

Stay Healthy

by

Sravani Donepudi

B.Tech., GITAM University, 2016

A REPORT

submitted in partial fulfilment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2018

Approved by:

Major Professor
Dr. Mitchell Neilsen

Copyright

© SRAVANI DONEPUDI

2018.

Abstract

It's difficult to keep track of sleep patterns, medicine and water intake. Having sufficient sleep every day is very important as not getting enough sleep can lead to difficulties remembering things, concentrating and other problems. This application provides tools to track the sleeping pattern of the user based on the inputs given. When the user picks a date, a weekly graph is generated to see how the user's sleep cycle looks. It also displays the average hours the user slept each day and the sleep pattern type below the graph. Medicine reminder is another tool which reduces the risk of skipping medicines. As many people tend to complicate the severity level of the disease by forgetting to take the required medicine, this app can help them to prevent such problems. It reminds them to take their medicine at the entered times until the end of the course duration.

People often tend to forget to drink enough water. Staying hydrated is very important for our body to function properly and flush out toxins. For this purpose, this app features a water reminder to remind the user to drink water every hour until the daily limit is reached. This is done by calculating the amount of water that should be consumed every day based on the user's details like age, weight, physical activity and gender. Apart from these, to stay fit and healthy one should do some physical activity every day which helps to activate their body. The instructions and videos of some exercises, information on health tips, and natural remedies to some common health problems that can be cured without any medication are provided in the app. This would be a very convenient application for those who are very busy with their work life and have difficult time tracking their sleep, medicine times and staying hydrated at work.

Table of Contents

List of Figures.....	vii
List of Tables	viii
Acknowledgements.....	ix
Chapter 1 - Introduction.....	1
1.1 Project Description	1
1.2 Motivation	2
Chapter 2 - Background and Technologies Used	3
2.1 Android Studio	3
2.1.1 Setting up Android Studio	5
2.2 Android.....	6
2.2.1 Android Architecture.....	6
2.2.1.1 Linux Kernal.....	6
2.2.1.2 Hardware Abstraction Layer.....	6
2.2.1.3 Android Runtime (ART).....	6
2.2.1.4 Libraries	7
2.2.1.5 Application Framework	8
2.2.1.6 Applications	8
2.3 STORAGE.....	9
2.3.1 SQLite.....	9
2.3.2 SHARED PREFERENCES	9
Chapter 3 - Requirements and System Design	10
3.1 Requirement Analysis.....	10
3.1.1 System Requirements	11
3.1.2 Hardware Requirements for running the application.....	11
3.1.3 Software Requirements.....	11
3.2 System Design	12
3.2.1 UML Diagrams	12
3.2.1.1 Use Case Diagram.....	12

3.2.1.2 Activity Diagram	13
Chapter 4 - Building the System.....	15
4.1 Login Module	15
4.2 Home Screen Module	15
4.3 Exercise Module	15
4.4 Sleep Logger Module	16
4.5 Remedies Module	16
4.6 Reminder Module	17
4.7 Lines of Code.....	17
Chapter 5 - Using the System	18
5.1 Login Screen.....	18
5.1.1 Login screen validation:	19
5.2 Registration Screen.....	19
5.3 User Screen.....	20
5.4 Sleep Logger.....	21
5.4.1. Sleep Duration Calculator.....	21
5.4.2. Time pickers	22
5.4.3. Sleep Graph Generator and Sleep Analysis	23
5.5 Exercises Screen	24
5.6 Remedies Screen.....	25
5.7 Reminder Screen.....	26
5.7.1 Water Reminder.....	27
5.7.2 Medicine reminder.....	28
Chapter 6 - Security	30
6.1 Password Encryption	30
6.2 Facebook token validation.....	30
6.3 Proguard Tools for code obfuscation.....	31
6.4 SQL Injection:	31
Chapter 7 - Testing	32
7.1 Unit Testing	32
7.2 Integration Testing.....	35

7.3 Performance Testing.....	37
7.4 Accessibility for elders	39
Chapter 8 - Conclusion and Future Work.....	40
8.1 Conclusion	40
8.2 Future Work.....	40
Chapter 9 - References.....	41

List of Figures

Figure 2-1 Android Studio.....	4
Figure 2-2 Android Architecture.....	7
Figure 3-1 Use Case Diagram.....	13
Figure 3-2 Activity Diagram.....	14
Figure 5-1 Splash Screen.....	18
Figure 5-2 Login screen.....	18
Figure 5-3 Invalid User Details.....	19
Figure 5-4 Registration Screen.....	20
Figure 5-5 Password Validation Screen.....	20
Figure 5-6 User Home Page.....	21
Figure 5-7 Sleep Logger Screen.....	22
Figure 5-8 Logging Sleep.....	22
Figure 5-9 Time Picker 1.....	23
Figure 5-10 Time Picker 2.....	23
Figure 5-11 Date Picker.....	24
Figure 5-12 Sleep Analysis.....	24
Figure 5-13 Exercise Screen.....	25
Figure 5-14 Exercise Content.....	25
Figure 5-15 Remedies Screen.....	26
Figure 5-16 Remedies Content.....	26
Figure 5-17 Reminder Screen.....	27
Figure 5-18 Validation of Inputs.....	27
Figure 5-19 Water Reminder Portion.....	28
Figure 5-20 Medicine Reminder Portion.....	29
Figure 7-1 Memory Usage.....	37
Figure 7-2 CPU Usage.....	37
Figure 7-3 CPU, Memory and Network Usage.....	38

List of Tables

Table 7-1 Unit Testing for Application	33
Table 7-2 Integration Testing.....	35

Acknowledgements

I would like to express my sincere gratitude to my Major Professor, Dr. Mitchell Neilsen for his support and for trusting my abilities to complete this project. His constant guidance and feedback led to the successful completion of this project. Thanks to him for patiently answering all my questions and helping me whenever needed.

I'm also grateful to my committee members Dr. Daniel Andresen and Dr. Torben Amtoft for their encouragement and for taking the time to serve on my committee.

Also, I would like to thank my parents, Giri Donepudi and Padmaja Donepudi for their immense love and support. I would like to thank my sister, Nandini Donepudi for constantly motivating and supporting me throughout my masters. I wouldn't have made it this far without them. Finally, thanks to my friends Nithin Kumar and Ajay Mandapaka for pushing me forward when I was in doubt. This accomplishment would not have been possible without them.

Chapter 1 - Introduction

1.1 Project Description

It is very important for a person to keep track of the amount of sleep one gets and the pattern of sleep: regular, moderate and irregular to maintain good health. Keeping track of sleep data will help to improve bedtime habits and wake up feeling energetic. A weekly graph is generated to show the amount of sleep the user gets. This graphical visualization helps the user to analyse how one's sleep patterns look like easily with different variations. It also gives the average number of hours the user slept based on the weekly graph. Another module of this app is the reminder section which has two parts: water and medicine. Staying hydrated is very important for the body to function properly at all times. Not having enough water required by the body daily can lead to several complications. This app focuses on reminding the user every hour to drink certain amount of water. Water reminder takes input from the user to give their gender, age, weight and amount of physical activity the user does. Based on those inputs, it calculates the amount of water the user has to consume daily. Once that is calculated, it reminds the user to drink portions of the calculated amount every hour by giving reminders. Coming to the medicine reminder, it takes the medicine name, start and end dates along with the times it has to be taken every day. Once the user inputs these details, the app reminds the user to take the mentioned medicine at the specified time. This way it will help people with the tendency to forget and busy schedules to drink enough water daily and take the required medicine on time until the course is completed.

This system consists of other parts, which include basic exercises, new feed and natural remedies to some common problems. The exercises section consists of instructions and videos of some basic exercises to keep a person fit and active. The health feed section consists of some

helpful tips needed to stay healthy. And the natural remedies section consists of some natural ways to cure the most common health problems.

1.2 Motivation

With the busy schedules, it has become almost impossible for a person to keep track of his sleep patterns, medicine and water intake. As all these factors are very important to maintain good health, the user needs a system that can remind them to take their medicines on time and drink enough water based on individual requirement. It also gives the sleeping patterns of the user which can help the user to keep track of how many hours of sleep he is getting and how regular the pattern is. Along with these which directly affect one's health, giving some basic health tips, natural remedies to common health problems that can be cured without the use of any medication and some basic exercises to keep the user fit and healthy are listed on the app. The idea of developing an application that would benefit the overall health of the user directly which is very important in the busy world is the motivating factor to work on this app. Apart from this, I was motivated to build this application in order to learn android development and gain some hands-on experience with these.

Chapter 2 - Background and Technologies Used

This system is built using Java as the programming language. The IDE used to build this system is Android Studio. As part of database management, I choose SQL Server to achieve it. XML is used for the design layout of the application. The detailed explanation of all the technologies used for this system are described in detail below.

2.1 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA and is designed specifically for Android Development. It is to Android like Eclipse is to Java. This can be downloaded on Windows, macOS and Linux. This IDE provides many useful features to make Android development flexible for developers. It is a very user-friendly IDE as one does not require an android phone to run and test the application. Emulator is provided with it to stimulate a real-time device to run and debug the applications in Android Studio itself with the help of AVD manager. When an application is developed on Android Studio, it will be automatically installed on the emulator to run and debug which is faster than running on an android device.

It is a gradle-based system which runs as an integrated tool from the Android Studio menu. It provides many features which can be achieved without modifying the app's core source files. Features include customizing, configuring and extending the build process, can create multiple APKS for the app with different features done with the same project and modules and lastly, the code and resources can be reused. The dependencies of the project are also specified in the build.gradle file which are made available in the build by the gradle.

It is easy and flexible to design a layout for the application being developed as it has the feature of dragging and dropping the UI components which automatically generate the components

for it in the xml files. It has instant run option which makes it easier to push all the changes made in the application to the running app without having to rebuild a new APK file for that. Android project view contains all the project files. Each app module will contain manifests, java and res files. The manifest files contain the AndroidManifest.xml file, java files contain the java source code files along with the Junit test code and res code consists of all the xml layouts, UI strings and bitmap images.

Android Studio also provides debugging and performance analysis tools. Android profiler is used to check the performance of the app built by tracking app's memory and CPU usage. It can also be used to find out memory leaks, analyse network requests and optimize app performance. Inline debugging is done to enhance the code with inline verification of references, variable values and expressions. The inline debugging can be enabled in the debug window under settings. Features like this make Android studio very effective for application development. [1]



Figure 2-1 Android Studio (<https://www.androidpolice.com/2017/05/18/android-studio-3-0-brings-kotlin-support-ton-new-debugging-tools/>)

2.1.1 Setting up Android Studio

Android Studio can be downloaded from the following link:

<https://developer.android.com/studio/index.html>

To run the Android Studio on the machine, java JDK is required. Java JDK and Android Studio should be first downloaded. The environment variable should be referencing to the JDK location before setting up Android Studio to let the launcher to know the proper location of JDK and make it compatible with Android Studio. After this process is completed, the executable file of the Android Studio will be launched, and the setup is followed. After installation is done, the Android Studio is launched. To configure it with all the additional packages suggested by the SDK manager, one has to go to the SDK Manager in the system settings and download those. After this step, we are ready to start a new project.

2.2 Android

Android is a mobile operating system was initially developed by Android Inc which was later bought by Google, based on a modified version of the Linux kernel and other open source software and designed primarily for touchscreen mobile devices such as smartphones and tablets.

2.2.1 Android Architecture

The main layers of Android architecture are as follows. [2]

2.2.1.1 Linux Kernel

The foundation of Android platform is the linux kernel. It provides functionalities such as threading and low-level memory management. Using linux kernel gives the advantages of key security features.

2.2.1.2 Hardware Abstraction Layer

This layer provides standard interfaces that expose device hardware capabilities to the higher-level Java API framework. When a framework API makes a call to access device hardware, the Android system loads the library module for that hardware components.

2.2.1.3 Android Runtime (ART)

It is written to run multiple virtual machines on low-memory devices by executing DEX files, a bytecode format designed for Android which is optimized for minimal memory footprint. Important features of ART include optimized garbage collection, better debugging support and ahead-of-time and just-in-time compilation. It contains a key component called Dalvik Virtual Machine that is specially designed and optimized for Android.

2.2.1.4 Libraries

It consists of java-based libraries for Android development. There are some core libraries like android.webkit, android.widget, android.view, android.os, and so on which are important for the application to run properly.

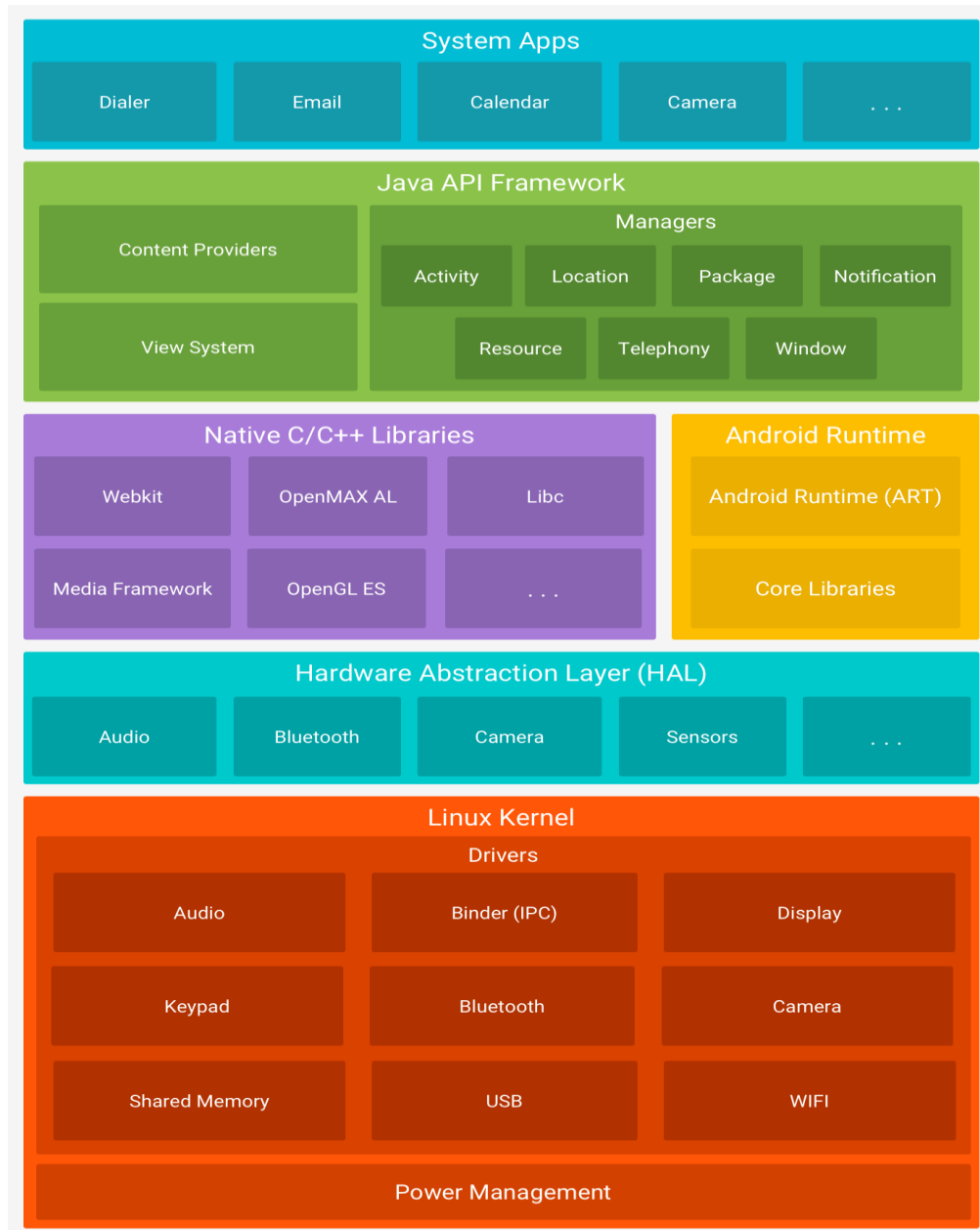


Figure 2-2 Android Architecture (<https://developer.android.com/guide/platform/index.html>)

2.2.1.5 Application Framework

This layer provides higher level services in the form of java classes to the applications. Some of the key services it provides are activity manager, content providers, resource manager, notifications manager and view system.

2.2.1.6 Applications

This is the top layer of the architecture. The application to be installed is written on this layer.

2.3 STORAGE

This application uses SQLite primarily for storage and some details are stored in shared preferences in key-value pairs.

2.3.1 SQLite

SQLite is a relational database management system. It is not a client-server database, but is embedded into the end program. It is a popular choice to be used as embedded database software for local/client storage in application software. SQLite library becomes an important part of the application, and its functionality is used by the application through simple function calls that can reduce the latency in database access. It stores the entire database which includes tables, definitions, indices, and the data itself as a single cross-platform file on the machine. [3]

As it is a server-less design, less configuration is required by SQLite applications than client-server databases. Here, several processes may not be able to write to the database file due to its design. For simple queries with little concurrency, SQLite performance profits from avoiding the overhead of passing its data to another process. It is a high-reliability storage solution and has been used without problems in billions of smart phones around the world. [4]

2.3.2 SHARED PREFERENCES

Shared Preferences allow you to save and retrieve data in the form of key, value pair. The first parameter is the key, and the second parameter is the mode. In order to use shared preferences, you have to call a method `getSharedPreferences()` that returns a `SharedPreferences` instance pointing to the file that contains the values of preferences. One can save data in shared preferences using `SharedPreferences.Editor` class. [5]

Chapter 3 - Requirements and System Design

3.1 Requirement Analysis

Requirements gathering is the most important step in the development cycle. It is during this phase that the developer understands how to build the system based on the requirements. Errors in the requirements given can lead to huge loss in the future as the developer has to re-build the whole system with the new requirements. It is always required to specify the requirements clearly at the beginning of the project for the developer to build the system correctly. Keeping this in mind, I initially jotted down the requirements for Stay Healthy that would be flexible for the user, which is outlined as follows:

- The user should be able to input the details of his sleep timings i.e. the time the user slept and woke up every day and the user should be shown a weekly graph of his sleep patterns based on the number of hours slept along with the average amount of sleep the user got, and the sleep pattern.
- The user should be able to see the instructions and videos to some basic exercises.
- The user should be able to input the gender, age and amount of physical activity done to calculate the amount of water that should be consumed daily. This should also remind the user to drink water every hour by giving a notification in the specified time range.
- The user should be able to see some basic health tips and natural remedies to cure common health problems.
- The user should be able to input the medicine name and the times at which should be taken every day along with the start and end dates. This should remind the user to take that medicine every day until the course duration is over with a notification.

3.1.1 System Requirements

- Operating System: Windows 7 or higher, Mac OS
- RAM: 3GB or higher
- Java version: JDK 8
- Disk space for Android Studio: 500 MB
- Space for Android SDK: 1.5 GB or higher
- Processor speed: 2.3 GHz
- RAM: 8GB

3.1.2 Hardware Requirements for running the application

- RAM: 512 MB
- Disk space: 250MB or more
- Android phone with API level 16 or higher (Jelly bean)
- Processor: i3 or higher

3.1.3 Software Requirements

- IDE: Android Studio
- Technologies: Java, XML, Android
- Database: SQLite
- Debugger: Emulator, Android mobile device

3.2 System Design

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. It is the process of developing a system for the user that will satisfy their requirements. It is a systematic approach to build the design of a system based on the user specific requirements. UML is widely used for object-oriented analysis and system design.

3.2.1 UML Diagrams

The Unified Modeling Language (UML) is a general-purpose, developmental, modelling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system. This helps to visualize the system's architecture in the form a diagram that will represent the individual components of a system, activities and the interaction between the components. UML diagrams are of two types: dynamic and static views. Dynamic diagram is also known as behavioural view which shows the changes to the internal states of objects and dynamic behaviour of the system. This consists of activity diagrams, sequence diagrams and state machine diagrams. The static view is also known as structural view which shows the static nature of the system using relationships, attributes, objects and operations. It consists of class diagrams and composite structure. [6]

3.2.1.1 Use Case Diagram

Use case diagrams fall under the behavioural diagrams which describe the set of actions that a system can perform with one or more users. The set of actions are the use cases, and the users are known as actors. Every use case should give some information useful to the user. These diagrams are used to specify the requirements and functionality of the user. It shows a pictorial representation of the use cases, actors and relationships involved between them.

Use case diagrams provide a high-level view and blueprints of the system. It gives a graphical representation of the system which tells what it will do and helps o understand how the system is going to be designed. They convey the intention of the system in a simple manner. The following diagram represents the use case diagram of this system.

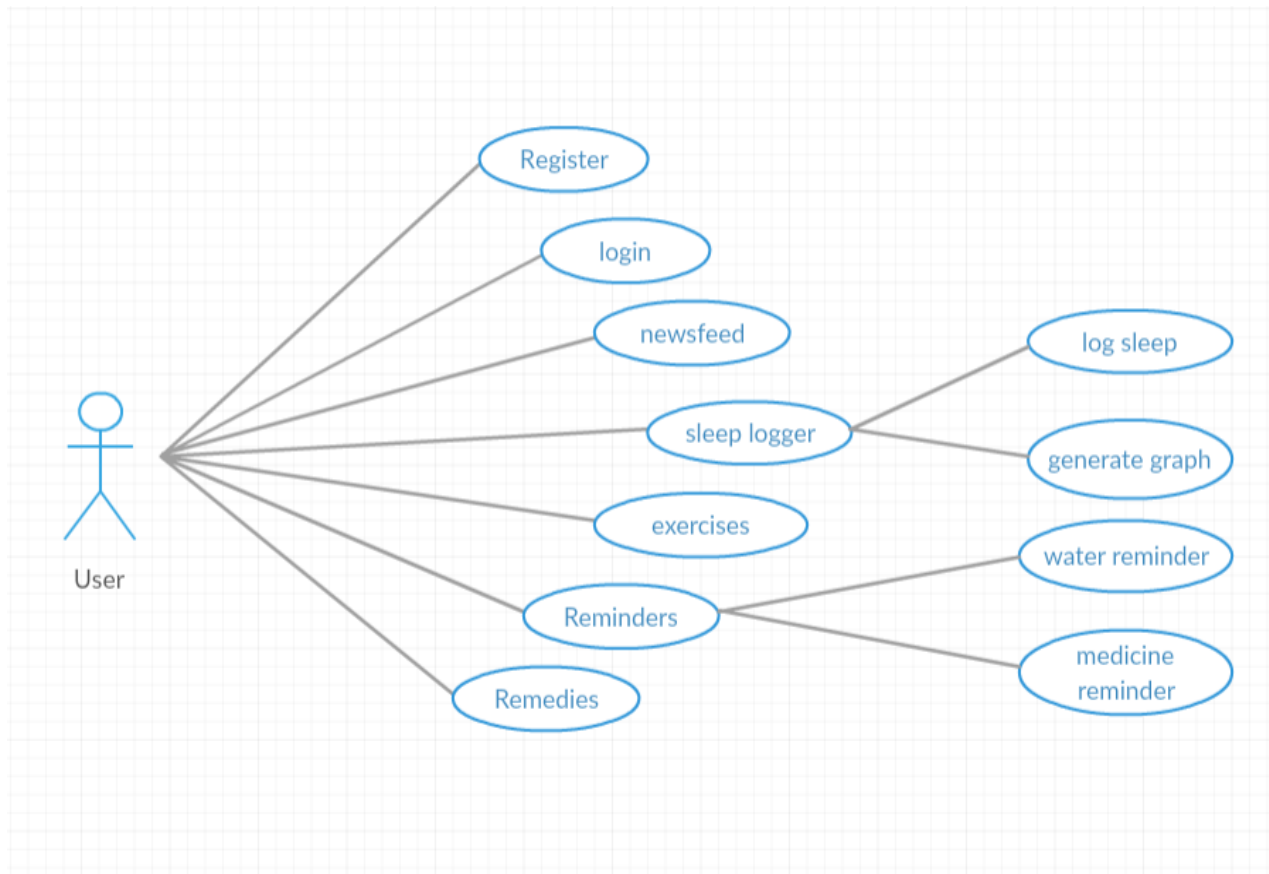


Figure 3-1 Use Case Diagram

3.2.1.2 Activity Diagram

Activity diagram describes the dynamic nature of the system. It is a flow chart that represents the flow from one activity to another. Here, an activity is an operation of the system. The flow from one operation to another can be sequential, branched or concurrent. Other diagrams show the flow from one object to other, but activity diagram shows the flow between 2 different activities. It can show different flows like parallel, branched, concurrent and single.

The main factors for using an activity diagram is to model the work flow using activities, to provide high level understanding of the system's functionalities, model business requirements and investigate them at a later stage. There are different elements that are used for activity diagram which include identifying the activities, association, conditions and constraints. After identifying all the elements, a layout of the diagram is fixed and implemented. [7] The activity diagram in this system is shown below.

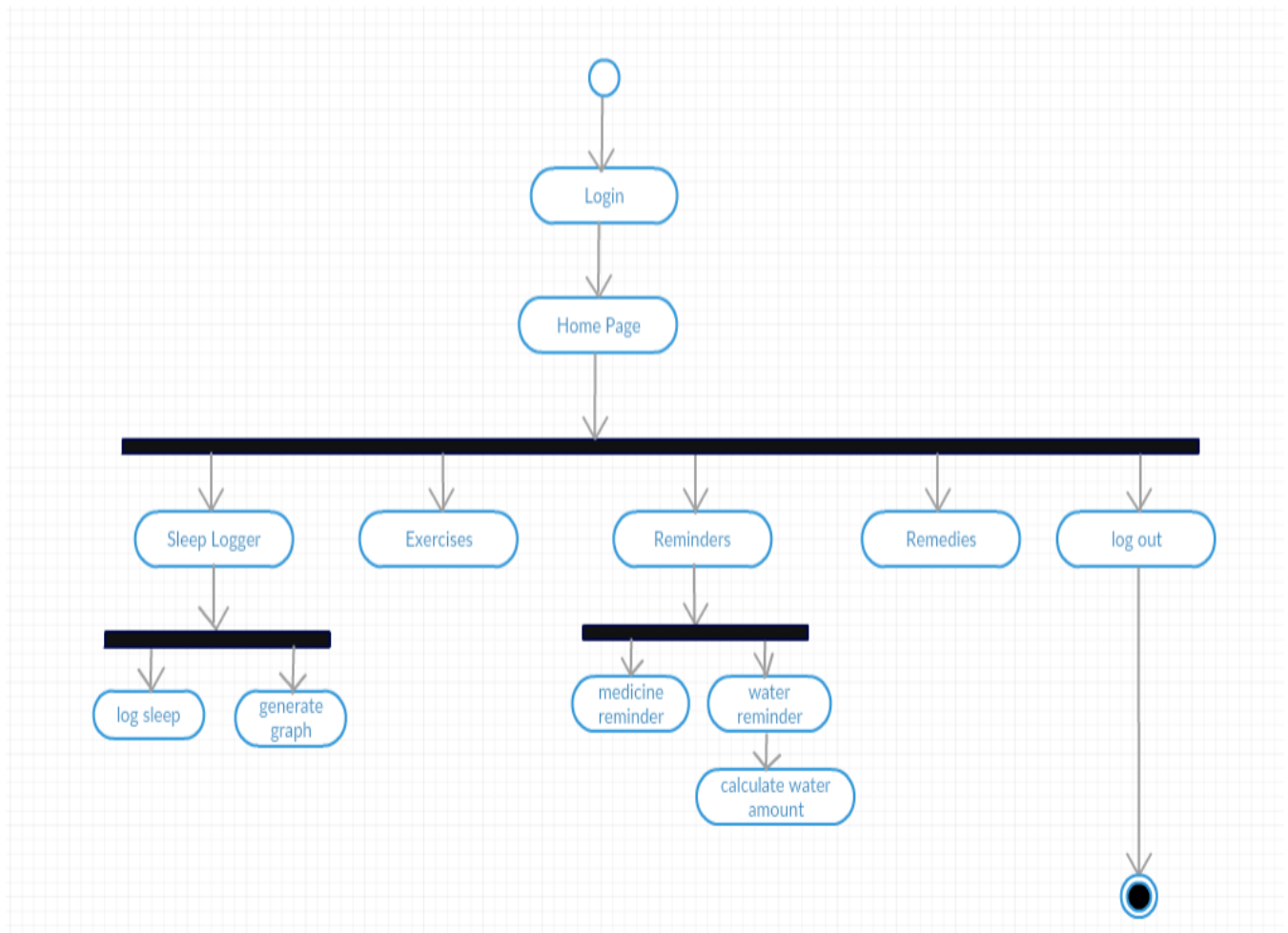


Figure 3-2 Activity Diagram

Chapter 4 - Building the System

This system consists of different modules which include login, exercises, sleep logger, newsfeed, reminder and remedies. The most important modules in this system are sleep logger and reminder section.

4.1 Login Module

The application starts with a splash screen that displays the logo and title of the app. The next screen that can be seen is the login screen where the user has to enter valid credentials: username and password in order to login successfully. The user can also login through facebook which is integrated into the application using the facebook SDK. When the user tries to login through facebook, information about the facebook id of the user is requested. Access token is generated which can be used by the user to login in. The input fields are validated when the user enters details. If the user does not have any login credentials, he has to register first. When the user clicks on the register button, details like email, username and password should be provided. For this purpose, there is a login and register activity.

4.2 Home Screen Module

Once the user logs in successfully, a home screen is displayed. This home screen consists of some health tips for the user to read if he scrolls down the page. The content in the home screen is static. On the bottom of the screen, there is a menu from which the user can navigate to different screens. The implementation of this module is done mostly in the newsfeed fragment and newsfeed adapter.

4.3 Exercise Module

When the user navigates to the exercise screen, he can see a list of exercises along with the instructions on how to do it. Each exercise also provides a link to watch the exercise if the

instructions are not clear for any user. It provides flexibility for the user to choose how he wants to learn the exercise listed on the screen. There is difficulty level associated with each exercise that is displayed to let the user choose according to his comfort level and stamina. The implementation of this module is done mostly in the exercise fragment and exercise adapter.

4.4 Sleep Logger Module

In the sleep logger section, the user has to input the time at which he slept and the time at which he got up on a selected date. After the user saves these times, he will be able to see the number of hours slept. Once the user logs sleep data for at least 7 days, he will be able to pick a date from which data for the next 6 days is logged. When the user picks a date, a weekly sleep graph is generated starting from the selected date. The graph is generated based on the number of hours the user slept each day. After the graph is generated, the sleep pattern which can be either regular, irregular or moderately irregular is displayed and the average number of hours the user slept each day is displayed. The sleep pattern is categorized into one of the three choices available based on the standard deviation among those values. The implementation of this module is done mostly in the sleep logger fragment, time picker fragment and date picker fragment.

4.5 Remedies Module

Remedies module is very similar to the newsfeed module. This screen displays some natural remedies to common health problems for the user to read if he scrolls down the page. The content in the remedies screen is static. The user can tap on each remedy to read in detail. The implementation of this module is done mostly in the remedies fragment and remedies adapter.

4.6 Reminder Module

The reminder screen is divided into portions horizontally. The top portion is the water reminder and the bottom one is the medicine reminder. In the water reminder, the user has to enter age, gender, weight and amount of physical activity he does: low, medium and high. Based on these inputs, the amount of water to drink daily by that user is calculated. This calculation is done based on the information from web resources. Once the total water amount is calculated, it is divided into a certain number of portions equally. An hourly notification pops up with the calculated portion of water until the daily limit is reached. For now, I set it in a way that water reminder starts at 8 AM and ends at 10 PM. So, an hourly reminder pops up as a notification with the portion of water to be taken in that hour. The water notifications should be enabled to remind the user to drink water. This data is stored in the shared preference.

In the medicine reminder section, the user will input the medicine to be taken along with start and end dates of the course duration for that medicine. The times of day at which the medicine should be taken is also logged. Once this information is given and the medicine reminder is enabled, the user gets notifications at the specified times until the course duration is over. Also, the user will be able to enter multiple medicines with different times and duration. The user will also be able to delete an existing medicine from the list to remind. The implementation of this module is done mostly in the reminder fragment, date picker fragment, time picker fragment, time list fragment, medicine times adapter, MedicineAlarmManager.java and MyAlarmManager.java classes.

4.7 Lines of Code

This project has 2289 lines of code in java and 1116 lines of xml code.

Chapter 5 - Using the System

This system has a newsfeed section, sleep logger section, reminder section, exercise and remedies section. Below is the description of each module along with the login module

5.1 Login Screen

The first screen of the application has the login form. The application starts with a logo screen and opens the login page. You will see the option login and register on this page. If the user is already registered, user can directly login with the credentials. If not, click on the register button which will direct the user to the register page. Once the user successfully registers, he will be automatically logged in for the first time. This is the only way the user can enter the system.

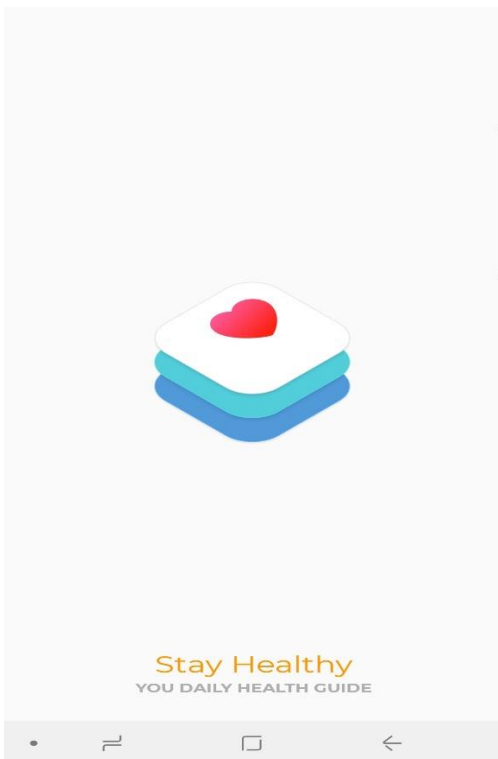


Figure 5-1 Splash screen

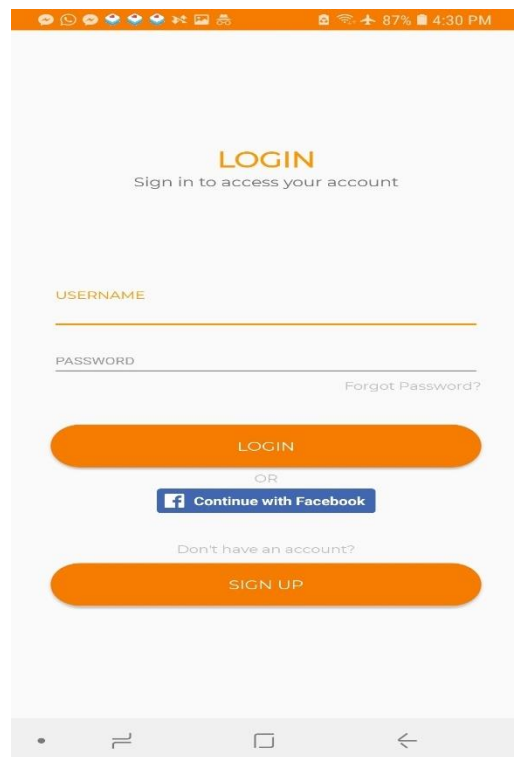


Figure 5-2 Login screen

5.1.1 Login screen validation:

When users enter incorrect credentials, it should not allow the user to log in. Instead, it should show a message saying invalid username or password.

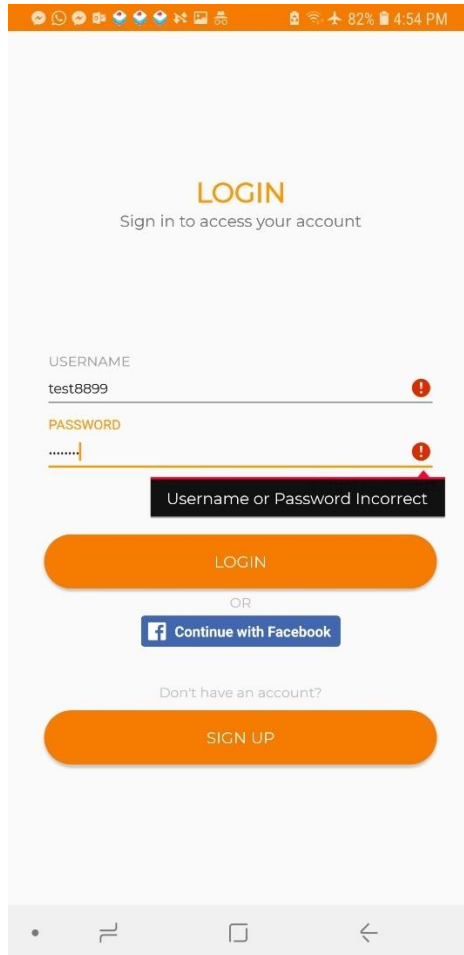


Figure 5-3 Invalid User Details

5.2 Registration Screen

A user has to register first before logging in. In order to register, details like email, username and password should be given. Once the user clicks on the register button and registers successfully, the user will be able to login with those credentials. The registration page and its validation for password matching is shown in the below screenshots.

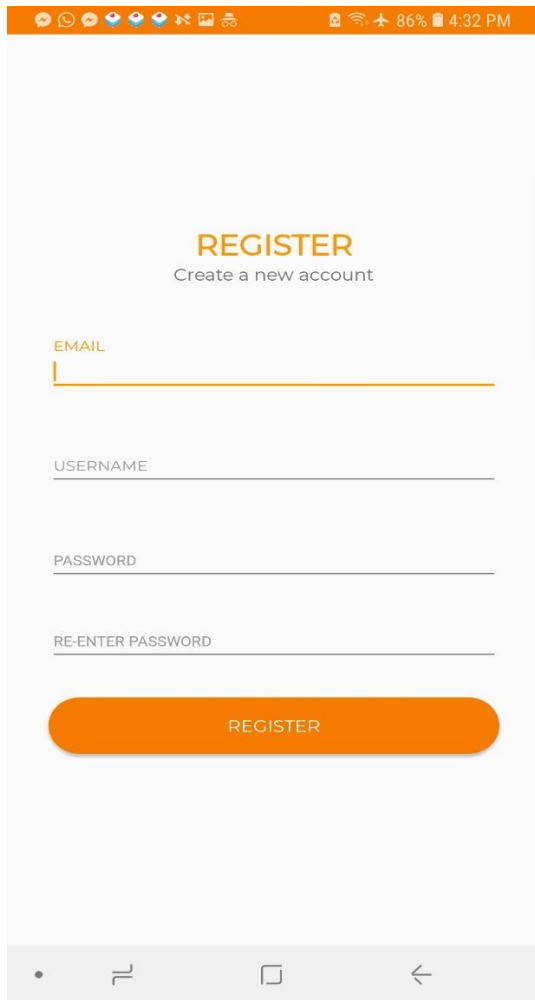


Figure 5-4 Registration screen

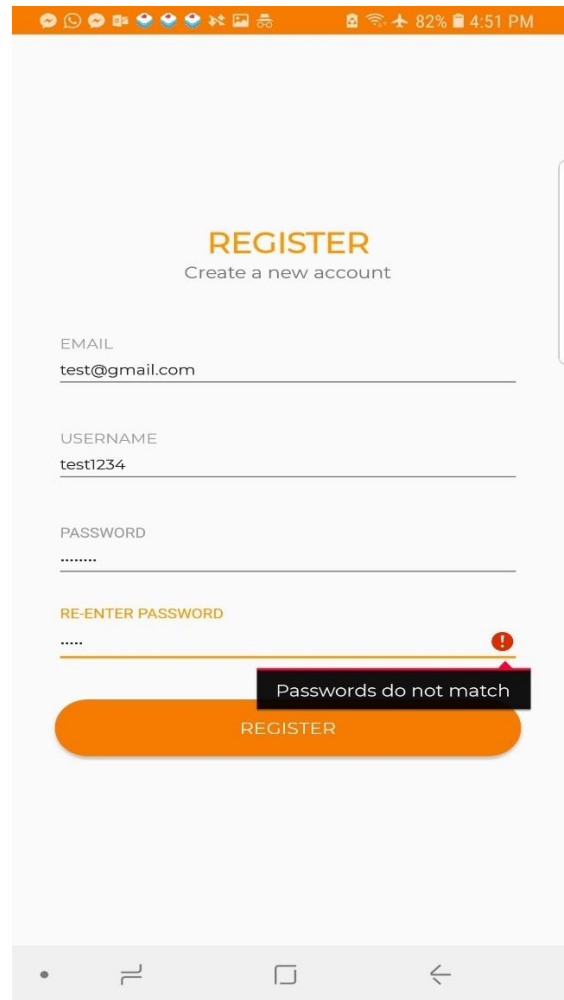


Figure 5-5 Password Validation screen

5.3 User Screen

After successful login, the user will be navigated to the home page which consists of some health tips for the user to check. This will be a static page, and the user will only be able to check the health tips listed on this newsfeed page and read the content. The user has to scroll down the list to look at different health tips. When the user clicks on each tip, the content of that health tip will be displayed completely. The newsfeed page looks like the below screenshots.

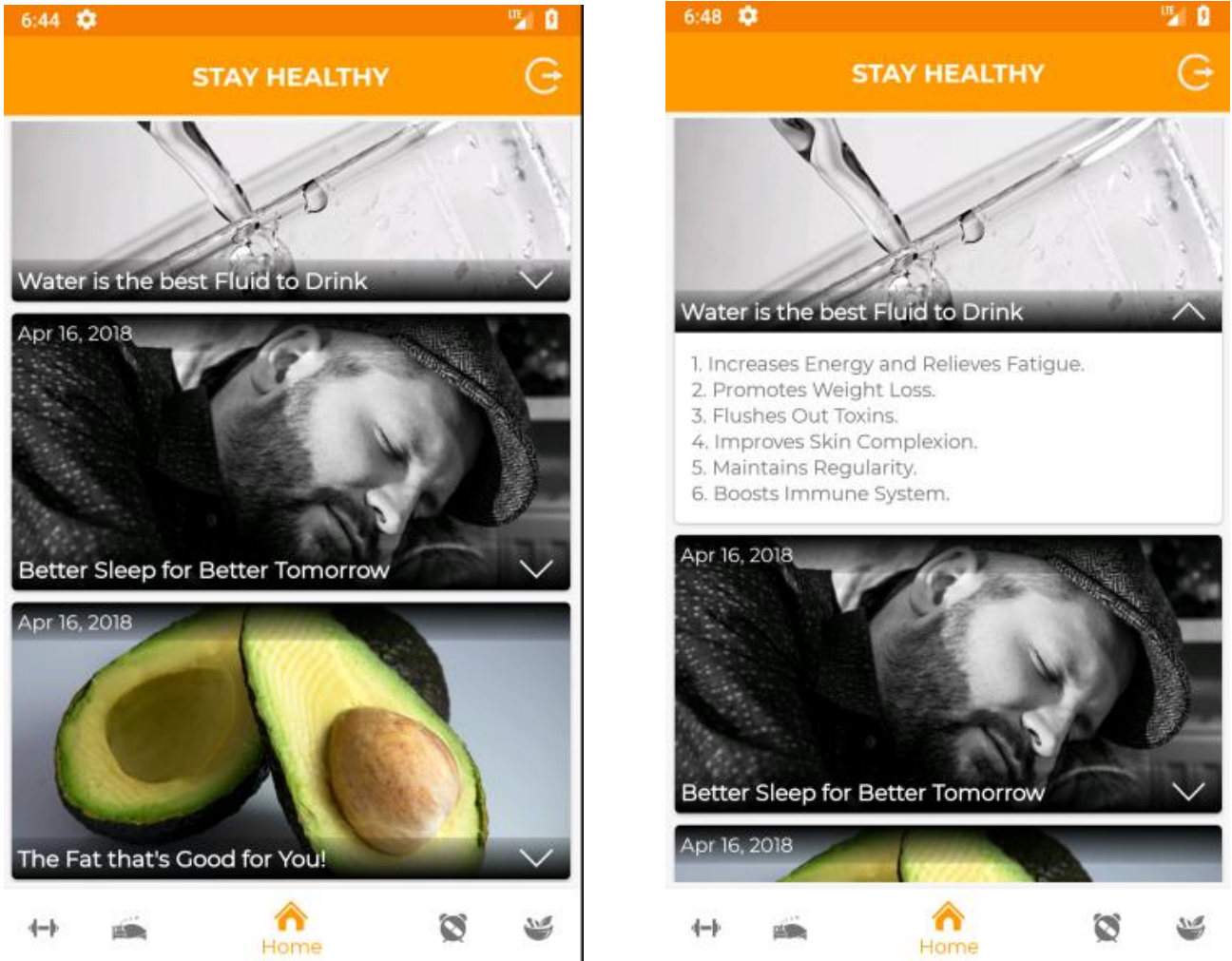


Figure 5-6 User Home Page

5.4 Sleep Logger

5.4.1. Sleep Duration Calculator

Sleep logger module will need the user to select the date to log sleep times. The user needs to provide inputs for slept at and woke up at fields. When the user clicks on the save button, the total number of hours the user slept will be calculated and displayed beside the sleep duration label. The screenshots below will show the described functionality.



Figure 5-7 sleep logger screen

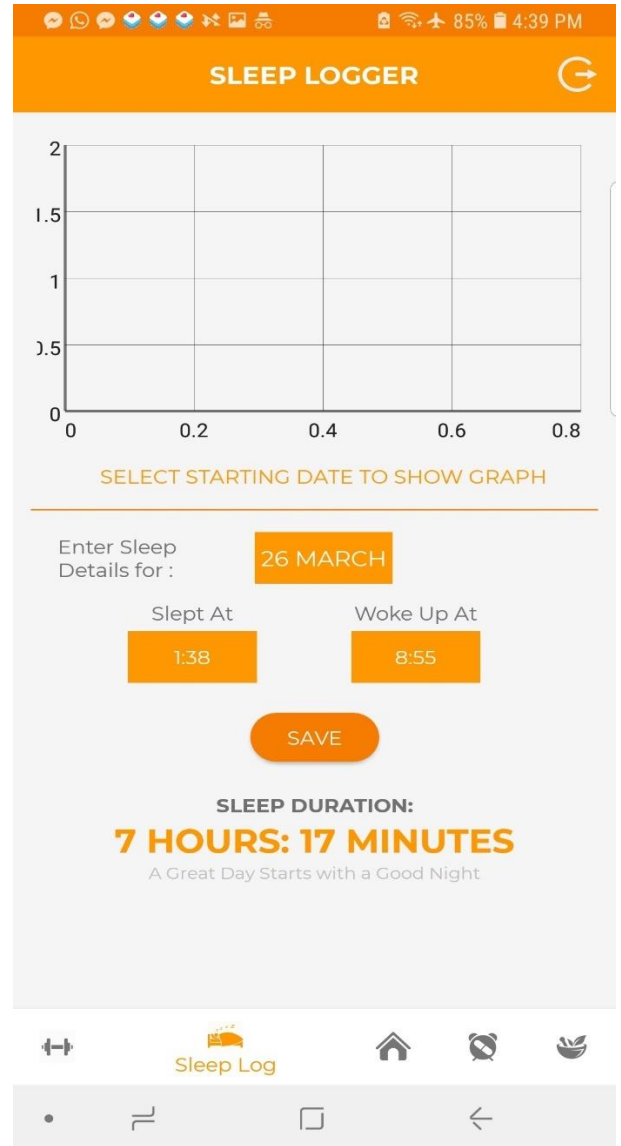


Figure 5-8 Logging sleep times

5.4.2. Time pickers

A user has the flexibility to set the time in two ways. One way is to pick the time from the clock and the second way is to enter the time using keyboard input which is below in the screenshot.

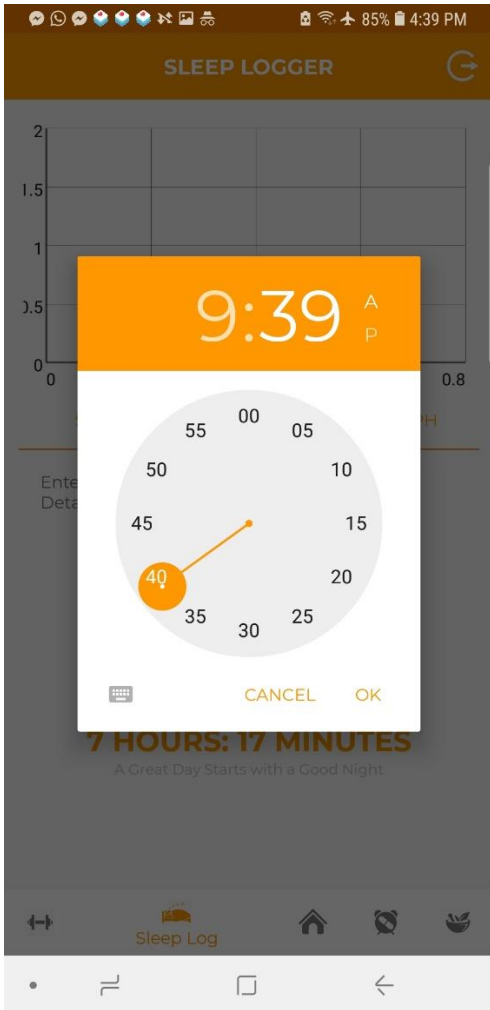


Figure 5-9 Time picker 1

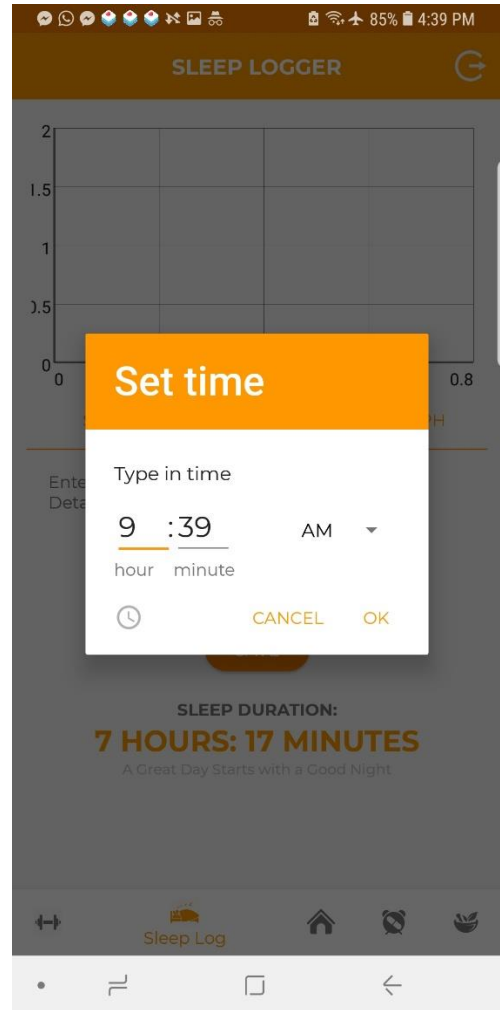


Figure 5-10 Time picker 2

5.4.3. Sleep Graph Generator and Sleep Analysis

Also, in the sleep logger section after a week's data is saved, the user should be able to pick a date from the calendar and generate a graph based on sleep duration to analyse the user's sleep pattern which can be either regular, moderately irregular or irregular and average number of hours the user slept during that week. The screenshots below show this functionality.

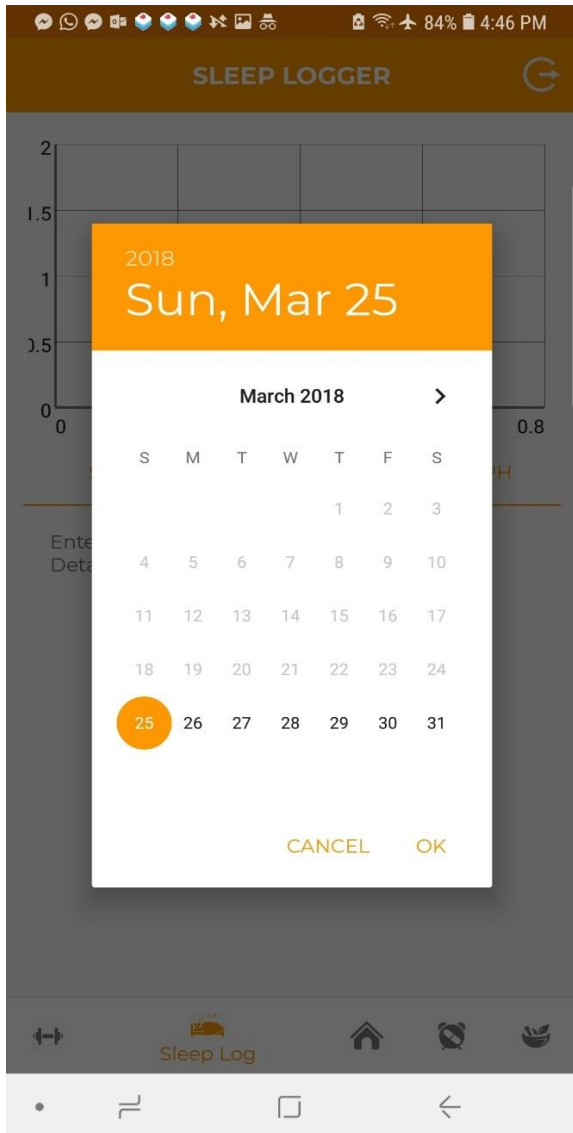


Figure 5-11 Date picker

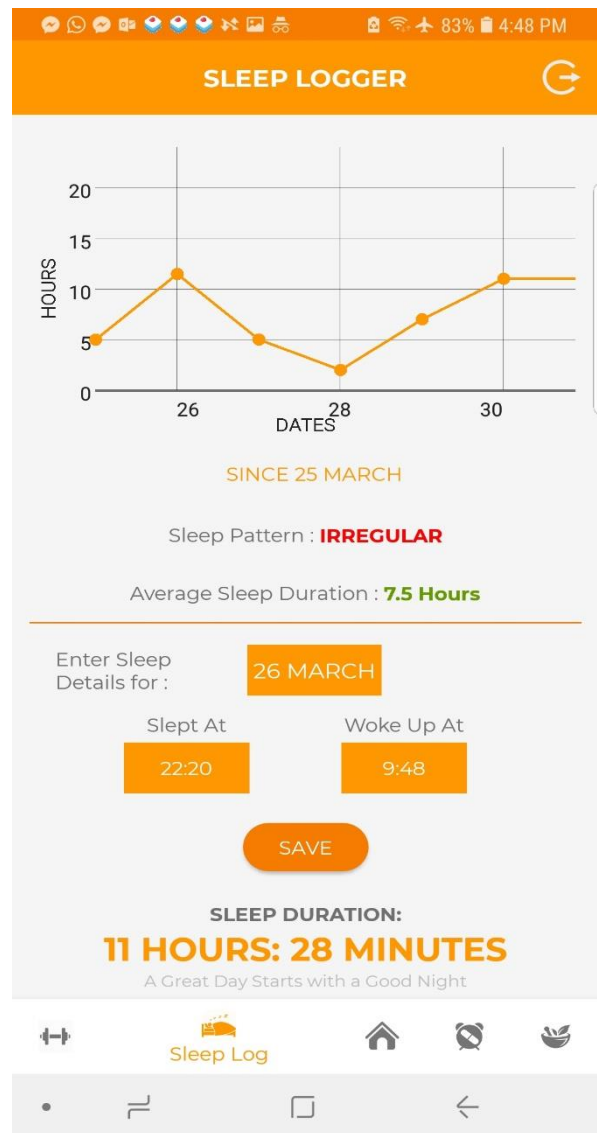


Figure 5-12 Sleep analysis

5.5 Exercises Screen

When the user navigates to the exercise page after clicking on the exercise button on the menu bar at the bottom, a list of exercises will be shown on the page along with the instructions and links to the videos on how to do those exercises.

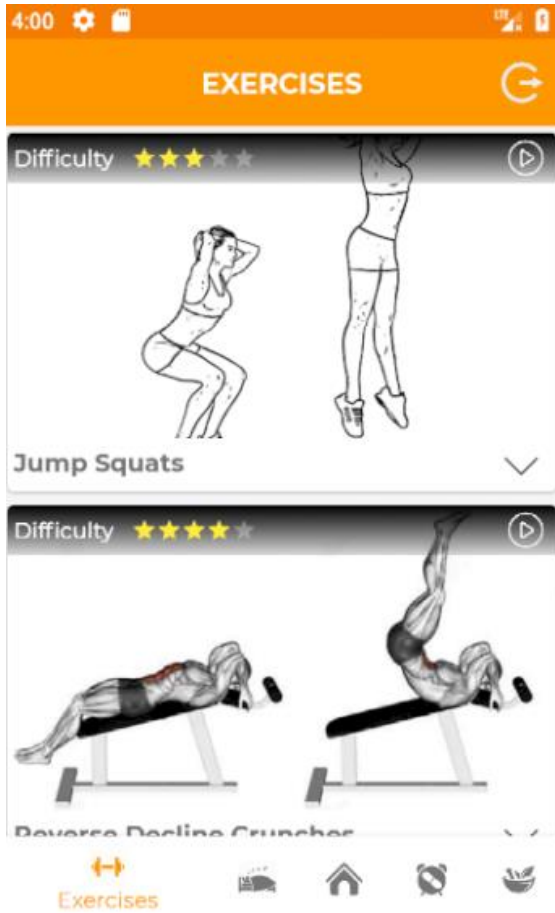


Figure 5-13 Exercise screen

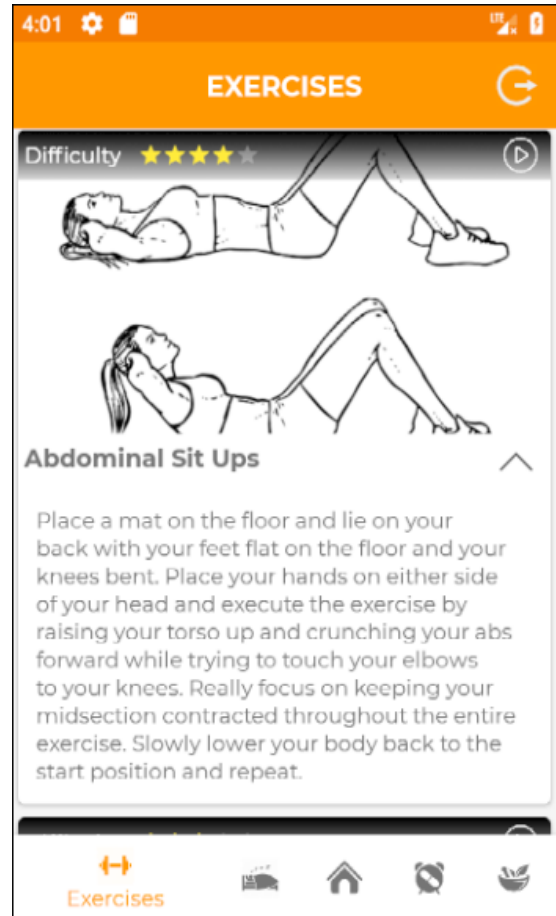


Figure 5-14 Exercise Content

5.6 Remedies Screen

As this application also deals with medicine reminder, remedies module is added to provide some natural remedies to the most basic health problems that can be treated without the use of any medicines. The user has to scroll down the page to see the listed remedies. The following screenshots show the remedies page.



Figure 5-15 Remedies screen

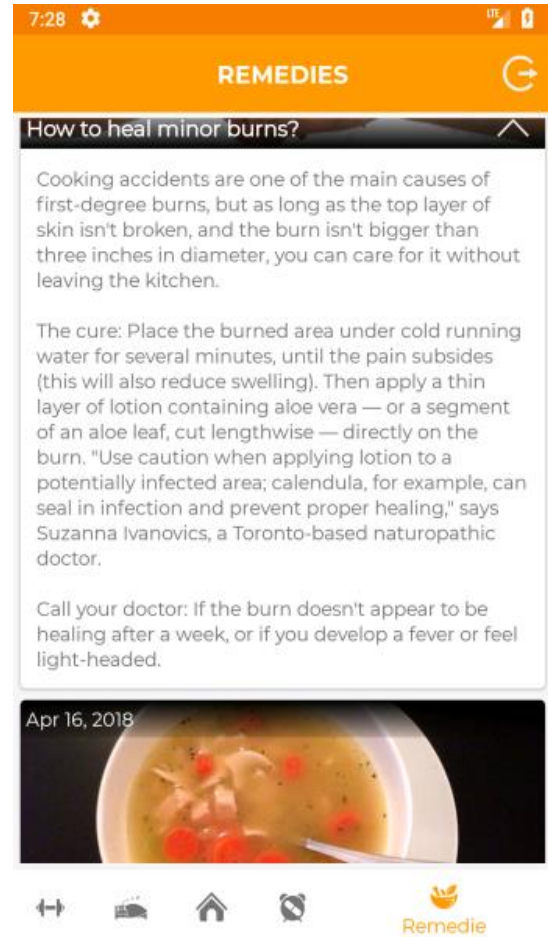


Figure 5-16 Remedies content

5.7 Reminder Screen

This module of the application is divided into two portions: water reminder and medicine reminder. The water reminder portion should be able to remind the user to drink water by popping a notification every hour with the information of how much water should be consumed. The medicine reminder should remind the user to take medicines by popping a notification with the medicine name. The validations for the input boxes are shown with the following screenshot.

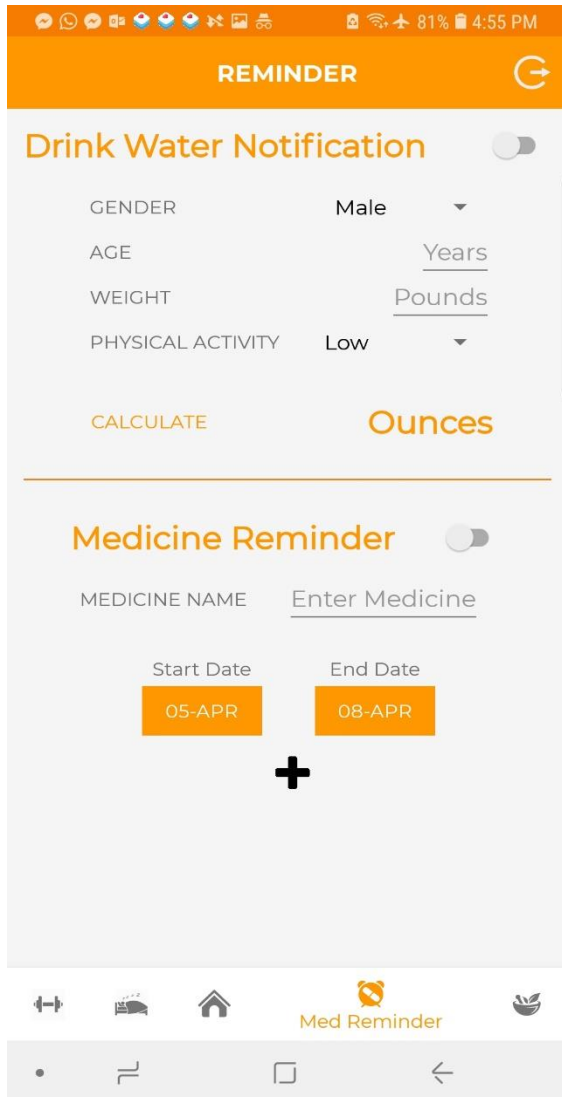


Figure 5-17 Reminder screen

5.7.1 Water Reminder

Water reminder portion will take inputs like age, gender, weight and physical activity rate to calculate and display the amount of water that should be taken by the user daily. After the water notifications are turned on, the user should get a reminder every hour in form of a notification on the screen within the specified time slot. The whole amount of water that should be taken by the user is divided into hourly portions to keep track of the amount of water consumed. The screenshots of the water reminder are as follows.

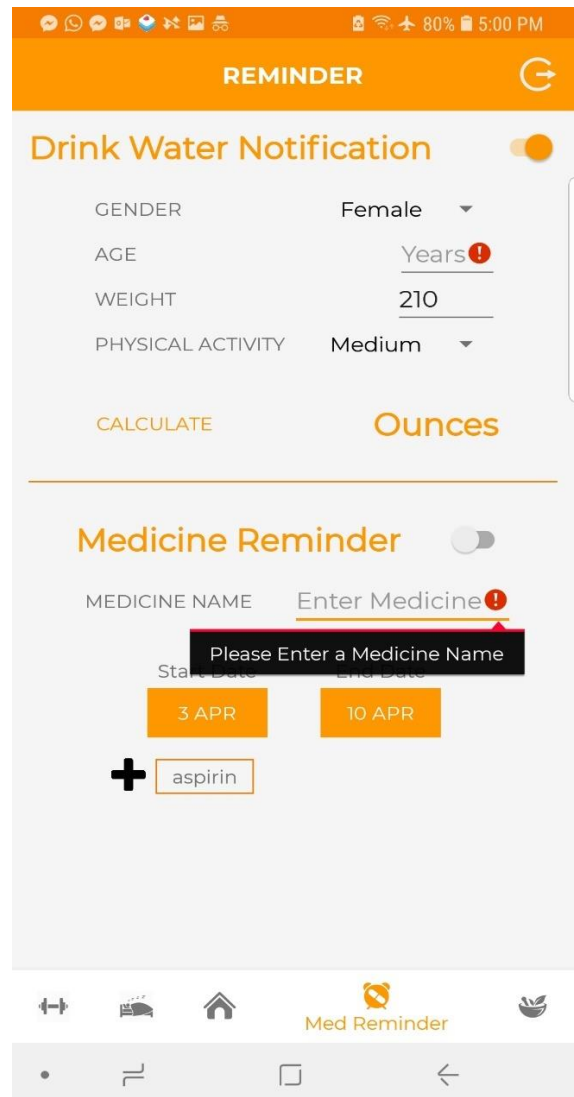


Figure 5-18 Validation of inputs

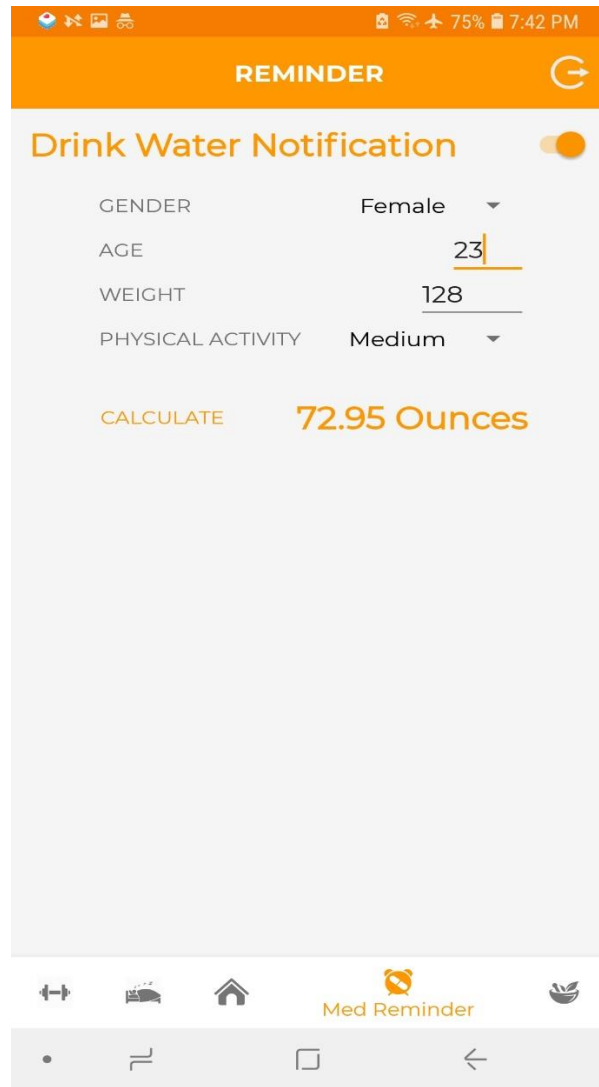


Figure 5.19 - Water Reminder Portion

5.7.2 Medicine reminder

The medicine reminder portion will take medicine name, start date, end date and the times at which the medicine should be taken during the day as inputs. A user will be able to add more than one medicine in the reminder section. After turning on the medicine reminder, the user should receive notification when it is time to take a medicine to remind the user. The following screenshots will show the medicine reminder portion.

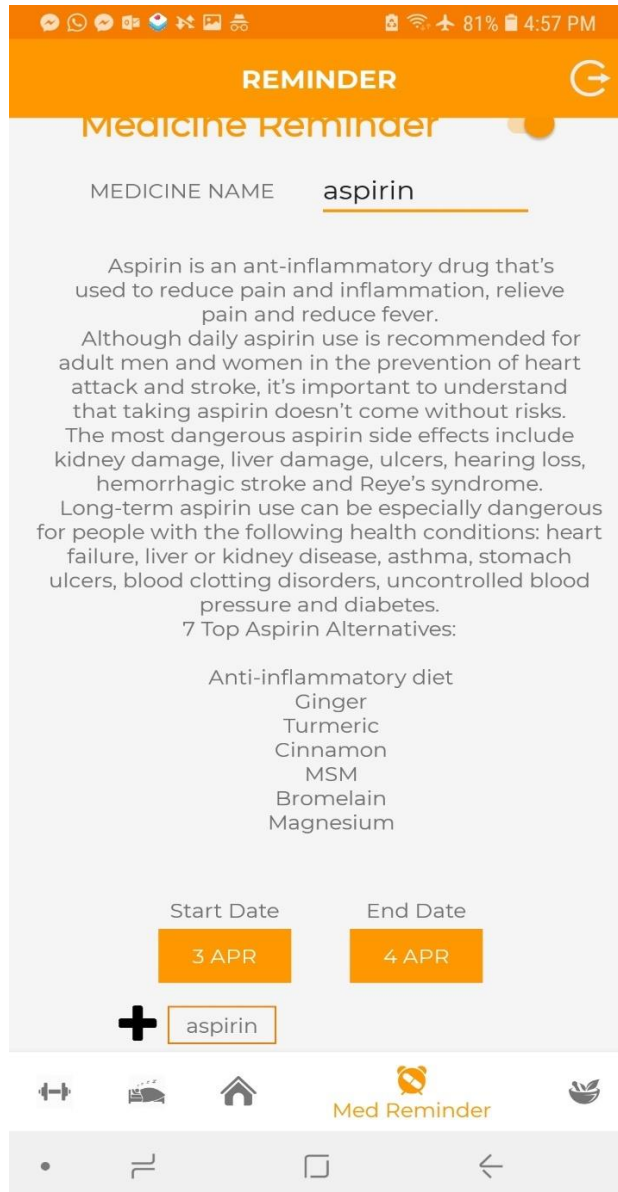


Figure 5-20 Medicine Reminder Portion

Chapter 6 - Security

Security is one of the most important factors when building an application. It is implemented in a system with an idea to protect it from hackers, any threats and other malicious attacks. The software should function correctly even under potential risks without any impact on it. It is necessary to provide integrity, availability and authenticity. Compromising on any one of these factors can put your system into risk and make it unsecure to use. When a system is attacked due to poor security, it leads to compromising important information of the user, damage the behaviour of the system and may cause the system to crash in some cases. The security of the system should be improved by finding faults beforehand, fixing them and preventing any security vulnerabilities. For this application, the data is not sensitive. So, I worked on other parts to ensure security.

6.1 Password Encryption

The password has been encrypted using the technique of hashing and then stored in the database. Now, all the passwords are encrypted in the database. Anybody who has access to the database will not be able to see the user passwords and misuse them.

6.2 Facebook token validation

When a app is connected using facebook login and the requested access is approved, it provides the app with an access token that gives access and provides security to facebook APIs. Access token is a string that identifies the user or the app and can be used by the app to call graph API calls. Android automatically manages user access tokens through `com.facebook.AccessToken`. This token can be retrieved by inspecting `FBSDKAccessToken.currentAccessToken` [8].

6.3 Proguard Tools for code obfuscation

Proguard is used to optimize and obfuscate code for java applications on mobile devices. It reduces the start time of the application and improves its performance. It ensures minimum protection against reverse engineering. [9]

6.4 SQL Injection:

SQL injection is a technique that can destroy the database of the application by injecting unwanted code in the SQL statements which is harmful to the system. It is one of the most common hacking techniques used on the applications. These injections can change the SQL statement and compromise the security of the application. This is usually done through input boxes or query strings. Giving SQL command that is malicious in the input box will lead to unfair access. For example, in the login inputs, one can inject malicious SQL commands and get access of the application and data. This can compromise the passwords of the user due to poor validation. One common approach to prevent SQL injections is to validate the user inputs so that SQL commands are not accepted as inputs. [10]

Chapter 7 - Testing

Testing is done to evaluate the application based on the requirements obtained from the user and system specifications. Testing should be done in a modular level in the software code. It is always suggested to test the software to check if it has met the business and technical requirements at an early stage. If a bug occurs at the end product, it becomes difficult and expensive to fix it. The complexity level to fix the bugs will also increase if one waits for the product to finish to start testing. The main targets for testing are to look for errors made by the developer in the code, difference between the expected output and obtained output in the software, to look for any bugs that cause the system to fail and to look for failures in the system that occur due to faults in the system.

Software testing consists of validation and verification. Validation is the process of checking to see if all the user requirements are satisfied in the system. It makes sure that the end product is obtained in a way that satisfies the user requirements. And, verification insists on the system specifications and design. It checks to see if the product being designed is meeting the system specifications. Again, testing can be manual or automated. In manual testing, the tester prepares different test cases to check if the code passes those at different levels and reports the results. In automated testing, different testing tools are used to test the code. For this project, unit testing, integration testing and performance testing is performed. [11]

7.1 Unit Testing

Unit testing is a technique in which individual components of the software are tested to check for any errors in the system. It is the smallest testable part of the system. In this, each test case is different

from the rest and will test to see that individual parts of the program are correct they are created by programmers during the development process to test and see if the code meets its design and specifications [12]. Following table shows the test cases used to perform the testing.

S.No	Test Case	Expected Result	Result
1	User should be able to log in with correct credentials	User should successfully login	Pass
2	User should not be able to log in with wrong credentials	User should not Login	Pass
3	New user should be able to register successfully	New user should be successfully registered	Pass
4	After successful login, user should be displayed news feed	User should be displayed newsfeed after successful login	Pass
5	Click on exercises section	Exercises section should open	Pass
6	Click on remedies section	Remedies section should open	Pass
7	Click on sleep logger section	Sleep logger section should open	Pass
8	When user inputs sleep and wake up times for a selected date, the number of hours slept should be calculated and displayed	User should see the total number of hours slept on a selected date after logging sleep times	Pass
9	User should be able to save sleep times input on selected dates	Sleep times of the user should be saved on the selected dates	Pass

10	On selecting a date from the calendar, the user should be shown a weekly graph based on total number of slept on each day from that date	User should see the weekly graph based on the number of hours slept from selected date	Pass
11	A graph should not be generated without enough data	Graph is not generated without enough data	Pass
12	On generating the sleep graph, sleep pattern and average hours slept in that week should be displayed	Sleep pattern and average hours slept should be displayed after the graph is generated	Pass
13	Open reminders section	Reminders section should be open	Pass
14	In the water reminder portion, user should input details to calculate the amount of water to drink daily	The amount of water to drink daily should be displayed	Pass
15	User should be able to turn on water notifications	Water notifications should successfully turn on	Pass
16	User should provide inputs in the medicine reminder portion	User inputs should successfully be saved	Pass
17	User should be able to add multiple medicines	Multiple medicines added successfully	Pass
18	User should be able to select different dates and times for each medicine	Different dates and times can be selected for each medicine	Pass
19	User should be able to turn on medicine notifications	Medicine notifications should successfully turn on	Pass

20	User should be able to login with FB account	User logged in successfully	Pass
21	On clicking on Logout user should log out	User should log out	Pass

Table 7-1 Unit Testing for Application

7.2 Integration Testing

In integration testing, individual units are combined and tested as a group together. This is done after the unit testing. This is done to check that the application works correctly even after the integration of different units together. It is done to bring out the faults between the integrated units when they interact with each other. This testing done by the developers or independent testers. [13]

Following table shows the integrated test cases for this application.

S.No	Test Case	Expected Result	Result
1	On clicking register button, register screen should load	Register screen should load	Pass
2	On successful login user should go to home screen	Home screen should load	Pass
3	On clicking exercise button, exercise screen should load	Exercise screen should load	Pass
4	On clicking on the play icon on one of the exercises listed, the user should navigate to the exercise video	Exercise video should load	Pass

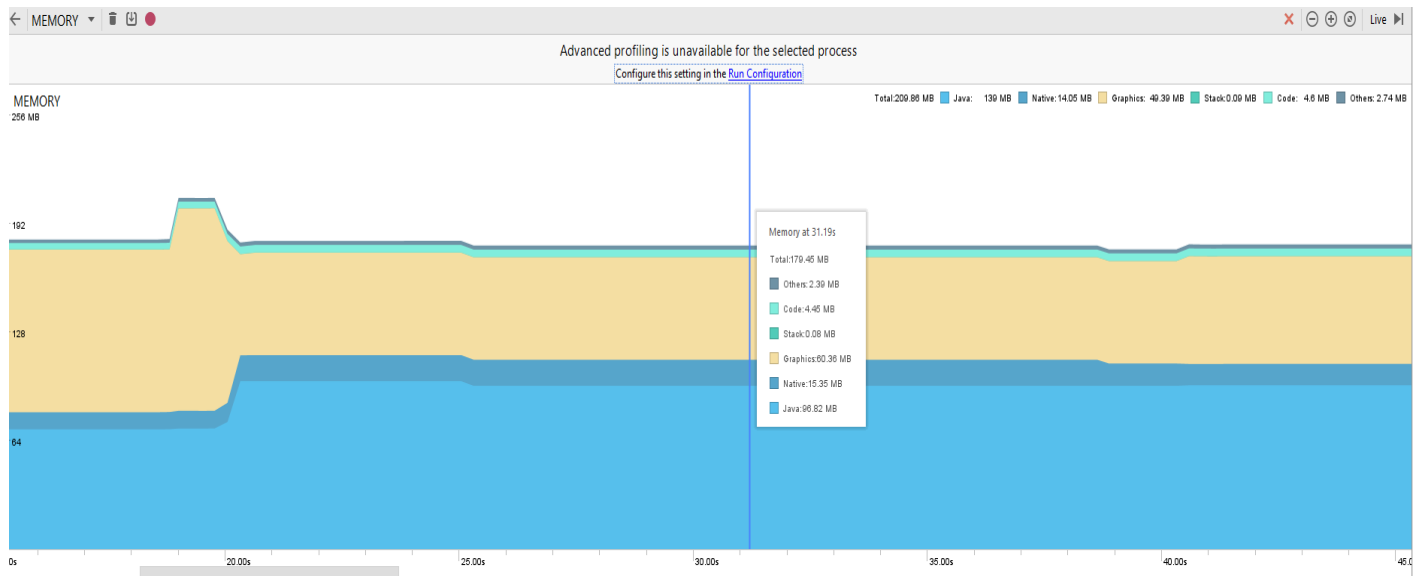
5	On clicking on remedies button, remedies screen should load	Remedies screen should load	Pass
6	On clicking on Sleep logger button, sleep logger screen should load	Sleep logger screen should load	Pass
7	On sleep logger section, after selecting a valid date, graph should	On valid date input, sleep graph should load	Pass
8	On loading sleep graph, sleep pattern type and average number of hours slept should be visible	Average amount of hours slept are calculated and sleep pattern type displayed	Pass
9	On selecting new valid date in sleep logger, new graph should be loaded	Updated graph should load	Pass
10	On clicking on Reminders button, reminder screen should load	Reminder screen should load	Pass
11	On reminder screen, after entering valid details on water reminder portion, daily water consumption amount should be calculated	On entering valid details on water reminder section, amount of water to drink daily is calculated	Pass
12	On reminder screen, after entering valid details on medicine reminder portion and turning on the reminder, notifications should pop up at the entered times.	Medicine notifications successfully given at input times	Pass
13	On turning on water reminder, notifications should be displayed every hour to consume some amount of water	Water notifications successfully given every hour within the specified time slot	Pass

Table 7-2 Integration Testing

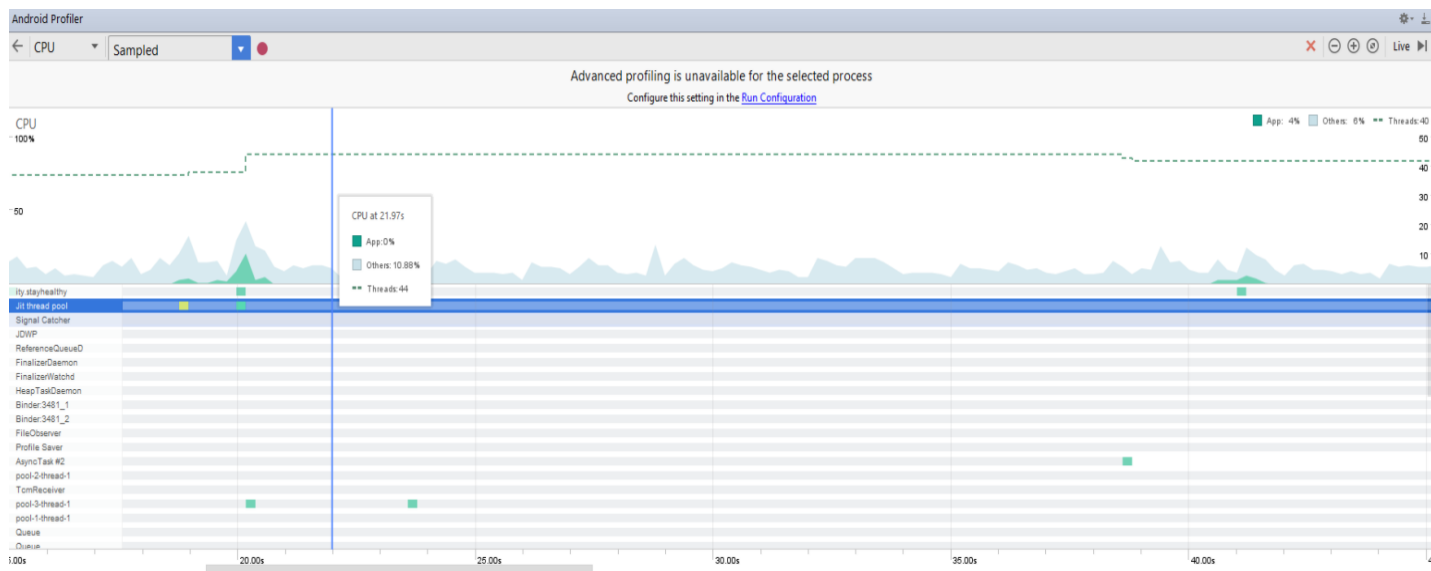
7.3 Performance Testing

Performance testing is done to check the responsiveness and stability of the application under different workloads. It is used as a measure to test the quality and capability of the system in terms of resource utilization, scalability and reliability. In this application, testing is done to check the CPU and memory usage. [14] The results obtained are as follows.

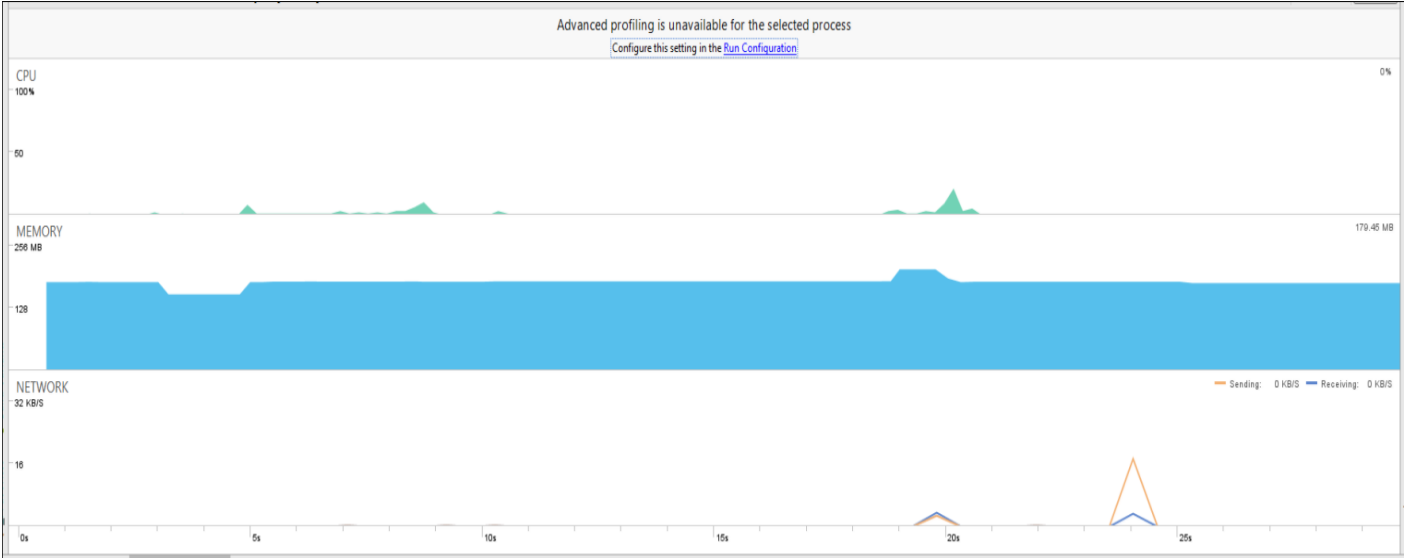
Memory Usage



CPU Usage



CPU, Memory and Network Usage



7.4 Accessibility for elders

In order to ensure that it is accessible to elderly people, the controls that the user will be using are set to a minimum of 48dp width and length. Another important factor is the talkback audio prompts which function correctly when used for this application. It is a voice assistant that works properly on this application after testing it manually. As this application has water and medicine reminder, the alerts will have a sound along with a visual notification. Also, the alerts can be made more accessible by turning on the flashlight when an alert pops up by using the feature available for applications. The exercises module is flexible and more accessible to users as there is content available for the instructions to the video along with a link to the video which has the instructions. It can be an advantage to people who cannot read the content. Also, it is useful for people who can just see the video. This application can be accessed using directional controls to do primary tasks using the emulator with keyboard. These are some of the features can help the elderly people or people with disabilities to use this application. [15]

Chapter 8 - Conclusion and Future Work

8.1 Conclusion

This application is very convenient for users who want to focus on the overall health improvement. This application combines different factors like sleep, exercises, medicine & water reminders, health tips and natural remedies that contribute to the betterment of the user's health. In today's busy world, it is difficult for a lot of people to keep track of these things. This application focuses on improving the sleep patterns of the user by giving analysis of his sleep cycle, reminds the user to stay hydrated and take medicines on time, and provided them with health tips, natural remedies and exercises which will improve the health of the user.

While developing this application, I learnt android development and understood its components. The process of developing an application undergoes different phases like requirements gathering, design, development and testing. Stay Healthy application gave me an overall insight of how important each of these phases are.

8.2 Future Work

This Application can be improved further by integrating APIs to provide different health tips on the newsfeed and update remedies. An API can be integrated to give useful information like purpose, side effects, and alternatives of the drug when it is entered by the user. Apart from API integration, improvements can be made in the sleep logger section by automatically generating a dynamic graph without the user picking a date. It should show an updated graph every day based on user inputs. Also, the sleep pattern generated should depend on the time at which the user slept instead of just depending on the number of hours slept in future work.

Chapter 9 - References

- [1] “Android Studio user guide,” Android Studio, [Online]. Available: <https://developer.android.com/studio/intro/index.html>. [Accessed January 2018].
- [2] “Android Studio Architecture,” Android Studio, [Online]. Available: <https://developer.android.com/guide/platform/index.html>. [Accessed January 2018].
- [3] “SQLite,” Wikipedia, [Online]. Available: <https://en.wikipedia.org/wiki/SQLite> . [Accessed Feb 2018].
- [4] “Save Data Using SQLite,” Android Studio, [Online]. Available: <https://developer.android.com/training/data-storage/sqlite.html>. [Accessed Feb 2018].
- [5] “Android Shared preferences,” [Online]. Available: https://www.tutorialspoint.com/android/android_shared_preferences.htm. [Accessed Feb 2018].
- [6] “UML Tutorial,” [Online]. Available: https://www.tutorialspoint.com/uml/uml_class_diagram.htm . [Accessed Feb 2018].
- [7] “Activity Diagram,” Tutorials Point, [Online]. Available: https://www.tutorialspoint.com/uml/uml_activity_diagram.htm. [Accessed Feb 2018].
- [8] “Access Tokens,” Facebook, [Online]. Available: <https://developers.facebook.com/docs/facebook-login/access-tokens#usertokens>. [Accessed March 2018].
- [9] “ProGaurd,” Gaurd Square, [Online]. Available: <https://www.guardsquare.com/en/proguard>. [Accessed March 2018].
- [10] “SQL Injection,” W3 Schools, [Online]. Available: https://www.w3schools.com/sql/sql_injection.asp . [Accessed March 2018].
- [11] “Software Testing Overview,” [Online]. Available: https://www.tutorialspoint.com/software_engineering/software_testing_overview.htm. [Accessed March 2018].
- [12] “Unit Testing,” Software Testing Fundamentals. [Online]. [Accessed March 2018].
- [13] “Integration Testing,” Software Testing Fundamentals, [Online]. Available: <http://softwaretestingfundamentals.com/integration-testing/>. [Accessed march 2018].
- [14] “Performance Testing,” Tutorials Point. [Online]. [Accessed March 2018].
- [15] “Testing Your App's Accessibility,” Andorid Studio, [Online]. Available: <https://developer.android.com/training/accessibility/testing.html>. [Accessed 07 April 2018].

[16] “Creative Commons Images,” Creative Commons, [Online]. Available: <https://search.creativecommons.org/>. [Accessed 2018].