**STUDY SPACE AVAILABILITY   
  
Report 4**

Group 20

**Mohammed Maaz Ahmed 100522349**

**Carlos Fabregas 100572307**

**Angelie Kennedy 100506702**

**Anjali Shah 100555990**

**Table of Contents:**

1. **Executive Summary…………………………………………..…………....4**
2. **Problem Identification……………………………………………………..5**
3. **Project-related Background and Research Review…………………....…6**
4. **Design Process…………………………………………………………....…7**
   1. Software architecture
   2. Degree to which work products are identified and required
   3. Project Tracking and Team Role
   4. Customer and Stakeholder Involvement in the Project
5. **Scenarios and Use Cases…………………………………………….…....10**
6. **Stakeholder Requirements and Traceability Matrix…………………...16**

6.1 Stakeholder Requirements

6.2 Engineering Requirements

6.3 Marketing Requirements

6.4 Test Cases (Use Cases)

6.5 Traceability Matrix

1. **Definition of acceptance Test……………………………………………..19**
2. **Project Plan…………………………………………………..……………21**
3. **Concept Generation and Analysis………………………………………..22**
4. **Conceptual Design……………………………………………………..….24**
5. **Definition of Integration Tests………………………………………...…27**
6. **Estimated Cost of the Project……………………………………………28**
7. **Updated Project Plan………………………………………………….....30**
8. **Detail Design……………………………………………………………...32**
   1. Architecture
   2. User App
   3. Admin App
   4. Frameworks
9. **Integration and/or Unit Testing…………………………………………38**
   1. Web Login Test Results
10. **Updated Project Plan…………………………………………………….39**
11. **Acceptability Planning/Test……………………………………………...41**
12. **Test Results………………………………………………………………..44**
13. **Ethical Considerations…………………………………………………...47**
14. **Safety Considerations …………………………………………………..47**
15. **Conclusion………………………………………………………………..48**
16. **Acknowledgements…………………………………………………….49**
17. **References………………………………………………………………..50**
18. **Appendix……………………………………………………………..…..51**

**List of Figures**

Figure 4.1: Diagram for the interaction between sensor, controller and database………6

Figure 4.2: Diagram for the interaction between the User, the User interface…………...6

Figure 4.3 Project Tracking and Team Roles: Team Leader: Carlos……...……………...7

Figure 5.1: Use Case…….……………………………………………………………………10

Figure 10.1: Architecture Model of the Design…..………………………………………....26

Figure 14.1: Overall Architecture Diagram………………………………………………….32

Figure 14.2: Item Detection App……………………………………………………………..33

Figure 14.3: displays a sequence diagram on how authentication is handled……………………………………………………………………………….…………34

Figure 14.4: Users App…………………………………………………………………….....35

Figure 14.5: Users App - sensors…………………………………………………………....35

Figure 14.6: Admin App……………………………………………………………………….36

**1. Executive Summary**

The capstone project, Study Space Availability, focused on solving the issue where students waste time trying to find available seats around the library. This issue is occuring due to students leaving belongings on the table while it is not occupied and limited space inside the library for the amount of students attending the university. While working on the capstone project the approach to solving the problem in the library was changed multiple times. The first option that was considered was using a PIR sensor to detect if a person is sitting at the table. After doing some testing using this sensor, we found deficiencies such as a lack of movement resulting in false positives of an empty seat. Researching other options the team decided on working with only a camera as the best option. The camera was attached to a Raspberry Pi which allowed us to capture pictures every minute of the tables status. The table status included the table having a person at it, only an object at the table, a person and object at the table and lastly, there is no person or object at the table.

The initial step to the approach was to attach the Raspberry Pi camera to the table at an area which would only capture the objects on the table and not the persons face. The team also had to consider privacy issues involving the recognition of person’s identity. The camera was placed in an area of the table where it would be out of the way for students but also capture the data accurately. The images are then processed to detect whether there is an object or a person at the table. These images are taken once every minute. Then the web page will be updated with this information to indicate whether the table is available or not.

In order for users to view the libraries availability of current seats, they are required to log in using their UOIT email. This gives the students and library staff a layer of privacy that will keep the libraries environment safe from intruders who wish to use the information for unethical acts.

**2. Problem Identification:**

As UOIT is a rapidly growing campus, it comes with its growing pains. As a group of students looking to give back to the campus community, the library has been identified as a place that could benefit from an IoT system. As the library is the largest and most notable study area on campus, with around 350 tablespaces available to students to use at a first come first serve basis. One of the identified problems at the library is students spending a significant amount of time looking for an available seats when they could use that time to study, this particularly frustrates students during exam seasons. Further investigating this issue one of the key reasons students are not able to find a seat is because other students leave their belongings for excessive amounts of time on the tables or chairs to reserve the spot and leave the library to do other activities. Currently the library has a policy that you cannot leave your belongings unattended for more than 30 minutes. Once this 30 minutes is over and the student has not returned the library security is allowed to remove the students’ belongings from the table. An additional problem that occurs with this is the library security is currently walking around multiple times on each floor of the library noting down which tables have belongings on them but no student and come around for a second check after half an hour before any action can be taken. This is a hassle for security since they have to continuously walk around and leads to inefficient information.

The goal for this capstone project is to focus on a solution to aid the students in finding seating quickly but also allow the library to track which tables have not been occupied for a certain amount of time without having to constantly check all the tables. While looking further into this problem an issue that came up was that there around 4 different styles of tables in the library. Due to the limited time given for the project it was necessary to narrow down our scope to a certain portion of the library that would prove useful and effective in its current state. The project also has a large focus on scalability allowing for future expansion and enhancement of the project to other areas of the library.

A problem that can come up is that the overall project needs to have the ability to be implemented to the UOIT website in the future. Since the focus of the project is to work on a certain type of table in the library the problem that could come up is that can these sensors be implemented to any type of table in the library and work efficiently. Lastly, the cost of the materials required to implement the project to the entire library in the future and any maintenance such as replacing batteries if they are used could be an issue.

**3. Project-related Background and Research Review:**

When researching ideas for how this project could be done the idea of using different types of sensors was an option. A human sensor and weight sensor can be used to detect human presence at tables and any objects left on table. A similar project found was the UCL (University College London) study space availability. UCL had placed a small device under tables in study spaces which used infrared technology to detect if the table is available. The information received from the device is then displayed on the UCL library website where students are able to see how many tables are available in each of the libraries on campus. For the capstone project the idea is very similar to the UCL library space availability but the capstone project will give students an exact location of which table is empty rather than telling the students how many spaces are available. The project will also give library staff access to see how long tables have not been occupied for even though belongings are left on the tables.

Another similar project done and being used currently that was found was the parking lot occupancy system. This system has sensors on the parking spaces which allows users to know which parking spots are taken. A LED is also placed at each parking spot which turns green for open spaces. The system counts the number of vehicles coming and going and keeps track of how many spots are still available while the advanced part of the system will monitor the available spaces on each level of a parking garage.

Other researches found were done by students at other universities which dealt with putting sensors on the seats rather than the tables. This was an option for this project but the issue of the chairs being moved to other tables was a problem. A tracking device would be necessary for the chairs to make sure each chair was with its corresponding table. Putting the sensors at the tables is more reasonable because the tables cannot be moved.

After deciding what the plan for the project was the next step was to research what materials would be needed. When doing so many sensors that could be a possible fit were found. A final decision of using a human sensor and a weight sensor was made. These sensors will be connected to a Microcontroller, and likely an Arduino board. The human sensor chosen is called Aukru 3x HC-SR501 Human Sensor Module Pyroelectric Infrared PIR Motion Sensor Detector.

**4. Design Process**

4.1 Software Architecture

The design architecture used for the application will be the Model-View-Controller (MVC). We have three different controllers which are the Sensors(View), Arduino WIFI (Controller), MySQL (Model). This will make the application robust as making changes to each component separately will make errors easier to diagnose.

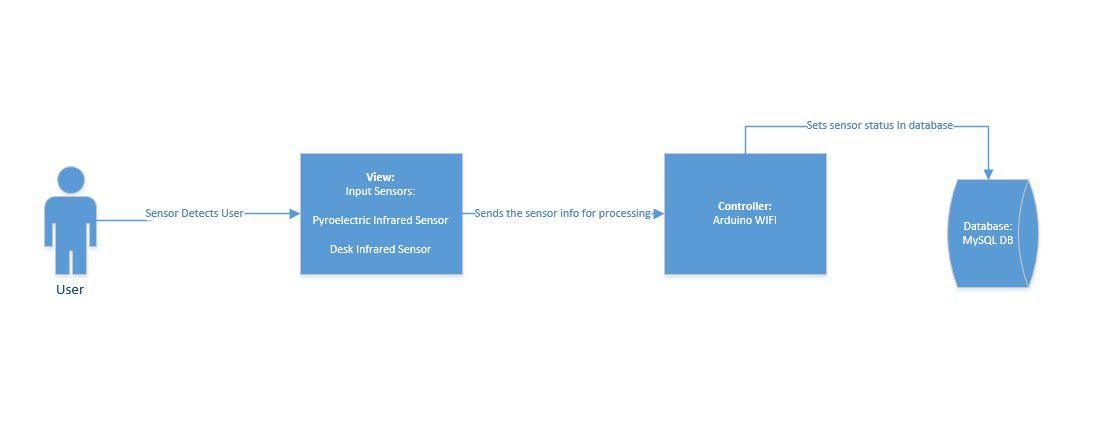


Figure 4.1: Diagram for the interaction between sensor, controller and database.

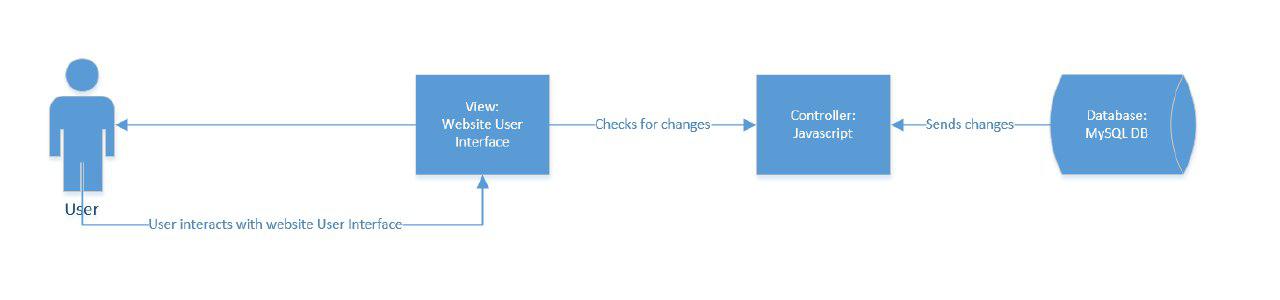


Figure 4.2: Diagram for the interaction between the User, the User interface

# 4.2 Degree to which work products are identified and required: *Software Requirements:*

Web application must display seating arrangement in library, and show students which tables are in use, including if unattended belongings are left at table, and which are available meaning no person or belongings at table.

Web application must also show security which tables have unattended belongings and show how long those belongings have been left unattended.

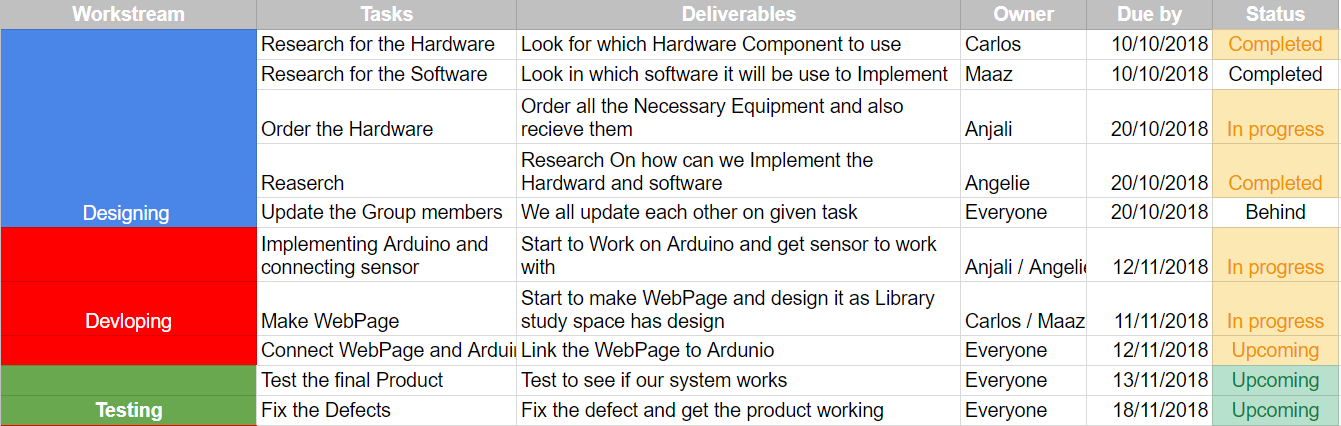
# *Hardware Requirements:*

Infrared camera for sensing individuals sitting at a table.

Weight sensor to detect if objects are placed on the table

Microcontroller will need to take readings from weight sensor if no one is at the table. If infrared sensor is reading no one at the table, and if weight sensor is detecting objects on table, Microcontroller must allow 30 minutes before sending notification to security

4.3 Project Tracking and Team Roles: Team Leader: Carlos



4.4 Customer and Stakeholder Involvement in the project

The main outcome of the project is to develop a System that shows if the Study tables are empty or not in the library .The targeted audience are the students who will be using the system to find study spaces. This means that the students would be the primary stakeholder of the system. The secondary stakeholder of the system would be the UOIT library because the library will benefit from the system. The system will simplify the tasks of the security in the library.

**Stakeholders**

*Primary Stakeholder*

· Users of Study space (Student)

o The student will be able to access the system to see which spaces are available for them to study.

· Developers

o The Developers can further improve the system or also add the functionalities to the make system better and more useful for the student

*Secondary Stakeholder*

· UOIT Library

o UOIT library will be benefit from this system because the system tell the security how long has the person been away from the desk since the time limit for students will be just 30 min. So the security doesn’t have to keep waking around noting down the number

**Customer**

1. Students will need a system that is simple to use

2. Students will require the system to constantly update which seats are available by using a colour system. Green will be available and red is unavailable.

3. There needs to be layout of library on the system so that students can find the available seats easily.

**5. Scenarios and/or Use Cases:**

**Actors**

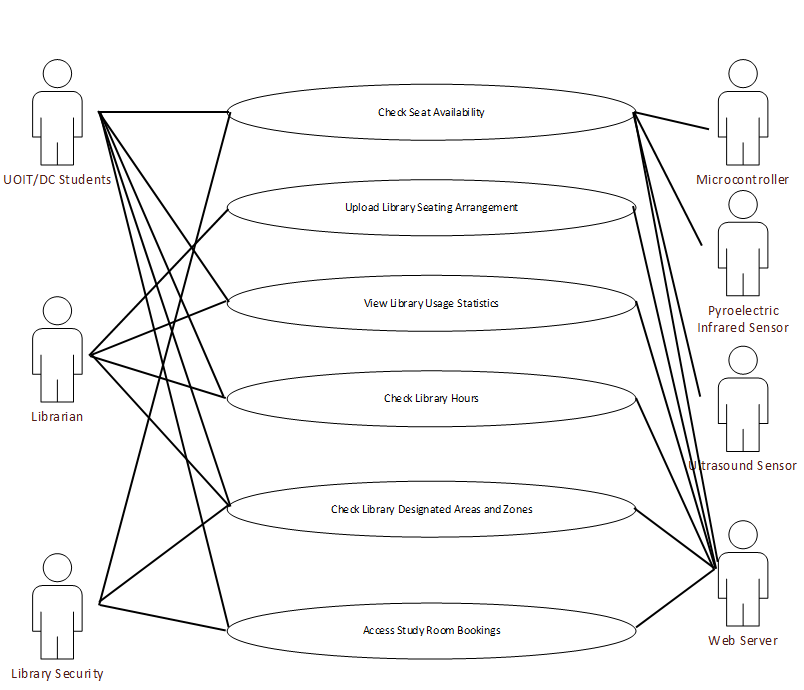
UOIT/DC Students - Students of the shared campus/ web users

Library Security - Security of the library

Librarian - The library’s administrator and assistant

Pyroelectric Infrared Sensor - Used to detect students at tables

Ultrasound Sensors - Used to detect items on table surface



|  |
| --- |
| Figure 5.1: Use Case |

**5.1 Use Case 1: Check Seat Availability**

**Actors:** UOIT/DC Students, Microcontroller, Web Server, Pyroelectric Infrared Sensor, Ultrasound Sensor

**Stakeholder and needs:** To check if there are available seats in the library where students can sit down and study.

Beyond this, the system must not record any personal data of students sitting on tables. (No pictures preferably, and at the least no saving of pictures or facial recognition). Preferably the system should also inform security of table being empty for 30 minutes, although this is not necessary.

**Preconditions:** Sensors have to be fully functioning and uploading accurate data to the Microcontroller. Microcontroller needs to maintain stable wifi connection to constantly upload data. Server must remain running to allow users to access data.

**Postconditions:** Library Space Availability is uploaded to server and User can check to see how many seats are available

**Trigger:** Microcontroller communicates with sensors and web server to update the mode

**Basic Flow:**

1. Pyroelectric Infrared Sensor updates reading to Microcontroller.

2. Microcontroller determines based on Pyroelectric Infrared sensor’s reading that no Student is present at table.

1. Student is present at table

3. Ultrasound Sensor updates reading to Microcontroller.

4. Microcontroller determines based on sensor’s reading that tabletop is empty and ready for use.

1. Table has belongings left on it

5. Microcontroller updates web server that the study space is available for usage.

6. Web server displays table availability based on information received from Microcontroller (If table has belongings, it displays as taken, and if student present it displays as taken, otherwise space is available for use.)

**Alternative**

A1. Student is present at table.

1. Microcontroller updates web server that the seat is taken
2. Return to basic flow step 6.

B1. Student belongings are left on table.

1. Microcontroller updates web server that the seat is taken
2. Microcontroller Times how long table remains in this state (No Student present, belongings on table).
3. If Time exceeds 30 minutes send update to web server
4. Web Server updates security that table has had un attended for 30 minutes with belongings on it.
5. Return to basic flow step 6.

**5.2 Use Case 2: Upload Library Seating Arrangement**

**Actors:** Librarian, Web Server

**Stakeholder and needs:** Librarian should be able to upload seating arrangement data.

**Preconditions:** Library must have some form of identification for tables to connect microcontrollers on each table with the system. This will allow for the librarian to place tables by there unique identification in different categorized areas.

**Postconditions:** System (Web server) now knows which seats (microcontrollers) are in which section and can categorize the information accordingly.

**Trigger:** Librarian logs into System and accesses ‘update library layout’

**Basic Flow:**

1. Librarian accesses ‘update library layout’

2. Librarian can move tables by there unique id’s to different sections of the library.

3. Librarian also has option to upload a new schematic of the different library sections to reflect the current state of the library.

**5.3 Use Case 3: View Library Usage Statistics**

**Actors:** UOIT/DC Students, Web Server, Librarian

**Stakeholder and needs:** The system should record long term data over the years to show usage of different areas. This could help the library in better understanding what zones (green, yellow, and red zones) are in highest demand and how occupied they are on average. This will help in long term planning for libraries expansions or redesigning.

**Preconditions:** All data needs to be safely stored and organized to allow these statistics to be made.

**Postconditions:** Library will be able to view ratios of occupied to available seating for each designated area (hence the usage of different parts of library

**Trigger:** Statistics will be generated on a monthly and yearly basis.

**Basic Flow:**

1. Data from seat availability which is displayed on website will also be stored.

2. Stored data will then be analysed into usage of different areas by comparing number of available seats in each area to number of used seats throughout the open hours of the library.

**5.4 Use Case 4: Check Library Hours**

**Actors:** UOIT/DC Students, Web Server, Librarian

**Stakeholder and needs:** Students need to be aware of library hours to help them know when extra hours are being offered around exam seasons etcetera. It would help increase usage to capacity if people are aware of hours ahead of time and can plan out there day accordingly

**Preconditions:** This information would need to be provided by UOIT library.

**Postconditions:** Library hours of operation will be displayed on the website so students can quickly find hours while checking library status.

**Basic Flow:**

1. Data of library hours will be retrieved.

2. Data will be displayed on website to users.

**5.5 Use Case 5: Check Library Designated Areas and Zones**

**Actors:** UOIT/DC Students, Web Server, Librarian, Library Security

**Stakeholder and needs:** Students need to be aware of different designations and rules for different parts of the library to help enforce a respectable environment in each of the different zones, and also to help students realise these zones exist to help there studying needs. Librarian would need to update these zones and possibly change certain zones if the usage of certain zones is seen to have higher demand than others. Security would also need to be able to access this information to help them keep up to date with any updates to zones to help enforce the different zones according to the noise level restrictions.

**Preconditions:** This information would need to be provided by Librarian.

**Postconditions:** Library zones will be displayed on a map of the library to help students understand which areas are meant for different types of studying, and what level of noise is permitted in each area.

**Basic Flow:**

1. Data of library zones will be retrieved.

2. Data will be displayed on library map to users.

**5.6 Use Case 6: Access Library Room Booking Services**

**Actors:** UOIT/DC Students, Web Server, Library Security

**Stakeholder and needs:** Students should be able to book (the bookable) library rooms from this site to help keep everything in one place and allow easy access to library facilities. Security also needs access to this information to help them monitor and track usage of bookable study spaces in library.

**Preconditions:** This information would need to access the current library site for room bookings. (Possibly merge the sites in future)

**Postconditions:** Library booking should be accessible from a link on this website to help make usage simple and keep all library features accessible from one place.

**Basic Flow:**

1. Student clicks on room booking option.

2. Student is redirected to library bookings site.

**Scenario 1:**

1. Student is finishing up class and planning out his day.
2. Student has to workout, eat lunch and study. Checks the website to check if library has available seats.
3. Library has available space, student plans to study after class.

**Alternative**

A1: Library has no available space

1. Student decides to eat lunch and check website again later to find seat.

**Scenario 2:**

Student plans out the day, checks the app to see when the library peak hours are. User adds studying time

Students in class

User wants to know when seat is available for him to study

User wants to check library peek hours

**Scenario 3:**

User is at the library, cannot find a seat in one of the sections of the library. Instead of wasting his time walking around the library looking for a seat he checks the website to see where exactly an available seat is located.

**Scenario 4:**

The library is about to close and would like to know an estimated capacity of the library prior to closing.

**Scenario 5:**

The desk sensor loses connection with the server. The server is updated with this information and will change the availability on set desk to information currently not available. The librarian/security will get a notification that a sensor has currently failed. This will allow them to start troubleshooting the sensor currently at the desk.

**Scenario 6:**

Student has a long commute to school to attend class. The student arrives to find out that his class was canceled. The student is now looking for a place to sit in campus but does not want to waste time walking around. Student accesses the UOIT website to find a seat in school to spend a few hours studying before he decides to commute back home.

**Scenario 7:**

A student has a quiz due in a few hours.They are currently at the fishbowls but they are finding that students are talking loud so they are finding it hard to concentrate. Instead of wasting time walking around school risking wasting time the students decides to go on the library website to check if there are seats available in the quiet area. If the area is currently not available the student has the option to receive a notification once the seat has become available.

**6. Stakeholder Requirements and Traceability Matrix**

**6.1 Stakeholder Requirement**

**Stakeholder Requirements:**

S.1: System Displays seat availability and can be accessed by students

S.2: System should refresh every 15 seconds

S.3: System should be accessible from UOIT website, mobile devices, etc

S.4: System should track and update to website which tables have unattended belongings

S.5: System should track amount of time belongings have been left on table

S.6: System should inform security when belongings have been left at unoccupied table for more than 30 minutes so that it can be removed

S.7: System should uphold standards of privacy and not be able to identify individuals. Data kept on secure server and database

S.8: Access must be limited to only UOIT/Durham students

S.9: System should be at a reasonable cost

**6.2 Engineering Requirements:**

E.1: System should hold 90% accuracy

E.2: Each table should cost no more than $90

E.3: System must be fast and accurate. System should have automatic update every 15 seconds

E.4: Security should receive notification when table has unattended belongings for more than 30 minutes

E.5: System needs to be on the UOIT website and available to be accessed by users

**6.3 Marketing Requirements:**

M.1: System should be user friendly

M.2: Cost should be minimized

M.3: System should uphold privacy and maintain safety of users

M.4: Information must be accurate

M.5: System should be able to store data for long term use

M.6: In the future system should be able to be modified to work with other systems (websites, apps, etc)

**6.4 Test Cases (Use Cases):**

T.1: Check seat availability

T.2: Upload library seating arrangement

T.3: View library usage statistics

T.4: Check library hours

T.5: Check library designated areas and zones

T.6: Access study room bookings

**6.5 Traceability Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stakeholder**  **Requirements** | **Engineering Requirements** | **Marketing Requirements** | **Test Cases** | **Justification** |
| S.1 | E.1  E.3 | M.1 | T.1  T.4  T.5  T.6 | Easy to use system. Users can navigate how to use system on their own |
| S.2 | E.1  E.3 | M.4 | T.2  T.3 | Refreshing to make sure system is giving up to date information on desk availability |
| S.3 | E.5 | M.6 | T.1  T.2 | The system needs to adapt to other existing systems such as the UOIT website and app |
| S.4 | E.1  E.3 | M.4  M.5 | T.2  T.3  T.4  T.5  T.6 | System can check which tables with belongings on it and update to website that table is taken |
| S.5 | E.3  E.4 | M.4  M.5 | T.2  T.3  T.4  T.5  T.6 | System checks duration of belongings left on unattended tables so that system functions correctly |
| S.6 | E.3  E.4 | M.4 | T.1  T.3 | System sends notification to security to check on unattended tables so that it can be freed for other students |
| S.7 | E.5 | M.3  M.5 | T.2  T.3  T.4  T.5  T.6 | System should not identify individuals and data should be kept secure and cannot be accessed by anyone |
| S.8 | E.5 | M.1  M.3 | T.2  T.3  T.4  T.5  T.6 | System should only be open to UOIT and Durham students to secure safety |
| S.9 | E.2 | M.2 | T.5 | System needs to have a reasonable cost because it will be implemented to many tables in the future which can become costly |

**7. Definition of Acceptance Test**

**Web Page Acceptance:**

* Web page being given dummy data should be able to reflect number of occupied and unoccupied seats in different sections of the library. This test should have 100% accuracy at this stage.
* If 5 different users are able utilise the web page from different devices with various different screen resolutions, they should be able to see the same webpage or similarly functional web page 100% of the time
* If users, using different devices are able to navigate through the web page and perform various different scenarios testing all of the web page’s functionality, Users should be able to perform 95% of the scenarios expected of them to perform.

**Microcontroller Acceptance:**

* Given dummy inputs to resemble inputs from Pyroelectric Infrared Sensor and Ultrasonic Sensors, the microcontroller should be able to determine if a seat is occupied, not occupied and have fully functional timing to track how long a seat has been occupied by personal *‘unattended’* belongings for. This functionality should have 100% accuracy given dummy data with set values is being used. (not with real sensors connected yet)

**Pyroelectric Infrared Sensor Acceptance:**

* Sensor should be able to detect an individual sitting at table and differentiate that from individuals in background (approximately 1 m, will be further investigated), with an accuracy of ‘at the very least’ 60%. This sensors accuracy maybe slightly lower quality, and this would be acceptable as the Ultrasound sensors will also be detecting if the seat is in use given there are objects on the table. In terms of timing the tables time of unattended belongings, noise in the signal (false positives) created by people walking by etcetera can be negated by the microcontroller’s software (processing multiple readings).

**Ultrasound Sensor Acceptance:**

* Sensor should be able to detect items on table with an accuracy of 80%. It is important that this accuracy is high to avoid false positives with the seat being occupied for extended periods of time. Also few items on table would also need to be detected as unattended belongings which may need to be removed after 30 mins.

**Pyroelectric Infrared Sensor & Ultrasound Sensor & Microcontroller Acceptance:**

* When the two sensors are connected to the Microcontroller, after all filtering is done by controller and data collected from sensors, this combined unit should have an accuracy of at least 75% to detection of table occupied or available.

**Full System Acceptance:**

* The overall detection and display of information should have at the least 70% accuracy at any single given time. Given that this information is real time and is constantly updated, overall the accuracy will balance out to be much higher than 70%. This is the minimum acceptance we will tolerate to assume the project functional.

**8. Project Plan**

The following project plan is for fall term 2018. The project plan covers all the requirement for this term which includes a Progress report, prototype and the Presentation

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Plan**  **Application Finding Seats in Library** | | | |
|  | Days | Start | Finish |
| **Progress Report 1** | **36** | **17/09/2018** | **19/10/2018** |
| Project Topic | 1 | 15/09/2018 | 15/09/2018 |
| Background Research | 7 | 15/09/2018 | 22/09/2018 |
| Stakeholders Requirement | 4 | 22/09/2018 | 25/09/2018 |
| Resources, Planning | 17 | 26/09/2018 | 12/10/2018 |
| Finalize Report | 7 | 12/01/2018 | 19/10/2018 |
| **Developing Prototype** | **30** | **20/10/2018** | **18/11/2018** |
| Design | 3 | 20/10/2018 | 23/10/2018 |
| Develop | 20 | 23/10/2018 | 11/11/2018 |
| Test | 1 | 12/11/2018 | 12/11/2018 |
| Fix Defects | 6 | 12/11/2018 | 18/11/2018 |
| Present to Stakeholders | 1 | 19/11/2018 | 19/11/2018 |
| In-Class Presentation | 1 | 26/11/2018 | 26/11/2018 |

**9. Concept Generation and Analysis:**

The project team had identified the existing problem of library study space being un-efficiently used causing students to have a hard time finding study space on campus. We then conducted some research on how monitoring of table usage can be done. This way we can detect when tables are empty and where they are in the library to help students find seating. Furthermore one of the major issues that currently affects the UOIT/DC Library is people ‘claiming’ library tables by leaving their personal belongings at the table all day, while they leave and go to classes or carry out other duties around campus.

As we started our research for similar projects we started to find a certain sensor, Passive Infrared Sensor (PIR), constantly being used in various forms of room occupancy projects. Often times these are used to monitor rooms and turn off lights when no heat motion is detected within the room.

One of the similar projects that caught our attention was Pressac’s Room Occupancy Project (Pressac, 2018). This project focuses on monitoring using small, unobstructive sensors to monitor occupancy of rooms, spaces, and tables. The sensors used in this project are all wireless allowing for seemingless streaming of data for analysis and displaying data. Furthermore the PIR sensors are kept in a small secure box that are attached to the bottom of the desks. This is something we will most likely need to implement as well to help keep the sensors safe from any form of damage or tampering.

Another similar project that stood out was library at University College London (UCL). This library has implemented a seat detection system to display to users how many seats are currently available at the libraries different sections. (UCL Library, 2018). While there library is much larger with various different buildings, we can still take away from the design they have used to create their website and display information to users.

While these projects offering good options for monitoring table usage and library availability based on individuals being at the table, this still doesn’t tackle our second issue of students claiming space in the library by leaving personal belongings on the tables. We found some studies indicating usage of load cells to detect weight on tables with four legs (Murao, K. & Terada, T, 2017). Unfortunately we would be unable to use such a set up in the library as the readings would likely be inaccurate given the carpet underneath causing a cushion effect and also the tables not having four distinct legs upon which to place these sensors.

The team also considered Ultrasound Sensors that could potentially be used for object detection. Although after performing some research the project team decided that ultrasound sensors would have too much scatter and deflection leading to inaccurate readings depending on how and where objects may be placed in reference to the sensor itself.

Finally, the project team did research on image processing. While the team was unable to find previous implementation of a birds eye image processing system, the option of converting the conventional image processing from a direct angle to a birds eye view appears to be feasible. The projects we found using image processing happened to be using it to determine occupancy to control building lighting (National Renewable Energy Laboratory, 2012). The image processing can be achieved using cellular phone cameras which can be found at relatively affordable prices helping maintain a low cost project plan (National Renewable Energy Laboratory, 2013).

After the project team had completed research on various viable solutions, the team had generated six concept plans to effectively enact a table monitoring system for the library, in essence these six consist of 2 concept options for each component of our system.

**Concept 1-A:** Wired system with cheaper sensors that do not contain wifi connectivity. This will allow for a cheaper costs for sensors although will also require a microcontroller to link the two sensors and upload the data to a server.

**Concept 1-B:** This system in comparison with the previous instead uses sensors with wifi connectivity, this way each sensor would automatically update data to the server without a need of micro controllers for each table.

**Concept 2-A:** Using Passive Infrared Sensor (PIR) for detecting if an individual is at the table. These sensors are very common for light management systems and can be found used in various projects. Often times these sensors are implemented as Time of Flight Sensors in which they detect how long a space has been empty for. PIR sensors are very commonly used and hence are easy to find at better prices and often readily available with different options such as wifi connectivity which will most likely be needed for an effective system.

**Concept 2-B:** Using Infrared Sensor to detect if an individual is present at the table. This sensor while not found used across as many projects more research and consideration may be needed here. Our research has found these sensors to be equally cost effective as compared to PIR. The benefit for the Infrared Sensor compared to the PIR is the ability to capture and recognize people who are sitting still at the table and not moving enough to activate the PIR sensor which would then lead to a false positive reading of the table being available.

**Concept 3-A:** Ultrasound Sensors to detect personal belongings being left behind at the table. Ultrasound sensors placed on each table would require a sensor on each table in order to collect data. Beyond this, the sensors them self would prove highly inaccurate due to the scatter of ultrasound waves. This may lead to many false positives in our system and seats appearing as available while they are not.

**Concept 3-B:** Using cameras and image processing to determine if table surface is clear or covered by any form of personal belongings. The camera approach will require setting up roof mounted cameras facing down at the tables, this may be more complicated to initially set up. The advantages are, this form of monitoring will have much higher accuracy and one camera would be able to manage a section of tables. Some testing will need to be done as to how many tables can be processed by one camera.

**10. Conceptual System Design:**

The tables below show a comparison of the concepts created in the previous section to understand which concept or concepts would be best suited for our project.

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Weight** | **Concept 1-A** | **Concept 1-B** |
| Ease to Update Software as Needed | 3 | 2 | 5 |
| Ease to set up Devices on Table | 4 | 3 | 4 |
| Ease of Software Development | 2 | 4 | 3 |
| Project Flexibility | 4 | 2 | 5 |
| Low Cost | 5 | 2 | 3 |
| Total Weighted Score | | 44 | 72 |

Weight Scale: 1: Insignificant to 5: High Priority

Score Scale 1: Does not Satisfy Needs to 5: Completely Satisfies Needs

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Weight** | **Concept 2-A** | **Concept 2-B** |
| People Detection Accuracy (Without considering Software filtering of data) | 4 | 4 | 5 |
| Portability | 3 | 5 | 5 |
| Projet Flexibility (Wifi capabilities in sensors) | 5 | 5 | 4 |
| Low Cost | 4 | 5 | 4 |
| Total Weighted Score | | 76 | 71 |

Weight Scale: 1: Insignificant to 5: High Priority

Score Scale 1: Does not Satisfy Needs to 5: Completely Satisfies Needs

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Weight** | **Concept 3-A** | **Concept 3-B** |
| Object Detection Accuracy | 4 | 2 | 5 |
| Ease of Software Development | 2 | 4 | 2 |
| Portability | 3 | 3 | 5 |
| Projet Flexibility (Wifi capabilities in sensors) | 4 | 5 | 5 |
| Low Cost | 5 | 3 | 5 |
| Total Weighted Score | | 60 | 84 |

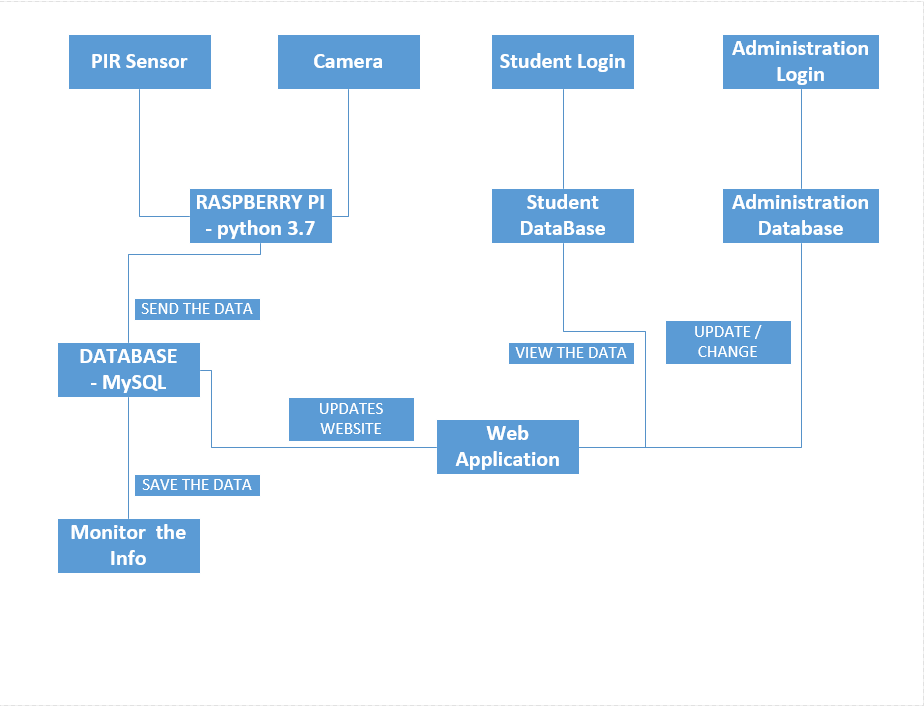
Weight Scale: 1: Insignificant to 5: High Priority

Score Scale 1: Does not Satisfy Needs to 5: Completely Satisfies Needs

Based on these analysis and comparisons of our different concepts the final design that best suits the project needs appears to be a combination of Concept 1-B. 2-A, and 3-B. Moreover the project will use a wireless system with wireless sensors, we will most likely use PIR sensors over pure infrared sensors, although this score in particular was very close, and we may reconsider if we can find a better infrared sensor with the utilities the project requires. Finally as for concept 3, it appears to be a better decision to tackle the problem using cameras and image processing. The main issue with ultrasound is the ultrasound tendency to bounce around causing a lot of miss readings, and different sensors on different tables may miss read each others signals causing more false positives.

Beyond the hardware components of our project mentioned thus far, the team also needs to complete various software components to integrate the system together and allow students and stakeholders alike easy access to the gathered data. One of the obvious software components is a website that will allow students to view number of available seats by library sections, as was mentioned in part one similar to the way in which ULC in London England currently displays its library availability. Furthermore in order to display this data we will need to first gather data from the sensors and compile the information in a database. From the database the project team will create various queries to retrieve meaningful information, such as number of available seats on each floor at the current time (last updated information). Long term statistics of what sections of library tend to be busiest on what times and days. Along with other queries to help monitor what seats have been left with unattended belongings for more than 30minutes.

Figure 10.1: Architectural Model of the Design



**11. Definition of Integration Tests:**

To define a set of integration tests we first looked at what components of our final product would need to individually be tested before it can be tested with the other components. The sensor we will be working with will have to detect the presence of the individual sitting at the table. We also need to consider individuals who are working at tables close by who could be detected by the sensor which could interfere with the results. To test this we could set up a sensor at a couple of tables near each other and have individuals sit at some tables to see if the sensor is able to detect the individual sitting at the specific table. Doing this will allow us to observe what limitations we have with the sensor and what constraints need to be put on the sensor.

Another integration testing that needs to be done is with the camera we will be using to detect objects on the table. Along with detecting objects on the table the camera needs to be work alongside the sensor so that we can have more accurate data being sent to the database. To test this we would need to connect our chosen sensor with the camera and when the sensor detects any movement the camera should be able to distinguish objects which are on the table. The camera will allow us to verify our results from the sensor.

Lastly, we should test if the complete system comprised of the sensor, camera, and web application are connected properly. We must check if the correct data is being displayed on the web application from the data taken from the sensor and camera. To do so we could have multiple tables set up with the sensor and camera and see what happens when items are placed on the tables or if a person is present at the table. When testing if object or person is found we should see on the web application that that specific table is occupied. If this is not seen corrections can be made to achieve this.

**12. Estimated Cost of the project:**

**Hardware Component**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | Name | Type | Manufacture | Supplier | $ CAN | Quantity |
| 1 | Raspberry Pi Zero W/ Camera | Logic Board | Raspberry Pi Foundation | BuyaPi.ca | $52.00 | 1 |
| 2 | Sandisk Ultra 16 GB | SD Card | Sandisk | Amazon | $10 | 1 |
| 3 | Infrared PIR Motion Sensor | Sensor | Aukru Power | Amazon | $12.99 | 1 |
| 4 | Camera V 2.1 | Camera | Rasberry Pi Foundation | Buyapi.ca | $0.00  Included W/ Pi | 1 |

**Total cost of Hardware: $74 + tax**

**Software Component**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name and Version | Type | $ CAN |
| 1 | PyCharm W/ Python 3 | IDE | Free |
| 2 | Brackets | Code Editor | Free |
| 3 | MQTT/Mosquitto | Messaging protocol | Free |
| 4 | XAMPP | Web Server | Free |
| 5 | Tensorflow | Python Library | Free |
| 6 | MySQL | Database | Free |

**Total cost for Software: $0.00**

**13. Updated Project Plan:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Plan- 1**  **Application Finding Seats in Library** | | | | |
|  | Days | Start | Finish | Progress |
| Project Topic | 5 | 10/09/2018 | 15/09/2018 | Completed |
| Background Research | 14 | 15/09/2018 | 29/09/2018 | Completed |
| **Progress Report 1** | **32** | **17/09/2018** | **19/10/2018** | **Completed** |
| Stakeholders Requirement | 14 | 29/09/2018 | 13/10/2018 | Completed |
| Resources, Planning | 5 | 13/10/2018 | 18/10/2018 | Completed |
| Design Website | 5 | 18/10/2018 | 23/10/2018 | Completed |
| Design Database | 5 | 18/10/2018 | 23/10/2018 | Completed |
| Develop Website | 11 | 23/10/2018 | 03/11/2018 | Completed |
| Develop Database | 11 | 23/10/2018 | 03/11/2018 | Completed |
| Research Hardware Components (Sensors and Microcontroller) | 7 | 17/10/2018 | 24/10/2018 | Incomplete |
| Purchase Hardware Components (For Prototype) | 14 | 24/10/2018 | 08/11/2018 | Complete  (No research) |
| Create Hardware Prototype | 10 | 08/11/2018 | 18/11/2018 | Incomplete |
| Integrate Database with Website | 4 | 03/11/2018 | 07/11/2018 | Completed |
| Integration Testing for Database and Website and fix defects as needed | 7 | 07/11/2018 | 14/11/2018 | Completed |
| Integrate Database and Website with Hardware Prototype (Sensors and Microcontroller) | 4 | 14/11/2018 | 18/11/2018 | Pending |
| Integration Testing for complete prototype and fix defects as needed | 4 | 18/11/2018 | 22/11/2018 | Pending |
| In-Class Presentation | 1 | 22/11/2018 | 22/11/2018 | Done |
| **Progress Report 2** | **15** | **22/11/2018** | **07/12/2018** | **Done** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Plan- 2**  **Application Finding Seats in Library** | | | | |
|  | Days | Start | Finish | Progress |
| Finalize Research of Hardware Components and Order components | 7 | 15/12/2018 | 22/12/2018 |  |
| Design – Hardware | 4 | 22/12/2018 | 26/10/2018 |  |
| Develop Database (Complete Statistical Analysis and all Queries required) | 39 | 01/01/2019 | 09/02/2019 |  |
| Develop Website (Complete all Pages required) | 39 | 01/01/2019 | 09/02/2019 |  |
| Develop Hardware and Hardware Related Software (Sensors, image Processing of Cameras, uploading data using MQTT) | 53 | 01/01/2019 | 23/02/2019 |  |
| Software Integration Testing (Website + Database) and Fix Defects | 14 | 09/02/2019 | 23/02/2019 |  |
| Software and Hardware Integration Testing and Fix Defects (Entire System) | 22 | 23/02/2019 | 17/03/2019 |  |
| Present to Stakeholders | 1 | TBD | TBD |  |
| Final Report | 1 | TBA | TBA |  |
| In-Class Presentation | 1 | TBA | TBA |  |
| Show Case | 1 | TBA | TBA |  |

**14. Detail Design**

**14.1 Architecture**

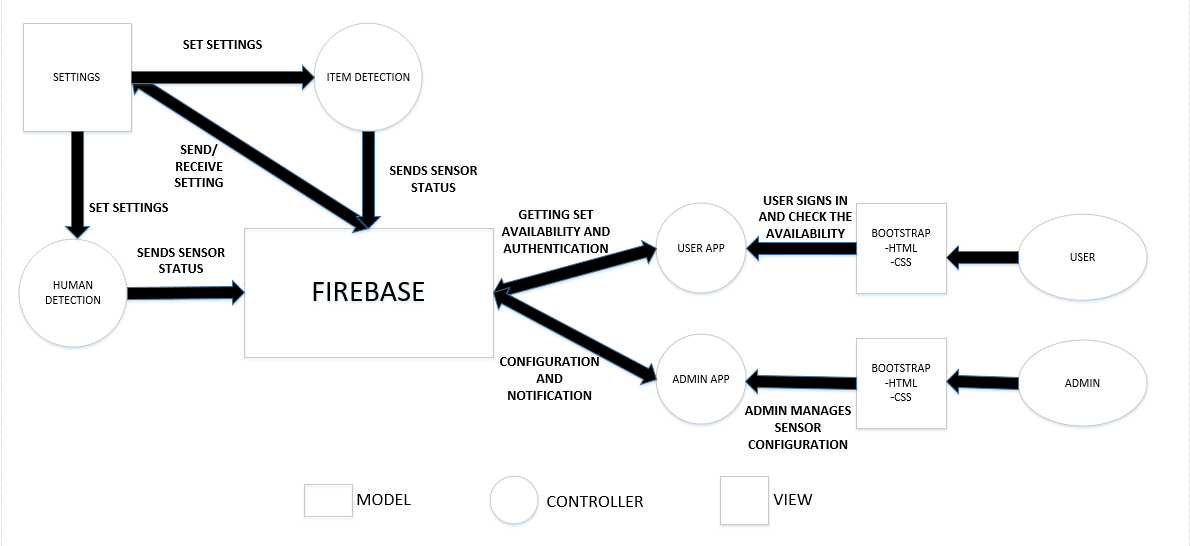
· User Application - Allows students and staff to log in and check library space availability Using Django Framework

· Firebase Database – Synchronizes data between the web app and the detection system

· Admin Application – Allows admin to change sensor and system settings

· Human Detection – This System detects if a person is present at a table

· Item Detection – Detects if items are left on the table.



|  |
| --- |
| Figure 14.1: Overall Architecture Diagram |

**14.2 Item Detection**

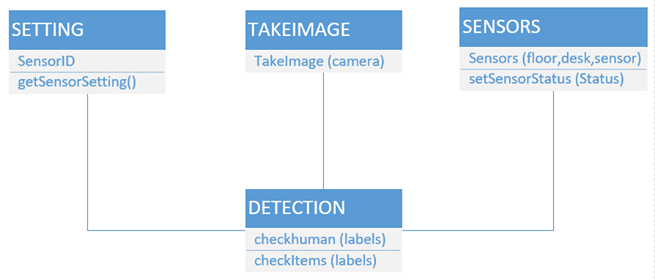
Image Processing will be performed using Google Cloud Vision API. This process will have the Raspberry Pi’s take pictures with a mounted camera module. The picture will be temporarily saved on the pi and uploaded to Google Cloud Vision for image processing

Google Cloud Vision API returns what objects it detects in the image and confidence ratings as to how confidently it feels those objects are in the image. This results in a large list of objects some with high confidence while others with very low confidences. These results are then processed by our system to determine if a student is present in the study space and if any objects are present in said space.

Hardware:

* Raspberry Pi Zero - Most compact Raspberry Pi in the market.
* Raspberry Pi Camera Module - Kuman Camera Module 5MP 1080p

**14.2.1 Item Detection App**



|  |
| --- |
| Figure 14.2: Item Detection App |

Theitem detection app imports in the settings, takeimage, and sensors class. The settings class is responsible for loading the detection settings which which involves attributes such as the floor number and the desk number of the specific Raspberry Pi. The TakeImage class takes care of taking a picture using the built in Raspberry Pi camera. The Sensors class is responsible for setting the sensor status to the firebase database which takes in the floor, desk and sensor attributes

**14.3 User App**

User account authentication is handled by the Firebase API. To communicate to the API from Python a wrapper by the name of Pyrebase is used. Pyrebase allows for use of multiple firebase services such as Authentication, Database and Storage. In our particular case we will just be using the Authentication and the Database.

In order to store and retrieve user info in the database the users.py class is used. The user session is handled by Django middleware application that will create a cookie based session once the user authenticates through the Firebase API. Each user has a unique UID that can be retrieved through the session cookie.

The SignIn function will communicate with the Firebase API to find a matching Username and Password. Once that’s found the Firebase API will return the UID which is then stored in the

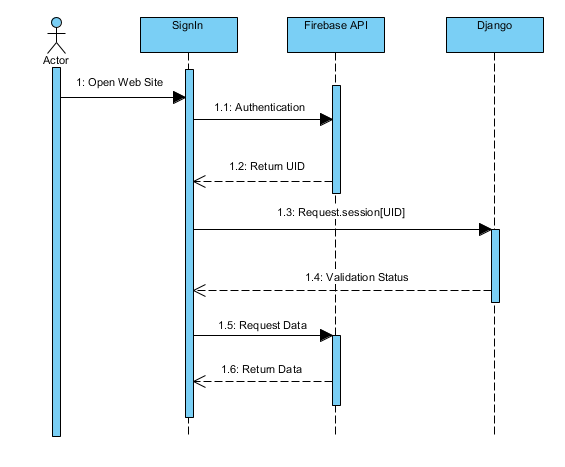
****

Figure 14.3: displays a sequence diagram on how authentication is handle

**Users.py (users app)**

The UID is then used to set and retrieve the user data from the database using the set and getter functions in Users.py

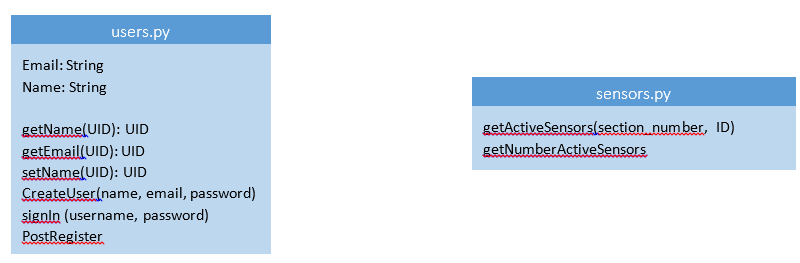


Figure 14.4: Users App

**Sensor.py(user app)**

The user app is also composed of the Sensor.py class. In order to get the number of active sensor the getNumActiveSensors function is used which takes the section\_number of the library space and the UID used to authenticate with the database that it’s a user with a valid UID.

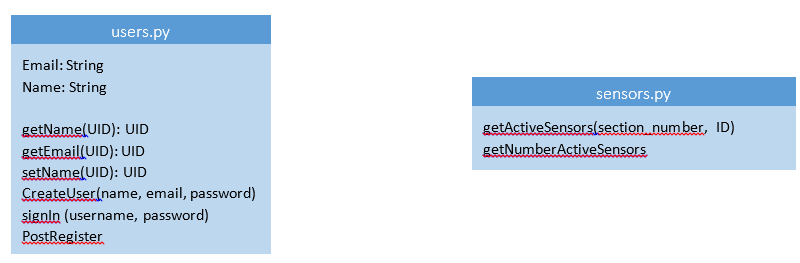


Figure 14.5: Users App - sensors

**Admin App**

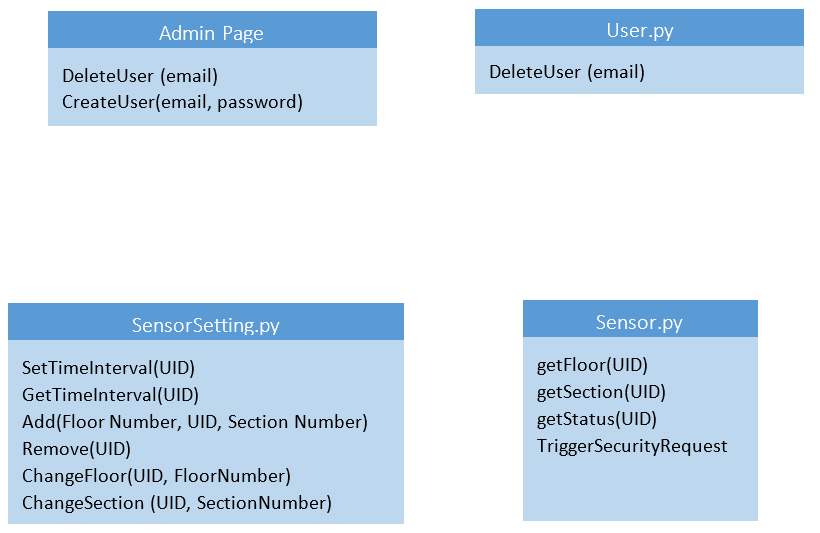


Figure 14.6: Admin App

User.py (admin app)The admin app is composed of four major components. There is the admin class which allows a user to create and delete users. a user class which allows the admin to delete regular users. The Sensor settings class allows for setting the time interval of when a tabletop has had items on a desk for more than a certain period of time for a particular sensor as well as displaying the current time interval set. Also allows for adding and removing sensors as well as updating the floor and section attributes and the sensor class allows for getting various sensor information from the database.

**14.4 Frameworks:**

**Bootstrap Framework:** Front end API: HTML, CSS and JavaScript library which contains templates for front-end website development.

**Django Python Framework:** Handles functions required by the website. This includes the Sign In request which communicates with the Firebase API to authenticate. Logout request which terminates session and retrieving sensor status from the database.

**Firebase API**:Handles user authentication log in. Once logged in it allows for access to the sensor status and limited account info such as student’s first name.The firebase API will also handle the Real Time Database triggers. If sensor1(seat) in the database has been active for 30 minutes longer than sensor2(tabletop) then an email notification will be sent with the Floor, Section number and timestamp of the table.

**Google Cloud Vision API:** Used for image processing. Initially we planned to use Tensorflow, although the Raspberry Pi’s were not capable of handling the image processing. Google Cloud Vision API allows us to make use of cloud computing to process the image, and returns objects with confidence values as to how sure the API is about the object being present in the image.

**15. Integration and/or Unit Testing:**

**14.1 Web Interface Login Test Results**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Test Case Name** | **Input/Precondition** | **Post Condition** | **Expected Output** | **Observed Output** | **Test Result** |
| **1** | Empty Credentials | Login:  Password: |  | Alert: Fill out this field | Alert: Fill out this field | OK |
| **2** | Empty Username | Login:  Password: ‘pass’ |  | Alert: Fill out this field | Alert: Fill out this field | OK |
| **3** | Empty Password | Login: ‘carlos.fabregasmejia@uoit.net’  Password: |  | Alert: Fill out this field | Alert: Fill out this field | OK |
| **4** | Invalid Email | Login: ‘wrongemail@uoit.net’  Password:capstone123 | Return: SignIn.html, “Alert Message” | Alert: Invalid Email or Password | Alert: Invalid Email or Password | OK |
| **5** | Invalid Password | Login: ‘carlos.fabregasmejia@uoit.net’  Password:‘wrongpassword’ | Return: SignIn.html, “Alert Message” | Alert: Invalid Email or Password | Alert: Invalid Email or Password | OK |
| **6** | Invalid Email and Password | Login: ‘wrongemail@uoit.net’  Password:‘wrongpassword’ |  | Alert: Invalid Email or Password | Alert: Invalid Email or Password | OK |
| **7** | Valid Username and Password | Login: ‘carlos.fabregasmejia@uoit.net’  Password: ‘capstone123’ | Return welcome.html | Login redirect to homepage | Login redirect to homepage | OK |

**16. Updated Project Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Plan- 2**  **Application Finding Seats in Library** | | | | |
|  | Days | Start | Finish | Progress |
| Finalize Research of Hardware Components and Order components | 7 | 15/12/2018 | 22/12/2018 | Done |
| Design – Hardware | 4 | 22/12/2018 | 26/10/2018 | Done |
| Develop Database (Complete Statistical Analysis and all Queries required) | 39 | 01/01/2019 | 09/02/2019 | Done |
| Develop Website (Complete all Pages required) | 39 | 01/01/2019 | 09/02/2019 | Done |
| Develop Hardware and Hardware Related Software (Sensors, image Processing of Cameras, uploading data using MQTT) | 53 | 01/01/2019 | 23/02/2019 | Done |
| Software Integration Testing (Website + Database) and Fix Defects | 14 | 09/02/2019 | 23/02/2019 | In progress |
| Software and Hardware Integration Testing and Fix Defects (Entire System) | 22 | 23/02/2019 | 17/03/2019 | Done |
| Present to Stakeholders | 1 | TBD | TBD | Done |
| Final Report | 1 | TBA | TBA | Done |
| In-Class Presentation | 1 | TBA | TBA | Done |
| Show Case | 1 | TBA | TBA | Done |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Plan- 3**  **Application Finding Seats in Library** | | | | |
|  | Days | Start | Finish | Progress |
| Develop Database (Complete Statistical Analysis and all Queries required) | 31 | 09/02/2019 | 12/03/2019 |  |
| Develop Website (Complete all Pages required) | 71 | 01/01/2019 | 12/03/2019 |  |
| Develop Hardware and Hardware Related Software (Sensors, image Processing of Cameras, uploading data using MQTT) | 71 | 01/01/2019 | 12/03/2019 |  |
| Software Integration Testing (Website + Database) and Fix Defects | 3 | 12/03/2019 | 15/03/2019 |  |
| Software and Hardware Integration Testing and Fix Defects (Entire System) | 2 | 15/03/2019 | 17/03/2019 |  |
| Present to Stakeholders | 1 | 25/03/2019 | 25/03/2019 |  |
| Final Report | 1 | 05/04/2019 | 05/04/2019 |  |
| In-Class Presentation | 1 | 02/04/2019 | 02/04/2019 |  |
| Show Case | 1 | 05/04/2019 | 05/04/2019 |  |

**17. Acceptability Planning/Test**

Scenario 1:

If the study table is occupied by the person the website will be updated

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test case # | Test case name | Preconditions | Input | Output | Purpose | Process |
| 1 | Human Body Detection and Updating it on Website | Person sitting at the table | Camera signal | Website was updated when Camera detected the person | The purpose of this experiment was that to observe how accurate the camera detects the person and update the website | To set a camera at the certain angle where it can detect the person sitting at the table and send data to the database so that website can be updated |

Scenario 2:  
If Human is not detected at the table check and if there is any object on the table and then update the website

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test case # | Test case name | Preconditions | Input | Output | Purpose | Process |
| 1 | Object Detection | Objects placed on the tables | Different objects on table | Camera was able to detect the different object | The purpose of this was that to see if there is object on the table | To set the camera at the certain angle where it can detect the object on the table |
| 2 | Updating the Website | Detecting the objects | Object is Either at the table or table is empty | It was able to update the database that there is object but it was not able to update on website. | The purpose to update the website if object is detect so that we know if the table is empty or not | The process is to get camera to send data over to database which will update the website. |

Scenario 3  
If students are able to login successfully and able to view the study space

**Purpose**: Purpose of this scenario was to see if the student can successfully login in to the website by using the correct credential, and also it gives error if the credentials are not right

**Process**: The process is that to add the user’s Id and the password into the database and only after that user can login into the Web application

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case #** | **Test Case Name** | **Input/Precondition** | **Output** |
| **1** | Empty Credentials | Login:  Password: | Alert: Fill out this field |
| **2** | Empty Username | Login:  Password: ‘pass’ | Alert: Fill out this field |
| **3** | Empty Password | Login: ‘carlos.fabregasmejia@uoit.net’  Password: | Alert: Fill out this field |
| **4** | Invalid Email | Login: ‘wrongemail@uoit.net’  Password:capstone123 | Alert: Invalid Email or Password |
| **5** | Invalid Password | Login: ‘carlos.fabregasmejia@uoit.net’  Password:‘wrongpassword’ | Alert: Invalid Email or Password |
| **6** | Invalid Email and Password | Login: ‘wrongemail@uoit.net’  Password:‘wrongpassword’ | Alert: Invalid Email or Password |
| **7** | Valid Username and Password | Login: ‘carlos.fabregasmejia@uoit.net’  Password: ‘capstone123’ | Login redirect to homepage |

Scenario 4

In this Scenario if 2 cameras detect people and a third doesn't, website should show 1 seat available

**Purpose:** The Purpose of this Scenario to see all the camera are working together and also updating the website.

**Process:** The process was that having two camera to detect the person and leaving one table empty and letting those data get updated on database and which will update on the website.

|  |  |  |  |
| --- | --- | --- | --- |
| Test case # | Test case name | Input | Expected output |
| 1 | Camera 1 | Person at sitting table | Table unavailable |
| 2 | Camera 2 | Person sitting at table | Table unavailable |
| 3 | Camera 3 | Nothing detected at table | Table available |
| 4 | Updating Website | Person is Either at the table or table is empty | One table available |

**18. Test Results**

Scenario 1:

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Date | Pass or Fail | Comments |
| 1 - Human Body Detection and Updating it on Website | 29/03/2019 | PASS | * Show which table # is available and not Space * Need to increase the accuracy of the human detection * Adjust the Threshold for detecting the human |

Scenario 2

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Date | Pass or Fail | Comments |
| 1 - Object Detection | 29/03/2019 | PASS | * Need to increase the accuracy of the object detection * It is better to detect the stuff rather then the Identify the Stuff * Test for all the four state   + No Human, No Object   + No Human, Yes Object   + Yes Human, No Object   + Yes Human, Yes Object |
| 2 - Updating On Website | 29/03/2019 | Fail | * Make website more attractive * Have a Study table # showing not the Number of study tables are available |

Scenario 3

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case #** | **Date** | **Test Result** | **Comments** |
| **1**- Empty Credentials | 29/03/2019 | Pass | N/A |
| **2**- Empty Username | 29/03/2019 | Pass | N/A |
| **3**- Empty Password | 29/03/2019 | Pass | N/A |
| **4-** Invalid Email | 29/03/2019 | Pass | N/A |
| **5**- Invalid Password | 29/03/2019 | Pass | N/A |
| **6**- Login: ‘wrongemail@uoit.net’  Password:‘wrongpassword’ | 29/03/2019 | Pass | N/A |
| **7** - Login: ‘carlos.fabregasmejia@uoit.net’  Password: ‘capstone123’ | 29/03/2019 | Pass | N/A |

Scenario 4

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Date | Pass or Fail | Comments |
| 1 - Camera 1 Detection | 29/03/2019 | PASS | * Adjust the threshold for the detecting Human |
| 2 - Camera 2 Detection | 29/03/2019 | PASS | N/A |
| 3 - Camera 3 Detection (No Human) | 29/03/2019 | PASS | N/A |
| 4 - Updating website | 29/03/2019 | PASS | * Make it more accurate |

**19.Ethical Considerations**

One of the main ethical concerns for this product would be when dealing with the Raspberry Pi camera. Since people walking around the perimeter of the table will be caught on camera as well as the individual sitting at the table this will bring a concern to people. People will want to know why there is a camera and where these images are being kept. To prevent this students should be given a warning before installation of the product and they should be informed that these images will not capture the faces of the person sitting at the table. Due to the use of the Google Cloud Vision API, Google currently has full control of the images being taken and this could cause some concern to people as Google can do whatever they want with those images.

**20. Safety Considerations**

When considering the safety of the users of the system the project team had spoken to Student Life at UOIT, who had recommended that we keep the system to only be open to UOIT students. The system could later be adapted for Durham College students as well. The team had decided the best solution for preventing people outside of UOIT using the system is to only allow login if the person has an UOIT email. Another safety consideration would be if the camera were to get corrupted. Having a camera involved could raise concerns of students as they are going to have images of them taken. If the camera gets corrupted by the system getting hacked in any way this could possibly result in the images being stolen. Since the Raspberry Pi is using WIFI, it can be easily hacked if someone is able to find out the password.

**21. Conclusions**

The Capstone Project helps the UOIT library efficiently use an existing space for students to study. This is an issue that has been personally affecting the project team during the team’s time at UOIT. While many students feel a library expansion is needed, the team decided to help offer an alternative solution. Rather than expanding the library, we could help make its usage more effective. Earlier in the year the project team had spoken to the stakeholders who had informed us about the issue of students leaving their belongings on tables for excessive amounts of time while the student were not there. The team decided, the best solution to perform detection, was to use a Raspberry Pi with a Raspberry Pi camera module. Images from the Pi are uploaded to the Google Cloud Vision API, to perform image processing. The results from the API are then returned and processed by the system to determine if a student or any objects are present at the table. All the table state data was then continuously sent to the database and uploaded to the web page for students to access. The web page currently shows how many seats are available on each floor and students can access this by logging in with an UOIT email.

In the future we plan on improving our human and object detection accuracy. Along with this we want to be able to use the data we have gathered to observe what the peak hours of use in the library are. The team would also like to incorporate a library floor plan to the web page so that it is easier for students to identify which tables are available. Another future plan would be to expand to UOIT’s various other study spaces around the campus. Lastly, to make it easier for students to access the web page the team would like to create Android and iOS applications for the system to be deployed on.

While working on the capstone project, the team was able to build on previous concepts learnt in other courses. The team was also able to use knowledge from courses such as Software Architecture and Design when having to create the software architecture and other diagrams such as use case diagrams in this project. The team has also been able to understand how to test a product and what documentation needs to be prepared before testing the product. Along with these the team has been able to work on new platforms such as the Google Cloud Vision API when having to do human and object detection. In conclusion, the capstone project has allowed the team to further explore the engineering curriculum and execute this project idea successfully.

**22. Acknowledgements**

Our capstone group would like to thank Dr. Ramiro Liscano for guiding and assisting us throughout our project. We would also like to thank Dr. Ramiro Liscano for giving us access to the lab room for us to work on our project and giving us suggestions along the way on how to refine our project. We would also like to thank Student Life and UOIT library for being supportive stakeholders. The UOIT Library had given us ideas on how we can improve our original design and gave us requirements to follow. We would also like to mention the UOIT librarian Karin Downie for giving us space in the library to work on our project and perform testing. The team would also like to thank Likhith Areekkal from Library IT for pointing us in the right direction and suggesting how we can implement our design. Lastly, we would like to thank UOIT for giving us this opportunity to work on a project which enabled us to apply our engineering curriculum and further expand our knowledge.

**23. References**

Murao, K., & Terada, T. (2017, January). Activity Recognition and User Identification based on Tabletop Activities with Load Cells. Retrieved December 2, 2018, from <https://www.researchgate.net/publication/312351947_Activity_Recognition_and_User_Identification_based_on_Tabletop_Activities_with_Load_Cells?fbclid=IwAR15qcY1W5P-kDxo0lZixnTahLTtaZmejCN_Nc6rUwDh_nouxiTP9T2xGJc>

National Renewable Energy Laboratory. (2012, December 08). Image Processing Occupancy Sensor. Retrieved December 08, 2018, from <https://techportal.eere.energy.gov/technology.do/techID=986>

National Renewable Energy Laboratory. (2013, June 22). Image Processing Occupancy Sensing. Retrieved December 2, 2018, from https://3k8h4uqzt88x5.cloudfront.net/sites/default/files/publications/Low-Cost Sensing\_L\_GENTILdE\_POLESE.pdf

Pressac. (2018, November 29). Occupancy Monitoring. Retrieved December 1, 2018, from <https://www.pressac.com/occupancy-monitoring/>

UCL Library. (2018, September 12). Available Study Spaces. Retrieved December 1, 2018, from https://www.ucl.ac.uk/library/libraries-and-study-spaces/available-study-spaces

**24.Appendix**

**API:** also known as Application Programming Interface, is a set of subroutine definitions, communication protocols and tools which are used to build different types of software.

**Authentication:** is the action of showing that something is true or valid. It is also the process used to verify a person’s identity.

**Database:** is a collection of data which is organized to be accessed, managed and updated easily. Databases also query data and run applications against it.

**Django:** is an open source web framework used to develop database-driven websites. Django also uses Python to create files, settings and data models.

**Framework:** it provides a way for users to build and deploy their applications.

**Firebase:** is a mobile and web application platform which belongs to Google. The platform allows users to quickly develop apps.

**Google Cloud Vision API:** is used to quickly classify images into many categories while also detecting objects and faces within the image. It also can help in reading words found in images.

**Image Processing:** is a form of analyzing an image to be used to do detection of what is found in the image or to alter an image.

**Integration:** is the process of combining components in a system together and making sure that the components all function together.

**PIR sensor:** also known as a passive infrared sensor, is an electronic sensor used to measure infrared light which radiates off objects. Usually used in motion detectors.

**Raspberry PI:** is a low cost, small computer which can be used as a desktop computer which can have many functionalities similar to a regular computer.

**Ultrasonic sensor:** sensors which measure distance which can be done using the ultrasonic signals that are being transmitted.

**Contribution Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Carlos | Angelie | Anjali | Mohammed Maaz |
| Title Page |  |  | ✔ |  |
| Executive Summary |  | ✔ |  |  |
| Table of Contents/Tables/Figures | ✔ | ✔ | ✔ |  |
| Report 1 | ✔ | ✔ | ✔ | ✔ |
| Report 2 | ✔ | ✔ | ✔ | ✔ |
| Revised Report 3 | ✔ | ✔ | ✔ | ✔ |
| Ethical Considerations | ✔ | ✔ | ✔ | ✔ |
| Safety Considerations | ✔ | ✔ | ✔ | ✔ |
| Acknowledgements |  | ✔ |  | ✔ |
| References |  |  | ✔ |  |
| Appendices |  | ✔ |  |  |