ParkODU Product Description

Professor Thomas Kennedy
3/19/2018
Isaac Asante
Lab 1 version 3

Table of Contents

List of Figures

List of Tables

1. Introduction

Old Dominion University has an enrollment of 24,828 students as of 2017. (3) Roughly 76% of the students enrolled are based off campus; majority of whom commute to classes every day. There are five garages here at Old Dominion University which totals 3013 parking spaces. Out of the 3013 spaces, the 1511 faculty members as well as the roughly 9400 student commuters need to park every day. As time goes on, there will be more incoming students who own a car but currently, already existing parking lots are being replaced by academic buildings. Geographically, there is constraint as to where to build more garages. In recent years, Old Dominion University has demolished parking structures and replaced them with academic buildings as they hold a higher priority over parking lots. The national average to build a parking garage is about $8.5 million. Even if the university were to build more garages, they would all be far away from the main buildings on campus. The current state of ODU parking demands a solution that can efficiently utilize the parking spaces that are already available.

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

2. Product Description

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

2.1. Key Product Features and Capabilities

Real-time vehicle counts in garages means users will know exactly where a spot will be available. Statistical analysis will be supported to predict where a spot may be available during certain times based on the user's imported schedule. On the management side, the customer will be able to edit the availability of all garages to make way for special events.

2.1.1 Vehicle Counts and Displays

ParkODU's capabilities with vehicle counts and displays are its main features. The application:

- Tracks and monitors garage parking availability in real-time
- Counts vehicles by garage, floor, and space
- Displays a detailed floor plan of each garage floor and the available spaces
- Displays average vehicle counts at each location by time of day

- Allows users to filter garages, floors, and space by their parking permit type
- Allows users to sort garages by walking travel time to another building

2.1.2 Navigation

ParkODU is equipped with navigation for a personalized user experience. The application:

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

2.1.3 Management

ParkODU's management team will have the following capabilities:

- 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
- Send notifications of the events going on at ODU that affect parking
- Interfaces with any vehicle detection system

2.1.4 Other

Other notable ParkODU features:

- Utilizes open source software
- Mobile support planned

[Space intentionally left blank]

2.2. Major Components (Hardware/Software)

ParkODU relies on data from the vehicle detection technology in the garages. Figure 1 illustrates the major functional components in ParkODU. The customer will choose whichever detection technology that works best for them. Some detection devices like IR sensors will be able to detect vacancy in each individual parking, while some detectors like inductive loops are only capable of counting each vehicle on a garage floor. Vendors of these detection devices gather and store the data obtained from the detection devices onto their server. These vendors also provide APIs which allows software developers to create their own app or interface. Data from the APIs will be sent to ParkODU through requests to REST endpoint and stored in Hazelcast, an in-memory data grid. MongoDB, a document database, will be used as a secondary to Hazelcast to backup data. The database for ParkODU's web application will be stored on the machines the customer provides. Be it virtual or physical.

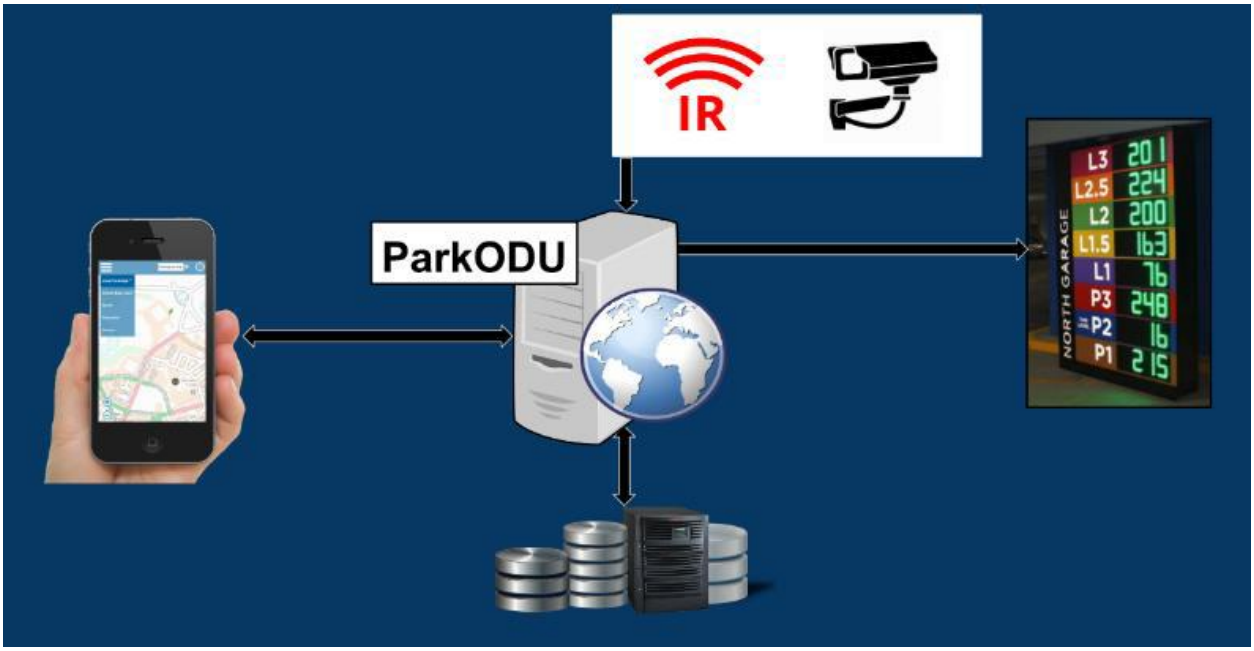

Figure 1 - Major Functional Components

2.2.1 MFCD Summary

The installed vehicle detection technology will send the space count data to servers hosted by ParkODU. ParkODU will store the data on Hazelcast and back them up in MongoDB. ParkODU will then update the availability of spaces on the web app and supported mobile devices, and on the signage outside the garages.

2.2.2 Out of the Box Requirements

There are a few ParkODU requires for it be in use. These needs are to be met by the customer:

- ParkODU is accessed via the web on a smartphone or computer.
- The customer will need some sort of 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.
- The customer will need to host the application on a server (physical or virtual).
- The software is open source and available for download on our website.

2.2.3 Development Tools

Development tools used to build ParkODU are:

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

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 constraint and academic buildings having priority over parking lots, ParkODU offers a more effective solution to parking. Signage outside of every parking lot will make it easier for any driver, especially those visiting, to know

about vacancies. The ParkODU software is also more feature packed and less costly than any other parking software out there. 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 ParkODU prototype will have all the features and capabilities defined in the product description except for the digital signs outside of the garages and lots. The software will run on the web and on android/iOS devices which will connect with ParkODU's servers and then its database. Vehicle detection will also be simulated in place of detection systems. The data gathered will be fed into the software to demonstrate ParkODU's features and capabilities. Table 2 illustrates features of the real-world-product and the Prototype.

Feature	RWP	Prototype
Real-time vehicle counts on every level of each garage	*	*
Display floor plan to show counts by space on each floor	*	*
Display average vehicle count at each location by time of the day	*	*
Allow users to sort garages by walking travel time	*	*
Allow users to filter garages, floors, and space by their parking permit type and space types	*	*
Allow ODU parking staff to configure parking garages, floors, and spaces.	*	*
Provide directions to each garage from user's current location	*	*
Predict future vehicle counts based on the current and historical traffic pattern	*	*
Upload special event schedules and allow the apps to display notification to end users	*	*
Send data to digital signs at the entrance of every garage	*	

Table 1- Features of Real-World Product and Prototype

4.1. Prototype Architecture (Hardware/Software)

Vehicle detection will be simulated by a REST client application. ParkODU's REST endpoint will receive requests from the vehicle detection simulation's REST client application and update the vehicle counts accordingly. Data from the simulation will be similar to how ODU's parking traffic flows in the real world during a normal weekday and also during special events. The web and mobile applications will process data from the simulation and deliver available parking spaces to the user. Figure 2 illustrates Prototype Major Functional Components.

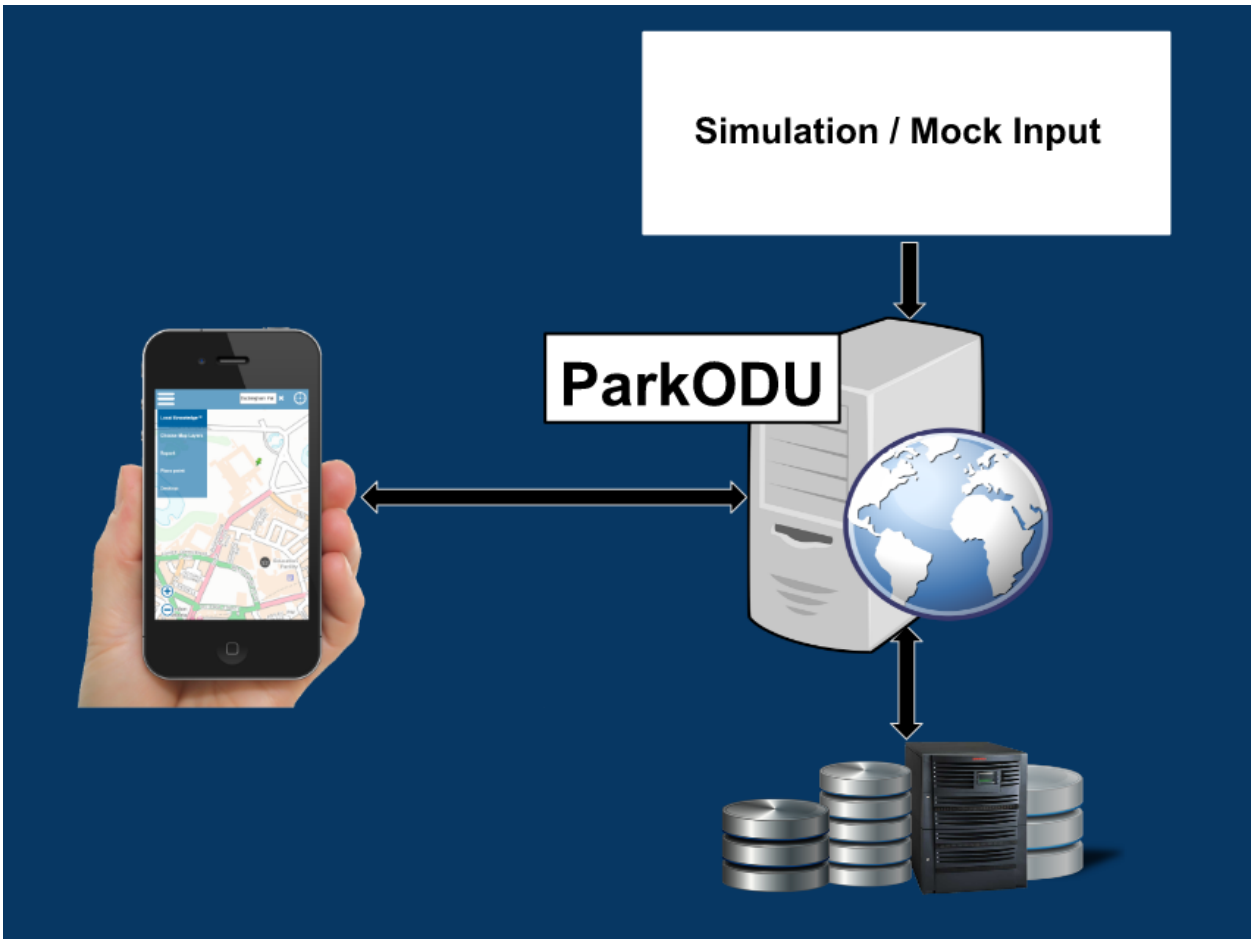

Figure 2 - Prototype Major Functional Components

[Space intentionally left blank]

Hazelcast - 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 - 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.

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)

Gradle - A build automation system that draws from already established ideas from Ant and Maven and improves upon them. Gradle follows a build-by-convention approach while using domain specific language. Gradle makes it easy for developers to add onto a build script in an expressive and maintainable way.

Git - 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 repository.

4.2. Prototype Features and Capabilities

The ParkODU prototype will simulate an input and display a real-time vehicle count by floor in every garage. The prototype will provide the detailed floor plan along with navigation to the vacant space. The user will be able to import his/her schedule and the application will generate the nearest parking options. It will also analyze the users' previous parking data in for improved recommendations.

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 allowing this to become possible.

[Space intentionally left blank]

4.2.1. Risks

ParkODU's risks have been broken into technical, customer, and user risks. Each type of risk is discussed and mitigated below in Tables 3, 4, and 5.

(T1) Web Connection Failure
(T2) Database/Web Application Failure
(T3) Software Bugs
(T4) Hardware Failure
(T5) Failure to Notify User of an Event
(T6) Lack of Technical Knowledge
(T7) Incompatible Input Format
(T8) Inability to Scale Under Load

(C1) University Implements a Better Solution
(C2) University Does Not Allow Access to Network
(C3) Customer Unable to Maintain Servers/Hardware
(C4) University Replaces All Garages and Lots with Other Buildings
(C5) Customer Unwilling to Purchase Hardware
(C6) Parking Lot Replaced by Parking Garage
(C7) Customer Purchases Partially Compatible Technology

(U1) User is Distracted While Using the Application
(U2) No Internet Device
(U3) User Cannot Find a Parking Spot

[Space intentionally left blank]

4.2.1.1. Technical Risks

Table 2 presents the technical risks associated with ParkODU and their mitigations.

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	Very High	Very Low

Risk	**Description**	**Mitigation**	**Impact**	**Probability**
		• Regression testing and continuous integration		
(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 Scheduled Task 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

Risk	Description	Mitigation	Impact	Probability
(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

[Space intentionally left blank]

4.2.1.2. Customer Risks

Table 3 presents the customer risks associated with ParkODU and their mitigations.

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.	Medium	Medium

Table 3 - Customer Risks

4.2.1.3 User Risks

Table 4 presents the user risks associated with ParkODU and their mitigations.

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

Table 4 - User Risks

4.3. Prototype Development Challenges

There are a lot of tools and components that go into developing the prototype for ParkODU. Challenges will come from acquiring the knowledge in database software such as Hazelcast and MongoDB. The prototype will also require knowledge in RESTful web services, Java programming, HTML and JavaScript. Development for a native App for android and iOS will also be a challenge. Team members who require more knowledge on these tools and services will be expected to research those areas, and team members who know more than most will have to devote some time in mentoring those who need more understanding. Time will also be a challenging factor as ParkODU is a solution that needs to meet many requirements.

[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 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

[Space intentionally left blank]

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. (F.4.)t

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

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

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

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

How Much Does a Parking Garage Cost? Retrieved November 02, 2017, from http://www.parking.org/2016/01/19/tpp-2013-09-how-much-does-a-structure-cost/. (6)

“IntelliJ IDEA: The Java IDE for Professional Developers by JetBrains.” *IntelliJ IDEA*, Jet Brains , Retrieved January 18th, 2018, fromwww.jetbrains.com/idea/.

Operating Budget and Plan. Old Dominion University. Retrieved November 02, 2017, from https://www.odu.edu/content/dam/odu/offices/budget-office/docs/opplan2017.pdf. (5)

ODU Campus Parking Map. Retrieved October 23, 2017, from https://www.odu.edu/content/dam/odu/offices/parking-and-transportation-services/docs/odu-student-parking-map-mm.pdf. (F.1.)

Parking and Traffic Procedures. Old Dominion University. Retrieved November 02, 2017,

from https://www.odu.edu/content/dam/odu/offices/parking-and-transportation-services/docs/parking-transportation-rules-and-regulations.pdf. (4)

Providence Place mall enhances parking garage with $20M in improvements (2016, December 15). Retrieved October 30, 2017, from https://pbn.com/providence-place-mall-enhances-parking-garage-adds-more-pay-stations-improves-signage119194/. (F.3.)

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

“Spring: the source of modern java by Pivotal” Retrieved January 23rd, 2018 from http://spring.io

Team Gold. “ParkODU.” December 2017. PowerPoint presentation.

The Problem at Hand - The Expansion of Parking At Old Dominion University. (n.d.). Retrieved November 02, 2017, from https://sites.google.com/a/odu.edu/the-expansion-of-parking-at-old-dominion-university/home/the-problem-at-hand. (2)

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

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

Vehicle counting & detection systems. (n.d.). Retrieved October 10, 2017, from https://www.swarco.com/stl/Products-Services/Parking-Solutions/Parking-guidance/Vehicle-counting-detection-systems. (11)

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