50 Stone Rd. East
Guelph, ON, N1G 2W1
519-824-4120 x. 54012
petros@uoguelph.ca

December 3rd, 2020

Dear Dr. Spachos,

As with all campuses, the University of Guelph's student population is growing. This steady growth rate has placed capacity and resource pressured on campus facilities, such as the Library and other student spaces, so much so that the University of Guelph underwent a renovation plan in order to create additional student spaces on campus at the University Center atrium in the summer of 2019. Furthermore, there are plans for further renovations in the University Center basement to incorporate group study rooms and student meeting rooms.

However, due to the recent COVID-19 pandemic, the space issue began to not only pose questions of capacity but also safety. As of right now, the University of Guelph has no infrastructure in place to actively track the capacity of a given space. The present system in order to monitor campus flow consists on an optional online questionnaire for University students and staff members to complete prior to arriving on campus. Thus, not only can physical resources not comfortably accommodate the growing campus population, but during a pandemic the available space is both limited and unmonitored, leading to health and safety concerns. This phenomenon only increases with every year as the University of Guelph's student population increases and the demand for space becomes more urgent.

To make better use of existing university facilities and resources, this project proposes to track the student populations' movement patterns across campus throughout all semesters and use these findings to generate a report for the university's own records. As well, the real-time de-identified student movement data will be made available enable students to make more educated decisions as to where they may study. In addition, this solution will be implemented as a contact-less system, which stands to have societal, public health and safety benefits by improving on the allocation of the infrastructure and resources. The scope of the project encompasses the deployment of an IoT device used for collection of data and transmission to a remote server, which can then be accessed by students via a mobile application allowing them to avoid crowded areas on campus.

Warm regards,

Bilal Ayyache
Dima Ismail
Meera Divakaran
Safwan Hossain
Siddharth Sadhu

Occupancy Counter Interim Report

University of Guelph, School of Engineering
ENGG 41X0, Group 1
Dr. Petros Spachos, PhD, PEng, Course Coordinator

Bilal Ayyache
bayyache@uoguelph.ca
0988616

Dima Ismail
dismail@uoguelph.ca
0919328

Meera Divakaran
mdivakar@uoguelph.ca
0929357

Safwan Hossain
safwan@uoguelph.ca
0918742

Siddharth Sadhu
sadhus@uoguelph.ca
0949213

January 17, 2021

1 Executive Summary

The crowd monitoring project aims to address the issue of safe and reliable access to adequate study spaces and on-campus physical resources available to students. Accessing shared study spaces such as labs, design rooms, and computer rooms has always been a challenge for the growing student population at the University of Guelph. These capacity challenges are only amplified this year due to the provincial social distancing by-laws set in place in light of the COVID-19 pandemic. Keeping a 6-foot distance between each person, and maintaining clean and sanitized shared public spaces is quite difficult to manage under the growing student population competing for these spaces. In order to tackle this problem, we have proposed an IoT tracking and broadcasting solution to make on-campus physical resources a safe and accessible atmosphere.

Key constraints and criteria include that the design should be cost effective, and a Minimum Viable Product (MVP) should be available to be tested by the end of the semester. The design should have an accuracy above 80 percent for both hardware and software components. The UI should be user-friendly and should be able to clearly display relevant sensor and room information, as well as the available room capacity and number of room occupants in a space at a given time. The design should not capture or store any facial (or otherwise sensitive biometrics information) data and should not use any unethical method of engineering to hinder with the privacy of an individual.

The initial brainstorming included three design alternatives, all including a raspberry pi as the core. The key differences lay in using different sensors and methods to identify human motion ranging from IR sensors to RFID tags to a camera module. After performing strong sensitivity analyses and gathering information from credible sources, the camera module was selected as the final design as it was the most innovative and accurate in all testing and literature review models. Using the camera module also opens up many boundaries for the project to expand in scope and application in the future.

The selected design has two major components: the hardware model and a full-stack web application. The hardware model includes a Raspberry Pi with the camera module as discussed above. The back-end of the application is completed and is hosted on the Google Cloud Platform (GCP), with scripts running on Google Fire-base which pulls sensor information onto the database storage. The front-end part of the application includes the use of React framework, and bootstrap components for styling. A database that includes data from all sensors was designed and utilized by the back-end using Django to send information about the monitored facilities to the front-end dashboards.

This application stores real-time and historical data on the database which allows clients and users to understand human interaction with the space being monitored through the utilization of Open-CV. Using an Open-CV model and the Raspberry pi camera, the software application can predict whether an individual is walking into or out of the frame. Pixel changes from the frame compared to the main background frame are tracked to determine whether the individual is leaving or entering the building.

Using this technology, the problem proposed is provided with a very reliable and inexpensive solution. This solution will provide occupancy information for students and faculty members which can be used to improve space utilization at the University Of Guelph.

2 Acknowledgements

The design team would like to thank Dr.Petros Spachos for the time he devoted into this project to ensure success. We would also like to thank him for the mentorship time and consultation provided during implementation. His expertise in IoT and wireless sensors helped us improve this project to meet the scope of the project.

Contents

List of Tables

List of Figures

3 Introduction

The project presented is an occupancy counter that is used to monitor the number of people in a room or globally in a building. The main purpose of this project is to create an easily accessible guide to finding study spaces on campus while maintaining safe distancing protocols and adhering to reduced maximum room occupancy guidelines. The project was achieved by combining a physical IoT device paired with an app interface displaying real time occupancy data. This app can be downloaded by students on their mobile phones and tablets and used further by building maintenance to indicate areas of high and low usage for energy and resource distribution purposes.

4 Problem Description

4.1 Problem Statement

The problem addressed by this project is the study time wasted seeking appropriate study spaces on campus. During exam season and other busy times during the semester the library study spaces available are few and the other study spaces around campus get full very quickly. Finding a space to study often takes up a lot of time and becomes almost impossible when trying to quickly get a couple hours of studying in between classes. Additionally, in order to follow social distancing guidelines, buildings have to operate at reduced capacity which makes safe study spaces even fewer and harder to find. This problem is common among many university campuses with growing student populations. While resident students may retire to their own living spaces to study, this is simply not an option to commuting students who live far from campus. Many commuting students rely on campus study spaces to do school work as they must spend extended periods of time on campus. It is imperative to these students to have ample study space in order to facilitate studying and encourage a strong work ethic. Unfortunately, it is not economically feasible to create new study spaces on campus. Instead, the university must find a way to better utilize existing space and accommodate the growing need for study spaces due to the growing student population. Due to the spread of COVID-19 there is an additional level of safety that must be considered. Major institutions must operate at reduced capacity to support social distancing. It is important to make social distancing as easy as possible for students while studying to keep everyone safe and still allow for productivity on campus. From an environmental perspective, there is a need for optimization in energy consumption and resource distribution. Empty spaces do not need to be powered and can operate at a minimum energy consumption when not in use. Data on building occupancy and usage will help in the redistribution of energy and resources to reduce the environmental impact of large unused space.

4.2 Design Objectives

The project focuses on the following objectives as a measure of success:

- The design must be able to monitor traffic through various study spaces in order to detect occupancy.
- The system must be paired with real time data analysis to provide information on current study space vacancies and provide space usage data to facilitate a reduction in energy consumption and resource distribution efforts.
- The final product must include a user interface to allow for review of available study spaces and occupancy levels.

To meet objective 1 the team designed an IoT device which uses a raspberry pi camera which collects motion capture and heat map data in real time. This allows the device to detect people moving in and out of a room and keep count of the number of people within at all times. To meet objective 2 the physical device captures and transmits the real time data captured to Google Cloud which hosts the back-end. This allows the system to analyze the data and provide accurate information on the occupancy of the space in real time.

To meet objective 3 the team designed an app interface which users can download onto their mobile devices. Through this app users can search for occupancy data on specific rooms in a building and review available study spaces in real time.

4.3 Literature Review

There are a few technologies which are currently in the market using various techniques to count occupancy. Most common amongst the technologies used are Radio Frequency Identification or RFID. Different methodologies have been used, using RFID technology to count occupancy of a room. One such technology as proposed by researchers at Cornell University and Montana State University is called "Indoor Occupant Counting by Ambient RFID" [1]. This methodology uses multiple RFID antennas on the ceilings and swarms RFID tags on the walls, spanning from what one may assume from tens to hundreds of RFID tags across the walls of the rooms [1]. The presentation claims to hide the tags behind wallpapers to make the process of counting occupancy non-invasive [1]. However, the ideal setup for such a methodology also includes the having less furniture inside the room, to remove obstruction and better estimation of the people inside the room [1]. The installation of such RFID tags and antenna also will be dynamic based upon how the room is set up and the furniture inside the room. Making installation of such an implementation a challenge. To overcome such, a challenge another technology proposed by the company Tagit Global uses RFID antenna at the ceiling near the entrance of the door and counts the tags of the people coming in to check how many people walked in or out of the room [2]. This system although overcomes the problem of using vast swathes of tags all over the walls of the room, it poses problems of accuracy especially in an academic setting. If a student is not carrying a tag with them system by default would not count such a person. Furthermore if the student is carrying the tag, if the tag is in their backpack or wallet the systems may or may not be able to read the tag. Accuracy proposes a problem whenever it is implemented to count people in an unobtrusive manner. This is further underscored by another research which proposes people counting by RFID by the University of Southern California called "Measuring and monitoring occupancy with an RFID based system for demand-driven HVAC operations" [3]. This proposal shows hat the accuracy of their system is 88% for "stationary occupants" and 62% for "mobile occupants" [3]. Both being far lower than the system we are proposing.

4.4 Assumptions

The two main assumptions for this project include that there will be a power supply nearby to connect the raspberry Pi and power it and that there will be Wi-fi available which will be connected to the raspberry Pi. The Raspberry Pi can also be connected to the internet via an ethernet port. The power supply will help power the raspberry Pi and the internet connection via Wi-fi or ethernet will allow the Pi to communicate with the back-end of the system. The back-end of the system is where the current data of the room or space where the Pi is placed can be updated.

4.5 Other Considerations

The environmental impact of such a project is minimal. The project uses batteries Raspberry Pi, cameras and batteries, all of which can be recyclable. Raspberry Pis are also do not consume a lot of power and do not require a lot of power to run, therefore the environmental concerns for such a project is minimal. One major criticism of our proposal could be privacy since, cameras are being used. However, since the cameras are not collecting image of peoples faces, or saving the images and the images are greyed out in a manner which makes it very difficult to make peoples faces discernible by a human, it uphold privacy. Furthermore we do not and cannot collect who is in the room based upon the data we collect, only how many people are in the room, regardless of whether they are carrying a tag or not, thereby upholding privacy while improving accuracy.

5 Project Constraints and Criteria

Constraints and criteria must be designated for the project in order to ensure the project meets a specific set of goals whilst following a certain set of guidelines. The criteria encompasses the aim of the project and what the project aims to accomplish. The constraints outline a certain set of guidelines that the project must meet in order to successfully implement the project. Failing to meet the constraints would make the design unfeasible. The constraints are limitations on the project which the project must not exceed. The constraints consist of accuracy, cost, time, physical constraints and software constraints. The following table extensively dives into each constraint and explains the current progress of the project against each constraint.

5.1 Project Constraints

The following table (Table 1) summarizes the constraints for our project and the current status of the project against the stated constraints.

Constraints	**Definition**
Cost	The project must not exceed \$ 100 per unit. This ensures that the project is kept within a budget which can make it scalable across a University campus or any other setting.
Time	The initial prototype of the project must be completed within mid to late December starting from September. This means the project prototype must be completed within 3.5 months.
Accuracy	The project must at least have an accuracy of 80% this ensures that there is enough accuracy such that people will use the product.
Physical Constraints	The physical dimensions of the project must not exceed 100 cm3 such that it can easily be installed and also does not cause an obstruction in the environment it is being used.
Software Constraints	There must be a UI to display the data being collected by the device, otherwise, the entire project loses it's meaning since the data collected is not utilized.

Table 1: Constraints Table

- Cost: A raspberry Pi 3 being used costs \$50 and the a camera which is used which costs another \$40 dollars. This brings the physical cost of the setup to \$90. The other \$10 is used for physical storage of data in Google Firebase which provides free storage up to 1 GB and free 10 GB storage to be downloaded from a real-time database. This brings the the current cost of the project to \$90. When the project will be used at scale depending on the number of devices the cost can be \$5 per GB store and \$1 for every GB downloaded. Hence, the back-end will need to be designed in a manner such that the costs of \$10 per device for storage is not exceeded. At the same time when the product is bought in bulk. The price of the camera and Pi will come down meaning more of the budget can be utilized for the software. For the current status of the product, however, the cost is at \$90 meaning it is within the budget of the project and meets the cost constraint.

- Time: The current status of the project indicates that the the project can be completed on time. This is because the physical component of the project is currently fully completed. As the back-end and front-end of the project are currently under work. As the Gantt chart will indicate the back-end and front end work will have to continue until the almost end of the project. As it currently stands we are 1.5 months in and have fully completed the capturing and counting portion of the project.

The prototypes for the app are also completed and the back-end also collects real-time data from the physical prototype. What is left to be completed is creating all the endpoints in the back-end to collect data for the different types of data and the front end to be able to show the graphs to show the real-time data on an individual phone. As for the prototype of the project, the project is on schedule. For a more scalable product, more time will be needed to ensure proper testing of the software and more features will need to be added to make it a sellable product.

- Accuracy: The product was tested under different situations and met the minimum constraint of 85% accuracy. The product was tested 100 times, under different situations such as multiple people going into and out of rooms simultaneously, people walking in and out with service dogs, multiple people walking in and out simultaneously, etc. and in our tests about 85% of the time the project succeeded. This meets the minimum constraint for accuracy and makes the product viable.

- Physical Constraints: The dimensions of a raspberry Pi 3 are: 8.5 cm by 5.6 cm by 1.7 cm. This gives a total volume of 80.92 cm3. This camera of the raspberry pi has dimensions of 2.5 cm by 2.4 cm by 0.9 cm giving us a total volume of 5.4 cm3. This brings the total physical volume of the Pi and camera to 86.32 cm3. This is well below the 100 cm3 dimensions constraint and hence, will not be an obstruction to the physical space. This constraint is currently being met with no issues.

- Software Constraints: The current progress of the frontend portion of the project only consists of the prototype and the graphs are currently being worked on. This constraint currently is not met since the data from the backend is not being displayed. In order to meet this constraint, the real time data of each room or space were a physical device is put must be displayed along with the last 24 hours data in the form of a bar graph. The prototypes for the graphs explain this extensively.

5.2 Project Criteria

The table below (Table 2) summarizes the defined criteria for the system:

Criterion	Explanation
Accuracy	The accuracy of the system should be greater than 90%. The machine learning model should be able to accurately predict the number of people in a building. An accurate prediction makes this system a reliable system.
Usability	The effort required by the user to learn the usage of the software should be minimal. To satisfy this criterion, a well designed graphical user interface will be designed. The graphical user interface designed for a specific system will guarantee usability of the software application by providing data on a clean dashboard.
Integrity	Code integrity will be satisfied through extensive unit testing under different possible scenarios. Proper code integrity will result in shorter development time, lower development costs, and ensure efficient quality assurance. Code integrity will be measured by subtracting 100% from the non-covered bugs divided by total bugs.
Efficiency	Maximum efficiency of algorithm will result in effective usage of storage space, and faster execution time. To achieve maximum efficiency, proper programming methods and tools will be utilized. A good efficiency can save power and increase the lifetime of product.
Reliability	The software application should be defect-free and should not fail during operation. The system should be able to recognize whether the object in frame is a person or not. The system should also be able to recognize whether a person is going in or out of the frame.
Safety	The software application should not be hazardous to the environment/ life.
Security	The software application should not have ill-effects on data or hardware. This could lead to hardware vulnerability and data breach. Data collected should not be accessible to anyone but the client.
Maintainability	Software application should be easy to maintain such that time and effort required to troubleshoot defects is minimal.
Testability	Software application should have preset testing models to ensure system is functional, and software issues is minimal.
Flexibility	Software code should be dynamic and modifiable to ensure time and effort required for making modifications or implementing new features is minimal.
Scalability	Software application should have the ability to increase performance in demand to meet requirements. The database application that gives a good response time for 100 users should be scalable for 1000 users if required.
Extensibility	Software application should be easy to modify and enhance if new features are implemented.
Modularity	The code of the system should be easily modifiable and testable. This can be achieved by separating independent parts of system (creating subsystems).
Portability	The system designed should not encounter issues if environment is altered. System should be easily transferred from one environment to another without changing the functionality.

Table 2: Criteria Table

6 Design Methodology

6.1 Design Alternatives

There were a total of 3 design alternatives that were taken into consideration for this crowd monitoring project. Considering all the constraints and criteria in the above sections, the hardware options for the project revolved around Raspberry Pi since it's one of the cheapest and smallest computing machines out there for the state of the art IoT projects. The accuracy and extensibility are one of the most important criteria for this crowd monitoring project since the estimation needs to be absolutely correct in order to predict the number of people gathered in a shared space. The Raspberry Pi is the go-to device for the project since it has a complete Linux server with both Bluetooth and WiFi compatibility, which is great for wireless communication. [1]

The design alternatives included: Raspberry Pi paired with an IR sensor Raspberry Pi paired with a camera module Raspberry Pi with individual RFID tags for each student's identity

Since the project involved a study on human motion detection, the privacy standards were kept in mind to reach these design ideas. No design involved recognizing a human's face or store their movement behaviour. All the designs were brainstormed in such a way that the data points collected for the motion would be totally anonymous, and any identities stored in the database would be randomly generated and would have no unethical existence. All the ideas involved a physical device located at the entrance of a shared space that students use on an ad-hoc basis.

The first alternative involved a raspberry pi device containing an IR sensor to sense human motion detection through Infrared waves. The waves would be generated through the device and if they encounter any interference, it would assume that there was a moving object entering the space. [2] The second alternative involved the device attached with a small camera module which would use OpenCV and machine learning algorithms to detect motion, and possibly even generate a heat map to decide between different moving objects. The third alternative required a stationary Raspberry Pi device with an RFID reader on the entrance of the shared space with RFID transmitter tags distributed to students to attach to their student IDs. [3]

6.2 Alternative Evaluation Procedures

For evaluation of the design alternatives, a strong sensitivity analysis was performed to select the best design among the three. Below is shown how the analysis was conducted:

		RANK(1, 2 or 3)		
Constraint/ Criteria	**Weight**	**Rasp Pi w/IR**	**Rasp Pi w/ Cam**	**Rasp Pi w/RFID**
Cost	0.8	2	1	3
Time	0.8	3	1	2
Accuracy	1	3	2	1
Efficiency	0.7	2	1	3
Extensibility	0.5	3	2	1
Safety	1	2	1	3
Reliability	0.9	3	1	2
Portability	0.8	3	1	2
Usability	0.8	3	1	2
Total		**19.4**	**8.8**	**15.6**

Table 3: Sensitivity Analysis Table

The total is calculated by the sum of Rank x Weight which makes the second alternative the best out of the three options. Here's break down the categories for the explanation of weights and ranks selected.

The cost and time to finish the project are very important aspects of the decision and the Raspberry Pi with camera setups costs far less than the Infrared and RFID sensors and hence it ranks higher on the list. As far as accuracy is concerned, the RFID solution ranks the highest on the list, since it requires almost no image capturing and machine learning techniques to identify human motion, it just gets a reading when someone carrying the RFID tag passes the device at the entrance of the shared space. For efficiency and safety, the camera alternative ranks much higher than the IR and RFID solution, since it does not emit any radio/infrared waves into the human body, while it is still considered very low risk by scientists and doctors, it is still to be taken account for. [4] The camera solution ranks at the top when it comes to reliability, portability, and usability. The small device with effective image recognition is extremely reliable and user-friendly and can be mass-produced and shipped very quickly due to the better availability of simple hardware. After this deep analysis, it was decided to move forward with the second alternative which is a Raspberry Pi device with a camera module.

6.3 Selected Alternative

The design chosen is a raspberry pi device paired with a camera module to detect human motion entering/leaving a shared space. The device will collect data and information including motion capture and heat maps in real-time and transmit that information to Google Cloud which hosts the back-end. The data will be processed using some open-source algorithms as well as in-house solutions and would be also saved in the database for future use. The hardware devices would be placed in various locations on the campus, and the software application would be available for both web and mobile, to create an ultra-modern IoT system where the user can view the data through the devices from any location on any platform. Google functions will run on the data collected and with the use of cloud tools, all the information will be sent back and forth to the front-end of the web application. The front-end of the web application would include an interactive user-experience system where the user would be able to remotely see all the devices and interactive crowd graphs across different locations on the campus. The application would have a counter of how many people are currently present in a shared space as well as the general trend of the crowd over a particular hour/day of the week. The whole IoT full-stack application will be robust and extensible using a combination of open-source APIs as well as natively written endpoints and algorithms.

6.4 Design Approach

The current status of the project has the physical device functioning. This means the Raspberry Pi is able to utilize the camera and be able to detect people going in and out of a room or a space. The Raspberry Pi utilizes OpenCV to be able to distinguish people from other objects and is able to accurately detect with about 85% accuracy based off of 100 tests. Alongside this the Raspberry Pi is also able to send real time data to the back-end about the certain room or space. In terms of front end work the initial mock ups of how the product should look like is completed and the framework being utilized to create the front end is also set up. What remains to be completed is the endpoints for the different types of data that will be displayed. In our case the different types of data are real time graphs which means the endpoint will need to send a timestamp along with the number of people. Also, the real time data the number of people in the rooms the raspberry pi is operating will need to be another back-end endpoint. This will complete the back-end portion of the project. The front end will need to work on displaying the line graph and the bar graphs for the data received from the back-end. There will also be a need to add a feature of searching for all the rooms for which information of its capacity is available. Therefore both back-end and front-end have three major components still left to be done with about 2 months left. This team is confident that this is sufficient time to complete the project since the entirety of it is software related and there is enough software experience within the team to complete the project.

6.5 Plan of Attack & Milestones

The plan of attack for this project, as well as key milestones and deliverables throughout the Fall semester, have been tracked in using the GANTT chart below.

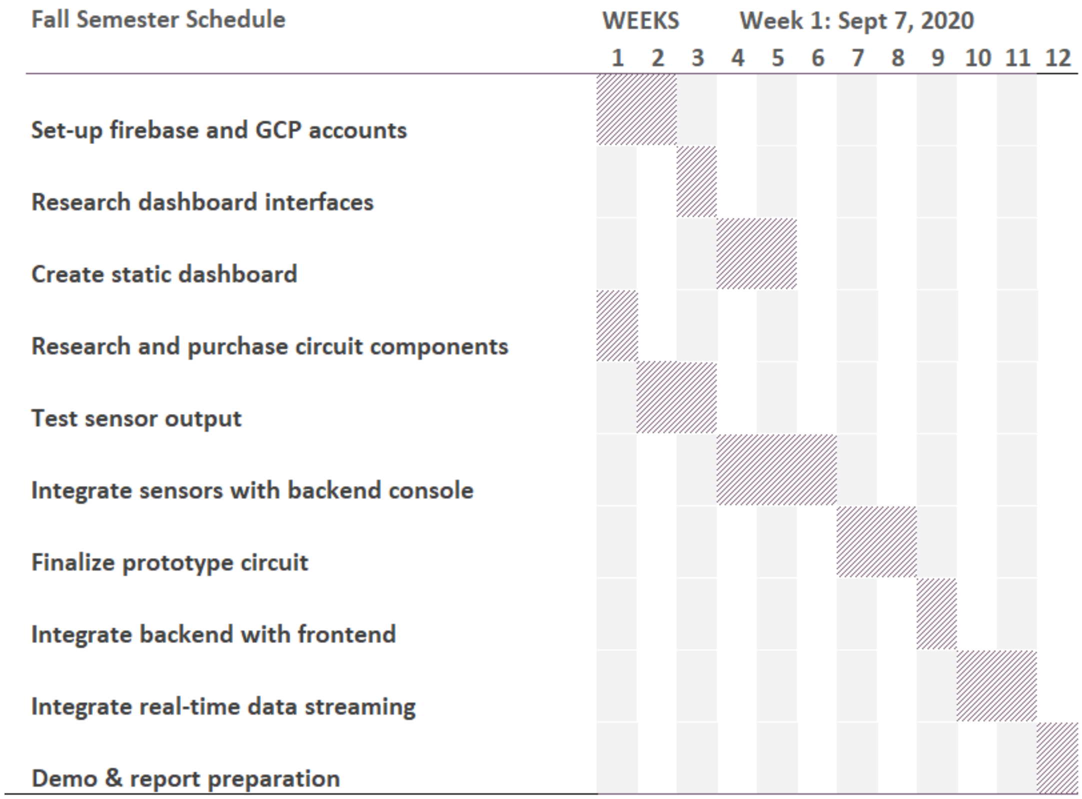


Figure 1: GANTT Chart for the Fall Semester

Key milestone events and deliverables include sensor testing and integration within the back-end console, due around the end of week 6. This key milestone has recently passed, and has been met. Upcoming key milestones include the finalization of the prototype circuit, and backend-to-frontend integration towards the end of weeks 8 and 9, respectively.

As of the submission of this report, it is presently the end of week 7, and all key milestones within weeks 1 to 6 indicated on the GANTT chart above have been met in a timely manner. The project is on schedule and approaching completion.

6.6 Analysis Techniques

To design this software application, different hardware and software tools were used. A simulation model was designed to test the system and ensure functionality. In this section the resources, calculations, and simulation results are presented. This section also highlights the design stages and iterations, and how specific outcomes and critical events impacted the design.

7 Results

7.1 Hardware Resources

In this section the hardware resources and tools used to design the software application is highlighted. A Raspberry Pi Model 3 was used as a micro-controller in companion with the Raspberry Pi Camera to build the design (figure 2).

The following table highlights the specifications of the hardware system:

Rasperry Pi Model 3	
SOC	Broadcom BCM2837
CPU	4x ARM Cortex-A53, 1.2GHz
GPU	Broadcom VideoCore IV
RAM	1GB LPDDR2 (900 MHz)
Storage	10/100 Ethernet, 2.4GHz 802.11n wireless

Camera Module V2	
Resolution	8 Megapixels
Video Modes	1080p30, 702p60, and 640x480p60/90
Pixel Size	1.12 um x 1.12 um

Table 4: Hardware Specification

In this project, the rgb video input from the camera is converted to 480x360 still resolution with 30 frames per second (24*480*360*30). This setting makes the data processed 12 times smaller compared to the normal image provided by the Raspberry Pi Camera. This will improve the efficiency and accuracy of application. The CPU runs on a 1.2 GHz 4 core ARM Cortex-A53, this can easily handle the data provided by the camera with no lag. The camera module will be sending about 124 mb/s which is around 10 times smaller than max data transfer speed a Raspberry Pi can provide.

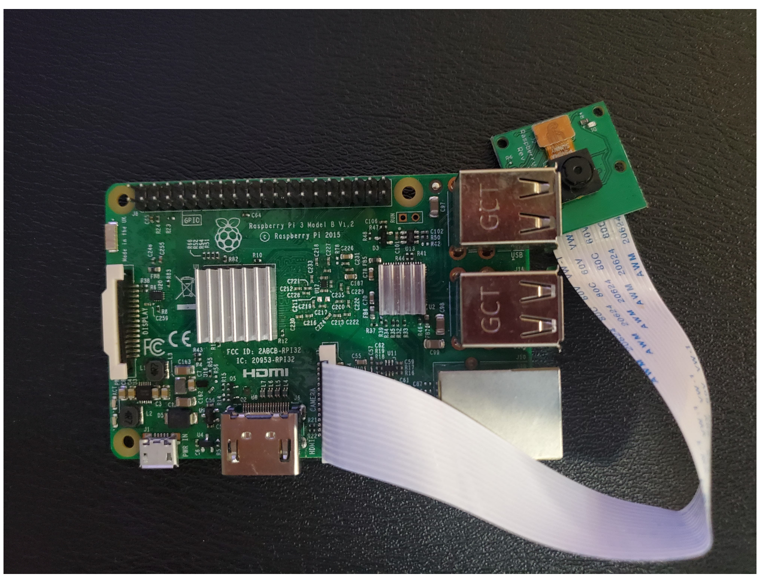


Figure 2: Hardware Components Used

The figure below (figure 3) shows how the system is installed on ceilings. In this figure, double sided tape was used to mount the casing on the ceiling.

Figure 3: Installed Hardware with Casing

The figure below, figure 4, demonstrates the cross-sectional view of the hardware implementation, with each individual component labeled.

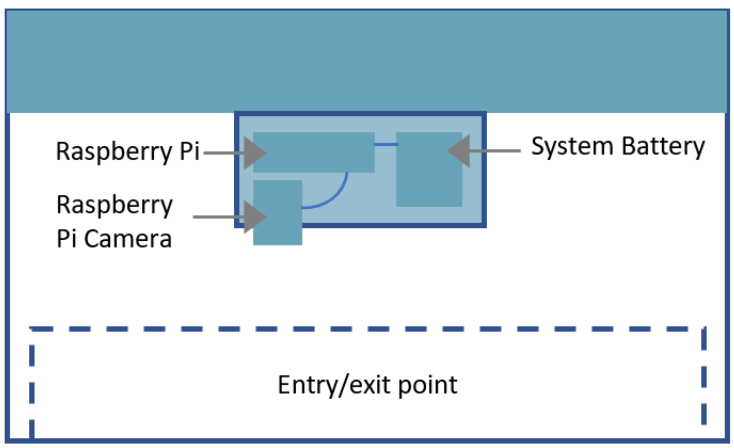


Figure 4: Cross Sectional Diagram of the Hardware Implementation

7.2 Software Resources

The back-end of the software application uses python as a programming tool to interact with the hardware. Python 3 was used to program a script that utilizes Open-CV for image processing. Changes in the x-values were monitored to identify the movement of the person in the frame. Coordinates of the motion from the Pi camera were stored in a list for processing. To make the system flexible a max area threshold variable was added to the program. The max area threshold allows the user to improve accuracy depending on the height of the beacon.

in real-time, data from the pi camera is first filtered then processed. The frame is first resized from 1080 to 500 for faster processing and less data transfer. The Raspberry Pi 3 have limited specifications, therefore the data processed should be minimal to maintain speed. A grey image filter is applied to improve accuracy and make it easier for the Open-CV model to accurately estimate the direction of the object in the frame. A blur filter is then applied to saturate the colors to improve the accuracy even further. After filtering is completed, processing of the image starts. Using Open-CV, the running average of the image vector is updated to find the difference in the original background and the current frame. After differences are found, the pixels are turned into black and white to make it easier for the model to estimate an accurate output. The vector is then dilated to compute the maximal pixel value overlapped by the current frame and replace the image pixel in the anchor point position with that maximal value. Finally, Open-CV is used to find the contours and track the object as it moves across the frame. A find majority algorithm was used to find max area in frame. The back-end system provides the front-end system 3 different variables. Those variables are the number of people inside, number of people that entered the facility, and number of people that left the facility.

The front-end of the software application uses the React framework with Material-UI bootstrap components. The first phase of the front-end development has been completed which includes a login page, a main dashboard with a side navigation bar which connects the user to multiple components including sensors, battery usage, campus map etc. The dashboard can be customized according to the user's preference and is fully optimized. The second half of the development process includes connecting graph components to the endpoints defined in Google Firebase, and use Google functions to carry real-time JSON data from our sensors to the front-end of our web application. Comparing this to the design objective and software constraints mentioned in the beginning of the report, this aspect of the project is not fully met. Due to not being able to interact with the sensors, and other colleagues physically, the front-end part of the website remains a future improvement when the sensor data could be properly seen in a pretty manner.

7.3 Design process

The first step in the design process was to determine whether the hardware resources can achieve the desired goal. The system was focused on solving the problem statement in the cheapest and most efficient ways while achieving a high accuracy. The design was first implemented on a Qualcomm dragon board but due to the low GHz from the CPU, frame drops occurred resulting in inaccurate results. The microcomputer was changed to a Rasberry Pi to meet the requirements of the system. During the first phase, different sensors were used but output was inaccurate. The sensors was later changed to a Pi camera while depending on the OpenCV model to ensure high accuracy in detection.

After hardware was implemented and application was validated, the back-end and front-end development of the software application was set in motion. To prevent the front-end team from waiting for completion of back-end, a hardware simulator was designed using python. The simulator was hosted on a fire-base server. This allowed both teams to work on the project at the same time with no dependencies on each other.

The back-end was first implemented and tested using existing data (not real-time data). This made it easier to debug and ensure accuracy is maximized. After designing and testing the system, real-time data was provided to the software application for processing.

The front-end utilized the artificial data provided by the fire-base server. The front-end was being developed as the back-end was being implemented.

7.4 Network Analysis

7.4.1 System Flow

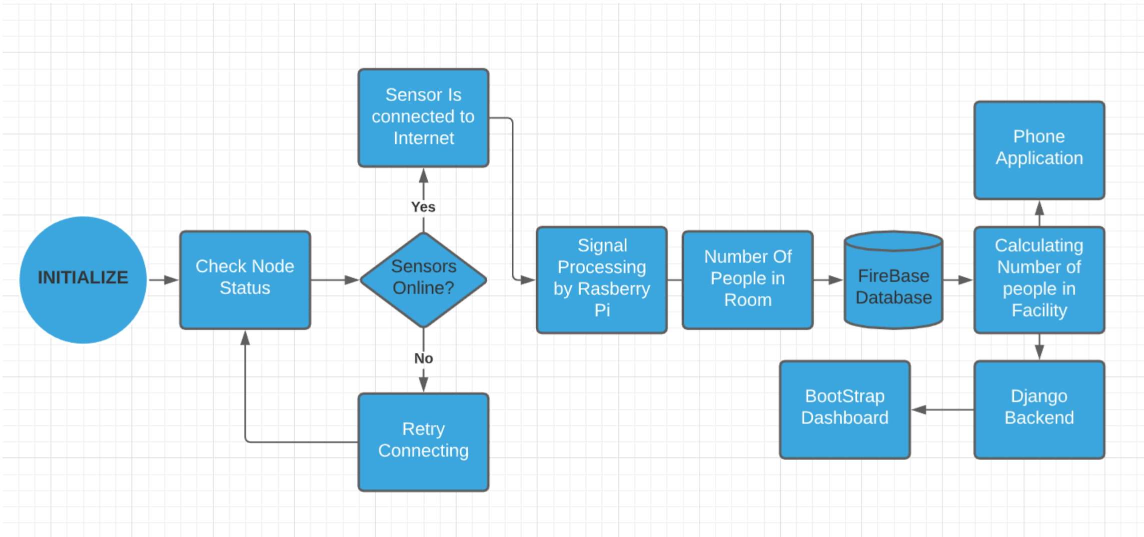


Figure 5: Network Flow Diagram

The network utilizes a star topology network where sensors send information to main base. The main base in this system is a Fire-base server. The sensor only sends the number of people in the building that passed through the system. This information is utilized by a second python script that collects all the information from the server and calculates the number of people in the facility depending on the organisation of the sensors. This information is later used to develop the dashboard on both the phone application and the web application.

7.4.2 Software application Flow

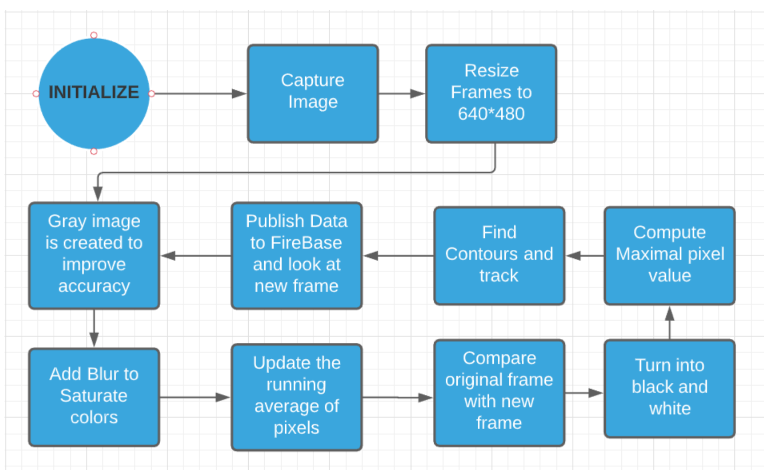


Figure 6: Software application Flow Diagram

The software application starts off by initializing the hardware connected to the Raspberry pi. The image is then captured using the Raspberry Pi camera and resized to a smaller resolution to make it easier on the model to process the data. A large frame is not needed because the quality of the image is not important. The image is used to create a gray scale image which improves the accuracy of the system. A blur is added to saturate the colors. The program then updates a running average and finds the difference in the original background and the current frame. The image is then turned into a black and white for easy and accurate detection. The script then compute the maximal pixel value overlapped by current frame and replace the image pixel in the anchor point position with that maximal value to find the contours. The contours are tracked along the x axis to determine whether the person is entering or leaving the facility. The information is later uploaded to the database and the system will loop forever to provide real-time data.

7.5 System Simulation

To run the simulation, a python script was used. The python script sent artificial data to the Fire-base server to test and design the front-end of the application. This allowed the team to work on different parts of the system without having to actually implement the hardware in real environments. The python code of the simulation can be found in the appendix section A.0.1. Figure 7 presents the simulation in action. 10 different Raspberry Pis are simulated to provide the three different variables, enter, exit, and inside.

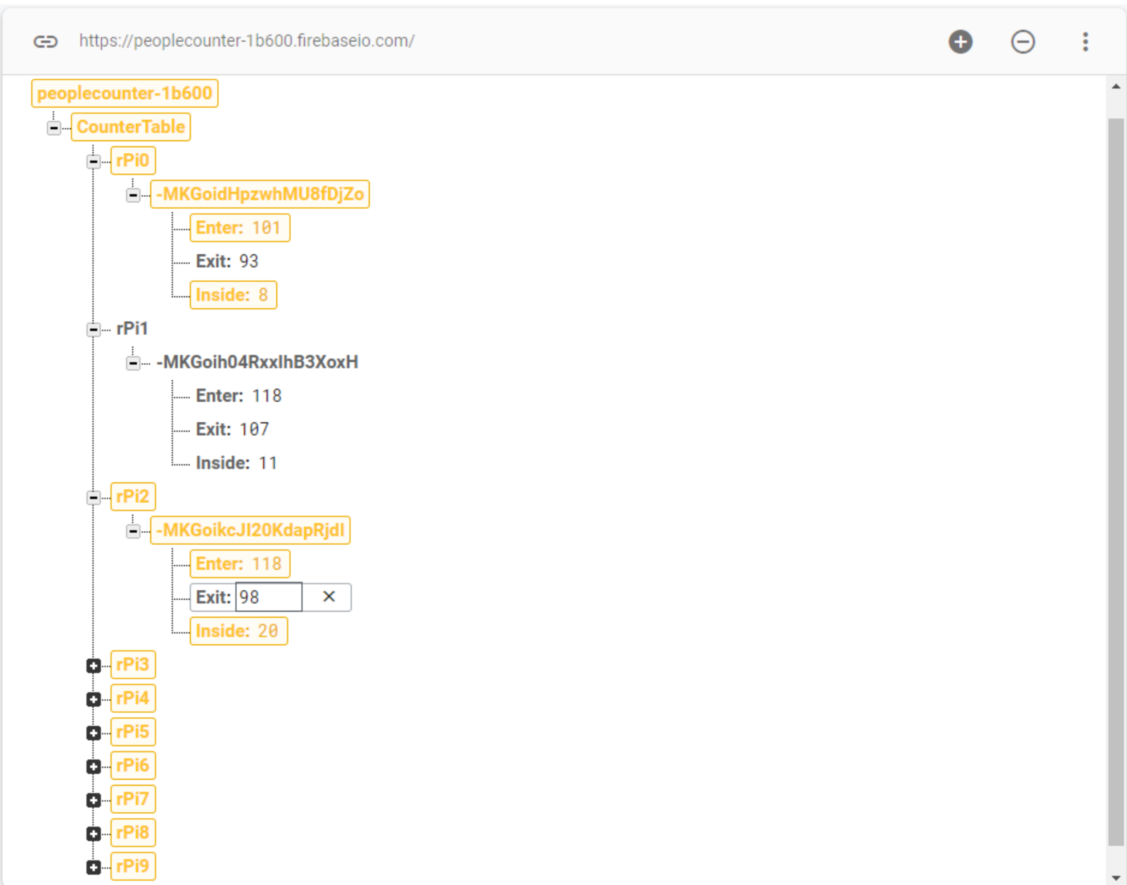


Figure 7: Simulation Results

7.6 Simulation Data

Figure 8 presents a processed image before dilation. In this figure, the beacon is placed on a house entrance and detects whether a person is leaving or entering the premises. Figure 9 represents the image vector after dilation where pixels have been converted to black and white. Open-CV does a great job in subtracting the subject from the background and detect the movement of the person. As the person moves across the x-axis, the model is able to tell whether the person is entering or leaving the frame.

An Online Video Demo was prepared in order to present the functionality of the hardware system, in a simple case in a typical application: tracking an individual walking into and out of a room.

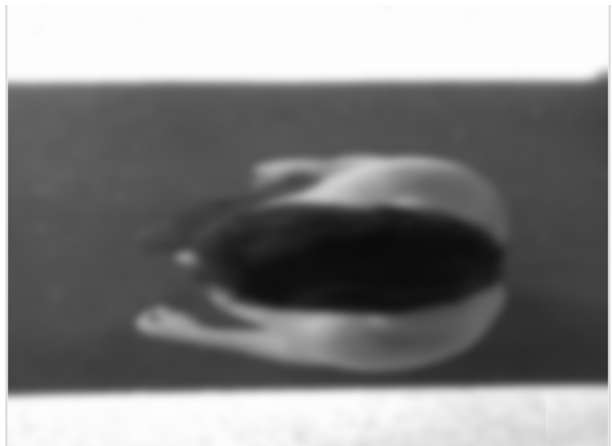


Figure 8: Image Result- GrayScale

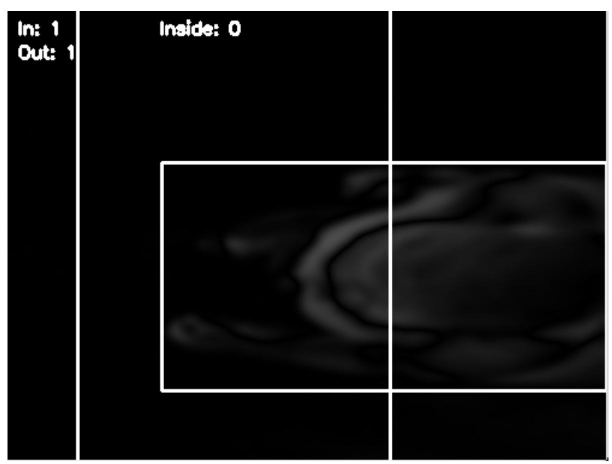
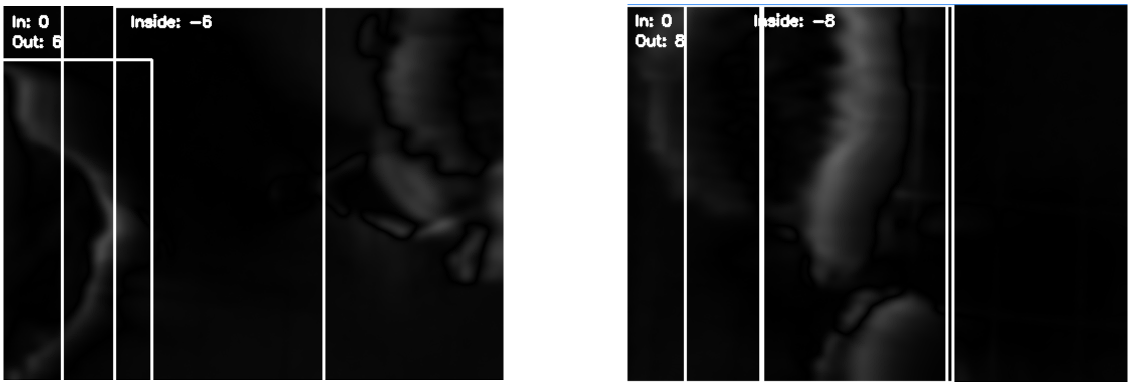


Figure 9: Image Result- DarkScale

In figure 10, a simulation with a cluster of people walking in at the same time was simulated. Simulation results can be improved with the use of a wide lens. Figure 10a refers to two different individuals walking out of the monitored location at the same time. Two contours are created and tracked by the software application. The out number increments from six to eight proving that the software application is accurate and functional. In figure 10b, Two people walk in side by side resulting in two different contours being created and tracked over each other. The out value is incremented from eight to ten people.



(a) Multiple People into Frame

(b) Multiple People Side By Side

Figure 10: Cluster of People Simulation

This simulation proves that the system is still accurate and functional when more than one person walks into a monitored facility at the same time. Due to Covid measures, a simulation with more than two people was not conducted. A higher roof was needed, but from experimenting with the system, the higher the roof or the higher the sensor is off the ground, the system is able to achieve a higher accuracy.

7.7 Designed Graphical User Interface (Mobile App)

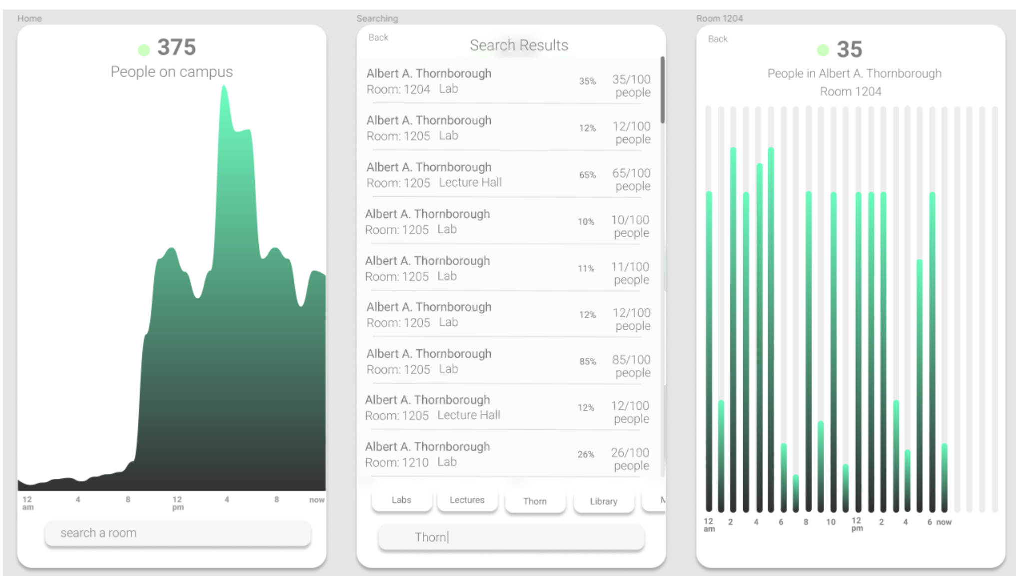


Figure 11: Interface Page 1

Figure 12: Interface Page 2

These screenshots were created using Figma, a software for app prototyping. These screenshots represent the front-end design of the software application. Figure 11 presents the number of people on campus which is represented using a graph. The user can utilize the search bar at the bottom of the screen to find a specific room at the university. The search bar provides the user with the occupancy percentage, and the number of people in the room. The user can access the entry history to gain a better understanding of how busy the room gets at different times.

Figure 12 presents the pinning feature. This feature allows the user to save their favorite locations. The location information is presented to the user as a shortcut when user logs into application. The data provided is real-time but user can still access historical data.

7.8 Designed Graphical User Interface (WebApp)

A dashboard is created as part of the design to make this application accessible to everyone. This dashboard is designed using React and Bootstrap. Figure 13 represents the initial stage dashboard designed. The dashboard will allow users to access room occupancy information of most rooms on campus. The dashboard will also provide occupancy information of services on campus such as gym facilities, the health clinic, food cafeteria, and more. The next step of the development process will include connecting the front-end to our Firebase back-end and different react components to interact with each other.

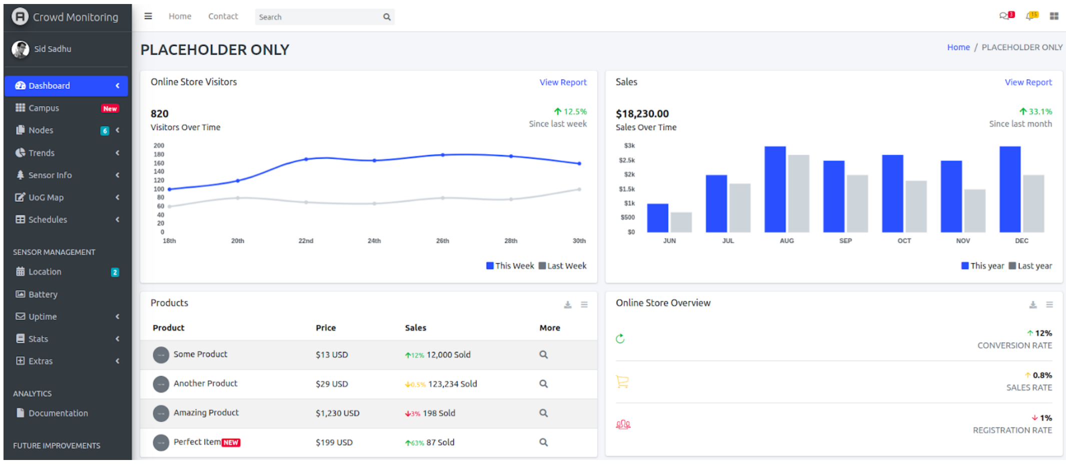


Figure 13: Front-end of the web application

7.9 Specifications

The table below shows a bill of materials including the hardware and software components of the project for a 3-year period. As discussed before, the back-end of the project which stores the data received through the sensors will potentially be all hosted on Google Cloud Platform. Services like Google Storage queue, Firebase, API Manager, Functions, IoT Core are all taken into account when calculating the cost per unit. The links to all the costs associated with GCP platform can be found in the appendix. Operation and Maintenance cost of 100 CAD is added for a 3-year period to take into account any unforeseen circumstances, and regular operations in order for the device to send accurate data to the servers.

Costing and Billing	
Raspberry Pi Model 3	50 CAD
Camera Module V2	30 CAD
Casing	10 CAD
GCP (Google Cloud Platform) Services	70 CAD
Operation and Maintenance	100 CAD
Total Cost per unit: 260 CAD	

Table 5: Effective cost per unit on a 3-year period

7.10 Suppliers

All the hardware for the project would be purchased through BuyaPi.ca, they are one of the largest Raspberry Pi distributors for North America. They take volume quotes, and their prices are very competitive. Also it's a much safer way to make transactions rather than dealing with the factories itself.

7.11 Discussion

The project is an important proof of concept application that serves as a modern tool to better manage study spaces, and prevent overcrowding on the campus. The final prototype includes two basic elements: Raspberry Pi and a 1080p camera module in a PVC casing. The hardware for the IoT device is very simple to assemble and can be operated on any door entrance of a shared space or labs. The scope of the project limits to Thornbrough building which would solve the hassle for engineering students to find better study spaces as well as being able to socially distance themselves properly. The software side of the project is in faily early stages and the dashboards shown above are demonstrations of how the websites/mobile applications would look like in the future implementations of the project. Currently the IoT device is able to send useful information in JSON data packets to Google Firebase. The hardware prototype as a whole is very sleek and minimal. Several constraints and criteria were set in the beginning of the project to make it successful. Cost, time, accuracy, physical; all these constraints were successfully met by the end of this project. The hardware cost was able to be kept under 100 CAD, the accuracy floated between 90-95 percent well over the 80 percent threshold as well as time and physical constraints of the project were successfully met. The software constraint of the project, however, still remains in very early stages, and the web application currently serves as placeholder elements which would require further development and would remain on the things to be done differently list. The project remains very accurate, user-friendly, efficient, reliable, maintainable, highly extensible and salable, and extremely modular. All criterion of the project were successfully met by the end of the semester.

8 Conclusions & Recommendations

8.1 Preliminary Conclusion

In conclusion, in order to assist students find study spaces in their university or college campus quickly and with accuracy, our solution proposes a system involving an IoT node, a raspberry pi camera, and a mobile application which work together to inform students about the number of empty spaces available in a room or on a floor in different buildings on campus.

8.2 Recommended Course of Action

In the original proposal, an RFID-based approach was discussed. However, after a design evaluation with alternatives, the final concept was selected to replace the RFID chip system with an image-based system using the raspberry pi camera. Alternatives to occupancy detection systems discussed include RFID chips paired with RFID readers, IR sensors, and raspberry pi cameras. The RFID chip-based system was eliminated due to issues with accuracy, information security and power consumption. The constraints we had defined for the project requires over 90% accuracy in occupancy detection. Additional constraints include budget and information security. As the RFID tags would be directly linked to each student through their student card, the storage and use of that personal information may lead to a breach in privacy and information security. Although IR sensors are far more accurate than a raspberry pi camera paired with OpenCV, IR systems can be very expensive and difficult to implement.

8.3 Revised Plan & Opportunity For Feedback

Overall, for demonstration purposes of the prototype, a raspberry pi camera will work well to detect occupancy when paired with OpenCV. In future iterations of the project, improvements can be made to the overall accuracy of the system by training the model further. This can be done on a continuous basis as new information and technology becomes available. Additionally, continuous improvement allows for client feedback to be implemented through each deployment and update. Clients may decide to opt for higher or lower funding options depending on the resources available to them and the functionality they require. A low-cost system which may be built upon to improve accuracy and functionality is ideal as clients may choose the level of complexity they require.

Other potential areas of improvement include using the building's own power source to power the monitoring devices in order to reduce maintenance costs over time. As well, implementing a proximity sensor in order to detect when someone is walking into the room and signalling for the camera to begin taking image captures may also prove valuable in preserving the system's battery life as well as cloud storage and computing resources, as when the system is inactive (i.e. the proximity sensor does not detect an individual walking in the room), the camera is in a dormant state and not capturing or storing any images.

References

[1] *E. C. Kan, D. Hysell, H. Park, R. Zadeh, and J. Mukhopadhyay, "Indoor Occupant Counting by Ambient RFID." Washington, 2020.*

[2] *T. Global, Ed., "Footfall Counter," Tagit, 2020. [Online]. Available: https://www.tagitglobal.com/industries/footfall-counter. [Accessed: 04-Dec-2020].*

[3] *G. C. Nan Li and B. Becerik-Gerber, "Measuring and monitoring occupancy with an RFID based system for demand-driven HVAC operations," Elsevier, Mar. 2012.*

[4] Aruquia Peixoto, Manuel Castro, Manuel Blazquez, Sergio Martin, Elio Sancristobal, German Carro, Pedro Plaza: *"Wireless Network tips and tricks for inclusion and integration of students"* Global Engineering Education Conference (EDUCON) 2018 IEEE, pp. 2037-2041, 2018.

[5] Sylvia T. Kouyoumdjieva, Peter Danielis, Gunnar Karlsson: *"Survey of Non-Image-Based Approaches for Counting People"* Communications Surveys & Tutorials IEEE, vol. 22, no. 2, pp. 1305-1336, 2020.

[6] ens Weppner, Paul Lukowicz, Ulf Blanke, Gerhard Tröster: *"Participatory Bluetooth Scans Serving as Urban Crowd Probes"* Sensors Journal IEEE, vol. 14, no. 12, pp. 4196-4206, 2014.

[7] Sheppard AR, Swicord ML, Balzano Q: *"Quantitative evaluations of mechanisms of radiofrequency interactions with biological molecules and processes."* Health Phys. 2008;95(4):365–96

[8] X. Liu, P.H. Tu, J. Rittscher, A. Perera, N. Krahnstoever *"Detecting and Counting People in Surveillance Applications"* IEEE Conference on Advanced Video and Signal Based Surveillance (2005), pp. 306-311

[9] V. Prabakaran, A.M. Arthanariee, M. Sivakumar *"Crowd Safety: A Real Time System For Counting People"* International Journal Of Innovative Technology Creative Engineering, Volume 1 (Issue. 1) (2011), pp. 6-11

[10] C. Raines, L. Williamson *"Using visual aids: The effective use of type, color, and graphics (Rev. ed.)"* Lanham, MD: National Book Network (1995)

[11] A. Stoytchev *"Face Detection Code. Retrieved April 11,2012, from (2008) http://home.engineering.iastate.edu/alexs/classes/*$2009_{S}pring_{5}75X/code/11_{F}aceDetection/$."

[12] D.Y. Chen, K.Y. Lin *A Novel Viewer Counter for Digital Billboards* Proceedings of the 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 09) (2009), pp. 653-656

[13] A. Kozłowski, A. Królak *Teaching image processing and pattern recognition with the Intel OpenCV library* Proceedings of SPIE, Volume 7502, (750205) (2009)

[14] X. Zhao, E. Delleandrea, L. Chen *A People Counting System Based on Face Detection and Tracking in a Video* Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance (2009), pp. 67-72

Appendix A Appendix

A.1 Python Program

A.1.1 Python code for the simulator

```
from firebase import firebase
from threading import Thread
import time
import random

def setup_table(firebase,NumOfTables):
    tableNames = []
    RasPIdata = {
        'Enter':0,
        'Exit':0,
        'Inside':0
    }

    for tableNum in range(0,NumOfTables):
        data = firebase.post('CounterTable/rPi'+str(tableNum), RasPIdata)
        tableName = data["name"].split("u\'-")
        tableNames.append(tableName[0])

    return(tableNames)

def run_sim(firebase,tableName,tableNumber):
    RasPIdata = {
        'Enter':0,
        'Exit':0,
        'Inside':0
    }

    #Setting up Rasberry Pi DataSet
    data = firebase.get('CounterTable/rPi'+ str(tableNumber) +'/', '')
    #Initialization
    if(data == None):
        #data = firebase.post('CounterTable/rPi', RasPIdata)
        print("table not Found. Make Sure that table is created!")
        exit
    else:
        RasPIdata.update({"Enter":data.get(tableName).get("Enter")})
        RasPIdata.update({"Exit":data.get(tableName).get("Exit")})
        RasPIdata.update({"Inside":data.get(tableName).get("Inside")})

    #Run Forever until Program is shut
    while(1):
        time.sleep(1) #Wait 5 Seconds
        enterValue = random.randint(0, 1)
        exitValue = random.randint(0, 1)
        RasPIdata.update({"Enter":RasPIdata.get("Enter") + enterValue})
        firebase.put('CounterTable/rPi'+ str(tableNumber) +'/'+tableName,'Enter',RasPIdata.get("Enter")
```

```python
        if(RasPIdata.get("Enter") <= RasPIdata.get("Exit")):
            continue
        else:
            RasPIdata.update({"Exit":RasPIdata.get("Exit") + exitValue})
            firebase.put('CounterTable/rPi'+ str(tableNumber) +'/'+tableName,'Exit',RasPIdata.get("Exit

        RasPIdata.update({"Inside":RasPIdata.get("Enter") - RasPIdata.get("Exit") })
        firebase.put('CounterTable/rPi'+ str(tableNumber) +'/'+tableName,'Inside',RasPIdata.get("Inside


    # # print(data.get("-MFm4FHmlgBwZMN3qsbI").get("Inside"))
    # print(RasPIdata)
    # print(data)

def main():
    firebaseLink = firebase.FirebaseApplication("https://peoplecounter-1b600.firebaseio.com/",None)
    TablesToCreate = input("How Many Rasberry Pis would you like to Simulate: ")
    tableNames = setup_table(firebaseLink,TablesToCreate)

    print("Starting Simulation!")
    for i in range(0,TablesToCreate):
        Thread(target=run_sim,args=(firebaseLink,tableNames[i],i,)).start()


if __name__ == '__main__':
    main()
```

A.1.2 Python script for processing

```python
import numpy as np
import time
import imutils
import cv2

avg = None    #Will be changed when Camera is implemented
video = cv2.VideoCapture("input.mp4")

xvalues = list() #Will be watching changes across the X direction
motion = list() #Store the coordinates of the motion in Video

inCount = 0
outCount = 0

#Depending on how far your camera is from the door, this value will affect results
#Adjust depending on high of camera. The higher the smaller the number
maxAreaThres = 8000

#Function to find max area
def find_majority(motionInput):
    Map = {}
    maxValue = ( '', 0 ) # (occurring element, occurrences)
```

```
    for n in motionInput:
        if n in Map: Map[n] += 1
        else: Map[n] = 1

        # Keep track of maxValue on the go
        if Map[n] > maxValue[1]: maxValue = (n,Map[n])
    return maxValue


#Run a loop forever
while 1:
    ret, frame = video.read()
    flag = True
    text = ""

    frame = imutils.resize(frame, width=500) # Resizing Video Here
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # Creating a grey image to improve accuracy
    gray = cv2.GaussianBlur(gray, (21, 21), 0) #Adding blur to saturate colors

    # no input inserted. Next step would be to take this and allow user to input Video
    if avg is None:
        print "[INFO] starting background model..."
        avg = gray.copy().astype("float")
        continue
    #IMAGE PROCESSING

    #Updates a running average
    cv2.accumulateWeighted(gray, avg, 0.5)
    #Find the difference in original background and current frame
    frameDelta = cv2.absdiff(gray, cv2.convertScaleAbs(avg))
    #Tutn into black and white for easy and accurate detection
    thresh = cv2.threshold(frameDelta, 5, 255, cv2.THRESH_BINARY)[1]
    #compute the maximal pixel value overlapped by current frame and replace the image pixel in the anc
    thresh = cv2.dilate(thresh, None, iterations=2)
    #Finding Contours/objects
    (_, cnts, _) = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)


    for c in cnts:
        #to improve accuracy change the maxAreaThres
        if cv2.contourArea(c) < maxAreaThres:
            continue
        #Calculates all of the moments up to the third order of the rasterized shape.
        (x, y, w, h) = cv2.boundingRect(c)
        #appending the x direction
        xvalues.append(x)
            #Display a box around the object
        cv2.rectangle(frameDelta, (x, y), (x + w, y + h), (255, 255, 255), 2)
        #Initialize flag as false
        flag = False
```

```
    no_x = len(xvalues)

    if (no_x > 2):
        difference = xvalues[no_x - 1] - xvalues[no_x - 2]
        if(difference > 0):
            motion.append(1)
        else:
            motion.append(0)

    if flag is True:
        if (no_x > 5):
            val, times = find_majority(motion)
            if val == 1 and times >= 15:
                inCount += 1
            else:
                outCount += 1

        xvalues = list()
        motion = list()


    #Displaying information on frame.
    cv2.line(frameDelta, (60,0), (60,480), (255,255,255), 2)
    cv2.line(frameDelta, (320,0), (320,480), (255,255,255), 2)

    cv2.putText(frameDelta, "In: {}".format(inCount), (10, 20), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 25
    cv2.putText(frameDelta, "Out: {}".format(outCount), (10, 40), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 2
    cv2.putText(frameDelta, " Inside: {}".format(inCount-outCount), (120, 20), cv2.FONT_HERSHEY_SIMPLEX


    #cv2.imshow("Frame",frame) #Main frame Ignore for now
    cv2.imshow("Gray",gray)
    cv2.imshow("FrameDelta",frameDelta)

    #exiting Program
    key = cv2.waitKey(1) & 0xFF
    if key == ord('q'):
        break

video.release()
cv2.destroyAllWindows()
```

A.2 Project Poster

IoT Occupancy Monitoring for Social Distancing Applications

Safwan Hossain · Bilal Ayyache · Siddharth Sadhu · Meera Divakaran · Dima Ismail

Scan the QR-code below to see system backend in action!

Background

A major concern for academic institutions in recent times has been providing sufficient and safe resources to their students in order to ensure quality education. Resources are limited and require efficient allocation in order to meet students demands.

Keeping students informed about the status of these resources in real time is the challenge that our project poses to solve, with the assistance of image recognition and real-time streaming of data to people.

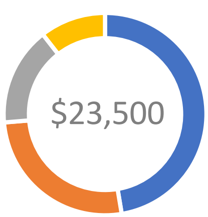
Results

The app can detect and display real time room occupancy to an accuracy of 85% , allowing students to quickly find available study spaces while maintaining social distancing and adhering to building and occupancy rules.

The device transmits real time motion capture and heat map data to the back end (hosted by google cloud) while maintaining an overall cost of less than $100 per unit. The cost is projected to further reduce when purchasing equipment and materials in bulk.

Cost

Total capital cost of the project is $23,500, which includes a 10% contingency, detailed design and implementation of 100 IoT devices covering every shared space in the Albert A. Thornborough building, data storage and hosting of the web application on Google Cloud Platform (based on a 3-year period), and 3 years of operations and maintenance for the physical devices.



Conclusion

This design serves as an important first step towards a more efficient and modern method of allocating shared study spaces uniformly throughout the campus

Design Solution

The design solution utilizes an image processing/open-CV model that predicts whether an individual is walking into or out of a video frame, captured using a raspberry pi camera.

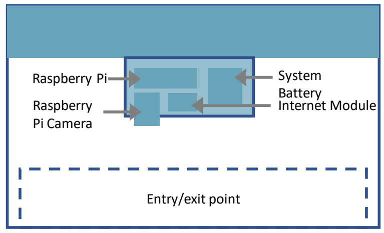
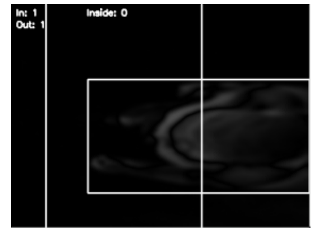


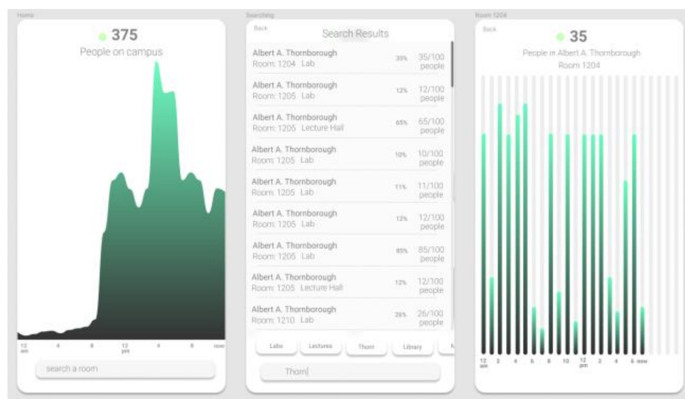
Diagram of ceiling mounted IoT device with raspberry pi camera



Camera output depicting bird's eye view of individual walking below

The raspberry pi captures real time data and sends information to a firebase database where data from all other nodes are collected. This provides accurate information of how many people are in the monitored facility. The designed dashboard utilizes this information to provide useful insight such as occupancy in lab rooms and study rooms.

The designed image processing model saves the original background of the image and compares it to every new frame captured to detect changes in image. If change occurs, the model tracks the changed pixels as the object moves across the frame. The direction of movement determines whether a person is entering or leaving the facility.



Screen capture of the designed dashboard depicting output results of the Albert A. Thornborough building.

The image captured is a filtered image which prevents privacy policy breach and ensures that the subject being tracked is unidentifiable.

SCAN ME

Live Demo

Scan using Google Lens app

A.3 Original Letter of Request

A.3.1 Letter of Transmittal

July 5, 2020

School of Engineering
University of Guelph
50 Stone Rd E
Guelph, ON N1G 2W1

Dear Dr. Spachos:

The University of Guelph's student population growth rate far exceeds the available campus facility's growth rate. However, the City of Guelph's bylaws for the city's proposed growth and development plan makes it very difficult for the University to expedite campus expansion. As a result, the University of Guelph is currently facing a space issue, in which there are not enough physical resources available on campus to comfortably accommodate the growing campus population in the Fall and Winter semesters. This phenomenon is only increasing with every year as the University of Guelph's student population increases and the demand for space becomes more urgent. To make better use of existing university facilities and resources, this project proposes to track the student populations' movement and studying habits across campus throughout all semesters, and use these findings to generate a report for the university's own records. As well, the real-time de-identified student movement data will be made available to students so that they may make educated decisions as to where they may choose to study.

In addition, this solution will be implemented as a contact-less system, which stands to have societal, public health and safety benefits by improving on the allocation of the infrastructure and resources available to students. The scope of the project encompasses the deployment of an IoT device used for collection of data and transmission to a remote server,which can then be accessed by students via a mobile application allowing them to avoid crowded places on campus.

Please find attached project proposal and provide feedback as you deem fit.

Regards,

Group 1: Meera, Safwan, Dima, Sid, and Bilal.

A.4 Project Management

A.4.1 Trello Project Manager

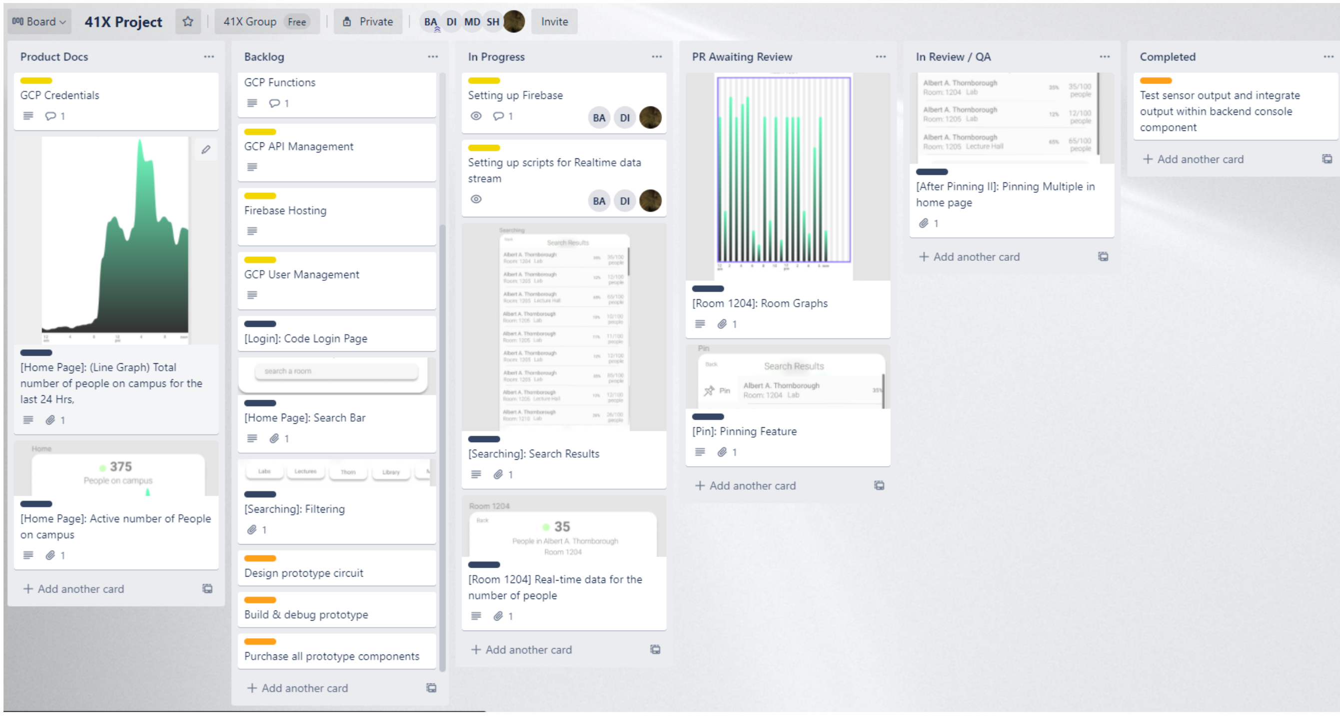


Figure 14: Trello Project Management Tool