Lab 2 – ParkODU Prototype Product Specification

Cody R. Coughenour

CS 411

Professor Thomas J. Kennedy

26 February 2018

Version 1

Table of Contents

1. Introduction 2
1.1 Purpose 3
1.2 Scope 3
1.3 Definitions, Acronyms, and Abbreviations 4
1.4 References 6
1.5 Overview 9
2 General Description 9
2.1 Prototype Architecture Description 9
2.2 Prototype Functional Description 10
2.3 External Interfaces 11
2.3.1 Hardware Interfaces 11
2.3.2 Software Interfaces 11
2.3.3 User Interfaces 13
2.3.4 Communications Protocols and Interfaces 15
3. Specific Requirements 15

List of Figures

Figure 1 – Prototype Major Functional Components Diagram 10
Figure 2 – Parking Garages Database Schema 12
Figure 3 - Events Database Schema 12
Figure 4 - User Accounts Database Schema 13
Figure 5 – Search Prototype Web Page 14
Figure 6 - Search Results Prototype Web Page 14
Figure 7 - Floor Plan Prototype Web Page for Floor 5 of Garage D 15

List of Tables

Table 1 – Features of Real-world Product and Prototype 4

1. Introduction

1.1 Purpose

ParkODU gathers and displays parking space availability for the user to view on the web. This information is organized by garage, floor, and space. The user can enter their preferences for parking predictions and recommendations. ParkODU also comes with tools for the organization and management of parking structures.

A driver will use ParkODU to make parking decisions. The driver will be able to use real-time data to select their desired parking garage. They will be able view a floor plan of the garage to see exactly which spaces are available. The driver will also be able to refine their search with additional factors such as the permit type of the space and the driver's estimated time of arrival. ParkODU will also provide directions to the selected garage.

An administrative user will be able to use ParkODU in all the same ways that a normal user, or driver, will with a few additions. The administrator will be able to add, edit, and delete garages, floors, and spaces. They will also be able to create and send parking notifications that affect space availability.

ParkODU will interface with a chosen vehicle detection system of management's choice. However, it is important to note that ParkODU is not itself a vehicle detection system. ParkODU is instead a software solution for sharing and utilizing the information provided by vehicle detection systems.

1.2 Scope

ParkODU aims to help ODU better utilize its parking spaces. The number of spaces available should be sufficient, yet frustrations are still expressed. Through better information

sharing, drivers should be able to avoid guessing incorrectly about which spaces are available and avoid such frustrations.

This prototype will use simulated parking data that will mimic how physical vehicle detection systems share information with ParkODU. The prototype will be a web application that demonstrates nearly all the real-world product's features except for the mobile application and the ability to display parking information on digital signage. Table 1 summarizes these features.

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 Real-world Product and Prototype

1.3 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)

Google Maps API - API provided by Google for mapping, navigation, and places.

High Availability - High availability is a characteristic of a system, which aims to ensure an agreed level of operational performance, usually uptime, for a higher than normal period. Modernization has resulted in an increased reliance on these systems.

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 tim

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

1.4 References

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

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

Car counting solutions. (n.d.). Retrieved on October 10, 2017 from http://www.puretechsystems.com/solutions-car-counting.html. (9)

Coughenour, Cody. Lab 1 – ParkODU Description. Version 2. (2018, February).

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

Hazelcast the Leading In-Memory Data Grid. Retrieved on January 23, 2018 from https://hazelcast.com

How Much Does a Parking Garage Cost? Retrieved on November 02, 2017 from http://www.parking.org/2016/01/19/tpp-2013-09-how-much-does-a-structure-cost. (6)

Inductive Loops. Retrieved on January 24, 2018, from http://diamondtraffic.com/technicaldescription/124 (17)

IntelliJ IDEA: The Java IDE for Professional Developers by JetBrains. *IntelliJ IDEA*, Jet Brains, Retrieved on January 18th, 2018 from http://www.jetbrains.com/idea/.

Is University Parking a Common Grievance? Parking Today Media. September 2011. Retrieved on September 2017 from http://www.parkingtoday.com/articledetails.php?id=1072. (8)

ODU Campus Parking Map. Retrieved on 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.)

ODUOnline at a Glance. ODU Online. Retrieved on January 24, 2018 from https://online.odu.edu/about/at-a-glance (14)

Old Dominion University | Student Life. U.S. News Best Colleges. Retrieved on January 24, 2018, from https://www.usnews.com/best-colleges/old-dominion-3728/student-life (12)

Old Dominion University | Campus-Info. U.S. New Best Colleges. Retrieved on January 24, 2018, from https://www.usnews.com/best-colleges/old-dominion-3728/campus-info (13)

Operating Budget and Plan. Old Dominion University. Retrieved on November 02, 2017 from https://www.odu.edu/content/dam/odu/offices/budget-office/docs/opplan2017.pdf. (65)

Parking and Traffic Procedures. Old Dominion University. Retrieved on November 02, 2017 from https://www.odu.edu/content/dam/odu/offices/parking-and-transportation-services/docs/parking-transportation-rules-and-regulations.pdf. (64)

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

Robust Vehicle Detection under Various Environments to Realize Road Traffic Flow Surveillance Using an Infrared Thermal Camera. Retrieved on January 24, 2018 from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4339787/ (16)

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

Spring: the source of modern java by Pivotal. Retrieved on January 23, 2018 from http://spring.io

Team Gold. "ParkODU." December 2017. PowerPoint presentation.

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. (2)

Ultrasonic Sensors vs Infrared (IR) Sensors. Retrieved January 24, 2018, from https://www.maxbotix.com/articles/ultrasonic-or-infrared-sensors.htm (15)

University Facts & Figures. Old Dominion University. Retrieved on November 2, 2017, from https://www.odu.edu/about/facts-and-figures. (3)

Vehicle Counter. Retrieved on October 10, 2017 from https://www.kiwisecurity.com/.

Vehicle counting & detection systems. Retrieved on October 10, 2017 from https://www.swarco.com/stl/Products-Services/Parking-Solutions/Parking-guidance/Vehicle-counting-detection-systems. (11)

What Is MongoDB? Retrieved on January 23, 2018 from https://www.mongodb.com/what-is-mongodb.

Why Hazelcast IMDG? Retrieved on January 24, 2018, from https://hazelcast.com/why-hazelcast/imdg/ (18)

1.5 Overview

This product description paper continues with details on ParkODU's prototype architecture, functionality, and interfaces. The specific requirements are also given.

2 General Description

2.1 Prototype Architecture Description

The prototype for ParkODU will be separated into three main components. These components are the ParkODU web application, the simulated input from a REST client application, and Hazelcast databases backed up by a MongoDB.

The web application queries for and processes data based on user interaction with the application's front end. The web application also provides administrative tools.

The simulator represents the hardware for vehicle detection via a REST client application. The REST client application will send requests to ParkODU REST endpoint and update the vehicle counts.

The databases store json documents that represent the parking garages, events, and user accounts.

Figure 1 shows the prototype major functional components.

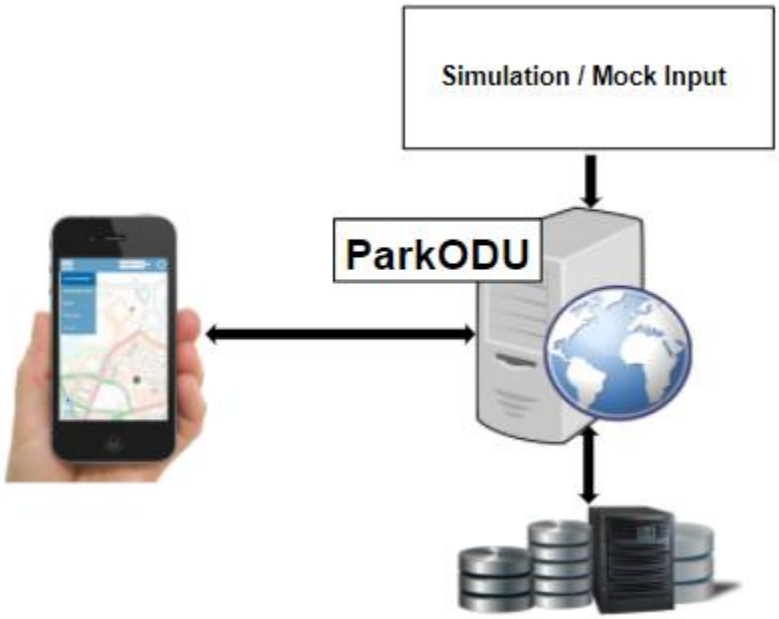

Figure 1 – Prototype Major Functional Components Diagram

2.2 Prototype Functional Description

ParkODU is a web application accessed via the user's internet device. The user can connect to the application as a guest or with a user-created account. Once connected the user can view vehicle counts, which are updated in real time. If they would like, they can view a detailed floor plan where colors are used to show exactly which spaces are available. Occupancy information is stored to a database, allowing the user to view average vehicle counts at different locations and times. ParkODU can also filter space availability by permit type(s) and sort garages by their walking distance to another building.

ParkODU uses historical and current data to recommend a parking location based on a predicted arrival time. Users can add their schedules and preferences to the application if they create an account. This allows the user to quickly be recommended the closest available parking spot to their destination. The application also utilizes Google Maps navigation if the user desires.

ParkODU will have tools to make parking administrative tasks easier. Users with access to these tools will be able to add, edit, and delete garages, floors, and spaces. Administrative users can also add, edit, and delete user roles. Management will also be able to send notifications of events that affect parking and the system will display space availability appropriately.

2.3 External Interfaces

There are four types of external interfaces for ParkODU: Hardware, Software, User, and Communications Protocols.

2.3.1 Hardware Interfaces

ParkODU is a web application, thus it must be accessed via the web on an internet device.

2.3.2 Software Interfaces

ParkODU interfaces with three databases and provides an API for querying for garage and floor status.

The databases store parking data, administrator created events, and user accounts.

The Parking Garages Database stores information about the garage, floors, spaces, and sensors. Figure 2 shows the Parking Garages Database schema.

Figure 2 – Parking Garages Database Schema

The Events Database stores information about events and the affected parking locations. These events are created by administrative users. Figure 3 shows the Events Database Schema.

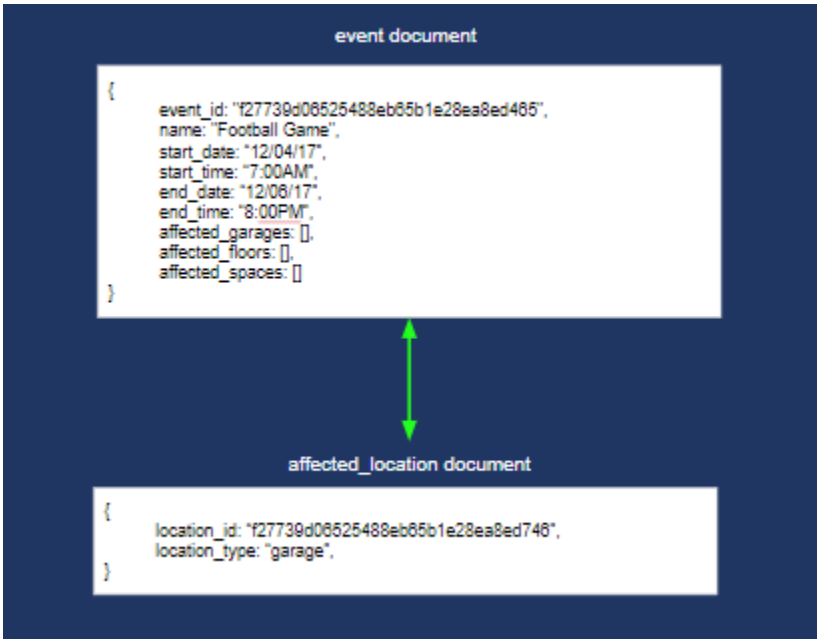

Figure 3 - Events Database Schema

The User Accounts Database stores information about the account such as the user's name, address, permissions, permit type, and schedule. Figure 4 shows the User Accounts Database Schema.

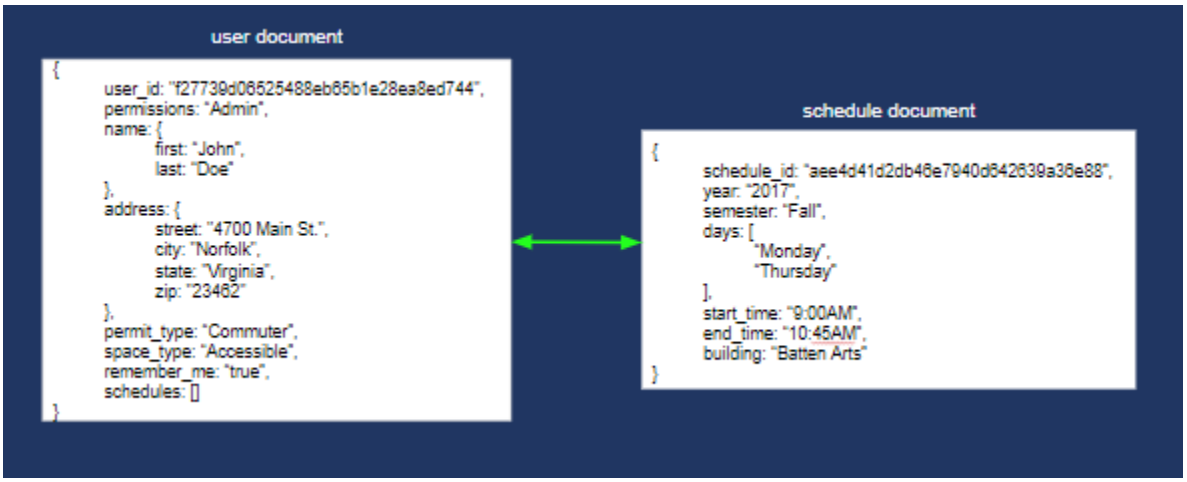

Figure 4 - User Accounts Database Schema

[This space intentionally left blank]

2.3.3 User Interfaces

- Flat-screen color display: to display the different pages of the web application. Figures 5, 6, and 7 show prototype versions of some of these pages: Search, Search Results, and Floor Plan.

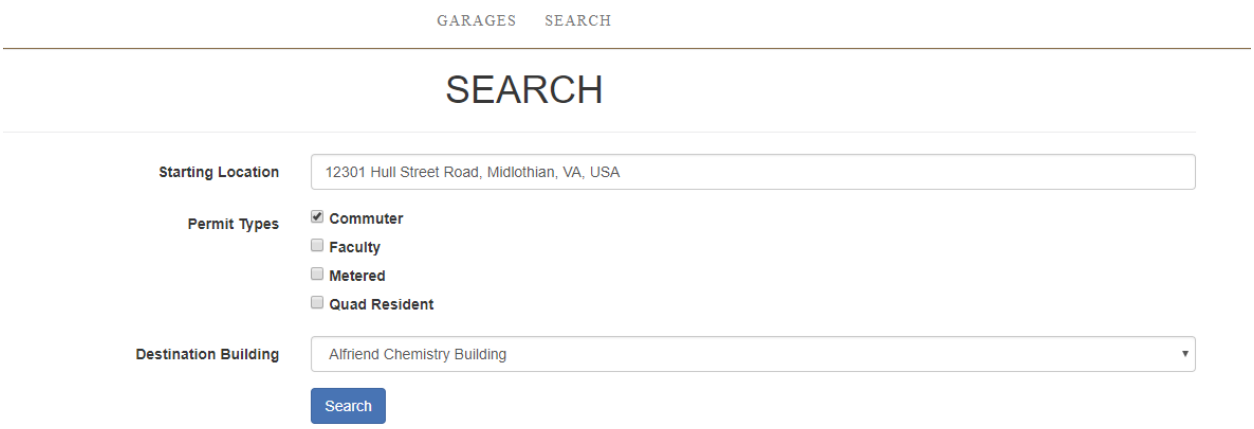

Figure 5 – Search Prototype Web Page

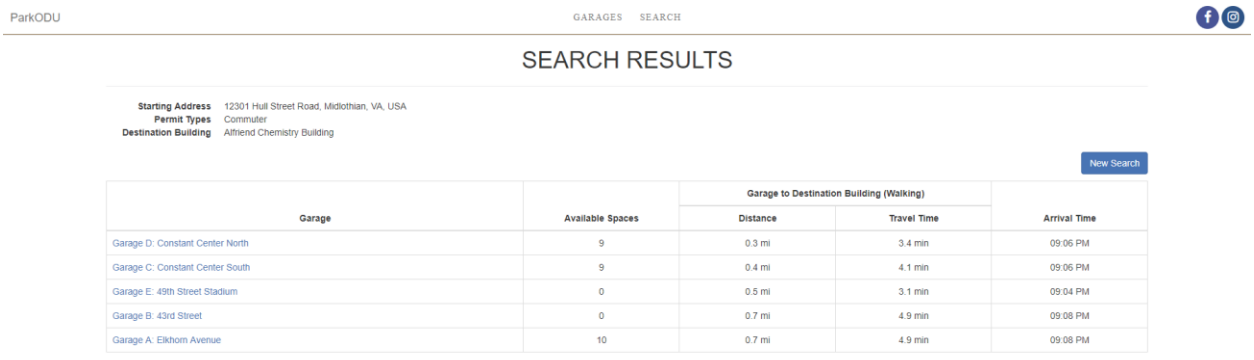

Figure 6 - Search Results Prototype Web Page

[This space intentionally left blank]

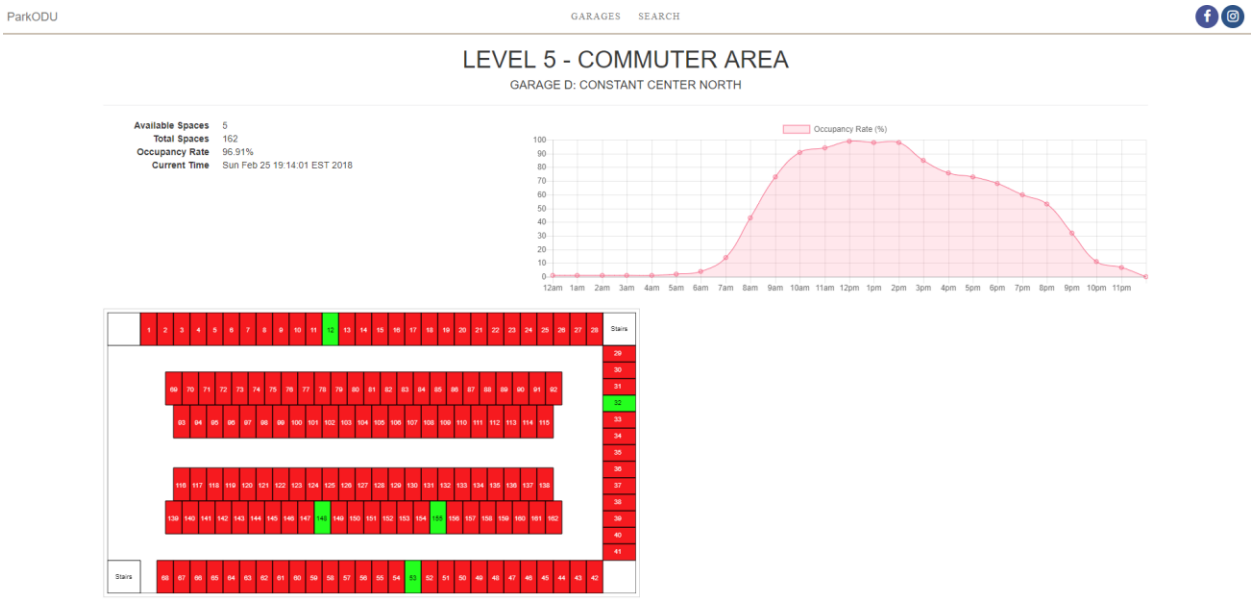

Figure 7 - Floor Plan Prototype Web Page for Floor 5 of Garage D

- Keyboard: for some features which require typed input, such as the login page or the starting location of a search.
- Mouse/Touchscreen: to select menu options and traverse the web pages.

2.3.4 Communications Protocols and Interfaces

ParkODU will be using a TCP/IP to establish internet connections to its databases. Users will need an internet connection to connect to the web application via SSH and TCP/IP.

3. Specific Requirements

This section was submitted separately.