

Booking Application for University Facilities



**Bachelor of Engineering in Software Engineering
International College
King Mongkut's Institute of Technology Ladkrabang
Academic Year 2017
KMITL-2018-IC-B-003-005**

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use

Final Report - academic year 2017

Bachelor of Engineer in Software Engineering
International College
King Mongkut's Institute of Technology Ladkrabang

Title - Booking Application for University Facilities

Authors:

Tankhun Suriyanonrin 56090028

Pisit Kittichaovanun 56090035

Dol Duangmanee 56090036



Isara Anantavasilp

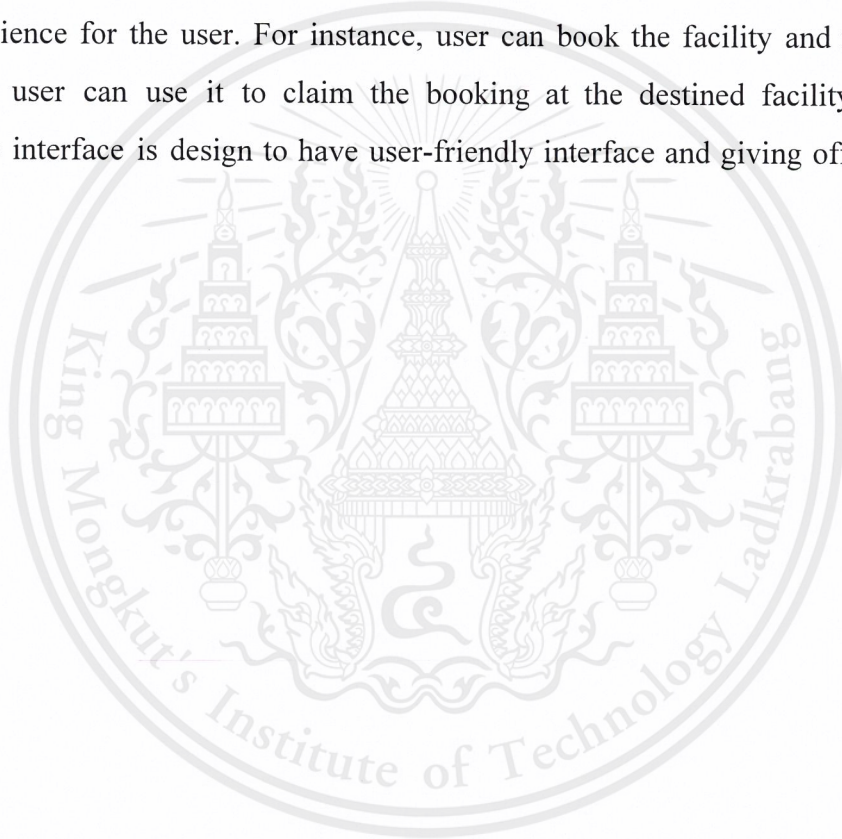
Dr. Isara Anantavasilp

Advisor

Date: *8/6/2018*

Abstract

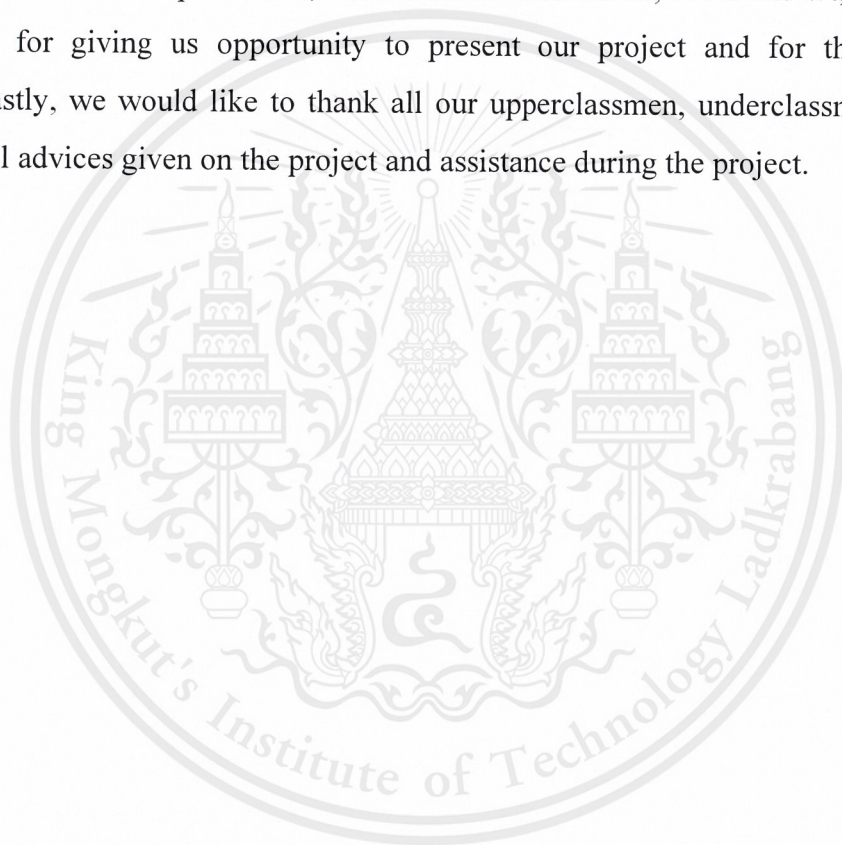
In the present day, smartphone has become one of the technology that human cannot live without. It took part in almost every task of our daily life, from the alarm clock to start our day to the music we listen to before bed. However, it is not the phone that is essential, it is the applications that run on the phone. This paper discusses the overall process of developing the Booking Application, an smartphone application for students and staff to use to book the facility within KMITL. The application has various features that provided convenience for the user. For instance, user can book the facility and from the QR code given user can use it to claim the booking at the destined facility easily. Furthermore, the interface is design to have user-friendly interface and giving off easy to use vibe.



Acknowledgement

The completion of this project, Booking Application, could not have been possible without the support and assistance of so many people. Therefore, in this section we would like to show our gratitude toward the following people:

First of all, we would like to thank our advisor, Dr. Isara Anantavrasilp, for the valuable guidance and suggestions which help us throughout the entire project. Secondly, our thank also goes to all the professors, Asst.Prof.Dr. Chaiwat N., Dr. Ukrit W., and Dr. Natthaphong J., for giving us opportunity to present our project and for their kind cooperations. Lastly, we would like to thank all our upperclassmen, underclassmen, and our friends for all advices given on the project and assistance during the project.



Contents

1. Introduction	7
1.1 Motivation	7
1.2 Objective	7
1.3 Scope of Work	8
1.4 Procedure	9
2. Problem Description and Related Works	10
2.1 Problem Description	10
2.2 Review of Related Works	10
2.2.1 Service Reservation System of Kasetsart University's Library	10
2.2.2 Tutoring Room Reservation System of Thammasart University	11
2.2.3 Eatigo - Discounted Restaurant Reservation Application	12
2.2.4 Vaniday - Beauty Booking Application	13
2.2.5 Skedda - Management System Application	14
3. Requirement and Analysis	17
3.1 System Requirements	17
3.1.1 Functional	17
3.1.2 Usability	17
3.1.3 Performance	17
3.1.4 Supportability	17
3.1.5 Reliability	17
3.2 Use Case Diagram	18
3.2.1 User Use Case Diagram	18
3.2.2 Use Case Description	18
3.2.3 Fully Dressed Use Case	19
4. Software Architect and Diagrams	25
4.1 Software Design	25
4.2 Sequence Diagram	26
5. Software Design	29
5.1 Overall Architecture	29
5.2 Activity Flow Diagram	30

6. Development and Tools	33
6.1 Development Process	33
6.2 Development Tools	34
6.2.1 Android SDK	34
6.2.2 WebStorm	34
6.2.3 React Native	34
6.2.4 Mobx	34
6.2.5 Axios	34
6.2.6 Node.js	34
6.2.7 Git and Github	35
6.2.8 SourceTree	35
6.2.9 BitBucket	35
6.2.10 PostgreSQL	35
6.2.11 PgAdmin	35
6.2.12 Heroku	35
7. Result	36
8. Evaluations and Discussions	43
8.1 Effectiveness	43
8.2 Future Work	43
9. Conclusion	44
9.1 Summary	44
9.2 Lesson Learned	45
9.2.1 Time Management	45
9.2.2 Learn to be Flexible	45
9.2.3 New Techniques Learned	45
9.2.4 UI/UX Design and Guidelines	45
9.3 Problem and Obstacles	45
9.3.1 Problem With Other Project's Tools	45
9.3.2 React Native State Management	45
9.3.3 Different in Version	46

2.2.5 Skedda - Management System Application

Skedda is a management system that helps users organize their venue. This system is able to integrate to user's website or mobile application. The integration enables user's product online booking, online payment, booking management, customer management, and showing venue's availability.



Application	Category	Platform	Front -End	Back End	Reserving Time Display	Time Limited
Service Reservation System of Library	Library's Facilities	Website	HTML / PHP	SQL	Time Period	1 Hour
Tutoring Room Reservation System	Tutoring Room	Website	HTML / PHP	SQL	Time Period	2-3 Hours
Eatigo - Restaurant Reservations	Restaurant	Mobile Application	Java / Javascript	SQL	Time Period	No Time Limit
Vaniday - Beauty Booking App	Beauty Salon	Mobile Application	Java / Javascript	SQL	Custom by User	No Time Limit
Skedda	Manage System	Website / Mobile Application	HTML / Javascript			
Our Application	University's Facilities	Mobile Application	HTML / Java / Javascript	SQL	Custom by User	No time Limit

Table 2.2.6 the conclusion table of the developing tools, platform, category, reserving time display, and time limited of reservable time for user of the relevant applications.

Application	User ID Require	Need Verification	Verification Type	Multiple User Reservation	Invite Friend
Service Reservation System of Library	Yes	Yes	Manual	No	No
Tutoring Room Reservation System	Yes	Yes	Manual	No	No
Eatigo - Restaurant Reservations	No	Yes	Digital Code	Yes	No
Vaniday - Beauty Booking App	No	Yes	Digital Code	No	No
Our Application	Yes	Yes	Digital Code	Yes	No

Table 2.2.7 the conclusion table of the core function that are similar to the project and a comparison between all listed applications. Which the applications with User ID require is an application that need to login to access the system. As for the verification to claim the booking all application require a mean to verify themselves but some applications using manual process and some using digital code to verify.

Requirement and Analysis

3.1 System Requirements

3.1.1 Functional

User

- User able to view all facilities open for booking
- User able to view detailed of certain facility such as operating time, vacant, contact number, and etc.
- User able to make reservation of the facility if the place is vacant.
 - all reservation of individual place have their own requirement and regulation.
 - the input required for booking: date and time range.
- User must send the minimum invitation required by the facility to the participants.
- User must scan the given QR code at the QR code reader to verified as the reserved person.

3.1.2 Usability

- The program UI follows Android/IOS design guideline.
- The application use Java language to implement. The interface design is well-managed, simple, and organized.

3.1.3 Performance

- The program has a fast response time.
- The program dynamically loads data to display.

3.1.4 Supportability

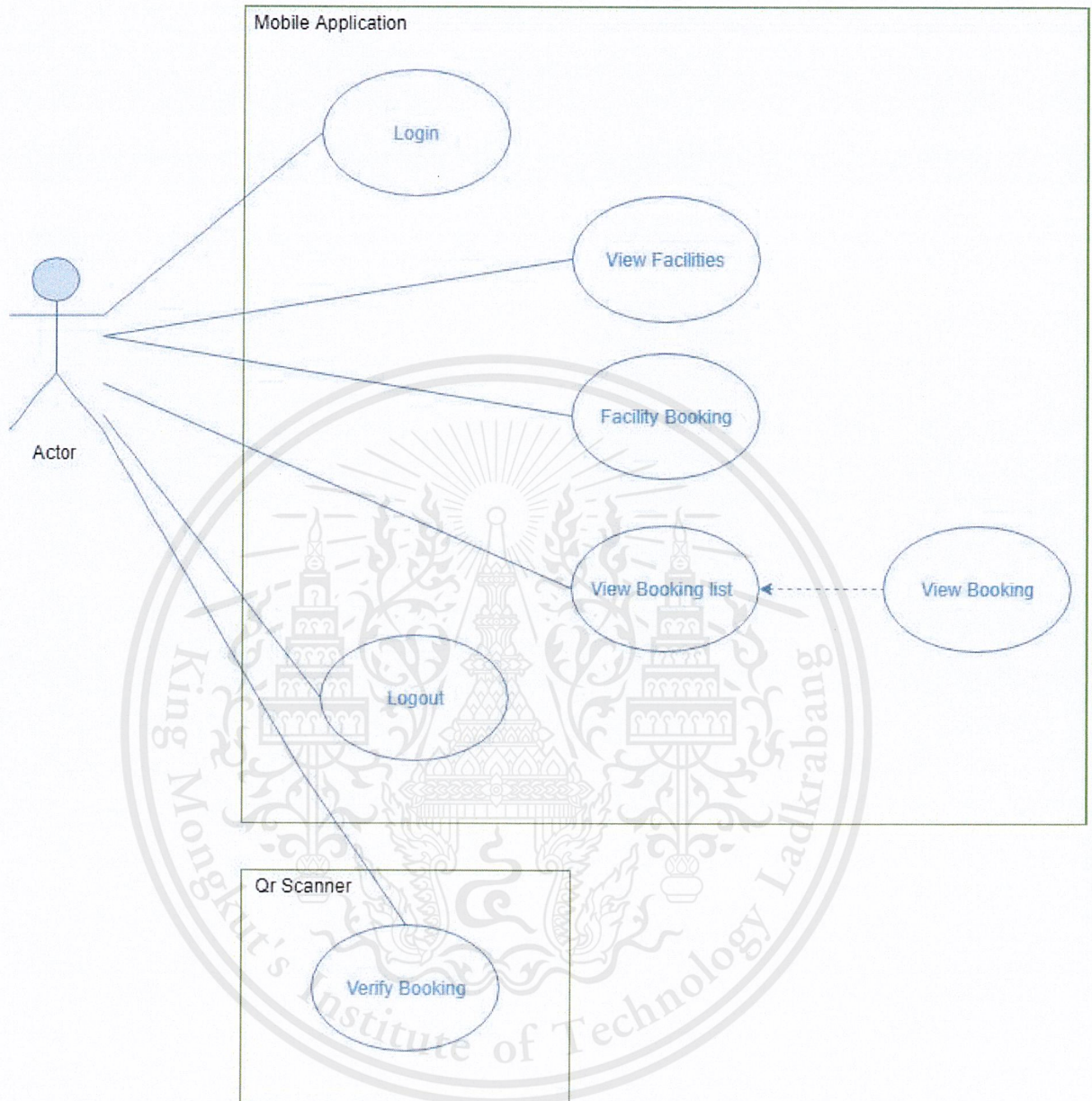
- System will notified the user if there are any update to the system. System provided tutorial for how the program work.
- The program is able to run on both Android 4.2+ and IOS 9.3+.

3.1.5 Reliability

- The application is bug free and work perfectly without crash.
- The program UI follows Android/IOS design guideline.

3.2 Use Case Diagram

3.2.1 User Use Case Diagram



3.2.2 User Use Case Description

- Login: The user enters username and password then press login. The system verifies and sends respond back to the user client. If login is successful, the client will locate the user to the main screen, or else it will show the error that receives from the server.
- Logout: The user selects profile tab then press logout. The client shows dialog that asks to confirm whether to logout. If user confirm, the client will be logout and will be redirect to the login screen.
- View Facility: In the main page or home tab bar. The list of facilities will be shown.

This material is reserved for educational use only, not allowed for commercial use.

- Facility Booking: User fill all the requirement text fill to make reservation.
- Claim Booking: Scan QR code at the facility to claim the booking.
- View Booking list: Select profile tab and press My Booking to view all active reservation.

3.2.3 Fully Dressed Use Case

Table 3.2.1.1 Login

Use Case	Login	
Primary Actor	User	
Precondition	-	
Postcondition	logged in	
Flow of Events	User	System
	1. Fill in correct id and password	
		2. System verify the password
		3. If system return true
		4. redirect user to the main screen of the application
Alternative Flow : Incorrect id or password		3. System return false and notified user that it's incorrect.
	4. User able to re-enter id and password again	

Table 3.2.1.2 View facilities

Use Case	View facilities	
Primary Actor	User	
Precondition	login	
Postcondition		
Flow of Events	User	System
	1. press on the “Explore” tab	
		2. System get the facility that open for booking with the application from database
		3. System show list of all the data obtained into list view
	3. press the facility	
		4. redirect user to the selected facility info page
		5. Show all info about the facility(operating time, time available, regulation)

Table 3.2.1.3 Facility booking

Use Case	Date select, Facility booking, Send Invitation	
Primary Actor	User	
Precondition	login and selected facility	
Postcondition	fill in all required text field	
Flow of Events	User	System
		1. System show list of facilities from database
	2. User select the facility	
		3. Lead to main page of the selected facility
	4. Fill all required text field and choose a time range that are available	
	5. press "Send invitation"	
		6. System send all the information to server in order to make reservation.
		7. System book the facility and generate QR code and pass it out to all users participated
	8. System display the respond from the server which is QR code and important info.	

Table 3.2.1.4 Claim booking

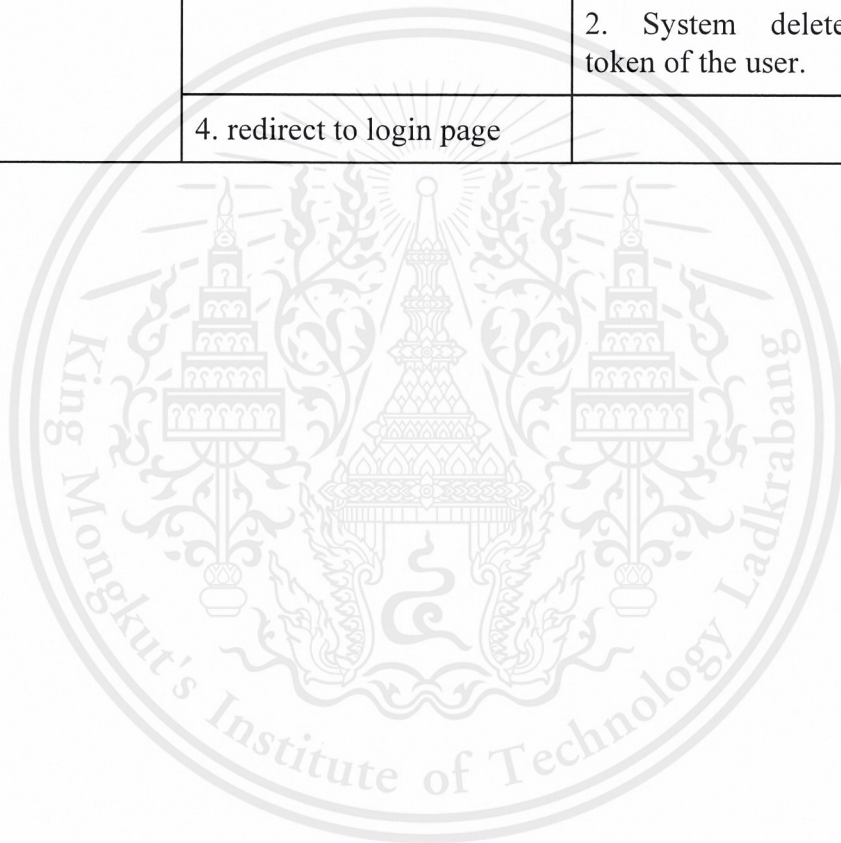
Use Case	Claim booking	
Primary Actor	User	
Precondition	Successfully booked the facility	
Postcondition		
Flow of Events	User	System
	1. go to the selected facility with the generated QR code	
	2. Scan the QR code at the QR code reader	
		3. System verify the scanned QR code using QR code id
		4. System assigns and shows a court/field number to user
	5. User can use the facility at their own leisure	
Alternative Verification Failed/Booking expired	Flow:	3. System shows error message back to user

Table 3.2.1.5 View Booking list and View Booking

Use Case	View Booking list and View Booking	
Primary Actor	User	
Precondition	login and selected profile tab	
Postcondition	-	
Flow of Events	User	System
		1. System show list of booking that match the user_uuid from the server
	2. User select certain facility from the shown list.	
		3. Send booking_uuid to the server, user respond all the info of the booking.
	4. Display the respond from server.	

Table 3.2.1.6 Logout

Use Case	Logout	
Primary Actor	User	
Precondition	login and select profile tab bar	
Postcondition	user got logout and go back to login page	
Flow of Events	User	System
	1. press “Logout” button	
		2. System delete login token of the user.
	4. redirect to login page	



Software Architect and Diagrams

4.1 Class Diagram

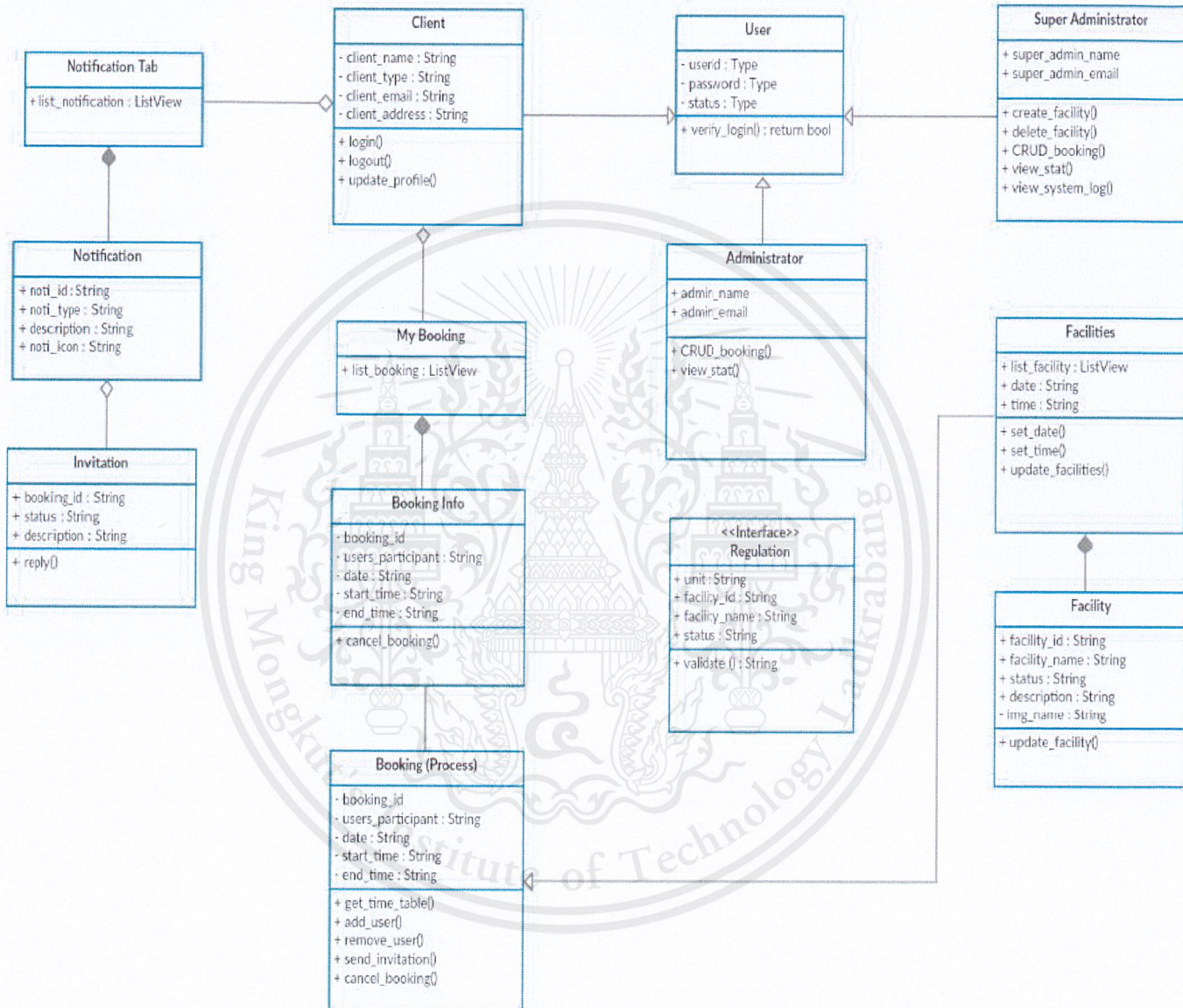


Figure 4.1 class diagram

4.2 Sequence Diagram

4.2.1 User Sequence Diagram

Login

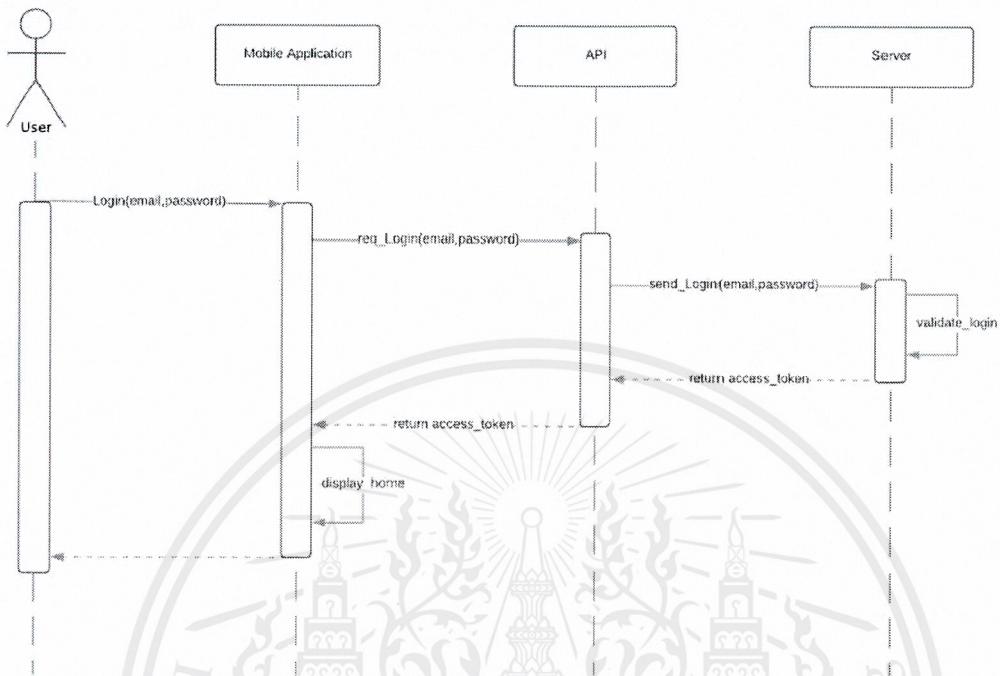


Figure 4.2.1.1 Sequence diagram of user logging in to application.

View facilities

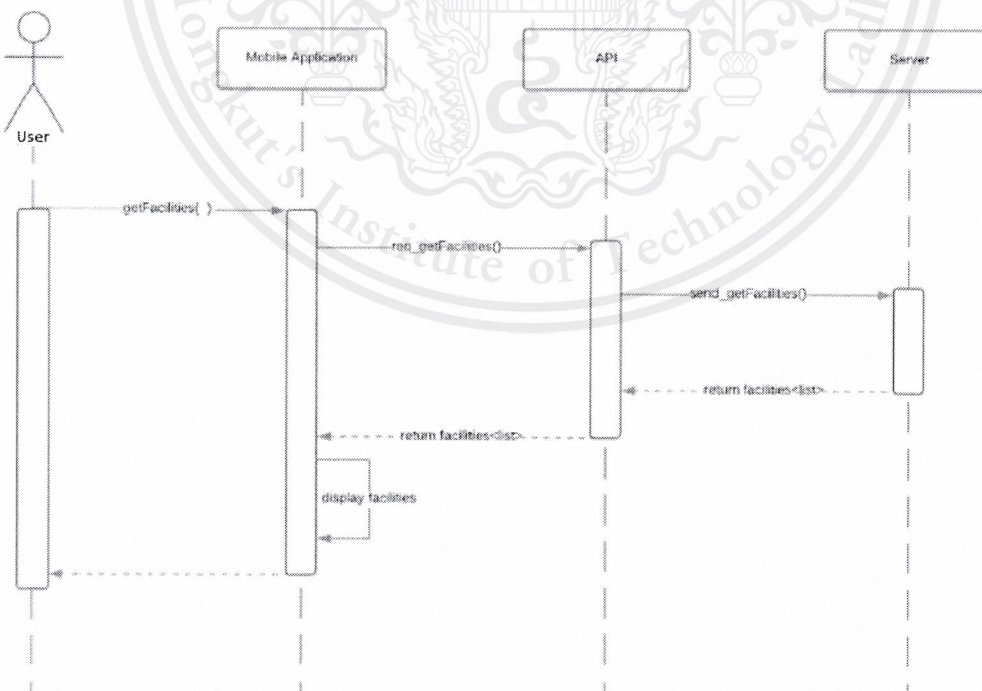


Figure 4.2.1.2 Sequence diagram of user successfully login and get accessed to home page of the application

Facility Booking

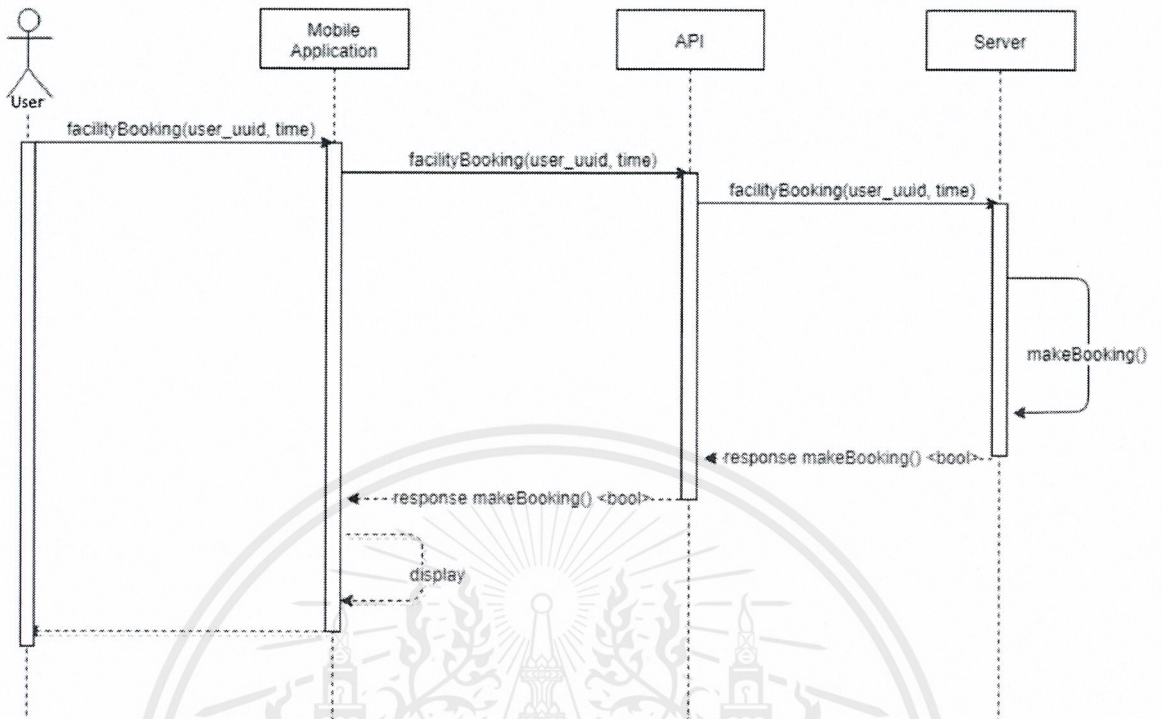


Figure 4.2.1.3 Sequence diagram of user booking facility from the listed facilities shown in home tab.

View Booking List and View Booking

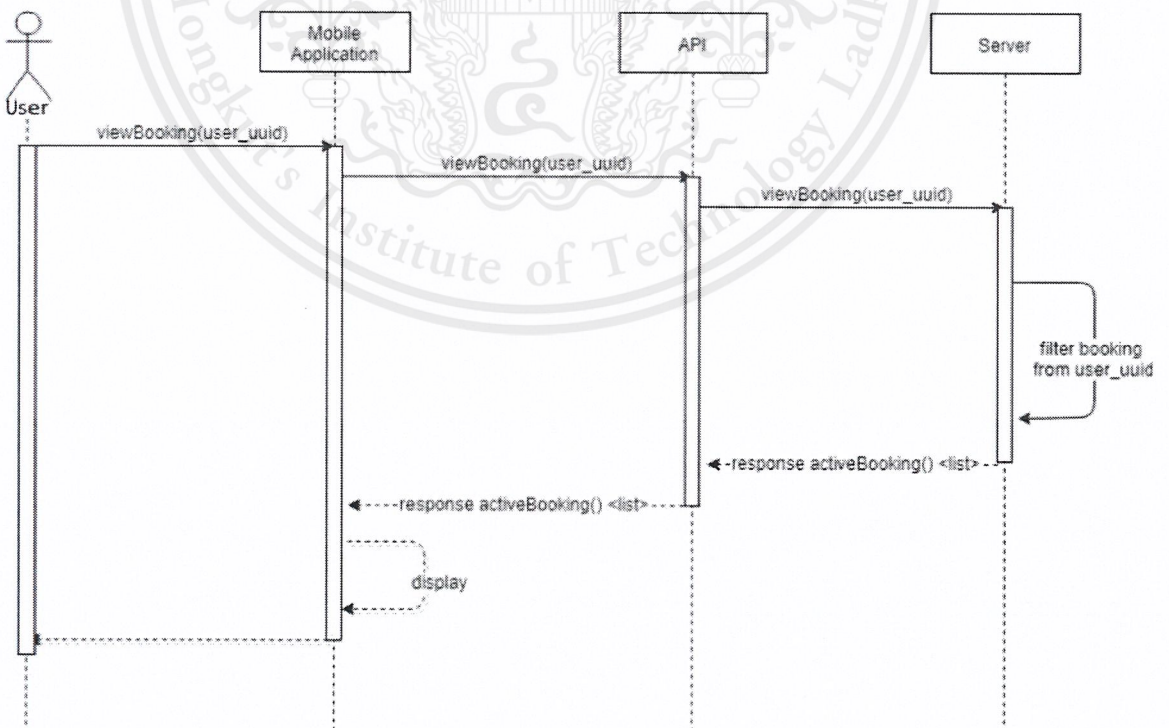


Figure 4.2.2.2 Sequence diagram of admin adding new available facility to the home page.

This material is reserved for educational use only, not allowed for commercial use.

Claim Booking

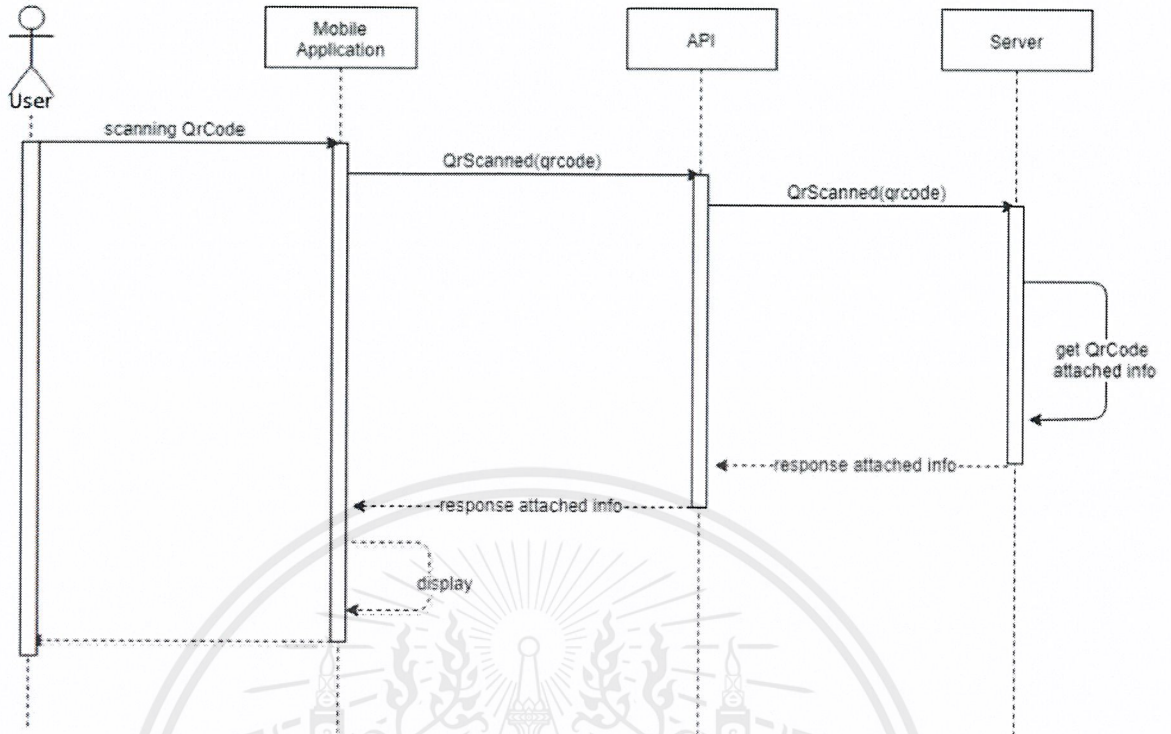
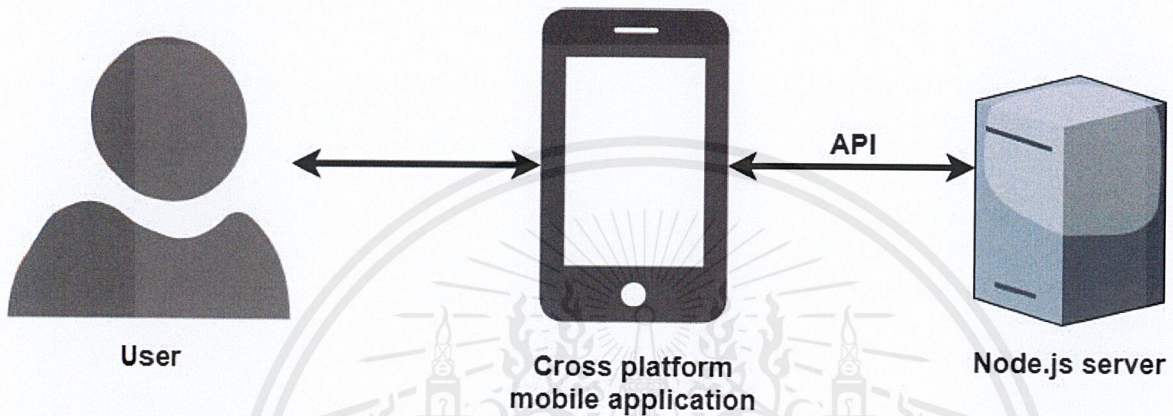


Figure 4.2.2.3 Sequence diagram of admin removing facility from the home page.

Chapter 5

Software Design

5.1 Overall Architecture



Our application consists of two main parts:

- Cloud Server: manage the database and any data request from the mobile application.
- Mobile application: provided for students and staffs of the university to use in order to book the facility.

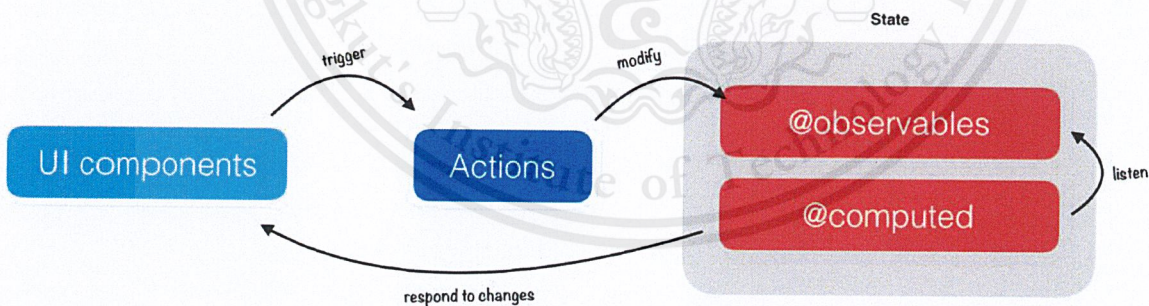


Figure 5.1.1 This diagram illustrated the system component of client mobile application state management of each screen or UI components. From the user triggering the action which server return data which need to be used later such as token and user uuid. The value can be store as an observables object. The observables object can be call from any UI components and can be change or compute.

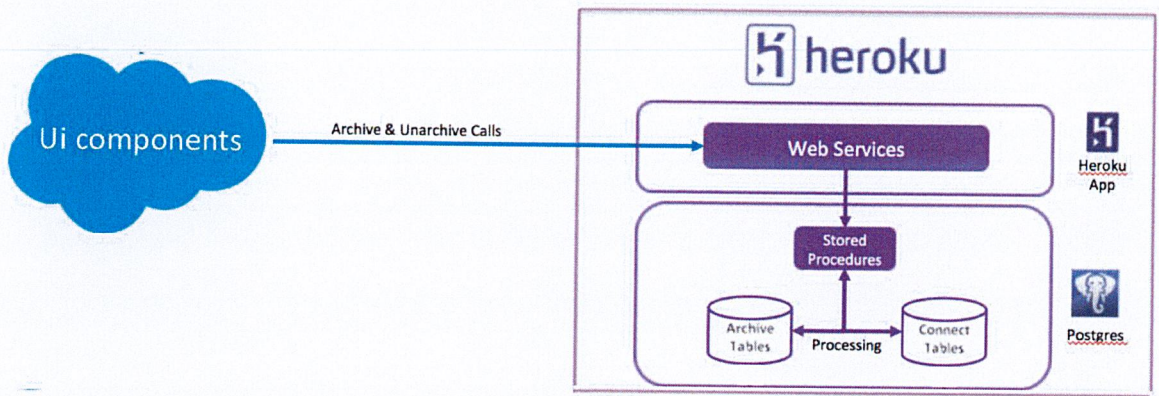


Figure 5.1.2 This diagram illustrated the UI components communicating with the server and database on cloud platform called Heroku. From **Figure 5.1.1** the action which user trigger is this diagram. The UI components sending the request to the destination API and the web service will called the database which is PostgreSQL to get the result and response back as a Json data format to the client system component.

5.2 Activity Flow Diagram

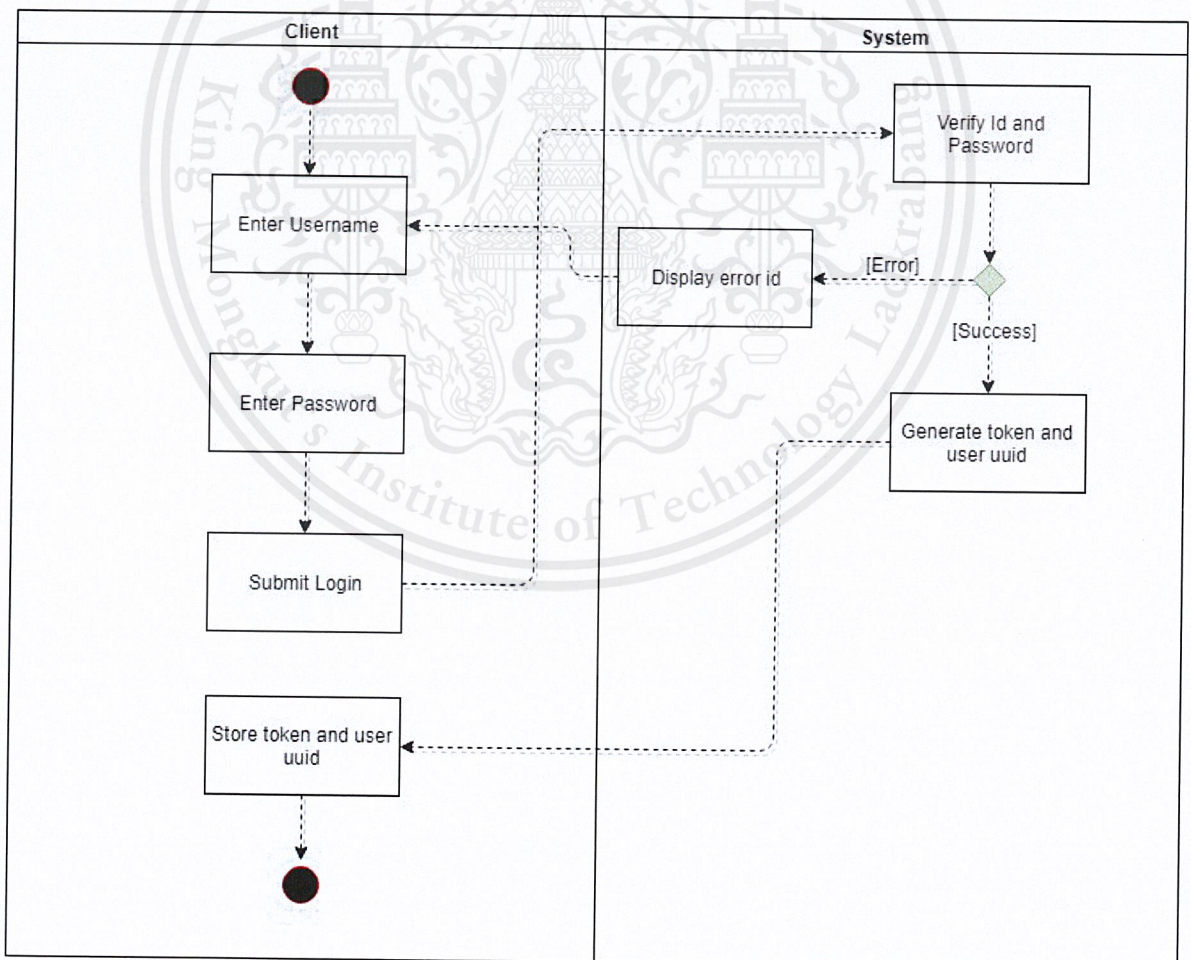


Figure 5.2.1 Acitivity diagram of user login in to the system.

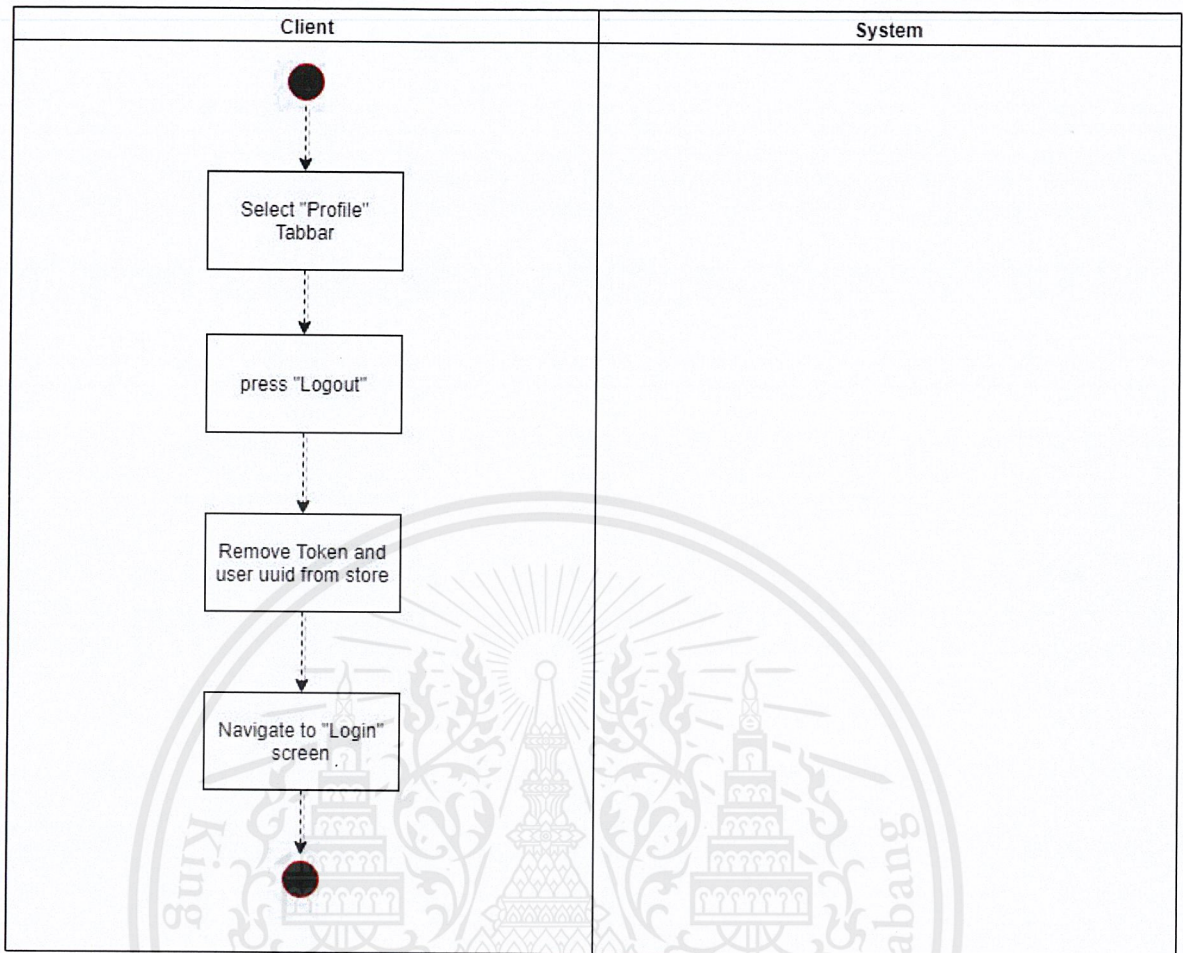


Figure 5.2.2 Activity diagram of user login out of the system

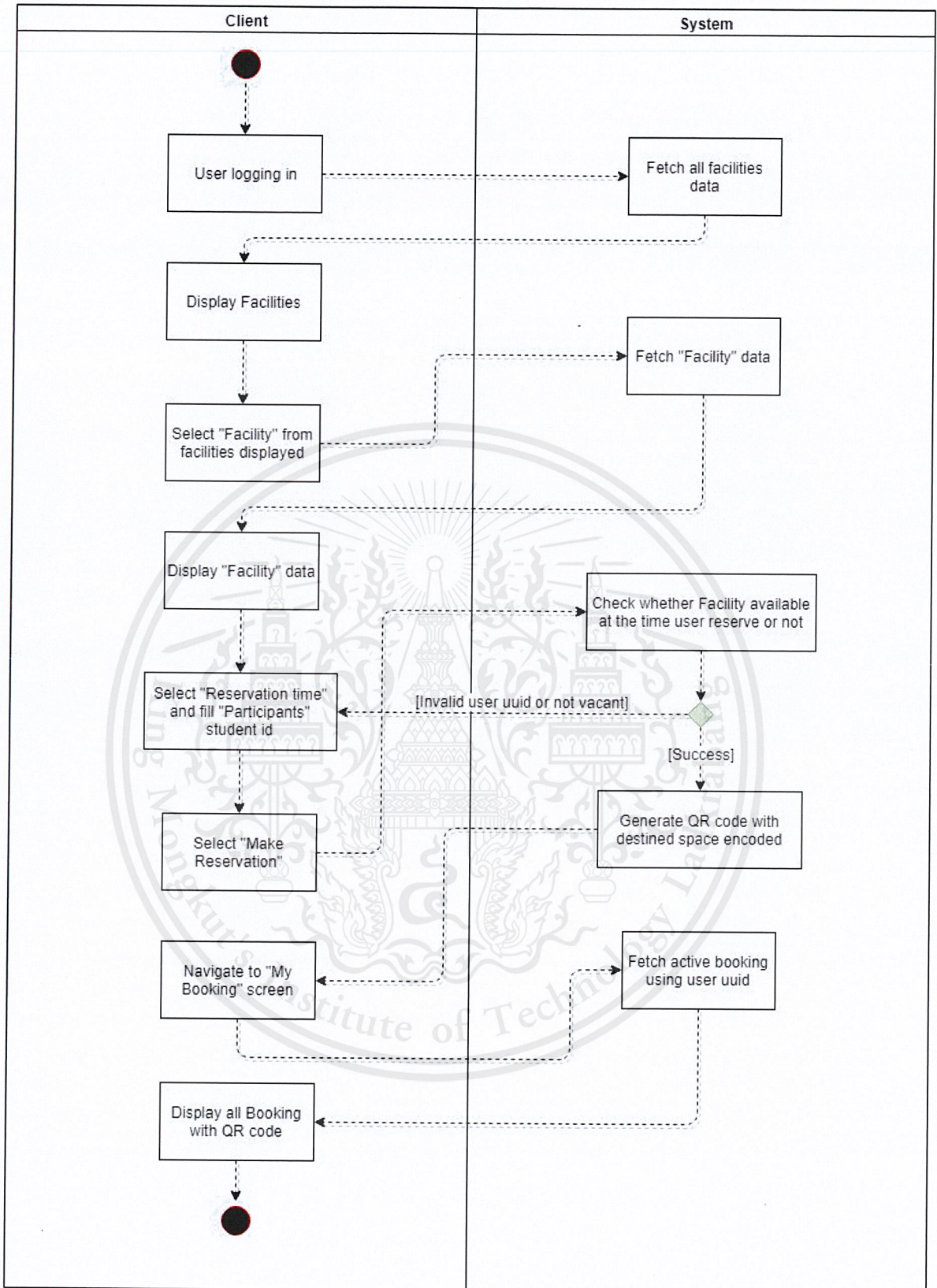
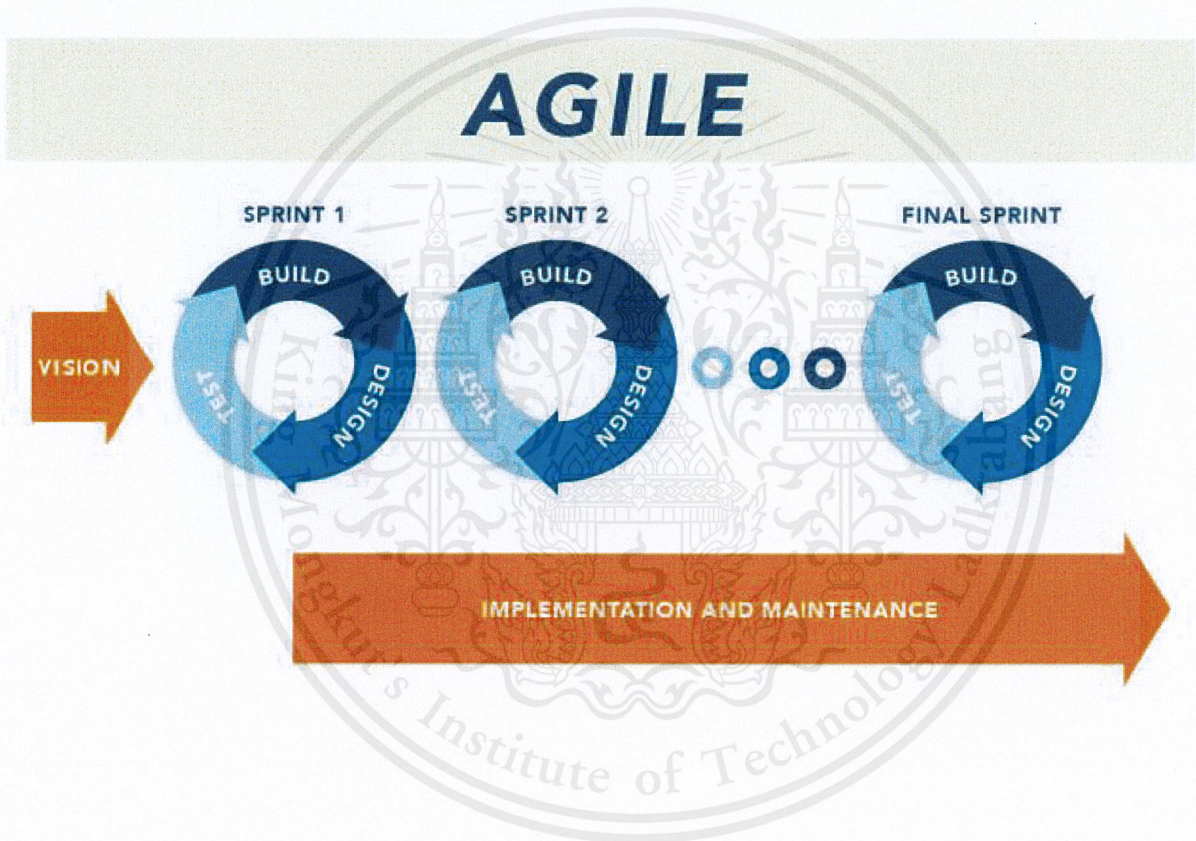


Figure 5.2.3 Activity diagram of user choosing the desire facility, make a reservation of the facility. Then send the information filled to the server and check whether the space is vacant or not. Then view the active booking of the user.

Development and Tools

6.1 Development Process

Agile Development Process, as can see from the figure from sprint to sprint more function will be add to the project. As the project adds more function to it and keep repeating the sprint, it is an incremental and iterative development process.



6.2 Development Tools

6.2.1 Android SDK

The Android SDK is a set of development tools used to develop applications for Android platform. Although we use Ionic as our main SDK, we need to use Android SDK pairing with Apache Cordova when compiling our program on Android mobile devices.

6.2.2 WebStorm

WebStorm is a powerful and smart editor. WebStorm is an intelligent code editor with various functions such as on-the-fly error detection, powerful navigation and refactoring for JavaScript, TypeScript, and stylesheet language. Furthermore, as the project use React native there is a built-in React Native debugger provided.

6.2.3 React Native

React Native framework is a software development kit (SDK) build on top of Node.js for developing web application on mobile devices. Need to use in conjunction with Node.js as a development server when building on mobile devices.

6.2.4 Mobx

MobX is a state management library which is simple, scalable, and battle tested state management solution. People often use MobX as alternative for Redux. But please note that MobX is just a library to solve a technical problem and not an architecture or even state container in itself.

6.2.5 Axios

Axios is a library use to handle json request and response with the user and application header handler. Axios also capable of make http requests from node.js and support the promise API.

6.2.6 Node.js

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

6.2.7 Git and Github

We use Git as a main method for version control/source control while hosting our git repository of our project on Github.

6.2.8 SourceTree

SourceTree is a GUI for git version control. It helps increase our productivity in the project since it renders many git commands into a click button so we don't need to mess with git command through the command line.

6.2.9 BitBucket

BitBucket has been used to share files and update to other member in the team so the files can be easily edit. And just in case something happen to our laptop or in case that we lost the files then we still have it store in the cloud.

6.2.10 PostgreSQL

PostgreSQL is an object-relational database management system (ORDBMS) with an emphasis on extensibility and standards compliance. As a database server, its primary functions are to store data securely and return that data in response to requests from other software applications.

6.2.11 PgAdmin

PgAdmin is the most popular and feature rich open source administration and development platform for PostgreSQL. Pgadmin is a web application that use to manage the database through the browser.

6.2.12 Heroku

Heroku is a cloud platform that lets companies build, deliver, monitor, and scale apps. Heroku is a platform for data as well as apps - providing a secure, scalable database-as-a-service with tons of developers tools like database followers, forking, dataclips and automated health checks. Heroku is an ecosystem of cloud services, which can be used to instantly extend applications with fully-managed services.

Chapter 7

Result

7.1 Key Functional

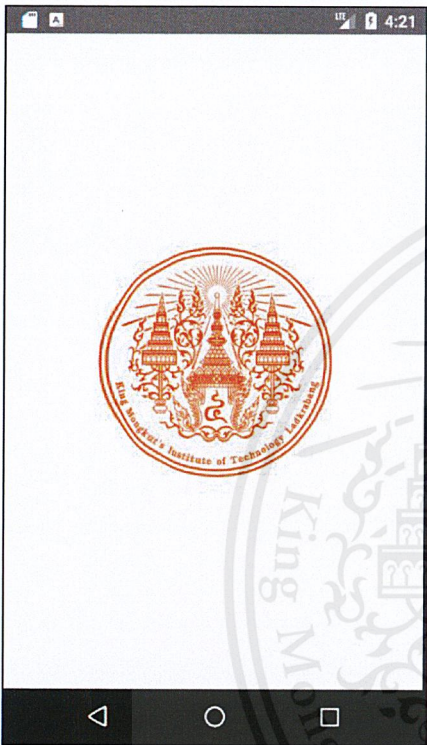


Figure 7.1.1 Splash Page

Purpose of the splash screen is to wait for the device to check whether there is token stored or not. If there is no token stored the screen will redirect to the login screen, On the other hand, redirect to the main screen of the application.



Figure 7.1.2 Login Page

If user already have their ID and Password for the application user can login using this page or else user need to sign up using sign up button down below.

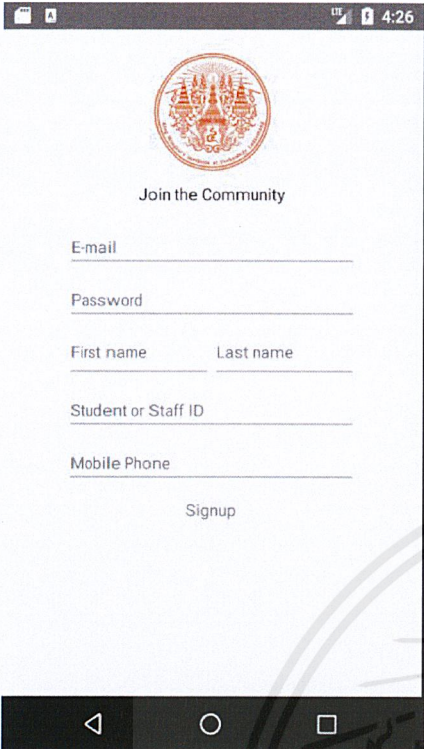


Figure 7.1.3 Sign up Page

User need to fill all the blank space in order to sign up after user successfully sign up the screen will redirect to the main screen and store the token use for persistent login.



Figure 7.1.4 Home Tabbar

Home Screen is a tab bar navigator with two tab bars which is home and profile.

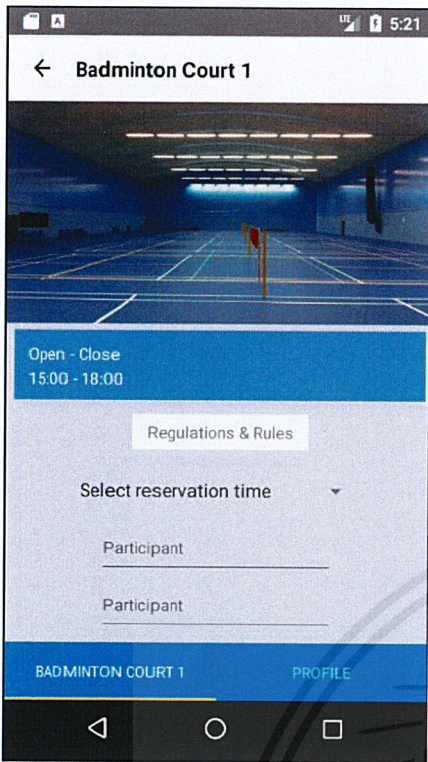


Figure 7.1.5 Facility Screen

Facility screen is a screen that show necessary information of the facility that have been chosen by the user from the home screen. In this page user able to book the facility by choosing the reservation time and fill in the amount minimum requirements for participants and press “Make Reservation” button down below.

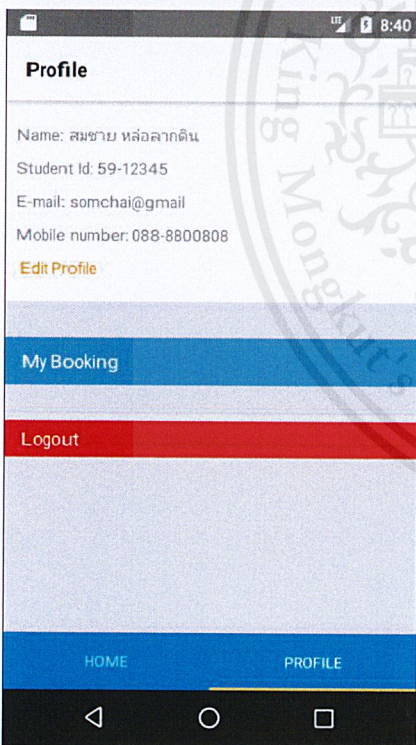


Figure 7.1.6 Profile Tabbar

Profile tabbar is a tabbar which show user information, In this screen user can edit their profile, view the booking user have book, and logout.



Figure 7.1.7 My Booking Screen

Show the booking that user has successfully booked. Which the information will inform the facility, time, and a QR Code.



Figure 7.1.8 QR scanner home screen

QR Code scanner application main interface for facility caretaker. Caretaker can choose from the interface which facility they are in charge of.

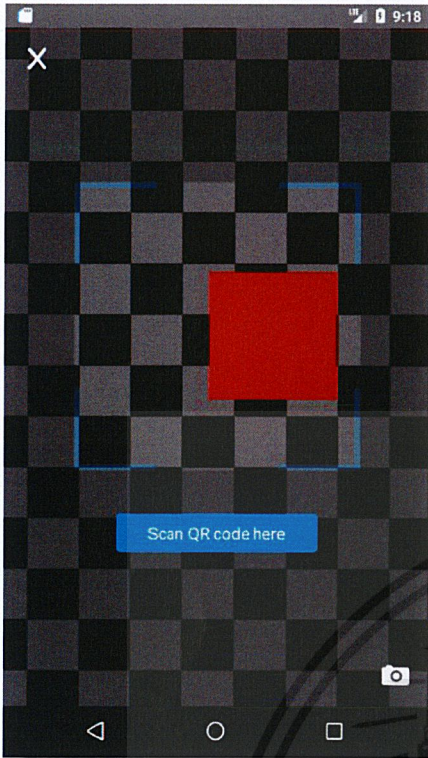


Figure 7.1.9 QR scanner screen

QR scanner after choosing the facility, user can claim their booking by going to the caretaker and scan the qr code.

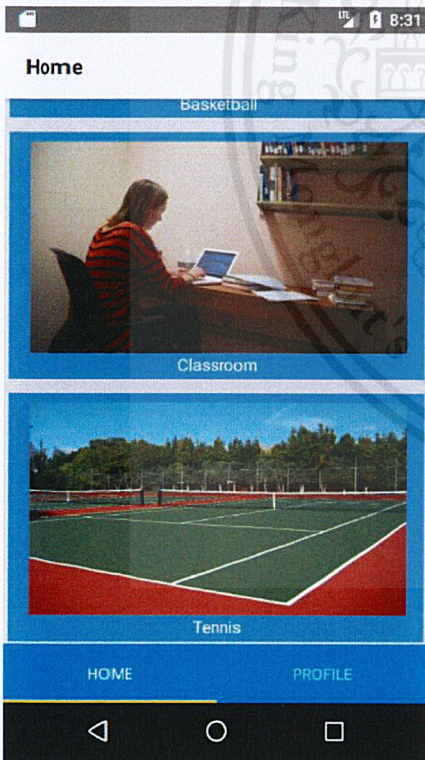


Figure 7.1.10 Home Tabbar Adding Facility

In this screenshot the classroom facility has been added to the facility/list api. Which will show the facilities that open for making reservation.

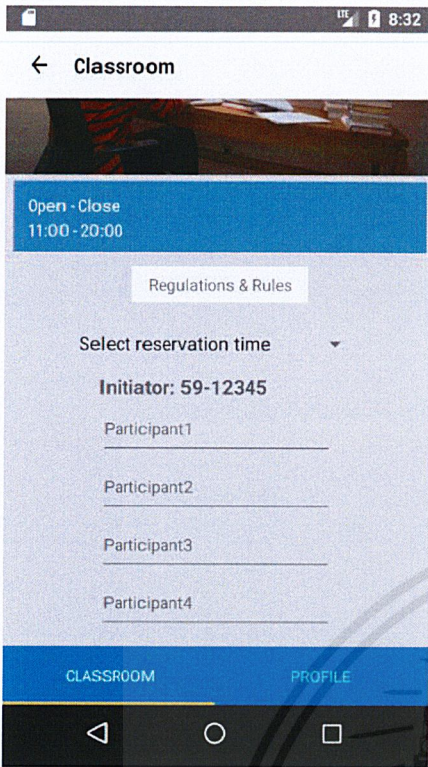


Figure 7.1.11 Classroom Facility Screen

Upon choosing newly added Classroom facility. The system will fetch the information of the facility from the server.

Id [PK] integer	facility_uuid text	image text	facility_name text	open_time text	close_time text	min_participants integer	max_participants integer	rules text	reserve_unit integer
1	1 229d3f63-5c4...	data:image...	Badminton Cou...	10:00	21:00	2	2	4 สวรรค...	30
2	2 33a8bd25-928...	data:image...	Tennis	10:00	21:00	3	3	4 เทนิส...	60
3	3 a55fb08c-c97...	data:image...	Basketball	10:00	21:00	3	3	8 บาสเกต...	60
4	4 a55fb08c-c97...	data:image...	Classroom	11:00	20:00	3	3	6 โป้...	30

Figure 7.1.12 facility/list API

In order to use this API user need to attach the token as the header but as for the application the device will manage and store the token as header. This API will return the data of facilities which the application will arrange and display as a Home Tabbar.

10	10	b45dc08c-c97...	1	11:00	13:00	4	4
11	11	b45dc08c-c97...	2	14:00	17:00	4	3
12	12	b45dc08c-c97...	3	18:00	20:00	4	4

Figure 7.1.12 Facilities Slot

The database of facilities slot will keep track of the time which separate as morning, afternoon, and evening respectively. The single digit number from the last two columns are max quota and a vacant space open for booking.

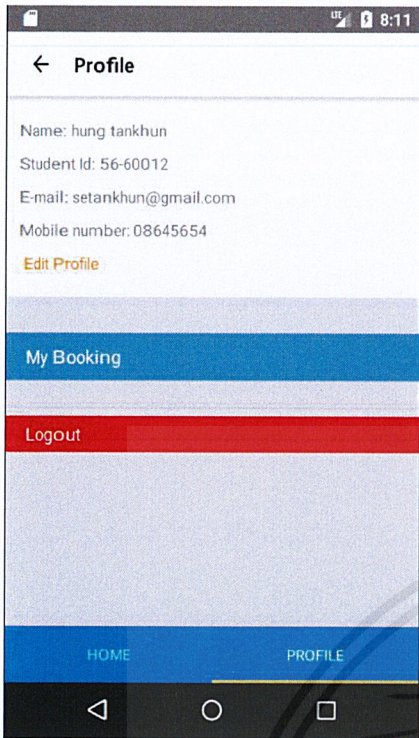


Figure 7.1.13 Profile Tabbar

In this screenshot show the info of the newly signup user that been signing up using mobile application.

8	8	66ad3ccd-cf...	password	user	tankhun12	hung tankh...	56-60012	setank...	08645654
---	---	----------------	----------	------	-----------	---------------	----------	-----------	----------

Figure 7.1.14 /user/list and /user/read/:user_uuid API

user/list API can be use to fetch all user in the database. But as for user/read with user_uuid as a header is the API which the mobile application use for getting particular user info. As for the response the application will arrange and display it in the Profile tabbar.

Evaluations and Discussions

8.1 Effectiveness

- User able to make reservation for the facility following the guidance of the application as follow upon have chose the facility:
 - read the rule and regulation abide by it.
 - choose available reservation time from the drop down or picker.
 - fill all the participants text input using other users email.upon successfully book user will recieve QR code use to claim the booking
- User able to claim the booking using the given QR code.
- User able to view the booking that is not expire by going to the “My booking” screen from Profile tabbar.
- In the main page is a scrollable view that expand according to the item in the list. Therefore, user can view all of it by scrolling down the table.
- User able to edit their profile by going to the profile tab bar and press edit profile in order to change their shown information.

8.2 Future Work

- In order to book the facility the initiator and all participants must all accept the invitation to book the facility within the time limit.
- the invitation and the booking success should've shown in the notification tab.
- Have interface for admin to add the facility and add functions of viewing the behavior of the users.
- Have a friend list in the interface in case user can't remember other users email or student id. So that user can select participants from the friend list.

Conclusion

9.1 Summary

In this project work has been separate into two part:

1. Front-end
2. Back-end

The purpose of this project is to develop an application that can use to book the university facility via our application. The application is crossed-platform which mean can be use on both operating system.

The main functionalities are as follow:

- Students and staffs of the require to sign up and logging in to the system in order to book the facility.
- User can choose any facility to book upon the shown facility in the home tab bar.
- User require to fill in all the participants according to the rule and regulation of the university and also abide by the rule and regulation other than inviting participants as well. As user have proceed through all the step correctly the reservation will be made and user will get the QR code in return.
- User able to view all the active reservation by going to profile tab bar and to “My Booking” section.
- User able to edit personal information from the profile tab bar.
- Upon user arrived at the facility user need to scan the QR code given by the system with the QR scanner provided for facility caretaker.

The pages that are available for user are as followed:

- Login screen
- Sign up screen
- Home tab bar
- Profile tab bar
- Facility screen
- Edit profile screen
- My booking screen
- Booking info screen

9.2 Lesson Learned

9.2.1 Time Management

Since the project must be done by the given time. Therefore, developer must use the time as given as efficient as possible since there are many things need to be done such as experiment and research. For that reason, the team comes up with a Gantt chart as project time schedule and to remind each teammate when the work should be done.

9.2.2 Learn to be Flexible

Flexible is also one of the most important since during the work not everything will going according to the project's plan and time schedule. So sometime something tends to be change and adapt according to the situation and time limit.

9.2.3 New Techniques Learned

Since the language use for development is new to the developer. Therefore, developer tend to do the research about the tools, libraries, project structure, and etc. In order to achieve the objective of this project as much as possible. The same also goes for the QR code generator and scanner as well.

9.2.4 UI/UX Design and Guidelines

Designing user experiences and interfaces are some things that is not taught in school. They might even be things that are unteachable. Learning the essence of designing UI/UX requires a lot of practice. Before all of this, we did not even know that there are guidelines to follow for each platforms.

9.3 Problem and Obstacles

9.3.1 Problem With Other Project's Tools

The tools to develop the project sometime encounter unknown error. Some of them can be solved and some of them cannot be solved or consuming too much time.

9.3.2 React Native State Management

As stated above, the development of crossed-platform application is new to the developer. Consequently, developer tend to research for the state management that suit to the project including various factors such as learning curve, time spending, and etc. At first the state management using was React-Native redux but upon doing and learning. We realized that redux have high learning curve and complex syntax comparing to Mobx. Therefore, the entire project structure need to be change from Redux to Mobx the same also goes for request and response library as well since the current one using Axios instead of the built in.

9.3.3 Different in Version

Development of React Native is a hard feat since React native need to be jointly works with other libraries and server as well. For instance, the development server for React Native for the project is Node.js since the Node.js version need to be compatible with React Native or else it would not work. The same also goes for the emulator and React native as well. Furthermore, some libraries required lower Android or Ios lower version but some also required higher version. Therefore, the project need to change the structure and libraries several times to get all the things in place.



References

- [1] Marcel. (2017). MobX. A Redux alternative you should consider. [Online]. Available: <https://hanno.co/blog/mobx-redux-alternative/>
- [2] Cervello. (2016). Salesforce Data Archiving Leveraging Lightning Connect & Heroku: Part III. [Online]. Available: <http://mycervello.com/blog/salesforce-data-archiving-leveraging-lightning-connect-heroku-3/>
- [3] นางสาวเกศชฎา ยิ้มแย้ม. (2016). ระบบจองห้อง TUTORING ในศูนย์การเรียนรู้กรมหลวงนราธิวาสราชนครินทร์. [Online]. Available: <http://203.131.219.167/km2559/2016/03/02/%E0%B8%A3%E0%B8%B0%E0%B8%9A%E0%B8%9A%E0%B8%88%E0%B8%AD%E0%B8%87%E0%B8%AB%E0%B9%89%E0%B8%AD%E0%B8%87-tutoring-%E0%B9%83%E0%B8%99%E0%B8%A8%E0%B8%B9%E0%B8%99%E0%B8%A2%E0%B9%8C%E0%B8%81%E0%B8%B2%E0%B8%A3/>
- [4] B.K. Hong. (2017). Eatigo- Dining Deals and Online Restaurant Reservation for your "App"etite. [Online]. Available: <http://www.itsberyllicious.com/2017/08/eatigo-dining-deals-and-online.html>
- [5] S. Gonzaga. (2017). Skedda - Features, Pricing, Alternatives, and More. [Online]. Available: <https://zapier.com/apps/skedda>