System Design for Park'd

Team #29, caPstOneGroup
Albert Zhou
Almen Ng
David Yao
Gary Gong
Jonathan Yapeter
Kabishan Suvendran

April 5, 2023

1 Revision History

Date	Version	Notes
Jan 18, 2023	1.0	Revision 0
Apr 3, 2023	1.1	Revision 1

2 Reference Material

This section records information for easy reference.

2.1 Abbreviations and Acronyms

See SRS Documentation at Park'd Software Requirements Specification.

Symbol	Description
Park'd	Parking Lot Application

Contents

List of Tables

List of Figures

3 Introduction

This System Design document provides an overview of the design of the Park'd system. A detailed specification and guide of the system modules can be found in the Module Interface Specification and Module Guide documents, respectively.

4 Purpose

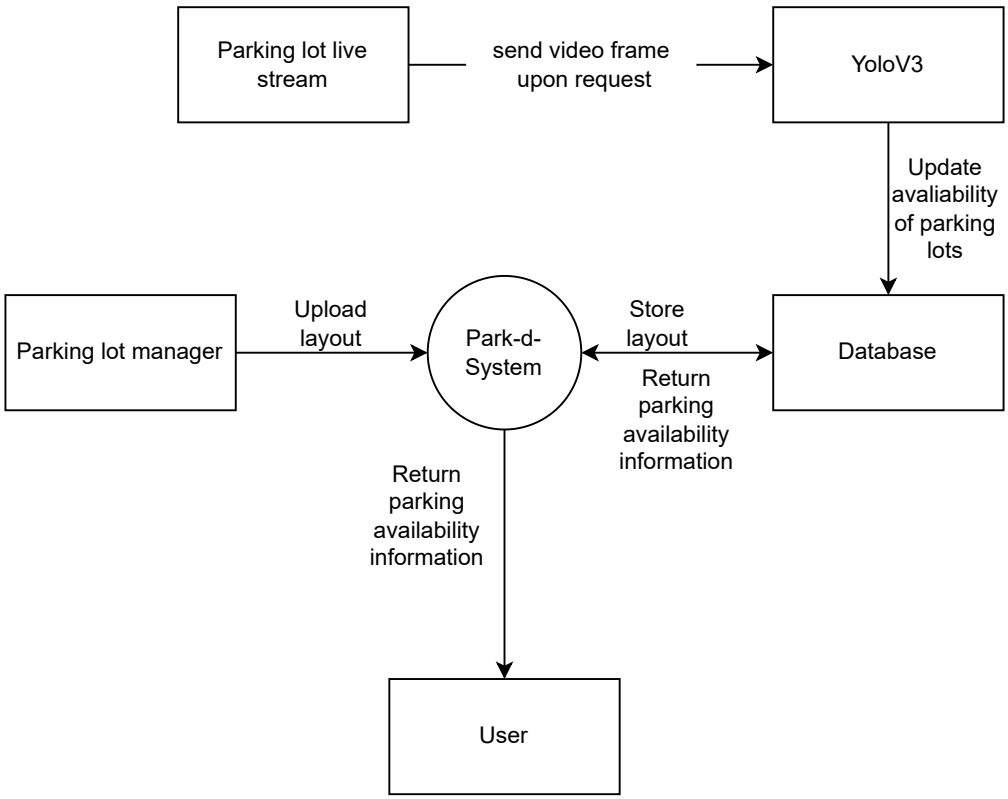
The purpose of our system is provide a way for users to quickly and conveniently identify available parking spaces, thereby reducing frustration. The system also provides instructions for the user to navigate to their desired parking space. In addition, the administrator for the parking space has the ability to modify certain elements of the parking space, such as the layout and which parking spaces are marked as reserved or accessible.

The purpose of this design documentation is to identify the overall system components of our project, their behaviours, and how they will work with one another to comprise the project as a whole. Furthermore, we will use this report to discuss how error handling will be implemented to account for undesired behaviours.

5 Scope

The system is meant to direct drivers to open parking spots. The driver will provide their location and a layout of their parking lot is displayed. Drivers will select an open spot and follow directions to it.

Figure 1: Context diagram



6 Project Overview

The following section is to outline the normal behaviour of Park'd and undesired event handling.

6.1 Normal Behaviour

Normal behaviour entails regular users being able to sign in, so that parking lots that are shown to them are tailored specifically to their profile, which consists of their vehicle type and their parking preference (regular or accessibility). Afterwards, users can select available parking spaces and then receive instructions to arrive at the selected parking space.

For administrators, the experience involves the ability to modify the layout of the parking space and designating certain parking spaces as either reserved or accessibility parking.

6.2 Undesired Event Handling

For the normal behaviour described above, there are many points of failure, which can result in undesired events. For example, the user may not be able to log in due to database connection issue. In this case, we would allow the user to still use the application, but the parking spaces shown to them may not be necessarily tailored to their needs. Another example is when the parking directions that are provided for a specified parking space are blocked by obstacles. To prevent this issue, there will be multiple algorithms that all compute the parking directions using different techniques.

In general, undesired events will be handled using multiple algorithms, which build upon the redundancy principle, or by using software that will still function if back-end features do not work correctly.

6.3 Component Diagram

The following is a component diagram of the Park'd system.

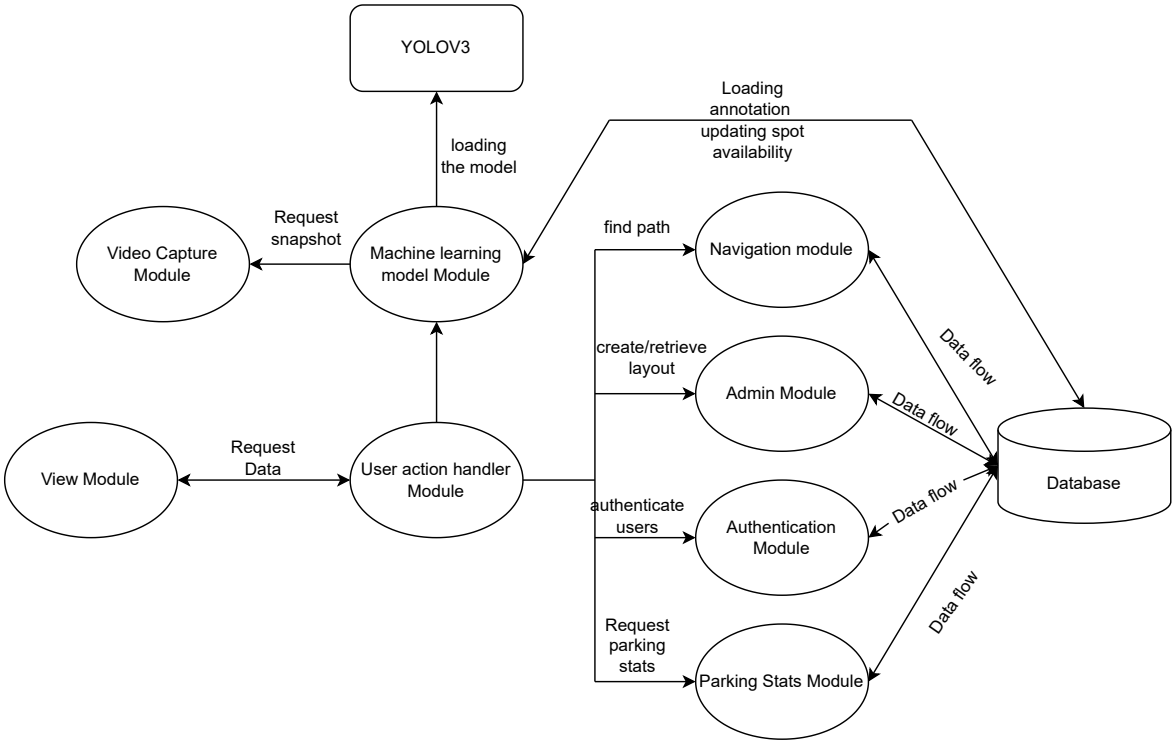


Figure 2: Component Diagram

6.4 Connection Between Requirements and Design

The following table is to document decisions that are made between the requirements and the design.

Table 1: Design Decisions and Requirements

Design Decision	**Requirements**
The software sends a prompt asking for users' current location information in order to use it for navigation	FR1, FR2
A sign-up function will be considered in future revisions as currently, there are privacy and security concerns regarding sensitive information being leaked out. This feature has been implemented for admins using Firebase Authentication. FR2	FR31-FR37
Admin console is created to allow parking lot administrator to recreate the parking lot layout and thus provide information for users later when they are looking for parking spaces	FR5, FR6, FR7, FR31-FR37
Mapping API is used to display routing information as clearly as possible to the user	FR11, FR13
The time interval between every update of the parking lot state and display must be short	FR17, FR18, FR20
The information of the parking lot provided to user should also be available to the parking lot admin	FR26, FR27, FR28, FR29

7 System Variables

This section is not applicable to software engineering projects.

7.1 Monitored Variables

N/A

7.2 Controlled Variables

N/A

7.3 Constants Variables

N/A

8 User Interfaces

Figma was used to prototype the user interface. The user interface consists of three different parts - Login Screen, User View, and Admin View.

For the Login Screen, the design allows both parking lot managers and users to login to an account on the system and prompts them to provide the appropriate information to do so (Figure 4).

With the User View, users are able to see parking lots nearby and their availability, select a parking lot, select a parking spot to navigate to, as well as look at the parking lot analytics (Figure 7, Figure 8).

Lastly, for the Admin View, admins are also able to see the analytics of their parking lot including how many available parking spaces there are, the division of parking spaces based on specialty spaces, as well as make edits to the parking lots they own (Figure 5, Figure 6).

9 Design of Hardware

This section is not applicable to the Park'd project.

10 Design of Electrical Components

This section is not applicable to the Park'd project.

11 Design of Communication Protocols

This section is not applicable to the Park'd project.

12 Timeline

A timeline of tasks and responsibilities leading up to the Revision 0 Demonstration can be found in Figure 3 below.

Figure 3: Timeline

A Interface

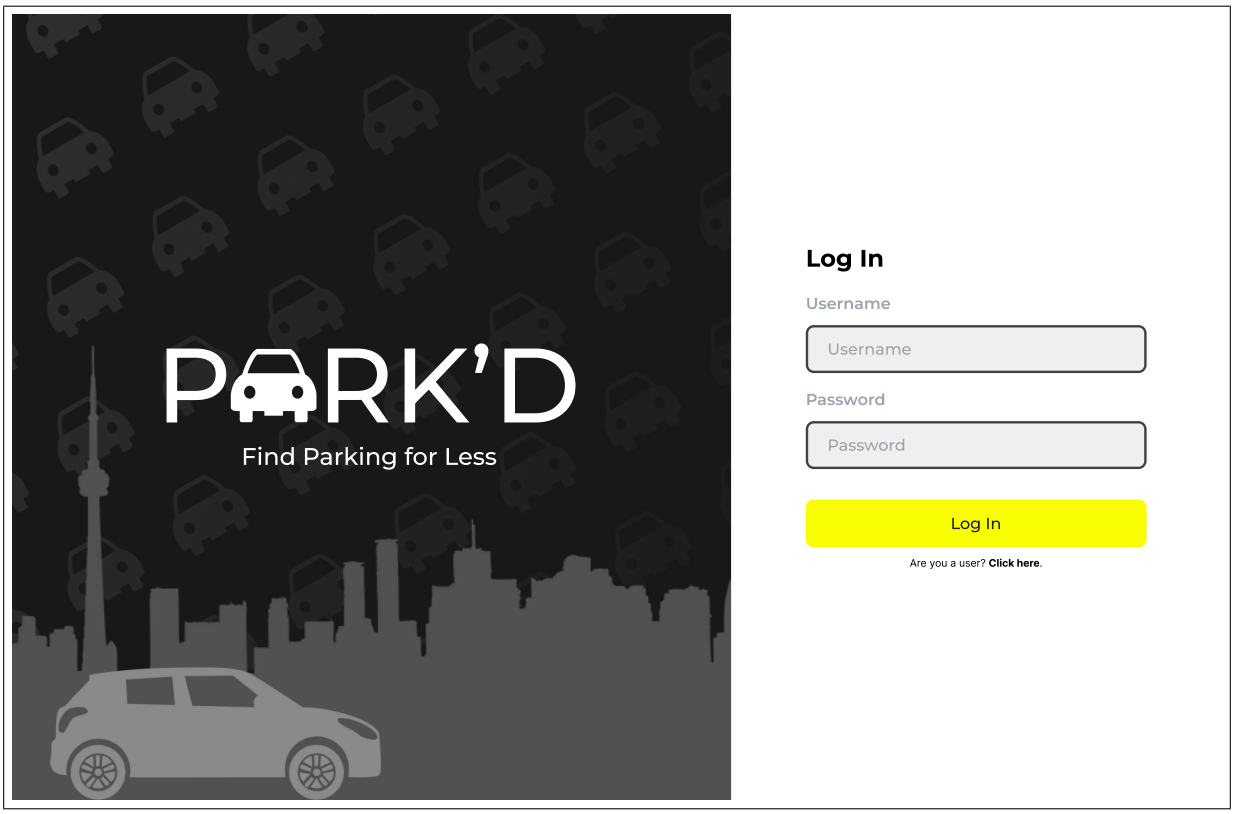


Figure 4: Login Screen

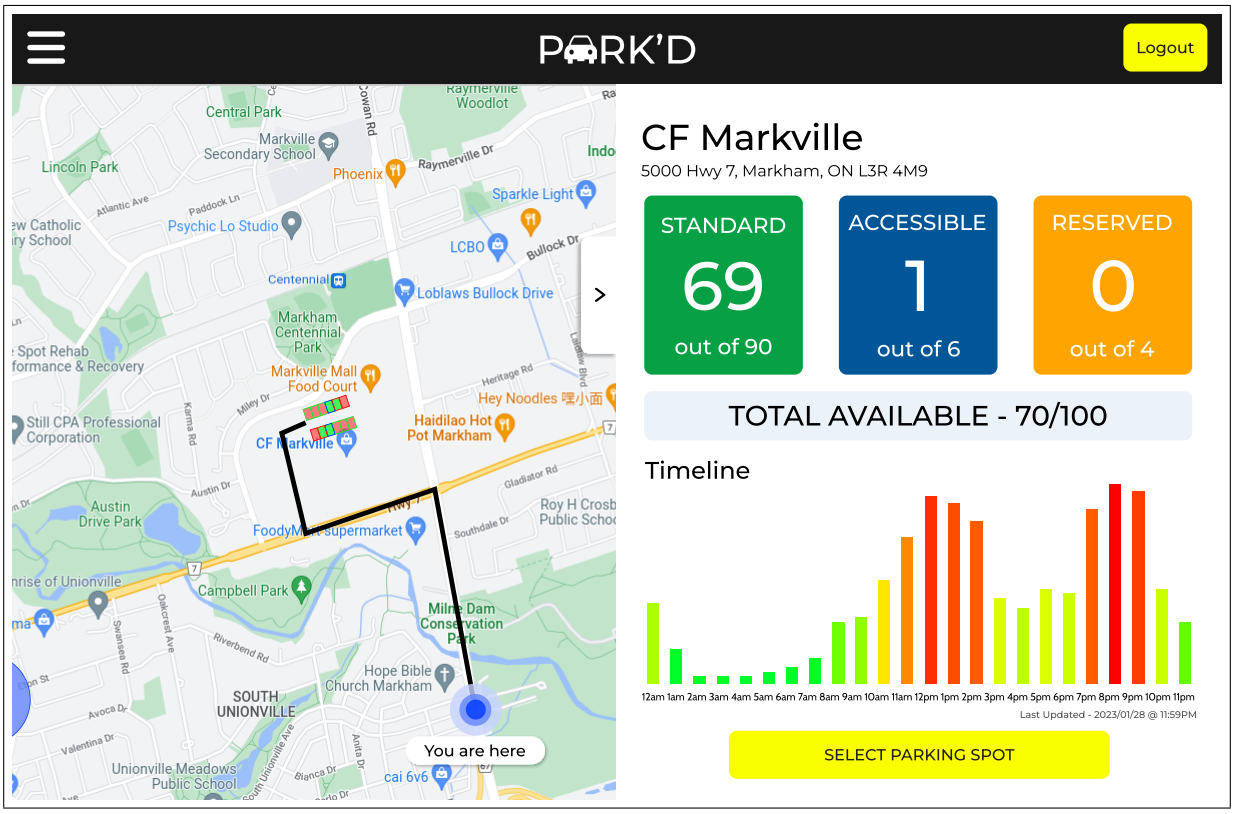


Figure 5: Admin View - Analytics Screen

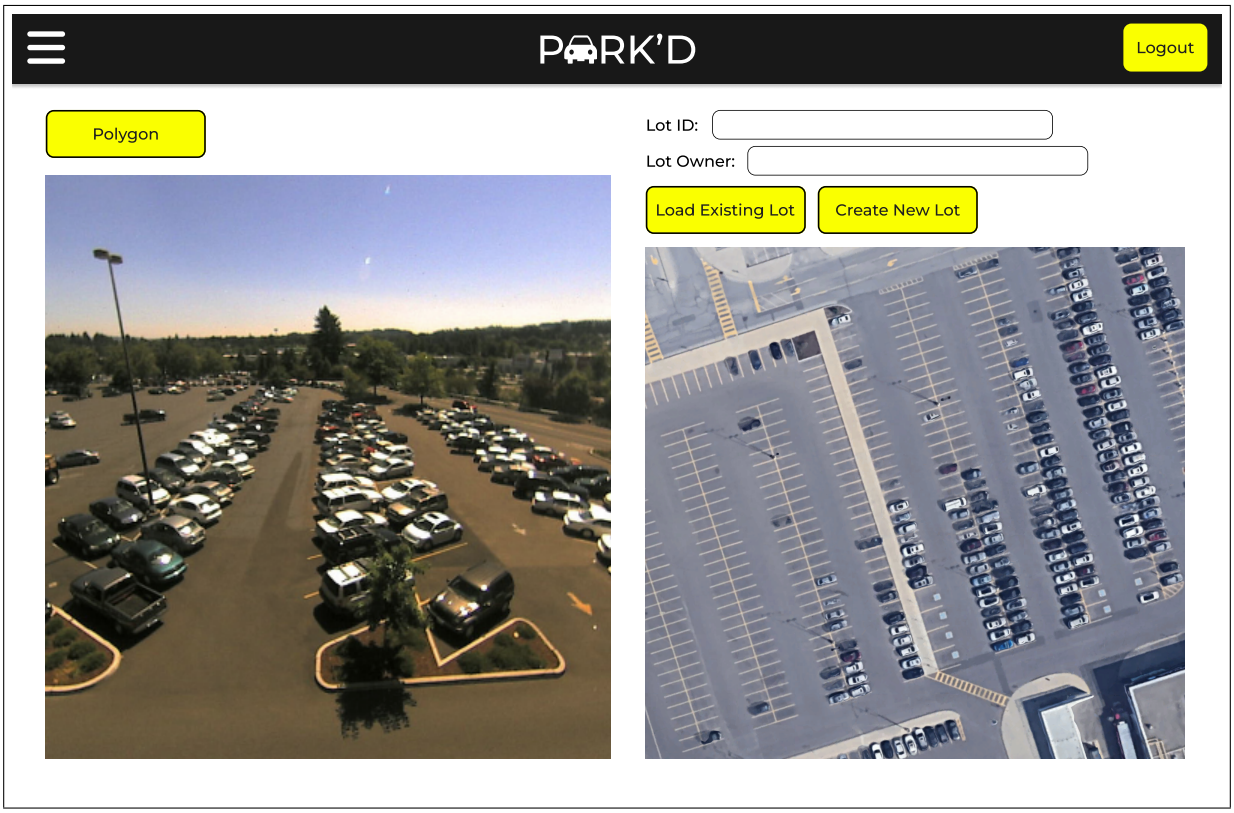


Figure 6: Admin View - Edit Screen

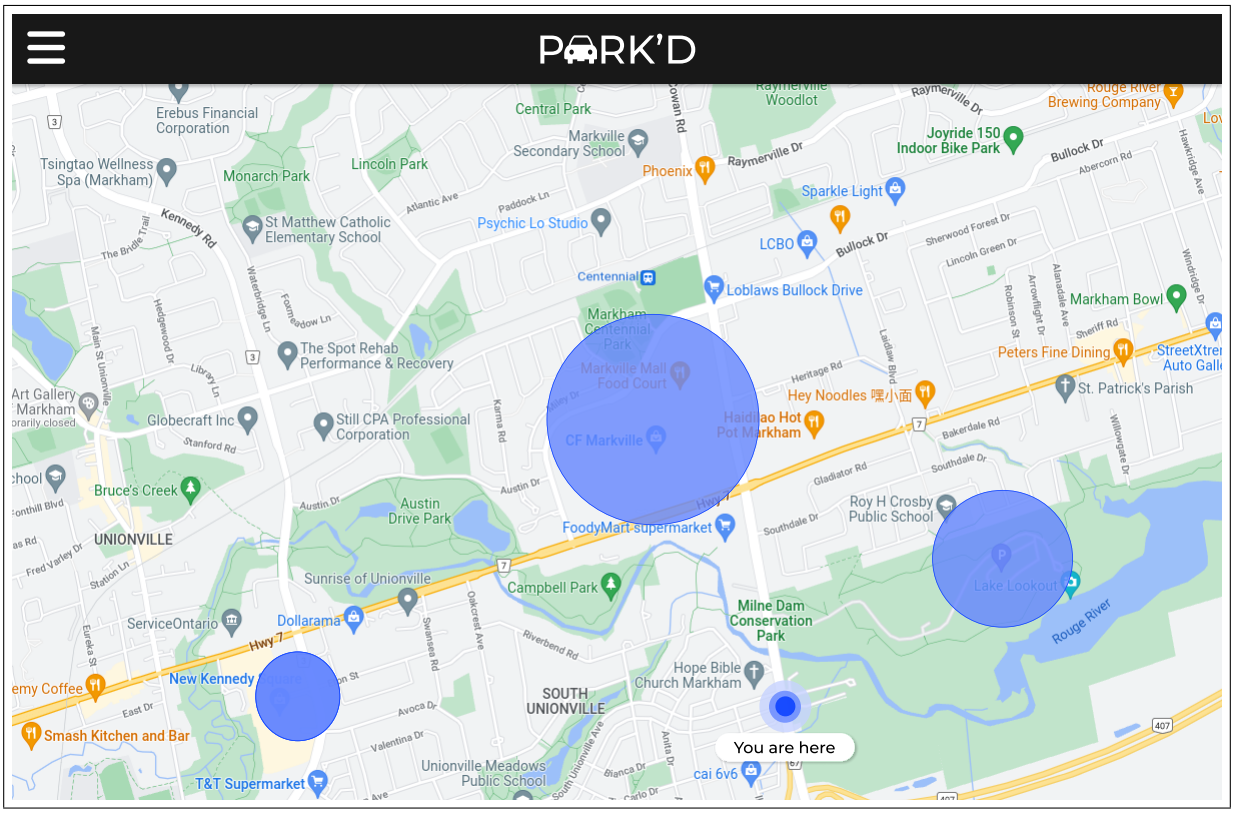


Figure 7: User View - You Are Here

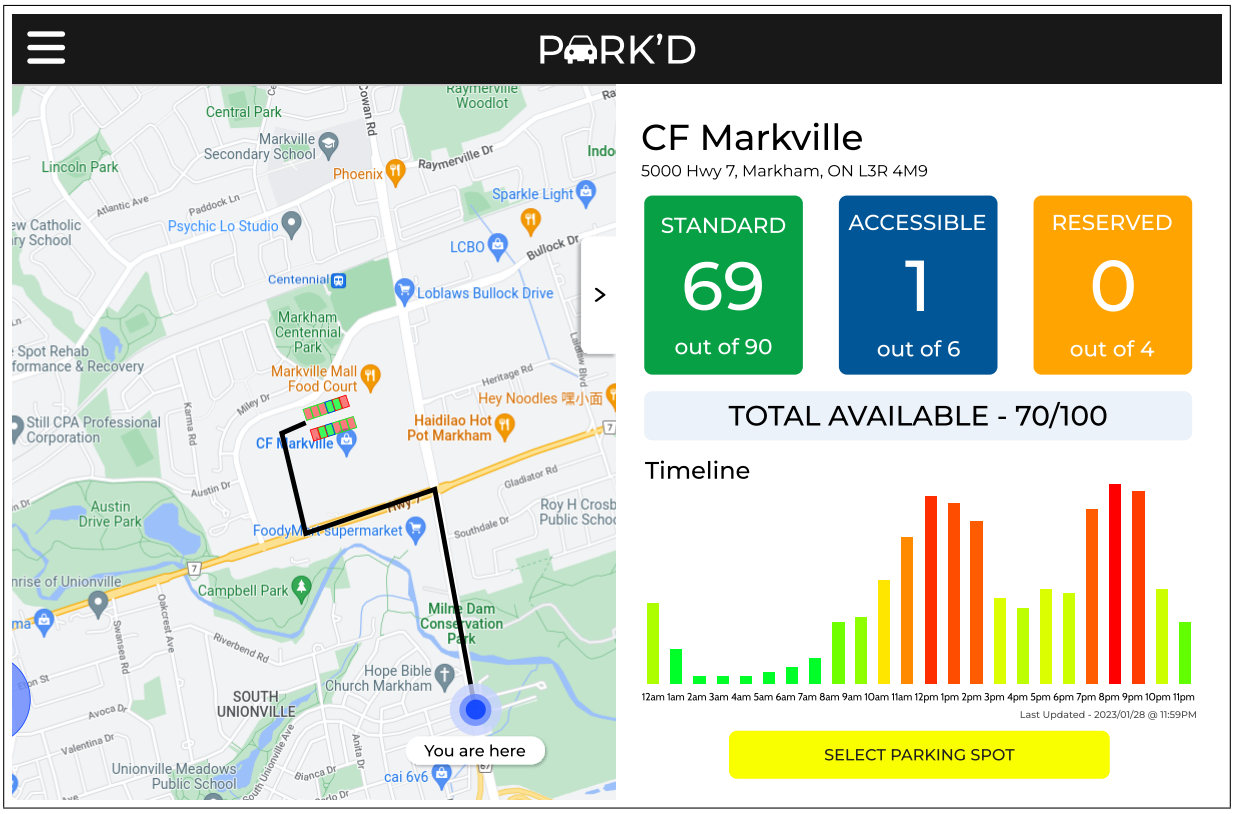


Figure 8: User View - Analytics Screen

B Mechanical Hardware

N/A

C Electrical Components

N/A

D Communication Protocols

N/A

E Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Problem Analysis and Design. Please answer the following questions:

1. What are the limitations of your solution? Put another way, given unlimited resources, what could you do to make the project better? (LO_ProbSolutions)

    (a) Our navigation system is simple and intended only to get the user within the general vicinity of their parking space, with the assumption that they will be able to locate it from there. The plan is to use a pre-existing mapping API, which for most parking lots will only map the main lanes. Given additional resources it may be possible to implement a more granular navigation, akin to those used in malls to locate specific stores, for example.

    (b) The current solution for determining the layout of a parking lot relies on an administrator to specify relative locations for each space. The back-end model is only intended to poll spaces for their current status, not to determine where those spaces are located. Extra time and resources could allow for a more complex model capable of returning such information to be directly displayed by the front-end.

    (c) Due to our solution's reliance on overhead cameras and clear aerial visibility, the user interface will not function with enclosed parking garages, or parking areas with multiple levels. Spaces are displayed to the user on a two-dimensional plane, so there is no way to switch between different levels for spaces that are stacked on top of one another. Additional resources could be allocated to a custom-built view to support such situations.

    (d) With the current solution, our navigation system does not offer appropriate accessibility features such as voice commands/prompts and minimal number of taps

to perform an action. Currently, once selections are made, minimal further interaction is needed, but additional resources could be allocated to user interface improvements to make hands-free operation as easy as possible.

2. Give a brief overview of other design solutions you considered. What are the benefits and tradeoffs of those other designs compared with the chosen design? From all the potential options, why did you select documented design? (LO_Explores)

   (a) The use of GPS, rather than overhead cameras, was considered to map parking lot layouts and mark occupied spaces. By noting at what specific coordinates app users tended to park their cars, the idea was to crowd-source the layout of a given parking lot over time. Spaces would be marked as occupied if an app user's vehicle was parked at the given coordinates. This would eliminate the need for cameras, as well as manually creating a layout.

   There are two main issues. First, the application would need to see broad adoption for it to be useful. Many users would be needed for an entire parking lot to be mapped out in an acceptable time frame, and then spaces would only be marked full if occupied by an app user. Second, mobile GPS varies widely in its accuracy, and is often not precise enough to determine exactly which parking spot has been occupied.

   (b) We considered the possibility of installing hardware on each parking space responsible for polling the state of its space and reporting back to a central system. Using specifically designed hardware would eliminate the issues stemming from variations in camera quality and position, or GPS accuracy.

   The issue with this idea is the difficulty of scaling up over large or multiple parking lots. Enough hardware would be needed for every parking space, and the hardware would need to be installed manually for every space.

   We selected the documented design as it needs very little from the end user in comparison to other candidates. It relies solely on separate camera hardware, and not GPS or specialized hardware. It is intended to work just as well regardless of the number of people using the application, and without the need for extensive hardware installation.