



*Report of the project*

# WORTH CARE – A Health Care App

**CSE299 – Junior Project Design**

Submitted By:

Name: Md. Tarbiar Swat  
Tashfiq Choudhury

ID: 1620918042  
ID: 1611508042

Submitted To:

**Intisar Tahmeed Nahin (ITN) Sir**

---

## TABLE OF CONTENTS

---

Title	Page
ABSTRACT -----	01
INTRODUCTION -----	02
BACKGROUND -----	03
OBJECTIVES -----	03
• BMI CALCULATON -----	04
• CONSULTATION -----	06
• MEDICINE DIRECTORY -----	08
• MEDICINE REMINDER -----	10
• WORKOUT TRAINER -----	11
• HEALTH TIPS AND ARTICES -----	11
USER REQUIREMENTS -----	13
METHODOLOGY AND APPROACH -----	14
BASIC LAYOUT -----	14
EXPECTED OUTCOMES -----	14
REFERENCES -----	15
CONCLUSION -----	15

---

## INDEX OF FIGURES

---

Title	Page
BMI CALCULATION -----	04, 05
CONSULTATION -----	07
MEDICINE DIRECTORY -----	10
WORKOUT TRAINER -----	11
HEALTH TIPS AND ARTICLES -----	12

## ABSTRACT

As medical and biological science has progressed, access to knowledge regarding one's health and the will to implement ways to keep oneself healthy has taken precedence more and more among the general population. This application will help its users to track their health status and maintain them with a proper system. Basically it will follow the BMI of the users and keep them updated daily what they need to do to improve their health condition. Moreover, it will have a medicine directory that will help users to know about the details about a medicine. There will be a medicine reminder, users have to set the medicine names and the timing when they take those, our app will notify about that. This project addresses an opportunity to provide a tool which supports both patient and provider in the personalization of the healthcare experience with the goal being optimization of overall health and wellness.

Why have we selected this? We can see the usage of smart devices will never be less. And people nowadays are less conscious about their health, so why not we put An Extra Addition of technology in their daily usage?

Additionally it will communicate with its users and will help to maintain his/her health properly based on the standard health status. And later on, there are some upcoming features those will be "All in One".

## INTRODUCTION

Since the emergence and popularization of smartphones, many mobile applications that track and record data about their users have been created. The classic example of this is the pedometer which utilizes the mobile device's built-in accelerometer to track the number of steps the user takes each day. Applications in this category, that track and record health or activity data about their users, are typically called Wellness or Fitness Apps. These Wellness Apps are designed to assist the user in pursuing a healthy lifestyle by encouraging them to perform positive activities, and improve lifestyle choices. Factors that are typically targeted by such applications include exercise, sleep, and diet. Understanding the nature of this relationship is crucial when designing a Wellness App. Applications like this have the potential to motivate its users into maintaining a cycle of positive lifestyle decisions and/or breaking a cycle of negative lifestyle decisions. Diet, exercise and sleep can influence several physiological pathways associated with depression and a bidirectional relationship likely exists between depression and these lifestyle factors, thereby creating a potentially increasing cycle of influence.

This android application should be able to help users in being more knowledgeable about the status of their health and about ways they can maintain the said status. They can hope to achieve this by keeping track of their own BMI and also being regularly updated of the different procedures, dietary regimes and tips needed to retain or improve the condition at which their health is in. Moreover, the app would have a medicine directory that will allow users to know about the details of certain medicines before they go on to consume it. A medicine reminder will be set in place to notify users to take medicines they are prescribed in the appropriate times.

We believe the need for the development of this application is imminent as it can be observed that the people today rely heavily on using their smart devices to conveniently manage their daily dealings. Thus, we hope to have a further positive contribution to the use of technology in everyday lives.

In this report the following sections are going to be briefly discussed:

1. Background
2. Objectives
3. User Requirements
4. Methodology and Approach
5. Conclusion
6. References

## **BACKGROUND**

Previous works have been done regarding this prospect, the most common of which is Google Fit. According to its developers, Google Fit is a health-tracking platform that uses sensors in a user's activity tracker or mobile device to record physical fitness activities (such as walking, cycling, etc.), which are measured against the user's fitness goals to provide a comprehensive view of their fitness. In theory, organizing this information in an easily interpretable way, motivates the user to continue a healthy lifestyle. Each of these apps has a unique way of expressing data to the user, and some methods have proven more effective than others.

Besides having the features that are integrated in Google Fit, our application, additionally, will communicate with its users and help them in maintaining their health setting the standard health status as the bar. This project will be successful when a software tool is produced which enables a patient to more easily and effectively achieve optimal health and wellness.

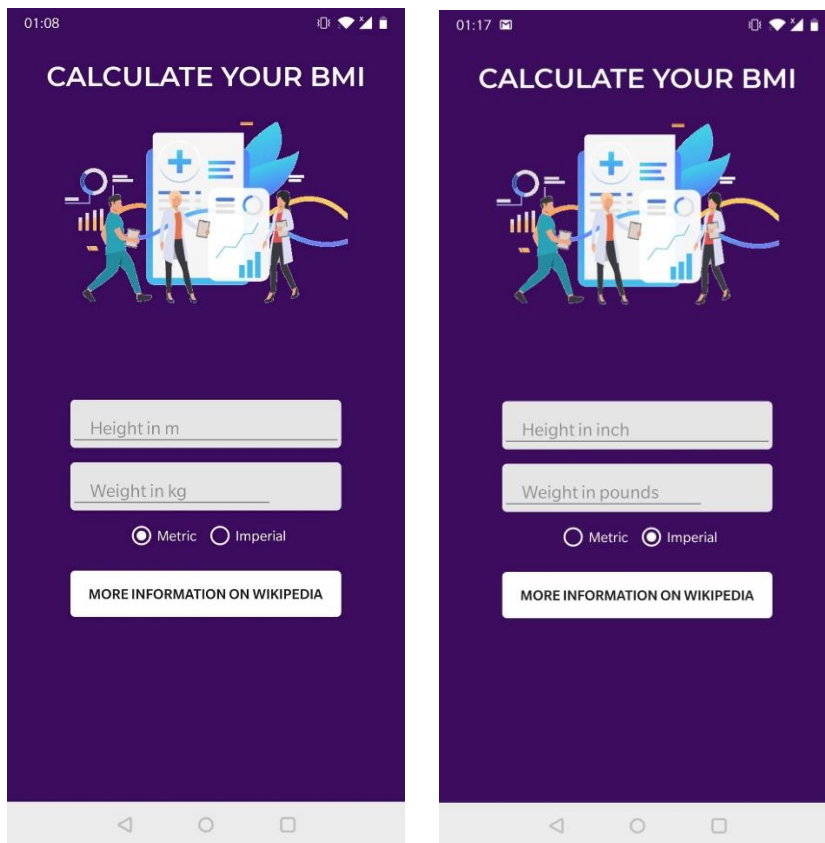
There are several factors which can affect the outcome of this project. The technology employed in this project may not be the most currently available when the project is complete. The goal of this project is to provide a user with an application which will facilitate access to providers, ensure that appointments are not missed, track current and new medications, display fitness regimens and goals, and also keep track of dietary constraints. It is anticipated that the patient market will embrace this tool.

## **OBJECTIVES**

The application will have some key features that it will act as a base for new developments. The following are the key features:

- **BMI Calculation**

On installation, the app will ask the user to insert height and weight to calculate BMI and send push notifications to improve health on a daily basis.



As shown in the figures above, the user will be able to type in their height and weight measurements in any of the two unit systems: metric (height in meters, weight in kilograms) and imperial (height in inches, weight in pounds).

On entering the numbers in the given spaces for height and weight, the BMI of the user will be calculated and an inference message will be shown along with the result. This is shown in the figure below.

Let's look at the code snippet so that we can understand how it worked:

```

126     private double calculateBmiAndCastIfNeeded(double height, double weight) {
127         height = isMetric() ? height : height / 39.37008;
128         weight = isMetric() ? weight : weight / 2.204623;
129         return calculateBmi(height, weight);
130     }
131
132     public static double calculateBmi(double height, double weight) {
133         return Math.round(weight / Math.pow(height, 2) * 10d) / 10d;
134     }

```

Here in this basic calculation function we have casted the BMI type module whether if it is metric or Imperial and calculated the user given input with ***calculateBmi*** method.

```

136 @ private String getCategory(double bmi) {
137     if (bmi < 15) {
138         return "Very severely underweight";
139     }
140     if (bmi < 16) {
141         return "Severely underweight";
142     }
143     if (bmi < 18.5) {
144         return "Underweight";
145     }
146     if (bmi < 25) {
147         return "Normal (healthy weight)";
148     }
149     if (bmi < 30) {
150         return "Overweight";
151     }
152     if (bmi < 35) {
153         return "Obese Class I (Moderately obese)";
154     }
155     if (bmi < 40) {
156         return "Obese Class II (Severely obese)";
157     }
158     if (bmi < 45) {
159         return "Obese Class III (Very severely obese)";
160     }
161     if (bmi < 50) {
162         return "Obese Class IV (Morbidly Obese)";
163     }
164     if (bmi < 60) {
165         return "Obese Class V (Super Obese)";
166     }
167     return "Obese Class VI (Hyper Obese)";
168 }

```

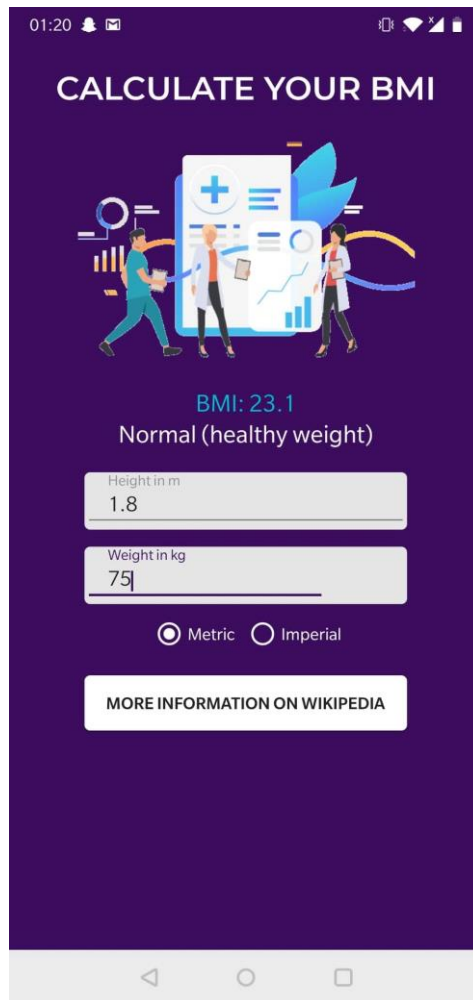
We have compared the user given inputs by category with our predefined BMI structure taken from Wikipedia within the *if else* condition.

To set the system of units the previous methods and objects were used.

```

170 private void setSystemOfUnits() {
171     RadioButton btn_metric = findViewById(R.id.btn_metric);
172     RadioButton btn_imperial = findViewById(R.id.btn_imperial);
173     btn_metric.setChecked(isMetric());
174     btn_imperial.setChecked(!isMetric());
175
176     TextInputLayout txt_weight_outer = findViewById(R.id.txt_weight_outer);
177     TextInputLayout txt_height_outer = findViewById(R.id.txt_height_outer);
178     txt_weight_outer.setHint(isMetric() ? "Weight in kg" : "Weight in pounds");
179     txt_height_outer.setHint(isMetric() ? "Height in m" : "Height in inch");
180 }
181
182 private boolean isMetric() {
183     boolean defaultToMetric = "@string/metric".equals("Metric");
184     return sharedPreferences.getBoolean(PREF_IS_METRIC, defaultToMetric);
185 }
186
187 @ public void setSystemOfUnits(View v) {
188     sharedPreferences.edit().putBoolean(PREF_IS_METRIC, v.getId() == R.id.btn_metric).apply();
189     setSystemOfUnits();
190     calculateBmiIfPossible();
191 }

```



Finally the BMI result is being shown just above the inputs with text high-lightening.

On clicking the “More Information On Wikipedia” button, the user will be taken out of the app to the Wikipedia page of BMI so that they are able to educate themselves further on the science behind the measure.

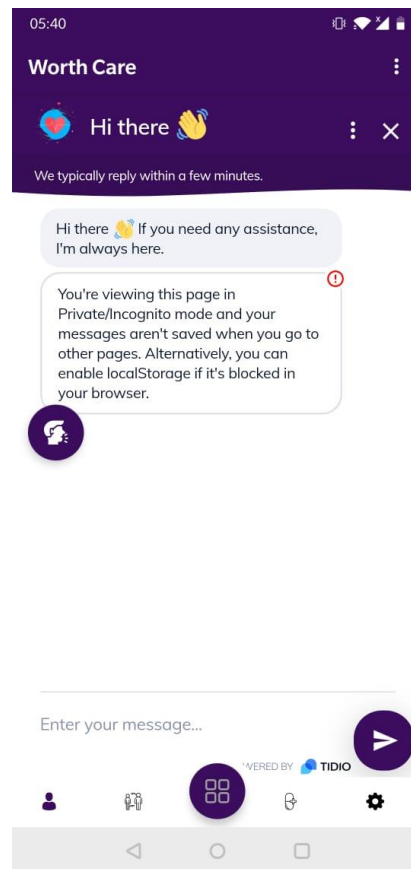
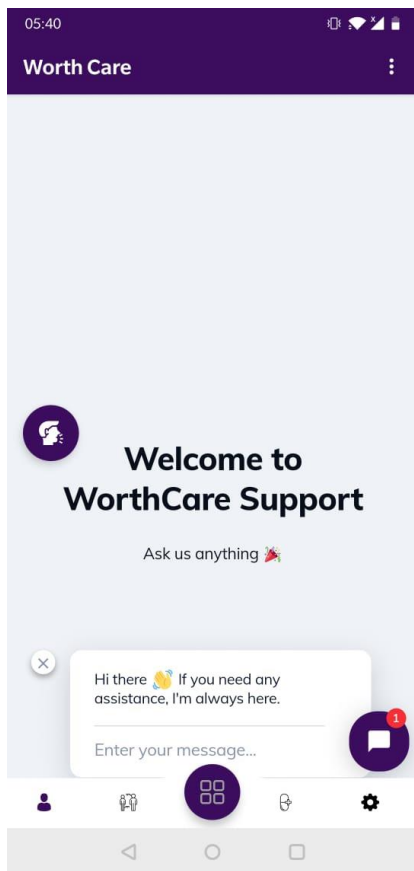
- **Consultation**

The application will ask the user how they are feeling. The user can tell the app about health status as well by using the available checkbox options and in response they will receive suggestions and tips. The user will be able to interact with the app in the following two main ways:

#### **Voice Command**

Users can speak directly onto the microphones of their devices and ask about basic queries such as what they should do in case of being affected by a certain disease and so on and so forth. Receiving and interpreting these voice commands, an AI generated voice would be responding to the queries.





## Chatbot

Users can also chat with the AI bot via text and help themselves through recorded conversations which will also be used to develop the quality of the conversations and the response of the bot through application updates later on.

```

196         if (result_message.indexOf("what") != -1) {
197             if (result_message.indexOf("your name") != -1) {
198                 speak("My Name is Ultra. Nice to meet you!");
199             }
200             if (result_message.indexOf("time") != -1) {
201                 String time_now = DateUtils.formatDateTime(context, this, new Date().getTime(), DateUtils.FORMAT_SHOW_TIME);
202                 speak("The time is now: " + time_now);
203             }
204             if (result_message.indexOf("slang") != -1) {
205                 String time_now = DateUtils.formatDateTime(context, this, new Date().getTime(), DateUtils.FORMAT_SHOW_TIME);
206                 speak("No, I'm just your voice companion, not your crime partner");
207             }
208         }
209         else if (result_message.indexOf("earth") != -1) {
210             speak("Don't be silly, The earth is a sphere. As are all other planets and celestial bodies");
211         }
212         else if (result_message.indexOf("fever") != -1) {
213             speak("Ace Plus and Napa can be your help, try it.");
214         }
215         else if (result_message.indexOf("browser") != -1) {
216             speak("Opening a browser right away master.");
217             Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("https://www.google.com"));
218             startActivity(intent);
219         }
220     }
221 }
222
223 private void initializeTextToSpeech() {
224     tts = new TextToSpeech(context, this, (status) -> {
225         if (tts.getEngines().size() == 0) {
226             Toast.makeText(context, UltraMainActivity.this, "There were no Text to Speech Engines located in the device", Toast.LENGTH_LONG).show();
227             finish();
228         } else {
229             tts.setLanguage(Locale.US);
230             speak("Hello, how can I help you?");
231         }
232     });
233 }
234
235 }

```

Here we have Android API level restriction. **API level Less than 21** can not have this voice companion functionality. In the below snippets we can see the predefined strings of our voice companion.

- **Medicine Directory**

Here, users can search for the medicine they're looking for using the search tool .A chatbot is featured on this interface as well. The platform will consist of a directory that will inform the users of intricate details about the medicine(s) in question. Details such as benefits, reactions and side effects, and average scheduling times will be spoken of. Here is a figure that shows how the medicine directory looks upon entering:

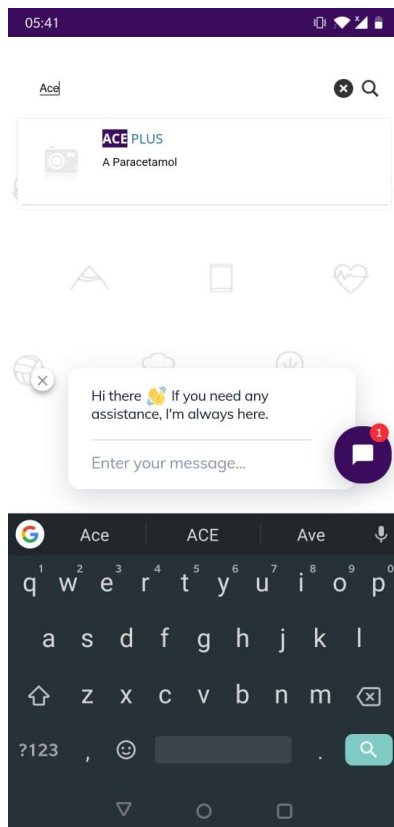
Here Medicine directory is being fetched from a website and currently it is in webview mode.

Let's have a look at this short code snippets:

```
18      webView = (WebView) findViewById(R.id.webview);
19      webView.setWebViewClient(new WebViewClient());
20      webView.loadUrl("https://www.chinton.org/medicine-directory/");
21
22      WebSettings webSettings = webView.getSettings();
23      webSettings.setJavaScriptEnabled(true);
24  }
```

Here we have made a website of a medicine directory on our own test domain. It has also been integrated with chatbot from 'Tidio Chat' API.

Our Medicine directory Interface:



- **Medicine Reminder**

Users, using the suggestion of the application or otherwise, can provide an input as to which medicine they are to consume and at which time they do it. Using this information, the app will initiate reminders at the given time. This could specially be beneficial for senior citizens. The user is able to register new reminder entries and also view the history of reminder notifications they had entered in the past. The following figures shows the user interface on entering the feature, followed by the interface that appears when the user decides to register a new reminder for the future.

Now let's see how the codes worked to set the reminder in the frontend. We will look forward to the very basic methods and objects with adapter class as well.

```

35     final NumberPicker numberPicker = (NumberPicker) view.findViewById(R.id.picker1);
36     numberPicker.setMinValue(2);
37     numberPicker.setMaxValue(999);
38     numberPicker.setWrapSelectorWheel(false);
39
40     final NumberPicker repeatPicker = (NumberPicker) view.findViewById(R.id.picker2);
41     repeatPicker.setMinValue(0);
42     repeatPicker.setMaxValue(4);
43     repeatPicker.setWrapSelectorWheel(false);
44     repeatPicker.setDisplayedValues(getRepeatValues(numberPicker.getValue()));
45
46     numberPicker.setOnValueChangedListener((picker, oldVal, newVal) -> {
47         repeatPicker.setDisplayedValues(getRepeatValues(newVal));
48     });
49
50
51
52
53     builder.setPositiveButton("OK", (dialog, which) -> {
54         String repeatType = repeatPicker.getDisplayedValues()[repeatPicker.getValue()];
55         String text = "Repeats every: {numberPicker.getValue()} {repeatType}";
56         listener.onAdvancedRepeatSelection( type: repeatPicker.getValue() + 1, numberPicker.getValue(), text);
57     });
58
59
60
61
62     builder.setNegativeButton(android.R.string.cancel, (dialog, which) -> {
63         dialog.cancel();
64     });
65
66
67
68
69     builder.setView(view);
70     return builder.create();
71 }
72
73 public String[] getRepeatValues(int number) {
74     String[] values = new String[5];
75     values[0] = hours;
76     values[1] = days;
77     values[2] = weeks;
78     values[3] = months;
79     values[4] = years;
80     return values;
81 }

```

Here we will be able to set the reminder by choosing from current date and time and afterwards. There will be an error dialogue if the date is previous than the current date.

```

28     @Override @NonNull
29     public Dialog onCreateDialog(Bundle savedInstanceState) {
30         final String[] repeatArray = getResources().getStringArray(R.array.repeat_array);
31         AlertDialog.Builder builder = new AlertDialog.Builder(getContext(), R.style.Dialog);
32         builder.setItems(repeatArray, (dialog, which) -> {
33             if (which == Reminder.SPECIFIC_DAYS) {
34                 DialogFragment daysOfWeekDialog = new DaysOfWeekSelector();
35                 daysOfWeekDialog.show(getActivity().getSupportFragmentManager(), tag: "DaysOfWeekSelector");
36             } else if (which == Reminder.ADVANCED) {
37                 DialogFragment advancedDialog = new AdvancedRepeatSelector();
38                 advancedDialog.show(getActivity().getSupportFragmentManager(), tag: "AdvancedSelector");
39             } else {
40                 listener.onRepeatSelection( dialog: RepeatSelector.this, which, repeatArray[which]);
41             }
42         });
43     }
44     return builder.create();
45 }
46

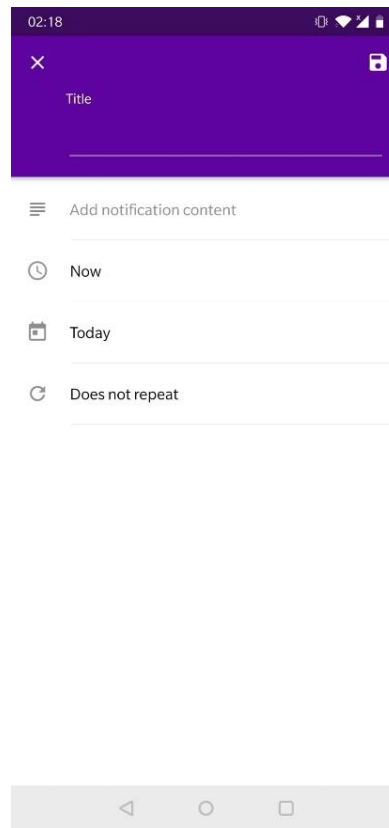
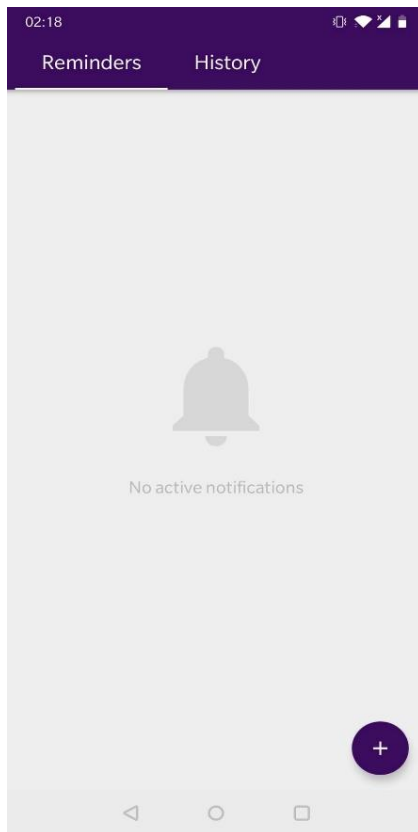
```

```

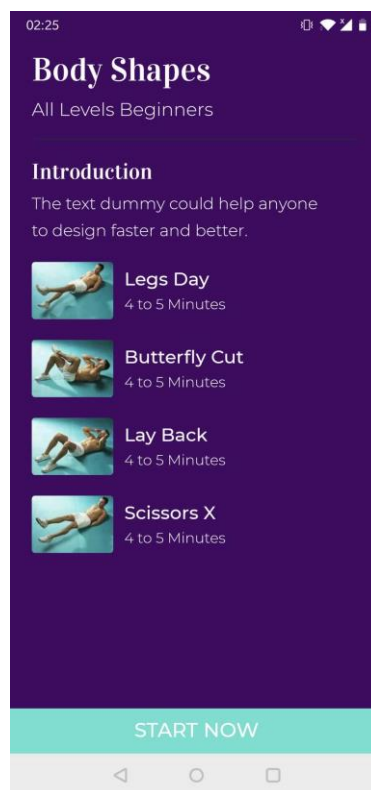
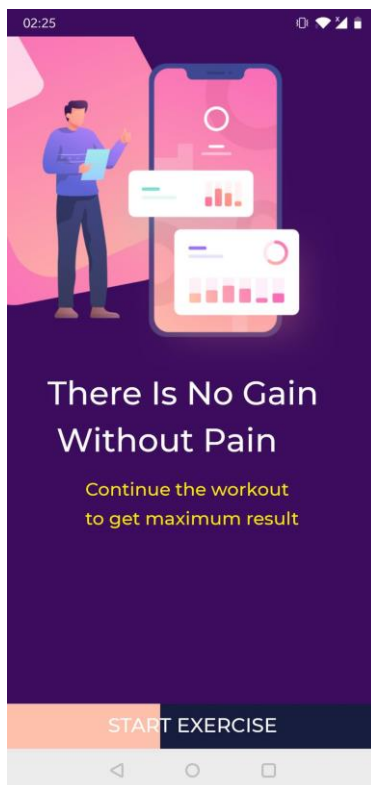
75     public void updateList() {
76         reminderList.clear();
77         reminderList.addAll(getListData());
78         reminderAdapter.notifyDataSetChanged();
79
80         if (reminderAdapter.getItemCount() == 0) {
81             recyclerView.setVisibility(View.GONE);
82             linearLayout.setVisibility(View.VISIBLE);
83         } else {
84             recyclerView.setVisibility(View.VISIBLE);
85             linearLayout.setVisibility(View.GONE);
86         }
87     }

```

Let's have a look at our medicine reminder interface:



- **Workout Trainer**



A workout regimen is chosen using this feature with names and optimal duration of the exercises. Users can take better care of their fitness this way not just through prescribed medication but through taking steps to actively keep themselves fit in order to try and prevent physical ailments from happening in addition to looking for cures and solutions through the app.

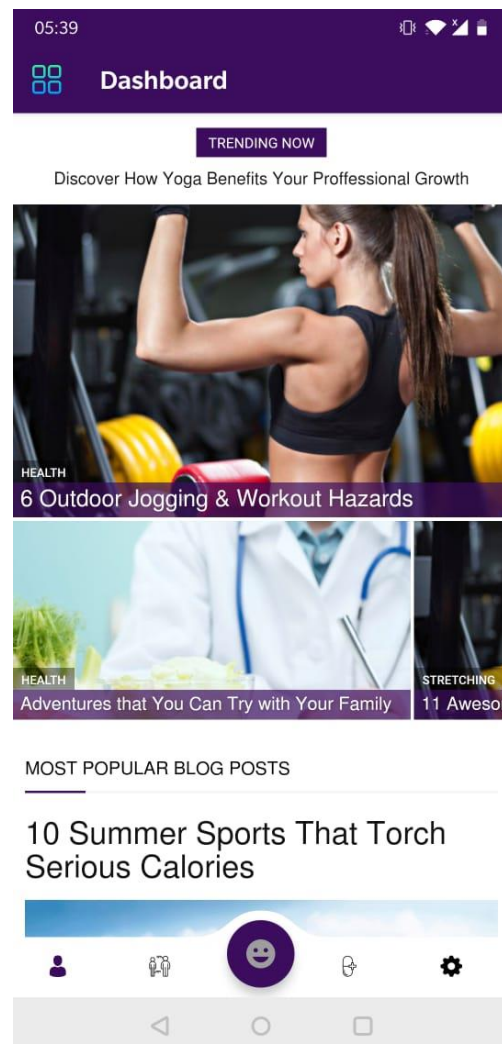
```

73      // item fit
74      fitonetitle = (TextView) findViewById(R.id.fitonetitle);
75      fitonedesc = (TextView) findViewById(R.id.fitonedesc);
76
77      fittwotitle = (TextView) findViewById(R.id.fittwotitle);
78      fittwodesc = (TextView) findViewById(R.id.fittwodesc);
79
80      fitthreetitle = (TextView) findViewById(R.id.fitthreetitle);
81      fitthreedesc = (TextView) findViewById(R.id.fitthreedesc);
82
83      fitfourtitle = (TextView) findViewById(R.id.fitfourtitle);
84      fitfourdesc = (TextView) findViewById(R.id.fitfourdesc);

```

- **Health Tips and Articles**

As the user enter the app, they are shown the dashboard on which there are a lot of trending topics, articles, blog posts and YouTube videos concerning fitness, health and also reliable news items. The figure below shows an instance of how it looks.



## **USER REQUIREMENTS**

This project will deliver the set of functionalities described within the Project Scope. These are necessary and sufficient conditions for satisfying the needs of the user as well as delivering business value. Following are additional requirements necessary for the full implementation of this software tool and its ultimate success.

### **Functional**

- An android smartphone or tablet (OS version Lollipop or greater)
- Internet access

### **Non-functional**

- Basic computer-use skills (i.e. navigation, user name/password use, etc.)
- Motivation to use a new software tool
- Ability to read and understand a printed, online, or video user's guide
- Motivation to engage in an increased level of self-care
- Understanding benefits of a patient portal

## **METHODOLOGY AND APPROACH**

A challenge for developing the app is to deal with all the different screen sizes and devices that the app will need to fit. Designing and developing an app for only a select few devices and screen sizes is simply not an option. The principal challenge is to develop an app that can run fluidly across as many devices with as many screen sizes as possible. The solution here is Responsive Design for apps. Responsive App Design gives screens more flexibility and thus can be adjusted to suit different screen sizes and various formats.

We have to consider different Operating Systems as well. The three main systems – iOS, Android and Windows – each of which has its respective set of UI objects and patterns. The requirement is to conduct bug fixing on a regular basis across all three. Preparation with a marked testing phase can also be done to deal with technical OS issues. A beta version of the app can be released to develop on later with regular updates.

Security issues can be a concern. Malware problems may arise along with software/hardware fragmentation. There is a lot of work required to address such app security issues, which consumes a great deal of time and money.

While there are existing software tools which provide the functionality present within any one of the components of this application, the objective of this project is to create a tool which will facilitate access to multiple elements of the healthcare experience from one user-interface thereby enabling a patient to more easily and effectively achieve a state of optimal health and wellness.

## BASIC LAYOUT

**Simplistic Design:** Overall, we wanted to implement a design a clean and simplistic layout that presented important information as soon as it was opened. We decided to model our app with this same mentality of keeping screens simplistic and present important data upfront.

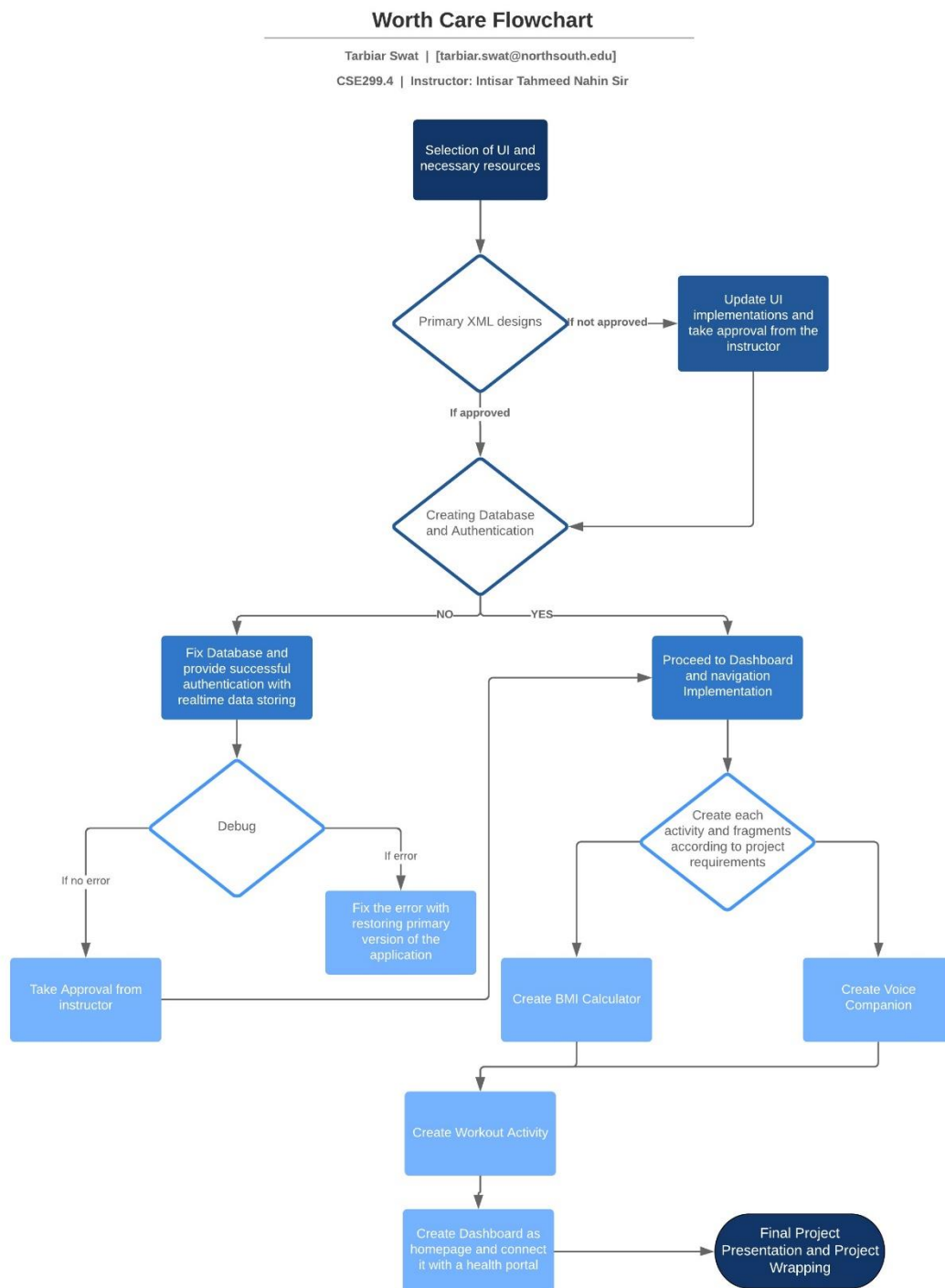
**Separate screens to display different lifestyle aspects:** In addition, we decided to employ separate screens for a breakdown of activities and features, as well as the other data the app tracked. We drew inspiration for separate detailed screen for each activity, function and section. We found it useful to see the data gathered by the app and felt it left the door open to later expansions. Overall, we found inspiration in a number of different apps that helped to shape the various features of our project.

## CONTRIBUTORS:

Md. Tarbiar Swat	<ol style="list-style-type: none"><li>1. Splash, Login and Registration UI, their corresponding XMLs,</li><li>2. Dashboard and Navigation UI,</li><li>3. Firebase Database configuration and project builds,</li><li>4. Splash screen animation and Registration with firebase with Realtime database,</li><li>5. Navigation drawer XML with required fragments based on the project,</li><li>6. Login with Firebase and logout functionality,</li><li>7. Workout Activities XMLs,</li><li>8. Debugging on multiple devices and internal testing using Logcat,</li><li>9. Workout details implementations,</li><li>10. Voice recognition and strings pass using TTS,</li><li>11. API integration with Tidio Chat,</li><li>12. Bottom Navigation and Medicine reminder XML design,</li><li>13. BMI calculator full functionality,</li><li>14. Dashboard with Blog API integration and</li><li>15. Debugging.</li></ol>
Mohammad Tashfiq H. Choudhury	<ol style="list-style-type: none"><li>1. XML Profile Activity and</li><li>2. XML Privacy Policy,</li></ol>

## WORKFLOW:

Here is our workflow in flowchart that describes how it was started:





## **EXPECTED OUTCOMES**

The potential for the utilization of this application can be significant. Future possibilities may include and not be limited to integrating appointment of physicians, consultation, building a forum based on conversations about health and proper diet.

## **CONCLUSION**

This application has been developed in the hopes that we would be creating something that further establishes the role of technology as a blessing for humanity. The project, made from scratch, is open to many future possibilities and is open to user suggestions on ways to make improvements on the laid groundwork.

## **REFERENCES**

- [1] Space Navigation View - <https://github.com/armcha/Space-Navigation-View>
- [2] Fitness App UI from Angga Risky - <http://garudapixel.com/en/how-to-build-fitness-android-application>