Park This Car

Requirements and Specifications Document

Group #24

Milestone 2

February 25, 2018

Software Engineering 2XB3

McMaster University

Group Members:

Arif, Hareem - 400023154

Choudhri, Kanak - 400084778

Currie, Ian - 400052672

Malik, Ibrahim - 001043905

Table of Contents

By virtue of submitting this document we electronically sign and date that the work being submitted by all the individuals in the group is their exclusive work as a group and we consent to make available the application developed through [SE]-2XB3 project, the reports, presentations, and assignments (not including my name and student number) for future, teaching purposes.

Stakeholders

Our team as the developers and management

Main relationship

The relationship between the developers and this project is the construction of this product for the usage of the Drivers that will correlate to the benefit provided to the commuting public of Seattle.

Goal

The goal is to provide a useable product that benefits the users with respect to saving time, providing ease of travel and an easier experience with the problem at and with the use of this product.

Drivers

Main relationship

They are the users of this product primarily as the product is constructed to provide them with an easier experience finding a parking spot suitable to their needs and requirements.

Goal

The goal is to achieve the optimal parking spot within a reasonable distance to their destination, that is in accordance with current laws and regulations as well as being safe.

Owners of paid parking lots and garages

Main relationship

They provide the physical aspect of interest, the parking spots available to the public.

Goal

The goal is to have their parking lots being easier to find so that they are chosen above other options, especially if they are paid as that will provide them with revenue. This product will help with the ability to advertise in a way without the extra cost of physical signs and such. However there is a conflicting goal which is that this product gives the ability to see parking lots and garages with cheaper rates or potentially free ones at certain times and days, with options like that being presented with contrast, the paid parking lots and garages may fear that they might experience a loss in revenue.

Seattle Parking Authority

Main relationship

The Seattle Parking Authority is the one responsible for regulating the proper use of parking lots and garages in abidance with the laws. They foresee any tickets and apprehending any violators.

Goal

The goal is to help drivers make smarter, informed decisions with regards to parking as it will avoid penalties and mistakes. A conflict of interest is present with this goal as a lack of tickets being given out will result in a lack of revenue for the city.

Common commuting public of Seattle

Main relationship

They are members of the traffic that is directly affected by drivers that are unable to find parking, which may lead to lags in traffic and potentially disruptive parking on streets that also slows down traffic which slows down the commute.

Goal

The goal is to have drivers spend less time searching or awkwardly making mistakes within busy streets. This is crucial for the city of Seattle whose downtown suffers heavy transit times during rush hours that last from 6:00-10:00am and 4:00-7:00pm in the evening. It can help improve the flow of traffic as the drivers will make better decisions and reduce negative interference within the flowing traffic to a certain degree.

Professor and TAs facilitating this project

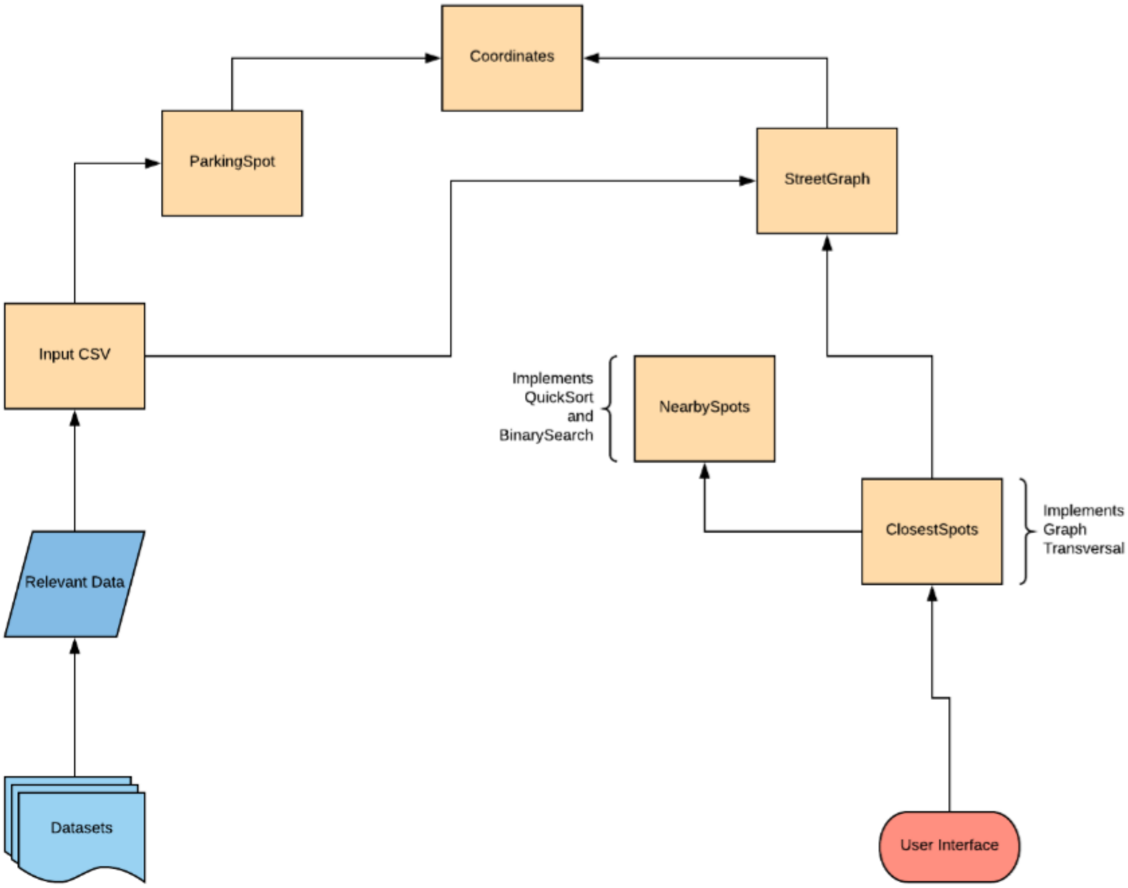
Main relationship

As they are assessing this project the relationship is to see an appropriate application of the concepts learned within the courses Software Engineering 2XB3, Software Engineering 2AA4, and Software Engineering 2C03.

Goal

The goal is to provide the students working on this project with an opportunity to apply the procedure of software development learned in the course by analyzing client requirements and formulating an appropriate set of specific solutions. As this will provide a realistic experience with software development in a real world application within a professional atmosphere.

Functional Requirements



Non – Functional Requirements

Reliability – availability, integrity, safety

This product will provide options to the user that are in accordance with all laws and regulations present for the city of Seattle. This product will not distribute the information taken from the user (locations searches, current location, etc.) to any third party.

Accuracy of results

This product will provide results that are appropriate and applicable for the user and will do so in a timely manner.

Performance

This product will provide the results in a timely manner to make optimal use of the user's time. The product will provide accurate results based on what input it is given by the user.

Operating constraints

The constraints present with operating this product are that the product will only operate on the data present for the city of Seattle, for the time being. It will not operate on searches made for other cities as it does not have the required information on different locations.

Physical constraints

The present physical constraints consist of those presented by the limitations from the datasets available for use that can only account for a select amount of locations within Seattle.

Portability issues

This product is currently being designed to be used on a computer as of yet, with intentions present to make it into a mobile application. The issues present with this intent consist of the different operating systems present on mobile devices and compatibility of the application with the operating system present.

Requirements on the development and maintenance process

Quality control procedures

A J-Unit test suite must be completed upon each module before said module can be considered complete. Testing of a module is only complete if all methods are tested, and each method passes 100% of test cases. Test cases of a method must cover all possible actions which could result from running said method. All test suites must execute within reasonable time (< 1 second) in order for the test to be considered a pass.

- To improve efficiency and prevent overwriting code, any hidden module will assume it will receive input which works as expected, and as such, inputs which would not be received by the module will not be tested. Every assumption made in these hidden modules must be noted explicitly in the testing process however.
- Any module which is available to the external user must handle all possible inputs from the user, including non-sensical inputs. This result can be obtained either through exception handling, or ideally, by preventing the user from inputting non-sensical values at all.

Priorities of the required functions

Basic functionality is valued above completeness of the user interface. Therefore, direct user interface modules are to be considered of lowest priority. Next lowest in priority is dataset input methods, due to them being simple, and easy to implement by other coders, while still being important to the workings of the application. Finally, of highest priority is the searching and graph traversal methods. These functions are difficult and highly specific to this application, therefore they must be completed in order for the the project to be considered complete.

Likely changes to system maintenance procedures

Execution time required for a code to be considered reasonably fast is an arbitrary value. In certain situations it could grow or shrink with greater evaluation. In addition, the order of the test elements is expected to stay largely static, however, it is possible if due to unforeseen circumstances, a certain method, or module is required in the short term, significantly more than those which have agreed upon higher priority, then the order of priority can shift to represent this.

Other requirements

Code should be designed for maintenance with little to no repetition of similar or identical code.