NEA COMPUTER SCIENCE – MOTION DETECTOR

VIVEK MISTRY

CANDITATE NUMBER:

CENTRE NUMBER:

ASHTON SIXTH FORM COLLEGE, TAMESIDE



**Contents Page**

**Introduction .…………………………………………………………………………………………………………………. 3**

**Analysis …………………………………………………………………………………………………………………… 3 -10**

***Why use a computational approach? …………………………………………………………………. 3***

***Existing solutions ……………………………………………………………………………………………….. 4***

***Stakeholders and their uses ……………………………………………………………………………….. 9***

***Key features of my solution ……………………………………………………………………………….. 11***

***Limitations of my solution …………………………………………………………………………………. 12***

***Software and Hardware Requirements for the solution ………….…………………………. 12***

***Success Criteria ………………………………………………………………………………………………… 13***

**Design……………………………………………………………………………………………………………………. 16 - 42**

***Systematic breakdown of problem……………………………………………………………………. 16***

***Structure of the solution……………………………………………………………………………………..18***

***Algorithms………………………………………………………………………………………………………….20***

***Usability Features …………………………………………………………………………………………….. 29***

***Variables/Data Structures/Classes………………………………………………………….………….30***

***Iterative Development………………………………………………………………………………………..33***

***Post Development ……………………………………………………………………………………………. 39***

**Development ………………………………………………………………………………………………………… 43- 84**

***Welcome Screen……………………………………………………………………………………………….. 44***

***Settings Screen ………………………………………………………………………………………………… 48***

***Launch Screen …………………………………………………………………………………………………. 57***

***OpenCV …………………………………………………………………………………………………………… 67***

**Appendices …………………………………………………………………………………………………………… 85 - 86**

***System Diagram [1] .…………………………………………………………………………………………. 85***

***Video Testing [2 – 14] ……………………………………………………………………………………….. 86***

**Introduction**

In my project, I am going to develop a motion sensor using Python and various libraries which can be implemented by any device which has a camera, this could be inbuilt or external. This will have a variety of usage as it can be sold to businesses as well as homeowners and I have a range of stakeholders to take this into account. The key purposes are that it can be used to detect criminals using footage playback and can also act as a deterrent. The reason this project is software based is because there are many hardware products that already implement this which I will research in order to decide which features to implement. It will also be very accessible as many households have devices with cameras built in or if this is not the case, they can always buy a Raspberry Pi Zero to implement this on.

**Analysis**

*In this section, I shall investigate my problem in order to start planning a suitable solution. My problem is that people can`t constantly monitor their house or property which makes it vulnerable to burglaries. Therefore, there needs to be a software solution which is easily to use and can be implemented by equipment already in the house.*

Why Use a Computational Approach?

We can decompose this problem into two parts: creating a GUI and using an OpenCV window processing. The key reason, for using a computational approach is the amount of repetitive processes it will have to do. It will analysis and compare every single frame, which would take so much longer if a human did it. The key repetitive process is that it will have to detect contours in every frame then save it as a list. Another method, would be using image subtraction which compares pixels and outputs an image that demonstrates the differences. This is an abstraction because we are ignoring the features of each frame we don’t need. The human equivalent of this would be marking on contours then reading of a graph to see at which contours the pixels are at which would be subjective. However, if an algorithm does it, it will have an accuracy will cannot be achieved by humans. If I use the matching pixels method, the key repetitive process is that I will have to get each pixel value which cannot be achieved by using non-computational methods within the time frame. Also, in Python, there is already a function within the OpenCV library that can do this, and this would be regularly updated for bugs which makes this reliant. Computers are good at repetitive processes because a modern day CPU can process billons of instructions per second. Using OpenCV function can be seen as decomposition because I am using libraries to aid my program in order to simplify my solution.

Although, a human would not have to do this as they use human judgement, we cannot trust an exact match in the contours or pixels. Therefore, we would have to create a percentage in terms of how many matches there is to the previous frame. For example, if there is 1000 contour matches out of 1050, we would not say motion was detected. The program would decide this my calculating that it is approximately a 95% match (1000%1050 times by hundred). This would simply be calculated be creating a running total of how many times comparison as taken places (in this case, it would be 1050) and how many times a match is found (in the example, it would be 1000). If a human carried out this process, people may disagree whether they match or not as pixels are so small which introduces the risk of human error. If a computer did this it could just use a calculation which has a fixed output due to the same script being ran. This is a simple repetitive process as it would just be imbedded within the loops, resulting in a iterative process, however the numbers will be different every time, but the script will adjust accordingly.

Most importantly, if a human did this, they would have to sit in the same position for a certain number of hours which is simply impractical. This solution would also allow you to use it in different locations as it is a mobile solution as you could simply move the device you’re running the script in a different location which allows you to test the program with different values as the contours or pixels will be different.

Kivy uses Object oriented programming which is good because it allow each screen layout to be split up which means that it allows me to break each screen layout into smaller problems. Using different classes allows each screen to have different variables, buttons and labels that can`t be accessed in any other class.

Existing Solutions

Ring Indoor Camera: <https://en-uk.ring.com/products/mini-indoor-security-camera>

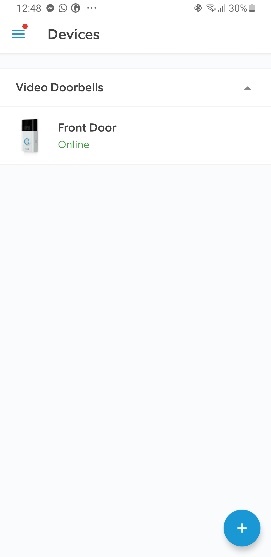
Key Features – HD Videos, Two-Way Walk, Motion-Activated Notifications, Plug in Power, Night Vision, Easy Installation

Price – £50 RRP

Equipment Needed – Ring Protect Subscription, Smartphone (iOS or Android)

How does it work? – You plug it in to a wall socket and then use the Ring Application on Apple or Android to pair the device to the ring device. Then, you can use the app to look at a live feed. If you pay for the “Ring Protect Plan”, it will save clips where motion was detected to their servers which will then be accessible using the event history button in the app. There is also a live view feature which is shown in the screenshot below. Detailed analysis of the Ring Application is below:

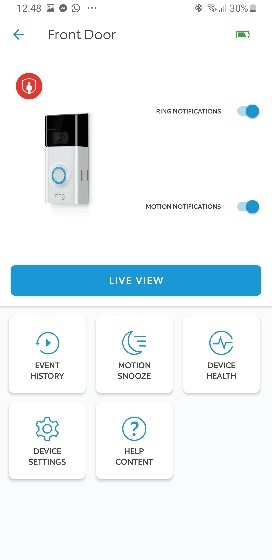
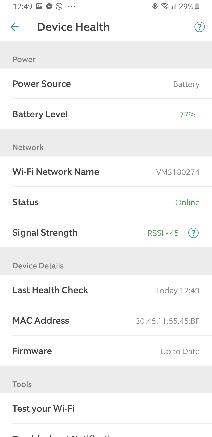
Categories products based upon their function.



Presents list of device that are linked to your ring account and you have the ability to access.

Ability to look at device in more detail.

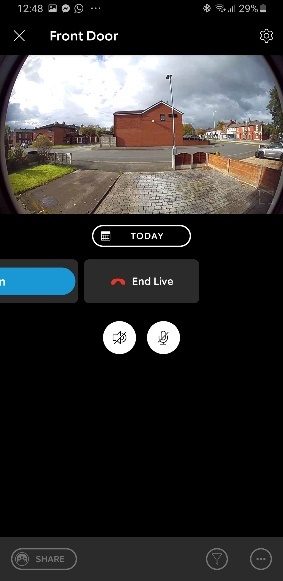
Ability to add additional devices that are part of the Ring eco-system.



Different button to help manage the device. Motion snooze is way to temporarily mute notification. Device help presents details such as the MAC address as well as the Wi-Fi connection and battery life in terms of a percentage.

Button to view live feed of camera. This can be accessed anywhere with the app if you have a stable internet connection.

View battery level. This device can be mains powered or powered by a rechargeable battery.



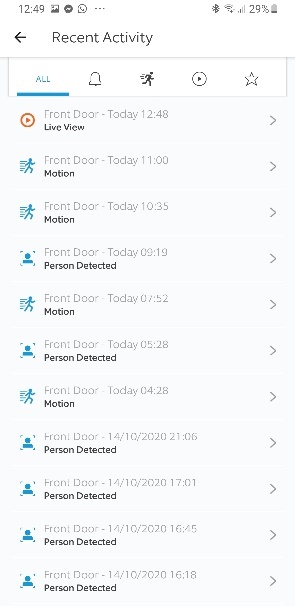
Option to share video clips in case they are required by others.

Option to use the Microphone (Input Device) and the Speaker (Output Device) in order to communicate.

Timeline to see previous video clips that are saved using cloud service to the Ring servers.

Option to exit this mode.

Live video feed.



Can click button to look at video feed. Each motion notification says whether a person is detected as sometimes cars passing by are recorded.

Gives the name of device, time and what event with every entry.

Ability to filter by looking at the type of notification (live view, motion, ring, favourites)

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| 1080p Video | Subscription to record video |
| Motion Detection | Mains Powered |
| Easy to Install | Occasionally laggy software |
| Compact design |  |

Sources: <https://uk.pcmag.com/home-security-cameras/123891/ring-indoor-cam>

<https://www.techradar.com/uk/reviews/ring-indoor-cam-review>

<https://www.stuff.tv/ring/indoor-cam/review>

<https://www.theverge.com/2019/10/9/20906191/ring-indoor-cam-home-security-camera-review-price-specs-features>

<https://www.theverge.com/2019/10/9/20906191/ring-indoor-cam-home-security-camera-review-price-specs-features>

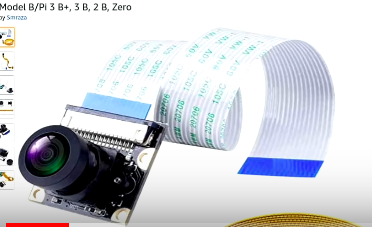
*My solution will include motion detection alerts and it will be very easy to set up as you simplify run the python script and it could be ported over to different Operating Systems as I will use Kivy as my main GUI. The camera quality will depend on the device that the user decides to use but this can always be changes if they want to change the quality by buying an external Webcam which they could plug in via USB or Type C. If I use Local storage, they will not have to spend money to save footage but even if the user decides to use cloud storage, they have unlimited unless they go over the limit but then they would have to pay for extra via their cloud storage provider or delete some older footage depending upon their preference. I shall modify the live view feature as it will not be based upon an application instead it will be based upon a web browser which is better because it will not require storage space and will be able to use on more Operating Systems.*

Motion Pie/MotionEyeOS: <https://www.techradar.com/uk/how-to/build-a-raspberry-pi-cctv-camera-network>

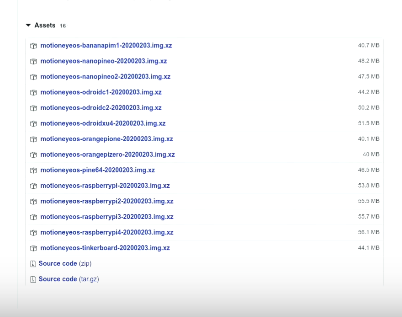
Key Features – Scheduling, Email Notifications, Take Still Imagines, Local and Cloud Storage, Access Media via FTP server

Equipment – Raspberry Pi, Camera

How it works? – How must flash an image file dependant on the model of your raspberry pi onto an SD card. Then either attach a camera directly onto the board using a 15-pin CSI connector and a ribbon cable. Once this is complete, you turn on the pi and enter the IP address shown onto your browser and everything can be controlled from this section. You can connect multiple cameras by: plugging one into via USB and one via CSI, use one pi as the main device and display cameras connected to this as well as other MotionEyeOS cameras.

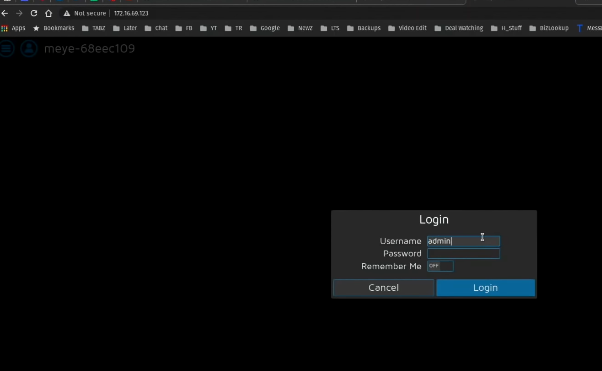


An example of a camera that can be attached via a ribbon cable to a Raspberry Pi as this OS works on most single board computers.

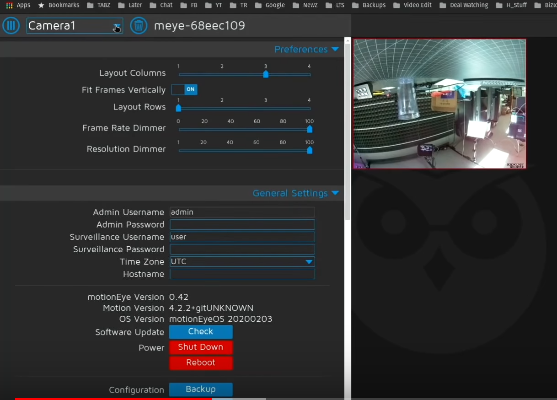


Access the camera by typing the assigned IP address into your browser.

Download image depending on your device which can be downloaded from GitHub for free.



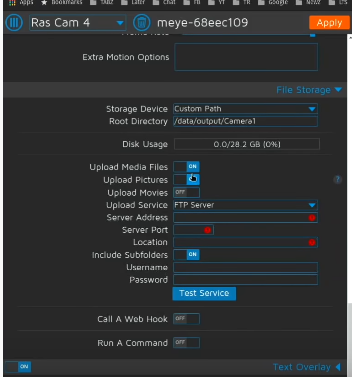
Enter credentials: the default one`s are shown.



Can remotely apply software updates and turn the device off.

This OS allows you to setup multiple cameras and choose between them.

Live video feed.



Test that provider’s credentials written are correct.

Option to add cloud service providers.

Can choose where files are stored locally.

Applies changes by restarting servies.

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| Supports multiple security cameras | Complex to set up |
| Can access footage via “Streaming URL” | Need to buy cameras separately |
| Can work independently – doesn’t need keyboard and mouse | Low frame rate. |

*This product is the closest match to my solution as it is software based. I will attempt to create a Streaming URL so the user can view the cameras when they are aware from the device. However, I will not support multiple security cameras because my intention is for the software to provide a comprehensive solution using one camera however it will be easy to deploy so you can use multiple devices in different rooms. Often laptop and all in one PCs have built in cameras however if this is not the case, user will have to buy these cameras, so this is also a problem for my program. However, I don’t think this is an issue as this is the only thing that needs to be bought so it is still a good value solution. There is a clearly a gap in the market because the closest alternative is MotionPy, but this requires knowledge of how to download an image (.iso file) to an SD card and use a Raspberry Pi. Therefore, I shall create simpler solution.*

*Source:* [*https://dronebotworkshop.com/motioneyeos-raspberry-pi/*](https://dronebotworkshop.com/motioneyeos-raspberry-pi/)

Security Guard:

*This solution is only a valid solution for large businesses due to the cost of this. All technological security items aim to let a range of people access this and ultimately carry out the role of security guards using algorithms in one fixed position. Unless you use multiple cameras to get a greater field of view.*

Stakeholders

As this software will have a range of uses for different people, I have identified a range of Stockholders which this would be suitable for. Generally, this solution would be most useful to property owners who could like to keep rooms under surveillance when they are away. Therefore, the age range would be approximately over 21.This could be implemented in a small scale in house, by just placing it in the bedroom. It could be implemented on a larger scale because different devices could be deployed throughout a building, for example, in a Supermarket, one could be in the staff room and one in the Cash Office. I am going to make the Graphical Interface user friendly which would mean that it will be accessible to anyone who has the equipment required(detailed further in the report).

My Stakeholders are:

* A homeowner that has bought his a house recently. He is concerned about the safety of his property and hopes this would act as a deterrent. – Mr Makinson
* A pub owner that is open from midday to early morning next day. He needs a solution that will work for multiple members of staff due to the rota which means the day is split into the afternoon and night shift. – Mr Mistry
* This Stakeholder is the manager of a hotel that is part of an international chain. The guests come in and out at various times and the reception is not 24/7. Therefore, they need to check no one unauthorised enters certain areas. – Mrs Mistry

I have created a set of question to help me understand how they will use my solution.

Q1. How would you prefer to activate it (Pattern, Pin, Password, etc)?

*Enables me to design the login screen based upon the user’s preferences.*

Q2. Where would you place the device?

*Enables me to think about the most suitable ways to compare frames (e.g. Detecting Contours or Comparing Pixels)*

Q3. Where would you want the footage to save (local or cloud). If cloud, which provider do you prefer (iCloud, Google Drive, OneDrive, etc)?

*Enables to know which Cloud API to use and whether I need to use built in OS module for local storage.*

Q4. How long would you need to deactivate the detector, before the alarm goes off?

*Enables me to know how much of a delay to have before playing the audio clip and when motion is detected.*

Q5. What Surveillance Products have you use before?

Q6. What are the advantages?

Q7. What are the disadvantages?

*Enables me to understand which features are the most useful for the Stockholders based upon pasted experiences.*

Q8. Why would you use my product?

*Enables me to ensure my product is designed with my stakeholders in mind.*

The responses have been recorded in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | *(Homeowner)* | *(Pub Owner)* | *(Hotel Owner)* |
| *1* | *Password* | *Pin* | *Pattern* |
| *2* | *Hallway* | *Cash Office + Behind the Bar* | *Reception + Lobby* |
| *3* | *Google Drive* | *Local + Google Drive* | *Google Drive* |
| *4* | *15 Seconds* | *20 Seconds* | *30 Seconds* |
| *5* | *Ring Camera – Outside their front door* | *Video Doorbell – Personal Use at Home* | *CCTV - At entrances to hotel.* |
| *6* | *Capture footage if robbed.*  *Serves as a deterrent.* | *Easy to set up.*  *Mobile app regularly updated.*  *1080p Video Feed.* | *Clear view.*  *Easy to Access via Mobile App.*  *Good while you are away from property.* |
| *7* | *Lots of notification any time motion is detected.*  *Expensive.* | *Have to pay for yearly subscription. for full functionality.*  *Battery Powered so it needs charging.* | *Expensive set up cost.*  *Hard Drive takes space.*  *Delicate to set up.* |
| *8* | *Makes me feel safer and more secure.* | *To protect valuable assets such as Till.*  *To get a good value solution to monitor rooms when the bar isn’t open.* | *It will take up less space.*  *Use equipment that the Stakeholder already has.* |

Key Features

Breakdown of program:

1. Setup Screen – Display key information and configure settings.
2. Main Screen – How to activate detector.
3. OpenCV Tracking Window – The software while the key feature is in operation.

1. In the welcome screen, I will use a Label (textual output) which would display the IP address for the user to access the webcam feed remote as this is part of the success criteria and it needs to be easy to use for the Stakeholder. Also, there will be button (set input) which will select the cloud service provider. Google Drive allows the user to sign in via the web browser if the json file is saved in the directory which will output in the default browser selected on the machine which means that I will create a button in order to activate this. I will do this as this is the cloud service provider which all my stakeholders prefer.

2. The main screen could contain a variety of options depending on the Stakeholders requirements. If I decide to use a pin, I will use a button which will be binded to a function and a label which will act as a display. The code will be outputted as text as they enter it so they know what they have typed. I will use a pin as each stakeholder wanted something different and I think the pin is the best match for all of them. This is because a password is prefer as it is could be viewed as secure due to a wide possibility of combinations but if you have a pin you have 10,000 combinations anyway. A pattern would be preferred as it is easy to use although it would take longer to input. The reason a stakeholder wants this, is because it is easy to enter, which the pin will also be as you can just enter it using the fixed input keypad.

3. The OpenCV will show a real time webcam feed which will be analysed frame by frame using an iterative process. To exit this window, I could use a character such as “Q” in order to exit this. This is where most of the complex processing will take place as this is the key feature of the program. The percentage change between frames will be calculated and if it drops below a certain value it will return to the security screen using a conditional statement. If the key isn’t entered within a certain amount of time, the audio will be outputted using the device speakers, this creates a motion alert.

|  |  |
| --- | --- |
| Key Features | Justification |
| Setup Screen | * To allow users to configure the program how they want so they can be satisfied with the program. * They can understand how to use the program by reading the key information. * Sign into the cloud service provider using the web browser which looks like a generic Google Login screen which makes it easier to use. |
| Main Screen | * Allows the user to input data. * Allows the user to navigate the program through easy to use buttons. * Using button restricts use of a keyboard which is easier. |
| OpenCV | * Allows computer to perform calculation to view motion detection. * Allows frame to be saved to a file in order to create a video. * Allows computer to view webcam. |

Limitations

One of the key limitations is related to storage space. The video clips will be saved to the local hard drive. Once this runs out of storage and there is not enough space, Python will run into run time errors. Therefore, the user will have to delete footage using the file explorer or I can program the script to automatically delete the oldest video file if needed. This may also be a problem when using Cloud Service Providers as you can often have a limit although you can increase this for a fee. However, this would be the user choice whether they want to purchase it and the same alternates will apply like with Local storage. The user can fix this by simply deleting old files after a certain amount of time or there may even be programs to do this.

Another issue is the number of users as the security file will be saved as a text file securely by being hashed. However, you could always overwrite the security details if needed. This will not be a problem for the cloud storage because you can just sign into another account if required. Therefore, if you share the system, the people who use it will all have to be aware of the security details. This will mostly be a problem when being used for businesses. However, this software is designed for one user and the PIN can just be distributed to everyone who requires it.

Finally, when comparing my solution to Motion Pi earlier in the report, I mentioned the inability for multiple cameras. However, this is just a motion detector designed for one part of a building. However, this software could be a small element of a CCTV system which would allow access multiple cameras. My final program could be used as a library to implement this.

Software and Hardware Requirements \*Optional requirements – will not limit key functionality.

Software Requirements:

|  |  |  |
| --- | --- | --- |
| **Equipment Required** | **Why is this required?** | **What will I use for testing?** |
| \*Another device with a Compatible Web Browser | To view real time footage using an address provided and entering this into your web browser. | Samsung S8+ (Android 9) with Firefox for Android |
| A desktop environment compatible with Python3 (Windows, Mac OS, Linux, etc) | In order to run Python script and corresponding packages. | Windows 10 |
| \*Media Player with .mp4 compatibility. | To view video that have been saved to the local drive. | VLC Media Player for Windows |

Hardware Requirements:

|  |  |  |
| --- | --- | --- |
| **Equipment Required** | **Why is this required?** | **What will I use for testing?** |
| Mouse | In order to navigate the graphical user interface by clicking on buttons. | Built in Trackpad |
| Keyboard | Enter user id and password if using the cloud option and to exit the open CV window using a button. | Built in Keyboard |
| Webcam or External USB Camera | In order to process frames and to detect motion. | Built in Webcam |
| Monitor/Screen | To view webcam feed while OpenCV window is open. See Graphical User Interface so they can interact with it. | Built in Display |
| Speakers | In order for the audio to play when the motion is detected and if the PIN is not entered in time. | Built in Speakers |

Other Requirements:

Google Drive account to save video clips.

Kivy Minimum Requirements (GUI Framework): (**Source: Stack overflow)**

* 1GHz CPU (32/64 bit)
* 1GB of RAM
* 128MB of graphics memory.

Python Libraries required for Open CV: **(Source: Official Documentation)**

* Numpy
* Matplotlib

Success Criteria

|  |  |  |
| --- | --- | --- |
| **Requirements** | **How to prove?** | **Why?** |
| The stakeholder should be able to enter pin to active the motion detector. | Screencast of a Stakeholder entering a pin which then launches the OpenCV window. | This ensures the users knows the pin before starting the detector. This ensures that they will be able to deactivate the alarm. |
| The stakeholder use be able to choose security details when setting up the system. | Screencast of a Stakeholder entering a security details during setup which should then save it securely. | This enable the users to set a PIN that is memorable and is safer than using a default pin. This is because anyone who knows about the software could deactivate the alarm with the default pin even if they are authorised to use it. |
| The stakeholder should be able to sign in to Google Drive when setting up the system. | Screencast of Stakeholder signing into Google Drive during setup. | This enables to users to sign into a platform that they may already have, avoiding another account needing to be created. This is also more reliant as Drive regularly receives patches so it is secure and results in one less risk in my program and maintenance of my own server. |
| If motion is detected, Stakeholder should hear a noise after 20 seconds if the pin isn’t entered. | Screencast of motion being detected, and a sound being played. | This should act as a deterrent for anyone unauthorised that has entered the zone. This also alerts anyone nearby so they can call someone appropriate (e.g. Shift Manager, Police, etc.) |
| If motion is detected, Stakeholder should not hear a noise after 20 seconds if the pin is entered correctly? | Screencast of motion being detected and the pin deactivating the alarm so a sound is not played. | The motion would be detected even if an authorised person comes in. The PIN pad should be displayed to allow the user to deactivate the alarm so the alert doesn`t go off as it would not be needed in this scenario. |
| The stakeholder should be able to view the file of when motion was detected using their file browser. | Screencast of the user clicking on the file and playing it once Python saves the file. | This is designed in case the user needs to review footage and is one of the ways of accessing the footage. |
| The stakeholder should be able to download the file of when motion was detected using their Google Drive account. | Screencast of the user clicking on the file on their Google Drive and downloading it. | This enable the user to access the footage elsewhere in case the local hardware device becomes corrupts or they are unable to access the device. |
| The stakeholder should be able to view the live feed on a browser on another device when the alarm is armed which can be accessed by the address on the settings screen. | Screencast of the user using another devices web browser to view the window. | This allows any one authorised to see the footage remotely. This can be seen from any device that has an active internet connection. |
| When the alarm is armed, the user should be able to see a live feed on a window on their screen? | Screencast of live video feed being shown once the alarm is activated. | This can be seen as indicator that the alarm has been activated. It also allows the user to position the device in the correct way in order to get the correct field of view. |

The way user feedback will be monitored is by three surveys, one is for the setup stage, one is related to operating the system and the final one is about accessing the files which are saved.

*Questionnaire – Setup*

*Q1. How easy to it to select a pin?*

*Q2. How easy was it to sign into your Google Drive account?*

*Q3. Did the instructions give you enough guidance to set this up?*

*Questionnaire – Usage*

*Q1. Can you activate the alarm by activating your pin?*

*Q2. Did the alarm deactivate when you entered your pin?*

*Q3. Did you have enough time to deactivate the alarm?*

*Q4. When you didn’t deactivate the alarm, did you hear a sound?*

*Q5. Can you view the live feed on your other device?*

*Q6. Are there any improvements that could be made to the user interface?*

*Questionnaire – Viewing footage.*

*Q1. Can you view the footage on your device that was recorded?*

*Q2. Is the quality of the video satisfactory?*

*Q3. Can you download the footage from the cloud provider?*

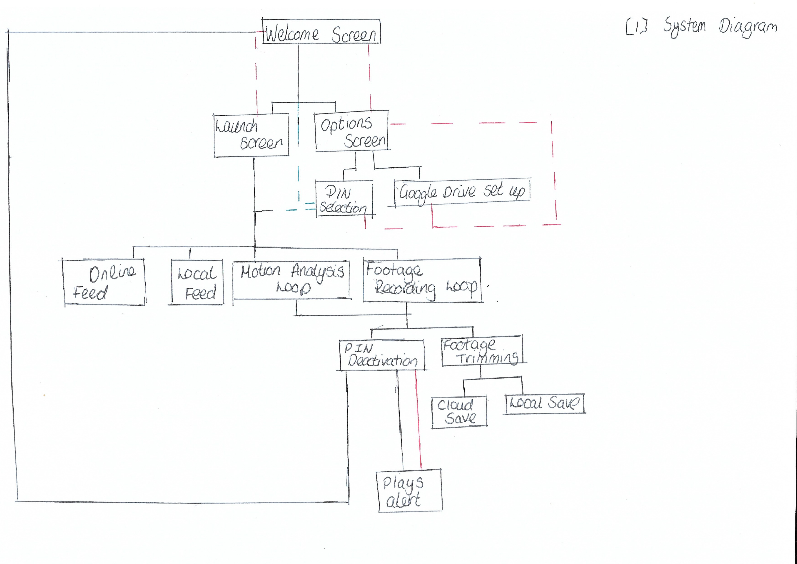
*Q4. Is the footage on the cloud, the same as what is saved locally?*

*Q5. Are the files easy to find locally and on Google Drive? If not, where would you prefer this to be saved?*

**Design**

After analysing my problem, I have decided on my success criteria with Stakeholders in mind. Therefore, I will go on to design by solution with my success criteria in mind.

Systematic Breakdown of Problem



*\*Figure one in the Appendices shows my System Diagram which will be explained below. It explains how each sub routine will interlink and I have justified below why I have done it like this. In order to break it down, I have shown how each part connects up.*

The “Welcome Screen” is required as it will immediately have instructions which explains to the user how to use the software which is key to a smooth user experience especially for my two stakeholders who are business owner as this would allow all members of staff to use the software if required.

From this screen, you will be able to choose whether to launch the motion detector software or whether you need to access the settings. Once you access the settings screen, there will be two options. One will allow you to reset your PIN and one will allow you to change the Google Drive account associated with the Program. This is particularly important from the Stakeholders perspective in case someone who shouldn’t have the PIN gains access to it or their Google account has been comprised which would make the Program more secure.

The dotted red path shown on the diagram shows the ability to navigate back and forth between windows as the option screen allows you to get to the two variables needed for the program to run successfully: the pin which will be saved in a text file as a hashed string and the Google account credentials which will be acquired using the Java Script file which will launch a login screen in the users default browser.

The “Launch screen” will display the keypad. This is required before the OpenCV module is used to do all the image processing as it confirms that the user knows the pin, I think this is valuable as it decreased the changes of the user forgetting the pin and triggering the alarm after a motion event. The dotted green path will be followed if the program attempts to run the “Launch Screen” but there isn’t a configuration text file which contains the hashed pin. This is appropriate to the program as the user will be stuck when the user is asked to the pin once motion is detected and they return.

Once the data that is required to run the program is validated, the key part of the program is ran. This is split into many programs. There is the online feed which is shown and this is required to meet the stakeholder requirement that allows them to view it on the browser on another device. This will require a library such as ImageXMQ and that is why it has its own function. The local feed will simply be using the OpenCV library and display every frame in a window on the host device. The Motion Analysis loop is needed to analyse each frame compared to the previous using image subtraction and looking at the amount of coloured pixels. The footage recording loop is required to save the video which then can be stored in two places and this function is key in case the user needs to see what triggered it.

Then, once the detection loop is broken, the keypad will be displayed and it will wait for 30 seconds which has been decided with Stakeholders in mind. Once, that occurs the footage will be trimmed so only the part where the motion is detected is saved to the cloud. This is key for user experiences as it limits storage space usage which is important as the users don’t need footage of nothing happening, they just need to see why a motion event has occurred.

Finally, if the PIN isn’t entered within the allocated time, it will play an alert from the speakers. The solid red line shows what happens if this doesn’t happens, the keypad will stay there and it will wait for user input to deactivate the detector. If the detector is deactivated in time, it will go back to the Welcome Screen ready to use again. This is particularly important if used on a device which is only used to run this software such as a Raspberry Pi Zero which could be very useful to the Stakeholders as it allows them to use it discretely and become a very good value solution compared to other products listed in the analysis stage.

|  |  |
| --- | --- |
| **Part of System Diagram** | **Justification** |
| Welcome Screen | * Display instructions as soon as user opens program. * Allows the user to run the main program or configure the program. |
| Option Screen | * To choose what part of the program to configure. |
| PIN Selection | * Gives the user an option to choose a PIN that they can remember. |
| Google Drive Setup | * Launch web browser so the user can login to Google Drive. |
| Launch Screen | * User can enter the PIN to verify that they know it. * Allows the user to run the main program- the motion tracking window. |
| Online Feed | * Allows the user to look at the feed on another device. |
| Local Feed | * Allows the user to look at the feed on the device it is running. |
| Motion Analysis Loop | * Allows the program to compare frames. * Allows the program to trigger the alarm if needed. |
| Footage Recording Loop | * Allows the program to collate frames. * Temporarily store the full video file. |
| PIN Deactivation | * Allows the program to check whether the designated user is there. * Allows the intended user to deactivate the alarm. |
| Plays Alert | * Acts as a deterrent to any unauthorised people. * Play audio if PIN isn’t entered in designated time. |
| Footage Trimming | * Trim file so only useful parts are saved. * Optimises use of storage space. |
| Cloud Save | * Saves trimmed footage to Google Drive. * Automatic backup of footage. |
| Local Save | * Allows the user to view files locally. |

Structure of Solution

Welcome Screen: This will be a class within the Kivy script and be managed my window manager which will allows us to move between different Kivy windows. There will be a Label contain multiple strings. The Label which display instructions on how to use the program and the IP Address for web access. This is the only place where this data will be available because this screen which appear on launch so this is the most logical part for this information. This relates to the “Online Feed” function as this is what user need in order to access the feed. Also, it will have two button which will navigate to either the Launch Screen or Options Kivy class. We can use the inbuilt OS library to get a static (which means it will not change) IP address for the window screen.

Options Screen: This class is separated as the functions aren’t callable from the Launch Screen because I think that it is inefficient to load the configuration file in the Welcome Screen if the user doesn’t intend to change the PIN. This option can be used to go up back to the “Welcome Screen” which is essential because the user may want to run the surveillance after changing the settings. It allows the function to use the PIN Selection class or the Google Drive function. This class will continue running once Drive is clicked and can be accessed by going back after you select your PIN.

PIN Selection: This screen will display a keypad using Kivy buttons. There will also be a Label which will be where the pin shows as the user enters it. This will be limited to 4 digits because of the time limit to deactivate the alarm. This screen will be accessible from the Launch Screen via a selective statement if the program can`t find a configuration containing the configuration text file containing the PIN. The buttons can be split up into 3 further functions: one that puts them on the display, one that allow you to delete one digit and one that allows you to save in a hashed file when entered which makes it more secure.

Google Drive Setup: This function will use a JavaScript file which is saved from the Google Developers website to launch a login page linked to the NEA application. This is more efficient as it provides a 1st party method of collecting the data which will result in a reduction in the size of the program and also can deal with issues such as user forgetting their password or two step authentication. This will be done from a Python Script with functions dedicated for Google Drive as this allows it to be used in multiple classes.

Launch Screen: This will be the same class as the “PIN Selection” as the display will look the same so it reduces the length of the code. However, once the PIN is verified it will launch the OpenCV functions which will be in a different script where all the image and video processing will take place. This is only method of launching the OpenCV script which is more secure as it prevents any users from using it without knowing their PIN which would cause issues when deactivating the sensor.

Online Feed: This function will also be accessible once the local feed function is ran as this is what is displayed to the web browser. This is one of the only independent functions as this is not a base requirement but the Stakeholders would like a way to remotely monitor this device.

Local Feed: This function will find a camera on the selected device and this is linked to the Motion Analysis and Footage Recording Loop as the frame from camera are the key variables in the processing. This function will launch a window which shows the each frame from the camera which produces a video feed.

Motion Analysis Loop: This function will use an in built OpenCV function which looks for differences in pixels between the current frame and previous. Then, I can use another in built function that looks for pixels that aren’t black. Then compare it to my match percentage as an object like a piece of paper may due to wind and this shouldn’t trigger an event as it is not a Stakeholder requirement as they only want people to be detected.

Footage Recording Loop: This function will take each frame and combine it in order to produce a video. This will be trimmed later on but we don’t know when the alarm will be triggered so we must record it all temporarily. This can only be accessed after a Motion Event is triggered otherwise there will be no video to record.

Footage Trimming: This function requires data from the “Motion Analysis Loop” as we need to know when the motion was detected and start recording a few seconds before that because this maximises storage space on the host device as we don’t use save video of a still frame. This is only accessible after the recording loop as this creates a variable which is what will be trimmed.

Local Save: This step is not essential but I think it is key to use a logical file name so we can use the date and time of the motion event which means we can use the variable from the footage recording loop. We will use the same variables that is needed in Footage Trimming.

Cloud Save: This will use the User Credentials which has been acquired from the Google Drive set up function. Again, this isn’t a key requirement so we can use a try loop to avoid further errors. This loop would only not work if there is no credentials and this isn’t a problem as some user may just want to save it locally. The reason for two possible save locations is in case files become corrupt or you would like to access it from another device.

PIN Deactivation: This is only accessible after the motion activation stage because you don’t need to deactivate it unless an event occurs. There will be a keypad and a label which allows you to look at the code as you enter it. This will be accessible from the Kivy Script under a class. The program will wait 30 seconds until waiting for the next function to run. This is to give the user enough time to deactivate the alarm. Once it is entered correctly it will return to the Welcome Screen class.

Plays Alert: This is used is to deter any intruders. This will continue playing until the PIN is entered correctly. The way this will work is by using the inbuilt Kivy audio function which has play and stop. This is only accessible after 30 seconds and the PIN not being entered as the alarm could be triggered from the user returning and in this case it is not required.

Algorithms

Welcome Screen:

*.kv file (Welcome Screen class)*

GridLayout:

Label:

Text: “For web access: IP Address - ” + self.os.getstaticip()

“Please click on settings to set your PIN and link your Google Account.”

“Once activated, you have 30 seconds to clear the frame and 30 seconds to deactivate the alarm.”

GridLayout:

Button:

Text: “Launch”

On\_Release:

self.launch\_verification ()

Button:

Text: “Settings”

On\_Release:

window\_manager = settings\_screen

*python script*

IMPORT os

PROCEDURE launch\_ verification DEFINE

IF os.find(“config.txt”) == FALSE THEN

window\_manager = pin\_selection

ELSE THEN

window\_manager = pin\_verify

*The .kv file uses the built in os function in order to get a static IP address which is needed for the user to access the video feed. It also changes screen if the Settings Button is pressed and it runs a function within the Python Script if the Launch Screen is pressed. The python script contains a function that verifies whether a PIN exists so the user can be rediverted if they haven’t created a PIN yet.*

Options Screen:

*.kv file (Options Screen Class)*

GridLayout:

Button:

Text: “PIN selection”

On\_Release:

window\_manager = pin\_selection

Button:

Text: “Google Drive Login”

On\_Release:

self.drive\_login()

*python script*

IMPORT drive\_script

PROCEDURE drive\_login DEFINE

drive\_script.login()

*This algorithm shows that there is two buttons. Within the .kv there is two buttons, the PIN selection button will change the screen to PIN Selection. The Google Drive Button is linked to a function that runs the login function. I have used a separate file in order to handle all Google Drive related code to make it easier debug. This is also more logical as it allows me to program the Option Screen and get that working fully first.*

PIN Selection:

*.kv file (PIN Pad Class : This class also applies to “Launch Screen” and “PIN Deactivation”)*

GridLayout:

Label:  
 Id: display

Text: “Enter a four digit PIN.”

GridLayout:

Cols: 3

Button:

Text: “1”

On\_Release:

self.number\_entered(1)

Button:

Text: “2”

On\_Release:

self.number\_entered(2)

Button:

Text: “3”

On\_Release:

self.number\_entered(3)

Button:

Text: “4”

On\_Release:

self.number\_entered(4)

Button:

Text: “5”

On\_Release:

self.number\_entered(5)

Button:

Text: “6”

On\_Release:

self.number\_entered(6)

Button:

Text: “7”

On\_Release:

self.number\_entered(7)

Button:

Text: “8”

On\_Release:

self.number\_entered(8)

Button:

Text: “9”

On\_Release:

self.number\_entered(9)

Button:

Text: “Enter”

On\_Release:

self.completed()

Button:

Text: “0”

On\_Release:

self.number\_entered(0)

Button:

Text: “”

On\_Release:

self.backspace()

*Python Script*

IMPORT hash

IMPORT time

display\_list = []

PROCEDURE number\_entered DEFINE *(This procedure also applies to “Launch Screen” and “PIN Deactivation”)*

IF length(display\_list) == 4 THEN

text.display = “Digit limit reached”

time.sleep(2)

ELSE THEN

display\_list.append(x)

text.display = displayasinteger(display\_list)

PROCEDURE backspace DEFINE *(This procedure also applies to “Launch Screen” and “PIN Deactivation”)*

IF NOT length(display\_list) == 0 THEN

display\_list.pop()

text.display = displayasinteger(display\_list)

ELSE:

text.display = “Nothing has been entered.”

PROCEDURE completed DEFINE

IF length(display\_list) == 4 THEN

file = (“config.txt”,”w+”)

file = hash(text.display)

file.close()

ELSE THEN

text.display = “Must be 4 digits”

time.sleep(2)

text.display = displayasinteger(display\_list)

*The .kv file is used to set up all the buttons. I have used buttons as it is easier to input and creates less errors as you cannot input any invalid data such as letters and special characters. This class has been split up into three methods. Number entered is used to register a user input. However, it must valid the length first so it doesn`t go over 4 digits. Then it adds it’s to a list. I have used a list because they are easy to manipulate as I can use pop and push which are inbuilt libraries. Then, it appears on the label at the top which acts as a display. The function backspace verifies there is nothing on the list before attempts the list as this would lead to an error. Completed is used when enter is pressed, this hashes the password and saves it to a file if it is 4 digits. Otherwise, it displays an error message.*

Google Drive Set Up

*Python script*

IMPORT pydrive

FUNCTION login DEFINE

google\_login = pydrive.Authentication()

google\_login.BrowserAuthentication()

credentials = GoogleDrive(google\_login)

*This is on an external file which will be imported and this function will be ran. It used the built in authentication function to launch a web browser in order to allow the user to sign in to the program. Then, the user details are saved to a variable called “credentials”.*

Launch Screen

*.kv file (PIN Pad Class - same as in PIN SELECTION)*

*python script*

IMPORT hash

IMPORT opencv\_script

FUNCTION completed DEFINE

file = (“config.txt”,”r”)

IF hash(text.display) == file THEN

opencv.main()

ELSE THEN

text.display = “Incorrect PIN. Please try again!”

*This uses the same code in the kv file as shown in PIN selection as it looks the same. The only method that is different is completed. This is because it has to read the file which contains the hashed pin and compare it. If the pin is correct, it will allow it to open the OpenCV window. All operations related to the OpenCV library will be done on another Python Script. This is easier to debug because if there is problem with the video processing I can just debug the OpenCV file which means there is less code to look through which makes it more efficient.*

Local Feed

IMPORT OpenCV

IMPORT Time

FUNCTION local\_feed DEFINE

WHILE True:

source = OpenCV.get\_source()

frame = source.load\_frame()

OpenCV.display\_window(frame)

RUN PROCEDURE online\_feed(frame)

RUN PROCEDURE footage\_recording(frame)

RUN PROCEDURE motion\_analysis(frame, previous\_frame)

IF motion\_analysis() RETURN FALSE:

time.sleep(10)

RUN PROCEDURE footage\_trim(time.gettime())

BREAK

frame = previous\_frame

*This is part of the OpenCV file. This is main part of file as all operation must occur as part of this OpenCV loop. Every time a new frame appears it must be processed the same way. Which is why there is a lot of function within this and the Pseudo code for these are below. The code gets the video source, gets a frame. If the last frame, is different to the current, which will be found out using the motion\_analysis function. The loop will break which means that motion has been detected.*

Online Feed

IMPORT web

IMPORT OS

PROCEDURE online\_feed (frame) DEFINE

web.bind\_address(os.getstaticip)

web.refresh\_frame(frame)

*This function takes it the parameter frame which is then displayed on the web page by refreshing the frame and putting a new frame on. This is done so quickly it appears as a video.*

Motion Analysis

IMPORT numpy

PROCEDURE motion\_analyis (frame,previous\_frame) DEFINE

frame\_prepared = OpenCV.GuassianBlur()

previous\_frame\_prepared = OpenCV.GuassianBlur()

differences = cv2.subtract(frame\_prepared,previous\_frame\_prepared)

image\_comparision = differences.getBlackPixels()

IF (image\_comparision/frame.dimensions (height) \* frame.dimensions (width)) < 80 THEN

RETURN False

*This procedure requires the NumPy function because that is the data Stucture that the built in function “subtract” saves it in. A Gaussian Blur is applied because this will smooth the image which results in image noise being reduced. The program doesn’t look for an exact match because there may be simple inconsistences such as a change in lighting or glare. Therefore, it creates a percentage based upon the amount of pixels.*

Footage Trimming

PROCEDURE footage\_trim(time) DEFINE

file\_name = time + date + “.mp4”

start\_trimming\_from = time – 30

RUN PROCEDURE local\_save(file\_name, start\_trimming\_from)

*This procedure creates a suitable time name by using the date and time as well as using the .mp4 file extension to save it. Then it uses the time and plays from 30 seconds before it to get the person in the recorded footage. Then these variables are passed in to the parameters to the next procedure called “local save.”*

Local Save

IMPORT ffmpeg

IMPORT drive\_script

PROCEDURE footage\_trim (file\_name, start\_trimming\_from) DEFINE

ffmpeg.trim(file\_name , start\_trimming\_from , time)

RUN PROCEDURE drive\_script.cloud\_save(file\_name)

*This procedure trims the footage using the ffmpeg library. The parameters needed for .trim are: file that needs to be trimmed, start time, end time. Then, this file is passed in to be processed within the Google Drive file so it can be saved on the cloud as well.*

Cloud Save

PROCEDURE cloud\_save(file\_name) DEFINE

try():

save\_on = pydrive.credentials

folder = “root/motion detector”

file = open(file\_name, “r”)

pydrive.save(save\_on,folder,file)

*The file is saved to the cloud using credentials that would have been set up using the Google Drive setup which would have been activated during the setup screen. It saved in a folder within their drive called motion detector to keep it organised and make it easy to access. Try is used because all users may not want to save it this way so if this doesn’t work it will continue with the program preventing further errors.*

PIN Deactivation

*.kv file (PIN Pad class: same as PIN Selection)*

*python script*

IMPORT time

FUNCTION completed DEFINE

start\_time = time.gettime()

file = (“config.txt”,”r”)

WHILE True:

IF hash(text.display) == file THEN

window\_manager = “Welcome Screen”

RUN PROCEDURE play\_alert(stop)

ELSE THEN

text.display = “Incorrect PIN. Please try again!”

IF start + 30 => time.gettime() THEN

RUN PROCEDURE play\_alert(play)

*The .kv file is the same as the PIN selection Pseudo Code. The only method that is different is complete because it needs to move back to the welcome screen if the PIN is correct. It will also stop the audio using a different function. If not, it will display an error message. Once, it goes 30 seconds after the screen was launched it will run the play alert screen.*

Play Alert

IMPORT kivy.audio as audio

PROCEDURE play\_alert(operator) DEFINE

if operator == stop:

audio.stop()

elif operator == play:

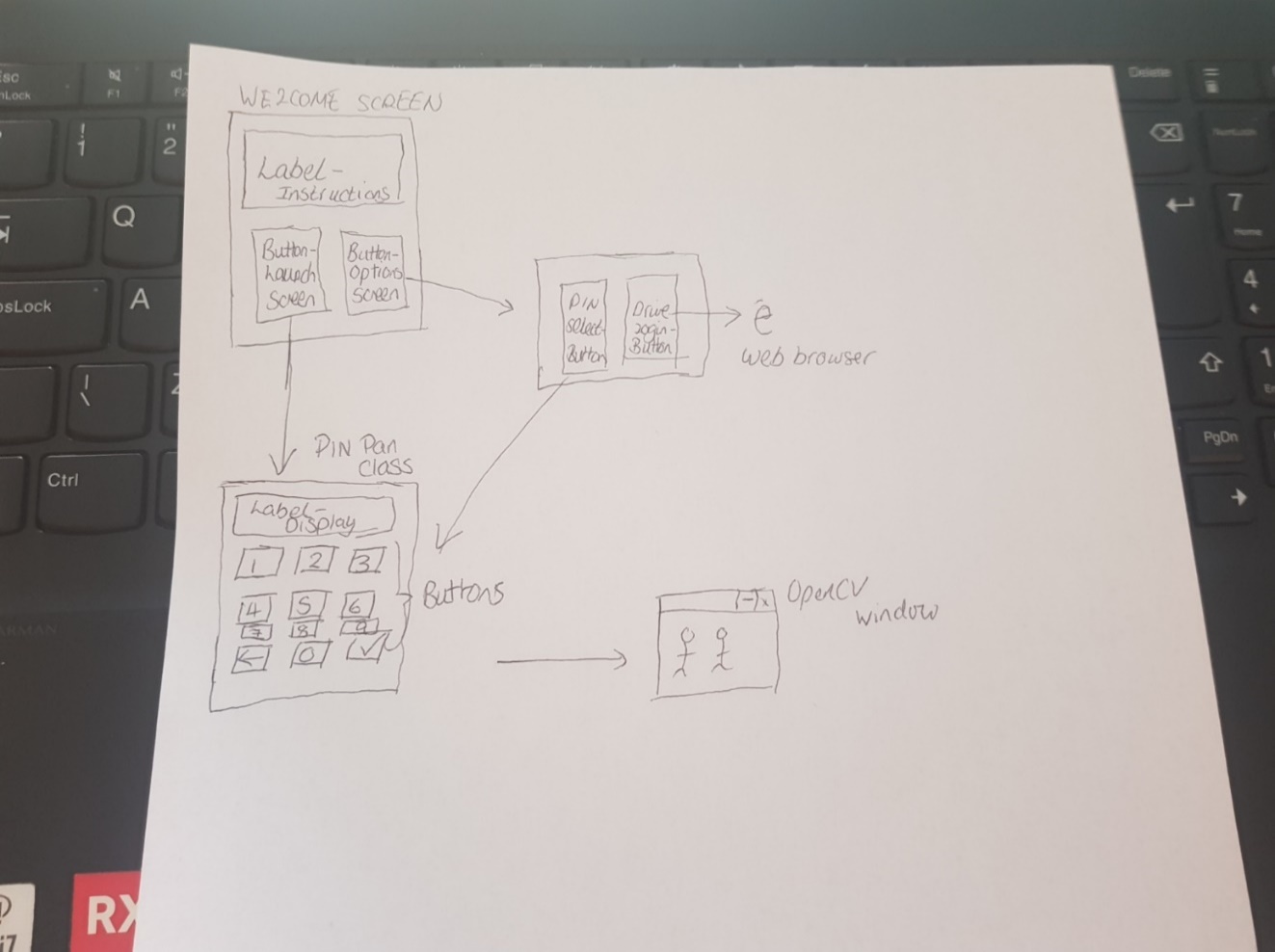
audio.play(“alert.mp3”)

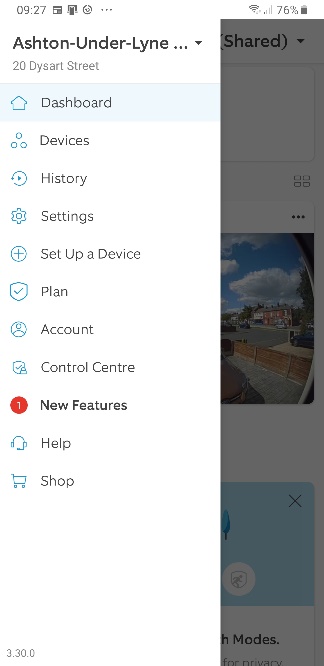
*This uses the kivy audio library to play and pause the audio according to the parameters passed into the procedure. It will select what to do using a selective statement.*

**Overall, these algorithms form a complete solution because they contain everything in the systems diagram. Some parts of the system diagram have been broken into smaller subroutines in order to split up the separate tasks even further**

Usability Features

My design is designed to work with a keyboard and mouse. Within the Kivy GUI, there is only buttons, I did this as one of Stakeholders said they liked the easy to use application on their Video Doorbell. Therefore, I limited text input although I couldn’t avoid this during the Drive Login section as you need to enter your account name and password. Kivy also automatically reshaped the button based on the screen resolution and how the user designed to resizes it which makes it usable on a range of devices.



As you can see from diagram above, I have limited the amount of different screen, this is to ensure that the program doesn’t take up too much storage space as this limits the amount of code and you also have enough space to save video footage. This has been done in order to avoid the stakeholders having to use external storage devices as this is one of the issues that one of the stakeholder had with their CCTV system.

I didn’t put the settings on the welcome screen as when researching the RING app which the stockholders thought was easy to navigate, they have different sections for the feed and settings (as shown to the right) so I thought as I would adopt this approach.

Also, I decided use the Google Drive browser login as all the Stakeholders prefer this cloud storage option so they may already have an account which means they may be already signed on their browser. In this case, they don’t have to re-enter their password and just click on their account. If they don’t have an account, they can easily create one there and it will also handle two step verification. In addition this, it is the layout is the same as if u were going to sign into any Google service (shown below).



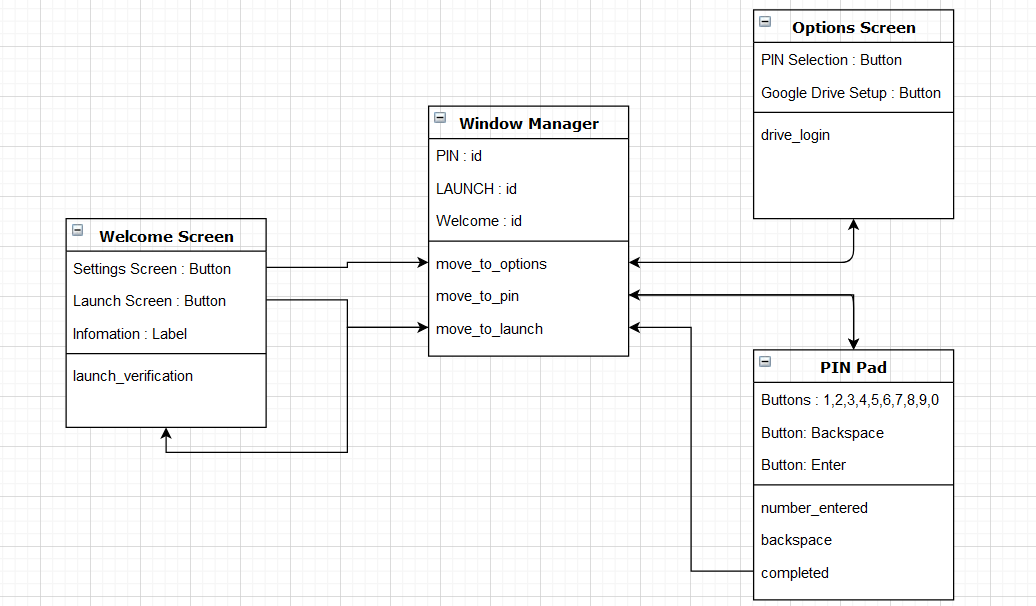
The users have 20 seconds to deactivate the alarm because I used the median time from the three times that the stakeholders suggested.

When using Kivy, you can use inheritance in order to get the properties of a layout in the screen. This is a usability feature because this means the window will automatically reshape for each widget so all the interface fits even if the user reshaped the window. This is also useful for portability to other devices.

Variables/Data Structures/Classes

*Kivy Classes:*

|  |  |
| --- | --- |
| **Class Name** | **Use** |
| Window Manager | To manage the switch between the different screen layouts in the .kv file and Python script. |
| PIN Pad | This class displays the keypad to activate, deactivate and choose a PIN. |
| Welcome Screen | This class display instructions on how to use the program and diverts you to options or activation depending on what button the user presses. |
| Option Screen | This class allows the user to configure their profile depending on their personal preference. |



This class diagram demonstrates how these classes interact with each other. The window manager has methods that help navigate between screens. Some methods are related to moving screen (completed, settings, launch).

*Variables:*

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Use** |
| display\_list | Array containing integers. | Saves number that are entered via the PIN Pad. It is saved as a list as you can use the pop function when backspace is used and push when a number is entered. Also, I can format the array to appear as an integer when it needs to appear on the screen. |
| File | String | This variable will be used to load the text file containing the hashed PIN. |
| Credentials | String | This variable will be used to save the Google Drive account details so it can be called later on in the program when the file needs to be saved. |
| Frame | Image which is saved using a Numpy array | This variables is used to save the current frame from the webcam and is needed in order to process the video and do the comparison. |
| previous\_frame | Image which is saved using a Numpy array | This variable is used to save the old frame before the next frame is fetch so the image comparison can take place. |
| frame\_prepared | Image which is saved using a Numpy array | This applies the Gaussian blur to reduce image noise and detail. This is done to before the comparison is done because it will speed up processing. |
| Differences | Image which is saved using a Numpy array | This will use the inbuilt subtract function within OpenCV so it will save an image which shows which area are the same and which aren’t. |
| image\_comparision | Integer | This variable shows the number of pixels which aren’t matching. This is done so we can calculate the percentage of the match which can then be used in a selective statement. |
| file\_name | String | This variable combines the date and time that the motion was detected in order to create a file name that is easy to identify for the user. |
| start\_time | String | This variable is used to get the time when the motion event is triggered so it can crop it for approximately 30 seconds before to create a 30 second clip of when motion was detected. |

*Additional Files:*

|  |  |  |  |
| --- | --- | --- | --- |
| **File Name** | **Use** | **File Size(KB)** | **Justification of File Size** |
| config.txt | Contains a hashed PIN code created by the user within the program. | 0.275 | The script will generate a SHA-256 hash value which always produces a fixed 32 bit value. |
| A series of .mp4 files (name will contain date and time) | This will contain cropped video footage which contain 30 seconds of video footage in colour when the motion event is detected. | 30000 | Assuming we are getting a SHD(720p) output at 60 frames a second:  - Bitrate: 1 Mbps  - Duration: 30 seconds |
| alert.mp3 | This contains the audio file that will play once the motion event is detected and the user doesn’t deactivate it within the correct time frame | 467 | Found exact size using file manager. |

Iterative Development

Welcome Screen – 1st Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Selection of option via button | Launch Screen | Crash, as this is yet to be created in the kv File. |  |
| 2 | Selection of option via button | Settings Screen | Crash, as this is yet to create in the kv file. |  |

Welcome Screen – 2nd Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Selection of option via button | Launch Screen | Displays the Launch Screen. |  |
| 2 | Selection of option via button | Settings Screen | Displays the Settings Screen. |  |

|  |  |
| --- | --- |
| Test Number (ITERATION.TEST NUMBER) | Justification |
| 1.1 , 1.2 | To see whether the screen appears in the correct format. |
| 2.1 | To see whether the user can access the Launch Screen from the initial screen. |
| 2.2 | To see whether the user can access the Settings Screen from the initial screen |

Settings Screen – 1st Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Selection of option via button | Google Drive Setup | Crash, as Python will not be able to find the function. |  |
| 2 | Selection of option via button | PIN Selection | Crash, as this is yet to create in the kv file. |  |
| 3 | Find “config.txt” in file manager. | 1111 – Enter using keypad in Test Data stage. | Cannot find as user has not entered the PIN yet. |  |

Settings Screen – 2nd Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Selection of option via button | Google Drive Setup | Displays the Google Drive sign in page on browser. |  |
| 2 | Selection of option via button | PIN Selection | Displays the PIN screen so the user can enter it. |  |
| 3 | Find “config.txt” in file manager. | 1111 – Enter using keypad in Test Data stage. | Cannot find as the function has not been defined yet. |  |

Settings Screen – 3rd Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Selection of option via button | Google Drive Setup | Displays the Google Drive sign in page on browser. |  |
| 2 | Selection of option via button | PIN Selection | Displays the PIN screen so the user can enter it. |  |
| 3 | Find “config.txt” in file manager. | Should contain hashed file entered. | Should be contains the same folder as Python Script. |  |

|  |  |
| --- | --- |
| Test Number (ITERATION.TEST NUMBER) | Justification |
| 1.1 , 1,2 | To see whether the screen appears in the correct format. |
| 2.1 , 3,1 | To check the function correctly runs and launches the Web Browser so the user can sign in. |
| 2.2 , 3.2 | To check whether you can access PIN Selection Screen from the Settings Screen. |
| 2.3 | Check that the correct function has been binded to enter button. |
| 3.3 | To check that the Hashed PIN is in the correct location so it can be used in other parts of the program. |

Launch Screen – 1st Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | User enters PIN using keypad on screen – they have set this up in Settings. | 1111 – Valid Data | Open the OpenCV window. |  |
| 2 | User enters PIN using keypad on screen – they haven`t set this up in Settings. | 1111 | Error , cannot find “config.txt” |  |
| 3 | User enters incorrect PIN. | 1222 – Invalid Data | Open the OpenCV window. |  |

Launch Screen – 2nd Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | User enters PIN using keypad on screen – they have set this up in Settings. | 1111 – Valid Data | Opens the OpenCV window. |  |
| 2 | User enters PIN using keypad on screen – they haven`t set this up in Settings. | 1111 | Test cannot run – automatically redirect to create PIN. |  |
| 3 | User enters incorrect PIN. | 1222 – Invalid Data | Error message – “Incorrect PIN” |  |

|  |  |
| --- | --- |
| Test Number (ITERATION.TEST NUMBER) | Justification |
| 1.1,1,3 | Check whether entering a PIN opens the OpenCV window. |
| 1.2 | To see whether the it attempts to open the config.txt file |
| 2.1 | Check whether valid data allows it to proceed to the OpenCV window. |
| 2.2 | Check whether it allows the user to set up a PIN if one isn`t already set up. |
| 2.3 | Check whether the comparison works so the user can only proceed if they have the correct PIN. |

OpenCV – 1st Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Video Feed | No change in what is appearing in video | Continues to show feed in window. |  |
| 2 | Video Feed | Change what is happening in video. | Continues to show feed in window. |  |
| 3 | File Location | Check file manager for a file with the video feed. | Will not find – function is not created yet. |  |
| 4 | Kivy Window | Change in video. | Will not change – function is not created yet. |  |
| 5 | File Location | Check Google Drive for file with the video feed. | Will not find – function is not created yet. |  |

OpenCV – 2nd Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Video Feed | No change in what is appearing in video | Continues to show feed in window. |  |
| 2 | Video Feed | Change what is happening in video. | Stops showing feed in window. |  |
| 3 | File Location | Check file manager for a file with the video feed. | Saves full footage with the name “test\_footage.mp4” |  |
| 4 | Kivy Window | Change in video. | PIN deactivation window appears. |  |
| 5 | File Location | Check Google Drive for file with the video feed. | Will not find – function is not created yet. |  |

OpenCV – 3rd Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Video Feed | No change in what is appearing in video | Continues to show feed in window. |  |
| 2 | Video Feed | Change what is happening in video. | Stops showing feed in window. |  |
| 3 | File Location | Check file manager for a file with the video feed. | Saves full footage with a logical name containing date and time. |  |
| 4 | Kivy Window | Change in video. | PIN deactivation window appears. |  |
| 5 | File Location | Check Google Drive for file with the video feed. | Will be able to find the file with the same name as it is saved locally. |  |

|  |  |
| --- | --- |
| Test Number (ITERATION.TEST NUMBER) | Justification |
| 1.1,1,2,2.1,3.1 | Check that the live feed is showing up. |
| 1.3,1.4,1.5,2.5 | Check that the correct function has been binded. |
| 2.2,3.2 | Check that change in motion results in the OpenCV loop breaking. |
| 2.3 | Check that the video feed has been saved fully so it can be trimmed and saved. |
| 2.4,3.4 | Ensure that the correct screen appears after motion is detected. |
| 3.3 | Ensure the file is saved with a sensible file name that is easy to find. |
| 3.5 | Check that the user can access the file on their Drive as well. |

PIN Deactivation – 1st Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | User input via Button – Correct PIN | 1111 – Valid Data | Returns to Welcome Screen. |  |
| 2 | User input via Button – Incorrect PIN | 2222 – Invalid Data | Stays on screen. |  |
| 3 | Audio Playback | No Input | Nothing – function not created yet. |  |
| 4 | Audio Deactivation | 1111 – Valid Data | Nothing – function not created yet. |  |

PIN Deactivation – 2nd Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | User input via Button – Correct PIN | 1111 | Returns to Welcome Screen. |  |
| 2 | User input via Button – Incorrect PIN | 2222 | Stays on screen. |  |
| 3 | Audio Playback | No Input | Plays audio if input isn’t entered within 20 seconds. |  |
| 4 | Audio Deactivation | 1111 – Valid Data | Nothing – function not created yet. |  |

PIN Deactivation – 3rd Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | User input via Button – Correct PIN | 1111 | Returns to Welcome Screen. |  |
| 2 | User input via Button – Incorrect PIN | 2222 | Stays on screen. |  |
| 3 | Audio Playback | No Input | Plays audio if input isn’t entered within 20 seconds. |  |
| 4 | Audio Deactivation | 1111– Valid Data | Audio is stopped as this is the correct PIN. |  |

|  |  |
| --- | --- |
| Test Number (ITERATION.TEST NUMBER) | Justification |
| 1.1,2.1,3.1 | Ensure the entered pin is compared to the file that contains the PIN. Valid data is entered to see if it progresses to the next screen. |
| 1.2,2.1,3.1 | Ensures that an invalid PIN doesn`t allow the user to progress. |
| 2.3,3.3 | Ensures that the programs waits 20 seconds before the alarm goes off |
| 3.4 | Ensure that the audio stops once that the correct PIN is entered so it doesn’t continue even with valid data. |

Online Feed – 1st Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Kivy Label | Run program. | Display IP address that is going to be used to view web cam on browser. |  |
| 2 | Online Webcam feed on web browser. | Enter IP address from Welcome Screen into Browser. | 505 error – function not defined yet. |  |

Online Feed – 2nd Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Kivy Label | Run program. | Display IP address that is going to be used to view web cam on browser. |  |
| 2 | Online Webcam feed on web browser. | Enter IP address from Welcome Screen into Browser. | See live webcam feed on web browser. |  |

|  |  |
| --- | --- |
| Test Number (ITERATION.TEST NUMBER) | Justification |
| 1.1 | Ensures that the user can find the IP address in order to access it so it can be accessed via the web. |
| 2.2 | Ensure that the feed can be seen on another device. |

Post-development phase

The stakeholder should be able to enter pin to active the motion detector.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | PIN Input – Correct PIN | 1111 | Open CV Window. |  |
| 2 | PIN Input – Incorrect PIN | 1222 | Displays message saying that PIN has been entered incorrectly. |  |

|  |  |
| --- | --- |
| Test Number | Justification |
| 1 | Ensures that the user can continue with a valid PIN. |
| 2 | Ensure that the user can`t continue with an invalid PIN. |

The stakeholder use be able to choose security details when setting up the system.

The stakeholder should be able to sign in to Google Drive when setting up the system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | User Input via Button | Settings Screen | Opens Settings Screen. |  |
| 2 | User Input via Button | PIN Selection | Opens PIN Selection Screen. |  |
| 3 | User Input via Button | Google Drive Setup | Opens Web Browser with Google sign in screen. |  |
| 4 | User selects PIN. | 1111 | Saves as an hashed password under “config.txt” |  |
| 5 | User selects PIN with an Insufficient amount of characters. | 123 | Display error messages saying there isn`t enough digits. |  |
| 6 | User selects PIN with more than 5 digits. | 12345 | Doesn`t allow you to enter more than 4 digits. |  |
| 7 | User enters Google Account password and username. | [mistryviv03@gmail.com](mailto:mistryviv03@gmail.com) | Successful sign in and return to Python Program. |  |

|  |  |
| --- | --- |
| Test Number | Justification |
| 1 | Ensures the user can open the settings screen from the welcome screen. |
| 2 | Ensures that the user can open PIN Selection from the settings screen. |
| 3 | Ensures that the user can sign into Google Drive via the Web Browser when they click the Google Drive Button. |
| 4 | Ensures that the user`s PIN is correctly stored in a text file so it can be used later on in the program. |
| 5 | This erroneous data ensure that the user selects a 4 digit pin. |
| 6 | The attempts to enter this erroneous data ensure that the user can’t enter more than 4 digits. |
| 7 | Ensures that you can sign into Google. |

If motion is detected, Stakeholder should hear a noise after 20 seconds if the pin isn’t entered.

If motion is detected, Stakeholder should not hear a noise after 20 seconds if the pin is entered correctly?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Video Frame | Walk towards webcam | PIN Pad appears. |  |
| 2 | PIN Deactivation | 1111 – Valid Data | Returns to Welcome Screen. |  |
| 3 | PIN Deactivation | 1222 – Invalid Data | Shows that PIN is incorrect. |  |
| 4 | PIN Deactivation | Enter no data. | Alarm starts going off. |  |
| 5 | Alarm Deactivation | 1111 – Valid Data | Returns to Welcome Screen. |  |
| 6 | Alarm Deactivation | 1222 – Invalid Data | Shows that PIN is incorrect. |  |

|  |  |
| --- | --- |
| Test Number | Justification |
| 1 | Ensure that once motion is detected, it allows you to deactivate the alarm. |
| 2 | Ensure that valid data deactivates the alarm. |
| 3 | Ensure that invalid data doesn`t deactivate the alarm. |
| 4 | Ensure that after 20 seconds, the alarm goes off. |
| 5 | Ensure audio stops once you entered valid. |
| 6 | Ensure that audio continues if invalid data is entered. |

The stakeholder should be able to view the file of when motion was detected using their file browser.

The stakeholder should be able to download the file of when motion was detected using their Google Drive account.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Search for file locally. | Video Feed | File is in the same folder as the Python Script. |  |
| 2 | File Name | Video Feed | The file name contains the date and the time. |  |
| 3 | Search for file on the cloud | Video Feed | The file with the same name as the Locally saved video file is found. |  |
| 4 | Full video file. | Download and watch Video Feed | The video is the same as the local file. |  |

|  |  |
| --- | --- |
| Test Number | Justification |
| 1 | Ensure video files are stored in the correct location. |
| 2 | Ensures that the video has an easily identifiable file name. |
| 3 | Ensures there is an exact copy of the local file on the cloud. |
| 4 | Ensures that file hasn’t been corrupted during upload. |

The stakeholder should be able to view the live feed on a browser on another device when the alarm is armed which can be accessed by the address on the settings screen.

When the alarm is armed, the user should be able to see a live feed on a window on their screen?

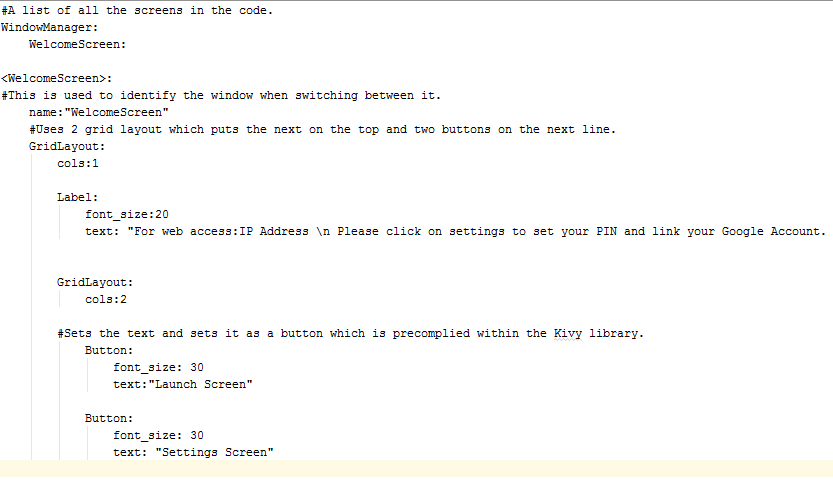
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Online Feed | Live Webcam Feed | Video feed appears on Browser. |  |
| 2 | Live Feed | Live Webcam Feed | Video appears in a Window. |  |

|  |  |
| --- | --- |
| Test Number | Justification |
| 1 | Ensure video can be seen on web browser. |
| 2 | Ensure that video can be viewed on the device that is running the program. |

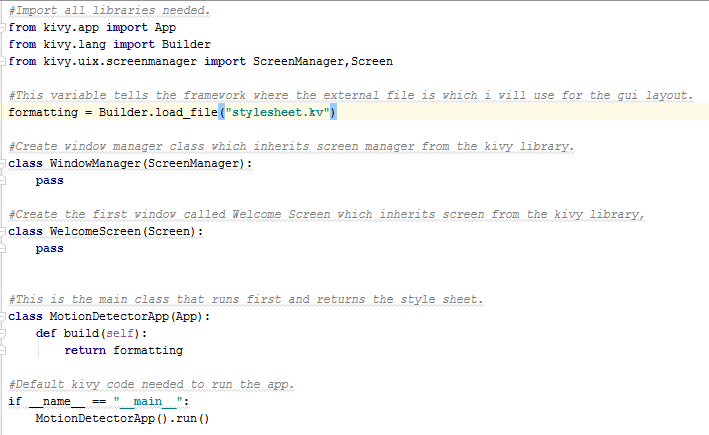
**Development**

In this section I will develop my code in an iterative manner while using my success criteria and test tables in order to develop a solution to my code which can will be evaluated by my stakeholders.

Welcome Screen – Iteration One



This part of the Kivy file describes how the Graphical Interface will work. The welcome screen uses two Grid Layouts because I want the text to be appears at the top and the two buttons to appear on the same line.



This prototype of code should work because all the needed libraries have been imported. The kv file is loaded in using the builder function. I didn’t need this as you don’t need to include it if you call it the same name as the main class. However, I thought this may have ran into more errors because it is not clearly stated and “stylesheet.kv” is a more sensible name and would make sense if I returned into the code in future in order to maintain the program.

The classes which inheritate screen only have pass inside of it. This is because the layout will be in the kv file as it is a more logical approach. This is because if the graphics are wrong I can just go straight to the kv file so it is easier to debug.

The class MotionDetectorApp is the main class because this uses the inbuilt build function which will automatically run at the start of the Kivy program because this is where the first part that needs to be ran is always kept.

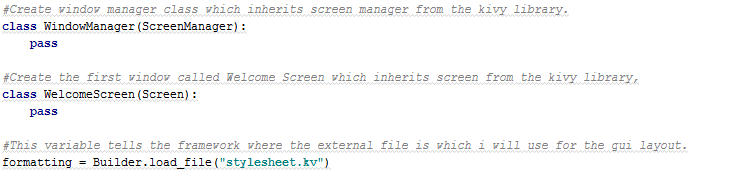
The if statement is default code because this tell the program to run the main class on launch.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Selection of option via button | Launch Screen | Crash, as this is yet to be created in the kv File. |  |
| 2 | Selection of option via button | Settings Screen | Crash, as this is yet to create in the kv file. |  |

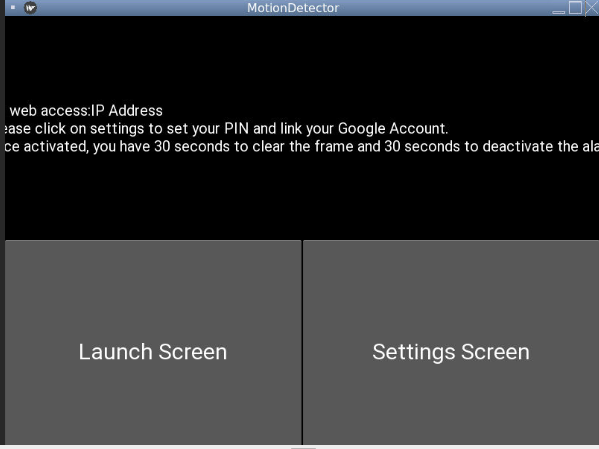
Once I ran this code both tests failed because it ran the following error:



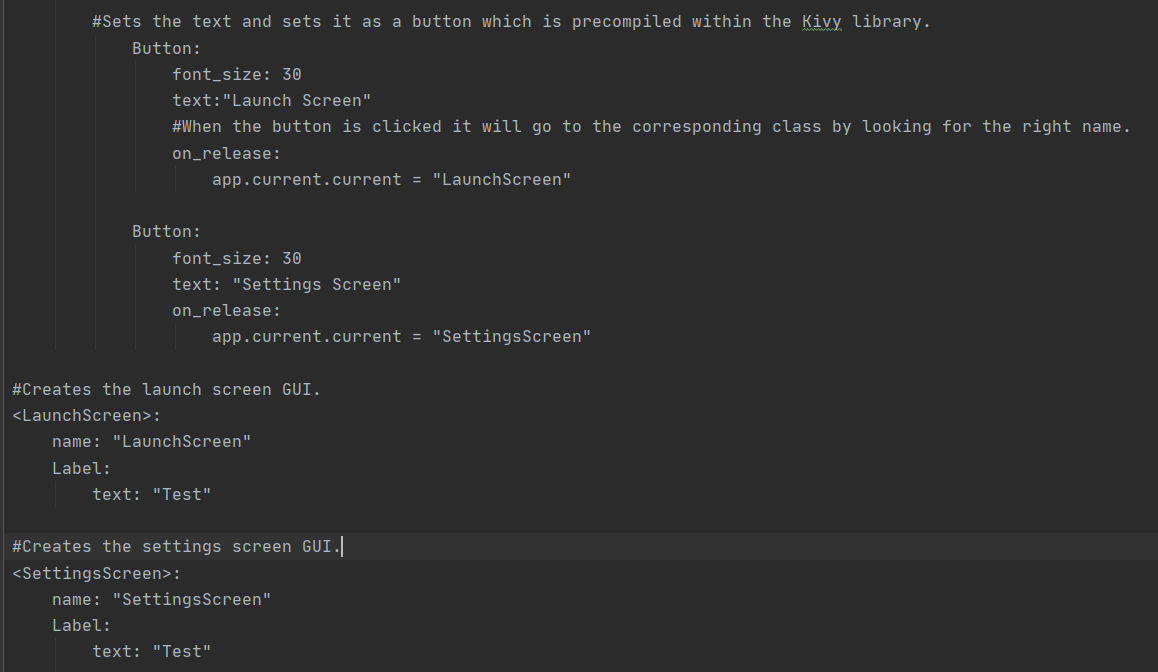
The issue with my code was the position of the formatting variable being assigned. Window Manager was being loaded in although it is not defined yet so the class is defined. Therefore, I moved the variable assigning to after the Window Manager class is defined so it doesn`t run into this error.



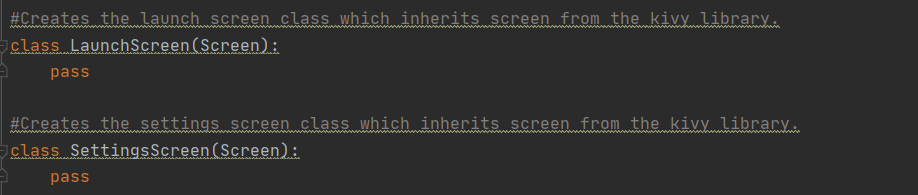
The result of test 1 and 2 was that the correct format appeared although clicking the button doesn`t do anything. This is different to my expected result because I haven`t assigned any functions yet. However, this iteration is now complete because the aim was to get the GUI to appear correctly. A screenshot of this screen is below:



Welcome Screen – Iteration Two

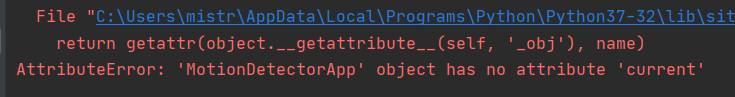


This prototype of the .kv file shows the button changing screen upon release of the button. It will transfer to the screen that has the corresponding name. This is why there is a name assigned at the start of every class.

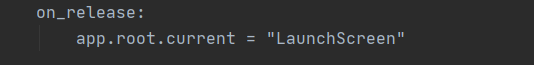


This Python code sets up the two new screens which is used by screen manager. The screen properties are inherited from the Screen class so properties such a grid layout, label and button can be used as this is inbuilt. The only thing in there is pass because the Graphical User Interface is coded within the .kv file.

This prototype code failed the test and returned the following results:



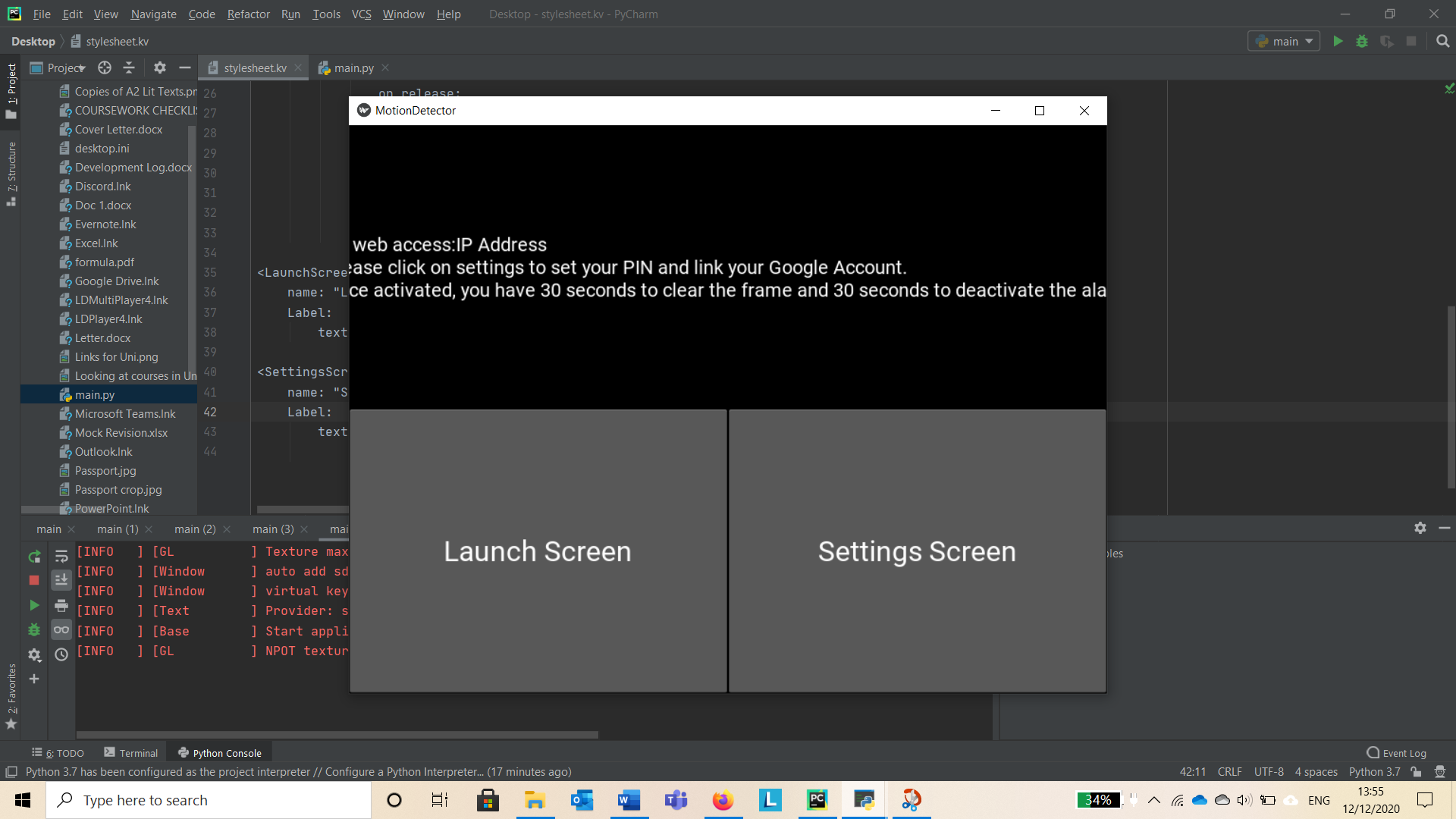
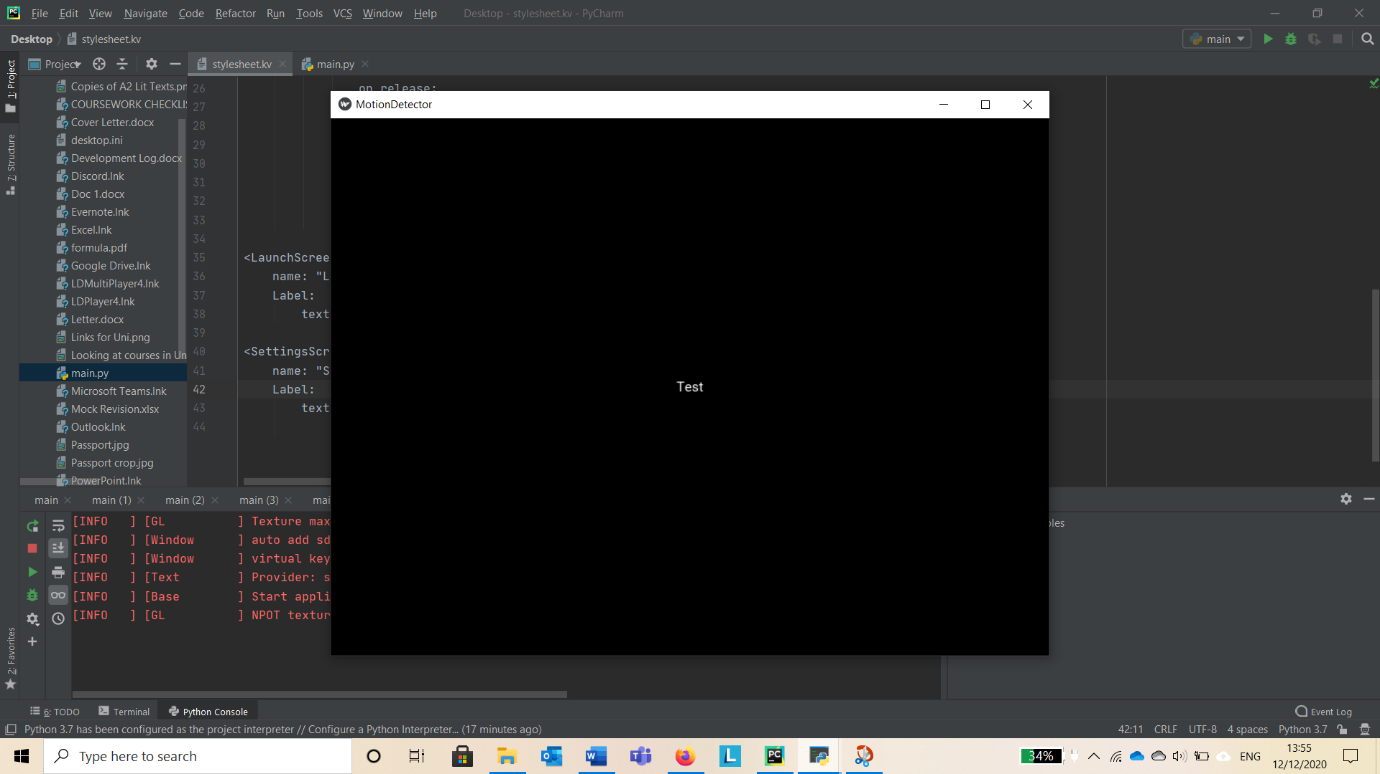
I fixed this by correcting a typo I made within the .kv. The program attempted to find current from going to the main class although it must go to the root of the program before it is able to do it. The fixed is shown below:

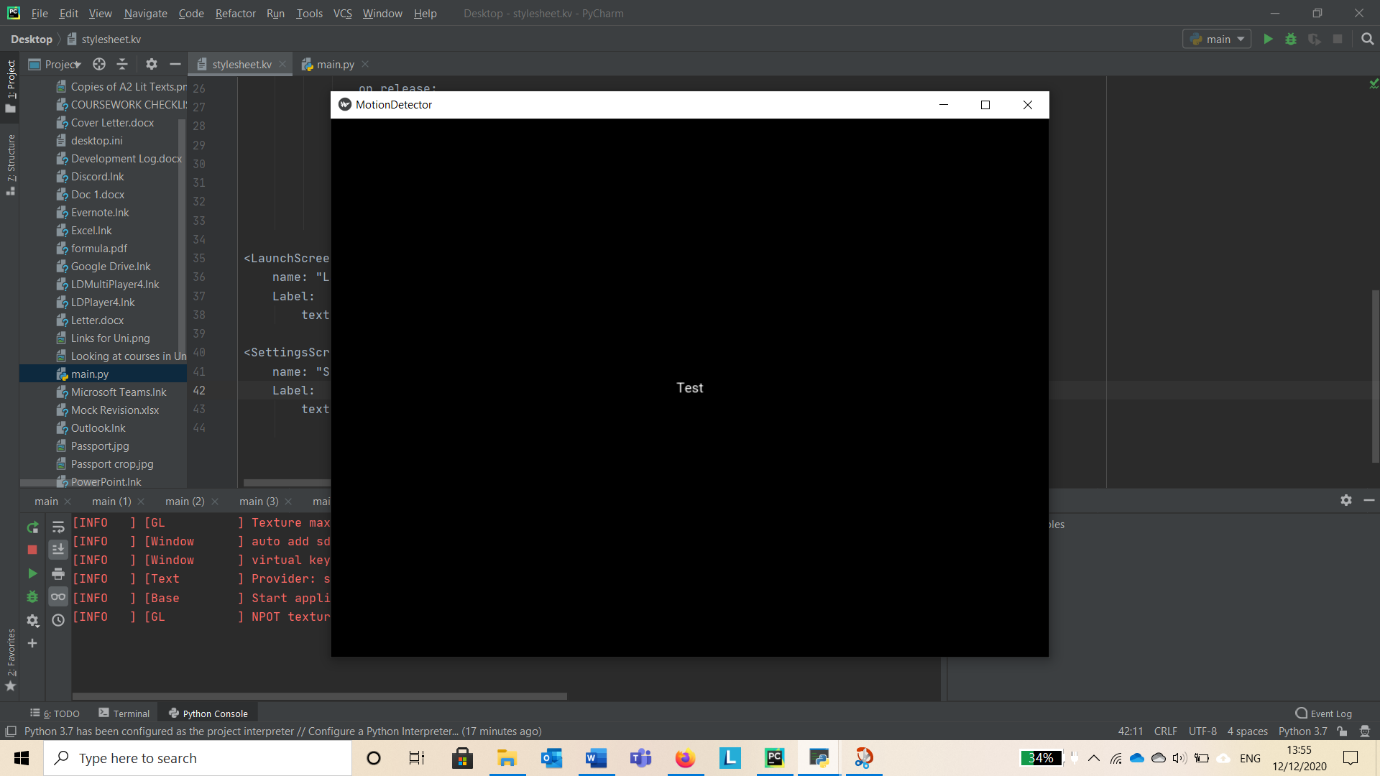


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Selection of option via button | Launch Screen | Displays the Launch Screen. | Shows the correct screen. |
| 2 | Selection of option via button | Settings Screen | Displays the Settings Screen. | Shows the correct screen. |

I decided to check whether it moves on to the next screen by simply making a label that says “test”. I did it this way to ensure it wasn`t just attempting to move on and not finding the screen. If this was the case, it may displays black empty screen.

The evidence of the test is shown via staged screenshots below:





Launch Screen – Review

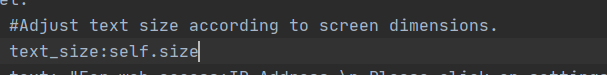
So far, I have not completed any of the success criteria. However, this is require to provide good usability features as it has instructions and it split up settings and the main program using a limited user input to avoid errors.

At this stage, I met with Mr Mistry(A Pub Owner) to see if this first screen is clear enough for multiple staff to use. He said that it was clear how to use but the text isn`t clear as it doesn`t fit the screen so I took time to change properties of text label.

As I wanted it to automatically adjust based upon the device that is being used, I used the internet to find out how to do this as I was unsure and ended up using this website:

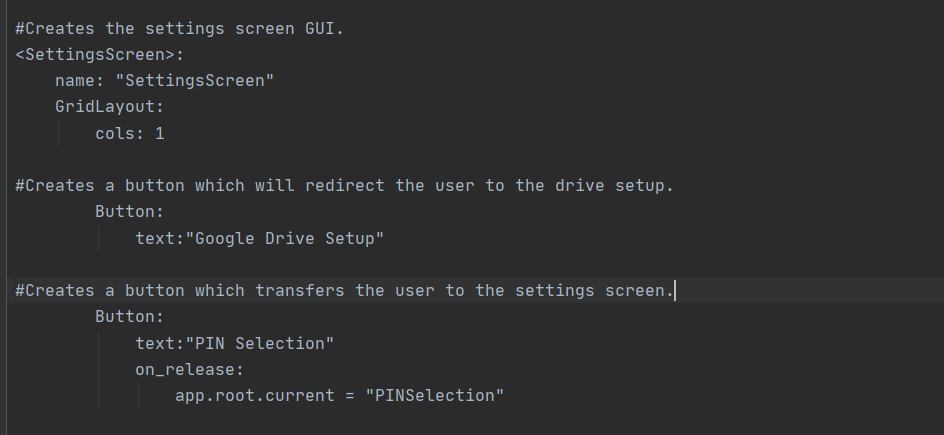
<https://blog.kivy.org/2014/07/wrapping-text-in-kivys-label/>

The fix uses a variable called size which is initialised at the start-up of the program and saves the dimensions of the program. The variable assignment is not in my python program because it is part of the Screen class which is inherited. The fixed is shown below:

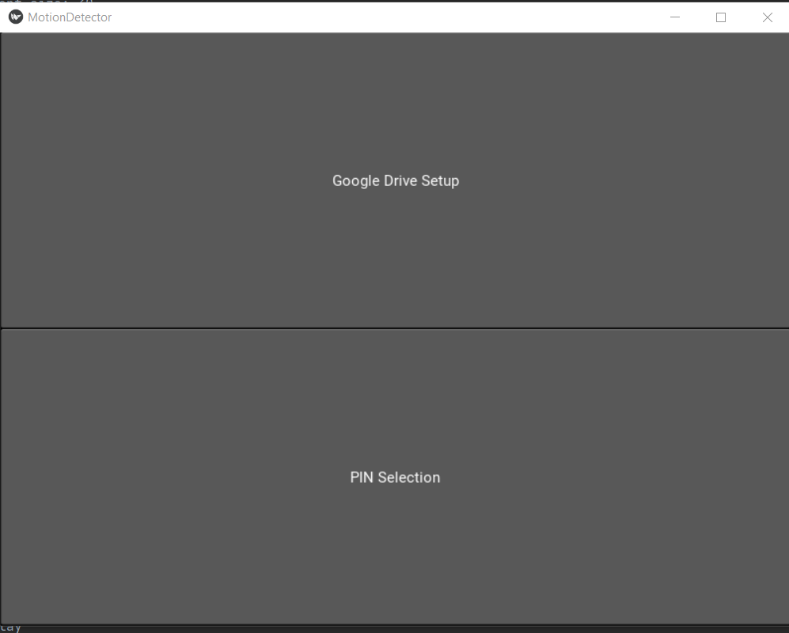


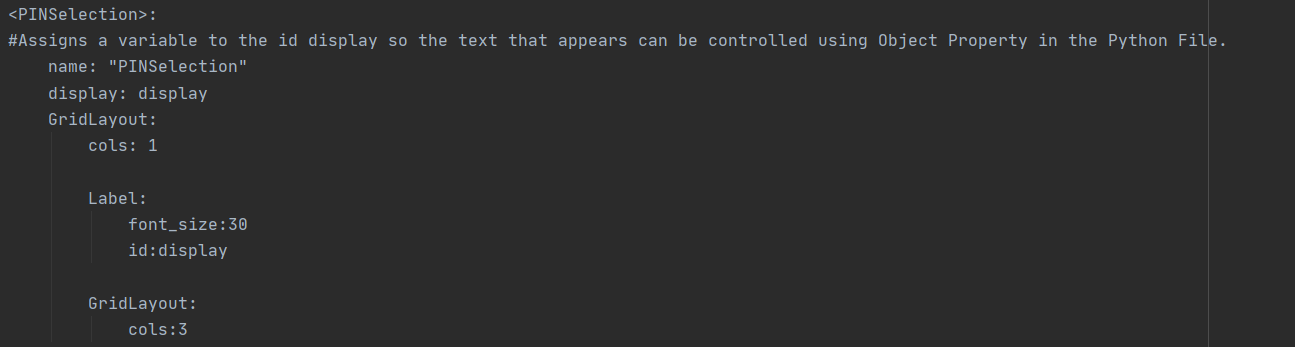
Settings Screen – 1st Iteration

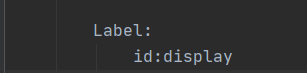
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Selection of option via button | Google Drive Setup | Crash, as Python will not be able to find the function. |  |
| 2 | Selection of option via button | PIN Selection | Crash, as this is yet to create in the kv file. |  |
| 3 | Find “config.txt” in file manager. | 1111 – Enter using keypad in Test Data stage. | Cannot find as user has not entered the PIN yet. |  |



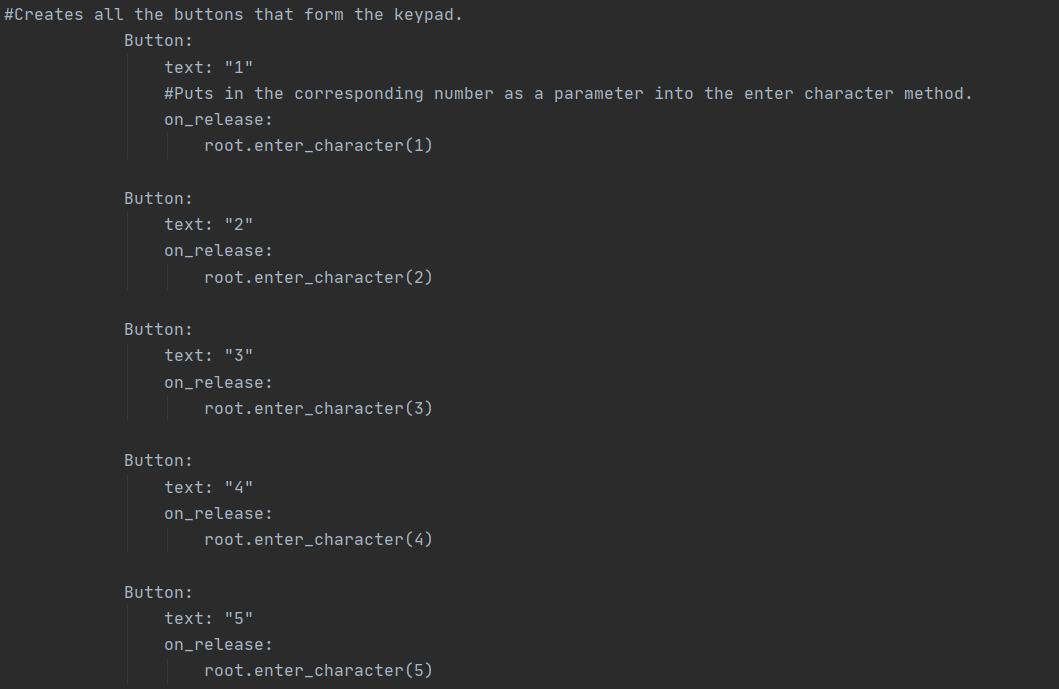
This part creates a new class which displays a new button. Each button takes up its own line. Once “PIN Selection” is pressed it will transition to the corresponding window. I ran test 1 and 2 at this point and it passed as the GUI was layout correctly. Evidence of the passed tests are below:



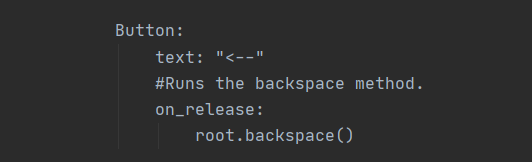


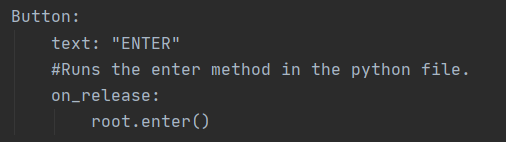


This extract gives the id called “display” and identifier called “display”. This allows the Python file to manipulate properties of this label within the script. The display has its own Grid Layout with one column so it fits the top of the screen completely and has nothing next to it.

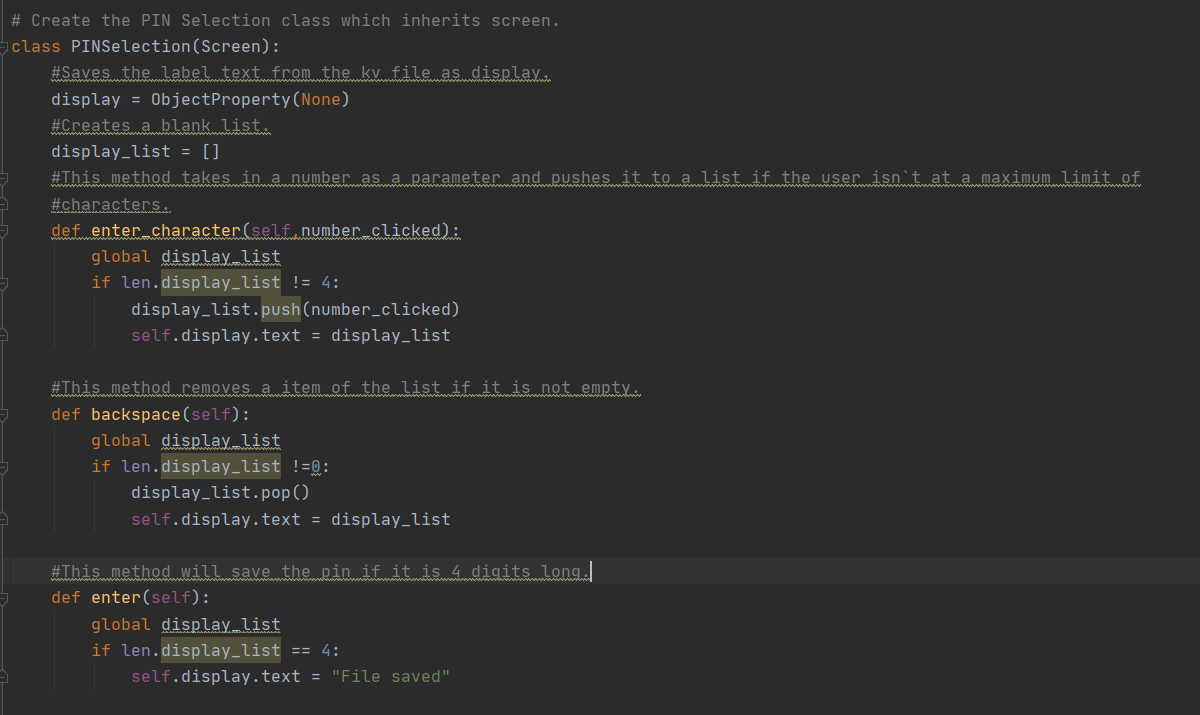


This extract of the .kv file for the PIN Pad shows a buttons being created in a 3x3 grid to form a pad. I have decided do this for validation purposes. This enables the program to limit user input and ensure they only enter valid integers. This usability features reduced input error as no erroneous data can be entered.





These are the only other methods of the screen on to enter the password and one for backspace. Again, I could have simplified the code by using a text input label and it wouldn`t need this code. However, it means that a keyboard may not be needed to run this input. As well as making it less likely to run into error for the reasons I have mentioned in previous paragraphs.



This new PIN Selection class inherits the screen class from the imported Kivy library. It saves the label in the formatting returned by the Object Property function from the imported library and saves it under the variable “display”. The function Object Property returns the value of display in the correctly format so the properties of the label can be changed within the Python Program. “None” is passed in as a parameter because we wants it to save with the default properties.

Then it creates a list which will be created in order to save the entered numbers as an array. This list must be global as it is needed to accessed within each method.

The method “enter\_character” require a parameter because this takes in the number that has been entered. I could have done a different method for each function although this would have inefficient because I would be repeating code for no reason. I have validated this by only doing this if the length is not equal to 4 as this is required length of the PIN.

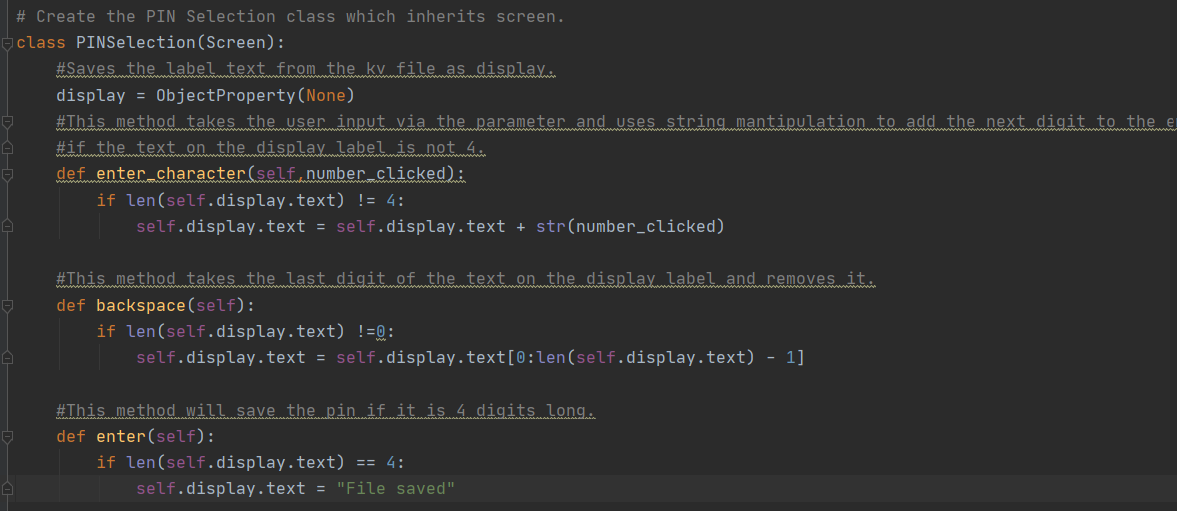
The “backspace” method takes the last integer entered out. It only does this if the list isn`t empty, this is because you cannot delete if there is nothing there. I have used to the Pop function because the logic behind this is like a stack. The last thing entered needs to be the first thing out. Finally, it outputs this back to the GUI so the user can see the result of pressing this button.

The “enter” method ensures that the length of the list of 4 and then says that the file has been saved to the user. I have not implemented saving to file yet because I want to ensure that the conditional statements work.

At this point, I ran test 3 and it failed as I got the following error:



An attribute error means that there is a problem with the variable. This must mean that “display\_list” is assigned in the wrong place. However, I came up with a more efficient solution which is different to my designed algorithms in the design section. I thought of a way that involves using string manipulation and doesn`t use a list. This increases the space efficiency of the program as less variables need to be stored which results in less memory space being used. My improved method is shown below:



In the design section, I didn`t intent to complete this much of the PIN screen in this much detail in this point in the process. However, it is more beneficial because this can be used as a super class for similar screen later to in the process which can avoid repetition.

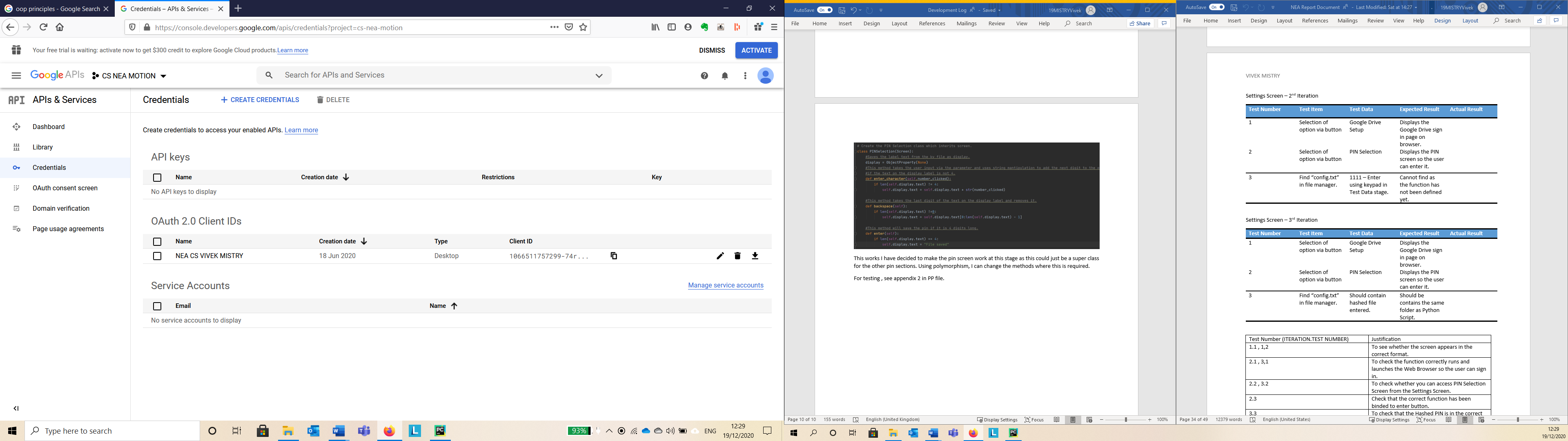
This prototype updates the screen when a character is entered by converting the integer to the screen and add it to the end of the existing code.

The backspace method works by display the start to the one less than the end. I have used the inbuilt function that calculates the length of a string and takes one away from it. The minus one is required so the last digit of the code doesn`t display.

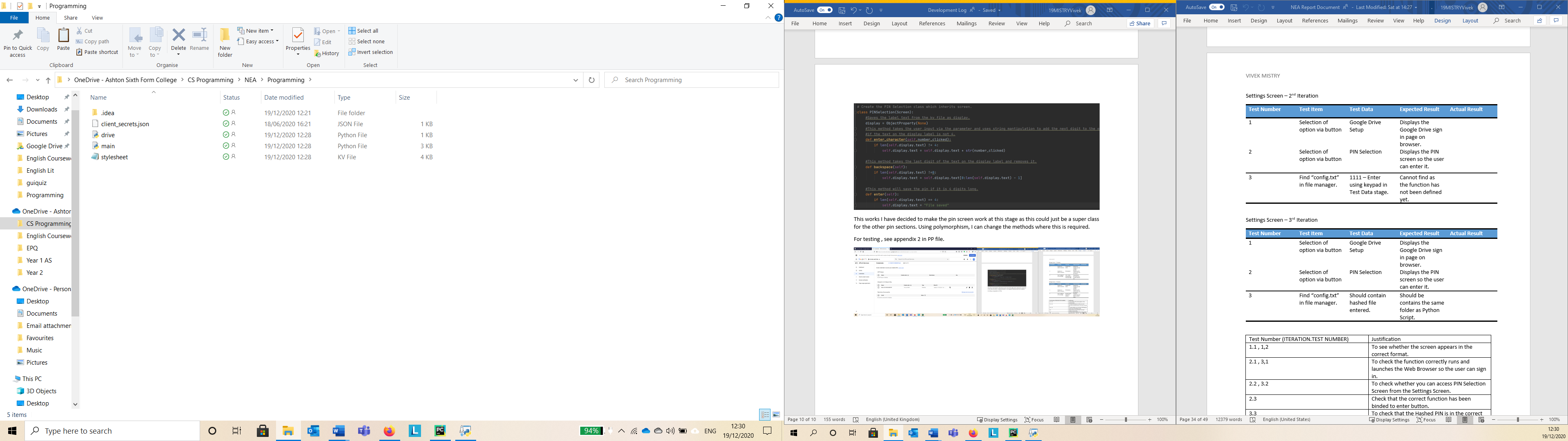
For testing see appendix two. This screencast shows the user entering the pin “1111” and the validation being passed so it display the corresponding message on screen. This shows test 3 passing.

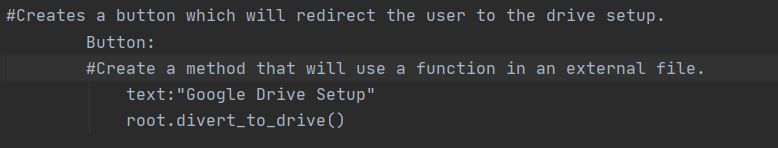
Settings Screen – 2nd Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Selection of option via button | Google Drive Setup | Displays the Google Drive sign in page on browser. |  |
| 2 | Selection of option via button | PIN Selection | Displays the PIN screen so the user can enter it. |  |
| 3 | Find “config.txt” in file manager. | 1111 – Enter using keypad in Test Data stage. | Cannot find as the function has not been defined yet. |  |

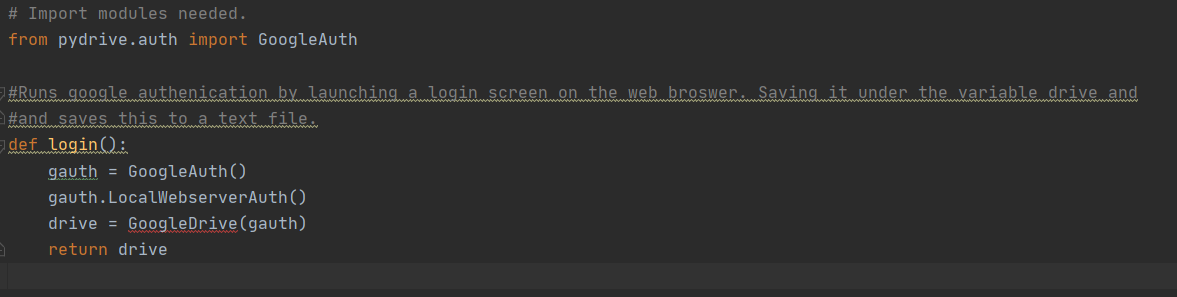


The first step was to sign into my google account on the developers page. Create a new app so I can download the json file that launches the browser so the user can sign in. This file has to be saved in the same location as your python script so it can locate it. The file “client\_serects.json” is a javascript file of the developers website that is linked to my app.





The following .kv file runs the function which is at the root at the class when the button is clicked.



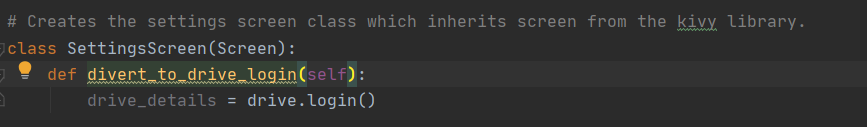
This is another file called “drive.py”. I did this to make the program modular and to ensure that the GUI and cloud services are separate in order to make it easier to debug as I can just go to the corresponding program. When disturbing software, if the user doesn`t want cloud services, they don`t need to download this script will save them disk space.

The procedure “login” runs the function GoogleAuth and saves it under the variable gauth. This then is used to run authentication in the browser. This is done this way as it is direct from the provider which means that it will be regularly debug and it reduces the amount of errors that will be ran into to. It then saves the user credentials under the variable drive which I then return so it can be used back in the main program.

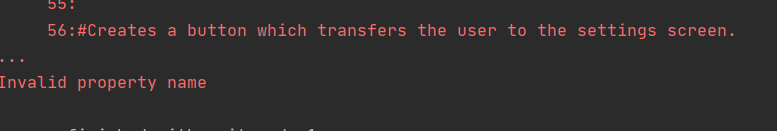


This file acts as a user created library.

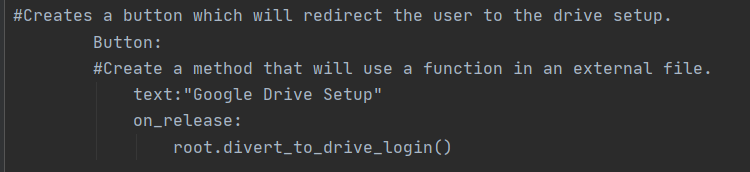
I have used Pydrive which is a third party module instead of the Google API. This is a third party library which has designed to simplify cloud usage. The commands are much simpler due to the focus on drive and no other Google Services. I have used libraries in this program because it is useful as it stops me from writing code that has already been created.



When the button is pressed, saved the value returned by the drive library that I have created and saves it under drive details so I can use it later.



I ran test one and it fails as I forgot to put “on release”. This tells Python that after the button is clicked run “divert\_to\_drive\_login”. This simple missing line in the .kv file created this issue. The fix is shown below.



Test 2 and 3 have not been repeated because they are the same as the previous iteration. Test 1 has now passed. Evidence of this is shown in appendix 3. The video shows the user clicking the Google Drive Button so it redirects the user to the web browser and they successfully sign in using their account.

Settings Screen – 3rd Iteration

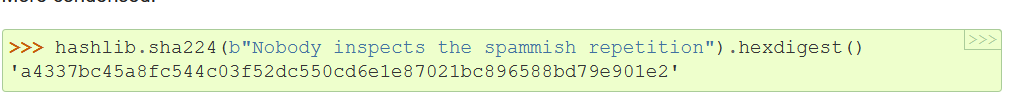
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Selection of option via button | Google Drive Setup | Displays the Google Drive sign in page on browser. |  |
| 2 | Selection of option via button | PIN Selection | Displays the PIN screen so the user can enter it. |  |
| 3 | Find “config.txt” in file manager. | Should contain hashed file entered. | Should be contains the same folder as Python Script. |  |

Test 1 and 2 are the same as the previous iteration so they will not repeated. This section will focus on the development of the hashing and saving. In order to complete this, I had to import an inbuilt Python Library as shown below:  


I haven’t got much experience with this so I decided to use the Python documentation for this module. The link I used is:

<https://docs.python.org/3/library/hashlib.html>

I used an example as a template. This example uses the sha 244 algorithm and Hexadecimal. This was the best method because less bits (this uses 244 bits) results in a smaller file size. Converting the file to hexadecimal also reduces the file size. Some users may want a hashing algorithm that uses more bits to avoid collisions. However, there is 10,000 possibilities for a 4 digit pin. The hashing algorithm can have 2^244 unique hashes so this will be fine. This is the example I used from the website:





This protype code hashes the pin and saves it under a variable called “hashed\_pin”. It applies the sha-244 hashing algorithm to the text displayed in the GUI once it is converted to a string. Then, this is converted to Hexadecimal which makes it shorter. This had to before the window is updated otherwise the string “File saved” would be hashed everytime. This would be result in a faulty program as the user would never be able verify the PIN on the Welcome Screen so the main OpenCV script will never be able to run. I have not saved the hashed file yet because I want to check whether this hashing algorithm is working first.

When running this, I ran into this error:



A type error means that there is an issue with how the variable is stored. I was stuck so I decided to google it and found this forum:

<https://stackoverflow.com/questions/7585307/how-to-correct-typeerror-unicode-objects-must-be-encoded-before-hashing>

This is the fix:



I had to encode this into Unicode to fix the error because the hashing algorithm uses Unicode in order to do mathematical operations to get a hash value. UTF-8 is an 8-bit format to encode characters using Unicode. By adding this before the value is hashed it should work. I didn`t format it as a string because it can just translate into the Unicode values for integers and it enables less processing to occur.

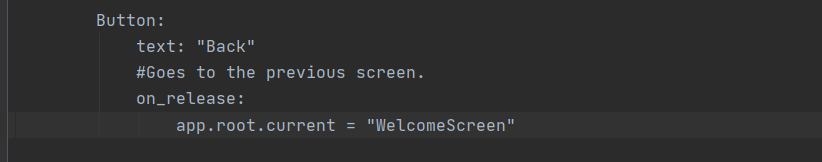
For evidence of testing, please see appendix 4. In this video, the user chooses a code and shows the hashed file stored in a text file in same location as the Python scripts. Therefore, it has passed the test.

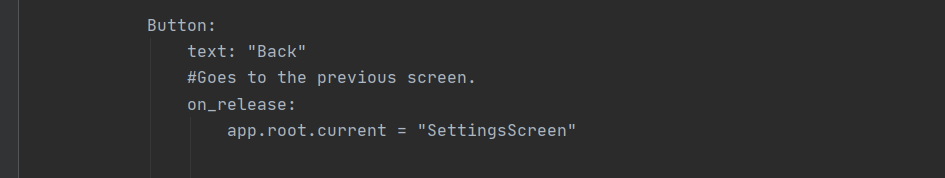
Settings Screen – Review

At this point, I’ve met the following success criteria:

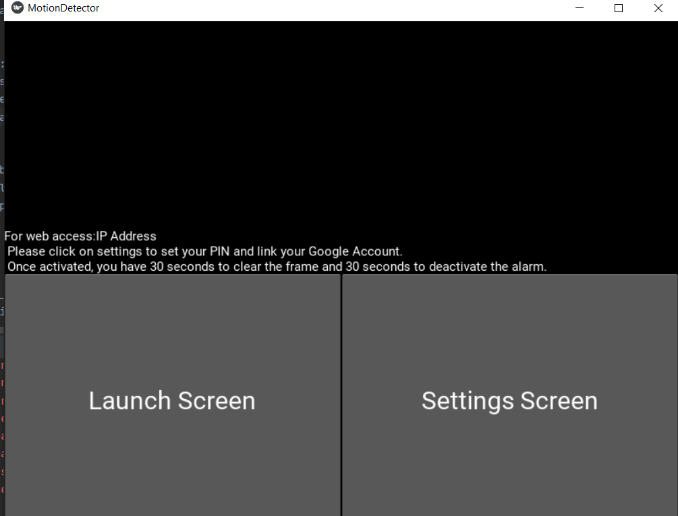
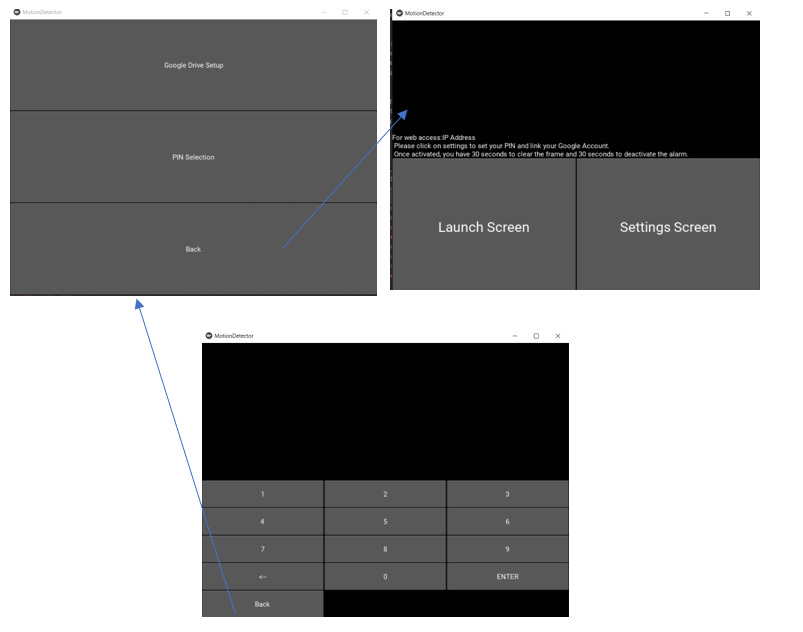
* The stakeholder use be able to choose security details when setting up the system.
* The stakeholder should be able to sign in to Google Drive when setting up the system.

I also met up with a stakeholder called Mrs Mistry. The feedback I got so far was that a back button would improve the usability as you don`t have to open and close the program if you need to go back a screen. Also, she said that the text on the first screen is not centred and that would look clearer.

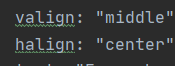


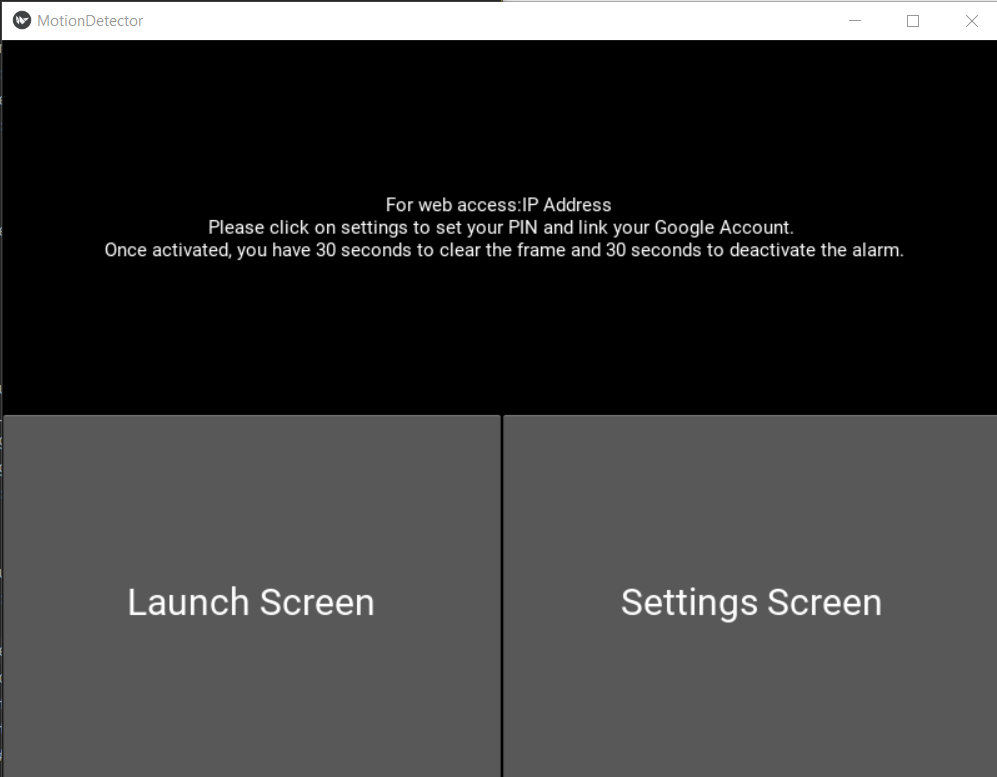


This part of the .kv file goes to the previous class using window manager when the button is released. Evidence is shown via staged screenshots below:

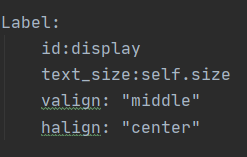
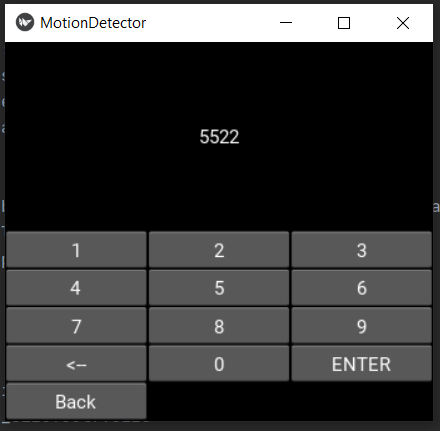


The text has been improved on the home screen by setting the virtual alignment to the middle and the horizontial alignment to the middle.





I have changed the alignment and make the size auto adjust on the PIN Pad display as well so this can be improved.

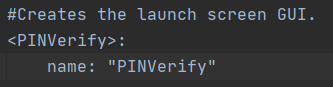


Launch Screen – 1st Iteration

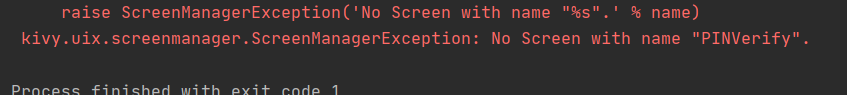
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | User enters PIN using keypad on screen – they have set this up in Settings. | 1111 – Valid Data | Open the OpenCV window. |  |
| 2 | User enters PIN using keypad on screen – they haven`t set this up in Settings. | 1111 | Error , cannot find “config.txt” |  |
| 3 | User enters incorrect PIN. | 1222 – Invalid Data | Open the OpenCV window. |  |

Initially, in the Design Stage, my plan was to copy and paste the same code from the PIN Selection Screen. However, using my knowledge of OOP, I thought that it would be best to inherit the PIN Selection screen. The only difference would be the “enter” function. I could use polymorphism to ensure that the current function is used. As “number\_entered” and “backspace” would be the same, if I don`t create the a method with the same name, this would automatically use method from the super class. A new function for “enter” will be created. Due to polymorphism, it will use the method in the sub class because this takes first priority.

A slight change I made to the original plan was to call the class “PINVerify” instead of “Launch Screen” as this is a clearer name. First of all, I created this new class in both the kv file and “main.py” script as shown below:

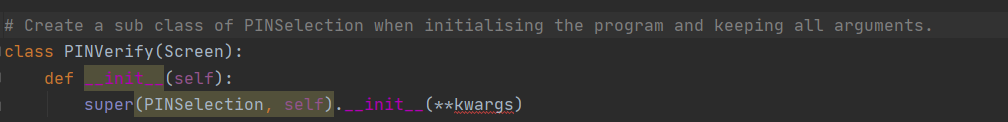






However, this resulted in the error screenshotted above because when I clicked on the “Launch Screen” button, it attempted to look for “PINVerify” but it can not do this unless the “Screen” class is inherited because this is needed for multi-screen Kivy programs.

I also attempted to use a constructor function to create a superclass of itself by inheriting all parts of the “PINSelection” screen. The protype code is below:



This protype creates a new class called “PINVerify” which inherits screen. Screen is needed so screen manager can identify this window and transition between it. When the script is ran, it will inherit all properties from “PINSelection”. However, this failed the tests because of this error:



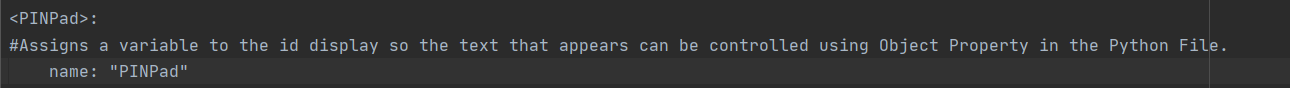
This is because in Kivy, you can only use the \_\_init\_\_ method once the application is launched. The program has attempted to run the constructor for this class but it can`t because the kv file has not been returned yet and this needs to happen first. This will happen last as Python is an interpreter and it will run code line by line. This means I cannot use this method as rearranging code will results it more errors because all the classes need to be set up before the .kv is returned.

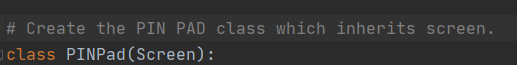
I also realised that this method would not work as the functions and variable would be the same but the parts of GUI (in the .kv file) will not be inherited so this would not work even if I didn`t run into these errors.

I hit a brick wall at this stage and decided to take a break for a few days.

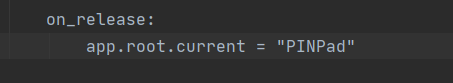
**RETURN TO PROGRAMMING:** When returning to this program, I had a new idea. Rather than using inheritance , I could rename the class to “PINPad” and use a variable how a method is ran. This would make the more space efficient as rather than creating another class and method, I could just use a conditional within classes.

Before, I used this approach, I had to edit earlier iterations and retest them to ensure they were working. The first step was to change the name of “PINSelection” to “PINPad” as this is more appropriate class name due to the new use.





Due to this change, I had to change the part of the Settings which divert to the PIN Pad Screen when PIN Selection is clicked.



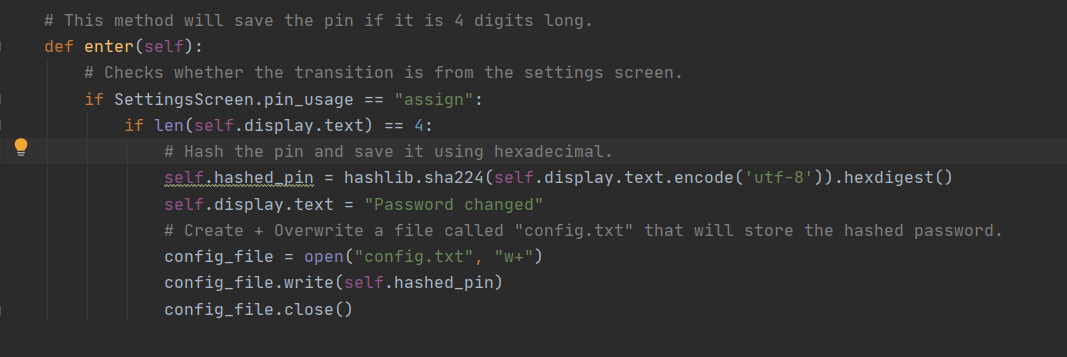
However, in order to identify what needs to be done, I had to add a function for when the button is clicked which would save a variable to identify what needs to be done on the PIN Pad screen.



The updated Settings Screen class creates a variable called pin\_usage when the script is ran. The variable selection then changes it to “assign” when PIN Selection is clicked in the GUI. I could have used a text box to ask what the user what they want to do once the screen transition is done. However, using the approach shown below, makes a better usability feature because it enables limited user input which reduces the amount of errors. This links to one of the features of system software, the buttons enable the user to use the program without knowing how the code works.



I also edited the method from the “PINPad” class because I needed to include a conditional with the new variable to ensure that it does the correct thing.



This takes the variable from the class “SettingsScreen”, if it is assign it runs the code that I previously programmed. This was designed to hash the pin and save it to a text file.

At this point, I thought it would be useful to that the variable has been saving correctly so I added a print statement on the first line of the “enter” method.



When testing it, it did not run the code within the first conditional because it was saved as “None” as shown by the output of the print statement below.



I fixed this error as shown below:

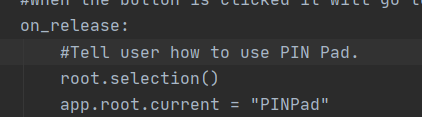


I had to use “SettingsScreen.” Instead of “self.”. This is because I needed to change it for all instances of this class instead of just one instance of this class. Once I fixed this, I ran this code again, I works as the variable had the correct value as shown below:

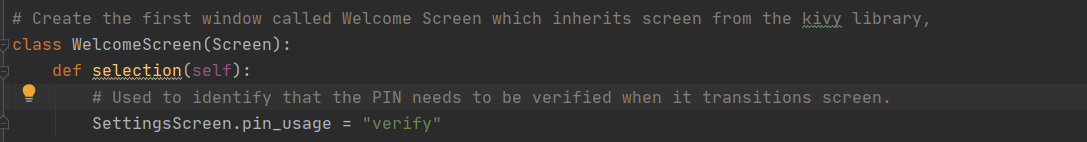


Appendix 5 shows the testing of the PIN Selection Pad. I enter a 4 digit PIN and it successfully changes which shows this new approach of using this class hasn`t broke the program.

Now, I can program the code related to this iteration. First of all, I had to change the .kv file for the on\_release part of the “Launch Screen” program. This is shown below:

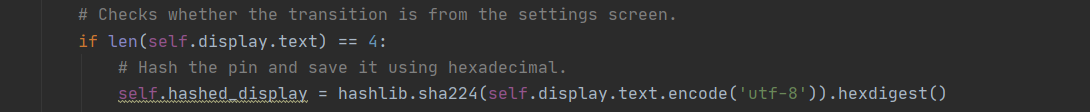


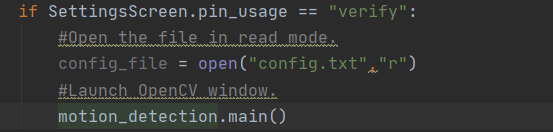
The method has the same name as the Settings Screen but it doesn`t matter because the methods are local to the class and they aren`t linked. The protype code for changing pin\_usage when the launch screen button is pressed is shown below:



This is very similar to the selection method on the Settings Screen is. The only difference is that PIN Usage is set to “verify”.

I changed the order of a some of the statements in the “enter” method (as shown below) because the PIN must be 4 digits for anything to happen and the display need to be hashed for file manipulation and conditionals. The variable is now called “hashed\_display” as it may not be the correct PIN so this is a more sensible name.





The protype code shown above opens the file with the hashed pin previously set up under the variable “config\_file” as well as running a function called “main” from my own library called “motion\_detection”. This is where all Open CV related operation will happen and gives my program a modular design. For testing purposes, I created the file and just put a print statement for testing in the motion\_detection script.



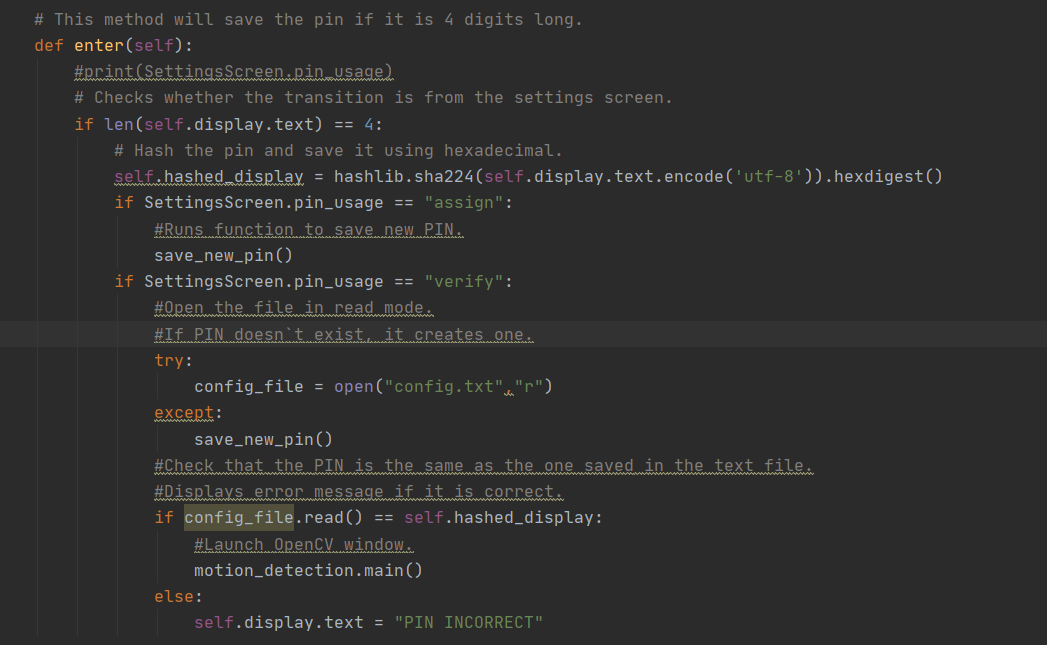


I ran this code and all tests were passed. Appendix 5 shows the results of Test 1 on the table. I entered the valid code and pressed entered and it successfully ran the main function because “test” was printed in the console. Appendix 6 shows the result of Test 2, it passed as it ran to an error as “config.txt” was not set up. I will create a fix for this in the next iteration. Appendix 7 shows the result Test 3, it passed as it ran the main function as soon as the PIN was entered. I shall work on PIN verification in the next iteration.

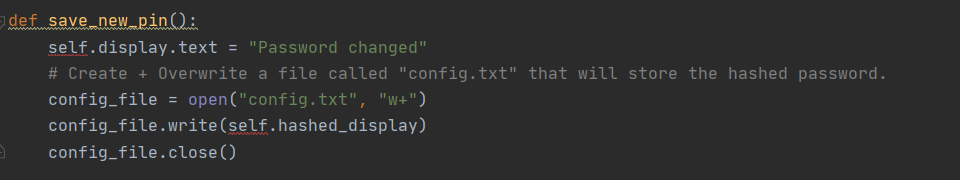
Launch Screen – 2nd Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | User enters PIN using keypad on screen – they have set this up in Settings. | 1111 – Valid Data | Opens the OpenCV window. |  |
| 2 | User enters PIN using keypad on screen – they haven`t set this up in Settings. | 1111 | Test cannot run – automatically redirect to create PIN. |  |
| 3 | User enters incorrect PIN. | 1222 – Invalid Data | Error message – “Incorrect PIN” |  |

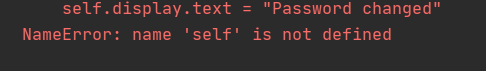
In the design stage, I thought that I would redivert the user to the create a PIN Screen if they haven`t already set one. However, at this stage, I think that it is best to just save the PIN if the “config.txt” file doesn`t exist. This is because of the changed approached related to the PIN Pad classes. This means that I have to attempt to the open the text file and if it doesn`t open I need to save the PIN that is on the display at that moment in time.



In order to do this, I decided to use try and expect. This means that Python will attempt to open the file and if it can`t it will run the function “save\_new\_pin”. This function is simplify the code within “assign” condition put into a function as shown below;



When testing this code, I ran into the following error:

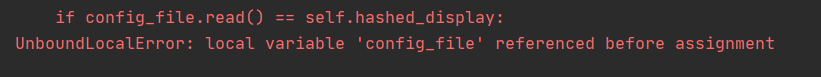


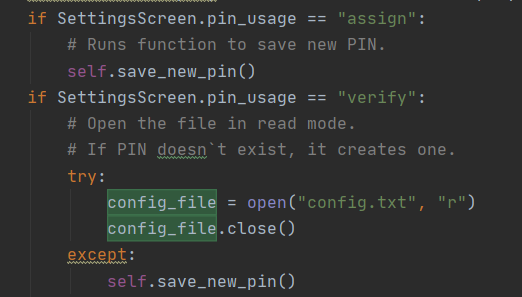
This means that self is required as a parameter so I fixed it by adding this:



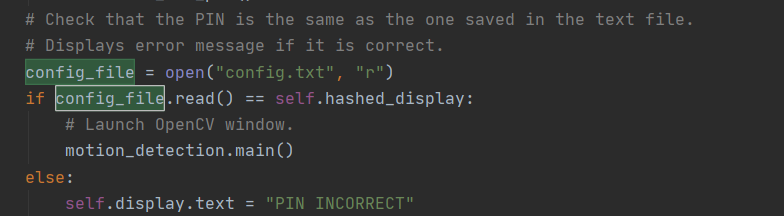


Again , test 2 failed due to the following error:



At this stage, I realised that I should change it and just make it a local method within the PIN Pad class as it is not required. This protects the function from everything that is happening outside the class which means that it is not part of the global scope. The error above is caused by the code in the try part of the code. This is because the method attempts to open the file in append plus mode although it is already open. I fixed this by closing the file before the function is ran. Using it within the class means that self is not required within the parameter as it is already part of the instance of the class. This prototype code is shown below:  


Test 2 passing is shown in appendix 9 because there is no config.txt file so it saved the PIN the user decides to enter and runs the main function from the motion\_detection library.



The protype code above shows the verification of the PIN the user has entered. This code verifies that the pin on the display is the same as the one saved on the text file. The file has to be opened again as it has been closed within the try so the method can be ran without errors. If the PIN is wrong, it displays an error message on the screen. The .read() function is needed so it checks the contents of the file. If it wasn`t there, it would check it against the properties of the file so the verification would not work. The verification at this stage is a usability feature because it is needed to check that the user knows the PIN before the motion detection starts so they can deactivate once they finish using it.

Appendix 10 shows Test 1 passing because it results in the OpenCV function running once the correct PIN is entered. Appendix 11 shows Test 3 passing because it results in an error message once the incorrect PIN is entered.

Launch Screen – Review

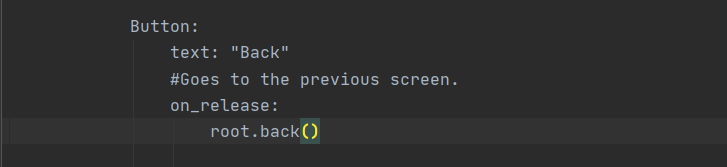
During the development of the Launch Screen, I met the following success criteria:

* The stakeholder should be able to enter pin to active the motion detector.

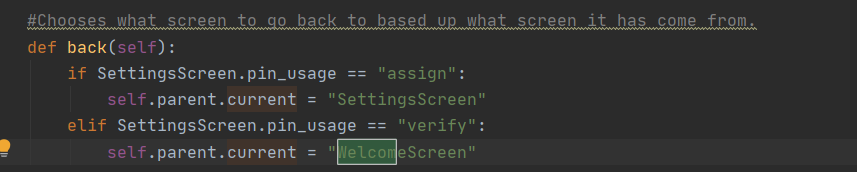
I met with Mr Makinson (A Home Owner) to get some feedback at this stage. The feedback I got was that:

* It would be more user friendly if the messages didn`t overwrite the text and were more specific.
* The back button from the Launch Screen doesn`t move to correct place.
* Need a message when a 4 digit pin isn`t entered.
* PIN should be wiped once enter is pressed.

In order to fix the back button , I would have to create a function with conditionals that chooses the screen that the window is changed to once the button is pressed.

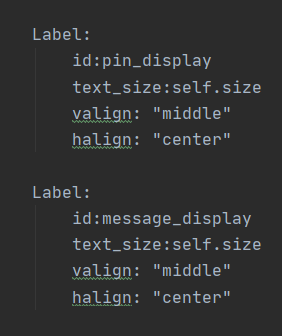


The prototype code runs a method called back when the “Back” Button on the PIN Pad is pressed.



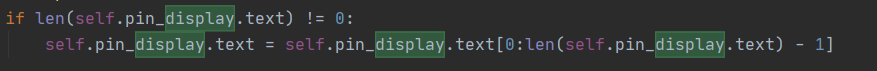
The following prototype code uses the variable “pin\_usage” which is set when a screen transitions from this screen. “Self.parent.current” goes to the parent class of the current instant of the class which is Window Manager. Window Manager has the property current which is set to the screen which needs to be displayed.

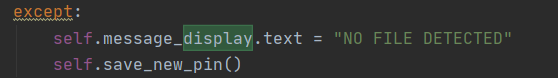
In order to improve the display, I could split it into two label one for the digits and one for the error messages. I have done this in the .kv file as shown below:



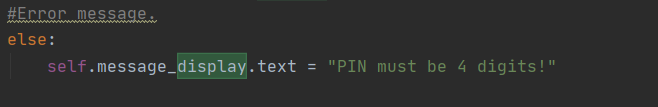
Appendix 12 shows the back button going to the corresponding screen.

As a result of doing this I had to change all the variable names as shown below (I have only shown a few screenshots as there is many) :

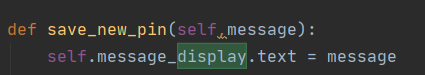


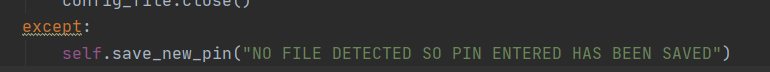


In order to create an error message when 4 digits aren`t entered I added an else statement in order to display an error message to the interface.



I have made my messages more specific by passing in what appears on the GUI as a parameter depending on what has happened. This is shown below:



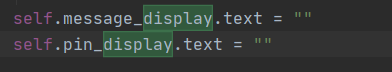




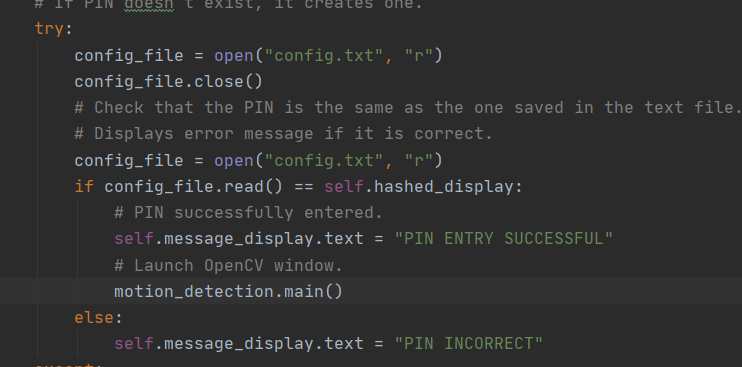
I have cleared the text after all the operations have occurred to ensure that all file manipulation and comparision have been done before it is cleared. In order to ensure this it is the last thing that is done in the “enter” method.



The text is also cleared when the back button is pressed. This is because due to it being the same class, if this doesn’t happen, all entered text and message will still be there. This would defeat the objective of the PIN verification as the PIN entered on the selection screen would still be there. The message is also cleared as this would not apply anymore.



When testing this, the “NO FILE DETECTED…” error message has stopped being displayed. Therefore, I changed the order of the code. I put the conditional that were previously outside of the try inside this. This means that the verification will only happen if the file is there. This is better for the user as this means that they have to re-enter it if they have set it on that screen. In order to make it clearly to the user that they have to re-enter, I changed the message to tell them to re-enter the PIN once it has been saved.



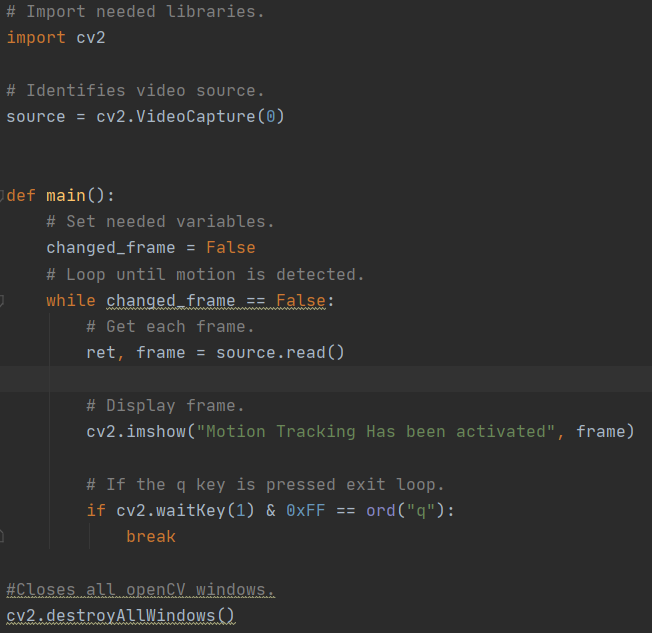


Appendix 13 shows the rest of the improvements and has audio commentary to explain what is happening.

OpenCV – 1st Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Video Feed | No change in what is appearing in video | Continues to show feed in window. |  |
| 2 | Video Feed | Change what is happening in video. | Continues to show feed in window. |  |
| 3 | File Location | Check file manager for a file with the video feed. | Will not find – function is not created yet. |  |
| 4 | Kivy Window | Change in video. | Will not change – function is not created yet. |  |
| 5 | File Location | Check Google Drive for file with the video feed. | Will not find – function is not created yet. |  |

The purpose of this iteration is to ensure that the webcam feed appears in a separate window so the user can see it.



The protype code imports the libraries need. “cv2” is the name for importing OpenCV for Python. Then, the variable “source” saved the default Video Capture device on the device. The number in the brackets indicates what source to use. 0 represents the default one. This could be modified in future versions of the program if I want to let the user decide what device to use. Then, the function main is needed so it can be ran from the other Python script as this file acts as a library.

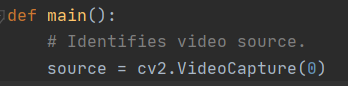
It runs a loop which will run until “changed\_frame” (a Boolean variable used to identify whether the frame has changed). This is set to False at the start of this procedure because the user has just activated the alarm. The rest of the code will run until this variable is changed.

It will then save the frame under two variables info and frame. The key part which is needed is “frame” as this saves the image. Then, it uses .imshow. This function has two parameters. The first parameter is for the title of the window and the second is the image that needs to be displayed.

During this while loop, it will exit the open CV window if the “q” key is pressed. This is not planned in my design section. I have decided to this add this as the user can just click “q” and enter the pin to deactivate their software once they return.

Once this loop is exited, the OpenCV window will be closed by calling the function shown upon the last line.

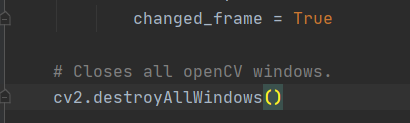
Appendix 14 shows a failed test because it does not launch the main python window once the enter button is pressed. In order to fix this, I have moved the position of the assigning of the source variable because it this may just be the system crashing due it using too many resources. I have moved in to the main function in the motion\_detection script because it does not need to be used until then.



Appendix 15 shows another failed test because of two reasons listed below:

* Using “q” to exit doesn`t work.
* The text saying “PIN Entry successful doesn`t appear straightaway”

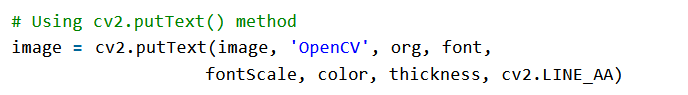
I have attempted to fix the exiting issue using the protype code below:



Although break was doing the correct thing, I changed the variable name as this loop depends on it so it will do the same thing. On Appendix 15, it shows the frame freezing when clicking q. This must been the while loop has stopped but the window has not closed. This issue was occurring because the in built function that closes the window wasn`t within the “main” procedure.

I was a bit confused why the next error was occurring because the changing of the text was before running the function. As Python is an interpreter, I could not understand why it was doing this.

Therefore, I decided to put this as text on top of the video. I did not know how to do this so I had to use the internet. I used the site: <https://www.geeksforgeeks.org/python-opencv-cv2-puttext-method/> and used the part of the website shown below:





This line of code take the frame of the webcam as the “image” parameter. The text needed is the second parameter. Org is the third parameter and are a set of coordinates telling the program where the text should go. It is in the format vertical then horizontal. The next parameter is the font, I have picked a simple font that is clear for the user. The fontScale is basically the font size in relation to the default size. The colour in a BGR format, the colour code I used is for red. “CV2.LINE\_AA” makes the text anti-aliased which blurs the edges together so it looks more uniform.



<https://graphicdesign.stackexchange.com/questions/41069/correct-anti-aliasing-color-when-moving-to-a-new-background> - SOURCE OF IMAGE

After doing this, I reran the test which can be seen in Appendix 16. As you can see, I entered the correct code which ran the function from the motion\_detection library. There was a change in frame as I closed and opened the privacy shutter but the program did not detected this change yet as it needs to be programmed first. This means that all tests (1-5) have passed.

Stakeholder Review – 1st OpenCV Iteration

I have met these success criteria at this stage:

* The stakeholder should be able to enter pin to active the motion detector.
* When the alarm is armed, the user should be able to see a live feed on a window on their screen.

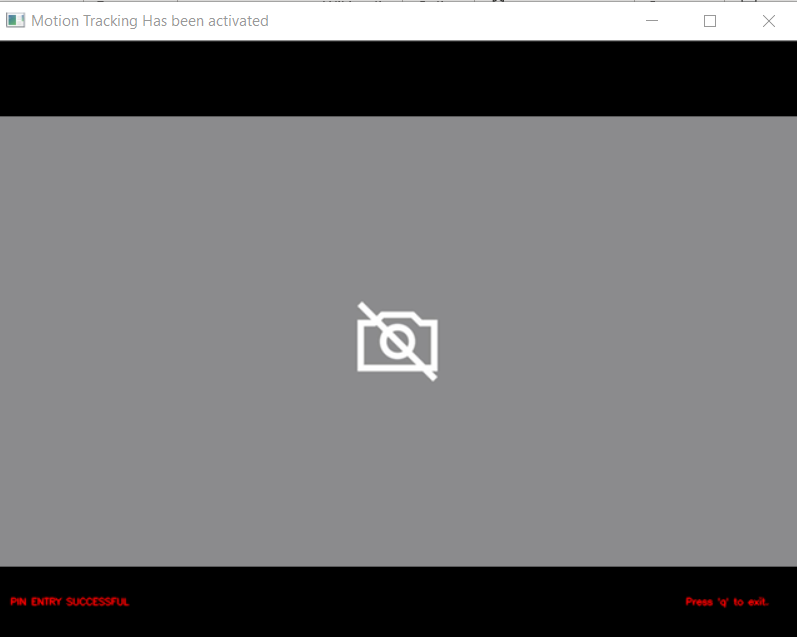
I thought this would be a good point to get some feedback on this program as a whole. The feedback I got was:

* Add instruction that tells you how to exit on the OpenCV Window.

In order to do this, I added it to the text on the screen using the code below:



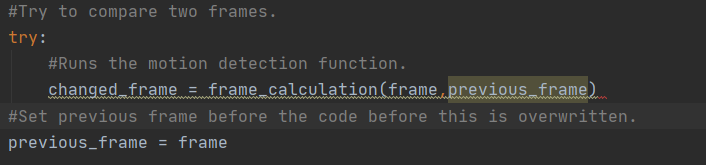
It is similar to the code for the “PIN Entry Successful” message. The only thing different is the text displayed and the coordinates on where it is pin. I want it to be at the bottom right because this is where Mr Makinson wanted it to go. This line cannot be merged with the line before as I want them to be in different positions. Therefore, they have to be done one at a time.



OpenCV – 2nd Iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Test Item | Test Data | Expected Result | Actual Result |
| 1 | Video Feed | No change in what is appearing in video | Continues to show feed in window. |  |
| 2 | Video Feed | Change what is happening in video. | Stops showing feed in window. |  |
| 3 | File Location | Check file manager for a file with the video feed. | Saves full footage with the name “test\_footage.mp4” |  |
| 4 | Kivy Window | Change in video. | PIN deactivation window appears. |  |
| 5 | File Location | Check Google Drive for file with the video feed. | Will not find – function is not created yet. |  |

At the first stage of this iteration, I would like to get the motion detection to work. I will do this by creating another function (as it returns something) within the “motion\_detection” file. This function will take in two parameters, one will be the current frame and the other will be the previous frame. The reason I have chosen to pass in parameters instead of using global variables is because this could cause issues related to variables which would be harder to debug. I have decided to make it a function that should return a Boolean value because this can directly overwrite the variable and this will have an effect on the loop and whether it continues. The reason it is in a try loop is because “previous\_frame” will not be defined the first time the loop is ran. If the try loop wasn`t there it would run into an error. “Previous\_frame” is set to “frame” at the end of one iterative loop because “frame” will be overwritten with the image of the webcam frame straight after this if motion is not detected. This is completely normally because there is nothing to compare first frame to. This protype code is shown below:

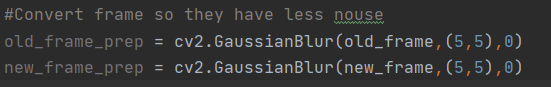


This function must be defined before the main function otherwise the program can`t find it. It is set up and looks like this:

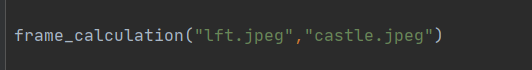


The first step is to change it to a gaussian blur so all noise is reduced. This includes things such as glare and changes in colour due to lighting. This is done at the start and this is the point where the frames are prepared. This is converted using a built in OpenCV function which is used for image editing. Technically, these frames are image are videos made up of multiple frames played back to back so it looks like the images are moving to the human eye. This is an example of decomposition as I am breaking down the video into frames so I can provide a solution for my Stakeholders. At this point, I was stuck one the parameters required so I used the link below:

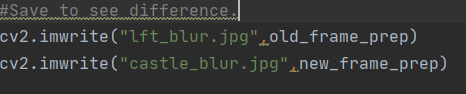
<https://docs.opencv.org/master/d4/d13/tutorial_py_filtering.html>



In order to test this, I put in images (taken by a phone camera) into the algorithm by passing them in as parameters as shown below. You can clearly see the differences and how the images have been smoothed which helps me when using the image subtraction function. I did this by using .imshow and passing the images in as parameters.



I also added code that will save the image to a new file so they can be compared. This code saves the image in parameter two using the name in the first parameter. Although, this will not be part of the final code.



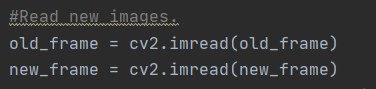
This resulted in this error:



At this point, I referred to a program that I have previously written. The purpose of this program was to compare two images and find similarities and differences. When writing these code, I was learning how to use OpenCV and the basis of the program was following this tutorial:

<https://pysource.com/2018/07/19/check-if-two-images-are-equal-with-opencv-and-python/>

The program that I had was that I hadn`t actually opened the file yet so no operations can be performed on it yet. Therefore, I used .imread as shown below:



Once running the code, it generated the images with the Gaussian Blur. The images are shown below to show the effect of this.

**Figure One: Original Picture of Concert Figure Two: Gaussian Blur of Concert**

**Figure Three: Original Picture of Castle Figure Four: Gaussian Blur of Castle**

****

Initially, I thought this process had not worked as the image don`t look like they have changed because of the similarities. Although, the original images have not got much noise which means that we will not see much effect. This is due to the fact that it has been taken from a good quality camera that has focused the image and some processing has occurring when they were taken. However, this will not be the case when webcam footage is being processed. This is because these often have a low amount of megapixels. As well as this, no image processing will occur before the frame is passed as a parameter to the function. In order to check if changes have happened, I decided to use a hash calculator online (MD5 <http://onlinemd5.com/> ). The values returned were:

Fg. 1 - B8F39BEC245F8E581972E6D8FCCAE795

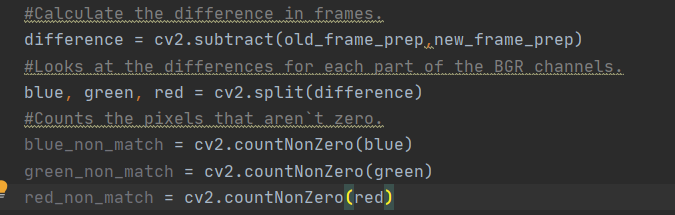
Fg. 2 - 5B002C2CEE9C736A4E6E6169D2AF4B69

Fg. 3 - 805E082D9388890ED11D9DC6DDC44C32

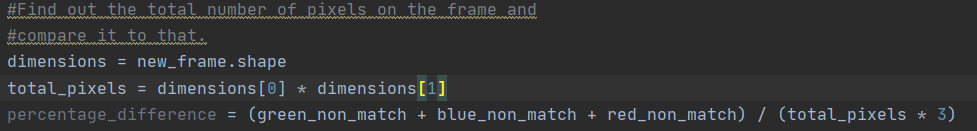
Fg. 4 - AADD751A9ED17847503FB234F00297E9

This shows different hash values for the images and this been the .Gaussian function is running correctly.

The next step is to compare the images.



This part of code uses the inbuilt .subtract functions. The two parameters are the Gaussian blur of the old and new frame. This inbuilt function looks at the RGB values of the pixels and compare them. If the colour of the pixel is the same it will be stored as 0 and if it is different will be stored as 1. After this I need to split up the images into the three channels (these are the main colours – RGB) to look at the difference. Then, I save the amount of pixels that aren`t zero to a variable called \*colour\*\_non\_match. I have used this approach because it saves me from writing code that has already been created from a library. I have used variables as these values will be required for calculations in the next step. The .countNonZero function is used because I can compared the number of pixels colours that don`t match in comparison to the dimension of the frame in the next part.



The next step was to get the dimensions of the frame for comparision. There is two methods of doing this. The first method involves assuming that every video recording device has the same dimensions. However , this is simplify impractical because this may result in a incorrect detection of change of frame in further calculations. Therefore, I have decided to the inbuilt .frame function. This function returns an array which has 3 values : height , width and colours. Then, I decided to calculate the amount of pixels that are needed for the comparision. I decided to make this its own variable instead of using it within the if statement because it may be needed again. Then, I have decided to create a variable which adds together the pixels that don`t match then divides it may the amount of pixels times by 3 as you have to take all three colours channels into consideration.

Then, i needed to make a conditional statement that chooses when to say motion has changed. In order to do this, I decided to take a burst shot on my phone (to replicate two continuous frames). Put the two images into the algorithm then round “percentage\_difference” to two decimal place as I think this would be the best method. This is because it would take into the order the slight changes in lighting that may occur during motion detection.

The two images I used are shown below:



This was done by passing the two images in as parameters like I did when testing the Gaussian Blur as shown below:

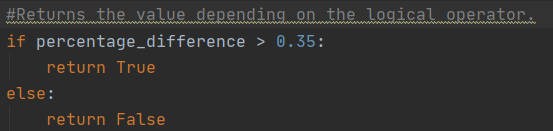


As well I this, I decided to add a print statement so the different is outputted to the terminal.



When running this code, it results in the following output:  


Therefore , I made the following conditional statement. I did this because if the “percentage\_difference” is above this value, we can assume that a person has come into the frame which means that the loop can break. This is done by running this a function which can assign the variable every time the variable is ran. I have taken this approach as it uses a more modular approach which will be easier to debug and all the separate parts of the systems diagram will be different.



Now once this is complete I decided to run tests 1 and 2. However, I ran into the following error:

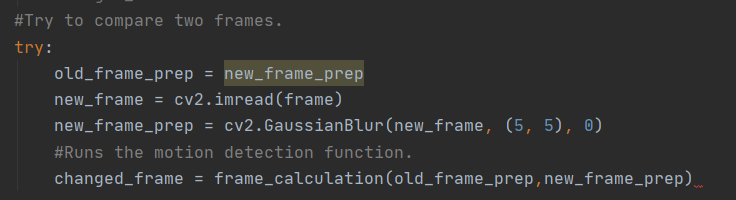


This is quite unusual because I am just assigning a variable. I decided to set the variable as “None” at the start to see if this was the issue. This was done as shown below:



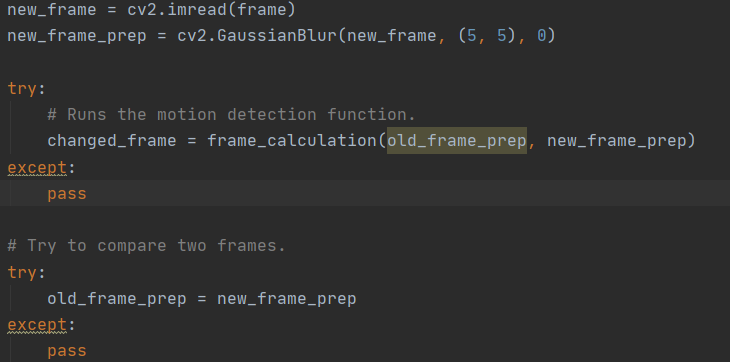


However, it still ran into the error so I decided to use the internet to create a fix for this issue. I couldn’t really find anything. I think it was because I setting a variable as an image which is not a primitive data type and maybe stored as an NumPy array. Although, I didn`t originally decide to take this approach, I decided to blur the image before this function is ran as shown below:

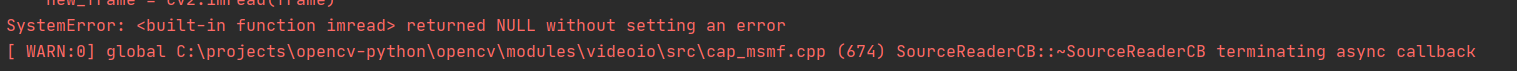


This may work because it will be a different data structure that the variable can be changed to. This code will change the variable “old\_frame\_prep” to “new\_frame\_prep”. This variable would have been assigned from the last time the loop was ran. This is why the try procedure is needed , as we need to make sure that it doesn`t run into errors when it attempts to do this the first time. All this code has been explained previously, it has just been moved from the “frame\_calculation” function. This was the next point where I could retest it.

Look at Appendix 17 to see testing at this point. At this point, I released that it the motion isn’t been detected as I used “q” deactivate it. To decide whether the try loop is the issue, I put a print statement within the “frame\_calculation” saying “TESTING” to see whether the function is even running. When I ran this again, it wasn`t which means that this is the issue. I then realised that the order is the issue. As soon as one line of the try function doesn`t work it will go to the “expect”. Therefore, I wanted to split it into two different try functions as shown below:



This part of the code, does the blur to the current frame first. Then it runs the function to see if there is a different in frame. This is done first as you need to overwrite “old\_frame\_prep” last as this is needed to use it the next time the loop iterates. However, this resulted in the following error as it has already been opened at the start of the iteration of the while and it does not need to be redone as it is not passed in as a parameter to the function.



Therefore, I deleted the imread line and replace “new\_frame” with “frame” as this is what the current frame is saved as shown below:



Another error, I detected was on this line:  


I have changed the parameters that have been passed in. As this function runs within a try function, it will not display an error saying that “new\_frame” is not defined. Therefore, I had to put the blurred version in the frame. This will not make a difference as the dimensions will not change when they are blurred.



When testing this, it works when the privacy shutter is on as the laptop shows an icon of a camera with a cross on it as it is a static image. Then, when I turn the privacy shutter off it closes the OpenCV window as this detects a change in image. However, if I just start the program with the privacy shutter off, it automatically goes off after one frame which means that it has been detecting a change which may be due to lighting or may be related with an issue with camera quality. This would make sense as the .subtraction function works in the program I have borrowed the code from as this was designed for image comparison. Image comparison is much different because you don`t have to take into account slight changes in frame caused by lighting, a slight movement, etc which means that this is an inefficient solution.

The main problem of this program is the comparison statement that tells it to exit if “percentage\_difference” is greater than 0.3. This number has been decided by taking two burst shots and comparing the value which means that it is inefficient. This doesn`t take into account how the person enters the frame. Depending on the position of the camera, only their head may be showing or their full body may be showing. This will change the comparison completely.

Looking for matching pixels is also a bad idea, this is because of the camera issues and lighting. Light can affect the colour of each pixel greatly. Also, if the camera uses auto focuses the pixel will change the computers image will become clearer but the computer will not acknowledge this. Ultimately, with all this in mind, I think the best thing to do is use a tutorial that uses an alternative method. One that looks for certain elements would be more efficient because it could look for the human outline. Therefore, the outline of a person would appear once someone enters the frame. If we looking for this, it removed the need for having a comparison statement which could make the program faulting in certain scenarios. Due to making this decision, my next step was to use the following tutorial:

<https://www.youtube.com/watch?v=MkcUgPhOlP8>

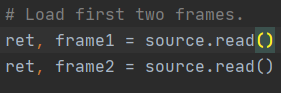
Before, is start redeveloping this, I would like to talk about the elements from this tutorial that aren`t required. I don`t need a box drawn around the person instead I just need to break the loop once this happens. Also, I don`t need to say when my room is occupied or not occupied using text on my image.

I decided to change my code quite a bit as it was better to use the tutorials code as a template and redesign my code according to this. I am going to take screenshot of my code along with the code from the Video tutorial (which was in the link in the description and stored on GitHub - <https://gist.github.com/pknowledge/623515e8ab35f1771ca2186630a13d14>).

First of all, I commented out the previous code I had development in case this didn`t work. I didn`t delete it in case I had to go back to it. The start of this tutorial was very similar to my previous attempt because it imported needed libraries as shown below.

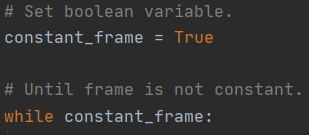
\*For future reference, my code has a dark background and the tutorial code has a white background.



As you can see, I only had to import OpenCV. Numpy arrays are used to store most OpenCV objects and this library is needed in order to work with them. However, I will not need to operate on any of these data structures for the code I have adapted. Then, I had to read the frames. This is heavily adapted from the protype because I have decided to get the first two frames before it entered the While loop. I have done this because it removes the need for the try which reduces space complexity which makes my program more efficient.

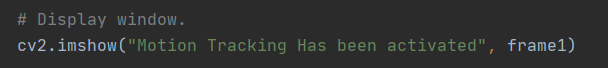


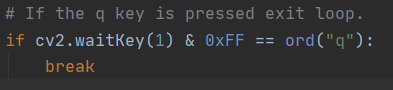
The only difference here is the cap. Instead of reading the capture, I have to read the frame. This does a very similar thing as it returns the image from the webcam. However, the code from the tutorial, take a pre-recorded .mp4 file. Another difference is how the while Loop runs.



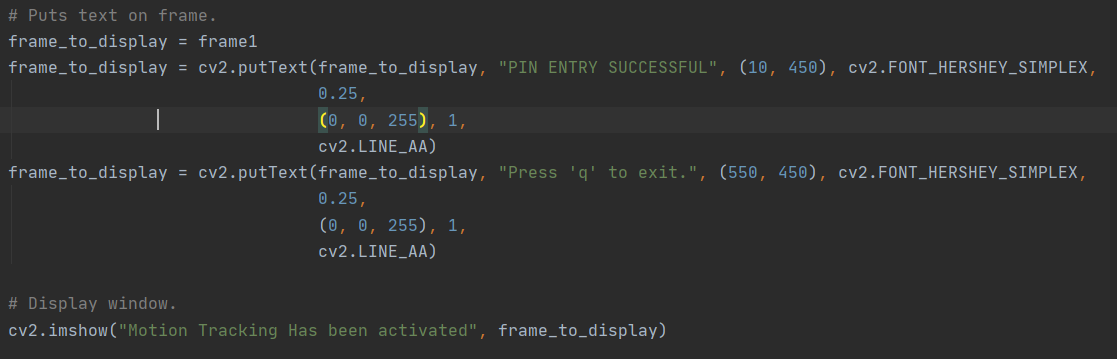
I have assigned a frame that is set to true as the frame is constant until motion has been detected. In order to prevent unnecessary code, I have left “While constant\_frame == True” because the default statement for a while loop is true. This is different to the tutorial as the while loop only works till the end of the video. While on the topic of video, the tutorial has implement recording to a video file and parts related to this has been removed because I am using an iterative approach where I do one part at a time.

Other parts that have been saved include the window title and how to exit the window because this was an improve suggested by a stakeholder which have previously been explained.

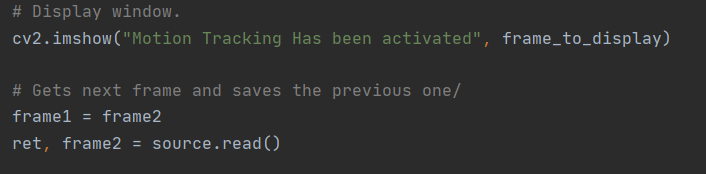


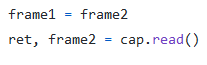


Another change is how the text at the bottom of the window appears. As this may be detected as contour (will explain soon), we can`t let this be part of “frame1”. Therefore, I copy the context of the variable frame1 into a variable called “frame\_to\_display” which then adds text as I previously explained. Then, instead of frame1 being displayed “frame\_to\_display” is shown to the user.



Similarly, like last prototype, the frame needs to be read again and the old one needs to be saved somewhere in order for comparison purposes. Rather then preparing it first, we just save the frames as they are. I do this because with background subtraction (explained soon), the blurring ,thresholding and conversion to greyscale must happen before the contours are found. Therefore, we display the image, then read the next frame and save the old one as shown below.

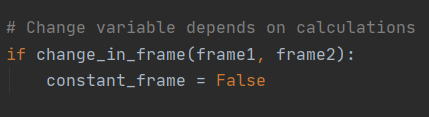




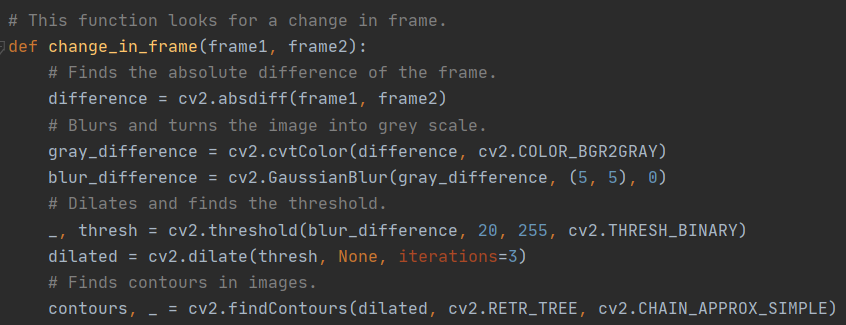
Again, this looks slightly different as the tutorial is reading from a video stored on the device. Then, I had to start comparing frames. Unlike the tutorial, I don’t do this immediately after the program loading up. This is to combat focus issues in case the camera needs to adjust for reasons like lighting which was one of the flaws of the last solution.

This then builds the main() function. I have decided to make it modular unlike the tutorial in order to keep the approach I choose in the design stage. I think this is key as it is easier to debug if each separate component on the system diagram is in a different function.

There is a function which returns true or false depending on whether there is a change in frame. If there is a change in frame, “constant\_frame” is now equal to False which means the loop will be exited. I have not done an else statement that sets it to True as it is already this so it doesn`t need to be done.



This function takes in two parameters which are the two frames that need to be compared. The image is compared using inbuilt functions as shown below:

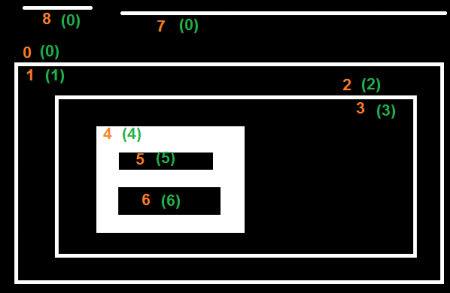
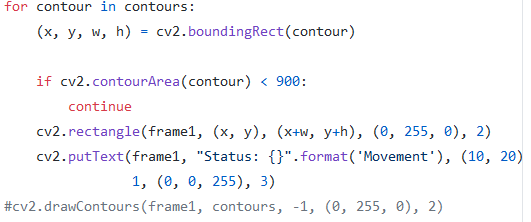


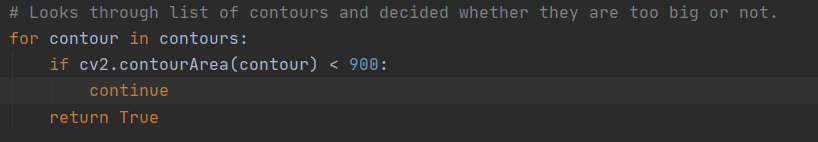
This part is the same as the tutorial. However, I now understand how it works. The variable difference is calculated by finding the absolute difference of the two frames. This approach is similar to my last approach but rather then looking at the whole image it looks at the background. This should work as the background is static and this means that there shouldn`t be slight differences in the frame.

After this the image of the compared frame (saved under difference) is converted to Black and White. The reason this is required is because thresholding only works if the image is in this format. It is then blur to removed image noise. Image thresholding then occurs, this is needed to create an almost cartoon like image which simply contains the outlines of shapes and objects. The reason the underscore is there is because it returns two variables however we only need to the second returned one. The underscore is standard practice in Python for variables that don`t need to be used.

Dilating isolates some parts of the images in order to make the edges clearer. This process is done three times which is indicated by the iterations parameter. Finally, the dilated image is taken and the contours are found. The second parameter is to decide how you want the returned data to be structured. I use .RETR\_TREE as it uses an abstraction of a tree data structure in order to create an image representation of this. An example is shown below:

**SOURCE:** [**https://docs.opencv.org/3.4/d9/d8b/tutorial\_py\_contours\_hierarchy.html**](https://docs.opencv.org/3.4/d9/d8b/tutorial_py_contours_hierarchy.html)

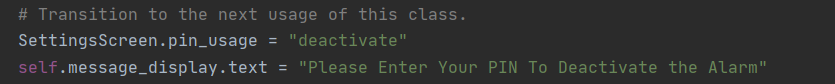




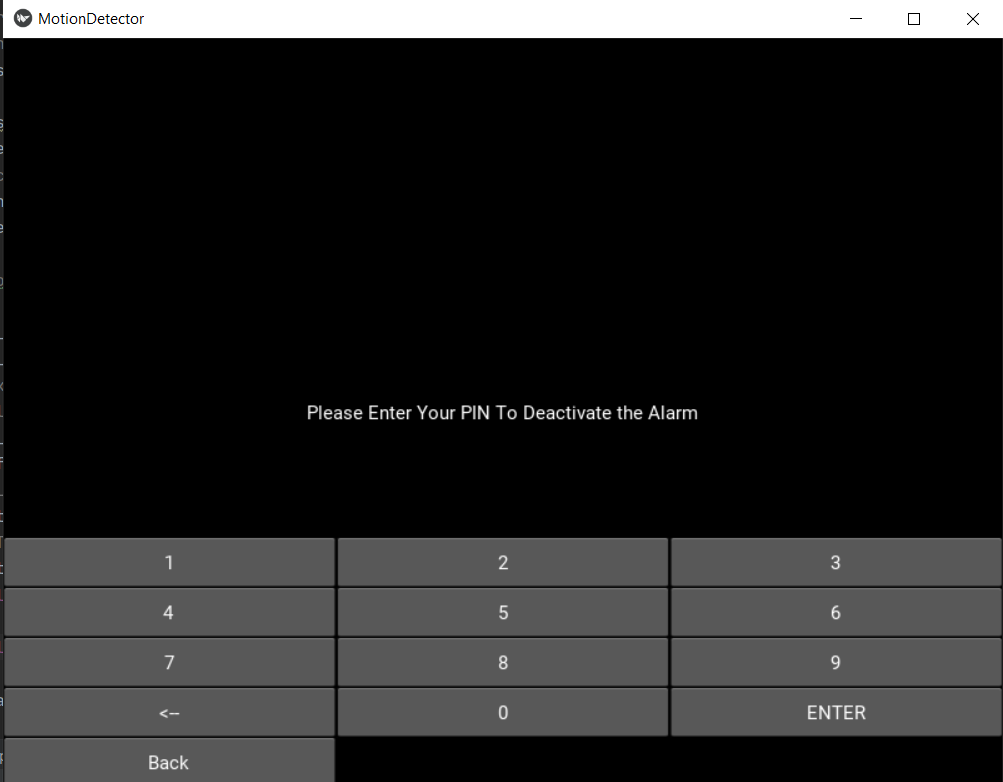
This is one of the most heavily adapted parts of the tutorial. The tutorial draws an box around where the contour is detected although this is not a requirement for my program. The if statement checks whether the contour is too small or not. This is needed to prevent accidental deactivations. If this is not the case, it will return true resulting in the alarm deactivating. The reason the else is not needed is because the continue keyword will result in the programming going to the next iteration without completing the current iteration.

Appendix 18 shows the user successful entering a PIN and then the window closing once I wave at the camera which means that Test 2 has passed.

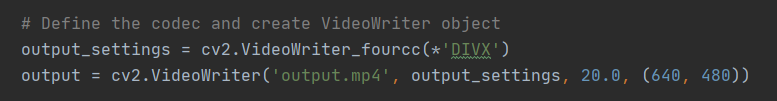
Next, I need to make the PIN Deactivation screen appear. At this stage, it would make sense to display instruction which instruct the user to enter the valid pin at this stage. As this is already set up within the PIN class, we can just change the text within the python script. This is different to my Design section as each different PIN Pad was meant to be a different class. However, by doing this, I can simply run this immediately run this after it runs to the open cv program instead of needing extra subroutines which increases the space efficiency of my program.



This code shows me changing a variable. Although, this variable will not be needed in order to complete this iteration it will be needed for the next one. As well as this, if I don`t put it in, it will remove one of the purposes of this variable which is to keep track of how this class is being used. Anything done after this, will be related to deactivation. Hence, it is worth doing this, just to avoid any logic errors from this point on. Once it exits the OpenCV function, it will display this text. This code is design to meet test 4. The image below shows evidence of a successful test.



The next step consists of saving a video file of the whole time when the alarm had been activated. In order to do this I referred to a program, I had previously developed. The intend of this program was to save all webcam footage while the webcam was running and was developed while I was using the OpenCV library.



The first step is doing this is by telling the program how it is going to saved. “Fourcc” is a code which tell the program which format to used . I have used DIVX as this is tested and supported for windows so this would work. Then, I tell the program to output this to the file name specified using the settings provided on the previous line with the dimensions shown on screen.



Every time the while loop is ran, it will add the frame to the video. The reason it is put inside the while loop is because a new frame from the webcam is captured every time the loop is ran which means that the output needs to be updated.

Now condition 3, has been met the evidence of the video successfully saving is shown below. The file is shown in file manager as well as a screenshot of the video playing:





Stakeholder Review – OpenCV 2nd Iteration

The following success criteria have now been met:

* The stakeholder should be able to view the file of when motion was detected using their file browser.

I met with Mr Mistry (A Pub Owner) who said that the timings of the motion detection could be improved. He thinks that the program should start monitoring after 20 seconds so people have time to move out the frame. In addition to this, he thinks that the video should only go off after 20 seconds so the criminal may be caught on camera so it can be used as evidence.

In order to do this, I would have to use the date and time module so I started by importing this.



This library would allow me to operate on time in the most simplistic way possible. As this is built into Python, I did not have to install any additional libraries. The first step was to record the time that the program begin. In order to do this at the earliest time possible, I did this as at the very start of the main program which is essentially the start of motion detection.



Then, I had to find out the time that had occurred since the tracking had begun. I did this using the subtract operator. This is done between the current time which is given by the .now function. The reason subtraction can occur on a non-primitive data types is because they are both the same format and this library will have a precompiled program which tells the user how to do this. Once it can do this, it will see this value exceeds twenty seconds. I could have found out how many seconds have elapsed using string manipulation but this would have taken more lines of code. Also, libraries are written to avoid library of code and use stable program which would make the program more efficient and less likely to have bugs.

Once it does this it must check whether it has been twenty seconds since the motion event has occurred before exiting the program. I had to be careful here because I had to make sure that the time\_detection variable is not overridden every time the loop is ran otherwise it would be stuck in an endless loop. Therefore, I decided to use try and expect. The program will attempt to use the variable and if it can`t it will be assigned to the time when it is ran.

The next comparision works similar to the previous one. It will find out how much time has occurred since the motion was detected by comparing it to the time as it is ran. Once this is over twenty seconds, it will changed the variable to False which will result in the loop existing. This will automatically change how much is recorded as recording stops once the while loop is existed.



However, I did have issues creating this because my original attempt looked like this:

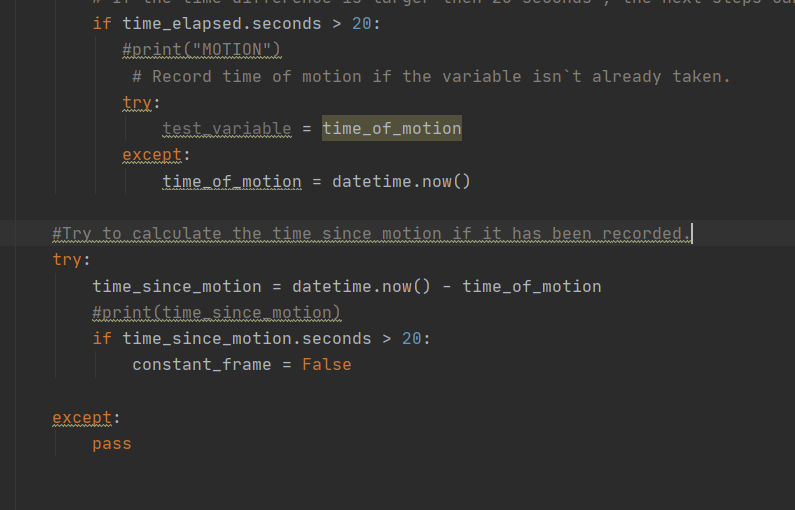


Instead of using try and expect in this loop. I attempted to use an if statement. This if statement would try and find the local variable “time\_detection”. If it was there, it could do all the operations needed for comparision as explained below. However, I was not sure why it didn`t work but it resulted in the program freezing on the frame once motion was detection. I found out this method as I originally was wondering whether you can check whether a variable exists and do it that way. Therefore, I used the website: when developing this stakeholder suggested improvement.

<https://stackoverflow.com/questions/843277/how-do-i-check-if-a-variable-exists?noredirect=1&lq=1>

However, when testing this, it only worked to a certain extent. The issue was that it was within the if loop so this code only runs when motion was detected. Therefore, what had to happen was a second motion event after 20 seconds of the first motion event. This was the only way that the pin deactivation window would appear.

In order to fix this, I would have to rearrange the code in some way.



I did this by attempting to use the “time\_of\_motion” variable if a motion event is detected. This means that if the code inside the try function works, the variable will already be saved so a motion event has already been logged. The 20 seconds will be recorded from the first valid motion event. If this variable isn`t set. It will set it if the try function doesn`t work.

Then every time the while loop is ran, it will attempt to compared the current time to the time that the motion was detected using subtraction. Once this difference becomes larger than 20, it will change the Boolean value of the “constant\_frame” which will result in the code breaking from the loop once the current iteration of the loop is finished.

I also have an except part which just does nothing as the loop must continue as normal if there isn`t a motion event. I sent this code back to the stakeholder and he was happy with the improvements which means that I can progress onto the next iteration. Appendix 19 shows me testing it by triggering motion events in this first twenty seconds but this doesn`t do anything. However, once I trigger a motion event after this, the alarm deactivates from 20 seconds at this point. A stopwatch is also present on the screen so the timing can be verified.

Diagram, schematic

Description automatically generated

Testing Screencasts: [2][3][4][5][7][8][9][10][11][12][13][14][15][16][17] [18]

<https://staffasfcac-my.sharepoint.com/:p:/g/personal/19mistryvivek_asfc_ac_uk/ESUI_rlFRl1MjWTtuO7K1L4BlsvKs1Pp6ol6gtCbXBOA4w?e=RFIQ6S>

or in zip file with this document.