

A PROJECT REPORT
ON
“FITNESS MONITORING USING MACHINE
LEARNING”

Submitted to
UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR’S DEGREE IN
COMPUTER ENGINEERING

BY

SHAIKH SHAFIQUE NAUSHAD SITARA	16CO15
SHAIKH ALTAMAS SHAKEEL SHABANA	16CO11
ULDE FAHMI NISAR SEEMA	16CO17

UNDER THE GUIDANCE OF
PROF. MUBASHIR KHAN



DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam’s Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY

Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2019-2020

AFFILIATED TO
UNIVERSITY OF MUMBAI

**A PROJECT II REPORT
ON**

“FITNESS MONITORING USING MACHINE LEARNING”

**Submitted to
UNIVERSITY OF MUMBAI**

In Partial Fulfilment of the Requirement for the Award of

**BACHELOR’S DEGREE IN
COMPUTER ENGINEERING**

BY

**SHAIKH SHAFIQUE NAUSHAD SITARA 16CO15
SHAIKH ALTAMAS SHAKEEL SHABANA 16CO11
ULDE FAHMI NISAR SEEMA 16CO17**

**UNDER THE GUIDANCE OF
PROF. MUBASHIR KHAN**



**DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206**

**2019-2020
AFFILIATED TO**



UNIVERSITY OF MUMBAI

Anjuman-i-Islam's Kalsekar Technical Campus

Department of Computer Engineering

SCHOOL OF ENGINEERING & TECHNOLOGY

Plot No. 2 3, Sector - 16, Near Thana Naka,

Khandagaon, New Panvel - 410206



CERTIFICATE

This is certify that the project entitled

“FITNESS MONITORING USING MACHINE LEARNING“

submitted by

SHAIKH SHAFIQUE NAUSHAD SITARA	16CO15
SHAIKH ALTAMAS SHAKEEL SHABANA	16CO11
ULDE FAHMI NISAR SEEMA	16CO17

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Engineering) at *Anjuman-I-Islam's Kalsekar Technical Campus, Navi Mumbai* under the University of MUMBAI. This work is done during year 2019-2020, under our guidance.

Date: / /

(Prof. MUBASHIR KHAN)
Project Supervisor

(Prof. KALPANA R. BODKE)
Project Coordinator

(Prof. TABREZ KHAN)
HOD, Computer Department

DR. ABDUL RAZAK HONNUTAGI
Director

External Examiner

Acknowledgements

We would like to take the opportunity to express our sincere thanks to our guide **MUBASHIR KHAN**, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for his invaluable support and guidance throughout our project research work. Without his kind guidance & support this was not possible.

We are grateful to him for his timely feedback which helped us track and schedule the process effectively. His time, ideas and encouragement that he gave helped us to complete our project efficiently.

We would like to express deepest appreciation towards **DR. ABDUL RAZAK HONNUTAGI**, Director, AIKTC, Navi Mumbai, **Prof. TABREZ KHAN**, Head of Department of Computer Engineering and **Prof. KALPANA R. BODKE**, Project Coordinator whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped US directly or indirectly during this course of work.

SHAIKH SHAFIQUE NAUSHAD SITARA
SHAIKH ALTAMAS SHAKEEL SHABANA
ULDE FAHMI NISAR SEEMA

Project I Approval for Bachelor of Engineering

This project entitled *Fitness Monitoring Using Machine Learning* by *Shaikh Shafaque Naushad Shabana, Shaikh Altamas Shakeel Shabana and Ulde Fahmi Nisar Seema* is approved for the degree of *Bachelor of Engineering in Department of Computer Engineering*.

Examiners

1.

2.

Supervisors

1.

2.

Chairman

.....

Declaration

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Shaikh Shafaque Naushad Sitara
Roll Number: 16CO15

Shaikh Altamas Shakeel Shabana
Roll Number: 16CO11

Ulde Fahmi Nisar Seema
Roll Number: 16CO17

ABSTRACT

The enhanced performance of modern Artificial Intelligence algorithms has opened up limitless possibilities in the development of smart systems and devices. One of the most complex tasks for interactive devices is the analysis of human motion. However, using neural networks, movement can be classified and even understood. Our system proposes the use of an enhanced Pose Estimation algorithm[8] and Alternating Least Square(ALS)[4] for developing an application that functions as a personal exercise trainer. As the problem of staying healthy & fit is important as well as considering the use of smartphones, an easy-to-use application is a great solution.

This report provide details of this application that offers a full workout program, monitors the user's workout, and alerts the user when the move is performed incorrectly. Pose estimation detects human figures in images and videos[8].The algorithm is simply estimating where key body joints are and then joining them to know human pose. ALS is Alternating Least Square method for recommendation that we are using in diet and exercise recommendation system after categorising the user[4].

Keywords: BMI Calculation, Diet Recommendation, Exercise Recommendation, ALS, Pose estimation, Machine learning, Body detection, Schedule

Contents

Acknowledgement	iii
Project I Approval for Bachelor of Engineering	iv
Declaration	v
Abstract	vi
Table of Contents	ix
1 Introduction	2
1.1 Purpose	2
1.2 Project Scope	3
1.3 Project Goals and Objectives	3
1.3.1 Goals	3
1.3.2 Objectives	3
1.4 Organization of Report	3
2 Literature Survey	5
2.1 m-Health : A Complete Health Care Solution	5
2.1.1 Advantages of Paper	5
2.1.2 Disadvantages of Paper	5
2.1.3 How to overcome the problems mentioned in Paper	5
2.2 Android Based Nutritional Intake Tracking Application for Hand-held Systems	6
2.2.1 Advantages of Paper	6
2.2.2 Disadvantages of Paper	6
2.2.3 How to overcome the problems mentioned in Paper	6
2.3 SmartCoach Personal Gym Trainer	7
2.3.1 Advantages of Paper	7
2.3.2 Disadvantages of Paper	7
2.3.3 How to overcome the problems mentioned in Paper	7
2.4 Android-based Exercise Application	7
2.4.1 Advantages of Paper	8
2.4.2 Disadvantages of Paper	8
2.4.3 How to overcome the problems mentioned in Paper	8
2.5 Technical Review	8
2.5.1 Advantages of Technology	8

2.5.2	Reasons to use this Technology	8
3	Project Planning	9
3.1	Members and Capabilities	9
3.2	Roles and Responsibilities	9
3.3	Assumptions and Constraints	9
3.4	Project Management Approach	9
3.5	Ground Rules for the Project	10
3.6	Project Budget	10
3.7	Project Timeline	10
4	Software Requirements Specification	13
4.1	Overall Description	13
4.1.1	Product Perspective	13
4.1.2	Product Features	13
4.1.3	User Classes and Characteristics	13
4.1.4	Operating Environment	13
4.1.5	Design and Implementation Constraints	14
4.2	System Features	14
4.2.1	System Feature	14
4.2.2	System Feature	15
4.3	External Interface Requirements	15
4.3.1	User Interfaces	15
4.3.2	Hardware Interfaces	16
4.3.3	Software Interfaces	16
4.3.4	Communications Interfaces	16
4.4	Nonfunctional Requirements	16
4.4.1	Performance Requirements	16
4.4.2	Safety Requirements	16
4.4.3	Security Requirements	16
5	System Design	17
5.1	System Requirements Definition	17
5.1.1	Functional requirements	17
5.1.2	System requirements (non-functional requirements)	20
5.2	System Architecture Design	22
5.3	Sub-system Development	23
5.3.1	Registration and Login Module	23
5.3.2	Diet and Exercise Recommendation Module	23
5.3.3	Schedule Module	23
5.3.4	Pose Detection Module	24
5.4	Systems Integration	24

5.4.1	Class Diagram	25
5.4.2	Sequence Diagram	25
6	Implementation	29
6.1	Module 1	29
6.1.1	Pose Detection Module	29
6.1.2	app.py	29
6.1.3	home.html	32
6.1.4	trainer.html	36
6.1.5	homepage.css	38
6.2	Module 2	40
6.2.1	Diet Module	40
6.2.2	AndroidManifest.xml	40
6.2.3	LoginActivity.java	41
6.2.4	MainActivity.java	44
6.2.5	DietFragment.java	47
7	System Testing	51
7.1	Test Cases and Test Results	51
7.2	Sample of a Test Case	51
7.2.1	Software Quality Attributes	52
8	Screenshots of Project	53
8.1	pose estimation module	53
8.2	Diet module	56
9	Conclusion and Future Scope	66
9.1	Conclusion	66
9.2	Future Scope	67
	References	67
	Achievements	68

List of Figures

5.1	Usecase diagram	18
5.2	DFD Level 0 diagram	19
5.3	DFD Level 1 diagram	19
5.4	DFD Level 2 diagram	20
5.5	System Architecture of Fitness Monitoring using Machine Learning	22
5.6	Module Diagram for Diet and Exercise	23
5.7	Module Diagram for User Schedule	24
5.8	Module Diagram for Pose Detection	24
5.9	Class Diagram	25
5.10	Sequence diagram for exercise module	26
5.11	Sequence diagram for diet module	27
5.12	Sequence diagram for schedule module	28
6.1	Module Diagram for Pose Detection	29
6.2	Module Diagram for Diet and Exercise	40
8.1	Home page	53
8.2	Training page	54
8.3	Workout page	54
8.4	Body detection page	55
8.5	Performing Exercise page	55
8.6	Registration	56
8.7	Login	57
8.8	BMI calculation	58
8.9	categorising The User	59
8.10	Selecting Diet Plan	60
8.11	Selecting Diet Plan	61
8.12	Displaying Calories	62
8.13	Displaying calories	63
8.14	Displaying Diet Plan	64
8.15	Displaying Diet Plan	65

List of Tables

3.1	Table of Capabilities	9
3.2	Table of Responsibilities	9



Chapter 1

Introduction

In this era of fast paced life, a person is very busy in his/her work. They don't have time to talk to their loved ones and family, so taking care of their health and fitness is much far. A person's health is very important to make his/her work productive. Because of work load and busy schedule, it is very crucial to maintain a healthy life style and stay fit. Also, not many people can afford going to gym because of less time or money. In this modern world where smartphone has everything, there is a need of such an application that can monitor both health and fitness of an individual effectively. Our system is android based mobile application which provides: BMI (Body Mass Index) calculator, Suggestion of what an individual has to include or exclude from their diet, A full workout program according to exercise plan, Alert the user when any exercise move is performed incorrectly and daily/monthly exercise schedule.

1.1 Purpose

Identify the product whose software requirements are specified in this document. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem. Describe the different types of user that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers.[1] Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.

1.2 Project Scope

Our system is android based mobile application[6] which provides: BMI(Body Mass Index) calculator, Suggestion of what an individual has to include or exclude from their diet, A full workout program according to exercise plan, Alert the user when any exercise move is performed incorrectly and Daily/monthly exercise schedule.

1.3 Project Goals and Objectives

1.3.1 Goals

Provide a solution to the peoples who can't afford a gym membership but want themselves to be fit. Provide a platform for a people to who don't have time to go the gym due to their busy schedule. Making a full use of AI and new technologies to solve new generations problem.

1.3.2 Objectives

In this era of fast paced life , A person don't even get a chance to talk to their loved ones and family. this shows that how much they busy in their work.Because of work load and busy schedule they don't get a time to care about their health and fitness. It is very crucial to maintain a healthy life style and stay fit.In this modern world where smartphone has everything, there is a need of such an application that can monitor both health and fitness of an individual.So basically this application is for those who don't get a time to go the gym .They get a full training of exercise and diet without going to the gym.

1.4 Organization of Report

The report is organized as follows : The introduction is given in Chapter 1.It describes the fundamental terms used in this project.It describes the Goal,Objectives and scope of this project. The Chapter 2 describes the review of the relevant various techniques in the literature systems. It describes the pros and cons of each technique with how to overcome those cons using new technology

The project planning includes members and capabilities of this project ,roles and responsibilities of each member,Budget of Project and Project timeline is describe in Chapter 3. The Chapter 4 describes Functional and Nonfunctional Requirements of project.Along with this it also explain features of system and constraints of system.

The Chapter 5 includes Design Information with Class Diagram, Sequence Diagram , Component Diagram and System Architecture. Implementation of each module is explained in Chapter 6. Chapter 7 shows final Test Cases and Test Results. Chapter 8 includes Screenshot of outputs and Conclusion and Future Scope of Project is described in Chapter 9.



Chapter 2

Literature Survey

2.1 m-Health : A Complete Health Care Solution

This paper presents details of smart android application on how to solve health-care issues. It has Fitness module, Doctor-Patient module, Medicine reminder module, Emergency module and Blood-request module. It uses Android Studio, MySQL, Firebase and PHP technologies.[1]

2.1.1 Advantages of Paper

- a. It provides many useful features to users like fitness tracking.
- b. User can manage appointments with doctors easily.
- c. It gives reminder for medicine to user.
- d. This application provides help to users in emergency.

2.1.2 Disadvantages of Paper

- a. This application cannot track distance travelled, heart rate of user and calories consumed by user.
- b. No video conferencing between patient and doctor which leads to no help in finding best advise.
- c. In emergency, user has to click on panic button which is sometimes not possible and hence no help is provided.
- d. It does not provide exercise plans for user to lose or gain weight.

2.1.3 How to overcome the problems mentioned in Paper

- a. Implement a new algorithm which can track distance travelled, heart rate of user and calories consumed by user.

- b. Need to add a video conferencing feature in the application
- c. Implement a diet and exercise to lose weight plan based on the person's health

2.2 Android Based Nutritional Intake Tracking Application for Hand-held Systems

This paper presents details of fully functional Android Application, Health Buddy, for health management specifically targeting the young people in India. Features of this application includes Graphical representation, traffic light approach, health tips and suggestions and additional features like Body Mass Index (BMI) calculator, waist to hip ratio, food and exercise suggestion. It has meal-intake record module, nutritional intake tracker module, BMI, Waist to Hip ratio module, amazing facts module and health recommendation module.[2]

2.2.1 Advantages of Paper

- a. This app can indicate high levels of potassium and sodium in packaged food items.
- b. Food facts are displayed to make people aware and breaking different stereotypical opinions about certain foods.
- c. According to the severity in nutritional aspect, with the help of traffic light approach- red, amber and green colour marks are shown with statistical data. Here, red indicates high, amber indicates medium and green indicates low severity.

2.2.2 Disadvantages of Paper

- a. It is highly customized for Indian users keeping in mind the Indian servings.
- b. Larger audience cannot use this application.
- c. There is no location tracker for finding nearby restaurants or cafe to give proper guidance for nutrition.
- d. It only recommends the diet suitable for user and not provide complete diet plan.

2.2.3 How to overcome the problems mentioned in Paper

- a. Make it a budget friendly for ease of every user.
- b. Implement a google map API to access user's location
- c. Need to add a complete diet plan for a particular user

2.3 SmartCoach Personal Gym Trainer

This paper presents details of a device that functions as a personal exercise trainer which uses Adaptive Modified Back propagation (AMBP) algorithm. It includes information of wearable device which offers full-workout program, monitors the user's workout, count down the reps and alerts the user when the move is performed.[3]

2.3.1 Advantages of Paper

- a. It uses AMBP algorithm which is must faster than both Standard Back propagation (SBP) algorithm and Back propagation with Momentum (BPM) algorithm.
- b. Due to AMBP, lower Mean Square Error (MSE) is obtained in long run.
- c. It is easy to use.

2.3.2 Disadvantages of Paper

- a. The device is only related to exercise of user and does not have any diet plan recommendation.
- b. It is critical to reach the global minimum error for optimal performance since the system relies on speed and accuracy.

2.3.3 How to overcome the problems mentioned in Paper

- a. Need to implement a recommendation algorithm which gives diet plan as well as exercise plan .
- b. Improving the over all accuracy of the application by reducing the complexity
- c. Need to add the track meal feature

2.4 Android-based Exercise Application

This paper presents details of android-based mobile phone prototype which calculates and determines duration of physical exercise, time to exercise and the types of exercise needed daily by users. It includes user height and weight module, BMI calculation module, user body condition module, user working hours module and result generation module.[4]

2.4.1 Advantages of Paper

- a. It suggests suitable types of exercise.
- b. It also provides suitable time of exercise to users.
- c. Duration of exercises is also suggested to users.

2.4.2 Disadvantages of Paper

- a. It does not provide live detection of user doing exercise.
- b. It does not give a proper exercise posture for the user.

2.4.3 How to overcome the problems mentioned in Paper

- a. Need to implement an algorithm which provide live detection
- b. Make it more user friendly by using effective GUI
- c. Improving accuracy

2.5 Technical Review

pose estimation detect human figures in images and videos. The algorithm is simply estimating where key body joints are. The key points detected are indexed by "Part ID". Bottom up approaches involve first detecting the parts or joints for one or more humans in the image, and then assembling the parts together and associating them with a particular human. In simpler terms, the algorithm first predicts all body parts/joints present in the image.

2.5.1 Advantages of Technology

- a. Detecting a pose of a person becomes easy by using this algorithm.
- b. Light weight and easy technology to use
- c. Uses new technology like AI and Computer Vision

2.5.2 Reasons to use this Technology

- a. pose estimation detect human figures in a live videos.
- b. Involve first detecting the parts or joints for one or more humans.
- c. The algorithm first predicts all body parts/joints present in the image

Chapter 3

Project Planning

3.1 Members and Capabilities

Table 3.1: Table of Capabilities

SR. No	Name of Member	Capabilities
1	Shaikh Shafaque Naushad Sitara	Machine Learning, Backend Developer
2	Shaikh Altamas Shakeel Shabana	Database, UI Designer , Android Developer
3	Ulde Fahmi Nisar Seema	Android Developer

Work Breakdown Structure

3.2 Roles and Responsibilities

Table 3.2: Table of Responsibilities

SR. No	Name of Member	Role	Responsibilities
1	Shaikh Shafaque Naushad Sitara	Team Leader	ML Model , Flask Web App
2	Shaikh Altamas Shakeel Shabana	Team Member	Database, UI Design
3	Ulde Fahmi Nisar Seema	Team Member	Android App , Backend

3.3 Assumptions and Constraints

3.4 Project Management Approach

- a. Planning of project.
- b. Defining the scope of the project.
- c. Estimation of time and It's management.
- d. Creating Gantt Charts and properly assigning tasks to members.
- e. Reporting the progress of project with the guide.

3.5 Ground Rules for the Project

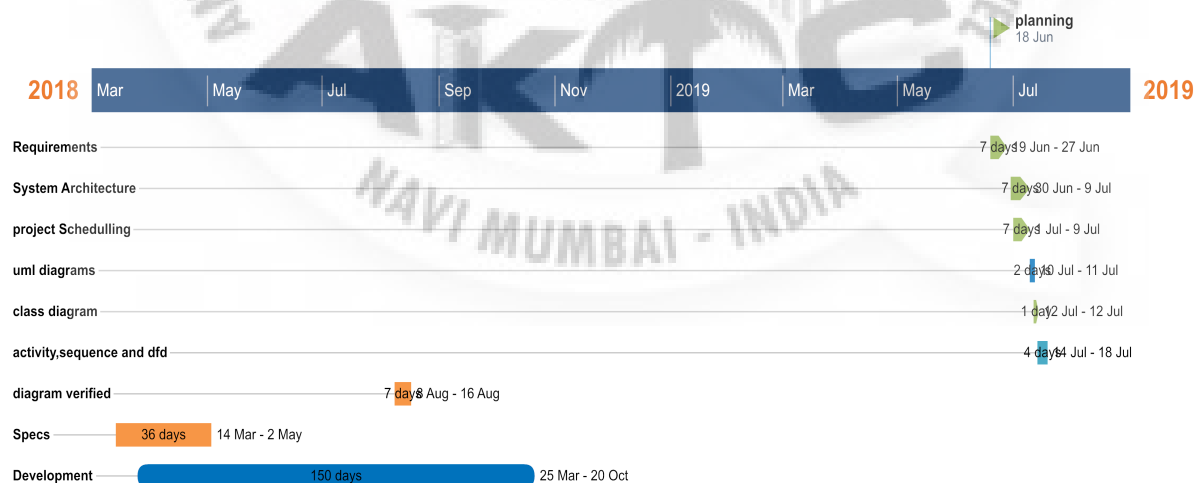
- Properly planning and gathering relevant information is very important.
- Developing a Blueprint of the project and work accordingly.
- All the members should report to the guide whenever required.
- Setting up small goals every week.
- Achieving the small goal within that span of time.
- Keeping tracks of the progress towards project.

3.6 Project Budget

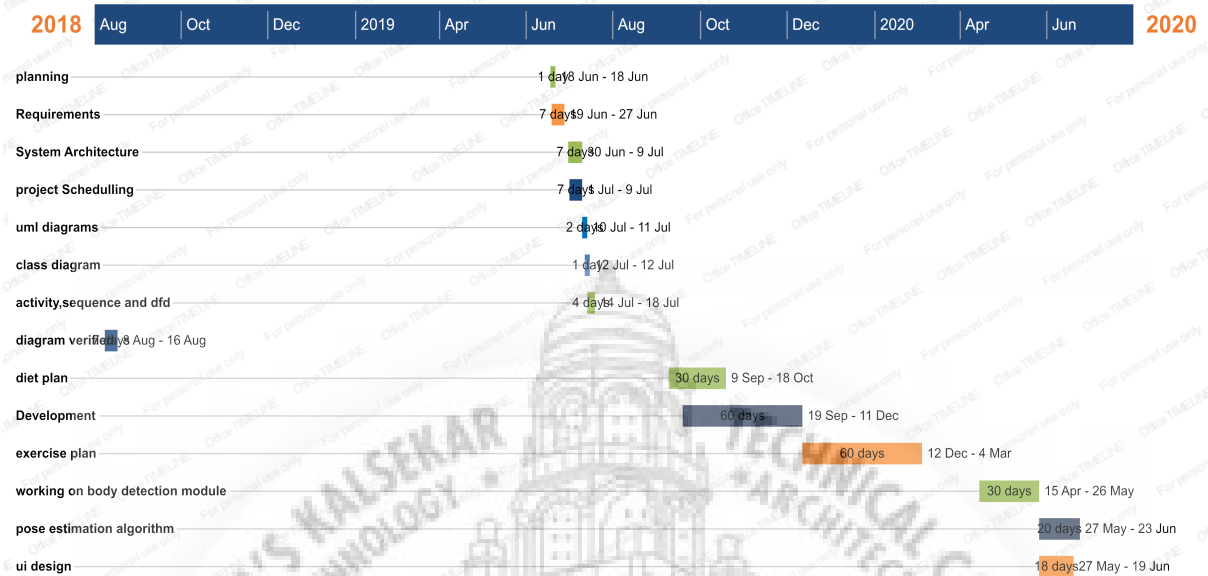
- It is a light project.
- Cost of the project is very low and efficient.

3.7 Project Timeline

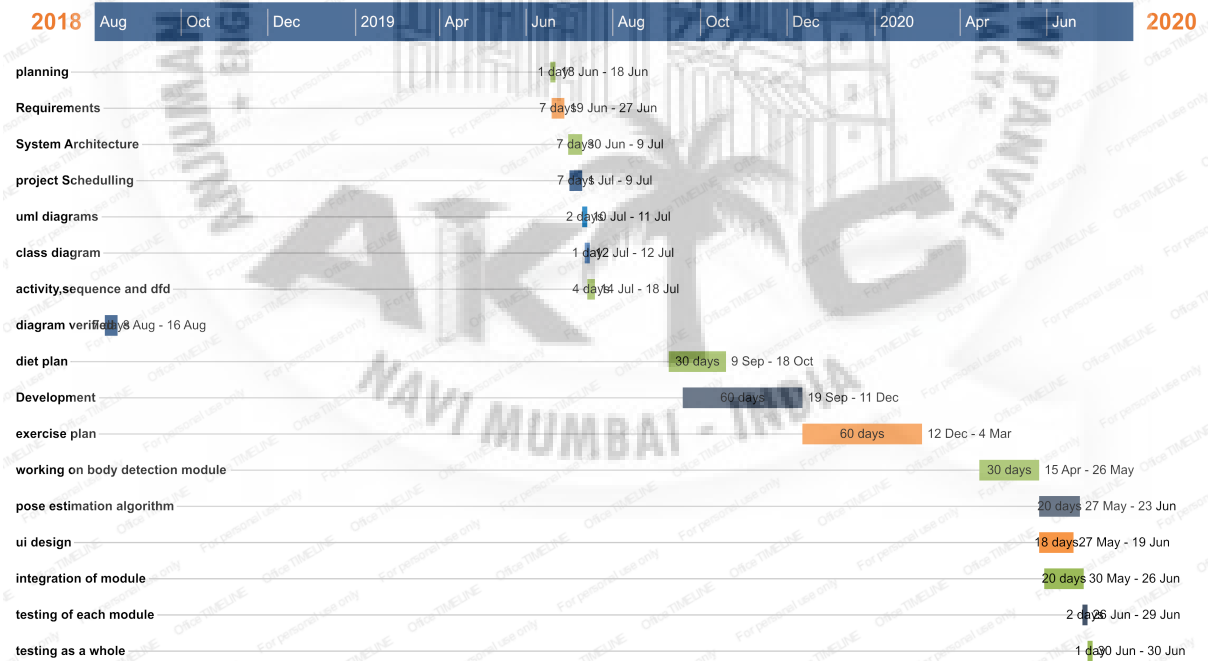
Fitness Monitoring



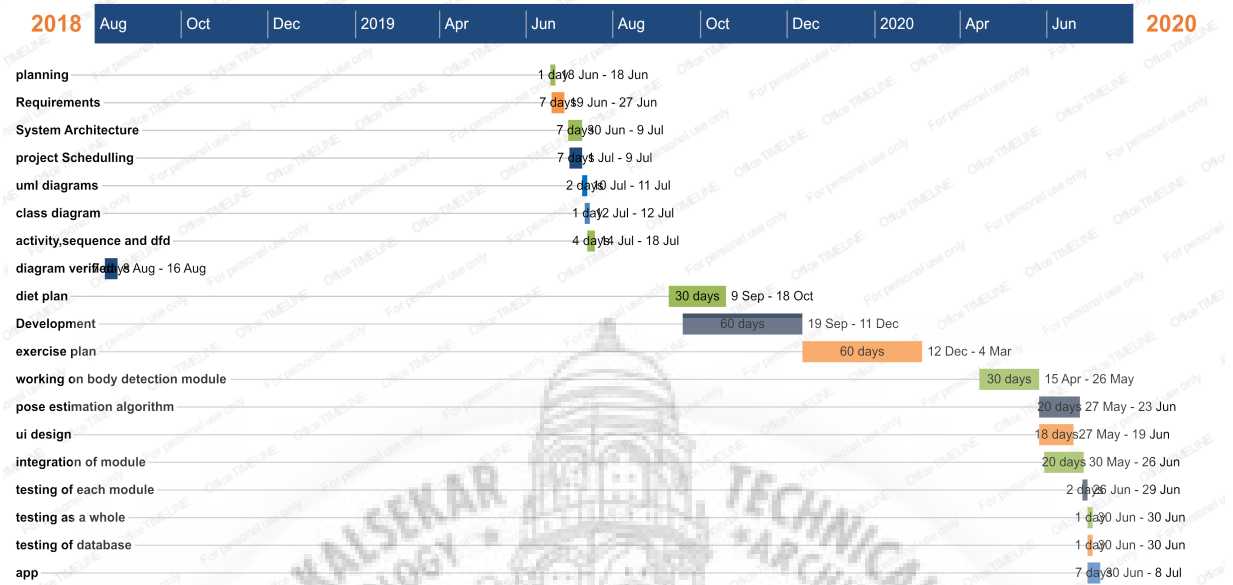
Fitness Monitoring



Fitness Monitoring



Fitness Monitoring



Chapter 4

Software Requirements Specification

4.1 Overall Description

4.1.1 Product Perspective

The product is an open source. It is a web-app based system implementing clientserver model. This web-app provide a service to the people who can't afford to go to the gym and those who dont have time for gym. This system is an independent from any other third party system. The main outcome is an audio which help to assist people i diet and exercise. In this application , people can calculate their BMI and based on that BMI a particular diet and exercise plan is given to the person and also monitor the exercise of a person

4.1.2 Product Features

Two main feature are there is in application .diet recommendation and exercise recommendation. In Diet recommendation , user get their diet plan based on their BMI value and start the diet plan as per their convinience . In exercise recommendation , user clicks on the particular suggested exercise and starts exercise and the real time feedback is given to the user.

4.1.3 User Classes and Characteristics

This is project is a social project. The Users of this system are mostly those who don't have time to go the gym or can't afford to go to the gym and those who love to be fit .

4.1.4 Operating Environment

- a. processor pentium4 or above.
- b. Ram:2GB or Above

c. Hard-disk:40GB or Above

Operating System:

a. Windows: 7 or above.

Software:

a. Android Studio

b. Flask : 1.1.2

c. Python: 3.6 or Above

d. OpenCv: 3.3.1 or Above

e. Web-browser:chrome,Mozilla,Internet Explorer

4.1.5 Design and Implementation Constraints

This system focuses one of the features at time. It is not able to provide two or more services at a time.At any instant only one of the services is accessible.

4.2 System Features

According to BMI it classify a user as three category which is underweight,normal and overweight.As per the classification it recommends diet and exercise on daily/-monthly basis for the particular category.After selection user starts the diet and exercise plan.User progress is shown in the form of graph for both diet and exercise.User has to mark manually the ate food items to see their diet progress.

4.2.1 System Feature

BMI Calculation

Description and Priority

This is of the main feature of the application in which a user give the their height and weight as an input and its return a users BMI , based on that BMI its decides a users category whether the user is normal , overweight or underweight .

Stimulus/Response Sequences

1. Stimulus : User logins.
2. Response : Home Page opens and BMI option pop ups.

3. stimulus : user enter their details like height and weight and click on the calculate BMI button.
4. Response : Calculated BMI is shown to the user.

Functional Requirements

REQ-1:Authorization from external camera module.

REQ-2:: Access to the Disk for saving the resources like image, audio.

4.2.2 System Feature

Diet and Exercise Recommendation

Description and Priority

This is of the main feature of the application in which a user got the diet and the exercise plan as per their category of BMI and user can start the given diet and exercise plan as per their convenience.

Stimulus/Response Sequences

1. Stimulus : User clicks on the particular exercise.
2. Response : Exercise option opens for the user and camera is started.
3. stimulus : User performs exercise in front of camera.
4. Response : A real time feed back is given the user to correct their mistakes while performing exercise.

Functional Requirements

REQ-1:Authorization from external camera module.

REQ-2:Access to database.

4.3 External Interface Requirements

4.3.1 User Interfaces

It is very light web-app, so the GUI is very simple.Home pages provides a sign and login option and a BMI option After calculating BMI new window opens up to a user which consist of a diet and exercise plan for a user.

4.3.2 Hardware Interfaces

This web-app requires permission of some of the hardware commodities. One need to give camera access in-order to capture the video. Microphone access should also given for the feedback purpose

4.3.3 Software Interfaces

This software uses different libraries. Opencv is used to deal with the computer vision part. ALS algorithm is used to recommend the diet and the exercise plan .OS support also needed to manage the audio features

4.3.4 Communications Interfaces

Since this is a light web-app, there is no such large communication in the system. Only Databases access, that also done locally. Also https standard is used in-order to gain the access the camera from the server.

4.4 Nonfunctional Requirements

4.4.1 Performance Requirements

Performance of overall system is very efficient and well optimize. From the time taken to capture and process it everything is well organized. While processing a real time feedback to the user

4.4.2 Safety Requirements

This system does not contain any critical data. Still it provide. The databases that are accessed are locally executed. In case of any updates in libraries used can lead to the failure in systems.

4.4.3 Security Requirements

Images of known persons are well compressed before storing. None of them are given access to database. All the libraries used are certified and standard. Also camera access is until the process is done completely. After that is released.

Chapter 5

System Design

5.1 System Requirements Definition

5.1.1 Functional requirements

They define the basic functions that the system must provide and focus on the needs and goals of the end users: 19

- a. Sign Up and Login : User first need to login to the app
- b. Calculate BMI : User must need to enter the height and weight details to get their BMI
- c. Diet Recommendation : Based on the BMI , user is recommended a particular diet plan .
- d. Exercise Plan : User need to select the particular recommended exercise plan and start the exercise .
- e. Real time Feedback : A real time feedback is given to the user as he/she performs the exercise.
- f. Correction : App corrects the user's mistake while performing exercise if they do some wrong steps.

Use-case Diagram

Use case diagram depicts how user is interacting with system.

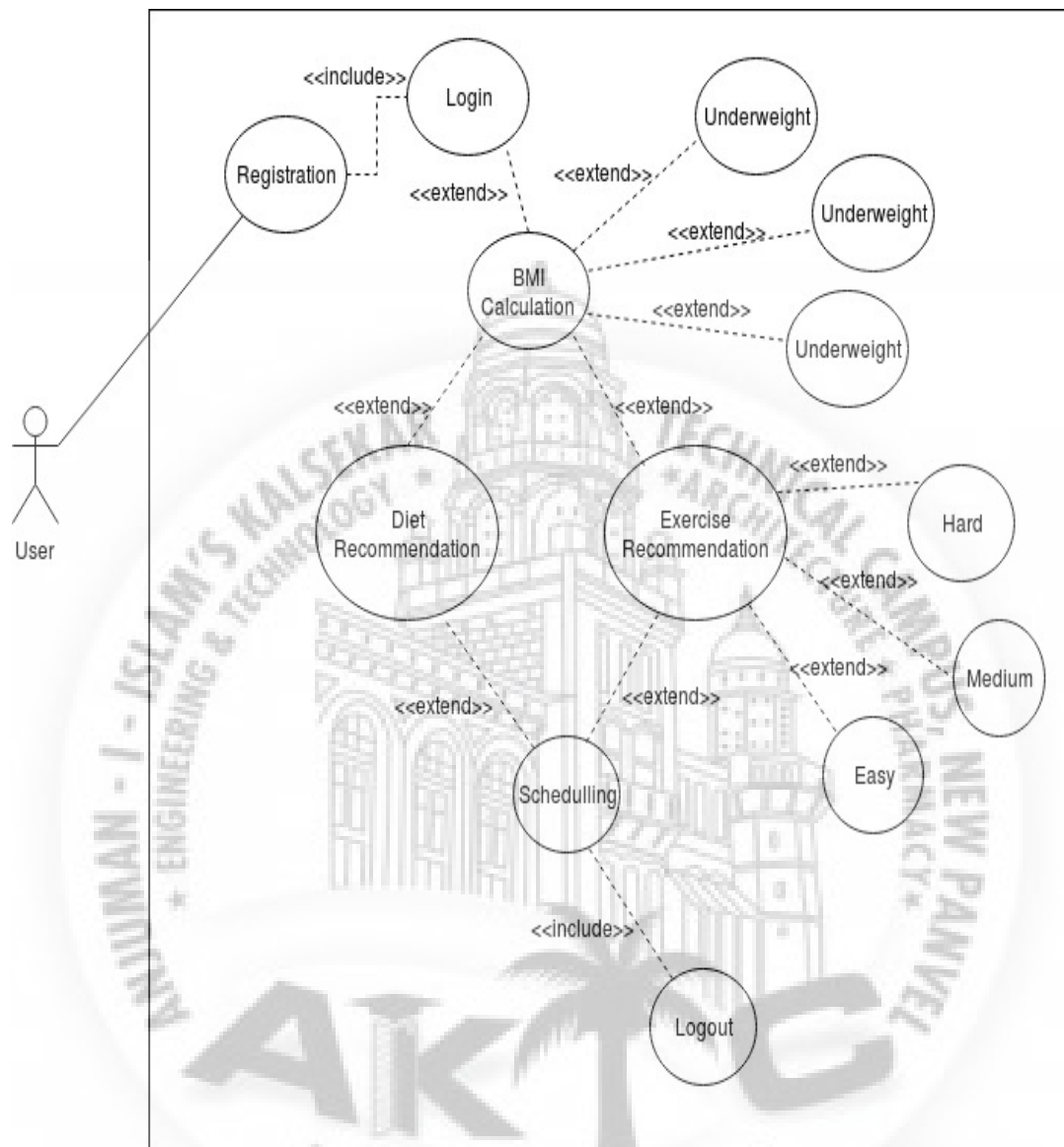


FIG-USE CASE DIAGRAM FOR FITNESS MONITORING

Figure 5.1: Usecase diagram

Data-flow Diagram

DFD Level 0 depicts main modules of the system.

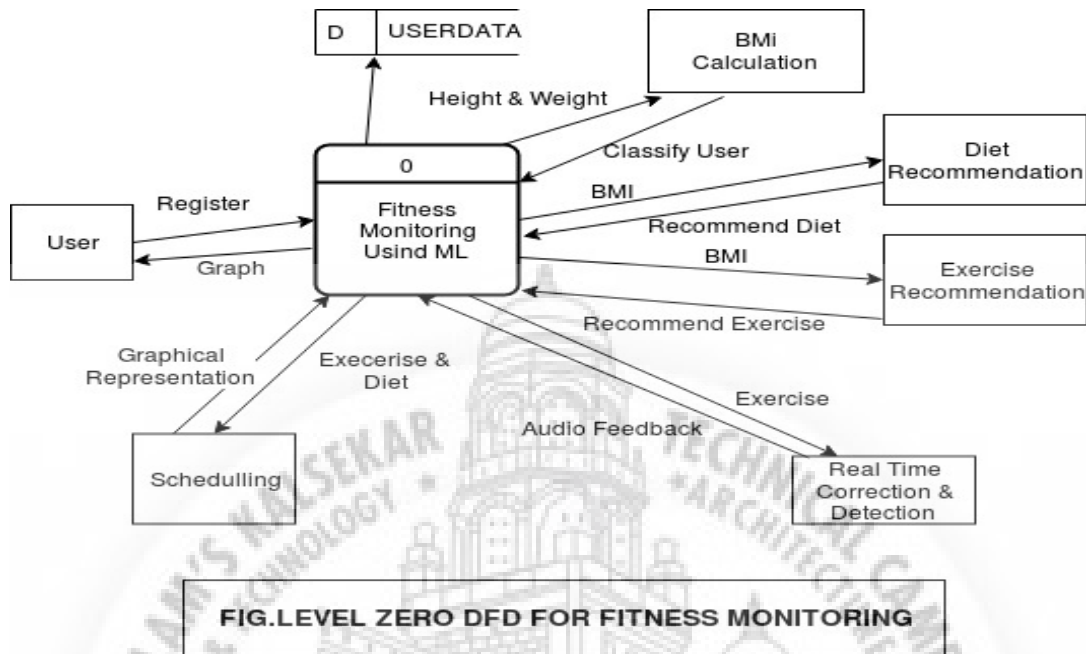


FIG.LEVEL ZERO DFD FOR FITNESS MONITORING

Figure 5.2: DFD Level 0 diagram

DFD Level 1 shows more details of the functions of each module of system.

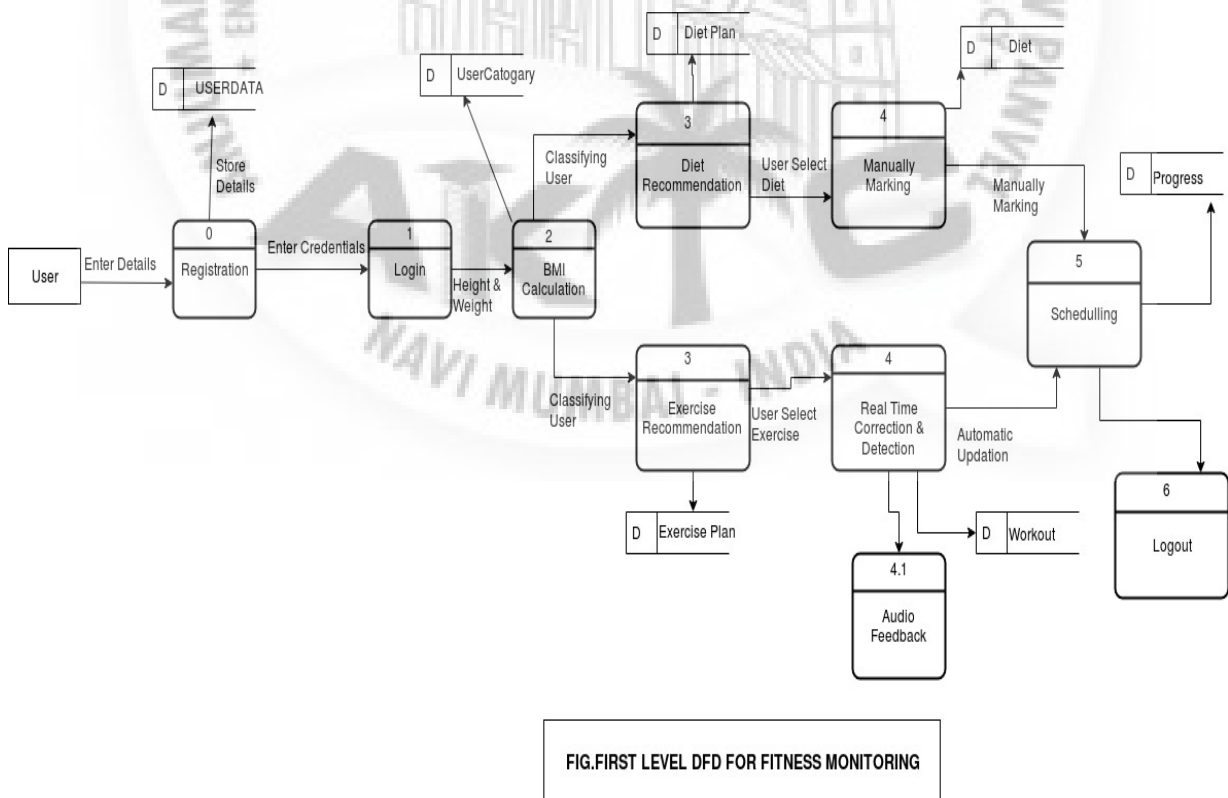


FIG.FIRST LEVEL DFD FOR FITNESS MONITORING

Figure 5.3: DFD Level 1 diagram

DFD Level 2 shows complete details of the functions of modules of system.

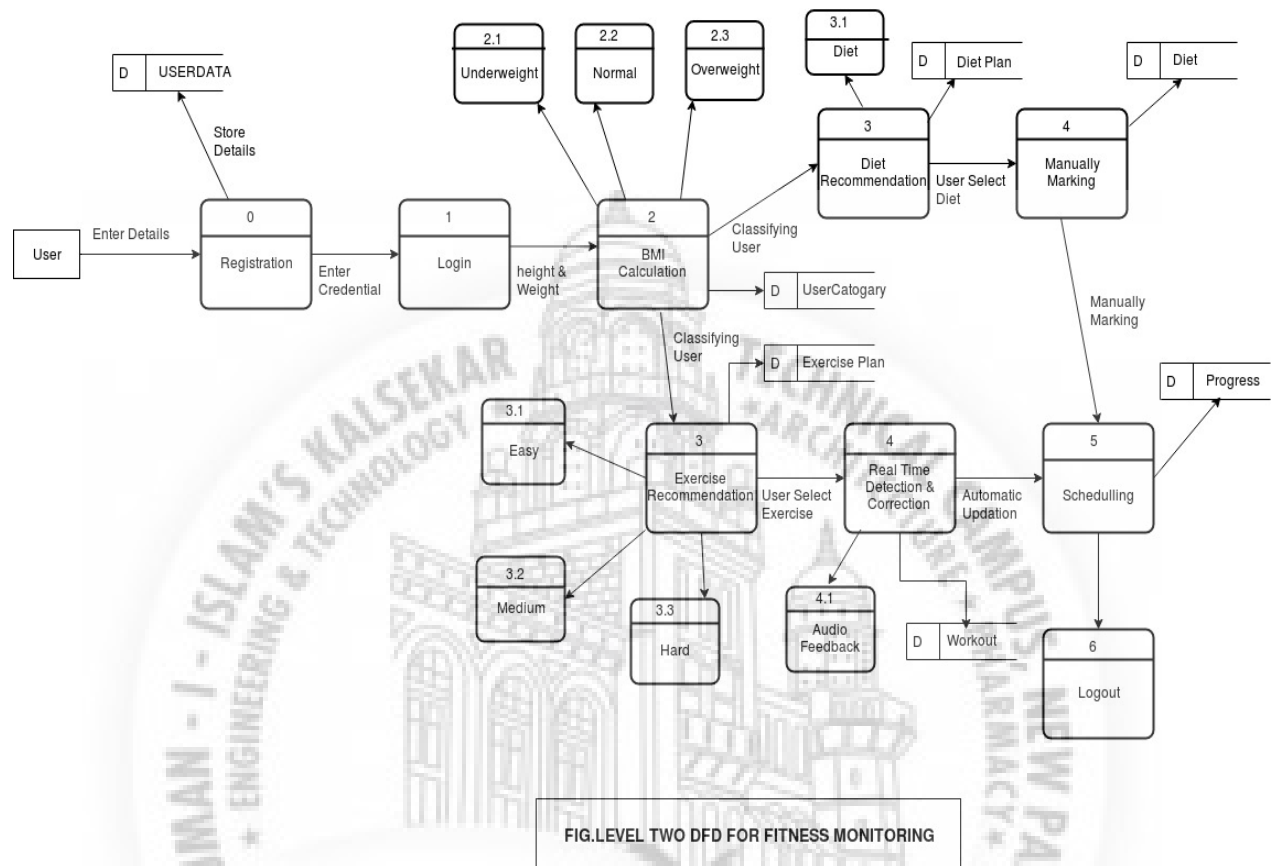


Figure 5.4: DFD Level 2 diagram

5.1.2 System requirements (non-functional requirements)

These are non-functional system properties such as availability, performance and safety etc. They define functions of a system, services and operational constraints in detail.

- Usability - Application implementation is feasible using technologies that are accessible to the end-users.
- Portability - The interfaces are compatible with Android.
- Space Efficiency - Saved audio is of relatively small in size. Also the images are well compressed.
- Performance Efficiency - Application is able to perform well in a proper time constraint.

- e. Multi User System -Application is able to consider the presence of more than one user in the same environment. All the features of the system operates properly for all users and provides proper transparency.
- f. Time Efficiency - Time taken for the executing of system is less.



5.2 System Architecture Design

System Architecture diagram consists of all different components of the system and their relationship with one another.

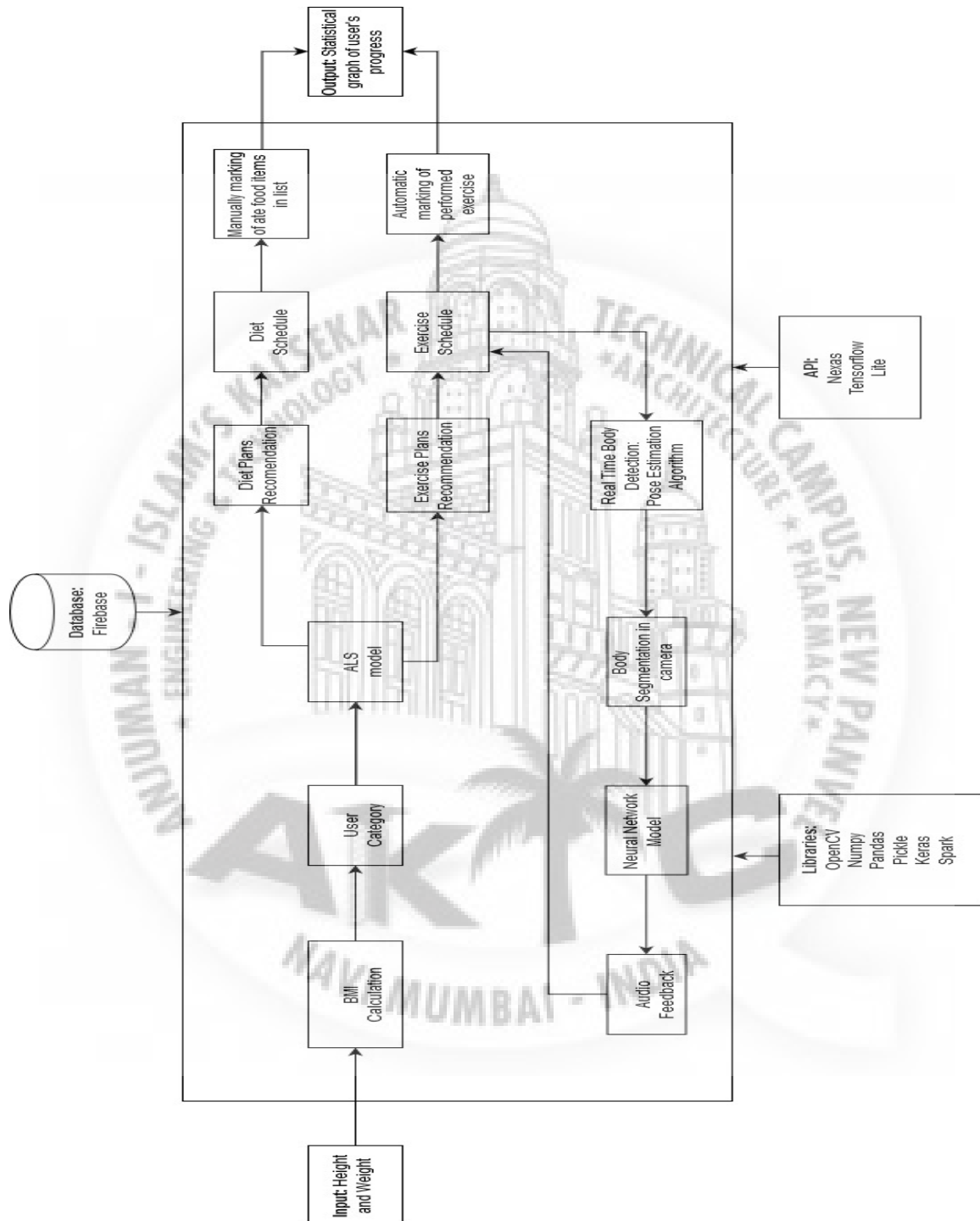


Figure 5.5: System Architecture of Fitness Monitoring using Machine Learning

5.3 Sub-system Development

Our system consists of four different modules: Registration and login, diet and exercise recommendation, schedule and pose detection possessing different functionalities needed in the system.

5.3.1 Registration and Login Module

The user enters the details: email and password. This email and password gets saved in firebase and user is registered[5]. The user has to login to his/her account to access all services of application, by simply entering the registered email with password.

5.3.2 Diet and Exercise Recommendation Module

According to BMI, user is classified in anyone of the three categories: underweight, normal and overweight. As per the classification, it recommends diet and exercise on daily/monthly basis for the particular category.

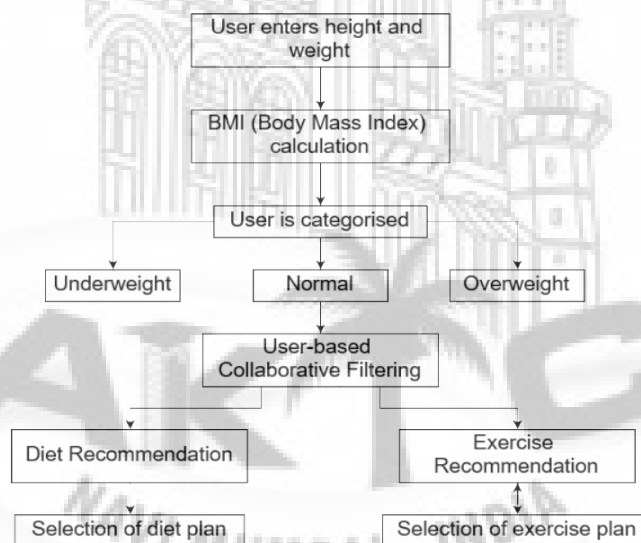


Figure 5.6: Module Diagram for Diet and Exercise

5.3.3 Schedule Module

After selection of any diet and exercise plan, user starts it. User's progress is shown in the form of graph for both diet and exercise. User has to mark manually the ate food items to see their diet progress. User's performed exercise is automatically updated.

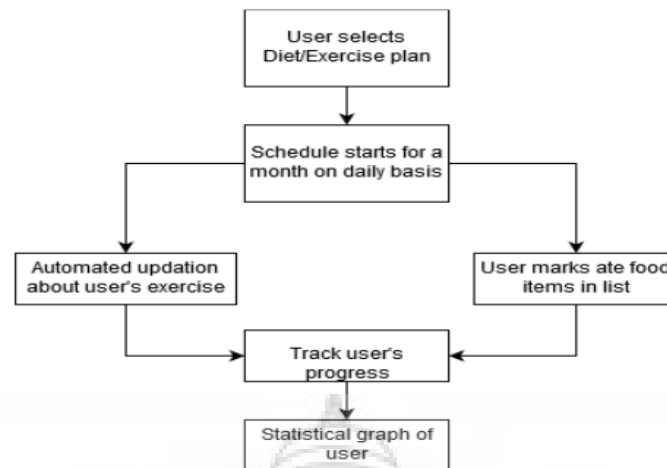


Figure 5.7: Module Diagram for User Schedule

5.3.4 Pose Detection Module

After selection of exercise camera opens in front of user. User starts performing exercise. If he/she performs exercise incorrectly, the real time audio feedback is given to the user to correct their mistakes while performing the exercise.

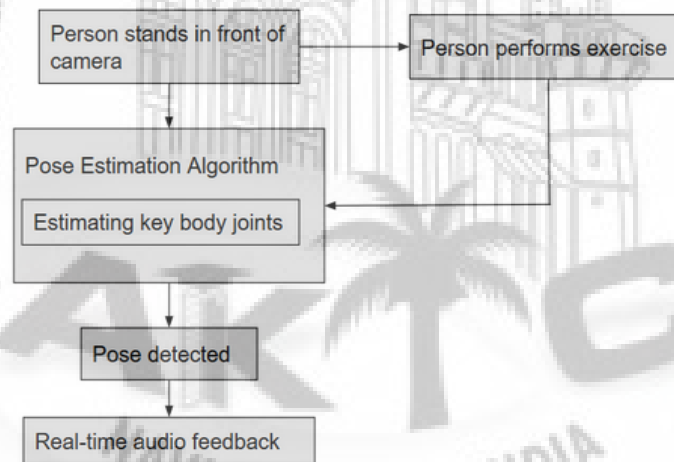


Figure 5.8: Module Diagram for Pose Detection

5.4 Systems Integration

System Integration is the process of integrating all the physical and virtual components of a system.

5.4.1 Class Diagram

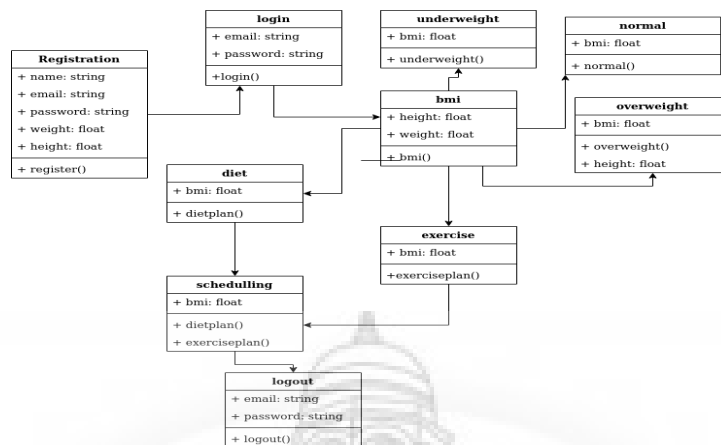


FIG-CLASS DIAGRAM FOR FITNESS MONITORING

Figure 5.9: Class Diagram

5.4.2 Sequence Diagram

Sequence diagram for exercise module shows the sequence of actions when user wants to select exercise plan.

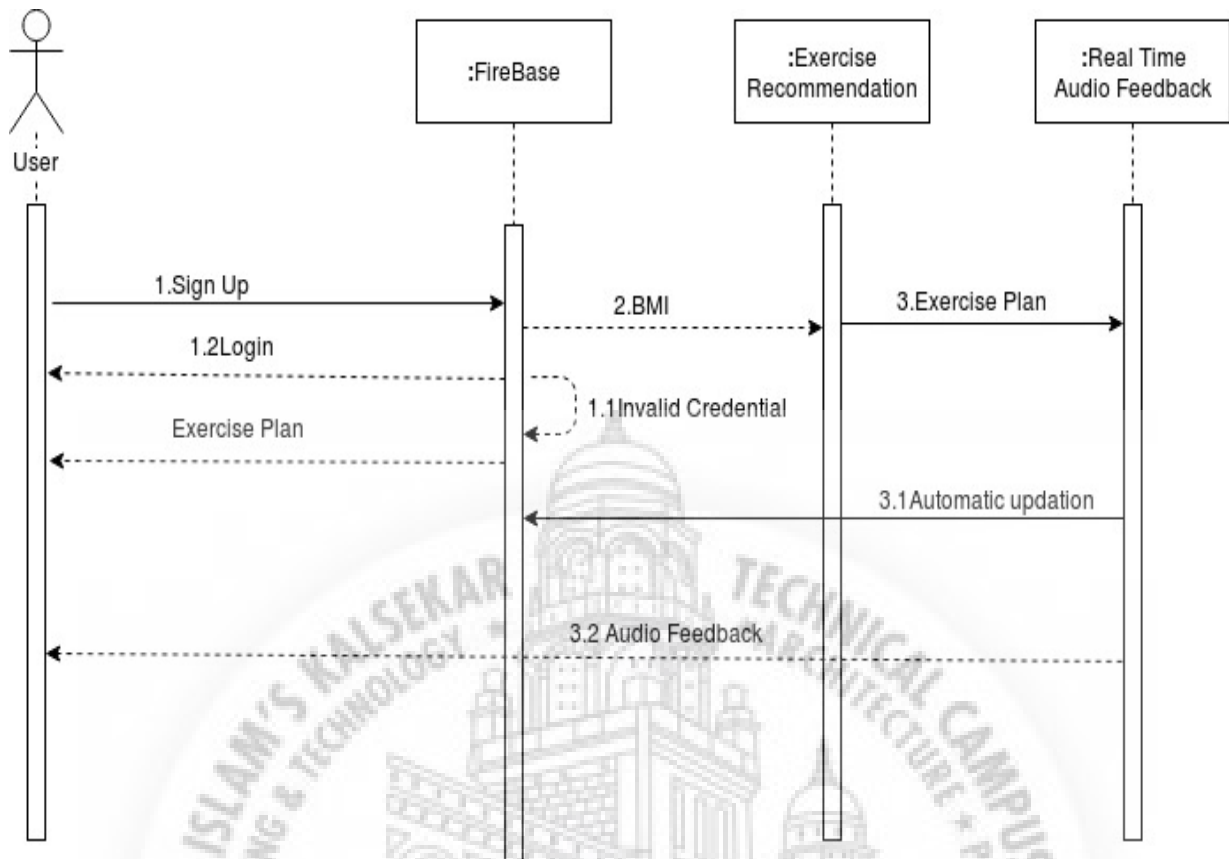


FIG-SEQUENCE DIAGRAM FOR EXERCISE MODULE

Figure 5.10: Sequence diagram for exercise module

Sequence diagram for diet module shows the sequence of actions when user wants to select diet.

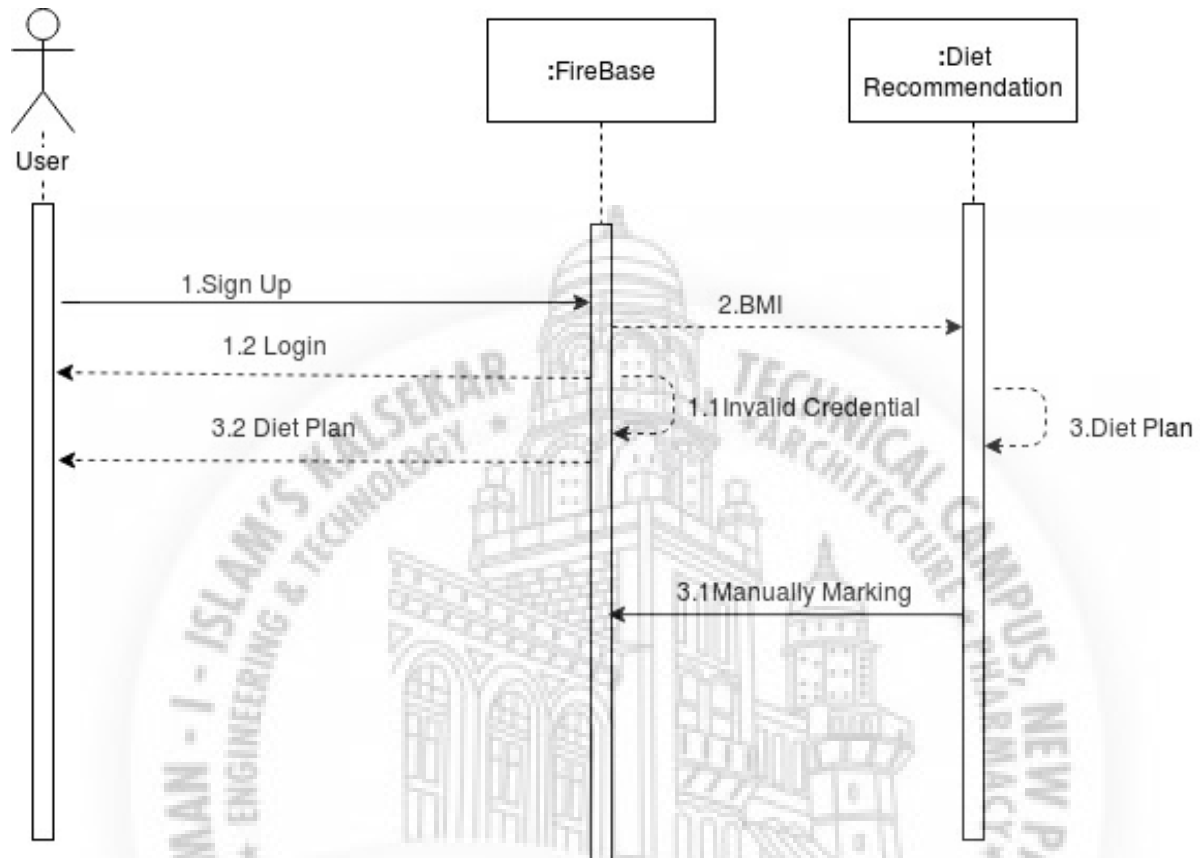


FIG-SEQUENCE DIAGRAM FOR DIET MODULE

Figure 5.11: Sequence diagram for diet module

Sequence diagram for schedule module shows the sequence of actions when user selects exercise and diet plan.

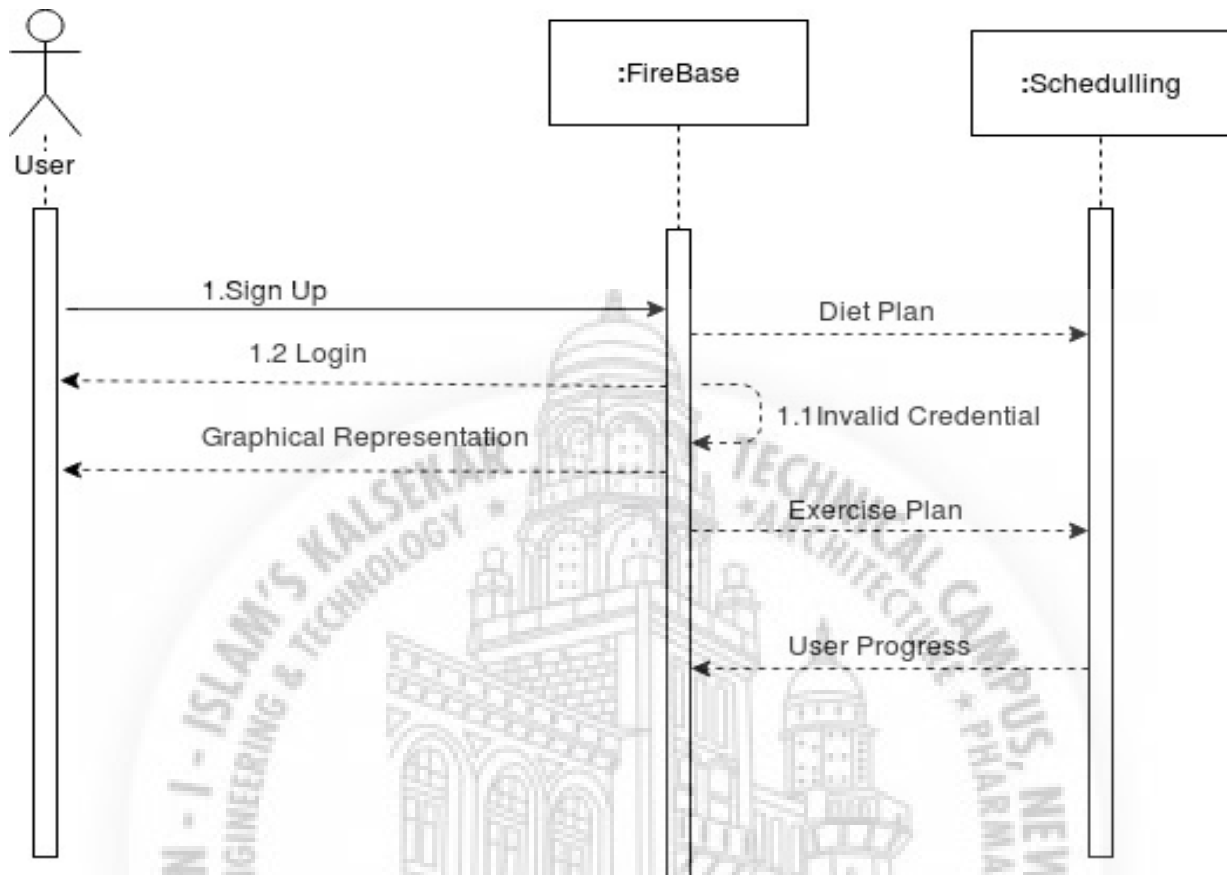


FIG-SEQUENCE DIAGRAM FOR SCHEDULE MODULE

Figure 5.12: Sequence diagram for schedule module

Chapter 6

Implementation

6.1 Module 1

6.1.1 Pose Detection Module

After selection of exercise camera opens in front of user. User starts performing exercise. If he/she performs exercise incorrectly, the real time audio feedback is given to the user to correct their mistakes while performing the exercise.

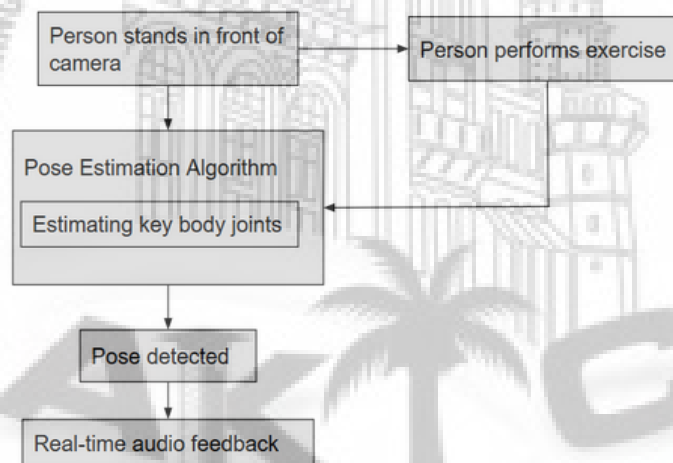


Figure 6.1: Module Diagram for Pose Detection

6.1.2 app.py

```

1 import sys
2 sys.path.append(".")
3 import os
4 import base64
5
6 import settings
7
8 from flask import Flask, request, jsonify, render_template
9 from flask_cors import CORS
10
11 import cv2
12 import imutils
  
```



```

13
14 from utils import convert
15 from vision import analyze
16
17 from helpers.estimator import TfPoseEstimator
18 from helpers.networks import get_graph_path, model_wh
19
20 # Setup Flask app
21 app = Flask(__name__)
22 app.config['DEBUG'] = True
23 app.config['STATIC_FOLDER'] = os.path.join(settings.BASE_DIR, 'static')
24 CORS(app, resources={r"/critique": {"origins": "https://localhost:5000"}})
25
26 # Load model
27 pose_estimator = TfPoseEstimator(graph_path=settings.GRAPH_PATH, target_size=
    settings.PROCESSING_DIMS)
28
29 """
30 Static Files Serving
31 """
32
33 @app.route('/static/<path:path>')
34 def send_static(path):
35     return send_from_directory(app.config['STATIC_FOLDER'], path)
36
37 """
38 Front-end serving
39 """
40 @app.route("/")
41 @app.route("/home")
42 def home():
43     return render_template('home.html')
44
45 @app.route("/login")
46 def login():
47     return render_template('login.html')
48
49 @app.route("/about")
50 def about():
51     return render_template('about.html')
52
53 @app.route("/trainer")
54 def trainer():
55     return render_template('trainer.html')
56
57 @app.route("/friends")
58 def friends():
59     return render_template('friends.html')
60
61 @app.route("/chat")
62 def chat():
63     return render_template('chat.html')
64
65 """
66 Computer Vision
67 """
68
69 @app.route("/critique", methods=['POST'])
70 def critique():
71     """
72     Endpoint for analyzing the video feed.

```

```

73     input:
74     {
75         "image": base64EncodedString ,
76         "workout": string ,
77         "state": int ,
78         "rotate": (optional) int => (degrees),
79         "side": (optional) char => ('F', 'L' or 'R'),
80         "repcount": int
81     }
82
83     return:
84     {
85         "data": {
86             "points": [],
87             "critiques": [],
88             "deviation": float ,
89             "state": int ,
90             "repcount": int
91         },
92     }
93
94     states:
95     0 - rest
96     1 - down
97     2 - up
98     ""
99
100     if request.method == 'POST':
101         # Read image from request
102         encoded_string = request.form['image'].split(',')[ -1]
103         workout = request.form['workout']
104         prev_state = int(request.form.get('state'))
105         side = request.form.get('side')
106         rotate = int(request.form.get('rotate'))
107         rep_count = int(request.form.get('repCount'))
108
109         if rotate == '0':
110             rotate = False
111
112         # Convert image string to image array
113         image = convert.base64_to_array(encoded_string)
114
115         # — Processing and analysis —
116         # Extract body parts
117         body_parts = analyze.extract_body_parts(pose_estimator, image, rotate)
118         if body_parts == -1:
119             return jsonify({
120                 "data": {
121                     "points": -1,
122                     "critiques": -1,
123                     "deviation": -1,
124                     "state": prev_state ,
125                     "repCount": rep_count
126                 },
127                 "status": -1
128             })
129
130         # Assemble points array
131         points = {}
132         for i in range(18):
133             try:
134                 points[i] = (body_parts[i].x, body_parts[i].y)

```

```

134     except KeyError:
135         points[i] = -1
136
137     # Analyze workout from body parts
138     deviation, critique, state = analyze.analyze_workout(body_parts, workout
139         , prev_state, side)
140     if prev_state == 2 and state == 1:
141         rep_count += 1
142
143     """
144     TODO:
145     logic for rep counting
146     logic for set counting
147     database for storing reps and sets
148     analysis for all workouts
149     """
150
151     # Send response with image
152     return jsonify({
153         "data": {
154             "points": points,
155             "critique": critique,
156             "deviation": deviation,
157             "state": state,
158             "repCount": rep_count
159         },
160         "status": -1
161     })
162
163 if __name__ == '__main__':
164     app.run(
165         threaded=True,
166         ssl_context=('cert.pem', 'key.pem')
167     )

```

6.1.3 home.html

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>H & F</title>
5     <link href="https://fonts.googleapis.com/css?family=Lato:400,700" rel="
6         stylesheet">
7     <link rel="stylesheet" href="/static/css/bootstrap.min.css" integrity="sha384-
8         BVYiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
9         crossorigin="anonymous">
10    <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/
11        font-awesome/4.7.0/css/font-awesome.min.css">
12    <link rel="stylesheet" type="text/css" href="/static/css/homePage.css">
13
14    <script
15        src="/static/js/jquery-3.3.1.js"
16        integrity="sha256-2Kok7MbOyxpgUVvAk/HJ2jigOSYS2auK4Pfbzm7uH60="
17        crossorigin="anonymous"></script>

```

```

18 <script>
19 // This is called with the results from from FB.getLoginStatus().
20 function statusChangeCallback(response) {
21   console.log('statusChangeCallback');
22   console.log(response);
23   // The response object is returned with a status field that lets the
24   // app know the current login status of the person.
25   // Full docs on the response object can be found in the documentation
26   // for FB.getLoginStatus().
27   if (response.status === 'connected') {
28     // Logged into your app and Facebook.
29     testAPI();
30   } else {
31     // The person is not logged into your app or we are unable to tell.
32     document.getElementById('status').innerHTML = 'Please log ' +
33       'into this app.';
34   }
35 }
36
37 // This function is called when someone finishes with the Login
38 // Button. See the onlogin handler attached to it in the sample
39 // code below.
40 function checkLoginState() {
41   FB.getLoginStatus(function(response) {
42     statusChangeCallback(response);
43   });
44 }
45
46 window.fbAsyncInit = function() {
47   FB.init({
48     appId      : '154689001865494',
49     cookie     : true,  // enable cookies to allow the server to access
50                 // the session
51     xfbml      : true,  // parse social plugins on this page
52     version    : 'v2.12' // use graph api version 2.8
53   });
54
55   // Now that we've initialized the JavaScript SDK, we call
56   // FB.getLoginStatus(). This function gets the state of the
57   // person visiting this page and can return one of three states to
58   // the callback you provide. They can be:
59   //
60   // 1. Logged into your app ('connected')
61   // 2. Logged into Facebook, but not your app ('not_authorized')
62   // 3. Not logged into Facebook and can't tell if they are logged into
63   //    your app or not.
64   //
65   // These three cases are handled in the callback function.
66
67   FB.getLoginStatus(function(response) {
68     statusChangeCallback(response);
69   });
70
71 };
72
73 // Load the SDK asynchronously
74 (function(d, s, id) {
75   var js, fjs = d.getElementsByTagName(s)[0];
76   if (d.getElementById(id)) return;
77   js = d.createElement(s); js.id = id;
78   js.src = "https://connect.facebook.net/en-US/sdk.js";

```

```

79     fjs.parentNode.insertBefore(js, fjs);
80 } (document, 'script', 'facebook-jssdk'));
81
82 // Here we run a very simple test of the Graph API after login is
83 // successful. See statusChangeCallback() for when this call is made.
84 function testAPI() {
85     console.log('Welcome! Fetching your information.... ');
86     FB.api('/me', function(response) {
87         console.log('Successful login for: ' + response.name);
88         hiLbl(response.name, response.id);
89     });
90 }
91
92 function hiLbl(name, id){
93     console.log(name, id);
94     var imgURL = "http://graph.facebook.com/" + id + "/picture?type=normal"
95     document.getElementById('fbdisp').innerHTML =
96         'Hi, ' + name + '! ' +
97         '
99         ' +
100         '<a style="display: inline;" onclick="FB.logout();revertLbl();">Logout </
101         a>';
102 }
103
104 function revertLbl(){
105     document.getElementById('fbdisp').innerHTML =
106         '<a onclick="FB.login();checkLoginState();" id="facebook">Login with <i
107         class="fa fa-facebook" aria-hidden="true"></i>acebook</a>';
108 }
109 </script>
110
111 <video autoplay muted loop id="myVideo">
112     <source src="/static/images/video.mp4" type="video/mp4">
113 </video>
114
115 <!-- particles.js container --> <div id="particles-js"></div> <!-- stats - count
116 particles --> <div class="count-particles"> <span class="js-count-particles
117 "></span> particles </div> <!-- particles.js lib - https://github.com/
118 VincentGarreau/particles.js --> <script src="https://cdn.jsdelivr.net/
119 particles.js/2.0.0/particles.min.js"></script> <!-- stats.js lib --> <script
120 src="https://threejs.org/examples/js/libs/stats.min.js"></script>
121
122 <nav class="navbar navbar-default">
123     <div class="container">
124         <!-- Brand and toggle get grouped for better mobile display -->
125         <div class="navbar-header">
126             <button type="button" class="navbar-toggle collapsed" data-toggle="
127             collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="
128             false">
129                 <span class="sr-only">Toggle navigation </span>
130                 <span class="icon-bar"></span>
131                 <span class="icon-bar"></span>
132                 <span class="icon-bar"></span>
133             </button>
134             <a id="logo" class="navbar-brand" href="/home"></a>
136         </div>

```

```

128
129 <!-- Collect the nav links, forms, and other content for toggling -->
130 <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
131   <ul class="nav navbar-nav">
132     <li><a href="/about">About <i class="fa fa-search" aria-hidden="true"></i>
133       > </a> </li>
134   </ul>
135 </div><!-- /.navbar-collapse -->
136 </div><!-- /.container-fluid -->
137 </nav>
138
139
140
141 <div class="container">
142   <div class="row">
143
144     <div class="col-lg-12">
145       <div id="content">
146         <h1>H & F</h1>
147         <h3>Using AI To Maximize Your Workout</h3>
148         <hr>
149         <a href="/trainer"><button id="lock" class="btn btn-default btn-lg"><i
150           class="fa fa-lock" aria-hidden="true"></i> Unlock Your Potential </
151           button></a>
152       </div>
153     </div>
154   </div>
155 </div>
156
157
158
159 <script type="text/javascript">particlesJS("particles-js", {"particles":{"number
160   ":{"value":80,"density":{"enable":true,"value_area":800}},"color":{"value":
161   "#ffffff"},"shape":{"type":"circle","stroke":{"width":0,"color":"#000000"}},
162   "polygon":{"nb_sides":5},"image":{"src":"img/github.svg","width":100,"height":
163   100},"opacity":{"value":0.5,"random":false,"anim":{"enable":false,"speed":
164   1,"opacity_min":0.1,"sync":false}},"size":{"value":3,"random":true,"anim":{"
165   enable":false,"speed":40,"size_min":0.1,"sync":false}},"line_linked":{"
166   enable":true,"distance":150,"color":"#ffffff","opacity":0.4,"width":1},"move":
167   {"enable":true,"speed":6,"direction":"none","random":false,"straight":
168   false,"out_mode":"out","bounce":false,"attract":{"enable":false,"rotateX":
169   600,"rotateY":1200}}},"interactivity":{"detect_on":"canvas","events":{"
170   onhover":{"enable":true,"mode":"repulse"},"onclick":{"enable":true,"mode":
171   "push"},"resize":true},"modes":{"grab":{"distance":400,"line_linked":{"
172   opacity":1}},"bubble":{"distance":400,"size":40,"duration":2,"opacity":8,"
173   speed":3},"repulse":{"distance":200,"duration":0.4},"push":{"particles_nb":
174   4},"remove":{"particles_nb":2}}},"retina_detect":true});var count_particles
175   , stats, update; stats = new Stats; stats.setMode(0); stats.domElement.style
176   .position = 'absolute'; stats.domElement.style.left = '0px'; stats.
177   domElement.style.top = '0px'; document.body.appendChild(stats.domElement);
178   count_particles = document.querySelector('.js-count-particles'); update =
179   function() { stats.begin(); stats.end(); if (window.pJSDom[0].pJS.particles
180   && window.pJSDom[0].pJS.particles.array) { count_particles.innerHTML =
181   window.pJSDom[0].pJS.particles.array.length; } requestAnimationFrame(update)
182   ; }; requestAnimationFrame(update);</script>

```

```

163
164
165
166
167
168
169 <script type="text/javascript"> //
170   $(document).ready(function () {
171     $('#lock').hover(function () {
172       console.log("HOVER");
173       $('#lock i').removeClass('fa-lock');
174       $('#lock i').addClass('fa-unlock');
175     }, function () {
176       $('#lock i').removeClass('fa-unlock');
177       $('#lock i').addClass('fa-lock');
178     });
179   });
180 </script>
181
182 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js
   " integrity="sha384-
   Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA712mCWNIpG9mGCD8wGNiCPD7Txa"
   crossorigin="anonymous"></script>
183 </body>
184 </html>

```

6.1.4 trainer.html

```

1 <!DOCTYPE html>
2 <!--
3  * Copyright (c) 2015 The WebRTC project authors. All Rights Reserved.
4  *
5  * Use of this source code is governed by a BSD-style license
6  * that can be found in the LICENSE file in the root of the source
7  * tree.
8  -->
9
10 <html>
11
12 <head>
13   <title>H & F</title>
14
15   <link rel="stylesheet" type="text/css" href="static/trainer/main.css">
16   <script src="/static/js/jquery-3.3.1.js"></script>
17   <script src="/static/js/bootstrap.min.js"></script>
18
19   <link rel="stylesheet" href="/static/css/bootstrap.min.css" integrity="
   sha384-BVYiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
   crossorigin="anonymous">
20   <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/
   font-awesome/4.7.0/css/font-awesome.min.css">
21 </head>
22
23 <body id="bdy" onresize="adjustSize();">
24
25   <div id="videoCanvasContainer">
26     <canvas id="videoCanvas"></canvas>
27

```

```

28     <div id="hiddenContainer" hidden=true>
29         <video id="video" playsinline autoplay></video>
30         <canvas id="transmit" width=320px; height=240px></canvas>
31         <!--canvas id="receiverCanvas" width=320px; height=240px></canvas-->
32     </div>
33 </div>
34
35 <div id='sidebar'>
36     <div class="jumbotron">
37         <div style="margin-bottom: 20px;"><a href="/home"></a></div>
39         <div class="select">
40             <label for="workoutMode">Workout Mode: </label>
41             <select id="workoutSelector" onchange="changeWorkout();">
42                 <option value="none">Select a workout</option>
43                 <option value="curls">Curls</option>
44                 <option value="squats">Squats</option>
45                 <option value="shoulderpress">Shoulder Press</option>
46             </select>
47         </div>
48         <div class="select">
49             <label for="videoSource">Video source: </label>
50             <select id="videoSource"></select>
51         </div>
52
53         <h1 class="timer">
54             <span class="minutes">00</span>
55             :
56             <span class="seconds">00</span>
57         </h1>
58         <h3>
59             Reps: <span id="repCounter">0</span>
60         </h3>
61         <h3>
62             Deviation: <span id="deviationDisplay">0</span>
63         </h3>
64         <h3>
65             State: <span id="stateDisplay">0</span>
66         </h3>
67         <h3>
68             Side: <span id="sideDisplay">Nones</span>
69         </h3>
70     <div id="indicator"></div>
71
72     <div class="timer-buttons">
73         <button id="start" class="btn btn-lg btn-default btn-success"
74             data-action="start">Start</button>
75         <button id="stop" class="btn btn-lg btn-default btn-danger" data
76             -action="stop">Stop</button>
77         <button id="reset" class="btn btn-lg btn-default">Reset</button>
78     </div>
79
80     <div id="chatOut">
81         <p></p>
82     </div>
83 </div>
84
85 <script src="/static/trainer/main.js"></script>

```



```

86
87 </body>
88
89 </html>

```

6.1.5 homepage.css

```

1 #content{
2   text-align:center;
3   padding-top: 25%;
4   text-shadow: 0px 4px 3px rgba(0,0,0,0.4),
5               0px 8px 13px rgba(0,0,0,0.1),
6               0px 18px 23px rgba(0,0,0,0.1);
7 }
8
9 /* —— reset —— */ canvas{ display: block; vertical-align: bottom; } /* ——
particles.js container —— */ #particles-js{ position:absolute; width:
100%; height: 100%; background-repeat: no-repeat; background-size: cover;
background-position: 50% 50%; } /* —— stats.js —— */ .count-particles{
background: #000022; position: absolute; top: 48px; left: 0; width: 80px;
color: #13E8E9; font-size: .8em; text-align: left; text-indent: 4px; line-
height: 14px; padding-bottom: 2px; font-family: Helvetica, Arial, sans-serif
; font-weight: bold; } .js-count-particles{ font-size: 1.1em; } #stats, .
count-particles{ -webkit-user-select: none; margin-top: 5px; margin-left: 5
px; } #stats{ border-radius: 3px 3px 0 0; overflow: hidden; } .count-
particles{ border-radius: 0 0 3px 3px; }
10
11 body > div:nth-child(10){
12 visibility: hidden;
13 }
14
15 body > div.count-particles{
16 visibility: hidden;
17 }
18
19 #logo{
20 background-repeat: no-repeat;
21 }
22
23
24 body{
25   background: url(victor-freitas-570027-unsplash.jpg);
26   background-size: cover;
27   background-position: center;
28   font-family: Lato;
29   color: white;
30 }
31
32 h1{
33   font-size: 5em;
34   font-weight: 700;
35 }
36
37
38
39 #facebook: hover{
40   color: #3b5998;
41   font-weight: 1000;

```

```
42 }
43
44 #lock: hover {
45     font-weight: 1000;
46 }
47
48 #myVideo {
49     position: fixed;
50     right: 0;
51     bottom: 0;
52     min-width: 100%;
53     min-height: 100%;
54 }
55
56 html {
57     height: 100%;
58 }
59
60 hr {
61     width: 400px;
62     border-top: 1px solid #f8f8f8;
63     border-bottom: 1px solid rgba(0,0,0,0.2);
64 }
```



6.2 Module 2

6.2.1 Diet Module

According to BMI, user is classified in anyone of the three categories: underweight, normal and overweight. As per the classification, it recommends diet on daily/monthly basis for the particular category.

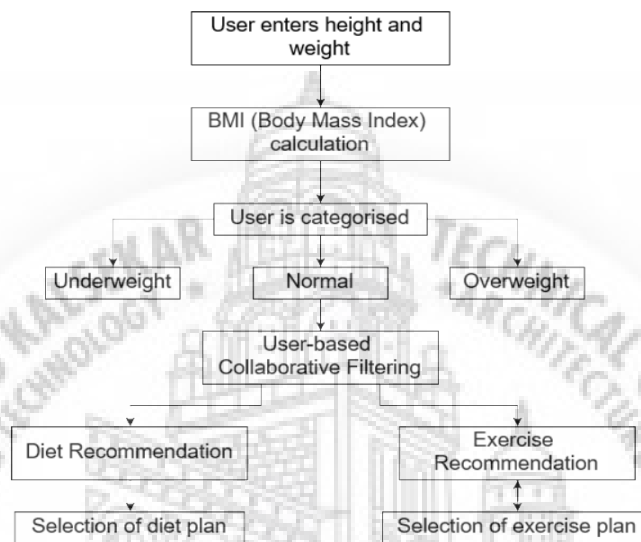


Figure 6.2: Module Diagram for Diet and Exercise

6.2.2 AndroidManifest.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.hf">
4
5     <uses-permission android:name="android.permission.INTERNET" />
6
7     <application
8         android:allowBackup="true"
9         android:icon="@mipmap/ic_launcher"
10        android:label="@string/app_name"
11        android:roundIcon="@mipmap/ic_launcher_round"
12        android:supportRtl="true"
13        android:theme="@style/Theme.AppCompat.Light.NoActionBar">
14        <activity android:name=".MainActivity">
15            <intent-filter>
16                <action android:name="android.intent.action.MAIN" />
17
18                <category android:name="android.intent.category.LAUNCHER" />
19            </intent-filter>
20        </activity>
21        <activity android:name=".LoginActivity" />
22        <activity android:name=".UserProfileActivity" />
23    </application>
24
25 </manifest>
  
```

6.2.3 LoginActivity.java

```
1 package com.example.hf;
2
3 import android.app.ProgressDialog;
4 import android.content.Intent;
5 import android.os.Bundle;
6 import android.text.TextUtils;
7 import android.view.View;
8 import android.widget.Button;
9 import android.widget.EditText;
10 import android.widget.Toast;
11
12 import androidx.appcompat.app.AppCompatActivity;
13 import androidx.appcompat.widget.Toolbar;
14
15 import com.google.android.gms.tasks.OnCompleteListener;
16 import com.google.android.gms.tasks.Task;
17 import com.google.firebase.auth.AuthResult;
18 import com.google.firebase.auth.FirebaseAuth;
19
20 public class LoginActivity extends AppCompatActivity {
21
22     // Creating EditText.
23     EditText email, password;
24
25     // Creating string to hold values.
26     String EmailHolder, PasswordHolder;
27
28     // Creating buttons.
29     Button Login, SignUP;
30
31     // Creating Boolean to hold EditText empty true false value.
32     Boolean EditTextEmptyCheck;
33
34     // Creating progress dialog.
35     ProgressDialog progressDialog;
36
37     // Creating FirebaseAuth object.
38     FirebaseAuth firebaseAuth;
39
40     Toolbar toolbar;
41
42     @Override
43     protected void onCreate(Bundle savedInstanceState) {
44         super.onCreate(savedInstanceState);
45         setContentView(R.layout.activity_login);
46
47         toolbar = findViewById(R.id.toolbar);
48         setSupportActionBar(toolbar);
49         getSupportActionBar().setTitle("H & F");
50         toolbar.setSubtitle("A fitness app");
51
52         // Assign ID's to EditText.
53         email = (EditText) findViewById(R.id.editText_email);
54         password = (EditText) findViewById(R.id.editText_password);
55     }
56 }
```

```

55
56 // Assign ID's to button.
57 Login = (Button)findViewById(R.id.button_login);
58 SignUP = (Button)findViewById(R.id.button_SignUP);
59
60 progressDialog = new ProgressDialog(LoginActivity.this);
61
62 // Assign FirebaseAuth instance to FirebaseAuth object.
63 firebaseAuth = FirebaseAuth.getInstance();
64
65
66 // Checking if stats already logged in before and not logged out
67 // properly.
68 if(firebaseAuth.getCurrentUser() != null){
69
70     // Finishing current Login Activity.
71     finish();
72
73     // Opening UserProfileActivity .
74     Intent intent = new Intent(LoginActivity.this, UserProfileActivity.
75         class);
76     startActivity(intent);
77 }
78
79 // Adding click listener to login button.
80 Login.setOnClickListener(new View.OnClickListener() {
81     @Override
82     public void onClick(View view) {
83
84         // Calling method CheckEditTextIsEmptyOrNot().
85         CheckEditTextIsEmptyOrNot();
86
87         // If EditTextEmptyCheck == true
88         if (EditTextEmptyCheck)
89         {
90
91             // If EditTextEmptyCheck == true then login function called
92             LoginFunction();
93         }
94         else {
95
96             // If EditTextEmptyCheck == false then toast display on
97             // screen.
98             Toast.makeText(LoginActivity.this, "Please Fill All the
99             Fields", Toast.LENGTH_LONG).show();
100         }
101     }
102 });
103
104 // Adding click listener to Sign up button.
105 SignUP.setOnClickListener(new View.OnClickListener() {
106     @Override
107     public void onClick(View view) {
108
109         // Closing current activity.
110         finish();

```

```
111
112     // Opening the Main Activity .
113     Intent intent = new Intent(LoginActivity.this , MainActivity.
114         class);
115     startActivity(intent);
116     }
117     });
118 }
119
120 // Creating method to check EditText is empty or not.
121 public void CheckEditTextIsEmptyOrNot(){
122
123     // Getting value form Email's EditText and fill into EmailHolder string
124     // variable.
125     EmailHolder = email.getText().toString().trim();
126
127     // Getting value form Password's EditText and fill into PasswordHolder
128     // string variable.
129     PasswordHolder = password.getText().toString().trim();
130
131     // Checking Both EditText is empty or not.
132     // If any of EditText is empty then set value as false.
133     // If any of EditText is empty then set value as true.
134     EditTextEmptyCheck = !TextUtils.isEmpty(EmailHolder) && !TextUtils.
135         isEmpty(PasswordHolder);
136
137 }
138
139 // Creating login function.
140 public void LoginFunction(){
141
142     // Setting up message in progressDialog.
143     progressDialog.setMessage("Please Wait");
144
145     // Showing progressDialog.
146     progressDialog.show();
147
148     // Calling signInWithEmailAndPassword function with firebase object and
149     // passing EmailHolder and PasswordHolder inside it.
150     firebaseAuth.signInWithEmailAndPassword(EmailHolder , PasswordHolder)
151         .addOnCompleteListener(this , new OnCompleteListener<AuthResult
152             >() {
153             @Override
154             public void onComplete(@NonNull Task<AuthResult> task) {
155
156                 // If task done Successful.
157                 if(task.isSuccessful()){
158
159                     // Hiding the progress dialog.
160                     progressDialog.dismiss();
161
162                     // Closing the current Login Activity.
163                     finish();
164
165                     // Opening the UserProfileActivity.
166                     Intent intent = new Intent(LoginActivity.this ,
167                         UserProfileActivity.class);
168                     startActivity(intent);
169                 }
170             }
171         });
172 }
```

```

165         else {
166             // Hiding the progress dialog.
167             progressDialog.dismiss();
168
169             // Showing toast message when email or password not
170             found in Firebase Online database.
171             Toast.makeText(LoginActivity.this, "Email or
172             Password Not found, Please Try Again", Toast.
173             LENGTHLONG).show();
174         }
175     });
176 }
177
178 }

```

6.2.4 MainActivity.java

```

1 package com.example.hf;
2
3
4 import android.app.AlertDialog;
5 import android.content.Intent;
6 import android.os.Bundle;
7 import android.text.TextUtils;
8 import android.view.Menu;
9 import android.view.View;
10 import android.widget.Button;
11 import android.widget.EditText;
12 import android.widget.Toast;
13
14 import androidx.annotation.NonNull;
15 import androidx.appcompat.app.AppCompatActivity;
16 import androidx.appcompat.widget.Toolbar;
17
18 import com.google.android.gms.tasks.OnCompleteListener;
19 import com.google.android.gms.tasks.Task;
20 import com.google.firebase.auth.AuthResult;
21 import com.google.firebase.auth.FirebaseAuth;
22
23 public class MainActivity extends AppCompatActivity {
24
25     // Creating EditText .
26     EditText email, password ;
27
28     // Creating button.
29     Button SignUp, ButtonGoToLoginActivity;
30
31     // Creating string to hold email and password .
32     String EmailHolder, PasswordHolder ;
33
34     // Creating Progress dialog.
35     ProgressDialog progressDialog;
36
37     // Creating FirebaseAuth object.
38     FirebaseAuth firebaseAuth ;

```

```
39
40 // Creating Boolean variable that holds EditText is empty or not status.
41 Boolean EditTextStatus ;
42
43 Toolbar toolbar;
44
45 @Override
46 protected void onCreate(Bundle savedInstanceState) {
47     super.onCreate(savedInstanceState);
48     setContentView(R.layout.activity_main);
49
50     toolbar = findViewById(R.id.toolbar);
51     setSupportActionBar(toolbar);
52     getSupportActionBar().setTitle("H & F");
53     toolbar.setSubtitle("A fitness app");
54
55     // Assigning layout email ID and Password ID.
56     email = (EditText) findViewById(R.id.EditText_User_EmailID);
57     password = (EditText) findViewById(R.id.EditText_User_Password);
58
59     // Assign button layout ID.
60     SignUp = (Button) findViewById(R.id.Button_SignUp);
61     ButtonGoToLoginActivity = (Button) findViewById(R.id.Button_LoginActivity
62         );
63
64     // Creating object instance.
65     firebaseAuth = FirebaseAuth.getInstance();
66
67     progressDialog = new ProgressDialog(MainActivity.this);
68
69     // Adding click listener to Sign Up Button.
70     SignUp.setOnClickListener(new View.OnClickListener() {
71         @Override
72         public void onClick(View view) {
73
74             // Calling method to check EditText is empty or no status.
75             CheckEditTextIsEmptyOrNot();
76
77             // If EditText is true then this block with execute.
78             if(EditTextStatus){
79
80                 // If EditText is not empty than UserRegistrationFunction
81                 // method will call.
82                 UserRegistrationFunction();
83             }
84             // If EditText is false then this block with execute.
85             else {
86
87                 Toast.makeText(MainActivity.this, "Please fill all form
88                 fields.", Toast.LENGTH_LONG).show();
89             }
90         }
91     });
92
93     // Adding click listener to ButtonGoToLoginActivity button.
94     ButtonGoToLoginActivity.setOnClickListener(new View.OnClickListener() {
95         @Override
96         public void onClick(View view) {
```



```

97
98 // Finishing current Main Activity.
99 finish ();
100
101 // Opening the Login Activity using Intent.
102 Intent intent = new Intent(MainActivity.this , LoginActivity.
103     class);
104     startActivity (intent);
105 }
106 });
107
108
109 }
110
111 // Creating UserRegistrationFunction
112 public void UserRegistrationFunction () {
113
114     // Showing progress dialog at stats registration time.
115     progressDialog.setMessage("Please Wait, We are Registering Your Data on
116         Server");
117     progressDialog.show ();
118
119     // Creating createUserWithEmailAndPassword method and pass email and
120     // password inside it.
121     firebaseAuth.createUserWithEmailAndPassword (EmailHolder , PasswordHolder)
122     .addOnCompleteListener (MainActivity.this , new OnCompleteListener<
123     AuthResult > () {
124         @Override
125         public void onComplete (@NonNull Task<AuthResult> task) {
126
127             // Checking if stats is registered successfully.
128             if (task.isSuccessful ()) {
129
130                 // If stats registered successfully then show this
131                 // toast message.
132                 Toast.makeText (MainActivity.this , "User Registration
133                     Successfully" , Toast.LENGTHLONG) .show ();
134                 firebaseAuth.signOut ();
135             } else {
136
137                 // If something goes wrong.
138                 Toast.makeText (MainActivity.this , "Something Went
139                     Wrong." , Toast.LENGTHLONG) .show ();
140             }
141
142             // Hiding the progress dialog after all task complete.
143             progressDialog.dismiss ();
144         }
145     });
146 }
147
148 public void CheckEditTextIsEmptyOrNot () {
149
150     // Getting name and email from EditText and save into string variables.
151     EmailHolder = email.getText ().toString ().trim ();

```

```

150     PasswordHolder = password.getText().toString().trim();
151
152     EditTextStatus = !TextUtils.isEmpty(EmailHolder) && !TextUtils.isEmpty(
153         PasswordHolder);
154 }
155
156 }

```

6.2.5 DietFragment.java

```

1
2 package com.example.hf;
3
4
5 import android.annotation.SuppressLint;
6 import android.content.Intent;
7 import android.graphics.Typeface;
8 import android.os.Build;
9 import android.os.Bundle;
10
11 import androidx.annotation.RequiresApi;
12 import androidx.appcompat.app.AlertDialog;
13 import androidx.fragment.app.Fragment;
14 import androidx.fragment.app.FragmentManager;
15 import androidx.fragment.app.FragmentTransaction;
16
17 import android.text.TextUtils;
18 import android.util.Log;
19 import android.view.LayoutInflater;
20 import android.view.View;
21 import android.view.ViewGroup;
22 import android.widget.Button;
23 import android.widget.EditText;
24 import android.widget.RadioButton;
25 import android.widget.RelativeLayout;
26 import android.widget.Spinner;
27 import android.widget.TextView;
28
29 import java.util.Objects;
30
31 import static android.view.View.INVISIBLE;
32 import static android.view.View.VISIBLE;
33
34
35 /**
36  * A simple {@link Fragment} subclass.
37  */
38 public class DietFragment extends Fragment {
39
40     NormalWeightFragment normalWeightFragment;
41     UnderWeightFragment underWeightFragment;
42     OverWeightFragment overWeightFragment;
43     RelativeLayout dietfr;
44     double bmrMen, bmrWomen;
45     float tbmrresult, calories, calorieintake, bmi;
46     TextView tcaloriesresult, tBMRResult;
47     RadioButton male, female;

```

```

48     private Button calculatebutton , clearbutton , dietbutton ;
49
50     public DietFragment() {
51         // Required empty public constructor
52     }
53
54     @Override
55     public View onCreateView(LayoutInflater inflater , ViewGroup container ,
56                             Bundle savedInstanceState) {
57         // Inflate the layout for this fragment
58         View view = inflater.inflate(R.layout.fragment_diet , container , false);
59
60         dietfr = (RelativeLayout) view.findViewById(R.id.diet_frame);
61         final EditText bheight = (EditText) view.findViewById(R.id.height);
62         final EditText bweight = (EditText) view.findViewById(R.id.weight);
63         final EditText age = (EditText) view.findViewById(R.id.age);
64         male = (RadioButton) view.findViewById(R.id.male);
65         female = (RadioButton) view.findViewById(R.id.female);
66         calculatebutton = (Button) view.findViewById(R.id.calculatebtn);
67         clearbutton = (Button) view.findViewById(R.id.clearbtn);
68         tBMRResult = (TextView) view.findViewById(R.id.bmrresult);
69         tcaloriesresult = (TextView) view.findViewById(R.id.caloriesresult);
70         dietbutton = (Button) view.findViewById(R.id.dietplan);
71
72         normalWeightFragment = new NormalWeightFragment();
73         overWeightFragment = new OverWeightFragment();
74         underWeightFragment = new UnderWeightFragment();
75
76         calculatebutton.setOnClickListener(new View.OnClickListener() {
77             @Override
78             public void onClick(View v) {
79                 String str1 = bheight.getText().toString();
80                 String str2 = bweight.getText().toString();
81                 String str3 = age.getText().toString();
82
83                 if(TextUtils.isEmpty(str1)){
84                     bheight.setError("Please enter your height (feet)");
85                     bheight.requestFocus();
86                     return;
87                 }
88
89                 if(TextUtils.isEmpty(str2)){
90                     bweight.setError("Please enter your weight");
91                     bweight.requestFocus();
92                     return;
93                 }
94
95                 if(TextUtils.isEmpty(str3)){
96                     age.setError("Please enter your age");
97                     age.requestFocus();
98                     return;
99                 }
100
101                 float bheightValue = Float.parseFloat(str1);
102                 float bweightValue = Float.parseFloat(str2);
103                 float age = Float.parseFloat(str3);
104                 tBMRResult = calculateBMR(bweightValue , bheightValue , age);
105                 tBMRResult.setText(String.valueOf("BMR: " + tBMRResult));
106                 calories = calculateCalories(tBMRResult);
107                 tcaloriesresult.setText(String.valueOf("Calories needed: " +
                    calories));

```

```

108     }
109 });
110
111 clearbutton.setOnClickListener(new View.OnClickListener() {
112     @Override
113     public void onClick(View v) {
114         bheight.setText("");
115         bweight.setText("");
116         age.setText("");
117         male.setChecked(true);
118     }
119 });
120
121 dietbutton.setOnClickListener(new View.OnClickListener() {
122     @Override
123     public void onClick(View v) {
124         float ht = Float.parseFloat(bheight.getText().toString())/100;
125         float wt = Float.parseFloat(bweight.getText().toString());
126         bmi = wt / (ht * ht);
127         if (bmi <= 18.5) {
128             Intent t1 = getActivity().getIntent();
129             String s = Float.toString(calorieintake);
130             t1.putExtra("calories", s);
131             setFragment(underWeightFragment);
132         } else if (bmi <= 24.9) {
133             Intent t1 = getActivity().getIntent();
134             String s = Float.toString(calorieintake);
135             t1.putExtra("calories", s);
136             setFragment(normalWeightFragment);
137         } else {
138             Intent t1 = getActivity().getIntent();
139             String s = Float.toString(calorieintake);
140             //Log.d("bmi is","value"+calorieintake);
141             t1.putExtra("calories", s);
142             //startActivityForResult(t1, 0);
143             setFragment(overWeightFragment);
144         }
145     }
146
147     private void setFragment(Fragment fragment) {
148         FragmentTransaction fragmentTransaction = getActivity().
149             getSupportFragmentManager().beginTransaction();
150         fragmentTransaction.replace(R.id.diet_frame, fragment);
151         fragmentTransaction.addToBackStack(null);
152         fragmentTransaction.commit();
153     }
154 });
155
156 return view;
157 }
158
159
160 //Calculate BMR
161 private float calculateBMR (float h, float w, float a) {
162     if (female.isChecked()){
163         bmrWomen = 655.1 + (9.563 * w) + (1.85 * h) - (4.676 * a);
164         bmrWomen = Math.round(bmrWomen);
165         return (float) bmrWomen;
166     }
167     else {

```

```
168     bmrMen = 66.47 + (13.75 * w) + (5.003 * h) - (6.755 * a);
169     bmrMen = Math.round(bmrMen);
170     return (float) bmrMen;
171 }
172 }
173
174 //Calculate calories
175 private float calculateCalories(float bmr){
176     calorieintake = bmr * (float) 1.55;
177     calorieintake = Math.round(calorieintake);
178     return calorieintake;
179 }
180 }
```



Chapter 7

System Testing

System testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

7.1 Test Cases and Test Results

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Login Page	Matching the correct pattern	Login successfully	Opens home page
T02	Calculate BMI	Valid Input	Shows BMI	Result of BMI
T03	Exercise Recommendation	User performing exercise	Real time Feedback	Please correct your posture

7.2 Sample of a Test Case

Title: Login Page – Authenticate Successfully on gmail.com

Description: A registered user should be able to successfully login at gmail.com.

Precondition: the user must already be registered with an email address and password.

Assumption: a supported browser is being used.

Test Steps:

1. Navigate to gmail.com
2. In the 'email' field, enter the email of the registered user.
3. Click the 'Next' button.

4. Enter the password of the registered user
5. Click 'Sign In'

Expected Result: A page displaying the gmail user's inbox should load, showing any new message at the top of the page.

Actual Result:

Write here description
upload the image of result

7.2.1 Software Quality Attributes

1. Availability-1 : The system shall be available to users all the time.
2. Availability-2 : The system shall always have something to function and always pop up error messages in case of component failure.
3. Efficiency-1 : The system shall generate the correct output with an accuracy of 80
4. Efficiency-2 : The system shall provide the right tools to support all its features.

Chapter 8

Screenshots of Project

8.1 pose estimation module

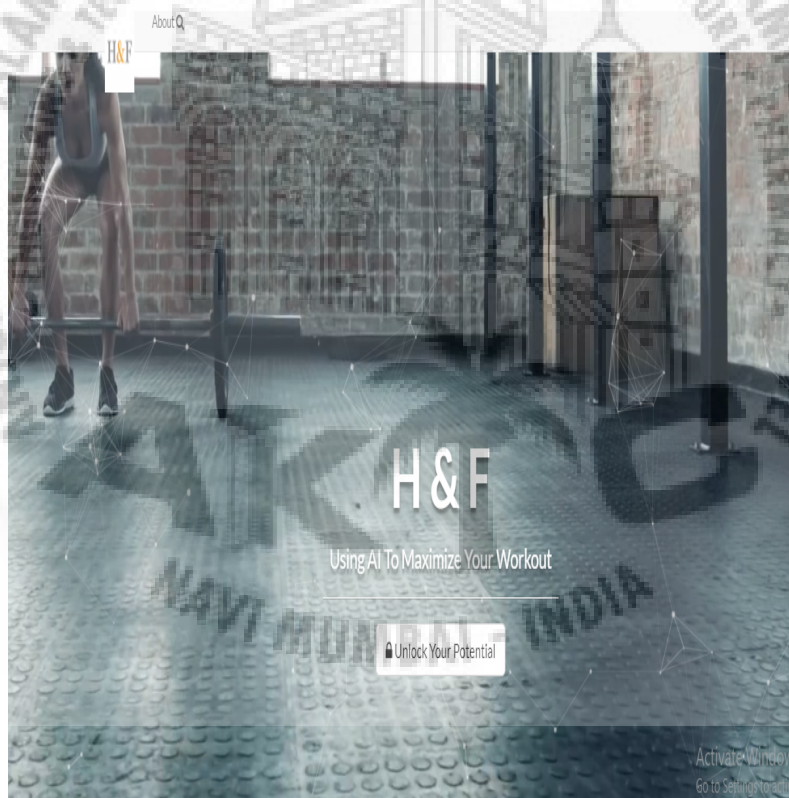


Figure 8.1: Home page

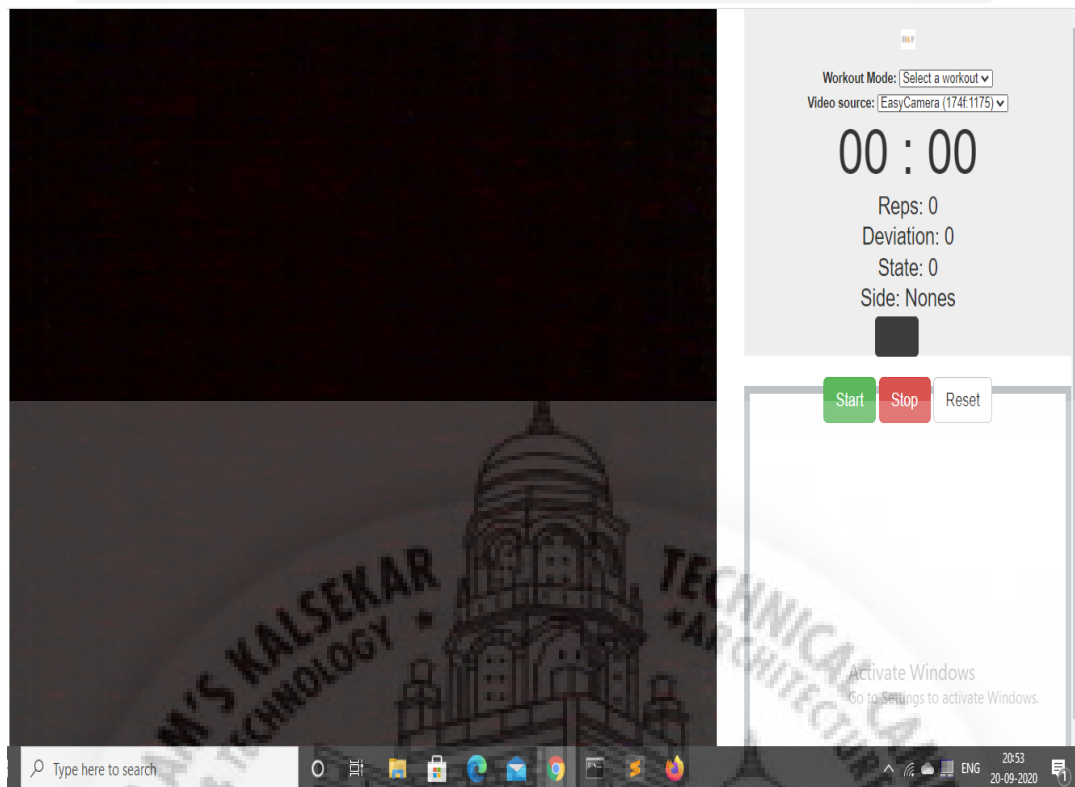


Figure 8.2: Training page

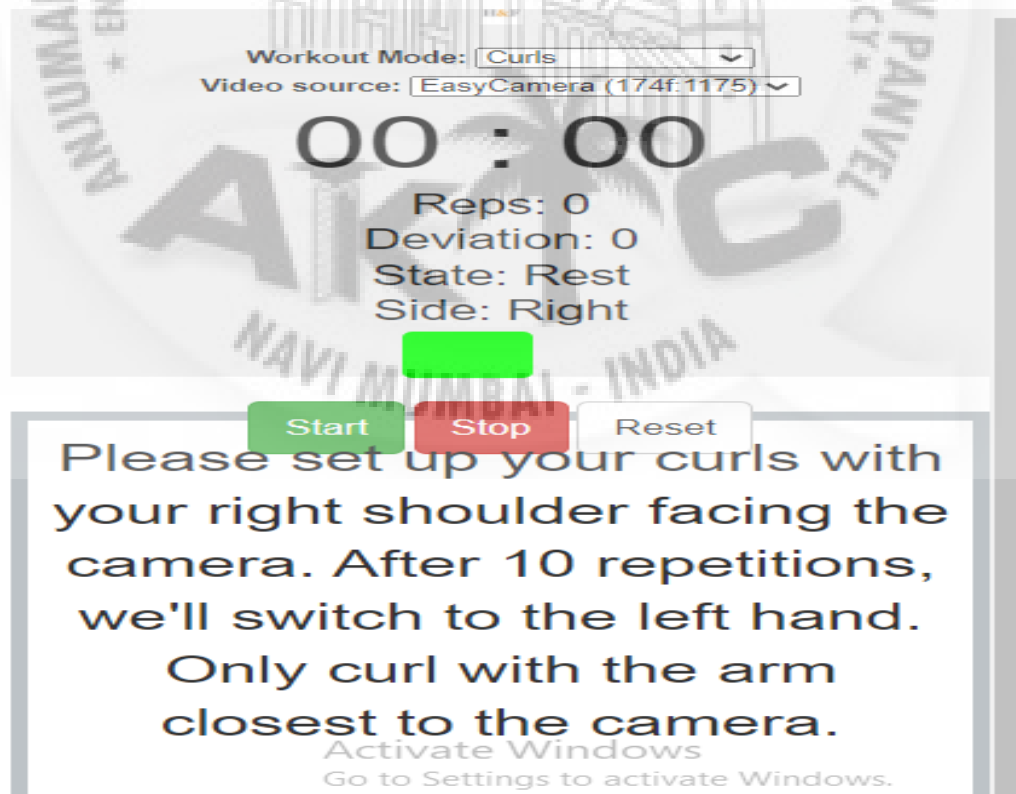


Figure 8.3: Workout page

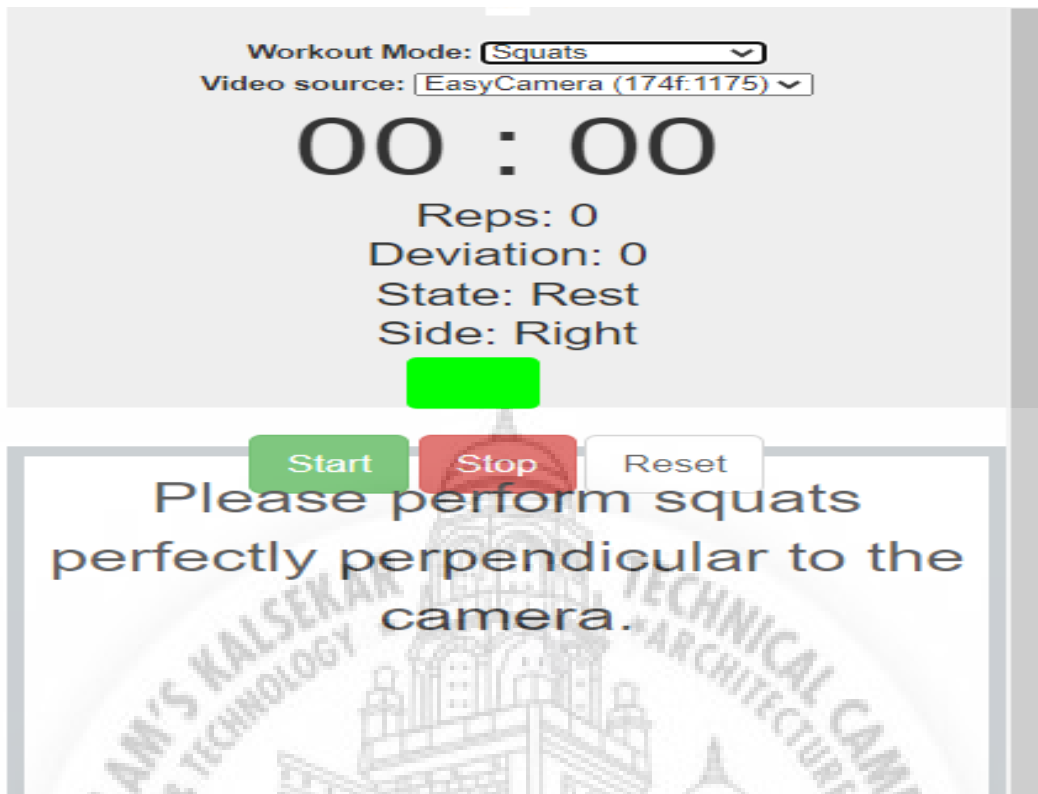


Figure 8.4: Body detection page

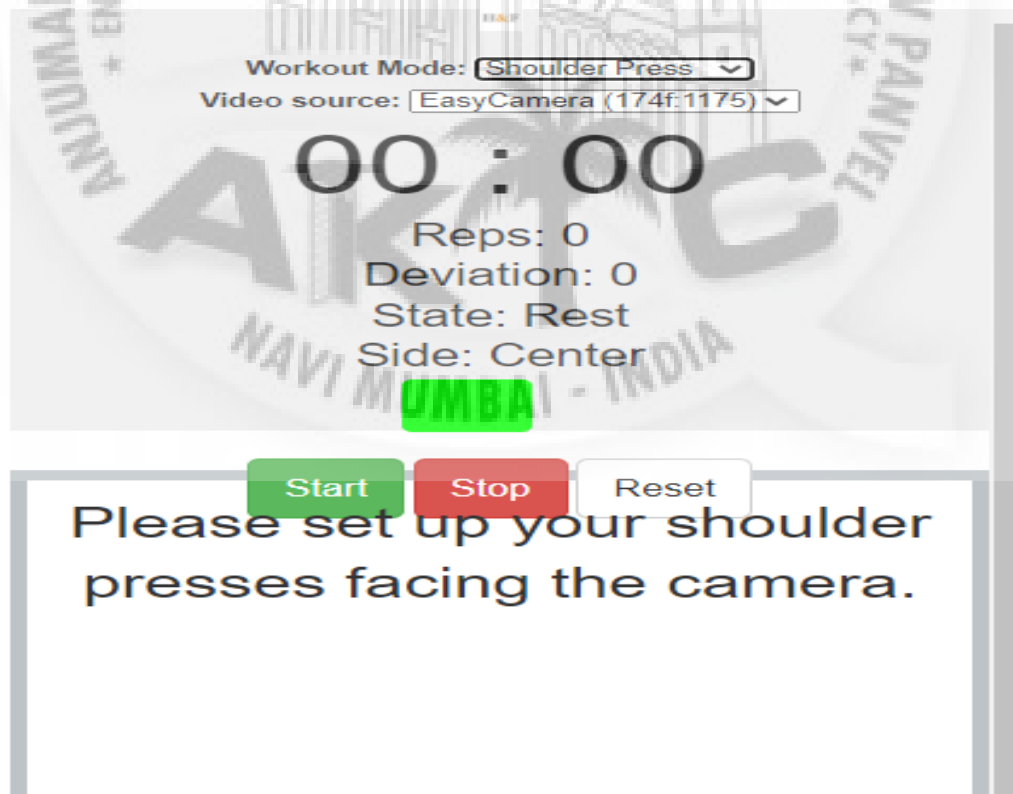


Figure 8.5: Performing Exercise page

8.2 Diet module



Figure 8.6: Registration

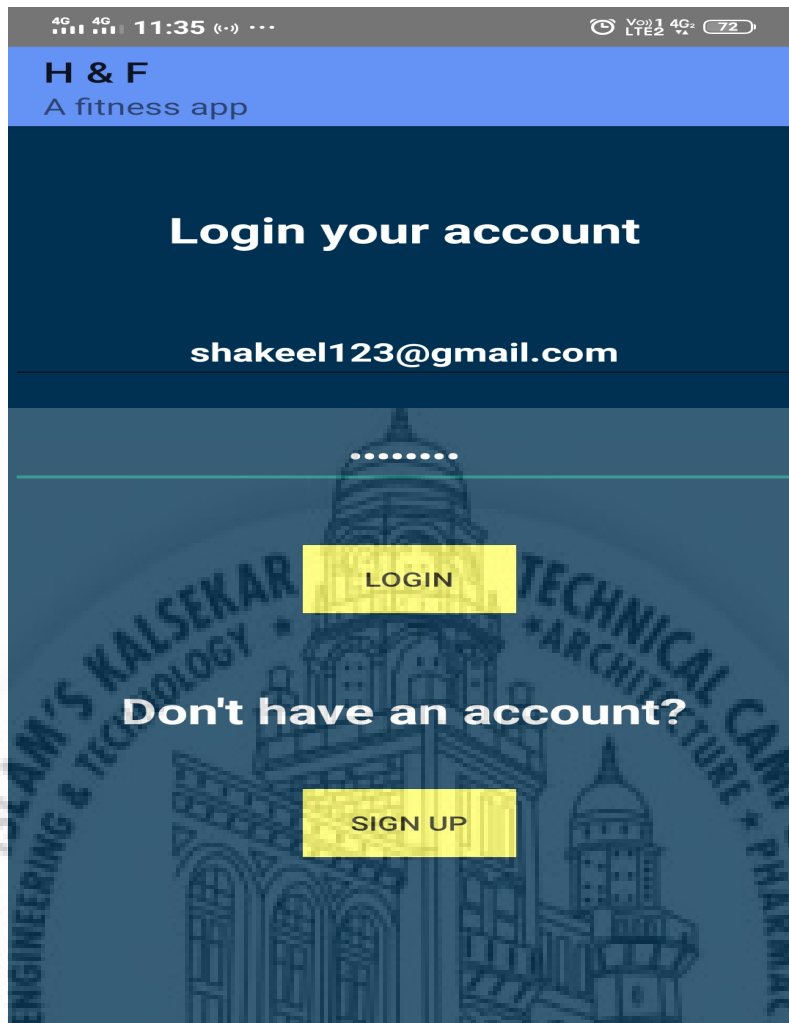


Figure 8.7: Login



Figure 8.8: BMI calculation



Figure 8.9: categorising The User



Figure 8.10: Selecting Diet Plan



Figure 8.11: Selecting Diet Plan

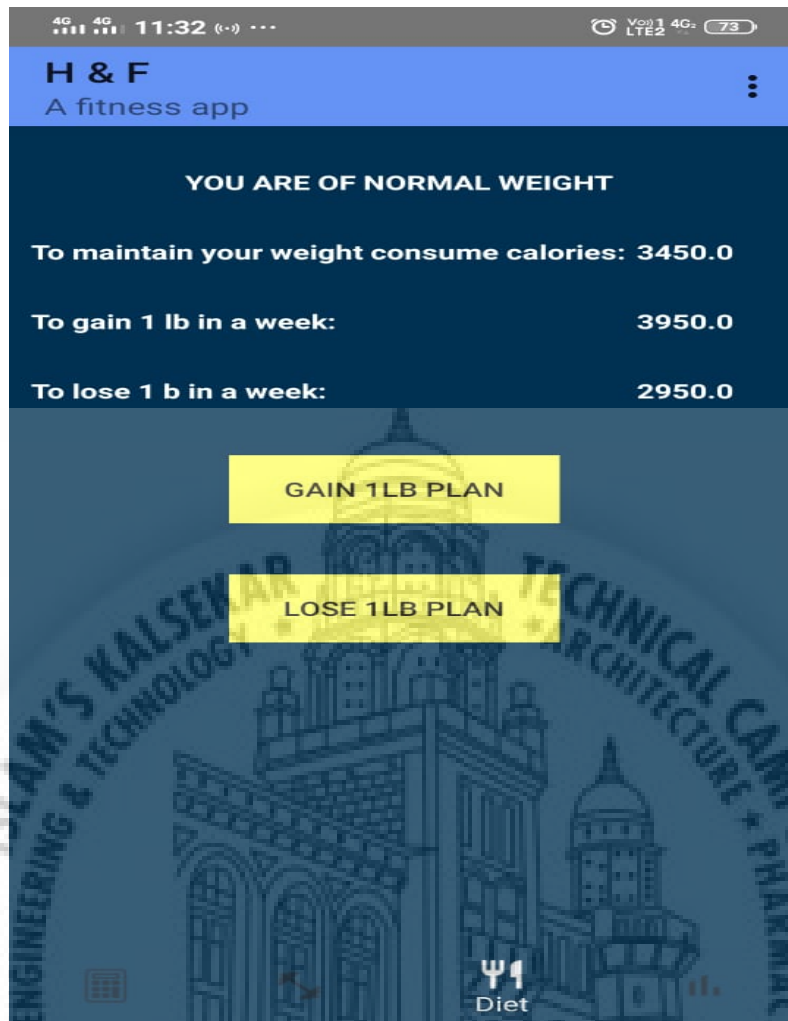


Figure 8.12: Displaying Calories



Figure 8.13: Displaying calories



Figure 8.14: Displaying Diet Plan

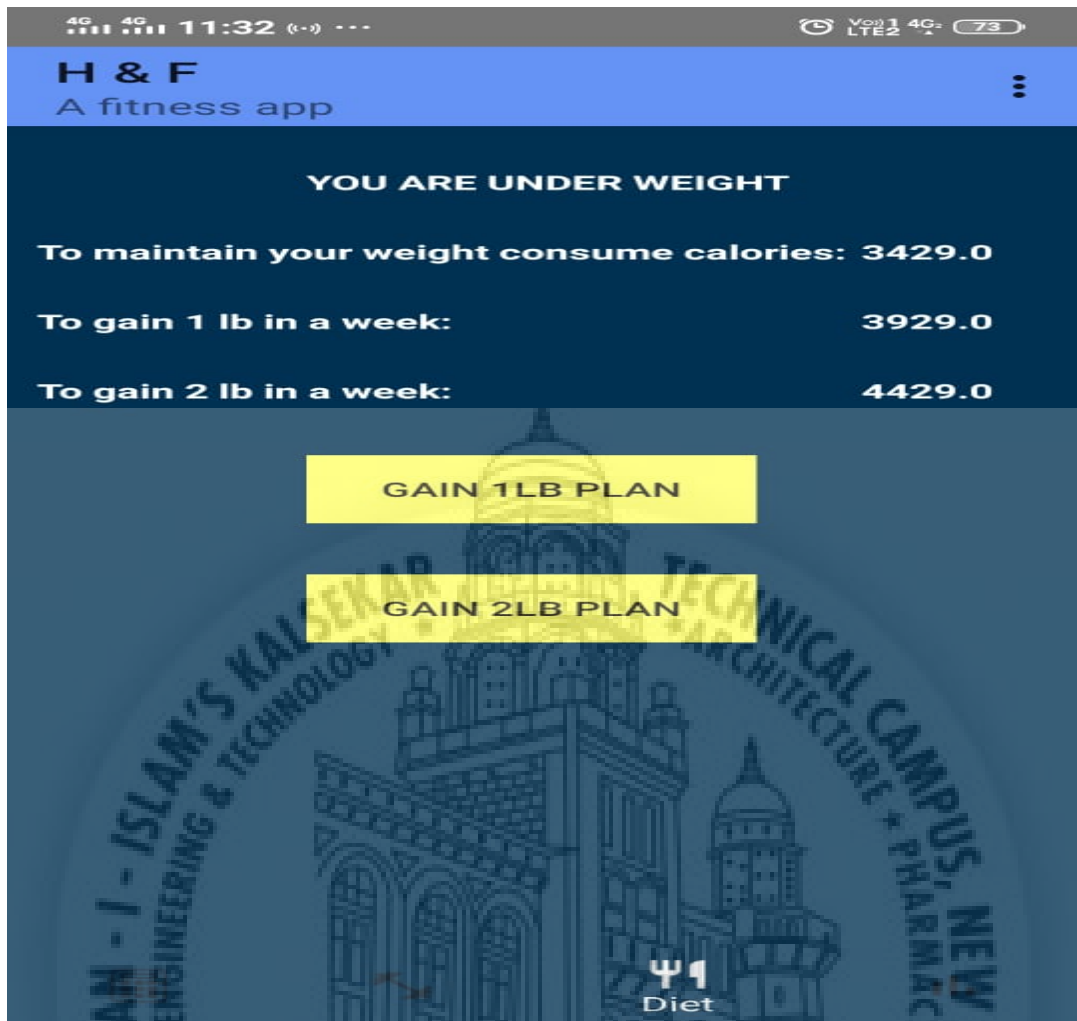


Figure 8.15: Displaying Diet Plan

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

In this world of technology, smartphones play an important role for many daily needs. The number of different applications installed by people is very large. Some are social apps, some are data sharing apps, some are games and there are many. Also, in this fast paced life, problem of not paying attention to health is rising. Some are getting slim, some are getting fat, some are having body pain problems and much more.

The Android application developed in this project is able to provide many useful features to the user to keep a track of his/her fitness. It recommends different diet and exercise plan on monthly and daily basis. When user performs exercise it gives alert in the form of audio feedback. User progress is shown in the form of graph. There are many improvements possible in the application in future.

So, from this project, it can be concluded that the problem has thereby reduced to some extent. However, different other modules can also be included for more advanced system. The future improvements of the application can be features like integration of fitness module with fitness band. A fitness band is a wearable device which can track various parameters of a person like calories consumed, heart rate, distance travelled and quality of sleep. This project is overall a good project to provide simply diet and exercise recommendations from which an

individual can be benefited.

It can be concluded that this project provides an answer or resolution of the problems related to an individual's health.

9.2 Future Scope

- Suggestion of what he/she has to include or exclude from their diet, Supplementary information like daily/monthly exercise, BMI (Body Mass Index).
- calculator, A full workout program and monitors it, Alert the user when move is performed incorrectly.
- Add live recording a particular exercise which a person wants to wish do it later.

References

- [1] *m-Health: A Complete Healthcare Solution*; Rohan V. Vaidya, Darshan K. Trivedi, ICCMC, 2017
- [2] *Android Based Nutritional Intake Tracking Application for Hand-held Systems*; Tewari, Dinika Saxena, Disha Sahani, Preetanjali Ray, ICCCNT, 2017
- [3] *SmartCoach Personal Gym Trainer- An adaptive modified back-propagation approach*; Roger Achkar, Rita Geagea, Hussam Mehio, Wael Kmeish, IMCET, 2016
- [4] *Android-based Exercise Application*; Goh Kim Nee, Muhammad Syazwan Bin Abu Bakar, ICCIS, 2012
- [5] Documentation *NAME OF site*;
<https://firebase.google.com/docs/>, July, 2019
- [6] Introduction to Android *NAME OF site*;
<https://developer.android.com./guide/index>, July, 2019
- [7] Recommendation system in python using ALS algorithm and Apache Spark *NAME OF site*;
<https://medium.com/@patelneha1495/recommendation-system-in-python-using-als-algorithm-and-apache-spark-27aca08eaab3>, July, 2019
- [8] Human pose estimation using OpenPose with TensorFlow (Part 2) *NAME OF site*; <https://arvrjourney.com/human-pose-estimation-using-openpose-with-tensorflow-part-2-e78ab9104fc8>, August, 2019

Achievements

1. Publications

- (a) *Fitness Monitoring Using Machine Learning*; Shaikh Altamas Shakeeel, Shaikh Shafaque Naushad, Ulde Fahmi Nisar, journal Name, month and year of published(<http://EXAMPLE.com>)

2. Conferences

- (a) *Fitness Monitoring Using Machine Learning*; Shaikh Altamas Shakeeel, Shaikh Shafaque Naushad, Ulde Fahmi Nisar, ICICN , DEC and 2019 of attend(Venue : Xi' an China)

3. Project Competitions

- (a) *Fitness Monitoring Using Machine Learning*; Shaikh Altamas Shakeeel, Shaikh Shafaque Naushad, Ulde Fahmi Nisar, Fourth National Level Tech Fest Connect-2020 , 22nd January 2020(Venue :Jamia Institute of Engineering, Akkalkuwa)