

# Mobile Application for Parking Lot Finding



Joshua Daniel Sevilla  
Patchanon Ekrungruangchai  
Patipat Suksanga  
Sorikorn Asawakunalai

Bachelor of Engineering in Software Engineering  
School of Engineering  
King Mongkut's Institute of Technology Ladkrabang  
Academic Year 2022

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

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



COPYRIGHT 2022  
SCHOOL OF ENGINEERING  
KING MONGKUT'S INSTITUTE TECHNOLOGY LADKRABANG

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

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

**Thesis – Academic Year 2022**

Bachelor of Engineering in Software Engineering

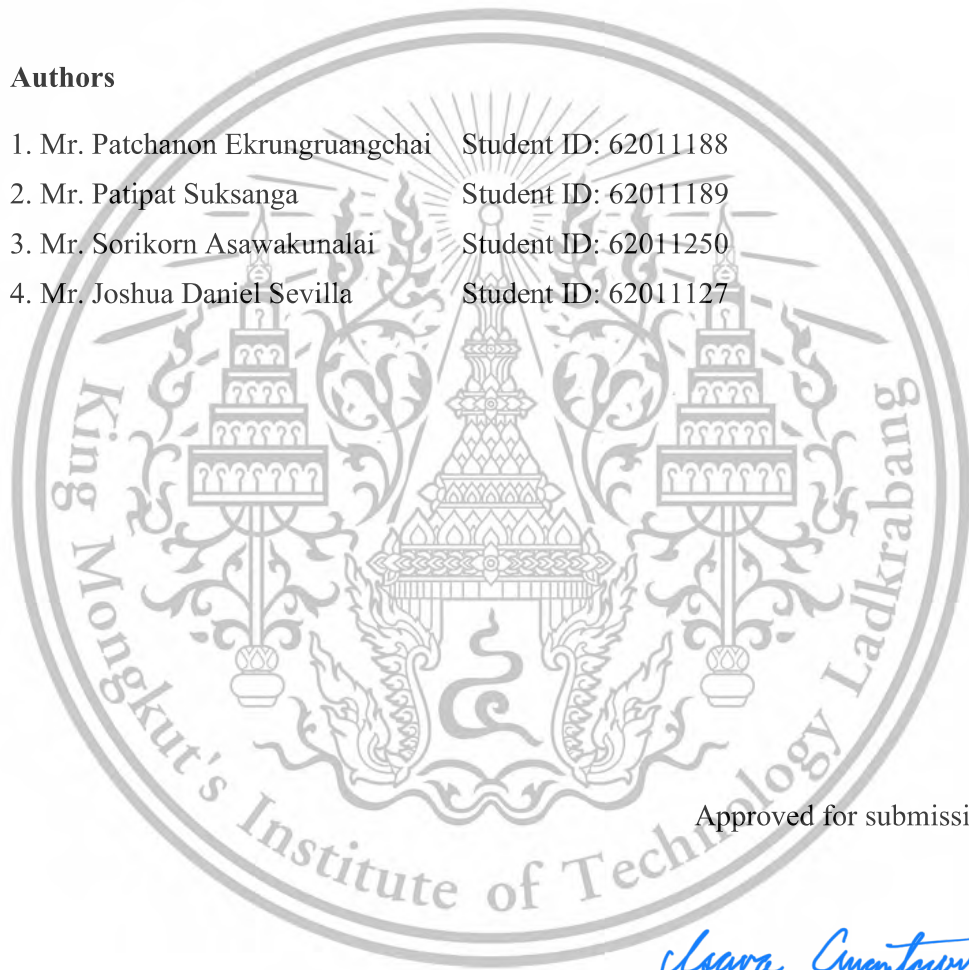
School of Engineering

King Mongkut's Institute of Technology Ladkrabang

**Title:** Mobile Application for Parking Lot Finding

**Authors**

1. Mr. Patchanon Ekrungruangchai Student ID: 62011188
2. Mr. Patipat Suksanga Student ID: 62011189
3. Mr. Sorikorn Asawakunalai Student ID: 62011250
4. Mr. Joshua Daniel Sevilla Student ID: 62011127



Approved for submission

*Isara Anantavasilp*

(Asst.Prof.Dr. Isara Anantavasilp)  
Advisor

Date *06* / *06* / *23*

## Acknowledgement

We also would like to express our deepest appreciation to Asst.Prof.Dr. Isara Anantavasilp for the valuable advice you provided to us on the subject of development. Your insights and guidance were instrumental in helping us better understand the nuances of this field and we are grateful for the time and effort you put into assisting us.



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

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

## **Abstract**

The aim of this software engineering project is to create a mobile application, that helps King Mongkut's Institute of Technology Ladkrabang (KMITL) students to find parking slots in specific parking areas / spots, to help find parking slots inside the campus quicker and easier.

Our main users in our application are KMITL students. The students can check the number of car parked in each slots in specific parking areas. Know the location of their parking area, and monitor the parking area and their vehicle when parked with a camera.



# Table of Contents

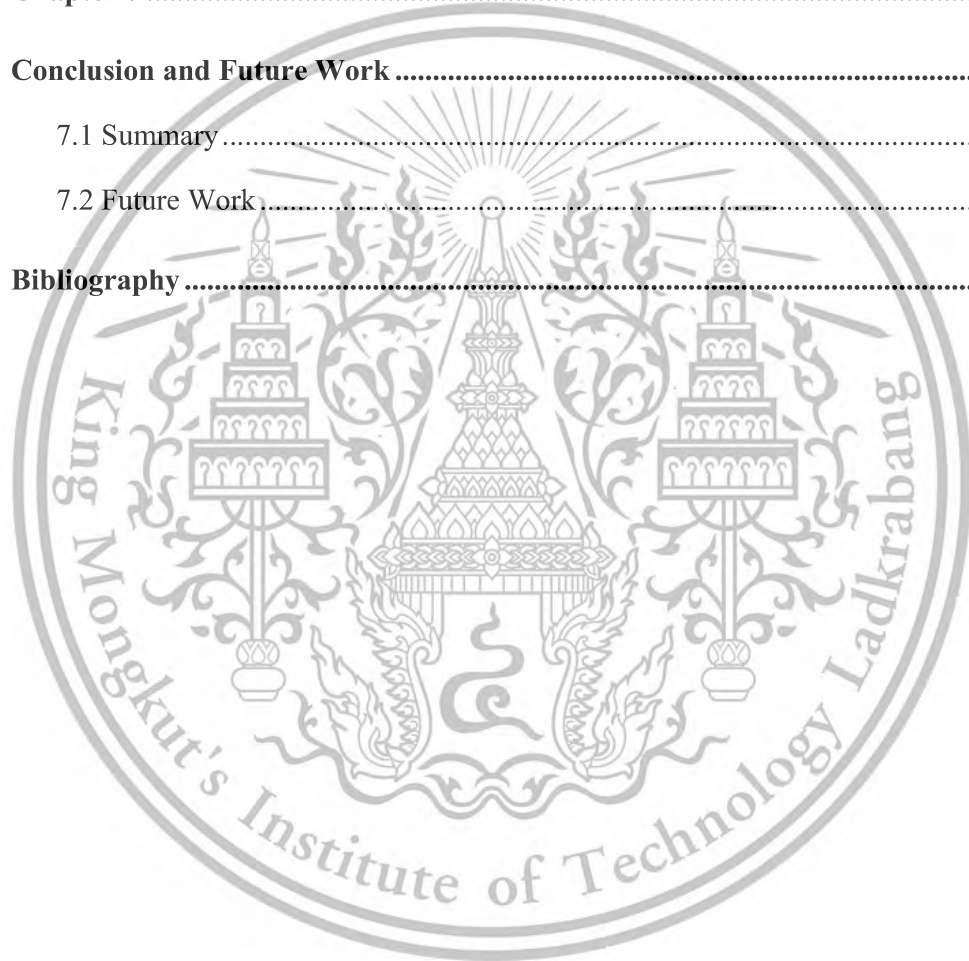
<b>Chapter 1 Introduction.....</b>	<b>1</b>
1.1 Motivation.....	1
1.2 Objectives .....	1
1.3 Scope of Work .....	2
1.4 Thesis Structure .....	2
<b>Chapter 2 Related Works.....</b>	<b>3</b>
2.1 SpotHero .....	3
2.2 Best Parking.....	4
2.3 Parkopedia.....	5
<b>Chapter 3 .....</b>	<b>6</b>
<b>Background Knowledge .....</b>	<b>6</b>
3.1 React Native.....	6
3.2 MySQL .....	6
3.3 OpenCV .....	6
3.4 IP CAMERA (SonOff Cam slim).....	7
3.5 Azure.....	7
3.6 Expo .....	8
3.7 YOLOv8 .....	8
3.8 COCO .....	8
<b>Chapter 4 .....</b>	<b>9</b>
<b>Requirement Analysis / System Architecture / Design.....</b>	<b>9</b>

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

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

4.1 Functional Requirement.....	9
4.2 Non-Functional Requirement.....	9
4.3 Use-Case Diagram .....	10
4.4 System Architecture.....	11
4.6 Class Diagram.....	15
<b>Chapter 5 .....</b>	<b>16</b>
<b>Software Development.....</b>	<b>16</b>
5.1 Development Tools.....	16
5.1.1 Visual studio code.....	16
5.1.2 GitHub.....	16
5.1.3 Figma .....	16
5.1.4 Jira.....	17
5.2 Technology / Concepts used for the System Development.....	17
5.2.1 React Native.....	17
5.2.2 OpenCV.....	17
5.2.3 MySQL .....	17
5.2.4 Microsoft Azure.....	17
5.3 Techniques / Algorithms used for the System Development.....	18
5.3.1 Agile development : Scrum .....	18
5.4 Development Process.....	18
1. Starting on application (Jan): .....	18
2. More discuss (Jan): .....	18
3. Backend designing (Feb): .....	19
4. The camera (Feb): .....	19
5. Development and Integration (Feb-May): .....	19

6. Verification of Completion (late May): .....	19
<b>Chapter 6</b> .....	<b>20</b>
<b>Results</b> .....	<b>20</b>
6.1 User Interface.....	20
View Parking via Camera .....	27
<b>Chapter 7</b> .....	<b>28</b>
<b>Conclusion and Future Work</b> .....	<b>28</b>
7.1 Summary.....	28
7.2 Future Work.....	28
<b>Bibliography</b> .....	<b>29</b>



# List of Figures

Figure 1. SpotHero.....	.....
Figure 2. Best Parking.....	.....
Figure 3. Parkopedia .....	.....
Figure 4. SonOff Cam slim.....	.....
Figure 5. Use-Case Diagram.....	.....
Figure 6. System Architecture.....	.....
Figure 7. CarDetect Algorithm.....	.....
Figure 8. Homepage UI .....	.....
Figure 9. Searching UI.....	.....
Figure 10. Class Diagram.....	.....
Figure 11. Login UI .....	.....
Figure 12. Drawer UI .....	.....
Figure 13. Home Screen UI (Signed Out).....	.....
Figure 14. Edit Profile.....	.....
Figure 15. Saved Place.....	.....
Figure 16. Sign In & Sign Up Screen .....	.....

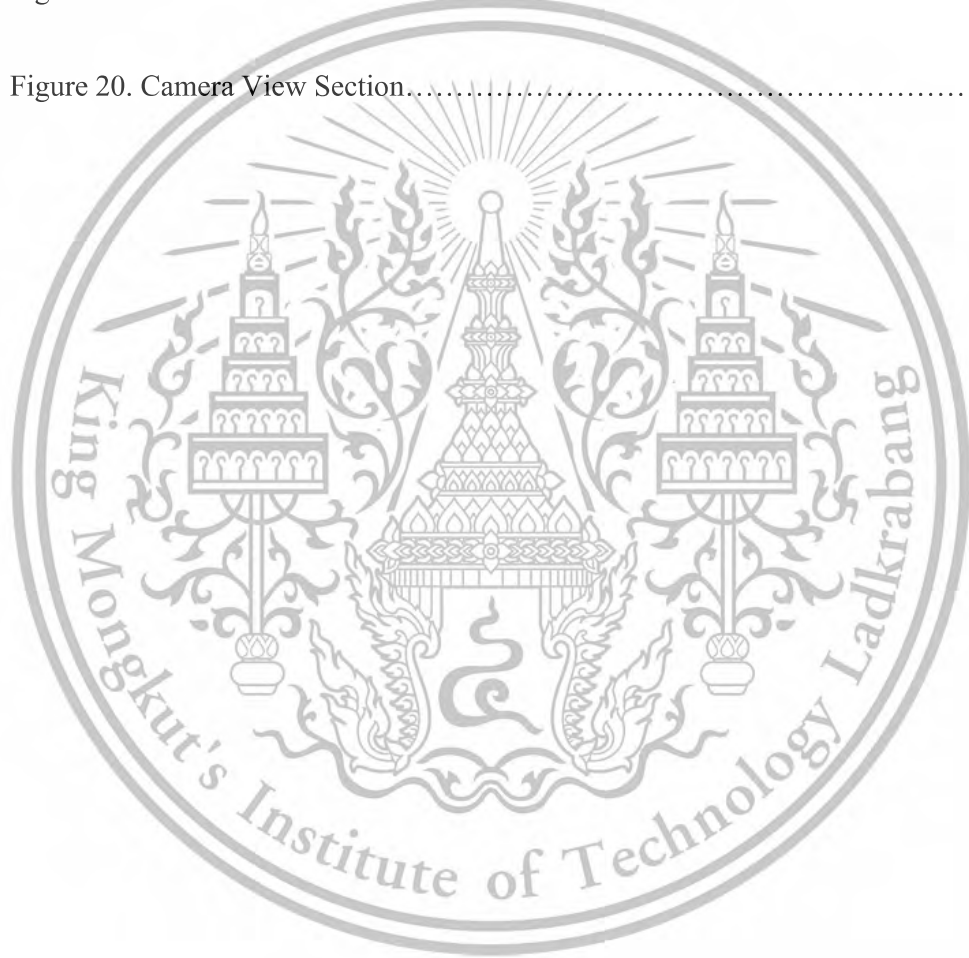
# List of Figures

Figure 17. Home Screen UI (Signed In).....

Figure 18. Saved Parking Lots Section & Search Section.....

Figure 19. Edit Profile Section.....

Figure 20. Camera View Section.....



# Chapter 1

## Introduction

### 1.1 Motivation

Many KMITL struggle to find parking spots in KMITL, since students who aren't familiar with the parking areas would find it hard to locate where to park, as KMITL does not provide information of the locations of the parking areas. Another problem is the number of students, professors, etc, who use vehicles inside the KMITL premises are increasing. This leaves them to spend significant amount of time just searching for parking slots, or in the worst case, if ever all slots inside the premises are full, that leaves them to waste their time and as well on their gas. Therefore, we aim to develop a mobile application that can help students find parking slots with ease with the addition of monitoring their vehicle as well.

### 1.2 Objectives

The objective of our project is to develop a mobile application that helps:

1. Students finds a parking slot
2. Students able to monitor their car.
3. Students maximize their time parking.

## 1.3 Scope of Work

The scope of this project can be listed as follows:

- The mobile application will be used by KMITL students around KMITL.
- The mobile application must be able to locate parking areas around the premises in a map.
- The mobile application must be able to operate cameras on specific parking areas.
- The mobile application must be available on iOS and Android platforms

## 1.4 Thesis Structure

This thesis consists of seven chapters which are arranged as follows:

- Chapter 1: Introduction
- Chapter 2: Related work
- Chapter 3: Background knowledge
- Chapter 4: Requirement analysis/ System Architecture and Design
- Chapter 5: Software Development
- Chapter 6: Results
- Chapter 7: Conclusion and Future work

# Chapter 2

## Related Works

This chapter primarily presents the comparison and evaluation between existing mobile applications related to helping people find parking slots.

### 2.1 SpotHero

Once known as Parking Panda, Spot Hero is a parking reservation service that works with different facilities around the country to provide customers with parking in 28 major cities. Customers can find, reserve, and pay for a parking spot in a lot or garage often at a discounted rate and receive a parking pass either via email or through the app. This service helps drivers find and reserve hourly, monthly and airport parking, as well as parking for events [1].



Figure 1. SpotHero

SpotHero, however, provides reservation for parking slots with a cost, while our mobile application doesn't. Our mobile application differs from how its more used for finding parking slots with ease without a cost. Our mobile application also is exclusively used for the university parking areas while SpotHero provides broader choices as the app operates on different areas of the country / region

## 2.2 Best Parking

The Best Parking application is a convenient and user-friendly mobile application designed to assist users in finding and securing parking spaces. It provides real-time information about available parking spots, rates, and locations in various areas. With its intuitive interface and advanced search features, users can easily locate nearby parking options, compare prices, and make reservations in advance. The application also offers additional features such as navigation to the selected parking spot and digital payment options, ensuring a seamless parking experience for users. Overall, the Best Parking application aims to simplify the process of finding parking and enhance convenience for users in urban areas.



Figure 2 - Best Parking

## 2.3 Parkopedia

Parkopedia is a comprehensive parking application that helps users find, compare, and reserve parking spaces worldwide. With Parkopedia, users can search for parking availability in their desired location, whether it's a city, airport, or specific venue. The app provides detailed information about parking facilities, including rates, opening hours, and available amenities. Users can view parking lot photos, reviews, and filter their search based on criteria such as price, distance, and security. Parkopedia also offers real-time parking space availability, allowing users to see which parking lots have available spots at any given time. The application provides a seamless booking experience, enabling users to reserve parking spaces in advance and receive digital parking passes. With its extensive database and user-friendly interface, Parkopedia makes the process of finding and securing parking spaces convenient and hassle-free.

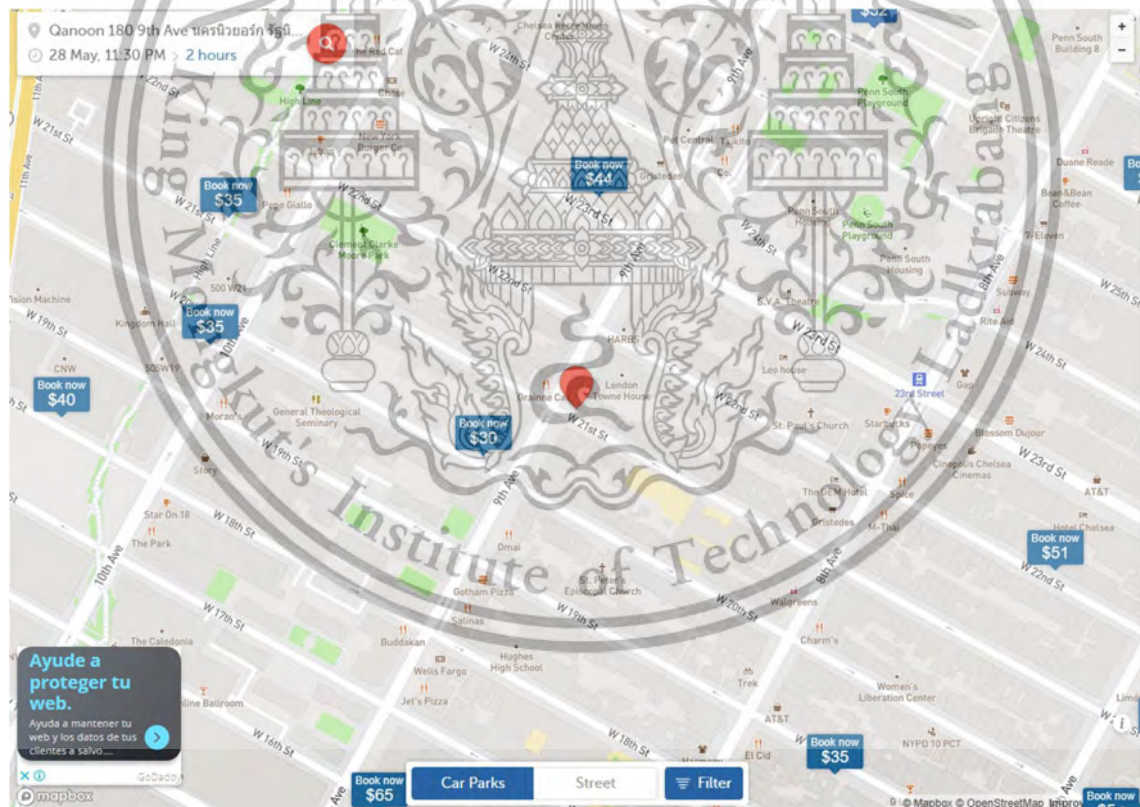


Figure 3 - Parkopedia

# Chapter 3

## Background Knowledge

### 3.1 React Native

**React Native** is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android. It's based on React, Facebook's JavaScript library for building user interfaces, but instead of targeting the browser, it targets mobile platforms. In other words: web developers can now write mobile applications that look and feel truly "native," all from the comfort of a JavaScript library that we already know.

### 3.2 MySQL

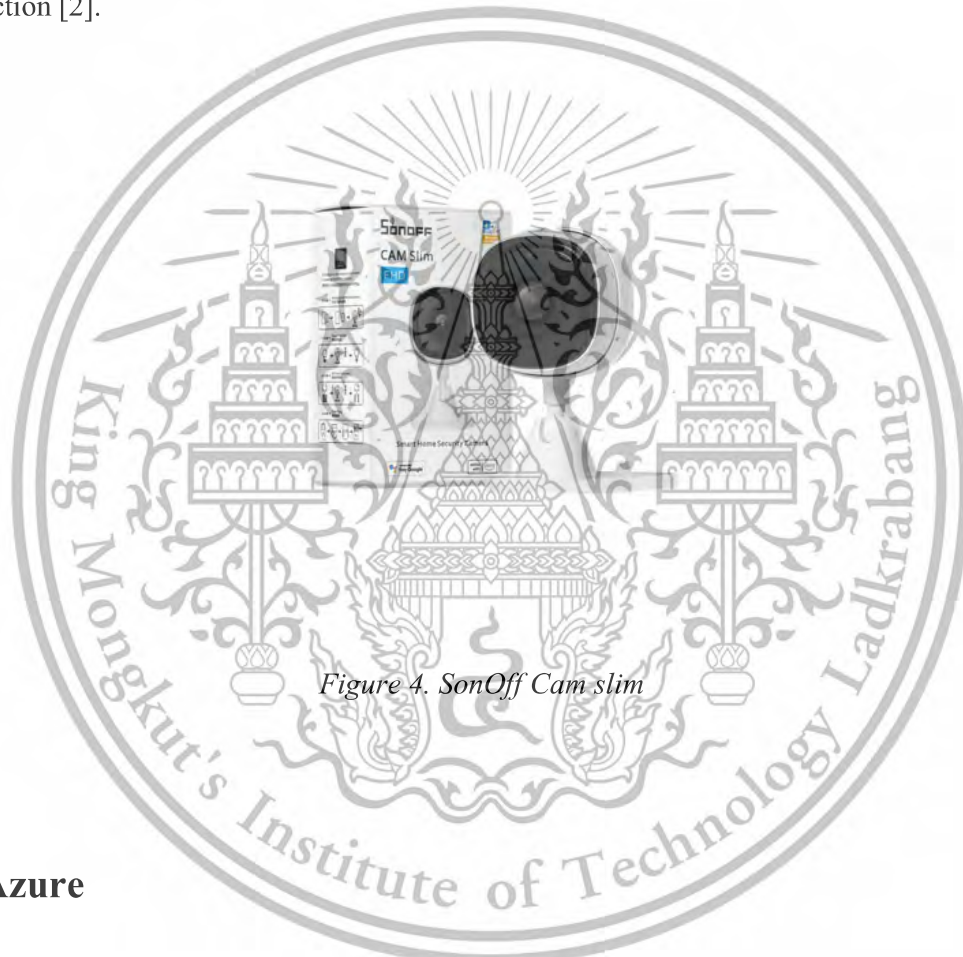
**MySQL** is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL). MySQL is one of the most recognizable technologies in the modern big data ecosystem. Often called the most popular database and currently enjoying widespread, effective use regardless of industry, it's clear that anyone involved with enterprise data or general IT should at least aim for a basic familiarity of MySQL.

### 3.3 OpenCV

**OpenCV** is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez. The library is cross-platform and free for use under the open-source Apache 2 License.

### 3.4 IP CAMERA (SonOff Cam slim)

An **Internet Protocol camera**, or **IP camera**, is a type of digital video camera that receives control data and sends image data via an IP network. They are commonly used for surveillance, but, unlike analog closed-circuit television (CCTV) cameras, they require no local recording device, only a local area network. Most IP cameras are webcams, but the term IP camera or **netcam** usually applies only to those that can be directly accessed over a network connection [2].



*Figure 4. SonOff Cam slim*

### 3.5 Azure

**Azure** is a cloud computing platform and infrastructure created by Microsoft. It provides a range of cloud services, including computing, storage, networking, analytics, machine learning, and AI, that can be used to build, deploy, and manage applications and services.

### 3.6 Expo

**Expo** is a free and open-source platform used for developing and building mobile applications. It provides a set of tools, libraries, and services that simplify the development process and allow developers to create cross-platform apps using JavaScript and React Native. Expo offers a variety of features and functionalities, including a development environment, pre-built UI components, device access APIs, and easy deployment options. With Expo, developers can write code once and deploy it to both iOS and Android devices without the need for separate native code development. It also provides a range of development tools, such as hot reloading, debugging, and testing capabilities, to streamline the app development workflow. Additionally, Expo offers services like push notifications, over-the-air updates, and analytics integration, making it a comprehensive solution for building mobile applications efficiently.

### 3.7 YOLOv8

**YOLOv8**, or You Only Look Once version 8, is an advanced object detection algorithm used in computer vision tasks. It is part of the YOLO (You Only Look Once) family of models. YOLOv8 is known for its real-time object detection capabilities and high accuracy. It uses a single neural network to simultaneously predict object-bounding boxes and their corresponding class labels. YOLOv8 incorporates various optimizations, including feature extraction techniques and anchor-based predictions, to achieve fast and accurate object detection in images and videos. It is widely used in applications such as autonomous driving, surveillance systems, and object recognition tasks.

### 3.8 COCO

**The COCO (Common Objects in Context)** dataset is a large-scale image recognition dataset for object detection, segmentation, and captioning tasks. It contains over 330,000 images, each annotated with 80 object categories and 5 captions describing the scene.

## Chapter 4

### Requirement Analysis / System Architecture / Design

#### 4.1 Functional Requirement

1. The System should allow users to search a location.
2. The System should allow users to see the pin of the parking lot.
3. The System should show the available of the parking lot.
4. The System should be able to update available slots.
5. The System should allow users to mark a favorite parking lot.
6. The system should allow users to search a history of location/parking lot.
7. The system should allow admins to manage a pin of the parking lot.
8. The system should allow admins to manage users.

#### 4.2 Non-Functional Requirement

1. The system should provide API documentation.
2. The system should receive HTTP requests.
3. The system should allow unauthorized users to view a parking lot.
4. The system should not allow unauthorized users to save their favorite parking lot.
5. The system should be a mobile application that supports both iOS and Android.
6. The system should be able to change the language



## 4.4 System Architecture



Figure 6. System Architecture

Let's review all of the components mentioned in the figure.

### 1. Application

- **React Native Map**

React Native Maps is a popular library that allows developers to integrate maps and geolocation features into their React Native applications. It is extremely helpful in our work. Whether it's displaying a map within our application for users to interact with or using the `<MapView>` component to configure various aspects of the map, it is convenient and highly efficient.

- **YOLOv8**

YOLOv8, or You Only Look Once version 8, is an advanced object detection algorithm used in computer vision tasks. It is part of the YOLO (You Only Look Once) family of models. YOLOv8 is known for its real-time object detection capabilities and high accuracy. It uses a single neural network to simultaneously predict object-bounding boxes and their corresponding class labels. YOLOv8 incorporates various optimizations, including feature extraction techniques and anchor-based predictions, to achieve fast and accurate object detection in images and videos. It is widely used in applications such as autonomous driving, surveillance systems, and object recognition tasks.

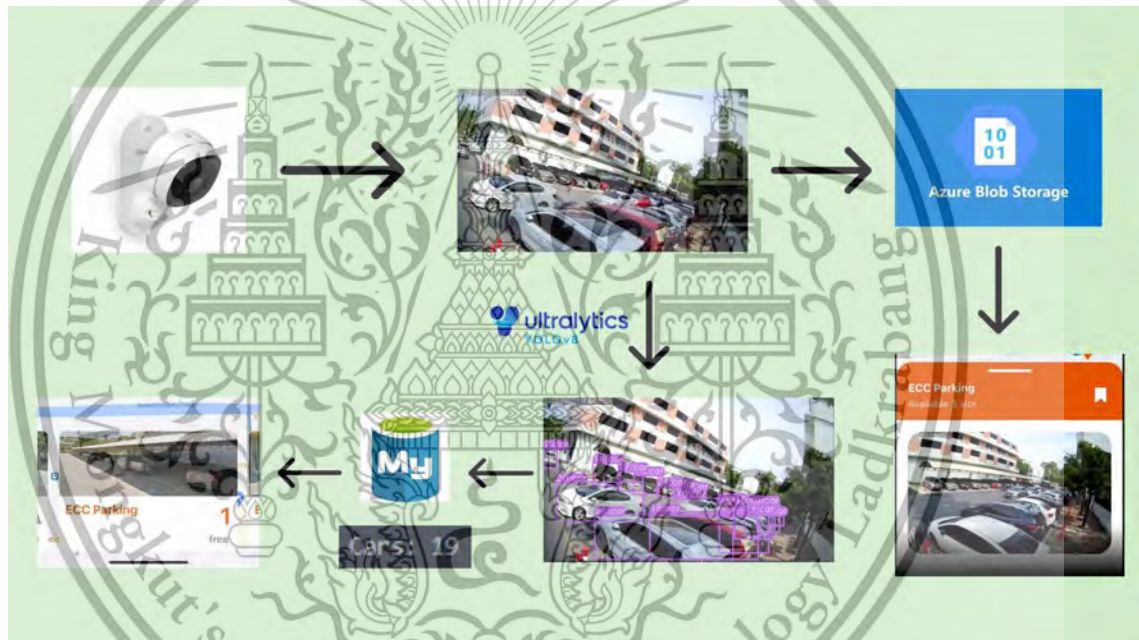


Figure 7 - CarDetect Algorithm

#### 4.5 System Components Design

KMITL Parking Lot is designed to be used by personnel within the university. Many features in the application are aimed at familiarizing users with various aspects,

such as setting familiar pin names based on the names of academic buildings, to enhance the efficiency of user experience.

#### 4.5.1 User Interface

We have designed the UI to be contemporary and user-friendly. Our emphasis is on simplicity and familiarity, providing users with a map interface that feels intuitive. As mentioned earlier, each pin (parking lot) is labeled with the names of buildings or names commonly used by students and staff, making it easier to understand. We have also designed the ability for users to save frequently used pins, allowing for easier access and convenience.

##### 1. User UI

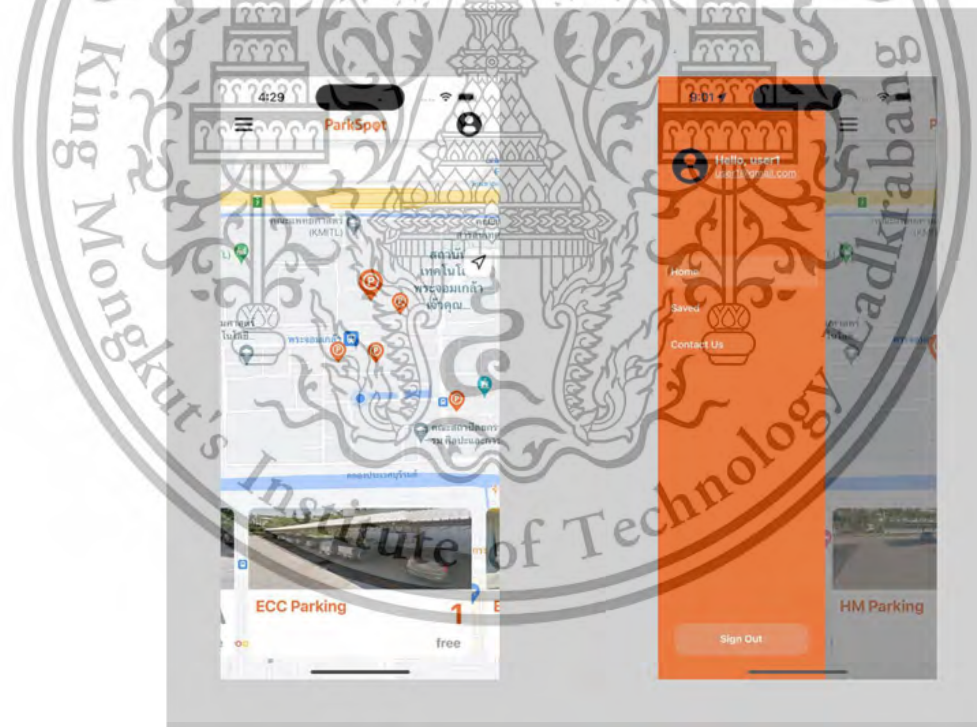
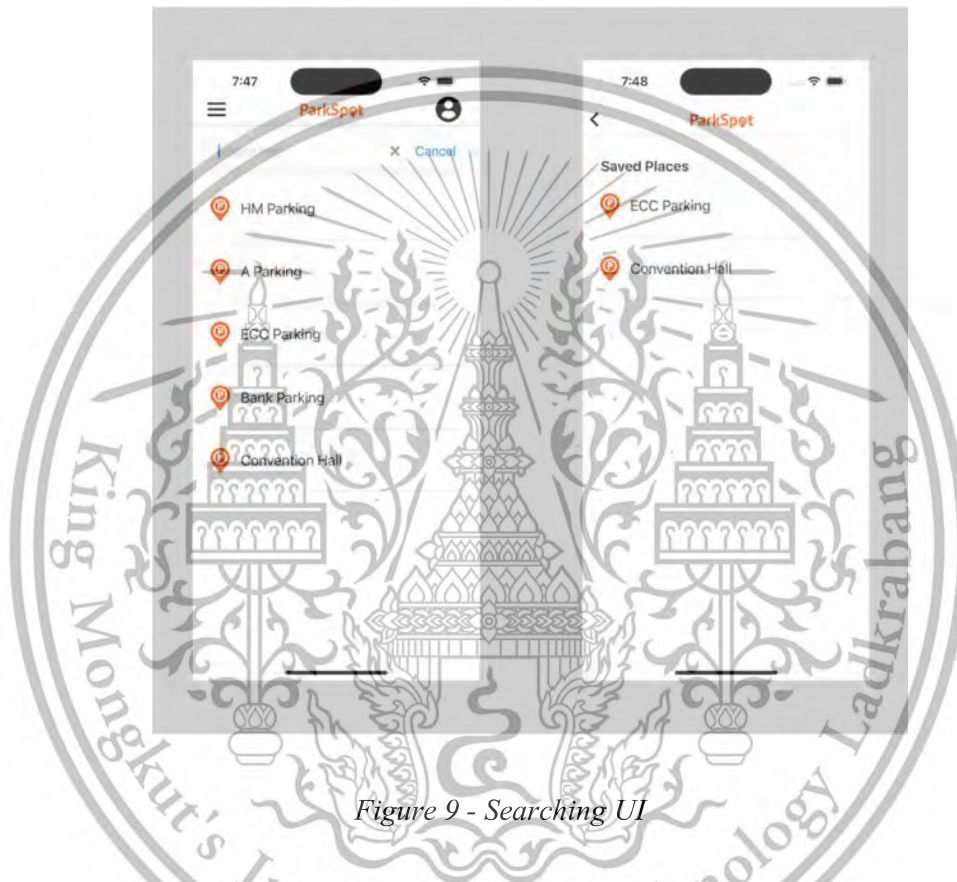


Figure 8 - Homepage UI

- We designed the sidebar to provide the most convenience for users.
- On the homepage, you will see a bottom strip of cards indicating whether each location has available space or not.
- The sidebar will have a "Save" function, where users can access and view the saved pins (Parking Lots) that we have stored



*Figure 9 - Searching UI*

- There is a search box available for searching and locating the pins you want to view the information of
- after we find the pin we want, we can click on it, and the application will take us to the information page of that pin

## 4.6 Class Diagram

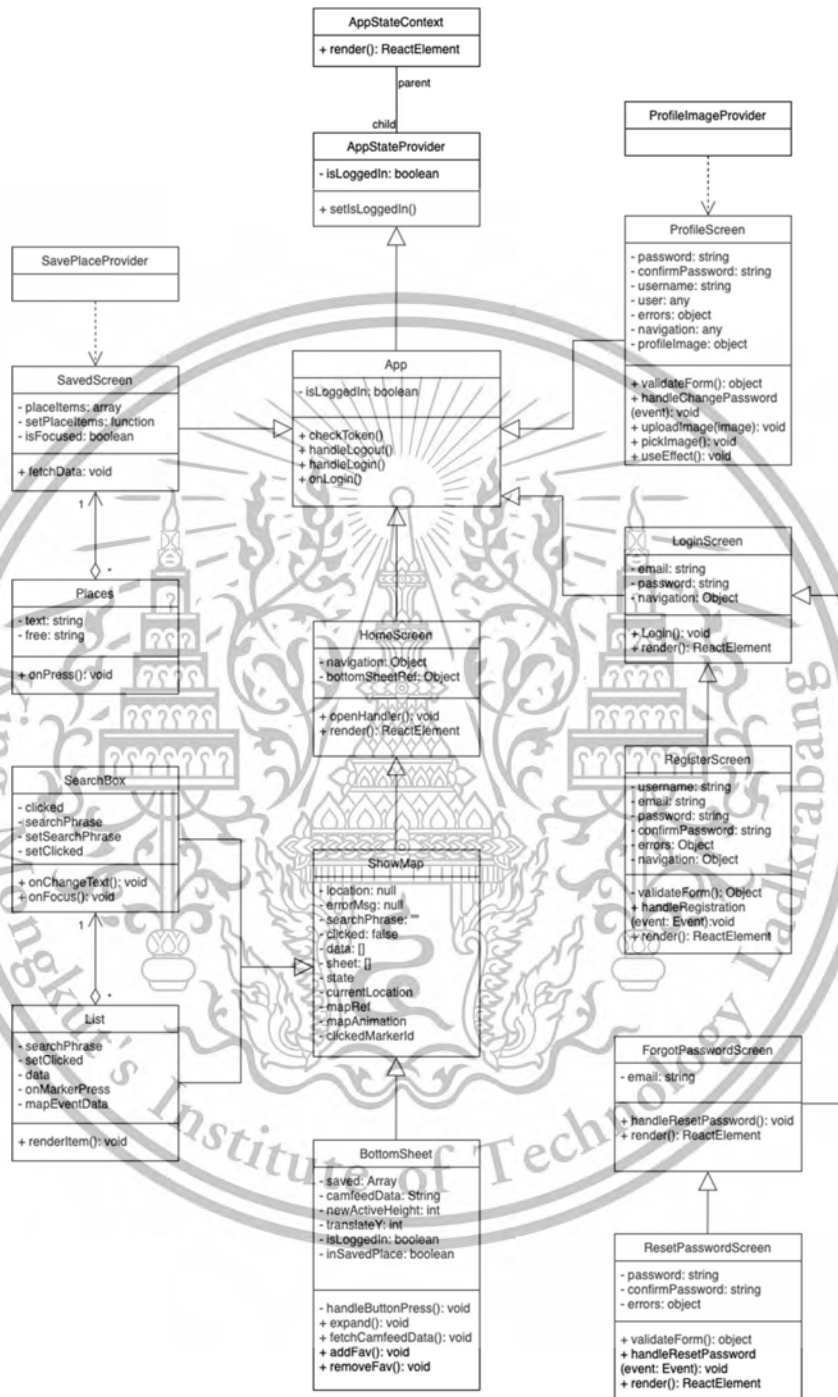


Figure 10 – Class Diagram

# Chapter 5

## Software Development

### 5.1 Development Tools

#### 5.1.1 Visual studio code

Visual Studio Code was selected as the IDE since it supports the majority of programming languages, including Java and javascript. We also connected Git to its built-in source control. It is easy to view the changes between the current version and the most recent committed version while developing with this IDE.

#### 5.1.2 GitHub

We used Github for team communication and version control. It allows us to create the software simultaneously. it offers a limit of three contributors per repository. Git includes the commands push, pull, commit, and clone to guarantee that everyone working on a project has access to the most current version.

#### 5.1.3 Figma

We used Figma to design a mockup version of our application. Figma allows multiple team members to work on the same design at the same time, making it easier to collaborate and stay on the same page.

### **5.1.4 Jira**

Jira is being utilized to track and keep an eye on the progress of our project. Jira Software is a member of a family of tools for managing projects in teams of various sizes. Jira was initially intended to be a bug and problem tracker. But now, Jira has developed into a potent task management solution for a variety of use cases, including agile software development and the administration of requirements and test cases.

## **5.2 Technology / Concepts used for the System Development**

### **5.2.1 React Native**

We used React Native because it's great for mobile apps. It provides a slick, smooth and responsive user interface, while significantly reducing load time. It's also much faster and cheaper to build apps in React Native as opposed to building native ones, without the need to compromise on quality and functionality.

### **5.2.2 OpenCV**

We used OpenCV to compute camera images to check if there is a car park available. then send it to work on the next part.

### **5.2.3 MySQL**

MySQL is free and open-source, which means that anyone can use it for their projects without having to pay any licensing fees. This makes it an affordable option for small businesses and startups.

### **5.2.4 Microsoft Azure**

We used Microsoft Azure because it allows you to quickly and easily scale up or down your computing, storage, and other resources as needed, without the need to purchase and maintain physical hardware. And we're not allowed to use Django.

## **5.3 Techniques / Algorithms used for the System Development**

### **5.3.1 Agile development : Scrum**

We apply Scrum knowledge to our projects, we have meetings every one-two weeks. We will talk about the work process that is carried out. and update the situation in the team.

## **5.4 Development Process**

Throughout the past 4 months, we have developed an application based on the designed blueprint. We have also started exploring various tools to maximize the efficiency of our application. Our focus has been primarily on frontend development to ensure that the app's appearance aligns with our design, as well as implementing the desired features. However, the most time-consuming aspect has been the backend, as we needed to store both user-side and camera-related data for processing and displaying on the application.

### **1. Starting on application (Jan):**

- We started frontend development in the first month, and we made efforts to find the most suitable maps solution to integrate. Eventually, we chose react-native-maps because it is the best option available. We clearly defined our roles and responsibilities and started working efficiently.

### **2. More discuss (Jan):**

- We encountered an issue in the first month where our discussions were not sufficiently clear to successfully complete the work. Therefore, we revisited the problems and goals to ensure everyone's understanding and realigned our understanding once again.

### **3. Backend designing (Feb):**

- After completing the frontend development to a certain extent, we started adding the backend to store various data of the application. Therefore, we proceeded to design the entire database.

### **4. The camera (Feb):**

- When we installed the cameras to monitor the number of cars in the parking lot, we encountered some issues. For example, we needed an image of an empty parking lot with no cars for accurate processing. However, initially, we couldn't find any such images without cars and we tried processing the available images to display in the application

### **5. Development and Integration (Feb-May):**

- We proceeded with the development of the application, with one person dedicated to working on the full backend while the rest of us collaborated on completing the frontend as quickly as possible. This allowed us to have time to address any potential issues that may arise. We worked well together and, for the most part, followed the planned timeline. Although there were some unexpected functions that took longer than anticipated, we managed to complete them successfully

### **6. Verification of Completion (late May):**

- Throughout the 4-month development period, we have been diligently working on this application. Along the way, we have gathered feedback from some users to make necessary adjustments and enhancements, aiming for the utmost completeness. Our goal is to ensure its efficiency and create a remarkable user experience for the KMITL Parking Lot app

# Chapter 6

## Results

### 6.1 User Interface

The UI outcome fulfills specific specifications and requirements. However, due to time limitations and the necessity to meet project deadlines, certain parts of the UI have undergone modifications to ensure timely completion and the ability to showcase its functionality but most of them are still the same as we planned.

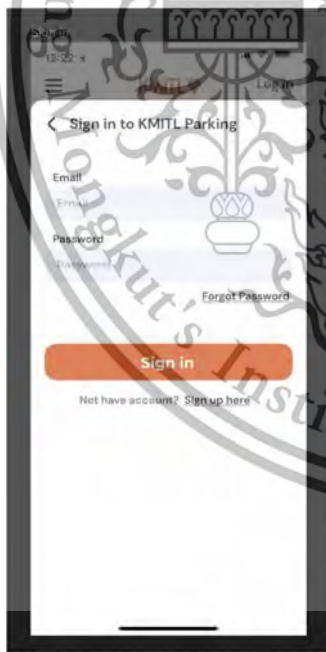


Figure 11 - Login UI

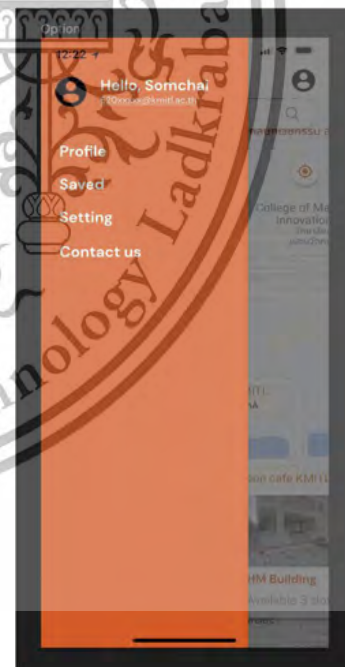


Figure 12 - Drawer UI

We have tried to make it as close as possible to the ideal prototype we designed. As you can see, nearly 90% of the design matches what we envisioned, and we are quite satisfied with it. There might be some differences in the appearance of the displayed cards, but they still function effectively.

### Main Menu Section (Signed Out)



Figure 13 - Home Screen UI (Signed Out)

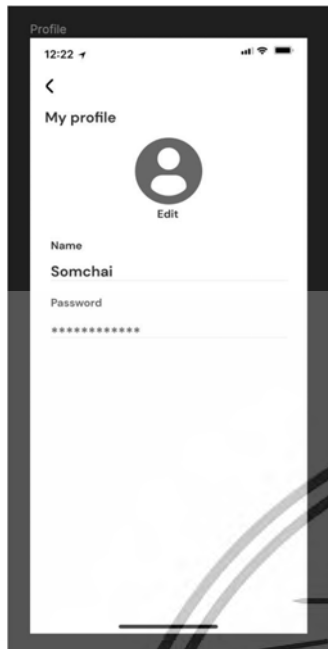


Figure 14. Edit Profile

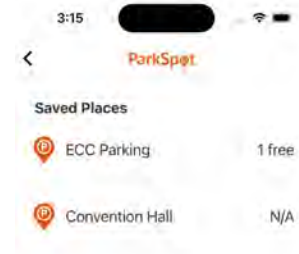


Figure 15. Saved Place

- When the user launches the app, the user is directed to the Main Home Screen, which shows the real-time location of the user along with the pin location of where the accessible parking lots are located and the ability to search them up in the search bar.
- There is also a Card list of parking lots in a row along the menu showing their names, how many spaces are left, and indicating if the parking lot has free space or not. But isn't accessible by the user who hasn't signed in yet.
- When trying to access the features, the user is then prompt to sign in to the Application.

## Sign In & Sign Up Section



*Figure 16 - Sign In & Sign Up Screen*

- When the user is prompted to the sign-in page, The UI for signing in is straightforward and simple for the user to understand.
- Sign In with Email. If the user doesn't have an account, the user then can sign up by entering their desired username, email, and password.

## Main Menu Section (Signed In)

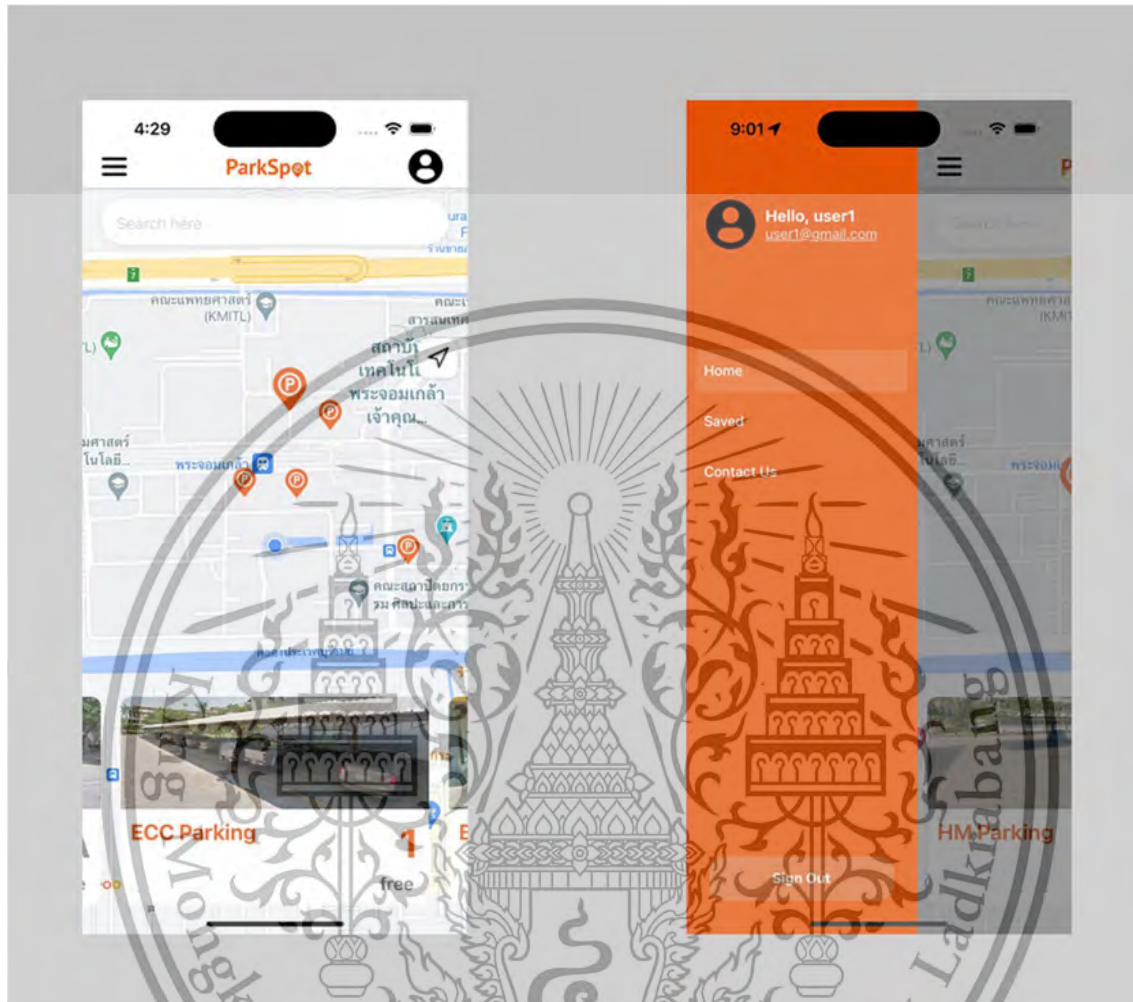


Figure 17 - Home Screen UI (Signed In)

- Once the user is signed in, The user is prompt to the Home Screen once again, with the ability to fully access the main features of the application,
- Users also have a saved area section where they can see and navigate to their specifically saved parking areas which will be shown in Figure x

## Search for Specific Parking Lot

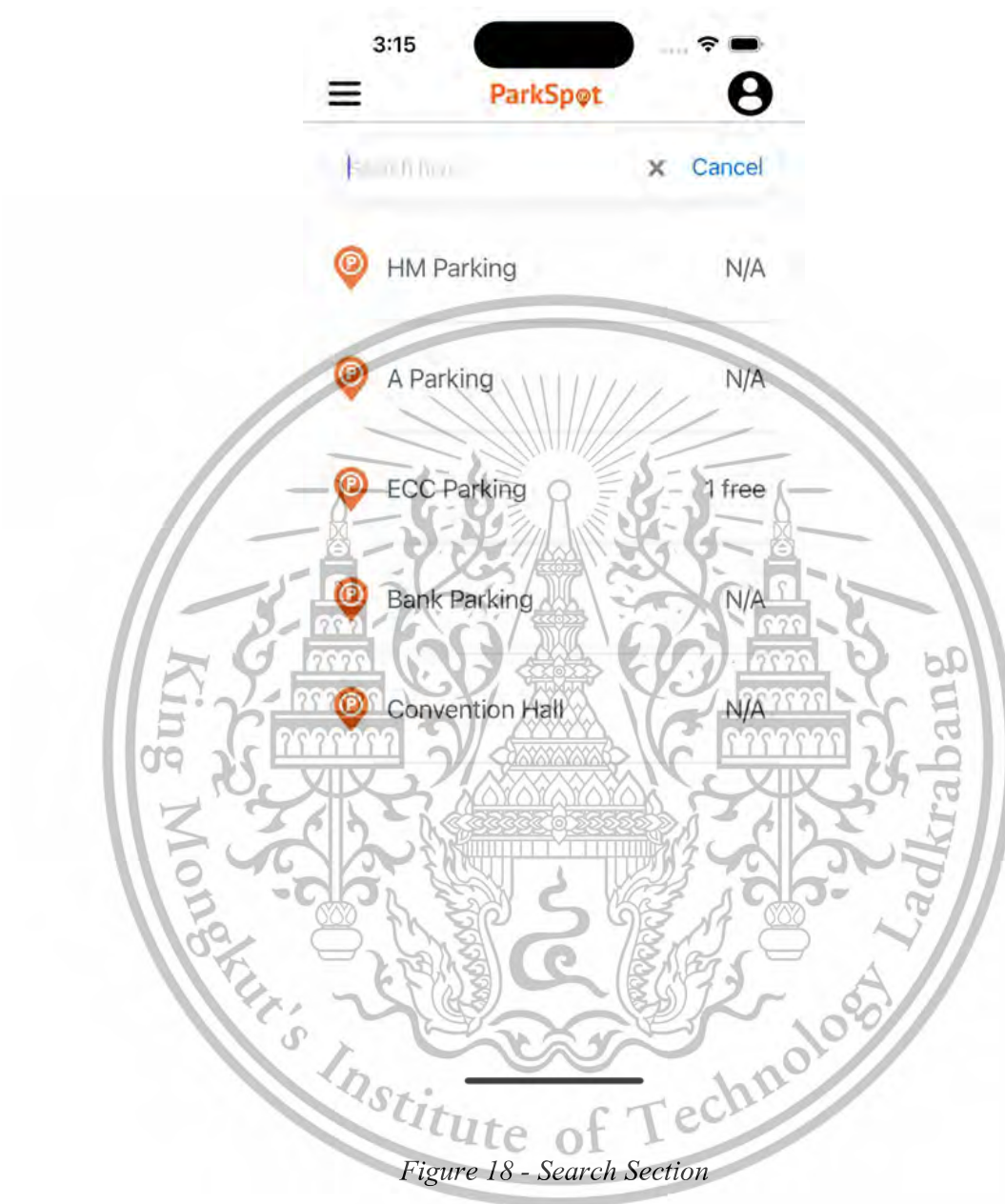


Figure 18 - Search Section

- If users want to search for a specific parking area they want to view or know its current location, the user can either go access the search bar and input the specific parking area they want to find.

- Another option is to access their saved places, where the user can tap on the specific saved parking area they want and it will navigate them to that specific parking area on the listed parking areas in the main menu and highlight its specific location.

### **Edit Profile Section**



*Figure 19 - Edit Profile Section*

- In this section the user can change and update their desired name, profile picture, and password. Upon update, the user then returns with their new name, profile picture, and as well a new password to access the application.

## View Parking via Camera



Figure 20 - Camera View Section

- In this section when the card container of the specific container is tapped, a bottom sheet would appear with the name of the parking area along with the camera view of that parking area.
- In this bottom sheet users also can click on the bookmark icon to add the specific parking area to their saved place, so they could access it easier on future use.

# Chapter 7

## Conclusion and Future Work

### 7.1 Summary

In conclusion, the Mobile Application for Parking Lot Finding Project succeeded to provide users the ability to check parking areas around the KMITL easier. By providing an extra eye for users to check specific parking areas in real-time and know how many spaces are left in those areas no matter where the users are located. This help users save time and efficiently find parking around the premises. Hence, the project eases the difficulty of parking around the premises.

### 7.2 Future Work

While the project achieved the features that we wanted it to have, there are some features we could've improved or implemented on our Parking Lot Finding Application.

- Implement a Visual Representation of each parking area, and show where each specific parking slot is free or taken by a car.
- Ability to guide the user's directions and show the distance between the user and the parking area.

By possibly implementing these features to our application, we believe that we could make the experience better for our users, with continuous development and updates. We can ensure that the project could help the community with parking at KMITL premises.

## Bibliography

[1] SpotHero - Best Parking Apps of 2022 - 10 Parking Apps to Help You Get the Best Spot. (2022). Available at: <https://smallbiztrends.com/2022/04/parking-apps.html> (Accessed: December 12, 2022).

[2] OpenCV - มาทำความรู้จักกับ Open CV กันเถอะ (2022). Available at: <https://kdbeer.dev/blogs/view/มาทำความรู้จักกับ-Open-CV-กันเถอะ-5eb961fbec76f75f73915b6e> (Accessed: December 12, 2022).

[3] IP CAMERA (IP CAMERA Outdoor 4213) - กล้องวงจรปิด Outdoor IP Camera ใช้งานภายนอก รุ่น 4213-Y App: YCC365 Plus (2022). Available at: <https://www.lazada.co.th/products/outdoor-ip-camera-4213-y-app-ycc365-plus-i3126625516-s11631452723.html?spm=a2o4m.tm80167383.6075338590.1.499figfjigfj32.499figfjigfj32&price> (Accessed: December 12, 2022).

[4] Microsoft Azure - What is Microsoft Azure (2022). Available at: [https://en.wikipedia.org/wiki/Microsoft\\_Azure](https://en.wikipedia.org/wiki/Microsoft_Azure) (Accessed: December 12, 2022).

[5] MySQL - What is MySQL (2022). Available at: <https://en.wikipedia.org/wiki/MySQL> (Accessed: December 12, 2022).

[6] React Native - React Native Learn once, write anywhere. (2022). Available at: <https://reactnative.dev> (Accessed: December 12, 2022).