Lab 1 - ParkODU Product Description

Gerard S. Silverio

CS 411

Professor Thomas J. Kennedy

29 Jan 2018

Version 2

Table of Contents

List of Figures

List of Tables

1. Introduction

Like many colleges and universities in the United States, Old Dominion University students, faculty, staff, and visitors face a challenging dilemma regarding the inefficiency of finding available parking spaces on campus. In 2017, Old Dominion University boasts a combined total of 24,375 undergraduate and graduate enrollments and an 18:1 student-to-faculty ratio. (3) With approximately 18,769 of students living off-campus, 11,692 students enrolled in online courses, 1,850 of on-campus students with vehicles, and 1511 faculty members (835 Full-time and 676 Part-time), approximately 10,438 commuters, and resident students are forced to share the 7,500 parking spaces available at Old Dominion University. (12, 4, 3, 13, 14)

ParkODU is a next generation parking solution platform designed to reduce the time and effort of identifying available parking spaces while increasing the efficiency of filling vacancies in parking lots, garages, and meters. By leveraging open-source technologies and with the ability to interface with new or existing vehicle detection systems, configuration and deployment is simple and cost-effective. In addition, as the popularity and the technology of distance learning advances, enrollment in online courses at ODUOnline will mitigate the need for additional parking spaces thus further reducing traffic flow within parking lots and garages.

2. ParkODU Product Description

ParkODU is designed to automate the search for an ideal parking space and to reduce the amount of time drivers normally take to identify vacant spaces through manual search. The platform uses continuous monitoring to detect parking space availability in real-time while collecting trend data to be used to generate analytical reports and forecasting. Customized queries based on starting location, permit types, and destinations allow drivers to quickly identify the best possible parking locations closest to their destination.

2.1 Key Product Features and Capabilities

ParkODU boasts a suite of innovative modules that simplify the process of identifying available parking locations through continuous monitoring, visualizations, notifications, and advanced querying. Through the administrative module, administrators can manage user access, parking locations, and events.

2.1.1 Analytics Module

The analytics module allows users to view historical trends and generate tabular and graphical reports. With availability forecasting, users have the capability of predicting future vacancy rates for desired parking locations allowing them to better estimate travel time.

2.1.2 Continuous Monitoring and Dynamic Floor Plan Visualizations

Dynamic floor plan visualizations display the vacancy status of all parking spaces within a floor allow for users to quickly pinpoint the exact location of their target parking space. Real-time vehicle counts through individual parking space monitoring can be interfaced by external signage allowing drivers to quickly view vacancies of parking lots, garages, and floors.

2.1.3 Events Management Module

The events management module executes scheduled tasks several times a day that poll ParkODU event calendars for current and future events. Event notifications are sent to users regarding the statuses of parking locations and on-campus events that may affect parking space availabilities.

(This section intentionally left blank)

2.1.4 Advanced Querying

Through the utilization of the Google Maps API, it is possible for users to generate advanced queries to identify the best possible parking locations based on the user's starting location, applicable permit types, and destination. Results are sortable and display the total availability count, destination, distance, and travel time.

2.1.5 Administrative Module

The administration module allows administrators to easily manage users, parking lots, garages, floors, and spaces. Administrators can assign custom roles and permissions to users in order to restrict access. Statuses of parking lots, garages, floors, and spaces can be updated within the administration module to notify users of planned and unforeseen events such as closures, campus events, construction, maintenance, and hardware failure.

(This section intentionally left blank)

2.2 Major Components (Hardware/Software)

ParkODU is an open-source parking solution platform that provides its customers with the flexibility to implement a vehicle detection system of their choice. The Major Functional Components of ParkODU depicted in Figure 1 display the flow of data throughout the system. Data is first gathered by integrating ParkODU with an existing vehicle detection system with the capability of monitoring parking space availability in real-time. The data collected by the vehicle detection systems can be either polled by or sent to ParkODU which in turn updates the parking space statuses of parking lots, garages, and floors at Old Dominion University. Once the parking space statuses are updated, the data is persisted into Hazelcast's In-Memory Data Grid and replicated into a MongoDB collection. The collected parking data can then be accessed through any browser or through the ParkODU RESTFul APIs.

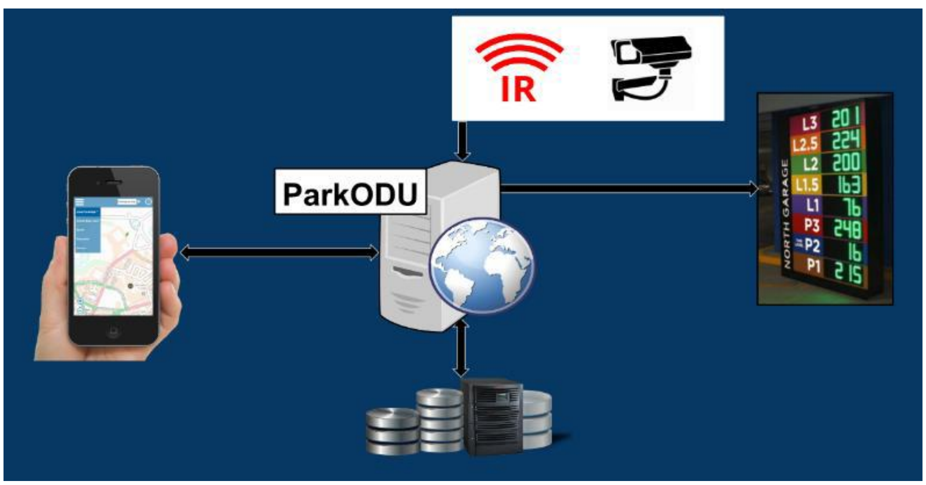

Figure 1 - Major Functional Components Diagram

(This section intentionally left blank)

2.2.1 Vehicle Detection Systems

An existing vehicle detection system is required for ParkODU to be fully operational. When in the market to purchase a vehicle detection system, customers weigh the pros and cons of each sensor technology used by the system. The accuracy of parking data collected and reported by ParkODU depends directly on the type of vehicle detection system installed. For example, the utilization of inductive loop detectors limits the vehicle count to monitor only entrances and exits as opposed to ultrasonic sensors which can be used to monitor individual parking spaces. Additionally, analytical software is included with the hardware implementation and is required for the sensors to communicate with ParkODU. Some examples of sensors used in vehicle detection systems include:

- **Passive Infrared Sensors** - An object detection device that emits infrared light while a nearby proximity sensor actively waits to detect the reflected light cause by an obstructed object. The distance of the object is measured based on the time interval from when the infrared light was emitted to when it is detected by the proximity sensor. (15)
- **Ultrasonic Sensors** - An object detection device that emits sound waves while actively listening for reflected sound waves caused by an obstructed object. The distance of the object is measured based on the time interval from when the sound wave was emitted to when it was returned. (15)
- **Hybrid Passive Infrared and Ultrasonic Sensors** - An object detection device that combines the technology of Passive Infrared Sensors and Ultrasonic Sensors.
- **Infrared Cameras** - Devices that detect infrared energy convert the data collected into thermal imagery. (16) Thermal imagery processing algorithms are used to classify objects detected by the infrared cameras.

- **Inductive Loop Detectors -** An occupancy detector that consists of wired coils installed within or underneath the surface of roadways that induces an electric current if a metal object were to trigger a relay. (17)

2.2.2 ParkODU Web Application and APIs

ParkODU consists of a responsive web application and RESTful APIs that users and external systems can connect to retrieve data ingested from vehicle detection systems. Utilization of the Java Spring Framework provides the capability to be deployed on any operating system. The parking solution comes packaged in a single JAR file along with required properties file. It is recommended that ParkODU is installed within a cluster environment to mitigate down time by implementing a failover strategy.

Communication between ParkODU and the vehicle detection system is possible through the exposure of RESTful APIs. Data transmissions can be configured to either poll the vehicle detection system for changes in sensor status or the vehicle detection system can send sensor status notifications to ParkODU.

2.2.3 Hazelcast In-Memory Data Grid (IMDG)

Hazelcast In-Memory Data Grid acts as both a distributed computing system and a data store. (18) As an industry leader with the fastest performance in distributed cache, data can be ingested into the data store within microseconds. By taking advantage of Hazelcast’s distributed computing platform, ParkODU has the ability to store its data within memory for lightning fast reads and writes and can be easily scaled vertically and horizontally with little to no downtime.

2.2.4 MongoDB

Due to the volatility of storing data in memory with Hazelcast, ParkODU uses MongoDB as a secondary storage to mitigate data loss in the event that all Hazelcast nodes were to be shut down. In addition, if there are no active ParkODU servers currently up, Hazelcast is configured to replicate all existing data from MongoDB onto its distributed maps.

2.2.5 ParkODU Android and iOS Mobile Applications

The ParkODU Android and iOS mobile applications will be developed and available for users with the same functionality as the ParkODU web application. All requests will be executed through the ParkODU API.

2.2.6 Additional Development Tools

- Integrated Development Environment (IDE): IntelliJ Community Edition
- Build Tools: Jenkins and Gradle
- 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. Without an effective parking solution, students, faculty, staff, and visitors are still experiencing high frustration levels when it comes to searching for available parking spaces. As an extremely cost-effective product, ParkODU offers a solution that will provide commuters with a tool to help quickly identify the best available parking space and help increase parking space utilization for parking locations on-campus.

Future instances of ParkODU can be implemented by other organizations with a need for a parking solution whether they are a business, hotel, mall, or university. With a simple installation process, the ParkODU software can be readily integrated with a new or existing vehicle detection system.

4. ParkODU Product Prototype Description

The prototype of ParkODU will contain all the features and capabilities listed in Section 2.1. The capabilities to be eliminated are the development of the ParkODU Android and iOS Mobile Applications. Generation of dynamic floor plans with real-time updates of parking space availability will be modeled. The vehicle detection system integration and external signage will be simulated by an external web application. The full feature list comparing the RWP and the Prototype is shown on Table 1.

Features	RWP	Prototype
Historical Trends	Yes	Yes
Report Generator	Yes	Yes
Availability Forecasting	Yes	Yes
Continuous Monitoring	Yes	Yes
Dynamic Floor Plans	Yes	Modeled
Events Management	Yes	Yes
Advanced Querying	Yes	Yes
Administrative Console	Yes	Yes
Android Mobile Application	Yes	No
iOS Mobile Application	Yes	No
Vehicle Detection System Integration	Yes	Simulated
External Signage	Yes	Simulated

Table 1 - RWP and Prototype Features

(This section intentionally left blank)

4.1 Prototype Architecture (Hardware/Software)

The prototype of ParkODU will demonstrate most of the key features and capabilities listed in Section 2.1 except for the implementation of a fully functional vehicle detection system, native Android and iOS mobile applications, and external signage. As shown on Figure 2, the prototype will consume simulated data from an external service mocking a vehicle detection system. Once consumed, the data will be evaluated by the ParkODU analytics engine and be readily available for consumers to view through the web application and APIs.

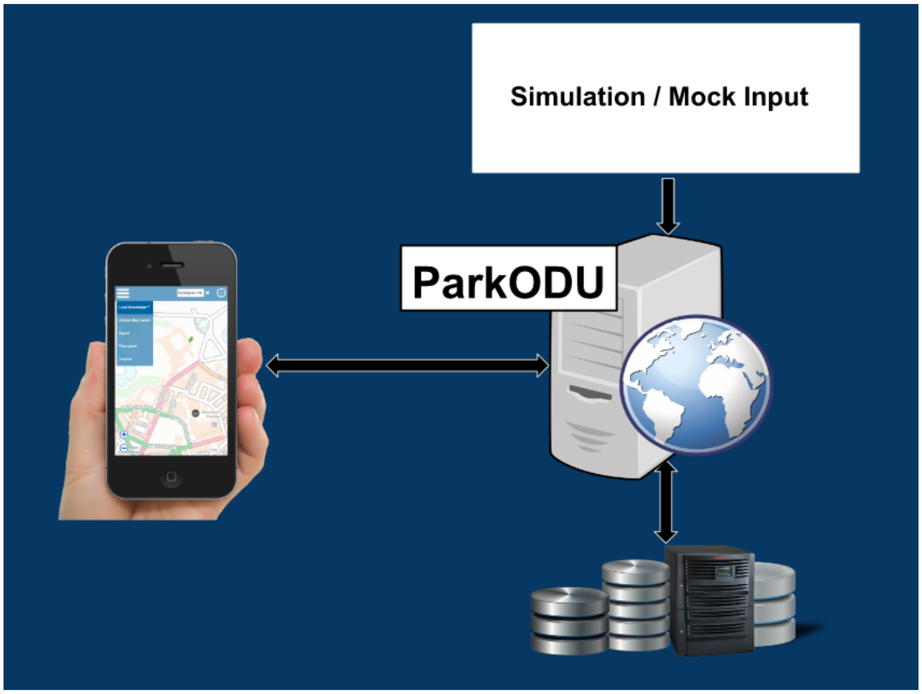

Figure 2 - Prototype Major Functional Components Diagram

4.1.1 Simulated Vehicle Detection Systems

Due to time constraints and the substantial cost of implementing modeled vehicle detection systems, the next best course of action is to develop vehicle detection system simulators. These simulators will act as mock vehicle detection systems installed at each parking location and will comprise of scheduled tasks that mimic the daily traffic flow of commuters finding parking around Old Dominion University. These tasks will consist of RESTful web service requests sent to ParkODU's APIs to update the availability statuses of individual parking spaces.

4.1.2 ParkODU Web Application and APIs

The ParkODU prototype will be made up of of a web application and RESTful APIs that users and external systems can request in order to retrieve data ingested from the simulated vehicle detection system.

4.2 Prototype Features and Capabilities

The ParkODU prototype demonstrates the capability of increasing the efficiency of fulfilling parking space vacancies while reducing the time it takes for drivers to identify ideal parking locations.

4.2.1 Mitigated Risks

The risk mitigations addressed by the prototype include:

- (T2) Database/Web Application Failure - the prototype is deployed in a clustered server environment with high availability thus reducing the likelihood of downtime. ParkODU integrates with Hazelcast's Management Center in order to help monitor server statuses by periodically checking for heartbeats. Alerts are configured to occur at the event of any expected and unexpected shut downs.
- (T3) Software Bugs - the prototype incorporates test-driven development methodologies during the development phase to ensure the mitigation of software bugs. Specific team members develop UI/UX testing standards and actively run regression tests throughout the continuous integration phase.

- (T7) Incompatible Input Format - the usage of rules based on regular expressions prevent incompatible input data to be persisted into the data store.
- (T8) Inability to Scale Under Load - with the use of Hazelcast's distributed computing platform, new members can easily be configured and added to the existing cluster.

4.2.2 Identified Risks

To reduce risk associated with the development and implementation of ParkODU, a mitigation plan is introduced and categorized into three separate categories: Technical Risks, Customer Risks, and User Risks. The impact and probability of these risks are also displayed to measure the level of severity of each individual risk.

4.2.2.1 Technical Risks

The technical risks listed in Table 2 show various steps for mitigation towards risks associated with the hardware and software components of ParkODU.

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
(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

4.2.2.2 Customer Risks

Table 3 displays risks that are associated with customer based decisions on the implementation of a different solution, the incompatibility with new and existing resources, as well as decisions based on geographical expansion.

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

Risk	Description	Mitigation	Impact	Probability
(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

Risk	Description	Mitigation	Impact	Probability
(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

4.2.2.3 User Risks

Risks associated with ParkODU end users that may potentially impact the identification of available parking spaces are shown on Table 4.

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

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

Overcoming the challenges due to the lack of knowledge and experience toward the chosen technical stack of ParkODU will prove to be the biggest obstacle when completing the objectives of the prototype. The development of the prototype will require team members to obtain a significant amount of knowledge in full-stack software development utilizing the Java Spring Framework, building RESTful web services, front-end web development (HTML5, CSS, and Javascript), Hazelcast, and MongoDB. In addition, the lack of time will also play a key role in determining the number of features and capabilities to be implemented into the prototype.

(This section intentionally left blank)

5. Glossary

Administrator - a special user with access to additional tools for user account and space management

Agile - a software development 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 can park without receiving a violation

Predictions - a guess based on current and historical data about garage space availability

Real-time - actual 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 on October 10, 2017 from http://www.access-automation.co.uk/car-park-count-systems.

Agile [Digital image]. (2017, May 8). Retrieved on November 29, 2017 from https://www.codingmart.com/uploads/post/image/57e0c0488ca7853c76dd986e/Agile_Development_Process.pngvehicle-count. (F.4.)

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

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

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

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

Inductive Loops. Retrieved on January 24, 2018, from http://diamondtraffic.com/technicaldescription/124 (17)

IntelliJ IDEA: The Java IDE for Professional Developers by JetBrains. *IntelliJ IDEA*, Jet Brains, Retrieved on January 18th, 2018 from http://www.jetbrains.com/idea/.

Is University Parking a Common Grievance? Parking Today Media. September 2011. Retrieved on September 2017 from http://www.parkingtoday.com/articledetails.php?id=1072. (8)

ODU Campus Parking Map. Retrieved on 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.)

ODUOnline at a Glance. ODU Online. Retrieved on January 24, 2018 from https://online.odu.edu/about/at-a-glance (14)

Old Dominion University | Student Life. U.S. News Best Colleges. Retrieved on January 24, 2018, from https://www.usnews.com/best-colleges/old-dominion-3728/student-life (12)

Old Dominion University | Campus-Info. U.S. New Best Colleges. Retrieved on January 24, 2018, from https://www.usnews.com/best-colleges/old-dominion-3728/campus-info (13)

Operating Budget and Plan. Old Dominion University. Retrieved on November 02, 2017 from https://www.odu.edu/content/dam/odu/offices/budget-office/docs/opplan2017.pdf. (65)

Parking and Traffic Procedures. Old Dominion University. Retrieved on November 02, 2017 from https://www.odu.edu/content/dam/odu/offices/parking-and-transportation-services/docs/parking-transportation-rules-and-regulations.pdf. (64)

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

Robust Vehicle Detection under Various Environments to Realize Road Traffic Flow Surveillance Using an Infrared Thermal Camera. Retrieved on January 24, 2018 from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4339787/ (16)

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

Spring: the source of modern java by Pivotal. Retrieved on January 23, 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)

Ultrasonic Sensors vs Infrared (IR) Sensors. Retrieved January 24, 2018, from https://www.maxbotix.com/articles/ultrasonic-or-infrared-sensors.htm (15)

University Facts & Figures. Old Dominion University. Retrieved on November 2, 2017, from https://www.odu.edu/about/facts-and-figures. (3)

Vehicle Counter. Retrieved on October 10, 2017 from https://www.kiwisecurity.com/.

Vehicle counting & detection systems. Retrieved on 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 23, 2018 from https://www.mongodb.com/what-is-mongodb.

Why Hazelcast IMDG? Retrieved on January 24, 2018, from https://hazelcast.com/why-hazelcast/imdg/ (18)