**DGMD S-598 Digital Media Design Capstone Tutorial**
**Proposal Template - Summer 2018**

# 1.0 Project Scope

## 1.1 Tentative Project Title

An Open Source Wi-Fi Based Device Detection System

## 1.2. Project Goal

The goal of my capstone project is to create a complete low-cost, open source Wi-Fi based device detection system with a machine learning module that is able to analyze and predict future user behavior which addresses the need to better understand user behavior in physical locations. There are existing solutions available today but they are primarily proprietary and much more expensive, requiring specific company made hardware. I do not see machine learning aspects applied to the people counting systems at this time. This solution would enable developers to easily integrate this functionality into their existing systems with options for different hardware suppliers.

## 1.3. Learning Goals

During this project, I plan to greatly improve my understanding of how 802.11 (Wi-Fi) works, explore machine learning techniques, and further my coding skills in languages/frameworks including CakePHP and Python,

## 1.4. Target Audience

Broadly, the target audience for this project is anyone that wants to better understand user behavior in physical locations. This could include marketing teams within larger companies as well as owners of small to medium sized businesses. The immediate target audience for this would be software developers working for these marketers and store owners as this is an open source platform that they could integrate with their existing systems. The technology requirements for the target audience are minimal. In order to successfully install and maintain the Wi-Fi Based Detection System, the user will need stable internet connectivity within the physical location in order to connect/communicate with the system servers and a simple computer terminal to access the system dashboard to view the data, configure the heat maps, and run reports.

## 1.5. Elevator Pitch

By incorporating the open-source community, this capstone project will provide a very cost effective and agile Wi-Fi Based Device Detection System with additional machine learning capabilities that will greatly expand the use of this technology in physical locations.

## 1.6. Metrics

Ultimate success of this capstone would be wide adoption and development of this project by the open source community. Wide adoption of this project would be measured by references and branches from this project's GitHub repository. Moderate success of this capstone would be incorporation of this project into my company's existing marketing intelligence platform.

The measurable outcomes are:

1.) An open source client library that will provides device identification in physical spaces using Wi-Fi
2.) an extension to my startup company's cloud platform that will enable users to visualize the collected information to gain valuable insights into end device user behavior
3.) A machine learning module that is able to determine future behavior of the device and number of people in a particular location

### 1.7. Life of the project beyond capstone

This capstone project will become part of the portfolio of solutions that my startup company, Digital2Go Media Networks (www.digital2go.com), will offer to our customers who fit the target customer criteria mentioned in section 1.4.

## 2. Competitor review

There are many Wi-Fi Based Device Detection Systems available today. The following are three similar systems.

**Meshlium**
http://www.libelium.com/products/meshlium/smartphone-detection/

In addition to Wi-Fi, the Meshlium product from Libelium can detect devices based on Bluetooth. Bluetooth scanning also provides Class of Device information which allows user to determine if the device is a smartphone, handsfree system, computer, or LAN/network AP. The Meshlium product also provides 3rd party cloud connectors which enables users to easily integrate the Meshlium to a variety of popular cloud platforms such as Axeda, Thingworx, and Esri. Although outside the scope of this capstone deliverable, 3rd party cloud connectors would be great to add as it simplifies the user experience if they can stream the device detection data to a platform they are already using.

The primary drawback for the Meshlium product is the price. The base model costs $1,080[1]. A couple other disadvantages for this system are that it does not provide heatmaps and it is not an open source system that users can modify.

**Accuware WiFi Location Monitor**
https://www.accuware.com/products/locate-wifi-devices/

[1] https://manualzz.com/doc/8165880/libelium-products-and-pricing-catalogue page 72

The nodes in the Accuware WiFi Location Monitor system communicate through a mesh network which simplifies installation with large numbers of nodes as it does not require every node to connect to the internet.

Similar to the Meshlium product, the Accuware WiFi Location Monitor is also expensive. Although not as expensive as the Meshlium, each node costs $195 for the cloud based versions and $390 for the standalone version. In addition, there is a minimum $5,000 licensing fee[2]. This product is also not open source so cannot be customized.

**TallyFi**
https://www.tallyfi.com/

TallyFi is a cloud based counting system that enables managers to monitor the number of people in physical locations and display the information in a common dashboard. The primary drawback of this system is that it requires human interaction to count people.

## 3. Technology requirements

**Technology #1:** CakePHP
**Reason:** The existing cloud platform is primarily written in this PHP framework.
**Alternatives**: Laravel
**Course Map**: CakePHP is similar to Laravel, which was taught in CSCI E-15.

**Technology #2:** Raspberry Pi
**Reason:** The system that detects nearby devices is essentially a Wi-Fi Access Point. The Raspberry Pi provides a low cost open platform to create the Wi-Fi Access Point.
**Alternatives**: BeagleBone Black, Intel Edison, Arrow Dragonboard

**Technology #3:** HTML/CSS3/JS/Bootstrap/Ajax
**Reason:** Combined with CakePHP, several web pages will need to be designed to effectively display/visualize the collected information.
**Course Map**: CSCI E-12, CSCI E-15, DGMD-20, DGMD-25, DGMD-27,

**Technology #4:** Python
**Reason:** Python is one of the more popular languages for machine learning.
**Alternatives:** Java, R, C/C++, JavaScript, Scala, Julia
**Course Map:** CSCI-50

**Technology #5:** Amazon Web Services (AWS)

---

[2] http://public.accuware.com/files/Accuware_WiFi_Location_Monitor_Price_List.pdf

**Reason:** AWS provides on-demand cloud computing platforms. The cloud based software for this project will be hosted on AWS.
**Alternatives:** Google Cloud and Microsoft Azure are similar on-demand cloud computing platforms.

## 4. Design workflow

The Open Source Wi-Fi Based Device Detection System will utilize inexpensive connected hardware and cloud processing power to provide insight into user behavior in physical locations. As most people these days have a smart phone with Wi-Fi enabled, a Wi-Fi based system should be sufficient for understanding overall user behavior.

The system will consist of at least one single board computer[3] such as the Raspberry Pi with Wi-Fi Access Point client software running on it and server side code embedded within another consumer engagement and marketing intelligence cloud platform. The system diagram is shown in **Figure 1**.

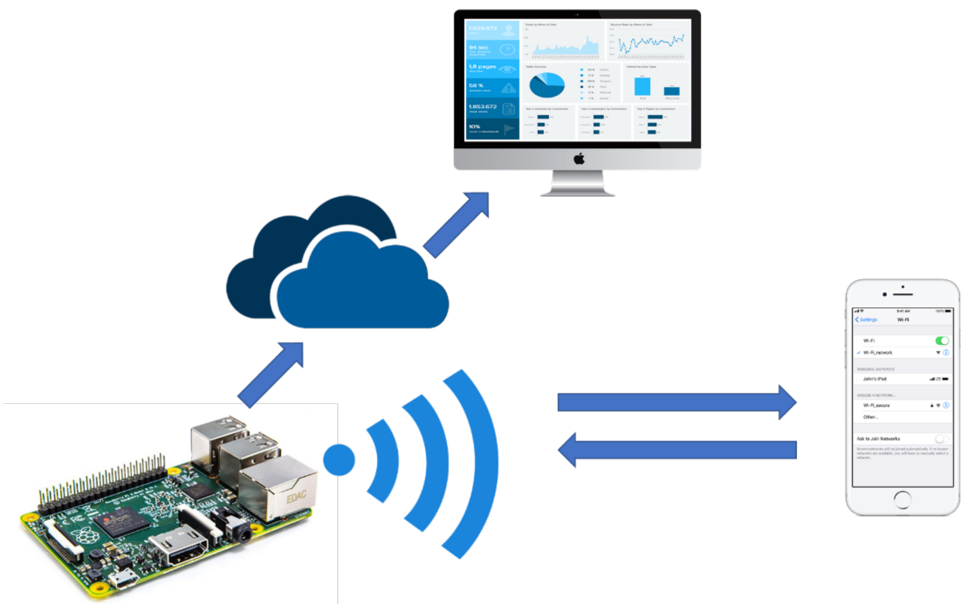

*Figure 1 - System Diagram*

The underlying mechanism that enables this system to function is the continual scan and response behavior of Wi-Fi enabled devices. When devices with Wi-Fi enabled become proximal to a Wi-Fi Access Point, they request basic information such as supported data rates in a probe request to see if it could potentially connect to that access point. In making that request, the device shares information such as its ID which allows the system to uniquely track that device when in range. Note this exchange of information occurs before any connection is made so the device does not actually have to connect to the access point for the system to identify it.

[3] A minimum of three (3) Wi-Fi Access Points are required to perform trilateration needed for the heat maps discussed later in this document.

Now there are examples of Wi-Fi Access Point software available but as this system will communicate to the cloud platform via a specific set of APIs, much of this software will need to be written from scratch. It is assumed that existing code will be used primarily as a guide for what needs to be done.

The cloud based portion of this system will be a module inside the consumer engagement and marketing intelligence platform which is hosted in Amazon Web Services (AWS). A rendering of how the module might be accessed from the sidebar navigation menu in the existing platform is shown in **Figure 2**.

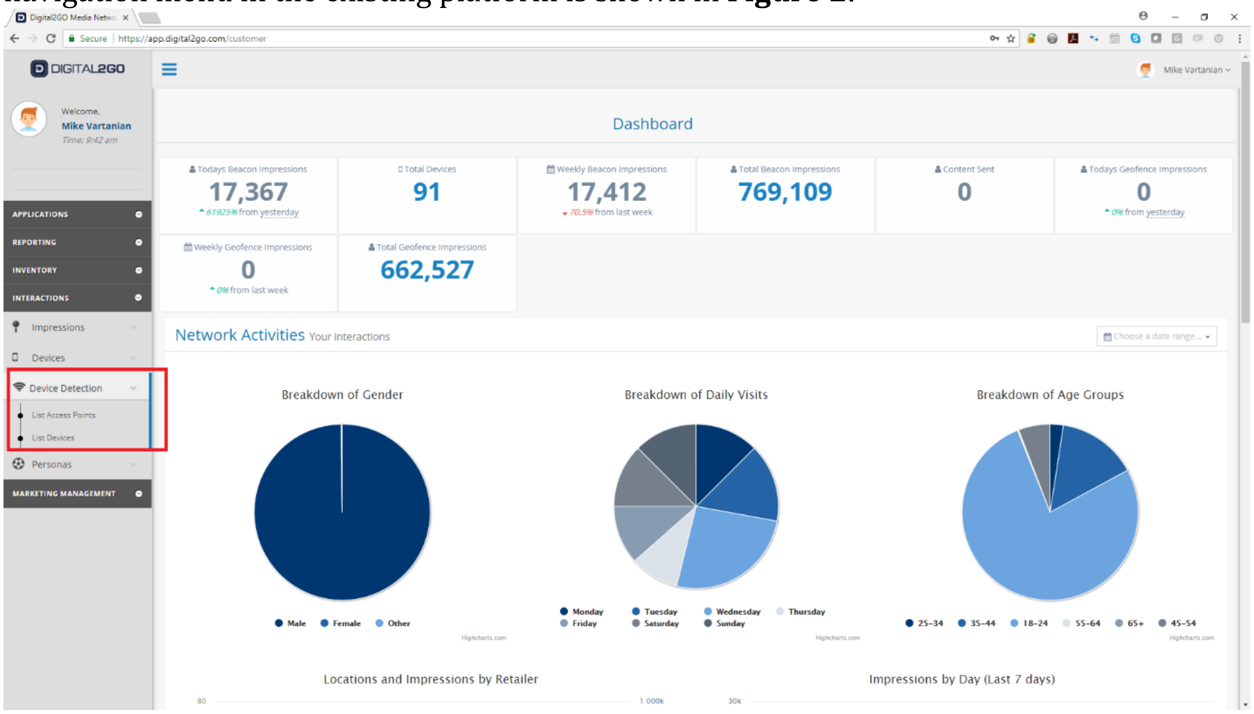


*Figure 2 - Proposed Module Navigation Interface*

As the existing platform is written in CakePHP, the device detection module will also be written in CakePHP. CakePHP is an open-source web framework similar to Laravel which I learned about in CSCI E-15 class here at Harvard Extension. Written in PHP, it follows the model–view–controller (MVC) approach modeled after the concepts of Ruby on Rails and is distributed under the MIT License.
Via RESTful API or stream based connection such as Amazon Kinesis (TBD), the information on devices that have interacted with the Wi-Fi Access Point will be sent to the cloud platform where it will be logged into a database. The proposed data to be collected and logged is shown in **Table 1**.

| Timestamp | Device MAC Address | Access Point MAC Address | Device Type |
|---|---|---|---|

*Table 1 - Proposed Data Collection Fields*

The device media access control address (MAC address) is a unique identifier assigned to network interface controllers for communications at the data link layer of a network segment. Similarly, the Access Point MAC address uniquely identifies the access point. Using the MAC Addresses and timestamp, we can track how many devices are within a particular location and for how long. Additional information such as Device Type will provide additional insight about these devices. Supplementary device information that could potentially be collected will be investigated.

To visualize the collected data, the module will use existing chart plugins such as Highcharts, Fusioncharts, Chart.js, and PHP Charts. The exact chart plugins to be used will be determined after experimenting with various plugins and observing the optimal chart per type. The desired charts for the module will be time series based charts that show number of devices per some specified, configurable time period, as well as number of new versus repeat devices and then a pie chart showing the breakdown of device type among the devices per a specified and configurable time period. Showing where the users are in a location is very useful to location owners and so the system will need to allow users to upload vector based images that will be used for heat map overlays. The heat map will look something like what is shown in **Figure 3** below.

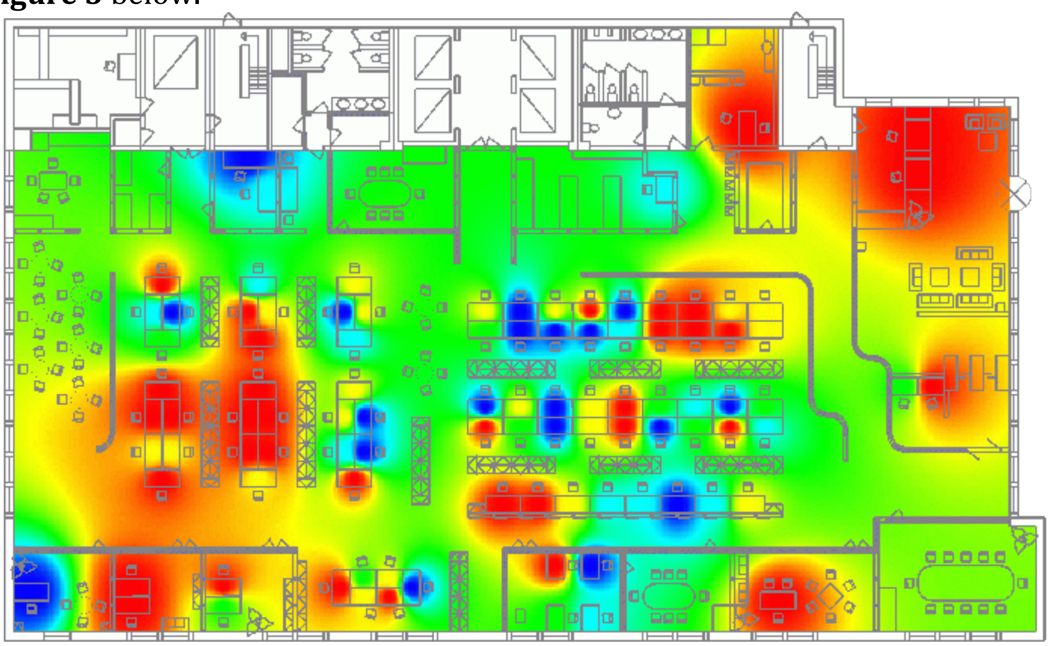

*Figure 3- Example Heat Map Display*[4]

To enable the heat map functionality, multiple Wi-Fi access points will need to be placed throughout the location. Using the concept of trilateration[5], we can determine the location of the user relative to the access points by measuring the Received Signal Strength Indicator (RSSI) from the devices at the access point.

If time permits, the final feature of this system would be to deploy a machine learning module to help make inferences or predictions on user behavior within

[4] Image source: http://f2017cpsc683.ucalgaryblogs.ca/2017/10/12/animated-heatmap-floorplan/
[5] https://en.wikipedia.org/wiki/Trilateration

locations. For example, use the module to infer how many devices will be present at a specific time. As the existing platform uses AWS, the machine learning module will utilize AWS Sagemaker[6]. As machine learning requires many training data sets, it is possible that additional data might need to be injected into this system to show functionality.

## 5. Work plan and milestones

There are five work streams that cover each of the primary system components. The work streams are: Data Interface Design, Access Point Software, System Integration, Data Visualization, and Machine Learning. The milestones consist of finishing each of these work streams.

**Milestone #1:** Data Interface Design
**Date:** 9/17/2018
**Description:** This delivery will define the API structure, database layout, error handling messages, and MAC hashing algorithm.

**Milestone #2:** Access Point Software
**Date:** 10/22/2018
**Description:** Create the software package on the Raspberry Pi that emulates a "pseudo" Wi-Fi Access Point. This delivery will implement the designs from Milestone #1 as it applies to the access point including configuring/streaming data to AWS.

**Milestone #3:** System Integration
**Date:** 11/2/2018
**Description:** This delivery will integrate the Wi-Fi based device detection system into the Digital2Go Consumer Engagement and Marketing Intelligence Platform (https://app.digital2go.com). The MVC files for configuring and viewing the access point settings will be created. Navigation links for the Wi-Fi sections will be inserted into the main menu. This delivery will also fetch the data streamed to AWS from the access point to the cloud platform.

**Milestone #4:** Data Visualization
**Date:** 12/3/2018
**Description:** This delivery will provide a mechanism to display the collected data in a variety of ways such as time series data, cumulative totals bar graphs, and floor plan-based heat maps. The method for correlating floor plan files with actual physical spaces and locations of Wi-Fi Access Points will be implemented. Trilaterization code that locates where the devices are within the floor plans will be developed.

**Milestone #5:** Machine Learning Module

[6] https://aws.amazon.com/sagemaker

**Date:** 12/17/2018
**Description:** This delivery will integrate a machine learning module that attempts to predict device behavior based on existing data sets. Once the Wi-Fi Access Points are functional, several of them can be placed in specific locations to collect data. The data is location specific which means that the data can only help predict future behavior in a similar location. In other words, training data from an office building will not be useful in predicting future data sets from a shopping mall.

Table 2 below shows the delivery schedule.

| WBS | TASK | START | END | DAYS | % DONE | WORK DAYS |
|---|---|---|---|---|---|---|
| **1** | **Data Interface Design** | | - | | | - |
| 1.1 | Define Data APIs | Tue 9/04/18 | Sun 9/09/18 | 6 | 0% | 4 |
| 1.2 | Define Error Handling & Messages | Mon 9/10/18 | Mon 9/10/18 | 1 | 0% | 1 |
| 1.3 | Define Local and Cloud Database Structures | Tue 9/11/18 | Tue 9/11/18 | 1 | 0% | 1 |
| 1.4 | Define MAC Address Hashing Algorithm | Wed 9/12/18 | Sun 9/16/18 | 5 | 0% | 3 |
| **2** | **Access Point Software** | | - | | | - |
| 2.1 | Internet Connectivity Code | Mon 9/17/18 | Wed 9/19/18 | 3 | 0% | 3 |
| 2.1.1 | Implement Error Handling Code | Wed 9/19/18 | Wed 9/19/18 | 1 | 0% | 1 |
| 2.2 | Configure Local Database | Thu 9/20/18 | Mon 9/24/18 | 5 | 0% | 3 |
| 2.3 | Platform Authentication Code | Tue 9/25/18 | Thu 9/27/18 | 3 | 0% | 3 |
| 2.3.1 | Implement Error Handling Code | Thu 9/27/18 | Thu 9/27/18 | 1 | 0% | 1 |
| 2.4 | Scanning Mechanism Code | Fri 9/28/18 | Thu 10/04/18 | 7 | 0% | 5 |
| 2.5 | Result Data Extraction and Storage Code | Fri 10/05/18 | Thu 10/11/18 | 7 | 0% | 5 |
| 2.6 | MAC Address Hashing Code | Fri 10/12/18 | Tue 10/16/18 | 5 | 0% | 3 |
| 2.7 | Data Stream to Platform Code | Wed 10/17/18 | Fri 10/19/18 | 3 | 0% | 3 |
| 2.7.1 | AWS Integration | Fri 10/19/18 | Fri 10/19/18 | 1 | 0% | 1 |
| **3** | **System Integration** | | - | | | - |
| 3.1 | Create MVCs | Sat 10/20/18 | Mon 10/29/18 | 10 | 0% | 6 |
| 3.1.1 | Setup/Settings/Inventory | Sat 10/20/18 | Wed 10/24/18 | 5 | 0% | 3 |
| 3.1.2 | Dashboard | Thu 10/25/18 | Mon 10/29/18 | 5 | 0% | 3 |
| 3.2 | Implement Navigation Menu Code | Tue 10/30/18 | Wed 10/31/18 | 2 | 0% | 2 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 3.3 | AWS Integration | Thu 11/01/18 | Thu 11/01/18 | 1 | 0% | 1 |
| **4** | **Data Visualization** | | - | | | - |
| 4.1 | Database Access Code | Fri 11/02/18 | Mon 11/05/18 | 4 | 0% | 2 |
| 4.2 | Dashboard Layout | Tue 11/06/18 | Tue 11/13/18 | 8 | 0% | 6 |
| 4.2.1 | Plugin Research | Tue 11/06/18 | Thu 11/08/18 | 3 | 0% | 3 |
| 4.3 | Floor Plan Import Code | Wed 11/14/18 | Sun 11/18/18 | 5 | 0% | 3 |
| 4.4 | Device Trilaterization Code | Mon 11/19/18 | Sun 11/25/18 | 7 | 0% | 5 |
| 4.5 | Floor Plan Visualization Code | Mon 11/26/18 | Sun 12/02/18 | 7 | 0% | 5 |
| **5** | **Machine Learning Module** | | - | | | - |
| 5.1 | Algorithm Determination | Mon 12/03/18 | Fri 12/07/18 | 5 | 0% | 5 |
| 5.2 | Collect Data Sets | Sat 12/08/18 | Tue 12/11/18 | 4 | 0% | 2 |
| 5.3 | Run ML Algorithms against Data Sets | Wed 12/12/18 | Wed 12/12/18 | 1 | 0% | 1 |
| 5.4 | Integrate into System | Thu 12/13/18 | Mon 12/17/18 | 5 | 0% | 3 |
| 5.4.1 | Create MVC | Thu 12/13/18 | Sat 12/15/18 | 3 | 0% | 2 |
| 5.4.2 | AWS Integration | Sun 12/16/18 | Mon 12/17/18 | 2 | 0% | 1 |

*Table 2 – Capstone Delivery Schedule*

All work items will be tracked using JIRA project management software.

## 6. References

Reference #1: Meraki | Location Analytics. (n.d.). Retrieved August 3, 2018, from https://meraki.cisco.com/solutions/location_analytics

The URL describes the company's location analytics products. Meraki have a WiFi based device detection product.

Reference #2: Location Services | Ruckus Networks. (n.d.). Retrieved August 3, 2018, from https://www.ruckuswireless.com/products/smart-wireless-services/location-services

The URL describes the company's location services products. Ruckus Wireless have a WiFi based device detection product.

Reference #3: Analytics | Aruba, a Hewlett Packard Enterprise company. (n.d.). Retrieved August 3, 2018, from http://www.arubanetworks.com/products/location-services/analytics/

The URL describes the company's location services products. Aruba have a WiFi based device detection product.

Reference #4: Heat Map Software | Tableau Software. (n.d.). Retrieved August 3, 2018, from https://www.tableau.com/trial/heat-maps

Tableau is a visualization software company. The URL describes their heat map product.

Reference #5: Hot Spot Heat Maps with eSpatial Mapping Software. (n.d.). Retrieved August 3, 2018, from https://www.espatial.com/features/hot-spot-heat-maps

URL presents information on the eSpatial's heat map product.

Reference #6: Automatic people counter for retailers – Counterest –. (n.d.). Retrieved August 3, 2018, from https://counterest.net/en/retail-automatic-people-counter/

URL presents information on the Counterest's people counting product.

Reference #7: Data analytics for the era of omni-channel retail operations | Qlik. (n.d.). Retrieved August 3, 2018, from https://www.qlik.com/us/solutions/industries/retail

URL discusses retail analytics products. URL provides heat map example picture.

Reference #8: CakePHP - The rapid development php framework. (n.d.). Retrieved May 4, 2018, from https://cakephp.org/

The URL provides an overview of the CakePHP framework. The capstone project will utilize these frameworks.

Reference #9: W3C. (n.d.). Retrieved May 4, 2018, from http://www.w3.org/

The URL provides an overview of the World Wide Web Consortium.

Reference #10: Lovely, Stephen. (2017, November 30). How to use your Raspberry Pi as a wireless access point. Retrieved from https://thepi.io/how-to-use-your-raspberry-pi-as-a-wireless-access-point/

Article demonstrates how to set up the Raspberry Pi as a wireless access point.

Reference #11: Amazon Web Services (AWS) - Cloud Computing. (n.d.). Retrieved May 4, 2018, from https://aws.amazon.com/

URL provides an overview of the various services offered by AWS. The cloud platform code for the capstone will be hosted on AWS.

Reference #12: Welcome to Python.org. (n.d.). Retrieved May 4, 2018, from https://www.python.org/

URL provides an overview about the programming language, Python. The machine learning code will probably be written in Python.

Reference #13: WiFi 101: Probe Requests and Responses. (n.d.). Retrieved May 1, 2018, from https://www.hak5.org/episodes/haktip-23

URL provides information about the WiFi mechanisms that enable device detection.

Reference #14: Wadsworth, Chris. (2014, August 26). Why Should I Use a People Counting System. Retrieved from https://www.trafsys.com/why-should-i-use-a-people-counting-system/

URL presents information about people counting systems and the benefits for using such a system.

Reference #15: McCrea, Nick. (n.d.). An Introduction to Machine Learning Theory and Its Applications: A Visual Tutorial with Examples. Retrieved from https://www.toptal.com/machine-learning/machine-learning-theory-an-introductory-primer

URL provides a primer on supervised and unsupervised machine learning techniques.

Reference #16: Jain, Kunal. (2015, June 11). Machine Learning basics for a newbie. Retrieved from https://www.analyticsvidhya.com/blog/2015/06/machine-learning-basics/

URL provides basic information on Machine Learning and how it differs from Big Data and Artificial Intelligence.