KMITL Bike Extension System

Nitravee Wongvitavas
Patipon Riebpradit
Pisit Duangdeekamoltat

Bachelor of Engineering in Software Engineering
International College
King Mongkut's Institute of Technology Ladkrabang
Academic Year 2017
KMITL-2018-IC-B-003-001

Thesis — Academic Year 2017

Bachelor of Engineering in Software Engineering

International College

King Mongkut's Institute of Technology Ladkrabang

Title: KMITL Bike Extension System

Authors:

1. Ms. Nitravee Wongvitavas Student ID 57090013
2. Mr. Patipon Riebpradit Student ID 57090016
3. Mr. Pisit Duangdeekamoltat Student ID 57090023

Approved for submission

(Dr. Isara Anantavrasilp)

Project Adviser

Date 8 / 6 / 2018

Abstract

KMITL Bike is a project which provide free bicycles for KMITL students and faculty members to use within the campus area using bicycle sharing platform. KMITL Bike is a station-less system where user can borrow and return the bike anywhere within the campus area. The project was launch in 2016 as part of KMITL Green Campus project, however there are still some problems and inconvenience issues exist in the system.

The intention of this project is to eliminate the problems exist in the KMITL Bike system and to provide more convenient ways for system administrators and the bike maintenance team to manage the system. The project contain administration application, geofencing usage area, and real-time monitoring system to prevent users from improper action using bikes.

The administration application should allow administrators to monitor the status of each bike closely and provide direct contact to the user. The geofencing area feature allows system administrator to setup the permitted usage area, the application will warn user when user get out of the area boundary and notify to admin via administration application.

Acknowledgements

Firstly, we would like to thank Dr. Isara Anantavrasilp, our adviser, for giving us a great opportunity to work in a real-world project and providing us with knowledge and experiences along with inspiration to be use our project. We would also like to thank Asst. Prof. Dr. Ronnachai Tiyarattanachai for his consolation in using different techniques to improve our system and showing us some of the existed work he found that helps us to shape the idea for the project. Another big thanks to Dr. Natthapong Jungteerapanich for providing us with a 3D printer and other 3D printing-related equipment. A thank to Kajornsak Peerapathananont who gave a great contribution in developing the iOS application and guidance throughout the project period. Furthermore, a thank to the InfiniLock team who created and developed the KMITL Bike project and providing us with necessary information and advises on our project, namely Ms. Chatchaya Chanchua, Mr. Pattarawut Imamnuaysup, and Mr. Sasawat Chanate. Lastly, we would like to thank our classmates, friends, and seniors who gave us suggestions and ideas so that helped improve our project during the development phase.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Transportation between faculties has been one of the concerned issues in the university campus. Motorized vehicles such as cars and motorcycles are the common ways students and university staffs choose out of this problem. However, these motorized vehicles are the greatest contributor to air pollution [1]. In 2013, automobile contributed more than half of the carbon monoxide and nitrogen oxides, and almost a quarter of the hydrocarbons emitted into the atmosphere. These gases may cause negative effect include irritation to the eyes, nose and throat, and can lead to serious disease such as lung cancer and heart disease as well [2].

To discourage the use of motorized vehicle, a team from International College of KMITL introduced a project called KMITL Bike. It is a community-based bicycle sharing system built to provide an alternative mode of transportation to students and faculty members of KMITL [4]. KMITL Bike allows user to borrow a bicycle from one place and return it in another place for a short period of time. The project was started in 2015 as part of KMITL Green Campus initiative, whose goal is to improve campus environment as a whole.

1.2 Problem Statement

KMITL Bike project provides free bike rental for students and faculty members in King Mongkut's Institute of Technology Ladkrabang. KMITL Bike allows users to unlock bicycles via mobile application which will connect to the electronic lock attached to the bicycle. However, several problems occurred after the implementations of the last team causing the system to be put out-of-service. Some of the problems include:

- Hardware defects and poorly assembled system hardware unit of the electronic locks
- Several system critical bugs that prohibits the unlocking process of the bicycles
- System does not proactively limit usage area
- Lacking a proper administration user interface
- UI is accidentally similar to competitor's application
- Deprecation of some core libraries
- Codebase that does not conform to proper design pattern techniques causing difficulties in the troubleshooting process and addition of new features

With such issues, a major revamp of the whole system is required to restart KMITL Bike service.

1.3 Objectives

The objectives of this project is to improve and extend the current KMITL Bike system as follows:

- Creating an extension feature that enables limitation of usage area.
- Creating a unique, clean, and intuitive UI/UX that further simplifies usage while enhancing user's experience.
- Creating a back-end page for system administrators and bike maintenance team.
- Creating a system that warns users and notify administrators in real-time when users make an improper action.
- Fixing the electronic locks to allow usage of the service.
- Revamping the codebase that conforms with industry standards coding techniques focusing on extensibility and maintainability of the system.

The rest of the thesis is organized as follows. Chapter 2 provides a review of related works. Requirements analysis of the system is described in Chapter 3. Chapter 4 describes background knowledge required to develop the system, followed by system designed in Chapter 5. Chapter 6 explains the development process along with the screen-shots of the application and back-end system. We also show the differences and improvements between previous and current versions of the system. Chapter 7 describes the results and evaluation of the system and it's usage. Finally, Chapter 8 concludes the thesis.

Chapter 2

Literature Review

This chapter presents the current existing products in the market and their comparison.

2.1 Bicycle Sharing System

This section presents some of the current existing bicycle sharing systems in the market which adopted the idea of station-less parking system.

2.1.1 SG Bike

SG Bike, a bicycle sharing company established in Singapore. To use SG Bike, user can locate the nearest bike via mobile application and scan QR Code on the bike to unlock it. User may also enter bike ID found on the bike to the application, as an alternative way to unlock the bike.

2.1.2 ofo

ofo is a Beijing-based company found in 2014. It is one of the world's leading bicycle-sharing system with more than 8 million bicycles in seven countries around the world [7].

The authentication method that are being used in ofo is in a form of combination lock. Users have to either input identification number or scan QR code of the target bicycle into its mobile application. Then users will receive combination numbers to unlock their bikes.

2.1.3 Mobike

A Chinese-based company found in 2015. In the past year, Mobike became one of the leading bicycle on-demand company with over 100 million registered users across 100 cities [10].

Mobike locates its bicycles using GPS chips built into their locks. After locating the bike, users have to scan QR code on the bike in order to unlock it. Users can also reserve their desired bike by select the bike and press "Make a Reservation" button on the map, the bike can be reserve up to 15 minutes per reservation.

2.1.4 oBike

One of the biggest player in the bicycle sharing market is oBike, the system was launched in Singapore in early 2017 and expanded into 11 countries [9].

User can located the available bike via the mobile application. User may also reserve the bike by tapping the bicycle icon shown in the map. The reservation will be cancel if user unlock other bike or 10 minutes after reservation. As same as Mobike, oBike authentication system let user scan QR Code on the bike to unlock it.

2.2 Geofencing

The emergence of station-less bicycle sharing system raises the concerns for parking these vehicles. Parking these bikes in an inappropriate area may create problems such as:

- Occupying parking space for motor scooter
- Blocking entrance and clogging pavement
- Parking bike in private property
- Littering public area (e.g. river, tree, under the bridge)
- Bike getting damage, stolen, or savage

Hence, to avoided these problems, many bicycle sharing firms have come up with geofencing parking policies, to ensure that users park their bicycle in the permitted area and to keep those bicycles available for other users. This section presents some of geofencing features implemented by each bike sharing company.

2.2.1 SG Bike

SG Bike is the first in the market to integrate geofencing in its system [6]. The feature enforced users to park bicycles in provided areas around yellow box-shaped devices called Geostations.

Geostation is a device which transmit the data to check whether the bike is parked in the provided parking zone. If the bike is not parked in the area at the end of use, the application will notify user to re-park the bike. Failure to comply will result in fine deducted from user's registered e-wallet [6].

2.2.2 ofo

In late 2017, the company introduce its new feature, geofencing, which encouraged users to park bicycles in the permitted parking area and to keep bicycles in the most needed place. User will receive phone call if the bike is parked out of the provided area for more than 24 hours [11]. The feature was first tested in Oxford, England and will be extend in to other countries in the future [8].

2.2.3 Mobike

As well as other bike sharing system, Mobike partnered with Manchester City council, launch its recommended parking area policy to encourage user to park the bike in the recommended parking area. The policy is aimed to providing bikes to areas of high demand and help operation team on bike redistribution operation.

The recommended parking zone is developed based on GPS data gathered from the bike and is indicated by red-line marking on the map within Mobike application [12]. In the current phase, the user will not be docked credit for parking out side the recommended area shown, however, will received a warning through the application.

2.2.4 oBike

Announcing oBike credit feature in early 2017, oBike aims to encourage its rider to have "positive behavior". The higher the point, the cheaper the ride. User can achieved oBike point by having "positive behavior" such as report broken bike and illegal park as well as encourage frequent ride. On the other hands, the point will be deduct when user performs "negative behavior" such as forgetting to lock the bike and not parking in permitted area. The permitted parking location is indicated by parking mark "P" in the oBike application maps [13].

2.3 Remote Control Bicycle Brake

As KMITL Bike system operates in a close service area, having a real-time physical action on bicycles help notify users that they are getting out of the permitted area. A brake will be attached to the rear wheel of a bicycle and apply force to slow down the bicycle automatically when it gets out of the usage area.

This section presents the existing works of remote control bicycle brakes.

2.3.1 MiniBrake

MiniBrake is a remote controlled bicycle brake designed for parental control. The product consists of two parts: a remote controller and a small brake attached to the rear wheel of the bike. The device is activate by pressing a button on its controller, the brake applied by putting pressure on to the wheel to slow down the bike without falling down [15]. The device works within 50 meters range and will activate automatically if the bike is out of range.

Figure 2-1: MiniBrake

2.3.2 iBrake

Similar to MiniBrake, iBrake is designed for kids safety during bicycle ride. The device contain two parts: iBrake which contain the brake mechanism to apply force to the rear wheel and the iBrick which contain the battery and its controller. The device is control using the mobile application via personal hotspot and can be use up to 100 meters range [14].

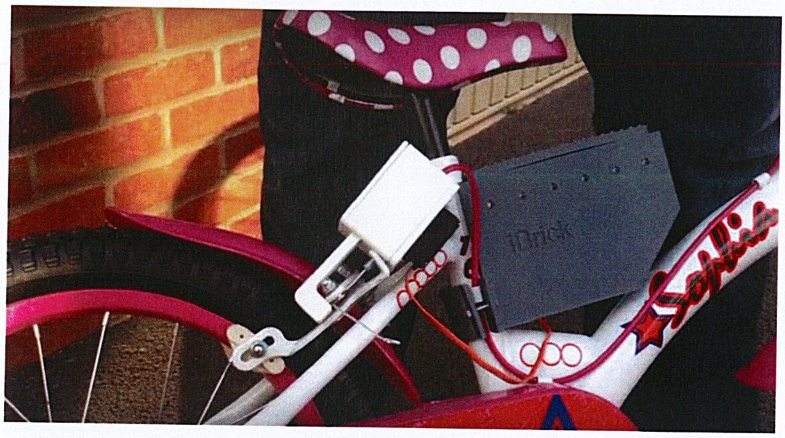

Figure 2-2: Brake

2.4 Comparison

Below is the comparison of each work in many major aspects.

Table 2.1: Comparison between SG Bike, ofo, Mobike, and oBike

Name	Authentication Method	Reserve Bike	Deposit Fee	Payment Method
SG Bike	Scan QR code / Enter bike's id to the app	No	No	E-wallet
ofo	Combination lock	No	No	Alipay / Credit and Debit cards
Mobike	Scan QR code	15 minutes	Yes	Credit and Debit cards
oBike	Scan QR code	10 minutes	Yes	Paypal / Credit and Debit cards

Table 2.2: Comparison between SG Bike, ofo, Mobike, and oBike in geofencing feature aspect

Name	**Phase**	**Technology**	**Operate Area**	**Penalty**
SG Bike	Implemented	Geostation device	Singapore	Fine through user's e-wallet
ofo	Testing	GPS	Oxford, England	Phone call
Mobike	Testing	GPS	Manchester City, England	Receive warning
oBike	Pilot	GPS	Singapore	Deduct user's credits

Table 2.3: Comparison between MiniBrake and iBrake

Name	**Target User**	**Maximum Range**	**Installed Position**	**Controller**
MiniBrake	Parent	50 meters	Rear wheel	Remote controller
iBrake	Parent	100 meters	Rear wheel	Mobile application

Chapter 3

Requirements Analysis

This chapter provides further details on the project requirements along with use case diagrams and descriptions. The requirements are represented in a form of FURPS+ model (Table 3.1) which identifies both functional and non-functional requirements of the project.

3.1 Requirements

Table 3.1: Table of Requirements

No	Requirement	Type
1	System administrator shall be able to login to the application.	Functional
2	System administrator shall be able to logout from the application.	Functional
3	System administrator shall be able to located active in the system.	Functional
4	System administrator shall be able to located inactive in the system.	Functional
5	System administrator shall be able to see the status of each bike.	Functional

Table 3.1 – continued from the previous page

No	Requirement	Type
6	System administrator shall be able to set availability status of each bike.	Functional
7	System administrator shall be able to see each user's information.	Functional
8	System administrator shall be able to ban users.	Functional
9	System administrator shall be able to unban users.	Functional
10	System administrator shall be able to see amount of available and unavailable bikes in the system.	Functional
11	System administrator shall be able to contact every user separately via chat function.	Functional
12	System administrator shall be able to see history usage of each bike.	Functional
13	System administrator shall be able to see history usage of each user.	Functional
14	System shall warn user if the bike get out of the boundary area via mobile application.	Functional
15	System shall take immediate action to the bike if user infringe terms and conditions.	Functional
16	System shall notify system administrator if the bike get out of the boundary area.	Functional
17	System shall be able to support at least 100 users.	Performance
18	System chat function shall update in real-time.	Reliability
19	The bike location shown in the application shall be the latest location within 1 minute.	Reliability
20	The bike location shown in the application shall contain no more than 3 meters error.	Reliability
21	The bike administration application shall be responsive.	Usability

Table 3.1 – continued from the previous page

No	Requirement	Type
22	The bike administration application shall runs on Android and iOS platform.	Supportability

3.2 Use Case Diagram

Figure 3-1 presents the use case diagram of KMITL Bike system where the white use case present the existed use cases in the previous version (2.0) and yellow use cases are addition use cases implemented.

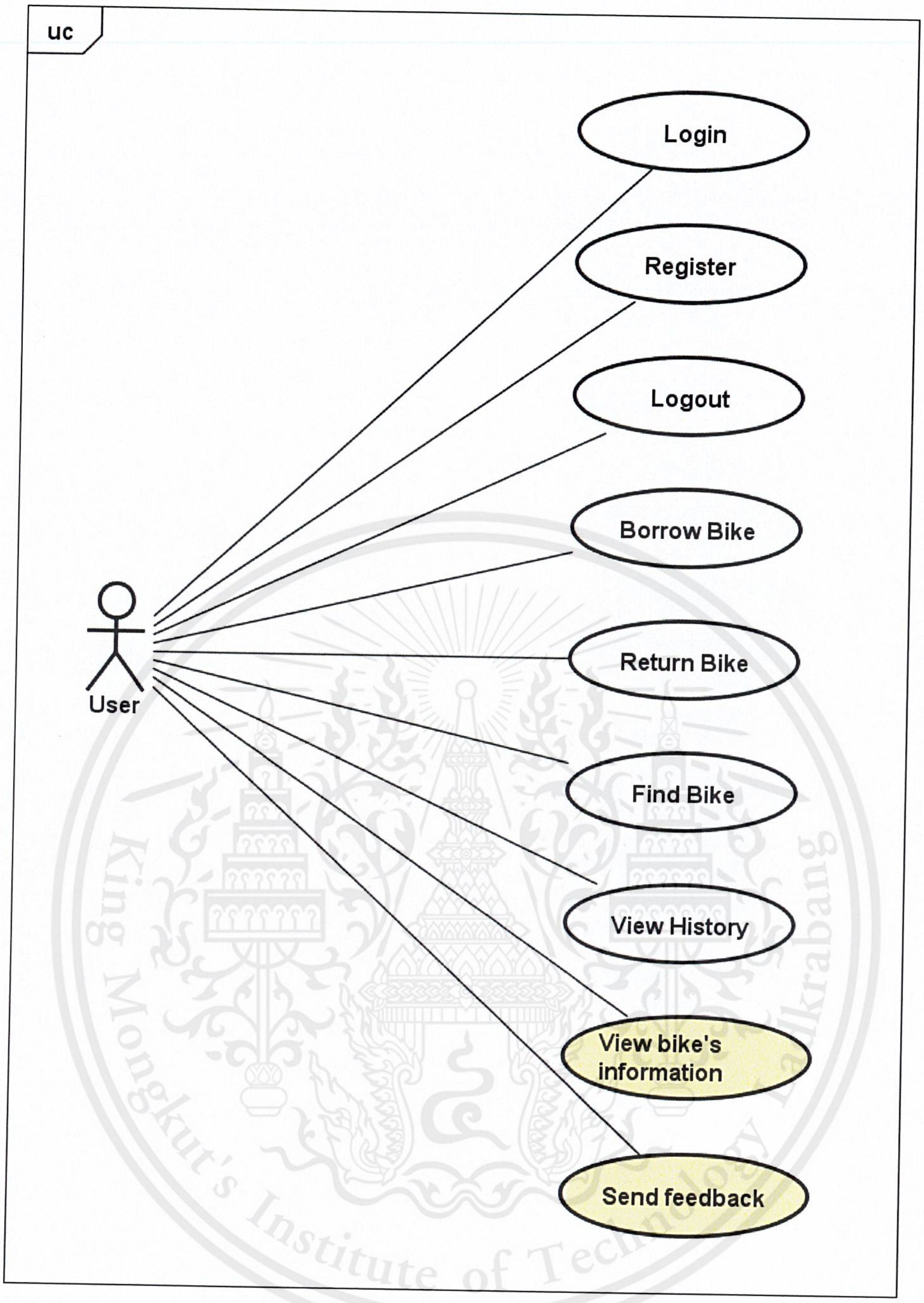

Figure 3-1: Use Case Diagram of KMITL Bike Application

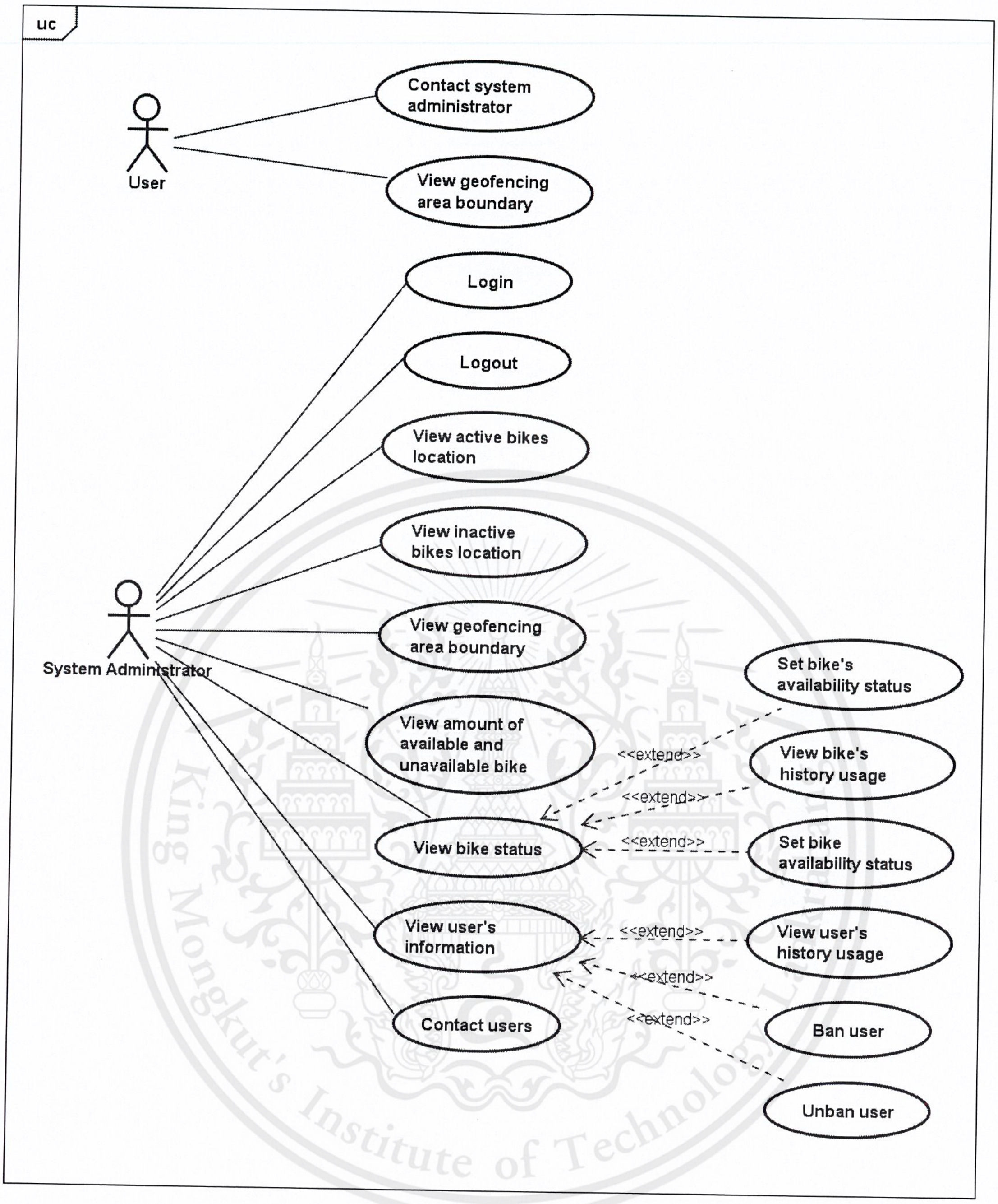

Figure 3-2: Use Case Diagram of KMITL Admin Bike Application

3.3 Use Case Description

Table 3.2: Use Case Description - Login

Use Case	**Login**
Primary Actor	System Administrator
Pre-condition	System admin has the application on their device.
Post-condition	System admin is logged in to the system.
Flow of events	1) System admin opens the application
	2) System admin enters their username and password
	3) Server verifies system admin's information
	4) System admin is logged in to the system
Alternative Flow	**Condition: Unrecognized username/password**
	3a) System notify system admin that the username/password is unrecognized
	Condition: Unable to login
	4a) System notify the user and asks user to login again

Table 3.3: Use Case Description - Logout

Use Case	Logout
Primary Actor	System Administrator
Pre-condition	System admin has the application on their device. System admin is logged in.
Post-condition	System admin is logged out from the system.
Flow of events	1) System admin opens the application
	2) System admin presses to "Logout" button on the application
	3) Application performs logout for system admin
	4) System admin is logged out from the system
Alternative Flow	**Condition: Unable to logout**
	4a) System notify the user and asks system admin to logout again

Table 3.4: Use Case Description - View Active Bikes Location

Use Case	View Active Bikes Location
Primary Actor	System Administrator
Pre-condition	System admin is already logged in to the system through the application.
Post-condition	-
Flow of events	1) System admin opens the application
	2) Application places active bicycle "markers" on its map

Table 3.5: Use Case Description - View Inactive Bikes Location

Use Case	View Inactive Bikes Location
Primary Actor	System Administrator
Pre-condition	System admin is already logged in to the system through the application.
Post-condition	-
Flow of events	1) System admin opens the application
	2) System admin select "Inactive Bike" filter button
	3) Application places inactive bicycle "markers" on its map

Table 3.6: Use Case Description - View Geofencing Area Boundary

Use Case	View Geofencing Area Boundary
Primary Actor	System Administrator
Pre-condition	System admin is already logged in to the system through the application.
Post-condition	-
Flow of events	1) System admin opens the application
	2) System admin select "Geofencing" menu
	3) Application displays geofencing area boundary indicated by red line on its map

Table 3.7: Use Case Description - View Amount of Available and Unavailable bikes

Use Case	View Amount of Available and Unavailable bikes
Primary Actor	System Administrator
Pre-condition	System admin is already logged in to the system through the application.
Post-condition	-
Flow of events	1) System admin opens the application
	2) System admin select "Bike" tab in the application
	3) Application displays number of available and unavailable bike and list all bicycle in the system

Table 3.8: Use Case Description - View Bike Status

Use Case	View Bike Status
Primary Actor	System Administrator
Pre-condition	System admin is already logged in to the system through the application.
Post-condition	-
Flow of events	1) System admin opens the application
	2) System admin select "Bike" tab in the application
	3) Application lists all bikes and its status in the system

Table 3.9: Use Case Description - View Bike's History Usage

Use Case	**View Bike's History Usage**
Primary Actor	System Administrator
Pre-condition	System admin is already logged in to the system through the application. System admin view bike status.
Post-condition	-
Flow of events	1) System admin select interested bike from the list displayed
	2) Application displays an in-detail view of the interested bike past usage

Table 3.10: Use Case Description - Set Bike's Availability Status

Use Case	**Set Bike's Availability Status**
Primary Actor	System Administrator
Pre-condition	System admin is already logged in to the system through the application. System admin view bike status.
Post-condition	The selected bike availability status is set
Flow of events	1) System admin select interested bike from the list displayed
	2) Application displays an in-detail view of the interested bike past usage
	3) System admin select "Available" toggle button in the bike information page
	4) System set the selected bike's availability status
	5) Application displays updated bike's availability status

Table 3.11: Use Case Description - View User's Information

Use Case	**View User's Information**
Primary Actor	System Administrator
Pre-condition	System admin is already logged in to the system through the application.
Post-condition	-
Flow of events	1) System admin opens the application
	2) System admin select "Users" tab in the application
	3) Application lists all users in the system
	4) System admin select interested user
	5) Application displays an in-detail view of the interested user

Table 3.12: Use Case Description - View User's History Usage

Use Case	**View User's History Usage**
Primary Actor	System Administrator
Pre-condition	System admin is already logged in to the system through the application. System admin view user's information
Post-condition	-
Flow of events	1) System admin select "History" button in the user's information page
	2) Application displays an in-detail view of the interested user past usage
Alternative Flow	**Condition: No history usage**
	2a) Application displays "No history usage"

Table 3.13: Use Case Description - Ban User

Use Case	Ban User
Primary Actor	System Administrator
Pre-condition	System admin is already logged in to the system through the application. System admin view user's information. The selected user status is "Not Ban"
Post-condition	System ban the selected user
Flow of events	1) System admin select "Ban User" button in the user's information page
	2) System set selected user status as "Ban User"
	3) The selected user is ban

Table 3.14: Use Case Description - Unban User

Use Case	Unban User
Primary Actor	System Administrator
Pre-condition	System admin is already logged in to the system through the application. System admin view user's information. Selected user's status is "Banned".
Post-condition	System unban the selected user
Flow of events	1) System admin select "Unban User" button in the user's information page
	2) System set selected user status as "Ban User"
	3) The selected user is unban

Table 3.15: Use Case Description - Contact User

Use Case	Conatct User
Primary Actor	System Administrator
Pre-condition	System admin is already logged in to the system through the application.
Post-condition	System administrator communicate with user
Flow of events	1) System admin opens the application
	2) System admin select "Contact" tab in the application
	3) Application all the issue report by user
	4) System admin select interested issue
	5) Application navigate to chat page with the issue user

Table 3.16: Use Case Description - Contact System Administrator

Use Case	Contact System Administrator
Primary Actor	All Users
Pre-condition	User is already logged in to the system through the application.
Post-condition	User communicate with system administrator
Flow of events	1) User opens the application
	2) User select "Contact" tab in the application
	3) Application navigate to chat page

Table 3.17: Use Case Description - Set Working Hours

Use Case	**Set Working Hours**
Primary Actor	System Administrator
Pre-condition	System admin is already logged in to the system through the application.
Post-condition	User is allowed to borrow the bike within working hours only
Flow of events	1) Admin opens the application
	2) Admin set date and time for each day

Table 3.18: Use Case Description - Set Holidays

Use Case	**Set Holidays**
Primary Actor	System Administrator
Pre-condition	System admin is already logged in to the system through the application.
Post-condition	User is not allowed to borrow the bike on the given day
Flow of events	1) Admin opens the application
	2) Admin set special unworking day

Chapter 4

Background Knowledge

4.1 Functional Programming

Functional programming languages are specially designed to handle symbolic computation and list processing applications. It is based on the concept of mathematical functions which use conditional expressions and recursion to perform computation [19]. Functional programming give several advantages over object-oriented programming as shown in Table 4.1

Table 4.1: Function programming vs. Object-oriented programming

Functional Programming	**Object-Oriented Programming**
Uses immutable data	Uses mutable data
Follows declarative programming model	Follows imperative programming model
Supports parallel programming	Not suitable for parallel programming
Flow control is done using function calls and function calls with recursion	Flow control is done using loops and conditional statements
Supports both "Abstraction over Data" and "Abstraction over Behavior"	Supports only "Abstraction over Data"

4.2 Bluetooth Low Energy Module (BLE)

Bluetooth low energy (BLE) or sometimes know as Bluetooth Smart is a light-weight subset of classic Bluetooth and was introduced as part of the Bluetooth 4.0 core specification [16]. It is usually powered by battery or USB. BLE module use the Generic Attributes (GATT) as a hierarchical data structure to transfers. GATT is built on top of the Attribute Protocol (ATT) which is a generic data protocol. By using GATT protocol, the connections are exclusive which means that each BLE peripheral can only be connected to one central hub such as mobile phone or tablet [17]. GATT transactions in BLE are based on high-level, nested objects called Profiles, Services and Characteristics, which can be seen in the Figure 4-1.

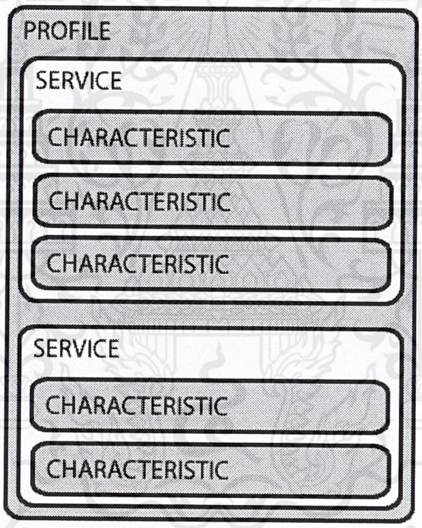

Figure 4-1: GATT Profile

A profile doesn't actually exist on the BLE peripheral itself, it's simple a pre-defined collection of Services that has been compiled by either the Bluetooth SIG or by the peripheral designers [17].

4.3 Servo Motor

Servo motors are widely used in many applications especially in robotic fields. They provide rotational movement controlled precisely in angular degrees. For example, they are used in in-line manufacturing to move belts and machines.

Inside of a servo motor consists of three main components which are potentiometer, DC-motor, and control circuit. The potentiometer is a three-terminal adjustable resistor that sends signal to control circuit, so it knows the direction and degrees as the motor moves.

Electrical pulse of changeable width is required to be sent to the motor to control a servo motor. Typically, a Pulse Width Modulator (PWM) is used to control a servo motor. Ordinary servo motor can be rotated only 180 degrees but the neutral position is at 90 degrees. When PWM is sent to the motor, it will move to desired angle and hold that position even if there are some force trying to change its position; the servo will resist that force to remain in the same position. However, different types of servo motor can resist variable level of force depending on its torque rating.

Chapter 5

System Design

This chapter provides an overview of the KMITL Bike system design including the KMITL Bike application and System Administration application. The design illustrated the system ranging from the basis for implementation to the flow of the system functions.

5.1 System Architecture

5.1.1 Server

In the server side, there are few modification made to the existed version implemented by the previous team. Figure 5-1 illustrated overview and changes made on the server where white modules represent the existed class, yellow modules serve as unused class, and blue module represents the implemented class.

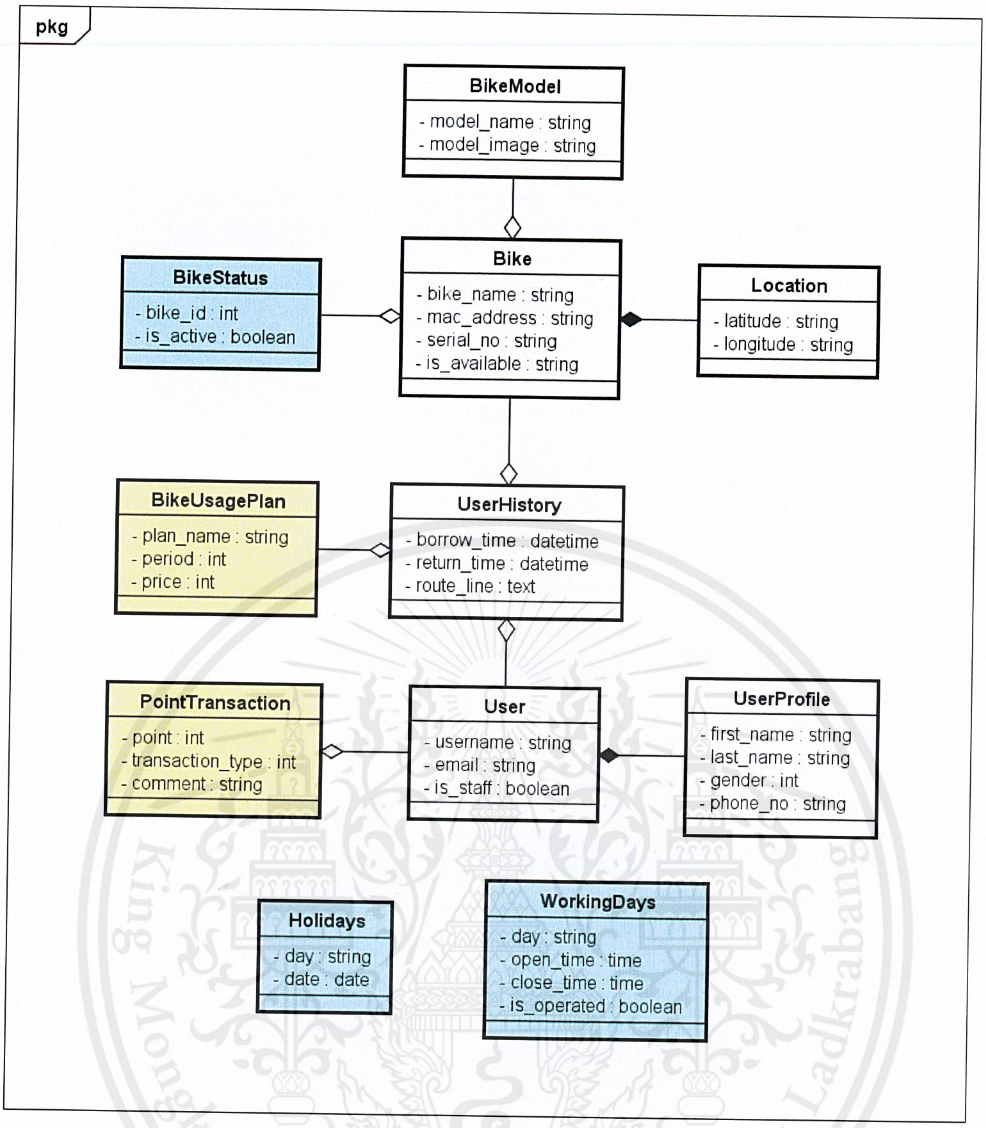

Figure 5-1: Overview of Server Class Diagram

5.1.2 KMITL Bike API

There are several services available on the server for the mobile application to use. However, modification of existing APIs and creation of new APIs are implemented for administrator to maintain the system more efficiently. Table 5.1 represents the existed APIs implemented by the previous team and Table 5.2 represents APIs implemented in this research.

Table 5.1: Existed KMITL Bike REST API

Template	**HTTP Verb**	**Description**
api/v1/auth/access_token	GET	Return credentials for a login session and identifies the user
api/v1/auth/login	POST	Login into the server, this will result a call to I AM KMITL account validation
api/v1/auth/logout	GET	Logout from the server
api/v1/auth/register	POST	Register a new account
api/v1/services/available	GET	Return a list of bikes that are currently available
api/v1/services/update_-bike_ location	POST	Update the current location of bike to the server
api/v1/user/status	GET	Return user's status whether he/she is still in riding session or not
api/v1/user/borrow	POST	Borrow the bike by bike ID
api/v1/user/return	POST	Return the bike
api/v1/user/history	GET	Return user's riding history
api/v1/user/update_user_-location	POST	Update the current location of user, this will be invoked while the user is riding

Table 5.2: Implemented KMITL Bike REST API

Template	HTTP Verb	Description
api/v1/users/<user_id> /profile	GET	Return user's profile
api/v1/users/<user_id> /histories/list	GET	Return list of user's riding histories
api/v1/users/<user_id> /points/list	GET	Return user's point histories
api/v1/bikes/current_used	GET	Return bike that are currently used
api/v1/versions/check	GET	Return version of the application

5.1.3 KMITL Bike 3.0 Application

KMITL Bike 2.0 application created in the previous implementation was not well-designed. It was not designed with a software architecture at its foundation. On top of that, the libraries and dependencies in used are from third-party which are deprecated and do not provide flexibility in customizing different parameters. These problems lead to difficulties in the troubleshooting process and addition of new features. Hence, revamping of the existing application (KMITL Bike 2.0) is required. The revamped version will be created from scratch using native languages and frameworks of each platform (e.g. Java for Android and Swift for iOS) to avoid further complication with libraries and dependencies.

Android Application

The latest Android application is being developed using MVP (Model-View-Presenter) as its architecture. To follow the the design principle, API client and data manager act as a model and activity modules serve as view. Hence, each view came with their own presenter that will handle all presentation logic and bridge between model and view. Figure 5-2 illustrates the overview of KMITL Bike 3.0 Android architecture.

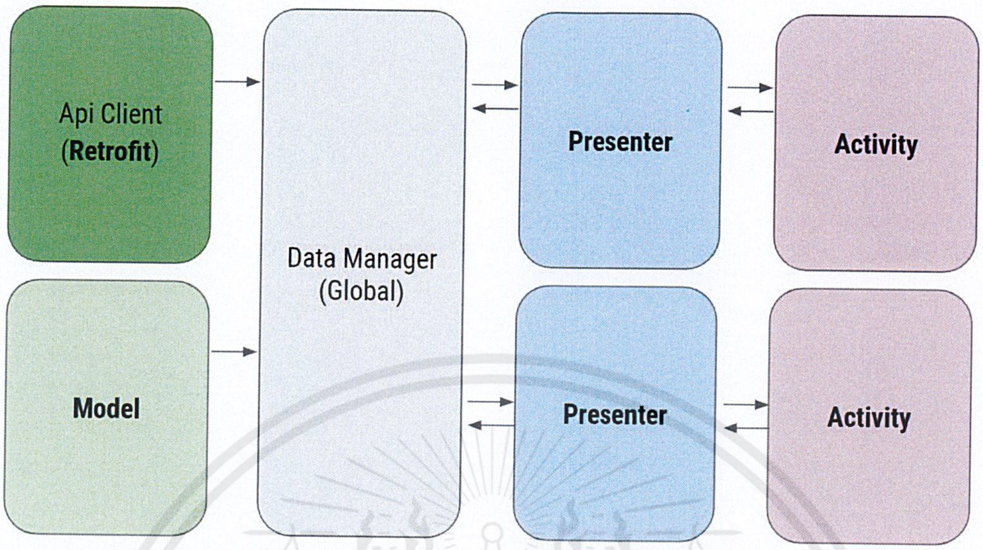

Figure 5-2: KMITL Bike 3.0 Android Architecture

Figure 5-3 depicts a class diagram of KMITL Bike 3.0 Android application.

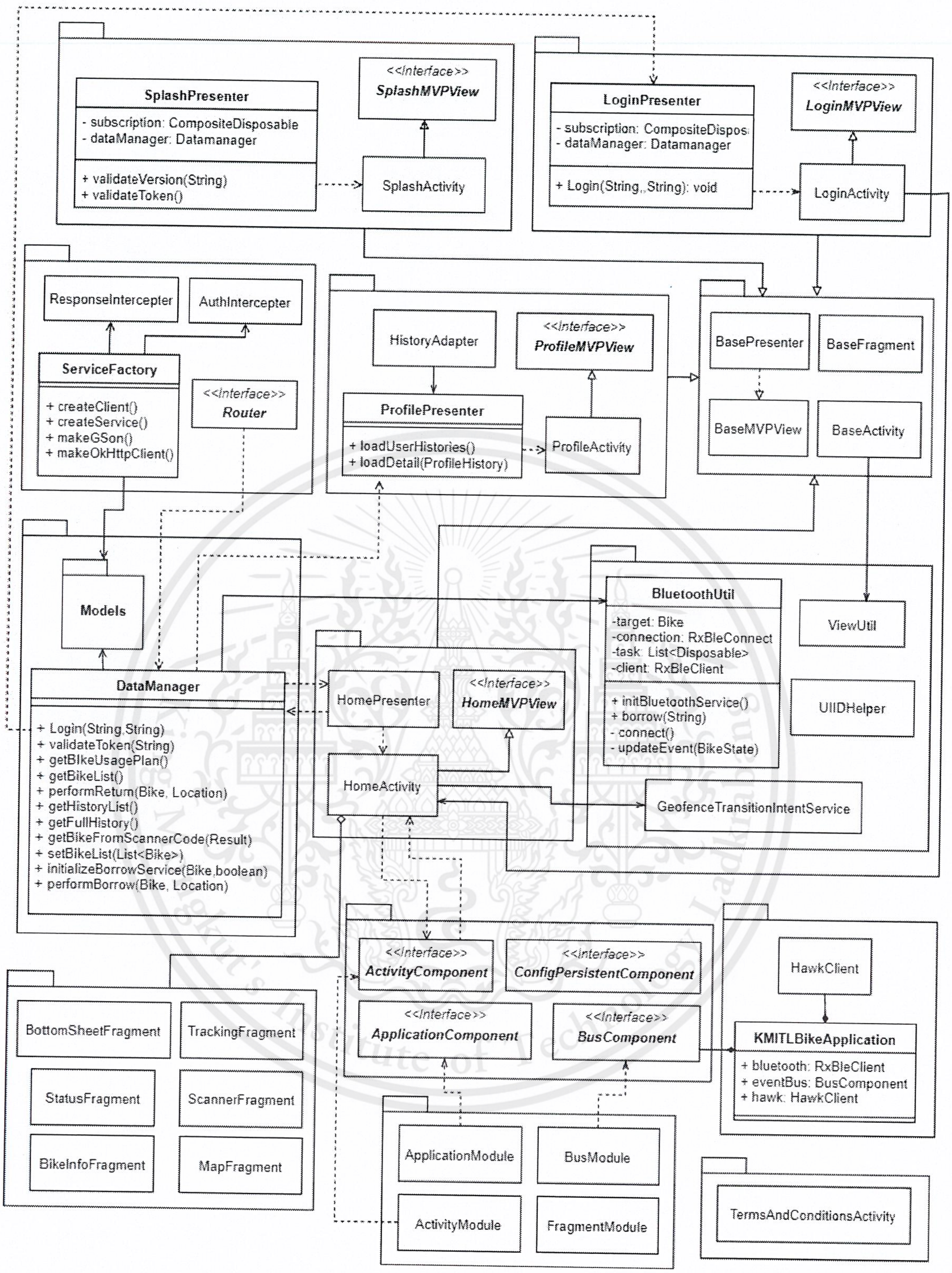

Figure 5-3: Class diagram of KMITL Bike 3.0 Android Application

iOS Application

The latest iOS application is being developed using MVVM (Model-View-ViewModel) as its architecture. The architecture intent is to provide a clean separation of concerns between the user interface controls and their logic . Figure 5-4 illustrates the overview of KMITL Bike 3.0 iOS architecture.

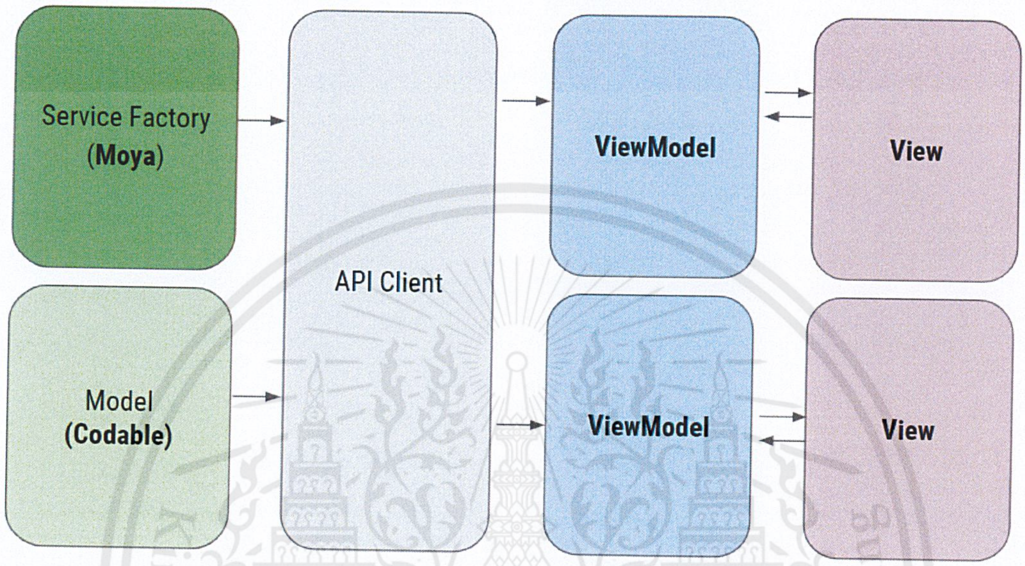

Figure 5-4: KMITL Bike 3.0 iOS Architecture

Figure 5-5 depicts a class diagram of KMITL Bike 3.0 iOS application. API services are used for any request from application to the server.

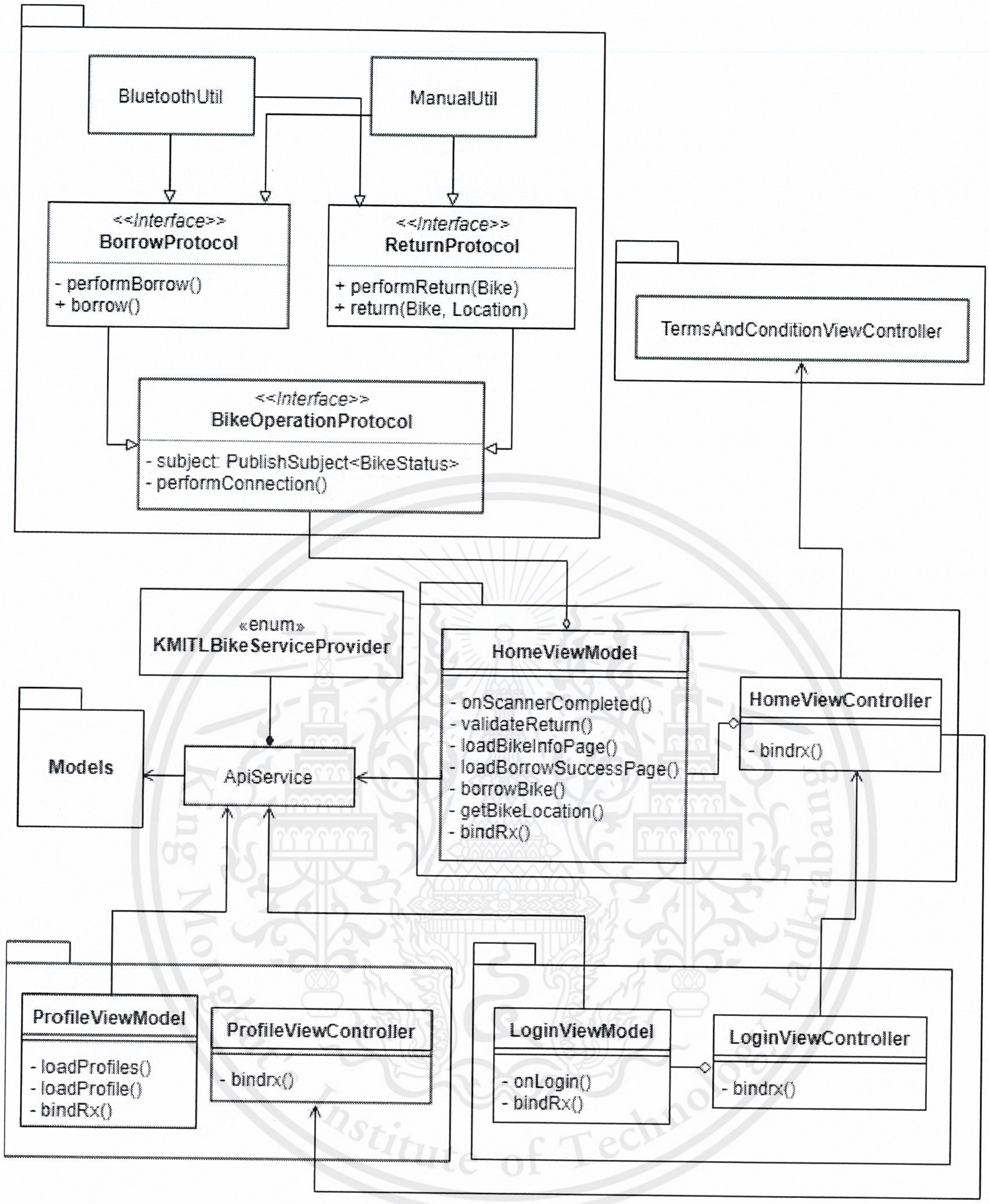

Figure 5-5: Class diagram of KMITL Bike 3.0 iOS Application

5.2 System Flow

As the system uses location service from user's mobile device, the user is asked to scan QR code attached on the bicycle to verify that both the user and the bicycle are in the same location. Figure 5-6 and Figure 5-7 illustrates the flow of borrowing and returning a bike in the system.

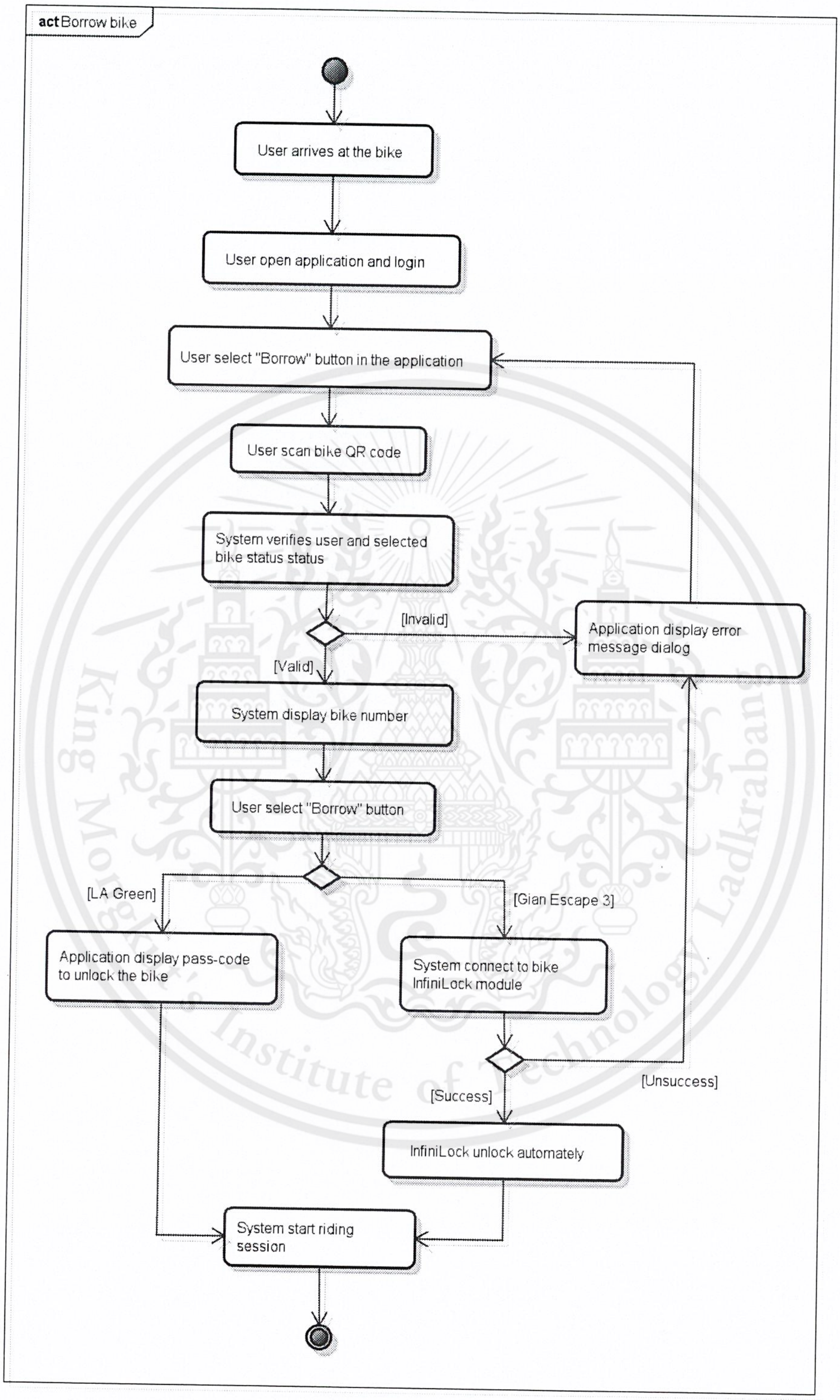

Figure 5-6: Flow of borrowing a bike

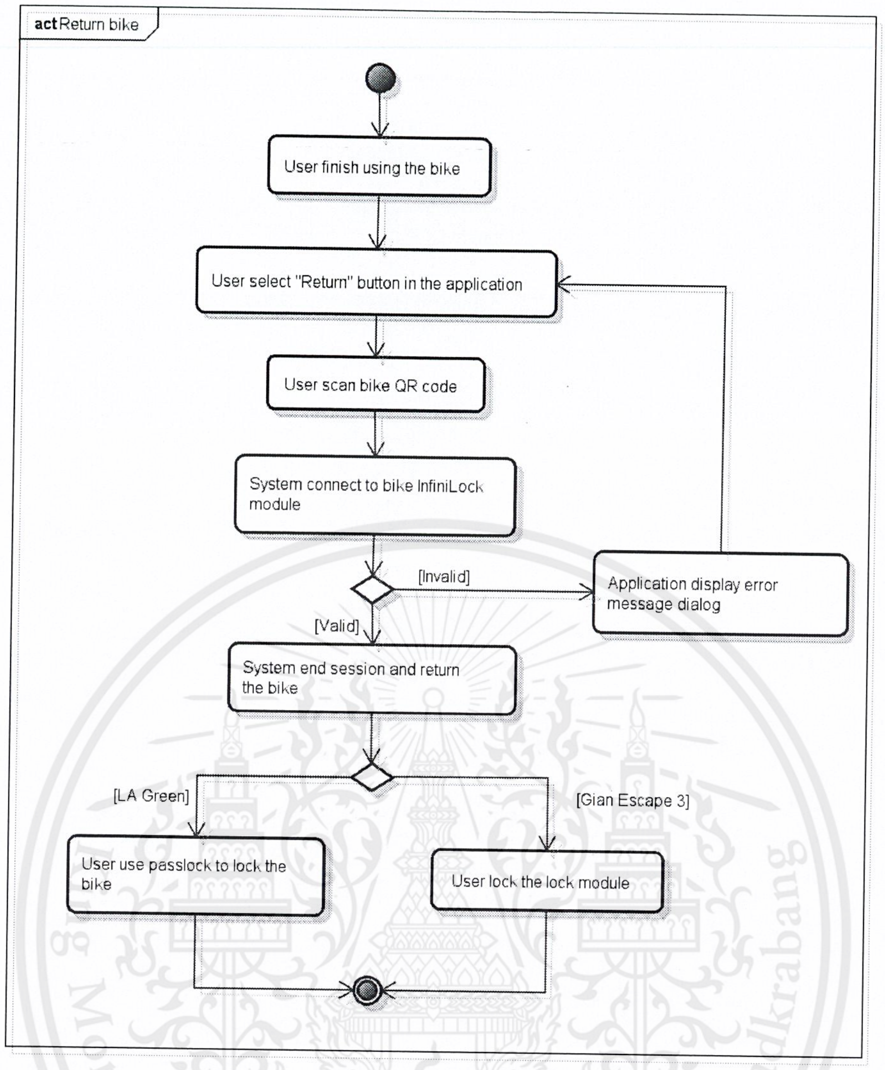

Figure 5-7: Flow of returning a bike

Chapter 6

Development

This chapter provides details of the development process of the project.

6.1 Development Tools

The development tools used in this project are:

1. Operating System:
   - Android 4.4 (KitKat)
2. Programming Language:
   - JavaScript
   - Java 8
   - Swift 4
3. Database Management System:
   - SQLite3
4. Utility Softwares:
   - Adobe Photoshop CC 2017
   - Adobe Illustrator CC 2017

- Autodesk Fusion 360
- FlashForge FlashPrint

6.2 Development Techniques

There are several development techniques used to ensure extensibility of KMITL Bike Application.

Functional Programming A programming concept such as pure function, map, filter, and reduce.

Reactive Programming A technique used to drive events and actions through data layer, communication layer, and between user interfaces using RxJava and RxAndroid.

Singleton Concept A design pattern which helps prevent memory leak and shared context in the application.

Dependency Injection Used for injecting objects into specific classes to solve Android's constructor problem and for simplifying testing purpose.

Protocol-Oriented Programming (POP) Used in iOS application, introduced in Swift language, to leverage the concept of composition over inheritance by programming abstraction layer of several classes.

6.3 Development Iterations

6.3.1 System Administration Application

To fix the maintenance problems stated in Chapter 3, KMITL Bike administration application was developed. The application was implemented using React and React Native frameworks.

The application, created for both mobile and web, was designed specifically to help system administrators maintain the system more efficiently.

Table 6.1: Frameworks & Libraries used

Frameworks & Libraries	Description
React & React Native	Core frameworks
Mobx	State management
React Native FCM	Create notification
React Native Maps (Airbnb)	Map controller
React Navigation	Navigate between pages
Axios	API client

Features in this iteration:

- View user's status
- View user's history
- View bike's status
- View bike's location
- Contact user

Mobile Application

KMITL Bike maintenance mobile application focuses on bike volunteer staffs who work on site with the bikes. Mobile application provides simplified tools and UI tailored to gracefully fits mobile phone screen so that they can be check and update bike and user status at an instant.

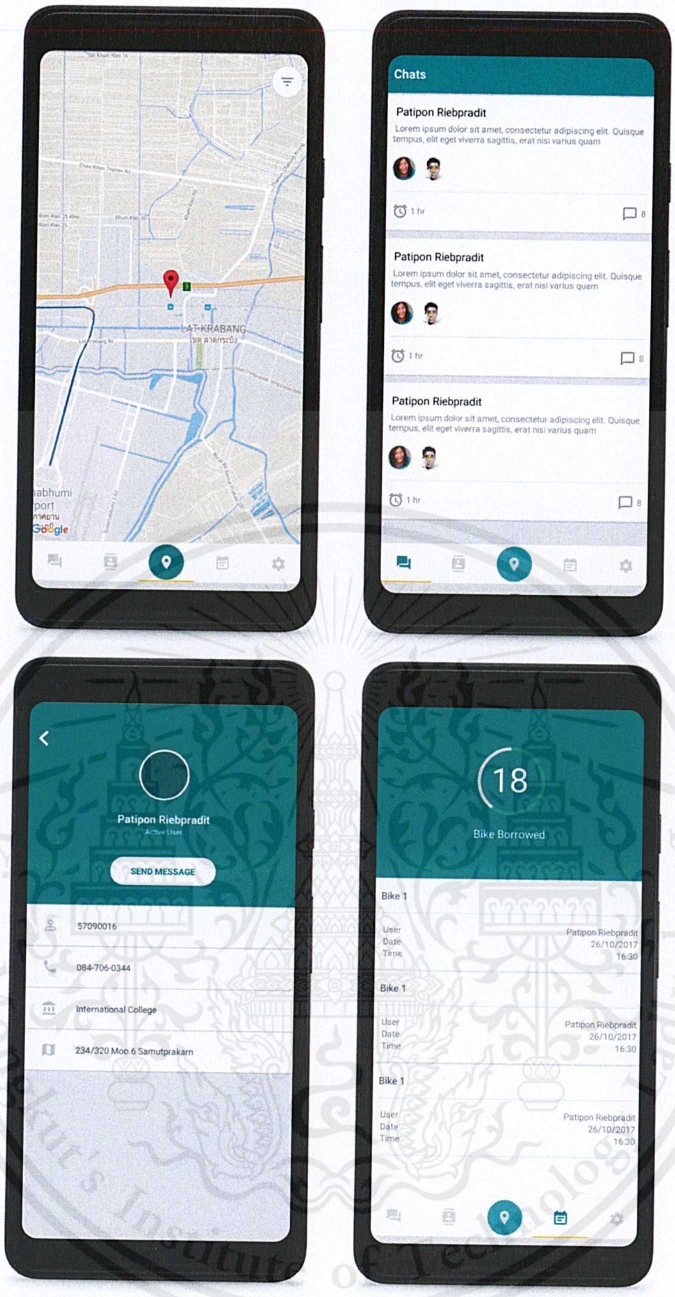

Figure 6-1: KMITL Bike maintenance mobile application

Web Application

As the maintenance team are not only working on site but also have to watch over the whole system. To fulfill this needed, having only the mobile application would not be enough for them to maintain the system properly. With advised from the project adviser, Dr. Isara, the decision was made to further extend the application to the web as well. The web application was create for system administrators to monitor overall of the system. The application provides easy-to-use interface for larger displays in personal computers.

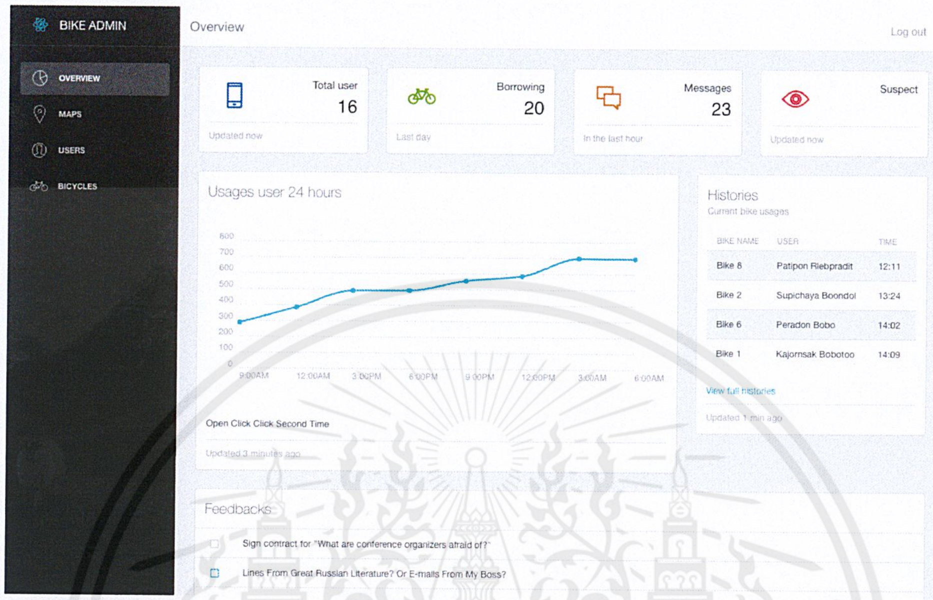

Figure 6-2: KMITL Bike maintenance web application - Overview

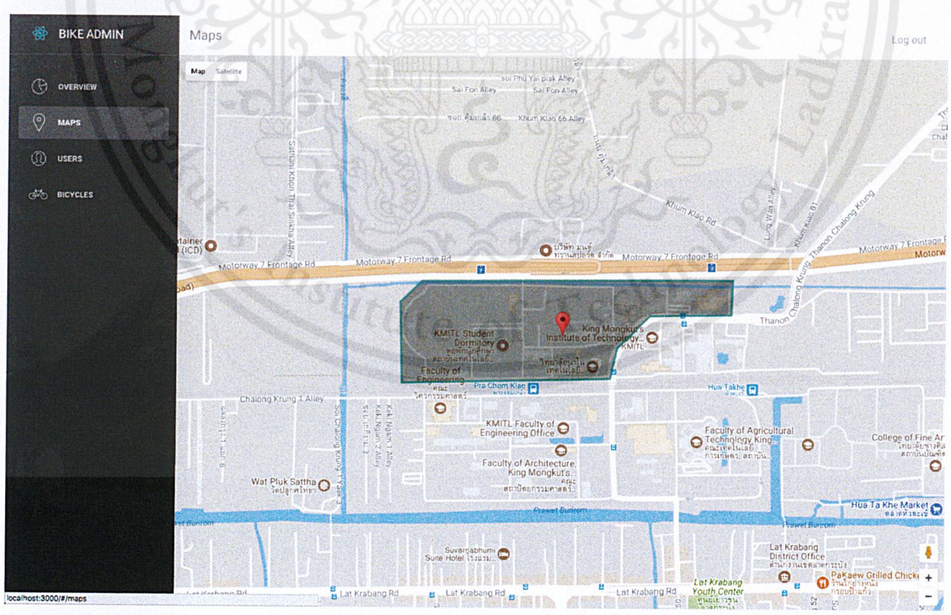

Figure 6-3: KMITL Bike maintenance web application - Maps

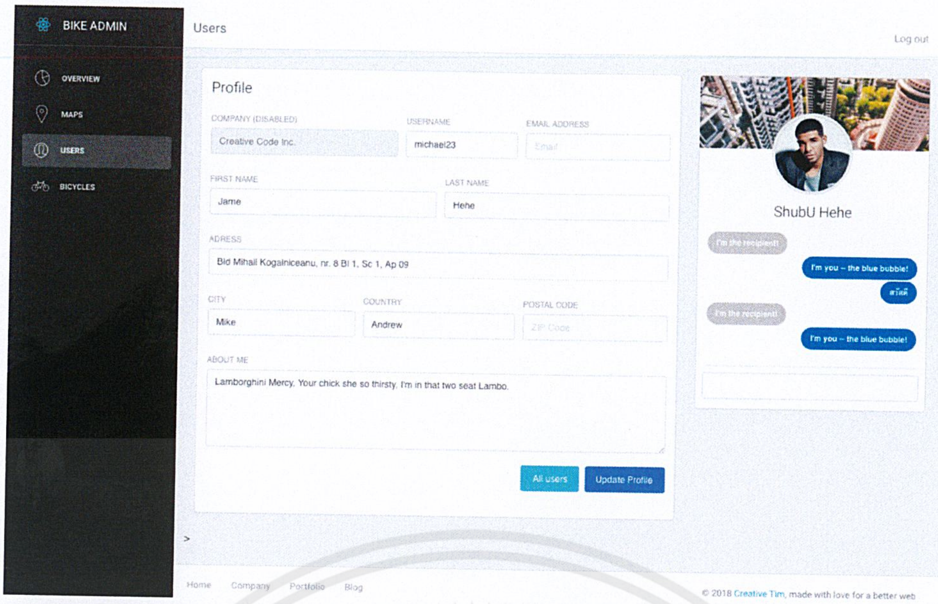

Figure 6-4: KMITL Bike maintenance web application - Users

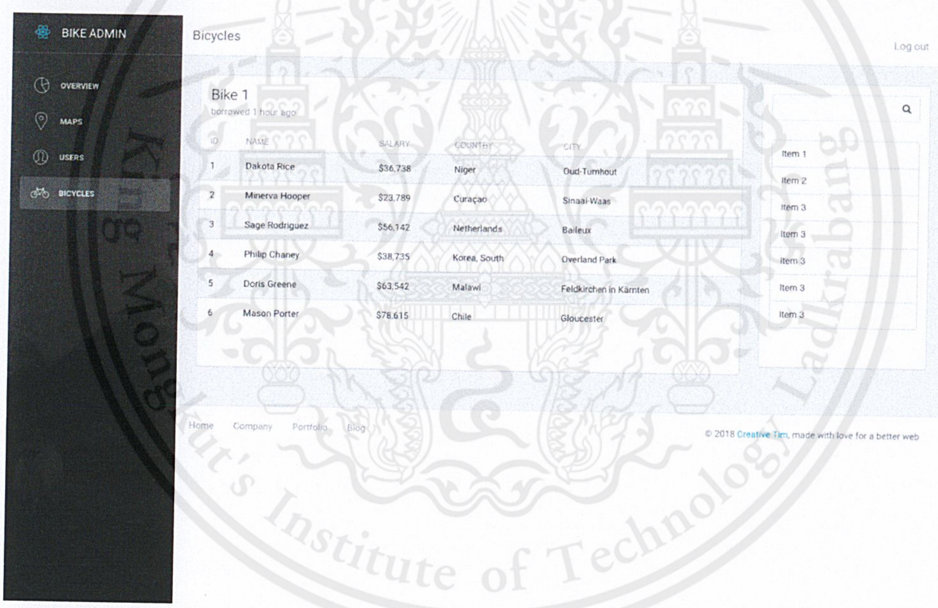

Figure 6-5: KMITL Bike maintenance web application - Bicycles

6.3.2 Electronic Lock Improvement

The electronic Lock is a bicycle lock module which was designed by the previous team (Ms. Chatchaya Chanchua, Mr. Pattarawut Imamnuaysup, and Mr. Sasawat Chanate). However, the modules contain critical defects that require modifications before installing to the bicycles. Figure 6-6 illustrated the overview of the electronic lock unit and its components.

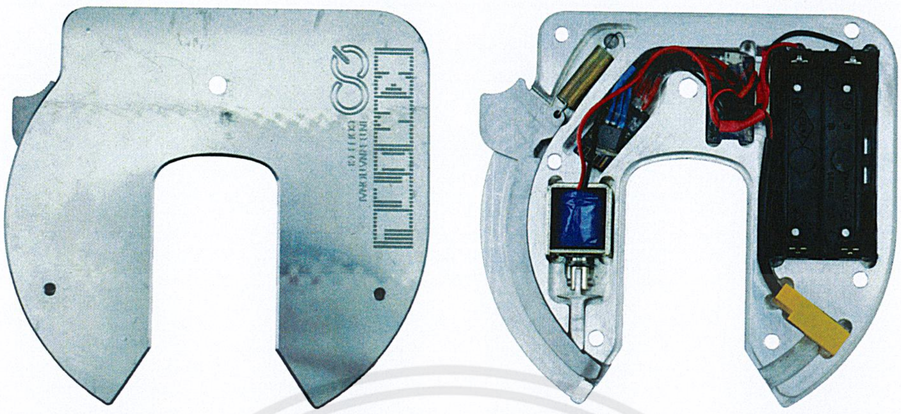

Figure 6-6: Electronic lock by previous team

There are two parts that were defected by the CNC production process. The defects affected the unlocking mechanism causing failure in the automatic unlocking process.

Shaft It is found that the niche on the shaft part has a steep angle causing high friction when attempting unlock procedure. However, it is challenging to figure the correct angle. If the angle is too low, it will make the lock unstable and easy to hack.

Sword Pin The part is too long in length that it could not be pull all the way through niche part of the shaft.

Although, due to time limitation, the modified parts were create using 3D printing method. The replace part was created by 3D printing method using PLA plastic filament. However, the part have low durability compare to the original aluminum part, hence, it have to be replace several times throughout the test flight period. Since the electronic lock module uses inductive proximity sensor to detect lock status, a metal screw was attached to the modified plastic lock shaft to trigger the sensor. Figure 6-7 illustrates the modified lock shaft and Figure 6-8 presents the modified lock sword pin.

Figure 6-7: Electronic Lock Shaft

Figure 6-8: Electronic Lock Sword Pin

The comparison between original lock components and the modified components is displayed in Figure 6-9. Additionally, Figure 6-10 illustrated the modified lock installed on GIANT Escape 3 bike.

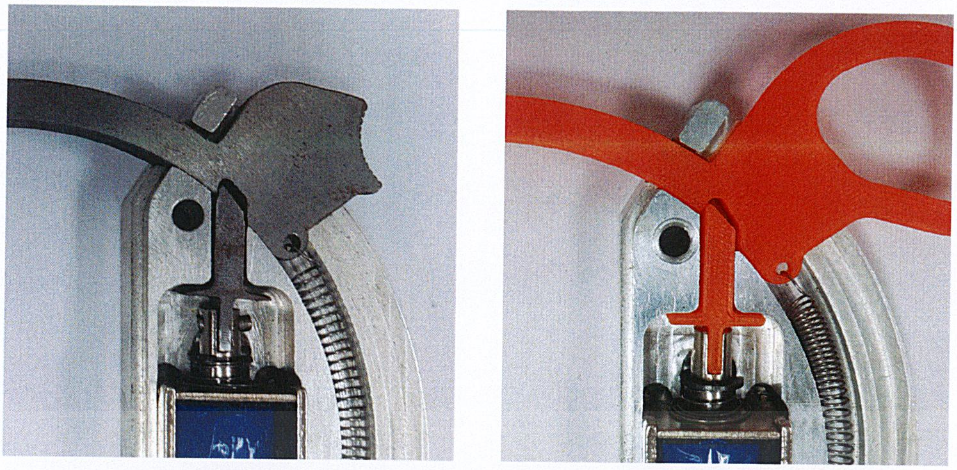

Figure 6-9: Comparison between original and modified mechanism

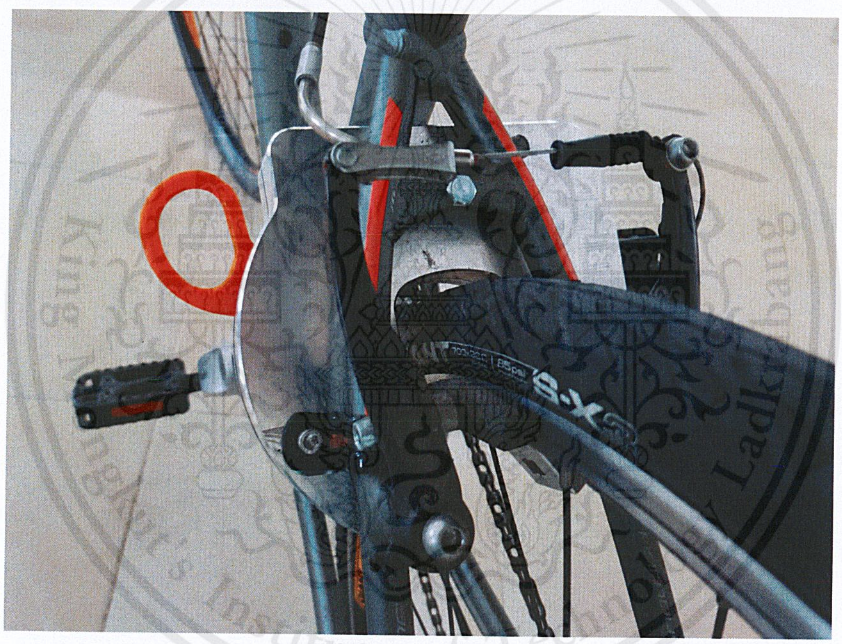

Figure 6-10: Modified electronic lock on GIANT Escape 3 bike

6.3.3 KMITL Bike Application

Relaunching KMITL Bike

After the modified the electronic lock modules were installed to the bikes and several fixes applied to the previous team's application, KMITL Bike 2.0, KMITL Bike system was relaunch on March 22, 2018.

There were several problems with the previous team's application. There is an issue with the iOS barcode scanner library which does not recognized the bike's barcode. With time limitation, instead of using barcode authentication, QR code is used as a replacement which can be recognized by libraries on both iOS and Android. The application is also modified to return the passcode in case the selected bike uses the old passcode lock (instead of the electronic lock) and will activate and unlock the electronic lock module if it uses the electronic lock.

Table 6.2: Frameworks & Libraries used

Frameworks & Libraries	**Description**
React & React Native	Main framework used
Mobx	State management
React Native FCM	Application Notification
React Native Maps (Airbnb)	Map controller
React Navigation	Navigate between pages
Axios	API client

Features in this iteration:

- Borrow bike
- Return bike
- Find bike
- View riding history
- Near real-time tracking
- Contact system administrator

Figure 6-11: KMITL Bike 2.0 Application - Relaunch

6.3.4 KMITL Bike 3.0 Application

KMITL Bike 2.0 application created in the previous implementation was not well-designed. It was not designed with a software architecture at its foundation. On top of that, the libraries and dependencies in used are from third-party which are deprecated and do not provide flexibility in customizing different parameters.

React Native location services are not reliable and contain errors in budget phone models. The library is not customizable which limits further development features such as geofencing. It became clear that revamping of the entire application will use less time than modifying the existing application. The revamped version will be created from scratch using native languages and frameworks of each platform (e.g. Java for Android and Swift for iOS) to avoid further complication with libraries and dependencies.

The revamped application aims to enhance the current system and supports for the future extensions.

Table 6.3: Features of KMITL Bike 3.0 Application

Existed features	Borrow bike Return bike Find bike View riding history Near real-time tracking Contact system administrator
New features	Automate report system Geofencing feature

Android Application

KMITL Bike 3.0 Android application is written in Java 8. The iteration solve location errors, borrow/return state, and other non-native issues occurred in the system.

Table 6.4: Frameworks and libraries used in Android version

Frameworks & Libraries	**Description**
RxJava	For implement reactive programming
Retrofit & OkHttp	API client
Google's Dagger 2	Perform dependency injection between modules & components
Butter Knife	Inject UI View and bind event into activity class
RxBleClient	Bluetooth controller
ZXing	QR code scanner

iOS Application

KMITL Bike 3.0 iOS application is developed using Swift 4 language and follow Android version's user interface.

Table 6.5: Frameworks and libraries used in iOS version

Frameworks & Libraries	Description
RxSwift	For implement reactive programming
Moya & OkHttp	API client
Bulletin board	Create dialog box in iOS style
RxBluetoothKit	Bluetooth controller
SwiftMessages	Perform local notification and error handling

Figure 6-12 and Figure 6-14 presents the UI design comparison of the previous version (2.0) and the revamped application (3.0) in both Android and iOS. Figure ?? shows the bike information page implemented in the revamped application.

Figure 6-12: UI Design Comparison - Main page

Figure 6-13: UI Design Comparison - Menu page

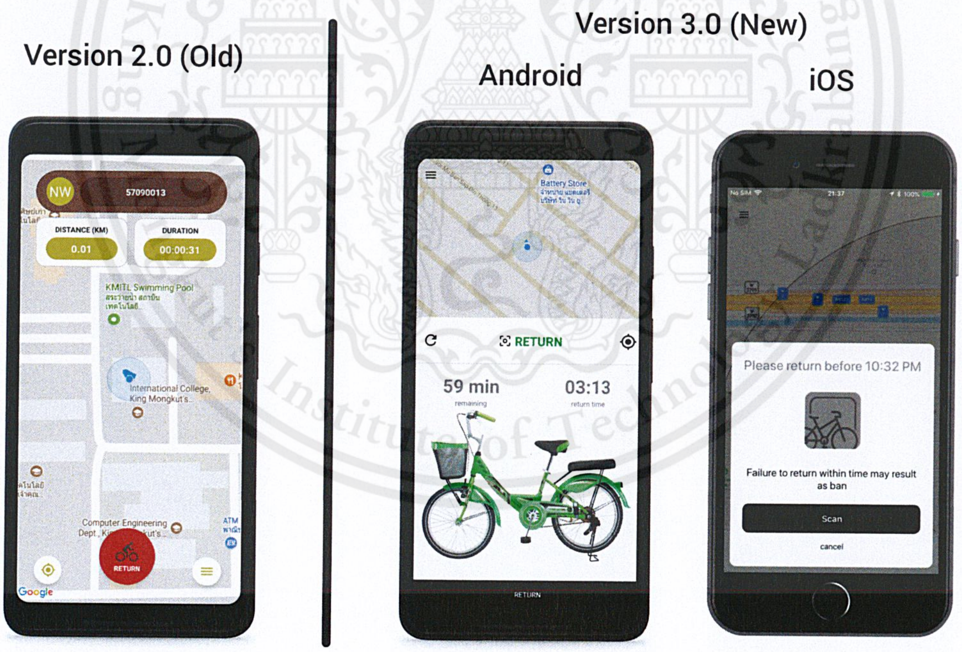

Figure 6-14: UI Design Comparison - Return page

Chapter 7

Results and Evaluation

7.1 Experimental Method

The experiment for this system was initiated through test flight. The test flight was conduct within a section of King Mongkut's Institute of Technology Ladkrabang main campus. The application available on both Android and iOS platforms. The system limits users to KMITL students and staffs only using their institution login credentials. The test flight focuses on users' feedbacks on the automate bike lock aspect and usage of the bike. The purpose of this test flight is to gain feedbacks from users and use it to improve the system in future development phases.

7.2 Test Flight

The test flight start from March 22, 2018 to May 4, 2018 for a total of 44 days. According to the previous test flights conducted by the previous team, female users tend to prefer bicycles with companion seat. Hence, the test flight was launched with two different bike models which are LA City Green and GIANT Escape 3. However, the electronic lock module was designed specifically to fit the GIANT Escape 3 model. Therefore, the LA City Green model was installed with an ordinary 5-digit passcode lock instead. After the test flight ended, the number of usage in each type of bike will be analyzed for future system development.

Table 7.1 lists the details of bike models used in the system. Figure 7-1 and Figure 7-2 are sample photos of each model.

Table 7.1: Bike Types

Model	LA City Green	GIANT Escape 3
Type	City Bike	Commuting Bike
Size (Seat Tube)	16″	XS/15″ and M/19″
Number of seats	2	1
Price (THB)	3,500	8,800
Amount in service	3	5

Figure 7-1: LA City Green

Figure 7-2: GIANT Escape 3

For security and maintenance reasons, the service hours is set to 6 AM – 7 PM daily and limit service area to be within a section of the institute. The permitted usage area are shown in Figure 7-3.

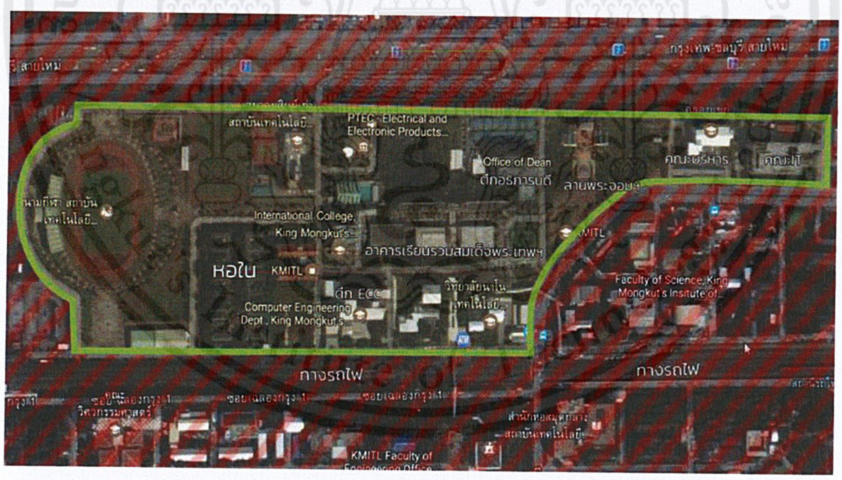

Figure 7-3: Test Flight service area

Since the system use user's mobile phone location to identify location of the bike. QR code were introduced to make sure that user user are physically with the selected bike. Figure 7-4 shows a picture of a sample QR code that was on one of the bikes.

Figure 7-4: Sample of Bike QR Code

7.3 Results

Figure 7-5 presents the number of application downloads in both Android and iOS platform between March 18, 2018 to May 4, 2018. The application starts off with 3 downloads and reaches its peak at 88 downloads in March 22, the relaunch date. After that, the number sharply drops. Overall, the total amount of application downloads grows linearly.

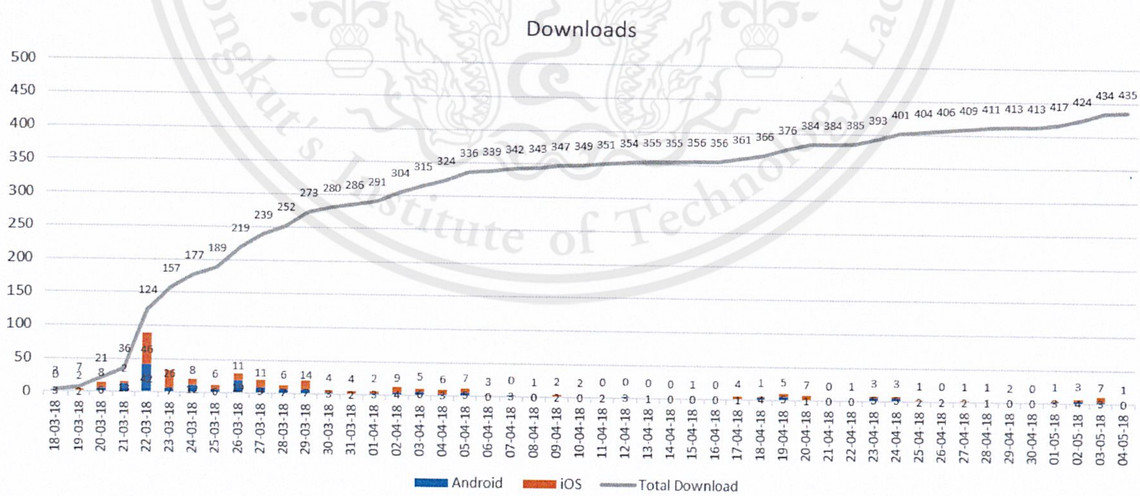

Figure 7-5: Numbers of downloads

Number of application download also reflected the number of new users which seems to have a similar trend as well. Figure 7-6 represents the actual number of daily new users.

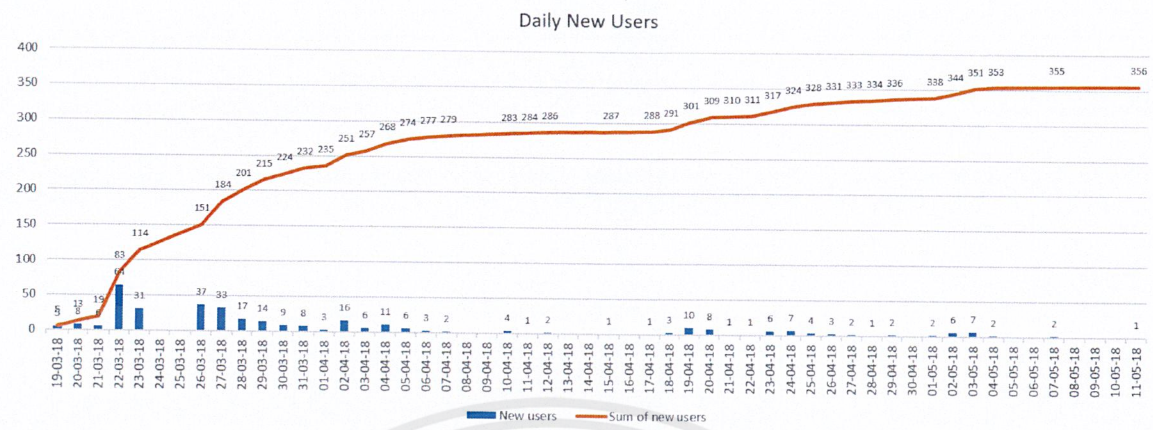

Figure 7-6: Numbers of daily new user

Figure 7-7 indicated number of usage of KMITL Bike application in each day during this test flight. There were approximately 500 total sessions during the test flight. According to the graph, the day of the week with the highest bike demand is on Thursday where flea market was held near service area.

Note that the system was closed during Songkran holiday on April 12 – 16, 2018.

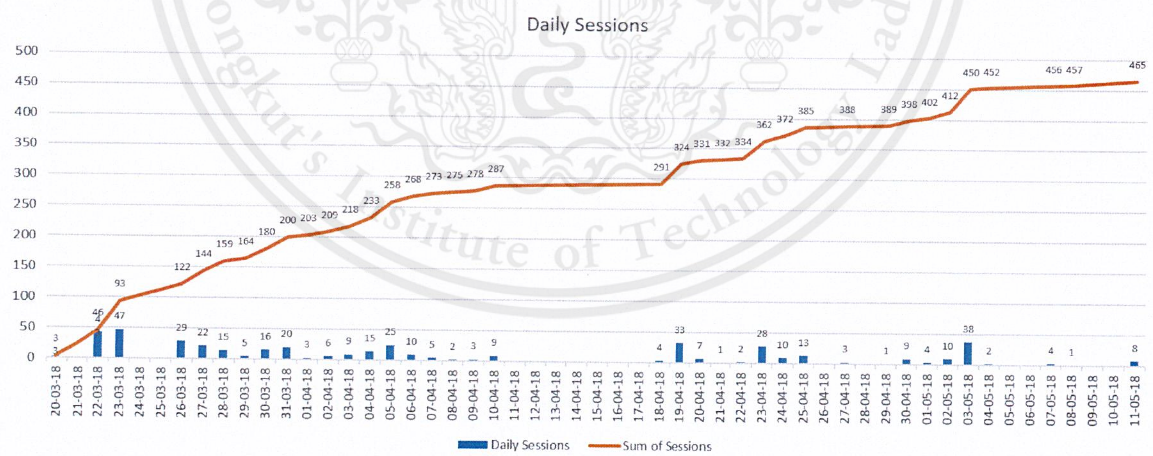

Figure 7-7: Numbers of daily session

Figure 7-7 is considerably lower compared to the test flight constructed by the team in the previous year. This could caused by the discontinuous of test flight period and the instability of the electronic lock module. Moreover, the introduction of two lock types may have confused users as well.

7.4 Users' Preferences

Throughout the test flight period, there were a total of 352 users registered in the system and over 500 sessions of borrowing bicycles. These information help understand users' preference to further improve the system towards users' demands.

Figure 7-8 presents the ration of user gender that were registered in the system.

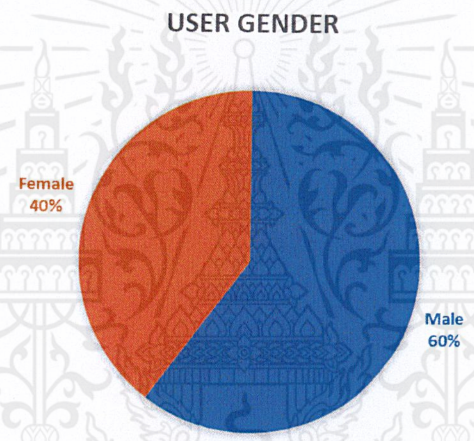

Figure 7-8: User categorized by gender

Figure 7-9 and Figure 7-10 visualizes the usage of each bike model categorized by their gender. The figures show that male users tend to use both model, while over 74 percent of female users prefer to use the LA City Green model.

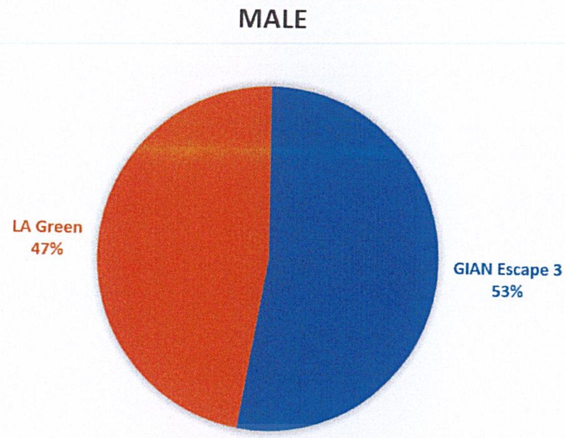

Figure 7-9: Male usage of different bike

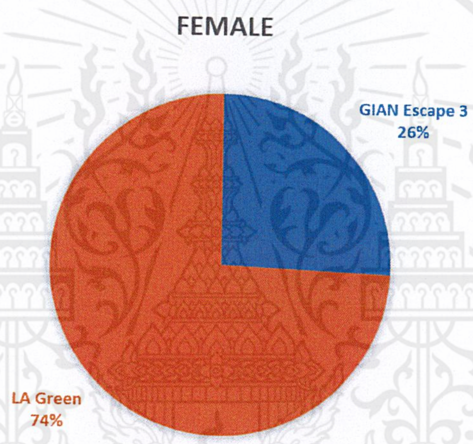

Figure 7-10: Female usage of different bike

The largest portion of users came from Faculty of Engineering followed by International College and Faculty of Science, respectively. The high usage may have come from the short distance between the service area and their faculties. The size of each department also have a significant effect on number of user as well. Figure 7-11 illustrated users categorized by their colleges or faculties.

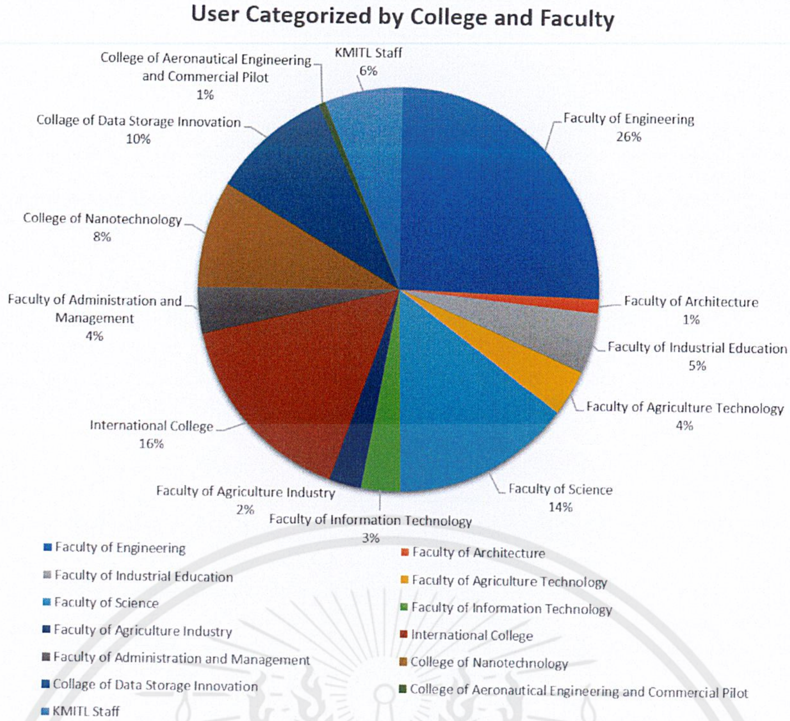

Figure 7-11: User categorized by college/faculty

7.5 Improvements over previous implementation

Table 7.2 shows list of problems found during the test flight and improvements made in this implementation.

Table 7.2: Improvements

Feature	Problem	Improvement
Location Service	Low accuracy Not customizable	Improve location accuracy Customizable to consume less power
Software Architecture	Not well-designed	MVP (Android) MVVM (iOS)
Report System	Not exist	Include automate report system
Geofencing	Not exist	Include geofencing feature
User feedback report	Not exist	Include user feedback report

Chapter 8

Conclusion

8.1 Summary

KMITL Bike system was originate by the team from last year, however, the system remains incomplete and contains many errors. After the relaunch of the system, there are over 300 new users and 450 sessions during the 44 days of test flight period. Many students and institute staffs from different colleges and faculties all over the institution showed their interest in the service. However, due to the difficulties in stabilizing the electronic lock module, there are fewer borrowing sessions than last year's test flight.

8.2 Problem and Lesson Learned

During development of this project, there are several issues needed to be fixed combined with inexperience in the new technique being used which results in consuming more time than anticipated.

8.2.1 Project Environments and Dependencies

The Reac Native on iOS has a lot of dependencies to required to start debugging the previous version of the application. Although cross-platform mobile application such as React Native can be coded once to be run on both Android and iOS but components inside the two platform are different. Therefore, several bugs occurred only on iOS while others occurred on Android. Solving consumes more time than expected which results in the delaying of the relaunch.

8.2.2 3D Model

To figure the precise angle and length of each component of electronic lock module, several trial-and-error attempts were made. Over 30 versions of lock shaft and sword pins were produced during this process. Moreover, 3D printer's speed and temperature of both nozzle and printing bed must be manually set. Achieving the correct configuration of 3D printer is also a great challenge due to inexperience in 3D model and printing.

8.3 Future Work

Regarding the aspect of the electronic lock module, here are some of the things that can be improved:

- Detect lock status
- Detect lock battery and increase its battery life
- Integrate alarm buzzer for geofencing feature
- Integrate real-time GPS tracking system
- Improve lock and mobile connection
- Optimize battery replacement method
- Optimize water resistant sealing and design

Regarding the aspect of the KMITL Bike services, here are some of the things that can be improved:

- Increase services hours and area
- Increase amount of bike in the system
- Add ride statistic (e.g. usage per month, usage heat map)
- Add ride analytic (e.g. calorie used, carbon reduce)

Bibliography

[1] "Vehicles and Air Pollution", Uvm.edu, 2017. [Online]. Available: https://www.uvm.edu/ empact/air/cars.php3. [Accessed: 12- Oct- 2017].

[2] "Cars, Trucks, and Air Pollution", Union of Concerned Scientists, 2014. [Online]. Available: http://www.ucsusa.org/clean-vehicles/vehicles-air-pollution-and-human-health/cars-trucks-air-pollution#.Wd9SR2iCyUk. [Accessed: 12- Oct- 2017].

[3] "benefits of alternative transportation", Morpc.org, 2017. [Online]. Available: http://www.morpc.org/transportation/CommuterChallenge/benefits-of-alternative-transportation/index. [Accessed: 16- Oct- 2017].

[4] C. Chanchua, P. Imamnuaysup and S. Chanate, *Universal Locking System: InfiniLock*. Bangkok: International College, King Mongkut's Institute of Technology Ladkrabang, 2017.

[5] "Benefits of Cycling", Walkandrollpeel.ca, 2016. [Online]. Available: http://www.walkandrollpeel.ca/cycling/benefits.htm. [Accessed: 16- Oct- 2017].

[6] J. HONG, "Bike sharing: New entrant to use geofencing", The Straits Times, 2017. [Online]. Available: http://www.straitstimes.com/singapore/new-entrant-to-use-geofencing. [Accessed: 16- Oct- 2017].

[7] "ofo - Crunchbase", Crunchbase, 2017. [Online]. Available: https://www.crunchbase.com/organization/ofo. [Accessed: 17- Oct- 2017].

[8] H. Cheng, "ofo Leads the Charge in Making Bike-Sharing Safe from Cybersecurity Attacks", Thetechrevolutionist.com, 2014. [Online]. Available: http://www.thetechrevolutionist.com/2017/09/ofo-leads-charge-in-making-bike-sharing.html. [Accessed: 16- Oct- 2017].

[9] P. Samsoon, "Singapore's oBike has a long way to go", Techinasia.com, 2017. [Online]. Available: https://www.techinasia.com/talk/obike-singapore-prove-itself. [Accessed: 18- Oct- 2017].

[10] J. Russell, "Meet Mobike, a billion-dollar bike-sharing startup from China", TechCrunch, 2017. [Online]. Available: https://techcrunch.com/2017/07/12/chinese-bike-sharing-startup-mobike/. [Accessed: 20- Oct- 2017].

[11] A. Bowden, "Oxford Ofo users who park bikes outside 'geo-fence' area will be getting a phone call", road.cc, 2017. [Online]. Available: http://road.cc/content/news/228129-oxford-ofo-users-who-park-bikes-outside-geo-fence-area-will-be-getting-phone. [Accessed: 16- Oct- 2017].

[12] Mobike UK,"Where to Mobike in Manchester", Mobike.com, 2017. [Online]. Available: https://mobike.com/uk/blog/post/recommended-parking. [Accessed: 20- Oct- 2017].

[13] L. Othman, "oBike to introduce points system that penalises 'ungracious' users", Channel NewsAsia, 2017. [Online]. Available: http://www.channelnewsasia.com/news/singapore/obike-to-introduce-points-system-that-penalises-ungracious-users-8720774. [Accessed: 20- Oct- 2017].

[14] D. Clarke, "iBrake Remote Braking System for Kids Bikes", Hackster.io, 2017. [Online]. Available: https://www.hackster.io/daveclarke/ibrake-remote-braking-system-for-kids-bikes-2a0bbe. [Accessed: 17- Oct- 2017].

[15] "MiniBrake - product information", MiniBrake, 2017. [Online]. Available: http://www.minibrake.com/product. [Accessed: 17- Oct- 2017].

[16] K. Townsend, "Introduction — Introduction to Bluetooth Low Energy — Adafruit Learning System", Learn.adafruit.com, 2015. [Online]. Available: https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction. [Accessed: 03- Mar- 2018].

[17] K. Townsend, "GATT — Introduction to Bluetooth Low Energy — Adafruit Learning System", Learn.adafruit.com, 2018. [Online]. Available: https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt. [Accessed: 04- Mar- 2018].

[18] "How Servo Motors Work", Jameco.com, 2018. [Online]. Available: https://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html. [Accessed: 07- Mar- 2018].

[19] "Functional Programming Introduction", www.tutorialspoint.com, 2018. [Online]. Available: https://www.tutorialspoint.com/func tional_programming/functional_programming_introduction.html. [Accessed: 08- May- 2018].