Lab 2 - Product Specification Outline

Ahsif A. Sheikh

CS 411

Professor Thomas J. Kennedy

February 26, 2018

Version 2

Table of Contents

List of Figures

List of Tables

1. Purpose

ParkODU is a web application designed for drivers that need to park at Old Dominion University. The software interface will allow users to view all current parking information related to the school so that users may find parking more efficiently. The ultimate goal of ParkODU is to automate and decrease the amount of time spent manually searching for the nearest parking vacancy. The main solution analyzes the parking availability in real-time and then helps drivers find a vacancy closest to their destination. The solution is optimized using the starting location, permit type, and the destination in order to narrow down the parking recommendations. Hence, allowing the user to save time, resources, effort and find the best parking spot.

The prototype, simulates a garage that demonstrates all of the products capabilities and features, shown in *Table 1*. ParkODU will also gather useful parking space usage statistics that will benefit Old Dominion University Transportation and Parking. It will compile data gathered in real time by various vehicle counting systems installed in the garages, parking lots and make this information open to the users.

The application will be compatible with major web browsers and will also be available for Android and iOS. The main objective of ParkODU is to inform drivers of the vehicle counts so that they can avoid driving to parking facilities that are full and go directly to another facility that has vacancies. The secondary objective of ParkODU is to provide Old Dominion University Transportation and Parking with the data from their parking facilities in order for them to use it for future strategic planning.

1.1 Scope

The ParkODU prototype will have all of the defined features and capabilities of the web app and the native Android/iOS App except support for digital signs. The data that would normally be obtained from vehicle detection systems such as inductive loops, IR sensors, and IP cameras, will be simulated. The ParkODU prototype will compile the simulated data to demonstrate all the features and capabilities of the web app and the native Android/iOS App.

The application prototype will simulate an input and display a real-time vehicle count by floor in every garage. The prototype will provide the detailed floor plan along with navigation to the vacant space. The user will be able to import his/her schedule and the application will generate the nearest parking options. It will also analyze the users previous parking data in for improved recommendations.

The functionality and purpose are vital in showing how this application has improved the users parking experience at ODU. By providing the user with a live count of parking spaces available by garage they are able to quickly and effectively locate parking. Success has been demonstrated because the overall goal of ParkODU is to reduce the amount of time spent manually searching for parking and the user of the application will be more informed allowing this to become possible.

The ParkODU prototype will support nearly all features and capabilities of the real working system. All of the features of ParkODU are designed to provide transparency to ODU parking availability; this will help drivers avoid full parking garages and go directly to parking garages that have available parking spaces. The prototype will demonstrate ParkODU's ability to accurately process the data received from the vehicle detection devices and display the results to end users.

1.2 Definitions, Acronyms, and Abbreviations

Administrator - a special user with access to additional tools for user account and space management

Agile - a methodology that anticipates the need for flexibility and applies a level of pragmatism into the delivery of the finished product

Best Garage - the closest garage to the destination building with the specified minimum number of available spaces

Driver - anyone who drives and parks at ODU

Driver Entry Rate - the number of vehicles entering the garage each minute

Driver Exit Rate - the number of vehicles exiting the garage each minute

Event - an occasion which affects garage and/or space availability

Garage Rate - Driver Entry Rate - Driver Exit Rate (a positive number denotes that the garage is filling up)

Operating Hours - 7:00AM - 10:00PM

Permit - a physical decal that specifies in which spaces the vehicle is allowed to park

Predictions - a guess based on current and historical data about garage space availability

Real-time - current time

Reconfigurable - software-based creation, deletion, or editing of spaces, floors, and garages

Rush Hours - 7:45AM - 9:00AM, 12:00PM - 1:00PM, 3:00PM - 4:30PM

Sensor - any device which indicates to the software whether a space is occupied or not

Signage - signs that indicate the number of available spaces

Statistical Analysis - the ability to use sample data to form predictions

User - an entity using Park ODU

Vehicle Detection Technology - any device which indicates to the software that a vehicle has entered a specified area

1.3 References

Access Automation Car Park Count Systems. (n.d.). Retrieved October 10, 2017, from http://www.access-automation.co.uk/car-park-count-systems. (1)

Agile [Digital image]. (2017, May 8). Retrieved November 29, 2017, from https://www.codingmart.com/uploads/post/image/57e0c0488ca7853c76dd986e/Agile_Development_Process.pngvehicle-coun(F.4.)

Burr, David W. "Is University Parking a Common Grievance?". Parking Today Media. September 2011. http://www.parkingtoday.com/articledetails.php?id=1072. September 2017. (3)

Car counting solutions. (n.d.). Retrieved October 10, 2017, from http://www.puretechsystems.com/solutions-car-counting.html. (4)

"Hazelcast the Leading In-Memory Data Grid" Retrieved January 23rd, 2018 from https://hazelcast.com (6)

How Much Does a Parking Garage Cost? Retrieved November 02, 2017, from http://www.parking.org/2016/01/19/tpp-2013-09-how-much-does-a-structure-cost/. (7)

"IntelliJ IDEA: The Java IDE for Professional Developers by JetBrains." *IntelliJ IDEA*, Jet Brains , Retrieved January 18th, 2018, fromwww.jetbrains.com/idea/. (8)

Operating Budget and Plan. Old Dominion University. Retrieved November 02, 2017, from https://www.odu.edu/content/dam/odu/offices/budget-office/docs/opplan2017.pdf. (10)

ODU Campus Parking Map. Retrieved October 23, 2017, from https://www.odu.edu/content/dam/odu/offices/parking-and-transportation-services/docs/odu-student-parking-map-mm.pdf. (F.1.)

Parking and Traffic Procedures. Old Dominion University. Retrieved November 02, 2017, from https://www.odu.edu/content/dam/odu/offices/parking-and-transportation-services/docs/parking-transportation-rules-and-regulations.pdf. (12)

Providence Place mall enhances parking garage with $20M in improvements (2016, December 15). Retrieved October 30, 2017, from https://pbn.com/providence-place-mall-enhances-parking-garage-adds-more-pay-stations-improves-signage119194/. (F.3.)

Rogers, Emily. Dear Future ODU Students. (2017, August 28). Retrieved November 02, 2017, from https://www.theodysseyonline.com/dear-future-odu-students. (5)
Git, Retrieved October 10, 2017, from git-scm.com/. (21)

Sheikh, Ahsif A. (2018, February). Lab 1 – Gold ParkODU Description

Solutions: vehicle counting. (n.d.). Retrieved October 10, 2017, from http://www.t2systems.com/solutions/vehicle-counting. (14)

"Spring: the source of modern java by Pivotal" Retrieved January 23rd, 2018 from http://spring.io (15)

Team Gold. "ParkODU." December 2017. PowerPoint presentation. (16)

The Problem at Hand - The Expansion of Parking At Old Dominion University. (n.d.). Retrieved November 02, 2017, from https://sites.google.com/a/odu.edu/the-expansion-of-parking-at-old-dominion-university/home/the-problem-at-hand. (17)

University Facts & Figures. Old Dominion University. Retrieved November 02, 2017, from https://www.odu.edu/about/facts-and-figures. Accessed November 1, 2017. (18)

Vehicle Counter. (2016, February 12). Retrieved October 10, 2017, from https://www.kiwisecurity.com/. (19)

Vehicle counting & detection systems. (n.d.). Retrieved October 10, 2017, from https://www.swarco.com/stl/Products-Services/Parking-Solutions/Parking-guidance/Vehicle-counting-detection-systems. (20)

"What Is MongoDB?" Retrieved on January 23rd, 2018 from *MongoDB*, http://www.mongodb.com/what-is-mongodb. (9)

1.4 Overview

This product specification provides the hardware and software configuration, external interfaces, capabilities and features of the ParkODU prototype. The information provided in the remaining sections of this document includes a detailed description of the hardware, software, and external interface architecture of the ParkODU prototype; the key features of the prototype; the parameters that will be used to control, manage, or establish that feature; and the performance characteristics of that feature in terms of outputs, displays, and user interaction.

2 Prototype Architecture Description

The prototype of ParkODU will have all of the defined features and capabilities of the web app and the native Android/iOS App except support for digital signs. The data that would normally be obtained from vehicle detection systems such as inductive loops, IR sensors, and IP cameras, will be simulated. The ParkODU prototype will compile the simulated data to demonstrate all the features and capabilities of the web app and the native Android/iOS App. Below in Table 1, the features of the ParkODU Prototype compared to RWP are shown.

Feature	RWP	Prototype
Real-time vehicle counts on every level of each garage	*	*
Display floor plan to show counts by space on each floor	*	*
Display average vehicle count at each location by time of the day	*	*
Allow users to sort garages by walking travel time	*	*
Allow users to filter garages, floors, and space by their parking permit type and space types	*	*
Allow ODU parking staff to configure parking garages, floors, and spaces.	*	*
Provide directions to each garage from user's current location	*	*
Predict future vehicle counts based on the current and historical traffic pattern	*	*
Upload special event schedules and allow the apps to display notification to end users	*	*
Send data to digital signs at the entrance of every garage	*	

Table 1 - RWP and Prototype Features

2.1 Prototype Functional Description

ParkODU possesses emerging technologies that provide a less time consuming method to identifying parking vacancies through real-time monitoring with counts and displays, navigation, notifications and other management.

2.1.1. Vehicle Counts and Displays

The application will be able to track and monitor garage parking availability in real-time. These counts are organized by garage, floor, and space. ParkODU will display a detailed floor plan of each garage floor and the available places. The average vehicle counts at each location by the time of day will be displayed. We have implemented filters and allowed users to sort from garages, floors, and space by their parking permit. Also added filters that sort garages by walking distance and travel time to another building.

2.1.2. Navigation

The application will be able to utilize Google Maps navigation features. It also predicts future vehicle counts (please see algorithms to expand beyond this point). It will also allow faculty and students to import their schedule to the application for the closest parking space recommendations.

2.1.3. Management

ParkODU allows the addition, editing, or removal of a garage, floor, or space and the removal of user roles. The ParkODU App provides configuration of occupancy signage and sends notifications of the events going on at ODU that may affect parking. ParkODU is a

universal application that will interface with any vehicle detection system (so the user can pick what hardware they are most comfortable managing).

2.1.4. Other

ParkODU will utilize open source software along with mobile support. The goal is to help users become more efficient in their quest for parking. Drivers can avoid full garages and minimize time spent searching for a spot.

2.2 Hardware Interfaces

The hardware for vehicle detection will be simulated by a REST client application. The application will send requests to ParkODU REST endpoint and update the vehicle counts. The application will closely mirror the actual ODU parking traffic during weekday normal hours and also simulate special events on weekends. The ParkODU web app and native Android/iOS App will perform operations on the simulated data to demonstrate the features and capabilities.

2.2.1 Software Interfaces

The software interfaces that will be used in order to accomplish ParkODU are Spring Framework, Gradle, MongoDB, and Git. Spring Framework is an open-source framework for Java-based applications. However, spring framework allows extensions for the developing of web applications along with the Java EE platform (15). This is the Spring Framework we will be using to create ParkODU. This will allow us to develop and android application and also a Web Application. Gradle is a build tool trusted by millions of developers. Gradle is a build automation system that can build programs in Java, C++, and also Python. Gradle can recognize which parts of a build are already up-to-date and offers the flexibility for incremental and multi-scale builds. It will be used to create and generate the builds of ParkODU. MongoDB is a free, open-source

database used for todays applications. MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time (9). Mapping is distributed within the MongoDB framework which allows the opportunity for flexible and horizontal scaling (9). Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency (21). It provides an interface for collaborating with other teammates on a project. It allows a team to make revisions based on the most recent source code uploaded on the hub.

2.2.2 User Interfaces

The user interface will consist of four sections including:

- Total Garage Count
- Filter Selection
- Administration Tools
- Account Management

The Total Garage Count will include real-time floor plans based on occupancy of each spot separated by floor. Each floor will be accompanied by a working graph representing the statistics of availability by day of the parking conditions being updated on an hourly basis.

The Filter Selection section will be able to filter the view by setting the, starting location, permit type, and the destination. The goal of this section is to provide a mechanism to allow the user to find the best parking space while using ParkODU.

The Administrative Tools sections allows for an administrative user to manipulate allocations of parking spaces if conditions change based on parking needs changes. This feature of the UI allows for flexibility using ParkODU to adapt to any parking allocations per implementation. Only administrative users will be able to access this portion of the UI and will

be protected by username and password created at time of installation of ParkODU. All administrative users will have the availability to add users as needed.

The Account Management section allows for an administrative user to add notifications for any future events that affect parking. Such events will display as a notification for user to notify them of future parking conditions and any preparations that need to be taken in accordance with the event. The availability to add, remove, and edit access roles that may need to access the administrative UI can also be configured. The components of the User Diagram can be seen in Figure 1.

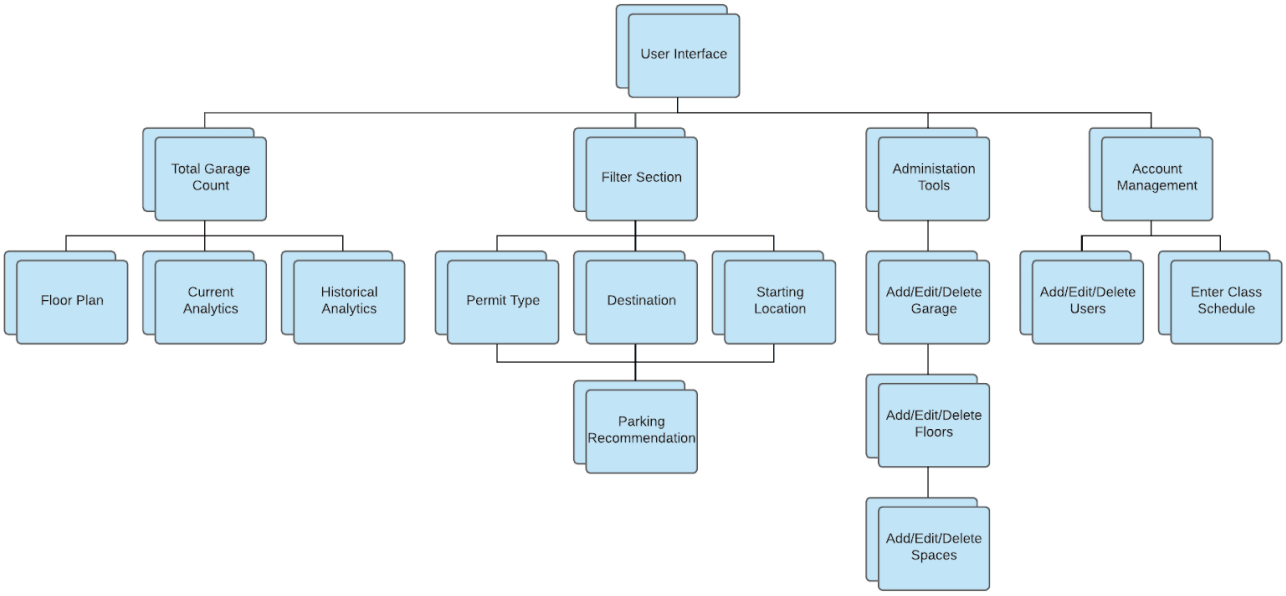

Figure 1 - User Interface Diagram

3 Specific Requirements

- This section was submitted separately