student / lecturer management module and Room Crowd Counter module. The data source of

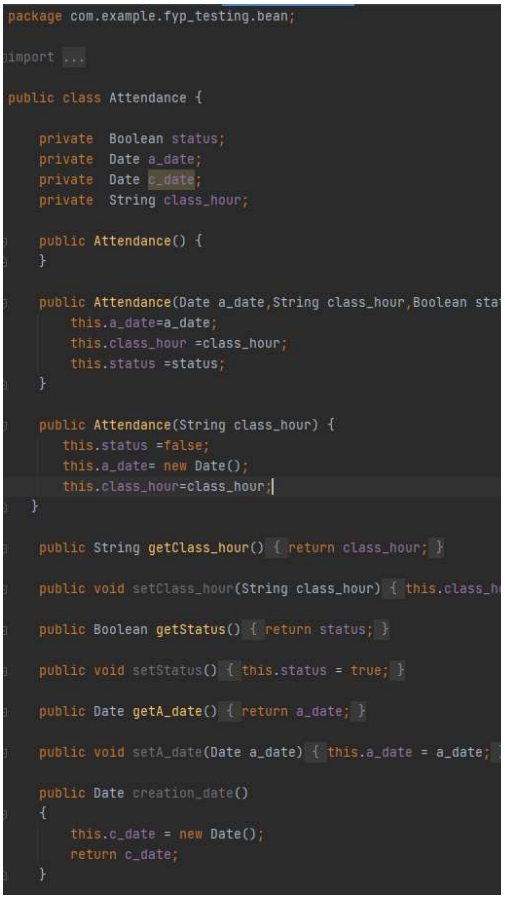
this system is Firebase database. System will using the Firebase service by using Firebase SDK. Firebase will provide Authentication services, Firestore database services and Cloud Firestore storage function. These service will work with the three module inside the user application and fulfil user's use case. On the other hand, a NFC tag will be also involve in this project as a data storage. It is use to store the attendance class code and Room Crowd Counter code. These two codes will be encode and decode by the system module.

4.2 System Components

Firestore Database

In this proposed system, Firestore database is used as data storing block. User could store their data to the Firestore database and all these data storing service is not in SQL format. Instead, it save user's data based on the document and key-value pairs. Therefore, it is easier to manage the database as it is more flexible compare to normal relational table. System will call the method set() method of store the data into the database. Firestore database could store the classes and object defined by users.

In this project, there are five classes and objects will be used. Each classes and object have their attributes and method. All these classes objects data will be saved on the firestore database.



```java
package com.example.fyp_testing.bean;

import ...

public class Attendance {

    private  Boolean status;
    private  Date a_date;
    private  Date c_date;
    private  String class_hour;

    public Attendance() {
    }

    public Attendance(Date a_date,String class_hour,Boolean sta
        this.a_date=a_date;
        this.class_hour =class_hour;
        this.status =status;
    }

    public Attendance(String class_hour) {
       this.status =false;
       this.a_date= new Date();
       this.class_hour=class_hour;
    }

    public String getClass_hour() { return class_hour; }

    public void setClass_hour(String class_hour) { this.class_h

    public Boolean getStatus() { return status; }

    public void setStatus() { this.status = true; }

    public Date getA_date() { return a_date; }

    public void setA_date(Date a_date) { this.a_date = a_date;

    public Date creation_date()
    {
        this.c_date = new Date();
        return c_date;
    }
```

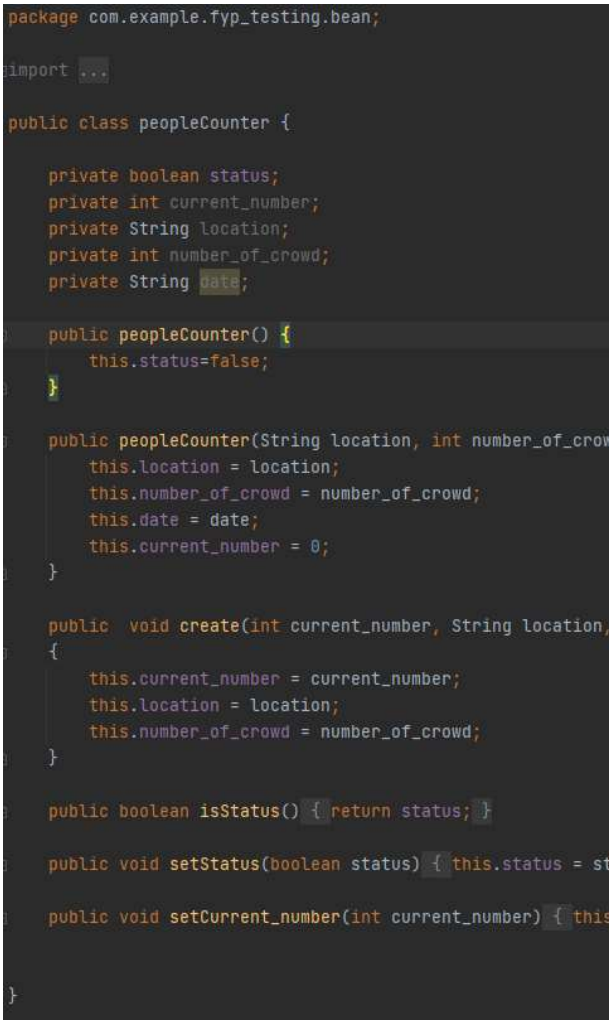
Figure 4.2.1 Attendance class

```java
package com.example.fyp_testing.bean;

import java.util.List;

public class student {
    private String stu_id;
    private String fname;
    private String lname;
    private String faculty;
    private String login_id;
    private String email;
    private String android_id;
    private List<String> classes;


    public student() { }

    public student(String stu_id, String fname, String lname, Stri
        this.stu_id = stu_id;
        this.fname = fname;
        this.lname = lname;
        this.faculty = faculty;
    }

    public String getStu_id() { return stu_id; }

    public void setStu_id(String stu_id) { this.stu_id = stu_id; }

    public String getFname() { return fname; }

    public void setFname(String fname) { this.fname = fname; }

    public String getLname() { return lname; }

    public void setLname(String lname) { this.lname = lname; }

    public String getFaculty() { return faculty; }

    public void setFaculty(String faculty) { this.faculty = facult

    public String getAndroid_id() { return android_id; }

    public void setAndroid_id(String android_id) { this.android_id

    public List<String> getClasses() { return classes; }

    public void setClasses(List<String> classes) { this.classes =
}
```

Figure 4.2.2 Student class

```
package com.example.fyp_testing.bean;

import java.util.List;

public class lecturer {
    private  String lec_id;
    private  String fname;
    private  String lname;
    private String faculty;
    private String android_id;
    private List<String> classes;


    public lecturer() {
    }

    public lecturer(String lec_id, String fname, S
        this.lec_id = lec_id;
        this.fname = fname;
        this.lname = lname;
        this.faculty = faculty;
    }

    public String getLec_id() { return lec_id; }

    public void setLec_id(String lec_id) { this.le

    public String getFname() { return fname; }

    public void setFname(String fname) { this.fnam

    public String getLname() { return lname; }

    public void setLname(String lname) { this.lnam

    public String getFaculty() { return faculty; }

    public void setFaculty(String faculty) { this.

    public String getAndroid_id() { return androi

    public void setAndroid_id(String android_id) {

    public List<String> getClasses() { return cla

    public void setClasses(List<String> classes) {
}
```

Figure 4.2.3 Lecturer class



```
package com.example.fyp_testing.bean;

import ...

public class peopleCounter {

    private boolean status;
    private int current_number;
    private String location;
    private int number_of_crowd;
    private String date;

    public peopleCounter() {
        this.status=false;
    }

    public peopleCounter(String location, int number_of_crow
        this.location = location;
        this.number_of_crowd = number_of_crowd;
        this.date = date;
        this.current_number = 0;
    }

    public  void create(int current_number, String location,
    {
        this.current_number = current_number;
        this.location = location;
        this.number_of_crowd = number_of_crowd;
    }

    public boolean isStatus() { return status; }

    public void setStatus(boolean status) { this.status = st

    public void setCurrent_number(int current_number) { this


}
```

Figure 4.2.4 Peoplecounter class (Room Crowd Counter)

```
package com.example.fyp_testing.bean;

import java.util.Date;
import java.util.List;

public class classes {
    String c_day,c_end,c_start,lecture
   List<String> student;
   List<String> att_date;

    public List<String> getAtt_date()
        return att_date;
    }

    public void setAtt_date(List<Strin
        this.att_date = att_date;
    }

    public classes() {
    }

    public String getC_id() {
        return c_id;
    }


    public classes(String c_id, String
String lecturer, String room, String s
        this.c_id = c_id;
        this.c_day = c_day;
        this.c_end = c_end;
        this.c_start = c_start;
        this.lecturer = lecturer;
        this.room = room;
        this.subject = subject;
        this.student = student;

    }

    public classes(String c_id, String
String lecturer, String room, String s
        this.c_id = c_id;
        this.c_day = c_day;
        this.c_end = c_end;
        this.c_start = c_start;
        this.lecturer = lecturer;
        this.room = room;
        this.subject = subject;
    }

    public String getC_day() {
        return c_day;
    }

    public void setC_day(String c_day)
        this.c_day = c_day;
    }
    public String getC_end() {
        return c_end;
    }
```

```
public void setC_end(String c_end) {
        this.c_end = c_end;
    }

    public String getC_start() {
        return c_start;
    }

    public void setC_start(String c_start) {
        this.c_start = c_start;
    }

    public String getLecturer() {
        return lecturer;
    }

    public void setLecturer(String lecturer)
        this.lecturer = lecturer;
    }

    public String getRoom() {
        return room;
    }

    public void setRoom(String room) {
        this.room = room;
    }

    public String getSubject() {
        return subject;
    }

    public void setSubject(String subject) {
        this.subject = subject;
    }

    public  List<String>  getStudent() {
        return student;
    }

    public void setStudent( List<String> stud
        this.student = student;
    }
}
```

Figure 4.2.5 Classes class

After user's data are stored into the Firestore database, system could call get() method to retrieve the document from the firestore database. The document retrieved from the Firestore database could be converted into class object that are readable by the system.

Whenever user wanted to update the data, system will call update() method to update the document inside the database.

Firebase Authentication

In this proposed system, user authentication function is supported by Firebase Authentication service. Admin user could manage the user authentication through the console of Firebase. Each users could register and log in into the system by using email and password. There are also different authentication options provided to users.

In this project, user will be register their account with the email and password. System will perform registration function by calling createuserWithEmailAndPassword() function.

```
fAuth.createUserWithEmailAndPassword(email,password).addOnCompleteListener(
new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if(task.isSuccessful()) { // Register succeed  }
        else { // Register failed }
    }
});
```

After user have registered their account, user can login into the application by using signInWithEmailAndpassword() method.

```
fAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if(task.isSuccessful()) { // Login succeed  }
        else { // Login failed  }
    }
});
```

At the same time, only student / lecturer already have a profile record on the Firestore database could perform this registration and login function. Which means that, only authorized student / lecturer by admin could register an account and login into the application for security purpose.

Attendance system component

This component is the main function of this application. This component is designed to manage the attendance activity for student, lecturer and admin. The use case for this component is create attendance, update attendance and take attendance. The class object involving in this component are Student class, Lecturer class, Attendance class and Classes class.

a) Create Attendance

One of the use cases for lecturer is attendance creation function. Lecturer could create the attendance for each class. When lecturer creating the class, a unique class code will be encoded into the NFC Tag. Lecturer must create the attendance by themselves before they start the class.

Figure below show the code that lecturer is selecting the class that he/she wanted to create an attendance record. Lecturer getting the class record from Firestore database and pass to a class object (class_obj). Then, it will add the selected date from lecturer into the class object using getAtt_date() method.

```
// Get the selected class by lecturer
db.collection("classes").document(global_selected_class).get().addOnComplet
eListener(new OnCompleteListener<DocumentSnapshot>() {
    @Override
    public void onComplete(@NonNull Task<DocumentSnapshot> task) {

        class_obj = task.getResult().toObject(classes.class);

        List<String> att_date= class_obj.getAtt_date();
        List<String> temp1 = new ArrayList<>();

        if(class_obj.getAtt_date()==null)
        {Toast.makeText(lec_create_attendance.this,"null" ,
Toast.LENGTH_SHORT).show();

            temp1.add(global_date);
            class_obj.setAtt_date(temp1);
        }
        else
        {
            att_date.add( global_date);
            class_obj.setAtt_date(att_date);
        }
```

After the class object is updated with the latest attendance date, then the system will add the new updated class object to the Firebase database.

```
db.collection("classes").document(global_selected_class).set(class_obj);
```

After that, the system will update each student that taken this class a new attendance record.

```
for (String temp : student_list) {

db.collection("attendance").document(temp).collection(global_selected_class
).document(global_date).set(att);
}
```

Then, lecturer is required to encode the unique class attendance code into the NFC Tag using write( ) method. In this method, the system will build a connection with the NFC Tag by using .connet() method. Then, it will call .writeNdefMessage() to write the message into the NFC tag. After that it will close the connection using .close() method.

```
private void  write(String text, Tag tag) throws
IOException,FormatException
{
    NdefRecord[] records={createRecord(text)};
    NdefMessage message = new NdefMessage(records);

    Ndef ndef = Ndef.get(tag);

    ndef.connect();
    ndef.writeNdefMessage(message);
    ndef.close();

}
```

The format of the attendance is class ID + date.

```
try{
    //if NFC is not detected
    if(myTag==null)
    {
        Toast.makeText(context,Error_Detected,Toast.LENGTH_LONG).show();
    }

    // Encode the attendance code into the NFC tag using write() method
    else
    {
        //The format of the attendance code is class ID + today's date
        write(global_selected_class+global_date,myTag);
        Toast.makeText(context,Write_Success,Toast.LENGTH_LONG).show();
    }

}
```

b) Update attendance

Lecture could also update each student attendance record. Lecturer will select the date and the class respectively to view the attendance record of the students. Firstly, lecturer get the selected class that lecturer wanted to view from firestore database.

```
db.collection("classes").document(selected_classes).get().addOnCompleteLis-
tener(new OnCompleteListener<DocumentSnapshot>()
```

Then, get all the student that taking the class into a array list (stu_list).

```
// Get all the student from the class
stu_list = (ArrayList<String>) task.getResult().get("student");
```

The next step is to loop through the student list and get the attendance record from each student from the attendance collection in Firestore database. After that, display each student attendance record into a recycle view.

```
// Get the attendance record from each student of the class
db.collection("attendance").document(stu_id).collection(selected_classes).d
ocument(selected_date).get()
```

On the student attendance recycler view, a menu will be implemented on it. Lecturer could click on the menu to update the student attendance.

```
public boolean onMenuItemClick(MenuItem item) {
    switch (item.getItemId())
    {
        case R.id.lec_update_attented:
            //Update student attendance to attended

db.collection("attendance").document(lec_view_stu_id.getText().toStri
ng()).collection(classid).document(date).update("status",true);
            lec_view_status.setText("Attended");
            return true;

        case R.id.lec_update_absent:
            //Update student attendance to absent
```

```
db.collection("attendance").document(lec_view_stu_id.getText().toStri
ng()).collection(classid).document(date).update("status",false);
            lec_view_status.setText("Absent");
            return                                                  true;

        default:return                                              false;
    }

}
```

c) Take attendance

Student could take the attendance by tapping their phone on the NFC tag.

Firstly, the system will read the raw NDEF message from the NFC tag using readfromIntent(Intent intent) method. Inside this method, the system will get the intent and call .getParcelableArrayExtra() method to get raw message. After the message is decoded and received by system, it will call buildTagView() method.

```
//Read raw NDEF Message data from the NFC tag
private void readfromIntent(Intent intend)
{
    String action=intend.getAction();

if(NfcAdapter.ACTION_TAG_DISCOVERED.equals(action)||NfcAdapter.ACTION_TECH_
DISCOVERED.equals(action)||NfcAdapter.ACTION_NDEF_DISCOVERED.equals(action)
)
    {
        Parcelable[] rawMsgs =
getIntent().getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES);
        NdefMessage[] msgs=null;
        if(rawMsgs != null)
        {
            msgs = new NdefMessage[rawMsgs.length];
            for(int i =0; i<rawMsgs.length;i++)
            {
                msgs[i]=(NdefMessage)rawMsgs[i];
            }
        }
        buildTagViews(msgs);
    }
}
```

In buildTagView() method, it will decide what to do with the message decoded from the NFC tag. Firstly, it will convert the raw NDEF message into a String of text(temp_attendance).

```
temp_attendance=new String(payload,languageCodeLength+1,payload.length-lan-
guageCodeLength-1,textEncoding);
```

Then, it will update the student attendance using collection of "Attendance" in Firestore database.

```
db.collection("attendance").document(task.getResult().get("id").toString())
.collection(global_tag_value)
        .document(global_date).update("status",true);
```

Crowd People Counter System component

This is a component that use to count the number of crowd people inside a class. The use case in this component is Create room and Update crowd people number.

a) Create Room

Admin could create a new room record that count the number of crowd people. Firstly, system will retrieve the room ID, maximum crowd people number value and the date and pass them to peopleCounter object.

```
// Create a new peopleCounter object that store all the information of the
room
people_counter1 = new
peopleCounter("room"+location_input1.getText().toString(),
limit_value,strDate);
```

Then create a new record on the Firestore database.

```
// Update the record on the Firestore database
Map<String, Object> update = new HashMap<>();
update.put(strDate,people_counter1);
DocumentReference docRef =
db.collection("area_crowd_check").document("room"+location_input1.getText()
.toString());
docRef.set(update, SetOptions.merge());
```

After the record is created on Firestore database, encode a unique Crowd People Counter code into NFC tag. Firstly, building a connection between the system with the NFC tag. Then call write() method to encode the unique code into the NFC tag. The format of the Room Crowd Counter code is “rcc” + room number.

```
//Format of Room Crowd Counter = rcc + room number
String code = "rcc"+location_input1.getText().toString();
write(code,myTag);
Toast.makeText(context,Write_Success,Toast.LENGTH_LONG).show();
```

b) Update crowd people number

System will update the current crowd people number inside the room to the Firestore database by decoding the Room Crowd Counter code from the NFC tag. The method to convert raw data taken from NFC tag to a readable string value in the system is similar with the take attendance function from Attendance component. The different is

when the code is in Room Crowd Counter format, system will update the current crowd people number inside the room.

Firstly, system will get the record from Firestore database based on the room number.

```
db.collection("area_crowd_check").document(room_number).get()…….
```

Then, system will check whether the room reach maximum number of people or not. If the room is not full yet, it will update the room record database.

4.3 System Flowchart

Admin flow chart

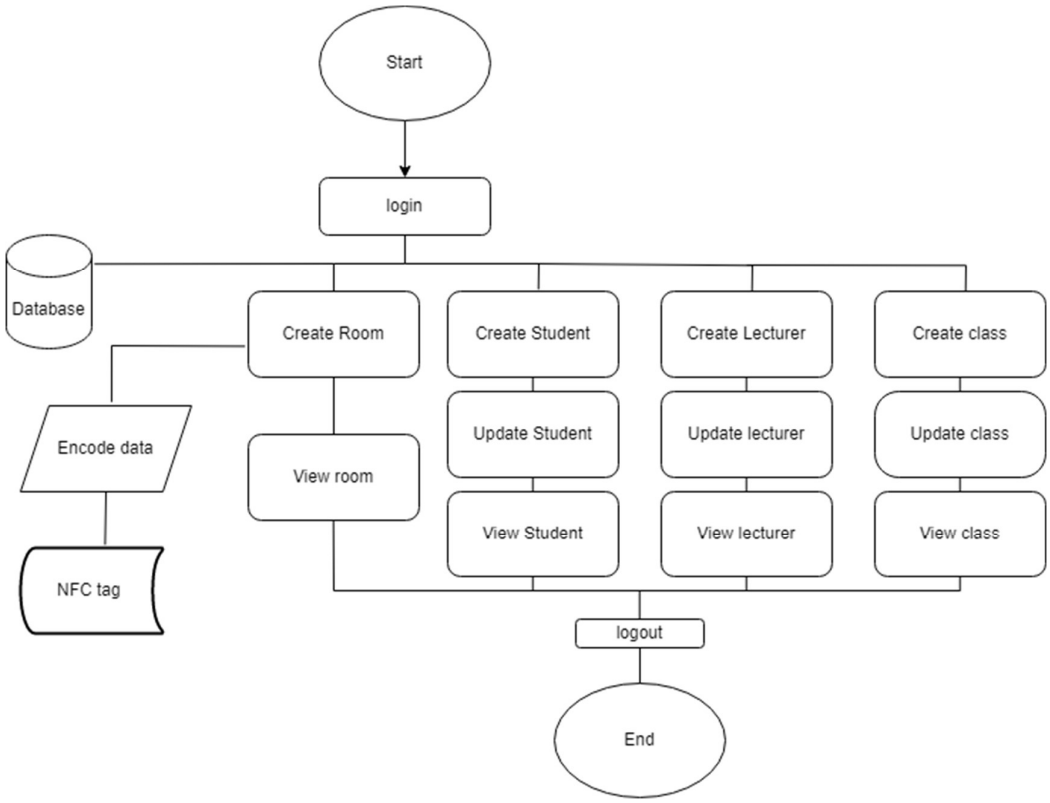


Figure 4.3.1 Admin flow chart

This is a flow chart for admin role as it show the overall flow of functionalities provided by system. Firstly, admin will login into the system using respective id and password.

Once admin login to the main page, admin could perform 11 different activities to have different functionalities. Admin could create a room, view room, create student, lecture and class, update student, lecturer and class, and view student lecturer and classes. Admin could click on different button on the admin main page that represent each functionality. Other than that, admin could access the database through these functionalities and perform create, update and view function. When admin wanted to create a new room, admin must also encode the unique Room Crowd Counter code to the NFC tag.

After admin are done using the application, they could logout the application by clicking on the logout button.

Lecturer Flowchart

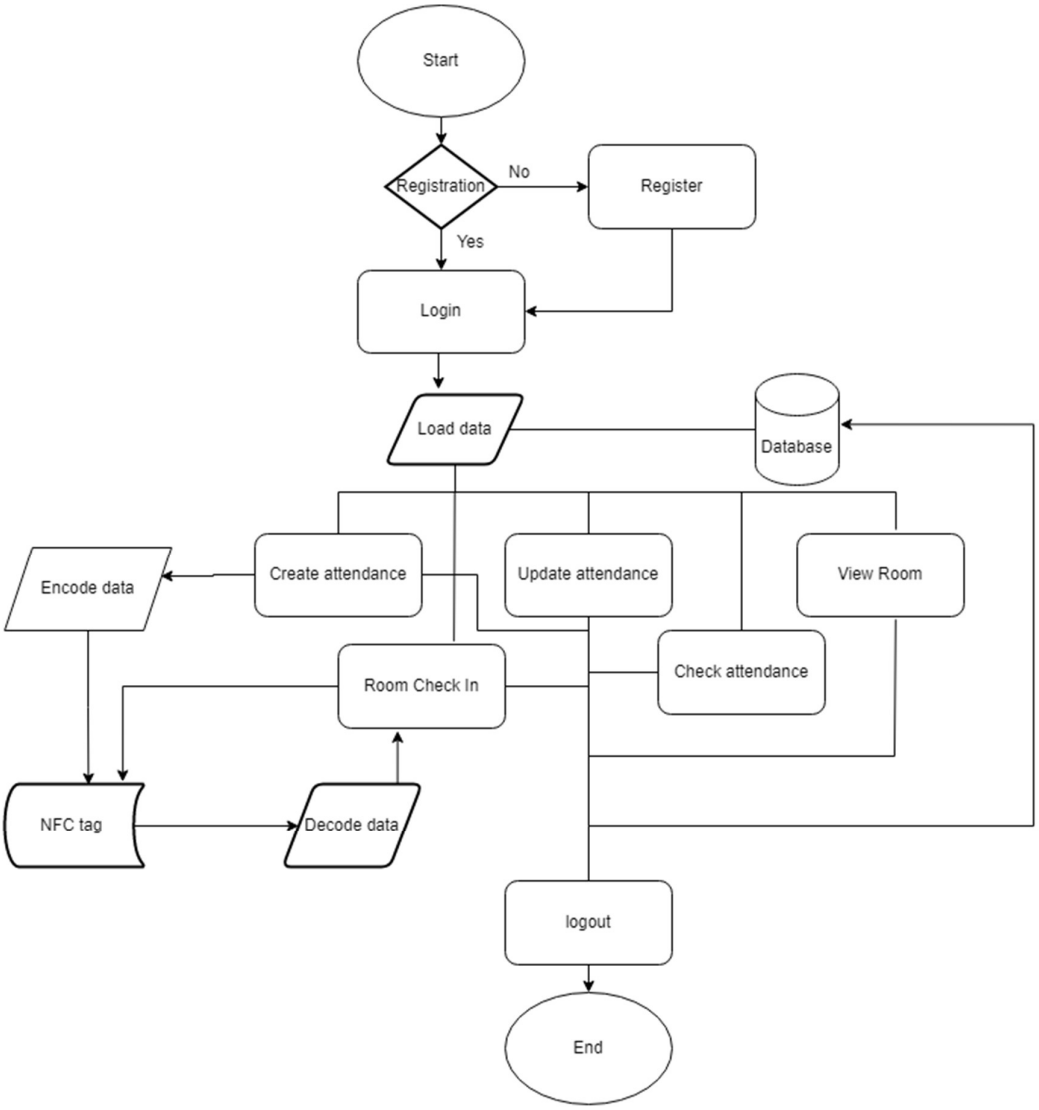


Figure 4.3.2 Lecturer flowchart

This is the flow chart for lecturer. Firstly, first time user will register an account before login into the system. After lecturer had registered, they will be login into the system. Once lecturer login successfully, they will access to the main page of the application. Lecturer are given the functionalities to create new attendance for the class, check in into room, update student attendance, check student attendance and view room. All these functionalities are connected to the database. Besides, lecturer also need to perform encoding and decoding data into NFC tag. Once they are done using the application and wanted to log out, they could log out by click on log out button.

Student Flow Chart

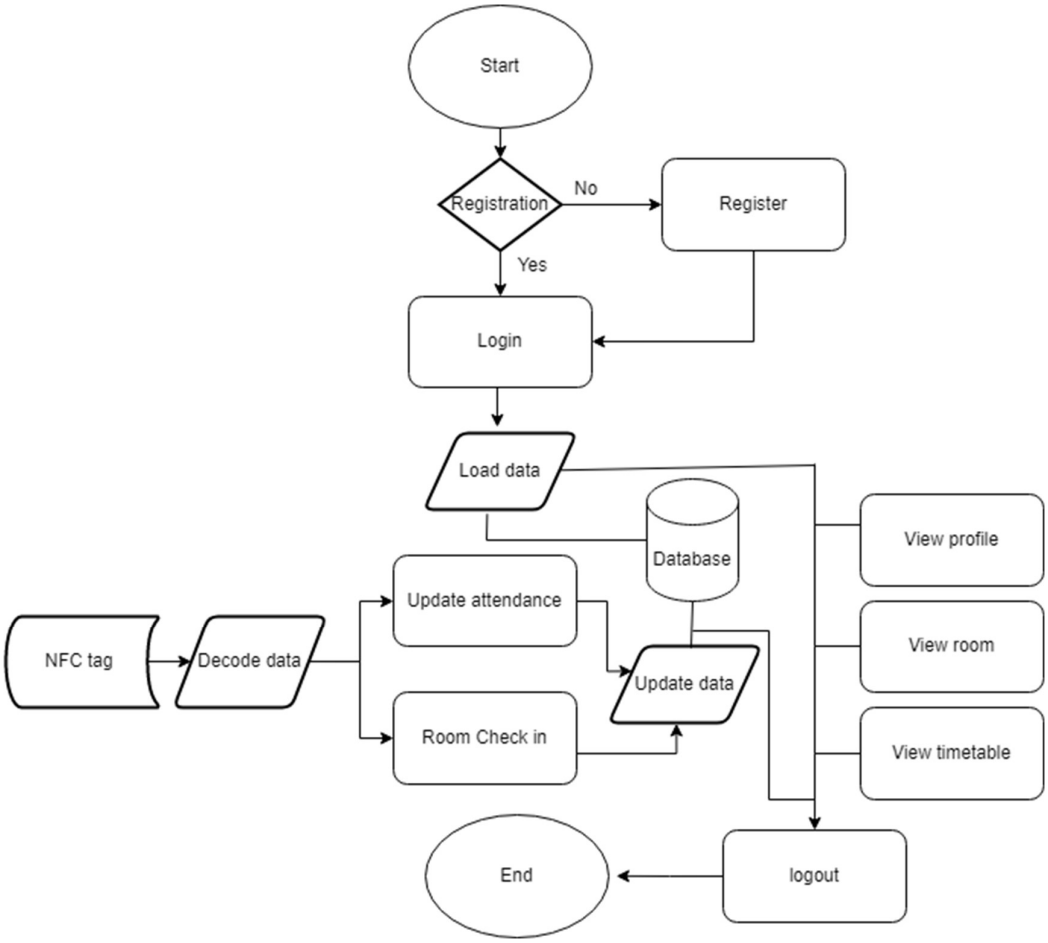


Figure 4.3.3 Student flowchart

This is the flow chart of student where it shows the flow of student using the application. Firstly, student who is the first time using this application must register themselves. After they successfully registered, they could login into the system. Once they log in, they will redirect to the main page. On the main page, student is provided with different button which represent different functionalities provided for student. Student could take their attendance, check in into a room, view their profile, view the room and their timetable. When student is taking class attendance and checking in into a room, student require to decode the data from the NFC tag by tapping their phone on the NFC tag. All the functionalities is connected to the Firestore database which provide the function storing the data. Once student is done with the application and want to log out, they could log out by clicking on the log out button on the main page.

Chapter 5

System Implementation

5.1 System Operation

Function: Create new student

Actor: Admin

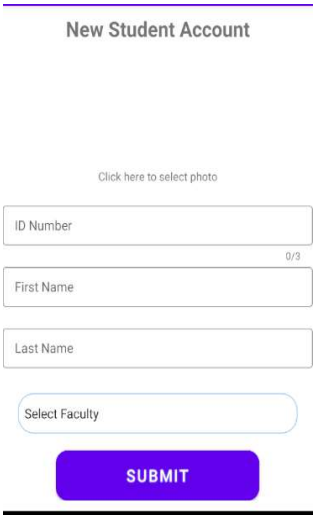


Figure 5.1.1 Student information input field

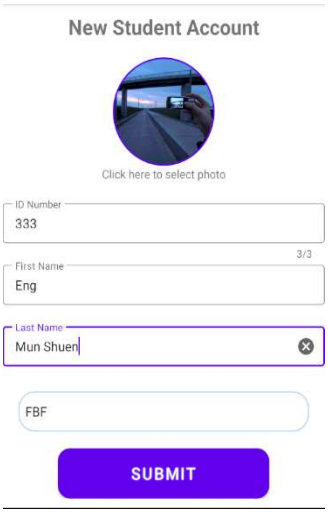


Figure 5.1.2 Complete student information input field

Function: Create new lecturer

Actor : Admin

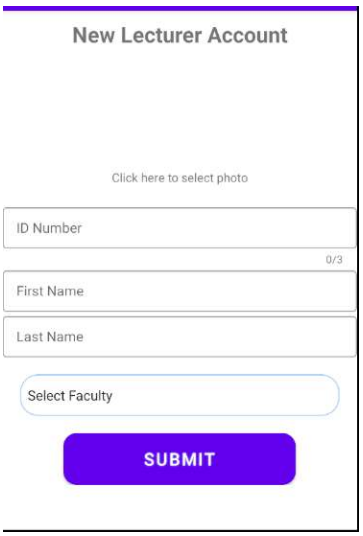


Figure 5.1.3 Lecturer information input field

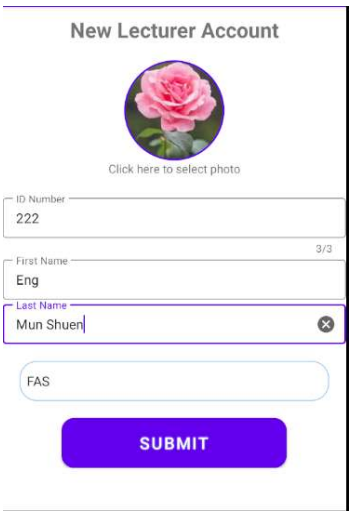


Figure 5.1.4 Complete lecturer information input field

Before student could perform registration and login function, admin must create each student and lecturer profile. Admin could enter the user ID number, first name, last name and they respective faculty. Besides, admin also could upload their profile picture. After student / lecturer account is created, admin could perform classes creation function.

Function: Update student

Actor: Admin

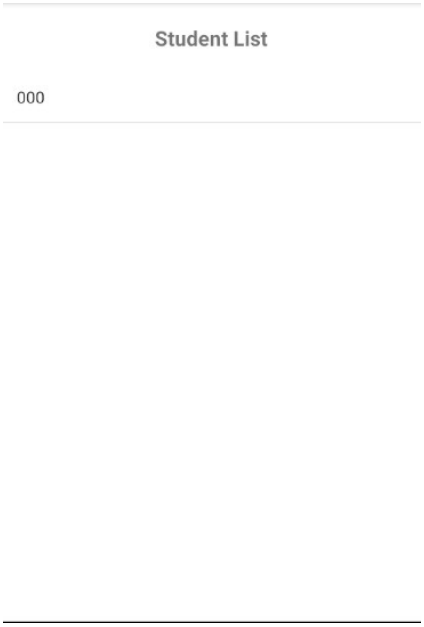


Figure 5.1.5 Student list

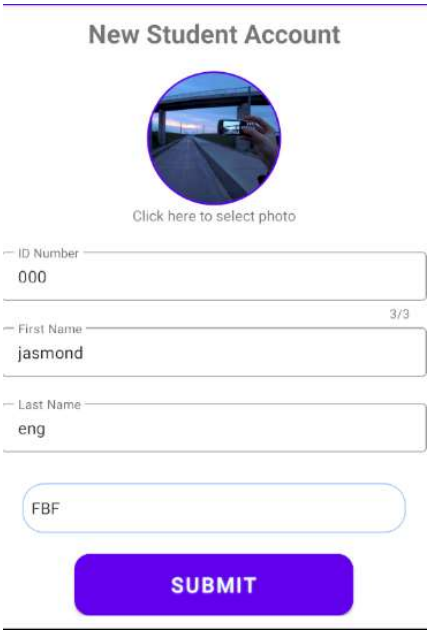


Figure 5.1.6 Update student information input field

Function: Update lecturer

Actor: Admin

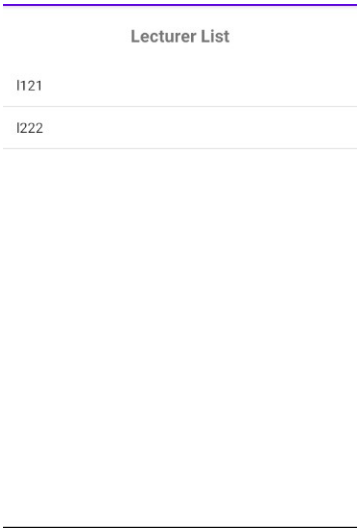


Figure 5.1.7 Lecturer list

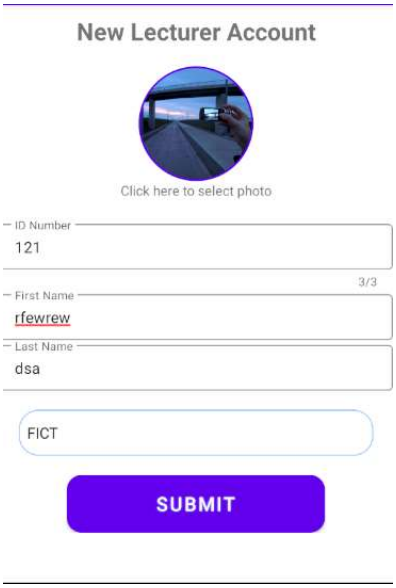


Figure 5.1.8 Update lecturer information input field

When admin wanted to update each student and lecturer identity information, firstly admin have to select one of the student / lecturer from the student / lecturer list. After that, system will redirect user to update page where all the student / lecturer identity information will be showed here. Admin could change ID number, first name, last name, faculty and reupload their profile picture. When admin is done, click on submit button and it will update the previous record of student / lecturer document in Firestore database.

Function: View student

Actor : Admin

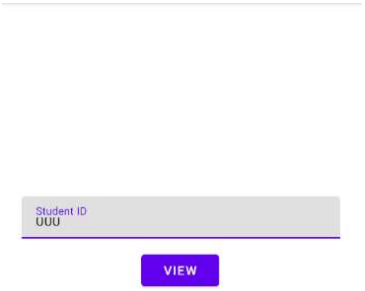


Figure 5.1.9 Student ID input

Figure 5.1.10 Student profile

Function: View lecturer

Actor : Admin

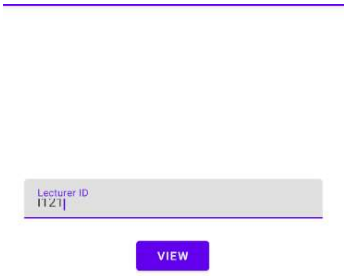


Figure 5.1.11 Lecturer ID input

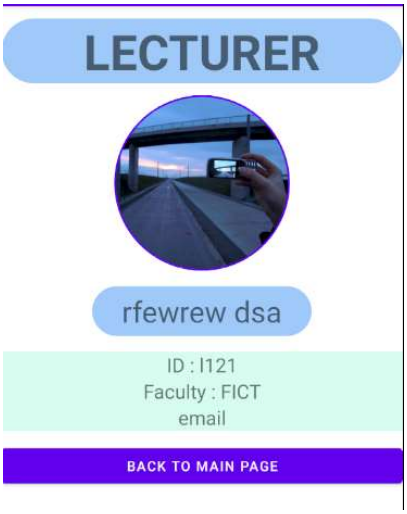


Figure 5.1.12 Lecturer Profile

Admin could view any student details by entering the student ID number.

Function: Create new class

Actor : Admin

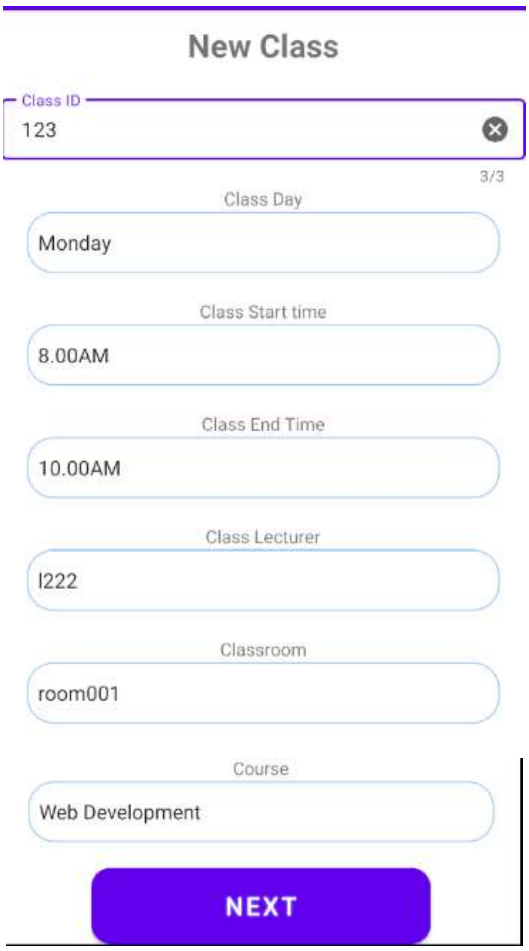


Figure 5.1.13 New class input field

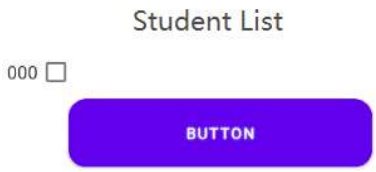


Figure 5.1.14 New class student list field

After admin had created student account and lecturer account, then admin could start to proceed to create new class. Firstly, admin is required to enter correct class details such as ID number, class day, class start time, class end time, class lecturer, class room and the course. After the system validate all the input values of the class, then admin click on next button. System will redirect admin to student list page where it require admin to add the student that participate in this class. Then click on finish button and the new class creation process will be completed.

Function: Update class

Actor : Admin

Figure 5.1.15 Class list

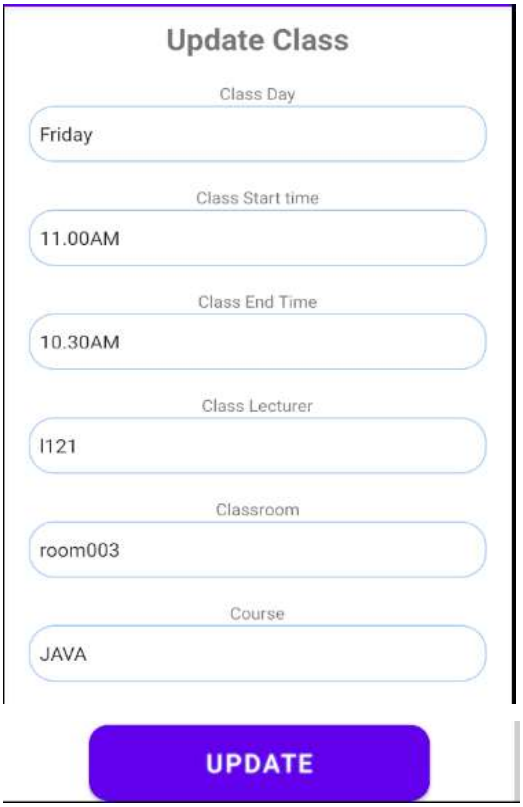


Figure 5.1.16 Update class input field

When amin wanted to update the class, when admin click on class update button, it will redirect admin to a list of class ID where admin need to select class that wanted to be updated. After selecting the class, then admin will be redirected to update class page where admin could update the details of the class. When admin complete filling all the class details, click on update button and the update process will be completed.

Function: View Class

Actor : Admin

Figure 5.1.17 Class ID input field

Figure 5.1.18 Class destribtion

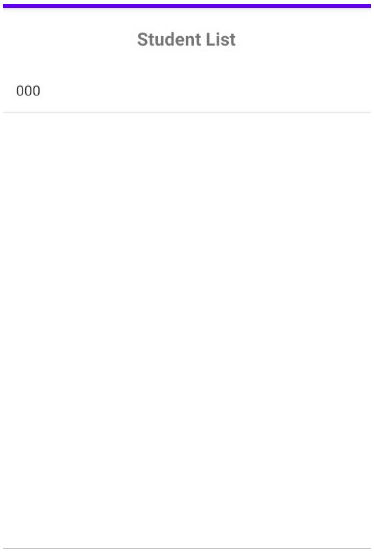


Figure 5.1.19 Class Student List

Figure 5.1.20 Student Profile

After admin created the class room, then admin could view each class room details. Admin could view the class details by entering the class ID, then it will display the class details. After that, admin could view the student list by clicking on view student (Figure

5.1.18). System will redirect user to student list page where it will list out all the student who are taking the class. Admin could also view each student profile by clicking on the student ID .

Function: Create Room

Actor : Admin

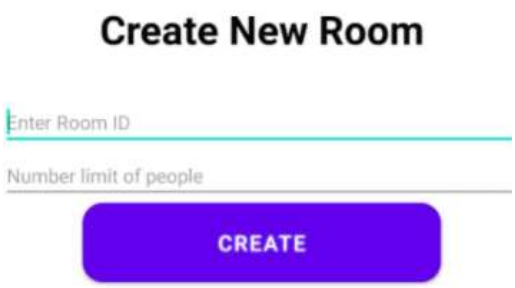


Figure 5.1.21 Room Details input field

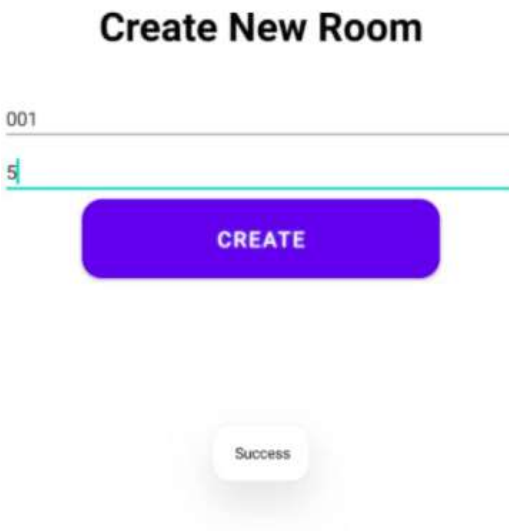


Figure 5.1.22 Success Room Details input

First of all, admin will create a new object of people counter record and store it into the database. Admin have to enter room ID number and maximum number of people. Then,

a random generated code with “rcc” string in front of the class ID will be encoded inside the NFC tag and uploaded into the database at the same time.

When user tap their phone on the NFC tag, the code that encoded inside the tag will be retrieved. When the first 3 character of the code is “rcc”, it means that it is the correct format of code for room crowd counter function. Then, the code will be used as keyword to search for the corresponding record in the firestore database. After that, the system will check on the status, if the status remain default which is false, then user could enter to the room. The current number will be updated by adding 1. If the current number meet the number of people, then system will update the status into true. Therefore, any other user could not come into the room again since the room is full

Function: View Room list

Actor : Admin , student, lecturer

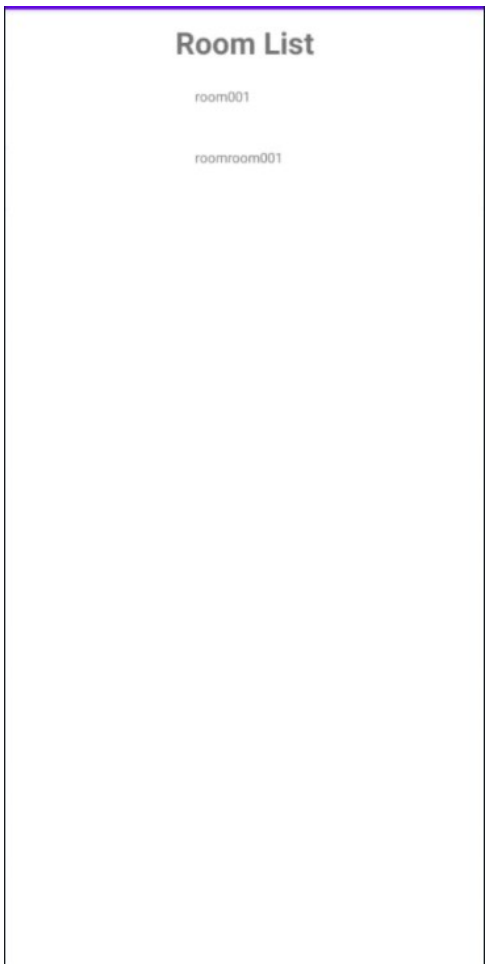


Figure 5.1.23 Room list

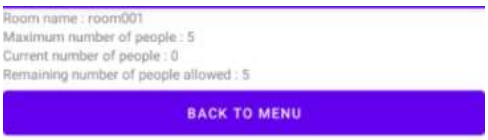


Figure 5.1.24 Room status

All the user of the application could view the room current number of crowd. They could click on the room list and view the current status of the room.

Function: Student / Lecturer View Profile

Actor : Student / lecturer

Figure 5.1.25 Lecturer Profile

Figure 5.1.26 Student Profile

After admin had created the record for the student /lecturer, they could view their profile after they registered and login into the application. They could view their profile picture, name, ID and faculty.

Function: Create class attendance

Actor : Lecturer

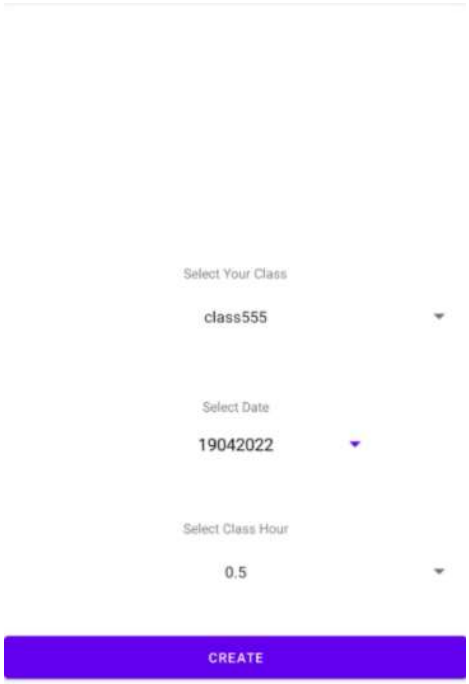


Figure 5.1.27 Class attendance creation default input field

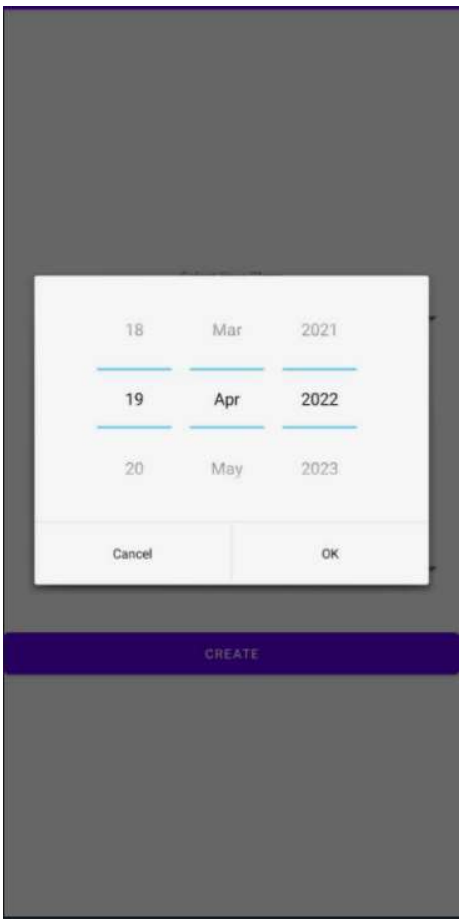


Figure 5.1.28 Attendance date selection

Lecturer could create a new attendance record for student. Firstly, lecturer will click on create attendance button and system will redirect user to attendance creation page(Figure 5.1.27). After that, lecturer will select a list of class ID which represent lecturer's classes. Then lecturer could start to pick a attendance date (Figure 5.1.28).

Figure 5.1.29 Attendance class hour selection

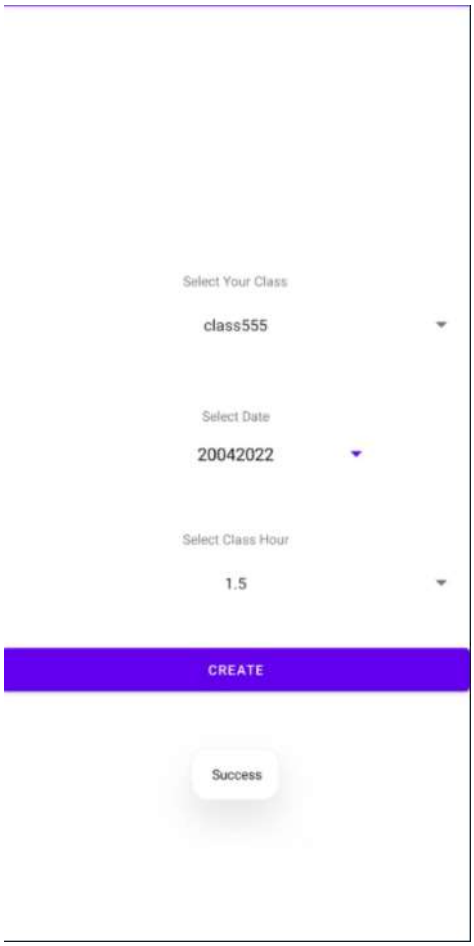


Figure 5.1.30 Success attendance class creation

After selecting the date, lecturer will pick the class hour for the class. Then, lecturer are required to approach their phone on NFC tag to encode the unique class attendance code into the NFC tag. When lecturer click on Create button, system will start to encode the attendance class code into the NFC tag and start to create a new attendance record for each student who taking this class. A success message will be display if the class successfully created and attendance code successfully encode into NFC tag.

Function: View class attendance / Update class attendance

Actor : lecturer

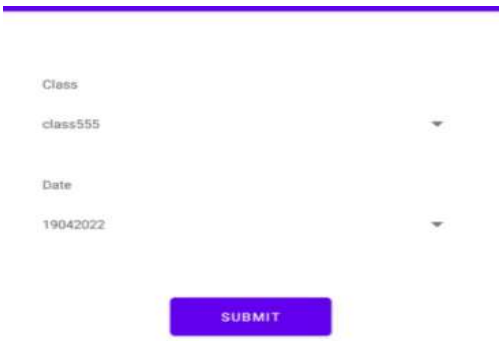


Figure 5.1.31 Lecturer view class attendance main page

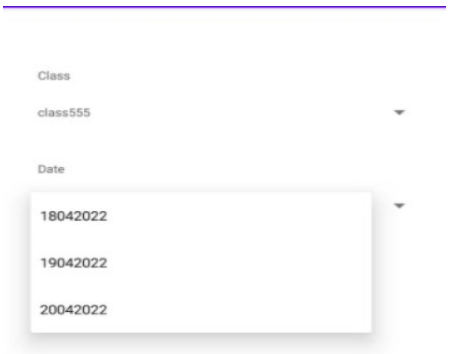


Figure 5.1.32 Lecturer view attendance date selection

Figure 5.1.33 Student class attendance status list

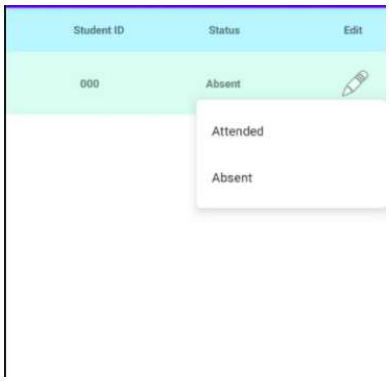


Figure 5.1.34 Attendance update menu

After lecturer created the class, lecturer could view all their student attendance based on the class. In figure 1, it showed that lecturer could select which class and date of the attendance to view. After clicking on submit button, system will redirect lecturer to the

list of student attendance status (Figure 5.1.32). It shows each student ID and the attendance status. Lecturer could also update the student by clicking on the edit image. After that, it will display a menu of attendance status (Figure 5.1.34) where lecturer could select an option to update the student attendance status.

Function: Check attendance history

Actor ; Student

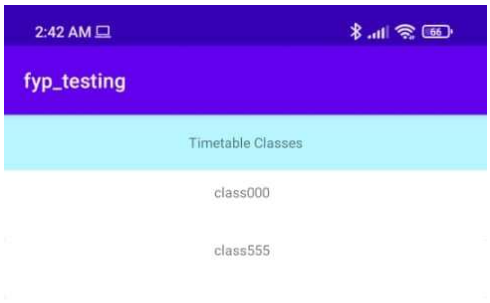


Figure 5.1.35 Student class attendance list

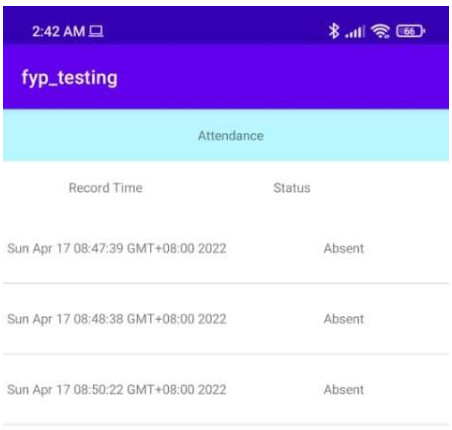


Figure 5.1.36 Student class attendance status

After lecturer had created the class attendance, student can always check their class attendance through the timetable classes. Firstly, student could view their class through timetable button. Then it will redirect user to a page where it list out the class taken by student. After that, when student click on the class ID, student will be able to view the class attendance (Figure 5.1.36). It will display the attendance record time and the status of the attendance.

Function: Take Attendance

Actor : Student

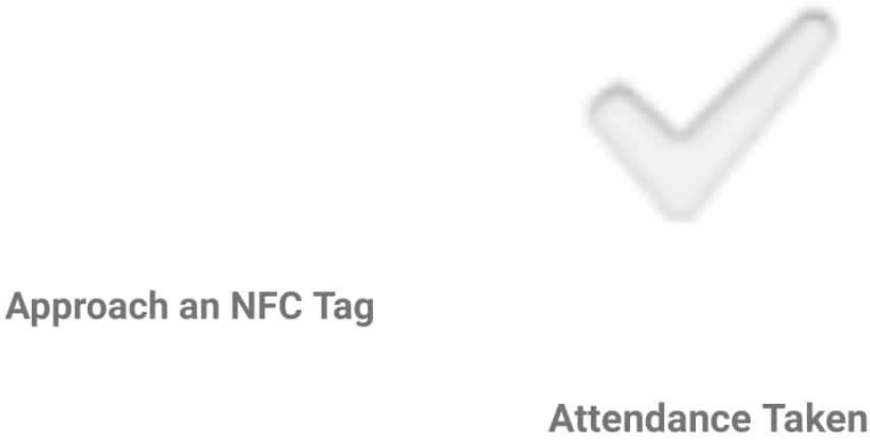


Figure 5.1.37 Attendance taking main page

Figure 5.1.38 Success attendance success page

When student wanted to take attendance, student could click on take attendance button. Then student will approach their mobile phone to the NFC tag. Then system will connect to the NFC tag and decode the NDEF message from the tag. If the raw message from the tag is in class attendance code format, then it will update student attendance status on the Firestore database, after that system will redirect user to attendance taken success page.

CHAPTER 6

System Evaluation and Discussion

6.1 Testing Setup and Result

Table 6.1.1 Register account

Test case ID: TC001				
Test case: Register account				
Test case Description: Test account registration flow in multiple scenarios				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Register with valid information	A message “Successful registered” is shown	A message “Successful registered” is shown	Pass
2	Register with an empty field	A message “The field is required” is shown	A message “The field is required” is shown	Pass
3	Register with existing email	A message “The email is already in use by another account” is shown	A message “The email is already in use by another account” is shown	Pass
4	Register with different passwords for the “Password” and “Repeat Password” field	A message “Password does not match” is shown	A message “Password does not match” is shown	Pass

Table 6.1.2 Login

Test case ID: TC002				
Test case: Login				
Test case Description: Test case login flow in multiple scenarios				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Login with a valid email and password	A message “Login successfully” is shown	A message “Login successfully” is shown	Pass
2	Login with an invalid email and valid password	A message “There is no user record corresponding to this email” is shown	A message “There is no user record corresponding to this email” is shown	Pass
3	Login with a valid email and invalid password	A message “The password is invalid” is shown	A message “The password is invalid” is shown	Pass
4	Login with an empty email	A message “Email is required” is shown	A message “Email is required” is shown	Pass
5	Login with an empty password	A message “Password is required” is shown	A message “Password is required” is shown	Pass

Table 6.1.3 Create student/ lecturer

Test case ID: TC003				
Test case: Create student/ lecturer				
Test case Description: Test the flow of student / lecturer creation				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Create student/lecturer with empty value	Display an error message “Empty field ”	Display an error message “Empty field ”	Pass
2	Create student / lecturer without profile picture	Display an error message “Missing profile picture ”	Display an error message “Missing profile picture ”	Pass
3	Create student / lecturer with valid input value and profile picture	Display success message “Successfully Created”	Display success message “Successfully Created”	Pass

Table 6.1.4 Create class

Test case ID: TC004				
Test case: Create class				
Test case Description: Test the flow of creating class				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Create class without class ID	Display an error message “Class ID is required ”	Display an error message “Class ID is required ”	Pass
2	Create class with valid input value	Display success message “Successfully Created”	Display success message “Successfully Created”	Pass

Table 6.1.5 View class

Test case ID: TC005				
Test case: View class				
Test case Description: Test the flow of viewing class and student list who taken the class				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Enter incorrect class ID in Class ID field	Display an error message “Class ID is not found”	Display an error message “Class ID is not found”	Pass
	Enter empty value in Class ID field	Display an error message “Class ID is required ”	Display an error message “Class ID is required ”	Pass

Table 6.1.6 Update class

Test case ID: TC006				
Test case: Update class				
Test case Description: Test the flow of updating class with new class details				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Empty Class list	Display a message “No Classes”	Display a message “No Classes”	Pass
2	Update class with valid input value	Display success message “Successfully Updated”	Display success message “Successfully Updated”	Pass

Table 6.1.7 View student / lecturer

Test case ID: TC007				
Test case: View student / lecturer				
Test case Description: Test view student / lecturer profile flow				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Enter incorrect class ID in student / lecturer ID field	Display an error message “student / lecturer ID is not found”	Display an error message “student / lecturer ID is not found”	Pass
	Enter empty value in student / lecturer ID field	Display an error message “student / lecturer ID is required ”	Display an error message “student / lecturer ID is required ”	Pass

Table 6.1.8 Update student / lecturer

Test case ID: TC008				
Test case: Update student / lecturer				
Test case Description: Test student / lecturer profile update flow				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Empty student / lecturer list	Display a message “No student / lecturer”	Display a message “No student / lecturer”	Pass
2	Create student/lecturer with empty value	Display an error message “Empty field ”	Display an error message “Empty field ”	Pass
3	Update student / lecturer with valid input value	Display success message “Successfully Updated”	Display success message “Successfully Updated”	Pass

Table 6.1.9 Create Room

Test case ID: TC009				
Test case: Create Room				
Test case Description: Test room creation flow				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Non NFC Device	Display an error message “This device does not support NFC ”	Display an error message “This device does not support NFC ”	Pass
2.	Invalid NFC Tag	Display an error message “No NFC Tag ”	Display an error message “No NFC Tag ”	Pass
3.	Failed to encode room code into NFC tag	Display an error message “Encode unsuccessful ”	Display an error message “Encode unsuccessful ”	Pass
4.	Successfully encode room code into NFC tag	Display success message “Encoded successfully”	Display success message “Encoded successfully”	Pass
5.	Create Room with empty value	Display an error message “Empty field ”	Display an error message “Empty field ”	Pass
6.	Create Room with invalid value	Display an error message “Invalid value ”	Display an error message “Invalid value ”	Pass

Table 6.1.10 Read and decode data from NFC tag

Test case ID: TC010				
Test case: Read and decode data from NFC tag				
Test case Description: Test decoding NDEF data from NFC tag flow				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Non-NFC Device	Display an error message “This device does not support NFC”	Display an error message “This device does not support NFC”	Pass
2.	Invalid NFC Tag	Display an error message “No NFC Tag”	Display an error message “No NFC Tag ”	Pass

Table 6.1.11 Update class attendance

Test case ID: TC011				
Test case: Update class attendance				
Test case Description: Test class attendance updating flow.				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Non-NFC Device	Display an error message “This device does not support NFC”	Display an error message “This device does not support NFC”	Pass
2.	Invalid NFC Tag	Display an error message “No NFC Tag”	Display an error message “No NFC Tag”	Pass
3.	Class attendance code format is not match with the NDEF data inside NFC Tag	Display an error message “Invalid Code”	Display an error message “Invalid Code”	Pass
4.	Class attendance code format is match with the NDEF data inside NFC Tag	Display a success message “Attendance Updated”	Display a success message “Attendance Updated”	Pass

Table 6.1.12 Update Crowd number in room

Test case ID: TC012				
Test case: Update Crowd number in room				
Test case Description: Test Room Crowd number update flow				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Non-NFC Device	Display an error message “This device does not support NFC”	Display an error message “This device does not support NFC”	Pass
2.	Invalid NFC Tag	Display an error message “No NFC Tag”	Display an error message “No NFC Tag”	Pass
3.	Room Crowd Counter code format is not match with the NDEF data inside NFC Tag	Display an error message “Invalid Code”	Display an error message “Invalid Code”	Pass
4.	Room reached maximum number of people	Display a message “Room full”	Display a message “Room full”	Pass
5.	Room have not reached maximum number of people	Display an message “Welcome”	Display an message “Welcome ”	Pass

Table 6.1.13 Check attendance history

Test case ID: TC013				
Test case: Check attendance history				
Test case Description: Test student class attendance checking flow				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Student does not have any class	Display an message “Empty class”	Display an message “Empty class ”	Pass
2.	Lecturer did not create any class attendance record	Display an message “Empty attendance record ”	Display an message “Empty attendance record ”	Pass

Table 6.1.14 Check room crowd status

Test case ID: TC014				
Test case: Check room crowd status				
Test case Description: Test Room Crowd checking flow				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Admin did not create room	Display an message “Empty room ”	Display an message “Empty room ”	Pass

Table 6.1.15 Create class attendance

Test case ID: TC015				
Test case: Create class attendance				
Test case Description: Test lecturer class attendance creation flow				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Lecturer does not assign with any classes	Display a message “Empty Classes”	Display a message “Empty Classes”	Pass
2	Create the class attendance with valid input	Display a message “All Done !”	Display a message “All Done !”	Pass

Table 6.1.16 Encode attendance code in NFC tag

Test case ID: TC016				
Test case: Encode attendance code in NFC tag				
Test case Description: Test attendance code encoding process				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Non NFC Device	Display an error message “This device does not support NFC ”	Display an error message “This device does not support NFC ”	Pass
2.	Invalid NFC Tag	Display an error message “No NFC Tag ”	Display an error message “No NFC Tag ”	Pass
3.	Failed to encode class attendance code into NFC tag	Display an error message “Encode unsuccessful ”	Display an error message “Encode unsuccessful ”	Pass

Table 6.1.17 View class attendance

Test case ID: TC017				
Test case: View class attendance				
Test case Description: Test class attendance viewing flow				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Lecturer does not assign with any classes	Display an message “Empty Classes”	Display an message “Empty Classes”	Pass
2.	Lecturer does not create any attendance record	Display an message “No attendance record”	Display an message “No attendance record”	Pass

Table 6.1.18 Update student attendance

Test case ID: TC018				
Test case: Update student attendance				
Test case Description:				
No	Test Scenario	Expected Result	Actual Result	Pass / Fail
1	Lecturer does not assign with any classes	Display an message "Empty Classes"	Display an message "Empty Classes"	Pass
2.	Lecturer does not create any attendance record	Display an message "No attendance record"	Display an message "No attendance record"	Pass

6.2 Project Challenges

During the implementation phases, this project encounters some problems and difficulties.

The first problem encounter is the file sharing API between two NFC supported device is deprecated. Started from android API level 29, which is Android 10, the method to share file and message feature is deprecated. Therefore, the origin idea of sending message between lecturer and student using NFC supported device is no longer working anymore. Luckily, the method to manage the NFC tag is still supported by Android. Therefore, user must perform the attendance taking process and communication process through the NFC tag.

In previous proposal, IMEI number will be retrieved to secure the identity of users. However, another issue is encounter during the implementation process is the method to retrieve mobile device's IMEI number is no longer supported by start from android 10. The reason behind this change is changing in user's privacy policy. Any third-party application could not declare the privileged permissions. Therefore, it will affect the method to retrieve device IMEI number. To solve this problem, Android ID is replaced with the IMEI number. Android ID is also a unique identify code for the Android device. It is similar with the IMEI number which use to identify user's device.

6.3 Objectives Evaluation

i) Enhance of the security to protect student information in mobile application

This application could enhance the security to protect student information in mobile application. This can be proven by the testing result. When user use different mobile phones to log in into his account, an error message will be pop up as user's account is registered with the android ID. A validation process will be taken place every time user login into the account. Therefore, any unauthorized users could not log in into the account and fake the student identity. On the other hand, the profile picture of the student will also help to prove the student identity.

ii) Utilize the student attendance system by using NFC technology

Using NFC to take attendance could speed up the attendance record process. There are only three steps for student to take their attendance. Launch the application, click take attendance button, tap their phone on the NFC tag, then student attendance will be updated. Besides, lecturer could also easily create a new class attendance by writing the attendance class code inside the NFC tag. The overall time taken to encode the attendance class code is short.

iii) To improve the attendance rate of student

Since every student account is registered along with the android ID, which means that each person could only take class attendance using his/her phone only. Students have to physically appear in front of the NFC tag with their own phone to decode the unique class attendance and update their attendance status in the database. Therefore, student could not fake their attendance since they must appear to the class with their phone. The chances of "attendance buddy" or taking attendance outside the class problems will be reduced.

Chapter 7

Conclusion and Recommendation

7.1 Conclusion

The final delivery of this project is an android mobile application that could secure student identity and implemented with advance attendance system using NFC.

This application is designed for replacing the old traditional student ID card that is unsecure, easily mislaid, and inefficient to manage. Identity thief could use the printed personal information on the student ID card to perform criminal. It is not only could danger the student but also the education institution as well. Besides, student ID card is small and could be easily mislaid by student when they are away from the campus area. Generally, student would not need to bring along their student ID card with them when they are not in campus. Therefore, student tends to forget the existence of their student ID card. The management of traditional student ID card could be also inefficient and wasting a lot of resources.

Replacing physical student ID card with mobile application not only could provide more security layer to protect student identity, but also provide a much more efficient management on the student identity. Since all student personal information is saved in cloud database, instead of printing student personal information on a one time use physical card, it could provide more convenient for student if their personal information could be displayed on their mobile phone. Therefore, user would not be punished from losing their student card and wasting time to replace a new student card. There is also another benefit of replacing physical card, which is mobile student ID card is much more eco-friendly. Since global warming is a real issue and society is starting to take care of protecting environment. Then it is a great opportunity to implement the mobile student ID card now.

Mobile student ID card could perform more than just securing and displaying the student information. It could also perform better in managing and taking the student attendance using NFC technology. There are a lot of different attendance taking method used by different education institution such as name calling attendance, signing attendance, QR code attendance, Bluetooth attendance and NFC attendance. Using NFC technology not only could provide a more secure environment for student to take

their attendance, but it could also improve the attendance rate. The attendance rate could be improved when every student could only use their registered mobile phone to take the attendance. Which means that every student must attend physically to the class to take their attendance by tapping their phone on the NFC tag.

7.2 Recommendation

There are a lot of potential with this application. Unfortunately, some of the potential functions could not be developed and implemented into the system due to time constraint, immature coding skills and budget.

The first function that wanted to develop but failed at the end is implementing statistical diagram in the Crowd People Counting module. For example, displaying the total number of people who enter to the room this month in bar graph diagram. This feature could provide users a lot of opportunities to analyse the data. The usefulness of this People Crowd Counter module is to control the total number of people inside the room does not exceed the maximum allowed people. By processing the data into graphical diagram could help the users figuring out the best ways to control the crowd number.

The second function that recommend being implemented into this application is encoding a data into NFC tag with a password protection. A protected NFC tag could enhance the security of the NFC tag so that the raw data inside the tag would not be decoded easily. Other than that, NFC tag also having a lot of opportunity such as saving URL link and redirect user to another web application

REFERENCES

[1] S. Reddy, "Why We Keep Losing Our Keys," The Wall Street Journal, 14 April 2014. [Online]. Available: https://www.wsj.com/articles/why-we-misplace-our-keys-phones-wallets-1397517054. [Accessed 10 April 2022].

[2] eSafety, "Deepfake trends and challenges," 11 May 2020. [Online]. Available: https://www.esafety.gov.au/sites/default/files/2020-05/Deepfake-position-statement.pdf. [Accessed 10 April 2022].

[3] GSMA, "Mobile Identity A Regulatory Overview Second edition," GSMA, London, 2015.

[4] A. Salim, "Rise of digital wallets forces student ID card to go mobile," IDG Connect, 14 July 2016. [Online]. Available: https://www.idgconnect.com/article/3577031/rise-of-digital-wallets-forces-student-id-card-to-go-mobile.html. [Accessed 10 April 2022].

[5] G. Staff, "Why attendance matters," Great! SCHOOL.ORG, 11 August 2011. [Online]. Available: https://www.greatschools.org/gk/articles/school-attendance-issues/. [Accessed 10 April 2022].

[6] I. Deliyannis, Interactive Multimedia, Norderstedt: BoD – Books on Demand, 2012.

[7] E. Haselsteiner and K. Breitfuß, Security in Near Field Communication (NFC) Strengths and Weaknesses, Gratkorn, 2006.

[8] R. R. Sabella, NFC For Dummies, United States: For Dummies, 2016.

[9] E. Desai and M. G. Shajan, "A Review on the Operating Modes of Near Field," *International Journal of Engineering and Advanced Technology (IJEAT),* vol. 2, no. 2, pp. 322-325, 2012.

[10] Y. W. LUN, "SMART ATTENDANCE SYSTEM USING QR CODE," 26 April 2019. [Online]. Available:

http://eprints.utar.edu.my/3386/1/FYP_report_corrected.pdf. [Accessed 10 April 2022].

[11] N. C. Office, "GPS Accuracy," National Coordination Office, 3 March 2022. [Online]. Available: https://www.gps.gov/systems/gps/performance/accuracy/#:~:text=GPS%20satellites%%2020broadcast%20their%20signals,4.9%20m%20(16%20ft.). [Accessed 10 April 2022].

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.:2**
Student Name & ID: Eng Mun Shuen 18ACB03271	
Supervisor: Ts Dr Wong Pei Voon	
Project Title: Digital Student ID Card using NFC technology	

1. WORK DONE

Completed Report Chapter 1, 2 and 3 of the written report.

2. WORK TO BE DONE

Development of application module.

i) Attendance module

ii) Room Crowd Counter module

iii) Lecturer/ student management module

3. PROBLEMS ENCOUNTERED

No problem encountered

4. SELF EVALUATION OF THE PROGRESS

Planning the steps of developing each module and make sure that each could completed within two weeks.

wong

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.:2**
Student Name & ID: Eng Mun Shuen 18ACB03271	
Supervisor: Ts Dr Wong Pei Voon	
Project Title: Digital Student ID Card using NFC technology	

1. WORK DONE

Completed Admin use case function. Admin could create room, view room, manage lecturer, manage student, manage class.

Completed lecturer / student management module.

Complete lecturer / student UI design

2. WORK TO BE DONE

Development of application module.

i) Attendance module

ii) Room Crowd Counter module

3. PROBLEMS ENCOUNTERED

Since admin have to manage student, lecturer, class and room and they are related to each other. When admin want to delete any document data that is corresponding with the child/parent table, cascading delete is required. However, Firestore database does not support cascading delete as it is not relational database. Therefore, user need to delete each value separately. It could cased a lot of problem as each document is related to each other.

4. SELF EVALUATION OF THE PROGRESS

UI design must be more user friendly and try to follow 3-click rule.

wong

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.:2**
Student Name & ID: Eng Mun Shuen 18ACB03271	
Supervisor: Ts Dr Wong Pei Voon	
Project Title: Digital Student ID Card using NFC technology	

1. WORK DONE

Completed Student use cases functions. Student could registered, login, update class attendance, update crowd number in room, view profile, view timetable, check room crowd status and check attendance history.

Completed Room Crowd Counter module.

Complete Room Crowd Counter module UI design

2. WORK TO BE DONE

Development of application module.

i) Attendance module

3. PROBLEMS ENCOUNTERED

Selected profile picture that is too big will affect the orientation of the picture. This could be solved by limiting the size of the image view in the XML.

4. SELF EVALUATION OF THE PROGRESS

A test for each function that is developed must be conducted immediately. This could ensure the module function do not affect with the function within the module or other function in other module.

wong

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.:4**
Student Name & ID: Eng Mun Shuen 18ACB03271	
Supervisor: Ts Dr Wong Pei Voon	
Project Title: Digital Student ID Card using NFC technology	

1. WORK DONE

Completed lecturer use case functions. Lecturer could create class attendance, view class attendance, update student attendance, registered, login, update class attendance, view profile, view timetable, check room crowd status.

Completed attendance module.

Complete attendance module UI design

2. WORK TO BE DONE

Testing for each module and report

3. PROBLEMS ENCOUNTERED

No problem encountered

4. SELF EVALUATION OF THE PROGRESS

Focusing on delivering a consistent and simple UI design. These could be done by reducing the number of click and apply similar theme, size and style to all UI component in the application.

wong

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.:6**
Student Name & ID: Eng Mun Shuen 18ACB03271	
Supervisor: Ts Dr Wong Pei Voon	
Project Title: Digital Student ID Card using NFC technology	

1. WORK DONE

Completed Testing for each module and debugging. Completed test case for report.

2. WORK TO BE DONE

Complete Report Chapter 3, Chapter 4, Chapter 5, Chapter 6 and chapter 7.

3. PROBLEMS ENCOUNTERED

No problem encountered

4. SELF EVALUATION OF THE PROGRESS

Testing report should be written after each functions is successfully developed and implement into the system. There are a lot of different test scenario and these scenario should be written and recorded immediately.

wong

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.:8**
Student Name & ID: Eng Mun Shuen 18ACB03271	
Supervisor: Ts Dr Wong Pei Voon	
Project Title: Digital Student ID Card using NFC technology	

1. WORK DONE

Report completed and ready to be submitted.

2. WORK TO BE DONE

FYP2 presentation to supervisor.

3. PROBLEMS ENCOUNTERED

No problem encountered

4. SELF EVALUATION OF THE PROGRESS

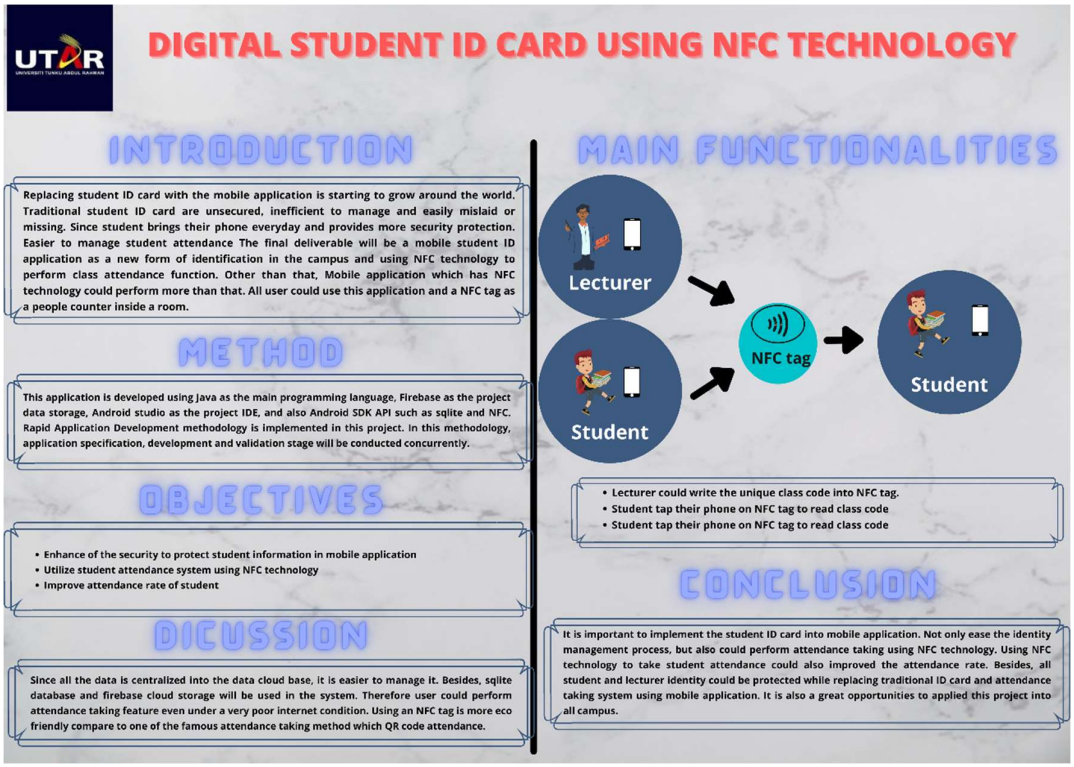
There are a lot of potential functions of this application. However, it could not be developed due to limitation of time and skills. A proper time management could help to implement more ideas into the application.

wong

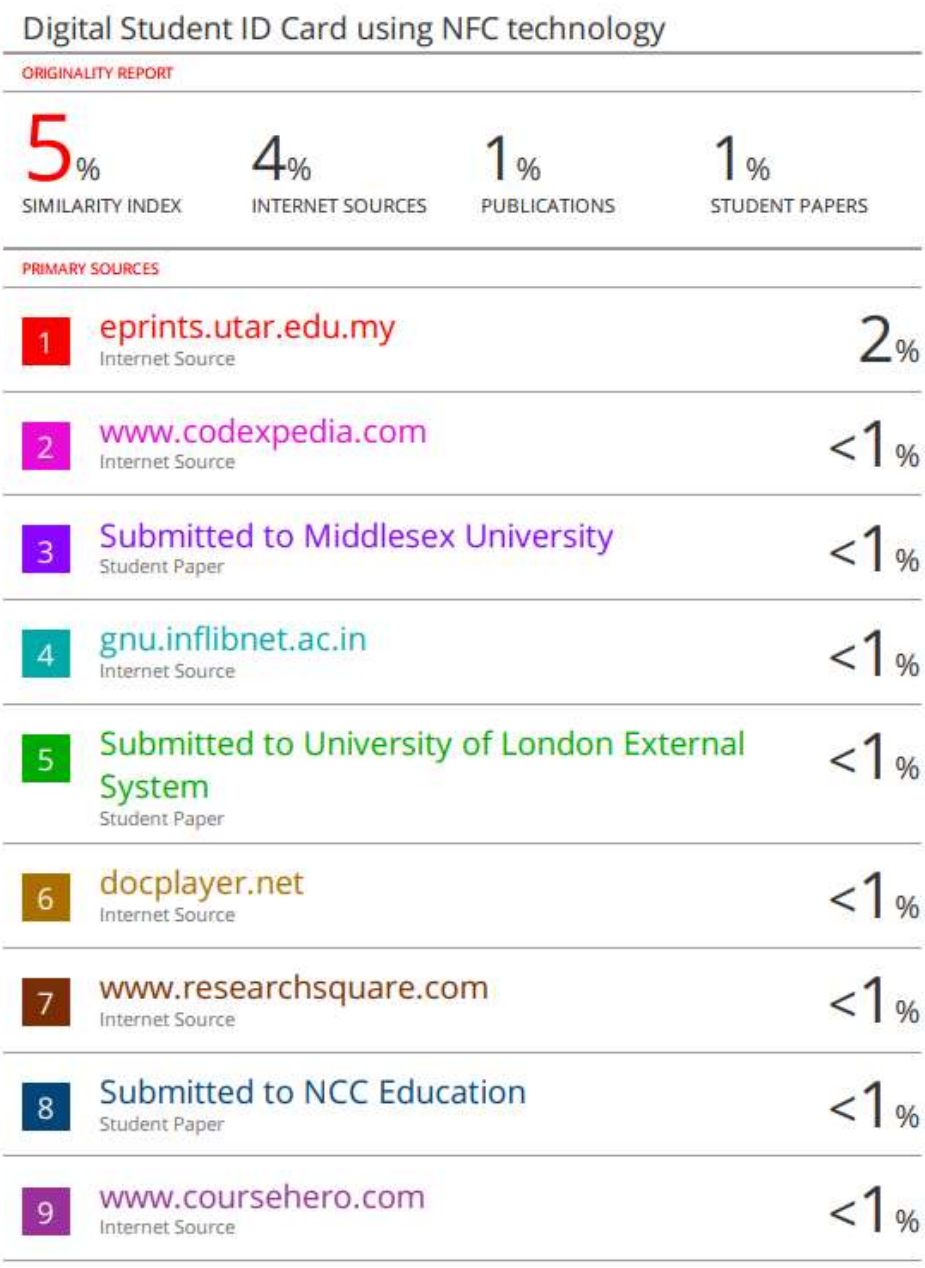
Supervisor's signature

Student's signature

POSTER


DIGITAL STUDENT ID CARD USING NFC TECHNOLOGY
INTRODUCTION
Replacing student ID card with the mobile application is starting to grow around the world. Traditional student ID card are unsecured, inefficient to manage and easily mislaid or missing. Since student brings their phone everyday and provides more security protection. Easier to manage student attendance The final deliverable will be a mobile student ID application as a new form of identification in the campus and using NFC technology to perform class attendance function. Other than that, Mobile application which has NFC technology could perform more than that. All user could use this application and a NFC tag as a people counter inside a room.
METHOD
This application is developed using Java as the main programming language, Firebase as the project data storage, Android studio as the project IDE, and also Android SDK API such as sqlite and NFC. Rapid Application Development methodology is implemented in this project. In this methodology, application specification, development and validation stage will be conducted concurrently.
OBJECTIVES
• Enhance of the security to protect student information in mobile application
• Utilize student attendance system using NFC technology
• Improve attendance rate of student
DICUSSION
Since all the data is centralized into the data cloud base, it is easier to manage it. Besides, sqlite database and firebase cloud storage will be used in the system. Therefore user could perform attendance taking feature even under a very poor internet condition. Using an NFC tag is more eco friendly compare to one of the famous attendance taking method which QR code attendance.
MAIN FUNCTIONALITIES
Lecturer
Student
NFC tag
Student
• Lecturer could write the unique class code into NFC tag.
• Student tap their phone on NFC tag to read class code
• Student tap their phone on NFC tag to read class code
CONCLUSION
It is important to implement the student ID card into mobile application. Not only ease the identity management process, but also could perform attendance taking using NFC technology. Using NFC technology to take student attendance could also improved the attendance rate. Besides, all student and lecturer identity could be protected while replacing traditional ID card and attendance taking system using mobile application. It is also a great opportunities to applied this project into all campus.

PLAGIARISM CHECK RESULT



Digital Student ID Card using NFC technology

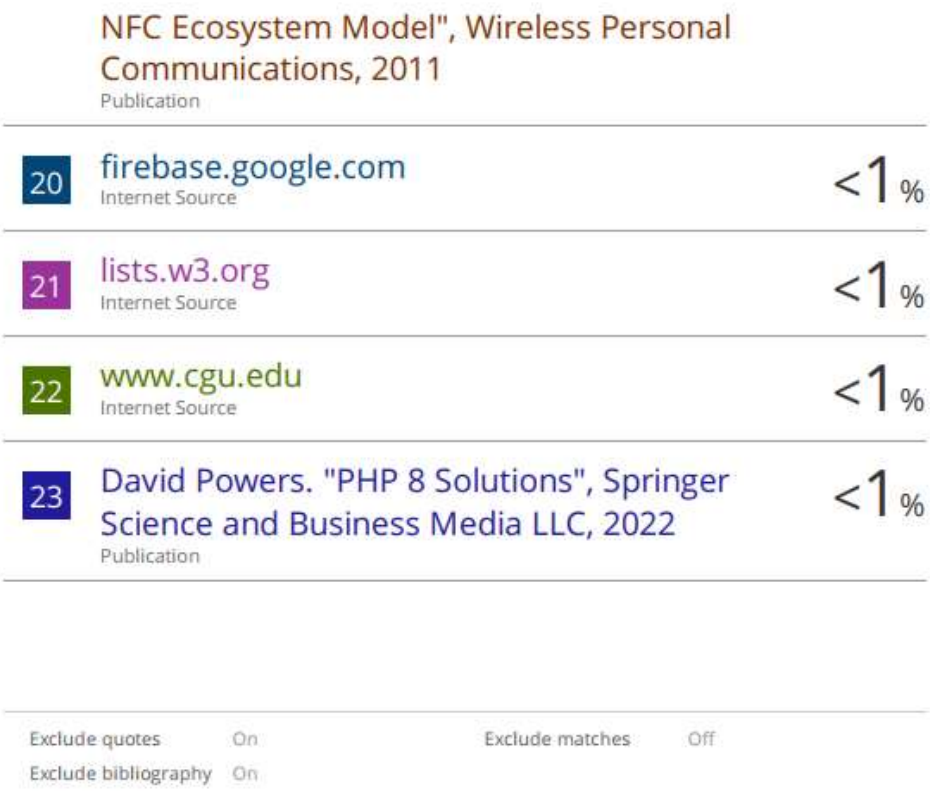
ORIGINALITY REPORT

5%	4%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	2%
2	www.codexpedia.com Internet Source	<1%
3	Submitted to Middlesex University Student Paper	<1%
4	gnu.inflibnet.ac.in Internet Source	<1%
5	Submitted to University of London External System Student Paper	<1%
6	docplayer.net Internet Source	<1%
7	www.researchsquare.com Internet Source	<1%
8	Submitted to NCC Education Student Paper	<1%
9	www.coursehero.com Internet Source	<1%

10	A. Ruiz, Y.W. Price. "Test-Driven GUI Development with TestNG and Abbot", IEEE Software, 2007 Publication	<1 %
11	Submitted to City University Student Paper	<1 %
12	fict.utar.edu.my Internet Source	<1 %
13	pdfcoffee.com Internet Source	<1 %
14	seminarprojects.blogspot.com Internet Source	<1 %
15	www.framentos.com Internet Source	<1 %
16	www.systemsjunction.com Internet Source	<1 %
17	Hsuan-Yu Liu, Chen-Yi Lee. "A Low-Complexity Synchronizer for OFDM-Based UWB System", IEEE Transactions on Circuits and Systems II: Express Briefs, 2006 Publication	<1 %
18	tools.ages.pucrs.br Internet Source	<1 %
19	Kerem OK, Vedat Coskun, Busra Ozdenizci, Mehmet N. Aydin. "A Role-Based Service Level	<1 %


NFC Ecosystem Model", Wireless Personal Communications, 2011
Publication
20 firebase.google.com
Internet Source
<1%
21 lists.w3.org
Internet Source
<1%
22 www.cgu.edu
Internet Source
<1%
23 David Powers. "PHP 8 Solutions", Springer Science and Business Media LLC, 2022
Publication
<1%
Exclude quotes On
Exclude matches Off
Exclude bibliography On

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)	Eng Mun Shuen
ID Number(s)	18ACB03271
Programme / Course	Bachelor of Information System(Honours) Information System Engineering
Title of Final Year Project	Digital Student ID card using NFC technology

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 5 % **Similarity by source** Internet Sources: 4 % Publications: 1 % Student Papers: 1 %	
Number of individual sources listed of more than 3% similarity: 0	

Parameters of originality required and limits approved by UTAR are as Follows:
- **(i) Overall similarity index is 20% and below, and**
- **(ii) Matching of individual sources listed must be less than 3% each, and**
- **(iii) Matching texts in continuous block must not exceed 8 words**

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.

wong

Signature of Supervisor		Signature of Co-Supervisor
Name: Ts Dr Wong Pei Voon		Name: _____
Date: 22/04/2022		Date: 22/04/2022

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

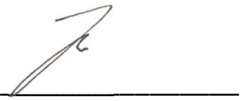
CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	18ACB03271
Student Name	Eng Mun Shuen
Supervisor Name	Ts Dr Wong Pei Voon

TICK (√)	DOCUMENT ITEMS 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.
✓	Front Plastic Cover (for hardcopy)
✓	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.



(Signature of Student)
Date: 19 /04/2022