

ParkODU Product Description

Professor Thomas Kennedy

3/29/2018

Isaac Asante

Lab 2 version 2

1 Introduction	3
1.1 Purpose	4
1.2 Scope	4
1.3 Definitions, Acronyms, and Abbreviations	Error! Bookmark not defined.
1.4 References	6
1.5 Overview	7
2 General Description	8
2.1 Prototype Architecture Description	8
2.2 Prototype Functional Description	9
2.3 External Interfaces	10
2.3.1 Hardware Interfaces	10
2.3.2 Software Interfaces	11
2.3.3 User Interfaces	11
2.3.4 Communications, Protocols and Interfaces	11
List of Figures	
Figure 1- Prototype Major Functional Components Diagram	10
List of Tables	
Table 1 - Features of Real-World Product and Prototype	9

1 Introduction

Old Dominion University has an enrollment of 24,828 students as of 2017. (3) Roughly 76% of the students enrolled are based off campus; majority of whom commute to classes every day.

There are five garages here at Old Dominion University which totals 3013 parking spaces. Out of the 3013 spaces, the 1511 faculty members as well as the roughly 9400 student commuters need to park every day. As time goes on, there will be more incoming students who own a car but currently, already existing parking lots are being replaced by academic buildings.

Geographically, there is constraint as to where to build more garages. In recent years, Old Dominion University has demolished parking structures and replaced them with academic buildings as they hold a higher priority over parking lots. The national average to build a parking garage is about \$8.5 million. Even if the university were to build more garages, they would all be far away from the main buildings on campus. The current state of ODU parking demands a solution that can efficiently utilize the parking spaces that are already available.

ParkODU is a software solution that analyzes parking availability in real time to help drivers find the vacant parking space closest to their destination at Old Dominion University. ParkODU will allow users to optimize the destination of the building they are going to, the type of parking permit they have, and find them the closest parking spot available. Users will be able to view all parking information related to Old Dominion University so visitors will also find parking with ease. The goal is to cut out time and frustration spent manually trying to find a vacant spot

while also avoiding a citation. To demonstrate all ParkODU's features, the prototype will simulate the garages.

1.1 Purpose

ParkODU uses vehicle detection technology to keep track of all vehicles in parking lots and garages. These detectors will gather and send parking space usage to information to ODU's Transportation and Parking Services and also display the parking availability in front all parking lots and garages. ParkODU will initially be intended for Old Dominion University and will be available on all major web browsers. ParkODU ultimately serves to better utilize the limited parking at Old Dominion University by providing the status of all parking lots and garages to any faculty member, student, or visitor in need of a place to park. This way, drivers relieve themselves of the hustle in finding an available spot and receiving a parking ticket.

1.2 Scope

The ParkODU prototype will have all the features and capabilities defined in the product description except for the digital signs outside of the garages and lots. The software will run on the web and on android/iOS devices which will connect with ParkODU's servers and then its database. Vehicle detection will also be simulated in place of detection systems. The data gathered will be fed into the software to demonstrate ParkODU's features and capabilities. Table 2 illustrates features of the real-world-product and the 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)
- 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 ParkODU
- 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 October 10, 2017, from <http://www.access-automation.co.uk/car-park-count-systems>.

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.)t

Burr, David W. "Is University Parking a Common Grievance?". Parking Today Media. September 2011. <http://www.parkingtoday.com/articledetails.php?id=1072>. September 2017. (8)

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

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

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

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/>. (6)

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

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>. (5)

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>. (4)

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.)

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

“Spring: the source of modern java by Pivotal” Retrieved January 23rd, 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)

University Facts & Figures. Old Dominion University. Retrieved November 02, 2017, from <https://www.odu.edu/about/facts-and-figures>. Accessed November 1, 2017. (3)

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

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>. (11)

“What Is MongoDB?” Retrieved on January 23rd, 2018 from *MongoDB*, www.mongodb.com/what-is-mongodb.

1.5 Overview

This product specification details the prototype architecture, functionality, and interfaces of ParkODU and how they differ from the real-world-product

[Space intentionally left blank]

2 General Description

ParkODU is a software solution that analyzes parking availability in real time to help drivers find the vacant parking space closest to their destination at Old Dominion University. ParkODU will allow users to optimize the destination of the building they are going to, the type of parking permit they have, and find them the closest parking spot available. Users will be able to view all parking information related to Old Dominion University so visitors will also find parking with ease. The goal is to cut out time and frustration spent manually trying to find a vacant spot while also avoiding a citation. To demonstrate all ParkODU's features, the prototype will simulate the garages.

2.1 Prototype Architecture Description

Vehicle detection will be simulated by a REST client application. ParkODU's REST endpoint will receive requests from the vehicle detection simulation's REST client application and update the vehicle counts accordingly. Data from the simulation will be similar to how ODU's parking traffic flows in the real world during a normal weekday and also during special events. The web and mobile applications will process data from the simulation and deliver available parking spaces to the user. Table 1 compares all the major features in the real-world product and the ParkODU prototype.

[Space 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 Real-World Product and Prototype

2.2 Prototype Functional Description

The ParkODU 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. Data from the APIs will be sent to ParkODU through requests to REST endpoint and stored in Hazelcast, an in-memory data grid. MongoDB, a document database, will be used as a secondary to Hazelcast to backup data.

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. Figure 1 illustrates the prototype major functional components.

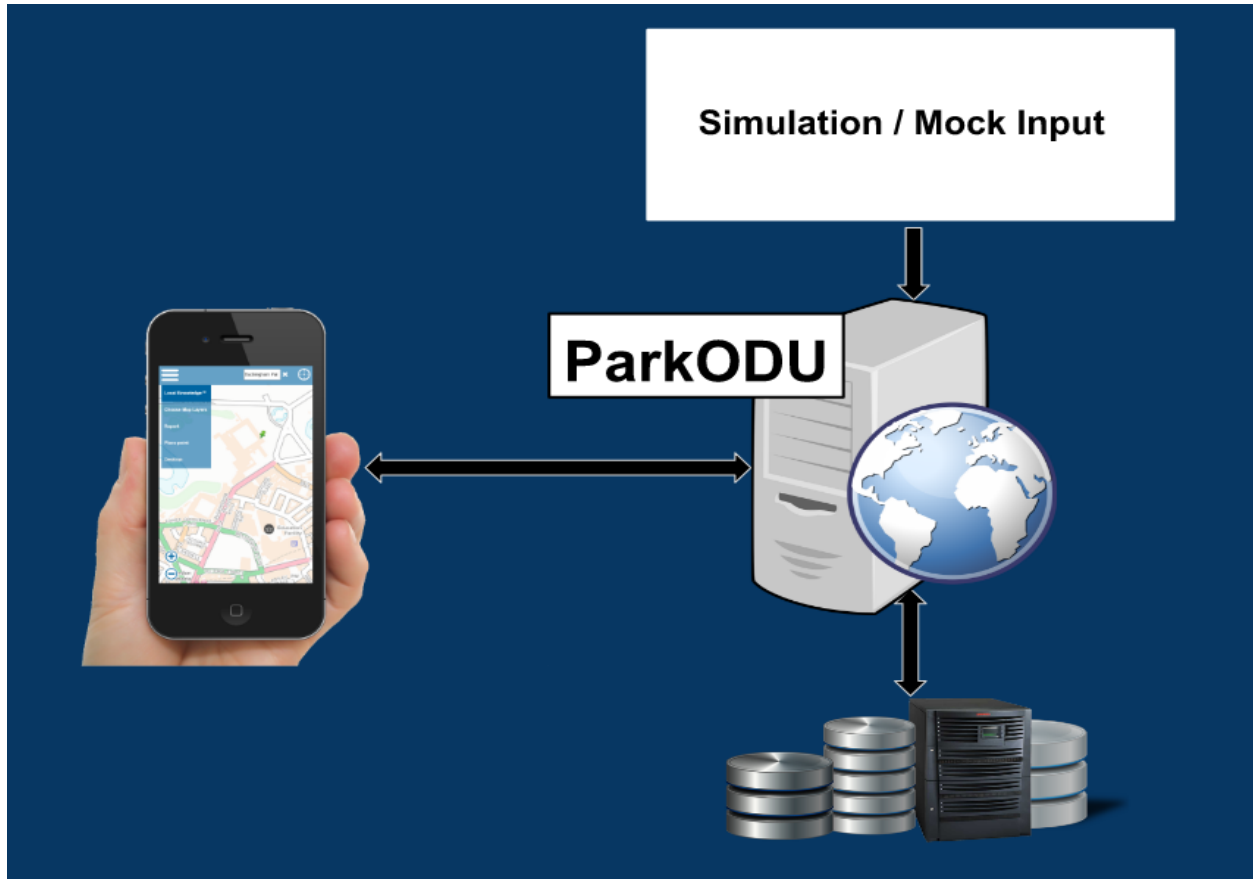


Figure 1- Prototype Major Functional Components Diagram

2.3 External Interfaces

There are four external interfaces in ParkODU:

- Hardware
- Software
- User
- Communications Protocols

2.3.1 Hardware Interfaces

The web-based application will run on servers at Old Dominion University

2.3.2 Software Interfaces

The ParkODU prototype will utilize the following software interfaces:

- Language: Java
- Framework: Spring Framework
- IDE: IntelliJ Community Edition
- Build Tools: Jenkins, Gradle
- Data Stores: Hazelcast, MongoDB
- Version Control: Git
- Third-Party API: Google Maps API

2.3.3 User Interfaces

- Users will access the ParkODU prototype via their preferred web browser

2.3.4 Communications, Protocols and Interfaces

ParkODU uses hazelcast nodes to communicate through TCP/IP to establish internet connection to its database.