Lab II - Product Specification Outline

CS 411W Lab II

Prototype Product Specification

For

ParkODU

Prepared by: Matthew Stevenson, Team Gold

Date: 02/26/2018

Table of Contents

List of Figures

List of Tables

1. Introduction

"Parking at ODU sucks, there are not enough spaces for everyone and if you are a commuter you better get to class an hour early if you want a spot. It is like The Hunger Games for parking spaces. May the odds be ever in your favor." (1) ParkODU intends to approach the frustrations with parking conditions to which drivers face while parking at ODU while providing a tool for both drivers and administers of the system.

1.1. Purpose

ParkODU is a web application composed of tools to help users find available parking spaces and allows ODU Parking and Transportation Services to manage parking more efficiently. The application is only a software solution that is dependent on hardware implementations of vehicle detection technology including infrared sensors or IP based cameras. The intent of the application is to improve both the parking experience for drivers while providing a more efficient method to improve traffic flow, management of peak parking times, and promote better assessments toward future planning based on parking needs.

1.2. Scope

ParkODU is designed to be used by students, faculty, parking administration, and any other campus visitors. Users can interact will mechanisms within the application for whichever task the wish to accomplish. For drivers, choosing the best parking location is conducted by searching for that parking space through the Google Maps API The algorithms describe how the calculation of best parking space is determined and are detailed more in depth in the Lab 1 paper, Team Gold ParkODU Description (Stevenson, 2018). Park ODU will provide access levels for management of garages within the web application otherwise users will not need to provide login credentials unless they intend the desire to save their preferences. User accounts will be stored within a database. ParkODU is written in Java and implemented within the Spring Framework as referenced within the Lab 1 (Stevenson, 2018).

[This section intentionally left blank]

1.3. Definitions, Acronyms, and Abbreviations

The below terms address any gaps in knowledge for technologies or terms referenced to be used alongside the ParkODU application.

IP Camera – Camera that allows network connectivity.

Infrared Sensor – Sensor that detects motion by projecting infrared beams in defined direction. If the beam is broken, motion is detected.

ODU Parking and Transportation Services – entity at ODU that manages parking.

Java – coding language that utilizes .jar files after compilation.

Spring Framework – framework designed for Java web applications as referred to in Lab1 paper, Team Gold ParkODU Description (1).

Google Maps API – integration with Google Maps to provide a resource in determining location based on geolocation as referred to in Lab1 paper, Team Gold ParkODU Description (1).

GUI – graphical user interface to which users can interact with the toolkit ParkODU displays.

Access Level – permissions on a user to which they are allowed to view within the GUI.

Hazelcast – memory store interleaved with Spring Framework to provide quick lookups of cached data stored as backup in database as referred to in Lab1 paper, Team Gold ParkODU Description (1).

MongoDB – database with schemas are referred to a documents as referred to in Lab1 paper, Team Gold ParkODU Description (1).

JSON object – information represented as a string object. MongoDB documents are written as JSON objects.

Administrator user – user with the ability to login to ParkODU and configure components which normal users cannot access.

Virtual Machine – host machine created in a virtual environment.

[This section intentionally left blank]

1.4. References

Rogers, Emily. Dear Future ODU Students. (2017, August 28). Retrieved November 02, 2017, from https://www.theodysseyonline.com/dear-future-odu-students. (1)

Stevenson, Matthew. (2018, February). Lab1 Team Gold ParkODU Description. (2)

1.5. Overview

Team Gold ParkODU's product specification gives the user a better understanding with how ParkODU can enhance parking at ODU with a detailed account to its functionality.

2. General Description

Figures 1 and 2, show the major functional component diagram for the real world and prototype of ParkODU. The two diagrams are almost identical with the main difference between the two is the prototype will have a simulated input for vehicle detection technology and signage output. The differences and implemented prototype that relates to the real-world product will be further explored in section 2.1, The Prototype Architecture Description.

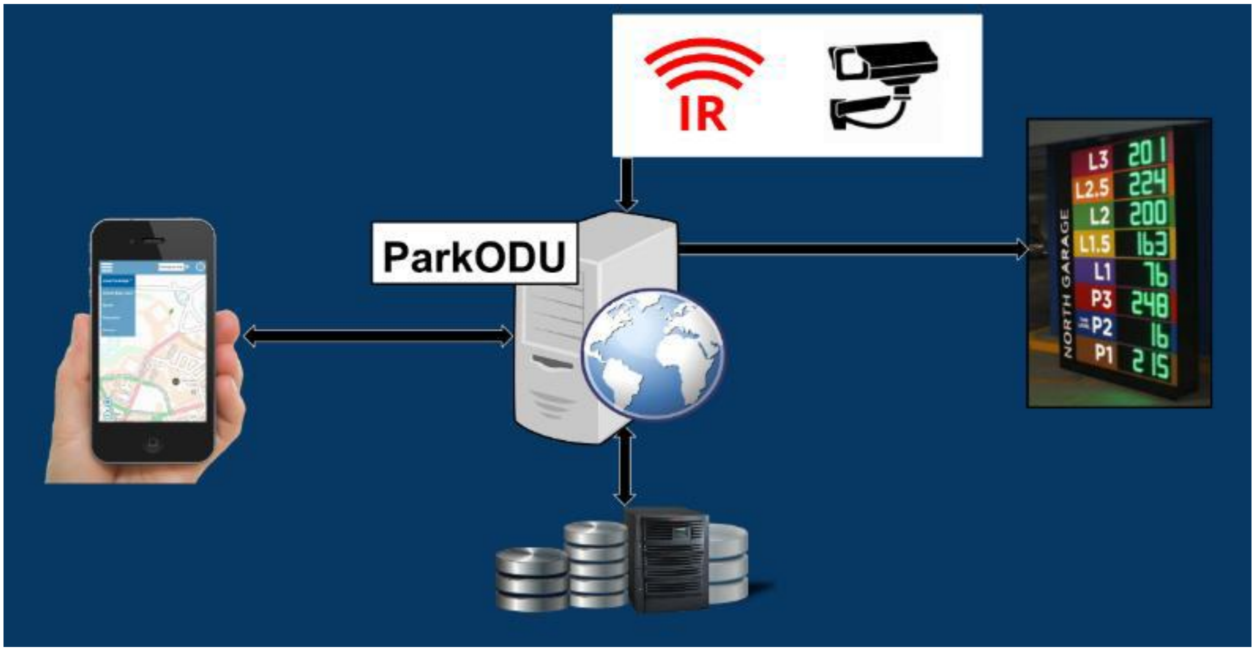


Figure 1. – Major Functional Components Diagram

[This section intentionally left blank]

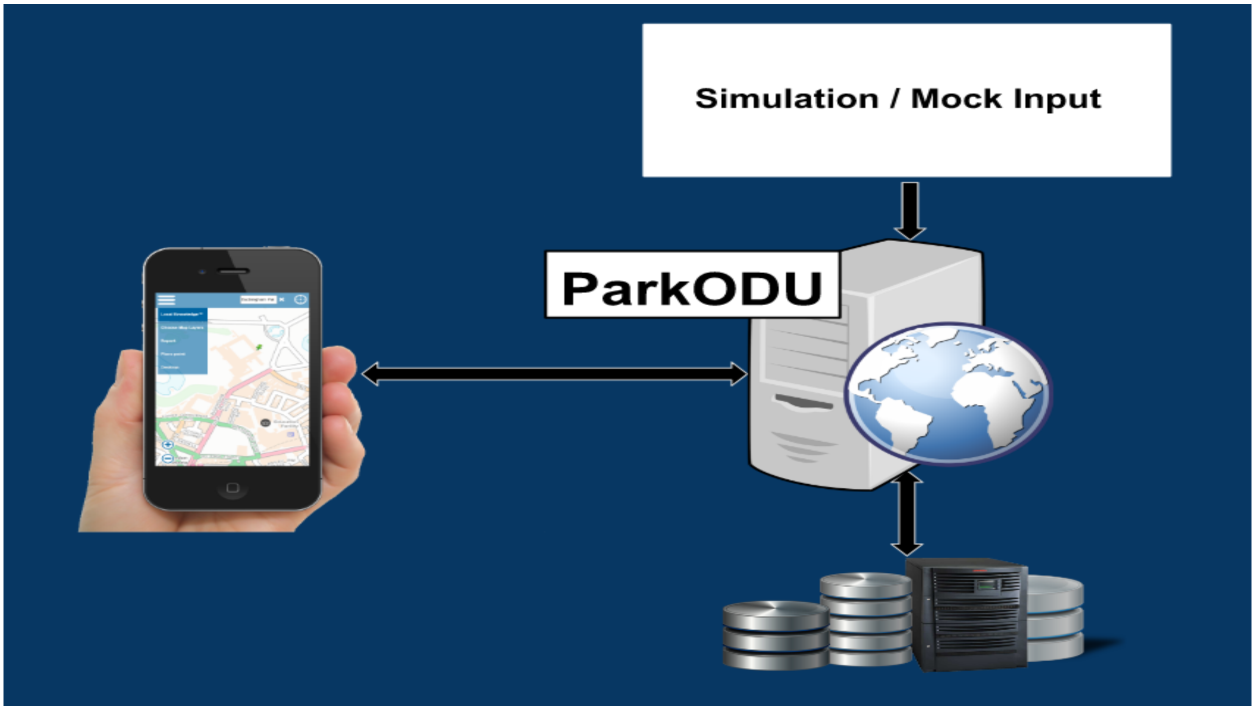


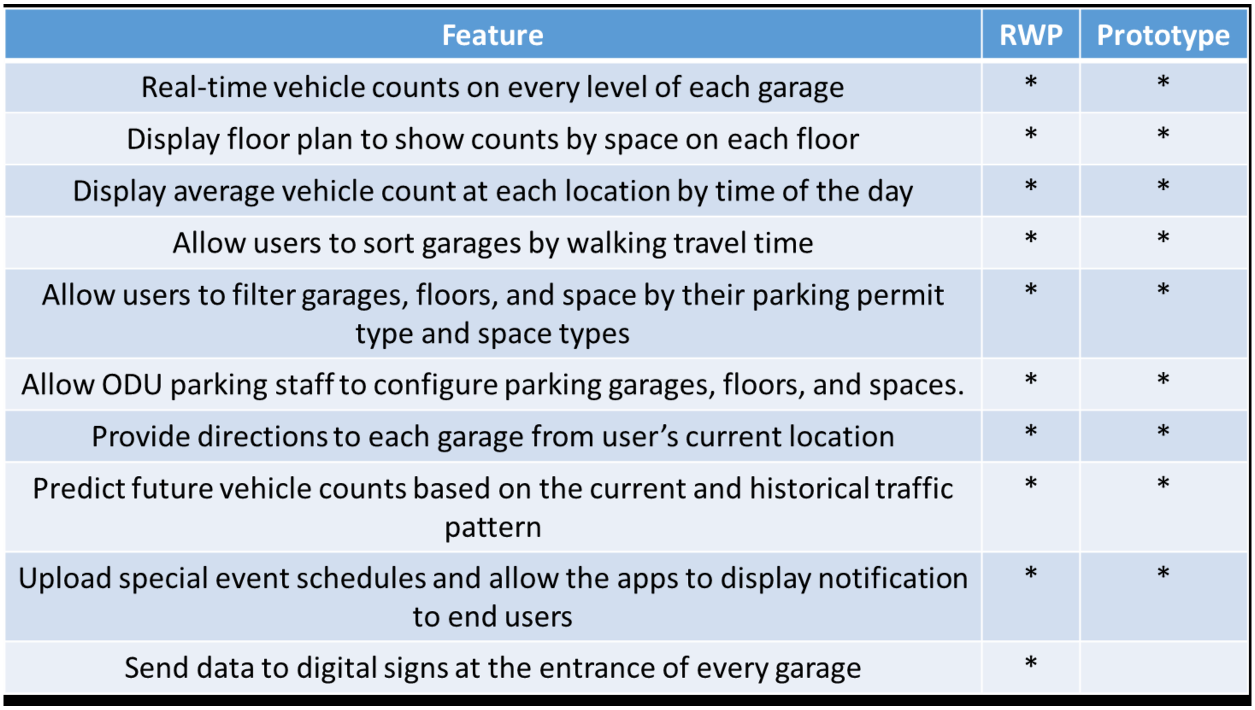
Figure 2. – Prototype Major Functional Component Diagram

The major components of ParkODU is the web application which will house the GUI to which users can interact with represented by the hand holding the phone depicted in both in relation to ParkODU in both Figures 1 and 2. ParkODU serves as a frontend to services implemented to which users are given access. Access is saved based on users created in both the database and memory store through Hazelcast. ParkODU

2.1. Prototype Architecture Description

The ParkODU prototype will consists of the same components as the real-world product and will provide all the same functionalities as the prototype, but will be simulated rather than actual input from vehicle. Figure 3 show the architecture of Team Gold ParkODU.

[This section intentionally left blank]



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. - Features of ParkODU between the Prototype and the Real-World Product

Both the prototype and the real-world product will be have full functionality, but the prototype will be unable to output data to digital signs as shown in Table 1. ParkODU will be designed to be compatible with all web browsers.

2.2. Prototype Functional Description

The functionality will be displayed in the prototype by creating a simulation program to represent the changes of parking conditions updated to ParkODU by sending JSON objects. JSON objects will also be used to update ParkODU in the real-world implementation for each functionality changed by a user with features they are able to access given their access level. The prototype will only consists of a single administrator user which has access to configure all parking the prototype would manage in the real-world implementation. The administrator user would have the ability to view all portions of the prototype. All users will have access search for the best parking location through the Google Maps API, view parking conditions, and import their schedule. User schedules will only be able to be saved for later use given they create a username and password suggested to be created after the schedule is imported. Tis feature is only an option to the user the retain preferences. The database will store both the credentials of the user and the schedule that was imported specific to the user.

2.3. External Interfaces

This sections is intended to define external devices needed for the ParkODU prototype. The prototype will consist of a virtual machine for both the prototype and the program running the simulation so no external devices will be necessary. The real-world implementation of ParkODU would require vehicle detection technology and digital signage as defined in 2 General Description.

Appendix

The software requirement for Team Gold ParkODU would include the user has a compatible browser.

[This section intentionally left blank]