Lab 1 – ParkODU

Imani Mason

Old Dominion University

CS411

Professor Thomas J. Kennedy

29 January 2018

Version 2

Table of Contents

Figures and Tables

1 Introduction

Old Dominion University currently faces a problem that many campuses encounter nationwide – inadequate parking. This is due to a high student population, residents and commuters, and insufficient parking. Unfortunately, the challenge of inadequate parking is a source of stress for students, staff, and visitors. According to the ODU website, in 2016 enrollment at Old Dominion was 24,828 students. Approximately 76% of students live off campus, including those taking online classes who take classes online and the majority of this population commutes to ODU (The Problem at Hand, n.d.). ODU currently has five parking garages specified for faculty, metered, commuters, and other, totaling 3,013 spaces (Parking and Traffic Procedures). However, roughly 9,400 student commuters need to park at ODU daily and 1,511 faculty members (835 Full Time, 676 Part Time) (Facts and Figures). A current ODU student says, "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" (Odysseyonline).

Drivers will continue to experience difficulty finding parking spaces, during the hours of 10:00AM - 2:00PM, due to: lack of signage and notifications for available spaces, preferences for specific parking locations, and limited choices during peak hours. ParkODU is a web application designed for any drivers that need to park at ODU. The interface will allow users to view parking information related to Old Dominion University so users may find parking with ease. The goal is to automate and decrease time spent manually searching for the optimal parking spot. The software solution analyzes parking availability in real-time and helps drivers find the vacant parking space closest to their destination. The application will optimize parking as it includes starting location, permit type, and destination to allow the user to find the best parking

space available. Ultimately, this application will save time, resources, and effort. The prototype will use a simulated garage and will demonstrate the product's features.

2 Product Description

ParkODU is a set of tools to help drivers find available parking spaces around the campus and gather parking space usage information for ODU Transportation and Parking. The application will be available in major web browsers and will also be available as an application on Android and iOS. ParkODU 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 ParkODU is to inform drivers of vehicle counts in real-time so drivers can avoid parking facilities that are full and go directly to another facility that has available parking spaces. The secondary objective of ParkODU is to provide ODU Transportation and Parking with usage data so they know how their parking facilities are utilized and use the information for future strategic planning. ParkODU is a solution that will increase the efficiency of parking.

2.1 Key Product Features and Capabilities

ParkODU has many unique features which include vehicle count and displays, navigation, management, and other capabilities to allow the user to find the best parking space while using ParkODU.

2.1.1 Vehicle Counts and Displays

ParkODU will track and monitor garage parking availability in real-time. Vehicle counts will be displayed by garage, floor, and space. A detailed floor plan of each garage floor and the available spaces will be displayed. In addition, the average vehicle counts at each

location by time of day. The user can choose to filter parking based on garages, floors, and space by their permit type. ParkODU will allow the user to sort garages by walking travel time to destination.

2.1.2 Navigation

ParkODU will utilize Google Maps for key navigation features and will allow for future prediction of vehicle count. Faculty and students will be allowed to import their schedules to the application. The Google Map API will sort the list of garages by walking distance (in minutes) to the destination building from the shortest to longest. ParkODU will recommend parking spaces closest to their destination, the first garage in the sorted list would be the best garage and each subsequent garage would be the next best parking option.

2.1.3 Management

Administrative personnel will have the option to add, edit, or remove a garage, floor, or space and user roles. The functionality of this section allows for an administrative user to manipulate allocation of parking spaces if conditions change based on parking needs. This feature of the user interface allows for flexibility using ParkODU to adapt to any parking allocations per implementation. Only administrative users will be able to access this portion of the user interface and will be protected by username and password created at the initial installation of ParkODU.

All administrative users will have the availability to add users as needed. Account management allows for an administrative user to add notifications for any future events that affect parking. Such events will display as a notification for user to notify them of future parking conditions and any preparations that need to be taken in accordance with the event. The

availability to add, remove, and edit access roles that may need to access the administrative user interface can also be configured. The customer will determine which vehicle detection hardware they are most comfortable managing.

2.1.4 Other Features

Additional features include utilizing open source software and mobile support. This helps solve the parking problem by directing drivers, in real time, to the nearest available parking space. Drivers can avoid full garages and minimize their time searching for a parking spot by driving directly to it. As a result, available parking infrastructure is utilized more efficiently.

2.2 Major Components (Hardware/Software)

The flow of information starts from the vehicle detection technology of customer's choice – IR sensors, inductive loops, IP cameras, or other means of detection. Some devices such as IR sensors installed at each parking space, are capable of counting by space, which means the system can detect vehicles in individual parking spaces. Some devices such as inductive loops are only capable of counting by floor as the installation of inductive loops for each space is infeasible. The vendors of the vehicle detection technology gather data from their devices and store them on their server. ODU would normally access the vendor's web portal to access the occupancy data. However, 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. Figure 1 - Major Functional Components Diagram (MFCD) displays how the data will be stored 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. In case

the customer fails to provide machines, ODU Computer Science department virtual machines will be used as the servers. ParkODU database and application servers will service the web app, Android/iOS app, and the digital signs.

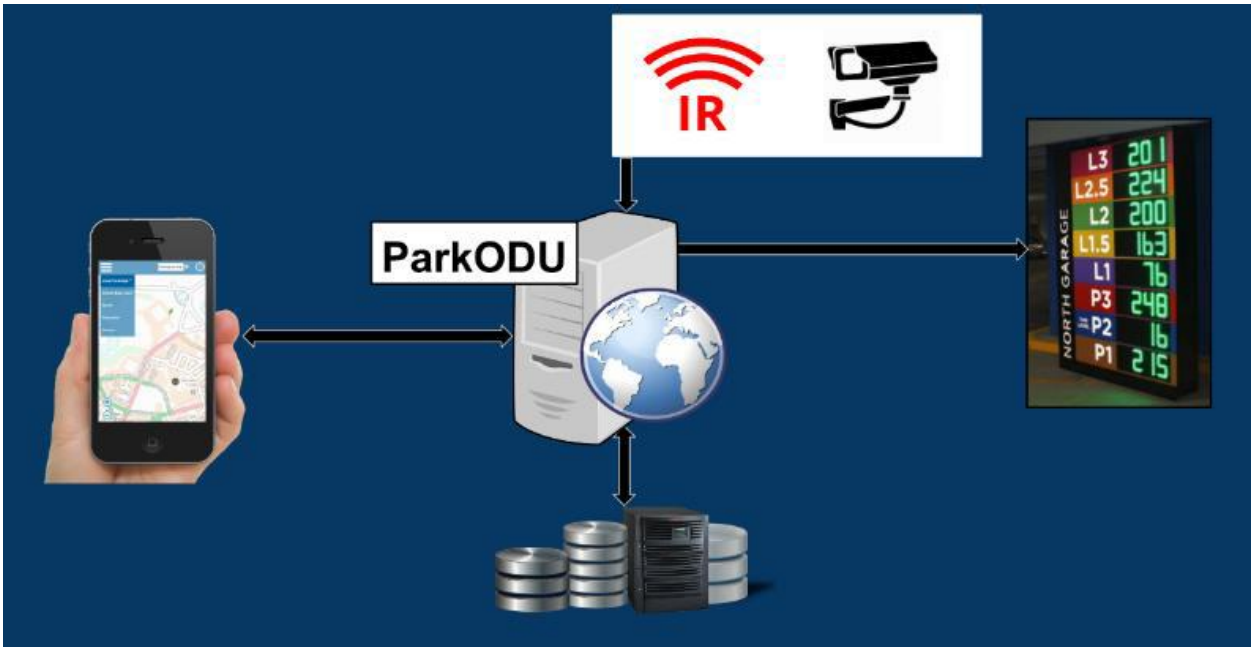

Figure 1 – Major Functional Components Diagram(MFCD)

2.2.1 MFCD Summary

ParkODU is hosted on a server or cluster of servers. These servers can be physical or virtual. ParkODU uses the installed vehicle detection technology to update garage counts and floor plans. ParkODU stores data in MongoDB, an open source database. This information can then be displayed on signage or accessed via the web on a smartphone or computer.

2.2.2 Out of the Box Requirements

ParkODU can be accessed via the web on a smartphone or computer. The customer will need to implement a vehicle detection technology to interface with ParkODU, such as IR sensors or cameras. Optionally, the customer can purchase signage and display information from

ParkODU on the signs for further updated parking information on each garage. Additionally, the customer will need to host the application on a physical or virtual server. The ParkODU software is an open source and available for download on via team gold's website.

2.2.3 Development Tools

ParkODU will be developed using Java programming language. The software development team will implement Spring Framework for infrastructural support at the application level. "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 an open source" (Spring, n.d.). IDE IntelliJ Community Edition will be used to assist with debugging the web application.

ParkODU will use build tools Jenkins and 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. Hazelcast and MongoDB will be ParkODU primary data stores. 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.

Git is used to regulate version control of the application. 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 Google Maps API will be used for key navigation features and future prediction of vehicle count. Google Maps API is a set of application programming interfaces developed by Google that allow Third-party apps to take advantage of or extend the functionality of existing services, for example embedded Google map on website.

2.2.4 Software Description

ParkODU is a software based solution for selecting a parking spot at ODU. The software provides the tools necessary to collect, store, and communicate parking information. By providing an interface for updating vehicle counts, any vehicle detection technology is useable. ParkODU also includes the necessary algorithms, functions, and data structures to provide the functionalities listed in section 2.1. The application will compute vacancies in each parking garage in real-time and analyze past parking data for future decisions. ParkODU will assist user in finding parking nearest to the user's building on campus and send notifications of available spaces to the user. The software based solution will allow users to be informed of all campus events that will impact parking, provide navigation to garage, and suggest parking spaces according to their schedule.

3 Identification of Case Study

ParkODU is being developed for the department of Transportation and Parking Services at Old Dominion University. Due to factors such as geographical constraints and academic buildings having priority over parking lots, many believe the solution to inadequate parking on campus is to build more garages. Although that is an option, the national average to build a parking garage is $8.56 million (Parking, n.d.). Further, the current geographic size of Old Dominion University poses a constraint. ParkODU offers a more effective and less costly solution to parking. Signage outside of every parking lot will make it easier for any driver, especially those visiting will know about vacant parking spaces. ParkODU could also be adopted by universities and businesses who are in need of a more efficient way to handle their parking lots.

4 Product Prototype Description

The prototype of ParkODU will have all of the defined features and capabilities of the web app and the native Android/iOS App except support for digital signs. The data that would normally be obtained from vehicle detection systems such as inductive loops, IR sensors, and IP cameras will be simulated. The ParkODU prototype will compile the simulated data to demonstrate features and capabilities of the web app and the native Android/iOS App.

4.1 Prototype Architecture (Hardware/Software)

The hardware for vehicle detection will be simulated by a REST client application. Figure 2 – Protoype Minor Functional Components Diagram displays how 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. The ParkODU web app and native Android/iOS App will perform operations on the simulated data to demonstrate the features and capabilities. ParkODU will be developed and tested using Hazelcast, MongoDB, Java Spring Framework, Gradle, and Git; for more details see Development Tools in section 2.2.

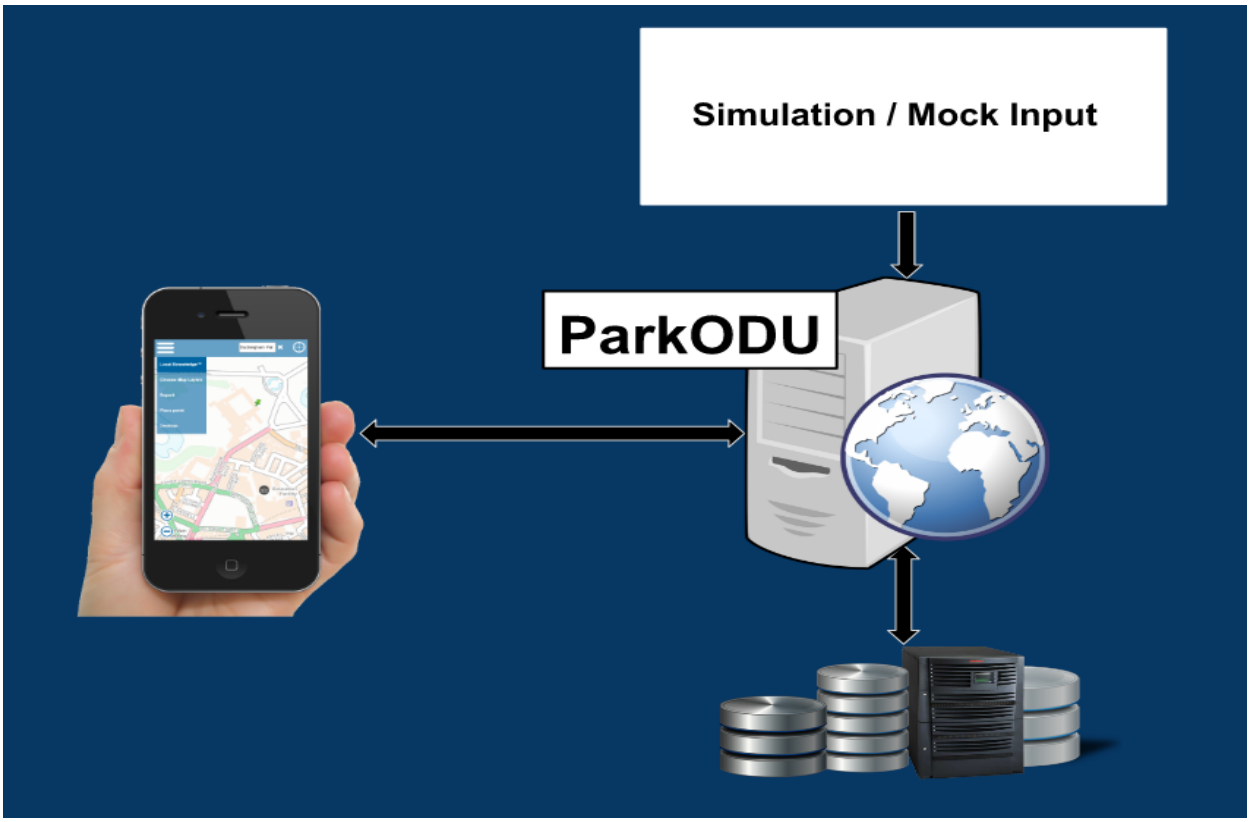

Figure 2 – Prototype Major Functional Components Diagram

4.2 Prototype Features and Capabilities

The application 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 user's previous parking data for improved recommendations. 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. Success has been demonstrated because the overall goal of ParkODU is to reduce the amount of time spent manually searching for parking and the user of the application will be more informed and satisfies with this new process.

The ParkODU prototype will support nearly all features and capabilities of the real working system. All ParkODU features are designed to provide transparency to ODU parking availability that will help drivers avoid full parking garages and go directly to parking garages that have available parking spaces. 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.

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 Prototypes

4.2.1 Risks

A major step while developing a web application is identifying risk, vulnerability identification, and determining appropriate mitigations. The ParkODU team has efficaciously developed technical, customer, and user risks, along with ways to mitigate the risks.

4.2.1.1 Technical Risks

Table 2 – Technical Risks provides a detailed list of technical risks and mitigations associated with ParkODU.

This space was 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
(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

4.2.1.2 Customer Risks

Table 3 – Customer Risks provides a detailed list of customer risks and mitigations associated with ParkODU.

This space was intentionally left blank.

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
(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

4.2.1.3 User Risks

Table 4 – User Risks provides a detailed list of user risks and mitigations associated with ParkODU.

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
(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

Table 4 – User Risks

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 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.

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

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.)

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/.

"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. (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)