CS411W – Lab 2

# ParkODU Prototype Product Specification

Imani Mason

Team Gold

Old Dominion University

Professor Thomas J. Kennedy

11 March 2018

Version 2

## Table of Contents

## List of 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. 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). 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 driver that needs 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.

(This space is intentionally left blank.)

## 1.1 Purpose

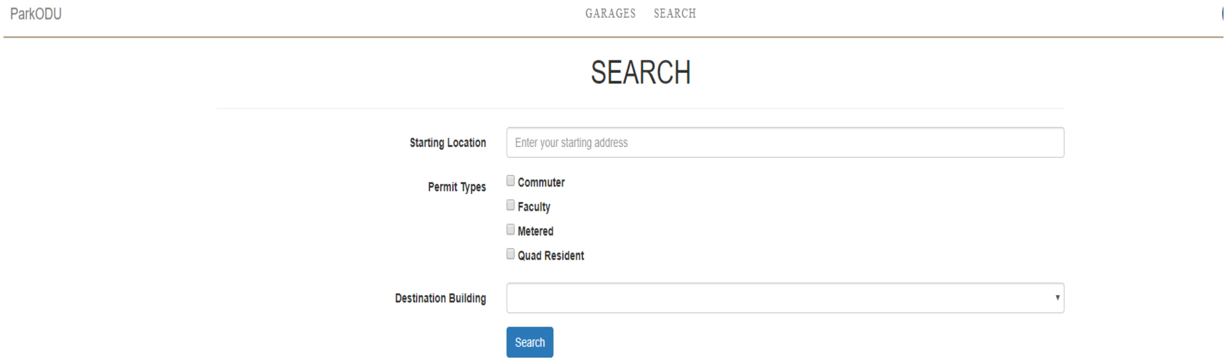
ParkODU is a set of tools to help drivers find available parking spaces around the campus and gather parking space usage information for ODU Transportation and Parking. The application will be available in major web browsers and will also be available as an application on Android and iOS. ParkODU will 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 count 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.

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. ParkODU offers an 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 that are in need of a more efficient way to handle their parking lots.

(This space is intentionally left blank.)

## 1.2 Scope

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. 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. Users will have the option to filter parking by destination building and permit types, see Figure 1. 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 user's 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 satisfied with this new process. See Figure 2 for a detailed flowchart of this process.



SEARCH

Starting Location: Enter your starting address

Permit Types:
□ Commuter
□ Faculty
□ Metered
□ Quad Resident

Destination Building: ▾

*Figure 1 – ParkODU Prototype Filter*

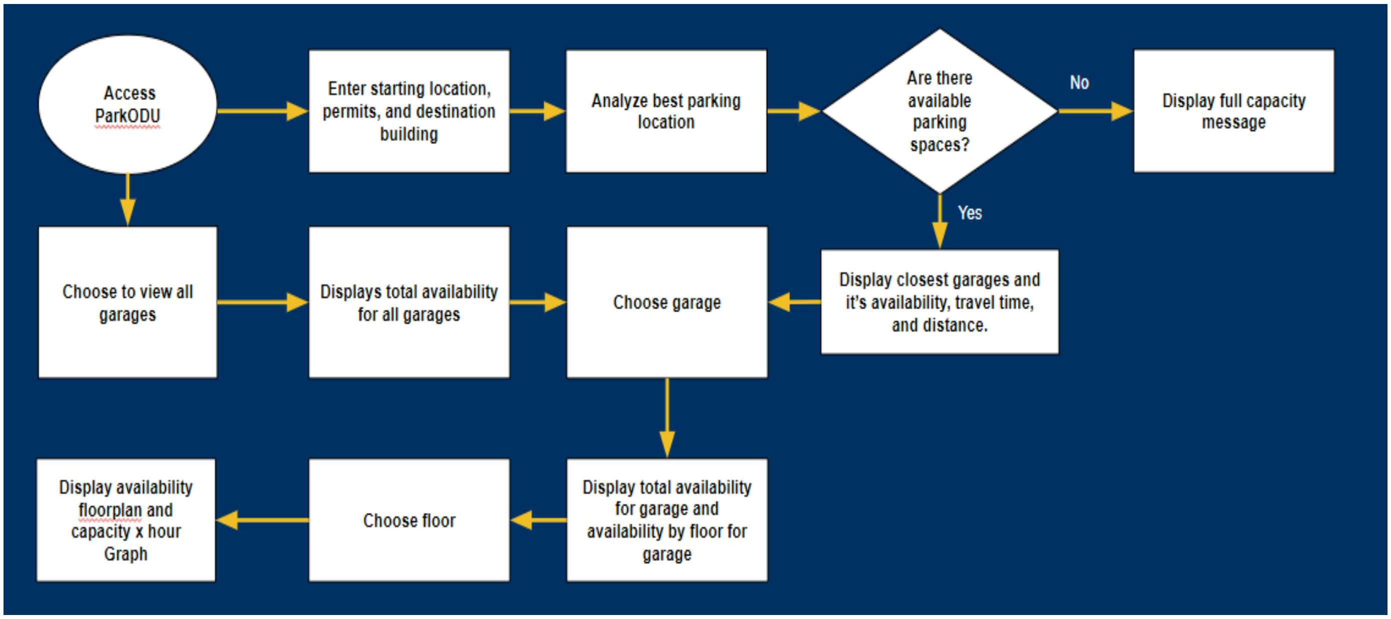

Access ParkODU
Enter starting location, permits, and destination building
Analyze best parking location
Are there available parking spaces?
No
Display full capacity message
Yes
Display closest garages and it's availability, travel time, and distance.
Choose to view all garages
Displays total availability for all garages
Choose garage
Display total availability for garage and availability by floor for garage
Choose floor
Display availability floorplan and capacity x hour Graph

*Figure 2 – ParkODU Use Flowchart*

(This space is intentionally left blank.)

### 1.3 Definitions, Acronyms, and Abbreviations

**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)

**Google Maps API**: API provided by Google for mapping, navigation, and places.

**High Availability**: High availability is a characteristic of a system, which aims to ensure an agreed level of operational performance, usually uptime, for a higher than normal period. Modernization has resulted in an increased reliance on these systems.

**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

(This space is intentionally left blank.)

### 1.4 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/.

Lab 1 – ParkODU. Version 2. (2018, February). ParkODU. Gold Team. CS411W: Imani Mason

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.

(This space is intentionally left blank.)

### 1.5 Overview

This product specification provides a general description and goal of the structure of the ParkODU prototype. The information provided in the remaining sections of this document includes architecture design of the prototype, functionality, capabilities, and features of ParkODU. Also detailed are the various ways the software interfaces with external components. Finally, the specification delivers a defined list of requirements, which the prototype must fulfill. The product specification requirements provided in Lab II Section 3 can be found in a separate document.

## 2 General Description

The primary objective of ParkODU prototype is to provide a working demonstration of the ParkODU product. This is accomplished by implementing only the necessary components and features of the full product. As a prototype, the product will compile the simulated data to demonstrate features and capabilities of the web app and the native Android/iOS App. The software provides the tools required to collect, store, and communicate parking information. The application will compute vacancies in each parking garage in real-time and analyze past parking data for future decisions. It will optimize parking as it includes starting location, permit type, and destination to allow the user to find the best parking. ParkODU 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. ParkODU aims to allow users a comprehensive tool for maximizing the effectiveness of finding available parking spaces around campus.

(This space is intentionally left blank.)

## 2.1 Prototype Architecture Description

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. The hardware for vehicle detection will be simulated by a REST client application. Figure 2 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.

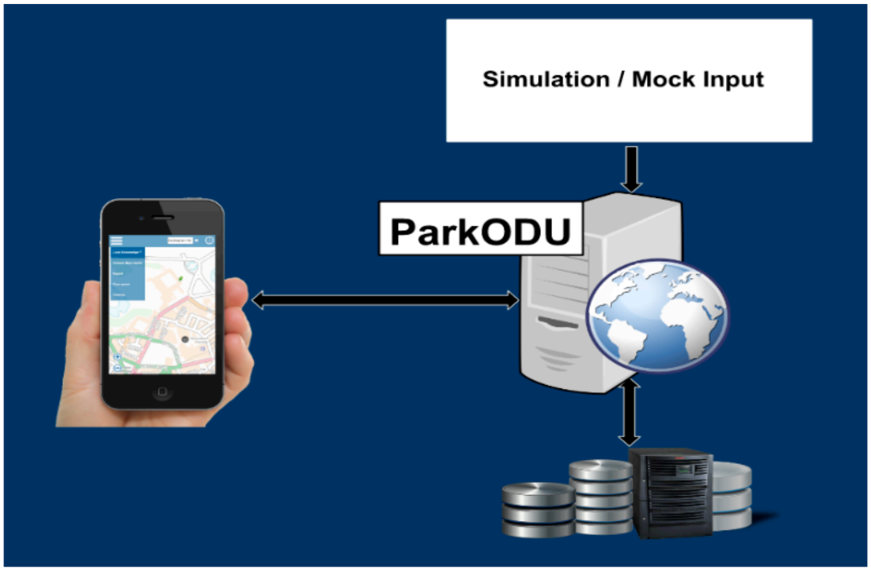


*Figure 3 - Prototype Major Functional Components Diagram*

(This space is intentionally left blank.)

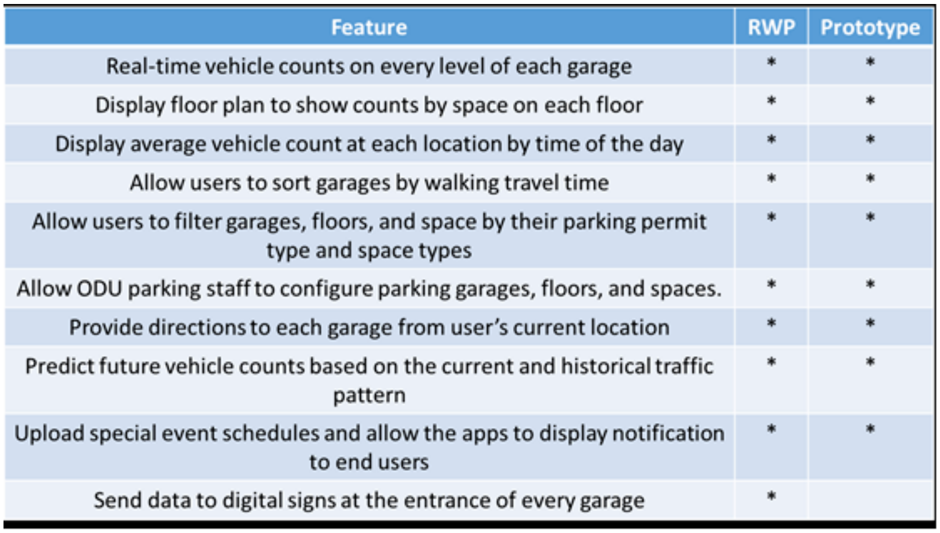
## 2.2 Prototype Functional Description

The application prototype will simulate an input and display a real-time vehicle count by floor in every garage. It 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 option. 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.

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. The prototype will utilize Google Maps API 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.

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 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.

The ParkODU prototype will support nearly all features and capabilities of the real working application. 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. The primary difference between the features of the prototype and complete version of ParkODU are displayed in Table 1.



| 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 Prototype and Completed ParkODU*

## 2.3 External Interfaces

There are four types of external interfaces for ParkODU. These external interfaces are Hardware , Software, User, and Communications Protocols.

### 2.3.1 Hardware Interfaces

The flow of information starts from the vehicle detection technology of a 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. 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. However, the prototype hardware for ParkODU information will be stimulated by a REST API client application. The REST client will replicate data that would normally be obtained from vehicle detection systems. The simulated data will demonstrate features and capabilities of the ParkODU prototype.

### 2.3.2 Software Interfaces

ParkODU is a software-based solution for selecting a parking space 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 appropriate algorithms, functions, and data structures to provide the functionalities listed in section 2.2. ParkODU's prototype will be developed using Java programming language. The software development team will implement Spring Framework for infrastructural support at the application level. IDE IntelliJ Community Edition will be used to assist with debugging the web application.

Hazelcast and MongoDB will be ParkODU primary data stores. The data obtained from the vendor's REST APIs will be sent to ParkODU via requests to REST endpoint. Figure 4 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.  Mapping will be distributed within the MongoDB framework to allow the opportunity for flexible and horizontal scaling. Git will be used to regulate version control of the application. Third-party Google Maps API will be used for key navigation features and future prediction of vehicle count.

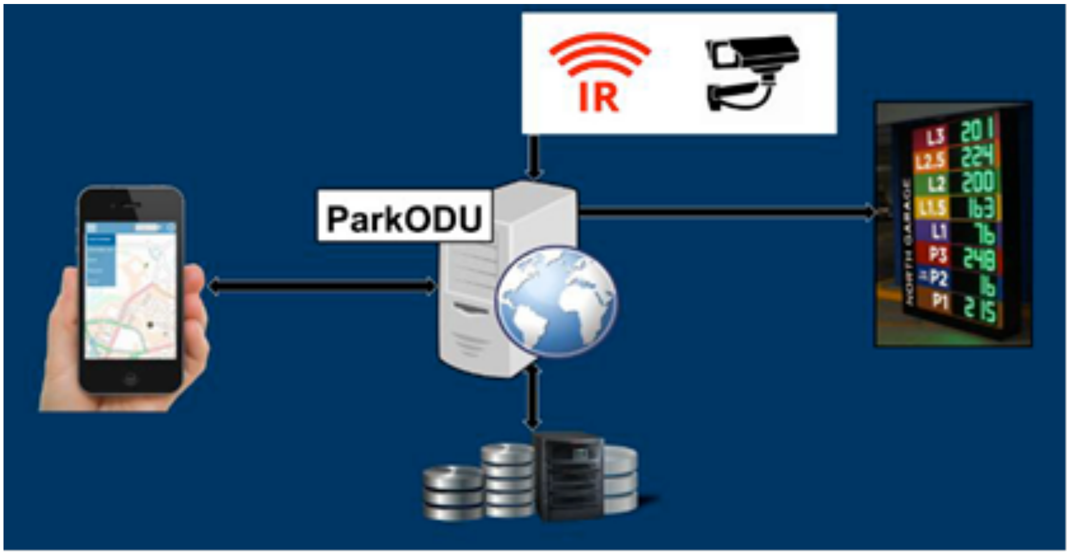
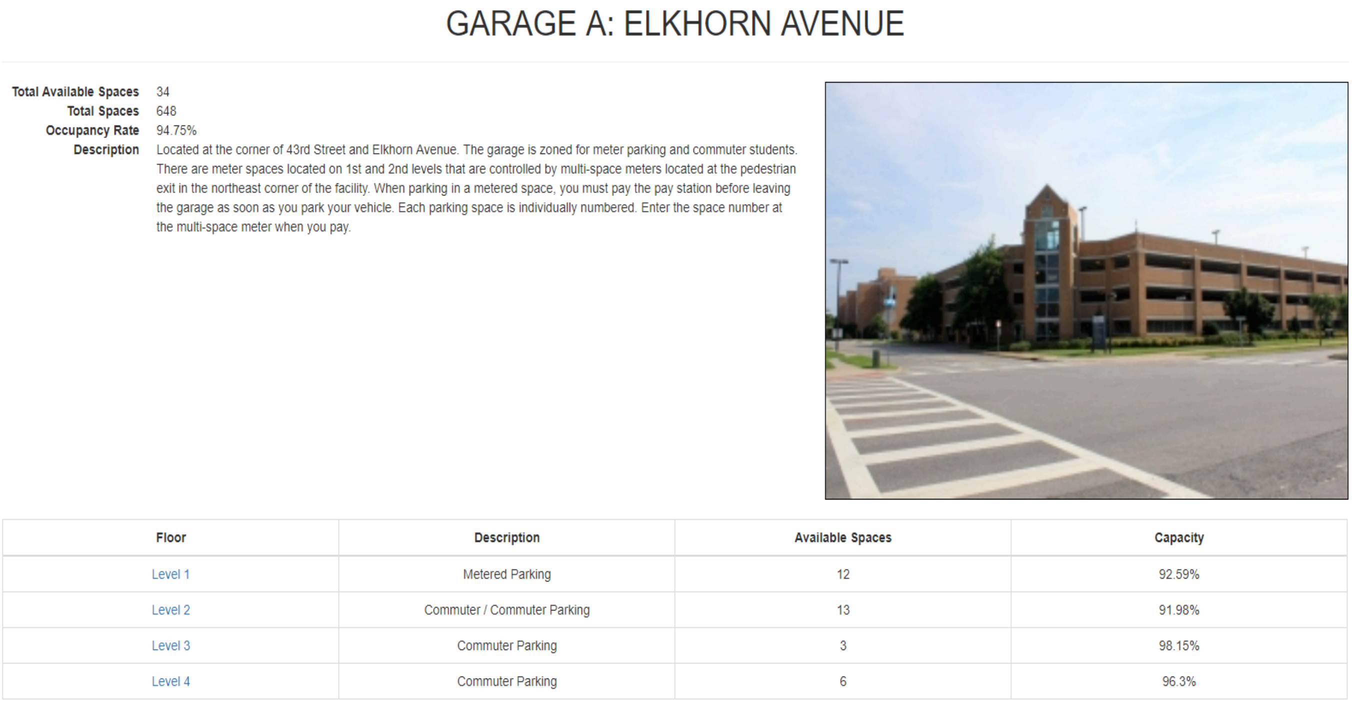


*Figure 4 –Major Functional Components Diagram (MFCD)*

### 2.3.3 User Interfaces

A computer screen or mobile user interface wil be needed to display a live count of parking spaces available by garage so they are able to quickly and effectively locate parking. Figure 5 shows a mockup of an interface a user could see displayed on the screen. Figure 6 displays occupancy rate and a map of available spaces.

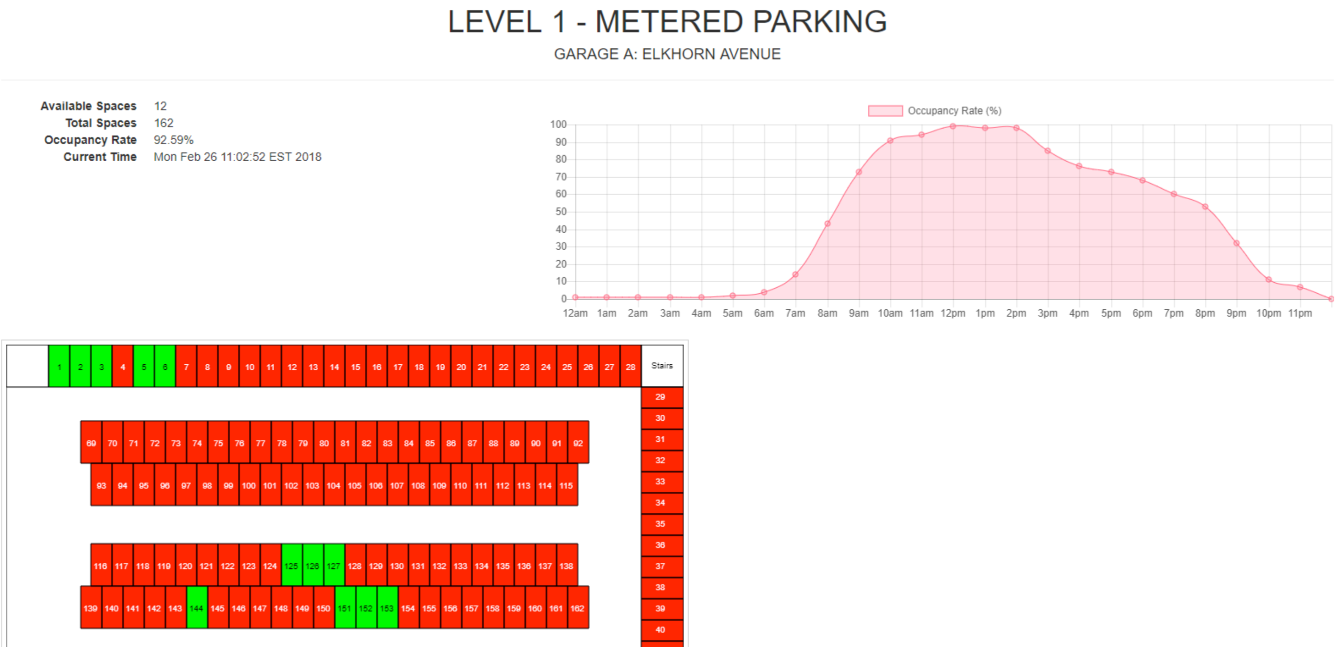


## GARAGE A: ELKHORN AVENUE

| | |
|---|---|
| **Total Available Spaces** | 34 |
| **Total Spaces** | 648 |
| **Occupancy Rate** | 94.75% |
| **Description** | Located at the corner of 43rd Street and Elkhorn Avenue. The garage is zoned for meter parking and commuter students. There are meter spaces located on 1st and 2nd levels that are controlled by multi-space meters located at the pedestrian exit in the northeast corner of the facility. When parking in a metered space, you must pay the pay station before leaving the garage as soon as you park your vehicle. Each parking space is individually numbered. Enter the space number at the multi-space meter when you pay. |

| Floor | Description | Available Spaces | Capacity |
|---|---|---|---|
| Level 1 | Metered Parking | 12 | 92.59% |
| Level 2 | Commuter / Commuter Parking | 13 | 91.98% |
| Level 3 | Commuter Parking | 3 | 98.15% |
| Level 4 | Commuter Parking | 6 | 96.3% |

*Figure 5 – ParkODU Prototype*



## LEVEL 1 - METERED PARKING

GARAGE A: ELKHORN AVENUE

| | |
|---|---|
| **Available Spaces** | 12 |
| **Total Spaces** | 162 |
| **Occupancy Rate** | 92.59% |
| **Current Time** | Mon Feb 26 11:02:52 EST 2018 |

*Figure 6 – ParkODU Prototype Mapping*

### 2.3.4 Communications Protocols Interfaces

ParkODU prototype will be using a TCP/IP and SSH to establish Internet connections to the servers. The database and application servers will service the web and Android/iOS applications.