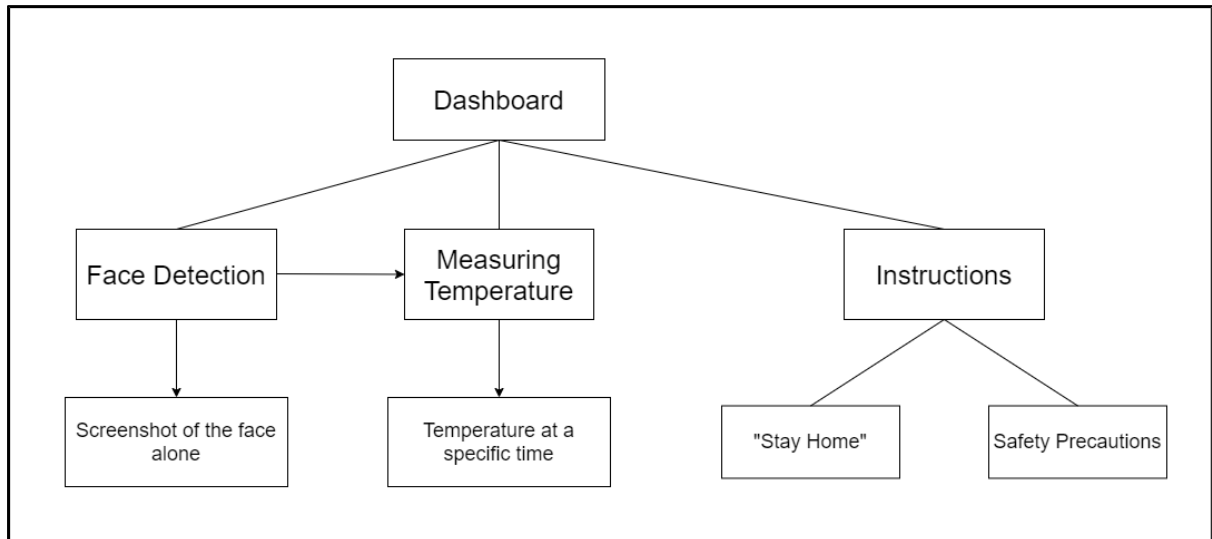
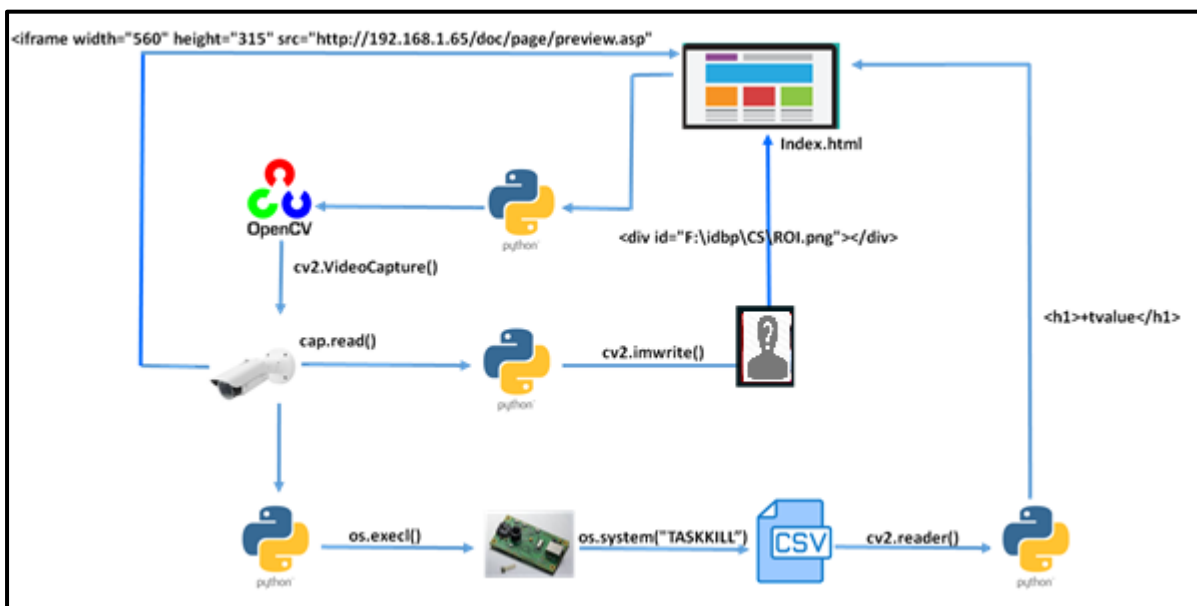


## CRITERIA B: DESIGN

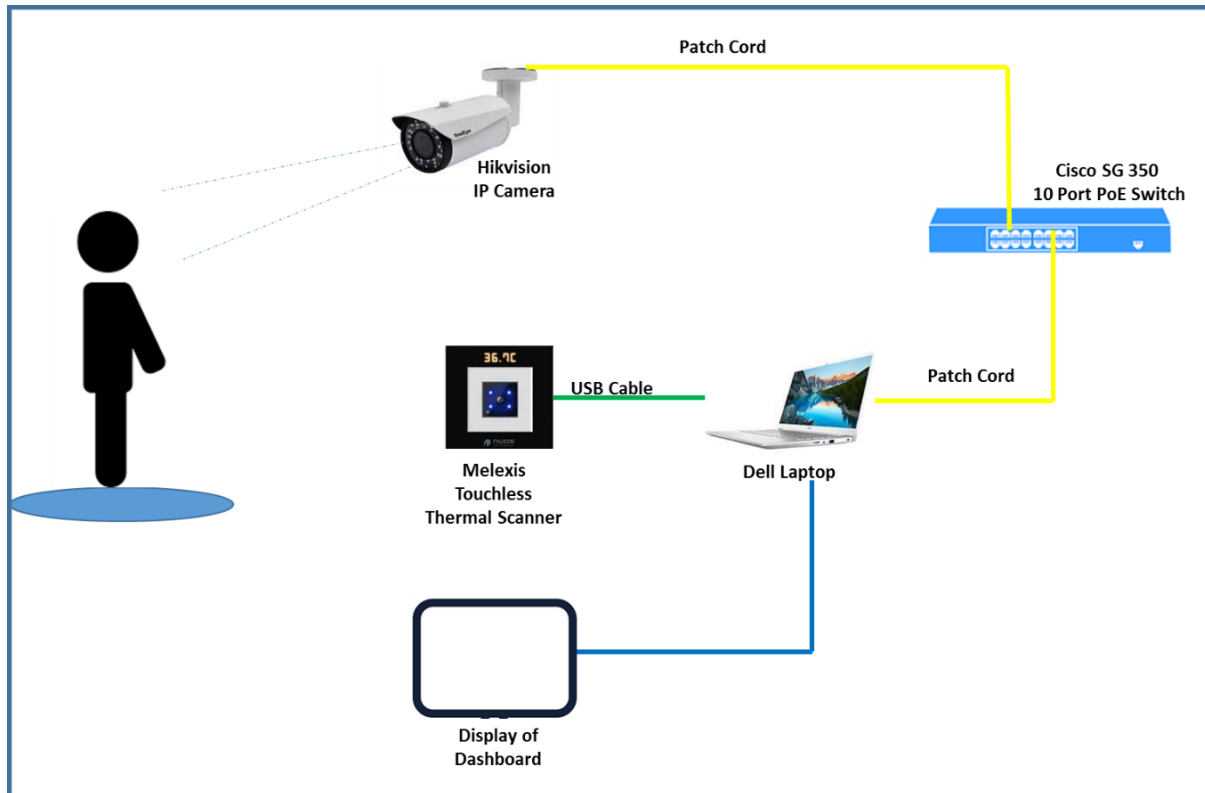
### GENERAL VIEW



### PROGRAM OVERVIEW



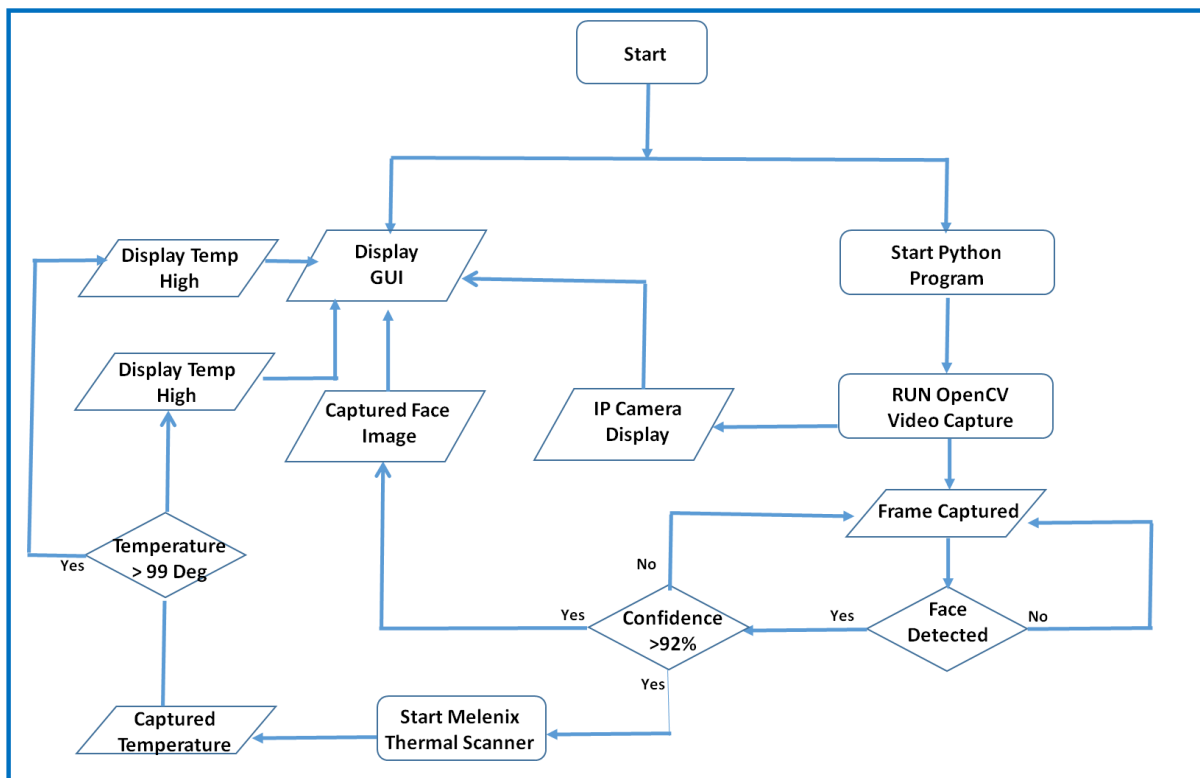
## HARDWARE LAYOUT



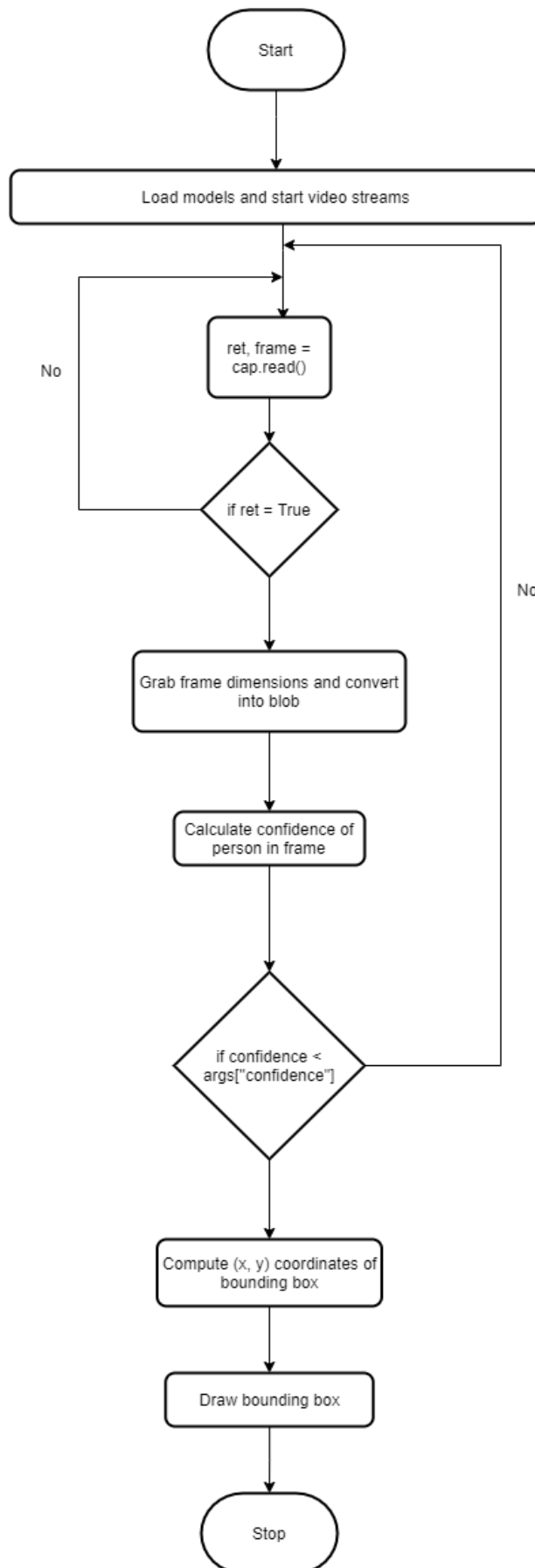
Hardware Component	Purpose	Aspect of project
<b>HikVision IP Camera</b>	The purpose of an IP Camera is to capture a video stream from the lens. Using OpenCV Python, this stream can be stored on the laptop or a certain screenshot of the stream can be captured and stored.	The IP Camera is connected to the HTML Dashboard and keeps running. When a person is identified in the stream, the camera captures their face and the related Python program starts the temperature detection program.
<b>Cisco SG 350 10 Port PoE Switch</b>	Different devices connected across a Switch Port have a shared LAN network and can share information.	The video stream from the IP Camera is shared with the laptop via the LAN Network by connecting both devices to the Cisco Port Switch
<b>Melexis Touchless Thermal Scanner (MlxCIPT 90632 3)</b>	Thermal scanners measure the temperature. Touchless ones do not require physical contact.	Since the overall context for this project is the coronavirus, the aim is to minimise physical contact. The touchless scanner measures the temperature, as required, with any contact. This reduced the risk of infection.
<b>Laptop</b>	Laptop is a portable hardware device composed of many hardware and software features.	The laptop presents the dashboard to the customer and the owner of the shops.

<b>USB Cable</b>	USB Cables and Patch cords are wires that connect two hardwires. This connection enables the transfer of data from one device to another	The USB Cable connects the Melexis Thermal Scanner to the laptop, feeding the temperature to the CSV file and then to the dashboard.
<b>Patch Cord</b>		The Patch Cord connects the devices to the Switch Port.

### **FLOW CHART FOR ENTIRE PROGRAM**



## **PROCESS FOR FACE DETECTION**



## Execution:

*Python detect\_faces\_IP.py --prototxt deploy.prototxt.txt --model  
res10\_300x300\_ssad\_iter\_140000.caffemodel*

- *detect\_faces\_IP.py*: The python file that begins the face detection and then spurs the rest of the program.
- *deploy.prototxt.txt*: It is a prototype machine learning model for Caffe. They create .caffemodel files.
- *res10\_300x300\_ssad\_iter\_140000.caffemodel*: This is a machine learning model by Caffe. It classifies and segments images. This helps capture the stream, capture the blob and the bounding box.

## Load Models and Start Video Streams:

*net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])*

This invokes the Prototxt and Caffemodel files

*url1="rtsp://admin:1234567a@192.168.1.65:554/doc/page/preview.asp"*

*cap = cv2.VideoCapture(url1)*

*print(cap.isOpened())*

- *url1*: This is the webpage of the video stream from the IP Camera.
- *cap.isOpened()*: This signals that the stream *url1* has been captured and is now streaming for the program

## Confidence:

*confidence = detections[0, 0, i, 2]*

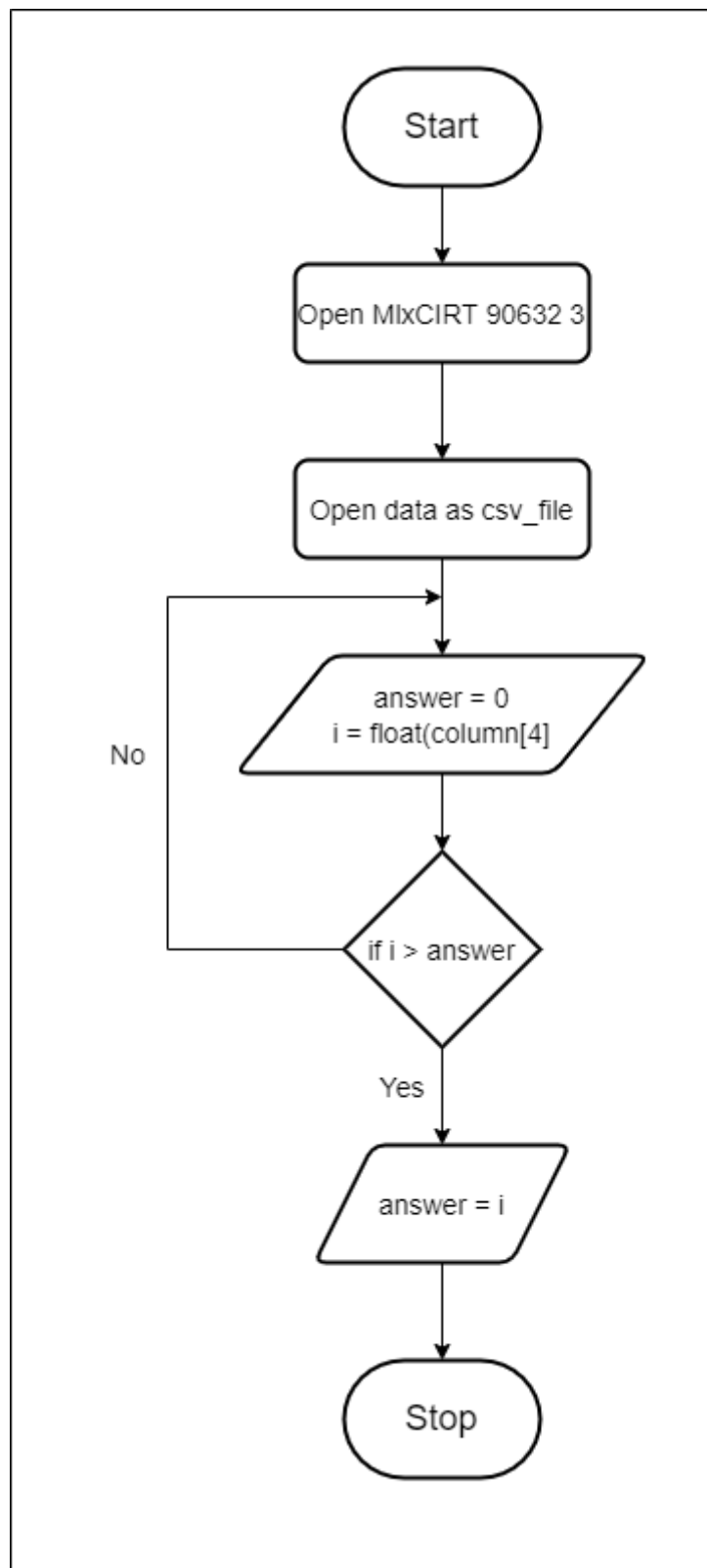
This extracts the confidence of the probability of the prediction (person being in the blob) being true.

*if confidence < args["confidence"]:*

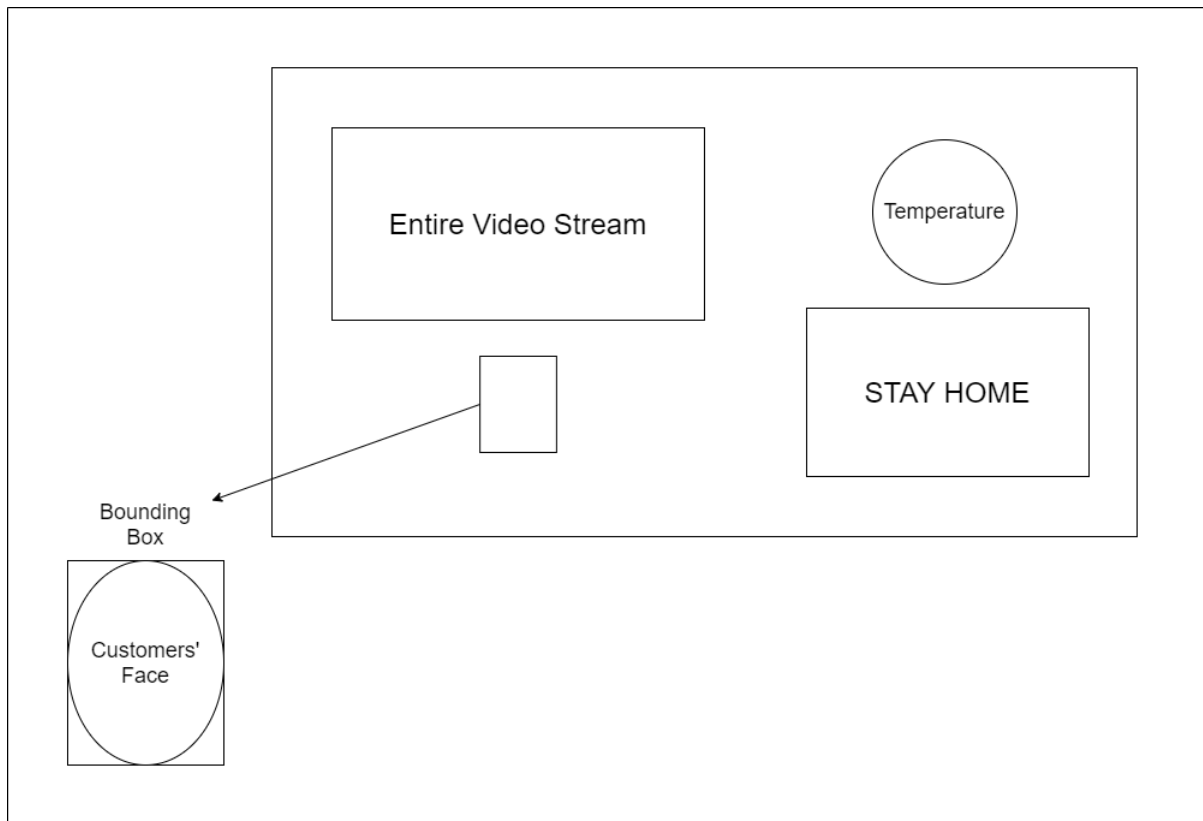
*continue*

The condition ensures that only blobs with a minimum confidence level would be considered by the program and for the temperature detection. This confidence level is predetermined by the owner or the programmer.

# PROCESS FOR MEASURING TEMPERATURE



## **DASHBOARD LAYOUT**



## **DEVELOPMENT PLAN**

This is the plan for the development of the project. As in the general overview of the project, this plan has been divided into three aspects: face detection, measuring temperature, and the HTML dashboard. Although the development of the code may take 1 week for each part, the connection to additional hardware equipment for the first two parts requires sensitive handling. Also, the face detection and temperature scanner code need to be connected to one another. This would extend the time required for the development of face detection and measuring temperature.

### **Face Detection (3 weeks)**

- Buy IP Camera, Patch cords, Switch ports
- Write the Prototxt file
- Write the Caffemodel file
- Write the main code:
  - Connect to the IP Camera
  - Verify if person is there
  - Capture bounding box

### **Measuring Temperature (2 weeks)**

- Buy Melexis Thermal Scanner and USB Cables
- Initialise Thermal Scanner
- Write code to sense temperature

### Developing Dashboard (1 week)

- Write HTML code independent of above two parts
- Connect face detection stream, bounding box and temperature to dashboard

### **TEST PLAN**

<b>Action test</b>	<b>Way of testing and result</b>
Test if the program runs	Enter the execution code in the Command Prompt. If the execution code gives an affirmative remark, it means that its running
Check if the dashboard opens	After a small amount of time, the dashboard needs to open with any Internet browser (Chrome, Microsoft Edge, Internet Explorer, Mozilla Firefox)
Check if the video stream is smooth	The video streams should have no interruptions
Check if face is identified properly	When a face is identified, the confidence must be higher than 95% and the image must be clear
Check if bounding box is captured	The bounding box in the bottom-left quadrant of the dashboard must show the image captured from the stream but focusing on the face, with good quality
Check if temperature scanner works	Open the MlxCIRT application and execute locally
Check if CSV File for temperature scanner works without interruptions	Open the related CSV file and test the temperature scanner
Check if temperature in dashboard is equal to the maximum temperature in column 4	Once the temperature is captured, the temperatures in the CSV file must be revisited. If the temperature is the highest temperature captured in a specific time period, the dashboard temperature is accurate
If temperature is greater 99 degrees Celsius, check if there is a warning	The warning is a large blue box that must appear in the middle of the screen.