

COMPLAIN BOX ANDROID APPLICATION

BY

**MD. ASHIKUR RAHMAN
ID: 161-15-7058**

**MD. NAIMUL ZANNAT
ID: 163-15-8315**

AND

**MD. NOOR NOBI JUWEL
ID: 163-15-8341**

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering.

Supervised By

Mr. AHMED AL MAROUF
Lecturer
Department of CSE
Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY DHAKA,
BANGLADESH SEPTEMBER 2020**

APPROVAL

This Project/internship titled “**COMPLAIN BOX ANDROID APPLICATION**”, submitted by Md. Ashikur Rahman, ID No: 161-15-7058, Md. Naimul Zannat, ID No: 163-15-8315 and Md. Noor Nobil Jewel, ID No: 163-15-8341 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 Thursday, 8th October 2020.

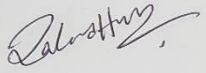
BOARD OF EXAMINERS



Dr. Syed Akhter Hossain
Professor and Head

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

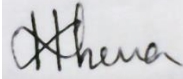
Chairman



Md. Zahid Hasan
Assistant Professor

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

Internal Examiner



Most. Hasna Hena
Assistant Professor

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

Internal Examiner



Dr. Mohammad Shorif Uddin
Professor

Department of Computer Science and Engineering
Jahangirnagar University

External Examiner

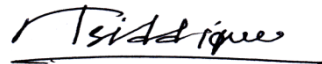
DECLARATION

We hereby declare that, this project has been done by us under the supervision of Mr. Ahmed Al Marouf, Lecturer, Department of CSE, Daffodil International University. We 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:

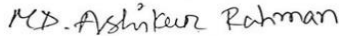


Ahmed Al Marouf
Lecturer
Department of CSE
Daffodil International University



Shah Md. Tanvir Siddiquee
Assistant Professor
Department of CSE
Daffodil International University

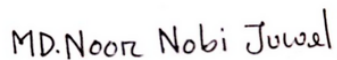
Submitted by:



Md. Ashikur Rahman
ID: 161-15-7058
Department of CSE
Daffodil International University



Md. Naimul Zannat
ID: 163-15-8315
Department of CSE
Daffodil International University



Md. Noor Nobi Jewel
ID: 163-15-8341
Department of CSE
Daffodil International University

ACKNOWLEDGMENT

At first we would like to thank to the Almighty for His blessing makes us possible to complete the final year project successfully.

We are grateful and wish our profound our indebtedness to **Md. Ahmed Al Marouf, Lecturer**, Department of CSE, Daffodil International University. Deep Knowledge and keen interest of our supervisor in the field of “*Android Application*” to carry out this project. His patience, guidance, encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We also want to thank our department head **Dr. Syed Akhter Hossain** from the deep of our heart for his support in this project.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

This project is based on the theme of complain management system. This is an android application which is very easy to use and users will be able to raise their issues in an efficient way. Nowadays, people always face various kinds of issues regarding society and system. But the problem is, people can not raise this such issues to the representative or Member of Parliament of that particular area in an efficient manner. People can not reach out to them for the lack of communication and time.

Our Android based solution “**Complain Box**” has brought the problem into an wonderful structure. This application is like a social app where normal users will be able to raise their issues according to their desired seats and MP’s of that particular area will be able to interact with that user regarding that issue. Through this app, people can communicate with MP’s in an efficient and faster way. Our main goal is to bring the normal people and representatives of government in a single platform, so that they can communicate to each other and discuss their problems easily.

TABLE OF CONTENTS

CONTENTS	PAGE
Approval	ii
Declaration	iii
Acknowledgments	iv
Abstract	v
CHAPTER	
Chapter 1: Introduction	1-3
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	2
1.4 Expected Outcome	2
1.5 Report Layout	3
Chapter 2: Background	4-6
2.1 Introduction	4
2.2 Comparative Studies	4
2.3 Scope of the Problem	5
2.4 Challenges	6
Chapter 3: Requirement Specification	6-11
3.1 Business Process Modeling	6
3.2 Use Case Model	8
3.3 JSON Data Model	9
3. 4 Design Requirements	11

Chapter 4: Design Specification	12-23
4.1 Front-end Design	12
4.2 Coding Environment and Languages	22
4.3 Back-end Database	22
4.4 How The App Works	22
4.5 Implementation Requirements	23
Chapter 5: Implementation and Testing	24-29
5.1 Project Code	24
5.2 Testing & Result Reporting	29
Chapter 6: Discussion & Conclusion	30-30
6.1 Future Scope	30
6.2 Discussion and Conclusion	30
REFERENCES	31

LIST OF FIGURES

FIGURES	PAGE
Figure 3.1: Business Process Model	7
Figure 3.2: Use Case Model	8
Figure 3.3: JSON Data Model	9
Figure 4.1: Login Screen	12
Figure 4.2: Signup Page	13
Figure 4.3: Password Recovery Page	14
Figure 4.4: User Dashboard	15
Figure 4.5: Filtering drop-down.	16
Figure 4.6: Filtered Posts	17
Figure 4.7: Create Post	18
Figure 4.8: Own Posts	19
Figure 4.9: App Drawer	20
Figure 4.10: Create MP	21
Figure 5.1: AndroidManifest.xml- part 1	24
Figure 5.2: AndroidManifest.xml- part 2	25
Figure 5.3: AndroidManifest.xml- part 3	25
Figure 5.4: Posts View Model	26
Figure 5.5: Comment View Model	27
Figure 5.6: Build Gradle	28
Figure 5.7: Gradle Dependencies	28

CHAPTER 1

Introduction

1.1 Introduction

This is an era of modern science and technology. Every one is looking forward to find an easy solution to bring the world in a single piece of smart device. Currently, the most of the people of our world is depended on smart devices. Android based application are the most popular among all types of users. An android application can be reached out to people so easily and so rapidly. Almost 81% of smart phone users in all over the world prefer to use android os based smart phone. This is the reason, we choose to develop an android solution for our country to bring the politician and normal people into a single platform, so that everyone can interact with politicians so easily regarding all kind of issues they are facing and they intend to get a solution from their policy makers.

This chapter mainly describes, why did we develop this application, how did we get motivation and what was our objectives through out the project.

1.2 Motivation

Smart OS based smart phone is the wonder of modern era of technology. Nowadays, we are so much involved in using smart devices. Such as Android, iOS, Windows platform. We choose to bring a solution for Android OS so that, in an initial release we can reach out to a large number of people.

One thing that motivated us so much is to bring the policy makers, politicians and normal people in a single platform. People can not bring their daily life problems in a sight of our country politicians because of the lack of communication and time, that is why development in our country infrastructure remain unchanged. If a bridge could be established directly between them then many issues would be solved so easily and in the fastest way which would take our country in one step ahead.

So, this was the thought to develop such a solution which can bring people and policy makers into a single platform so easily.

1.3 Objectives

The main objectives of this android application are given below,

1. An efficient solution to bring MP's and normal people in a single platform.
2. Providing an user friendly smooth experience to users.
3. Make the application faster to get data live by using real time database.
4. People can raise their issues so easily and MP can interact to users post so easily.
5. A direct bridge between normal people and policy makers.
6. Social site using experience.
7. Upvote and Downvote system to bring the important issue in MP's sight so easily.
8. Normal users will be able to register themselves so easily.
9. People would be able to see the reactions of MP and other people as well.

1.4 Expected Outcome

The outcome of this android based solution will bring up a new dimension in our daily life. This is a complete solution to bring those real life issues into light and sight of MP's and other people as well. People would be very pleased to see their desired problem is being discussed through out the app as a social discussion and everyone is being able to participate in the discussion. Since, normal people can not even think of bringing their desired issues in front of the politicians because of communication gap and other problems. Then this app will help them to bring people into a single platform and they will be able to communicate and discuss their problem so easily.

Since the politicians or MP's are so busy with their daily life, they can not give enough time to his people. This app will help them to come through the single platform by only get registered. MP's will be very happy to get interacted with his people so easily and in an efficient way.

This will keep a real time impact on society for the well being of policy makers and normal people.

1.5 Report Layout

We tend to illustrate layout as a result of indication of all our works in short way. So that a whole scenario gets floated in front of the reader so easily and they can go through this report by knowing what he/she will be reading in the very next chapter.

In Chapter 1, we briefly described the main objectives of this project. Along with that, we tried to describe how we got motivated about this android application. We also described the possible and expected outcome or impact this application will keep on the society.

In Chapter 2, we showed the background of this project and if there is any related work done by others or not.

In Chapter 3, the requirements specifications is described briefly. Which contains, Business process model, use case diagram, database schema, project flow model and json data model.

Chapter 4 describes design and development specifications with proper diagram and images.

Chapter 5 includes the testing and implementation done by us and a release among regular users.

Chapter 6 describes conclusion and the future scope of the application.

We also intentionally included the References at the end of this report to keep the track of data fetched over the internet.

CHAPTER 2

Background

2.1 Introduction

Social site has been a part of our daily life. People regularly uses social site for a longer period of time. There are lot of events and issues on a social site like FB, Instagram, Twitter etc. What actually interacts people into that social sites, there are lot of issues, news and etc to discuss upon and people even can communicate each other through the site.

Like that, we have developed an application which will create an impact like social site. Policy makers would be able to communicate with normal people so easily instead of making schedules from their busy life. Normal people will able to share their thoughts, issues and social problems. Interaction of MP's will be shown correctly, so that user might be able to communicate with them so easily and discuss the desired problem or issue.

Matter of the concept debate that its distinctness neglect its impact that it becomes sometimes the main key bearer for activists wishing to create certain issues and that it never helps authority to get what human safety means or give rules makers to take policies.

So, this application is an useful product for the community and policy makers. This will make an impact for the well being of society, normal people and policy makers.

2.2 Comparative Studies

While we were thinking about the idea and wanted to develop an android application. Then we tried to explore some real world applications which matches our idea. And as a matter of fact, we couldn't find any platform like this in our country which can bring politicians and common people in a single platform.

But in other country like Pakistan has a such app which is called **Pakistan Citizen Portal** [1]. It has a feature like creating issue with different types. Normal user can register themselves and create issue.

But the difference between this app and our app is, in complaint box MP's and other policy makers can directly see the peoples reaction by the up-vote and down-vote system. Which will help them to prioritize the issue among so many. People will be able to see the comment made from the MP's and policy makers which will give them some idea to lean on.

Apart from that, there is no such application which is comparatively closer to our android based solution complaint box.

2.3 Scope of the problems

Discussion of issues on social media is a trending thing nowadays. It has been a issue raising site for debating among people and sometimes policy makers notice things on. But this is not a proper way to have such discussion upon social issues. If there would be a proper platform to discuss about those issues then public representative and policy makers can notice things clearly and discuss with common people in a direct manner without delay. There will be no hazard of time management and communication gap. Plug and play type situation will remain there to move such things on with proper documentation.

Recently we can see that, on Facebook, there are lot of conversations and issues being raised and people comment on such things without any thinking of further attempt. We tried to bring things in a shape with proper criteria so that, people can see the reactions of policy makers and other people as well.

According to stat report, 91% people of our country use social websites to discuss such things which is really a huge portion of number [2].

People still like to talk or discuss about social issues whether on social site or in a direct approach. But all we wanted is to bring the whole thing in an integrated documented platform so that, the discussion doesn't go in vain.

2.4 Challenges

There are some challenges we faced while developing and thinking about the solution of the problem. Such as,

1. Data collection of 350 seats of MP's in our country.
2. Creating every MP's account in a default manner.
3. Real time data fetching and on-boarding.
4. User acceptance testing.
5. Finding the possible scopes of this particular problem.
6. Designing a user friendly interface to keep attraction.
7. Bring a mass number of users in a single platform.
8. Direct interaction with all the normal users and MP's.
9. Different types of user's level to identify their access level.

CHAPTER 3 Requirement Specification

3.1 Business Process Modeling

Requirement analysis was the most important part for this particular project, because the solution of this problem was meant for low level users, like mass people. So, we had to think all aspects to bring things in shape. Business process model shows the perfect portrait of the project flow and data flow of the project.

Business Process Model of our application is shown in figure 3.1, which was designed according to the analysis of requirements,

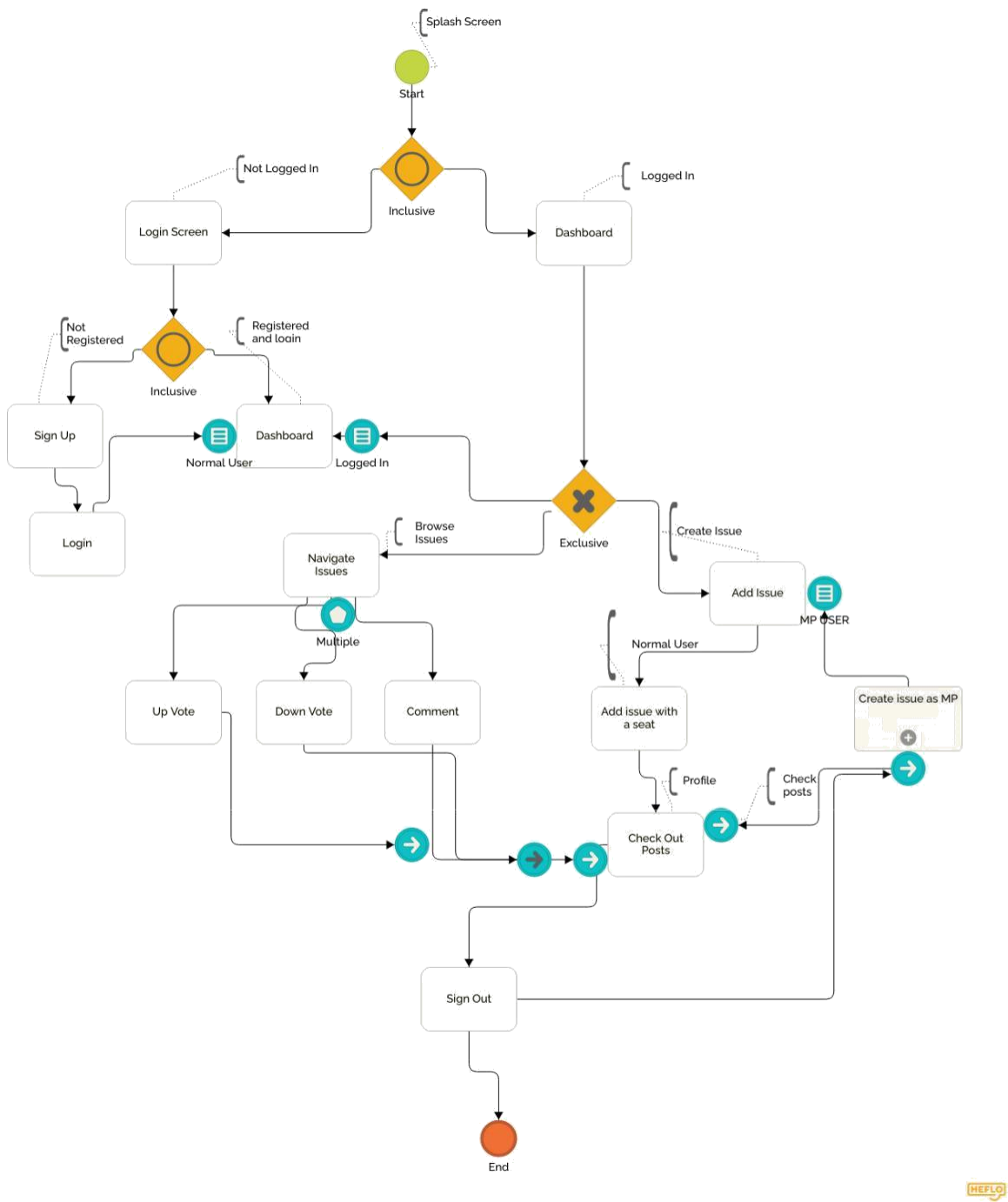


Figure 3.1: Business Process Model

3.2 Use Case Model

Use Case Model defines the flow of every type of users and their accessibility along with project flow. The use case model for this project is shown below as a diagram,

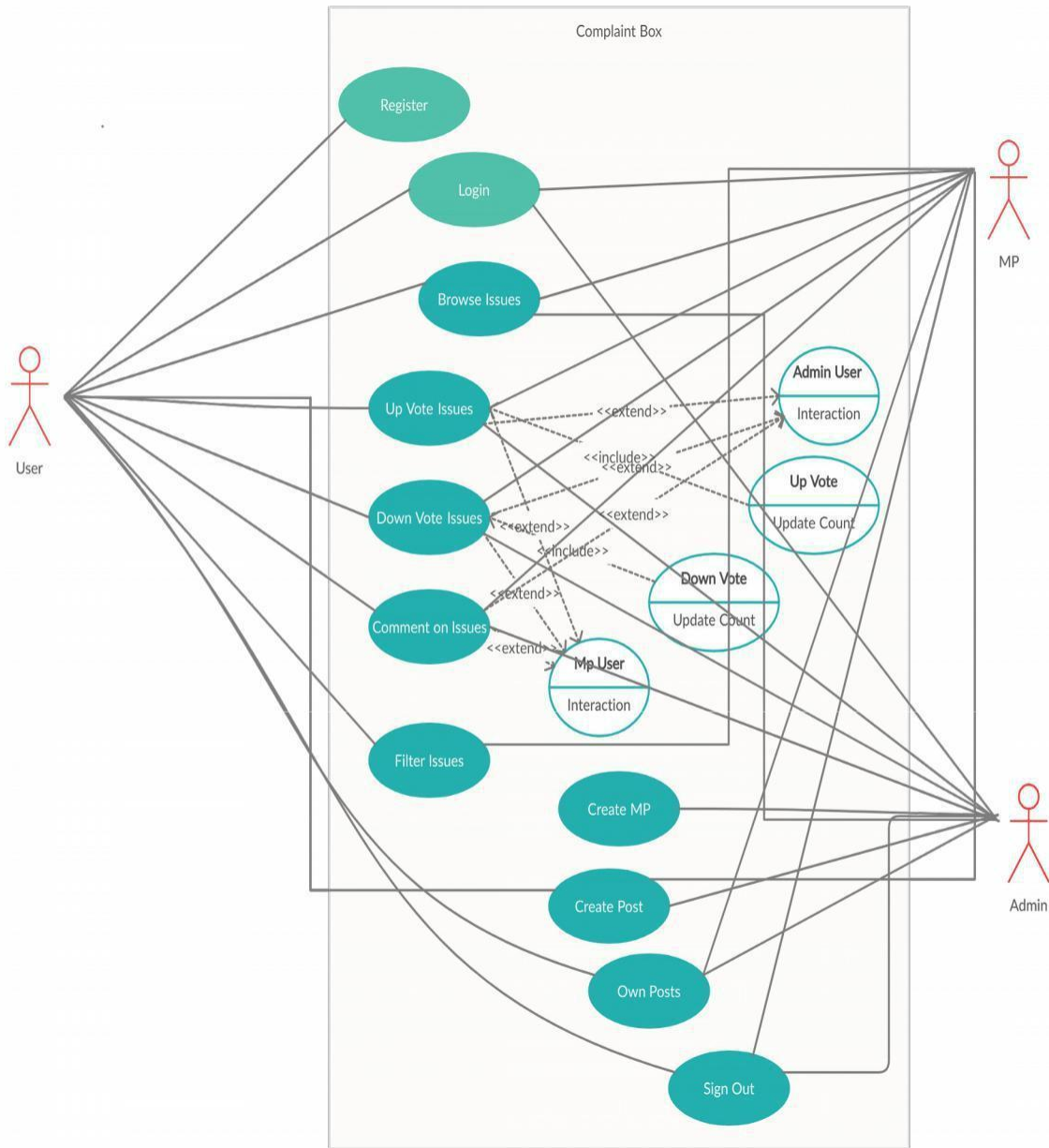


Figure 3.2: Use Case Model.

3.3 JSON Data Model

JSON data modeling is a vital part of using a document database like cloud-hosted. Here we have demonstrated a glimpse JSON data model of our mobile application and those are shown in figure 3.3.

- User table data:

```
[
  {
    "_id": "5f62134778fbc62240ed803c",
    "uuid": "2a71f8e0-a3f4-4b44-a86c-74d2c7adfca4",
    "isActive": true,
    "username": "ashik",
    "email": "complaint.boxwards@gmail.com",
    "seat": "Dhaka-1",
    "password": "letmein2",
    "re_pass": "letmein2",
    "name": "Ashik"
  }
]
```

- Issue table data:

```
[
  {
    "issue_id": "dhaka-2_001",
    "created_by": "2a71f8e0-a3f4-4b44-a86c-74d2c7adfca4",
    "is_admin": false,
    "is_mp": false
  }
]
```

- Issue details data:

```
[
  {
    "issue_id": "dhaka-2_001",
    "created_by": "2a71f8e0-a3f4-4b44-a86c-74d2c7adfca4",
    "upvote_count": "2",
    "downvote_count": "3",
    "comment": "{{comments()}}"
  }
]
```

- MP user's data:

```
{
  "344": {
    "Constituency": "Reserved women's seat-25",
    "ID": "344",
    "MP": "Nargis Rahman"
  },
  "345": {
    "Constituency": "Reserved women's seat-30",
    "ID": "345",
    "MP": "Gloria Jharna Sarker"
  },
  "346": {
    "Constituency": "Reserved women's seat-39",
    "ID": "346",
    "MP": "Parveen Haque Sikder"
  },
  "347": {
    "Constituency": "Reserved women's seat-26",
    "ID": "347",
    "MP": "Monira Sultana"
  }
}
```

Figure 3.3: JSON Data Model.

3.4 Design Requirements

To design the app UI and Logo we used some tools and markup language for giving users and enhanced experience. Such as,

1. XML (Extended Markup Language) language.
2. Adobe XD (Experience Design)
3. Figma (Experience Design)
4. Adobe Photoshop (Logo Design)

CHAPTER 4 Design Specification

4.1 Front-end Design

This chapter of the project report denotes the whole UI design of the app. This application is made and designed by researching and exploring a lot of applications to provide users the best UI experience. A true user friendly and simple gestured app with interactive and eye pleasing pixels of colors. The very first step of the app is a splash screen, where a logo gets shown to the user to create an attraction,

- **Login Screen:** Login screen contains, two fields such as, email and password. A login button along with create new account button and a password recovery page.

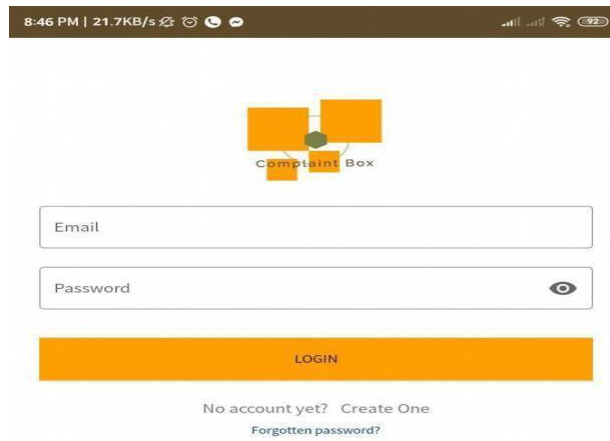


Figure 4.1: Login Page.

- **Sign Up page:** There is a sign up page for normal users to register themselves by using a valid email and information. User will be able to choose a seat from where he/she belongs. After giving an email, a confirmation email gets sent to that given email. Without confirming the email, user won't be able to login into the app.

8:46 PM | 15.6KB/s

Complaint Box

Name

Username

Email

Phone Number

Choose seat

Password

Confirm Password

CREATE ACCOUNT

Already a member? [Login](#)

Figure 4.2: Sign up Page.

- **Password Recovery Page:** There is a password recovery page. By giving the registered email, an email gets sent to that email to recover password.

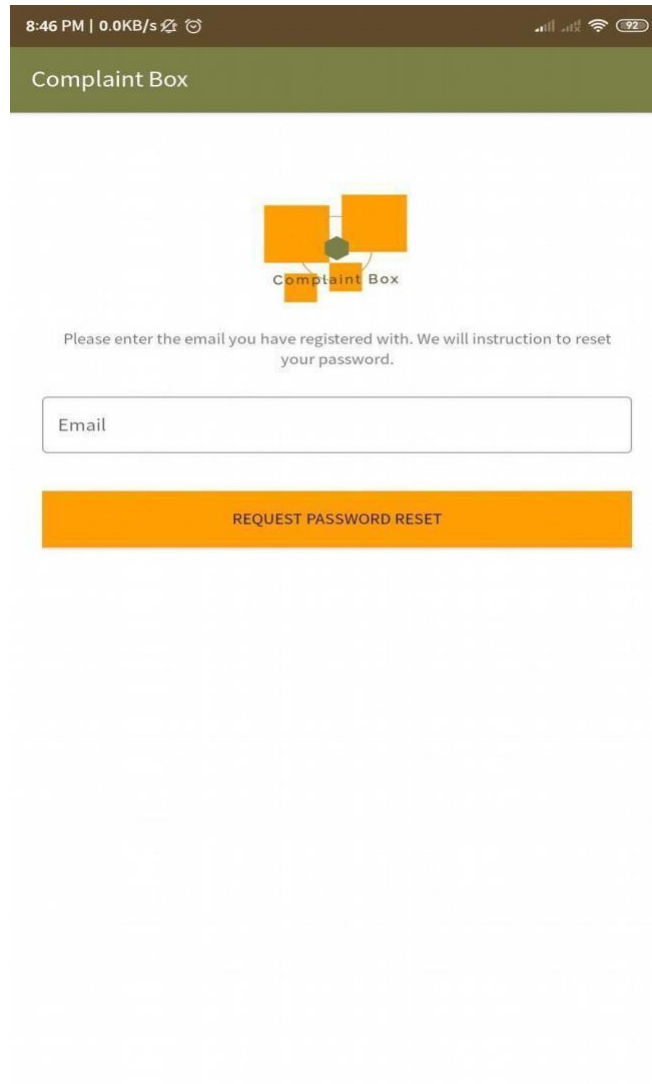


Figure 4.3: Password Recovery Page.

- **User Dashboard:** In dashboard, user will be able to see all the posts created by other users or MP or admin. MP and Admin will hold the badge like “**MP Post**” or “**Admin Post**”. And there is a button to add own issue. Every issue contains, *Upvote*, *downvote* and *comment* button. *Upvote* shows the count of total upvotes and *downvote* shows the total count of downvotes.



Figure 4.4: User Dashboard Page.

- **Filtering:** All the posts shown in Dashboard can be filtered by the seats of MP's. There are total 350 seats and user will be able to select one at a time for filtering the issue lists. When a seat will be selected, issues from that particular seat will be shown.

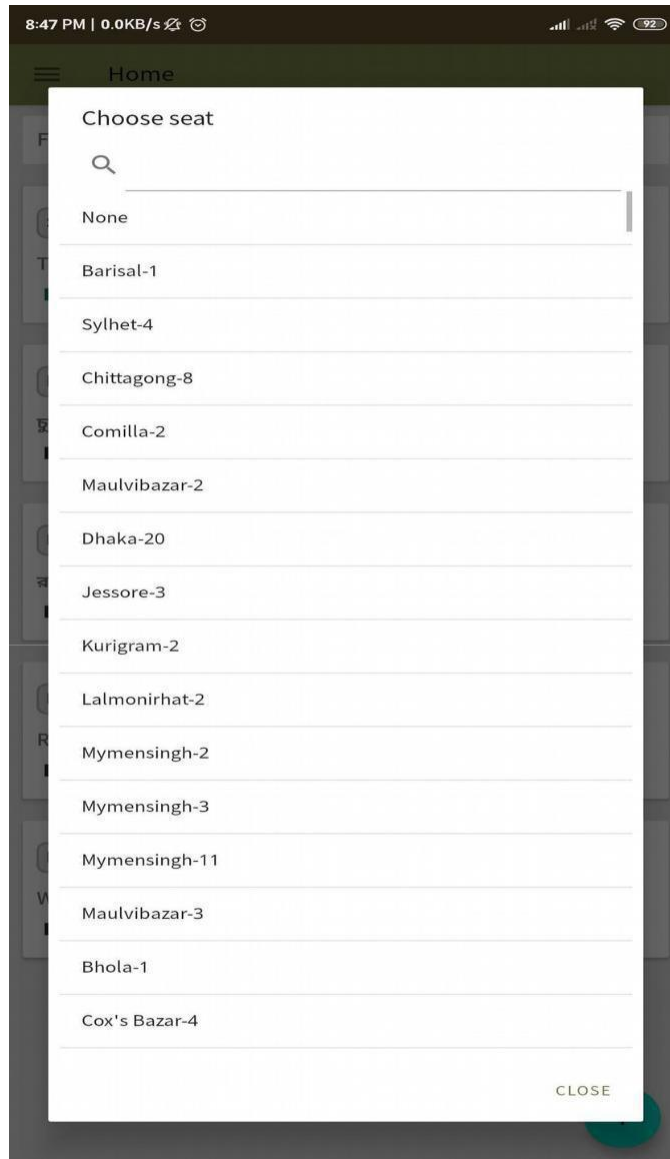


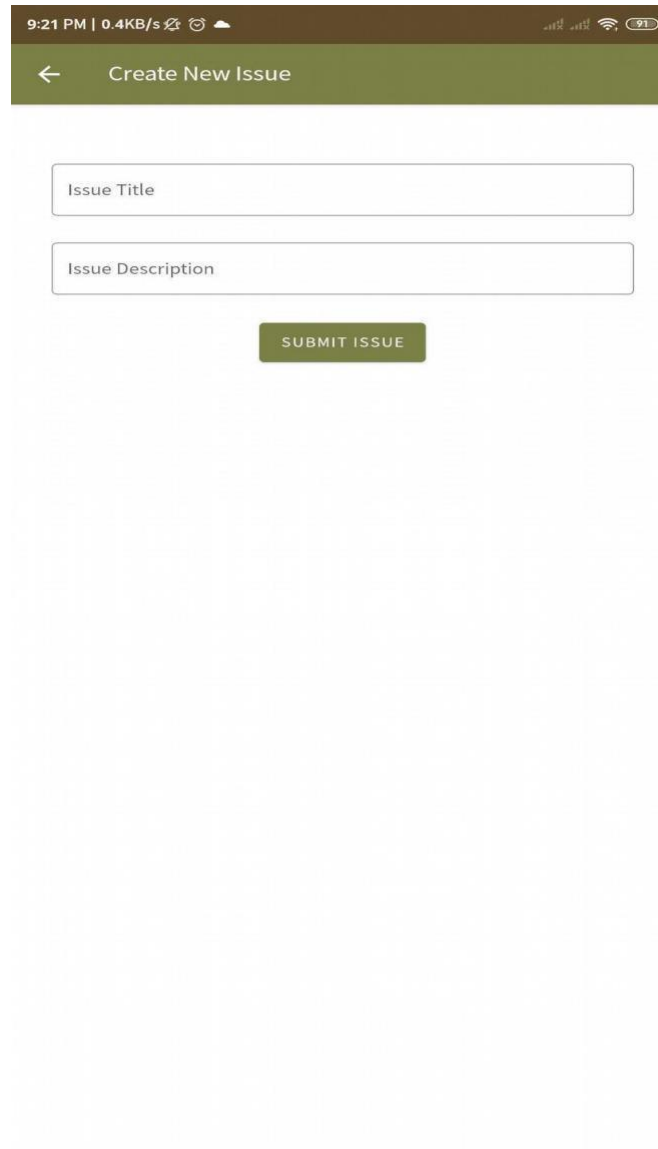
Figure 4.5: Filtering drop-down.

- **Filtered Posts:** After selecting a seat, filtered posts are being shown.



Figure 4.6: Filtered Posts.

- **Create Post:** There is a + button on the dashboard to add issue for all kinds of users. Very easy to submit an issue. After pressing the button, there will be a Issue title and Issue description filed along with submit button.



The screenshot shows a mobile application interface for creating a new issue. At the top, there is a status bar with the time 9:21 PM, a data transfer rate of 0.4KB/s, and a battery level of 91%. Below the status bar is a dark green header with a white back arrow and the text "Create New Issue". The main content area is white and contains two text input fields: "Issue Title" and "Issue Description". Below these fields is a dark green button with the text "SUBMIT ISSUE".

Figure 4.7: Create Post.

- **Own Posts:** Option in the navbar to see own posts.

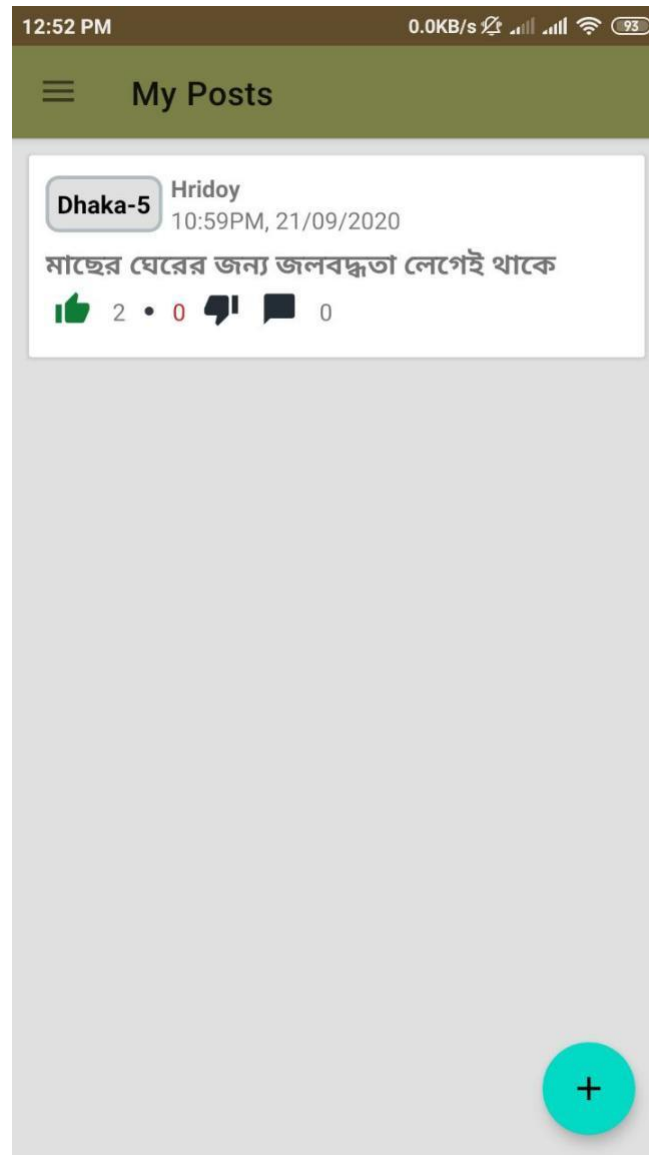


Figure 4.8: Own Posts.

- **App Drawer** : Drawer shows the Name and mail along with sign out option. Sign out button will navigate user into the login page.

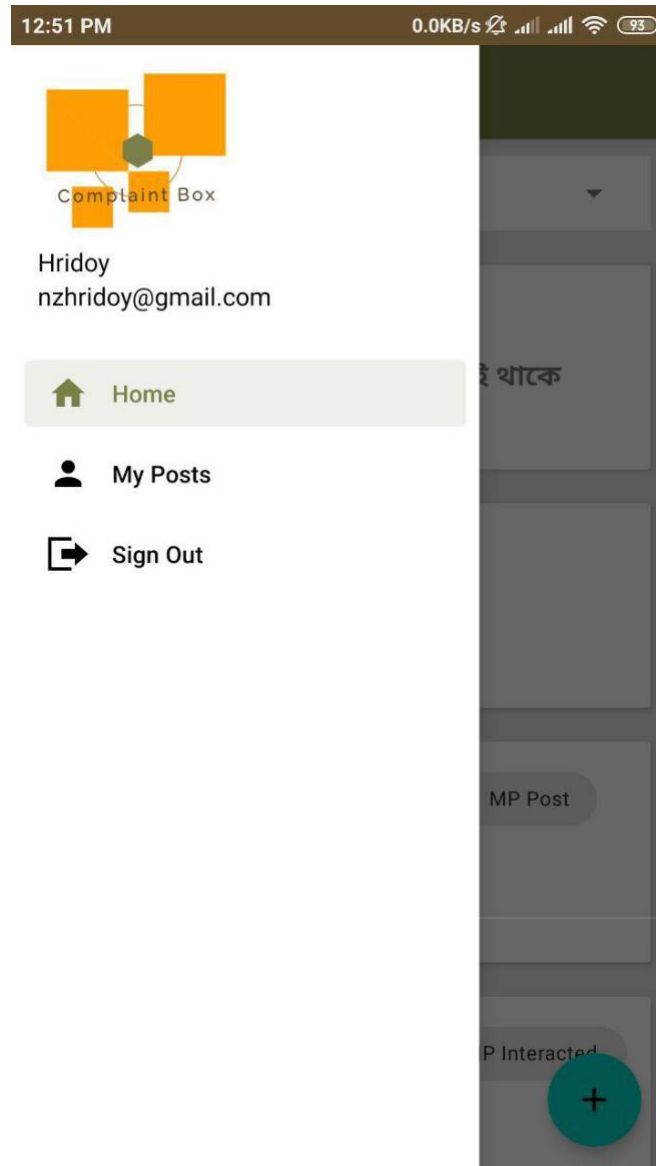


Figure 4.9: App Drawer.

- **Create MP:** While logging in with Admin credential, admin will be able to create MP account. There is “**Create MP**” Button in app drawer to create MP. By clicking the button, there will be a registration page to create MP according to the seat.

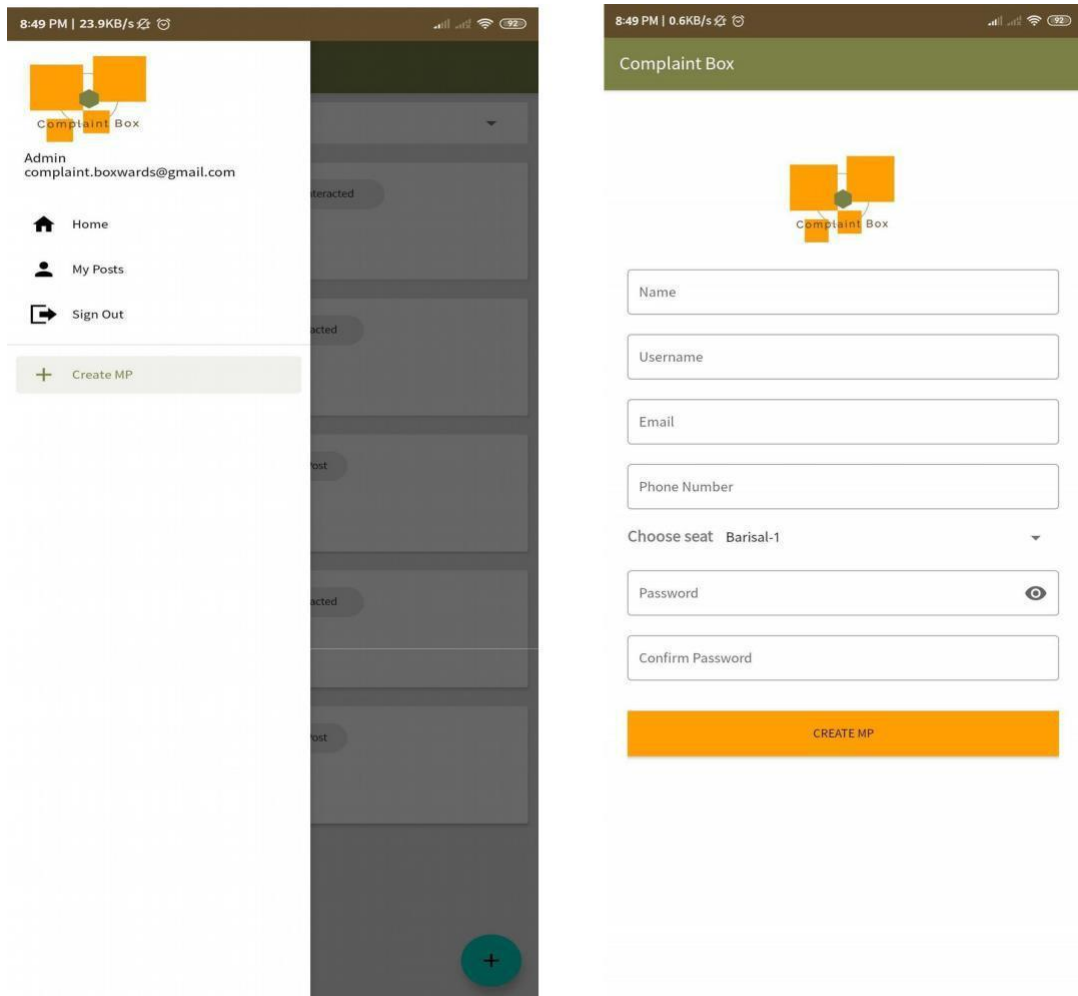


Figure 4.10: Create MP.

4.2 Coding Environment and Languages

Throughout the development phase of the project we have used Android Studio IDE which is developed by IntelliJ to work on android based project. To develop the backend we have used JAVA programming language and the front end was developed by default XML and Material UI. The in details coding ,material and class description will be described in the following chapter.

4.3 Back-end Database

To develop **Complaint Box** we have used a real time cloud based database Firebase by google to fetch data faster and in an efficient way. There is no way to loss the data since the database is cloud based and stores data on the cloud. To have the strong authentication process we have used the firebase authentication system.

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between users in real-time. Cloud Firestore enables you to store, sync and query app data at global scale. Realtime syncing makes it easy for your users to access their data from any device: web or mobile, and it helps your users collaborate with one another. [3]

When your users go offline, the Realtime Database SDKs use local cache on the device to serve and store changes. When the device comes online, the local data is automatically synchronized. [3]

The Realtime Database integrates with Firebase Authentication to provide simple and intuitive authentication for developers. You can use our declarative security model to allow access based on user identity or with pattern matching on your data. [3]

4.4 How The App Works

The work flow of *Complaint Box* is given below.

1. Complaint Box basically a social app to discuss social issues or problems of our country and a communication bridge between normal people and policy makers

- like politicians or elected MP's of different districts or seats. User can create issues according to their seats. MP can view, react and comment on the post and in the comment section their might be a discussion about the issue.
2. There are three types of users in complaint box. Such as, Normal users, MP and Admin. Normal users can create issues, edit them and interact with others post as well. MP users have a special badge to differentiate between normal user and MP user. Admin user have special access to create MP account.
 3. Normal users can register themselves normally by using a valid email. But MP account can not be created normally, Admin has to create the MP account.
 4. There is upvote and downvote system to get the reaction from users all over the country about an issue.
 5. Moreover, User can easily submit complains, reports, compliments, interacts and communicate with MP's using Complaint Box app.

4.5 Implementation Requirements

To implement our application we had to make sure these implementation requirements given below,

Tools:

1. JDK
2. Android Studio.
3. Firebase Real-time cloud based database

Collecting information:

1. Explore internet to find every information which are related to our project.
2. Research on issue discussion on social media.

Programming languages:

1. Java Programming language.
2. XML language.

CHAPTER 5

Implementation and Testing

5.1 Coding Overview

There are lot of classes written to make things run successfully. There are almost 60+ classes to develop this app. We will describe some important things like *AndroidManifest file* and *buildgradle file* in this particular chapter.

5.1.1 Android Manifest File

An Android app project must have an AndroidManifest.xml file (with precisely that name) at the root of the project source set. The manifest file describes essential information about an app app to the Android build tools, the Android operating system, and Google Play [4].

Here is snapshots of **Complaint Box** App *AndroidManifest.xml* file, which are shown in figure 5.1-5.3.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     package="com.howcow.complaintbox">
5
6     <application
7         android:icon="@mipmap/ic_launcher"
8         android:label="@string/app_name"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportsRtl="true"
11        android:theme="@style/AppTheme"
12        tools:ignore="AllowBackup">
13        <activity android:name=".PasswordResetActivity"></activity>
14        <activity
15            android:name=".SignupActivity"
16            android:theme="@style/AppTheme.NoActionBar" />
17        <activity
18            android:name=".LoginActivity"
19            android:theme="@style/AppTheme.NoActionBar" />
20        <activity
21            android:name=".PostDetailActivity"
22            android:theme="@style/AppTheme.NoActionBar" />
23        <activity
```

Figure 5.1: AndroidManifest.xml- part 1


```

<application
    android:name=".services.App"
    android:allowBackup="true"
    android:icon="@drawable/self_protect"
    android:label="@string/app_name"
    android:roundIcon="@drawable/self_protect"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity
        android:name=".MainActivity"
        android:configChanges="keyboardHidden|orientation|screenSize">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity android:name=".SliderScreenActivity"
        android:theme="@style/OnBoardingTheme" />
    <activity android:name=".UserInfoActivity"
        android:windowSoftInputMode="stateHidden|adjustResize"
        />
    <activity android:name=".LoginActivity" />
    <activity android:name=".SplashActivity" />
    <activity android:name="com.tangxiaolv.telegramgallery.GalleryActivity" />

    <service
        android:name=".services.SelfProtectService"
        android:exported="false">
    </service>

```

Figure 5.2: AndroidManifest.xml- part 2

```

    <activity
        android:name=".LoginActivity"
        android:theme="@style/AppTheme.NoActionBar" />
    <activity
        android:name=".PostDetailActivity"
        android:theme="@style/AppTheme.NoActionBar" />
    <activity
        android:name=".NewPostActivity"
        android:parentActivityName=".MainActivity" />
    <activity
        android:name=".MainActivity"
        android:label="@string/app_name"
        android:theme="@style/splashtheme">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=".EditPostActivity"
        android:label="Edit Post Activity" />
    <activity android:name=".CreateMpActivity" />
</application>

</manifest>

```

Figure 5.3: AndroidManifest.xml- part 3

5.1.2 View Models of Complaint Box

The purpose of the ViewModel is to acquire and keep the information that is necessary for an Activity or a Fragment. The Activity or the Fragment should be able to observe changes in the ViewModel. ViewModels usually expose this information via LiveData or Android Data Binding. You can also use any observability construct from your favorite framework. [5]

ViewModel's only responsibility is to manage the data for the UI. It should never access your view hierarchy or hold a reference back to the Activity or the Fragment. [5]

Here a class named *post view model* is shown to get a glimpse of our project's viewmodel

```
package com.howcow.complaintbox.viewmodels;

import android.app.Application;

import androidx.annotation.NonNull;
import androidx.lifecycle.AndroidViewModel;
import androidx.lifecycle.MutableLiveData;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.ValueEventListener;
import com.howcow.complaintbox.cache.FirebaseDatabaseSnapshot;
import com.howcow.complaintbox.cache.SharedPreferenceHandler;
import com.howcow.complaintbox.models.Post;
import com.howcow.complaintbox.models.User;

import java.util.ArrayList;
import java.util.List;

public class PostsViewModel extends AndroidViewModel {
    private final String TAG = "PostsViewModel";
    public MutableLiveData<List<Post>> posts;
    public MutableLiveData<List<Post>> userPosts;
    public MutableLiveData<User> user;

    public PostsViewModel(@NonNull Application application) {
        super(application);
        posts = new MutableLiveData<>(new ArrayList<>());
        userPosts = new MutableLiveData<>(new ArrayList<>());
        user = new MutableLiveData<>(SharedPreferenceHandler.getUserData(application));
    }
}
```

Figure 5.4: Posts View Model

Here is another class named *comment view model* is shown to get a glimpse of our project's another viewmodel

```
package com.howcow.complaintbox.viewmodels;

import android.app.Application;

import androidx.annotation.NonNull;
import androidx.lifecycle.AndroidViewModel;
import androidx.lifecycle.LiveData;
import androidx.lifecycle.MutableLiveData;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.ValueEventListener;
import com.howcow.complaintbox.cache.FirebaseDatabaseSnapshot;
import com.howcow.complaintbox.models.Comment;

import java.util.ArrayList;
import java.util.List;

public class CommentViewModel extends AndroidViewModel {
    private MutableLiveData<List<Comment>> comments;

    public CommentViewModel(@NonNull Application application) {
        super(application);
        comments = new MutableLiveData<>();
        comments.setValue(new ArrayList<>());
    }
}
```

Figure 5.5: Comment View Model

5.1.3 Build gradle file

Gradle is one type of build tool that builds the source code of the program. So it's an important part of Android Studio, and needs to be installed before starting developing your application. We do not have to install it separately, because the Android Studio does it for us, when we make our first project [6].

Gradel files of our projects are shown in figure 5.5.

```

apply plugin: 'com.android.application'
apply plugin: 'com.google.gms.google-services'

android {
    compileSdkVersion 29
    buildToolsVersion "29.0.3"

    defaultConfig {
        applicationId "com.howcow.complaintbox"
        minSdkVersion 21
        targetSdkVersion 29
        versionCode 10
        versionName "1.1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }

    android.applicationVariants.all { variant ->
        variant.outputs.all {
            def appName = "complaint-box"
            outputFileName = appName + "-v${variant.versionName}.apk"
        }
    }
}

```

Figure 5.6: Build.gradle

```

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])

    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    implementation 'com.google.android.material:material:1.2.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.0'
    implementation 'androidx.navigation:navigation-fragment:2.3.0'
    implementation 'androidx.navigation:navigation-ui:2.3.0'
    implementation 'androidx.lifecycle:lifecycle-extensions:2.2.0'

    implementation 'com.jaredrummler:material-spinner:1.3.1'
    implementation 'com.toptcoche.searchablespinner:searchablespinnerlibrary:1.3.1'
    implementation 'androidx.preference:preference:1.1.1'

    testImplementation 'junit:junit:4.13'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
}

dependencies {
    // firebase sdks
    implementation 'com.google.firebase:firebase-analytics:17.5.0'
    implementation 'com.google.firebase:firebase-auth:19.3.2'
    implementation 'com.google.firebase:firebase-database:19.3.1'
    implementation 'com.google.android.gms:play-services-auth:18.1.0'
    implementation 'com.google.code.gson:gson:2.8.6'
}

```

Figure 5.7: Gradle Dependencies

5.2 Testing Result and Reports

An well tested application is a blessing to the users and developers. If an application is developed and released without proper testing then there might be too many bugs and application might not be usable. It irritates users and user will not be interested to use that application in longer period.

So keeping that in mind, we released the *alpha* version of our app to some real users for getting the feedback. They used that version for 3 days and provided some valuable feedback. We solved all those issues they were facing.

Then we released another *beta* version of the app and let some users to use the. Then we finally solved some UI issues and app performance issues.

After two phases of testing we released our final product **Complaint Box v1.1.0**. Which is being presented to the honorable project supervisor and project supervision committee.

CHAPTER 6

Discussion & Conclusion

6.1 Future Scope

The android application Complaint Box have been developed to solve a problem to discuss and raise issues in an organized manner. People our country love to talk about issues but there is not a single dedicated platform for such activities. From that point of view, we have developed the app to make a bridge between normal people and policy makers.

The current version of the app is all about creating issues and discussing about it along with people interaction. In future, we plan to introduce share option of every issues. There will be a link generated if any user wish to share an issue through other platform.

The app will be hosted in a server to have that feature. In addition, there will be comment replying feature to track down the comments properly.

We intend to make this app our countries constructive social app which will be used so many people to communicate every policy makers. Day by day we will introduce new features which will enhance the app and improve user experience.

6.2 Discussion & Conclusion

Our one and only intention was to create a social platform which will be used to discuss social issues of our country. People from all over the country will be able to join the discussion and they will be able to register themselves so easily. Al the elected politicians who are ready serve the people of our country will get the proper way to discuss things on.

We have tried our best to bring on a social platform which can be used to make this country more special. If all the issues can be brought in front of policy makers, then the process will be faster to solve every issues easily. We hope and pray that, our app will be able to gain the glory and handle a structured way to discuss such issues.

REFERENCES

- [1] Play.google.com, 2020. [Online]. Available: <https://play.google.com/store/apps/details?id=com.govpk.citizensportal>. [Accessed: 16- Sep- 2020]
- [2] "Social Media Stats Bangladesh | StatCounter Global Stats", StatCounter Global Stats, 2020. [Online]. Available: <https://gs.statcounter.com/social-media-stats/all/bangladesh>. [Accessed: 16- Sep- 2020]
- [3] "Firebase Realtime Database | Store and sync data in real time", Firebase, 2020. [Online]. Available: https://firebase.google.com/products/realtime-database?gclid=CjwKCAjw74b7BRA_EiwAF8yHFItqMMwkaUpS_eeYGlp4LubDgcJyiaRHSrct1j5h8v7oVHemX6vGxoCLFQQAuD_BwE. [Accessed: 16- Sep- 2020]
- [4] "App Manifest Overview | Android Developers", Android Developers, 2020. [Online]. Available: <https://developer.android.com/guide/topics/manifest/manifest-intro.html>. [Accessed: 16- Sep- 2020]
- [5] "ViewModel | Android Developers", Android Developers, 2020. [Online]. Available: <https://developer.android.com/reference/android/arch/lifecycle/ViewModel>. [Accessed: 16- Sep- 2020]
- [6] "What is Gradle in Android Studio?", Stack Overflow, 2020. [Online]. Available: <https://stackoverflow.com/a/21214991/6681228>. [Accessed: 16- Sep- 2020]

Complain Box Plagiarism Check

ORIGINALITY REPORT

29%

SIMILARITY INDEX

%

INTERNET SOURCES

%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES



[Submitted to Daffodil International University](#)

Student Paper

21%



[developer.android.com](#)

Internet Source

2%



[dspace.library.daffodilvarsity.edu.bd:8080](#)

Internet Source

2%



[hackernoon.com](#)

Internet Source

1%



[ijrar.com](#)

Internet Source

1%



[firebase.google.com](#)

Internet Source

1%



[Submitted to National University of Ireland,](#)

[Maynooth](#)

Student Paper

<1%



[Submitted to Aston University](#)

Student Paper

<1%



[Submitted to Laureate Education Inc.](#)

<1%



dspace.daffodilvarsity.edu.bd:8080

Internet Source

<1%



Submitted to Koc University

Student Paper

<1%



imaginaryfriends1.wixsite.com

Internet Source

<1%



Submitted to University of Queensland

Student Paper

<1%



Submitted to Universitaet Dortmund

Hochschulrechenzentrum

Student Paper

<1%



stackoverflow.com

Internet Source

<1%



eur-lex.europa.eu

Internet Source

<1%

