**SECOND INDIVIDUAL DESIGN REPORT**

**Aslam Patel**

**Introduction**

DermDoc is a skincare app which provides the user with customised acne treatment. The initial design was for a tailored care system, the user will just have to take a picture on the app and instantaneously be given an evaluation of their skin condition alongside treatment options.

When deciding the project to study there were many ideas however the most exciting project of a facial recognition was chosen.

As one of 5 members of the group the initial workload was distributed evenly, and group members were set research and practical tasks to carry out.

When delegating the tasks facial recognition techniques/algorithms seemed the most ideal and challenging field to study hence this being chosen. Main tasks so far for the rest of group was heavy research into target market, flow of app, and GUI.

Since the last individual report, the group have made specific changes to research techniques, integration and design of the app which will be discussed in later parts of this report.

When implementing a facial detection app, and initial app prototype was created and the building off the app using HTML & CSS was followed. The bulk of the design where the most effort went into was the code/algorithm used for the actual facial detection which is discussed later in the report.

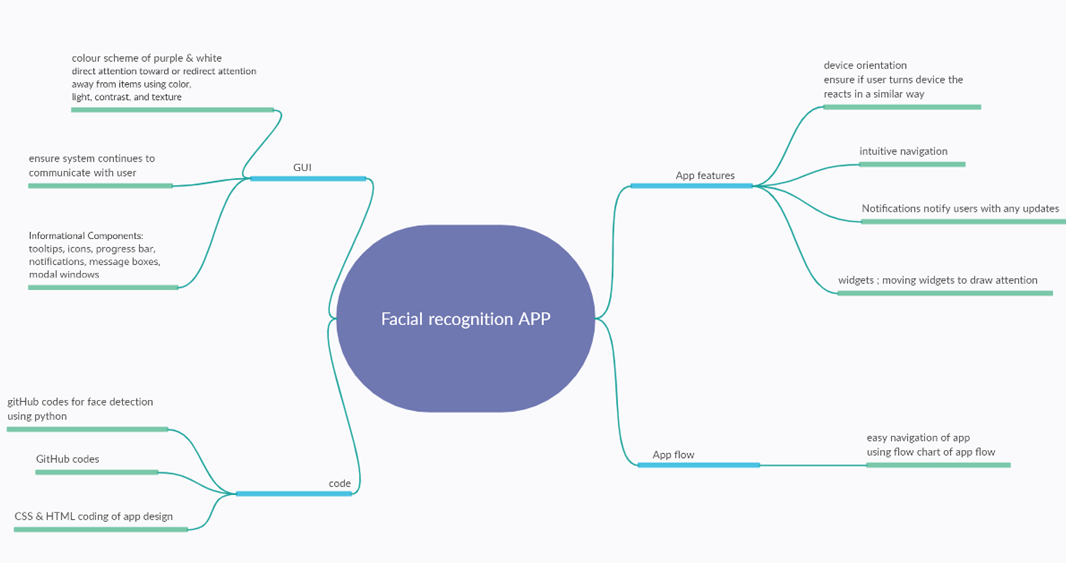
**Functional Overview**

The initial goal was to create an algorithm which would not only be able to detect a face but rather detect specific parts of a face and even marks on the user’s face. When beginning the construction of the code it was instinctively realised by the group that this will not be an easy task to complete.

As the construction began it was decided that the groups main focus would be to initially use a base code and add the specifics that the group wanted rather than just use an open-source code, again this task was challenging

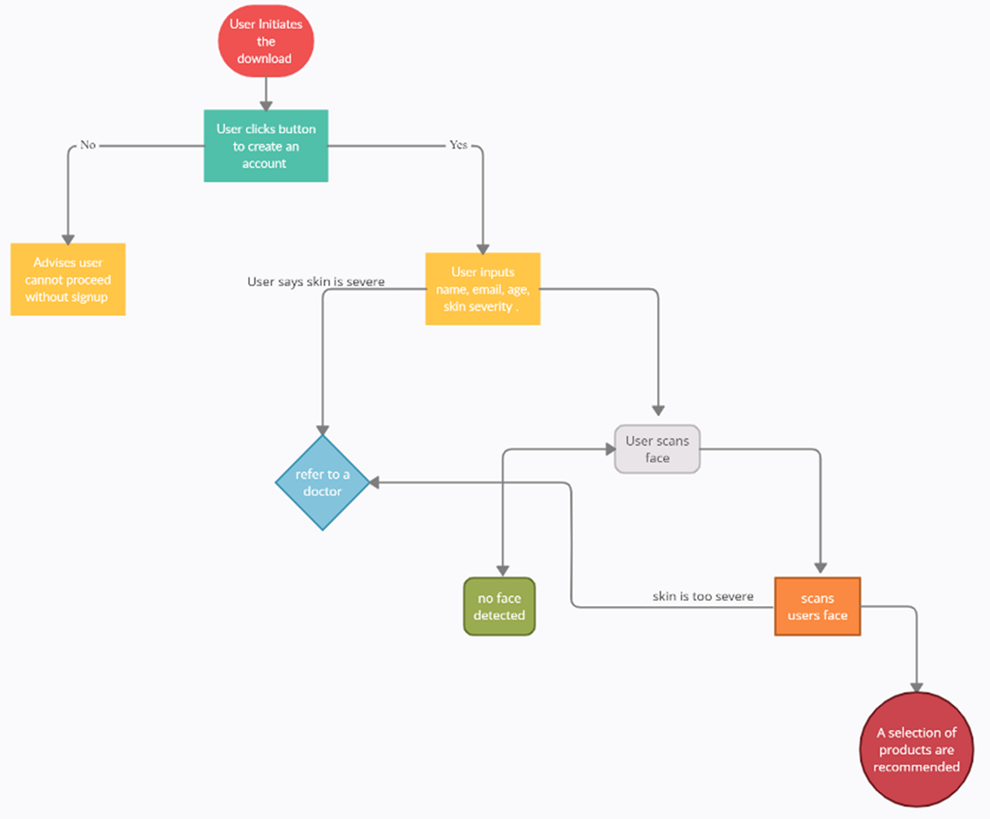
As the project being heavily focused on an app it was simple to identify the elements in the project that would need to be worked on.

The project was initially broken down into 4 main sections. To Begin With each section was worked on by all members of the group in which research tasks were completed, and then specifics were assigned to each of the group members. Figure.1 shows the initial topics of research.



Figure

The group attempted to use this diagram as base design and related all additions to design back to this one diagram. Ultimately it aided with the one goal that the group had which was to build a fully functioning app design.

A system overview flow design shown in figure 3.

A simple flow of the app was created by the group which the group attempted to follow when implementing the design

The main goal was to ensure that we stuck to this flow to not only enhance user experience but to ensure that the group are creating a free functioning program

An example of this is when the user wants to scan their face but due to the lack of visibility no face can be detected it will prompt the user back to scan their face again.

**Research and Design Update (10 marks)**

The initial research of the facial detection code is shown in figure 2. With annotations on what the different parts of the code actually does as function. As this was a pre-written code that was found on GitHub the group did not want to simply use a code that was found on the internet rather to build one that resembles this, but all code was written solely by the group.



Figure

**Constructional Detail (40 marks)**

The first thing the user would be prompted to do when initiating the app would be to input some personal details. In a live running app this would include email addresses, contact details for a company to collect user details and store them within a database. This would help the developers to see exactly how many people use their app and what they are called, how old they are, where they are from, how to contact them and many more personalised factors. To keep the simplicity of the groups’, design the group chose to keep this simple and just input 3 inputs of data, in our case Name, age and city.

A simple PHP was created to embed within our design when the time came to collate all designs and put them through the test.

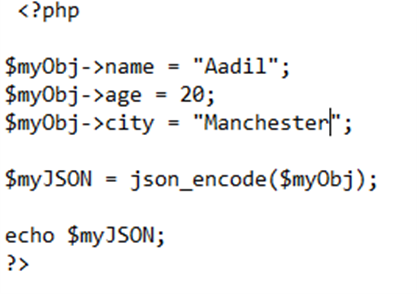


Figure 3

The PHP script which will present the users information back to them so they can confirm their details that they have just inputted within the system. Again, no database storage has been inputted within our design to keep simplicity.

Once the user has created their profile information and the details have been stored on the database in real life app we designed what it would like the user to input next. The app would need the user to sign in using the credentials the user has just created and created using the following HTML code shown in figure 4 and the output shown in figure 5.

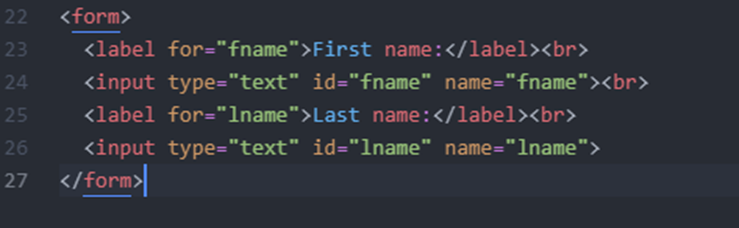


Figure 4

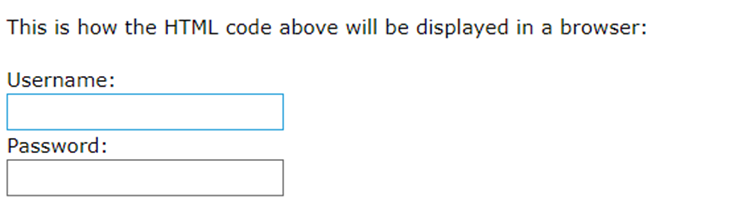


Figure 5

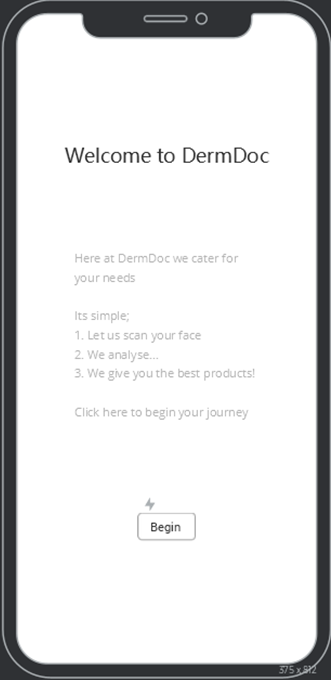


Figure 6 GUI Design of App

When constructing the facial detection code, the first step was to allow the code to be able to the detect the face itself and place a box around the face to ensure the programme knows what a face looks like. The code that was designed attempted to use the users face and show this by surrounding the users face with a box.



The first thing that was done was OpenCV was installed.

Atom was used to write code.

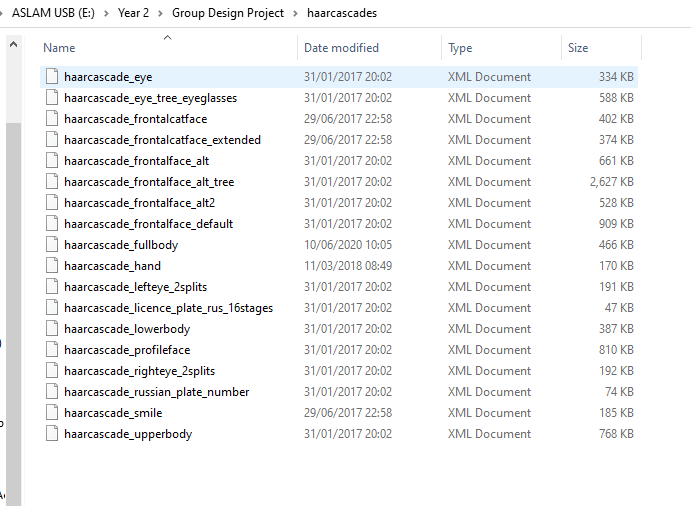
As OpenCV was used for all the heavy lifting it was crucial to add the line ‘import cv2’.

LINE 25, shows that the code was capturing the picture from the webcam of the device that was used.

It needed to be put into a while loop and was told to read the first frame from the video capture method.

Figure 7

The waitkey (1) on line 42 shows that it will read the frame within 1 millisecond.



The first thing the group was load the hard cascade which act as premade models to detect object. The frontal face hard cascade was used for the groups design.

The frontal face hard cascade instantiated within the code. As from research we knew that most machine learning methods work more efficiently when you feed a lower dimensional representation that the problem that is at hand.

Figure 8

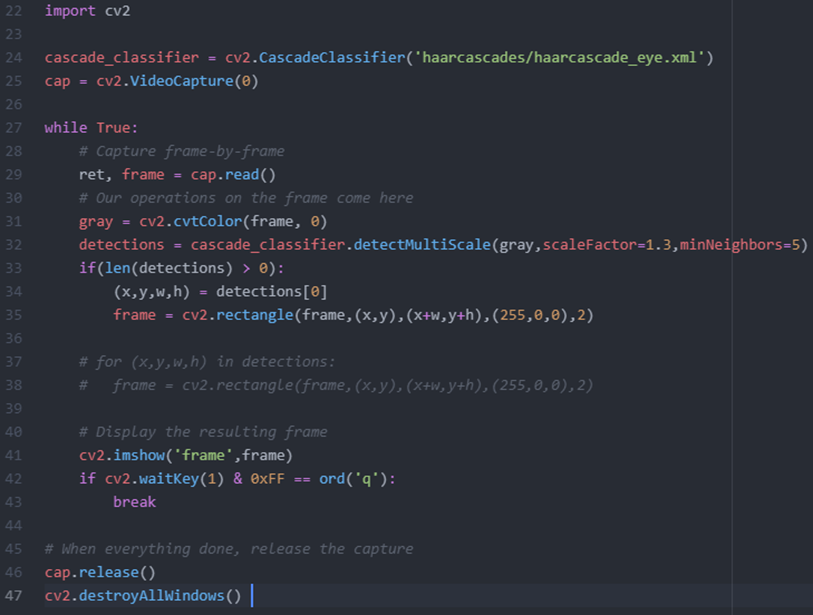
****So, in this case instead of feeding the 3 channel RGB image, rather converting the image to greyscale image that we can give it to the classifier.

Figure 9

We needed to use the detect cascade classifier method by detecting multi scale and setting hyper parameters.

These parameters were used to fine tune our results.

This detector gives all possible regions in image were the code thinks it has detected a face.

We can take the first item from this array and takes the coordinates of largest rectangle around the face.

The one thing that we needed to input was the ability to exit the window and once we exited out the loop we needed to close the video stream.

**App design and GUI**

Whilst creating the frameworks of the initial app the plan was followed to create all the frameworks of design embedded with CSS design and built with some python and Php scripts as an when needed.

An example of this was the loading bar that was created once the face was actually detected shown in figure 7.

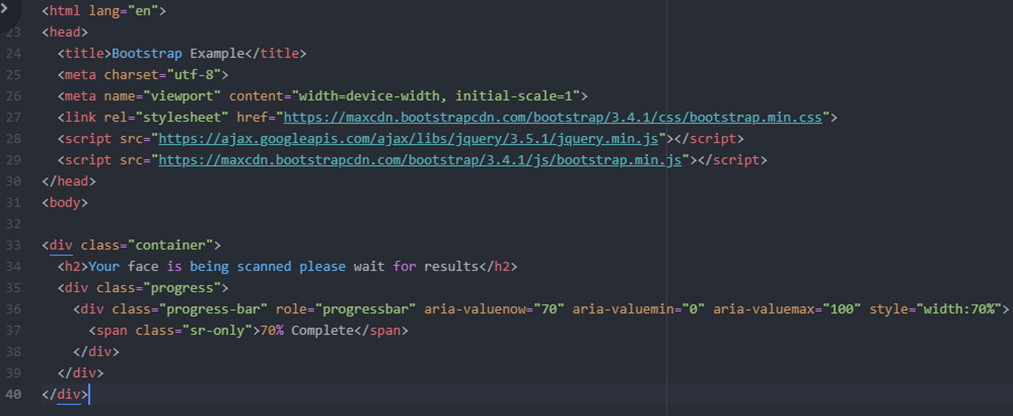
****

Figure 10

A CSS framework was embedded using bootstrap which can bee seen on line 27. This provided and outputted the following progress bar shown in figure 7.



Figure 11

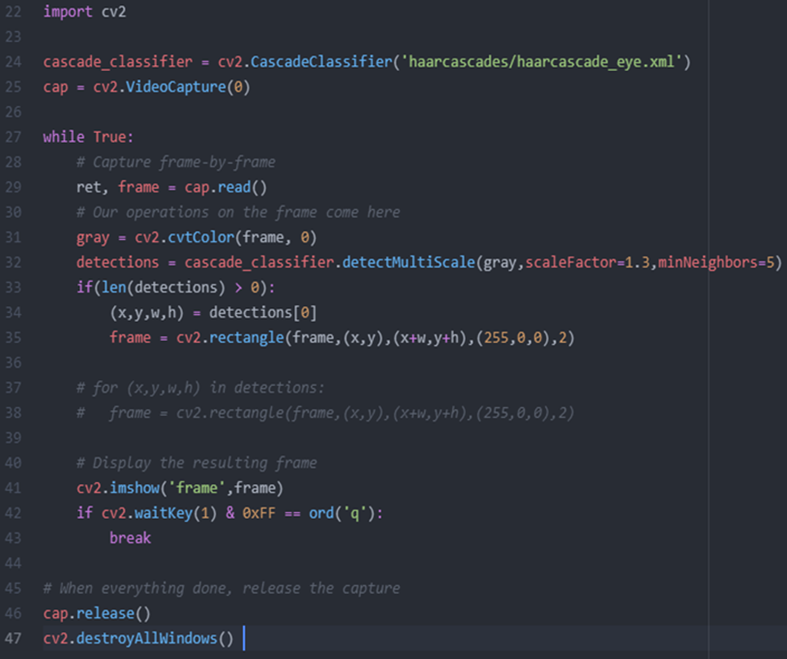
**Testing and Integration (20 marks)**

When testing the overall design, the group chose to test all design elements of the design as they were constructed. For example, when the HTML framework for the loading bar was constructed it was vital the group added this to the HTML index page to see how it will run when using the app.

Likewise, when testing the code for the facial detection it was equally if not more vital as this was the main part of the groups aim. There were certain aspects of the code that needed to be altered and additions to be made.

For example, in the first test that was run with the was without the use of any cascade properties. This meant the code was detecting any object in the frame whether that be a object, a persons hand etc. the main purpose of this test was to initially know that the code can actually scan something/anything and surround that object with a box.

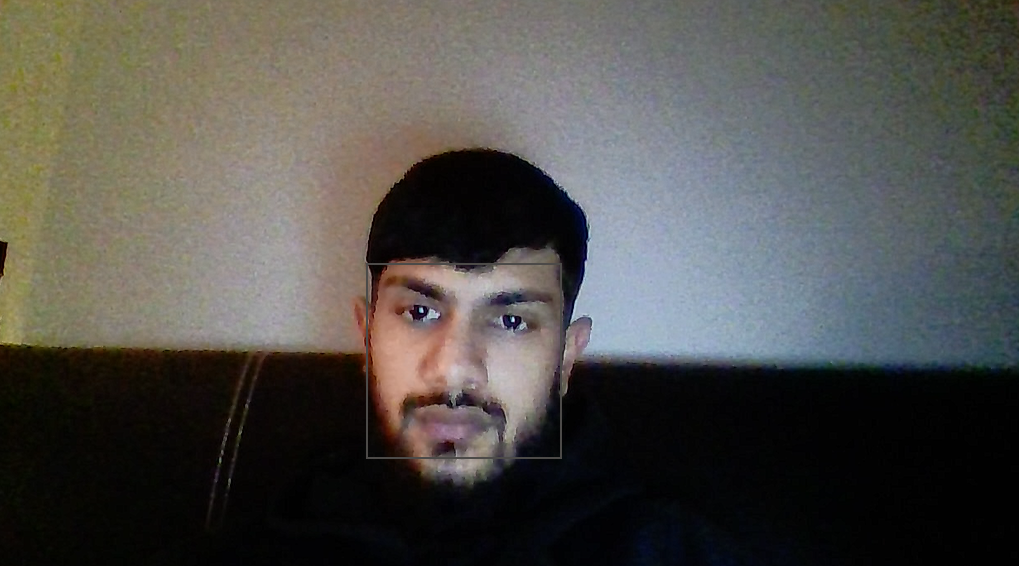
The next test that was completed was with the addition of the frontal face eye cascade which is highlighted in figure 12.



Figure

The output of this code now could actually scan any face that was present on the hardware webcam and not pick up any other objects such as hands, objects etc.

The output is shown in figure 13



Figure

When adding anything else to the frame, e.g., a mobile telephone the output shows that only the users face will be scanned. The key part was established to this part being the eye cascade.



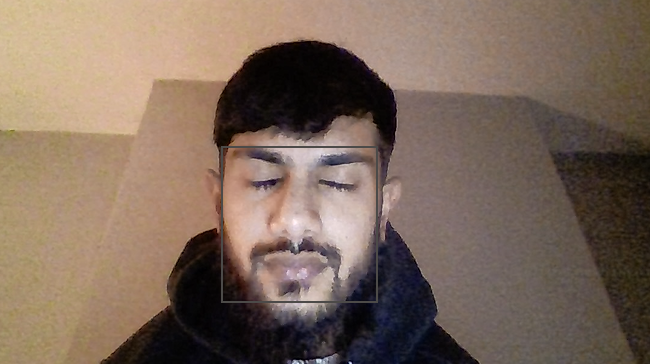
Figure

The next test that was completed was to see if when the user closes their eyes will the code still detect that there is a face as there are certain parameters in the eye cascade which suggests this may not work. Shown in figure 15.



Figure

Ultimately this proved that the eye cascade was they key to our design as there was no grey box surrounding the users face. So, the final stage was to implement both the frontal face and eye cascade to allow the code to scan both when eyes are open and closed. After amending the code figure 16 was the outcome.



Figure

There were a lot of other tests that were undergone to help the other members of the team such as the HTML framework designs, and example of one of them was the HTML script which was integrated with PHP which required the user to input their personal details within the system.

**Conclusion (10 marks)**

This design that we wanted to create initially was tough task from the initial stages and lived up to that expectation. I feel the group did not expect the amount of workload included in this design and the usage of multiple languages that were included. The initial idea was to create a design which could actually scan different features of the face i.e., the ears, nose, face and on them certain features scan for any marks however after construction began the group soon realised this was not a straightforward task as we thought it would be. And changed our aim to simply create a piece of code which could detect a face rather than recognise a face.

With the workload of other modules, the group as a whole found this task challenging but the aim was to complete the main part of the design (facial detection) and other parts such as HTML frameworks as much as we could in time to present.

The matters I would change from this project would be the initial planning of languages to use. For example, If we were to go back to week one we would’ve made all the HTML frameworks first, tested they are working, create a fully functioning design of GUI and then and only then move onto the code.

If the group were to start fresh from this topic again and it would be the groups only focus for the length of say 2/3 months I strongly believe we would have had a finished product nevertheless it is still a feeling of accomplishment knowing the main part of our design i.e., the code was complete and working.