Lab 1 – ParkODU Description

Cody R. Coughenour

CS 411

Professor Thomas J. Kennedy

29 January 2018

Version 1

Table of Contents

List of Figures

List of Tables

1. Introduction

Old Dominion University has a parking problem. There are approximately 7,500 parking spaces that service about 10,438 vehicles between commuters, faculty, and on-campus students with vehicles. While these numbers sound adequate, there are numerous complaints from students and faculty such as this one: “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 discrepancy stems from a problem of optimization. The most desirable spaces are those in the parking garages, which hold a more central position on campus. These spots save a driver around 8 minutes of walking time. ODU currently offers 5 parking garages specified by space for faculty, metered, commuters, and other, totaling about 3,000 spaces (4). The above student’s sentiment becomes much more defensible when there are only about 1,000 spaces designated for commuters in the garages.

Drivers do not have information about the current occupancy of these garages. They must guess which spaces are available. Enough incorrect guesses can result in significant stress on the driver and tardiness to class. The likelihood of such an event is increased by certain spaces being more desirable, as mentioned above.

Building additional parking garages would be misunderstanding the problem. There are enough spaces, but not enough information about the availability of those spaces. Additionally, the national average to build a parking garage is $8.56 million (6), the current geographic size of Old Dominion University poses a constraint, and new building priority goes to academic buildings over parking structures.

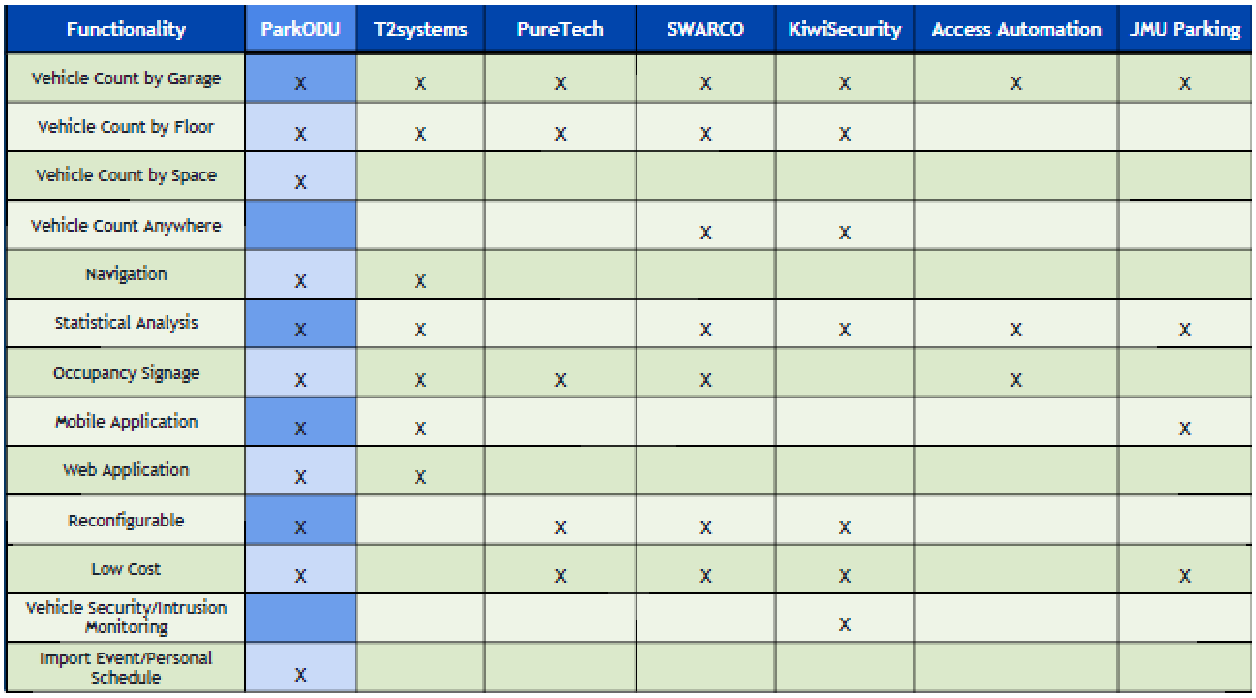
The current state of ODU parking demands a more efficient method to utilize existing parking without building additional parking garages. Without improvement, drivers will continue to have trouble finding parking spaces, during the hours of 10:00AM - 2:00PM, due to: lack of signage and notifications for available spaces, preferences for specific parking locations, and limited choices during peak hours.

ParkODU is a web application designed to address the issues of information availability and efficient parking space utilization. The interface will allow users to view garage occupancy from the web, down to the precise space. ParkODU can also recommend a parking location based on distance from the intended destination and current and historical data trends. The prototype will use a simulated garage and will demonstrate nearly all the product’s features.

[This space intentionally left blank]

2. Product Description

ParkODU gathers and displays parking space availability for the user to view on the web. This information is organized by garage, floor, and space. The user can enter their preferences for parking predictions and recommendations. ParkODU also comes with tools for the organization and management of parking structures. Table 1 shows features provided by ParkODU and other parking solutions.



Functionality	ParkODU	T2systems	PureTech	SWARCO	KiwiSecurity	Access Automation	JMU Parking
Vehicle Count by Garage	X	X	X	X	X	X	X
Vehicle Count by Floor	X	X	X	X	X		
Vehicle Count by Space	X						
Vehicle Count Anywhere				X	X		
Navigation	X	X					
Statistical Analysis	X	X		X	X	X	X
Occupancy Signage	X	X	X	X		X	
Mobile Application	X	X					X
Web Application	X	X					
Reconfigurable	X		X	X	X		
Low Cost	X		X	X	X		X
Vehicle Security/Intrusion Monitoring					X		
Import Event/Personal Schedule	X						

Table 1 – Features of ParkODU and other parking solutions.

2.1. Key Product Features and Capabilities

Table 1 shows that ParkODU primarily differentiates from the competition by providing a vehicle count by space and allowing users to import events, personal schedules, and preferences. However, ParkODU improves upon many of the overlapping categories it has with its competition.

2.1.1. Vehicle Counts and Displays

Vehicle counts are updated in real time. ParkODU then displays this information to the user. If they would like, they can view a detailed floor plan where colors are used to show exactly which spaces are available. Occupancy information is stored to a database, allowing users to view average vehicle counts at different locations and times. ParkODU can also filter space availability by permit type(s) and sort garages by their walking distance to another building. The entirety of the user interface is detailed in Figure 1.

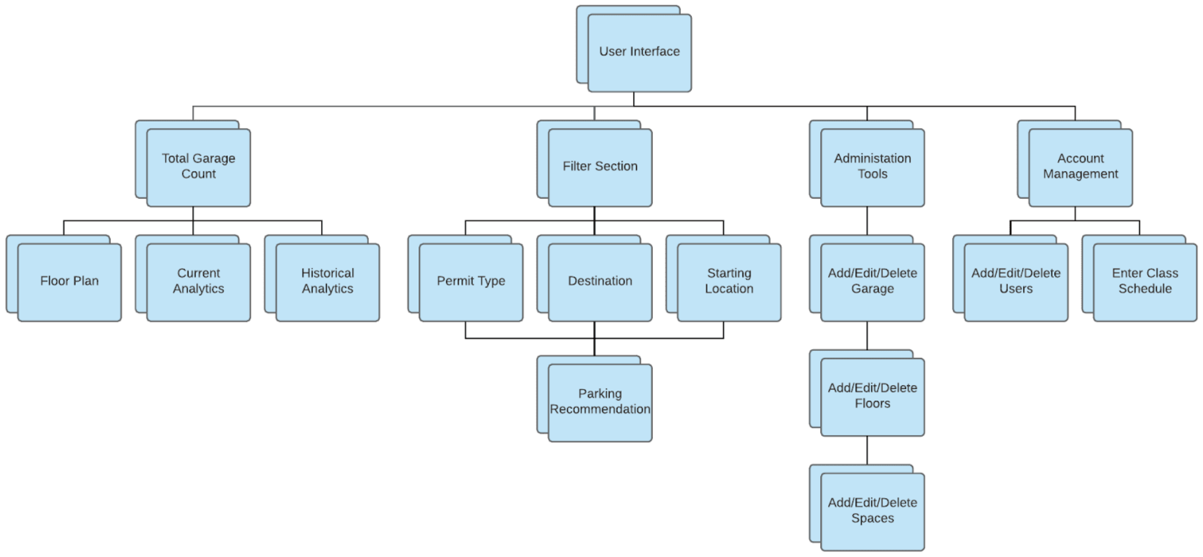


Figure 1 – User Interface Diagram

2.1.2. Navigation

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

2.1.3. Management

ParkODU has tools to make parking administrative tasks easier. Users with access to these tools will be able to add, edit, and delete garages, floors, and spaces. This allows management to reflect real-world changes such as expansions to the garage or a change of space-permit configuration. The administrative user can also add, edit, and delete user roles to give employees access to as many tools as is necessary. Management will also be able to send notifications of events that affect parking and the system will display space availability appropriately.

ParkODU is designed to easily interface with a vehicle detection system of management's choice. This gives the customer full control in deciding what hardware they feel most comfortable maintaining and affording. ParkODU will also support digital signage.

2.1.4. Other

ParkODU is built entirely with opensource software. This keeps the cost of development low and reduces maintenance barriers for future developers. A mobile application (Android/iOS) is planned, but not for the prototype.

2.2. Major Components (Hardware/Software)

The flow of information starts from the vehicle detection technology of the customer's choice; it could be IR sensors, inductive loops, IP cameras, or any other means of detection. Some devices, such as IR sensors installed at each parking space, can count by space. Some devices such as inductive loops are only capable of counting by floor. The vendors of the vehicle detection technology gather data from their devices and store them on their server. Vendors provide APIs, which allows ODU's software developers to create their own applications or interfaces. The data obtained from the vendor's APIs will be sent to ParkODU via requests to REST endpoint. 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 CS 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 2 shows the major components.

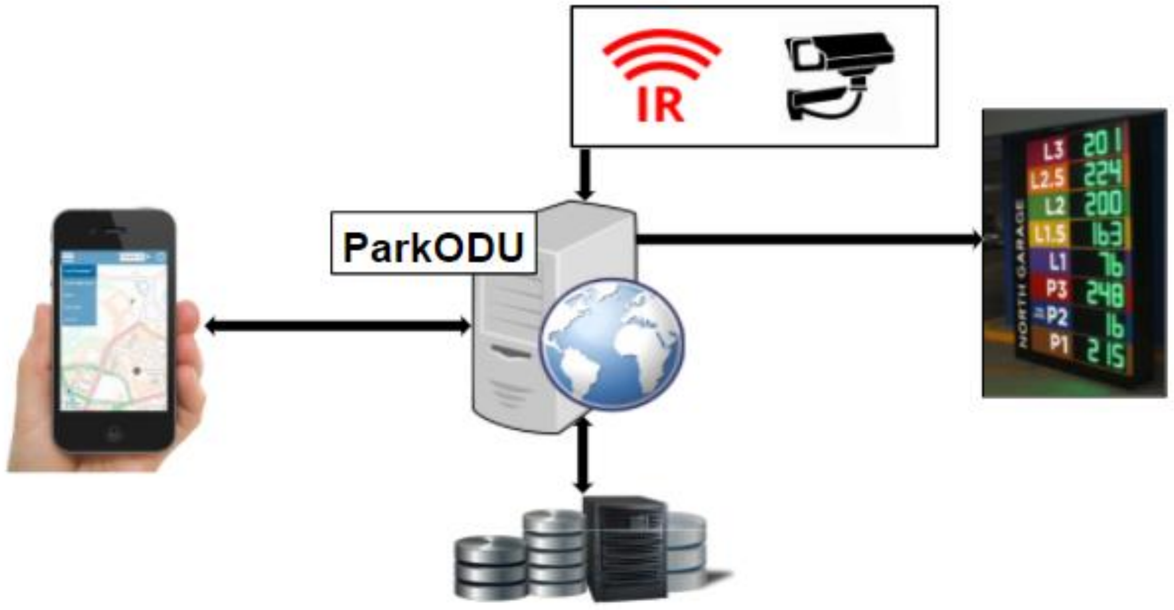


Figure 2 – Major Functional Components Diagram

2.2.1. Out of the Box Requirements

ParkODU is a software solution and will require the user to have a device to access either the web or mobile application, such as a computer or smartphone. The customer is required to purchase a vehicle detection technology to interface with ParkODU, such as IR sensors or cameras. If the customer wishes to display occupancy information on signage, they will need to purchase these signs. The customer will need to host the application on a server, either physical or virtual. The customer will also need to download the software for free from the Team Gold website.

2.2.2. Development Tools

ParkODU is written in Java, utilizing the Spring Framework. The team is using IntelliJ Community Edition as the IDE. The build tools are Jenkins and Gradle. Hazelcast and MongoDB are used for data storage. Version control is handled by Git. Google Maps API is also utilized. A few brief descriptions follow.

2.2.2.1. Java Spring Framework

"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" (12).

2.2.2.2. Hazelcast

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.

[This space intentionally left blank]

2.2.2.3. MongoDB

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

3. Identification of Case Study

ParkODU is being developed for the department of Transportation and Parking Services at Old Dominion University. ParkODU will enable greater utilization of available parking spaces. Parking Services has expressed interest in sharing parking space availability to the public. Many of ParkODU's features are designed for a campus setting, making implementation at other universities a possibility. It would also be possible for ParkODU to be adapted as a general-purpose application for other parking garages.

4. Product Prototype Description

The prototype of ParkODU will have all the defined features mentioned in Section 2, except for a mobile application and digital signage. 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 all the features and capabilities.

[This space intentionally left blank]

4.1. Prototype Architecture (Hardware/Software)

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 simulate special events on weekends. The ParkODU web application will perform operations on the simulated data to demonstrate the features and capabilities. Figure 3 shows the major components of the prototype.

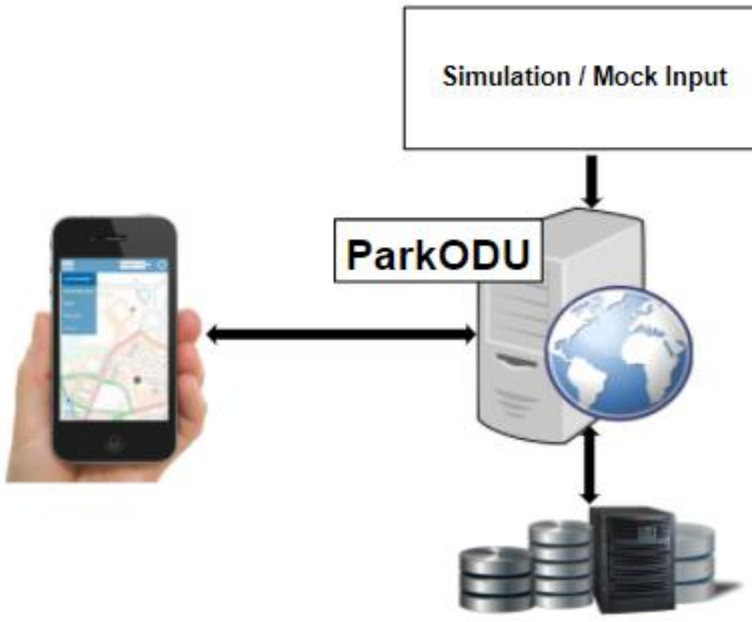


Figure 3 – Prototype Major Functional Components

[This space intentionally left blank]

4.2. Prototype Features and Capabilities

The prototype of our application 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 their schedule and the application will generate the nearest parking options. Table 2 shows the features of the real-world product and the prototype side-by-side.

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 2 – Features of Real-world Product and Prototype

The ParkODU prototype will support nearly all features and capabilities of the real working system. All the features of ParkODU are designed to provide transparency to ODU parking availability; this 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.

4.2.1. Risks

ParkODU has many technical, customer, and user risks that have been well documented. Each risk comes with a name, description, mitigation, impact, and probability. These are summarized in Tables 3,4, and 5.

4.2.1.1. Technical Risks

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 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.	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 3 - Technical Risks

4.2.1.2. Customer 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
(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 4 - Customer Risks

4.2.1.3. User Risks

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 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 can 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 5 - User Risks

[This space intentionally left blank]

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 technologies. The development of the ParkODU prototype will require substantial knowledge in MongoDB, Hazelcast, Java programming, RESTful web services, and web programming (HTML5 and Javascript). It is expected that every team member will devote a significant amount of time in personal research and mentoring to understand 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-count. (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)

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

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 18h, 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.)

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

Spring Framework. (October 1, 2005). Retrieved January 14th, 2018, from https://en.wikipedia.org/wiki/Spring_Framework. (12)

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)