Lab 1 - ParkODU Product Description

Sangeet Mokha

CS 411

Professor Thomas J. Kennedy

29th January 2017 Sign out

Version 1

Table of Contents

1. Introduction

Old Dominion University is located in a coastal city of Norfolk Virginia. It's a successful entrepreneurial minded doctoral research university facing a problem as a result of years of inadequate parking allocation. This is due to an increasing student body, that includes both residents and commuters to the campus. According to data from Old Dominion's website, last year's enrollment 24,828 undergraduate students. Around 76% of these students live off campus, this figure also includes students who take classes online, yet the majority of the aforementioned group commute to the main Norfolk campus. Unfortunately, the effects of the inadequate parking have become a source of stress for students, staff, and visitors. (2) ODU currently offers 5 parking garages specified for faculty, metered, commuters, and other, totaling 3013 spaces. (4) However, roughly 9,400 student commuters need to park at ODU daily and 1511 faculty members (835 Full Time, 676 Part Time). (3) A quote from a current ODU student describing the situation, "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) This generally sums up the attitude of the student body and the university staff.

A common belief is that building more parking garages on campus can help solve this problem, however the national average cost to build a parking structure is $8.56 million. (6) Furthermore the current geographic size of Old Dominion University poses a constraint. Moreover, the university is more focused towards increasing academic infrastructure. The current lack of ODU parking spaces provides a demand for a more efficient method to utilize existing parking without building additional parking structures. During peak hours, of 10:00AM - 2:00PM, is the most difficult time for commuters to find a parking spot. The root causes are: "lack of signage and notifications for available spaces, preferences for specific parking locations, and limited choices during those hours."

Another way of resolving the situation is a software solution that analyzes parking availability in real-time and helps drivers find the vacant parking space closest to their destination, which brings in ParkODU. ParkODU is a web application designed for the use of drivers that need to park at the university. The interface will allow the driver to view all relevant parking information related to Old Dominion University, so drivers may find parking with ease. Instead of wasting time and fuel manually roaming around for a parking spot the goal is to automate that process. The software solution provides real-time analysis of parking availability to help drivers find the closest vacant parking space to their destination. This application should optimize parking as it includes the starting location, permit type, and destination to allow the drivers to find the best parking space available, saving time, resources, and effort. The prototype will use a simulated garage and will demonstrate nearly all of the product's features.

2. Production Description

ParkODU is an application to help drivers find available parking spaces closest to their destination on campus. It is accessible on any web browser and will also be available as a mobile application on an Android and an iOS sometime in future. The application will compile data gathered in real time by various vehicle counting systems installed in garages and parking lots and make the information available to drivers. The main objective of this software is to inform drivers of the available parking spaces in garages in real-time so drivers can avoid overfull parking facilities and instead go directly to the open facilities. The secondary objective of ParkODU is to provide ODU Transportation and Parking services with parking data for analysis of current utilization rates of their parking facilities for use the information for future strategic planning.

2.1 Key Product Features and Capabilities

ParkODU has many features that have not been implemented in other parking applications. There are managing features that lets the driver import their schedule for custom parking recommendations. It Displays a detailed floor plan of each garage floor and the available spaces and average vehicle counts at each location by time of day, these features are further described in section 2.1.1, 2.1.2 and 2.1.3.

2.1.1 Vehicle Count and Display

ParkODU has the ability to track and monitor parking availability in the garages in real-time. These are done by vehicle counts in the garage by floor and by space, then updates are sent to the application. There are few unique capabilities that ParkODU has with Vehicle Count and Display:

- ParkODU displays a detailed floor plan for each garage floor with the available spaces.
- It updates and displays average vehicle count in every location during rush hours or operating hours.
- It gives every driver the option to filter garages, floors, and space by the parking permit type that they choose to purchase.
- It also allows the drivers to sort garages by the walking travel time to their desired building

2.1.2 Navigation

ParkODU provides navigation to the drivers which might be needed for providing eta or if they are not sure of the route. It uses Google Maps API for that functionality. There are few unique capabilities that ParkODU has that deal with navigation:

- ParkODU is able to predict future vehicle counts. It can predict the overall space availability in a particular garage at the drivers selected time of arrival. The application can predict the “best” garage to a driver by predicting the garage that has the maximum number of available spaces and is closest to the driver’s destination.
- An extension of the import schedule feature mentioned earlier gives the opportunity to the faculty members and the students to check for closest parking space recommendations at their time of arrival.

2.1.3 Management

ParkODU will be managed by ODU parking and transportation services. They will have the option to configure the application easily with an add, edit or remove feature for the garage, floor, space, a user role or any occupancy signage if they decide to change it the future. There are few unique capabilities that ParkODU has with management:

- The application has the feature to send notifications to the driver of the events going on at ODU that may affect the parking availability.
- It interfaces with any vehicle detection system that the customer will decide to use, such as IR sensors, inductive loops, IP cameras, or other means of detection to update garage counts and floor plans which will be updated on the application.

2.1.4 Others

ParkODU has Mobile support capability that has not yet been designed, that will be determined later if the team will have enough time to implement that feature.

2.2 Major Component

The major components for ParkODU will depend on the vehicle detection device the customer decides to choose. If the customer chooses IR sensors they will be installed at each parking space in all parking garages which will detect if the space is empty or not to count by space. If the customer chooses inductive loops which will not have the count by space feature, they will count only by floor and send the data to the vendor’s servers which will be further accessed by ODU for occupancy data. Sometimes, vendors also provide APIs, which allows ODU's software developers to create their own app or interfaces. The data obtained from the vendor's APIs will be sent to ParkODU via requests to REST endpoint. The data will be backed up in Hazelcast in-memory data grid for fast queries and MongoDB will be used as the secondary and backup data storage. The database and the web application will be installed on physical or virtual machines provided by the customer. ParkODU database and application servers will service the web application, Android/iOS app, and the digital signs.

2.2.1 MFCD Summary

ParkODU application is hosted on a server or cluster of servers. These servers can be physical or virtual. ParkODU uses the installed vehicle detection technology such as IR sensors, inductive loops, IP cameras, or other means of detection to update garage counts and floor plans which will be updated on the application. ParkODU stores the data in Hazelcast and MongoDB, both are open source databases. This information can then be displayed on signage or accessed via the web on a smartphone or computer.

2.2.2 Out of Box Requirement

There are few requirements that will be needed for ParkODU to be in use. As it is an application the users will need a web browser either on a smartphone or on a computer. For ParkODU the customer will need some sort of vehicle detection technology to interface with ParkODU, such as IR sensors or cameras. Lastly, the customer will need to host the application on a server either physically or virtually.

2.2.3 Development Tools

ParkODU is developed using seven different developing tools that includes the framework which is the platform of the language that is used, the language for the framework which is java, the IDE which is the IntelliJ community Edition , the build tools to build the program that are Gradle and Jenkins, the data stores to back up the data which are Hazelcast and MongoDB, the version control which is Git, and a third party API for the navigation feature in the application.

2.2.3.1 Language: Java

Java is a general purpose programming language designed for minimal dependencies at implementation allowing it to run anywhere once written. It is class based and object-oriented language that when compiled can run on any computer with the Java Virtual machine(JVM). Even though it is known to be slower and more of a memory hog it has “better code analysis (such as inner classes, the stringBuilder class, optional assertions, etc.).” Their are also numerous frameworks for client-server web interactions.

2.2.3.2 Framework: Spring Framework

The Spring Framework is an open source application framework and inversion of control container for the Java platform. Spring framework features could be used by any java platform. “Reactive Spring represents a platform wide initiative to deliver reactive support at every level of the development stack—web, security, data, messaging, etc.” (12)

2.2.3.3 IDE: IntelliJ Community Edition

IntelliJ community Edition is an open source version of IntelliJ IDE (Integrated Development Environment). It has an intelligent code editor that has the ability to understand codes in Java, XML, Groovy and many other languages. It also supports build tools like Ant and Maven which we will be using for ParkODU.

2.2.3.4 Build Tools: Jenkins and Gradle

Jenkins is an open source automation server written in Java. It helps in automating non-human parts of software development process, with continuous integration and providing technical aspect of continuous delivery. As it is a server-based system it runs on Apache Tomcat. It supports version control tool that will be used to develop ParkODU and execute Apache Ant and Apache Maven also used in developing ParkODU.

Gradle is an open source 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. Gradle can help teams build, automate and deliver any software much faster.

2.2.3.5 Data Stores: Hazelcast and 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.

Whereas, MongoDB is also an open source database solution that uses json documents, which is a language independent data format. 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.

2.2.3.6 Version Control: Git

Git is an open source version control content solution for source code management in a collaborative development environment. It allows group members to edit files in a non-linear, distributed fashion since it can track changes to files in the repository. This allows editing without interference from colleagues. The Git directory on each members local machine is a full repository with version tracking that can operate independent of the main repo. Members then push updates to the main and the central commit log.

2.2.3.7 Third-party API: Google Maps API

Google Maps API is used to create engaging web and mobile application using google maps. It is a powerful mapping platform that includes driving directions, street view imagery and much more. This API is used for the purpose of navigation by the ParkODU application.

2.2.4 Software description

ParkODU is an application that acts as a tool to collect, store, and communicate parking information to the drivers and the customer. It makes it easier for the driver to help them park at ODU with ease and it makes it easier for ODU parking and transportation services to manage their garages easily by providing an interface for updating vehicle counts with the help if the vehicle detection technology. ParkODU also includes the necessary algorithms, functions, and data structures to provide the functionalities listed in section 2.1.

3. Identification of case study

ParkODU is being developed as a software solution for drivers driving to Old Dominion University that find it difficult to find an available parking spaces in their desired garage. Most specifically students and the faculty members who drive to ODU daily. ParkODU will also help the Parking and Transportation Services at Old Dominion University to manage their garages by having it controlled through ParkODU.

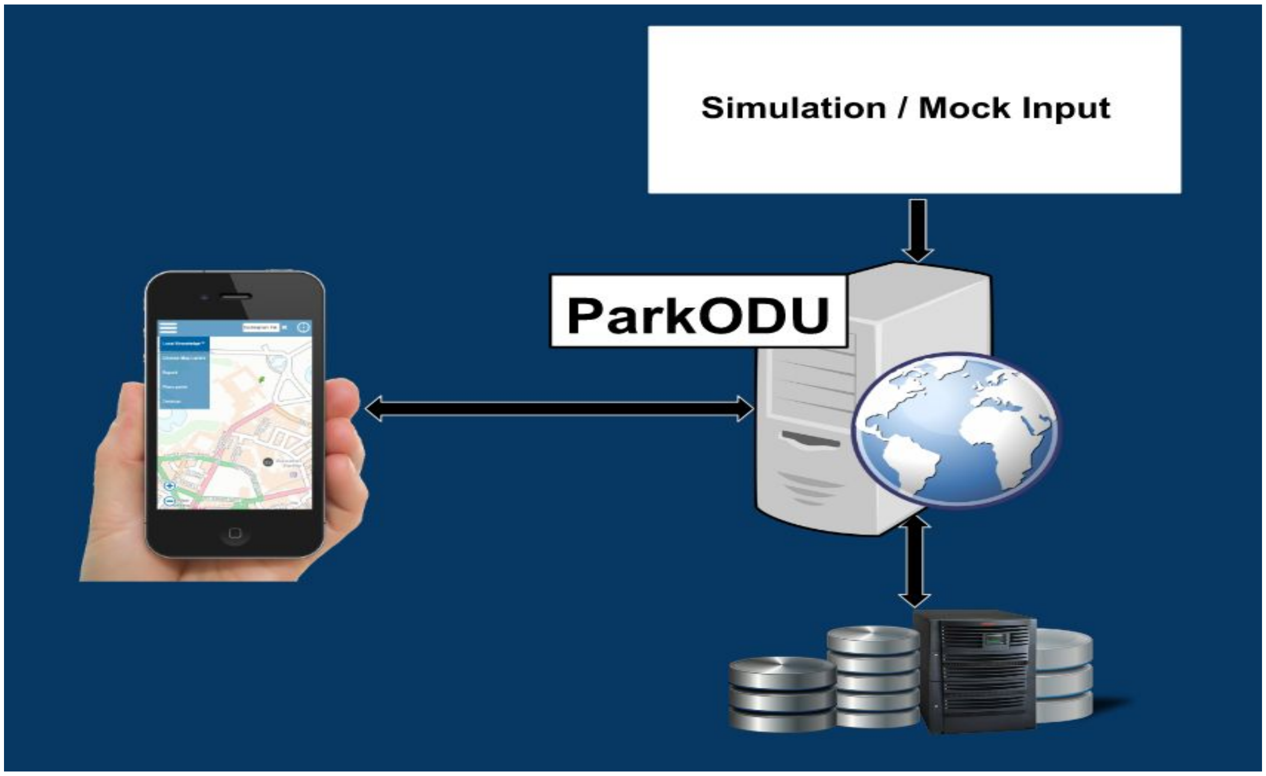
4. Product Prototype Description

The prototype of ParkODU will be totally simulation based as the data that would normally be obtained by the vehicle detection device will be simulated. The prototype of ParkODU will then the simulated data to demonstrate all the features and capabilities of the web app and the native Android/iOS App except support for digital signs.

4.1 Prototype Architecture

“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.” (13)

Figure 1. shows the major functional components of the prototype



4.2 Prototype Features and Capabilities

ParkODU prototype will simulate display a real-time vehicle count by floor in every garage. The prototype will provide the details of the floor plan along and option for navigation to the available space. The ODU faculty or students will be able to import his/her schedule, which will allow them to see the closest parking destination accordingly.

The functionality of ParkODU will make it easier for the driver to park at ODU quicker and more effectual by reducing the amount of time spent manually searching for parking.

Table 1. shows all the features that our real-world product will have versus our prototype.

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	*	

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.

4.2.1 Risks

When it comes to a web application, there are many risks that gets involved during its design phase and even after its production. Risks that can be technical, customer or user risks. Following are the possible technical risks associated with ParkODU:

- ParkODU application server can fails to connect to the web.
- The application or the database can fail, due to network settings being offline or unavailable.
- ParkODU might face software bugs before or after the production.

- Hardware such as IR sensor could fail.
- Due to lack of information from ODU ParkODU could fail to notify drivers of an event taking place or will take place at ODU.
- Lack of Technical knowledge about ParkODU and the developing tools ODU parking and transportation services can face some problems managing the application.
- Input Format provided to ParkODU could be incompatible for it to process.
- ParkODU might crash due to Inability to scale under load

Followed by the possible customers risks associated with ParkODU:

- ODU University ends up implementing or buying a better solution instead.
- ODU University refuses and does not allow access to their network in order for the ParkODU to run.
- ODU parking and transportation services could fail to maintain the servers or hardware
- ODU University replaces all Garages and lots with other buildings that will leave ParkODU for no use.
- The customer could be unwilling to purchase any hardware.
- ODU decides to replace all parking garages with parking lots which will not be compatible with ParkODU.
- The Customer could decide to purchase the vehicle detection device that will be partially Compatible with ParkODU.

Lastly, the possible user risks associated with Park ODU:

- The user could get distracted using the application and result in a crash.
- The user might not have internet connection or a device with internet.
- The user fails to find a parking space as the parking that are being monitored by the application could be full.

4.2.1.1 Technical Risks

Table 2 displays the description of different possible technical risks of ParkODU and how it can be mitigated.

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

Risk	**Description**	**Mitigation**	**Impact**	**Probability**
(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
(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

Risk	**Description**	**Mitigation**	**Impact**	**Probability**
(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.	Mediu m	Medium
(T7) Incompatible Input Format	Formatting of input is not compatible with ParkODU.	● Verify input formatting compatibility through testing.	Mediu m	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

4.2.1.2 Customer Risks

Table 3 displays the description of different possible customer risks of ParkODU and how it can be mitigated.

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

Risk	**Description**	**Mitigation**	**Impact**	**Probability**
(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
(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.	Mediu m	Medium

4.2.1.3 User Risks

Table 4 displays the description of different possible user risks of ParkODU and how it can be mitigated.

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

Risk	Description	Mitigation	Impact	Probability
(U3) User Cannot Find a Parking Spot	All parking that is being monitored by the application is full.	• Inform the user parking is full and application will notify ODU Parking Services. • Provide ODU Parking Services contact information.	Very High	Very Low

4.3 Prototype Development Challenges

The challenges while completing the objectives of the prototype will come from obtaining knowledge for various development tools, services and technology. The development of the ParkODU prototype will require substantial knowledge in MongoDB, Hazelcast, Java programming, RESTful web services, web programming (HTML5 and JavaScript), and native App development for Android and iOS. It is also based on each individual team member's prior knowledge to these tools which can either make it easier or more difficult for that individual in the developing process of ParkODU. It is expected that every team member will devote a significant amount of time in personal research and mentoring to understand all of the tools and services required to complete the objectives of the ParkODU prototype. 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.

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 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

6. 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.

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)

Rogers, Emily. 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/.

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

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.

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/.

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 (12)

Team Gold. “ParkODU. Design Presentation” 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)