



EIH

Emergency Info Hub

Final Report

- Osama Abou Hajar - C00220135.
- 4th Year Software Eng.
- Submission Date: 20/04/2020

Supervisor **Paul Barry**



INSTITUTE of
TECHNOLOGY
CARLOW

Institiúid Teicneolaíochta Cheatharlach

Abstract

Emergency Info Hub (EIH), is a central website helps the emergency services to prepare, respond and recover from a disaster, by providing all the needed data for the targeted building (E.g. Number of people, area size and emergency exits).

The main objective of this project is the count of the number of trapped people under rubbles or inside a building, by tracking their number using a simple movement sensor fitted on the main gate by Body detection technology, and save this number to the cloud to be used when a disaster happens

Table of Contents

Abstract.....	2
Table of Contents	3
Table of Figures.....	5
Section 1: Introduction.....	6
Section 2: Project Description.....	7
2.1. Hardware Description.....	7
2.1.1. Body Detection approach used in the current release.....	8
2.1.2. Face Detection approach did not work.....	12
2.1.3. Live Streaming Approach for future work.....	13
2.2. Front-End	15
2.2.1 Website Screenshots.....	15
A. the Main page.....	15
B. the results page.....	18
C. Login Screen.....	20
D. All Locations Screen.....	21
E. About Us Screen.....	24
F. Contact US page.....	25
2.3. Database Structure.....	26
2.4. Development challenges.....	27
2.4.1 Hardware challenges:.....	27
2.4.2. Software challenges:.....	30
Section 3: Python Unit Testing.....	32
Section 4: Learning outcome.....	33
3.1. Internet of things (IoT):.....	33
3.2. Sensors:.....	33
3.3. Programming Languages.....	33
3.4. Open-CV and object detection.....	33
3.5. Firebase and Google API.....	34
3.6. Full-stack project	34

3.7. Reusing the previous experience.....	34
3.8. Personal Learning.....	34
Section 5: Project review and Conclusion.	35
4.1. What went wrong?	35
4.2. Future work?	35
Acknowledgements	36
Plagiarism Declaration	37
Declaration	37

Table of Figures

Figure 1 connect the IR sensor to the Rasp PI	9
Figure 2 how to install the IR sensors on the doors	10
Figure 3 connect the camera to the rasp Pi	10
Figure 4 the camera installations for one direction	11
Figure 5 the final diagram of the two system directions.	11
Figure 6 miss detect for the faces in the picture with extra face.....	12
Figure 7 the original algorithm for the people counter	13
Figure 8 the new version of the People count algorithm	14
Figure 9 the front-end main page.....	15
Figure 10 menu screen	16
Figure 11 Address AutoComplete function	17
Figure 12 Google Map API	18
Figure 13 alert to show the search address not registered.	19
Figure 14 admin login Screen	20
Figure 15 add new admin to the system.....	20
Figure 16 the admin table menu	21
Figure 17 all locations screen for not admin users.....	21
Figure 18 add building steps.....	22
Figure 19 registration key for the added building	22
Figure 20 about us page	24
Figure 21 contact Us form.....	25
Figure 22 the confirmation page.....	25
Figure 23 falling test case with the direction	27
Figure 24 - 2 people chatting challenge.....	28
Figure 25 changing the direction	28
Figure 26 challanging case with 2 sensors and 2 people	29
Figure 27 the final picture of the system used.....	30

Section 1: Introduction.

This document contains the process of developing the Emergency Info Hub (**EIH**). EIH is a full stack project focusing on the number of people inside a building by counting the people entering and leaving the building through the main doors. The software (application) is developed using Raspberry Pi, camera and sensors as hardware to detect the movement. Body Detect algorithm is used to count the number of people entering/leaving the building and then store the data on the cloud as an API and web application to represent the data when it requested.

This report will outline the description of the final submitted project, it will contain a brief overview of the Hardware, Front-End, Database structure, all technology used to develop this project, detailed description of each application screen and its functionality. This section will also discuss the difficulties faced in development, approach changing, those that were solved and those that were not completely solved.

The next section of this document will explore how the final product compares to the original idea specification, it will define what changed during the development process, and what extra features have been added in or excluded. This section will discuss changes to the overall hardware approach of people detections all different approaches were taken.

The personal and technical learning outcome will be discussed in this document. The section dedicated to this will have a detailed analysis of the technologies used to develop the EIH will be explored.

This document will be concluded with the overview review of the completed project, both successes, and failures of developing EIH, what would change if the project will be started again from scratch and what could be done next if the project would be developed forward.

Section 2: Project Description.

Emergency info Hub is a full-stack product including the development from end-to-end. The hardware used as backend detects the number of people entering/leaving an area and stores the data gathered on the cloud. As for front-end the application will process the data from the cloud and will display the data to the user as requested.

As the project is made up of different components, each of its parts needs to be working smoothly and in real-time latency to prevent different problems. Every component must function correctly from passing data to the next related side to grantee the quality and efficiency of the project work.

A detailed description is provided in the next section about each element of this project.

2.1. Hardware Description.

At the time this project was done, there were no similar ideas implemented on the market before. This made the project a challenge as the application needs to be accurate and useful using only new approaches.

Several different approaches were taken and tested, and they were completely different from one to another, each approach was implementing a new type of sensors, algorithms, and software.

Face detection, Body detection, and Live people detection were tested. Body detection was giving the highest accuracy in counting people without errors. The face detection was failing as was missing a big number of people's faces and sometimes considering another object in the picture as a person (i.e. drawn face on a jumper) which resulted in the wrong total number of people inside the building. The live people detection was under development as mentioned below in details until the Irish government set the new lockdown instructions and closed all colleges campus and limit the gathering in public, which create a big obstruction in the testing part, resulting in holding the development for this algorithm for future work and additional support machines to be used once the COVID-19 pandemic finished.

Meantime, the body detection was under development to improve the quality, performance and was chosen in the end to be used as the main detection algorithm for the current release of the project.

2.1.1. Body Detection approach used in the current release.

The body detection approach is used in the current release of EIH. Hardware, software, and sensors described below work sequentially to complete the process.

The approach process starts when a person moves toward the door and breaks the beams from the IR sensors. Once the IR beams break-in sequence, IR Sensor 1 then IR sensor 2, will instantiate a subprocess that will take a picture of the entrance. This picture will be saved on the local machine and run another subprocess to run the body-detect algorithm using the open-cv library.

Once the person's body got detected this number will be stored on the cloud after the decision has been made: to add or subtract the detection from the total number on the cloud.

These two sensors for each direction were added to guarantee the person directly and to run the suitable process depending on this cut. More details are listed below to describe the hardware parts, collaboration, test cases...etc.

Requirements Hardware.

The hardware used:

- Raspberry Pi 4 with RAM 4GB LPDDR4-2400 SDRAM, 8 GB SD card
- Raspberry Pi Camera
- IR Break Beam Sensor LEDs

Software Dependencies

The dependencies:

- Requests
- python-firebase
- python-opencv
- python-scipy
- ipython
- firebase
- imutils
- python_jwt
- gcloud
- sseclient
- parse
- requests_toolbelt
- pyrebase

all can be installed with a single file created within the project back-end folder called *setup.py*, (*more details in the technical manual and the GitHub REAMME.md file*)

Collaboration:

To have the best performance of the hardware the following instructions must be followed:

IR Break Beam Sensor LEDs

- Sensor one should be connected to GPIO 4 PIN (7)
- Sensor two should be connected to GPIO 17 PIN (11) As is shown in **figure1**.

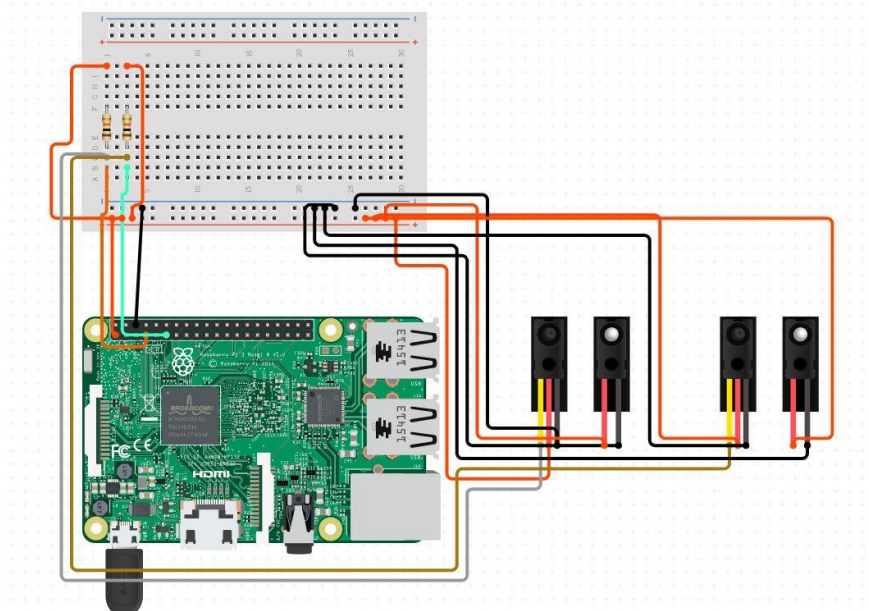


Figure 1 connect the IR sensor to the Rasp PI

Both sensors must be installed on both sides of the door, with range [25cm] to [50cm] height from the ground (taken the kids' and adults' height under consideration as shown in **figure 2**).

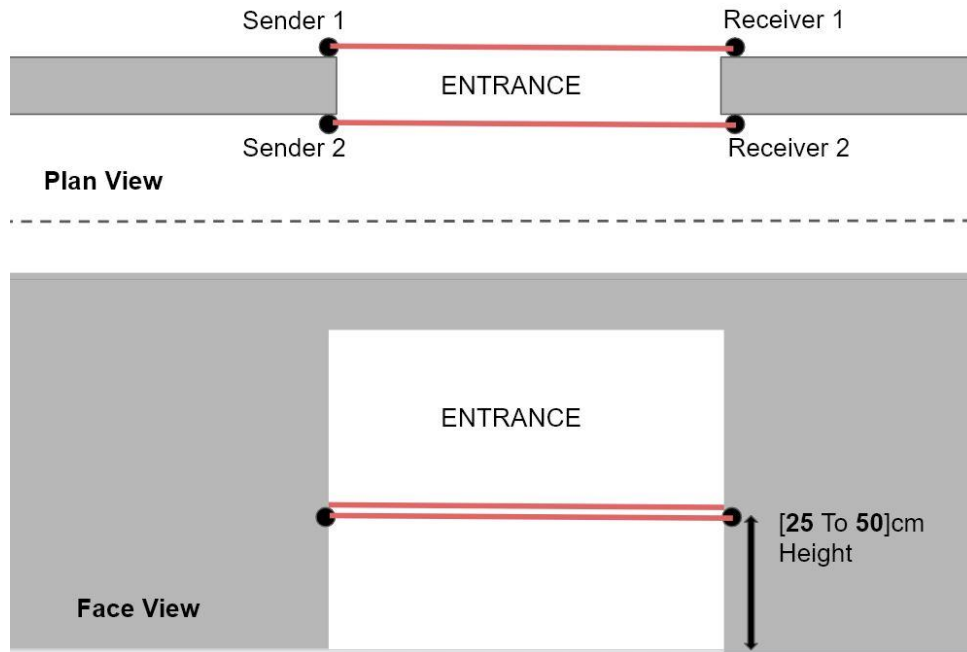


Figure 2 how to install the IR sensors on the doors

Raspberry Pi Camera

The camera connects to the camera port on the Raspberry Pi and can be enabled from the configurations as it is shown in **figure 3**.

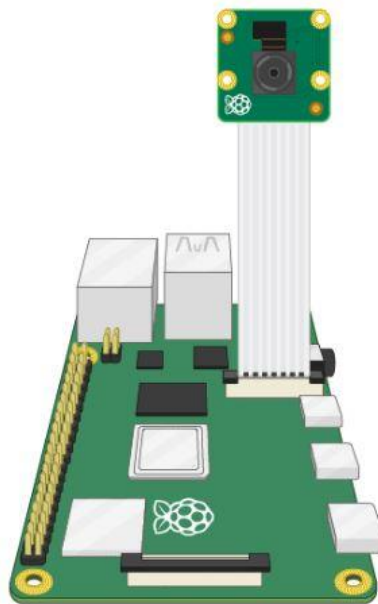


Figure 3 connect the camera to the rasp Pi

The camera should be installed closer to the midpoint of the entrance, with height range [190cm] to [220cm], with distance and an angle covers the full Entrance width and height **figure 4**.

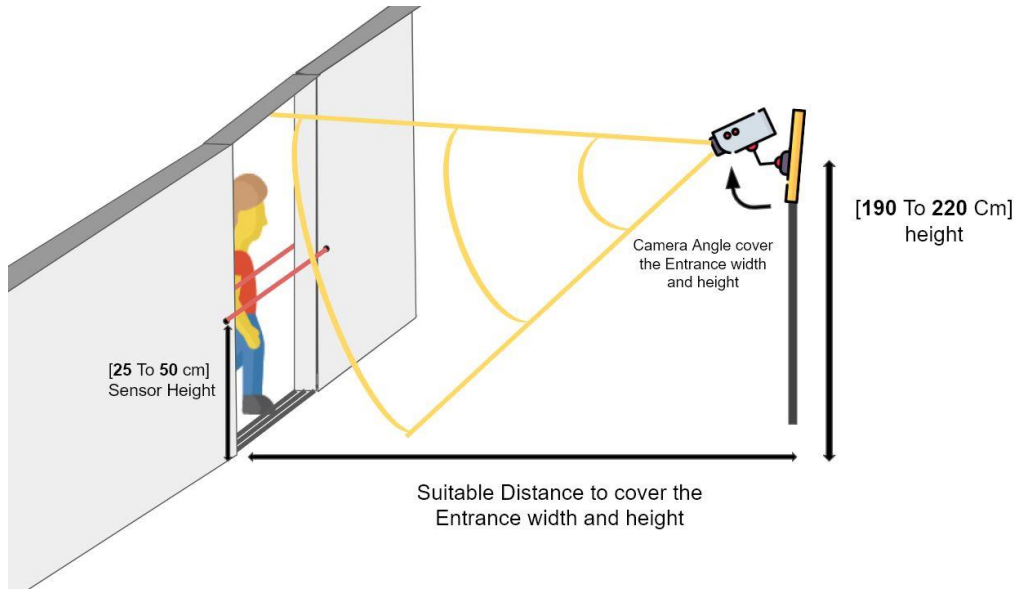


Figure 4 the camera installations for one direction

The picture above represents the system in one direction, another camera is needed from the opposite direction. The second camera and sensors are needed to have the full system ready in both directions **figure 5**.

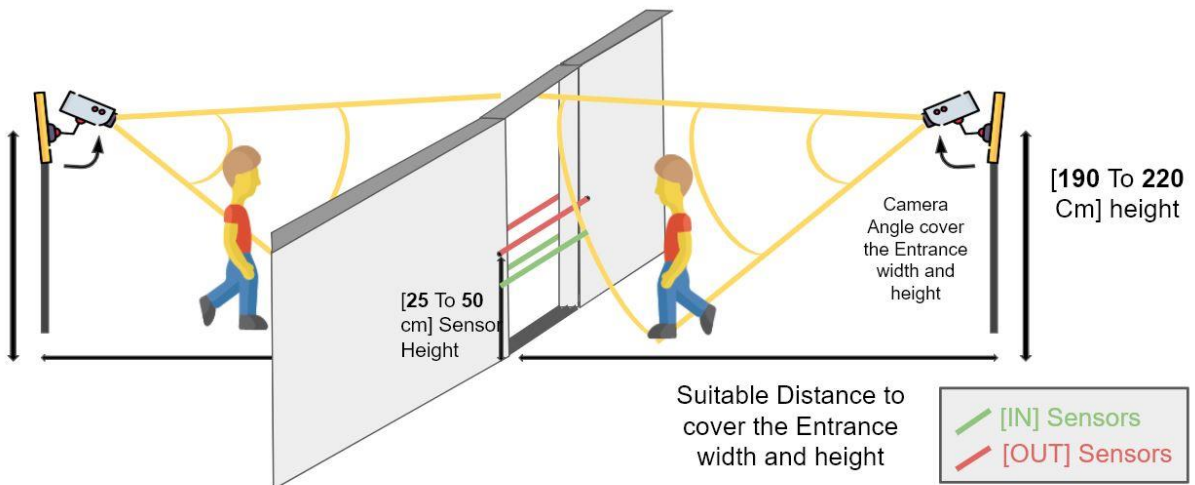


Figure 5 the final diagram of the two system directions.

Body-detect Test cases.

Several cases were tested in the Unum room of IT Carlow before the lockdown. The detection was working with an approximate 60% accuracy. After changing the photo format and the frame size the accuracy increased to 90% and in all cases, the bodies were detected successfully.

The first run of the system was using the PIR sensors which were taking a long time for the process to be completed (about or more than 10 seconds in total). The minimum delay time for the PIR is 3 seconds. This caused a problem as the person was not able to walk up to 4 meters within 3 seconds, resulting in that many persons were not counted as the system was not responding quickly enough.

The delay problem required a new IR beam breaker sensor, which had a shorter delay, with less than 2ms delay, and closer to the real-time detect. The IR sensors were tested, and the complete process was finished with 3 to 4 second which is was great and met the requirements.

The timeout has been assigned 1.5 seconds for each loop of the process to make sure that a person approaches the first sensor but does not continue to enter the door the system will ignore the person in question.

2.1.2. Face Detection approach did not work.

Face detection was the initial idea for the project, and the movement gets detected by the PIR. This approach was tested, and the results were not close to the requirements. The minimum delay time for the PIR sensors is 3 seconds which in real life is a long delay.

Several issues were founded during the testing of this approach:

- The face detection algorithm had low accuracy by not detecting all faces and sometimes consider some objects as faces which caused a big problem with the number of people detected as shown in **figure 6**.
- The PIR minimum delay time was 3 seconds, where the sensor holds its status and not switch before the 3 seconds which is a long delay, as the walking test was taking place and the adults can walk up to 4 meters within this 3 seconds.

The previous issues were causing a big problem with giving the wrong number of people and this is not what the project goals are.



Figure 6 miss detect for the faces in the picture with extra face.

2.1.3. Live Streaming Approach for future work.

Hardware Parts.

the parts used in this approach are:

- Raspberry Pi 4 with RAM 4GB LPDDR4-2400 SDRAM, 8 GB SD card.
- Live Streaming Camera.

Algorithm and software.

This software algorithm for this approach was built on the top of a **Pedestrian Detection** people counter, an open-source project that had been developed for other purposes. The algorithm has been modified and changed to fit the purpose of the EIH project. The original project was built to detect people walking in both directions of the road using one line down and cantered in the middle of the video and the increments will be depending on the object direction **figures 7**. The accuracy of this algorithm was less than 50% which caused problems counting, resulting in erroneous data.

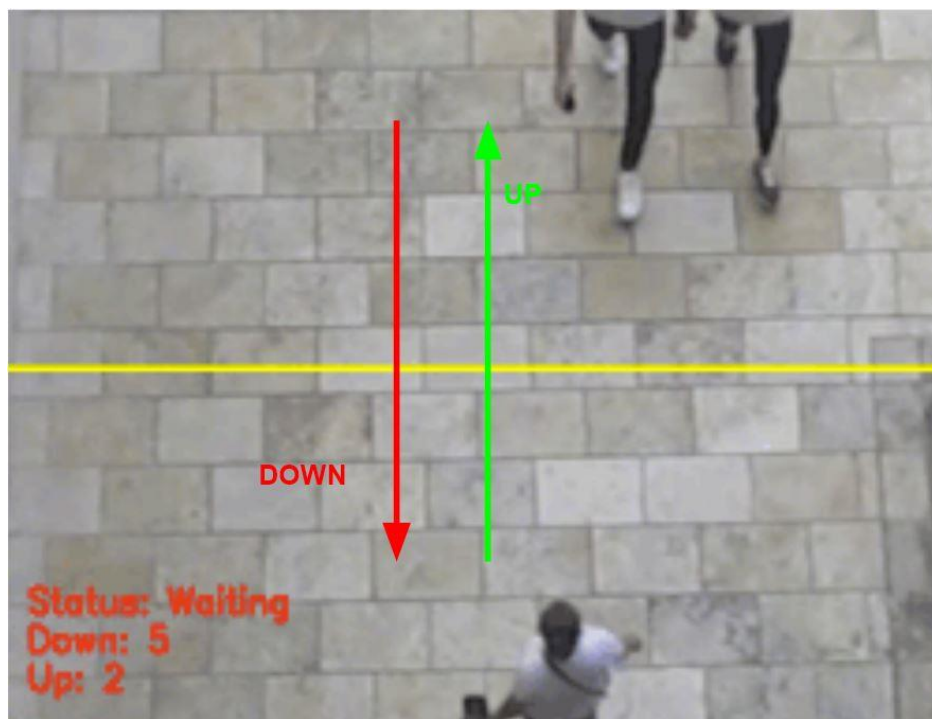


Figure 7 the original algorithm for the people counter

This algorithm was modified to increase performance and accuracy. The line was moved close to one end of the video frame **figures 8**, and take it to count only for one direction in addition to reduce the frame size for the processed video which gives the algorithm the ability to process

more frames in a shorter time. This change requires the double number of equipment working separately and update the same database. The modified algorithm was tested, and the accuracy was significantly increased to more than 75%.

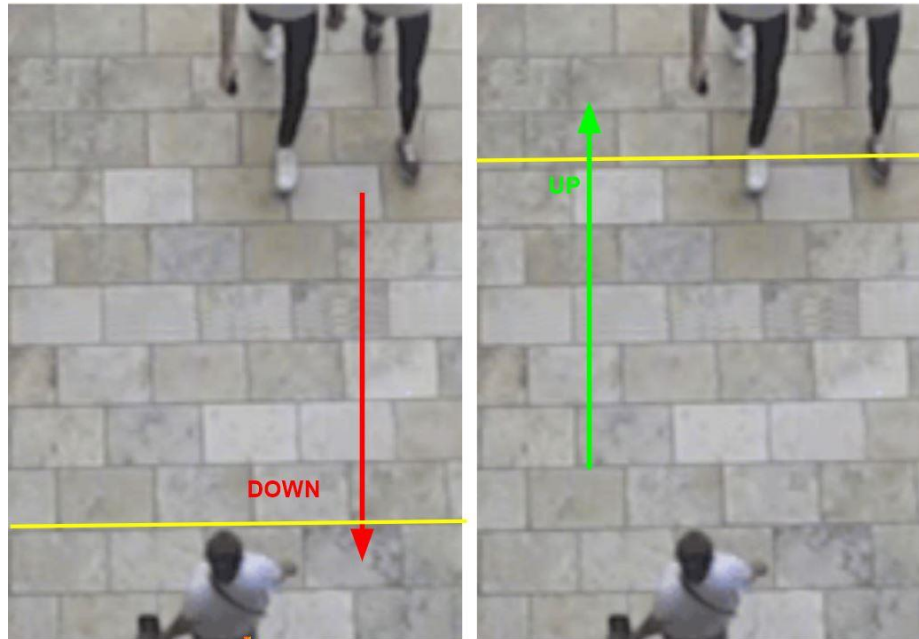


Figure 8 the new version of the People count algorithm

Future work.

As the COVID-19 pandemic happened and the governmental restriction was set during the development of this algorithm. This unfortunate situation did not allow more testing to be done and the development couldn't proceed, as one of the requirements is to set the camera in a public place with movement.

2.2. Front-End

Flask and jinja are the frameworks that had been used to design the front-end. In addition to other several programming languages that had been used to develop the project: Python, HTML, CSS, and JavaScript, with Google Maps API and Google Address autocomplete technologies. To design a simple and straight forward front-end website that provides data to the users at emergency services, the website was designed with all the mentioned frameworks, programming languages and Google maps technologies assuring that the users have all the functionality, performance, usability and reliability needed. Also, the users can reach all the time the information directly and in the fastest time. The screenshots below show all the front-end pages and describe the factuality for each page.

2.2.1 Website Screenshots

A. the Main page.

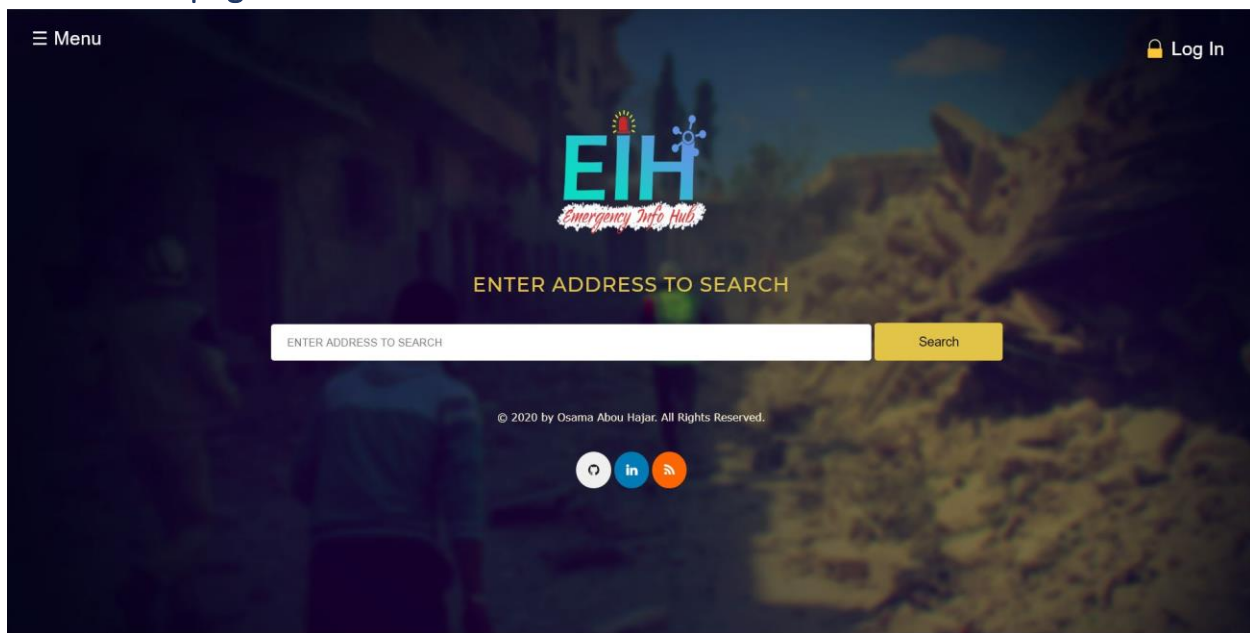


Figure 9 the front-end main page

This page **figures 9** Is the main page representing the project to the users. It had been designed with the lowest number of displayed elements as possible for simplicity. This screen points the users (emergency services mainly) directly to one input which is the targeted address.

The menu is on the top left-hand side of the screen shown on all the project pages. The menu is closed and can be accessed by the user's as needed, to not disturb the user during emergency time by losing focus. The picture below **figure 10** shows the menus screen after the user's clicked on the menu button.

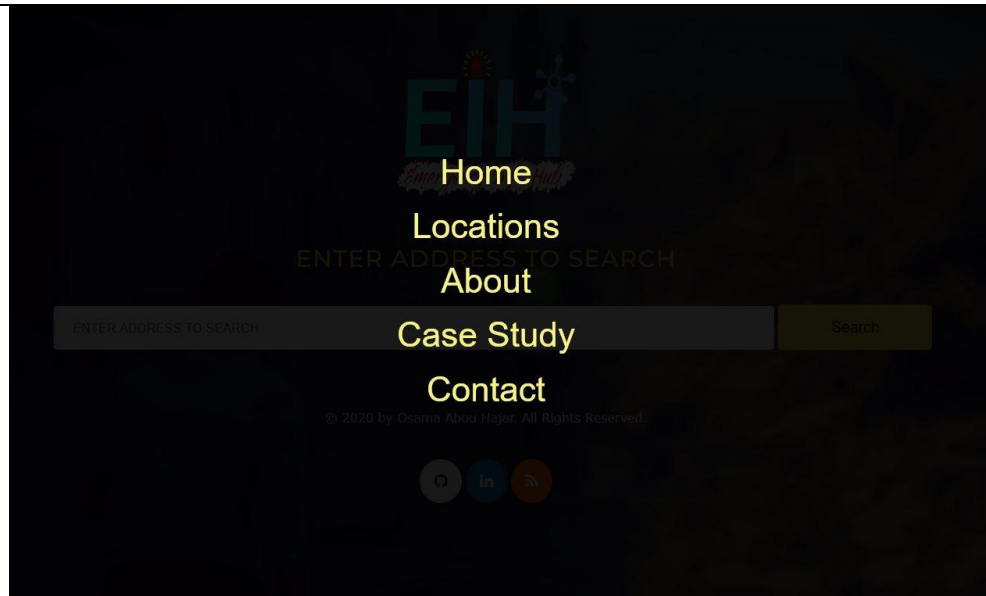


Figure 10 menu screen

The menu screen **Figures 10** has five options that will give the user the ability to choose between **Home**: directing the users to the main page of the project. **About**: gives the users more details about the EIH and the story of EIH. **Case Study**: direct the users to the showcase website where all documentations, guidance and basic information are provided. **Locations**: Shows the user a table with all the data and locations stored in the database and several options: add, delete, update and activate **Figure 16**. **Contact**: display a contact form to allow the users to send their feedback.

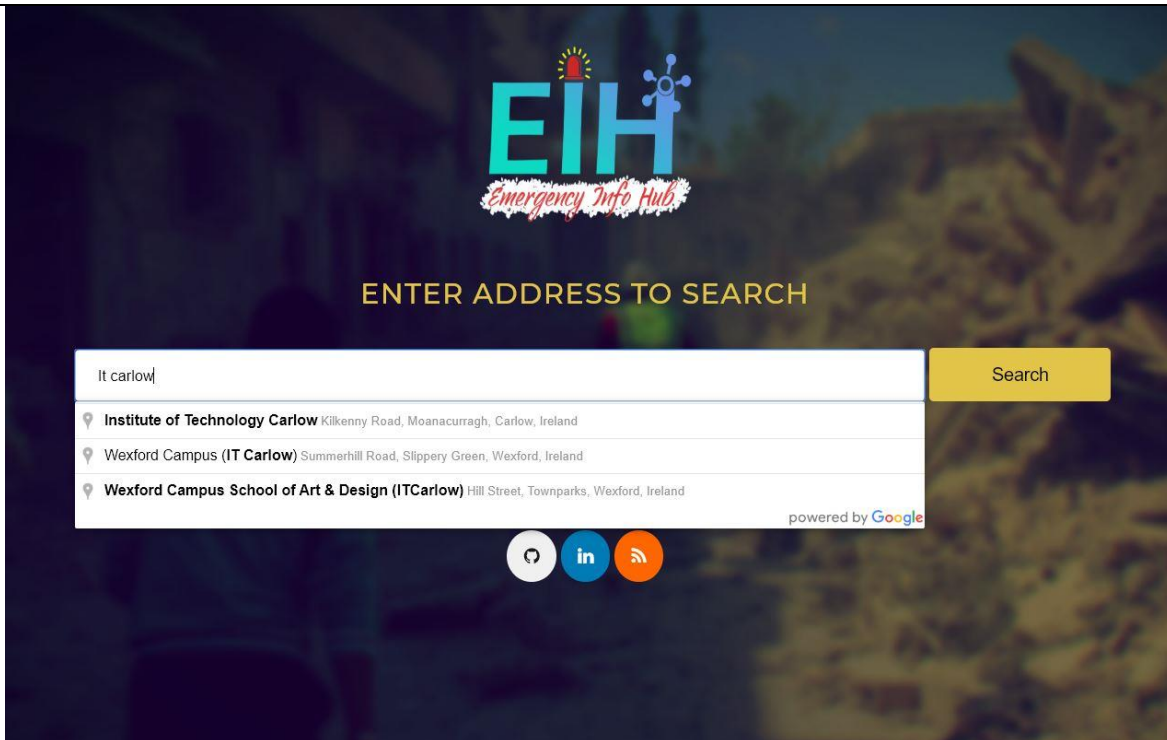
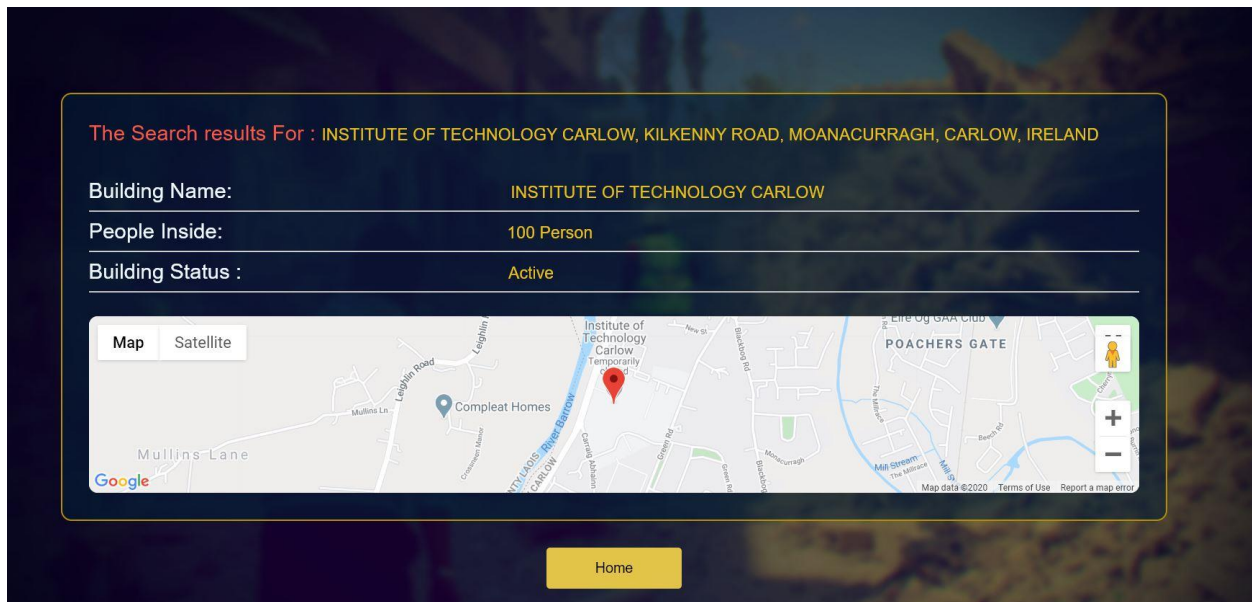


Figure 11 Address AutoComplete function

Address Auto-Complete is added to the search box on the main page as shown in **figures 11** real-time responds to correct and complete the user input address. This feature was implemented using Google Maps API, to save the time of users during the search.

B. the results page.



The second functional page in the project is the results page, shown after the users select the targeted address and click on the search button on the main page. This page displays the data coming from the database and show the real-time number of people inside the building. The page has only one button that redirects the user to the main page and displays the map for the targeted address as with the results.

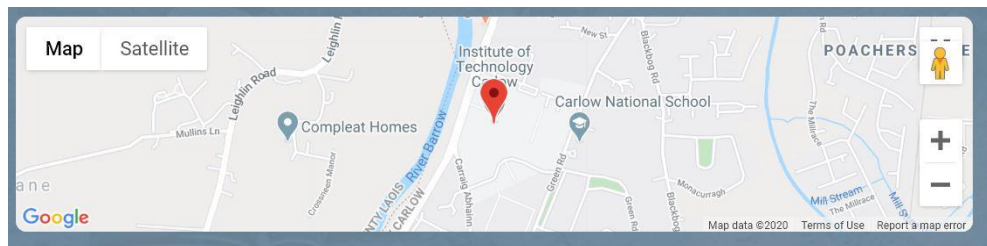


Figure 12 Google Map API

This function built using Google Maps API, where the targeted address will be sent to the API and the respond back from the API as it showed on **figures 12** A single click on the map box within the results page displays the full map for the around area. This will save time as no other external application is needed to get this information.

If the search address has no data in the database stored, the response message will be an alert box to the users that will inform the user that the address does not exist at this address. **figures 13**, This gives the user the ability to do another search in the shortest time, as they can perform another search from the same page.

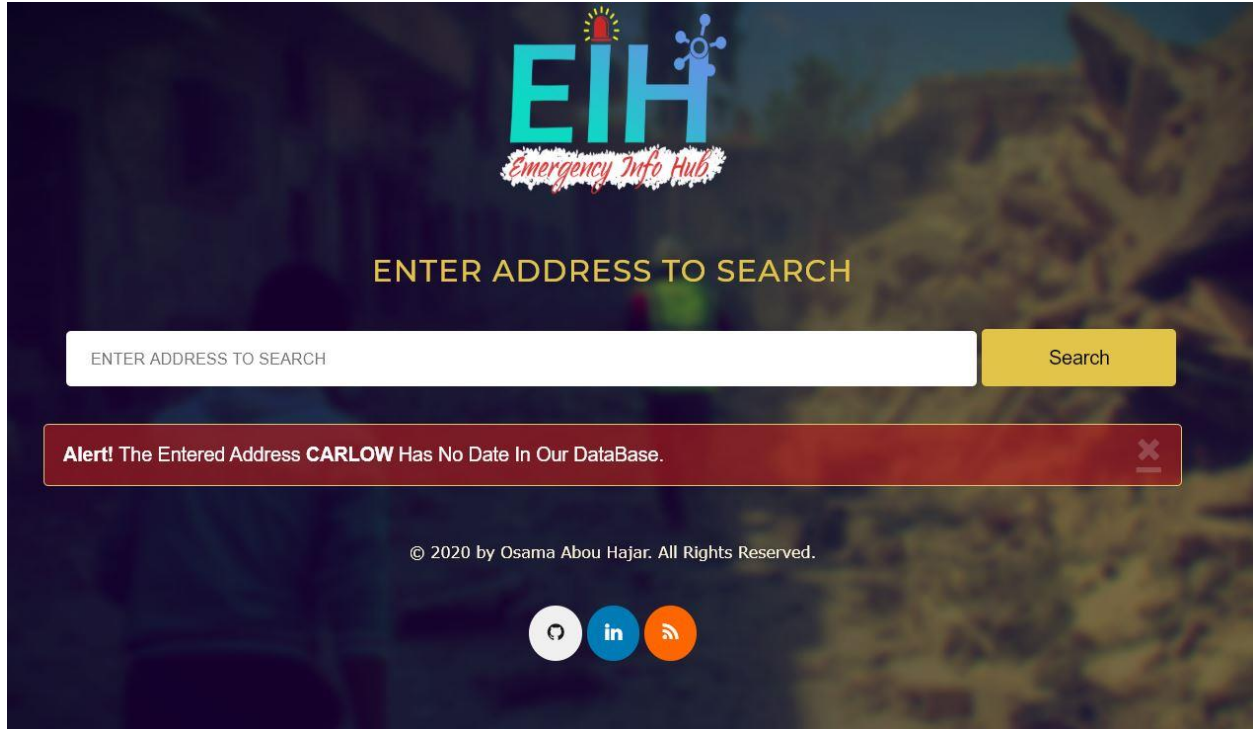


Figure 13 alert to show the search address not registered.

C. Login Screen.

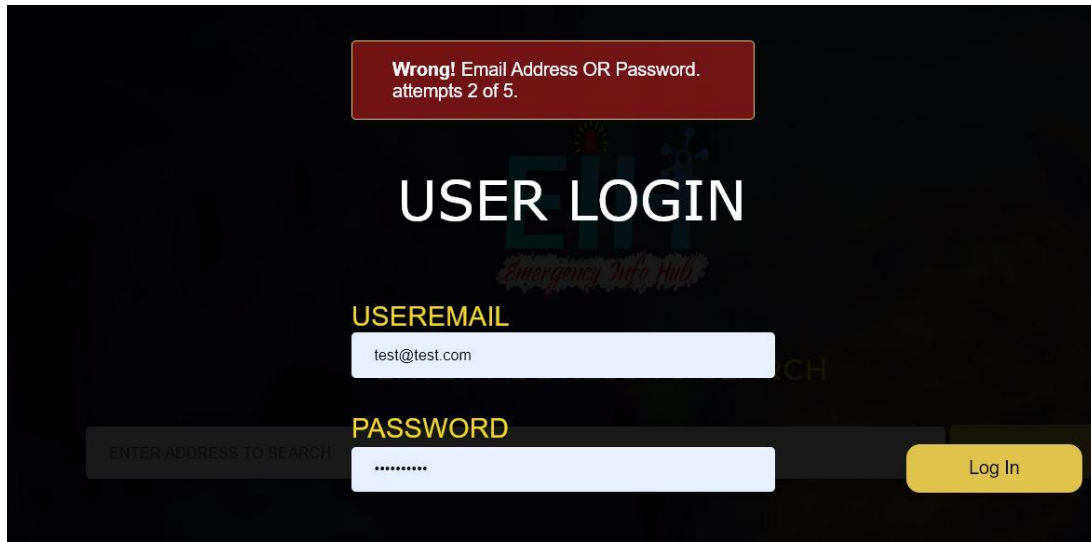


Figure 14 admin login Screen

The menu screen **Figures 14** take two user inputs, the user email and password once the admin login successfully the login page will redirect the admin to **All location** screen **figure 15**. to prevent the **brute force vulnerability** the users IP address will be locked out for 60 minutes after five failing attempts.

The registration from the front-end is not an option, to gain access to the website the administration office, should grant you with the login credentials, that by registering your email and password to the user’s authentications table in the firebase API side,

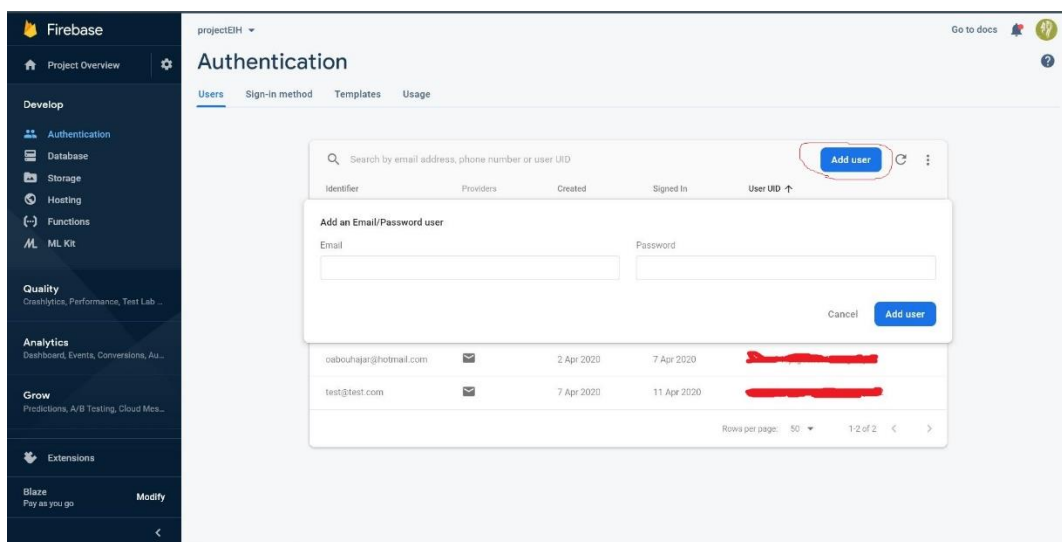


Figure 15 add new admin to the system

D. All Locations Screen.

This screen will display all the locations within the databases and give a fully dynamic table. The admins have full control with add, delete, update, and reset the number of people for each row **figure 16**.

* RIGHT CLICK TO SHOW THE OPTIONS MENU
* DOUBLE CLICK ON THE FIELD TO EDIT.

Search for names.. +/- New Row

#	Building ID	Building Name	Address	Eircode	Number Inside	Status	Last Update
	M4Eq45nYdyBOA8GgLf5r	INSTITUTE OF TECHNOLOGY CARLOW	INSTITUTE OF TECHNOLOGY CARLOW, KILKENNY ROAD, MOANACURRAGH, CARLOW, IRELAND	R93 V965	100	True	14:32:48 04/10/20 UTC
	M4FcSSKcym7LnVvHz_3	AVIVA STADIUM	AVIVA STADIUM, LANSDOWNE ROAD, DUBLIN 4, IRELAND	R93 V960	254	True	23:51:09 04/07/20 UTC
	-M4Fyj11wwWuu3jMXNHx	RDS MAIN ARENA	RDS MAIN ARENA, MERRION ROAD, BALLSBRIDGE, DUBLIN 4, IRELAND	R93 V960	78	True	20:32:36 04/06/20 BST
	M4FzQrd7V81_xOCSQBn	THE FOUNDRY NIGHTCLUB	THE FOUNDRY NIGHTCLUB, TULLOW STREET, GRAIGUE, CARLOW, IRELAND	TRUE	855	True	11:32:55 04/10/20 UTC
	-M4JyG7Cly7ZHJFEIC1P	CROKE PARK	CROKE PARK, JONES' ROAD, DRUMCONDRA, DUBLIN 3, IRELAND	R93 V960	85	True	14:09:00 04/07/20 UTC
	-M4KFFFv-VhrAHfn-gVf	DORA AND MIKE HOUSE	40 SAINT RIOCH'S COURT, KILKENNY, IRELAND	R95 Y802	554	False	15:30:44 04/07/20 UTC
	-M4MDIU4p1WQbx75gIF	DUBLIN ROAD GARAGE	DUBLIN ROAD GARAGE, DUBLIN ROAD, BALLYROAN, PORTLAOISE, COUNTY LAOIS, IRELAND	R93 V960	86	False	00:40:22 04/08/20 UTC

Note: An options menu is open over the first row, containing: Reset, Add, Delete, Activate, Update, +/- New Row.

Figure 16 the admin table menu

The above menu can be accessed only by the admin users. Public users won't be able to see the table menu and the Building ID as this is a private data **figure 17**.

Log In

Search for names..

Building Name	Address	Eircode	Number Inside	Status	Last Update
INSTITUTE OF TECHNOLOGY CARLOW	INSTITUTE OF TECHNOLOGY CARLOW, KILKENNY ROAD, MOANACURRAGH, CARLOW, IRELAND	R93 V965	100	True	14:32:48 04/10/20 UTC
AVIVA STADIUM	AVIVA STADIUM, LANSDOWNE ROAD, DUBLIN 4, IRELAND	R93 V960	254	True	23:51:09 04/07/20 UTC
RDS MAIN ARENA	RDS MAIN ARENA, MERRION ROAD, BALLSBRIDGE, DUBLIN 4, IRELAND	R93 V960	78	True	20:32:36 04/06/20 BST
THE FOUNDRY NIGHTCLUB	THE FOUNDRY NIGHTCLUB, TULLOW STREET, GRAIGUE, CARLOW, IRELAND	TRUE	855	True	11:32:55 04/10/20 UTC
CROKE PARK	CROKE PARK, JONES' ROAD, DRUMCONDRA, DUBLIN 3, IRELAND	R93 V960	85	True	14:09:00 04/07/20 UTC
DORA AND MIKE HOUSE	40 SAINT RIOCH'S COURT, KILKENNY, IRELAND	R95 Y802	554	False	15:30:44 04/07/20 UTC
DUBLIN ROAD GARAGE	DUBLIN ROAD GARAGE, DUBLIN ROAD, BALLYROAN, PORTLAOISE, COUNTY LAOIS, IRELAND	R93 V960	86	False	00:40:22 04/08/20 UTC
FACEBOOK	FACEBOOK, GRAND CANAL SQUARE, GRAND CANAL DOCK, DUBLIN, IRELAND	R93 V960	0	True	16:41:42 04/08/20 UTC
TESCO CARLOW	TESCO SUPERSTORE, CARLOW, IRELAND	R93 V960	0	True	15:22:39 04/11/20 UTC

Home

Figure 17 all locations screen for not admin users.

D.1. Add Building function.

The most important part of the project was to establish the connection between the hardware and the front-end: linking the device to the actual location. The location should be created on the front-end first with the building name, address (this will use the google address autocomplete) and the Eircode, **figure 18**, and the given Registration ID Key will be used on the hardware side to be export and give the hardware the access to any location **figure 19**.

#	Building ID	Building Name	Address	Eircode	Number Inside	Status	Last Update
ADD	Auto Generate	Building Name	Building Address	Eircode	Auto Generate	Auto Generate	Auto Generate
	M4Eq45nYdyBOA8GgL5r	INSTITUTE OF TECHNOLOGY CARLOW	INSTITUTE OF TECHNOLOGY CARLOW, KILKENNY ROAD, MOANACURRAGH, CARLOW, IRELAND	R93 V965	100	True	14:32:48 04/10/20 UTC
	M4FcSSKcym7LnVVHz_3	AVIVA STADIUM	AVIVA STADIUM, LANSDOWNE ROAD, DUBLIN 4, IRELAND	R93 V960	254	True	23:51:09 04/07/20 UTC

Figure 18 add building steps

Alert! the Building DUBLIN INSTITUTE OF TECHNOLOGY registration key is : '-M4e5FjAEcc3mHOZ5nC-' Add this key to your hardware to connect.

Figure 19 registration key for the added building

D.2. Edit Building option

This function will give the users the ability to modify the data stored already in the database. The data row is set to be editable; the admin can click on the field where the change should take place, then select the edited row and select update option from the menu.

```
data = {
    "known_name": request.form.get("known_name_update").upper(),
    "address": request.form.get("address_update").upper(),
    "eircode": request.form.get("eircode_update").upper(),
    "timeUpdated": time.strftime("%X %x %Z"),
}
```

D.3. Update Building option

To Reset a building, select the building row, and select the reset option from the table menu. The number of people inside the selected building will be reset to zero.

D.4. Delete/Activate Building option

The admin can set a building status to false or true which means this address system is active or not. This option may be used in different cases of the building is not using the EIH or at the time of maintenance.

The data will be affected by this option is the 'active' key in the data for the selected building.

Example:

```
{"active": False}
```

E. About Us Screen.

This screen has information about the project and external links to different resources and the project repository. In addition to EIH story will provide solid reasons why the project is needed.

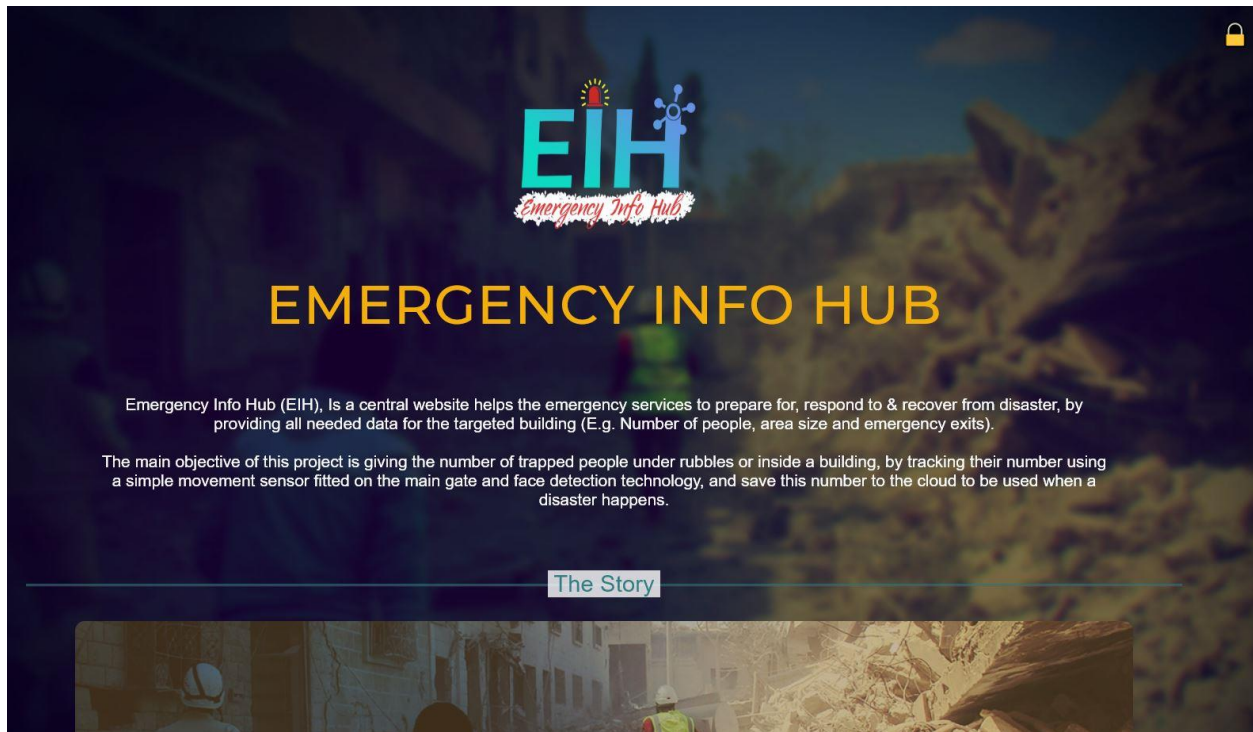


Figure 20 about us page

F. Contact US page.

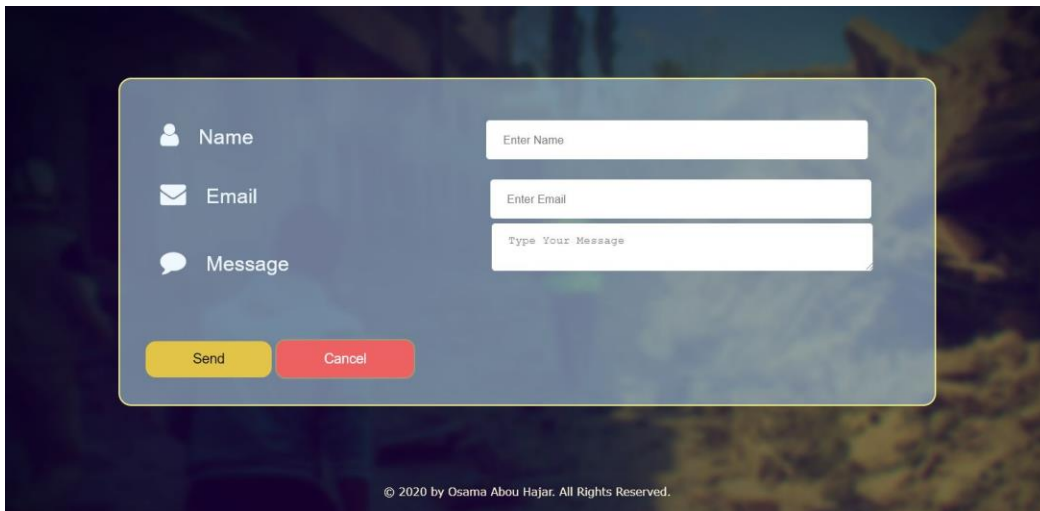


Figure 21 contact Us form

The contact form is the third page of the project. Its purpose is to give the users the ability to submit their feedback and report issues and bugs if found. The contact form is consisting of three parts: the sender name, email address and the message. The date and the time of the submitting will be auto-generated in the moment of submission. The page has two buttons **Cancel**: to stop the process and redirect the user to the main page of the project, **send**: to send the msg to the target destination and redirect the users to the confirmation page as shown in **figures 22**.

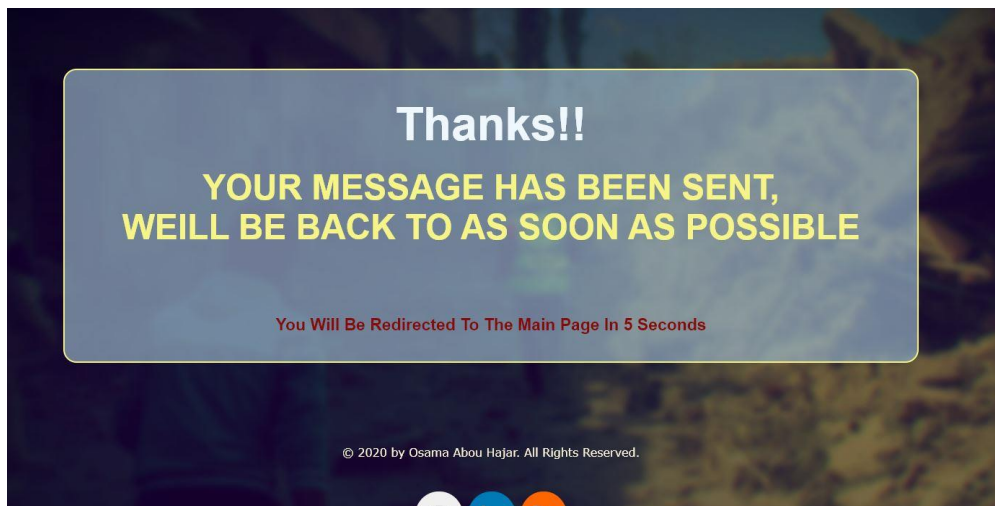


Figure 22 the confirmation page

The confirmation page displays a confirmation message to the users, and automatically redirect the users after 5 seconds to the main project page.

2.3. Database Structure.

EIH data stored as an API on the Google firebase real-time database.

The data is stored in JSON format as a dictionary pair using a key and a value. This allows more flexibility for updating information than the tabular databases.

```
data =
{
  '<BUILDING-ID>':
  {
    'active': '<BOOLEAN-VALUE>'
    "deviceId": '<STRING-SET-BY-THE-USERS-FROM-THE-BACK-END>'
    "known_name": '<THE-BUILDING-KNOWN-NAME>',
    "address": '<THE-ADDRESS-GENERATED-BY-GOOGLE-ADDRESS-AUTOCOMPLETE>',
    "eircode": '<THE-BUILDING-EIRCODE>',
    "numberOfPeopleINDetect": '<INTEGER-INITIALIZED-TO-ZERO>',
    "timeUpdated": '<AUTO-GENERATED-BY-THE-SERVER-TIME>'
  }
}
```

- **BUILDING-ID:** is auto-generated by the firebase with the push request, so when the user adds new building the data the key will be generated and associated with this submitted data.
- **ACTIVE:** Boolean data type, True to active and in work building, False for non-active or deleted address.
- **DEVICE-ID:** String data type, created by the user to differentiate between the hardware.
- **KNOWN NAME** is the known and public name for the address, which makes a better understanding by the reader than the proper address.
- **ADDRESS:** String datatype, the address will be autocompleted by the google autocomplete function, the address is considered the unique value to each address where the search will be depending on.
- **EIRCODE:** for the time of this project was done, the Eircode was not used by all the address in Ireland, which is hard to be used for the search and identity.
- **NUMBER-OF-PEOPLE-DETECTED:** this is the value where the number of people gets stored and updated, the initial value is zero when the building gets added.
- **TIME-UPDATED:** is the time when any changes happening to the data, this value will be assigned by the server time.

2.4. Development challenges.

2.4.1 Hardware challenges:

The hardware was the most challenging part for me, at the start of this project as I had never been working with this kind of device before. This is what makes the pre-plan for this project not well defined, working with the Raspberry Pi, the PIR sensors, IR sensors, and the camera module and make all this component linked and process in one system was a blind point for me. A simple question was in my head before the start of this project like how to read the data from the sensors? How to send this data to the cloud?

In the same area and during the test and the run time of EIH, several challenges appeared from at different stages:

People movement and sensors detection:

People are different, each has a different height, walking speed and walking habits. Several challenges appeared during the test and the run time for this project.

Figure 23 shows the challenge case where two people are within the sensor range, both walking in different directions. The camera will take the picture and send it to the detection algorithm to be processed.

The **face detection** algorithm was detecting only one face and did not catch the leaving one, causing one of the problems to use the face detection algorithm.

The **body detection** was able to detect the two people but without figure the direction for each of them.

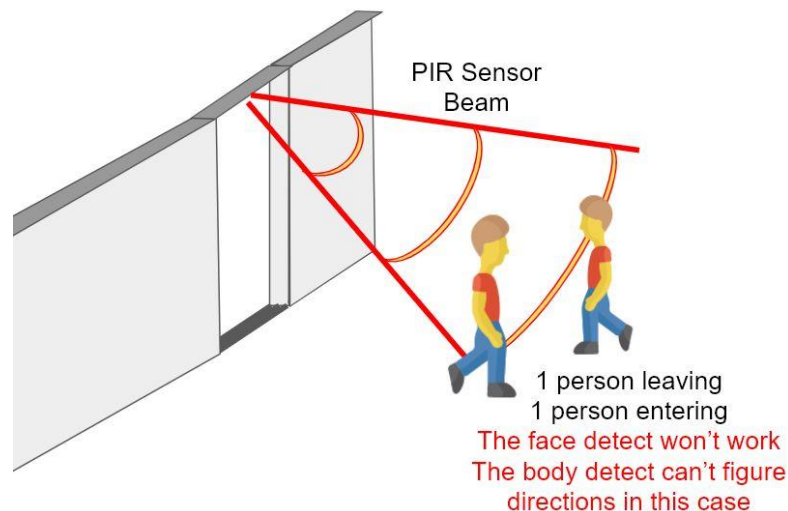


Figure 23 falling test case with the direction

Figure 24 shows the case where two people start having a chat within the sensor beam range, in this case, the people are seen but there is no confirmation if either any of two will enter the building.

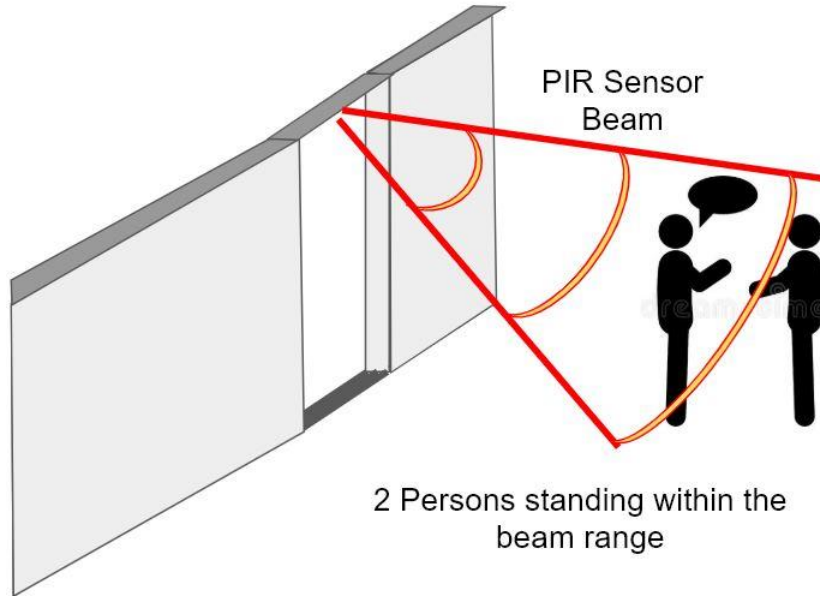


Figure 24 - 2 people chatting challenge

Figure 25 shows the case where a person walks toward the door then they change the direction without entering the room.

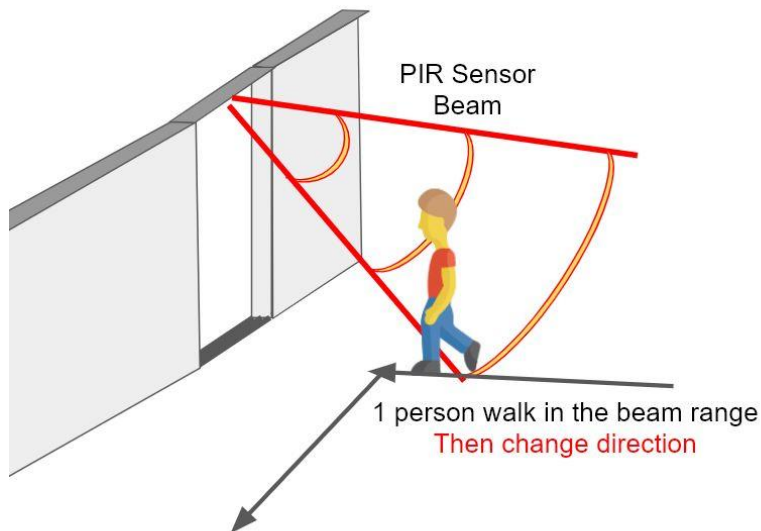


Figure 25 changing the direction

All the previous challenges were requiring the use of two sensors, a solution that was used in the final solutions of the project. Even with the two sensors some other issues were challenging and changing the development approach:

- Someone cut the first sensor beam and return without completing the process of sensor 1 then sensor 2.
- Using two sensors for both directions with cameras runs depending on the sequence of the sensors beam break **figure 26**, the approach failed many times as sometimes two people will cut the sensors together in the same instance of the time.

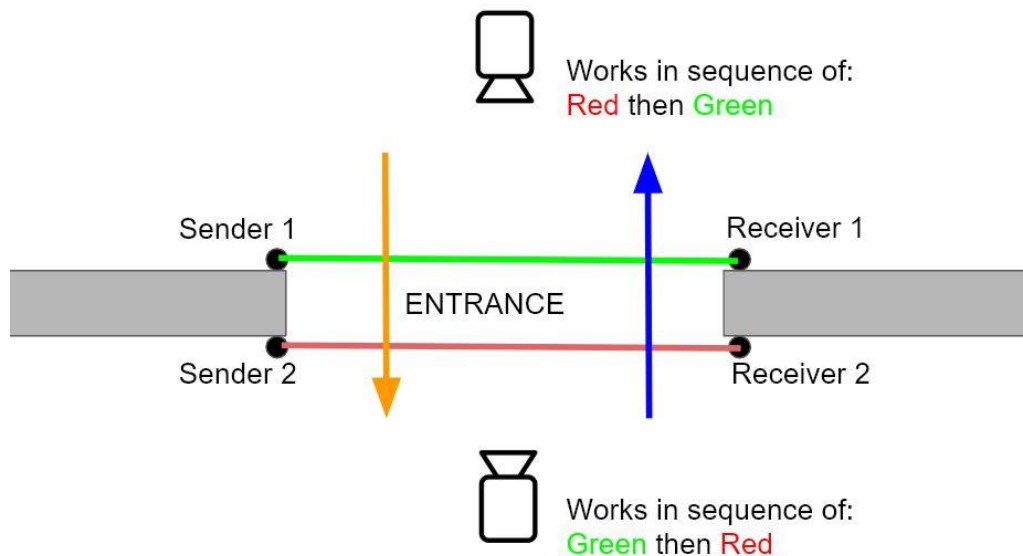


Figure 26challenging case with 2 sensors and 2 people

All the previous challenges and cases show, and with the supervisor advice, the new approach of using two compatible different systems each one working for one direction at the time as shown in **figure 27**.

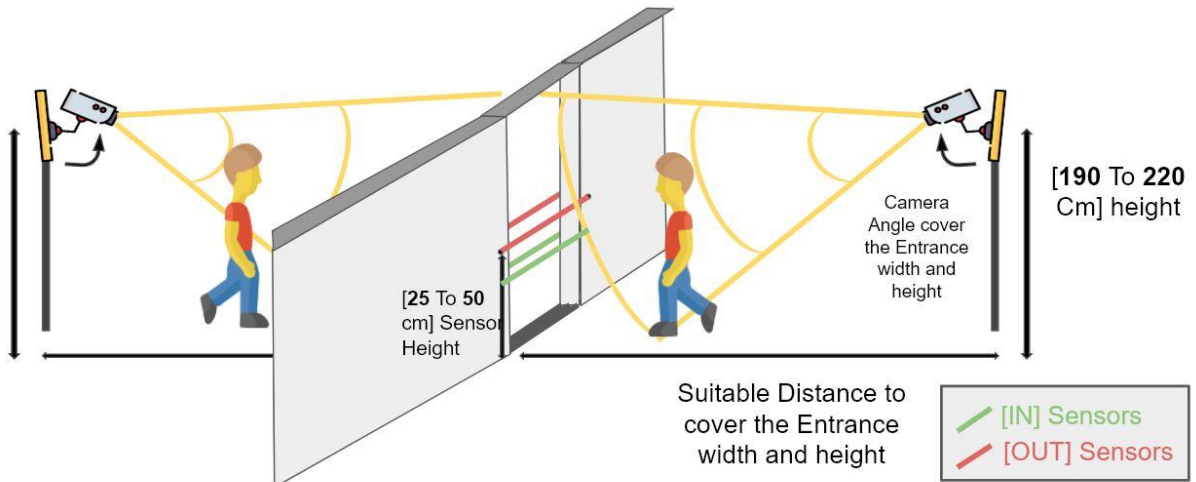


Figure 27 the final picture of the system used

2.4.2. Software challenges:

Alongside the hardware challenges, there were several challenges with the backend and the front-end software:

- Backend

The backend detection algorithm is requiring a high-performance device to process the images quickly and fast. Also, to all the requirements and dependencies.

- Front-end

While these numbers are representing real persons, all the data within the whole system should be considered as **sensitive data**. All the **security and website vulnerability** should be tested and prevented.

- All the connections were enforced to be over HTTPS not HTTP
- Flask security configurations were set to:

```

o SESSION_COOKIE_HTTPONLY= True
o ## Restrict how cookies are sent with requests from external sites.
  Can be set to 'Lax' (recommended) or 'Strict'
o SESSION_COOKIE_SAMESITE='Lax'
o ## The domain match rule that the session cookie will be valid for.
  If not set,
o ## the cookie will be valid for all subdomains of SERVER_NAME
o SESSION_COOKIE_DOMAIN = True
o SESSION_REFRESH_EACH_REQUEST = True
  
```

- In additions to set secure all the headers with the after-request call:

```
o @app.after_request
o def apply_caching(response):
o     response.headers['X-Frame-Options'] = 'SAMEORIGIN'
o     response.headers['X-XSS-Protection'] = '1; mode=block'
o     response.headers['Strict-Transport-Security'] = 'max-
age=31536000; includeSubDomains'
o     # response.headers['Content-Security-Policy'] = "default-
src 'self'"
o     response.headers['X-Content-Type-Options'] = 'nosniff'
o     return response
```

Establishing the connections between the hardware and the front-end was another challenge that needed a solution to make the hardware connected to an actual address and guarantee the number update.

Section 3: Python Unit Testing.

Python unit testing has been designed for the front end of this project; white box testing is the method used for this unit testing and it is a. The purpose White box (glass-box testing) is to test the input of this project and make sure any added code is executed at least once successfully, and all the frontend requests are loaded correctly. Where Black-box testing is done on the server-side.

The table below shows the White Box Test Inputs Generation

	Request Type	Branches Covered	Expected output
#1	Get request, /displayform, route	Index page load	200
#2	Get request, /allocations, route	all location page load	200
#3	Get request, /aboutus, route	About us page load	200
#4	Get request, /contactForm, route	Contact Us page load	200
#5	Get request, /logout, route	Logout redirect successfully	200
#6	Post request, / displayResults, route address = "Institute of Technology Carlow, Kilkenny Road, Moanacurragh, Carlow, Ireland"	Result Page load	True
#7	Post request, /login, route userlogin = "test@test.com" password = "noPassword"	Fail login and show alert Wrong! Email or password.	True
#9	-	Return the data from the database	Not None
#10	-	Check the database data type	Dict
#11	-	Configure the database successfully	Not None

Section 4: Learning outcome.

The completion of this project had a great impact on my experience in various fields of **modern technologies** that are used in many global projects used by large technology companies as discussed below. In addition to expanding all my previous experiences gained from previous studies, such as **electronics engineering** and **graphic design**, all combined to make the project integrated and easy to use.

3.1. Internet of things (IoT):

The Raspberry Pi is the mainboard used to connect the project components, before the start of the project I had no experience with such boards and devices. Working on this project gives me a good idea about how all these devices work and handle data and how data can be read from the devices to deliver it to the target. Working on the raspberry Pi had an impact on my experience by increasing my knowledge in the field of Internet of things such as its benefits and areas of its use.

3.2. Sensors:

During the work on this project, two types of sensors tested, the PIR sensor and the IR beam break sensors were under testing and development. Each of them has a different specification and they require a different approach to code development. Working with these sensors, reading their input data and sync that data with other components was a new learning experience for me. Right now, I do have a better understanding of a lot of different projects in our daily life using sensors and other components. This learning area opens a new idea for me for future projects with more input and output components involved.

3.3. Programming Languages

Python, Jinja2, Html, CSS, and JavaScript are the main languages used in this project, in general, my experience was developed and expanded within all the languages mentioned. Working with flask as a framework, implementing a new way of solving all the problems I faced, seeking help from different resources, working on a new way of code design patterns, using more functions, being more familiar with the use of git, able the security vulnerability and the final year college material during the work the greatest and the enjoyable part of the work.

3.4. Open-CV and object detection.

Body detect, face detection, face recognition...etc are the new terms we hear a lot in our daily life. In my project, I was able to learn about these technologies and use them. Open-cv is one of the biggest open-source libraries developed by intel. During the development, I was able to find some open source projects for face and body detection and live video detection. This project I used was developed by one of the professional Computer Vision/Deep Learning developers and researcher¹. This project needs to be fixed and improved to do exactly what I want for my EIH project. Working on such as this project had great learning within this field, exploring this area of images process, give a wider opportunity in with a wider range of projects.

3.5. Firebase and Google API.

Using firebase can be difficult to learn, but it's very useful to use. During the development I was looking for a database platformer to store the data for EIH, I was already decided to use a non-tabular database API and JSON data type getting more popular to use. Firebase has many products and one of these products is the real-time database, which stores the data in JSON formats, which what I was looking to store. JSON format gives my project the option of having a public API to be used by other developers they want to use my data for the same purpose.

3.6. Full-stack project

The term full-stack was always one of the terms I am asking myself a question about, how all the components from the end-to-end can work together? What about the latency between all this component? The performance? The functional and non-functional requirements?

All these questions were answered in addition to discover more useful information during the deployment. Working on the End-To-End project as much as it hard it's a great project where challenges and learning taking place. Doing a bit of everything over the hardware and the software increase my knowledge with different useful areas.

3.7. Reusing the previous experience.

Along with my programming experience, I have two-year prior experience in electrical/electronic engineering and around 6 years of graphic design experience. In the project I was able to refresh this experience by planning, implementing and creating new ideas to build the project in the best format in terms of the user experience.

3.8. Personal Learning.

In the completion of this project, I got the chance to develop my learning and skills in different areas.

- Public speaking and communications, by reaching a lot of people to collect useful information helps with the project plans and goals.
- Time management was one of the main challenges for me as the project is a self-development process that requires allocating daily, and a weekly number of hours to work on the project.
- Identify tasks and links, and schedule to achieve goals.
- Breaking down the project into small tasks and keep tracking of each task to deliver the product at the deadline time.
- Agile development process experience with teamwork, weekly meeting and early feedback from my supervisor.

In general, this project has allowed me to use the experiences and information that I studied over the 4 years of my software development course and apply them to real-life projects.

Section 5: Project review and Conclusion.

In general, this project was one of the most difficult projects I did during my studies or work placement within the software development area in terms of size and new ideas that I had never implemented before.

Many new aspects and challenges require a lot of research and experience, which made many points of the project subject challenge and fun. Through research and work on these challenges, I was able to have a new experience to be able to start a smarter and larger project that helps to develop my mental and practical consultations.

Besides, the effort in this project was to provide assistance and protection to the largest number of people who lose their lives and their loves daily as a result of wars, crises, and humanitarian disasters in many countries around the world, especially the Syrian war, which was the inspiration for the idea of this project after losing beloved close people.

4.1. What went wrong?

While working on the project, there were several points initially agreed to be invalid after testing it. Face detection was one of these points that were picked from the beginning without looking at other available algorithms that were more efficient for use. The time that was invested on the PIR sensor due to the error in reading the minimum value of the delay time that made me lose the long-time changing the codes repeatedly without reaching the desired goal to find out later the misunderstanding of the value reading

4.2. Future work?

As mentioned in the approaches section above, the direct detection algorithm that I already spent many hours developing and working on its efficiency to be appropriate to the objectives of my project, will be one of the future steps that I am looking forward to completing when Corona Virus lockdown will end and the life will go back to normal. I wish I will have the chance to set up the camera in a public place where a large number of people will interact with the algorithm allowing me to perform more testing and improve the quality of the project.

Implementing the **Valuation Office and Building Control Management System (BCMS) API** to the project, where these APIs provide more details to the emergency services such as area size, levels number, and the emergency exits plan map. During the research, I have been in touch with the two offices and I couldn't get access to these APIs. The **Valoff.ie** website does already publish a public API provide some data about the building's levels and area size but the **BCMS** does give access only for the Governmental bodies.

Also, I would like to do more research on Blockchain and its usability with this project, where it can make the data exchange between the emergency services more secure, easy to reach and wider range update.

Acknowledgements

First and foremost, I have to thank my project supervisors, **Dr. Paul Barry**. Without his assistance and dedicated involvement in every step throughout the process, this project would have never been accomplished. I would like to thank you and all the **lecturers** very much for your supports and understanding over these past four years.

I would like to thank, **Adrian Rosebrock**, for open-cv technology tutorials, which was a great resource to learn about the body and face detect from his website, www.pyimagesearch.com.

Getting through my project required more than academic support, and I have many, many people to thank for listening to and, at times, having to tolerate me over the past four years. I cannot begin to express my gratitude and appreciation for their friendship. **Dora, John, Grace, Brendan, Sam, Ana, Martin, Rhyder, and Holly**. You have been unwavering in your personal, professional and English language support during the time I spent at the **Institute of Technology Carlow**. For many memorable days in and out.

Finally, my deep and sincere gratitude to my family for their continuous and unparalleled love, help and support, none of this could have happened without **my family**, I am forever indebted to **my parents** for giving me the opportunities and experiences that have made me who I am.

Plagiarism Declaration

Declaration

- I declare that all material in this submission e.g. thesis/essay/project/assignment is entirely my/our own work except where duly acknowledged.
- I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams or other material; including software and other electronic media in which intellectual property rights may reside.
- I have provided a complete bibliography of all works and sources used in the preparation of this submission.
- I understand that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offense.


Student Name: Osama Abou Hajar

Student Number: C00220135

Signature:

X

Osama Abou Hajar



Date: 20/04/2020