CS2310 FINAL PROJECT REPORT: AYUSH MISHRA(amm428@pitt.edu)

Table of Contents:

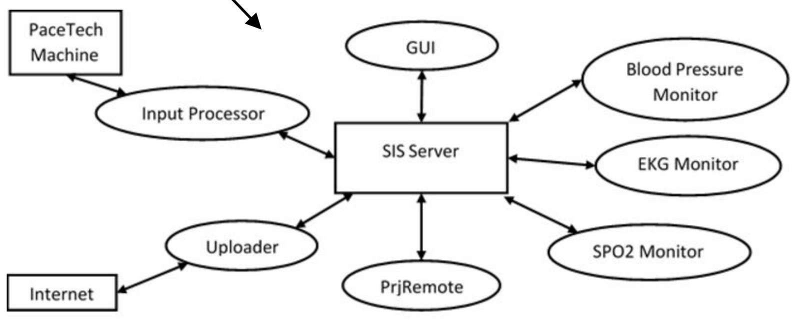
Introduction:

This project has been implemented on the basis of the healthcare system which makes use of existing peripherals to bring to the patients, “Ease-of-Access” to Medical resources. This has been designed keeping in mind the elderly, who should be able to access resources such as Urgent Medical attention in case of an emergency. The basis of the idea is the “Quick-reaction” healthcare system that enables the User to achieve immediate response from a Hospital or any Medical institution.

The focus of the project has been on “Gesture Recognition” which is the stepping stone towards an efficient healthcare system. In this project, I have implemented reading of motion and gesture recognition using one of the most accessible peripheral devices – the webcam (can also be used with a PC camera). The webcam successfully reads a gesture and saves the corresponding result to a common repository on the local machine. The SIS Server accesses this common repository to generate further messages to other components in the healthcare system.

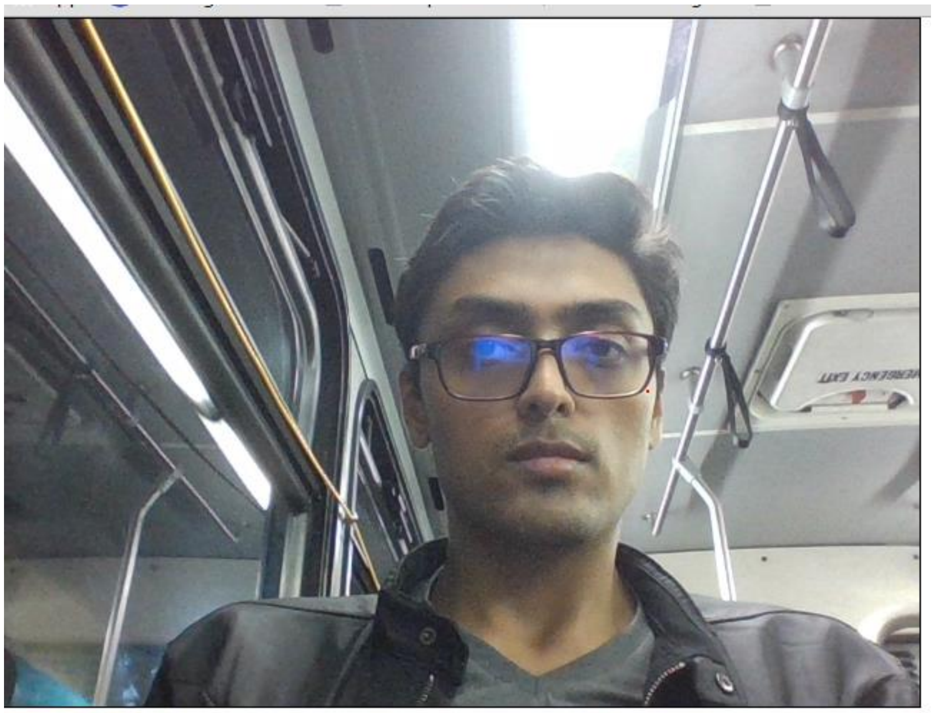
Flow:

Fortrek

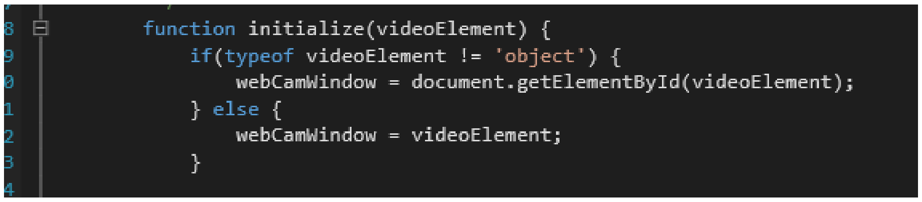

PaceTech Machine
GUI
Blood Pressure Monitor
Input Processor
SIS Server
EKG Monitor
Uploader
SPO2 Monitor
Internet
PrjRemote

Components:

i) Web Camera:

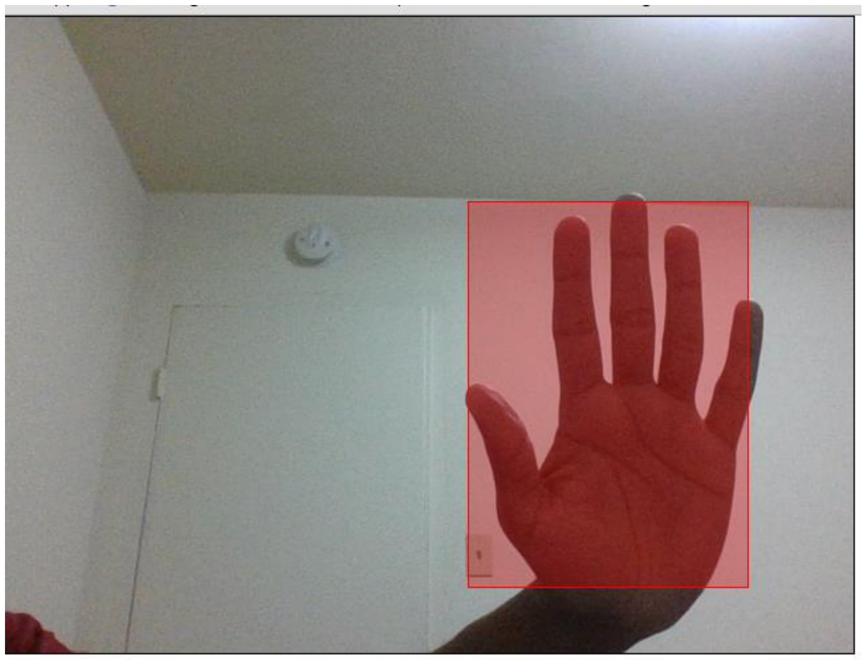


➢ The following code snippet enabled the webcam activation (in JavaScript):

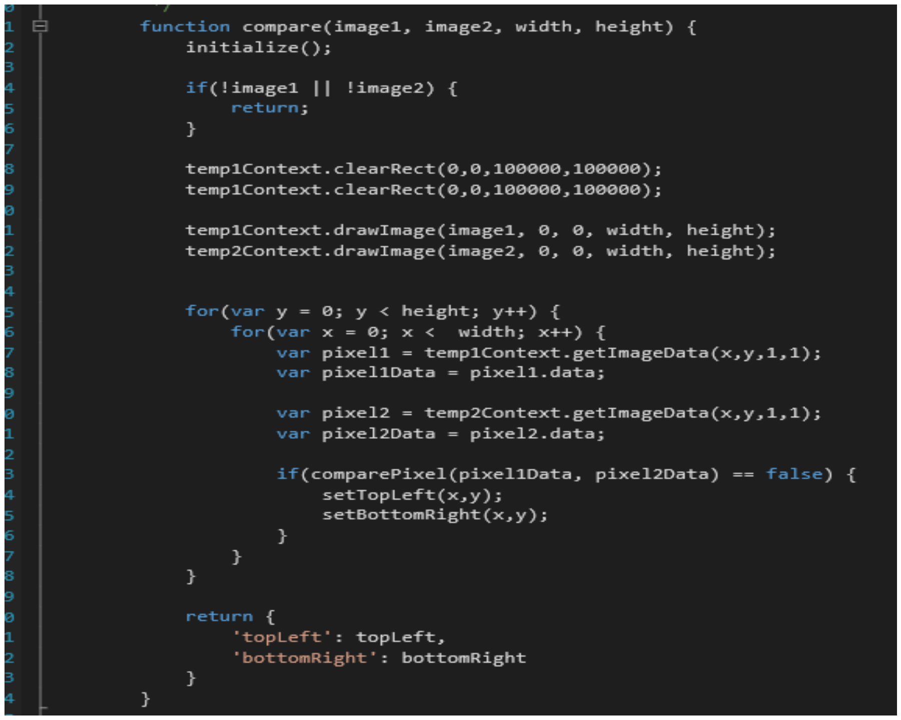


```
function initialize(videoElement) {
    if(typeof videoElement != 'object') {
        webCamWindow = document.getElementById(videoElement);
    } else {
        webCamWindow = videoElement;
    }
```

- The motion detection was implemented based on pixel comparison between the “currentframe” and the “oldframe”.
- The method of comparison implements a “credible” motion or movement on the screen if there is a difference of approximately 96% or more in the pixel values of the two current and old images. This is captured in the snapshot shown below:

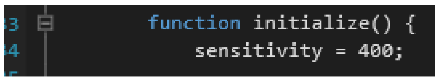


- The code snippet for enabling this motion detection is as shown below:



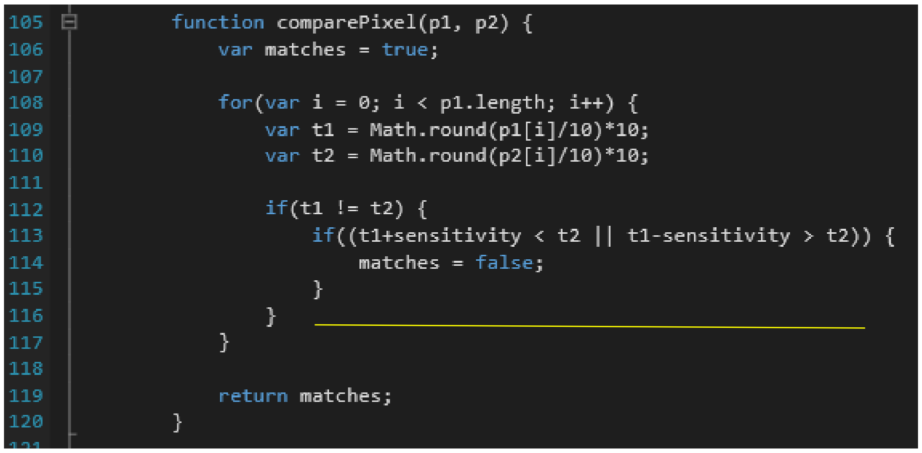
```
function compare(image1, image2, width, height) {
    initialize();

    if(!image1 || !image2) {
        return;
    }

    temp1Context.clearRect(0,0,100000,100000);
    temp1Context.clearRect(0,0,100000,100000);

    temp1Context.drawImage(image1, 0, 0, width, height);
    temp2Context.drawImage(image2, 0, 0, width, height);


    for(var y = 0; y < height; y++) {
        for(var x = 0; x <  width; x++) {
            var pixel1 = temp1Context.getImageData(x,y,1,1);
            var pixel1Data = pixel1.data;

            var pixel2 = temp2Context.getImageData(x,y,1,1);
            var pixel2Data = pixel2.data;

            if(comparePixel(pixel1Data, pixel2Data) == false) {
                setTopLeft(x,y);
                setBottomRight(x,y);
            }
        }
    }

    return {
        'topLeft': topLeft,
        'bottomRight': bottomRight
    }
}
```

- The pixel comparison between old and current image frames was done with a “sensitivity” counter.



```
function initialize() {
    sensitivity = 400;
```

➢ This was then compared with pixel data values:



```
        function comparePixel(p1, p2) {
            var matches = true;

            for(var i = 0; i < p1.length; i++) {
                var t1 = Math.round(p1[i]/10)*10;
                var t2 = Math.round(p2[i]/10)*10;

                if(t1 != t2) {
                    if((t1+sensitivity < t2 || t1-sensitivity > t2)) {
                        matches = false;
                    }
                }
            }

            return matches;
        }
```

➢ This worked well in detecting motion but the challenge however was in getting the system to detect "GESTURE". This was accomplished by creating a global variable "window.width_count" and a "window.wide" comparator variable. The corresponding gestures were recorded in the browser console to generate alerts on gesture recognition:

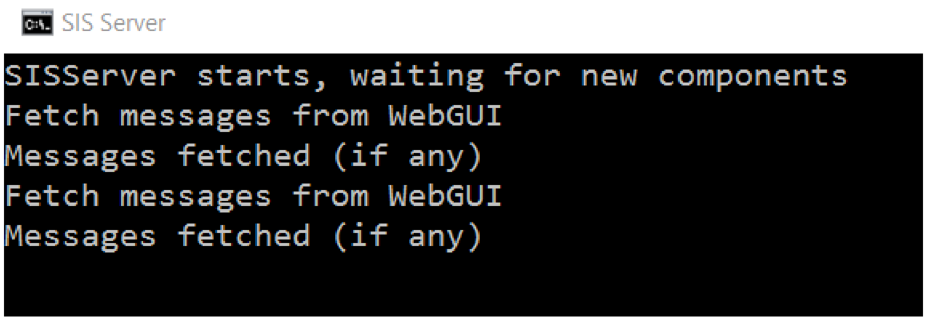
Youtube Link:

https://youtu.be/G7-omxKf8e4

➢ After achieving this, the next step was to activate the SIS server and other Components:
➢ The first step here was creating the "Register", "Connect", "Confirm" and "Alert" xml specification files for the GestureRecognition SuperComponent (included in the source code)
➢ After these, I created the following files for the super component:
    - CreateGesturerecognition.java
    - Helper.java
    - Initial.java
    - Pnml.xml

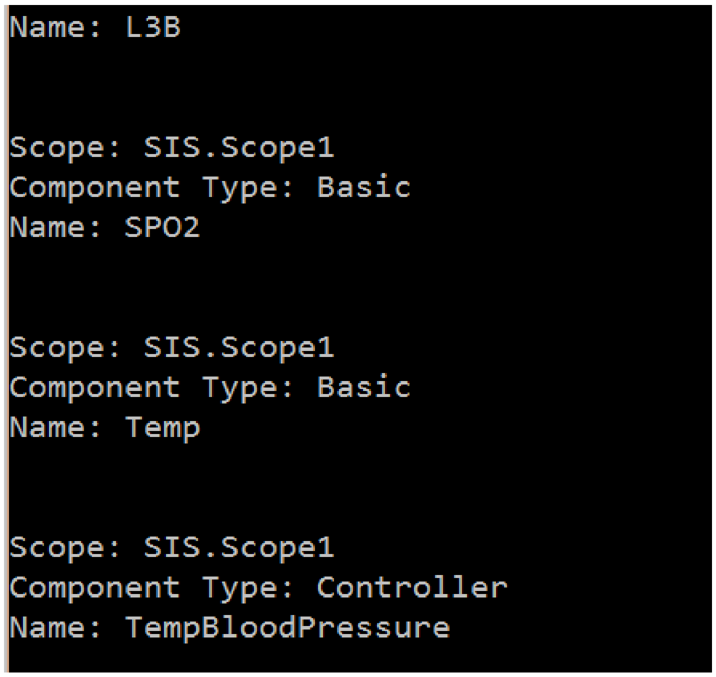
- For the runnable, I created a batch file runGestureRecognition which would establish the connection with the SIS server and would allow for the message transfer to and fro GestureRecognition component and a mock EmergencyMonitor Component (registered with PrjRemote)

- The screenshots are as follows:



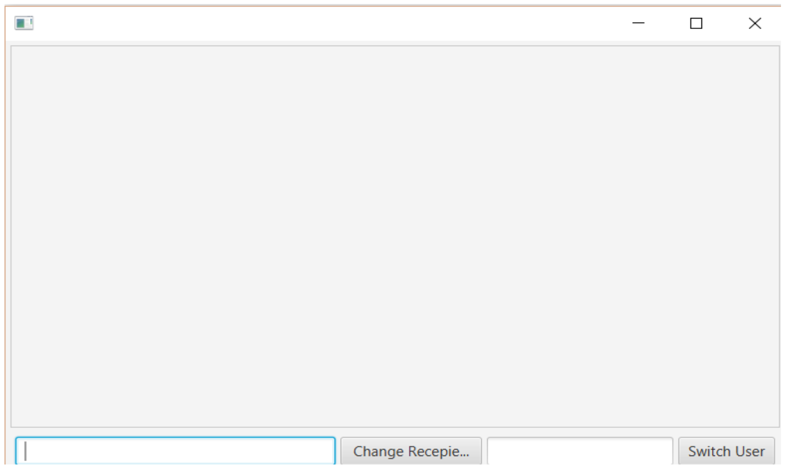
```
SISServer starts, waiting for new components
Fetch messages from WebGUI
Messages fetched (if any)
Fetch messages from WebGUI
Messages fetched (if any)
```

SIS server started

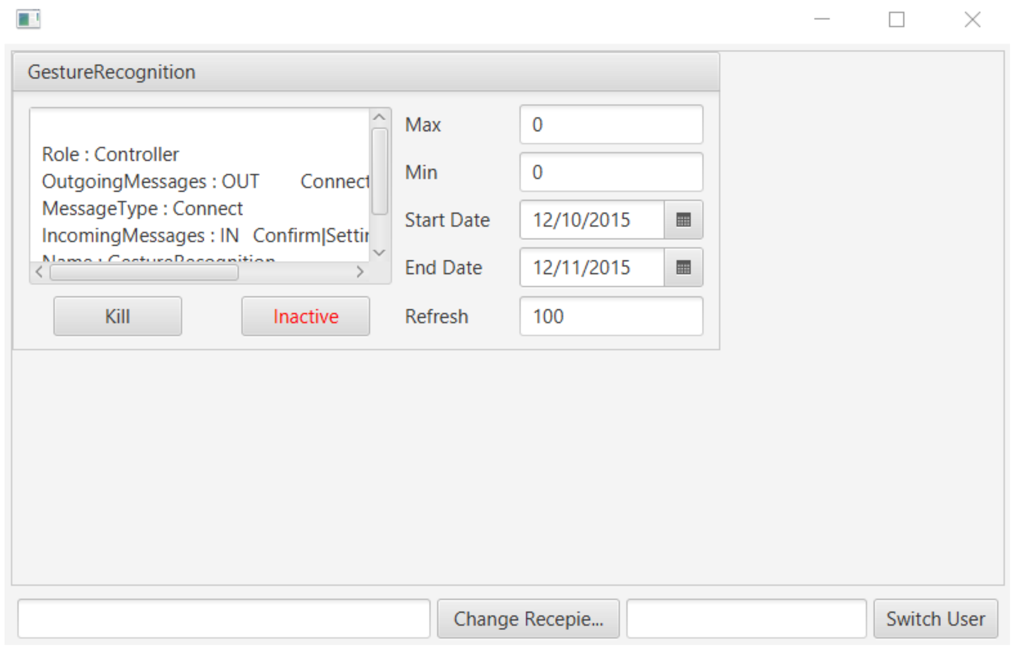


```
Name: L3B


Scope: SIS.Scope1
Component Type: Basic
Name: SPO2


Scope: SIS.Scope1
Component Type: Basic
Name: Temp


Scope: SIS.Scope1
Component Type: Controller
Name: TempBloodPressure
```

Components Initialized



GUI started



Role : Controller
OutgoingMessages : OUT Connect
MessageType : Connect
IncomingMessages : IN Confirm|Setti

Max	0
Min	0
Start Date	12/10/2015
End Date	12/11/2015
Refresh	100

Gesture Recognition initialized on the GUI successfully.

- The PrjRemote is used further to send messages from Gesture Recognition to Emergency Monitor.
- The alert generated by the gesture movement triggers the SIS component to initialize the Gesture Recognition GUI.

Conclusion:

The implementation of this project was limited to recognition of gestures which was successfully implemented and initializing of the SIS server to recognize the gesture and work towards sending it to the Emergency Monitor component which entails future work in the area. I would like to thank Professor Chang for guiding us through this wonderful journey and letting us explore so many interesting areas. I have gained invaluable insight into gesture and motion control due to my efforts in the project. I thank Professor Chang again for imparting his knowledge with us.