

# **ANDROID BASED DIET CONSULTANT**

**By**

**ESHITA YEASMIN MUNMUN**

**ID: 151-15- 4747**

This report presented in partial fulfillment of the requirements for  
The degree of Bachelor of Science in Computer Science and Engineering.

Supervised by

**Moushumi Zaman Bonny**

Senior Lecturer

Department of CSE

Daffodil International University

Co-supervised by

Lecturer

Department of CSE

Daffodil International University



**Daffodil International University**

**Dhaka, Bangladesh**

**May 2019**

## **APPROVAL**

This project titled “**Android based Diet Consultant**”, submitted by Eshita Yeasmin Munmun, ID: 151-15-4747, to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. In Computer Science and Engineering and approved as to its style and contents. The presentation has been held on May,04,2019.

## **BOARD OF EXAMINERS**

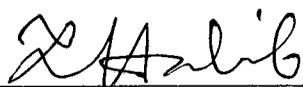


**Dr. Syed Akhter Hossain**

**Professor and Head**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Chairman**



**Md. Tarek Habib**

**Assistant Professor**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**



**Moushumi Zaman Bonny**

**Senior Lecturer**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**



**Dr. Md. Saddam Hossain**

**Assistant Professor**

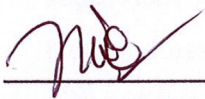
Department of Computer Science and Engineering  
United International University

**External Examiner**

## DECLARATION

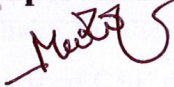
I hereby declare that, this project has been done by me under the supervision of **Moushumi Zaman Bonny, Senior Lecturer, Department of CSE** in Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

**Supervised by:**



**Moushumi Zaman Bonny**  
Senior Lecturer  
Department of CSE  
Daffodil International University

**Co-supervised by:**



Lecturer  
Department of CSE  
Daffodil International University

**Submitted by:**



**Eshita Yeasmin Munmun**  
Id:151-15-4747  
Department of CSE  
Daffodil International University

## ACKNOWLEDGEMENT

First, I express my heartiest thanks and gratitude to almighty Allah for his divine blessing makes me possible to complete this project successfully.

I feel grateful to **Moushumi Zaman Bonny, Senior Lecturer**, Department of CSE, Daffodil International University, Dhaka. Deep knowledge & passionate interest of my supervisor in the field of software development influenced me to carry out this project. Her endless endurance, scholastic guidance, continuous encouragement, constant and diligent supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to **Professor Dr. Syed Akhter Hossain**, Head and all Faculty members of Department of CSE, Daffodil International University, Dhaka for their kind help to finish our project and also to other faculty members and the staffs of CSE department of Daffodil International University.

Finally, with due respect I must acknowledge the constant support and patience of my parents.



## **ABSTRACT**

To stay healthy, we have to maintain some rules for our daily life. In this app it is shown that a complete health toolkit which will give us a proper guide to stay claim all day long with a fittest body. It contains some health tips those are scientifically proven. It has a BMI calculator which can calculate proper BMI of a human body. Which is very important to know every person that is he healthy or not. It has calories calculator to calculate proper calories of any with which are consumed by user. It also has a service which contains the doctors list. Anyone can see any doctor's profile from the list. A user can call or SMS his doctor. Every doctor will be shown by his designation. Users can also get doctor's location. There a user also can find a list of all members who are registered in this app. All of them can communicate with each other that they may know about progress of each other. They can also chat with doctors. It is also shown a list of exercises. Users can see video of any exercise from this list. Users have facility to track all of their health record by this. It has an alarm system which will notify user his meal time with alarm followed by the meal plan.

## TABLE OF CONTENTS

<b>CONTENS</b>	<b>PAGE</b>
Approval	i
Board of examiners	ii
Declaration	iii
Acknowledgements	iv
Abstract	v
 <b>CHAPTER</b>	
 <b>CHAPTER 1: INTRODUCTION</b>	 <b>9-10</b>
1.1 Introduction	9
1.2 Motivation	9
1.3 Objectives	9
1.4 Expected Outcome	10
1.5 Report Layout	10
 <b>CHAPTER 2: BACKGROUND</b>	 <b>11-14</b>
2.1 Introduction	11
2.2 Literature Review	11
2.2.1 MVP	11
2.2.2 Repository Pattern	11
2.2.3 Singleton Pattern	12

2.2.4 NoSQL Database (Firebase)	12
2.2.5 Rx Java	12
2.2.6 Data Binding	12
2.2.7 SQLite	12
2.3 Comparative Studies	13
2.4 Scope of The Problem	13-14
<b>CHAPTER 3: REQUIREMENT AND SPECIFICATION</b>	<b>15-20</b>
3.1 Introduction	15
3.2 Methods And Algorithm	15-17
3.3 Data Flow Model	18
3.4 ER Diagram	19
3.5 Use Case Model	20
<b>CHAPTER 4: DESIGN SPECIFICATION</b>	<b>21-32</b>
4.1 Front-End Design	21-28
4.2 Back-End Design	29-31
4.3 Implementation Requirements	32
<b>CHAPTER 5: IMPLEMENTING AND TESTING</b>	<b>33-38</b>
5.1 Implementation of Database	33
5.2 Model	34-35
5.3 View	36-37
5.4 Testing Implementation	38
5.4.1 Acceptance Testing	38

5.5 Test results and Reports	38
5.5.1 System Testing	38
5.5.2 User Acceptance Testing	38
<b>CHAPTER 6: CONCLUSION AND FUTURE SCOPE</b>	<b>39</b>
6.1 Discussion And Conclusion	39
6.2 Scope for Further Developments	39
<b>REFERENCES</b>	<b>40-41</b>
<b>APPENDIX</b>	<b>42-49</b>
<b>PLAGIARISM REPORT</b>	<b>50</b>

## **LIST OF FIGURES**

3.3 Data flow diagram	18
3.4 ER diagram	19
3.5 Use case model	20
4.1 (A) Sign in Page	21
4.1 (B) Sign up Page	21
4.2 BMI calculation	22
4.3 Food alarm	23
4.4 Health tips list	24
4.5 Workout video	25
4.6 Health tips details page	26
4.7 Doctor list	27
4.8 Front End Design Sample Code	28
4.9 Database Sample of Consultant's Information	29
4.10 Retrieving and Assigning Data	30
4.11 Calculation of BMI	31
5.1 Data structure of NoSQL database	33
5.2 Main Dashboard	35
5.3 View with Adapter pattern	37

# **CHAPTER 1**

## **Introduction**

### **1.1 Introduction**

Health is Wealth. If any person is not healthy he cannot play role of his daily life properly. So to stay productive and healthy we have to focus on our health.

The full system is concerned on user's health. We are human being always distracted by nature so to stay focus on our health, this project will be a very good assistant of a user to keep good health.

### **1.2 Motivation**

We all are so much busy in our daily life that we get no free time to do something special for our health to be fit. But in the last age of our life we feel the value of a good health but it is then so late to realize.

To keep proper data about health a lot of people are facing death. The main motivation of this project is to keep a proper record of health data with every aspect.

### **1.3 Objectives**

After study it is got that some factor which are most responsible for health is diet guideline, BMI report [1], calories intake measurement. So it is included here with a good user experience. It is also got after a study that lack of proper information of a doctor around a lot of patients faces trouble and go to fake doctors to get treatment which leads to the patient to death. It is also costly to hire a health trainer for everyone which is also solved by this app because it has a complete workout guideline

## **1.4 Expected Outcome**

It has some good outcome in future if it is used greatly. They are-

The system can be used as a personal health trainer. The system can be used as a diet instructor. The system can be suggested by doctor to their patient to keep their health good. The system can also be used for health education. This system can be used by government to give proper information about right doctors. This system can be used by hospital to give the patient proper calories by calculating calories. This system can be used to plan everyday meal.

## **1.5 Report layout**

The present study has divided into six chapters where the very Chapter 1, I have discussed about the introduction, motivation of the work, objectives and expected outcome of the research work and the report layout. In Chapter 2, I discussed about the background circumstances of our work. I also delivered the literature review, comparative studies and scope of the problem and challenges of the system. In Chapter 3, This Chapter is all about the procedure used to build the system. This section has the methods and steps of the proposed system. In this Chapter 4, all graphical interface has shown and display the system design. In this Chapter 5, all the experimental result that has been achieved by the proposed system is discussed along with the performance analysis and a summary of the result is covered. Chapter 6 contains the conclusion part and the ideas of implication of further study on this topic.



## **CHAPTER 2**

### **Background**

#### **2.1 Introduction**

Health is more valuable for a person than the money. Because money cannot buy good health if someone suffers from unbeatable disease however, using wealth one can maintain his/her health.

The full system is concern on user's health

The auxiliary purpose about this project is to implement real life software engineering with their best practice.

#### **2.2 Literature Review**

Some algorithms and techniques of modern software engineering have been developed by past researchers. This section is briefly discussed.

##### **2.2.1 MVP**

MVP consists of terms MODEL, VIEW and PRESENTER [2]. Models are the proper entities of real world object. In this section, database related code is written. Views are the main module in which user interactions are written it also contains the presentation logic. Finally, presenters are responsible for all business logic which links between model and view.

##### **2.2.2 Repository Pattern**

Repository pattern is used here to be separated the Model from rapid change of ORM because I know that ORM is updated most frequently. So keep them interdependence from Model is a very good practice to database operation. It has a data access layer which is very helpful to communicate with database.

### **2.2.3 Singleton Pattern**

Sometimes it is not acceptable to create a duplicate user object for a real life single user. So to solve this situation I have to create a good structure where I can create only one user object.

### **2.2.4 NoSQL Database (Firebase)**

Firebase is a new technology by Google which use NoSQL database underneath [7]. It has all facilities which are needed for a developer like as authentication, push notification, data storage, real time database etc.

### **2.2.5 RX Java**

To do some asynchronous task I need to do reactive programming in this project, So I needed to use RX java the reactive extension of java in this project [3].

### **2.2.6 Data Binding**

To make the UI implementation very easy, I use data binding in this project which makes the UI in a declarative way so that UI updates its structure without no coding interfere and reduce the Boiler plate code [11].

### **2.2.7 SQLite**

To keep offline data here I used a portable database called SQLite database [12]. SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files.

## 2.3 Comparative Studies

I have to do real life software engineering in this project. So I follow Agile methodology in this project. I firstly studied about health information. I have to get all possible proper information about good health and their factor. Then I did proper analysis from the raw data. Then I have to design the system. I have to get knowledge about the proper architecture for this project which can make this project more testable and maintainable. I also have to choose a correct testing framework to test every module.

## 2.4 Scope of the Problem

The following table shows the expected ow of work for the accomplishment of the required result.

### Time scheduling:

Analysis	2 Month
Design of the system	1 Month
Coding	3 months
Testing and implementing	2 Weeks
Total	6 Months and 14 days

So, in this time I complete the whole task. But that actually took more time to complete each task for many circumstances. At last I did complete our project not with time but with targets.

## 2.5 Challenges

To work with new technologies with this project I face a lot of challenges with it. One of them is lack of communities support and Ill documentations.

### Time Scheduling:

It was the difficult challenge because it was the main thing in which our project will be completed. If I am unable to complete on time it would be great hazard for me. So, I just divide my time and project work to complete each one of the tasks to complete the whole project.

### **Poor Communication:**

For completing our project, I have faced different types of problems and every time I have discussed our problem with our supervisor. She has given the right decision for completing our project properly.

### **Skills for the Project:**

A project sometimes requires skills that by the project's go on. So I had needed some skills and I gained those skills for completing our project and I had discussed about all the topic and took a decision for proper work.

## **CHAPTER 3**

### **Requirement and Specification**

#### **3.1 Introduction**

In previous section I illustrated the different techniques and methods of software engineering. Each category of procedure performs Ill in specific criteria and has difficulties as Ill. The precision and accuracy levels are still very far in the system. The observation of the field of view is proposed for the following architecture identification and software systems.

#### **Requirement:**

I need to know about

1. java,
2. XML,
3. Third party library.
4. Android studio

I can use another IDE for total development process, to test and debug I need emulators and Android debug tools [4].

#### **3.2 Methods and Algorithm**

##### **Diet Tips**

A user can show any diet tip from a list. If a user clicks any of the items, then he will go to the details page.

## **The Pseudo Code:**

### **Read Tips**

- i. Click Diet Tips Button.
- ii. Show list of tips.
- iii. Click any of the items
- iv. Go to the details tips activity.
- v. Come back to the Home activity.

### **BMI Calculator**

A user can calculate his BMI with this app [1]. He needs to input his height and Weight with a good interactive UI.

- i. Input Feet
- ii. Input inch
- iii. Input weight
- iv. System auto calculate the BMI
- v. Show a result of BMI.
- vi. Show how much weight to gain or loss.
- vii. Click suggested button of meal plan
- viii. Get meal plan chart
- ix. Set Alarm
- x. Reset the UI

### **C Calculator**

A user can calculate calories with the 'C Calculator' where there are some foods are shown. You can input food amount with the specific food item and get the Calories which contains.

- i. Select food
- ii. Select amount
- iii. Get Calories amount
- iv. Reset UI

## **Food Alarm**

A user can set some food alarm which will notify him what he has to eat on meal planned time.

- i. Set food
- ii. Set time.
- iii. System include time to the system alarm manager
- iv. Alarm counter start
- v. Reset alarm it wants.

## **Services**

There are collections of doctor on user nearby. If a user clicks any of the doctor's button, then he will be got all information about the doctor [14].

- i. Select doctor.
- ii. Call doctor.
- iii. SMS doctor.
- iv. Get Address of doctor.
- v. Chat with doctor.

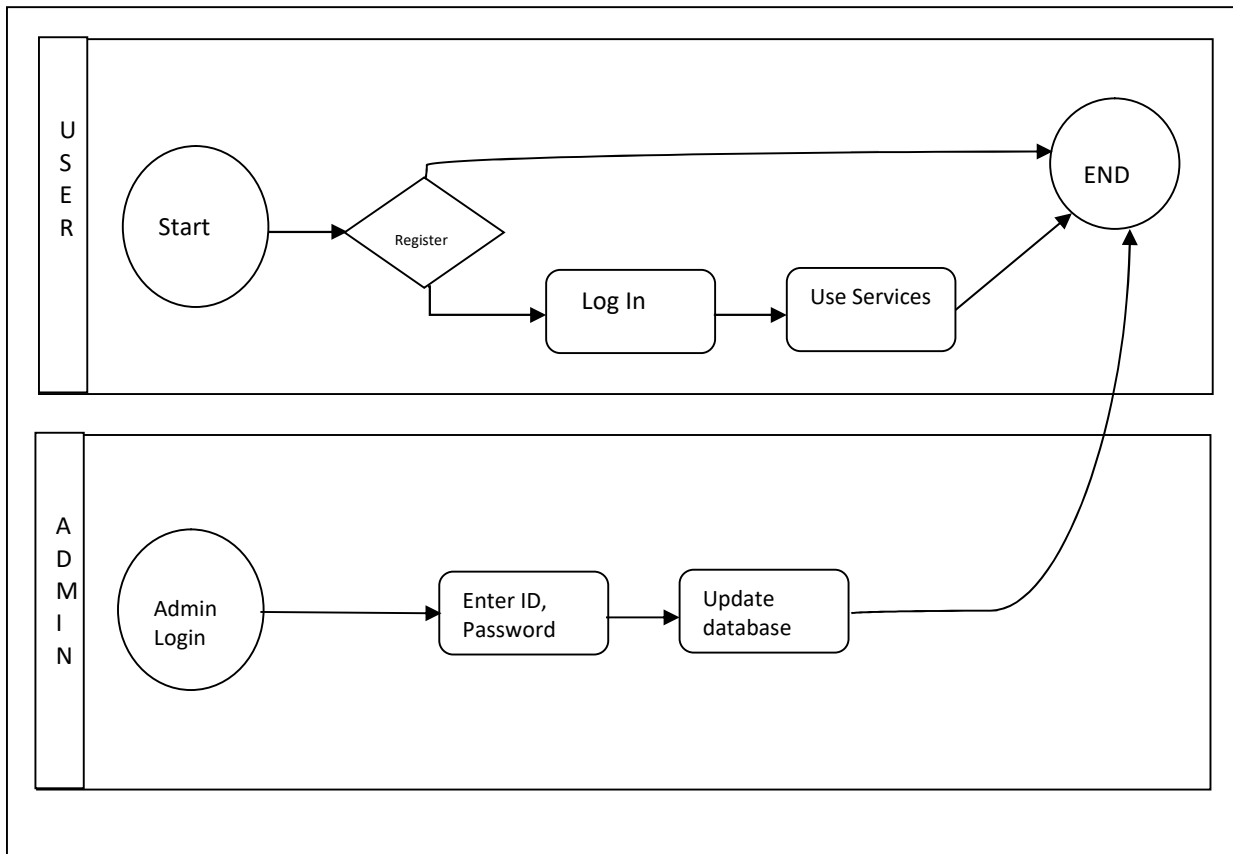
## **Exercise**

A user can see all exercises with in a single place with video control panel.

- i. Select Exercise part.
- ii. Go to video list
- iii. Select any video from there
- iv. See video an exercise by following them
- v. Return to home



### 3.3 Data Flow Diagram



**Figure 3.3 Data Flow Model.**

### 3.4 ER Diagram

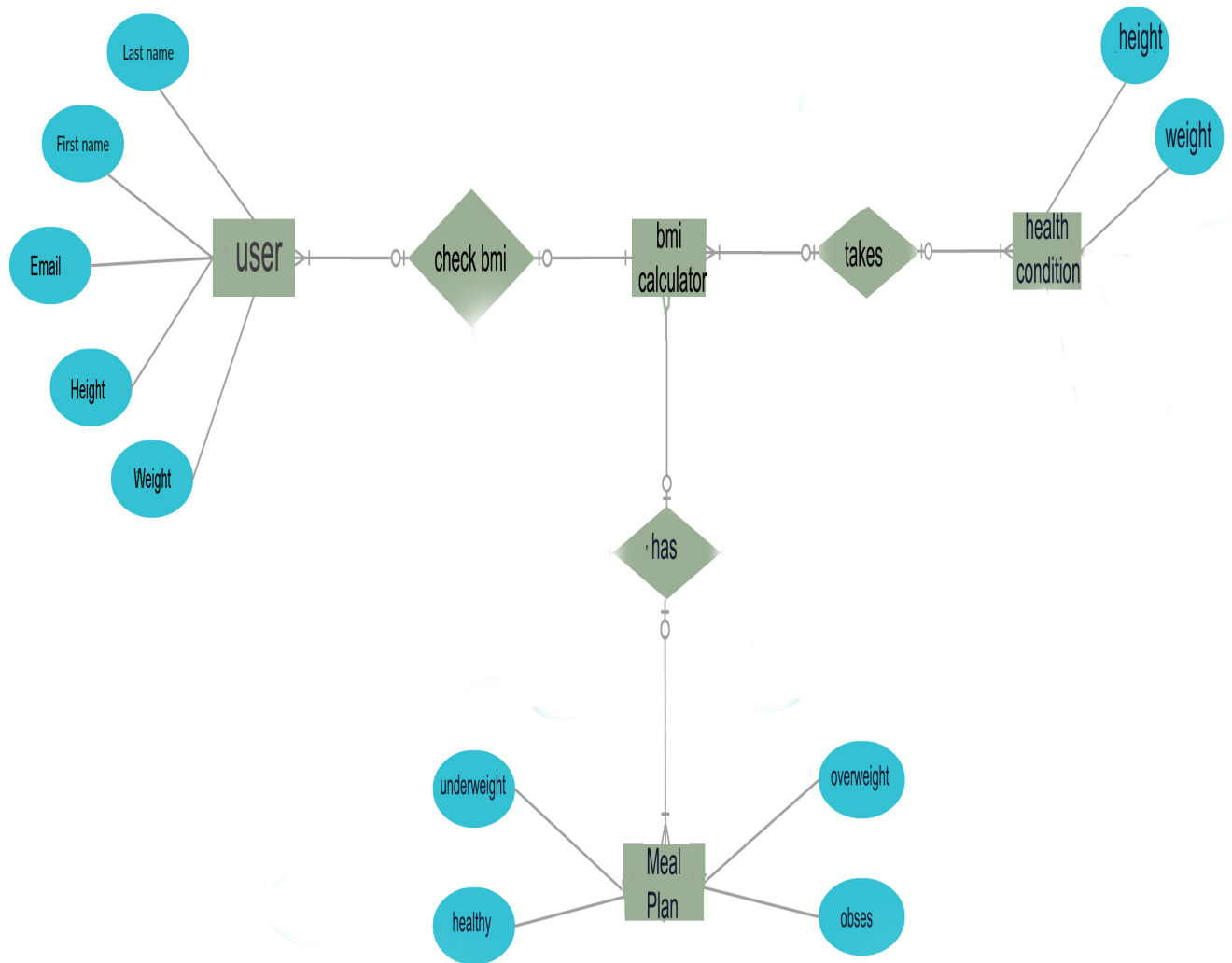
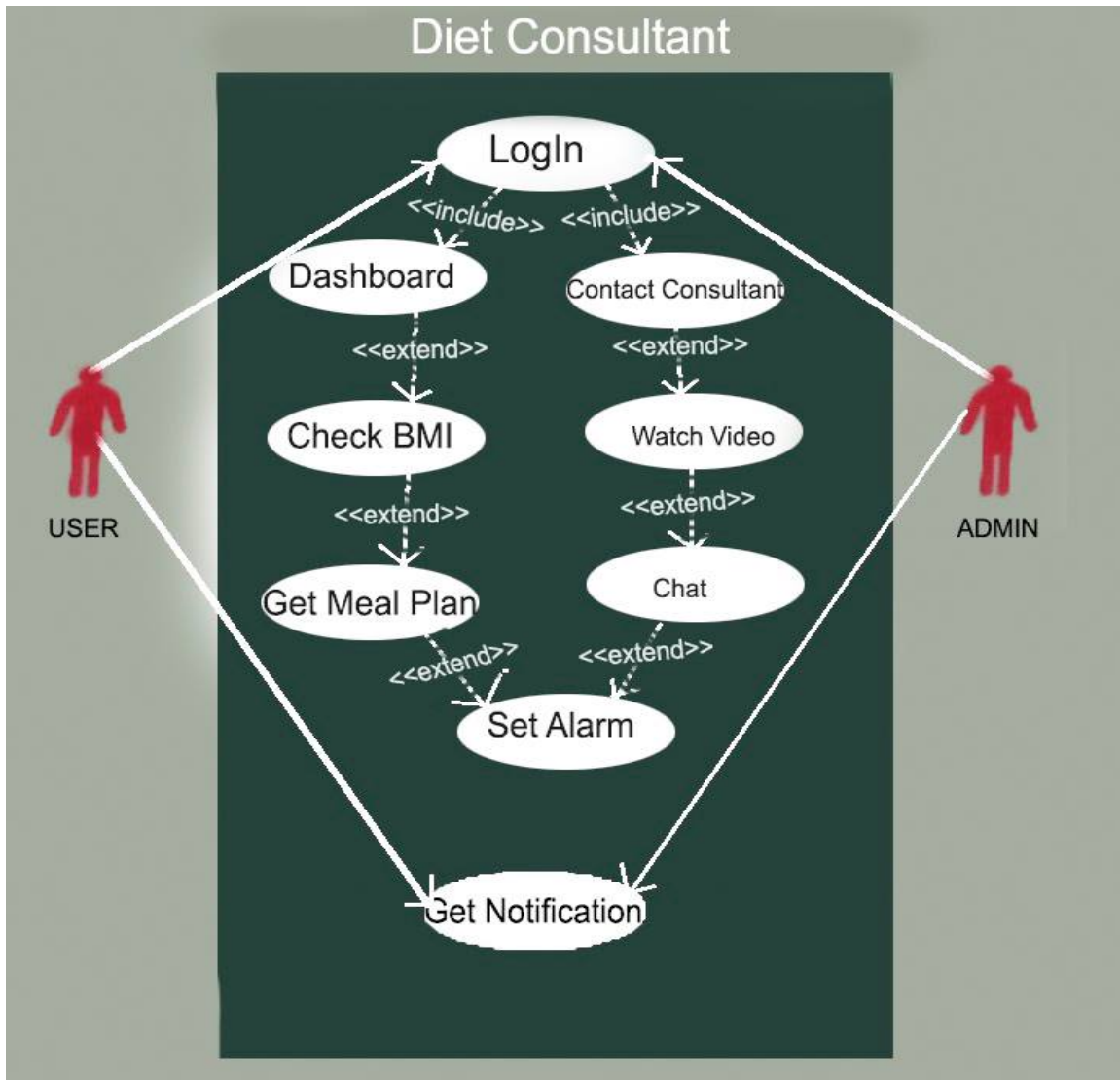


Figure 3.5 ER Diagram

### 3.6 Use Case Diagram



**Figure 3.6 Use Case Diagram.**

In this use case diagram, we can see that two actors- 1) Admin 2) Normal user. Normal user login first of all. Then he can see dashboard, check BMI, get meal plan, set alarm, chat with doctor, watch video and contact consultant. If it is admin then an admin can modify content of the app [17].

## CHAPTER 4

### Design Specification

#### 4.1 Front-End Design

Actually in front end design we used xml which is very flexible language to design UI instead of programming approach. I also use some third party library to give an attractive look of the UI. I research on some color psychology to select good color suit to it. We also some API like Youtube API to preview video in the app.

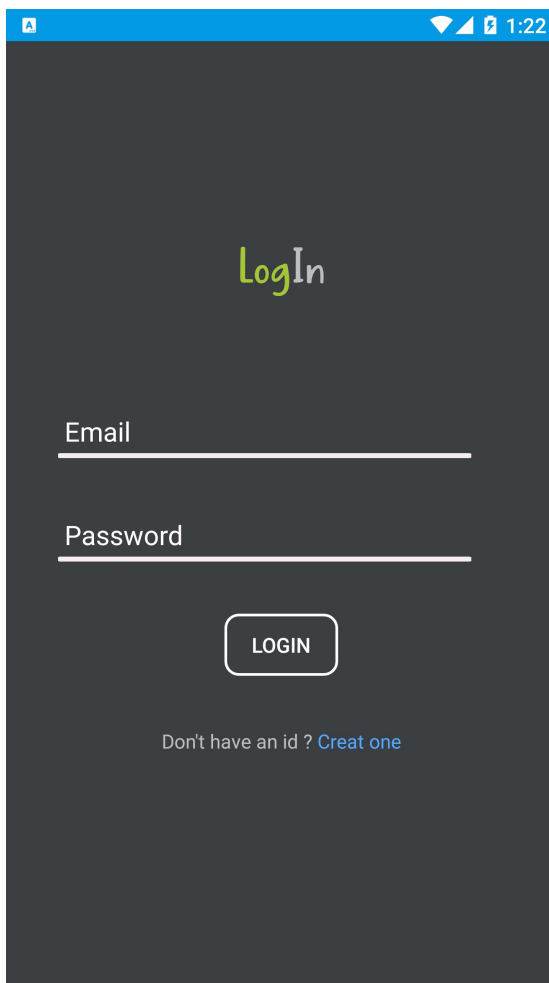


Figure 4.1 (A) Sign in Page.

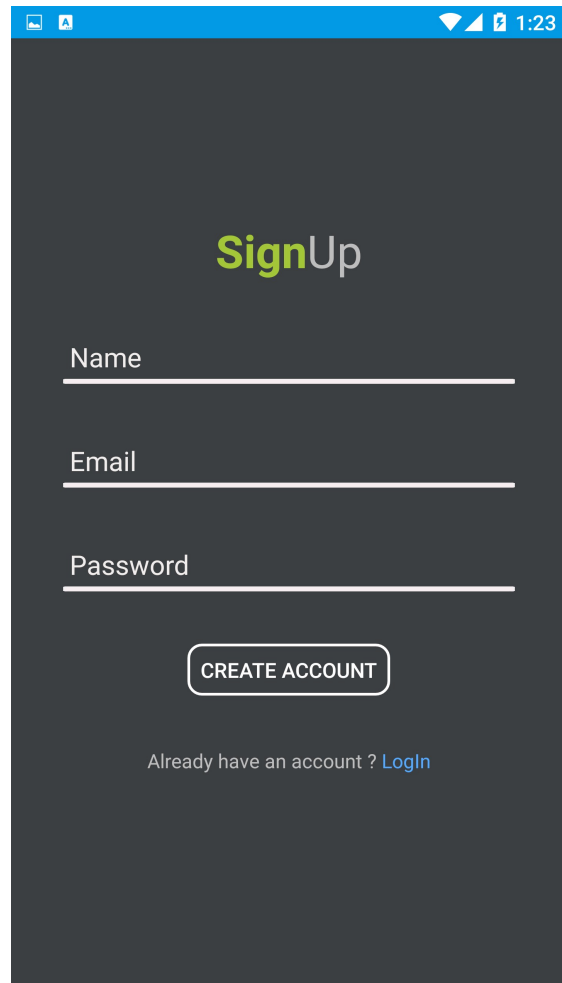
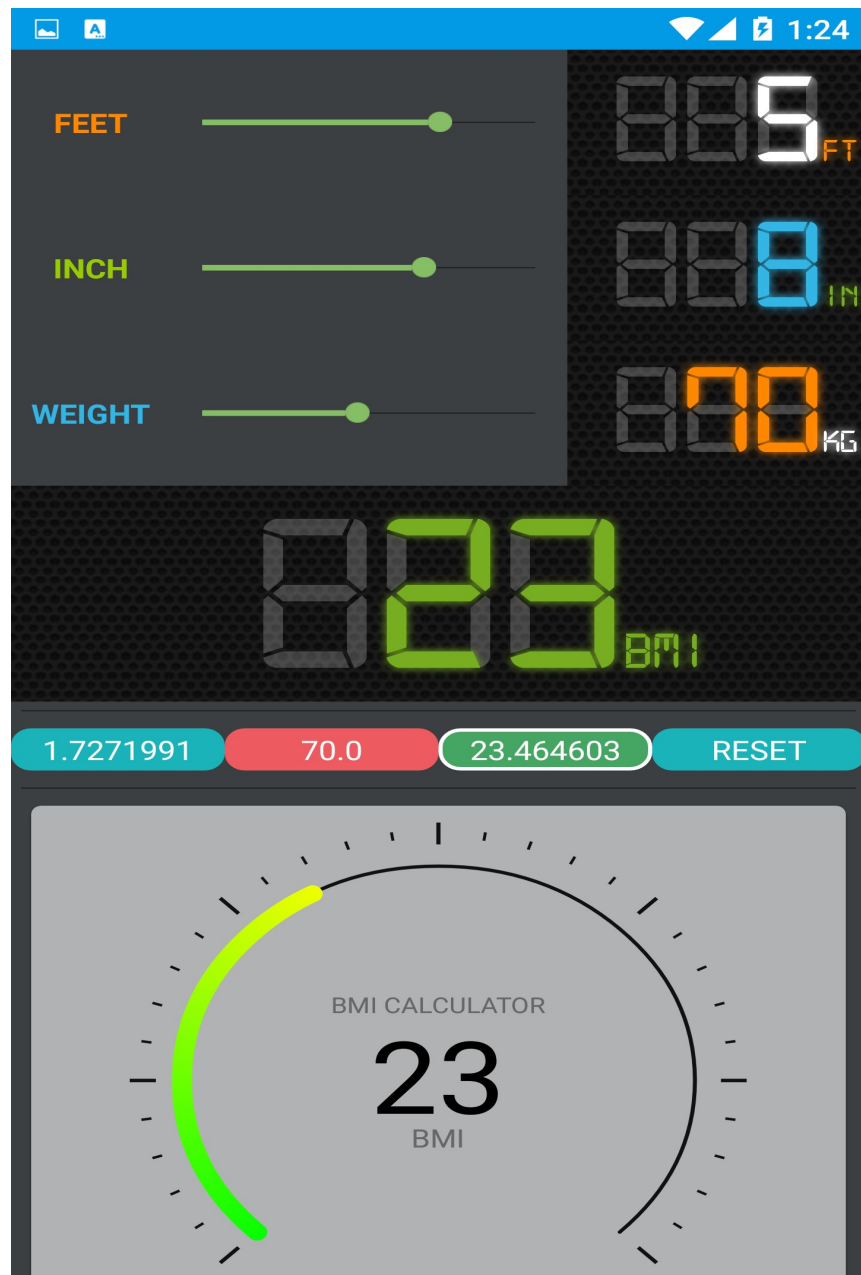
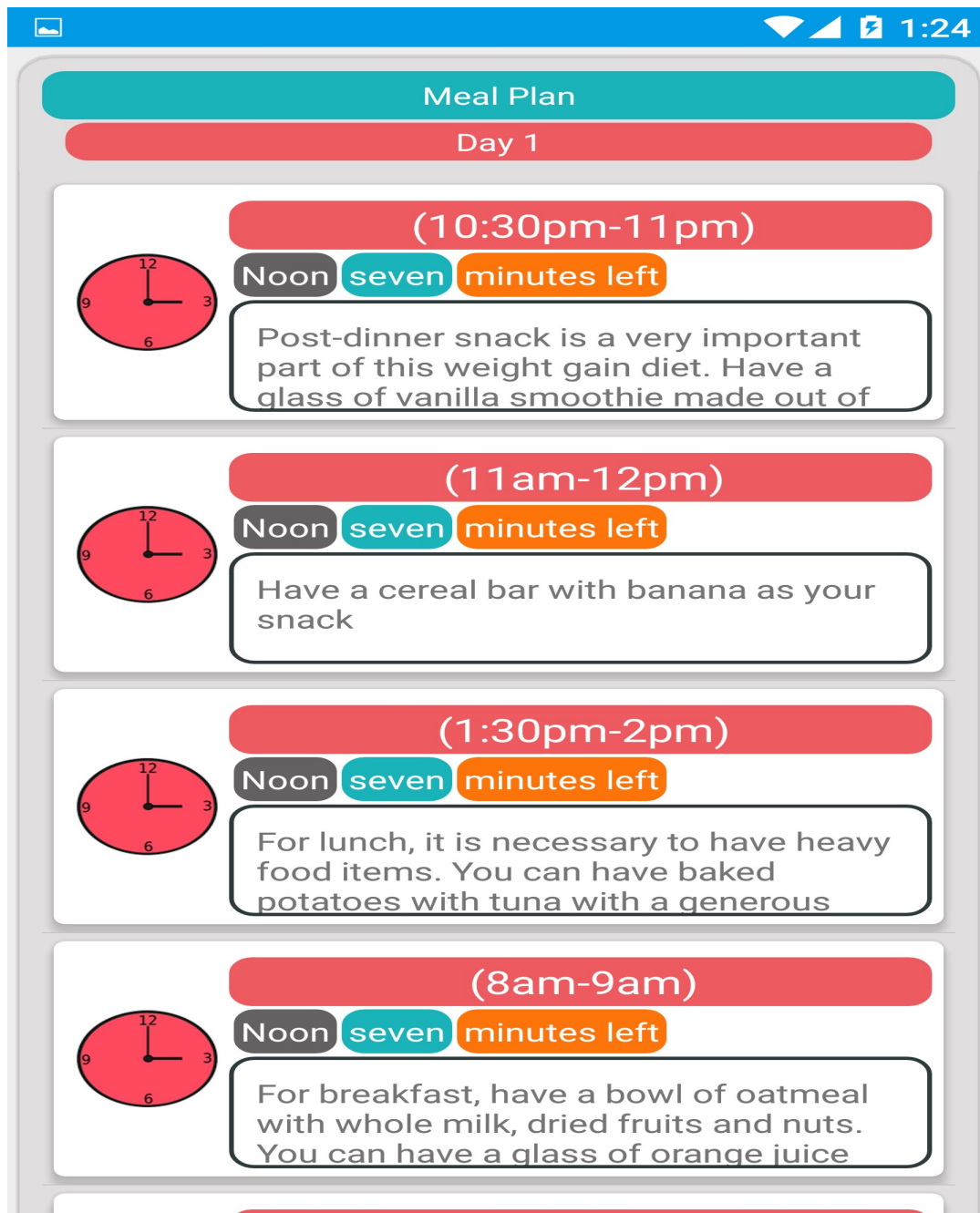


Figure 4.1 (B) Sign up Page.



**Figure 4.2 BMI Calculation.**



**Figure 4.3 Food Alarm**



**Figure 4.4 Health Tips List.**



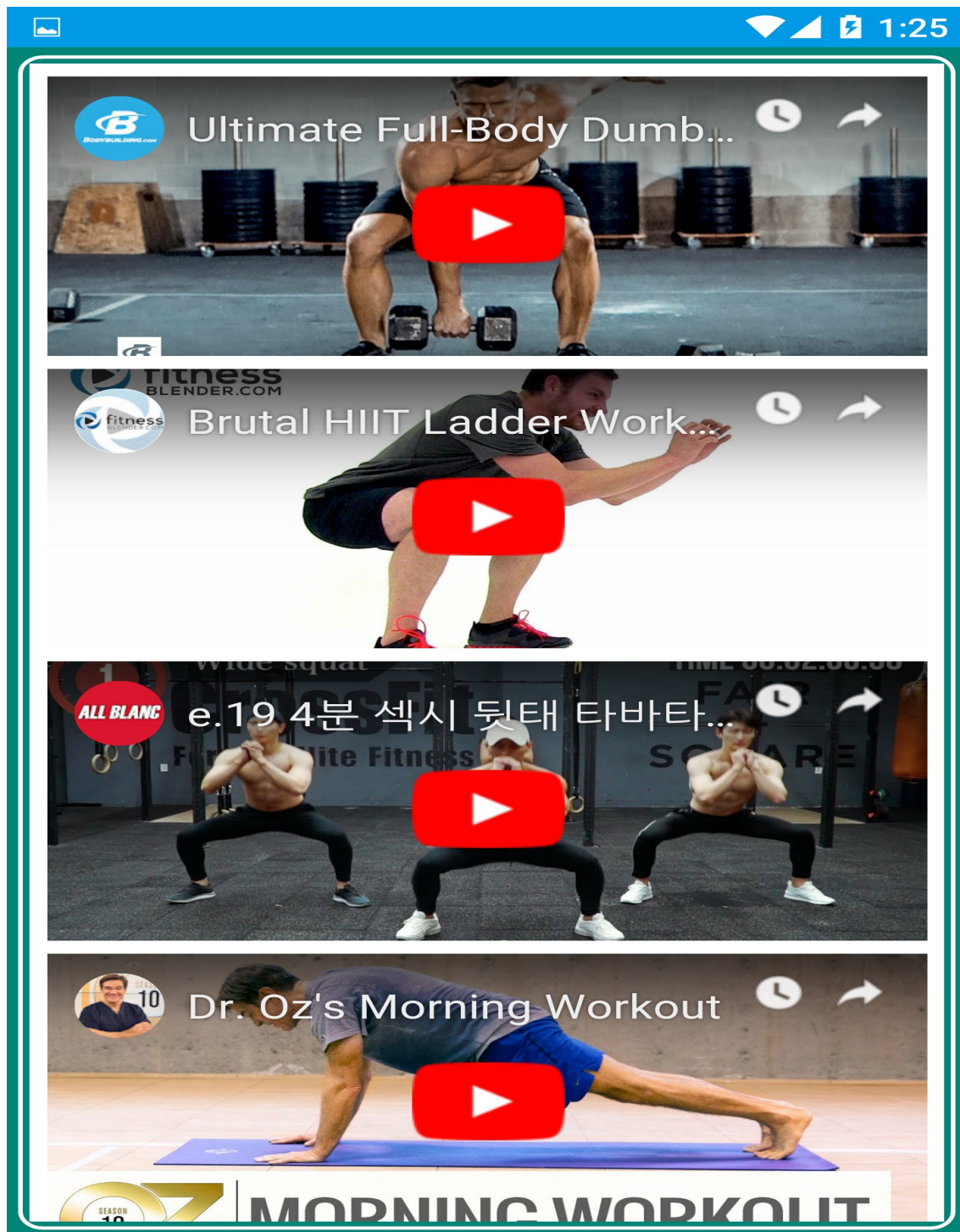
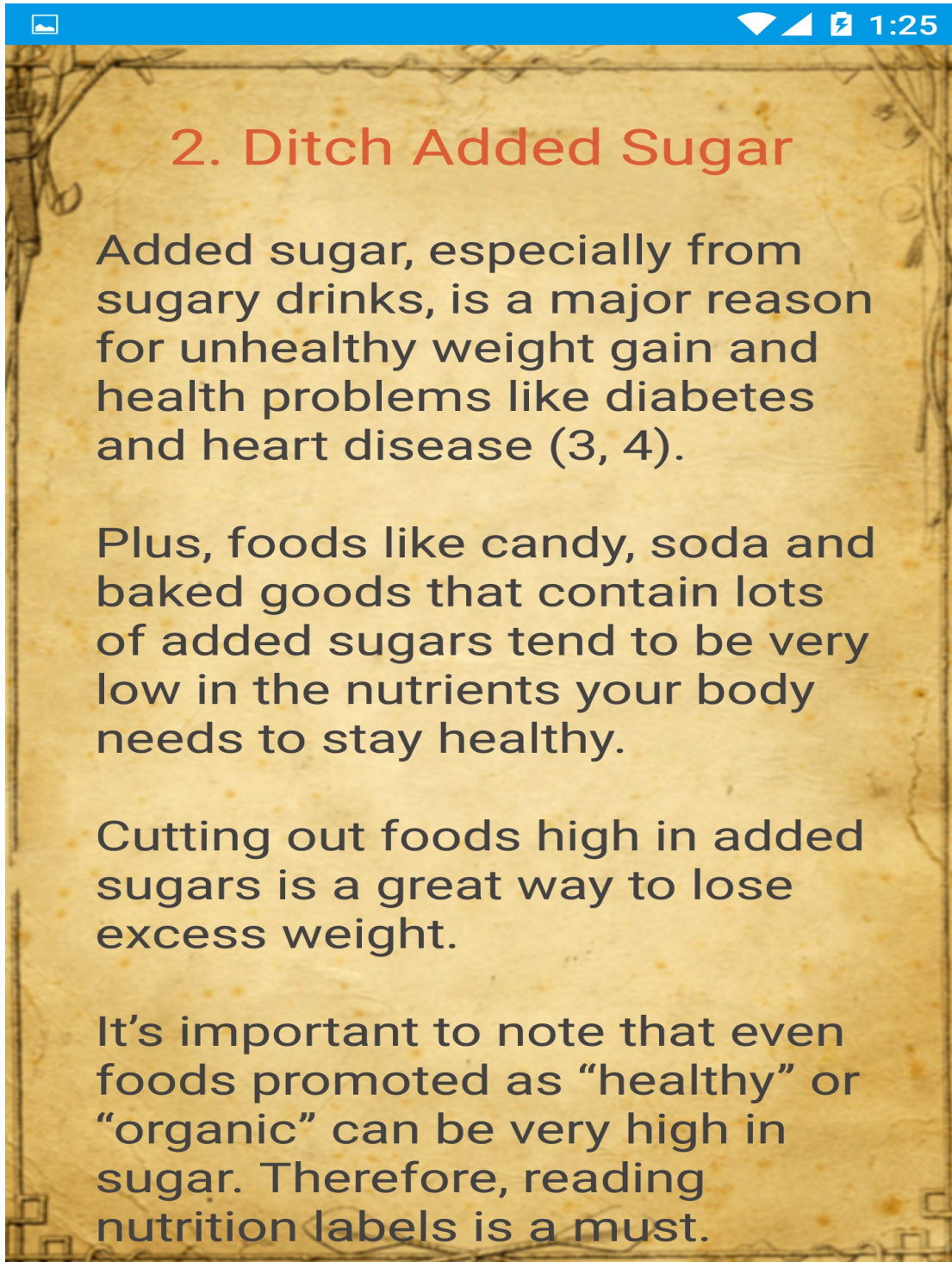
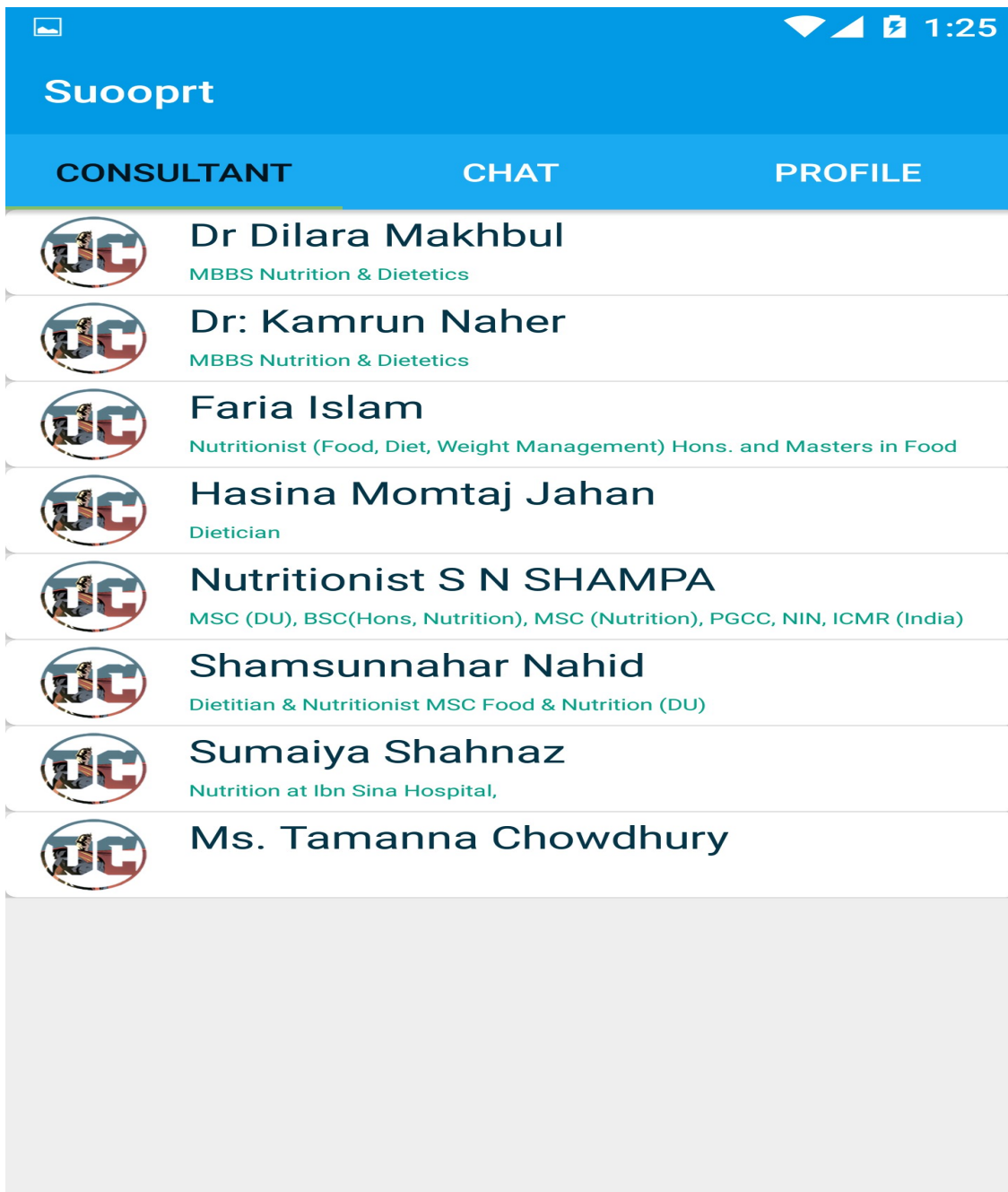


Figure 4.5 Workout Video.



**Figure 4.6 :Health Tips Details Page.**



**Figure 4.7: Doctor List.**





**Figure 4.8: Front End Design Sample Code**

## 4.2 Back-End Design

In back-end I actually used Firebase No SQL database where everything is stored as JSON object. I need to first define the access rules. Then need to configure database in the project. Firebase has online offline support together. I just need structure a tree of data. When the collection of data is formed then firebase has some method to send data to firebase. There firebase stores data as parent and child. There I also used SQLite database to store some data locally. SQLite is a portable database. I can store data by query. After retrieving data from firebase I need to display those data to an attractive pattern. So there I used list view and recycler list view [6]. For listing data to view I used Base Adapter and recycler list view adapter. To display plain text, I used text view and for video I used web view. In this app I created a chat system which was controlled by firebase and recycler list view on real time [8].



Figure 4.9: Database Sample of Consultant's Information

```

reff.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        int day = 1;

        for (DataSnapshot childSnapshot : dataSnapshot.getChildren()) {

            String oldtemp = childSnapshot.getKey();
            //text.append(oldtemp+"\n");
            switch (day) {

                case 1:
                    for (DataSnapshot childSnapshot1 : childSnapshot.getChildren()) {
                        String temp = childSnapshot1.getKey();
                        //data.add(temp);
                        //text.append(temp+" : "+childSnapshot.child(temp).getValue().toString()+"\n");
                        String desc = childSnapshot.child(temp).getValue().toString();
                        if (temp != null && desc != null) {
                            list1.add(new AlarmModel(temp, desc));
                            System.out.println(temp + " " + desc);
                        }
                    }
                    break;
            }
        }

        setAlarm.setOnClickListner((view) -> {
            Calendar calendar = Calendar.getInstance();

            for(int i=1;i<=7;i++){
                //startAlarm(i,time+=50000);
                switch (i){
                    case 1:
                        int alert1 = 1;
                        for(AlarmModel model:list1){

                            int hour,min;

                            String timeWithRegex = model.getTime();
                            String timeWitRegexToCharArray [] = timeWithRegex.split( regex: "[ ]");
                            if(timeWitRegexToCharArray[0].toLowerCase().contains("am")){
                                timeWitRegexToCharArray[0] = timeWitRegexToCharArray[0].replaceAll( regex: "am", replacement: "");
                                timeWitRegexToCharArray[0] = timeWitRegexToCharArray[0].replaceAll( regex: "\\(", replacement: "");

                                if(timeWitRegexToCharArray[0].contains(":")){

                                    String hourAndMin [] = timeWitRegexToCharArray[0].split( regex: "[ ]");
                                    hour = Integer.parseInt(hourAndMin[0]);
                                    min = Integer.parseInt(hourAndMin[1]);

                                }else{
                                    hour = Integer.parseInt(timeWitRegexToCharArray[0].replace( target: " ", replacement: ""));
                                    min = 00;
                                }

                                //Toast.makeText(HtmlView.this,"time is "+timeWitRegexToCharArray[0],Toast.LENGTH_SHORT).show();
                            }else{
                                timeWitRegexToCharArray[0] = timeWitRegexToCharArray[0].replaceAll( regex: "pm", replacement: "");
                                timeWitRegexToCharArray[0] = timeWitRegexToCharArray[0].replaceAll( regex: "\\(", replacement: "");

                                if(timeWitRegexToCharArray[0].contains(":")){

                                    String hourAndMin [] = timeWitRegexToCharArray[0].split( regex: "[ ]");
                                    hour = Integer.parseInt(hourAndMin[0].replace( target: " ", replacement: ""));
                                    min = Integer.parseInt(hourAndMin[1].replace( target: " ", replacement: ""));

                                }else{
                                    hour = Integer.parseInt(timeWitRegexToCharArray[0].replace( target: " ", replacement: ""))+12;
                                    min = 00;
                                }

                                // Toast.makeText(HtmlView.this,"time is "+timeWitRegexToCharArray[0],Toast.LENGTH_SHORT).show();
                            }
                            String desc = model.getDesc();

```

**Figure 4.10: Retrieving and Assigning Data**

```

public float calculateBMI(float height, float weight) {
    feet.setText(String.valueOf(height));
    w.setText(String.valueOf(weight));
    int wgoal = 0;

    float bmi = weight / (height * height);

    int ibmi = (int)bmi;
    if(ibmi <= 19){
        moonMoonDisabler( c 'u');
        wgoal = (int) ((weight/ibmi)*22)-weight;
        status.setText("you are underweight\nyou have to increase "+wgoal+" kg");
        status.setTextColor(Color.parseColor( colorString: "#FF495E"));
    }else if(ibmi > 19 && ibmi <= 25){
        moonMoonDisabler( c 'h');
        status.setText("you are Healthy");
        status.setTextColor(Color.parseColor( colorString: "#439700"));
    }else if(ibmi >25 && ibmi <= 29){
        moonMoonDisabler( c 'o');
        wgoal = (int) (weight-((weight/ibmi)*22));
        status.setText("you are overweight\nyou have to decrease "+wgoal+" kg");
        status.setTextColor(Color.parseColor( colorString: "#FF495E"));
    }else if(ibmi > 29){
        moonMoonDisabler( c 'b');
    }
    return bmi;
}

```

**Figure 4.11: Calculation of BMI**



#### **4.4 Implementation Requirements**

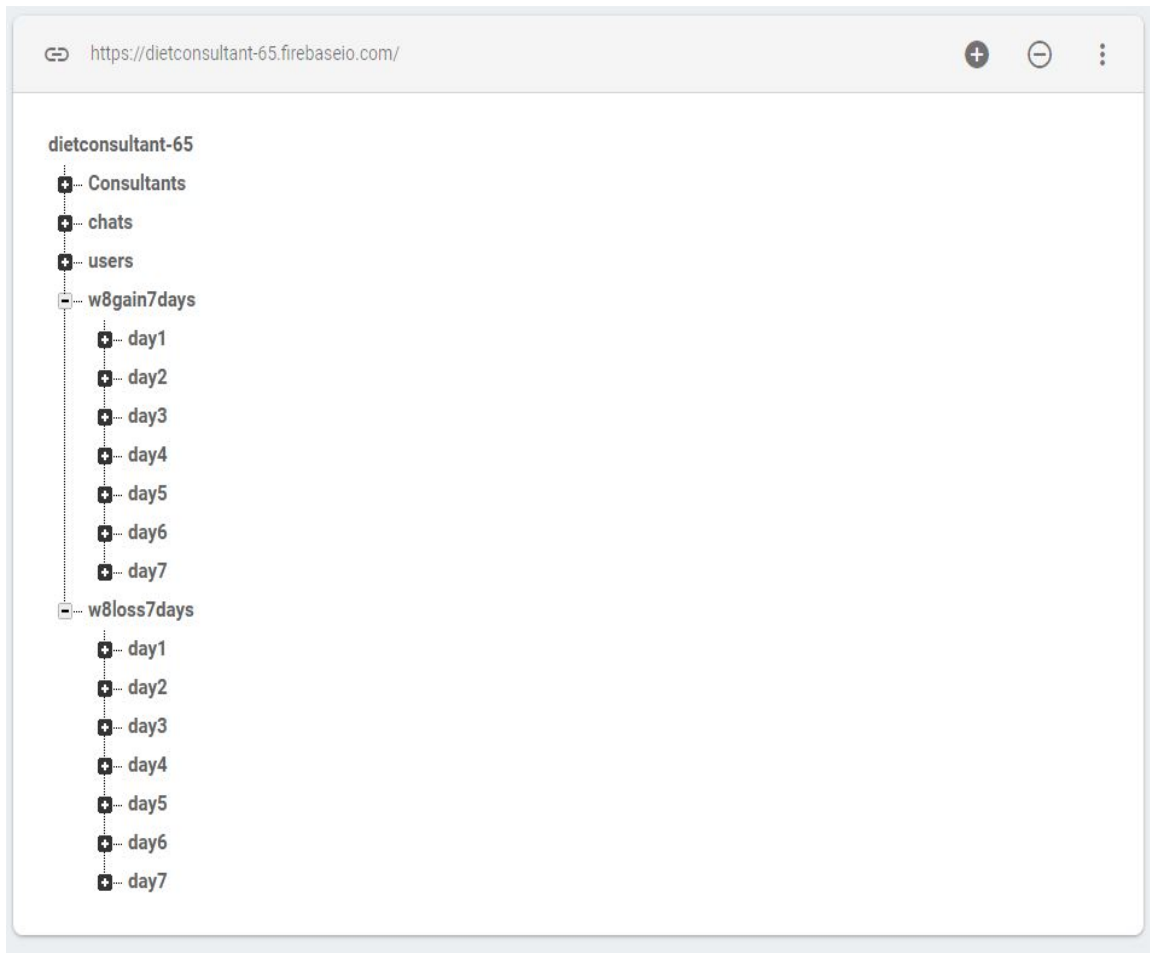
To avoid memory leak I need to do some extra steps in this projects. I need to use a library called Leak canary to avoid memory leak in this app which will lead an app to use extra space in ram which is very harmful to android application performance.

## CHAPTER 5

### Implementation and Testing

#### 5.1 Implementation of Database

I need some classes in our project which is actually responsible with database related work and calling servers. I need to enable our database to support in online and offline together. I separated the Model class from all database operation by Repository class which makes the project more controllable and maintainable with a good testable project.



**Figure 5.1: Data Structure of NoSQL database**

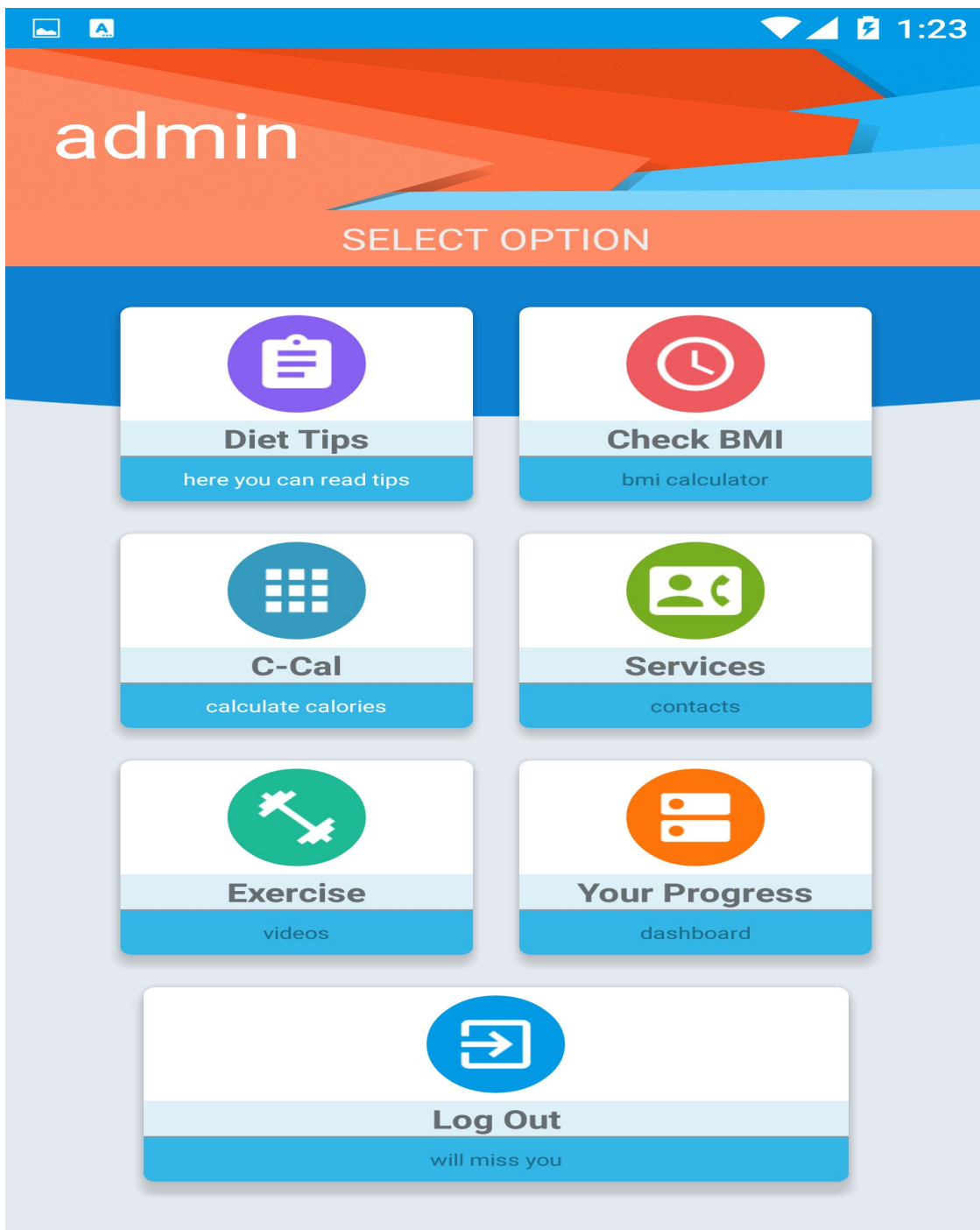
## 5.2 Model Class

Here I use actually 7 models. These are the real world entities which are used in this app as Model object.

1. User class
2. Consultant class
3. Massage class
4. BMI class
5. Calories class
6. Diet class
7. Food class

Every class above are represented the real world entities of the application as model class which is used to connect real world entity to programming basically Object oriented programming.

Many of those extended base class for getting all methods and attributes inside them. That the development of my android application become easy to implement.



**Figure 5.2: Main dashboard**

### **5.3 View**

It has some common views. They are-

1. Login
2. Registration
3. Diet list
4. Diet Detail
5. BMI interface
6. Caloric calculator
7. Exercise Videos
8. Doctors details
9. Chat interface
10. User profile interface
11. Progress viewer interface
12. User Logout

### **5.4 Controller**

There are some controllers which control the business logic of this project. They are-

1. Message controller
2. Diet list controller
3. BMI calculation controller
4. C calculator controller
5. Doctor list view controller
6. Doctor profile view controller
7. Progress view controller
8. Alarm set controller
9. Login Logout controller

```

public class UserAdapter extends RecyclerView.Adapter<UserAdapter.VideoViewHolder> {

    Context mContext;
    List<User> mUser = new ArrayList<>();

    public UserAdapter(Context context, List<User> user) {
        this.mContext = context;
        this.mUser = user;
    }

    @Override
    public VideoViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

        View view = LayoutInflater.from( parent.getContext()).inflate(R.layout.user_item, parent, false);

        return new UserAdapter.VideoViewHolder(view);
    }

    @Override
    public void onBindViewHolder(final VideoViewHolder holder, int position) {

        User user = mUser.get(position);
        holder.text.setText(user.getName());

        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(mContext, Message.class);
                intent.putExtra("name", holder.text.getText());
                mContext.startActivity(intent);
            }
        });
    }

    @Override
    public int getItemCount() {
        return mUser.size();
    }
}

```

**Figure 5.3: View with Adapter Pattern**

## **5.4 Testing Implementation**

I created a lot of test class for JUnit testing for this app which helped me to find out a lot of bugs. I also did some performance testing using performance monitor tools which reduced a lot of ram consumption.

### **5.4.1 Acceptance Testing**

The test case was far better from other software.it can recognize faster than others.

## **5.5 Test Results and Reports**

Everything is in testing result is positive. Here needs more work with the GUI to create more user interaction.

### **5.5.1 System Testing**

The whole particle of the system was tested properly. Now the beta version of the software will provide to some beta developer for their feedback.

### **5.5.2 User Acceptance Testing**

The beta testing was very appreciative. And user accept the system very easily. And give very positive feedback.

## **CHAPTER 6**

### **Conclusion and Future Scope**

#### **6.1 Discussion and Conclusion**

It will be a future virtual health assistant of every person. If I want to digitalize the system then, this project will be a good backbone to this system.

It will create a great impact on human health if everyone will use this.

#### **6.2 Scope for Further Developments**

Some of the work I could have done, had I continued this project -

- ☐ Gorgeous looking user interface
  - ☐ Using AI
  - ☐ Using Machine learning
  - ☐ Using Augmented reality
  - ☐ Using create centralized public database by this
  - ☐ Digital hospital management system by this
- Doctor patient interaction



## Reference

- [1] Learn about Body Mass Index (BMI) website, available at << <https://www.cdc.gov/healthyweight/assessing/bmi/> >>, last accessed on 04-03-2019 at 1:15 am.
- [2] Learn about MVC website, available at << [https://www.tutorialspoint.com/mvc\\_framework/mvc\\_framework\\_introduction.htm/](https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm/) >>, last accessed on 04-03-2019 at 1:55 am.
- [3] Learn about Rx java website, available at << <https://www.baeldung.com/rx-java/> >>, last accessed on 04-03-2019 at 1:45 am.
- [4] Learn about Android Development website, available at << <https://developer.android.com/studio/> >>, last accessed on 04-03-2019 at 1:34 am.
- [5] Learn about Android Adapter website, available at << <https://developer.android.com/reference/android/widget/Adapter/> >>, last accessed on 04-03-2019 at 1:25 am.
- [6] Learn about Android list view website, available at << [https://www.tutorialspoint.com/android/android\\_list\\_view.html/](https://www.tutorialspoint.com/android/android_list_view.html/) >>, last accessed on 04-03-2019 at 1:40 am.
- [7] Learn about Firebase website, available at << <https://firebase.google.com/docs/> >>, last accessed on 04-03-2019 at 2:10 am.
- [8] Learn about Chatting application website, available at << <https://www.raywenderlich.com/5359-firebase-tutorial-real-time-chat/> >>, last accessed on 04-03-2019 at 2:30 am.
- [9] Learn about Progress Bar website, available at << <https://developer.android.com/reference/android/widget/ProgressBar/> >>, last accessed on 04-03-2019 at 2:40 am.
- [10] Learn about Firebase authentication website, available at << <https://firebase.google.com/docs/auth/> >>, last accessed on 04-03-2019 at 2:50 am.
- [11] Learn about Android Data binding website, available at << <https://developer.android.com/topic/libraries/data-binding/> >>, last accessed on 04-03-2019 at 2:55 am.

- [12] Learn about SQLite website, available at << <https://www.sqlite.org/docs.html/> >>, last accessed on 05-03-2019 at 1:34 am.
- [13] Learn about Software development process website, available at << [https://en.wikipedia.org/wiki/Software\\_development\\_process/](https://en.wikipedia.org/wiki/Software_development_process/) >>, last accessed on 05-03-2019 at 1:34 pm.
- [14] Learn about Specialist doctor of Dhaka website, available at << <https://findoutadoctor.blogspot.com/> >>, last accessed on 05-03-2019 at 1:34 pm.
- [15] Learn about Diet List website, available at << <https://www.nhs.uk/live-well/healthy-weight/12-tips-to-help-you-lose-weight/> >>, last accessed on 05-03-2019 at 1:34 pm.
- [16] Learn about Repository pattern website, available at << <https://medium.com/corebuild-software/android-repository-pattern-using-rx-room-bac6c65d7385/> >>, last accessed on 05-03-2019 at 1:34 pm.
- [17] Learn about Use Case Diagram website, available at << [https://en.wikipedia.org/wiki/Use\\_case\\_diagram/](https://en.wikipedia.org/wiki/Use_case_diagram/) >>, last accessed on 05-03-2019 at 1:34 pm.
- [18] Learn about Software testing website, available at << [https://en.wikipedia.org/wiki/Software\\_testing/](https://en.wikipedia.org/wiki/Software_testing/) >>, last accessed on 05-03-2019 at 1:34 pm.
- [19] Learn about ER model website, available at << <https://beginnersbook.com/2015/04/e-r-model-in-dbms/> >>, last accessed on 05-03-2019 at 1:34 pm.
- [20] Learn about Data flow diagram website, available at << [https://en.wikipedia.org/wiki/Data-flow\\_diagram/](https://en.wikipedia.org/wiki/Data-flow_diagram/) >>, last accessed on 05-03-2019 at 1:34 pm.

## Appendix

### Sample Code:

BMI Calculation Code

```
package com.diu.finalproject.dietplan;

import android.content.Intent;
import android.graphics.Color;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.AppCompatSeekBar;
import android.support.v7.widget.CardView;
import android.view.View;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;
import com.diu.finalproject.dietplan.R;
import com.github.capur16.digitalspeedviewlib.DigitSpeedView;
import com.github.capur16.digitalspeedviewlib.OnSpeedChangeListener;
import com.shinelw.library.ColorArcProgressBar;

public class DashboardActivity extends AppCompatActivity {

    CardView underw8, overw8, healthy, obese;
    TextView feet, w, bmitxt, status;
    private DigitSpeedView fmeter, imeter, weightmeter;
    private AppCompatSeekBar fseek, isek, weightseek;
    private ColorArcProgressBar bar3;
    float weightKG, heightM, ft, in;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dashboard);
    }
}
```

```

        final DigitSpeedView digitSpeedView = (DigitSpeedView)
findViewById(R.id.digit_speed_view);
        fmeter = (DigitSpeedView) findViewById(R.id.fmeter);
        imeter = (DigitSpeedView) findViewById(R.id.imeter);
        weightmeter = (DigitSpeedView) findViewById(R.id.weightmeter);
        fseek = (AppCompatSeekBar) findViewById(R.id.fseek);
        isseek = (AppCompatSeekBar) findViewById(R.id.isseek);
        weightseek = (AppCompatSeekBar) findViewById(R.id.weightseek);
        bmitxt = (TextView) findViewById(R.id.bmitxt);
        bar3 = (ColorArcProgressBar) findViewById(R.id.bar3);
        feet = (TextView) findViewById(R.id.feet);
        w = (TextView) findViewById(R.id.w);
        status = (TextView) findViewById(R.id.status);
        underw8 = (CardView) findViewById(R.id.underweight);
        overw8 = (CardView) findViewById(R.id.overweight);
        healthy = (CardView) findViewById(R.id.healthy);
        obese = (CardView) findViewById(R.id.obese);

        underw8.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // startActivity(new Intent(DashboardActivity.this,HtmlView.class));
                Intent intent = new Intent(DashboardActivity.this, HtmlView.class);
                intent.putExtra("key", "w8gain7days");
                startActivity(intent);
            }
        });

        overw8.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

```

```

        // startActivity(new Intent(DashboardActivity.this,HtmlView.class));
        Intent intent = new Intent(DashboardActivity.this, HtmlView.class);
        intent.putExtra("key", "w8loss7days");
        startActivity(intent);
    }
});

fseek.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
        fmeter.updateSpeed(i);
        i *= 12;
        float fh = (float) i;
        ft = (float) (fh / 39.3701);
        heightM = calculateFtoC(ft, in);
        bmitxt.setText(String.valueOf(calculateBMI(heightM, weightKG)));

        if ((int) calculateBMI(heightM, weightKG) < 60) {
            bar3.setCurrentValues(calculateBMI(heightM, weightKG));
            digitSpeedView.updateSpeed((int) calculateBMI(heightM, weightKG));
        } else {
            //Toast.makeText(DashboardActivity.this, "MAX LENGTH REACHED",
Toast.LENGTH_SHORT).show();
        }
    }
});

@Override
public void onStartTrackingTouch(SeekBar seekBar) {

}

```

```

@Override
public void onStopTrackingTouch(SeekBar seekBar) {

}

});

iseek.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
        imeter.updateSpeed(i);
        float fh = (float) i;
        in = (float) (fh / 39.3701);
        heightM = calculateFtoC(ft, in);
        bmitxt.setText(String.valueOf(calculateBMI(heightM, weightKG)));
        if ((int) calculateBMI(heightM, weightKG) < 61) {
            bar3.setCurrentValues(calculateBMI(heightM, weightKG));
            digitSpeedView.updateSpeed((int) calculateBMI(heightM, weightKG));
        } else {
            //Toast.makeText(DashboardActivity.this, "MAX LENGTH REACHED",
Toast.LENGTH_SHORT).show();
        }
    }
});

@Override
public void onStartTrackingTouch(SeekBar seekBar) {

}

@Override
public void onStopTrackingTouch(SeekBar seekBar) {

```

```

    }
});

weightseek.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SearchBar seekBar, int i, boolean b) {
        weightmeter.updateSpeed(i);
        weightKG = (float) i;
        bmitxt.setText(String.valueOf(calculateBMI(heightM, weightKG)));
        if ((int) calculateBMI(heightM, weightKG) < 61) {
            bar3.setCurrentValues(calculateBMI(heightM, weightKG));
            digitSpeedView.updateSpeed((int) calculateBMI(heightM, weightKG));
        } else {
            //Toast.makeText(DashboardActivity.this, "MAX LENGTH REACHED",
Toast.LENGTH_SHORT).show();
        }
    }
});

@Override
public void onStartTrackingTouch(SearchBar seekBar) {

}

@Override
public void onStopTrackingTouch(SearchBar seekBar) {

}
});
}

```

```

public float calculateFtoC(float f, float i) {
    return f + i;
}

public float calculateBMI(float height, float weight) {
    feet.setText(String.valueOf(height));
    w.setText(String.valueOf(weight));
    int wgol = 0;

    float bmi = weight / (height * height);

    int ibmi = (int) bmi;
    if (ibmi <= 19) {
        moonMoonDisabler('u');
        wgol = (int) (((weight / ibmi) * 22) - weight);
        status.setText("you are underweight\nyou have to increase " + wgol + " kg");
        status.setTextColor(Color.parseColor("#FF495E"));
    } else if (ibmi > 19 && ibmi <= 25) {
        moonMoonDisabler('h');
        status.setText("you are Healthy");
        status.setTextColor(Color.parseColor("#439700"));
    } else if (ibmi > 25 && ibmi <= 29) {
        moonMoonDisabler('o');
        wgol = (int) (weight - ((weight / ibmi) * 22));
        status.setText("you are overweight\nyou have to decrease " + wgol + " kg");
        status.setTextColor(Color.parseColor("#FF495E"));
    } else if (ibmi > 29) {
        moonMoonDisabler('b');
    }
    return bmi;
}

```



```
public void moonMoonDisabler(char c) {  
  
    switch (c) {  
        case 'u':  
            underw8.setEnabled(true);  
            overw8.setEnabled(false);  
            healthy.setEnabled(false);  
            obese.setEnabled(false);  
            break;  
        case 'h':  
            underw8.setEnabled(false);  
            overw8.setEnabled(false);  
            healthy.setEnabled(true);  
            obese.setEnabled(false);  
            break;  
        case 'o':  
            underw8.setEnabled(false);  
            overw8.setEnabled(true);  
            healthy.setEnabled(false);  
            obese.setEnabled(false);  
            break;  
        case 'b':  
            underw8.setEnabled(false);  
            overw8.setEnabled(true);  
            healthy.setEnabled(false);  
            obese.setEnabled(true);  
            break;  
    }  
  
}
```

}