Customer Statement of Requirements

Parking at Sacramento State is a hassle. After arriving at the school, one can expect spending 5-15 minutes locating a vacant spot to park. At peak times, this time can become even greater. And at the beginning of the semester when people are trying to add classes, the parking situation can get so bad that students get so frustrated that they turn around and go home because they can’t find a vacant spot. According to source, students that don’t make it to class on time on average have GPA’s that are # points lower than their punctual counterparts and students that don’t make it to class have GPA’s that are # lower. This is a huge problem and needs to be resolved.

Currently, in an effort to ease the strain on the current parking lot system at Sac State, they have decided to build an entirely new 5 story parking structure that will cost around $25 million to complete. There is also an app on the Sac State App that provides “real time” parking availability that has proven to be completely worthless and does not provide good enough information to be useful. And finally, there is temporary road work sign that is used at the North entrance of the school to display which lots are full and recently they’ve actually removed the sign. Clearly this situation is a mess. The system is disorganized and instead of spending $25 million to build new structures, we believe that more connected and responsive parking lots will be able to help parking enforcement and students come to a more logistic friendly resolution when it comes to parking.

The current system uses manually inputted data entered by parking enforcement to display a percentage of how full each lot is. This leaves a lot of room for human error and laziness as lot info has been shown on the Sac State App to go unedited for hours. What we are proposing is coming up with a real-time system to not just monitor the percentage of how full each lot is, but with a preciseness of exactly which spots are still vacant. This information can then be passed on to parking enforcement and students alike to guide students to vacant spots.

Each parking spot will be equipped with a module that can be used to detect whether or not a vehicle is present at the spot with 99.99% percent accuracy. The module will use a HC-SR04 Ultrasonic sensor and will connect to a microcontroller. The microcontroller will be able to analyze the data coming into the sensor and be able to determine if a car is present or not. The microcontroller will then send information to a remote web server where the data will be stored. This data can then be used by parking enforcement and students/faculty alike to tell where parking is available.

At the entrances of each lot will be a display giving a number of how many spots are available within the lot. This will allow students/faculty to be directed towards lots that still have parking spot vacancies. Another thing the data can tell people driving around in a parking lot is if there is parking down a given aisle in the parking lot. All that is needed is an overhanging LED light bulb or sign that displays a "No parking available" sign to be hung up over the entrance of an aisle. This will reduce traffic in already full parts of the parking lots/structures and better direct students/faculty to vacant parking spots.

The average student/faculty user will be able to use the app on their own to find a spot. The home screen of the app will display a heat map of Sacramento State and all the parking lots/structured on campus. From there a user can do a sort, for example a faculty member can limit their query to faculty only lots. Or if they'd like they can click on a parking lot/structure to get a view in real time of where parking spots are available within that parking lot/structure. Once they have a map of what spots are open, all that's left is to drive to that spot and park there. Once they park in a vacant spot, the sensor gets triggered at that spot and the app updates, making the spot appear as "not vacant" on app.

Another scenario is student/faculty may want to park in closest proximity possible to a desired building. Actual commute time on foot is another major consideration that needs to be addressed when designing a system used to optimize logistics. There might be a sizable amount of parking available at lot 7 but let’s say the user's first class is at 9am at the opposite end of campus in Solano Hall. A better place to park would be Lot 1. There's a 0.5 mile difference and a 8 minute difference in walking time from lot to classroom! What a user would be able to do is pick what classroom they want to be closest to. The app will then return a list of spots it recommends parking at based on availability and odds that a parking spot will still be available by the time the user gets the parking lot/structure.

Sometimes the app itself can be very difficult to navigate through. The user opens the app, finds it confusing, and out of frustration may decide not to use it ever again. That’s why our app will be very user friendly. It will mainly focus on finding a vacant parking spot with a few other options that the user may wish to use, such as the sorting feature mentioned earlier. The user can easily choose whether to toggle these other options on or off. The app will be able to find a parking spot with just a few taps after first being set up. Say the user is in a hurry; the user will be able to open the app and within # taps, the app has already found a vacant parking spot for the user.

The database will have outbound communication with the cell phone app along with a possible desktop application. Using a homegrown proprietary REST API in conjunction with the SQL database, the app and web interface alike will have the capability of data extraction through JSON. An example implementation of the getLot method on the REST API would be to display the lot info through JSON. From there, the app and web interface alike can parse the data and generate kml overlays to be used on the google maps map. Possible additional implementation would be to have the web server generate these kml files but the speed practicality of this method is yet to be determined.

The system as a whole will largely be event based. The database becomes updated once it receives stimulus of a change in state from the sensors via the internet. From there, signs and displays will be updated based on the new overall state of the database. The web interface and mobile phone app will also be updated based on these events but may be updated incrementally to save bandwidth. The mobile app and interface are also event based systems in their own respect. A click to see the detailed state of a lot is an event trigger by the user. A request to receive directions to a certain spot is also an event created by the user.