ESE-519: Real Time Embedded Systems
Project Fit-O-Philia: Report
Fan Zhang, Karthik Anantha Ram & Sanjeet Phatak

1. Project Title: Fit-O-Philia (Revolutionizing Custom Fitness solutions)

2. Team: Fan Zhang (UI Lead)
Karthik Anantha Ram (Software Lead)
Sanjeet Phatak (Embedded Lead)

3. Motivation:

The scramble to lose weight and get fit is on the rise among today's generation. As a result of which more and more people tend to join the gym. Gyms across the country have seen significant growth over the past decade, and demand doesn't appear to be slowing down. But a recent study by a popular website proves that 80% percent of the people who join gyms quit within 5 months of joining a gym. There are several reasons behind people quitting a gym, some tend to get lazy, most don't find time to workout and some are not happy with the results of their workouts at a gym. A trainer at a gym won't be able to cater to all the needs of the user working out, basically there is no one to provide feedback to the user on how he/she is performing their workout routine.

What are we trying to do here? We want to build an independent system that provides custom fitness solutions delivered through media content [primarily in the form of videos], receive feedback from the user on how he/she maintains posture during the workout through a set of interactive devices & dynamically adapt based on the user's input. We are trying to bridge the gap between time & working out. Such a system does not compel you to workout at a gym, one can use this system at home at their convenience.

4. Goal:

Developing & designing a set of interactive devices to capture the user's posture, analyse if the user is working out in the right manner and provide suitable feedback to the user. The goal is to build a closed system that incorporates the above mentioned features along with a Central Management System [CMS] that coordinates among the various components of the system and the user.
`

5. Methodology:

1. Develop a Web Application [Meteor JS] to deliver media content in the form of videos which guides the user during his/her workout [Content Management System (CoMS)].
2. Detect the posture of user working out during the various workout routines [this is done using a combination of four wearable interactive devices and one head gear, all of them are embedded with 9 DOF IMU's].

3. The data from these devices is sent over to the Central Management System [CMS] which performs statistical analysis to determine if the posture is right or wrong.
4. After the data has been analyzed and once the CMS has estimated if the posture is right or wrong, it communicates with the Animation module which generates a human stick figure. The stick figure would automatically simulate the user's current exercise. By running in the background, it indicates the user where in the posture did he/she go wrong.
5. The CMS also communicates with the user via the interactive devices and the CoMS as well. It sends in commands to these two modules to perform the necessary actions. For example, if the CMS predicts that the user is not working out and he/she has remained idle for a long time the video will automatically pause and once the user hits the play button on the wearable device the video will resume from the beginning again.

6. Architecture:

Develop a Web Application [Meteor JS] to deliver media content in the form of videos which guides the user during his/her workout [Content Management System (CoMS)].

6. Detect the posture of user working out during the various workout routines [this is done using a combination of four wearable interactive devices and one head gear, all of them are embedded with 9 DOF IMU's].
7. The data from these devices is sent over to the Central Management System [CMS] which performs statistical analysis to determine if the posture is right or wrong.
8. After the data has been analyzed and once the CMS has estimated if the posture is right or wrong, it communicates with the Animation module which generates a human stick figure. The stick figure would automatically simulate the user's current exercise. By running in the background, it indicates the user where in the posture did he/she go wrong.
9. The CMS also communicates with the user via the interactive devices and the CoMS as well. It sends in commands to these two modules to perform the necessary actions. For example, if the CMS predicts that the user is not working out and he/she has remained idle for a long time the video will automatically pause and once the user hits the play button on the wearable device the video will resume from the beginning again.

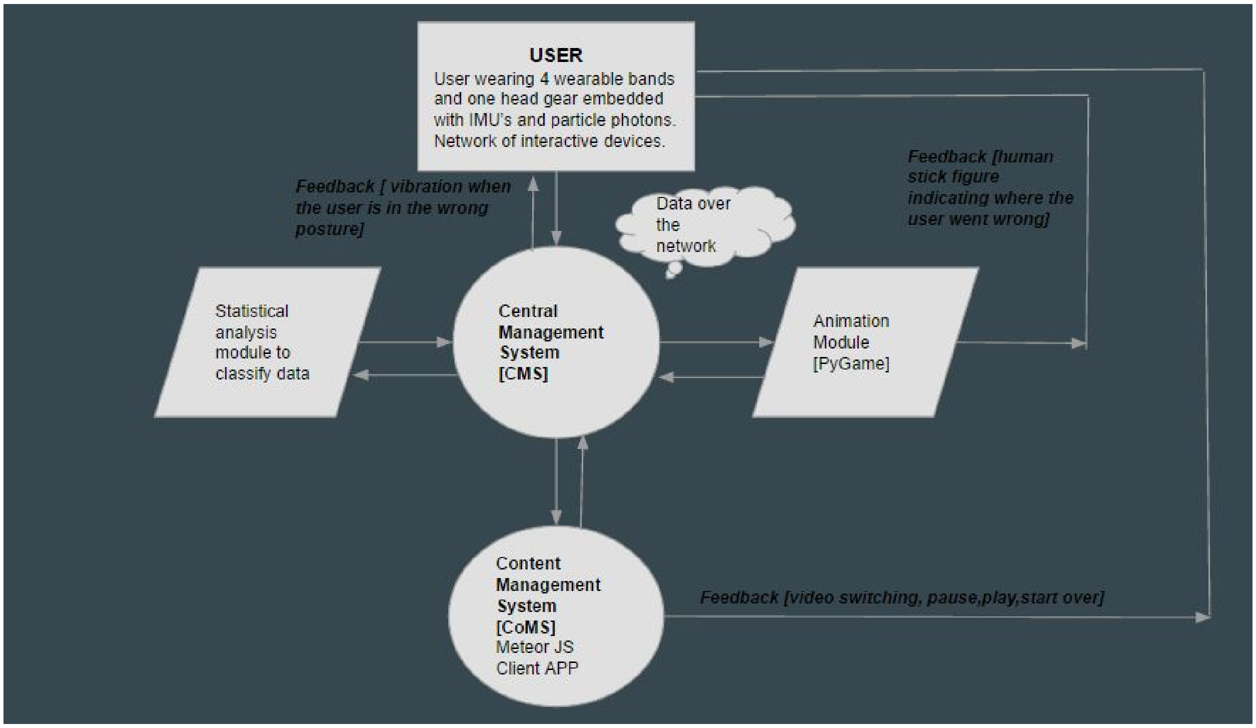

Figure 1:Fit-O-Philia [high level architecture]

- The CMS [main server] receives data from the interactive devices to process them.
- The CMS has two main modules one being the Animation module developed using PyGame that generates the stick figure and the other module is the one that performs the statistical analysis on the data to classify them.
- The CMS also talks to CoMS which is the Meteor JS web application that renders media content.
- From the above figure we can see the interactions between various modules of the system and how feedback is provided to the user from these individual modules in different forms.

7. Project Components:

Hardware: The main hardware component of the project are the five interactive devices. Four of them being wearable devices and one being the head gear. We designed five them in order to capture complex postures. We have one wearable device that goes on the wrist of each of the arms. One for each leg on the calf muscle, to detect motions involving the lower body. Each of these devices consists of a 9 DOF IMU, a particle photon & a power management circuit to power the devices. One of the bands [right arm -master device] we have developed a mechanism to control the videos being rendered by the CoMS by using a push button. We also have a vibration motor on the same band which is triggered when the user is not performing his/her workout in the right way to alert the user that they are doing it the wrong way.

Software: We build our web application the CoMS using Meteor JS. The meteor application has two components one is the server component that interacts with the external environment and the other is the client component of the applications that renders the front end. The CMS is a Python based multithreaded server which forms the heart of the system. The statistical module of the CMS is also developed in Python. The animation module that generates the human stick figure was developed using PyGame. It would generate stick figures according to the exercise that the user does, and indicate to the user where they are going wrong.

7.1 Hardware Effort:

- Sensor interfacing [9 DOF IMU with particle photon]
- Soldering
- Designing the band
- Placement of IMU and other components on the interactive devices.
- Programming the Photon

7.2 Software Effort:

- Roll & Pitch Estimation using IMU data
- I2C
- Wifi
- Web Sockets
- Multithreaded
- ISR-Interrupt Service Routine
- Python
- Meteor JS
- Node
- PyGame
- MatPlotLib
- AWS S3

8. Testing and Evaluation:

Testing:

- Tested the robustness of the devices by performing various workout routines wearing them.
- On testing our system we realized that we should not being making our system hard bound, we provided more leverage to the user in terms of how many errors he/she could make during the workout routine.
- Tested our statistical analysis model for the different exercises by performing them in different fashions to ensure that our model still gave a close estimation how the user is doing.
- Stress tested the Python Server [CMS] and the Meteor JS client APP [CoMS] by sending in concurrent requests and

Evaluation:

- Initially, we self tested the system in a closed environment

- We asked our fellow classmates to wear the devices and workout, and obtained feedback from them on whether they were comfortable wearing the device and performing the workout.
- Tweaked and implemented few modifications to the system based on user feedback and the feedback received on the baseline demo.

9. Project Deliverables:

Baseline goals:

1. Build a wearable wristband embedded with an IMU, power management circuit and a photon to acquire data of the user working out with a dumbbell.
2. The data from the IMU is wirelessly transmitted to a python server as UDP packets which performs the necessary computation. The server provides for live visualization of the acquired data
3. and simultaneously inserts the data into a Mongo database [pymongo].
4. Build a client based MeteorJS application that delivers video content [in this case workout videos using dumbbells] which the user follows during his/her workout routines. The app interacts with the server and depending on how the user is performing during the working smooth switching of videos takes place. We have also provided provision to play/pause the video using the wrist band.

Reach goals:

1. Expanding on the hardware front, incorporated four wearable bands and one head gear. There is one band each for one arm, one each for each leg and an head gear. These devices are embedded with IMU & Particle Photon.
2. Performing statistical analysis of data to estimate posture, improved over our algorithm from what was employed for baseline demo.
3. Incorporated more exercises compared to baseline. Currently the system supports 4 different exercises.
4. Providing feedback to the user on where the user is going wrong while working out. This is done by sending short vibration pulses to one of the bands to alert the user. Also a human stick figure which performs the workout with you will assist to indicate which part of the posture was wrong. [Ex. if the right arm of the user is in the wrong motion, the right arm of the stick figure will turn red]

10: Hardware Communication: Five photons were used totally four wearable devices and one head gear. Each photon continuously sends IMU data to the CMS on five different sockets in the form of roll and pitch. The photons are powered by 9v battery whose output is dropped to 3.3V using a Linear Regulator. The wristband on the right hand has a push button and a vibration motor. The user can use the push button to dynamically play and pause videos. The vibration motor vibrates whenever the user goes wrong in a certain exercise. This allows the user to look at the screen with the stick figure which will indicate the part he’s going wrong in.
Also the photon uses I2C communication to receive data from the IMU. This is done using the SDA and SCK pins.

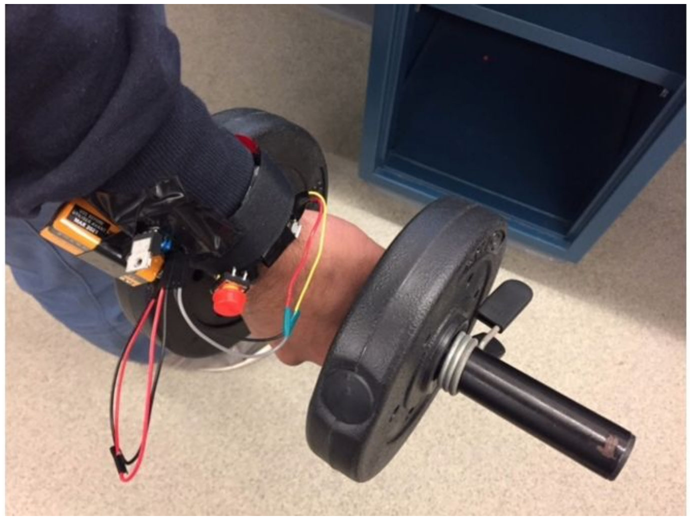

Figure2: Illustrates the push button on the right hand and the 9v battery connected through a 3.3v regulator to the photon.

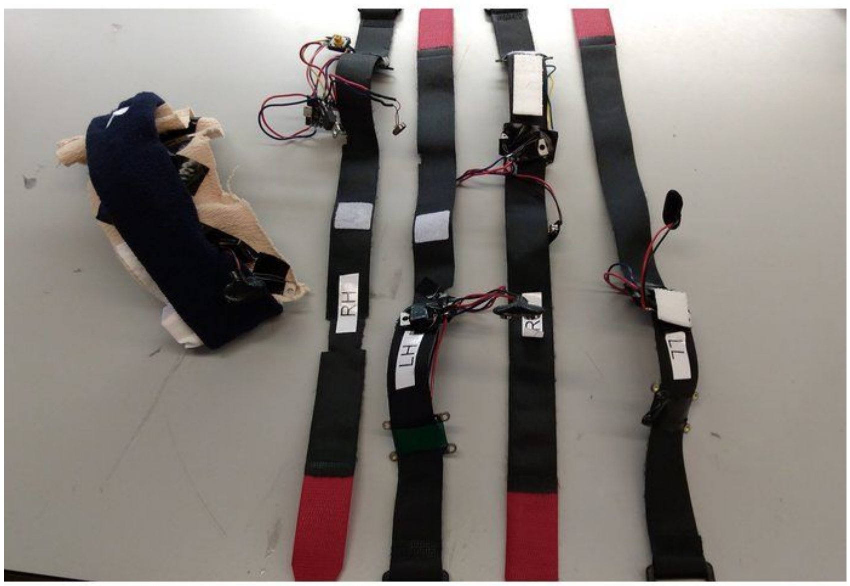

Figure 3: Illustrates the 4 band for the limbs and the head gear.

11. Central Management System[CMS]:

CMS is heart of our system that coordinates between the various modules of the system. CMS is a multithreaded application completely written in Python. CMS receives data from the network of interactive devices from the user, and analyzes that data. On the other end CMS also receives information from CoMS about which video is being played currently by the user. The data from the devices in combination with which video is being played is forwarded to the statistical analysis module that estimates if the posture of the user for that exercise is right or wrong. The statistical analysis module is run on an independent thread different from the thread that receives data from the devices. This ensure that the system does not clog and the computation time does not increase. The output of the statistical analysis is then forwarded to the animation module which renders the human stick figure for that exercise. Based on where the user is going wrong that part of the stick figure turns red indicating that an error has occurred. For example if the user is performing bicep curls and he/she is not lifting their left arm in the right way, the CMS will send that command over to the animation module which will then make that arm on the figure turn red to indicate to the user that an error has occurred. Apart from the animation module the CMS also sends feedback computed from the data received to CMS as well as back to the user. If the user is not doing it the right way, CMS sends a trigger signal to device as a result of which a short vibration pulse is generated on the band to alert the user. CMS also sends commands to CoMS to either pause/resume/resume from beginning/repeat video after a small break/ switch to a next video etc. [see figure 1 for better understanding]

12. Statistical Analysis Module: A pattern matching technique was used to analyze the dataset. Data samples from 6 people were collected and readings were taken from all the IMUs in different postures while doing the 4 exercises we covered. Since the IMU was well calibrated, we normalized the data such that values for IMU into three to six sections depending on the body part. For example- for right hand: bottom to middle, top to middle, top left to top right, bottom left to bottom right, top right to top left and bottom right to bottom left.
Whenever a video started playing, CMS was notified the name of the video by CoMS
Since we were handling streaming data, applying Hidden Markov Model on the data received from 5 IMUs in real-time became a difficult task. So we decided to go with a simpler approach. Data coming from the IMU was chunked every 2 seconds while performing the activity and prediction was done on whether the sequence of events for that particular exercise is correct - Pattern Matching. If the number of times the user goes wrong exceeds 5, the video would start over after a small 10 second interval(to give the user some time to catch his breath).

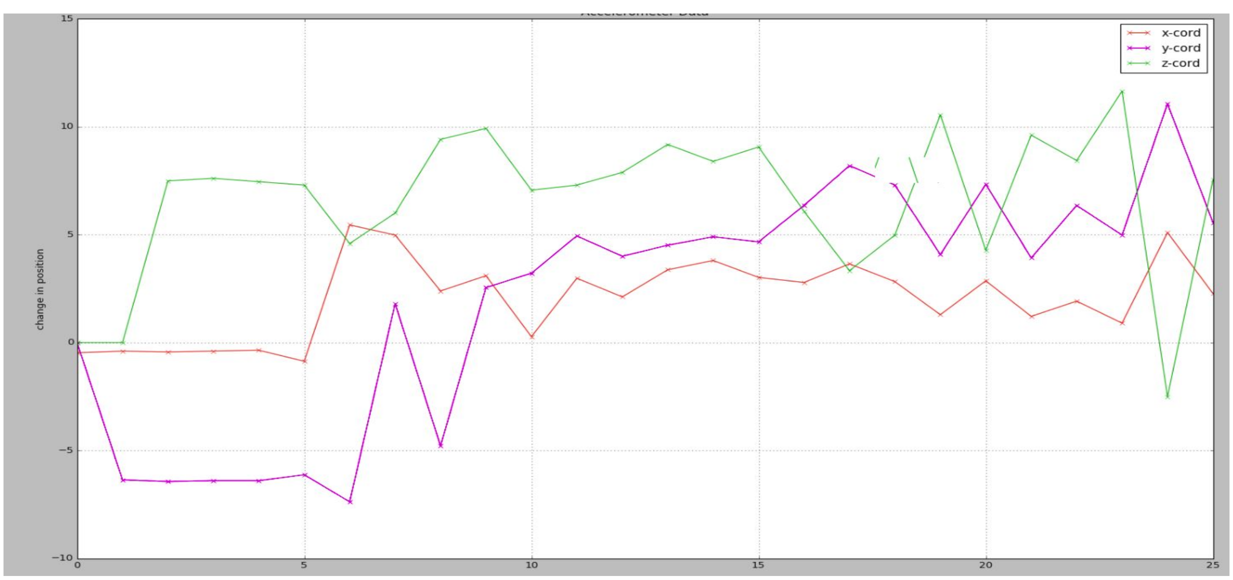

Figure 4: Data from the devices

13. Content Management System [CoMS]:

The Content Management System [CoMS], as the name suggests delivers media content in the form of videos that the user follows during his/her workout. The video library consists of videos that guide the user from a warm-up video, double arm bicep curl, alternate lunges & a cool down exercise. All the videos were hosted on Amazon AWS S3. The CoMS framework was built using Meteor JS. The server side of the Meteor APP communicates with the rest of the environment in this case the Content Management System [CMS]. The client side of the Meteor APP renders the front end interface. The Meteor APP constantly updates the CMS on it's status. For Example, when the user plays a video on the portal CoMS sends a message to CMS indicating that a video is being played. The Meteor APP then constantly polls every second to receive commands from the CMS on whether the user is performing the workout in the right manner. Based on the command from the CMS the CoMS can take multiple actions like pause the video if the user is idle for long time, repeat the same video after a 10 second break if the user has not performed well during that video, if the user does it right it takes him/her to the next video etc. The CoMS also provide provisions for the user to control the video player using his/her wearable wrist band. The user can play/pause the video from the device itself through a push button.

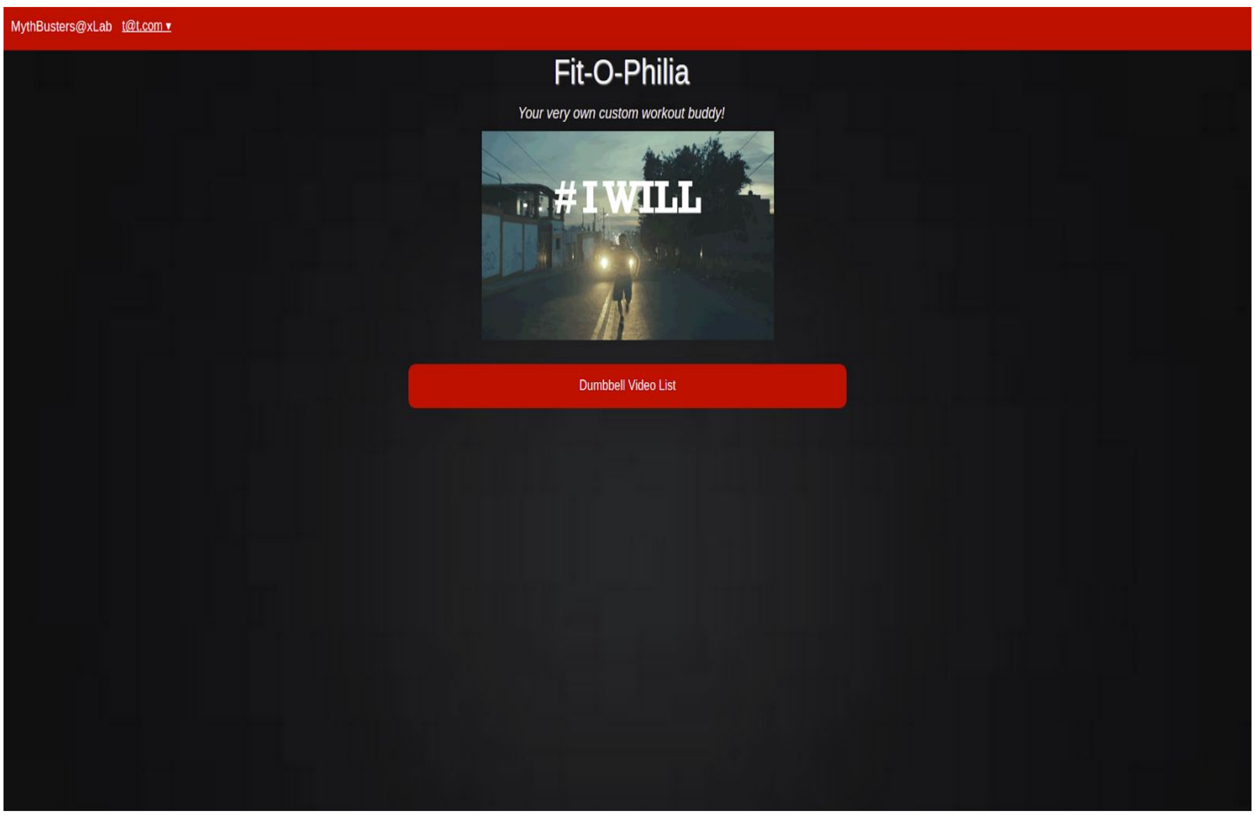

Figure 5:Home Page of the Meteor Application

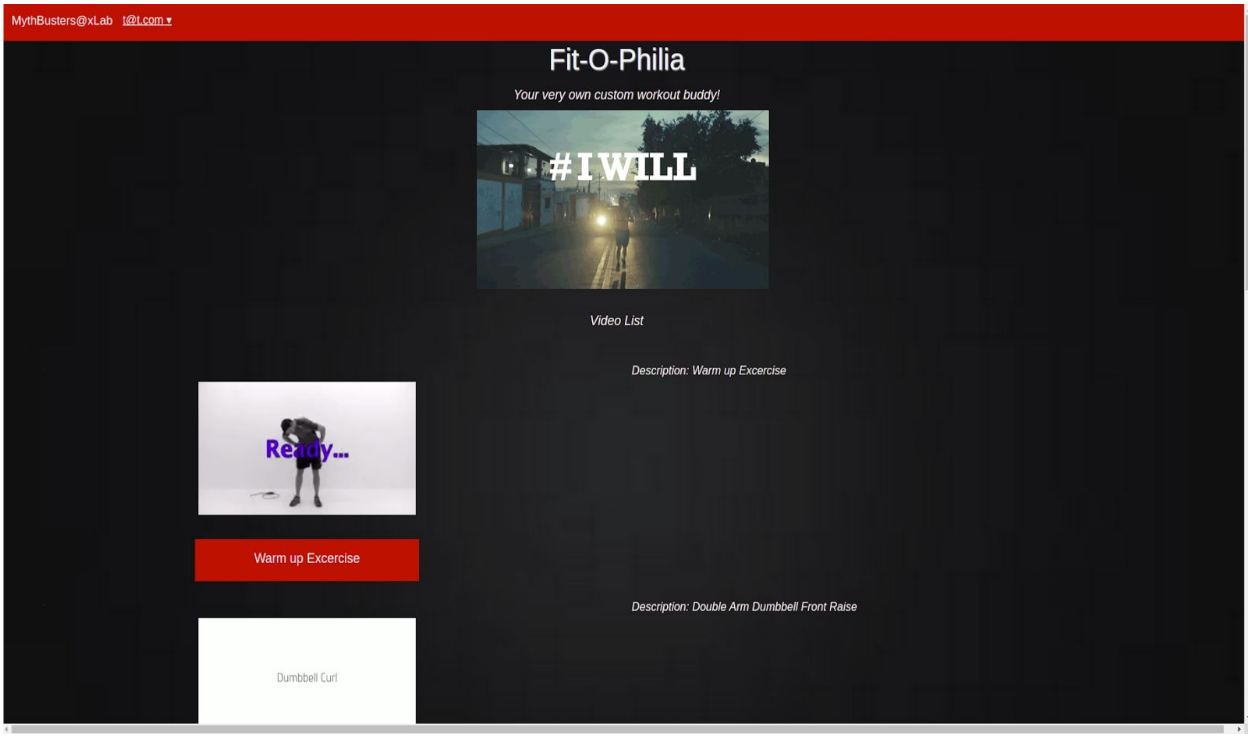

Figure 6: Video List.

14. User Interface - Animation Module:

Different from the previous modules which are mostly behind the scene, the animation module runs the main interface that the users would interact with during their exercise. When the user starts a video, the animation module receives message from the server and starts rendering a stick figure corresponding to current exercise. This module is realized by PyGame. The stick figure would indicate user where he/she is doing wrong by changing the corresponding body part to red. To properly simulate all the possible exercise, we draw each posture frame by frame in PyGame - like motion pictures. In total we implemented more than 30 frames to keep every posture animation running smoothly. We made the module multi-threaded in order to realize real-time color and posture change. The main thread would keep the animation running according to the flag received from the server indicating what exercise the user is currently doing. Meanwhile, other threads would also be running in background listening to the feedback from the server indicating whether the user is doing right or wrong, and whether it's time to stop the animation and switch to another exercise. The color variables are declared global in the Python file so that all the threads have access to them. In this way, colors could be changed as soon as there is any error detected by the server. At the end of the animation, all colors are reset to default in preparation for the next exercise. [see figure 2 and 3 for example]

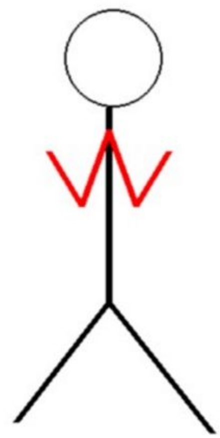

Figure 7:Sample stick figure [animation for bicep curl, red indicates that user is doing it wrong for both arms]

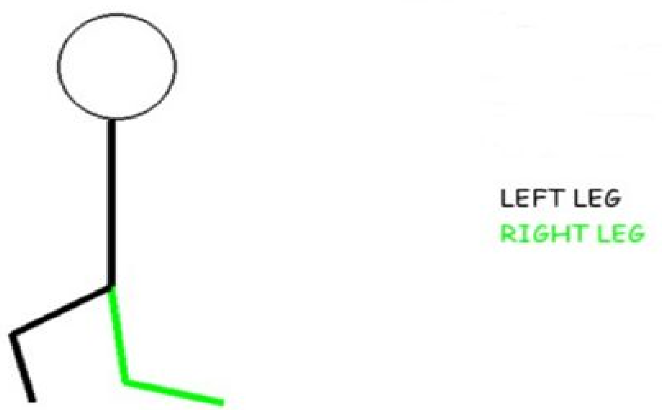

Figure 8:Sample stick figure [animation for alternative lunge]

15. Links:

Github: https://github.com/akarthik1991/Xlab-Cloud
Devpost: https://devpost.com/software/cloud-based-dumbbell-activity-tracker

16. Conclusion:

There are many websites that provide custom fitness solutions delivered through video content. But not a lot of systems provide feedback to the user on how he/she is performing the workout. We tried to bridge this gap by building a closed loop system where in there is interaction between the user and our system back & forth at all times. There is always scope for improvement and we can make our system better by improving the functionality of our different modules. One such area would the analysis performed on the data received from the devices, we are currently doing a statistical analysis on it which may not be that accurate when the device is used by multiple people of different physical builds. A better approach would be to collect data from a variety of people, chunk the data and apply a machine learning model such as HMM on it. Also building a Mobile App that serves as prognosis for the user would be a great addition to the project.