Lab 1 – ParkODU Product Description

Michael Park

Old Dominion University

CS411W

Thomas Kennedy

29 January 2018

Version 1

Table of Contents

Figures and Tables

1. Introduction

Old Dominion University has a substantial number of commuting students and faculty members. According to statistics released by the university, 24,828 students were enrolled in 2016. Around 76% of enrolled students in 2016 lived off campus or 18,870 students. ODU reports that most of students living off campus drives their own vehicle. Assuming that only 50% of the off-campus students drive their own vehicle, the number is still as high as 9,400. ODU Facts and Figures reported 1511 faculty members working on campus.

There is a number of websites that allow students to review their universities. One example found at www.theodysseyonline.com is an article titled “Dear Future ODU Students”, which was posted by a former ODU student. The poster’s comment on ODU parking is highly negative; he compares the parking situation at ODU to *The Hunger Games*.

All students, especially new students, generally suffer more from the parking space shortage than faculty and staff members do. ODU reserves some spaces and even an entire garage and lot for faculty and staff use only, but the faculty and staff members are also allowed to park in other spaces, garages and lots that students generally use. In other words, there are no parking facilities reserved for students only. Continuing students and faculty and staff members eventually learn the parking traffic patterns at ODU and they are prepared to find parking at a number of other locations if needed, but new students may not know where else to go if the parking facility of their first choice turns out to be full.

ODU Transportation and Parking admits that the time of day students arrive on campus has a big impact on where they can find parking and the students may not find parking close to their destination as they would prefer. However, with five parking garages and over thirty-five surface parking lots on campus, there is a total of approximately 7,500 parking spaces. While the

more favored garages and parking lots completely fill up during busy hours, there are always available parking spaces in other less favored parking facilities, which may be located farther away from most educational and administrative buildings unless there is a special event.

Drivers who are unaware of the regular parking traffic pattern often unexpectedly find that the parking facility of their first choice is completely full and then drive to another location in search of available parking spaces; such situations can be frustrating. With no immediate plans to expand existing garages or parking lots or to create additional parking facilities, ODU needs to utilize their existing spaces more efficiently to lessen the effect parking space shortage. Students, faculty and staff members, and any other drivers who seek to park at ODU need a way of knowing which parking facilities have available parking spaces in real time; this will reduce frustration among all commuters. ParkODU is proposed as a software solution to this problem.

2. Product Description

ParkODU is a software solution to help drivers find available parking spaces at various ODU parking facilities and compile parking space usage data. The solution will be available in major web browsers such as Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, and Safari. ParkODU will compile real-time data sent from various types of vehicle counting systems and present facility occupancy data to ODU drivers. The main objective of ParkODU is to present the real-time vehicle counts in each parking facility so the drivers can avoid traveling to full parking facilities and instead go directly to another facility that has available parking spaces. The secondary objective of ParkODU is to provide ODU Transportation and Parking Services with parking facility usage data so they have an idea of how their parking facilities are utilized and use that information for future strategic planning.

2.1. Key Product Features and Capabilities

Many organizations and companies developed a number of applications that display parking facility occupancy information. ParkODU stands out among them with its extra capabilities to satisfy the specific needs of the university and its drivers.

2.1.1. Vehicle Counts and Displays

ParkODU's main feature is displaying real-time vehicle counts in each garage. ParkODU will provide a floor plan of each garage level, which will show the availability of each parking space. ParkODU will support statistical analysis to display the average vehicle counts by time of the day for every level of the garage. It will show walking distance from each garage to the user's destination building and sort the garages by the walking travel time. Users will be able to select their preferred parking space types (Regular, Reserved, Motorcycle) and their parking permit types and ParkODU will allow them to filter parking facilities based on their preferences.

- Counts vehicles by garage, floor, and space in real-time
- Displays a detailed floor plan of each garage floor
- Displays average vehicle counts at each location by time of day
- Allows users to filter garages, floors, and space by their parking permit type and the space type
- Allows users to sort garages by walking travel time to another building

2.1.2. Navigation

ParkODU will include navigation to any parking facilities, prediction of future vehicle counts at parking facilities based on the current and historical parking traffic, and ability to upload class schedules for customized search that looks for the best garage based on the location and start time of the uploaded class.

- Utilizes Google Maps navigation features
- Predicts future vehicle counts based on the current and historical parking traffic flow
- Allows faculty and students to import their schedule to the application for closest parking space recommendations

2.1.3. Management

The ODU parking staff and system administrators will have the ability to log on ParkODU's manager portal to configure parking garages, levels, and spaces. ParkODU will optionally provide support for digital signs for displaying real-time counts at the entrance of each garage.

- Allows the addition, editing, or removal of a garage, floor, or space
- Allows the addition, editing, or removal of user roles
- Provides easy configuration of occupancy signage
- Sends notifications of the events going on at ODU that affect parking
- Interfaces with any vehicle detection system

2.1.4. Other

The cost of ParkODU is expected to be low because the solution will use open source libraries and APIs. While the web application will work in any major web browser, native mobile app is currently planned to provide better user experience on mobile devices.

- Utilizes open source software
- Mobile support planned

The features and capabilities of ParkODU are designed to increase the transparency of ODU parking facilities' location, occupancy, and general parking traffic patterns. The increased

transparency will greatly help ODU drivers find available parking spaces without aimlessly stopping at multiple parking facilities.

2.2. Major Components (Hardware / Software)

Figure 1 shows the major functional components. There are 5 total components. Two components are specific to ParkODU: web server (Center) and database (Bottom). One component is the customer's choice of vehicle detection system (Top). The last two components are end user interfaces: web application (Left) and digital signs (Right).

Figure 1: Major Functional Component Diagram

2.2.1. Vehicle Detection Systems

The customer will select and purchase the vehicle detection system of their choice. The common types of detection technology are IR sensor, inductive loop, and IP camera. Some technologies, such as IR sensors, are capable of counting by space because they are installed at each individual parking space. Some technologies, such as inductive loops, are only capable of counting by floor because the loop is installed on each floor. The data collected by the detection system will be sent to the vendor's server. Normally, ODU accesses the vendor's web portal to

see the occupancy data. However, the vendors provide APIs that ODU software developers can use with their own applications. The software developers in ODU ITS will be responsible for building applications to interface with the vendor's API and send proper HTTP requests to ParkODU's REST endpoint (See *2.2.2. ParkODU Server and Database* for more details on development tools).

2.2.2. ParkODU Server and Database

ParkODU will be hosted on a cluster of physical or virtual servers. The customer will be responsible for providing or purchasing the server hardware. The data from the vehicle detection system will be sent to ParkODU via HTTP requests to a REST endpoint. The data will be written to Hazelcast in-memory data grid and to MongoDB as the secondary and backup storage. The web server will interface with Hazelcast for accessing the necessary data to service end users. In case the customer fails to provide machines, ODU CS department virtual machines will be used as the server hardware. ParkODU will utilize a number of open-source tools.

- **Programming Language: Java**

  Because ParkODU will be developed mainly with the Spring Framework, Java will be the primary programming language.

- **Framework: Spring Framework to support RESTful web service**

  "The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an addition to, or even replacement for the Enterprise JavaBeans (EJB) model. The Spring Framework is open

source.” This framework will be used to create RESTful web services that will process HTTP requests.

- **IDE: IntelliJ Community Edition**

  While the choice of the integrated development environment is up to the individual developer, IntelliJ Community Edition is an excellent choice for the Spring Framework.

- **Build Tools: Gradle**

  “Gradle is a build automation system that derives its roots from Apache Ant and Apache Maven while using Domain Specific Language used by Groovy. Gradle can recognize which parts of a build are already up-to-date and offers the flexibility for incremental and multi-scale builds.”

- **Data Storage: Hazelcast, MongoDB**

  “Hazelcast is an open source in-memory data grid based on Java. Hazelcast provides high-availability access to frequently used data while offering scalable solutions in a multi-node environment. The data grid created by Hazelcast offers redundancy and increases performance.”

  “MongoDB is an open source database solution that uses json documents. Document objects can vary, however, are able to adapt automatically. Mapping is distributed within the MongoDB framework which allows the opportunity for flexible and horizontal scaling.”

- **Version Control: Git**

  “Git is an open source version control content solutions that provides an interface for collaboration when a team is working together on a project. Git offers the availability to

control changes and keep each member up-to-date with the most recent revision based on source code uploaded to the hub."

- **Third-Party API: Google Maps API**

  The Google Maps API will be used to implement navigation and calculating walking distances.

2.2.3. End User Interface

ParkODU will provide information to end users mainly via web application. An optional interface will be digital signs installed at the entrance of parking facilities. The web application will run in major web browsers such as Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, and Safari. End users will access the application on desktop computers and on smartphones using the web browser of their choice. The digital sign will require another application that can retrieve data from ParkODU's REST endpoint and convert it to the data format the digital sign can read.

3. Identification of Case Study

ParkODU is being developed for Old Dominion University, specifically the ODU Transportation and Parking Services department. ODU Transportation and Parking Services, along with ODU ITS software developers, will adopt this solution and make the application available to all ODU drivers.

ODU currently uses T2Systems and their inductive loops for counting vehicles. ODU in-house developers created a web application that utilizes T2System's APIs to show the vehicle counts in Garage A, Garage D, Parking Lot 42, and Parking Lot 43. However, the current web application has limited functionality. The features of this application are displaying the vehicle counts and using color codes to indicate which facility has more available spaces.

While ParkODU is projected to be for ODU use only as of now, the possibility of this solution being adopted by other universities with some modifications is not completely out of the question.

Figure 2 shows the current process flow diagram when ODU drivers are searching for available parking spaces. The current ODU web application mentioned earlier was released only recently and many ODU drivers are not aware of its existence. Due to the limited functionality and the low usage by ODU drivers, the application is not included in this current process flow diagram. The key in *Figure 2* shows the frustration level of the drivers in the search of a parking space. The drivers may suffer negative consequences such as being late or receiving citations due to the fact that they do not know beforehand which parking facilities are full. ParkODU aims to reduce the frustration level by providing information to the drivers.

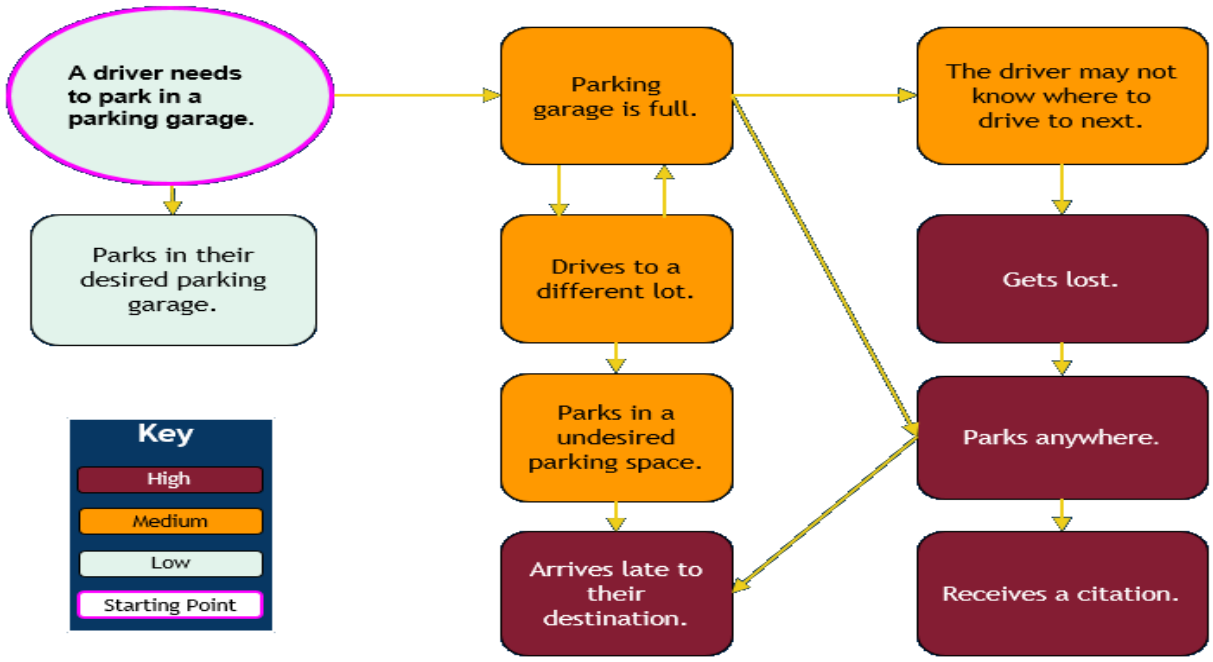

Figure 2: Current Process Flow Diagram

4. X Product Prototype Description

The prototype of ParkODU will support all of the features and capabilities of the web application defined in *2.1. Key Product Features and Capabilities*. However, it will not support digital signs. The data that would be normally obtained from vehicle detection systems, such as inductive loops, IR sensors, and IP cameras, will be simulated with a REST client application. The ParkODU prototype will compile the simulated data and demonstrate all the features and capabilities of the real working system.

4.1. Prototype Architecture

The major functional components of the prototype are shown in *Figure 3*. This prototype major functional component diagram is similar to the major functional component diagram of the real working system. Simulation will replace the vehicle detection technology and digital signs will not be supported. The simulation will be accomplished with a REST client application that will send proper HTTP requests to ParkODU's REST endpoint in order to update vehicle counts. The client will send the requests in a way that mirrors the actual ODU parking traffic patterns during normal operating hours on weekdays. The same development tools will be used for the development of the prototype as outlined in *2.2.2. ParkODU Server and Database*.

[This Space Intentionally Left Blank]

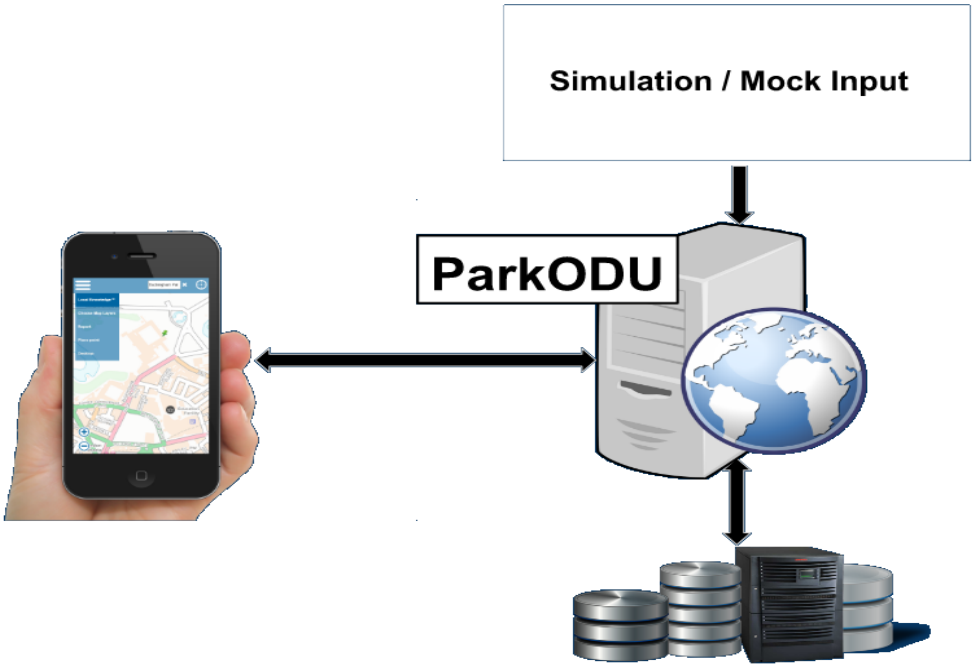

Figure 3: Prototype Major Functional Component Diagram

4.2. Prototype Features and Capabilities

The ParkODU prototype will support nearly all of the features and capabilities of the real working system. *Table 1* shows the features and capabilities that are available in the prototype. The technical, customer, and user risks along with possible mitigation are listed in *Table 2*, *Table 3*, and *Table 4*. *Figure 4* has the risk analysis matrix. The mitigation suggested for each risk is for the real working system. The way the prototype mitigates some of the risks is different. With the simulated data closely resembling the actual ODU parking traffic pattern, the prototype will demonstrate all of the non-trivial features with the same volume of data that the real working system is expected to handle; this addresses the technical risk, T8, in *Table 2*. The simulation of the vehicle detection technology addresses some of the risks arising from the coordination with the customer (C1, C5, and C7 in *Table 3*) and some of the technical risks from vehicle detection systems (T4 and T7 in *Table 2*). All of the features in *Table 1* are designed to accomplish the

objectives of providing real-time parking facility occupancy data to drivers and compiling parking facility usage data for the ODU parking staff.

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: Prototype Features and Capabilities

[This Space Intentionally Left Blank]

Risk	Description	Mitigation	Impact	Probability
(T1) Web Connection Failure	Application server fails to connect to the Web.	• Test the connection and ensure communication is regained.	Very High	Low
(T2) Database/Web Application Failure	Database/Web App failure may occur due to network settings being offline or unavailable.	• Verify the database and software communication through testing. • Establish dedicated clustered server environments for both database and web application server clusters to reduce possible downtime of ParkODU.	Very High	Low
(T3) Software Bugs	Software development opens up the possibility for bugs that may reduce functionality of the Web application.	• Software updates and debugging techniques will be administered routinely. • User Interface/User Experience (UI/UX) Testing • Regression testing and continuous integration	Very High	Very Low
(T4) Hardware Failure	Hardware including IR sensors, Garage Signage (optional), may not be functioning or require repair.	• Mark spot as hardware malfunction and send a service request to maintenance.	High	Medium

Risk	Description	Mitigation	Impact	Probability
(T5) Failure to Notify User of an Event	ParkODU is not updated with event schedules that may affect garage availability.	• Ensure ParkODU is updated with upcoming events. • Create a ScheduledTask to poll the event calendar through rest endpoints.	Very High	Low
(T6) Lack of Technical Knowledge	Minimal technical experience and/or programming familiarity needed to develop the application.	• Individually improve programming knowledge and provide training to less experienced members in area of deficiency.	Medium	Medium
(T7) Incompatible Input Format	Formatting of input is not compatible with ParkODU.	• Verify input formatting compatibility through testing.	Medium	Medium
(T8) Inability to Scale Under Load	As volume or customer count increases the database becomes slow or may fail.	• Expanding computing resources to handle the exponential growth of work with the use of database scalability. • Establish dedicated clustered server environments for both database and web application server clusters.	Very High	Very Low

Table 2: Technical Risks

Risk	Description	Mitigation	Impact	Probability
(C1) University Implements a Better Solution	The university implements a solution other than ParkODU.	• Show the customer how ParkODU's benefits and features are superior to competing solutions. • Offer ParkODU as an open-source solution.	Very High	Very Low
(C2) University Does Not Allow Access to Network	ODU ITS does not allow ParkODU to run on the university's network.	• Some departments within the university run things on their own networks. • Customer determines hosting location.	Low	Very Low
(C3) Customer Unable to Maintain Servers/Hardware	Customer will be unable to maintain the hardware utilized by ParkODU.	• Customer determines the most effective hardware solution with their implementation for ParkODU.	High	Medium
(C4) University Replaces All Garages and Lots with Other Buildings	All parking garages and lots are replaced by buildings.	• Parking will still be essential. The software will allow for reconfiguration as the university changes parking allocations.	Very High	Very Low
(C5) Customer Unwilling to Purchase Hardware	Customer may not agree to purchase hardware.	• Software will allow for multiple hardware implementations. • ParkODU will allow for manual toggling of parking space availability.	Very High	Medium

Risk	Description	Mitigation	Impact	Probability
(C6) Parking Lot Replaced by Parking Garage	The University builds parking more parking garages in place of parking lots.	• Ability to add/edit/delete parking objects.	Low	Low
(C7) Customer Purchases Partially Compatible Technology	Customer purchases detection hardware that does not support a certain functionality, such as count by space.	• The software can be reconfigured to support specific customer implementation.	Medium	Medium

Table 3: Customer Risks

[This Space Intentionally Left Blank]

Risk	Description	Mitigation	Impact	Probability
(U1) User is Distracted While Using the Application	User is distracted while using ParkODU.	• Provide safety notification. • Allow for ParkODU to auto-refresh in order to display current data without any additional user interaction.	High	High
(U2) No Internet Device	The end user does not have access to an internet device, such as a Smartphone or a computer, to use the mobile or web application.	• User is able to view occupancy signage while on campus. • User can use public resources such as a public library computer to access ParkODU. • User can utilize ParkODU's historical prediction feature to print future projections.	Very High	Very Low

Table 4: User Risks

[This Space Intentionally Left Blank]

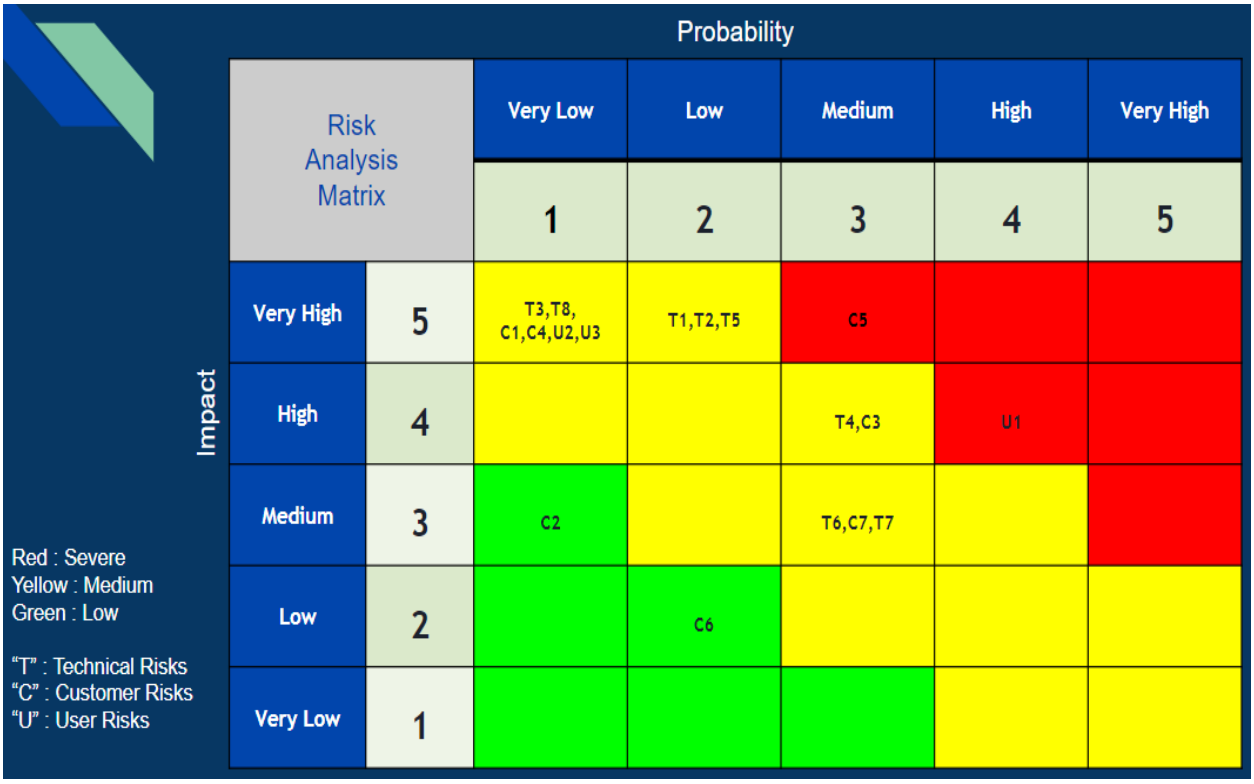

Figure 4: Risk Analysis Matrix

4.3.Prototype Development Challenges

The main challenge while completing the objectives of the prototype is obtaining knowledge and skills for various development tools, services and technology. The development of the ParkODU prototype will require substantial knowledge and skills in MongoDB, Hazelcast, Java programming, RESTful web services, and web programming (HTML5 and JavaScript). As seen in *Table 2*, the lack of technical knowledge (T6) can be detrimental to the development of the ParkODU prototype especially because ParkODU utilizes many open-source tools. It is expected that every team member devote a significant amount of time in personal research to understand all of the open-source tools required to complete the objectives of the ParkODU

prototype. It is also expected that team members mentor and help one another to understand how to use the development tools. Another challenge of the prototype development is the time constraint due to the complicated nature of the solution and many requirements that need to be met.

[This Space Intentionally Left Blank]

5. Glossary

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 value is the number of vehicles entering while a negative value is the number of vehicles exiting (Garage Rate = Driver Entry Rate – Driver Exit Rate)

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

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

Prediction - 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

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)

"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/.

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.)

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

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. "Lab 1 Collaborative Outline" January 2018. Lab 1 Outline.

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.