2020

User Documentation

CWU PARKING APPLICATION

KEVIN BERTELSEN – CAMILO JACOMET – COREY JOHNSON – HAILEY LAWTON
PAUL MCCAFFERTY – DRAKE WALD

HAILEY & THE BOYS | 400 E University Way, Ellensburg, WA

Table of Contents

Introduction

Parking lot conditions at Central Washington University (CWU) are currently one of the biggest complaints that the student body has about the campus. Students are spending a great deal of money, and they still aren't able to find parking in time to make it to class.

In order to solve this problem, we have developed the CWU Parking Application, which may be used as a proof of concept for the creation of an application which is able to track the number of parking spots available in each parking lot on campus in real-time. In the current version of the application, the backend of the system will push random number values to the database in order to simulate parked cars. This document may be referred back to in the case than an individual is attempting to install or use the CWU Parking application, as the application is currently unavailable in the App Store or Google Play Store.

Installation Instructions

The following are installation instructions which a user may implement in order to use the CWU Parking application from an Android or iOS device. The information included describes installation steps for the most recent version of the application, which is currently makes use of a Python script in order to generate random integer values and push these values to an online Firebase database for querying.

The following installation instructions are tailored to the system which the application was developed on. For the front end, including the installation of React Native, this means that the instructions are tailored to a Windows 10 Home operating system, as this is what the application has been developed on. In the case of the Python installation, instructions are tailored to a Linux Ubuntu 18.04 operating system.

React Native

React Native is the development platform which was selected for this application. Currently, since the application is not available in the App Store or Google Play Store, the installation of React Native is the only way to successfully run the application. React Native is available on Linux, Windows, and macOS operating systems.

In order to install React Native, one must first install Node.js. While the most recent version of Node.js was recommended to the development team, it has been discovered that this version causes some technical issues when attempting to build the code behind the user interface. In order to remedy this issue, the development team recommends an installation of the Node.js version 10.16.3, which has not been shown to have these problems. This version of Node.js is still available on online platforms for all operating systems.

Once Node has been installed, users must install the Expo CLI command line utility. The command line utility will give the user access to the ability to create new React Native projects as well as allowing the user to start a development server. This development server will allow the user to view the changes that they are making to their project in real-time, as they are saving their JavaScript files. To install the Expo CLI command line utility, users should run the command **npm install -g expo-cli** within their command prompt terminal.

Upon installation of node, users are nearly to the point where they will be able to run their project. In order to do so, the user should download the project .zip folder and extract the files within, saving them to the location of the user's choosing. If downloaded directly from GitHub, the folder should initially be named CWU-Parking-master. Once the files have been saved, the user should use the **cd** command in order to navigate to within the CWU-Parking-master folder. From within the project folder, the user should run the command **npm start** which will start the development server and allow the user access to the project, via the QR Code which is displayed within the window.

Troubleshooting

A common issue encountered by the development team occurs at times when running the command **npm start**. This issue may be remedied by first running the command **npm install** within the project folder, and repeating the **npm start** command.

Python

In order to configure the server to run the Python-based backend, one must install both Python 2 and Python 3. Python 2 is used for the OpenCV library used to operate the USB webcam through the server, and Python 3 is used to post the data to the Firebase from the server. In order to properly install Python 2, run the following commands:

- **sudo apt-get update**
- **sudo apt-get install build-essential checkinstall**
- **sudo apt-get install libreadline-gplv2-dev libncursesw5-dev libssl-dev libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev**
- **cd /usr/src**
- **sudo wget https://www.python.org/ftp/python/2.7.16/Python-2.7.16.tgz**
- **sudo tar xzf Python-2.7.16.tgz**
- **cd Python-2.7.16**
- **sudo ./configure --enable-optimizations**
- **sudo make altinstall**

In order to ensure that you have successfully installed Python 2, run the command **python2.7 -V**.

Then, install Python 3:

- **sudo apt-get update**
- **sudo apt-get install python3.6**

There is a script present inside of the PythonScripts github folder called auto_install.sh that can be made into an executable file using "chmod +x auto_install.sh" that can be used to automatically install all Python-related dependencies onto the Ubuntu server, but the above commands can be entered manually if the script fails to install some of the modules correctly.

Next up, you want to create a cron job for these scripts to run at a set interval of your choosing. In order to create a cron job:

- Open up a terminal in the Ubuntu server
- Enter the command **crontab -e** to open the crontab file editor
- Paste the following command after the printed instructions:

***/30 * * * * cd /home/taxi/ParkingApp; python db.py >> logs/log.txt; echo 'Success!' >> logs/log.txt**

- Save the file
- Create the logs/ folder inside of the PythonScripts folder, and create a logs.txt file inside of that logs/ folder.

This job will run at the 30th minute of every hour, but the timeframe can be altered using the https://crontab.guru/ online tool. The logs.txt file will assist in debugging the cron job, and will ensure that the information is being correctly inserted into the database.

Expo Client

The Expo Client mobile application can be downloaded from the App Store and the Google Play Store, making it available for both Android and iOS operating systems. This has been useful throughout the testing process, as the development team has been able to concurrently check the results of their design changes on both platforms. Once the application has been installed, there are differences regarding how the application is able to be run on Android and iOS operating systems. However, in the case of both operating systems, the user should ensure that their mobile device and the device which is running React Native are connected to the same wireless network. This may be accomplished using wifi, or by using a personal hotspot on the mobile device.

Android

In the case of Android, once the Expo Client application has been installed, accessing the application is a fairly simple process. The user should begin by ensuring that the 'Connection' which is set on the left side of the screen is set to the mode 'Tunnel'.

Once this is done, the user should select the option to scan a QR code. The QR code will be located within the development server opened on the computer console after the user has run the **npm start** command. Once the user scans the QR code, the JavaScript bundle will download, and once the download has completed the application should automatically open to the main screen.

iOS

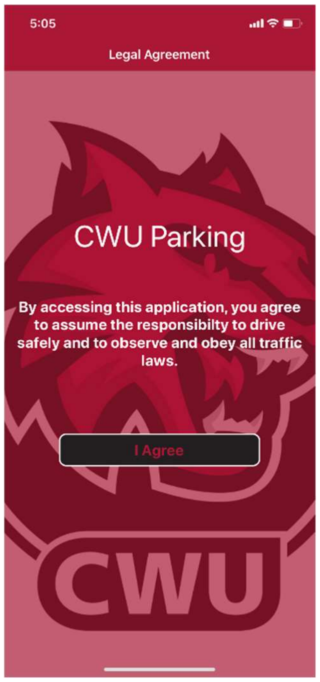
In the case of iOS, once the Expo Client application has been installed, accessing the application is slightly more complicated. The user should begin by creating an Expo Client account, an option which is provided within the application. Once the account has been created, the user should return to the command prompt from which they have run the **npm start** command.

By pressing the **?** key on the user's device, they will be displayed a list of available commands while they are working within the development server. In order to access the new Expo Client account which they have created, the user should use the **s** command. They will then be prompted to enter the email address and password that they used to create their Expo Client account. Once they have successfully signed in, the user should return to the development server window.

Rather than scanning the QR code, the user should select the option to 'Send link with email…', an option in the sidebars listed to the left. When this option is selected, the user should enter a convenient email address where they can receive a link to the application. Once the link appears in their email, the user is able to click the link. The JavaScript bundle will begin downloading, and once this download has completed, the application should automatically open to the main screen.

Firebase
No installation should be necessary in the case of the Firebase database which has been implemented for the application. This is due to the fact that the database is cloud hosted, and therefore accessible from the code which has been implemented for the system. The version of Firebase which is currently being used is the latest available, as it is being regularly updated by the Google provider.

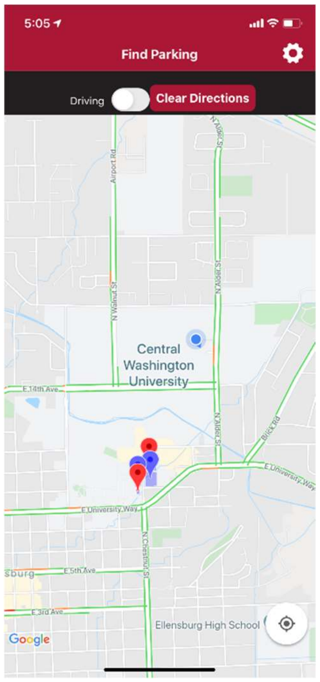
Use Instructions
Landing Screen
The landing screen of the application details a legal agreement between the user of the application and the development team. This has been included in order to remind the user to obey all the rules of the road while the application is in use. The main concern associated with this is the use of the application while the user is driving, as this is considered to be distracted driving.



By accessing this application, you agree to assume the responsibilty to drive safely and to observe and obey all traffic laws.

Functionality
In order for the user to continue from this screen, they should select the ‘I Agree’ button, which will redirect them to the accompanying Maps screen display so that they may begin the process of selecting the parking lot they wish to be directed to.

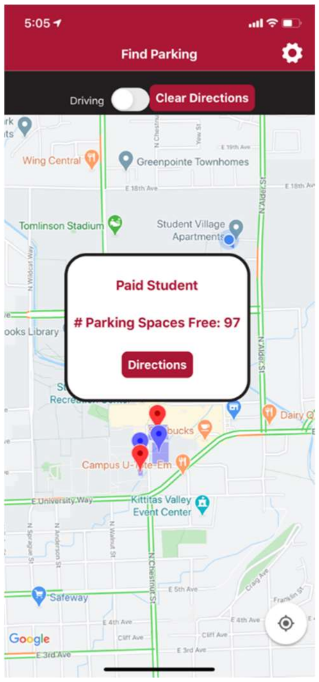
Maps Display
The maps display of the application automatically displays the generated polygons on the CWU campus, represented by colored polygons which appear on the map itself. Accompanied by these polygons are location pins, which are colored according to the type and location of parking lot which they are associated with.



Functionality

Pin Selection

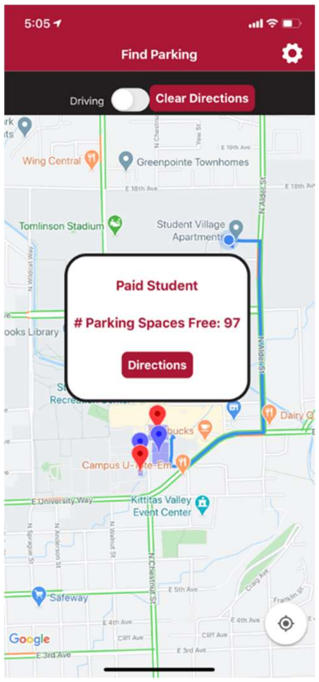
A pin may be selected by the user by tapping on the screen where the pin appears. Once this has touch has occurred, an automatic popup will display, providing the user with information regarding the type of parking lot they have selected and the current number of available parking spots. This value will change automatically as more or less parking spots become available. The user will also be displayed a button, 'Directions', for selection which will be discussed further.



Should the user wish to access information regarding another parking lot, they are able to do so by selecting any of the other location pins. They may also clear the information from the screen by selecting any other portion of the map screen.

Directions

Once the user has selected a parking lot that they would like to be directed to, they should select the 'Directions' button within the popup window. They may again do this by tapping the screen where the button appears. Once the button has been selected, a blue line should appear between the selected location pin and the location that the user is currently at. This line may be cleared from the screen when the user selects the 'Clear Directions' button, which appears near the top of the screen.



Mode of Transportation

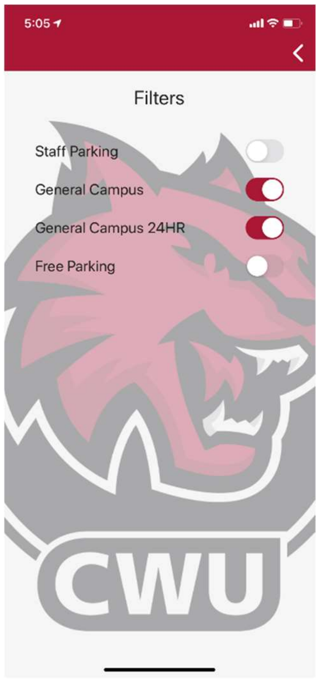
At the top of the screen, the user is provided a switch which is used to determine whether the user will be used to determine the mode of transportation being used in order to get to the selected parking lot. The two options currently include 'Driving' and 'Walking'. This will determine whether the user is directed to the parking lot of choice via sidewalks or main streets where cars are able to easily travel.

Page Navigation

Also at the top of the screen, the user is provided with a settings icon for selection. Once this icon is selected, the user will be automatically redirected to the accompanying Settings screen display.

Settings Options

On the settings screen, the user is displayed various filtering options for the type of parking lot which they would like to be guided to. These switches include 'General Campus', '24 Hour General Campus', 'Staff Parking', and 'Free Parking'.


5:05

Filters

Staff Parking
General Campus
General Campus 24HR
Free Parking

Functionality

Switch Activation

The switches listed may be activated by the user’s touch input. Once the user activates the switch, it will automatically toggle to the other mode option. This means that once the page navigation takes place, only those parking lots which are associated with active switches should be displayed on the map as selection options.

Page Navigation

At the top of the screen, the user is provided with a left-facing arrow for selection. The user should select this option by tapping the screen where the arrow appears. Once the arrow has been selected, the user will be navigated back to the Maps screen, where their changes have appeared.

Conclusion

The purpose of this document is for use as a reference by users so that they may make use of the CWU Parking Application is as efficient a manner as possible. If there are any questions or concerns regarding the functionality of the application or any of the download instructions, the development team may be reached through Hailey Lawton at lawtonh@cwu.edu or Kevin Bertelsen at bertelsenk@cwu.edu.