Department of Software Engineering
Lakehead University

ESOF 4969 – DEGREE PROJECT
2022-23

Progress Report 1 – Software Requirements

Group 3

Contributing Members:
Nicholas Imperius
Jimmy Tsang
Mason Tommasini

Date Submitted: 11/30/22

Approval

This document has been read and approved by the following team members responsible for its implementation:

PRINT NAME: Nicholas Imperius
Signature:

Nicholas Imperius

PRINT NAME: Jimmy Tsang
Signature:

Jimmy Tsang

PRINT NAME: Mason Tommasini
Signature:

Mason Tommasini

COMMENTS:
Individual Contributions are as follows:
Nicholas Imperius: Introduction, Literature Survey, Business Model, Functional Requirements
Jimmy Tsang: Introduction, Literature Survey, Business Model, Non-Functional Requirements
Mason Tommasini: Introduction, Non-Functional Requirements, Business Model, Conclusion

Smart Parking Application: Software Requirements Document

Nicholas Imperius Jimmy Tsang Mason Tommasini

Abstract

Parking can be quite an annoyance in many situations, whether you are in a rush, the weather is bad, or if you don't want to park very far from your desired destination. In this document, you will find information on our project that has the main goal of improving this experience. A background study into the domain and existing research was concluded. Functional requirements of the application are presented, and the non-functional requirements are discussed. This document serves as the specification document for our mobile application that we will be designing.

Index Terms— Parking application, machine learning, mobile application, use case diagrams.

Contents

1 Introduction **3**

2 Background Research **3**
2.1 Domain Analysis . . . 3
2.2 Literature Survey . . . 3

3 Business Model **4**
3.1 Description of Technology & Environment . . . 4
3.2 Cost and Time of Development . . . 4
3.3 Societal and Economic Impacts . . . 5
3.4 Target User Base . . . 5
3.5 Competing Software Products . . . 5

4 Functional Requirements **5**
4.1 Use Case Descriptions . . . 6
4.1.1 View Application Settings . . . 6
4.1.2 Modify Name . . . 7
4.1.3 Modify Parking Permit . . . 7
4.1.4 Modify Parking Spot Availability . . . 7
4.1.5 View Campus Parking . . . 7
4.1.6 View Parked Car Location . . . 7
4.1.7 Filter by Type of Parking Spot . . . 8
4.2 User Interface Designs . . . 8
4.3 Database Design . . . 8

5 Non-Functional Requirements **8**
5.1 Performance Capabilities . . . 8
5.2 Scope of the Mobile Application . . . 9
5.3 Privacy, Security, and Safety Concerns . . . 9

6 Conclusion **9**

1 Introduction

Many students and staff can attest to the parking issues at Lakehead University. To provide a simple, fast and effective solution to this problem we plan to create a software application that makes finding a parking spot much easier. The software would be implemented as an application for smartphones that displays information, such as free parking spaces, busy times, and the location of your parked car. When arriving to the university, it can be difficult to find an open parking spot and frustrating when you are running late. This application would provide a solution to this issue and provide further benefits, such as saving the user time, reducing unnecessary stress, and helping with the environmental impact of vehicles. Our initial testing and training of this project will be focused to the G campus parking lot on Lakehead University but could be further expanded to other fields, such as public parking structures and mall parking lots.

This application will be quite different to other parking applications that exist currently. Instead of relying on sensory data or IP cameras to locate and detect empty parking locations, this application will use machine learning techniques to determine whether you have parked your car or not and then use GPS location information to notify other users that the parking space has been filled. Furthermore, when you have left your parking spot, our model will be able to accurately detect that and mark the parking space as free. Different machine learning classification methods will need to be tested to ensure we have chosen the most appropriate model. The machine learning model will predict based off your speed before and after a full stop, it should be able to recognize the difference between walking and driving which will allow it to determine the stopping location.

The rest of this paper is organized as follows: Section 1 provides an initial introduction to the project. Section 2 discusses the information retrieved a background study into the domain and any related work. Section 3 describes the general business or technology that will be used, societal and economic impacts, potential customers and environment, and competing software. Section 4 explains the functionality behind this project, specifically, the inputs, outputs, algorithms, data storage, and so on. Section 5 discusses the performance related parameters of the project, limitations, risks, and restrictions. Section 6 gives the conclusion of the paper.

2 Background Research

2.1 Domain Analysis

When researching similar ideas in the domain, we came across many unique solutions. Most solutions relied heavily on sensory data to determine if parking spaces were filled. This includes using IP cameras and object detection models or even IR sensors pointed at the parking space which will send alerts when a car is parked. Almost all interfaces consisted of a mobile application as this is the easiest way for a user when they are in a vehicle. It was found that there were not many ideas involving the GPS data of the user and using machine learning models to determine where they have parked, we think our solution will help bring research to this area since there exists a lack of information regarding this technique.

2.2 Literature Survey

With the uptrend of the number of vehicles being on the road, finding available parking locations has been a frustrating procedure for many commuters. Reducing the time it takes to find a parking spot which, in turn, increases productivity and creates more time for more important tasks. In this section, we explain current research in this domain, the methods being used, and the success rate of these methods.

Aftab *eh al.* [1] studied the impact of a smart parking application reducing the time it takes to search for a parking space along with the environmental impacts associated with it. They designed an android application called ParkUs and users would be able to select their destination, get directions, and confirm when they have parked. Their approach to determining if a user has parked revolved around looking at the timestamps from the cruising data they had, determining where stops were made and then choosing the closest parking location. They used this approach because the majority of the time, a user would choose the closest parking space to their location. They trained 3 different models, decision tree classifier, k-nearest neighbours, and a support vector machines. It was concluded that the support vector machine performed the best in their trial. They noted that their was a decrease in carbon emissions due to the lesser amount of driving that was being conducted as well.

How *et al.* [2] discusses the current problems surrounding parking infrastructure in cities and their parking structures or areas. They presented a solution that used QR codes in conjunction with a mobile application. Their solution could use license plate recognition to detect cars coming to park that have already booked the parking spot. In addition, if the weather conditions were not good, a QR code was present that could be scanned to confirm the parking location. This allows parking authorities to have a better understanding of which cars are allowed to be parked in the parking spots. Their goal was to lower the time spent searching for a parking spot.

Torres *et al.* [3] presented an e-recommended parking mobile application called P.E.T.E.R Parking. Their approach used rapid application development to obtain a business model canvas for a start-up project. Their application also incorporated user input to help improved recommendations too. The model that they created also aimed to improve revenue for the owners of the parking spaces and in their findings was able to achieve a rating of *highly usable* from the users and the parking lot owners had a positive response too.

Anand *et al.* [4] presented a real time system that allowed drivers to locate unoccupied through a web or android application. Their proposed solution incorporated a hardware and software components. The drivers would be issued a

RFID card that would be used when entering the parking premises and the information would be sent to their Raspberry Pi database. From here, a parking token would be granted to the user and directions to the parking spot would be provided through their mobile application. In the end, their solution reduced the time spent searching for parking spaces and helped reduced traffic congestion at peak times while being user friendly enough for anyone to operate.

Mahendra *et al.* [5] described a solution to car parking issues by utilizing Internet of Things (IoT) devices for inter connectivity of devices which would allow devices to gain access to surrounding objects. Their model was an IoT sensor that would be used in conjunction with a mobile application to allow users to pre-reserve parking spaces. They used IR sensors and a Raspberry Pi to compute if a parking space was taken; however, their system was designed for 3 parking spaces with each having their own IR sensor. Thus, a limitation of this would be scale and cost associated with large scale configurations.

Ng *et al.* [6] presented an outdoor parking space detection model that would be used through a mobile application. Their model used a CNN and images of the parking spaces through a security camera to determine if parking spaces were empty or full. They used a university parking lot as a testing facility for real-time vacancy detection. Their mobile application provided a visual for their users of the empty parking spaces and users could provide feedback on the models correctness which would help it learn for future reference. It was found that most users found the app useful in determining the best location to park which they treated as a success.

Tahmidul Kabir *at al.* [7] discussed a solution to recognize vacant parking spaces for areas where free parking can be an issue. Their solution used both hardware and software; a YOLOv3-tiny model was trained to detect vehicles based off the images from their hardware. On the software side, they used a web application for the client side and a SQL database for the server side. Users were able to utilize the model through a mobile application that would display a map with pins showing available parking or parking that can be pre-booked too. There system was fully automated for online function and while performed duties offline, required very minimal intervention from a human. Finally, their system had a low cost associated with it which helps with large scale implementation concerns.

Lewis *et al.* [8] presented a mobile application for saving parking locations of your vehicle. Their solution was a mobile application that would be able to provide other features, such as a compass or directions to their parking location. The model had to correctly identify when their vehicle parked as well as when it has departed the parking location as well. To track this they observed the user's speed throughout time and stored information whenever it thinks the user has parked. When trying to provide directions to the parking spot, the model would then look up the collected data and give the user the best results to find their parking spot. The model had correctness measured at 94%, which is practical enough for usage in the real-world.

Mangiaracina *et al.* [9] developed a simulated model of urban parking in Milan, Italy before and after their smart parking solution was implemented. For their Intelligent Transportation System (ITS) model, they implemented several technologies, such as GPS, WSNs, and Gateways, in order to accommodate the traffic of both truck and car drivers. They placed 80,000 sensors around parking spots in the city and developed a mobile application that would point users to the nearest possible parking spot available. It was found from their results that this implementation could save the average person around 77 hours per year and cut fuel costs by €86.5. Most importantly, CO2 emissions would be reduced by 44,470 tons per year in the city of Milan.

3 Business Model

3.1 Description of Technology & Environment

This project will consist of a mobile application that will be used on a smartphone that users will be able to download from their respective app store. The application will utilize the user's GPS location and other sensors for a precise reading. We plan to develop our mobile application with Visual Studio (VS) Code using the React Native language. Supplementary GPS data will be provided by Google Maps Application Program Interfaces (API's), and a cloud server through Firebase will need to be established. In the application, a birds-eye-view of the parking lot will be visible so users can get a better sense of the parking that they have available to them. A mobile internet connection will be required so the application can properly communicate with the server to allow for real-time parking availability information.

3.2 Cost and Time of Development

As of right now, we do not predict the development of our prototype will have a high cost. There are four main software components required for our project:

- Mobile Application IDE – Visual Studio Code
- Google Maps API
- Backend Infrastructure – Firebase with NoSQL

For our prototype, we do not expect to incur any costs throughout development. Most IDEs are free open-source software that do not require any costs for basic functionality. As for accessing the Google Maps APIs and Firebase, a Google based cloud server, Google provides a $300 credit for new users, which should cover all required costs.

Regarding commercial deployment, there will be some costs incurred. In order to publish our application, React Native requires a fee of $100 in order to place the application on the respective Application Stores. For commercial use, more requests to access Google Maps APIs and the Firebase cloud server are required, which should total around $50 per month for each application. Lastly, for the SQL database, a license may be required, which can cost up to $1000. A detailed breakdown of all pricing is found in Table 1.

Table 1: Cost Breakdown

Software	Prototype Cost	Commercial Cost
React Native	$0	$100
Google Maps APIs	$0	$50 per month
Firebase Cloud Server	$0	$50 per month
NoSQL Database	$0	$1000

Regarding development time, we expect the time to create the prototype to be around 60 hours, while supporting documentation will be around 20 hours. However, the amount of time dedicated to development will vary on a week-by-week basis, at the minimum, we plan to dedicate at least 3 hours a week to the development of the prototype and any required documentation. In Table 2, a detailed breakdown of estimated time for development is displayed. As the project progresses, an updated timeline prediction will be provided.

Table 2: Development Time Breakdown

Task	Predicted Time
Mobile Application	20 Hours
Database and Cloud Server	20 Hours
Testing and Alterations	10 Hours
Software Requirements Doc.	4 Hours
SPMP Doc.	4 Hours
Specifications Doc.	6 Hours
Software Design Doc	6 Hours

3.3 Societal and Economic Impacts

Ideally, this application will provide users with a quick and easy way to find a parking spot and alleviate much of the stress and wasted time that is associate with parking. Many different businesses and institutions can get busy, so implementing an interface to minimize issues with parking and blockage can have a huge net positive effect. By providing clients with an alternative to conventional parking, they would have a more enjoyable experience moving from their vehicle to their destination. The economic impacts would also play a huge role in the effectiveness of the application as users do not have to spend excessive amounts of money on gas driving around looking for a parking spot, especially since gas and inflation have spiked significantly this would save users a large amount of money. Carbon emissions would also be decreased as the vehicle would be driving around for less time.

3.4 Target User Base

Our user base would primarily be towards students and staff of Lakehead University to start, with future plans on expanding the applications capabilities to other parking structures. It would be our hope to allow the app to recognize the geo-location of a parking spot through a satellite view and then the application would be able to let users know ahead of time if parking is available. Of course this would need full participation by every person.

3.5 Competing Software Products

Currently, October 2022, there are no parking applications that attempt to locate available parking through machine learning methods, i.e., not using a camera with computer vision detection methods. There were products found that rely on sensory data, such as proximity sensors or IR sensors, to confirm if a car was parked in a location but this has a larger cost and increased maintenance to it. The computer vision solutions were quite well-done; however, there is no existing infrastructure that allows us to use currently installed cameras and it would not be within our scope to install new cameras on the property.

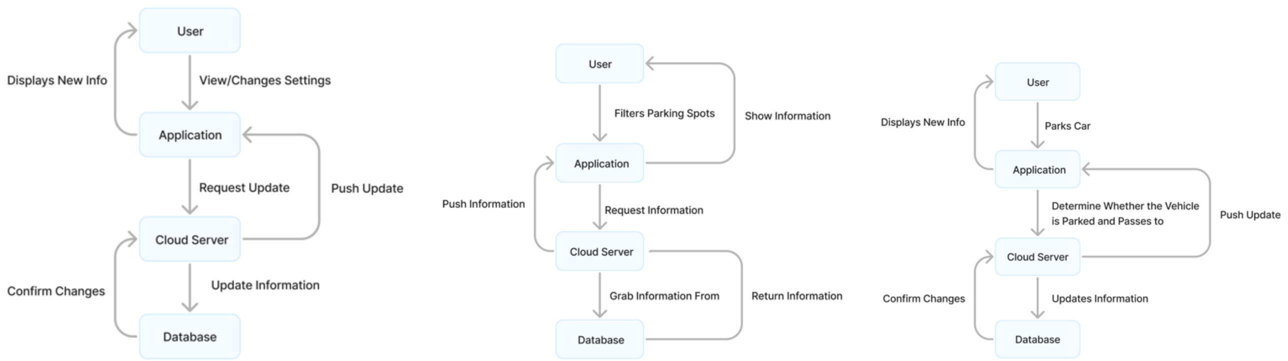
4 Functional Requirements

Functional requirements dictates the features that a system must be able to perform successfully. For our project, the following key functional requirements will need to be implemented:

- Display a birds-eye-view of the G parking lot on the campus of Lakehead University.

Figure 1: The geofence that will be surrounding the parking lot.

- Allow the user to zoom in or out on the map view to locate available parking locations.
- Users will be shown an indicator on each parking space in the parking lot to see if it is available or not.
- Users should be able to view the current parking location of their car.
- When a user has stopped driving, parked, and started walking, the machine learning model should be able to determine this and mark the location as not available.
- When a user has gotten in their vehicle and drove away, the machine learning model should be able to recognize this and mark the parking location as available.
- The mobile application will recognize when the user has entered in the geofence, as shown in Fig. 1, which will turn on precise GPS tracking.



(a) Block diagram depicting the update in the settings.

(b) Block diagram depicting the filter function.

(c) Block diagram depicting the process of parking a car.

Figure 2: The three main application functions are (a) the updating of user settings, (b) the filtering of parking spaces, and (c) the process of parking a car are shown in a visual format.

There are three main functions of the application, viewing the application settings, viewing the parked and unparked spots, and determining whether a parking spot has been taken. In Fig. 2a, the functionality of viewing and updating settings is depicted. In Fig. 2b, a block diagram depicting the process of filtering for parked and unparked spots is depicted. Lastly, we represent the process of determining whether a parking spot has been taken when a user parks their vehicle is depicted in Fig. 2c.

A use case diagram can be constructed from these functional requirements, as shown in Fig. 3. Descriptions of the use cases are described in Section 4.1.

4.1 Use Case Descriptions

4.1.1 View Application Settings

Enables the user to adjust their name and parking permit number directly inside the application:

(i) The user will click on the *gear* icon at the top right of the welcome screen.

(ii) The user will click on the name text field

(iii) The **Modify Name** use case is used.

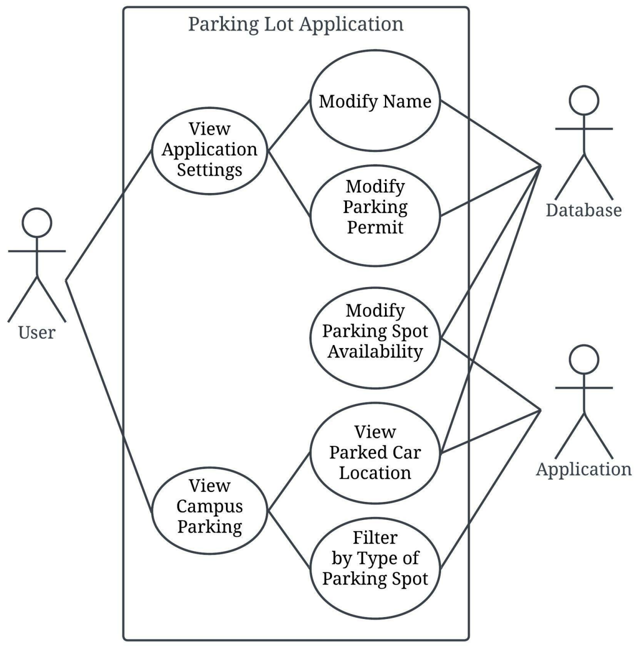


Figure 3: The use case diagram for our mobile application that displays the interactions between the user, database, and application.

4.1.2 Modify Name

Enables the user to have their name updated in the database and have these changes reflected in the application:

(i) The user will enter their name into the text field.

(ii) The application will send this information to the database.

(iii) The database with the new changes.

4.1.3 Modify Parking Permit

Enables the user to have their parking permit number updated in the database and have these changes reflected in the application:

(i) The user will enter their parking permit number into the text field.

(ii) The application will have the database updated with the new information.

4.1.4 Modify Parking Spot Availability

The application will update availability information once it has recognized that a user has parked or left their parking spot:

(i) The application will assess the user's speed before and after the stop.

(ii) If the model determines that the parking availability has changed, then update the database; otherwise, do nothing.

4.1.5 View Campus Parking

Enables the user to view an overview of the entire campus G parking lot that the application is supported for:

(i) Show a satellite view of the campus G parking lot along with the users current location.

(ii) The user will be able to zoom in and out on the map.

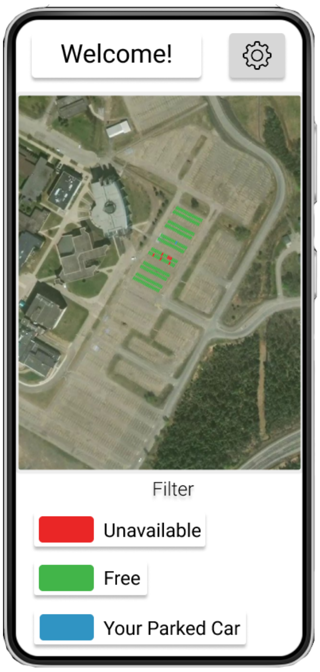
(iii) The user will be able to visually see which spots are available, taken, and their parking location.

4.1.6 View Parked Car Location

Enables the user to easily view where they parked their car on the satellite map:

(i) A visual indicator will describe the location of the vehicle to the user.

(ii) The user's current location will be shown to relate to where they are on the map.



(a) The mock user interface of the welcome screen.

(b) The mock user interface of the settings screen.

Figure 4: The two main user interface wireframe screens that are going to be implemented into the working prototype. In (a), the welcome screen is shown, and in (b), the settings screen is shown.

4.1.7 Filter by Type of Parking Spot

Enables the user to select what type of parking spots display on the satellite map image:

(i) The user chooses the filter they want.

(ii) The application will adjust the satellite image accordingly with information received from the database.

4.2 User Interface Designs

In Fig. 4a, the welcome screen is displayed. This is the main screen that the user sees once they open the application. We intend to each have each sub-lot of the parking lot clickable and then the map would zoom into that section, providing an easier to understand view for the user. On this screen there is a button in the top right that allows the user to go to the settings screen and three filter buttons at the bottom. These filter buttons allow the map to select only the specified type of parking spots to be shown, allowing for a better ease of use for the user.

In Fig. 4b, the settings screen is shown. This is where the user will be able to update their name and parking permit number, which will be stored as strings in the database. There will be a text field for the name and permit number fields, and a return button that will redirect the user to the welcome screen again.

4.3 Database Design

Our database will be designed based on the entity-relationship (ER) diagram in Fig. 5. From this diagram, we can see that each user can only have one parking space and one parking space can only be occupied by a single user. In addition, parking sub lots can have many parking spaces but not none, while each parking space can only belong to a single parking sub lot. Lastly, a parking lot must have at least one sub lot while a sub lot can only belong to a single parking lot. In Fig. 5, primary keys are identified with an underline notation and their variable types are listed as well.

5 Non-Functional Requirements

5.1 Performance Capabilities

There are many parameters that can affect the performance of the prototype, such as the capabilities of the user's smartphones. All smartphones are different, and they may have varying degrees of accuracy when it comes to global positioning. There could be devices that have a faulty magnetic sensor, which affects the user's orientation; however, as Google Maps relies on GPS data rather than then directional data from the phone's built-in magnetometer, this can be mitigated. In addition, since the application will be accessing a cloud server, an internet connection would be required for a real-time display of data.

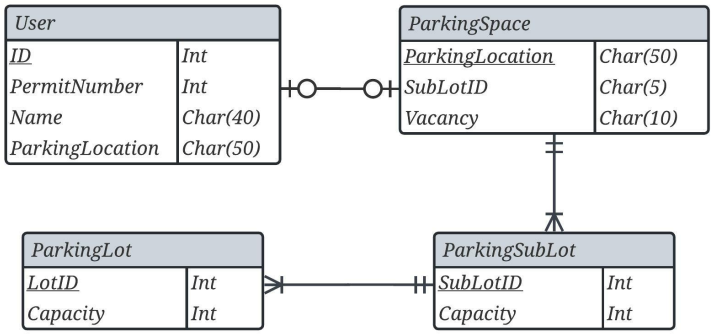


Figure 5: An entity-relationship diagram for our database. It is shown that the user and parking space table have a one-to-one relationship, the parking space and parking sublot of a one-to-many relationship, and the parking lot to parking sublot have a one-to-many relationship too.

5.2 Scope of the Mobile Application

As of right now, we are limiting the scope of our project to the G parking lot of Lakehead University. With a smaller scope, we will be able to implement our prototype without having to adjust the parameters of the parking lot depending on the size and location of the area. Future work involving this project would be to expand its functionality to other commercial parking areas, such as shopping centres or airports. There is a noticeable limitation of our device, all users require a functional smartphone with a mobile internet connection for real time communication with the server and application. Google Maps and GPS would still work properly without a mobile internet connection, although, to send a signal to the prototype that a parking spot is taken, an internet connection is required. However, in a study conducted by the Pew Research Center, it was found that 97% of college students own a smartphone device, with many of those having access to mobile internet [10]. Additionally, another limitation placed on our prototype involves public acceptance. If members of Lakehead University are unaware or unwilling to download our application, then this application will not run as expected. As this is a prototype, it is not a limitation as of now; however, if this product is to be commercially developed, it would require 100% user participation. The accuracy of our application is based on full participation, otherwise, we may run into false negatives where parking spots may be labeled as empty when they are full. This can lead to users expecting a spot to be vacant when it is in fact full; thus, the with a lower participation rate, comes a lower accuracy.

5.3 Privacy, Security, and Safety Concerns

There are some concerns of privacy since the mobile application would require information about the users precise GPS location. A possible remedy that may correct this issue involves having the application only use a relative or non-precise position until the user enters a geofence around Lakehead University, shown in Fig. 1, at which point the application will switch to precise location tracking. Another possible solution would be to have the application track the user's location only when the application is in use, and not when it is running in the background. As this is a prototype, a closed application, there is no concern for safety and security as of now. Although, in the future, if this idea is brought to market, security measures would need to be put in place to protect the confidentiality and privacy of the users.

6 Conclusion

By creating this software, we hope to provide a useful application to many people to assist in their parking troubles. The parking application will be applied mainly to the G parking lot at Lakehead university, so it would provide the most benefits to students and staff that use the parking lots regularly. The prototype will consist of a cloud server network and machine learning model connected to our application that will correctly identify available and taken parking spaces. This will be done by determining when a user has stopped driving, parked, and began walking to their destination. An intricate classification model will be created that would allow an automatic way of designated a parking space as filled or empty. As our main target environment will be Lakehead University, this software comes with many opportunities to expand its application on to other fields of parking as well. Thorough background research was conducted in order to understand the domain of the problem as well as look at pre-existing literature regarding similar implementations using location based tracking and how parking can be tracked in a given area. To develop the application, we must understand the business side of the implementation and define requirements such as the project development timeline, target user base, and pre-existing or competing products currently in the field. The requirements of this project need to be understood so that both functional and non-functional requirements are addressed. Furthermore, any concerns or risks must have a plan set in motion to mitigate or reduce the amount of risk by as much as possible.

Acknowledgment

We would like to express gratitude towards the department of Software Engineering at Lakehead University for this opportunity and Dr. Thangarajah Akilan, Chair of Software Engineering Department, for being our project supervisor.

References

[1] K. Aftab, P. Kulkarni, I. Shergold, M. Jones, M. Dogramadzi, P. Carnelli, and M. Sooriyabandara, "Reducing parking space search time and environmental impacts: A technology driven smart parking case study," *IEEE Technology and Society Magazine*, vol. 39, no. 3, pp. 62–75, 2020.

[2] C. C. How, I. Farhana Kamsin, N. K. Zainal, H. Aysa Abdul Halim Sithiq, and N. A. Abd Rahman, "Smart Parking Reservation Mobile Application," *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, 2022.

[3] C. L. Torres, V. R. Torres, M. C. Estira, J. D. Osias, and J. M. De Los Reyes, "P.E.T.E.R. parking: Park easier through e-recommended parking mobile application," *2021 1st International Conference in Information and Computing Research (iCORE)*, 2021.

[4] A. Anand, A. Kumar, A. N. Rao, A. Ankesh, and A. Raj, "Smart parking system (S-park) – a novel application to provide real-time parking solution," *2020 Third International Conference on Multimedia Processing, Communication & Information Technology (MPCIT)*, 2020.

[5] B. M. Mahendra, S. Sonoli, N. Bhat, Raju, and T. Raghu, "IOT based sensor enabled smart car parking for Advanced Driver Assistance System," *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 2017.

[6] C.-K. Ng, S.-N. Cheong, E. Hajimohammadhosseinmemar, and W.-J. Yap, "Mobile Outdoor Parking Space Detection Application," *2017 IEEE 8th Control and System Graduate Research Colloquium (ICSGRC)*, 2017.

[7] A. Z. Tahmidul Kabir, A. M. Mizan, P. K. Saha, M. S. Hasan, and M. Pramanik, "An IOT based intelligent parking system for the unutilized parking area with real-time monitoring using mobile and web application," *2021 International Conference on Intelligent Technologies (CONIT)*, pp. 1-7, 2021.

[8] J. Lewis, R. Abbas, S. Reisenfeld, and S. Maric, "Little locator: A smartphone parking-location application," *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, 2018.

[9] R. Mangiaracina, A. Tumino, G. Miragliotta, G. Salvadori, and A. Perego, "Smart Parking Management in a smart city: Costs and benefits," *2017 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, 2017.

[10] Silver, "Smartphone ownership is growing rapidly around the world, but not always equally," *Pew Research Center's Global Attitudes Project*, 25-Aug-2020. [Online]. Available: https://www.pewresearch.org/global/2019/02/05/smartphone-ownership-is-growing-rapidly-around-the-world-but-not-always-equally/. [Accessed: 01-Oct-2022].