

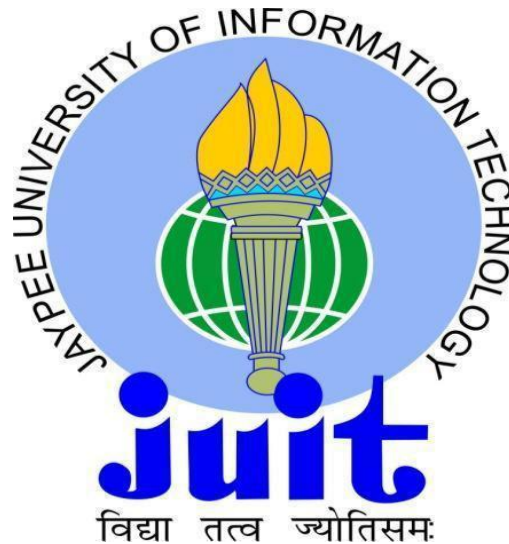
CAMPUS CONNECT APPLICATION

Project report submitted in partial fulfillment of the
requirement for the degree of Bachelor of Technology
in

Computer Science and Engineering/Information Technology

Nikunj Gandhi (141273)
Under the supervision of

MR.AMIT KUMAR



Department of Computer Science & Engineering
and Information Technology
**Jaypee University of Information Technology Waknaghat,
Solan-173234, Himachal Pradesh**

Certificate

Candidate's Declaration

I hereby declare that the work presented in this report entitled "**Campus Connect**" in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2017 to may 2018 under the supervision of MR. Amit Kumar

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Nikunj Gandhi (141273)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)

MR.AMIT KUMAR

Computer Science & Engineering Department

Dated:

ACKNOWLEDGEMENT

We are grateful and indebted to , Department of Computer Science And Engineering for his help and advice in completion of this project report. We also express our deep sense of gratitude and appreciation to our guide for his constant supervision, inspiration and encouragement right from the beginning of this project. We also want to thank our parents and friends for their immense support and confidence upon us. We deem it a pleasant duty to place on record our sincere and heartfelt gratitude to our project guide for his long sightedness, wisdom and co-operation which helped us in tackling crucial aspects of the project in a very logical and practical way.

Nikunj Gandhi
(141273)

Contents

CERTIFICATE	ii
ACKNOWLEDGEMENT	iii
LIST OF FIGURES	iv
LIST OF TABLES	v
ABSTRACT.....	vi
CHAPTER 1 INTRODUCTION	1
1.1 Problem Introduction.....	1
1.1.1 Objective	2
1.2 Previous Work Carried Out In This Field	2
1.3 Researching the Problem Studied.....	2
1.4 Summarization of Results Obtained.....	3
1.5 Organization of the Report	3
CHAPTER 2 LITERATURE REVIEW	4
2.1 Existing System.....	4
2.2 Problem Domain	4
2.3 Technology Used.....	4
2.3.1 Programming Platform.....	4
2.3.2 Android Features.....	5
2.3.3 Firebase	6
2.3.4 Firebase Features	7
2.4 Proposed Advancements	7
CHAPTER 3 PROPOSED SYSTEM	8
3.1 Broad Categories of Problems.....	8
3.2 Proposed System Modules.....	8
3.2.1 Backend Setup.....	9
3.2.2 Reported Problem	9
3.2.3 Modules.....	9
3.2.4 Expected Outcome	9
3.3 Working Of The Project.....	9

3.4 Proposed System Design	10
3.4.1 Use Case	10
3.4.2 Benefits Of Use Case Diagram.....	11
3.4.2 E-R Diagrams	12
3.4.3 Data Flow Diagram	15
CHAPTER 4 CODING	16
4.1 Android Manifest file	16
4.2 Login Xml File	17
4.3 Login Frame Java file	18
4.4 App User Xml File	18
4.5 App User Java file	19
4.6 Problem Administrator Xml file.....	21
4.7 Problem Administrator Java File	21
4.8 Report Problem Xml File	22
4.9 Report Problem Java File.....	23
4.10 View Problem Xml File	24
4.11 View Problem Java File.....	25
4.12 Complaint List Item Structure	27
4.13 Notification Service	28
CHAPTER 5 TESTING	29
5.1 Psychology Of Testing	29
5.2 Testing Objectives	29
5.3 Testing The Application.....	29
5.3.1 Testing Output.....	30
5.4 Testing Table	34
5.4.1 Testing Table Benefits	34
CHAPTER 6 RESULT.....	36
6.1 Output Screens.....	36
6.1.1 Splash Screen	36
6.1.2 Login Screen	37
6.1.3 Google Verification Screen.....	38

6.1.4 Problem List Screen.....	39
6.1.5 Report New Problem Screen.....	40
6.1.6 Reported Problem To Administrator.....	41
6.1.7 All Problem Reported To Administrator.....	42
6.1.8 Reported Problem Details.....	43
CHAPTER 7 CONCLUSION & FUTURE SCOPE.....	44
7.1 Conclusion.....	44
7.2 Key Features Provided.....	44
7.3 Future Scope.....	45
7.4 Future Advancements.....	45
CHAPTER 8 REFERENCES.....	46

LIST OF FIGURES

Figure No.	Figure Name	Page no.
Fig. 2.1	Android Benefits	5
Fig. 2.2	Firebase Services	6
Fig. 3.1	Use case diagram	12
Fig. 3.2	Use case diagram for Nodal officier	13
Fig. 3.3	E-R diagram for Staff Member	13
Fig. 3.4	E-R diagram for Student	13
Fig. 3.5	E-R diagram for Registration Process	14
Fig. 3.6	Activity diagram	15
Fig. 5.1	Internet connectivity error	30
Fig. 5.2	Internet connectivity resolved	30
Fig. 5.3	In-valid input error	31
Fig. 5.4	Wrong input error	31
Fig. 5.5	Valid input entered	32
Fig. 5.6	In-Valid Otp entered	32
Fig. 5.7	Valid Otp entered	33
Fig. 5.8	Button enabled false	33
Fig. 5.9	Button enabled true	34
Fig. 6.1	Splash screen	36
Fig. 6.2	Login screen	37
Fig. 6.3	Google verification	38

Fig. 6.4	Listed Problem	39
Fig. 6.5	Reported Problem	40
Fig. 6.6	Pending Problem	41
Fig. 6.7	All Problem	42
Fig. 6.8	Problem Details	43

LIST OF TABLES

Table No.	Table Name	Page no.
Fig. 5.1	Testing of application	35

ABSTRACT

This project aims at getting the weight off your shoulders. *In today's fast paced world people are constantly looking for ways to help them on the go, with best deals and convenient rates.*

With this app people will be able to book coolies on the go with ease and save time.

The proposed project is an android mobile based application which will be compatible with all the versions of android having version API level above 21 (Lollipop).

CHAPTER 1

INTRODUCTION

A university campus remain abuzz with activities round the clock. While performing their work users including faculty members, staff and students at times notice issues related to campus infrastructure and other problems but many a times fails to bring them to the notice of the concerned as they soon tend to forget while they move along doing their core activities. Due to limited administrative staff this leads to delay in action. It is therefore desired that users be provided with a handy application which let them bring such issues to the notice of concerned authorities on the move. A Campus Connect App would allow reporting such issues and problems to the notice of the concerned for quick redressal of issues.

1.1 Problem Introduction

Earlier there was no such application or e-solution available that provide the combined services of all these facilities. It will have two broad categories of users; Nodal Officers/Problem Administrators responsible for remedial action & users who reports problems including faculty members, university staff and students. Thus for App users, there will be registration for every member and they can report the issue by providing description of it and the problem area. User may re-open the issue in case of non-satisfactory response. So, it will provide a responsive, efficient and user friendly system for problem reporting. On the other hand, Problem Administrators will be assisted with dashboard to view the pending/all complaints reported to them & to change status of complaint with remark. Once status of complaint is changed, an email/SMS is sent to the user who reported the problem. Both the users will be assisted by the System Administrator by adding new problem types & Problem Administrators.

1.1.1 Objective

- The objective is to providing a one common platform to report problems related to any context and issue of notices to the students/staff. The reported problems will then be solved by the concerned authority in due time.
- The emphasis is to reduce the time and effort on all those works which don't help to improve the productivity of the organization.

1.2 Previous Work Carried Out In This Field

As there is no such mobile application developed that provide all such solutions at a one place. Earlier it was difficult as no such app was available so the work like spreading the information about issue of a new notice, registering a complaint regarding any physical damage or problematic infrastructure etc. was a difficult task. The amount of time it took only to register the complaint was quite high and now it only takes this much amount of time to solve the problem.

1.3 Researching the Problem Studied

Campus Connect Application is the mobile based application that will be available in every smart phone and thus will be used easily. We have developed campus connect application to provide services and information in a quick and easy manner to each individual. This application will aggregate all services that will be able to ingrate and create a product which becomes One Stop Solution. All the work is done online. By using this technology we can provide information properly and correctly.

The existing system carries more time and no proper information is provided and no process is carried and for this reason the online system is implemented. User will just have to select a option that to login as a student or a nodal officier with this app. In this project security is also maintain that is only the authorized user can access and use the application.

An online mobile application is an application that will provide services like google mobile number verificaton. The app starts with giving two options to the user to login as a problem administrator or a app user. When loged-in as a problem administrator a list appears that

shows the reported problems. The list divides itself into two parts showing all complaints and pending complaints. If clicked on a problem the status of it can be changed from reported to completed. And a email to the person is sent who reported the problems. And when logged-in as a app user a option will appear to report a new problem. This application uses the firebase real-time database to communicate between the app. The real-time database helps the application to be updated all the time like a web-chat like application. The notification service is also provided to let the user know that the problem is reported or the status of the reported problem has been changed. Also this application saves time by providing all information at one. Stepwise process will be provided so that the new user can access the application and use it with much easiness.

1.4 Summarization of Results Obtained

This project is design for the purpose to reduce the time and effort spent on the unnecessary activities that do not help to increase the efficiency of the work place. This application is used to keep the record of the reported probelms. The interface provided in this app is also very user friendly and offers the services at ease. As it is a mobile application so it makes easy to maintain the record. The android platform is chosen as it is the most common and is available to the large population of India. After the problem has been solved the email and a notification is send to the user who reported the problem.

1.5 Organization of the Report

The rest of the report is organised as follows: Chapter 2 describes the literature review. Chapter 3 describes the proposed System and the modules framed in the project, it includes the working of the whole system and data flow diagram is reperesented and some e-r diagram of the modules are presented. Chapter 4 describes the coding and the algorithm devised. In Chapter 5, Testinhg methodlogy is described. In Chapter 6, result analysis is described with the application images. Finally in Chapter 7, the report is concluded with the future scope of the present study.

CHAPTER 2

LITERATURE REVIEW

2.1 Existing System

There is no such app or the e-solution currently available that provide the combined services of all these facilities. As the students and staff at times face issues related to campus infrastructure and other problems but many a times fails to bring them to the notice of the concerned as they soon tend to forget while they move along doing their core activities. Due to limited administrative staff and time boundations this leads to delay in action. As the whole process of registering complaints was done manually till date the amount of time taken was also very high.

2.2 Problem Domain

It is difficult to maintain and arrange for all the issues and reports on daily basis in the organization. Sometimes in the institutions it is not clearly known that who is responsible for what so finding that person can be difficult and time consuming task to start with.

This app will be an effective and efficient way of disseminating information.

2.3 Technology Used

- Android
- Firebase Database

2.3.1 Programming Platform

Android is choosen as the programming platform as it is most common and is available to the large population of india. Android is a mobile operating system developed 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. In addition, Google has further developed Android TV for televisions, Android Auto for cars, and Wear OS for wrist watches, each with a specialized user interface. Variant of Android are also used on game consoles, digital cameras, PCs and other electronics.

Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android.

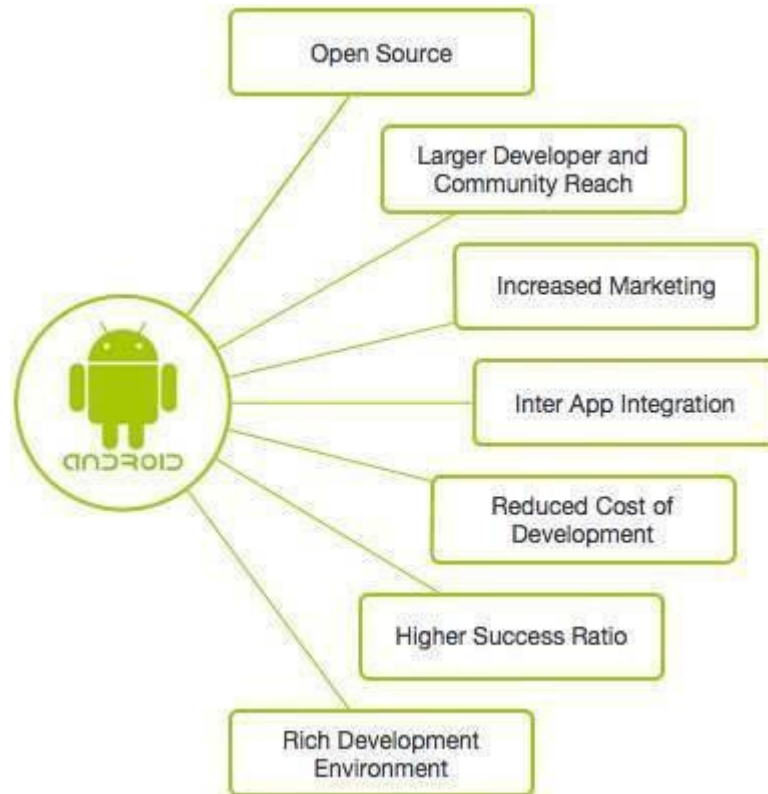


Figure 2.1: Android Benefits

2.3.2 Android Features

- Beautiful UI
- Connectivity
- Storage
- Media Support
- Messaging
- Web Browser
- Multi-Touch

- Multi-Tasking
- Android Beam
- Wi-Fi direct
- GCM – Google Cloud Messaging.

2.3.3 Firebase

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. Firebase frees developers to focus crafting fantastic user experiences. Users don't need to manage servers. Users don't need to write APIs. Firebase is user server, user API and user datastore, all written so generically that user can modify it to suit most needs. Firebase provides free database upto certain limit and with the facility of free web hosting.



Figure 2.2: Firebase Services

2.3.4 Firebase Features

- Realtime Database
- Crashlytics
- Authentication
- Google Analytics
- Performance Monitoring
- Cloud Messaging
- Remote Configs
- Hosting
- App Indexing
- Predictions

2.4 Proposed Advancements

The disadvantages present in the existing system can be overcome using the proposed system

- Only the authorized users can use the facilities.
- Coordination among different levels of administration will increase.
- The huge amount of time can be saved.
- The complaints can be registered within few seconds.
- No need to find who is responsible for what kind of work.
- The notification that complaint is successfully registered and completed are provided.
- This app provides the feature that if the user who registered the complaint is not satisfied with the result, then he can also re-open the issue.

CHAPTER 3

PROPOSED SYSTEM

The project titled as “Campus Connect” is an android application. It will have two broad categories of users; Nodal Officers/Problem Administrators responsible for remedial action & users who reports problems including faculty members, university staff and students. Thus for App users, there will be registration for every member and they can report the issue by providing description of it and the problem area. User may reopen the issue in case of non-satisfactory response. So, it will provide a responsive, efficient and user friendly system for problem reporting. On the other hand, Problem Administrators will be assisted with dashboard to view the pending/all complaints reported to them & to change status of complaint with remark. Once status of complaint is changed, an email/SMS is sent to the user who reported the problem. Both the users will be assisted by the System Administrator by adding new problem types & Problem Administrators.

3.1 Broad Categories of Problems

Although it is envisaged that system allow for reporting of any category of problem as required by the institution however, to start with some of the broad areas of problem reporting are as follows:

- Sanitation
- Security
- Electrical Work
- Civil Work
- Ragging
- Sexual Harassment

3.2 Proposed System Modules

As we all know database projects are so huge, so it is difficult to manage for an individual within a short period, hence our project is divided into different modules like

3.2.1 Backend Setup

This allows the System Administrator to create the following:

- Nodal-Officier/ Problem Administrator: Name, Email Address, Mobile Number etc.
- App User: Email Address, Phone Number.

3.2.2 Reported Problem

This module will store the reported problems. Problem Administrators will also have an interface to update the status of the reported problems.

3.2.3 Modules

System Administrator: To create problem types, Problem Administrators.

App-Users: To register themselves and report issues. Reopen the issue in case of non-satisfactory response.

Problem Administrators: To view the pending/all complaints reported to them, to change status of complaint with remark. Once status of complaint is changed, an email/SMS is sent to the user who reported the problem.

Dashboard: Summarize the complaints received, and their status with mean closure time etc.

3.2.4 Expected Outcome

- A user friendly system for problem reporting
- Responsive, efficient and accountable governance for complaints redressal.
- Analytical reports to support staff requisition. and deployment

3.3 Working Of The Project

Step 1: Firstly user and problem administrator have to download this application. Then user and problem administrator both can directly log-in as their credentials have already been saved in the database.

Step 2: The log-in mechanism works as follows the users only have to submit their already registered mobile number and a OTP will be generated via google service and if verified the user will be logged-in. And same is the case with problem administrator.

Step 3: After the user has logged-in he will have options to report new problems, check previously reported problems. After the report has been reported the user and problem administrator related to that problem area will get a notification.

Step 4: As if logged-in as a problem administrator he will have options like either to change the status of the reported problem from unsolved to solved to check new and previously reported problems.

Step 5: The database of all the problems reported, user, problems administrator will be saved in google firebase database “Real-time Database”. The benefit of using realtime database is that it helps the app to be updated all the time like a chat app.

Step 6: An email notification will be send to the user by the problem administrator after the problem has been solved or its status has been changed.

3.4 Proposed System Design

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

3.4.1 Use Case

The app will provide a common platform to users who reports problems as well as Nodal Officers/Problem Administrators. This app will help faculty members and students to bring problems to the notice of the concerned for quick redressal of issues. Although the app would allow for reporting of any category of problem as required by the institution. The App will enlist the some of the broad areas of problem reporting such as Sanitation, Security, Electrical Work, Civil Work, Ragging, and Sexual Harassment. The qualifications & experience records of expert will be enlisted.

The App will also provide the progress regarding the solution & option to rate the solution. If the solution is not satisfactory user can reopen the issue. Once status of complaint is changed app will trigger a notification to the user. App will also provide the directory of the regarding Nodal Officers in case of offline operation.

Who will use the App?

- Nodal Officers/Problem Administrators
- Users who reports problems including faculty members, university staff and students

What the user wants to do?

- Nodal Officer – View the pending/all complaints reported, Change the status of complaint, Add the Solution progress.
- Student/Staff – Report issues, View solution status, reopen the issue, Rate the work.

3.4.2 Benefits Of Use Case Diagram

- The use cases are mainly composed of narrative text. Hence, unlike many other modeling techniques, the non-technical stake holders are also able understand the model for the software system. This means that feedback can be obtained at a very early stage of the development from the customers and the end users.
- Another major advantage of use case modeling is that it requires the identification of exceptional scenarios for the use cases. This helps in discovering subtle alternate requirements in the system.
- The use case model can be utilized in several other aspect of software development as well, e.g. Cost Estimation, Project Planning, Test Case Preparation and User Documentation.
- The use case diagram provides a comprehensive summary of the whole software system in a single illustration.

Use case diagram for Nodal officer and Department Staff

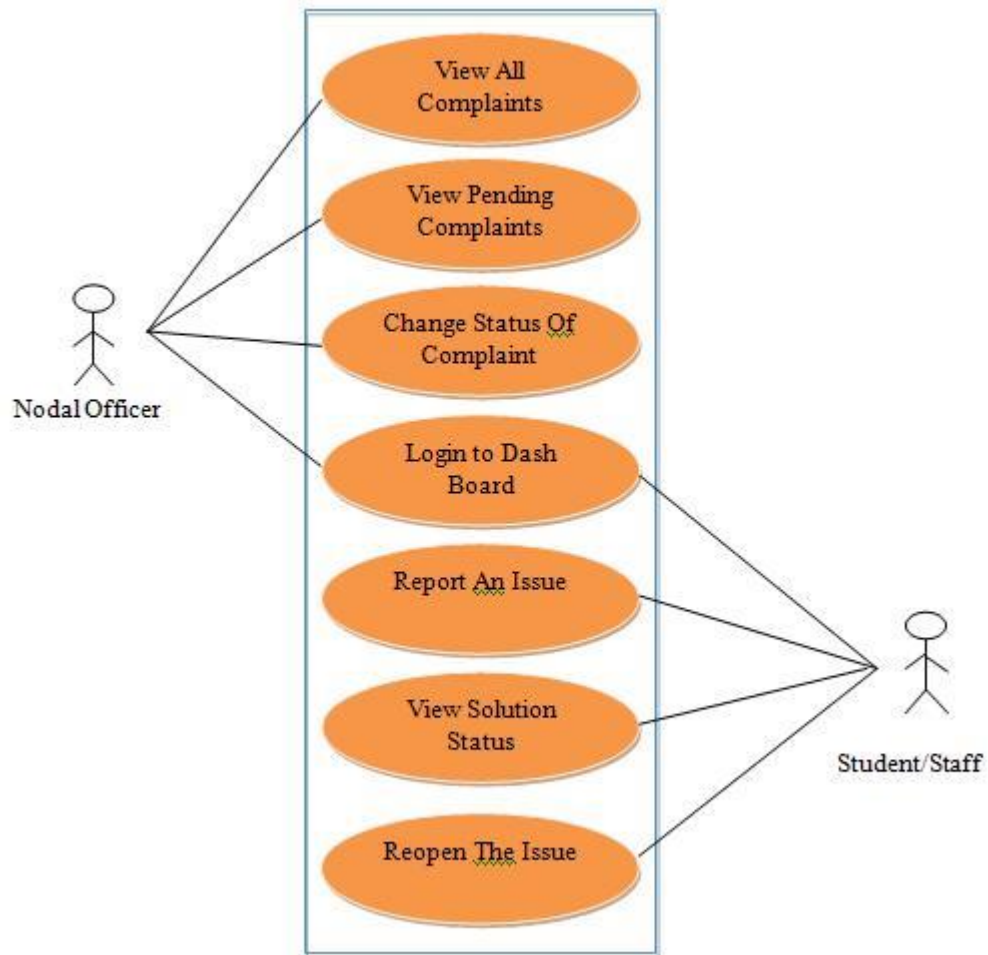


Figure 3.1: Use case diagram

3.4.2 E-R Diagrams

- Nodal Officer: He is head of the entire organization and have control and access to every piece of information. He can change the status of the complaints, grant and revoke permission from students/staff. The nodal officer maintains the integrity of the app and has the duty to provide and maintain consistant and upto date database record of the students and staff to the database maintainer of the application.

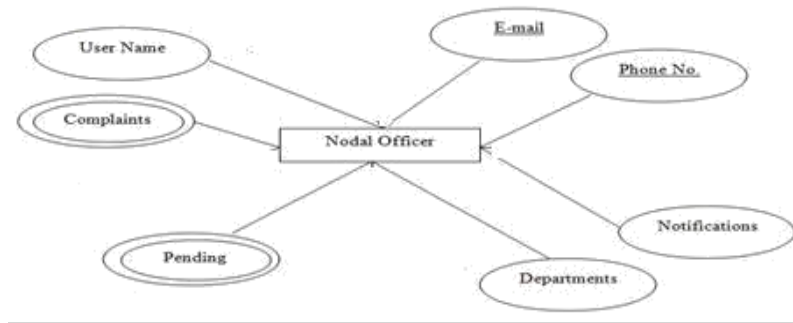


Figure 3.2: E-R diagram for Nodal Officer

- Staff Member: After the nodal level comes staff their main duty is too check complaints and maintain up-to date student record.

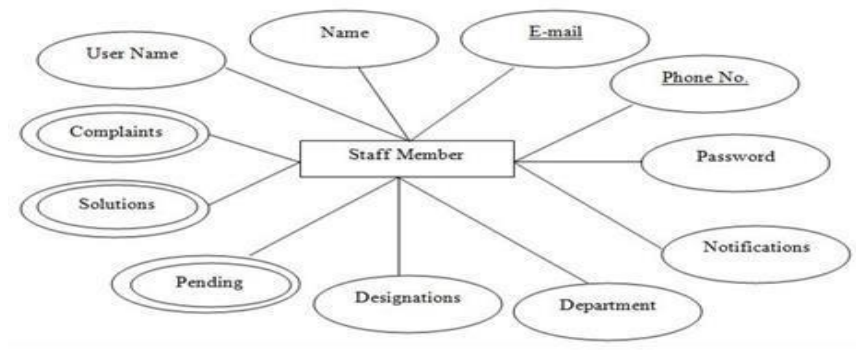


Figure 3.3: E-R diagram for Staff member

- Students: The students are at the base level they can new add complaints, see the status of their complaints, give suggestions etc. And read notices as issued by the nodal officer.

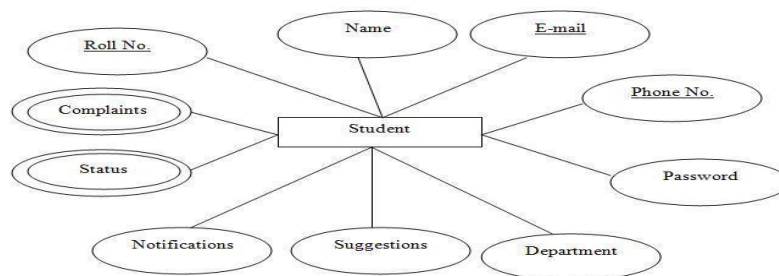


Figure 3.4: E-R diagram for Students

- Registration Process: The Nodal officer is added at the beginning and he further registers the staff members and further staff members registers the students after the authentication process.

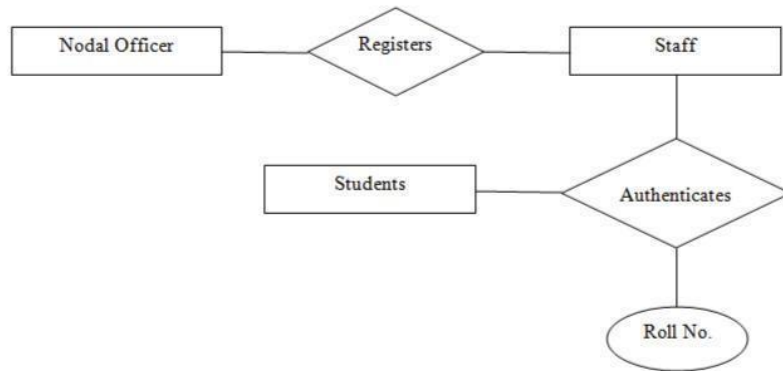


Figure 3.5: E-R diagram for Registration Process

3.4.3 Activity Diagram

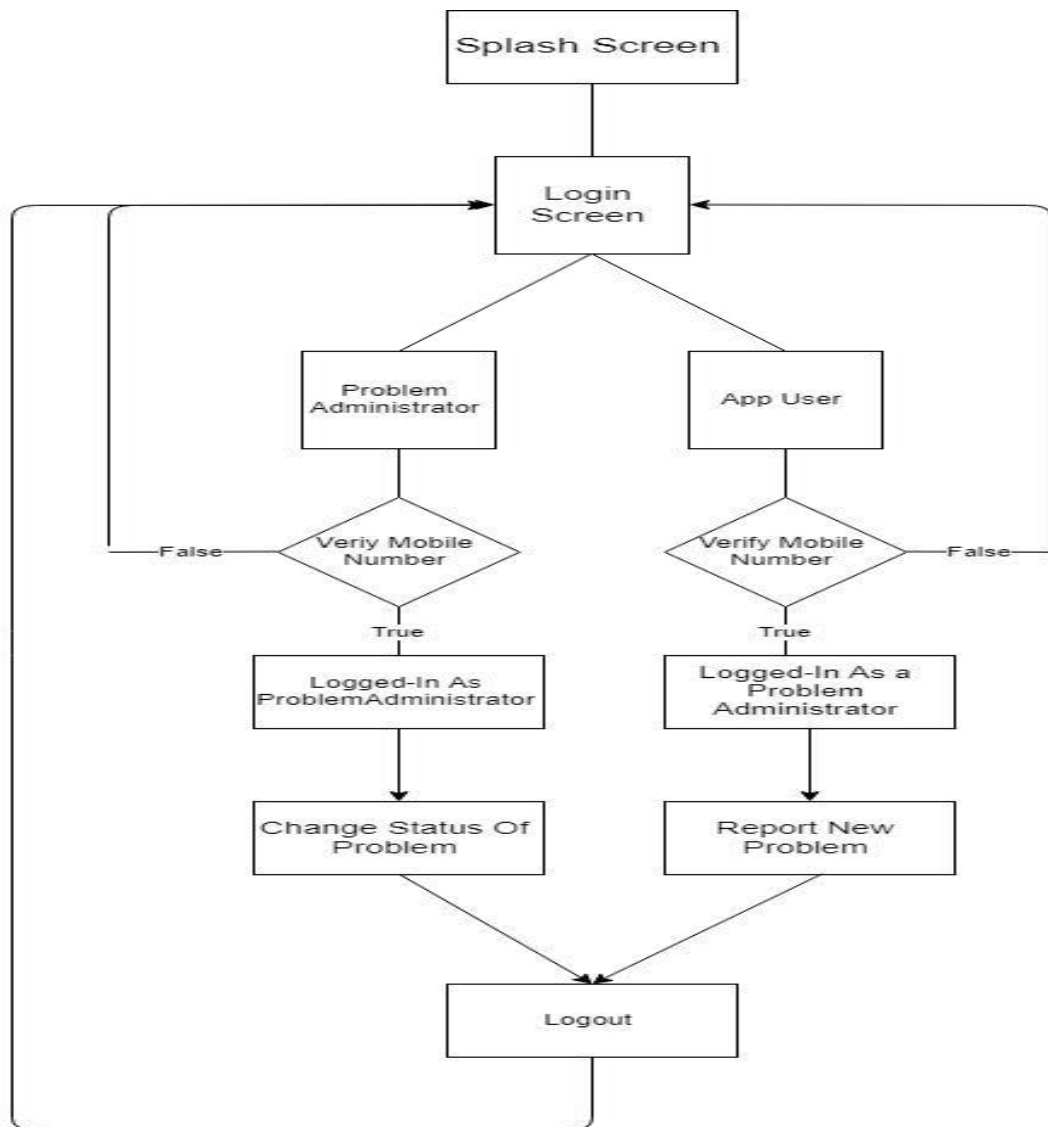


Figure 3.6: Activity Diagram

The activity flow diagram describes the overall run flow of the application. It helps in the better understanding of the application and shows the logical relationship between the different modules of the application.

CHAPTER 4

CODING

4.1 Android Manifest file

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
  xmlns:android="http://schemas.android.com/apk/res/android"
  package="in.blogspot.programmingbypc.campusmitra"> <application
  android:allowBackup="true"
  android:icon="@mipmap/ic_launcher"
  android:label="@string/app_name"
  android:roundIcon="@mipmap/ic_launcher_round
  " android:supportsRtl="true"
  android:theme="@style/AppTheme">
  <activity android:name=".LoginActivity">
    <intent-filter>
      <action android:name="android.intent.action.MAIN" />
      <category android:name="android.intent.category.LAUNCHER"
/>
    </intent-filter>
  </activity>
  <activity android:name=".AppUserHomeActivity" />
  <activity android:name=".ProblemAdministratorHomeActivity"
/> <activity android:name=".ReportProblemActivity" />
  <activity android:name=".ViewProblemActivity" /> <service

  android:name=".AppUserService"
  android:enabled="true"
  android:exported="true" />
  <service
  android:name=".ProblemAdministratorService"
  android:enabled="true"
  android:exported="true" />
</application>
```

4.2 Login Xml File

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FF33b5e5"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context="in.blogspot.programmingbypc.campusmitra.LoginActivity"
    >

    <RadioButton
        android:id="@+id/problem_administrator_radio_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Problem Administrator" />

    <RadioButton
        android:id="@+id/app_user_radio_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        android:text="App User" />
</RadioGroup>

<android.support.design.widget.TextInputLayout
    t android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <EditText
        android:id="@+id/phone_no_edit_text"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Phone No"
        android:inputType="phone" />
</android.support.design.widget.TextInputLayout>
```

4.3 Login Frame Java file

```
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    super.onActivityResult(requestCode, resultCode,
data); if (requestCode == RC_SIGN_IN) {
        if (resultCode == RESULT_OK) {
            // Successfully signed in finish();
            Toast.makeText(getApplicationContext(),
                "Successfully
signed in", Toast.LENGTH_LONG).show();
            SharedPreferences.Editor editor =
                sharedPreferences.edit(); editor.putString("systemUser",
                systemUser); editor.apply();
            if (systemUser.equals("Problem Administrator")) {
                startService(new Intent(getApplicationContext(),
ProblemAdministratorService.class));
                startActivity(new Intent(getApplicationContext(),
ProblemAdministratorHomeActivity.class));
            } else {
                startService(new
Intent(getApplicationContext(), AppUserService.class));
                startActivity(new
Intent(getApplicationContext(), AppUserHomeActivity.class));
            }
        }
    }
}
```

4.4 App User Xml File

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FF33b5e5"
    android:padding="16dp"
    tools:context="in.blogspot.programmingbypc.campusmitra.AppUserHomeActivity"
    >
```

```

<ListView
    android:id="@+id/app_user_list_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:dividerHeight="2dp" />

<android.support.design.widget.FloatingActionButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_marginBottom="12dp"
    android:layout_marginEnd="12dp"
    android:layout_marginRight="12dp"
    android:clickable="true"
    android:focusable="true"
    android:onClick="reportProblem"
    app:srcCompat="@android:drawable/ic_menu_edit" />
</RelativeLayout>

```

4.5 App User Java file

```

public class AppUserHomeActivity extends AppCompatActivity
{
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_options,
            menu); return true;
    }

    databaseReference.addValueEventListener(new ValueEventListener()
    {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot)
        {
            ArrayList<> arrayList = new ArrayList<>();
            for (DataSnapshot postSnapshot :
            dataSnapshot.getChildren()) {
                HashMap<String, String> hashMap =
                (HashMap<String, String>) postSnapshot.getValue();
            }
        }
    });
}

```

```

        "ProblemStatus", "TimeOfReporting"};

        int[] to = {R.id.problem_area_text_view,
R.id.problem_title_text_view, R.id.problem_status_text_view,
R.id.reporting_time_text_view};

        SimpleAdapter simpleAdapter = new
SimpleAdapter(getApplicationContext(), arrayList,
R.layout.complaints_list_item, from, to);

        listView.setAdapter(simpleAdapter);
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }

    });

    listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view,
int position, long id) {
            Intent intent = new
Intent(getApplicationContext(), ViewProblemActivity.class);
            intent.putExtra("problemKey",
arrayList.get(position).get("problemKey"));
            intent.putExtra("systemUser", "App
User"); startActivity(intent);
        }
    });
}

    public void reportProblem(View view) {
        startActivity(new Intent(getApplicationContext(),
ReportProblemActivity.class));
    }
}

```

4.6 Problem Administrator Xml file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FF33b5e5"
    android:orientation="vertical"
    android:padding="16dp"

    tools:context="in.blogspot.programmingbypc.campusmitra.ProblemAdministrato
r HomeActivity">

    <Spinner
        android:id="@+id/select_complaint_spinner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <ListView
        android:id="@+id/problem_administrator_list_view"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:dividerHeight="2dp" />
</LinearLayout>
```

4.7 Problem Administrator Java File

```
public class ProblemAdministratorHomeActivity extends AppCompatActivity
{
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_options,
            menu); return true;
    }
    () .substring(3);
    final Query databaseReference =
    firebaseDatabase.getReference("Reported
```

```

Problems").orderByChild("ReportedTo").equalTo(problemAdministrator)
        ; int position, long id) {
            if (position == 1) {
                databaseReference.addValueEventListener(new
ValueEventListener() {
                    @Override
                    public void onDataChange(DataSnapshot dataSnapshot)
{
                        arrayList = new ArrayList<>();
                        for (DataSnapshot postSnapshot :
dataSnapshot.getChildren()) {
                            HashMap<String, String> hashMap =
(HashMap<String, String>) postSnapshot.getValue();
                            if (hashMap != null &&
AdapterView.OnItemClickListener() {
                                @Override
                                public void onItemClick(AdapterView<?> parent, View view,
int position, long id) {
                                    Intent intent = new
Intent(getApplicationContext(), ViewProblemActivity.class);
                                    intent.putExtra("problemKey",
arrayList.get(position).get("problemKey"));
                                    intent.putExtra("systemUser", "Problem
Administrator"); startActivity(intent);
                                }
                            });
                        }
                    }
}
}

```

4.8 Report Problem Xml File

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FF33b5e5"

```



```

        android:orientation="vertical"
        android:padding="16dp"
tools:context="in.blogspot.programmingbypc.campusmitra.ReportProblemActivi
t
y">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
<android.support.design.widget.TextInputLayout
    t android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <EditText
        android:id="@+id/problem_description_edit_text"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Description"
        android:inputType="textLongMessage"
        android:lines="5" />
</android.support.design.widget.TextInputLayout>

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="reportProblem"
        android:text="Report Problem" />
</LinearLayout>

```

4.9 Report Problem Java File

```

public class ReportProblemActivity extends AppCompatActivity
{
    HashMap<String, String> hashMap;
    String reportedTo;

```

```

FirebaseDatabase firebaseDatabase;
DatabaseReference databaseReference;
Spinner problemAreaSpinner;
EditText problemTitleEditText, problemDescriptionEditText;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_report_problem);
    problemAreaSpinner = findViewById(R.id.problem_area_spinner);
    problemTitleEditText = findViewById(R.id.problem_title_edit_text);
    problemDescriptionEditText = hashMap.put("ProblemArea",
databaseReference = firebaseDatabase.getReference("Reported
Problems").push());

        databaseReference.setValue(hashMap);
        Toast.makeText(getApplicationContext(), "Your complaint has
been reported successfully!",
            Toast.LENGTH_LONG).show(); finish();
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }

});
}
}

```

4.10 View Problem Xml File

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FF33b5e5"
    android:orientation="vertical"

```

```
tools:context="in.blogspot.programmingbypc.campusmitra.ViewProblemActivity" >
```

```
<TextView  
    android:id="@+id/problem_area_text_view"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:textAppearance="@style/TextAppearance.AppCompat.Subhead" />
```

```
<TextView
```

```
<TextView  
    android:id="@+id/problem_status_text_view"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"
```

```
<TextView  
    android:id="@+id/remark_text_view"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:textAppearance="@style/TextAppearance.AppCompat.Body1"  
    android:visibility="gone" />
```

4.11 View Problem Java File

```
public class ViewProblemActivity extends AppCompatActivity  
    { HashMap<String, String> hashMap;  
    String reportedBy, reportedByEmailAddress;  
    TextView remarkTextView;  
    Button reopenIssueButton;  
    FirebaseDatabase firebaseDatabase;  
    DatabaseReference databaseReference;  
    EditText remarkEditText;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_view_problem);  
        firebaseDatabase = MyFirebaseDatabase.getFirebaseDatabase();
```

```

findViewById(R.id.problem_area_text_view); final
    TextView reportingTimeTextView =
findViewById(R.id.reporting_time_text_view);
    final TextView problemTitleTextView =
findViewById(R.id.problem_title_text_view);
    final TextView problemDescriptionTextView =
findViewById(R.id.problem_description_text_view);
    final TextView problemStatusTextView =
reopenIssueButton findViewById(R.id.reopen_button);
    final String systemUser = getIntent().getStringExtra("systemUser");
    String problemKey = getIntent().getStringExtra("problemKey");
    databaseReference = firebaseDatabase.getReference("Reported
Problems").child(problemKey);
    databaseReference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot)
            { hashMap = (HashMap<String, String>)
dataSnapshot.getValue();
            if (hashMap != null) {
                reportedBy = hashMap.get("ReportedBy");
            }
            if (hashMap != null && hashMap.containsKey("ReopenedOn"))
                { reopenedOnTextView.setVisibility(View.VISIBLE);
reopenedOnTextView.setText("Re-opened on : " +
hashMap.get("Remark"));
                if (systemUser.equals("App User")) {
                    reopenIssueButton.setVisibility(View.VISIBLE)
                    ;
                }
            } else {
                solvedOnTextView.setVisibility(View.GONE)
                ;
                if (systemUser.equals("Problem Administrator")) {
                    changeStatusSwitch.setVisibility(View.VISIBLE)
                    ; remarkEditText.setVisibility(View.VISIBLE);
                    saveButton.setVisibility(View.VISIBLE);
                    remarkEditText.setEnabled(false);
                    saveButton.setEnabled(false);

```

```

public void changeStatusWithRemark(View view) {
    databaseReference.child("ProblemStatus").setValue("solved")
        ; databaseReference.child("SolvedOn").setValue(new
Date().toString());
    final String remark = remarkEditText.getText().toString();
    databaseReference.child("Remark").setValue(remark);
    android.support.v7.app.AlertDialog.Builder builder = new
android.support.v7.app.AlertDialog.Builder(ViewProblemActivity.this);
    builder.setTitle("Send Email?");
    builder.setMessage("Do you want to send EMail (notifying the status
of the problem) to the user who reported the problem?");
    builder.setCancelable(false);
    builder.setPositiveButton("Yes", new builder.show()
;
}

public void reopenIssue(View view) {
    databaseReference.child("ProblemStatus").setValue("reopened")
        ; databaseReference.child("ReopenedOn").setValue(new
Date().toString());
    reopenIssueButton.setVisibility(View.GONE);
    remarkTextView.setVisibility(View.GONE);
}
}

```

4.12 Complaint List Item Structure

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorAccent"
    android:orientation="vertical"
    android:padding="4dp">

    <TextView
        android:id="@+id/problem_title_text_view"
        android:layout_width="match_parent"

```

```

<TextView
    android:id="@+id/reporting_time_text_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"

    android:textAppearance="@style/TextAppearance.Compat.Notification.Time"
/> </LinearLayout>

```

4.13 Notification Service

```

public class ProblemAdministratorService extends Service
    { public ProblemAdministratorService() {
    }

    @Override
    public void onCreate() {
        super.onCreate();

        final FirebaseDatabase firebaseDatabase
= MyFirebaseDatabase.getFirebaseDatabase();

        final String problemAdministrator =
            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);

            intent.putExtra("problemKey", dataSnapshot.getKey());
            intent.putExtra("systemUser", "Problem
Administrator"); PendingIntent pendingIntent =
PendingIntent.getActivity(getApplicationContext(), 0,
intent, PendingIntent.FLAG_UPDATE_CURRENT);

            builder.setTitle("New Problem
Reported!").setContentText(hashMap.get("ProblemTitle")).setContentIntent(
@Override
public void onChildChanged(DataSnapshot dataSnapshot, String s) {
        HashMap<String, String> hashMap = (HashMap<String, String>)
dataSnapshot.getValue();

        if (hashMap.get("ProblemStatus").equals("reopened")) {
            Intent intent = new Intent(getApplicationContext(),
ViewProblemActivity.class);

```

CHAPTER 5

TESTING

5.1 Psychology Of Testing

Software testing is a process of executing a program or application with the intent of finding the software bugs. The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn't work.

5.2 Testing Objectives

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

- Testing is a process of executing a program with the intent of finding an error.
- A successful test is one that uncovers an as yet undiscovered error.
- A good test case is one that has a high probability of finding error, if it exists.
- The tests are inadequate to detect possibly present errors.
- The software more or less confirms to the quality and reliable standards.

5.3 Testing The Application

The application is tested of the errors that are produced due to wrong input value or the error in coding, the basic purpose of all these type of testing is that to minimize the errors that can be generated in the app. To test the app different type of techniques are used like black box testing etc. The errors produced are recorded and rectified.

5.3.1 Testing Output

- Negative Startup Testing

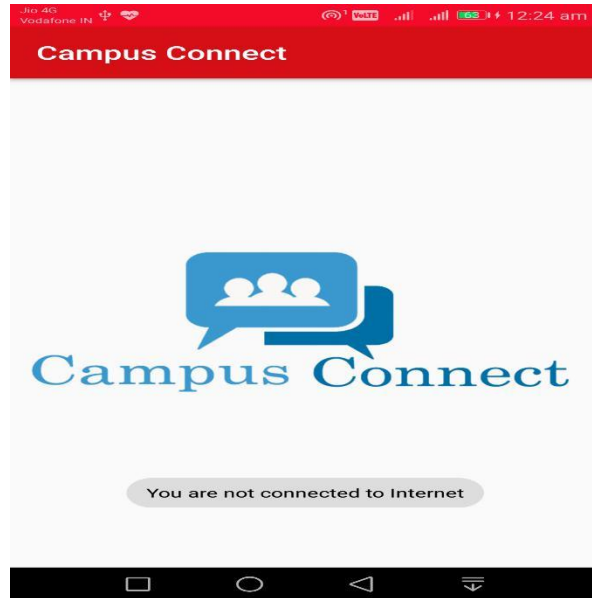


Figure 5.1: Internet Connectivity Error

- Positive Startup Testing



Figure 5.2: Internet Connectivity Resolved

- Negative Entry Testing

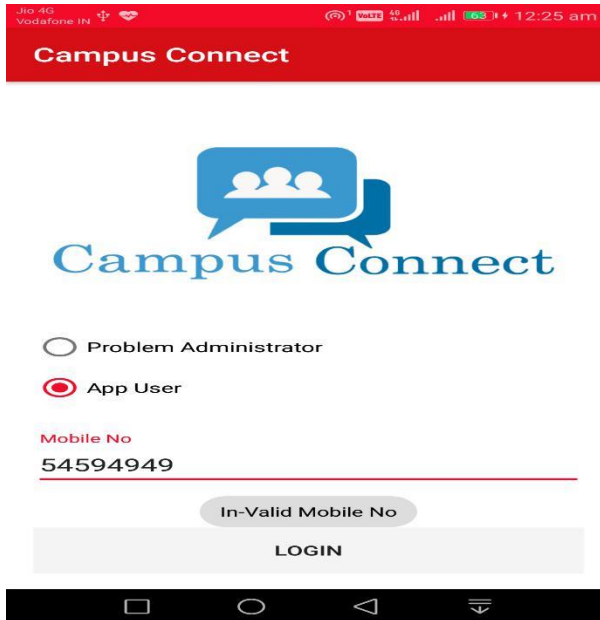


Figure 5.3: Invalid Input Error

- Wrong Input Testing



Figure 5.4: Wrong Input Error

- Positive Input Testing

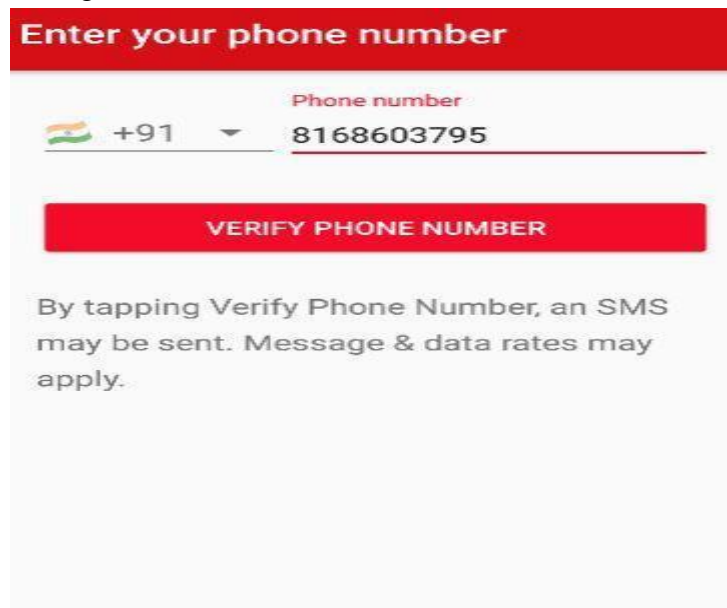


Figure 5.5: Valid Input Entered

- Negative Otp Testing

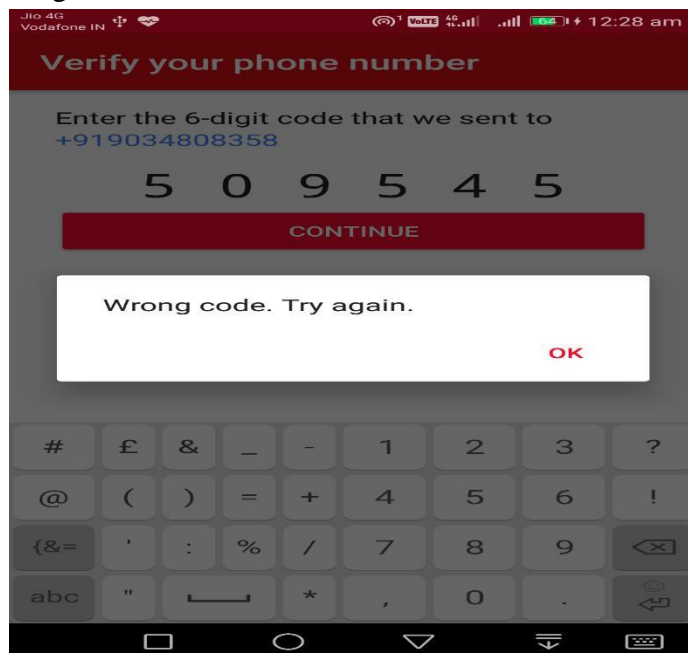


Figure 5.6: Invalid Otp Entered

- Positive Otp Testing

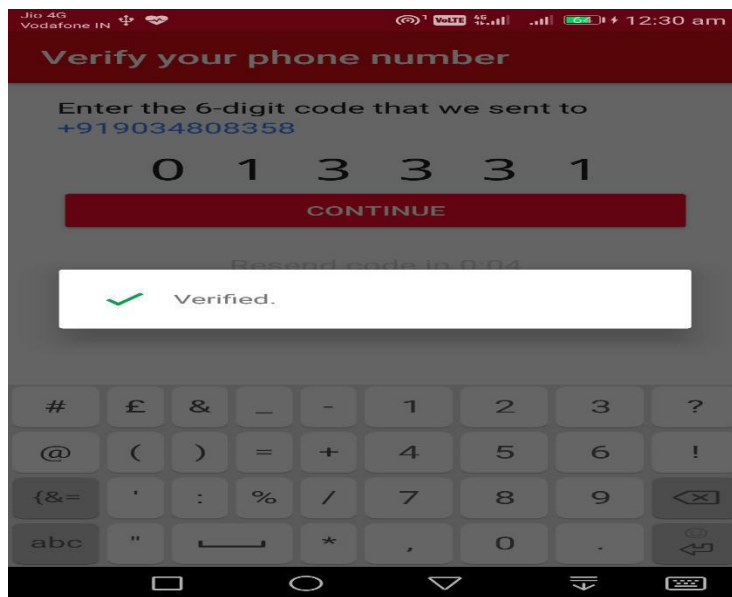


Figure 5.7: Valid Otp Entered

- Save Button Status Enabled False

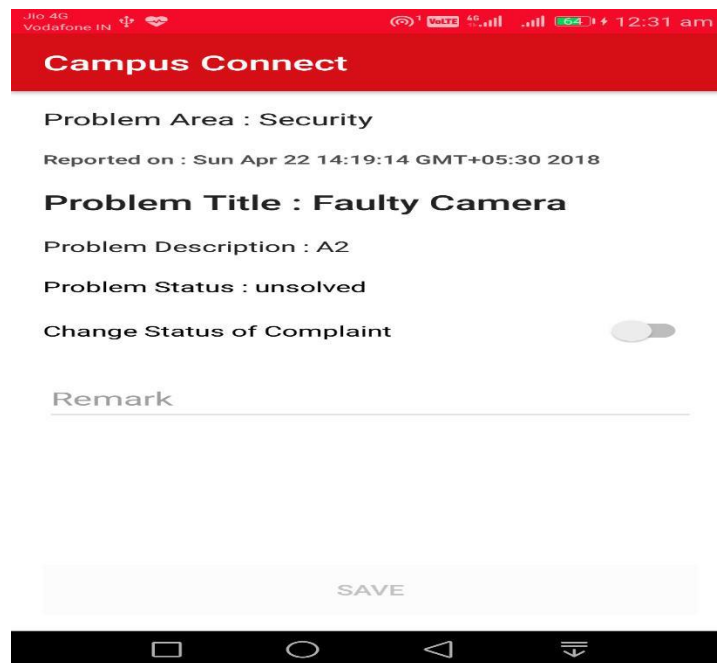


Figure 5.8: Button Enabled False

- Save Button Status Enabled True

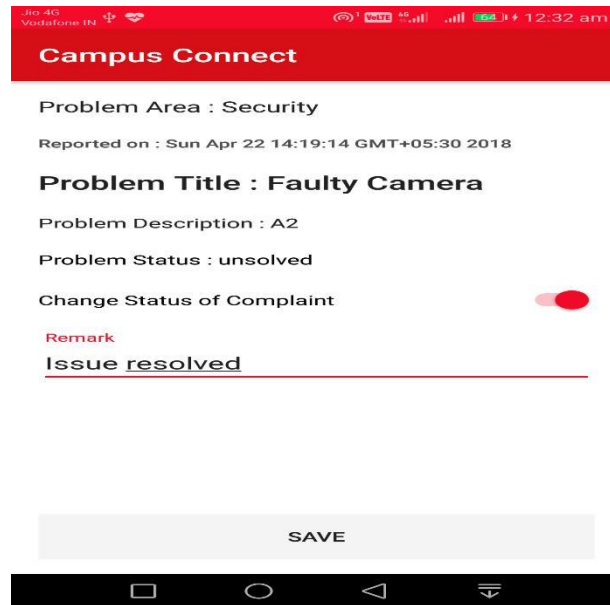


Figure 5.9: Button Enabled True

- The save button status changes after the all the data enteries are filled upon.

5.4 Testing Table

The testing table is prepared so that all the testing records can be maintained. The table helps to analyze the code in a better way and better predictions can be made. The effects of stress testing on the app can be recorded in a better way.

5.4.1 Testing Table Benefits

- Better understanding of the code.
- Helps to achieve the aim of error free app.
- Better analyzation on the code can be done.
- The limit of the working of app can be understood in a better way.
- A record of all the test cases and their respective outputs can be obtained and saved at a single place only.

S. No	Module	Scenario	Test Case 1 (positive)	Test Case 2 (negative)
1	Startup	Start Application	Application Started	An error occurred while starting the application if internet is not there.
2	Internet Connectivity	Start Application	Application run smoothly if internet was connected	Application hanged up on showing toast for no internet connection
3	Login	Phone Number	If A user was already entered into database then the user was sent to other activity	If a user was not entered into database the application shows a toast with text “unauthorized user”
4	Login	One Time Password	If OTP is received by sms then app auto verify the user and moves the user to next activity	If OTP is not received by user phone number then another activity is opened up asking user's OTP
5	Create Complaint	Complaint Is generated	If title contains some words and problem is reported it successfully created	If title is kept blank and a problem is submitted then a toast is shown with text “Title Cannot Be Blank”
6	Create Complaint	Complaint Is generated	If description contains some words and problem is reported it successfully created	If title is kept blank and a problem is submitted then a toast is shown with text “Title Cannot Be Blank”
7	Solve Problem	Problem switch	when switch is on the edit text becomes editable	When switch is off the edit text becomes editable

Table 5.1 Testing Of Application

CHAPTER 6

RESULT

6.1 Output Screens

6.1.1 Splash Screen



Figure 6.1: Splash Screen

- This is the first screen that opens when we open the app. At this screen the app checks in background if the data connectivity is there and for other permissions. If the data connectivity is absent then the app stops right here until data connectivity is regained.

6.1.2 Login Screen



The screenshot shows the login interface for 'Campus Connect'. At the top, there is a red header bar with the text 'Campus Connect' in white. Below this is the 'Campus Connect' logo, which consists of a blue speech bubble containing three white human figures, with the text 'Campus Connect' in blue below it. Under the logo, there are two radio button options: 'Problem Administrator' (unselected) and 'App User' (selected). Below these options is a text input field labeled 'Mobile No' in red. A horizontal red line is positioned below the input field. At the bottom of the screen is a grey button with the text 'LOGIN' in black.

Figure 6.2: Login Screen

- This is the application login screen from her both problem administrator and app user can login by just entering their registered mobile number and selecting the choice in which they want to login.

6.1.3 Google Verification Screen

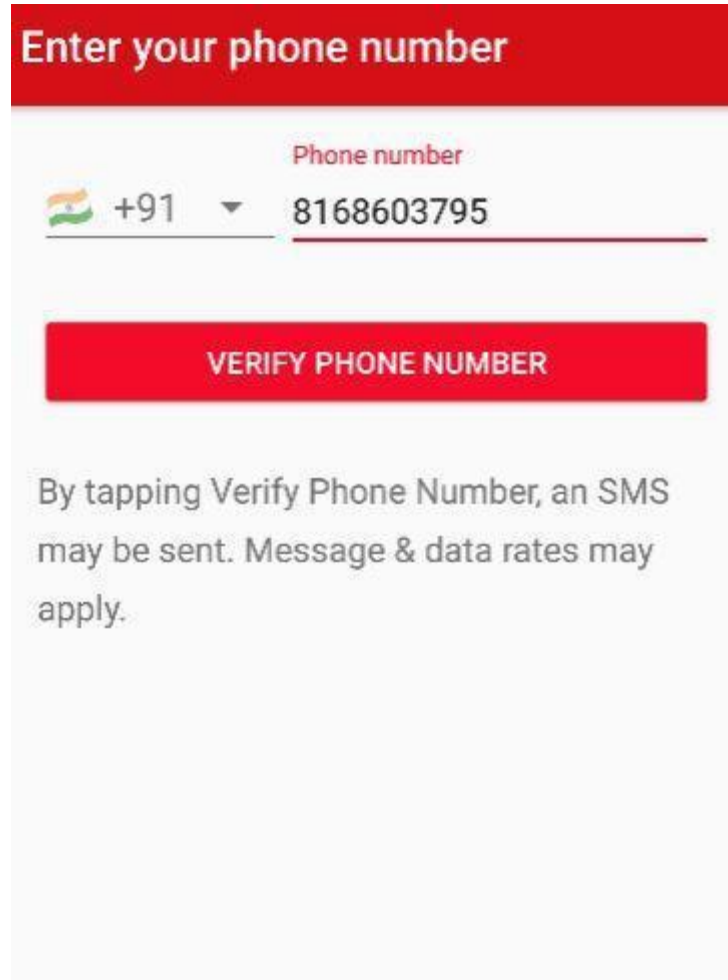


Figure 6.3: Google Verification

- At this screen the user mobile number is verified by the mechanism of one time password (otp) facility that is provided by firebase authentication service. The otp generated will be automatically read upon by the app or can be entered and if correct the user will be signed-in into the app.

6.1.4 Problem List Screen



Figure 6.4: Listed Problems

- These are the problems that are reported by the app user and the problems that are in green color are already solved and problems in red color are unsolved. To report a new problem a circular icon appears at the bottom of the screen and upon clicking it a new frame opens up.

6.1.5 Report New Problem Screen

Jio 4G Vodafone IN

VOLTE 4G

11:23 pm

Campus Connect

Problem Area Security

Title

Lock Problem

Description

Room No 303 Boys Hostle

REPORT PROBLEM

Figure 6.5: Report Problems

- This is the screen where the app user can report new problems by describing his problem in the area provided. Once reported the problem will be sent over to the problem administrator who is responsible for these kinds of problems.

6.1.6 Reported Problem To Administrator

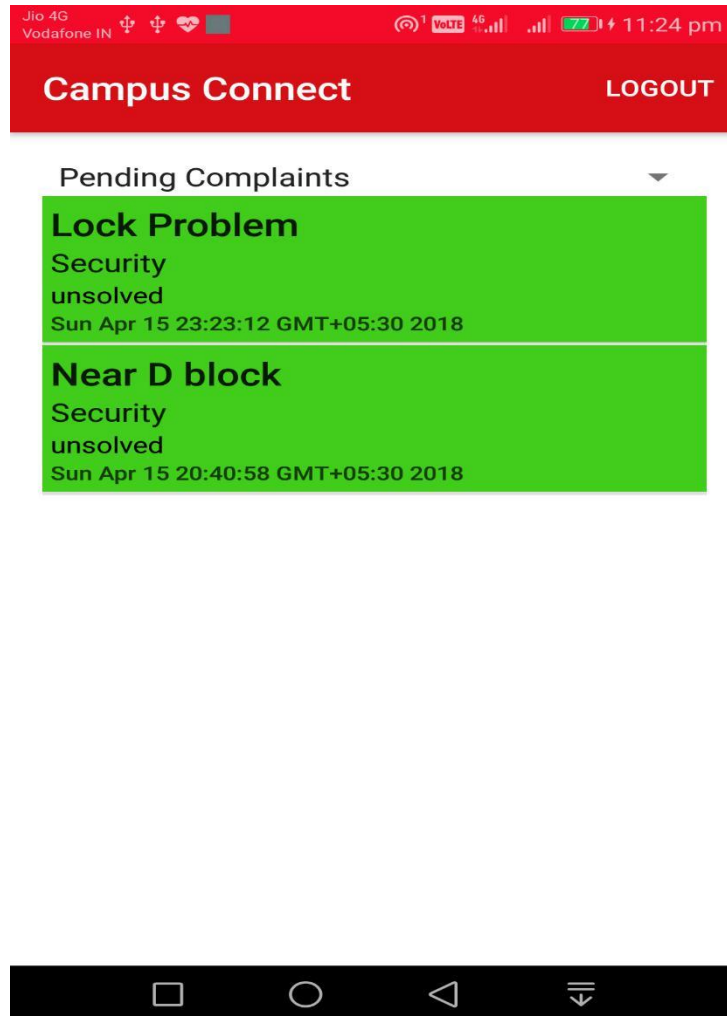


Figure 6.6: Pending Problems

- This is the screen that would appear to the problem administrator upon login. Here all the pending complaints are displayed that have not been solved yet. The problem administrator can select a problem to update its status to be solved and can send an email about its changed status to the app user who reported the problem.

6.1.7 All Problem Reported To Administrator



Figure 6.7: All Problems

- This is the screen that would appear to the problem administrator upon selection from the drop-down list. Here all complaints and pending complaints are displayed. The complaints that are displayed in red color are unsolved and in green color are solved. The problem administrator can select a problem to update its status to be solved

6.1.8 Reported Problem Details

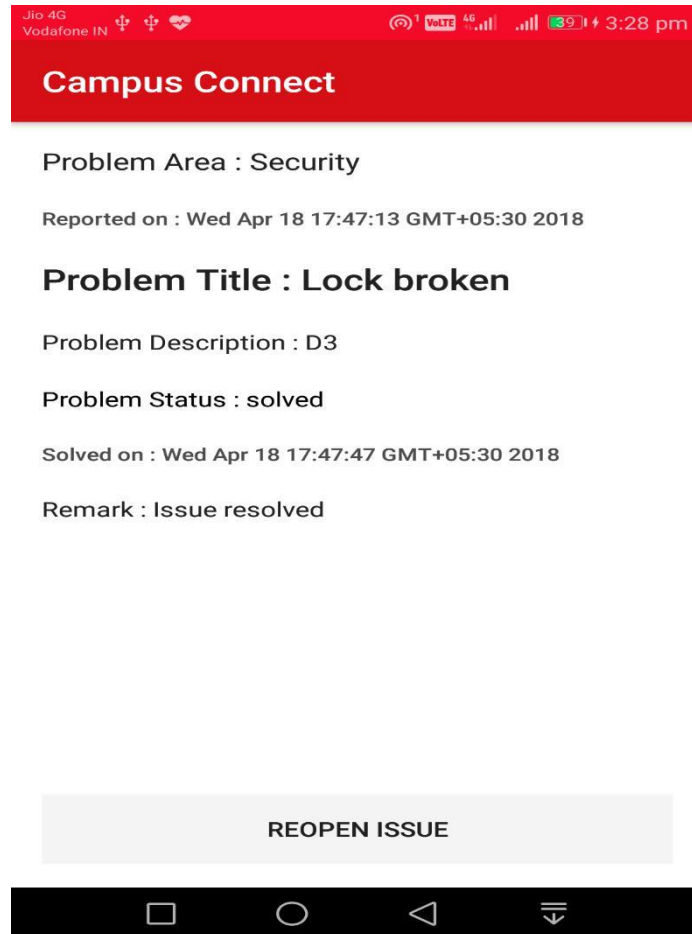


Figure 6.8: Problem Details

- This screen displays the status of the reported problem to the app user who reported the problem. Here all the description is provided as problem title, status, date and remark given by the problem administrator. An another option is provided here to the app user to re-open the problem if he is not satisfied with the result.

CHAPTER 7

CONCLUSION & FUTURE SCOPE

7.1 Conclusion

The application has been developed with much more care that it is free of errors and at the same time it is efficient and less time consuming. The important thing is that the system is robust. It goes through all phases of software development life cycle. So product is accurate. Also provision is provided for future developments in the system. Human being can commit mistake but machine can not commit mistakes, it can be failure like system failure, battery failure etc. hence by making this application we are reliable on machine. All the data is maintained in the firebase database provided by the google. It is well accepted that there is no such application established or the e-solution currently available that provide the combined service of all these facilities. The aim of reducing the time consumed in the whole manual process is achieved by automating the whole process via this app. This app has also enable the people to focus more on their core activities rather than wasting time on unwanted issues. The interface provided in this app is very user friendly and easy to use. The android platform is choosen as it is most common and is available to the large population of india.

7.2 Key Features Provided

- Security as this app provides two times user verification first as with registered mobile number and other one with google provided cloud messaging otp service.
- Different color combination to differentiate between the solved and unsolved problems.
- Very easy to report new problem with given options to choose from.
- Email service is provided to let the user who reported the problem to know that problem he reported has been solved.

7.3 Future Scope

This will help the organizations to do their work in more efficient way by a providing a one common platform to report problems related to any context. The reported problems will then be solved by the concerned authority in due time. There is no scope of confusion as where to report our problems or who is responsible and all.

The emphasis is to reduce the time and effort on all those works which don't help to improve the productivity of the organization. Any organization be it schools, colleges or company etc. can use this app as this app is flexible according to the user requirements.

7.4 Future Advancements

The advancements can be made like

- Enabling easy entry of data in firebase database.
- Enabling the nodal officier/ problem administrator to issue notices regarding any context.
- Dividing the app into the 3-tier module student, teacher and nodal officier.
- Add more features like adding informations about the on-coming events.
- Showing students details like name, section, semmester etc.

CHAPTER 8

REFERENCES

AndroidAuthority. 10 3 2018. <https://www.androidauthority.com> (accessed 3 10, 2018).

androidView. 11 2 2018. <https://stackoverflow.com> (accessed 2 11, 2018).

changeOfBackgroundColor. 15 3 2018. <http://mobiledevtuts.com> (accessed 3 15, 2018).

connectivity. 4 2 2018. <https://console.developers.google.com> (accessed 4 2 4, 2018).

firebaseAddData. 17 2 2018. <https://www.androidmaster.com> (accessed 2 17, 2018).

FirebaseDocs. 17 January 2018. <https://firebase.google.com/docs/>(accessed 3 10, 2018).