DUBLIN INSTITUTE of TECHNOLOGY

Institiúid Teicneolaíochta Bhaile Átha Cliath

Room-Occupancy-Measurement using Image Processing
Final Year Project Report

DT211C
BSc in Computer Science

Michael Kane

Dr Yupeng Liu

School of Computing

Dublin Institute of Technology

13/04/2018

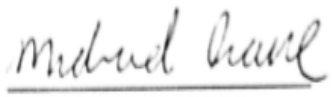
Abstract

The purpose of this project is to count the number of people in a room using image processing techniques. This application could be used to monitor a library checking the number of people in the room against its capacity. The reason for monitoring them could be for example to detect if there is free space for another person to occupy. To detect people in a room, a raspberry pi NoIR camera will be used. This camera will be combined with super bright infra red LED's to shine light on the room making people visible. Once an image has been captured, the result will be posted to a web server. The web server will display the image of the room and the number of people in the room. The image and result posted to the server will be stored in a database.

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:



Michael Kane

13/04/2018

Acknowledgements

I would like to thank Dr Yupeng Liu who is my project supervisor for all the support he has provided, I would also like to thank Dr Damon Berry who allowed me to use this idea for my forth year project and the support he provided, to thank Dr Jane Courtney who assisted me in the image processing aspect of this project and to thank everyone in both the School of Computing and Electrical Engineering who helped me with my project. Finally I would like to thank my friends and family especially my fiancée Elizabeth for all the help and support over the years.

Table of Contents:

1. Introduction....10
1.1 Overview of the project and the background behind it....10
1.2 Project Objectives....10
1.3 Project challenges....11
1.4 Structure of the document....11
2. What research has been done and what are the outputs....12
2.1 Background research....12
2.1.1 Digital Image Processing....12
2.1.2 Infrared VS digital camera....12
2.1.3 Segmentation....14
2.1.4 Flesh Detection....15
2.1.5 Contouring and object Detection....16
2.1.6 Haar cascades....16
2.1.7 Conclusion....17
2.2 Alternative existing solutions to the problem you are solving....17
2.2.1 Human Skin Detection by Visible and Near-Infrared Imaging....17
2.3 Technologies Researched....17
2.3.1 OpenCV VS Mat Lab....17
2.4 Resultant Findings / requirements....18
2.5 Bibliography....18
3. Design....19
3.1 Identification of design methodology including why it was chosen....19
3.1.1 What is agile scrum....19
3.1.2 Why use agile scrum....19
3.2 Design of each of the project components....20
3.2.1 Overall design approach....20
3.2.2 Raspberry pi....22
3.2.3 Web Server....25
3.2.4 Image Processing application....25

3.2.5 MySQL.....25
3.2.6 Upload and Gallery HTML pages.....26
3.3 Define a list of features and use cases supported within the project.....27
3.3.1 List of Features.....27
3.4 Other Design Documents.....28
4. Architecture and Development.....30
4.1 Overview of the system architecture and key elements.....30
4.1.1 What the system is now.....30
4.2 Details of each Component.....30
4.2.1 Person Detection algorithm.....30
4.2.2 Raspberry pi.....31
4.2.3 Web Server.....31
4.2.4 Storing data.....31
4.2.5 Accessing the Web Server and Client.....32
4.2.6 Security.....32
4.3 Development of each component, prototypes and solution created.....32
4.3.1 Problem and solution created.....32
4.3.2 Creating the prototype camera.....32
4.3.3 Creating the Web Server.....33
4.3.4 Setting up domain name and hosting.....33
4.3.5 Creating the database.....33
4.3.6 Creating the client.....34
4.4 Identification of external API'S.....34
4.4.1 JQUERY.....34
4.4.2 Cycle2.....34
5. System Validation.....34
5.1 Testing.....34
5.1.1 Testing Outputs.....34
5.1.2 Regular Expressions and test functions.....34
5.1.3 Unit Testing.....35

5.1.4 Usability testing....35
5.2 Demonstration....35
5.2.1 Demonstration of the application....35
5.2.2 Demonstration of the gallery....37
6. Project Plan....37
6.1 What was the plan in the beginning and how it changed....37
6.1.1 Interim plan....37
6.1.2 Final plan....37
6.1.3 What had changed and why....37
7. Issues and Risks....38
7.1 Issues....38
7.1.1 Prototype Camera....38
7.1.2 Person detection using infrared....38
8. Future work....38
8.1 Planned future work....38
8.1.1 Creating the prototype camera....38
8.1.2 Remote Access....38
8.1.3 Enhancing the algorithm....38
8.1.4 Usability testing....38
8.1.5 Creating unit tests....39
9. Conclusion....39
10. Bibliography....39

Table of Figures:

Figure 1. Project Architecture.....10

Figure 2. Image Denoising.....12

Figure 3. Fingers detected by infrared camera.....13

Figure 4. Standard picture left Face detection image right.....14

Figure 5. Standard picture left Face detection image right.....14

Figure 6. Left side Light skin Right side dark skin.....15

Figure 7. Left side Light skin Right side dark skin.....15

Figure 8. Object detection using contours and Haar cascades.....16

Figure 9. Haar cascades on single and group photograph.....17

Figure 10. Haar cascades on single and group photograph.....17

Figure 11. Week 12 sprint.....20

Figure 12. Low level diagram of architecture.....21

Figure 13. More detail into the architecture.....21

Figure 14. Prototype Camera.....22

Figure 15. Motorised Prototype camera.....23

Figure 16. Daisy chain solution to infrared LEDS.....24

Figure 17. System overview.....25

Figure 18. MYSQL ERD.....26

Figure 19. Early concept of the image gallery.....26

Figure 20. Use Case.....27

Figure 21. Interim Class Diagram.....28

Figure 22. Final Class Diagram.....28

Figure 23. Directory Structure.....29

Figure 24. Gantt Chart.....29

Figure 25. More detail into the architecture.....30

Figure 26. Breadboard Prototype.....32

Figure 27. IR LEDS.....33

Figure 28. Flesh Detection on Light skin.....35

Figure 29. Flesh Detection on Light skin.....35

Figure 30. Flesh Detection on Darker skin tones.....36

Figure 31. Flesh Detection on Darker skin tones.....36

Figure 32. Single Person VS Group.....36

Figure 33. Single Person VS Group.....36

Figure 34. Gallery.....37

1. Introduction

1.1 Overview of the project and background behind it

The main application of this project is to monitor a room for example a library and post the results to a webpage where the image and information about the room is displayed to the user. The application will take the image and using image processing techniques, detect how many people are in the room.

1.2 Project Objectives

The objective of this project is to create a System which will display to the user the occupancy of a room for a specific date and time, this is achieved by completing the below objectives.

- Create a python flask server to host the website, the server will contain a login, logout, register , upload and gallery page. The upload and gallery pages are allowed to accept post requests and read from the MYSQL database.
- Create a MYSQL database with two tables, the first will save the username and passwords, the second will store the paths to the images which are stored on the server.
- Create an image processing client on a raspberry pi to find a person in an image using a combination, thresholding, segmentation, modification of colour spaces and haar cascades.
- Create a domain name to host the website on the internet.

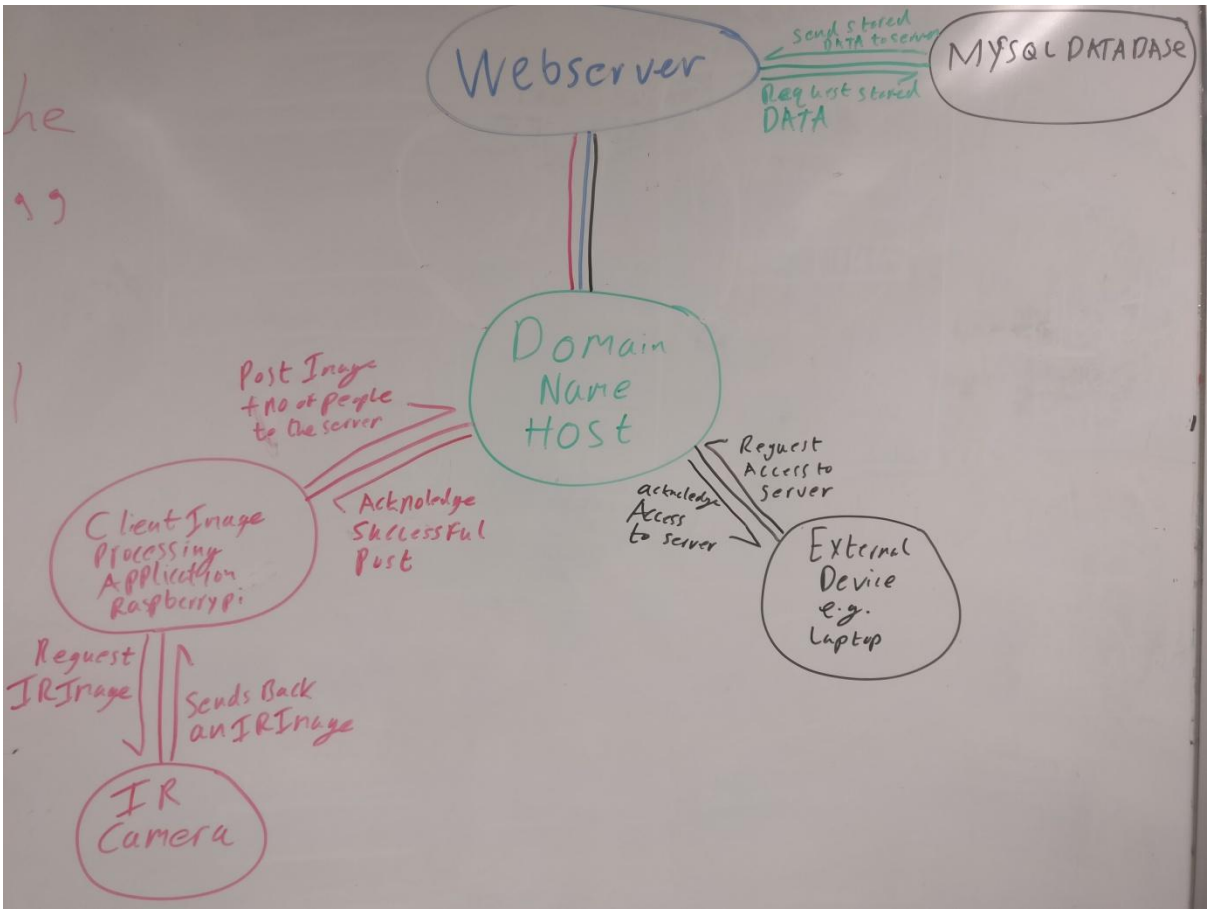


Figure 1 Project Architecture

1.3 Project Challenges

The main challenges for this project came from the image processing aspect of the assignment. The first attempts at the program to detect people had issues distinguishing between hands and facial features. This was eventually overcome through the use of haar cascades which will be discussed later. Other challenges faced are listed below:

- Creating a lighting system to project ultra violet light for the infra red camera to detect people in low light environments.
- The application will take a picture and send it to the web server every ten minutes, running the application for a large amount of time will require a large amount of storage space required to store the images.

1.4 Structure of the document

The structure of the document is as follows:

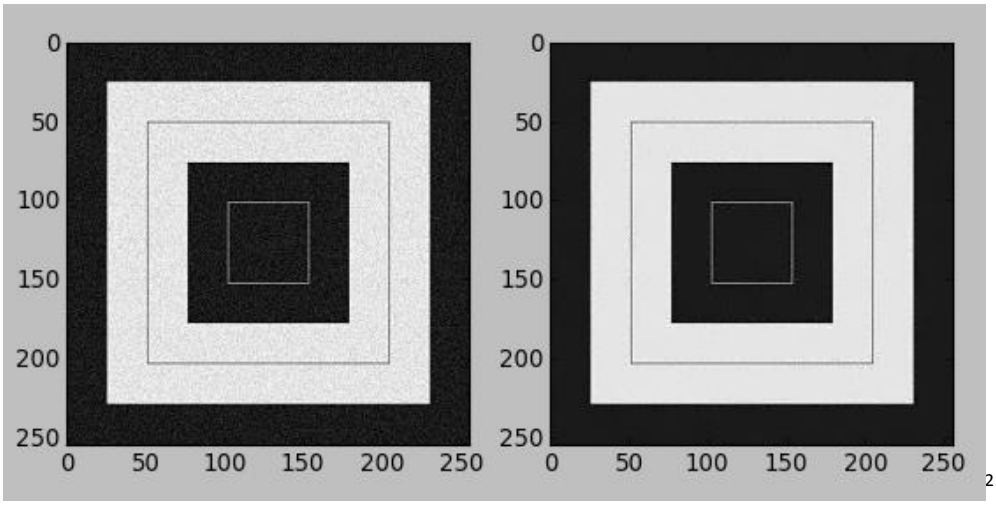
- The research into the image processing side of the project and web server.
- Design and methodology used.
- Architecture and solution.
- Testing and demonstration of the project.
- Comparing the project to the interim and what has changed.
- Results of the solution.
- Issues and risks.
- Future work.
- Conclusion.

2. What research has been done and what are the outputs?

2.1 Background research

2.1.1 Digital Image Processing

Digital Image processing is the ability to manipulate images through the use of a computer. The purpose of digital image processing is to create applications to generate a specific output on an image. There are some elements which can affect the quality of an image or video such as noise or blurring. These elements can be found in old images or videos and a user can use image processing to reduce or even remove them.[1]



[2]

Figure 2 Image Denoising

2.1.2 Infrared VS digital camera

Near infrared (IR) imaging can be captured using a camera with the IR filter removed. IR works on a different visual spectrum then we as humans can see. Clement Fredembach defines the wave lengths of infrared as follows "*The near-infrared spectrum is located just after the red wavelength and comprises wavelengths that range from 700 to 1100 nanometers*" This means that the infrared wavelength is outside of a humans visual spectrum. [3] IR LEDS can be used to illuminate dark areas with the area only being visible to an IR camera, as the IR wavelength is out of our visual spectrum. The IR camera also has the ability to take daytime photos, these images appear drastically different due to the way IR wavelengths interact with the environment.[4]

[1] https://www.tutorialspoint.com/dip/
[2] http://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_Image_Non-local_Means_Denoising_Algorithm_Noise_Reduction.php
[3] http://ivrl.epfl.ch/research/infrared/imaging
[4] C. Fredembach and S. Süsstrunk, Colouring the near infrared, Proceedings of the IS&T 16th Color Imaging Conference, pp. 176-182, 2008.

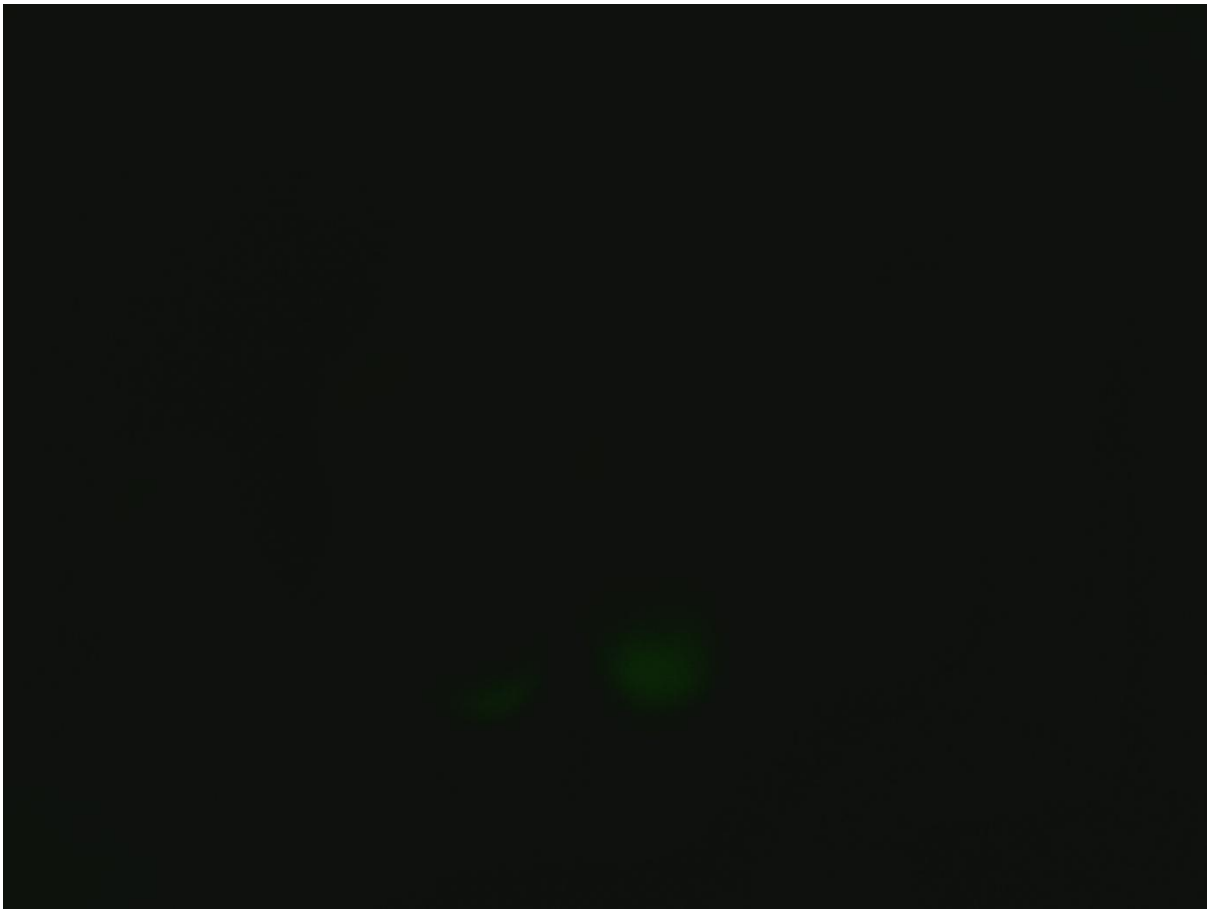


Figure 3 Fingers detected by Infrared Camera

Digital images can be taken with an ordinary camera; the light used to capture these images is in our visual spectrum. Meaning to capture day and night images we need a light source to illuminate the area E.G. the sun or a lamp. The reason to keep the IR filter on the camera is to prevent the IR radiation from interfering with the quality of the image as Clement Fredembach says here "*digital cameras' sensors are so sensitive to near-infrared that a "hot mirror", a filter that lets only visible light pass, is placed in front of them in order to prevent NIR contamination of the visible signal.*".[5] Digital cameras with an IR filter provide a higher quality image during daylight, and images can appear blurred in IR daylight images depending on the lighting conditions.

[5] http://ivrl.epfl.ch/research/infrared/imaging

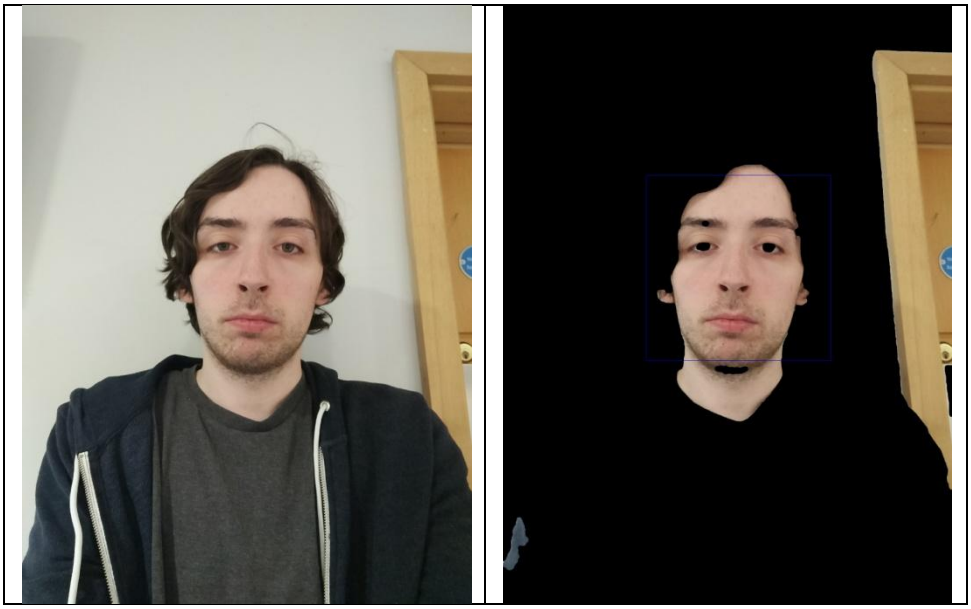


Figure 4, 5 Standard picture left, Face detection image right

There are two applications for the camera chosen to be used in. These are monitoring an elderly or disabled person in their home and detecting the occupancy of a room in the college to see if its free or occupied. For the applications to perform acceptably, the camera must be able to function is daylight, lowlight and dark conditions. The digital camera has the ability to produce high quality images in any light, but the issue is having to shine a bright light onto an area 24 hours a day in the case of monitoring a person. The advantage of using an IR camera in this situation is that the IR LED light is invisible to the human eye. While the room is dark for humans the room will be bright enough for the IR camera to pick up. The goal of this application is to detect a person, and the IR camera is in my opinion the best choice as it can provide an image where people can be segmented out and located using object detection.[6]

2.1.3 Segmentation

Image segmentation is the splitting an image into individual pixel values then transforming the image into something new. Examples of this can be done by reducing noise or extracting certain object or elements from the image. There are two examples of Segmentation these are thresholding and morphology. Image thresholding is converting a grayscale or coloured image to a binary mask. There are two types of thresholding, there's thresholding and adaptive thresholding. The difference between the two is that the adaptive thresholding will try to find the average value of the pixels and apply the answer as the binary mask. " *It assumes that (i) each peak coincides with the mean grey level for all pixels that relate to that peak and (ii) the pixel probability decreases monotonically on the absolute difference between the pixel and peak values both for an object and background peak.*"[7]

Morphology is the use of morphological transformations which are algorithms, to alter an image based on its shape. These algorithms are performed on binary masks which can have jagged edges or spots of noise in the background, and can be used to extract and clean the overall look of an image.

[6] https://pimylifeup.com/raspberry-pi-camera-vs-noir-camera/
[7] https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic3.htm

There are four main types of morphological transformations and these are Erosion, Dilation, Opening, and closing. Erosion and Dilation are opposites of each other; Erosion will shrink the white areas of an image while also removing noise. Dilation increases the white in an image and is usually paired with Erosion to return the image to its previous size. Opening is erosion followed by dilation, it's a algorithm which combines Erosion and dilation. Closing is Dilation followed by Erosion, this is useful for cleaning the objects in the image itself.[8]

2.1.4 Flesh Detection

One of the biggest challenges in this project is to detect a person. With a regular camera a user would have the ability to use colour to segment out a person from a background. With an IR camera it's based on what level wavelength an object such as skin will absorb light. Describes the wavelength where human skin is detected is 970nm while clouds and the sky are detected at 940nm. This means that the clouds and sky can cause interference with skin detection. Their solution to this problem was a 6 band camera which is a combination of a VIS camera and NIR(Near infrared camera) which is out of the scope for my prototype and capabilities.[9] Below is an image taken from the running project, we can see that the program is finding a variety of skin tones.

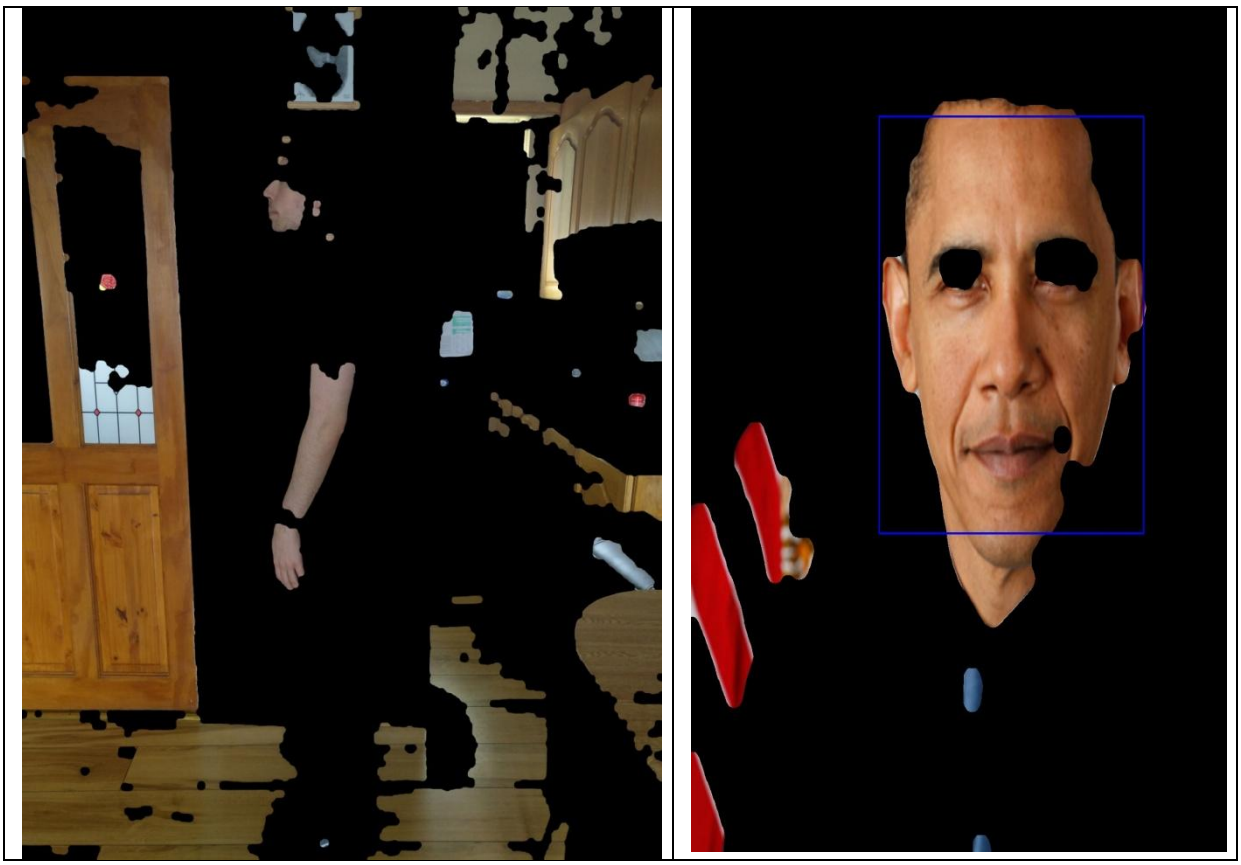


Figure 6, 7 Flesh Detection: Left side Light skin, Right side Dark skin

[8] https://docs.opencv.org/trunk/d9/d61/tutorial_py_morphological_ops.html
[9] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.389.6745&rep=rep1&type=pdf

2.1.5 Contouring and object Detection

Contours in image processing are a way of joining similar areas of interest. Contours are a great way of finding a particular object.[10] A list of contour object can be created on a binary mask for a specific region of interest. From this list of objects a specific object can be chosen by a developer to be displayed to the user. In the below image we can see the use of contours on one of the images taken. To find contours a binary mask must be created of the region of interest i.e. what the user wants to see. In the below image I wanted to segment out the face by itself, but due to the algorithm also picking up other colours related to skin tones it made it difficult to isolate a specific contour. The solution was to combine the current algorithm with Harr cascades to detect a face within the altered image, which can be seen working due to the blue box surrounding the face.

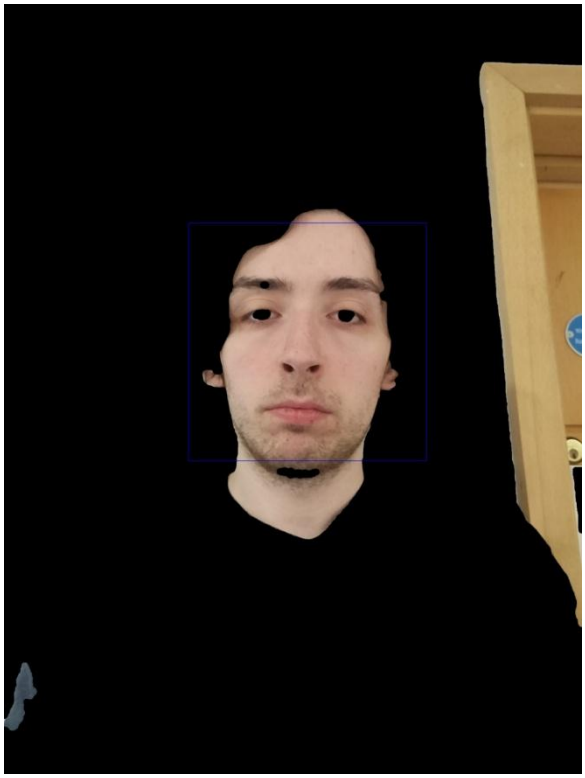


Figure 8 Object Detection using contours and Harr Cascades.

2.1.6 Haar Cascades

A Haar cascade is a machine learning technique where a program reads in a Harr cascade classifier to identify objects in images. A classifier is an xml file with lots of images with faces and without faces to train the classifier to recognise faces within an image. This project will be using a prebuilt classifier to detect faces. Haar cascades can be used to find a single face or multiple faces within an image and can do so for any skin tone. In the two below images we can see the classifier identifying myself in the first image and in the second image can locate the three faces within it.[11]

[10] https://docs.opencv.org/3.1.0/d4/d73/tutorial_py_contours_begin.html

[11] https://docs.opencv.org/3.4.1/d7/d8b/tutorial_py_face_detection.html

Figure 9, 10 Haar cascades on single and group photograph.

2.1.7 Conclusion

The research above shows what methods and algorithms are needed to get this project of the ground and the difference between the digital and IR camera. We will see further in the report that the Infra red camera is not as easy to work with as the digital camera is and how the face detection algorithm can be difficult to work with in certain situations.

2.2 Alternative existing solutions to the problem you are solving

2.2.1 Human Skin Detection by Visible and Near-Infrared Imaging.

The purpose of this application is to use spectroscopy to increase the effectiveness of pedestrian detection in Advanced Driver Assistance Systems (ADAS). Yusuke Kanzawa, Yoshikatsu Kimura and Takashi Naito define spectroscopy as “Spectroscopy is the study of how substances absorb, transmit, or reflect light."[12] They go on further to say that they use a method to detect human skin using spectroscopy.

This solution uses near infrared and visible processing to detect humans. They created their own camera to combine both processing types, as human skin, clouds and the sky share the same wavelength. The NIR camera will find the human skin and the visible camera will be used to filter out the sky and clouds.

2.3 Technologies researched

2.3.1 OpenCV VS Mat Lab

There are two main independent development environments (IDE) to use when developing an image processing application. For this application I chose OpenCV, OpenCV (Open Source Computer Vision)

[12] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.389.6745&rep=rep1&type=pdf

is an open source library of methods used for image processing.[13] Mat Lab is a pay to use development environment which is based of Java. I chose OpenCV because it is faster than Mat lab when it comes to processing algorithms, it's free and the resources needed to run it are less than they are to run Mat lab. The downsides to not using Mat Lab would be that it's a development environment and the advantages that would bring.

2.4 Resultant findings/requirements

- Infrared LED prototype to shine infrared light.
- Use python with OpenCV to program the client.
- Create a client on the raspberry pi to post images.
- Create a python flask web server to store and run analysis on the images and data.
- Use JQUERY to create a dynamic gallery.
- Create a MYSQL database to store the data.
- Stack Overflow, SO. (2017)How to detect edge and crop an image in Python.

2.5 Bibliography (research sources)

- karanjthakkar, KT. (2017)What is OpenCV? OpenCV vs mat lab
  Available at: https://karanjthakkar.wordpress.com/2012/11/21/what-is-opencv-opencv-vs-matlab/ (29/11/2017)
- raspberrypi, RP. (2017)Camera Module
  Available at: https://www.raspberrypi.org/documentation/hardware/camera/ (29/11/2017)
- ivrl epfl, IE. (2017)Near Infrared imaging
  Available at: http://ivrl.epfl.ch/research/infrared/imaging (29/11/2017)
- thepihut, PH. (2017)Raspberry pi camera module V2.
  Available at: https://thepihut.com/products/raspberry-pi-camera-module (29/11/2017)
- raspberrypi, RP. (2017)Pi NoiR camera V2
  Available at: https://www.raspberrypi.org/products/camera-module-v2/ (29/11/2017)
- pimylifeup, PL. (2017)Raspberry pi camera vs NoiR camera
  Available at: https://pimylifeup.com/raspberry-pi-camera-vs-noir-camera/ (29/11/2017)
- raspberrypi, RP. (2017) Pi NoiR camera V2
  Available at: https://www.raspberrypi.org/products/pi-noir-camera-v2/ (29/11/2017)
- tutorialspoint, TP. (2017)Digital Image Processing
  Available at: https://www.tutorialspoint.com/dip/ (29/11/2017)
- bogotobogo, BB. (2017)Image Noise Reduction: Non-local means denoising algorithm
  Available at: http://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_Image_Non-local_Means_Denoising_Algorithm_Noise_Reduction.php (29/11/2017)
- C. Fredembach and S. Süsstrunk, Colouring the near infrared, Proceedings of the IS&T 16th Color Imaging Conference, pp. 176-182, 2008.
- Opencv, OC. (2017)Contours getting started
  Available at: https://docs.opencv.org/3.1.0/d4/d73/tutorial_py_contours_begin.html (29/11/2017)
- citeseerx, CE. (2017)Human Skin Detection by Visible and Near-infrared Imaging
  Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.389.6745&rep=rep1&type=pdf (29/11/2017)
- navicat, NC. (2017)Navicat.
  Available at: https://www.navicat.com/en/download/navicat-data-modeler (29/11/2017)
- staruml, SU. (2017)StarUML
  Available at: http://staruml.io/download (29/11/2017)
- raspberrypi, RP. (2017)Infrared Camera you asked us so we are making them
  Available at: https://www.raspberrypi.org/blog/infrared-camera-you-asked/ (29/11/2017)
- DYNU, DYNU. (2017)DYNU.
  Available at: https://www.dynu.com/en-US/ControlPanel/BasicEdit?ATVPDKIKX0DER=SQBEAD0AOAAwADEANQA4ADMAMwA%7C (29/11/2017)
- tutsplus, TP. (2017)Creating a web app from scratch Using Python Flask and MySQL
  Available at: https://code.tutsplus.com/tutorials/creating-a-web-app-from-scratch-using-python-flask-and-mysql--cms-22972 (29/11/2017)

[13] https://karanjthakkar.wordpress.com/2012/11/21/what-is-opencv-opencv-vs-matlab/

- youtube, YT. (2017)setup a Raspberry Pi Web server with you own .COM using google domains
  Available at: https://www.youtube.com/watch?v=vzojwG7OB7c&t=2959s&index=6&list=WL (29/11/2017)
- Amazon, A. (2017)Raspberry Pi NoiR V2
  Available at: https://www.amazon.co.uk/Raspberry-Pi-1080p-Camera-Module/dp/B01ER2SMHY (29/11/2017)
- youtube, YT. (2017)Displaying images in server using flask and python
  Available at: https://www.youtube.com/watch?v=Y2fMCxLz6wM&t=472s&index=7&list=WL (29/11/2017)
- https://docs.opencv.org/3.4.1/d7/d8b/tutorial_py_face_detection.html

3. Design

3.1 Identification of design methodology including why it was chosen

3.1.1 What is Agile Scrum

Agile Scrum is a software methodology where scrum teams usually comprised of two or more people break down a large task into smaller tasks, the development team will then have a sprint planning session where they decide what task to bring into the weeks sprint. The team will then estimate the cost of each task i.e. how many hours it will take to complete. The person who owns the project is called the product owner and during the sprint planning meetings the will tell the dev team what they would like to see worked on in the next sprint. The scrum master is a person typically who is outside the team, they will try to keep the work on track by resolving blockers while keeping the product owner reasonable with their requests. At the end of a sprint the team will have a product backlog refinement, A PBR is where the product owner dev team and scrum master sit down and discuss the priority of upcoming tasks and if they should be prepared for the next sprint or three sprints down the line.[14]

3.1.2 Why use Agile Scrum

The reason I chose to use Agile Scrum is that it allows me to split the project into smaller parts and work on them as needed. It helps with the time keeping aspect of the project as you set a goal to get x amount of work done in a single sprint. Currently working eight hours a day seven days a week, will give fifty eight hours time to complete work. In the latest sprint which can be seen below, the weeks work takes fifty one hours leaving five hours of free time.

[14] http://scrummethodology.com/

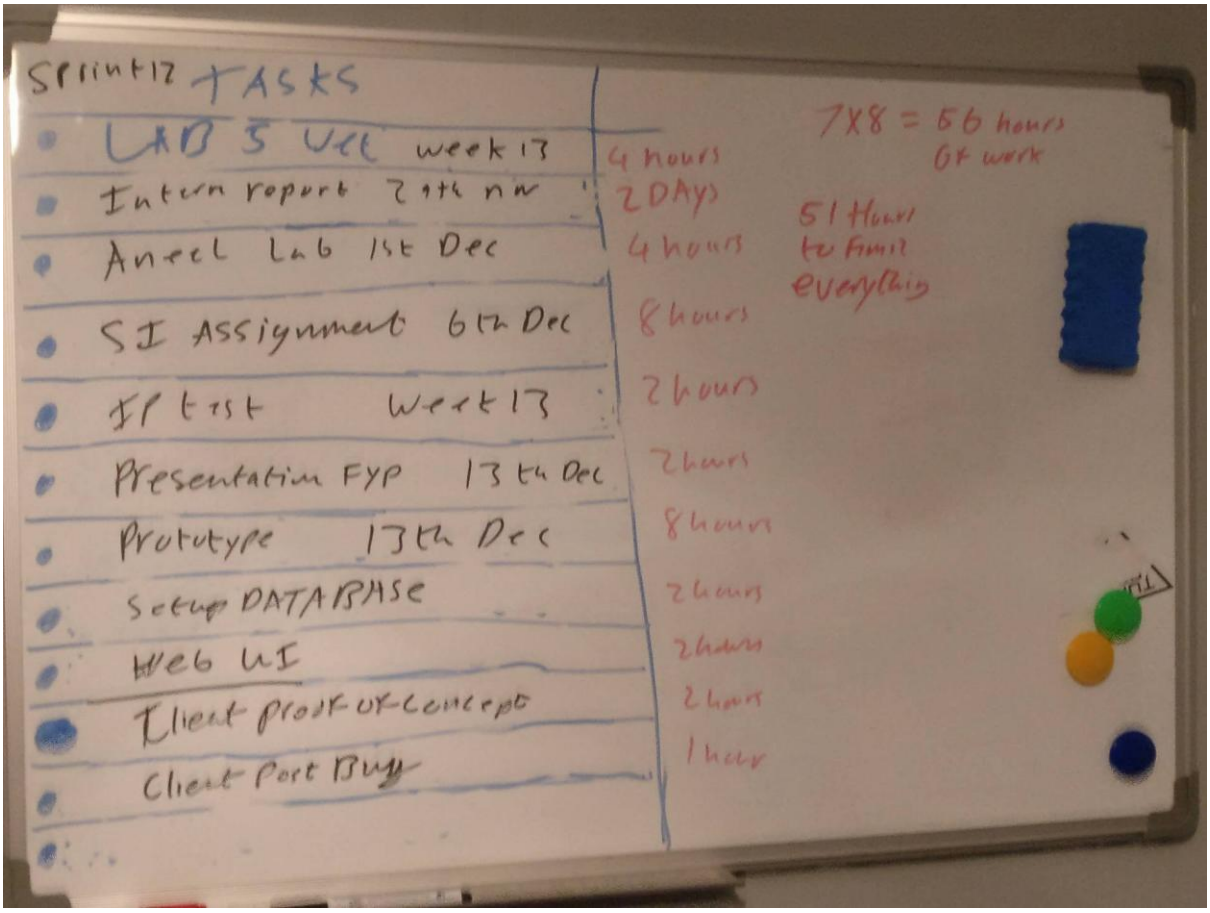


Figure 11 Week 12 sprint

3.2 Design of each of the project components

3.2.1 Overall design approach

The idea when designing this system was to create all elements independent of each other, then connect them together through the use of a web server. As the raspberry pi acts as a remote object there had to be a way of getting the output to a user, this is where the web server and MYSQL database come into play, with the creation of a web server hosted on a domain a user was now able to access the output from the raspberry pi.

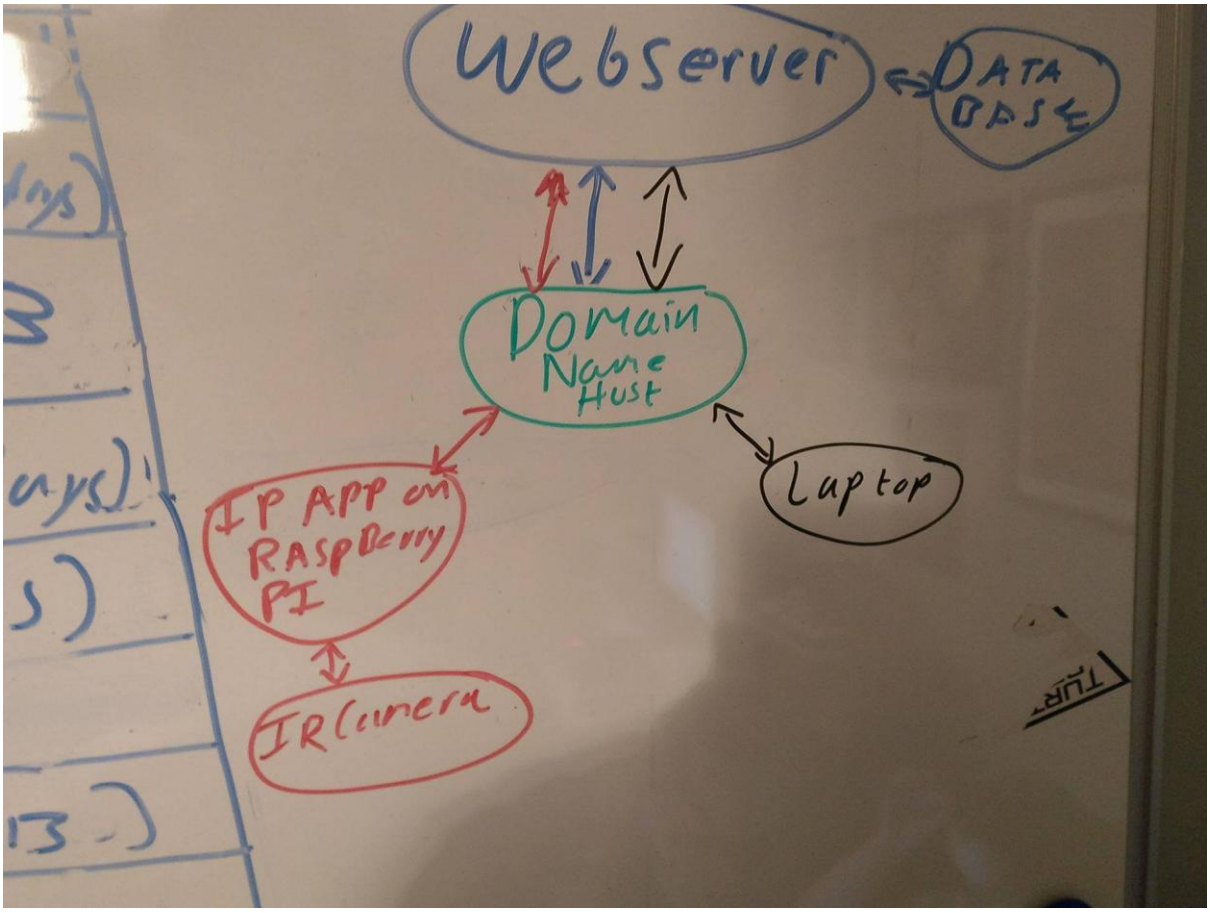


Figure 12 low level diagram of architecture

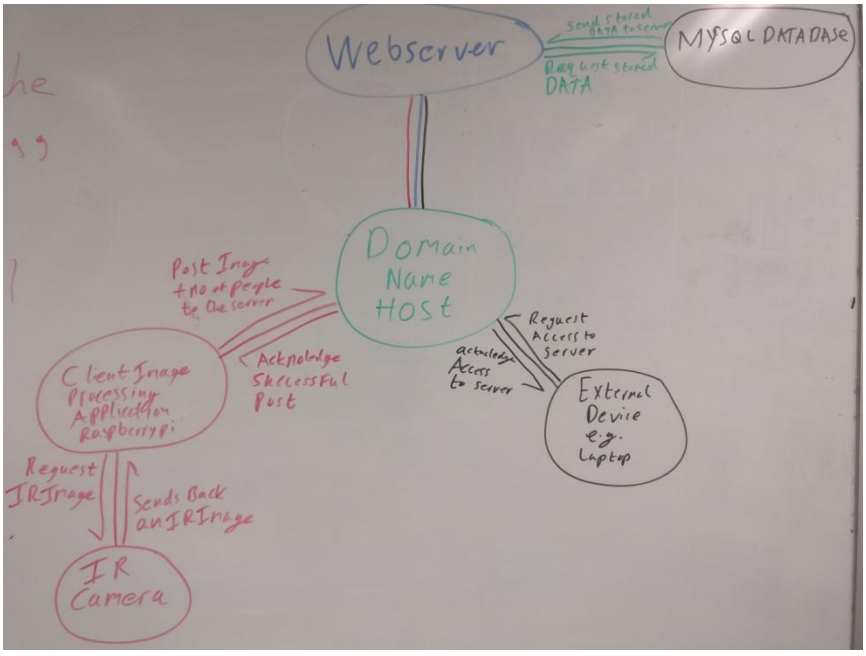


Figure 13 more detail into the architecture

3.2.2 Raspberry Pi

The raspberry pi was chosen for this project as it is portable, easy to setup anywhere, has internet access and can create its own web server. Combining the raspberry pi with an infrared camera and internet access, in regards to the project it can be used anywhere with a power supply and wifi. When designing this component for the project there needed to be a way to take pictures in low light. The solution was to create a mount with super bright infrared LEDS. All the parts were obtained to construct the mount but due to timing issues it was not constructed. Below we can see the designs for this prototype.

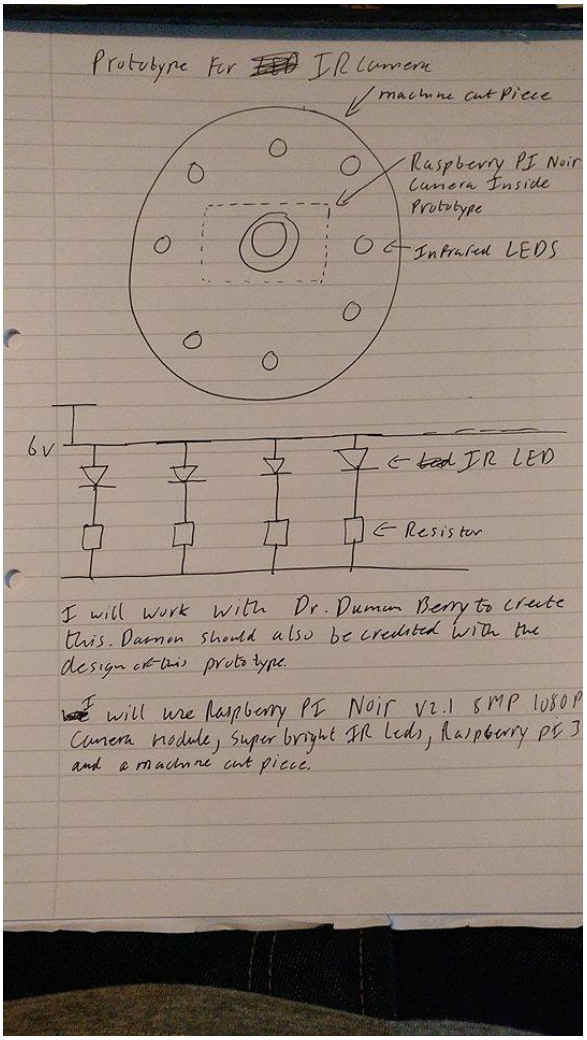


Figure 14 Prototype Camera. This image demonstrates how the frontal mount will work, showing how he LEDS will be shining on the area the camera is facing, illuminating the area with infrared light which is invisible to us but not the camera.

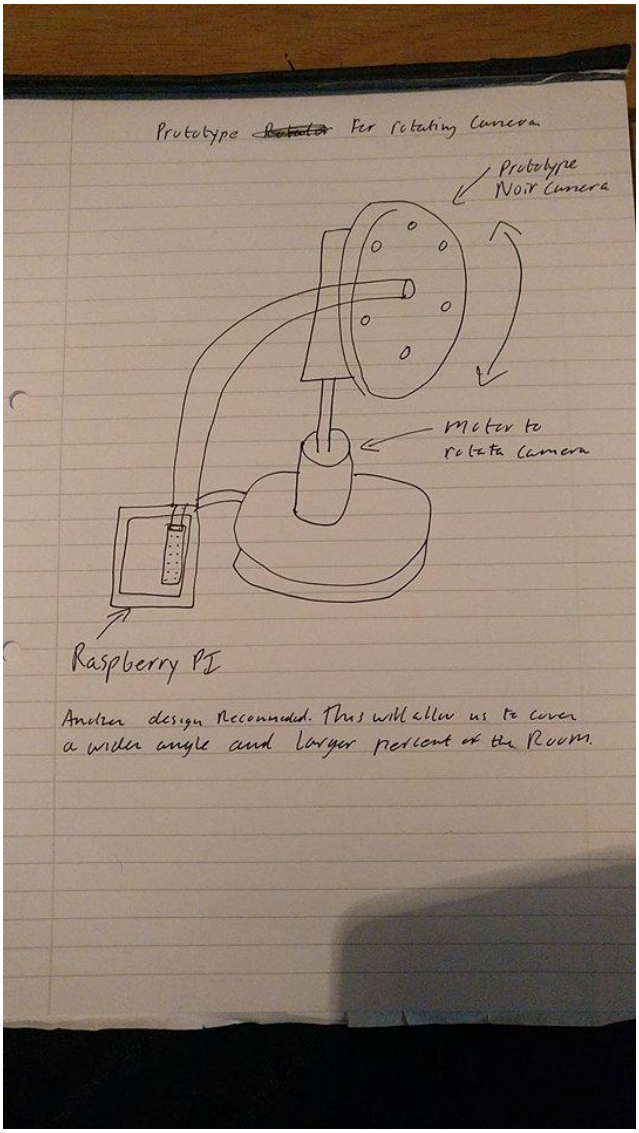


Figure 15 Prototype Motorised camera. This image shows how the prototype can be motorised to capture a greater angle within a room removing the static positioning of the camera.

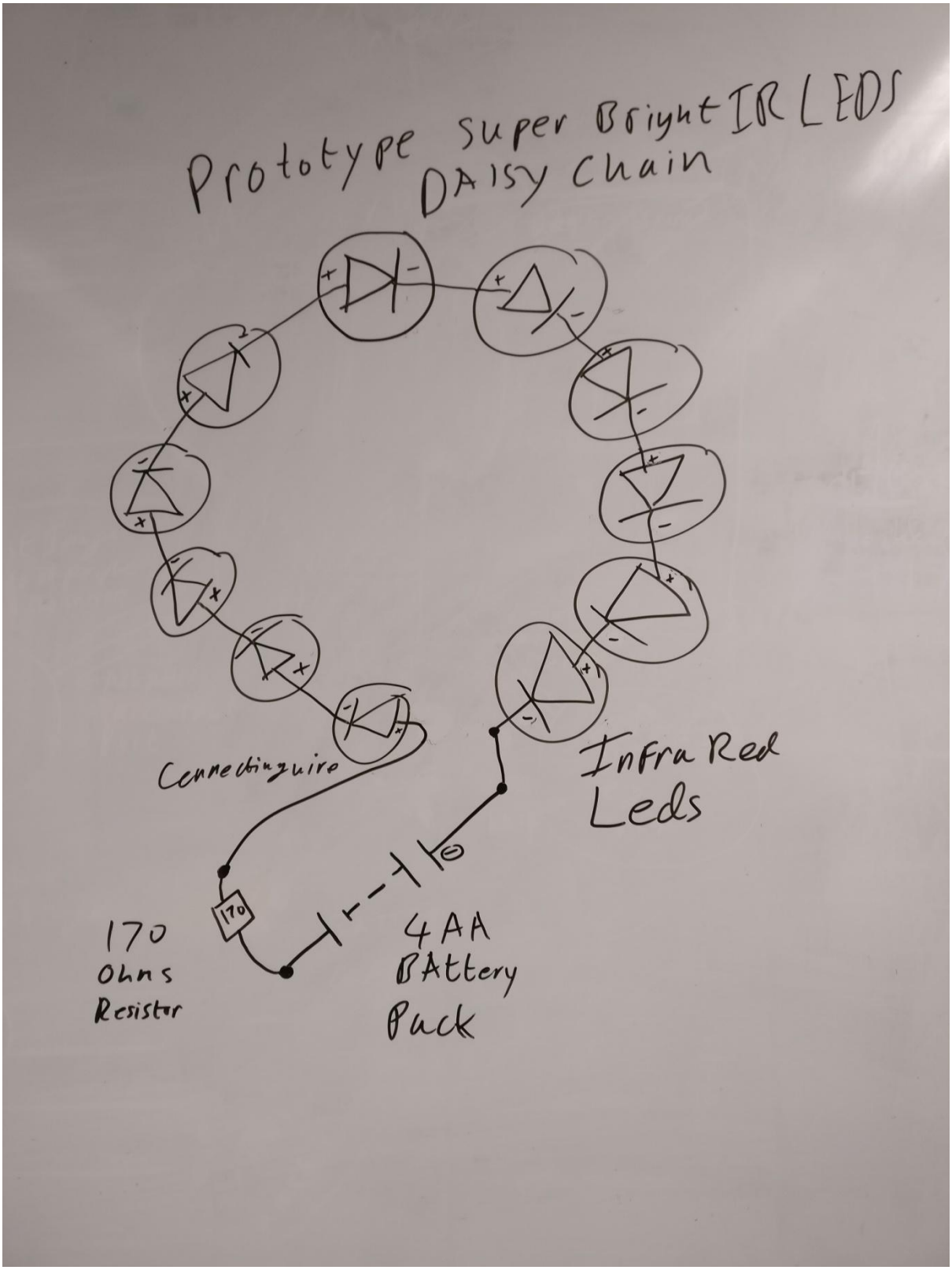


Figure 16 Daisy chain solution for infrared LEDS. This image shows the prototype for the LEDS, these LEDS would be placed inside the mount seen in images 11 and 12 what you see above is an example of how they would be connected and powered.

3.2.3 Web Server

When creating the web server python flask was chosen as it has many advantages for this type of project, one of these major advantages was the use of JINJA2 which is a module within flask. JINJA allows the application to pass data from Python to HTML and JavaScript, and then pass it back to python again. This came in very handy when the queries where being created as it required real time information from both the JavaScript and the html. Another reason for choosing flask was how easy it is to render the HTML templates and communicate with the MYSQL database. Below is a system overview showing how the web server integrates with the other components.

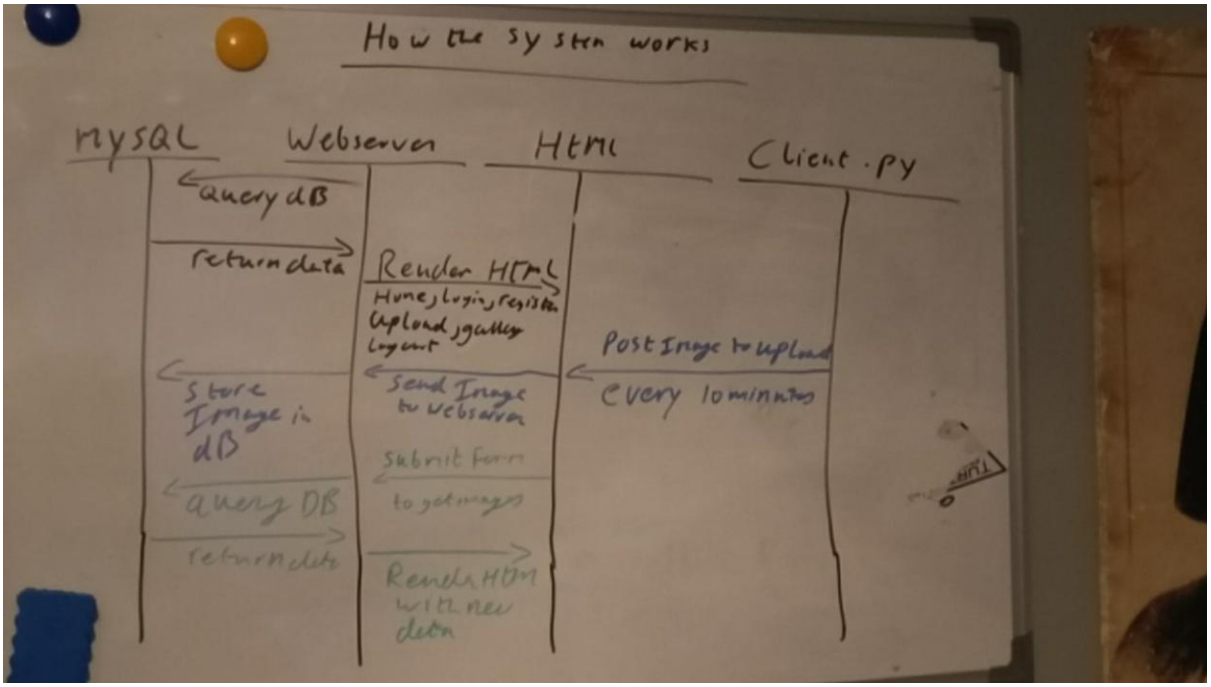


Figure 17 System Overview

3.2.4 Image Processing application

For the application python with OPENCV was chosen. OPENCV combined with python has great documentation is open source and can easily be installed on a remote device such as a raspberry pi. The design for the application was to create a class which can be run continuously taking images every ten minutes while running a face recognition algorithm. This was done with a combination of segmentation and Haar Cascades.

3.2.5 MYSQL

The design decision to choose MYSQL was due to it being compatibility with python, python has built in modules which allow an easy connection to the MYSQL database. Before rendering a html page python allows us to run the necessary queries such as checking login information or preparing the gallery with data.

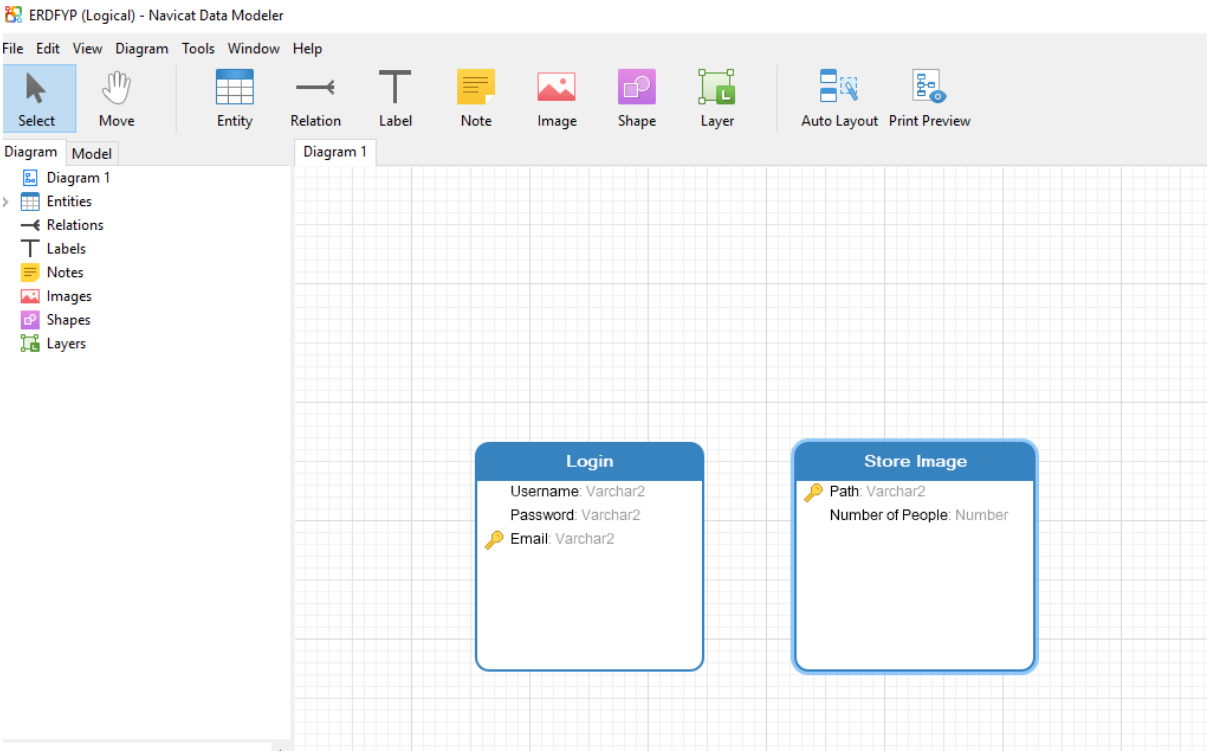


Figure 18 MYSQL ERD

3.2.6 Upload and Gallery HTML pages

The Upload and Gallery HTML pages are the parts of the project which can be demonstrated to a user, the upload page allows a remote device such as the raspberry pi to post on image to the web server and the gallery allows the user to view the most recent images and images based on a specific date time, the gallery also displays information about the pictures such as the number of people in them and the date time. Below you can see an early concept for the gallery.

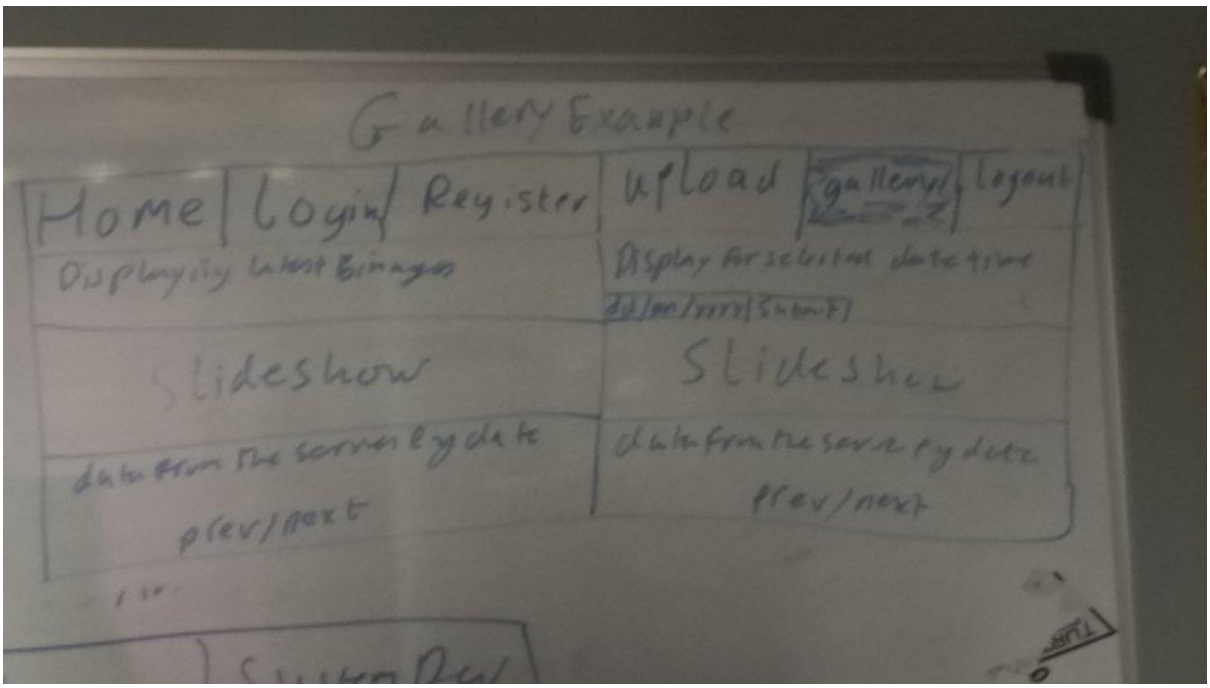


Figure 19 Early concept of the Image gallery.

3.3 Define a list of features and use cases supported within the project

3.3.1 List of features

The following features are implemented within the project:

User	Client	Server	Administrator
Login	Post image to Uploads	Query the database	Delete Account
Logout		Generate the gallery with data	Delete data from the database
Register		Accept and store posts from remote devices	Remove stored images
View Gallery		Encrypt user passwords	
Submit new Query to the gallery			
Delete Account			

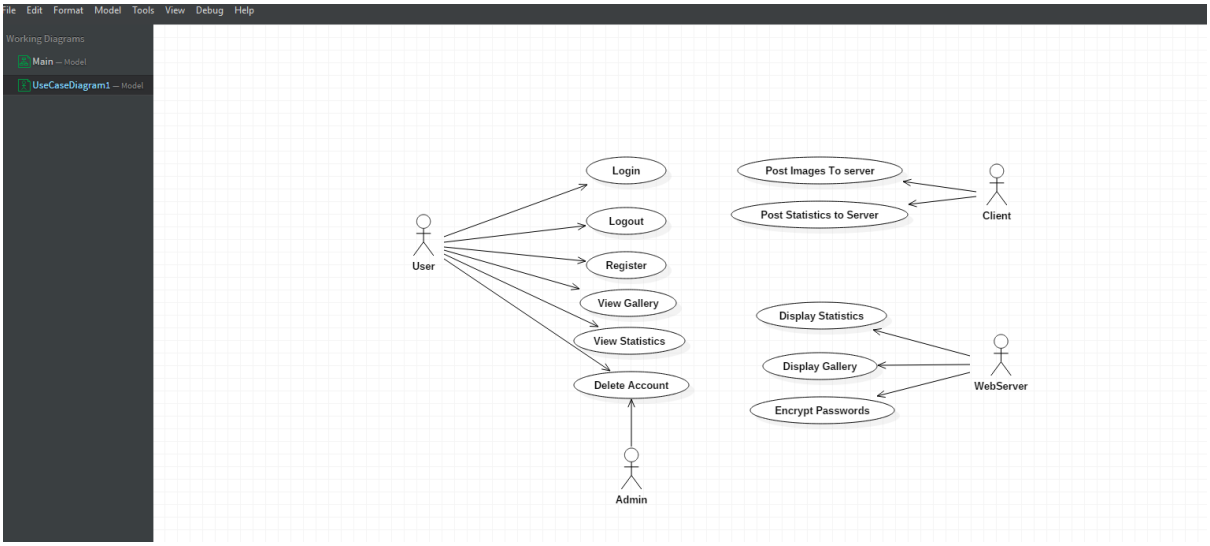


Figure 20 Use Case

3.4 Other Design Documents

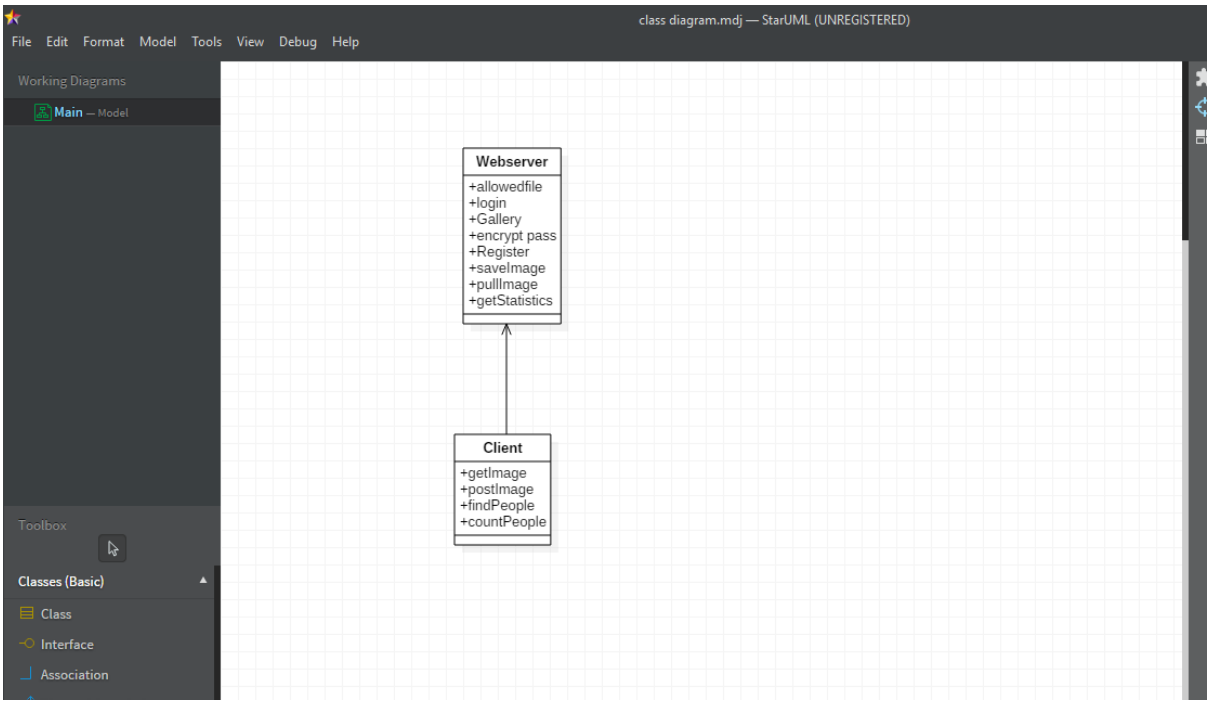


Figure 21 interim Class Diagram

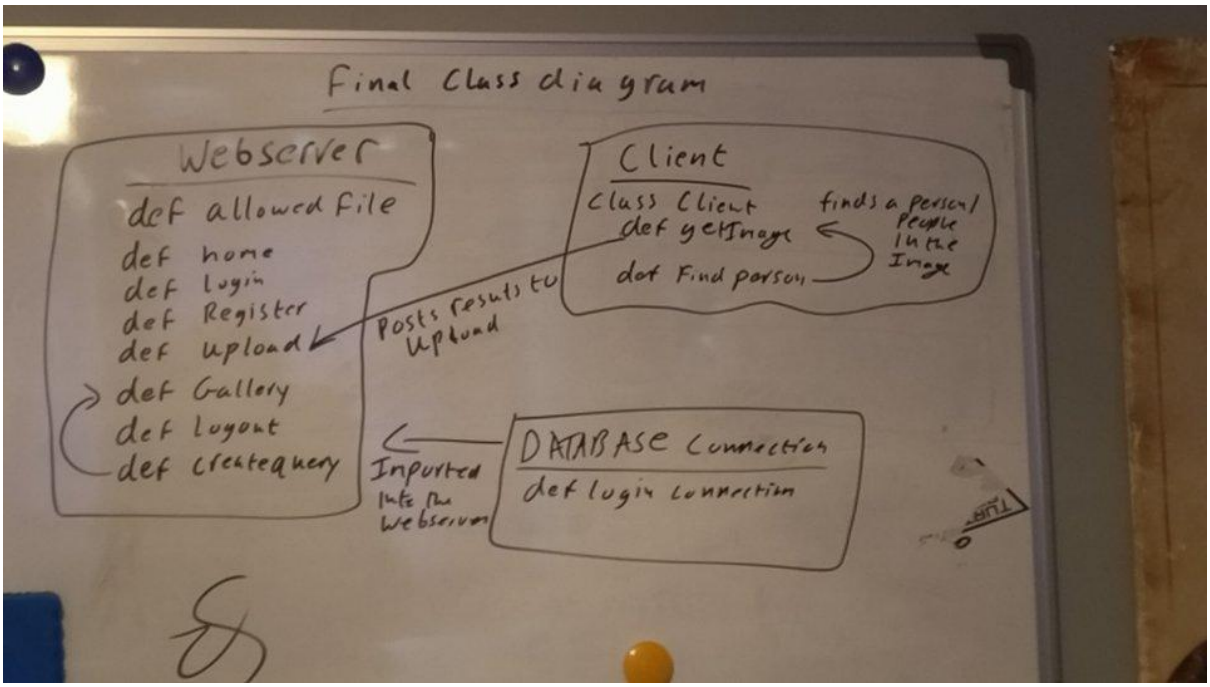


Figure 22 Final Class Diagram, the two images shown above are early designs and completed designs of what the class structures of the program are. They list the functions within the programs and give a low level account of how they interact with each other.

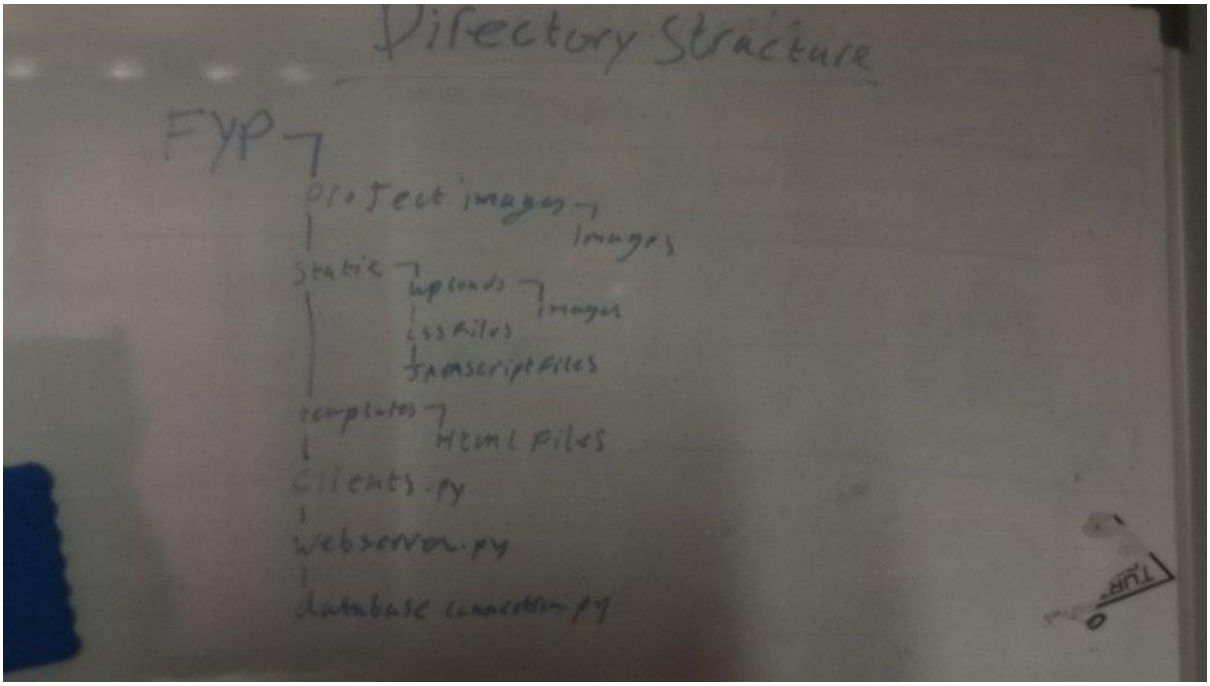


Figure 23 Directory Structure, the above image is an early example of how the directories would be structured in the project.

	Task	Start-Date	Proposed End-Date
2	Interim Report	30/09/2017	29/11/2017
3	Create Flask Server	30/09/2017	13/12/2017
4	Create Raspberry Pi Client	30/09/2017	13/12/2017
5	Setup MySQL Database	30/09/2017	13/12/2017
6	Host Server on Domain Name	30/09/2017	13/12/2017
7	Create Login System	30/09/2017	13/12/2017
8	Post Images from the client	30/09/2017	13/12/2017
9	Display Images in A gallery	30/09/2017	13/12/2017
10	Presentation	30/09/2017	13/12/2017
11	Create Detection Algorithm	30/09/2017	Project Fair
12	Build IR camera prototype	30/09/2017	Project Fair
13	Store the image and Number of people in a database	30/09/2017	Project Fair
14	Data analysis on the number of people e.g. Average per time of day	30/09/2017	Project Fair
15	Display Analysis to a user	30/09/2017	Project Fair
16	Final Report	30/09/2017	Project Fair
17	Project Fair	30/09/2017	Project Fair

Figure 24 Gantt chart, this image is from the start of the project outlining the time frames to complete the project.

4. Architecture and Development

4.1 Overview of the system architecture and key elements

4.1.1 What the system is now

The image below shows what the system is currently. What was set out to achieve with this project from an architecture point of view was achieved. All components are independent of each other, and can communicate only with necessary components, for example the raspberry pi cannot contact the MYSQL database it must use the python flask web server as a mediator between the two. The client, MYSQL and any other remote devices speak to the web server through a connection to the domain name which host the web server.

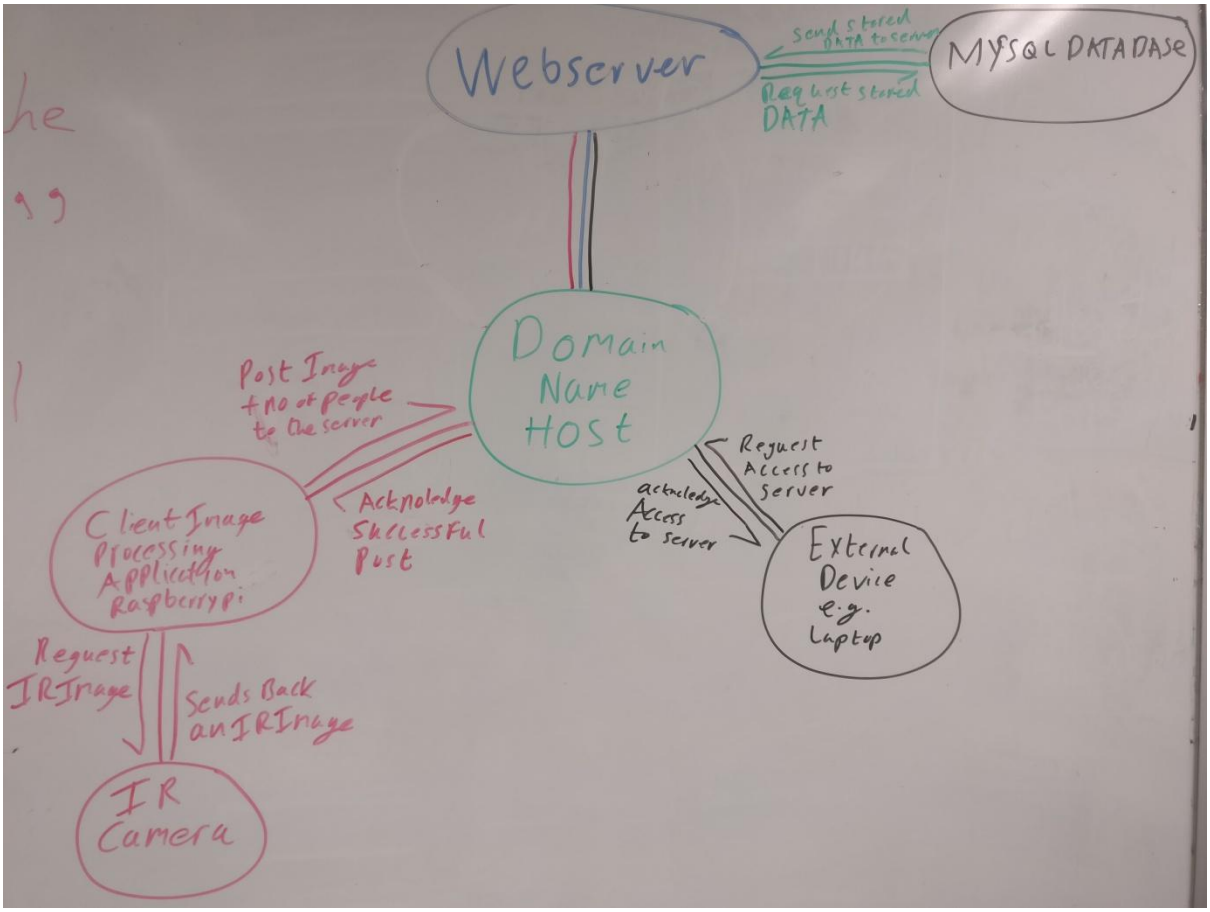


Figure 25 more detail into the architecture

4.2 Details of each component

4.2.1 Person Detection Algorithm

The algorithm will use thresholding to find lightest and darkest pixels. In the case of this application the darkest pixels will be the areas which do not have infra red light shining on them. Once a mask has been created from the threshold value the image will be segmented, the area we are looking for in the image for a proof of concept will be the brightest areas. For example the application will try to find skin over clothing. Once the person's skin has been found contours will be used to show how many areas in the image are recognised as skin, after applying contours to the image it will be ran

through the haar cascade part of the program. Haar cascades will look at each contour in the image and using the pre-defined facial classifier, decide if the contour is a face or not. For future work it is hoped that the application will use more advanced machine learning such as creating a classifier from scratch to understand that, contours of skin from a side view or from a top down view can be recognised as a person.

4.2.2 Raspberry pi

The raspberry pi 3 is used to host the client and is also used to mount the Pi NoiR camera to take photos. The reason for using the raspberry pi is that it can be taken and positioned anywhere with a wifi connection and can begin taking photos and posting them back to the server. The raspberry pi will run the person detection algorithm which involves thresholding, segmenting and contouring the image. The finished image is then posted back to the web server along with the number of people detected and the date time it was taken. The program is run continuously and will send an image every 10 minutes.

4.2.3 Web Server

The web server is based on my home pc and the IP of my machine is being projected from a domain name server. The client will post images to the server and the server will store it on the hard disk drive of the home computer. The web server will then take the path to the image and store it in the MYSQL database for querying later. A user can access the server from their laptop or pc and they will first have the ability to login or register. When a user has successfully registered or logged in they will have access to view the image gallery page. This page will show two dynamically filled JQUERY slideshows, the first will display the latest images taken and below it will show the details of the image such as the date and time it was taken as well as the number of people in the image, the second slideshow will display images for a selected date range and like the first slideshow display the information which accompanies it. Another piece of functionality the user will be able to see in the gallery is analysis which has been run on the data. This will be displayed in the form of for example "Average No. of people for day/month/year hour/minutes/seconds". Another example of a query which could be implemented is checking if rooms are empty at certain peak times during the day.

4.2.4 Storing Data

The MYSQL database will be installed on the home computer just like the web server. The images and data posted to the web server will be stored in this database. The schema for this database will consist of two tables. The first table is the login table and will store the username and password which will be varchars, and will also store the email address which will also be a varchar and a primary key for the table. The second table will be the data storage table, this table will have two values the first will be the directory path for the images, and this will be a varchar and primary key. The second attribute will be the number of people counted or occupancy measurement, this will take the form of a Number in the database.

4.2.5 Accessing the web server and client.

For administration purposes the user will be able to SSH into the server and the raspberry pi client. The purpose for this is to allow an administrative user to make changes to the server, database or the code running on the raspberry pi client. It also allows an admin to troubleshoot any errors off site.

4.2.6 Security

The web server will be available through a domain name so it is important to protect it from unauthorized users. The use of regular expression will help prevent SQL injections through the login and registration page, and encrypting the user passwords before they are stored in the database will help protect the user accounts. When using SSH between the client and server authentication mechanisms such as password and public/private SSH keys will be used.

4.3 Development of each component, prototypes and solution created

4.3.1 Problem and solution created

The problem to be solved is the use of Infrared image processing to measure room occupancy, and to display this information in a user friendly way. The solution to this problem was to design a system which connected all stages through a web server. The web server receives connections from both the user and the raspberry pi but the two never meet. The raspberry pi receives an image from the IR camera, runs an algorithm to count the people in a room and posts it to the web server, the web server stores this in a database where it analyses the data to present it in a user friendly format. The user can then log into the web server, and view the images and data analysed.

4.3.2 Creating the prototype camera

The camera I am using for this project is the raspberry pi NoIR camera. The reason for choosing this camera is that it has no infrared filter attached. The lack of an IR filter allows the taking of lowlight and infrared images. For this camera to take infra red (IR) images, IR LEDS must be shined onto the target area.

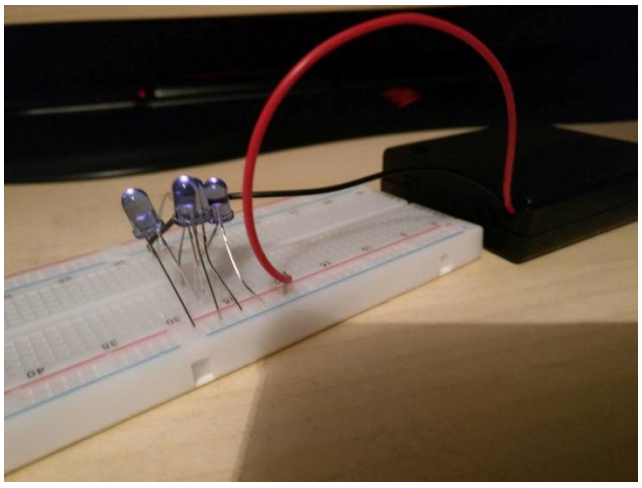


Figure 26 Breadboard Prototype

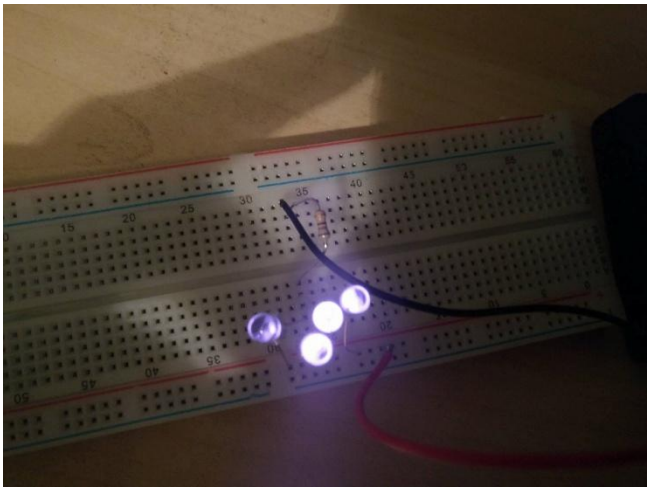


Figure 27 IR Leds

4.3.3 Creating the web server.

To create the web server the python flask framework was used. At first the web server was running off localhost on my home network. As of now it is being hosted on a domain name website using my home computers IP address. The web server will display five pages; these are the login, register, upload, gallery and logout. The login and registration pages are straightforward a user enters their details into a form and the server stores them in a database, the server will retrieve them to compare login credentials. The upload page allows a client application to post an image to the server; the server will then store this image on the Hard disk drive of my home computer and save the path, date time and number of people in the room to the database. The gallery page will query the database and fill four slideshows with information. There are two image slideshows the first displays the five latest images the second displays the max number of images taken within a specific date time. The last two slideshows are for information connecting to these slideshows when the next or previous button is clicked the text transitions along with the image showing the corresponding information such as date time and number of people.

4.3.4 Setting up the domain name and hosting.

To host my web server I used the site dynu.com and chose the domain name michaelkanefyp.dynu.net:5000/, this is open to change but for now it's just a temporary name. To allow traffic to connect to the web server the home router needed to enable port forwarding on the address and port number. With port forwarding enabled the domain name was configured to mask my IP address with the domain name. There were some issues with this as the port number would have to be appended to the domain name. This was fixed by adding the port number to the masked domain name.

4.3.5 Creating the database.

The database is created with MYSQL and will have two tables. The first table is the login table; this table will store the username, email address, and the encrypted password for the user account. The primary key for this table will be the email address. The second table will be the store image table,

this table will store the directory path to the image on the web server and will also store the number of people found in that image as an integer value. The primary key for this table will be the directory.

4.3.6 Creating the client.

The client is hosted on a raspberry pi and uses Python with OPENCV and Anaconda to create the image processing application. The client will threshold then segment the region of interest from the image, it then uses contours and haar cascades to find the people within the image. The client uses an import called request to post the processed image to the web server.

4.4 Identification of external API'S

4.4.1 JQUERY

JQUERY is a lightweight JavaScript library which allows easier integration between JavaScript and the web server. JQUERY provides useful prebuilt functions which would take large amounts of JavaScript to write and allow the user to import them into their project. Specifically in this project JQUERY is used for it features such as the manipulation of Document Object models (DOM) and its animation features such as slideshows.[15] [16]

4.4.2 Cycle2

Cycle2 is a JQUERY slideshow plugin which provides an easy way to display the information to the user in the gallery HTML page. Python runs a select query on the database then sends the data to JavaScript. The JavaScript then uses this data to select the images and place them into img tags which are then using a for loop inserted into the HTML.[17]

5. System Validation

5.1 Testing

5.1.1 Testing Outputs

The first step I chose to take when testing was that the expected output where being returned to the server. What I mean by this is that if I posted an image to the web server I would expect to see that image on the web server. The same can be said about the database, the point of me doing this was to understand the flow of data between the applications and to make sure that they are connecting to the correct areas.

5.1.2 Regular Expressions and test functions

Now that I have tested that physical data is being sent to the server, I created functions such as the function allowed file. The purpose of this function is to only allow jpg files into the server. This puts a layer of security in case that if an entity can post to the web server, that the web server will only accept a jpg image. The login and registration page for the user will also be tested for SQL injections

[15] https://www.w3schools.com/jquery/jquery_intro.asp
[16] https://jquery.com/
[17] http://jquery.malsup.com/cycle2/

through the use of regular expressions. Regular expressions allow the program to disallow certain characters such as ">" from being used in a username or password.

5.1.3 Unit Testing

The purpose of a unit test is to ensure that the output of a program does not change as a programmer is making changes to the codebase. Unless it is intended for the output to change, Unit Tests are a great way to ensure continuity of the data sent from client to server, or from server to database. Due to time constraints I was unable to add any unit tests to the application.

5.1.4 Usability test

To test the usability of the system I intend to get friends, family and people I do not know to test the system. This will take the form of a heuristic evaluation of the system. Due to time constraints I was unable to run a usability test on the application.

5.2 Demonstration

5.2.1 Demonstration of the Application

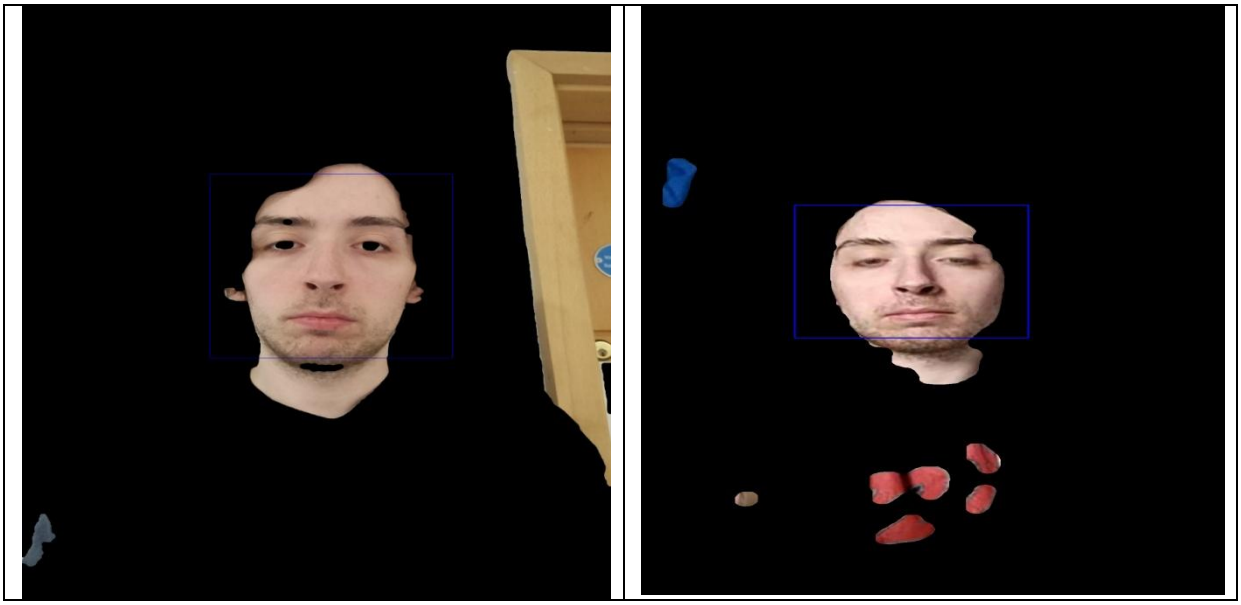


Figure 28, 29 Flesh Detection Light Skin

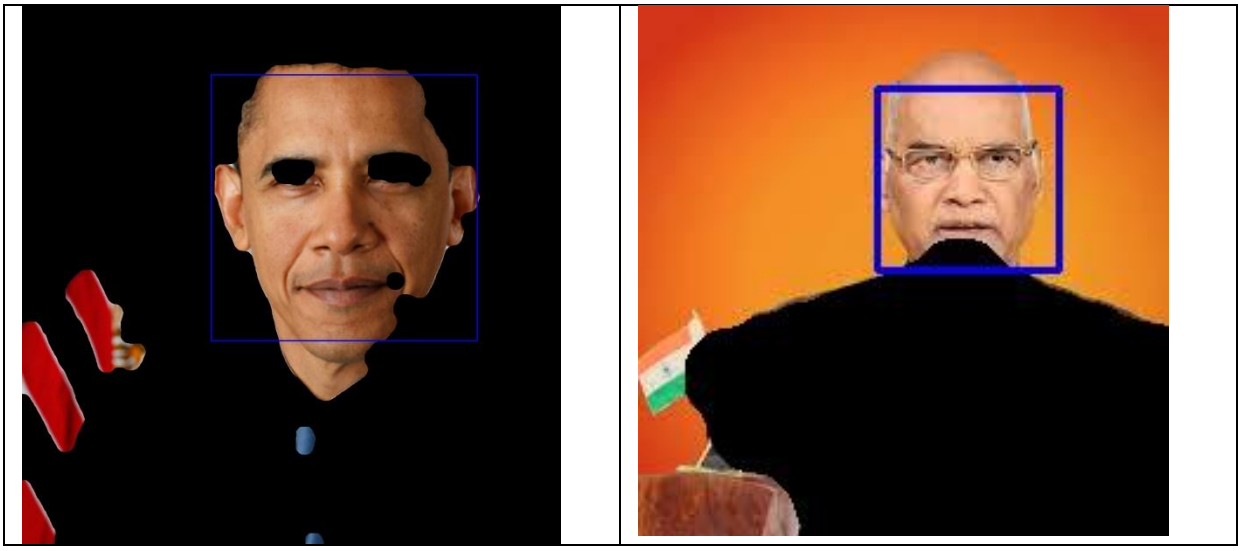


Figure 30, 31 Flesh Detection on Darker Skin tones

Figure 32, 33 Flesh Detection on a Single Person Vs Group

5.2.2 Demonstration of the Gallery

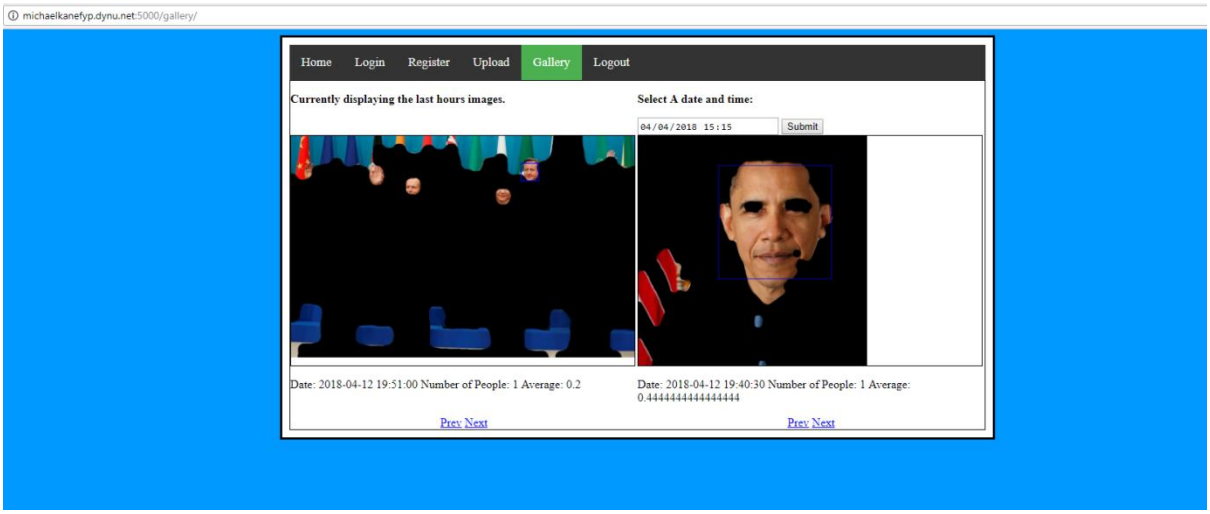


Figure 34 Gallery

6. Project Plan

6.1 What was the plan in the beginning and how it changed

6.1.1 Interim plan

The plan for the interim presentation was to have a prototype website, database and client. The website would allow a user to submit a file through the upload page and a remote device to submit a file through the upload page. The users can login logout and register, and also view any images by visiting the gallery.

6.1.2 Final plan

The plan for the final presentation was to finish the system. The website allowed only a remote device to post images to the server and the gallery would display images in a slideshow as well as data for each image. The client was upgraded with a machine learning technique called haar cascades which allowed the existing application to recognise faces.

6.1.3 What had changed and why

The upload HTML file was changed as there is no image processing done on the server side so it was redundant to allow a user to submit a file. The gallery was changed to display more information as the previous version had only a single image on screen. The client made use of haar cascades which allowed the application which could find hands arms and faces to distinguish between them and pick out a users face.

7. Issues and risks

7.1 Issues

7.1.1 Prototype camera

The prototype camera was not constructed due to time constraints all the parts are ready to be assembled.

7.1.2 Person Detection using Infra Red

From the research I have done so far I am worried that the IR camera may not be able to detect people at its current state. The project using spectroscopy to find people uses a 6 band camera to find a person, while I am using a lone infrared camera with IR led's. I hope to solve this problem by finishing the prototype and getting to test it with real world examples.

8. Future work

8.1 Planned Future work

8.1.1 Create the prototype Infrared camera

The work to be completed for the prototype camera is to daisy chain the infrared LEDS together, connect them to a low voltage resistor to increase the light intensity, and then connect them to a battery pack to power the prototype. Once the basic shell is completed the idea is to laser cut a piece of metal to hold the prototype, this will keep the prototype held in place and opens it up to be used with a motor to rotate the camera.

8.1.2 Remote Access

The prototype will also hopefully include SSH connections to both the server and client so public and private keys will need to be set up to ensure secure connections to the infrastructure.

8.1.3 Enhancing the Algorithm

The algorithm can currently find a person's face from the front. When in low light situations or large groups of people the average pixel size removes most of the images. This could be picked up on in the future to make the algorithm more accurate. Another improvement could be the creation of your own classifier for the project to better suit face detection for example from a side or top down view.

8.1.4 Usability testing

The web server will need a usability test to see how usable it is for a user. This will involve the use of Nielsen's heuristics and volunteers from family, friends, and people I don't know. They will test the interaction on the website and give feedback on what needs improvement and what they feel needs to be added.

8.1.5 Creating unit tests for the expected results from client, server and database

It will be important to have static unit tests where the result should always be the same. The code will change drastically as the project moves forward but as of now the output from each source should not change. So for future work unit tests will be created to make sure I keep on track with the work.

9. Conclusions

I believe I succeeded and failed with this project, I failed by not completing the core objective of this project which was to detect people in low light and dark situations, but I succeeded in creating a system which not only detects and can recognise a person of many different skin colours but also displays these results to a user on a dedicated web server. There are still lots of potential work in this project, the image processing algorithm can be refined even further to detect people with greater accuracy, the prototype camera can be constructed to obtain different room angles or even be paired with a second camera to compare images for a more accurate estimation. I enjoyed working on this project and I hope someone is inspired to take up where I left off or base a different project of what I have researched here.

10. Bibliography