

Radford University Parking App



User's Manual Document

*Angel Diaz-Soriano, Kevin McGruder, and Drew Richards
Group 3*

12/6/2022

USER'S MANUAL

TABLE OF CONTENTS

	<u>Page #</u>
1.0 GENERAL INFORMATION	3
1.1SYSTEM OVERVIEW.....	3
1.2AUTHORIZED USE PERMISSION	3
1.3POINTS OF CONTACT	3
2.0 SYSTEM SUMMARY.....	4
2.1SYSTEM CONFIGURATION	4
2.2USER ACCESS LEVELS	5
2.3KNOWN BUGS/SYSTEM FLAWS.....	6
3.0 QUICK START GUIDE	7
3.1LOGGING ON (GAINING ACCESS TO THE SYSTEM)	7
3.2SYSTEM MENU	7
3.2.1Who Menu	8
3.2.2 Pass Menu.....	9
3.2.3 Map Menu.....	9
3.2.4 Lot Menu.....	10
3.2.5 Spot Menu	10
3.4EXITING THE SYSTEM	11
3.5SPECIAL INSTRUCTIONS FOR ERROR CORRECTION.....	12
3.6CAVEATS AND EXCEPTIONS.....	12
4.0 FUTURE ENHANCEMENTS	13
4.1FUTURE FUNCTIONALITY.....	13
4.2SIMILAR SYSTEMS – ADDITIONAL FUNCTIONALITY	13
4.3MAINTENANCE CAPABILITIES.....	13

1.0 GENERAL INFORMATION

1.1 System Overview

The Radford University Parking App is a web based mobile / desktop application that displays information regarding university owned parking lots. The user can select their parking permissions and the map will display the lots they have permission to park in. The map also displays each lot in a heatmap fashion. This allows the user to find where an available parking spot may be located faster. Our application builds its predictions based on user input. Once the user selects a lot, they will then have the option to select their specific spot and select that they parked there. This functionality allows us to compare lots and parking spots to decide where the user is likely to find available parking spots. The user can access this application by visiting our website through university Wi-Fi or while logged onto Cisco AnyConnect VPN with their Radford credentials. Although the user must select their parking pass before seeing the map. The user can still select that they are parking at any spot on-campus.

For deploying this system, a folder within a webserver (in our case Apache) containing all necessary files will be needed. The system configuration is a Linux server distribution running a LAMP technology stack. For this instance, this application is running on Radford University's PHP server. Where various APIs are being used, namely LeafletJS, to complete the requirements of the application.

1.2 Authorized Use Permission

This project is open-sourced, where a lot of the information about our parking app comes from publicly known information. If a person were to obtain the source code, they would be able to retain a copy to do what they wish. However, this does not include the private information stored on our database through those who use our system in the implementation we have used. If all else, we will operate under the MIT License. Which states:

“Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

1.3 Points of Contact

Type of contact: Email.

Developers of system (Includes all POCs):

Drew Richards – darichards@radford.edu

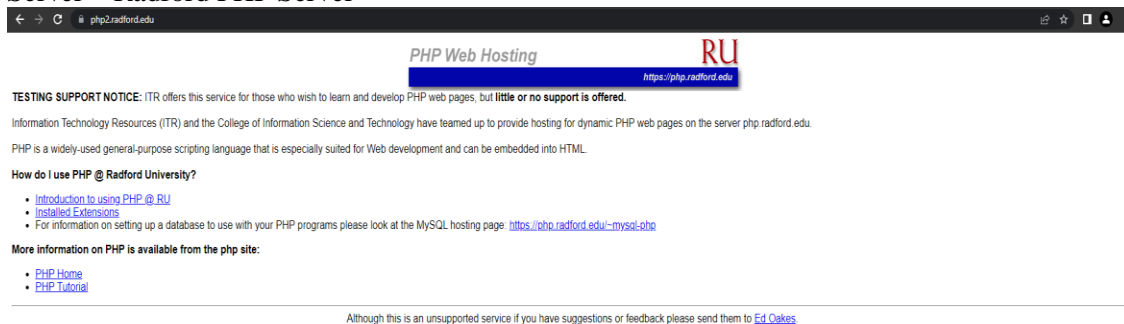
Angel Diaz-Soriano – adiazsoriano@radford.edu

Kevin McGruder – lmcgruder@radford.edu

2.0 SYSTEM SUMMARY

2.1 System Configuration

Server – Radford PHP Server



Database – Radford Database Accessed through MySQL Workbench

Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

[Browse Documentation >](#)

[Read the Blog >](#)

[Discuss on the Forums >](#)

MySQL Connections

[Filter connections](#)



Technology Stack – WAMP / LAMP



APIs – LeafletJS, fonts.googleAPI



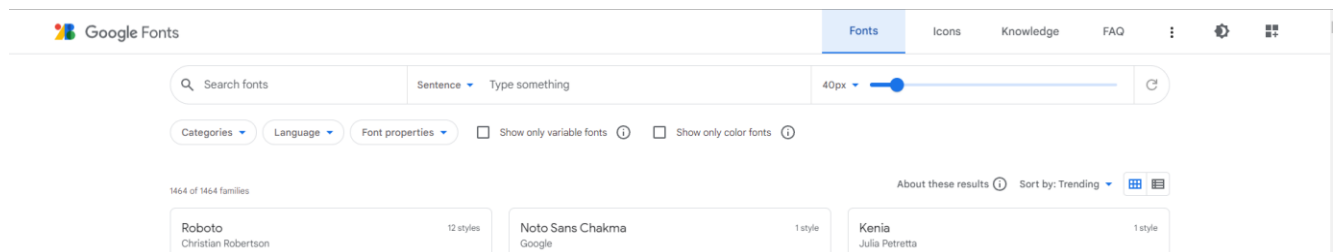
an open-source JavaScript library
for mobile-friendly interactive maps

[Overview](#) [Tutorials](#) [Docs](#) [Download](#) [Plugins](#) [Blog](#)

Sep 21, 2022 — [Leaflet 1.9](#) has been released!

Leaflet is the leading open-source JavaScript library for mobile-friendly interactive maps. Weighing just about [42 KB](#) of JS, it has all the mapping [features](#) most developers ever need.

Leaflet is designed with *simplicity*, *performance* and *usability* in mind. It works efficiently across all major desktop and mobile platforms, can be extended with lots of [plugins](#), has a beautiful, easy to use and [well-documented API](#) and a simple, readable [source code](#) that is a joy to [contribute](#) to.



Web Languages: HTML, JavaScript, & CSS

What's the Difference?



HTML

Hypertext Markup Language

Create the structure

- Controls the layout of the content
- Provides structure for the web page design
- The fundamental building block of any web page



CSS

Cascading Style Sheet

Stylize the website

- Applies style to the web page elements
- Targets various screen sizes to make web pages responsive
- Primarily handles the "look and feel" of a web page



Javascript

Increase interactivity

- Adds interactivity to a web page
- Handles complex functions and features
- Programmatic code which enhances functionality

2.2 User Access Levels

The system only can be edited and maintained by the admin through logging into the specific PHP server the site is hosted on. The administrator can view all files and change all files located on the server. The administrator also can log on to the database and see raw data, clean data, alter tables, and delete tables.

The client can access the site through the link. Once on the site they can either log in or continue as a guest. After using the site once, the following time the client uses the site, it will remember their cookie

and immediately load the map with their initial parking pass preference. Other than typing in a log in, and clicking on the buttons, lot, and parking spot options, the user will not have any more permissions or options. This is strictly a user preference, removing a lot of permission that the user has.

2.3 Known Bugs/System Flaws

- The user can lose their login by clearing cookies, thereby removing their ability to log in again.
 - This is because there is no explicit system to validate a user, and thus there is no foreseeable way to validate that the proper user is attempting to log in.
 - A workaround would be to simply be careful when deleting cookies from websites. Assuming the user understands this concept of cookies and sessions.
- The user can go back to previous pages after clicking on buttons.
 - This means that a user could accidentally create a record on the database and reattempt to do so by changing older preferences. However, this will result in a constant redirection to the map page.
 - The user shall not go back to older pages as a workaround, as there is no need to do so.
- The map legends can get in the way of a parking spot within the lots themselves.
 - This increases the difficulty for a user to park in lots of the upper left corner of the mini-lot map.
 - A workaround would simply be to click on the legend (it allows clicks through itself) or to zoom into the lots more to select a spot.
- The entire parking application does not restrict users to where they can park in the application.
 - This essentially means that a user can decide to park anywhere, which may not be representative of the pass they have.
 - The user can simply park where they are designated, although this is through discretion of the user.
- The user can enter any email (alphabetic & numeric) so long as it ends with @radford.edu
 - There is no explicit access to the userbase of emails, and thus there is no form of verification because there is no way to tell whether an email is valid or not.
 - The user could enter a proper email, or simply use the “Enter as guest” button to get that sorted.
- Some lots and their spots may be inaccurate and not representative of what they may contain.
 - There was no simple way to collect information on parking lots. Because satellite images were not up to date. Or numbers officially given by the parking services not being able to keep up with the changing parking lot structure of Radford University.
 - The user can attempt to use the app in a location where they park, omitting the other extra lots that may be inaccessible.
- LeafletJS may run into a couple of errors during operation.
 - There is a chance that LeafletJS can crash as the user uses the application. Causing the inability to use the map or pan over to other locations.
 - The user can refresh the page as a work around to this (this crash behavior is outside our control).

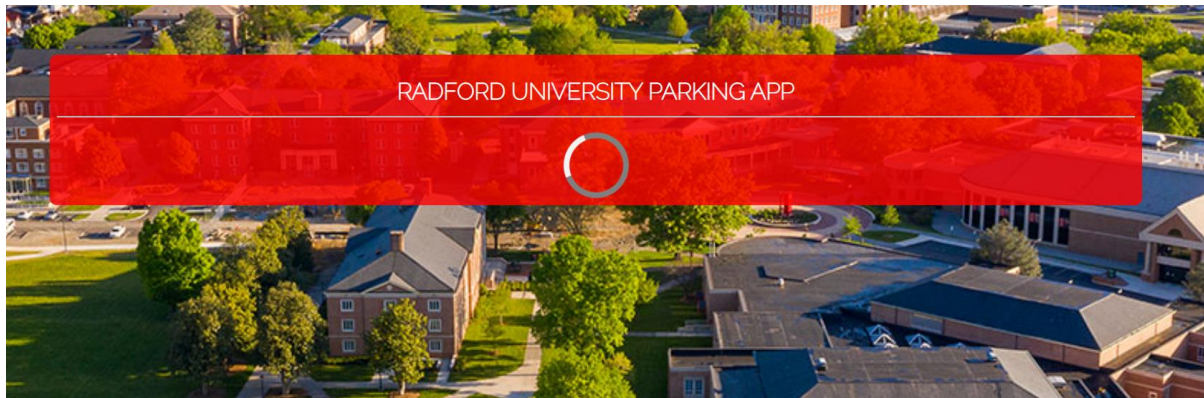
3.0 QUICK START GUIDE

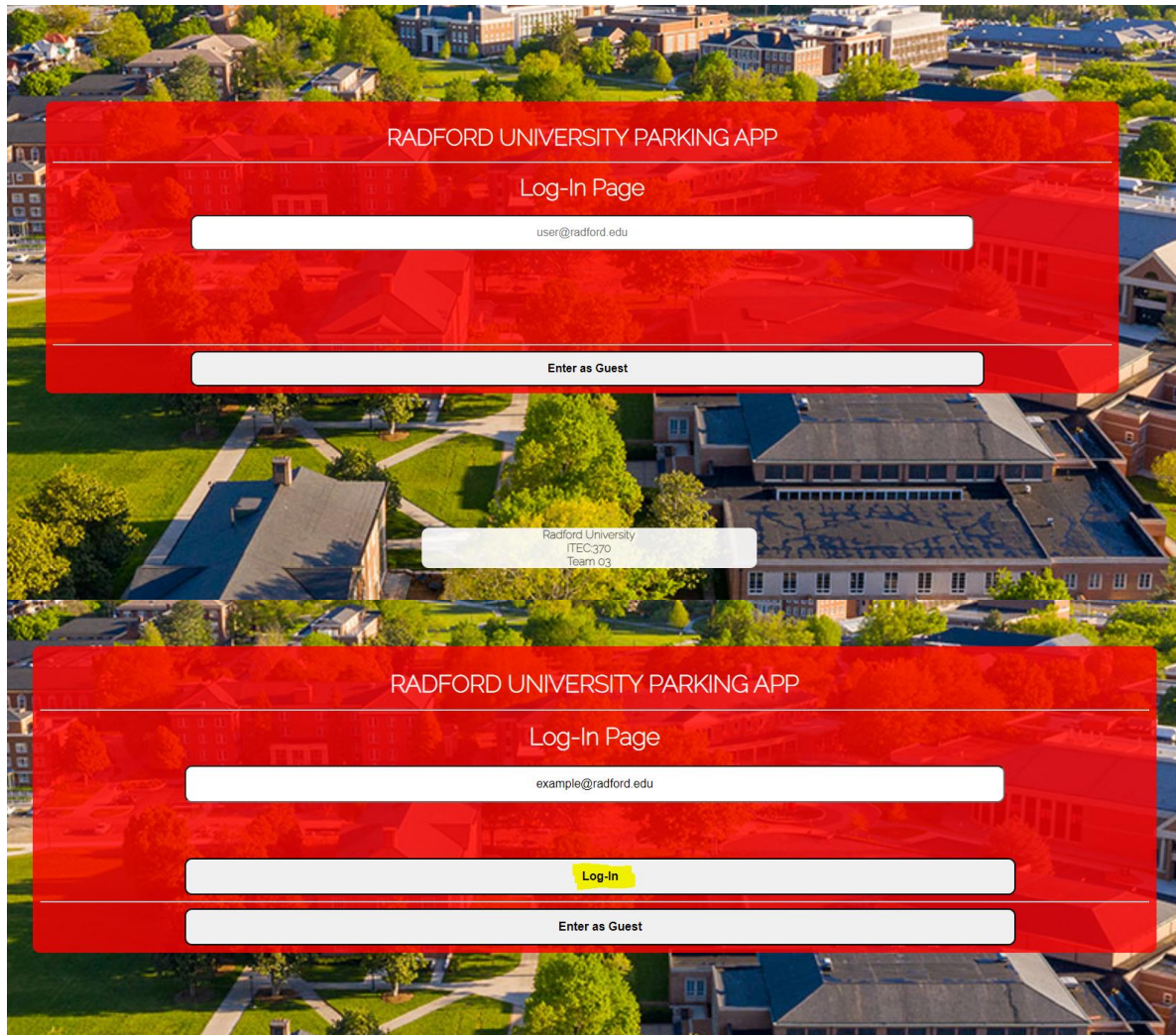
3.1 Logging On (Gaining Access to the System)

The client must be on Radford University Wi-Fi and visit our website. Once on the site page, the client will select their parking pass and view the interactive map. Once on the map, the client will select a lot they want to see the likelihood of having an open spot. The client can then select a spot and select to park there if it's open, or not select to park there.

3.2 System Menu

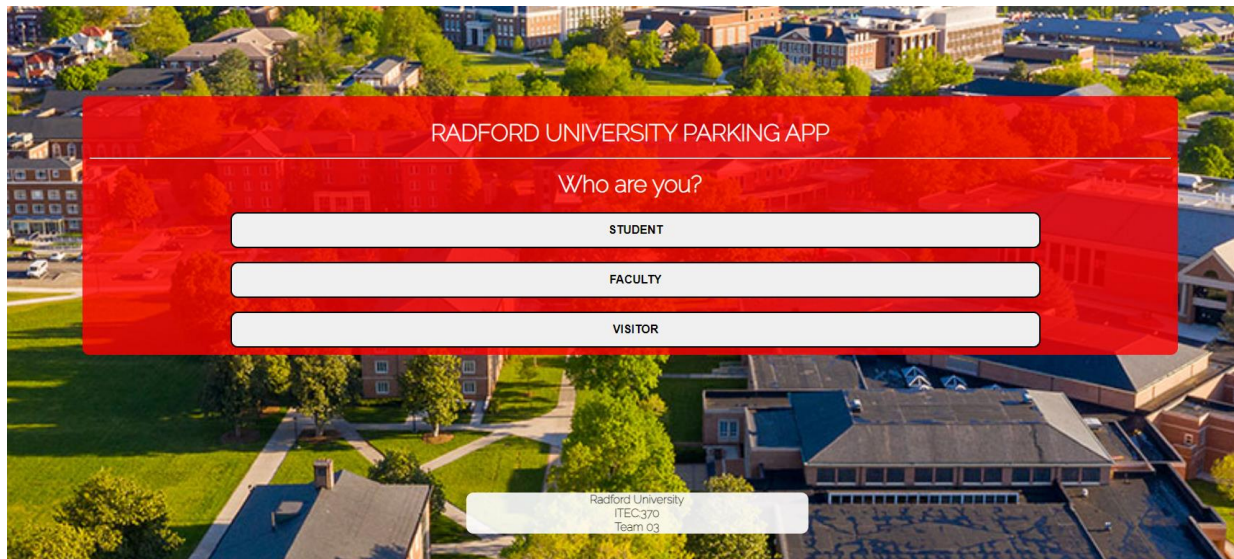
The user will land on the splash screen that will check their browser for a login cookie. If the user has a login cookie, then the site will load the map with their preferences. If no login cookie is detected, then the user will either login or enter as guest. Then they will select the identity they have at the school, such as a commuting student, or a faculty member and then which pass they have. Once these two preferences are selected the map will load showing the lots that they have access to park in. The map will also demonstrate each lot with the colors green to red, representing a heat map. Red meaning most used, and green meaning least used.





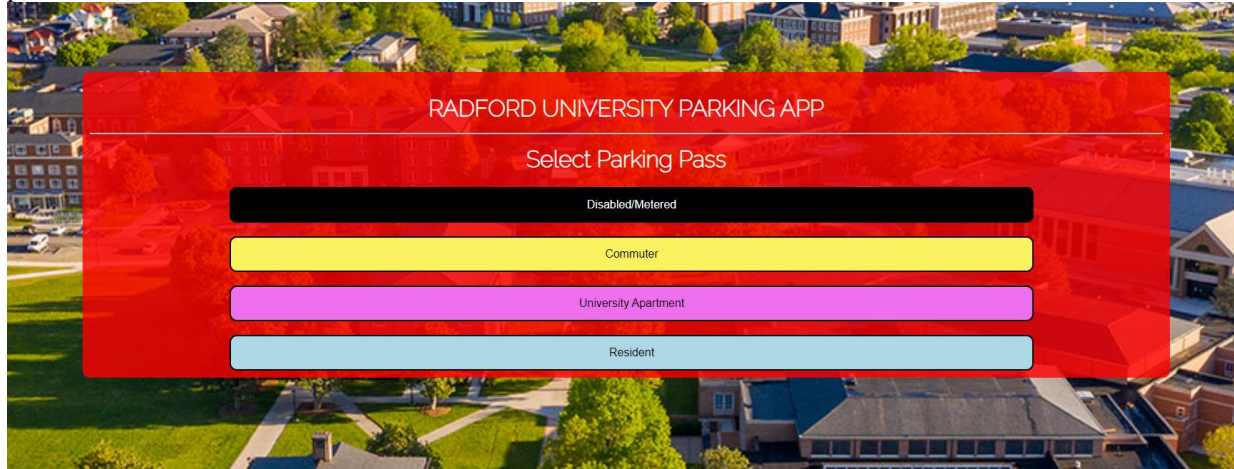
3.2.1 Who Menu

The Who Menu page provides the user with a couple of options. Which are “Student,” “Faculty,” and “Visitor.” The purpose of this page is to give the system information on whether it needs to send the user to the next page or straight to the map. For instance, selecting the “student” button will send the user to the pass menu page. Selecting the “faculty” button will send the user to the map right away. Selecting the “visitor” button will send the user to the map right away. This helps us streamline the process of giving the user their parking pass preference. The average response time would be a couple of seconds as the user reads and selects their identity.



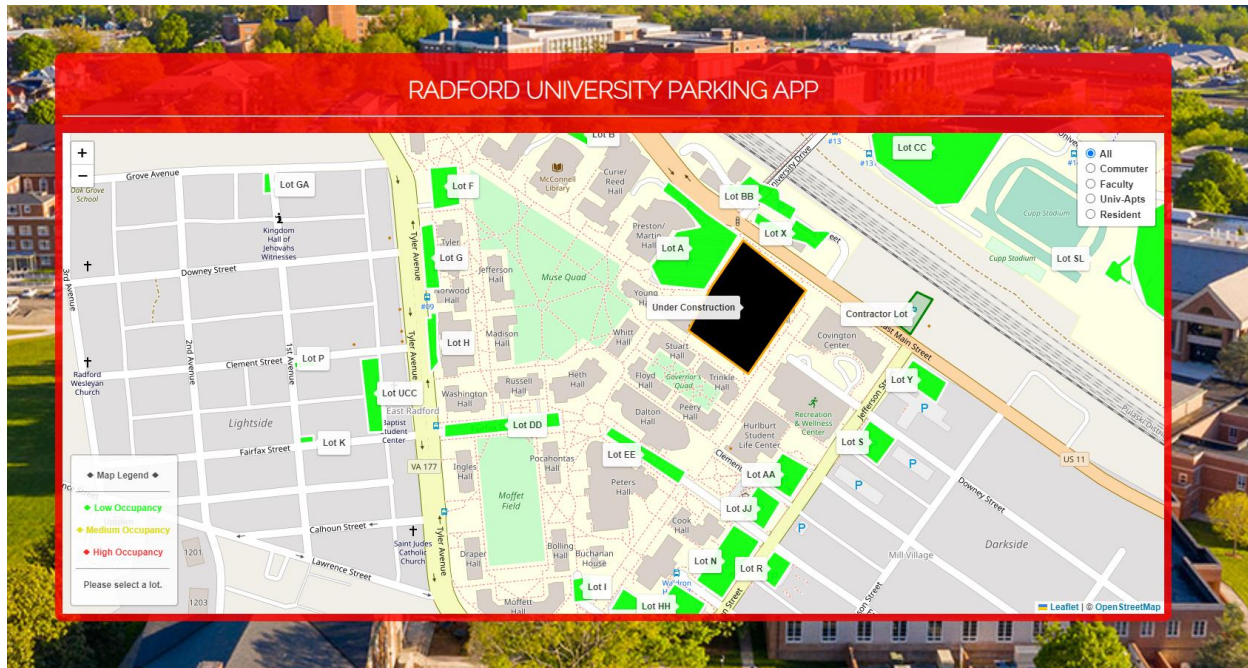
3.2.2 Pass Menu

The Pass Menu page provides the user with a couple of options. Which are “Disabled/Metered,” “Commuter,” “University Apartment,” and “Resident.” This page is only available to those who have selected “student” from the previous page (Who Menu). These further filters the input given by the user, so that the system can boil down what pass they have before sending the user to the map. This page would take a user a couple of seconds to a minute or so depending on what the user remembers about their parking pass.



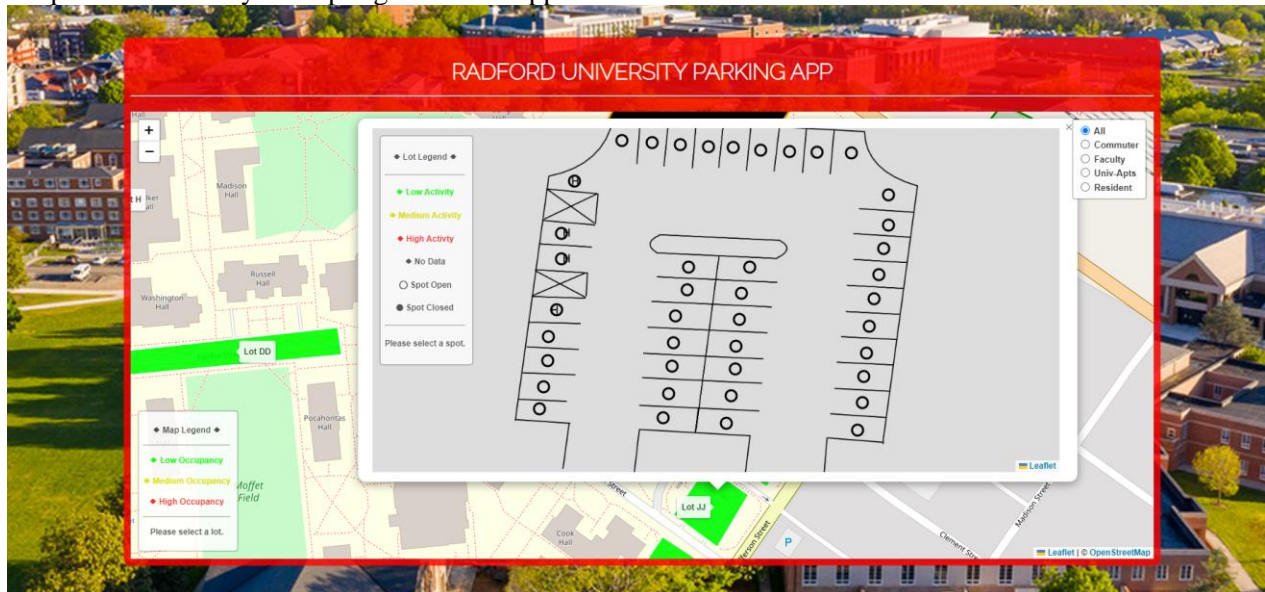
3.2.3 Map Menu

The Map Menu page provides the user with a graphical user interface of an interactive map. This is the heart of the application as it provides the user with a parking service. Based on the information that has been provided to the system before, it will determine what defaults to give the user whenever they choose to come back to the application. Depending on how much the user uses interactive maps, this may take a couple of seconds to potentially a couple of minutes for the user to figure out as they attempt to interact with the system.



3.2.4 Lot Menu

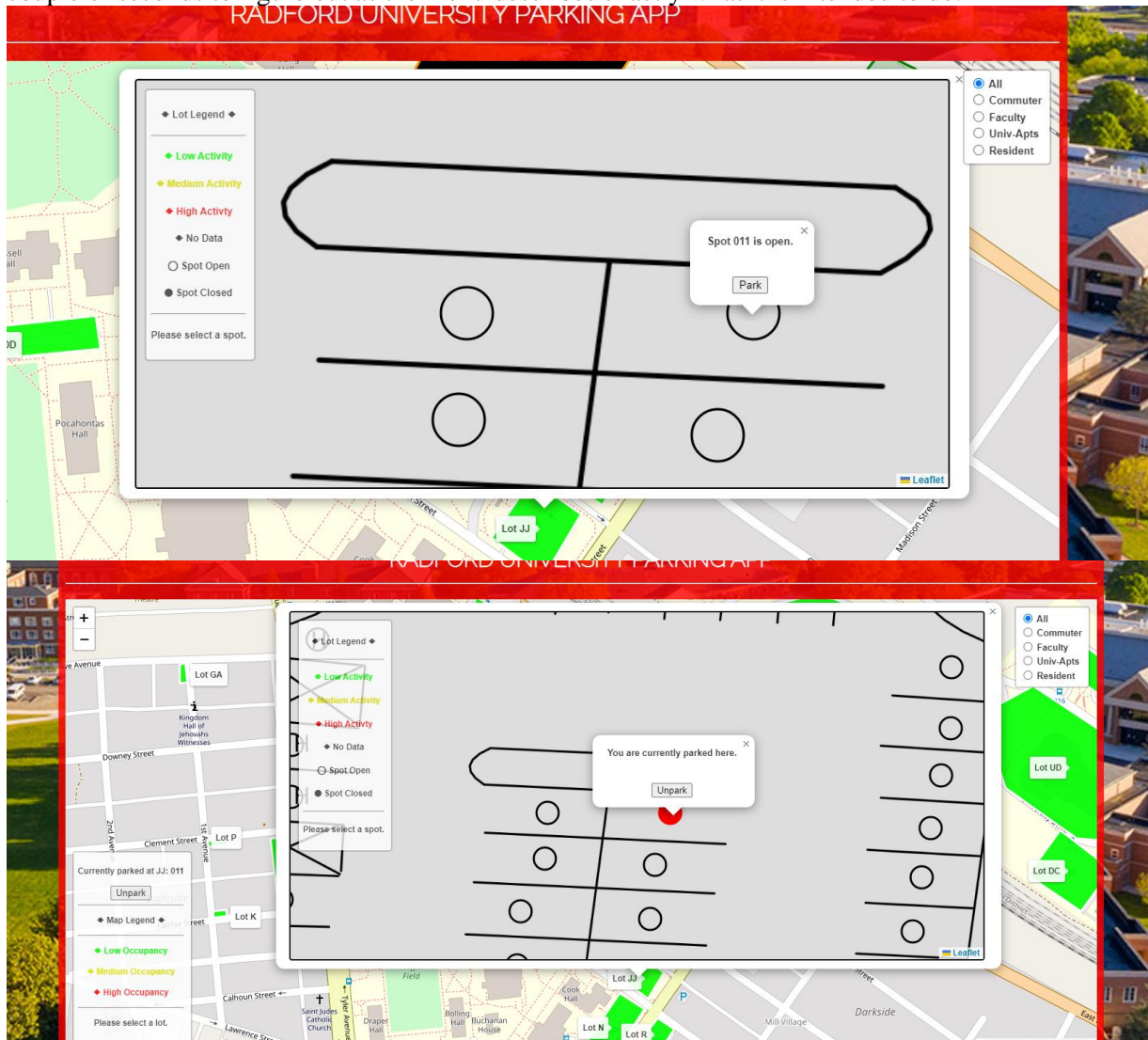
The Lot Menu provides the user with a graphical user interface of an interactive map of a lot. This is a core feature of the parking app. As the user can select a lot (such as this one) and interact with the parking lot to see useful information regarding parking statuses. Given this, the user has a couple of options, they can select any of the number of lots and choose to park there. Getting to this point would require the user to select a lot before being greeted by this page. Then to try and understand how to pan around a small interactive map of a lot. This could take anywhere from a couple of seconds to a minute to understand if the person is actively attempting to use the application.



3.2.5 Spot Menu

The Spot Menu provides the user with two choices for what to do next. This menu is not very extensive by any means; however, it plays an important role in our application. The user is given two

options depending on what they have done already. If the user is currently not parked anywhere, then the system will prompt them to park if they decide to click on a lot. However, if the user is parked somewhere, then they will no longer have access to park anywhere else until they unpark. The option to unpark will be given at the spot they've parked and the overall map legend (for convenience). This will take anyone a couple of seconds to figure out as the menu describes exactly what it is intended to do.



3.4 Exiting the System

To exit the system the user can simply close their phone or exit out of the site by closing it. No matter how they exit the site it will retain the user input.

3.5 Special Instructions for Error Correction

Because this is more of a web-based application. It is very difficult to have error conditions as the user is extremely limited to a handful of buttons to press. There will be moments where the application becomes unresponsive. It is observed that it would be either connection or one of the used APIs that would cause the system to not work as well. All the users would need to do if something like this happened is refresh the page.

3.6 Caveats and Exceptions

To unpark the user must manually select that spot to unpark or click unpark in the map legend. If the user is currently parked somewhere, it will be shown in the map legend, and they will be given the option to unpark. The user will also not be allowed to park anywhere until they unpark from their previous spot first. The parking system is generally something that needs to be used in order for this app to have utility for everyone else.

4.0 FUTURE ENHANCEMENTS

4.1 Future Functionality

Adding new lots and the new hotel the university is building. The university pdf we used stated all the spots, handicap spots, and metered spots we believe is slightly wrong. This would have to be revised by checking with the parking services about lot changes or just going to each lot and comparing the data.

Adding a more refined login system would provide greater sustainability to this system than anything else thus far, as it would simply be something that would enable people to keep more of a personalized experience, as well as providing the person with continual access to the account without the fear of losing information.

Including the use of GPS to ensure accurate parking by the user is a feature that has been suggested, where the location of the person could either confirm a park or enhance the viability of detecting and validating a park.

Having information on each parking lot, and enforcing the information given for each parking lot would be something that could help the integrity of the rules in place for each lot. For example, some lots may allow people to park there after a certain hour. The current system does not account for this.

4.2 Similar Systems – Additional Functionality

- Google Maps
 - The system features that would come from Google Maps would include the live information feed that will be useful for those wishing to know about traffic. This would be an additional system that would be helpful to those using it.
- SpotHero
 - The system features from this application rely more on booking a spot ahead of time. As well as the monetary side of applications like these ones. Where the application could utilize some “spot check out” system to reserve spots for people, thus giving the person who paid a priority over others.
- PassioGo
 - This system would provide more of a public transport information system, which would also be considered useful to those using our application. Where finding things like the nearest bus at a lot, for instance, would provide an incredible amount of assistance to the user using the application.

4.3 Maintenance Capabilities

Maintenance would consist of cleaning the database likely each semester. This would ensure that it did not retain unneeded or unused data. The only two tables that would need cleaning are parkHistory and Users. Every other table contains information pertaining to lots and spots that do not change. ParkHistory and users table should be cleared every semester and then in the spots table every spot should be set to not parked in.