BITZONE

A Project submitted in partial fulfilment of the

Degree of

BACHELOR OF COMPUTER APPLICATION

BY

NARENDRA CHATTERJEE

BCA/4536/16

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BIRLA INSTITUTE OF TECHNOLOGY

MESRA, RACHI-835215

2016-2019

DECLARATION CERTIFICATE

This is to certify that the work presented in the project entitled **"BITZONE"** in partial fulfilment of the requirements for the award of the Bachelor of Computer Application of Birla Institute of Technology is an authentic work carried out under my supervision and guidance.

To the best of my knowledge, the content of this project does not form the basis for the award of any previous Degree to anyone else.

DATE:

SIGN.: _____

Mrs. Mrinalini Mehta

Department of Computer Science And Engineering

Birla Institute of Technology

SIGN.: _____

Head

Department of Computer Science And Engineering

Birla Institute of Technology

CERTIFICATE OF APPROVAL

The foregoing project entitled **"BITZONE"** is hereby approved as a creditable study of analysis and has been presented in satisfactory manner to warrant its acceptance. As prerequisite to the degree for which it has been submitted.

It is understood that by this approval the undersigned do not necessarily endorse any conclusion drawn or opinion expressed therein, but approve the thesis for the purpose for which it is submitted.

(Internal Examiner) **(External Examiner)**

(DIRECTOR)

ACKNOWLEDGEMENT

Our sincere thanks to my mentor Mrs. Mrinalini Mehta, Assistant Professor BIT Noida for extending all the support and for her ideas and knowledge for all the project work. She was always ready with answers to all our doubts.

Finally, I would like to thank my friends for their valuable suggestions and inputs that proved to be of invaluable assistance and constantly motive us to learn as much as possible.

Narendra Chatterjee

BCA/4536/13

Birla Institute of Technology

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

BITZONE is a college interactive and informative android application that can be used to resolve many problems encountered daily in the college. This project offers mainly the following services and software to ensure ease and dependability for its customers.

- **Attendance Management:** This app enables teachers not to carry registers rather they can mark attendance on their smart phones or tablets. Students can also check their attendance time to time and it would help them to maintain it. This app will also indicate the low attendance (below 75%) of students and would help teachers to create report on it.
- **College Circular:** This app will enable circulating college related circulars to teachers and students.
- **Chatting:** This app enables messaging between teachers and students.
- **Links:** This app will also be linked with different college links like results, downloading different application forms etc.

1.2 Objective

- The aim of this project is to build an android application that would help in making college management easy.
- This app would be versatile based on what type of user is user it.
- This app would be directly linked to a cloud database to store and retrieve data for different users

1.3 ADVANTAGES

- Low Cost of Ownership
- Multiple Users
- One-Time Fee of the System
- Anywhere Availability
- Scalable and Customizable
- Effective communication between administrator, teachers, and students
- Centrally stored information with zero redundancy

CHAPTER 2

SYSTEM ANALYSIS

2.1 NEED OF NEW SYSTEM

BITZONE is the primary app for day to day college operations. This app is a college management system that allows the Teachers and Students to connect and interact with each other. BITZONE provide capabilities for marking student attendance, circulating the circulars amongst the staff and students, provides a chat portal and many other facilities.

Student Information System also supports:

- Maintaining Student attendance and creating reports
- Sending and viewing different circulars or notifications
- Maintaining the user details.
- Sending assignments or notes to the students.

2.2 FEASIBILITY STUDY

An important outcome of the preliminary investigation is the determination that the system requested is feasible. There are 3 aspects in the feasibility study

2.2.1 TECHNICAL FESIBILITY

This is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may vary considerably, but might include

- The facility to produce outputs in a given time.
- Response time under certain conditions.
- Ability to process a certain volume of transaction data to distant location.

In examining technical feasibility, configuration of the system is given more importance than the actual makes of the hardware. The configuration should give the complete picture about the system's requirements like how many workstations are required, how these units are interconnected so they could operate and communicate smoothly. What speeds of input and output should be achieved at particular quality of printing? Specific hardware and software products can then be evaluated keeping in view with the logical needs.

At the feasibility stage it is desirable that two or three different configurations will be pursued that satisfy the key technical requirements but which represent different levels of ambition and cost. Investigation of these technical alternatives can be aided by approaching a range of suppliers for preliminary discussions. Out of all types of feasibility, technical feasibility generally is the most difficult to determine.

2.2.2 OPERATIONAL FEASIBILITY

It is mainly related to human organizational and political aspects. The points to be considered are:

- What changes will be brought with the systems?
- What organizational structures are distributed?
- What new skills will be required? Do existing staff members have these skills? If not, can they be trained in due course time?

Generally, project will not be rejected simply because of operational infeasibility but such considerations are likely to critically affect the nature and scope of the eventual recommendations. This feasibility study is carries out by a small group of people who are familiar with information system techniques, who understand the parts of the business that are relevant to the project and skilled in system analysis and design process.

As far as this BITZONE is concerned the changes which we have to be brought were only organizational. Then our focus goes towards workstations. Keeping in view of their hardware requirements like network interface card etc.

Regarding this project distribution of organizational structures is also essential because of security concerns, as there are different departments having their particular tasks, I have already mentioned earlier like a system administrator should have the authentication to provide different access permission to its clients.

2.2.3 ECONOMIC FEASIBILITY

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as cost-benefit analysis; the procedure is to determine the benefits and savings that expected from a proposed system and compare them with costs. If benefits outweigh costs. A decision is taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

The feasibility also depends upon quality of staff hired and the proposed duration of time taken in this project sometimes it might be possible due to extension of time duration may fall the project under loss. The study of feasibility changes from phase to phase of the project development.

In this project although this feasibility study doesn't matter much in the case new setup of project because we start according to client specification but on the other hand if we have to modify over existing system, we must take care of our existing resources and must analyse specially the working condition of hardware like server quality etc.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 HARDWARE REQUIREMENT

- Android Smartphone

3.2 SOFTWARE REQUIREMENT

MINIMUM: -

- Android Lollipop
- API Level-23

MAXIMUM: -

- Android Oreo
- API Level-28

CHAPTER 4
SYSTEM DESIGN

4.1 System design consists of six phases-

- Requirement analysis and specification.
- Designing
- Coding
- Testing
- Implementation
- Post-implementation

4.1.1 Requirement analysis and specification-

It is related to getting information or data about the software to be developed. The main task involved in this phase is to know what the customer wants. This is achieved by taking information from the customer with the help of various tools like personal communication; review of thoughts etc. and by getting format of reports or other documents from the customer. This phase is of importance because if the information gathering is not conducted properly then there may be difference between the software being developed and the customer's requirements in the latest stages.

4.1.2 Designing-

In the designing phase there is arrange of data in a proper manner so that the programmers can work accordingly for developing the software. It basically consists of the analysis part of the data, which has been collected in the earlier phase. It also includes the designing aspect, which is related to drawing of data-flow diagrams, entity-relationship diagrams and making data dictionary, functional decomposition diagrams and various other issues, which are concerned with getting a general view of the software. This part of system development life cycle is of most importance because it includes all the study of minute details, which are involved in development of the software.

4.1.3 Coding-

The next phase of system development life cycle is related to coding part of the software i.e. in this phase the programmer develops the software by using various methodologies, which are available for developing software. We can write the code in certain language and can choose the database accordingly. The development team consists of various programmers who are efficient in their respective domains.

4.1.4 Testing-

The phase of system development life cycles is related to the testing of the software after it has been developed. In this phase the testing engineer carries out various testing strategies and tools to check whether the software developed y the programmer is according to the predefined standards and the quality of the software is matching the requirements. The various techniques used for testing by a testing engineer are white box testing, black box testing, unit testing and so on. These all techniques are useful in getting rid if the shortcomings or the failures which are there on the part of programmer. If certain failures are encountered the programmer is informed and he is supposed to eliminate those failures from the software, which is error free from bugs, and ready for implementation on the client side is developed.

4.1.5 Implementation-

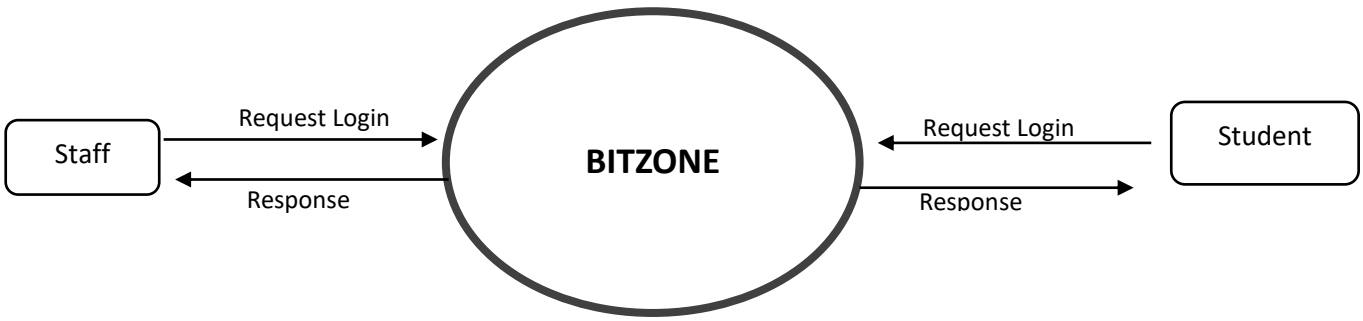
This phase relates to installing the software on the client-side. The various procedures that are involved for implementing a software includes – giving documentation of the details regarding the basic and the technical part of the software. It also includes checking of various hardware resources available with the client and making sure that the software is running properly on their machine.

4.1.6 Maintenance/post-implementation

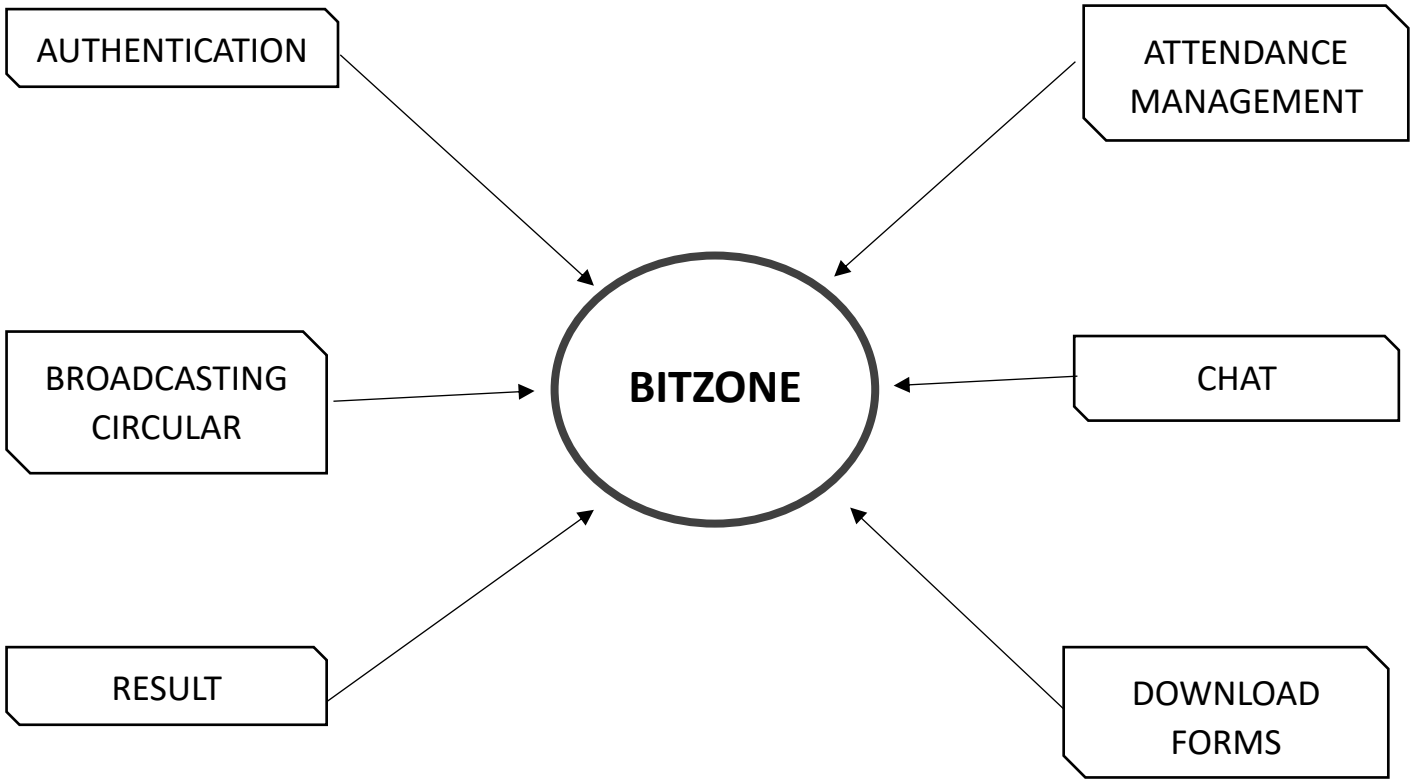
In this phase review is conducted y the technical support engineers on the client side to check whether the software is running properly according to the changing needs of the client from time to time. If there is any failure in the software or the client is not able to access the database then the same is intimated to the programmers or the developers and the certain cases it can be taken care by technical support engineer who is there with the client to check all these errors.

.2 DATA FLOW DIAGRAM

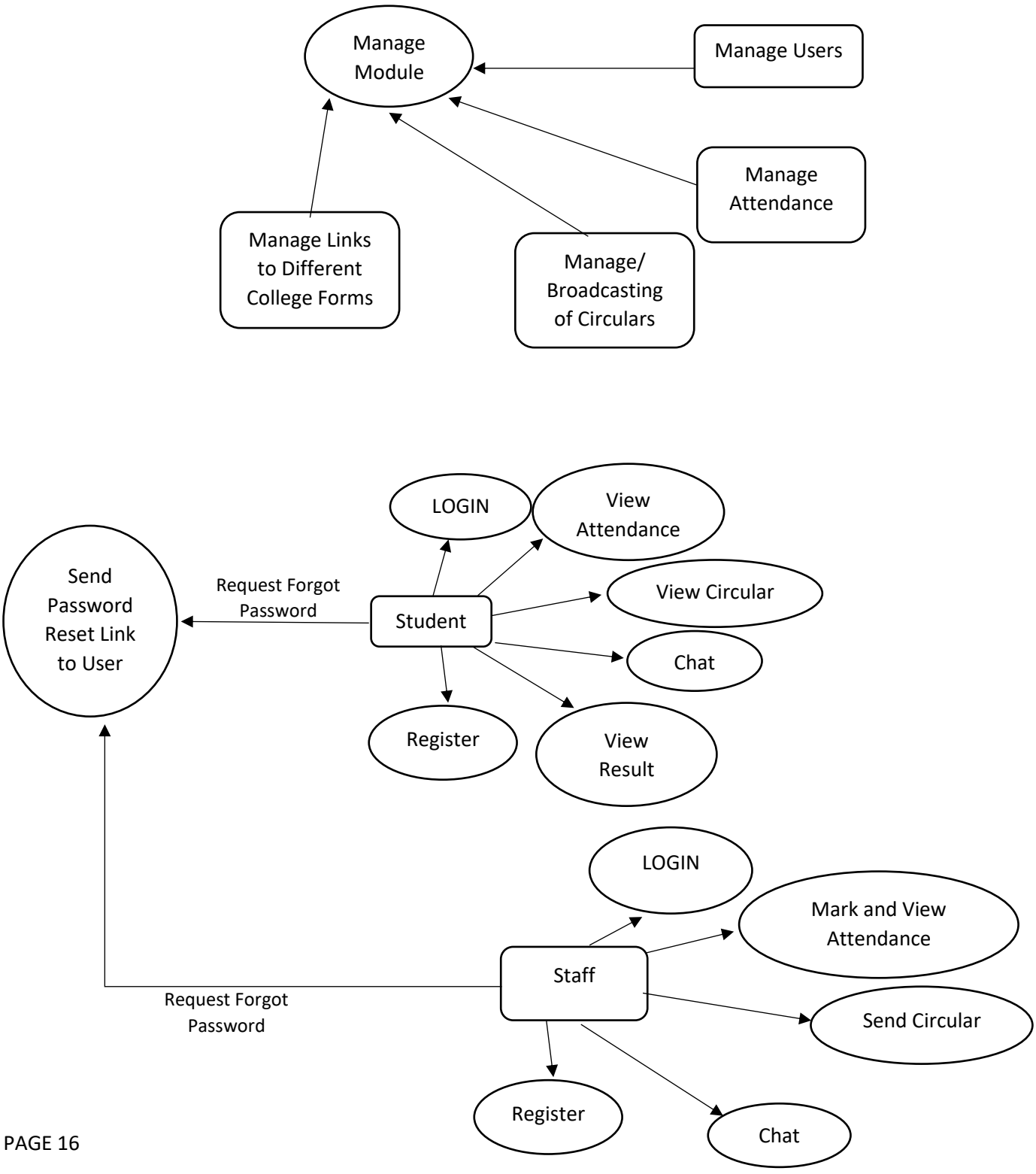
- **DFD CONTEXT LEVEL 0**



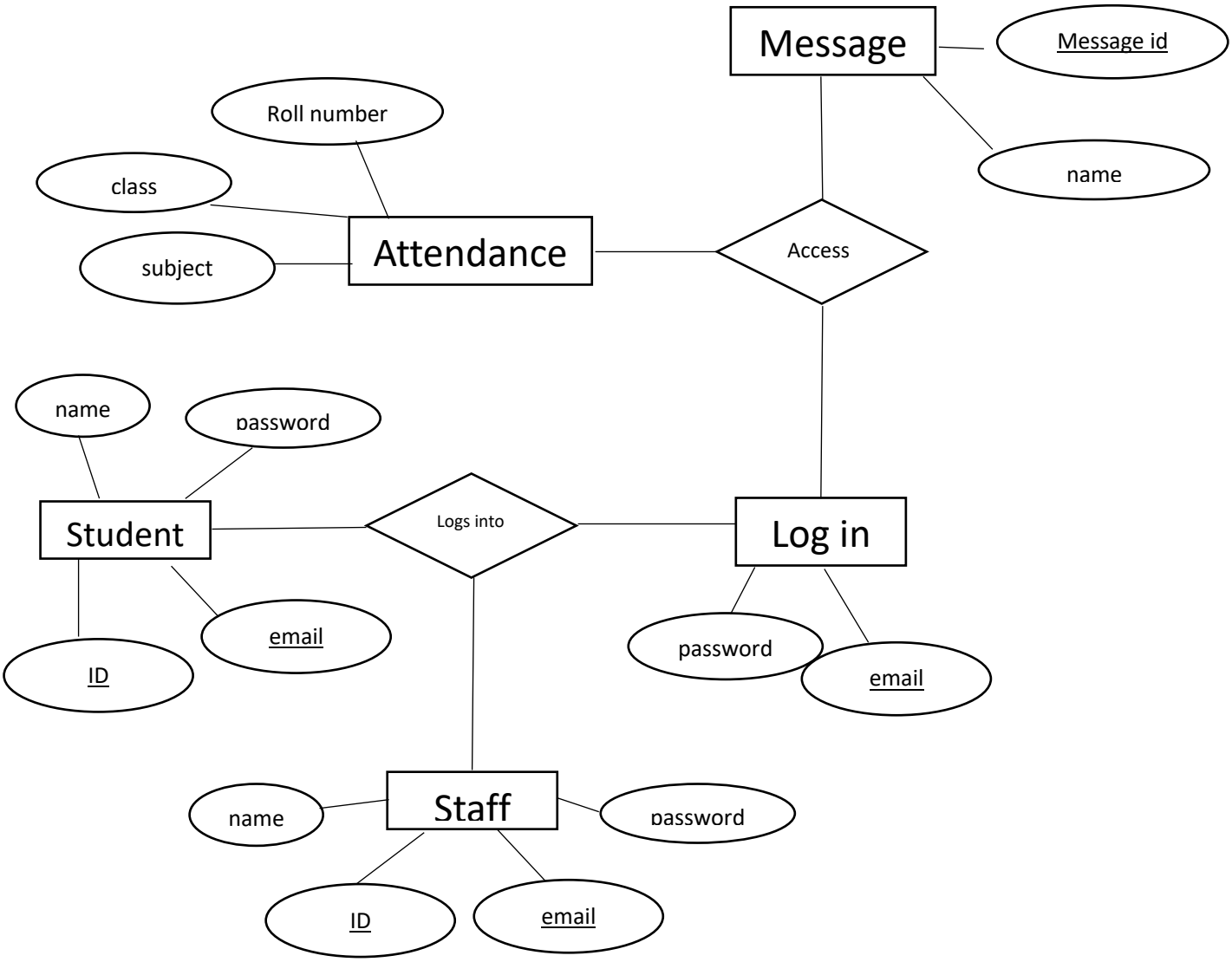
- **DFD CONTEXT LEVEL 1**



- **DFD CONTEXT LEVEL 2**



- **ER DIAGRAM**


Message
Message id
name
Roll number
class
Attendance
subject
Access
name
password
Student
Logs into
Log in
email
ID
password
email
name
Staff
password
ID
email

4.3 Process Description

- Users of the institution can register themselves as a staff or a student accordingly
- If you are already a user, you can directly log in to the application
- By chance the user forgot its password, they can retrieve it by their registered email id. The password reset link would be provided to your email.
- If the user is staff, they can do following activities:
    - Mark and edit Attendance of students
    - Send different circulars and notifications
    - Send assignments
    - Chat with users
- If the user is student, they can do following activities:
    - View Attendance of students
    - View different circulars and notifications
    - View assignments
    - Chat with users
    - View results and different forms of college

CHAPTER 5
CODING AND IMPLEMENTATION

CODING

- **Manifest:**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.bitzone">

    <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />

    <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=".AttendanceMain"
            android:theme="@style/AppTheme.NoActionBar"></activity>
        <activity
            android:name=".StudentDashboard"
            android:label="@string/title_activity_student_dashboard"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".StaffDashboard"
            android:label="@string/title_activity_staff_dashboard"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity android:name=".Main2Activity" />
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".StartActivity" />
        <activity
            android:name=".RegisterActivity"
            android:parentActivityName=".StartActivity" />
        <activity
            android:name=".LoginActivity"
            android:parentActivityName=".StartActivity" />
```

```
        <activity android:name=".Forgetpassword" />
        <activity
            android:name=".StatusActivity"
            android:parentActivityName=".SettingsActivity" />
        <activity
            android:name="com.theartofdev.edmodo.cropper.CropImageActivity"
            android:theme="@style/Base.Theme.AppCompat" />
        <activity
            android:name=".UsersActivity"
            android:parentActivityName=".Main2Activity" />
        <activity android:name=".ProfileActivity">
            <intent-filter>
                <action
android:name="in.tvac.akshaye.lapitchat_TARGET_NOTIFICATION" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity
            android:name=".ChatActivity"
            android:parentActivityName=".Main2Activity" />
        <activity android:name=".SettingsActivity" />
        <activity
            android:name=".AdditionActivity"
            android:label="@string/add"
            android:parentActivityName=".AttendanceMain"
            android:theme="@style/AppTheme" >
        </activity>

        <activity android:name=".SummaryActivity"
            android:parentActivityName=".AttendanceMain" >
        </activity>

    </application>

</manifest>
```

- **Main.xml**

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

   <ImageView
      android:id="@+id/main_logo"
      android:layout_width="85dp"
      android:layout_height="85dp"
      android:src="@drawable/logo"
      android:layout_centerVertical="true"
      android:layout_centerHorizontal="true"/>


   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="horizontal"
      android:id="@+id/main_activity_bottom_buttons"
      android:layout_alignParentBottom="true"
      android:layout_alignParentStart="true"
      >

      <Button
         android:id="@+id/join_btn"
         style="@style/Widget.AppCompat.Button"
         android:layout_width="100dp"
         android:layout_height="50dp"
         android:layout_weight="1"
         android:background="#f8f8f8"
         android:text="JOIN"
         android:textSize="8pt" />

      <Button
         android:id="@+id/login_btn"
         android:layout_width="100dp"
         android:layout_height="50dp"
         android:layout_weight="1"
         android:background="#f8f8f8"
         android:text="LOG IN"
         android:textSize="8pt" />

   </LinearLayout>

   <TextView
```

```
    android:id="@+id/welcome_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/main_logo"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="90dp"
    android:text="Welcome to BITZONE!"
    android:textAlignment="center"
    android:textSize="9pt" />

<LinearLayout
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:id="@+id/main_activity_icons"
    android:layout_centerHorizontal="true"
    android:layout_below="@id/welcome_text"
    android:layout_marginTop="10dp"
    android:orientation="horizontal"
    >

    <ImageView
        android:id="@+id/main_activity_assignment_icon"
        android:layout_width="25dp"
        android:layout_height="25dp"
        android:layout_alignStart="@+id/welcome_text"
        android:layout_below="@+id/welcome_text"
        android:src="@drawable/assignment"
        android:layout_weight="2"
        android:onClick="show_assignment_icon_toast"/>

    <ImageView
        android:id="@+id/main_activity_class_icon"
        android:layout_width="25dp"
        android:layout_height="25dp"
        android:layout_alignEnd="@+id/welcome_text"
        android:layout_alignTop="@+id/main_activity_assignment_icon"
        android:src="@drawable/bookmark"
        android:layout_weight="2"
        android:onClick="show_class_icon_toast"/>

    <ImageView
        android:id="@+id/main_activity_calendar_icon"
        android:layout_width="25dp"
        android:layout_height="25dp"
        android:layout_alignStart="@+id/main_logo"
        android:layout_alignTop="@+id/main_activity_assignment_icon"
        android:src="@drawable/calendar"
        android:layout_marginStart="4dp"
        android:layout_weight="2"
        android:onClick="show_calendar_icon_toast"/>
```

```
        <ImageView
            android:id="@+id/main_activity_announcement_icon"
            android:layout_width="25dp"
            android:layout_height="25dp"
            android:layout_alignEnd="@+id/main_logo"
            android:layout_alignTop="@+id/main_activity_class_icon"
            android:src="@drawable/announcement"
            android:layout_marginEnd="4dp"
            android:layout_weight="2"
            android:onClick="show_announcement_icon_toast"/>

    </LinearLayout>



</RelativeLayout>
```

- **Main.java**

```
package com.example.bitzone;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private Button login_btn,join_btn;
    public AlertDialog.Builder ex;

    @Override
    public void onBackPressed() {
//        super.onBackPressed();
        ex.show();
    }

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

        login_btn = (Button)findViewById(R.id.login_btn);
        join_btn = (Button)findViewById(R.id.join_btn);
```

```
        login_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(),LoginActivity.class);
                startActivity(i);
            }
        });

        join_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(),RegisterActivity.class);
                startActivity(i);
            }
        });

        ex = new AlertDialog.Builder(this).setTitle("Exit ?").setMessage("Do You Want To
Exit").setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                finish();
            }
        }).setNegativeButton("No", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.cancel();
            }
        });


    }

    public void show_assignment_icon_toast(View v){
        TextView toast_msg = new TextView(this);
        toast_msg.setTextSize(12);
        toast_msg.setTextAlignment(View.TEXT_ALIGNMENT_CENTER);
        toast_msg.setText("\uD83D\uDCCB Share assignments with your friends!
\uD83E\uDD1C\uD83E\uDD1B");
        Toast message = new Toast(this);
        message.setView(toast_msg);
        message.setDuration(Toast.LENGTH_SHORT);
        message.show();
    }

    public void show_class_icon_toast(View v){
        TextView toast_msg = new TextView(this);
        toast_msg.setTextSize(12);
        toast_msg.setTextAlignment(View.TEXT_ALIGNMENT_CENTER);
        toast_msg.setText("\uD83D\uDC4F View exam results! \uD83D\uDCAF ");
        Toast message = new Toast(this);
```

```
        message.setView(toast_msg);
        message.setDuration(Toast.LENGTH_SHORT);
        message.show();
    }

    public void show_calendar_icon_toast(View v){
        TextView toast_msg = new TextView(this);
        toast_msg.setTextSize(12);
        toast_msg.setTextAlignment(View.TEXT_ALIGNMENT_CENTER);
        toast_msg.setText("\uD83D\uDE4B Manage class attendance!
\uD83D\uDE4B\u200D♂");
        Toast message = new Toast(this);
        message.setView(toast_msg);
        message.setDuration(Toast.LENGTH_SHORT);
        message.show();
    }

    public void show_announcement_icon_toast(View v){
        TextView toast_msg = new TextView(this);
        toast_msg.setTextSize(12);
        toast_msg.setTextAlignment(View.TEXT_ALIGNMENT_CENTER);
        toast_msg.setText("\uD83D\uDCE3 Receive important notifications! \uD83D\uDCAC");
        Toast message = new Toast(this);
        message.setView(toast_msg);
        message.setDuration(Toast.LENGTH_SHORT);
        message.show();
    }}
```

- **LOGIN.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">



    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/login_con" />
```

```
</android.support.design.widget.CoordinatorLayout>
```

LOGIN.java

```
package com.example.bitzone;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.text.TextUtils;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
```

```
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.iid.FirebaseInstanceId;

public class LoginActivity extends AppCompatActivity implements AdapterView.OnItemSelectedListener {
    Spinner login_spin;
    public AlertDialog.Builder ex;

    EditText email, pass;
    String emailv, passv, log_typ;
    Button log_btn, goreg;

    private ProgressDialog mLoginProgress;
    private FirebaseAuth mAuth;

    private FirebaseDatabase db = FirebaseDatabase.getInstance();
    private DatabaseReference rootRef = db.getReference();

    @Override
    public void onBackPressed() {
//        super.onBackPressed();
        ex.show();
    }

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.login_activity);

        mAuth = FirebaseAuth.getInstance();
```

```
        mLoginProgress = new ProgressDialog(this);



        login_spin = (Spinner) findViewById(R.id.login_spin);

        log_btn = (Button) findViewById(R.id.loginbutton);
        goreg = (Button) findViewById(R.id.registerpage);
        email = (EditText) findViewById(R.id.login_email);
        pass = (EditText) findViewById(R.id.login_pass);

        String type[] = {"Staff", "Student"};
        login_spin.setOnItemSelectedListener(LoginActivity.this);
        ArrayAdapter<String> typelist = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, type);
        typelist.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        login_spin.setAdapter(typelist);

        ex = new AlertDialog.Builder(this).setTitle("Exit ?").setMessage("Do You Want To
Exit").setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                finish();
            }
        }).setNegativeButton("No", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.cancel();
            }
        });
```

```
    }


    public void gotoforget(View v) {
        Intent i = new Intent(this, Forgetpassword.class);
        startActivity(i);
    }

    public void gotoreg(View v) {
        Intent i = new Intent(this, RegisterActivity.class);
        startActivity(i);
    }


    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
        log_typ = parent.getItemAtPosition(position).toString();
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }

    public final boolean isInternetOn() {
        // get Connectivity Manager object to check connection
        ConnectivityManager connec = (ConnectivityManager)
getSystemService(getBaseContext().CONNECTIVITY_SERVICE);
        // Check for network connections
```

```
    if (connec.getNetworkInfo(0).getState() == android.net.NetworkInfo.State.CONNECTED ||
            connec.getNetworkInfo(0).getState() == android.net.NetworkInfo.State.CONNECTING ||
            connec.getNetworkInfo(1).getState() == android.net.NetworkInfo.State.CONNECTING ||
            connec.getNetworkInfo(1).getState() == android.net.NetworkInfo.State.CONNECTED) {
        // if connected with internet
        return true;
    } else if (
            connec.getNetworkInfo(0).getState() == android.net.NetworkInfo.State.DISCONNECTED ||
                    connec.getNetworkInfo(1).getState() == android.net.NetworkInfo.State.DISCONNECTED) {
        return false;
    }
    return false;
}

public void doLogin(View v) {
    if (isInternetOn()) {
        emailv = email.getText().toString();
        passv = pass.getText().toString();
        final ProgressDialog pdlg = new ProgressDialog(this);
        pdlg.setTitle("Loging In");
        pdlg.setMessage("Please Wait");

        if(!TextUtils.isEmpty(emailv) || !TextUtils.isEmpty(passv)){

            mLoginProgress.setTitle("Logging In");
            mLoginProgress.setMessage("Please wait while we check your credentials.");
            mLoginProgress.setCanceledOnTouchOutside(false);
            mLoginProgress.show();

            loginUser(emailv, passv);
```

```
        }

        if ((emailv.matches("")) && (passv.matches(""))) {
            //blank text field toast
            Toast.makeText(this, "Please Enter Detail", Toast.LENGTH_SHORT).show();
        } else if ((emailv != null) && (passv != null)) {
            pdlg.show();



            if(log_typ.equals("Staff")) {
                rootRef.child(log_typ).addChildEventListener(new ChildEventListener() {
                    @Override
                    public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {
                        String email = dataSnapshot.child("email").getValue().toString();
                        String password = dataSnapshot.child("password").getValue().toString();
                        if (emailv.equals(email) && passv.equals(password)) {
                            pdlg.cancel();
                            Toast.makeText(LoginActivity.this, "Login Succesfully",
Toast.LENGTH_SHORT).show();
                            Intent intent = new Intent(LoginActivity.this, StaffDashboard.class);
                            startActivity(intent);
                        }
                        else{
                            pdlg.cancel();
                            Toast.makeText(LoginActivity.this, "Login UnSuccesfull",
Toast.LENGTH_SHORT).show();


                        }


                    }
```

```
        @Override
        public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

        }

        @Override
        public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {

        }

        @Override
        public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
    });
}
if(log_typ.equals("Student")){
    rootRef.child(log_typ).addChildEventListener(new ChildEventListener() {
        @Override
        public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {
            String email = dataSnapshot.child("email").getValue().toString();
            String password = dataSnapshot.child("password").getValue().toString();
            if (emailv.equals(email) && passv.equals(password)) {
                pdlg.cancel();
```

```
                    Toast.makeText(LoginActivity.this, "Login Succesfully",
Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent(LoginActivity.this, StudentDashboard.class);
                    startActivity(intent);
                }
                else{
                    pdlg.cancel();
                    Toast.makeText(LoginActivity.this, "Login UnSuccesfull",
Toast.LENGTH_SHORT).show();

                }
            }

            @Override
            public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

            }

            @Override
            public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {

            }

            @Override
            public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {
```

```
                }
            });
        }



    }
} else {
    Toast.makeText(this, "Please Check Internet Connection ", Toast.LENGTH_LONG).show();
}
}

private void loginUser(String email, String password) {



    mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {

            if(task.isSuccessful()){

                mLoginProgress.dismiss();

                String current_user_id = mAuth.getCurrentUser().getUid();
                String deviceToken = FirebaseInstanceId.getInstance().getToken();


rootRef.child("Users").child(current_user_id).child("device_token").setValue(deviceToken).addOnSuccessLis
tener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {
```

```
//                    Intent mainIntent = new Intent(LoginActivity.this, Main2Activity.class);
//                    mainIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
//                    startActivity(mainIntent);
//                    finish();



                }
            });




        } else {

            mLoginProgress.hide();

            String task_result = task.getException().getMessage().toString();

            Toast.makeText(LoginActivity.this, "Error : " + task_result, Toast.LENGTH_LONG).show();

        }

    }
});

}
}
```

- **Register.xml**

```
<?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="#ffffff"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/register_activity">

    <LinearLayout
        android:id="@+id/linearLayout2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="69dp"
        android:orientation="vertical">

        <EditText
            android:id="@+id/reg_name"
            android:layout_width="match_parent"
            android:layout_height="40dp"
            android:layout_marginTop="8dp"
            android:background="@drawable/rounded_edittext"
            android:ems="10"
            android:hint="  Name"
            android:inputType="textPersonName" />

        <EditText
            android:id="@+id/reg_email"
            android:layout_width="match_parent"
            android:layout_height="40dp"
            android:layout_marginTop="8dp"
            android:background="@drawable/rounded_edittext"
            android:ems="10"
            android:hint="  Email"
            android:inputType="textEmailAddress" />

        <EditText
            android:id="@+id/reg_pass"
            android:layout_width="match_parent"
```

```
            android:layout_height="40dp"
            android:layout_marginTop="8dp"
            android:background="@drawable/rounded_edittext"
            android:ems="10"
            android:hint=" Password"
            android:inputType="textPassword" />

        <Spinner
            android:id="@+id/reg_spin"
            android:layout_width="fill_parent"
            android:layout_height="50dp"
            android:layout_marginTop="8dp"
            android:background="@drawable/rounded_edittext"
            android:spinnerMode="dropdown" />

    </LinearLayout>

    <Button
        android:id="@+id/Reg_btn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/linearLayout2"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_marginTop="35dp"
        android:background="@drawable/custom_button"
        android:onClick="doRegister"
        android:text="Register"
        android:textSize="30dp" />

    <TextView
        android:id="@+id/careiii"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="40dp"
        android:text="Register With Valid Detail"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#000000"
        android:textIsSelectable="false"
        android:textSize="20dp"
        android:textStyle="normal|bold" />

</RelativeLayout>
```

- **Register.java**

```
package com.example.bitzone;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
```

```
import com.google.firebase.iid.FirebaseInstanceId;

import java.util.HashMap;

public class RegisterActivity extends AppCompatActivity implements AdapterView.OnItemSelectedListener
{
    EditText regnm, regpass, regmail;
    Button reg_btn;
    String regnmv, regpassv, regmailv, regtypev;
    Spinner reg_spinner;
    public AlertDialog.Builder ex;
    private FirebaseDatabase db = FirebaseDatabase.getInstance();
    private DatabaseReference rootRef = db.getReference();

    private FirebaseAuth mAuth;
    private ProgressDialog mRegProgress;

    @Override
    public void onBackPressed() {
//        super.onBackPressed();
        ex.show();
    }

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.register_activity);

        mRegProgress = new ProgressDialog(this);

        mAuth = FirebaseAuth.getInstance();
```

```
        reg_btn = (Button) findViewById(R.id.Reg_btn);
        regnm = (EditText) findViewById(R.id.reg_name);
        regpass = (EditText) findViewById(R.id.reg_pass);
        regmail = (EditText) findViewById(R.id.reg_email);
        reg_spinner = (Spinner) findViewById(R.id.reg_spin);




        String type[] = {"Staff", "Student"};
        reg_spinner.setOnItemSelectedListener(RegisterActivity.this);
        ArrayAdapter<String> typelist = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, type);
        typelist.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        reg_spinner.setAdapter(typelist);


        ex = new AlertDialog.Builder(this).setTitle("Exit ?").setMessage("Do You Want To
Exit").setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                finish();
            }
        }).setNegativeButton("No", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.cancel();
            }
        });
```

```
    }

    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
        regtypev = parent.getItemAtPosition(position).toString();

    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }

    public final boolean isInternetOn() {
        // get Connectivity Manager object to check connection
        ConnectivityManager connec = (ConnectivityManager)
getSystemService(getBaseContext().CONNECTIVITY_SERVICE);
        // Check for network connections
        if (connec.getNetworkInfo(0).getState() == android.net.NetworkInfo.State.CONNECTED ||
                connec.getNetworkInfo(0).getState() == android.net.NetworkInfo.State.CONNECTING ||
                connec.getNetworkInfo(1).getState() == android.net.NetworkInfo.State.CONNECTING ||
                connec.getNetworkInfo(1).getState() == android.net.NetworkInfo.State.CONNECTED) {
            // if connected with internet
            return true;
        } else if (
                connec.getNetworkInfo(0).getState() == android.net.NetworkInfo.State.DISCONNECTED ||
                        connec.getNetworkInfo(1).getState() == android.net.NetworkInfo.State.DISCONNECTED) {
            return false;
        }
        return false;
```

```
}


public void doRegister(View view) {
    if (isInternetOn()) {
        regnmv = regnm.getText().toString();
        regpassv = regpass.getText().toString();
        regmailv = regmail.getText().toString();
        HashMap<String, String> usermap = new HashMap<>();
        usermap.put("name",regnmv);
        usermap.put("email",regmailv);
        usermap.put("password",regpassv);

        if(!TextUtils.isEmpty(regnmv) || !TextUtils.isEmpty(regmailv) || !TextUtils.isEmpty(regpassv)){

            mRegProgress.setTitle("Registering User");
            mRegProgress.setMessage("Please wait while we create your account !");
            mRegProgress.setCanceledOnTouchOutside(false);
            mRegProgress.show();

            register_user(regnmv, regmailv, regpassv);

        }


        if (((regnmv.matches("")) && (regpassv.matches(""))) && (regmailv.matches(""))) {
            Toast.makeText(RegisterActivity.this, "Please Enter Detail", Toast.LENGTH_SHORT).show();
        } else {
            if (((regnmv != null) && (regpassv != null)) && ((regmailv != null)) && (regtypev != null)) {
                if (regpassv.length() >= 8) {
                    final ProgressDialog pdlg = new ProgressDialog(this);
```

```
                pdlg.setTitle("Registering");
                pdlg.setMessage("Please Wait");
                pdlg.show();
                //crosscheck logic

                rootRef.child(regtypev).push().setValue(usermap).addOnCompleteListener(new
OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {
                        if (task.isSuccessful()) {
                            Toast.makeText(RegisterActivity.this, "Register Sucessfull",
Toast.LENGTH_SHORT).show();
                            pdlg.cancel();
                        } else {
                            Toast.makeText(RegisterActivity.this, "Register Unsucessfull" + task.getResult(),
Toast.LENGTH_SHORT).show();
                            pdlg.cancel();
                        }
                    }
                });



            } else {
                Toast.makeText(RegisterActivity.this, "Password Should Be 8 Digit Long",
Toast.LENGTH_SHORT).show();
            }
        }
    }
} else {
    Toast.makeText(this, "Please Check Internet Connection ", Toast.LENGTH_LONG).show();
}
```

```
    }


    private void register_user(final String display_name, String email, String password) {


        mAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {


                if(task.isSuccessful()){



                    FirebaseUser current_user = FirebaseAuth.getInstance().getCurrentUser();
                    String uid = current_user.getUid();


                    rootRef = FirebaseDatabase.getInstance().getReference().child("Users").child(uid);


                    String device_token = FirebaseInstanceId.getInstance().getToken();


                    HashMap<String, String> userMap = new HashMap<>();
                    userMap.put("name", display_name);
                    userMap.put("status", "Hi there I'm using BitZone Chat App.");
                    userMap.put("image", "default");
                    userMap.put("thumb_image", "default");
                    userMap.put("device_token", device_token);


                    rootRef.setValue(userMap).addOnCompleteListener(new OnCompleteListener<Void>() {
                        @Override
```

```
                public void onComplete(@NonNull Task<Void> task) {


                    if(task.isSuccessful()){


                        mRegProgress.dismiss();


                        Intent mainIntent = new Intent(RegisterActivity.this, MainActivity.class);
                        mainIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
                        startActivity(mainIntent);
                        finish();
                    }
                }
            });
        } else {


            mRegProgress.hide();
            Toast.makeText(RegisterActivity.this, "Cannot Sign in. Please check the form and try again.",
Toast.LENGTH_LONG).show();


        }
    }
});
  }
}
```

- **Attendance main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   android:id="@+id/main_content"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:fitsSystemWindows="true"
   tools:context=".MainActivity">

   <android.support.design.widget.AppBarLayout
      android:id="@+id/appbar"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:paddingTop="@dimen/appbar_padding_top"
      android:theme="@style/AppTheme.AppBarOverlay">

      <android.support.v7.widget.Toolbar
         android:id="@+id/toolbar"
         android:layout_width="match_parent"
         android:layout_height="?attr/actionBarSize"
         android:background="?attr/colorPrimary"
         app:popupTheme="@style/AppTheme.PopupOverlay"
         app:layout_scrollFlags="scroll|enterAlways">

      </android.support.v7.widget.Toolbar>
      <android.support.design.widget.TabLayout
         android:id="@+id/tabs"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"/>

   </android.support.design.widget.AppBarLayout>
   <android.support.v4.view.ViewPager
      android:id="@+id/container"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      app:layout_behavior="@string/appbar_scrolling_view_behavior"/>


</android.support.design.widget.CoordinatorLayout>
```

- **Attendance fragments.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
                android:orientation="vertical"
                android:layout_width="match_parent"
                android:layout_height="match_parent">


            <android.support.v7.widget.RecyclerView
                xmlns:android="http://schemas.android.com/apk/res/android"
                android:orientation="vertical"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:id="@+id/attendance_container_recycler_view">
            </android.support.v7.widget.RecyclerView>
        </FrameLayout>
```

- **Attendance.java**

```
package com.example.bitzone;


import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;

import java.util.UUID;


public class Attendance {


    private Lecture mLecture;

    private List<Student> mStudents;

    private UUID mId;

    private Date mDate;


    public Attendance(UUID lectureId)

    {

        this(lectureId, UUID.randomUUID(),new Date(),null);
```

```
}

//COnstructor for databaes

public Attendance(UUID lectureId, UUID id, Date date, String presentString)
{
   mLecture=LectureLab.get().getLectureById(lectureId);
   mStudents=mLecture.getKlass().getSpecificStudents(mLecture.getStudentStartingRollNo(),
         mLecture.getStudentLastRollNo());
   mId=id;
   mDate=date;

   SylveryteJoinSplit.setPresents(mStudents,presentString);
   mLecture.addAttendance(this);
}

public List<Student> getStudents() {
   return mStudents;
}

public UUID getId() {
   return mId;
}

public UUID getLectureId() {
   return mLecture.getId();
}


public Date getDate() {
```

```
        return mDate;
    }

    public static Date getDateByString(String s) throws ParseException {
        return new SimpleDateFormat("MM/dd/yyyy HH:mm:ss").parse(s);
    }

    public String getDateString ()
    {
        return new SimpleDateFormat("MM/dd/yyyy HH:mm:ss").format(mDate);
    }

    public String getExtraInfo()
    {
        return mLecture.getKlass().getKlassName()+
                "\nNo of students "+mStudents.size()+"\nDate : "+getDateString();
    }

    public void setDate(Date date) {
        mDate = date;
    }


    public String getAttendanceName()
    {
        return mLecture.getLectureName();
    }

}
```

- **Attendance Add Fragment.java**

```
package com.example.bitzone;

import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.support.v7.widget.GridLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;

import java.util.List;
import java.util.UUID;

/**
 * Created by sylveryte on 21/2/16.
 */
public class AttendanceAddFragment extends Fragment {


    private RecyclerView mAttendanceRecyclerView;
    private RollAdapter mRollAdapter;
    private Lecture mLecture;

    private Attendance mAttendance;
    private List<Student> mStudents;
```

```
    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        View view=inflater.inflate(R.layout.add_attendance_fragment, container, false);


mAttendanceRecyclerView=(RecyclerView)view.findViewById(R.id.attendance_container_recycler_view);
        mAttendanceRecyclerView.setLayoutManager(new GridLayoutManager(getActivity(), 4));

        getActivity().setTitle("Attendance");

        Intent intent=getActivity().getIntent();
        UUID uuid=(UUID)intent.getSerializableExtra(ListDialog.ATTENDANCECODE);

        mAttendance=AttendanceLab.get().getAttendanceById(uuid);

        mStudents=mAttendance.getStudents();
        mRollAdapter=new RollAdapter();

        mAttendanceRecyclerView.setAdapter(mRollAdapter);

        return view;
    }

    @Override
    public void onPause() {
        super.onPause();
        AttendanceLab.get().updateDatabaseOfAttendance(mAttendance);
    }
```

```
private class RollHolder extends RecyclerView.ViewHolder
{
    private Student mStudent;
    public Button mButton;

    public RollHolder(View itemView) {
        super(itemView);
        mButton=(Button)itemView.findViewById(R.id.roll_button_bro);
        mButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                mStudent.setIsPresent(!mStudent.isPresent());

                updateUI();
            }
        });
    }

    public void bind(Student student)
    {
        mStudent=student;
        mButton.setText(String.format("%d ", student.getRollNo()));
        if (!mStudent.isPresent())
        {
            mButton.setBackgroundResource(R.drawable.absent);
        }
        else {
            mButton.setBackgroundResource(R.drawable.present);
        }
    }
```

```
}
private void updateUI()
{
    if (mRollAdapter!=null)
    {
        mRollAdapter.notifyDataSetChanged();
    }
}

private class RollAdapter extends RecyclerView.Adapter<RollHolder>
{
    @Override
    public RollHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        LayoutInflater layoutInflater= LayoutInflater.from(getActivity());
        View view=layoutInflater.inflate(R.layout.roll_button_attendance,parent,false);
        return new RollHolder(view);
    }

    @Override
    public void onBindViewHolder(RollHolder holder, int position) {
        holder.bind(mStudents.get(position));
    }


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

```
    }
}
```

- **Attendance**

```
package com.example.bitzone;

import android.os.Parcelable;
import android.support.design.widget.TabLayout;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;
import android.support.v4.app.FragmentStatePagerAdapter;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;

public class AttendanceMain extends AppCompatActivity {

    /**
     * The {@link android.support.v4.view.PagerAdapter} that will provide
     * fragments for each of the sections. We use a
     * {@link FragmentPagerAdapter} derivative, which will keep every
     * loaded fragment in memory. If this becomes too memory intensive, it
     * may be best to switch to a
     * {@link android.support.v4.app.FragmentStatePagerAdapter}.
     */
    private SectionsPagerAdapter mSectionsPagerAdapter;
```

```
/**
 * The {@link ViewPager} that will host the section contents.
 */
private ViewPager mViewPager;
private FragmentManager mFM;

@Override
protected void onResume() {
    super.onResume();
    if (mSectionsPagerAdapter != null) {
        mSectionsPagerAdapter.notifyDataSetChanged();
    }
}

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



    //set db
    DatabaseLab.getInstance(this);

    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    // Create the adapter that will return a fragment for each of the three
    // primary sections of the activity.
    mFM=getSupportFragmentManager();
    mSectionsPagerAdapter = new SectionsPagerAdapter(mFM);

    // Set up the ViewPager with the sections adapter.
```

```
        mViewPager = (ViewPager) findViewById(R.id.container);
        mViewPager.setAdapter(mSectionsPagerAdapter);
        mViewPager.addOnPageChangeListener(new ViewPager.OnPageChangeListener() {
            @Override
            public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {

            }

            @Override
            public void onPageSelected(int position) {
                if (mSectionsPagerAdapter != null) {
                    mSectionsPagerAdapter.notifyDataSetChanged();
                }
            }

            @Override
            public void onPageScrollStateChanged(int state) {
//                mSectionsPagerAdapter.notifyDataSetChanged();
            }
        });

        TabLayout tabLayout = (TabLayout) findViewById(R.id.tabs);
        tabLayout.setupWithViewPager(mViewPager);

    }


    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
```

```
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        switch (id) {

            case R.id.add_klass :
            {
                startActivityForResult(AdditionActivity.fetchIntent(AttendanceMain.this,
AdditionActivity.ADDKLASS),0);
                break;
            }
            case R.id.add_lecture :
            {
                startActivityForResult(AdditionActivity.fetchIntent(AttendanceMain.this,
AdditionActivity.ADDLECTURE),0);
                break;
            }
            case R.id.add_attendance :
            {
                FragmentManager fragmentManager=getSupportFragmentManager();
                ListDialog dialog=new ListDialog();
                dialog.show(fragmentManager,"LISTDIALOG");
            }
```

```
    }

    return super.onOptionsItemSelected(item);
}


/**
 * A {@link FragmentPagerAdapter} that returns a fragment corresponding to
 * one of the sections/tabs/pages.
 */
public class SectionsPagerAdapter extends FragmentStatePagerAdapter {

    public SectionsPagerAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public Fragment getItem(int position) {
        // getItem is called to instantiate the fragment for the given page.
        // Return a PlaceholderFragment (defined as a static inner class below).
        switch (position) {
            case 0: return AttendanceFragment.newInstance();
            case 1: return LectureFragment.newInstance();
            case 2 : return KlassFragment.newInstance();
            default: return null;
        }
    }


    @Override
    public int getCount() {
```

```
        // Show 3 total pages.
        return 3;
    }

    @Override
    public Parcelable saveState() {
        return null;
    }

    @Override
    public int getItemPosition(Object object) {
        return POSITION_NONE;
    }

    @Override
    public CharSequence getPageTitle(int position) {
        switch (position) {
            case 0:
                return "Attendance";
            case 1:
                return "Lecture";
            case 2:
                return "Class";
        }
        return null;
    }
  }
}
```

- **Messaging.xml**

```
<?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="#ededed"
    >

    <include
        android:id="@+id/chat_app_bar"
        layout="@layout/app_bar_layout" />

    <android.support.v4.widget.SwipeRefreshLayout
        android:id="@+id/message_swipe_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_above="@+id/linearLayout"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/chat_app_bar">

        <android.support.v7.widget.RecyclerView
            android:id="@+id/messages_list"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_above="@+id/linearLayout"
            android:layout_alignParentStart="true"
            android:layout_below="@+id/chat_app_bar"></android.support.v7.widget.RecyclerView>
```

```
</android.support.v4.widget.SwipeRefreshLayout>

<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentStart="true"
    android:background="@android:color/white"
    android:orientation="horizontal"
    android:weightSum="10">

    <ImageButton
        android:id="@+id/chat_add_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:alpha="0.5"
        android:background="@android:color/white"
        android:padding="10dp"
        app:srcCompat="@drawable/ic_add_black_24dp" />

    <EditText
        android:id="@+id/chat_message_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="8"
        android:background="@android:color/white"
        android:ems="10"
        android:hint="Enter Message..."
        android:inputType="textPersonName"
```

```
        android:paddingBottom="12dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:paddingTop="14dp" />

    <ImageButton
        android:id="@+id/chat_send_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:alpha="0.5"
        android:background="@android:color/white"
        android:padding="10dp"
        app:srcCompat="@drawable/ic_send_black_24dp" />
  </LinearLayout>


</RelativeLayout>
```

- **Chat.java**

```
package com.example.bitzone;

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.widget.SwipeRefreshLayout;
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.text.TextUtils;
import android.util.Log;
```

```
import android.view.LayoutInflater;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ServerValue;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import de.hdodenhof.circleimageview.CircleImageView;

public class ChatActivity extends AppCompatActivity {

    private String mChatUser;
    private Toolbar mChatToolbar;

    private DatabaseReference mRootRef;

    private TextView mTitleView;
    private TextView mLastSeenView;
    private CircleImageView mProfileImage;
    private FirebaseAuth mAuth;
    private String mCurrentUserId;

    private ImageButton mChatAddBtn;
    private ImageButton mChatSendBtn;
    private EditText mChatMessageView;

    private RecyclerView mMessagesList;
    private SwipeRefreshLayout mRefreshLayout;
```

```
    private final List<Messages> messagesList = new ArrayList<>();
    private LinearLayoutManager mLinearLayout;
    private MessageAdapter mAdapter;

    private static final int TOTAL_ITEMS_TO_LOAD = 10;
    private int mCurrentPage = 1;

    private static final int GALLERY_PICK = 1;

    // Storage Firebase
    private StorageReference mImageStorage;


    //New Solution
    private int itemPos = 0;

    private String mLastKey = "";
    private String mPrevKey = "";


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

        mChatToolbar = (Toolbar) findViewById(R.id.chat_app_bar);
        setSupportActionBar(mChatToolbar);

        ActionBar actionBar = getSupportActionBar();

        actionBar.setDisplayHomeAsUpEnabled(true);
        actionBar.setDisplayShowCustomEnabled(true);

        mRootRef = FirebaseDatabase.getInstance().getReference();
        mAuth = FirebaseAuth.getInstance();
        mCurrentUserId = mAuth.getCurrentUser().getUid();

        mChatUser = getIntent().getStringExtra("user_id");
        String userName = getIntent().getStringExtra("user_name");

        LayoutInflater inflater = (LayoutInflater)
this.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View action_bar_view = inflater.inflate(R.layout.chat_custom_bar, null);

        actionBar.setCustomView(action_bar_view);
```

```
        // ---- Custom Action bar Items ----

        mTitleView = (TextView) findViewById(R.id.custom_bar_title);
        mLastSeenView = (TextView) findViewById(R.id.custom_bar_seen);
        mProfileImage = (CircleImageView) findViewById(R.id.custom_bar_image);

        mChatAddBtn = (ImageButton) findViewById(R.id.chat_add_btn);
        mChatSendBtn = (ImageButton) findViewById(R.id.chat_send_btn);
        mChatMessageView = (EditText) findViewById(R.id.chat_message_view);

        mAdapter = new MessageAdapter(messagesList);

        mMessagesList = (RecyclerView) findViewById(R.id.messages_list);
        mRefreshLayout = (SwipeRefreshLayout) findViewById(R.id.message_swipe_layout);
        mLinearLayout = new LinearLayoutManager(this);

        mMessagesList.setHasFixedSize(true);
        mMessagesList.setLayoutManager(mLinearLayout);

        mMessagesList.setAdapter(mAdapter);

        //------- IMAGE STORAGE ---------
        mImageStorage = FirebaseStorage.getInstance().getReference();


mRootRef.child("Chat").child(mCurrentUserId).child(mChatUser).child("seen").setValue(true)
;

        loadMessages();


        mTitleView.setText(userName);

        mRootRef.child("Users").child(mChatUser).addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {

                String online = dataSnapshot.child("online").getValue().toString();
                String image = dataSnapshot.child("image").getValue().toString();

                if(online.equals("true")) {

                    mLastSeenView.setText("Online");

                } else {
```

```
                GetTimeAgo getTimeAgo = new GetTimeAgo();

                long lastTime = Long.parseLong(online);

                String lastSeenTime = getTimeAgo.getTimeAgo(lastTime,
getApplicationContext());

                mLastSeenView.setText(lastSeenTime);

            }

        }

        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });


    mRootRef.child("Chat").child(mCurrentUserId).addValueEventListener(new
ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {

            if(!dataSnapshot.hasChild(mChatUser)){

                Map chatAddMap = new HashMap();
                chatAddMap.put("seen", false);
                chatAddMap.put("timestamp", ServerValue.TIMESTAMP);

                Map chatUserMap = new HashMap();
                chatUserMap.put("Chat/" + mCurrentUserId + "/" + mChatUser, chatAddMap);
                chatUserMap.put("Chat/" + mChatUser + "/" + mCurrentUserId, chatAddMap);

                mRootRef.updateChildren(chatUserMap, new
DatabaseReference.CompletionListener() {
                    @Override
                    public void onComplete(DatabaseError databaseError, DatabaseReference
databaseReference) {

                        if(databaseError != null){

                            Log.d("CHAT_LOG", databaseError.getMessage().toString());

                        }
```

```
                    }
                });

            }

        }

        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });



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

            sendMessage();

        }
    });



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

            Intent galleryIntent = new Intent();
            galleryIntent.setType("image/*");
            galleryIntent.setAction(Intent.ACTION_GET_CONTENT);

            startActivityForResult(Intent.createChooser(galleryIntent, "SELECT IMAGE"),
GALLERY_PICK);

        }
    });



    mRefreshLayout.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {
        @Override
        public void onRefresh() {

            mCurrentPage++;
```

```
                itemPos = 0;

                loadMoreMessages();


            }
        });


    }


    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if(requestCode == GALLERY_PICK && resultCode == RESULT_OK){

            Uri imageUri = data.getData();

            final String current_user_ref = "messages/" + mCurrentUserId + "/" + mChatUser;
            final String chat_user_ref = "messages/" + mChatUser + "/" + mCurrentUserId;

            DatabaseReference user_message_push = mRootRef.child("messages")
                    .child(mCurrentUserId).child(mChatUser).push();

            final String push_id = user_message_push.getKey();


            StorageReference filepath = mImageStorage.child("message_images").child( push_id +
".jpg");

            filepath.putFile(imageUri).addOnCompleteListener(new
OnCompleteListener<UploadTask.TaskSnapshot>() {
                @Override
                public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {

                    if(task.isSuccessful()){

                        String download_url = task.getResult().getDownloadUrl().toString();


                        Map messageMap = new HashMap();
                        messageMap.put("message", download_url);
                        messageMap.put("seen", false);
                        messageMap.put("type", "image");
```

```
                messageMap.put("time", ServerValue.TIMESTAMP);
                messageMap.put("from", mCurrentUserId);

                Map messageUserMap = new HashMap();
                messageUserMap.put(current_user_ref + "/" + push_id, messageMap);
                messageUserMap.put(chat_user_ref + "/" + push_id, messageMap);

                mChatMessageView.setText("");

                mRootRef.updateChildren(messageUserMap, new
DatabaseReference.CompletionListener() {
                    @Override
                    public void onComplete(DatabaseError databaseError, DatabaseReference
databaseReference) {

                        if(databaseError != null){

                            Log.d("CHAT_LOG", databaseError.getMessage().toString());

                        }

                    }
                });


            }

        }
    });

  }

}

 private void loadMoreMessages() {

    DatabaseReference messageRef =
mRootRef.child("messages").child(mCurrentUserId).child(mChatUser);

    Query messageQuery = messageRef.orderByKey().endAt(mLastKey).limitToLast(10);

    messageQuery.addChildEventListener(new ChildEventListener() {
      @Override
      public void onChildAdded(DataSnapshot dataSnapshot, String s) {


        Messages message = dataSnapshot.getValue(Messages.class);
```

```
                String messageKey = dataSnapshot.getKey();

                if(!mPrevKey.equals(messageKey)){

                    messagesList.add(itemPos++, message);

                } else {

                    mPrevKey = mLastKey;

                }


                if(itemPos == 1) {

                    mLastKey = messageKey;

                }


                Log.d("TOTALKEYS", "Last Key : " + mLastKey + " | Prev Key : " + mPrevKey + "
| Message Key : " + messageKey);

                mAdapter.notifyDataSetChanged();

                mRefreshLayout.setRefreshing(false);

                mLinearLayout.scrollToPositionWithOffset(10, 0);

            }

            @Override
            public void onChildChanged(DataSnapshot dataSnapshot, String s) {

            }

            @Override
            public void onChildRemoved(DataSnapshot dataSnapshot) {

            }

            @Override
            public void onChildMoved(DataSnapshot dataSnapshot, String s) {

            }

            @Override
```

```
            public void onCancelled(DatabaseError databaseError) {

            }
        });

    }

    private void loadMessages() {

        DatabaseReference messageRef =
mRootRef.child("messages").child(mCurrentUserId).child(mChatUser);

        Query messageQuery = messageRef.limitToLast(mCurrentPage *
TOTAL_ITEMS_TO_LOAD);


        messageQuery.addChildEventListener(new ChildEventListener() {
            @Override
            public void onChildAdded(DataSnapshot dataSnapshot, String s) {

                Messages message = dataSnapshot.getValue(Messages.class);

                itemPos++;

                if(itemPos == 1){

                    String messageKey = dataSnapshot.getKey();

                    mLastKey = messageKey;
                    mPrevKey = messageKey;

                }

                messagesList.add(message);
                mAdapter.notifyDataSetChanged();

                mMessagesList.scrollToPosition(messagesList.size() - 1);

                mRefreshLayout.setRefreshing(false);

            }

            @Override
            public void onChildChanged(DataSnapshot dataSnapshot, String s) {

            }
```

```
        @Override
        public void onChildRemoved(DataSnapshot dataSnapshot) {

        }

        @Override
        public void onChildMoved(DataSnapshot dataSnapshot, String s) {

        }

        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });

}

private void sendMessage() {


    String message = mChatMessageView.getText().toString();

    if(!TextUtils.isEmpty(message)){

        String current_user_ref = "messages/" + mCurrentUserId + "/" + mChatUser;
        String chat_user_ref = "messages/" + mChatUser + "/" + mCurrentUserId;

        DatabaseReference user_message_push = mRootRef.child("messages")
                .child(mCurrentUserId).child(mChatUser).push();

        String push_id = user_message_push.getKey();

        Map messageMap = new HashMap();
        messageMap.put("message", message);
        messageMap.put("seen", false);
        messageMap.put("type", "text");
        messageMap.put("time", ServerValue.TIMESTAMP);
        messageMap.put("from", mCurrentUserId);

        Map messageUserMap = new HashMap();
        messageUserMap.put(current_user_ref + "/" + push_id, messageMap);
        messageUserMap.put(chat_user_ref + "/" + push_id, messageMap);

        mChatMessageView.setText("");
```

```
mRootRef.child("Chat").child(mCurrentUserId).child(mChatUser).child("seen").setValue(true)
;

mRootRef.child("Chat").child(mCurrentUserId).child(mChatUser).child("timestamp").setValu
e(ServerValue.TIMESTAMP);


mRootRef.child("Chat").child(mChatUser).child(mCurrentUserId).child("seen").setValue(false
);

mRootRef.child("Chat").child(mChatUser).child(mCurrentUserId).child("timestamp").setValu
e(ServerValue.TIMESTAMP);

        mRootRef.updateChildren(messageUserMap, new
DatabaseReference.CompletionListener() {
            @Override
            public void onComplete(DatabaseError databaseError, DatabaseReference
databaseReference) {

                if(databaseError != null){

                    Log.d("CHAT_LOG", databaseError.getMessage().toString());

                }

            }
        });

    }

}
}
```

- **Firebase Messaging Service.java**

```
package com.example.bitzone;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.content.Intent;

import android.support.v4.app.NotificationCompat;

import com.google.firebase.messaging.RemoteMessage;

public class FirebaseMessagingService extends com.google.firebase.messaging.FirebaseMessagingService {
```

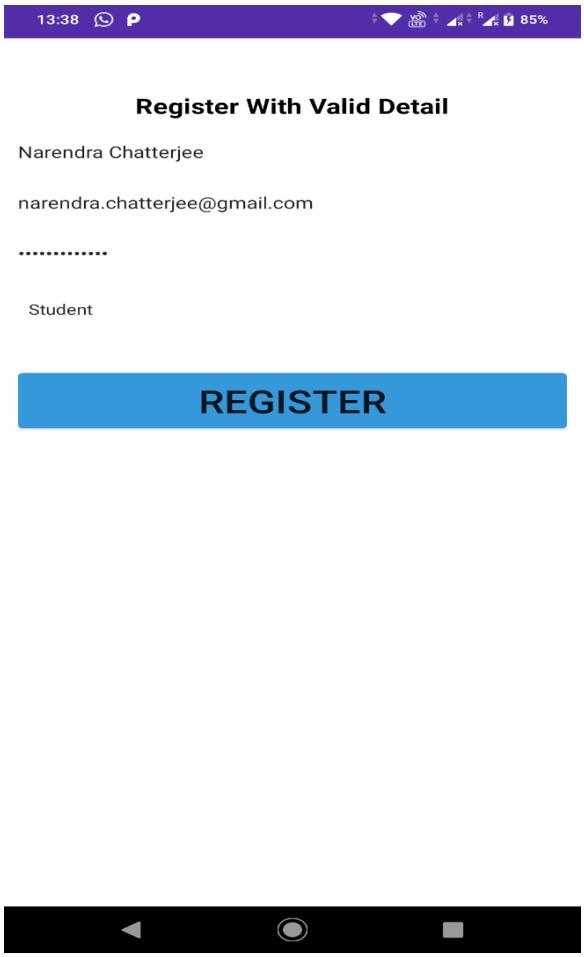
```
    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        super.onMessageReceived(remoteMessage);
        String notification_title = remoteMessage.getNotification().getTitle();
        String notification_message = remoteMessage.getNotification().getBody();
        String click_action = remoteMessage.getNotification().getClickAction();
        String from_user_id = remoteMessage.getData().get("from_user_id");
        NotificationCompat.Builder mBuilder =
                new NotificationCompat.Builder(this)
                        .setSmallIcon(R.mipmap.ic_launcher)
                        .setContentTitle(notification_title)
                        .setContentText(notification_message);
        Intent resultIntent = new Intent(click_action);
        resultIntent.putExtra("user_id", from_user_id);
        PendingIntent resultPendingIntent =
                PendingIntent.getActivity(
                        this,
                        0,
                        resultIntent,
                        PendingIntent.FLAG_UPDATE_CURRENT
                );
        mBuilder.setContentIntent(resultPendingIntent);
        int mNotificationId = (int) System.currentTimeMillis();

        NotificationManager mNotifyMgr =
                (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
        mNotifyMgr.notify(mNotificationId, mBuilder.build());



    }
}
```

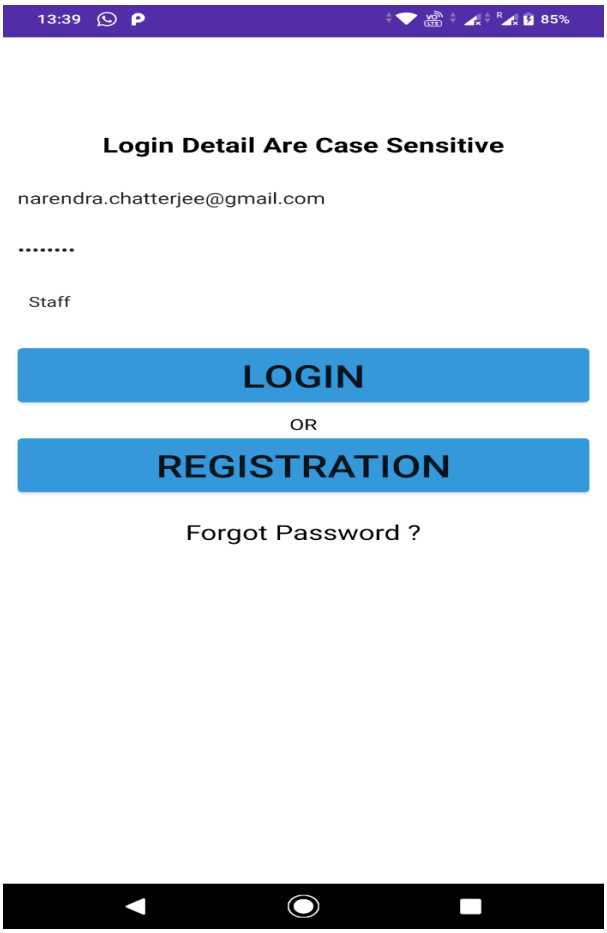
CHAPTER 6

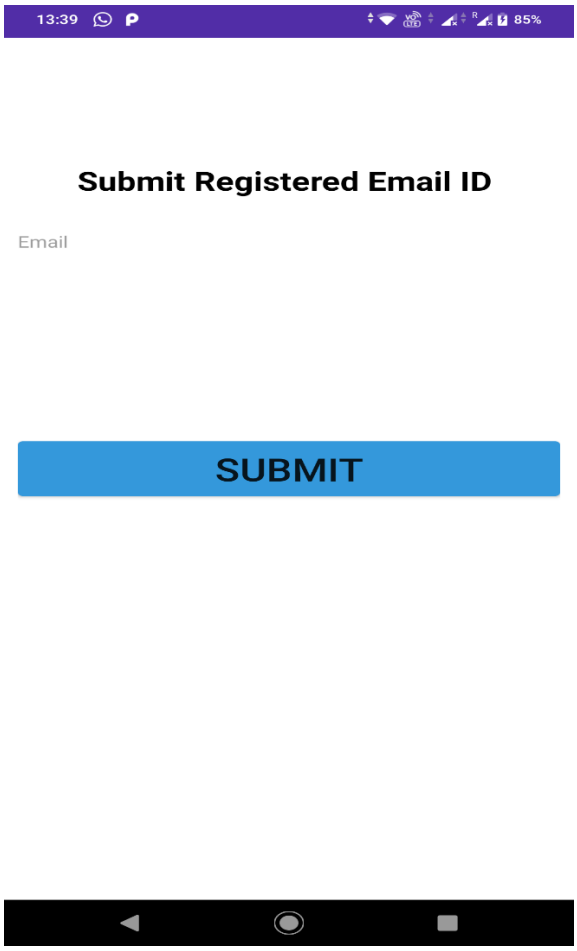
SCREENSHOTS

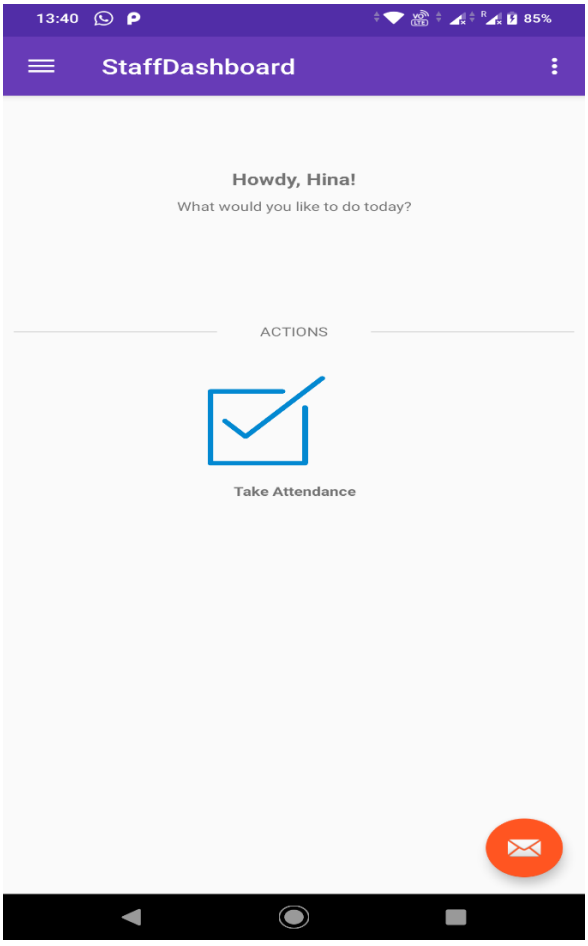

BIRLA INSTITUTE OF TECHNOLOGY
RANCHI
BitZone

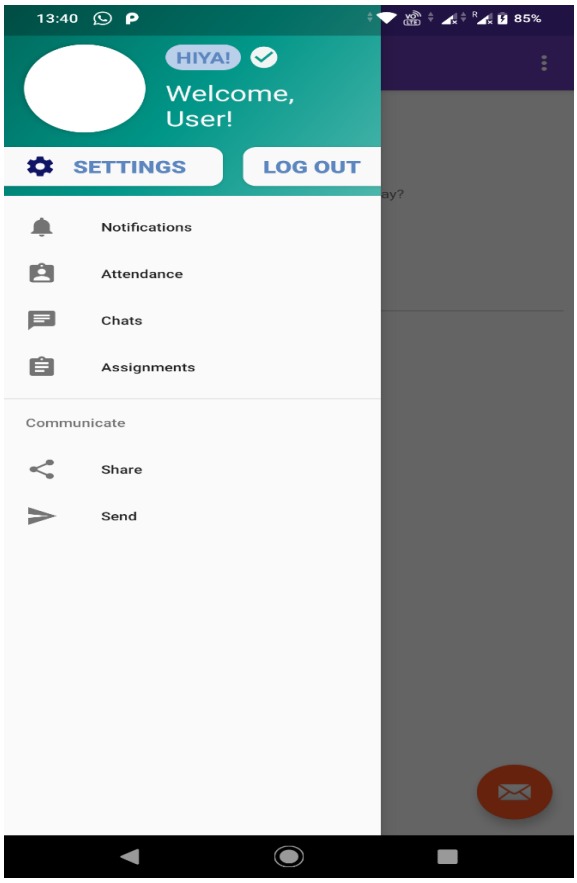

13:38
85%
Welcome to BITZONE!
JOIN
LOG IN


13:38
85%
Register With Valid Detail
Narendra Chatterjee
narendra.chatterjee@gmail.com
Student
REGISTER

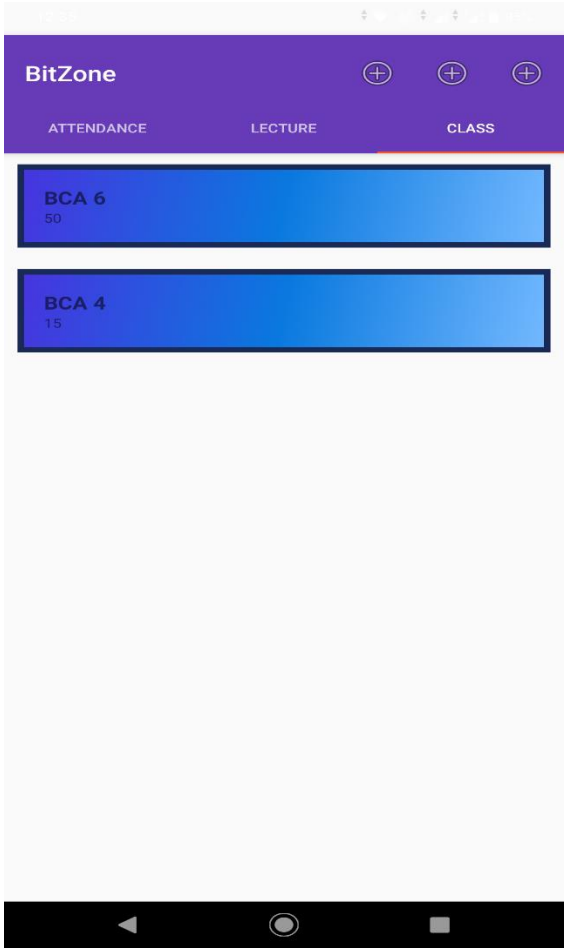

13:39
85%
Login Detail Are Case Sensitive
narendra.chatterjee@gmail.com
Staff
LOGIN
OR
REGISTRATION
Forgot Password ?

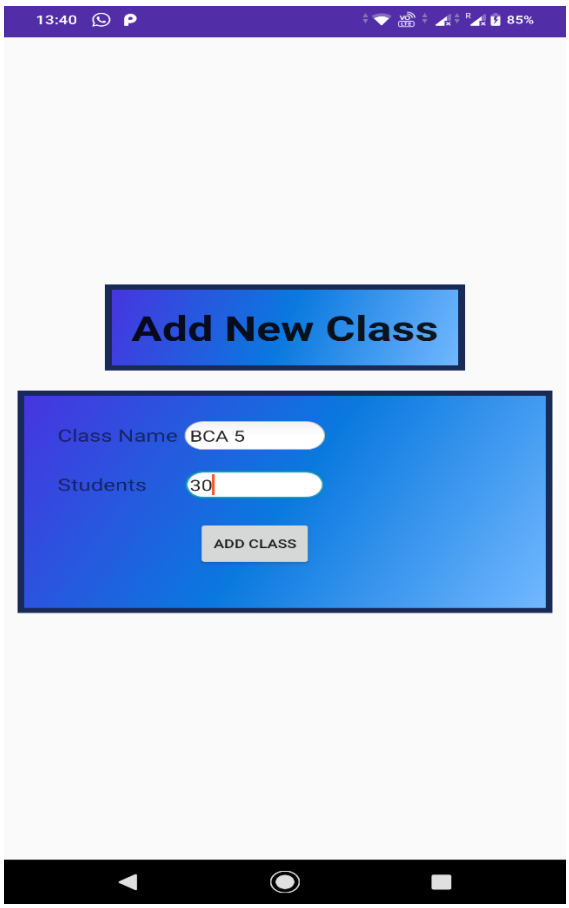

13:39
85%
Submit Registered Email ID
Email
SUBMIT

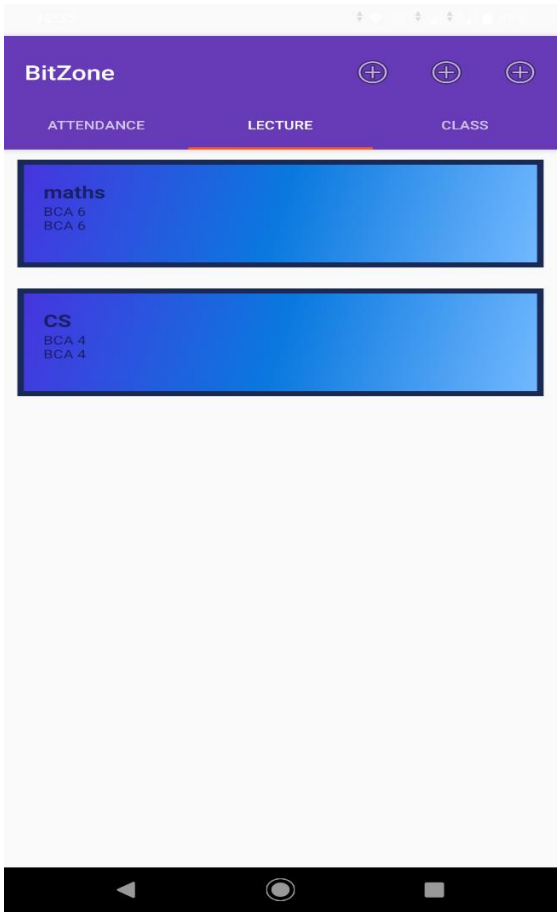

13:40
85%
StaffDashboard
Howdy, Hina!
What would you like to do today?
ACTIONS
Take Attendance

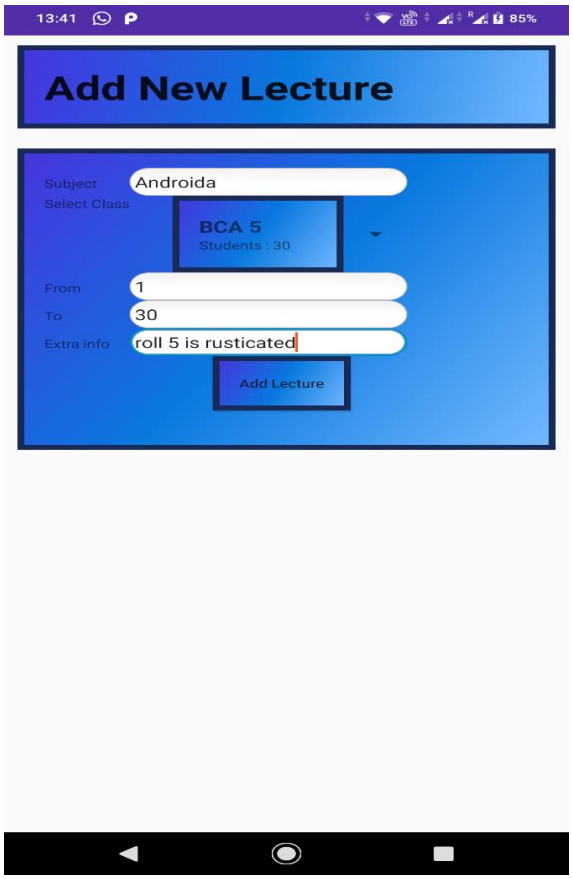

13:40
85%
HIYA!
Welcome,
User!
SETTINGS
LOG OUT
Notifications
Attendance
Chats
Assignments
Communicate
Share
Send

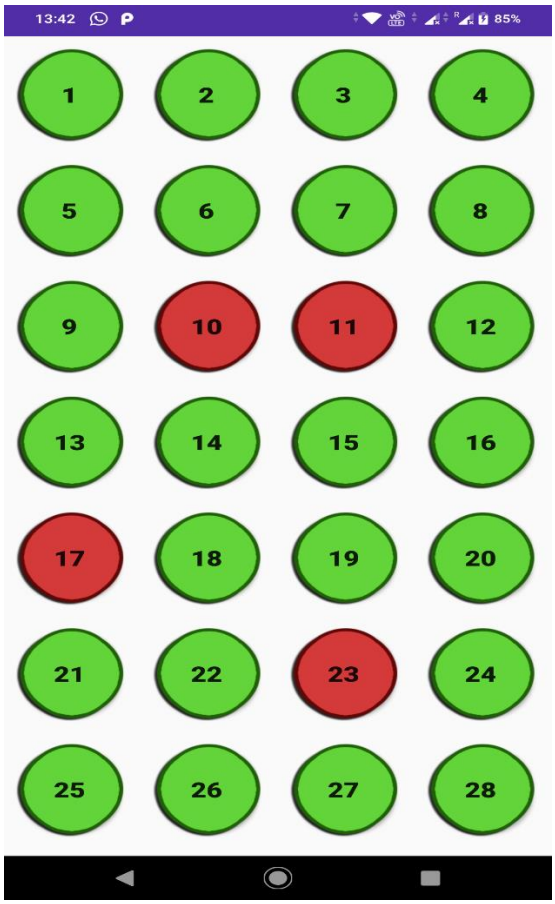

BitZone
ATTENDANCE
LECTURE
CLASS
CS
BCA 4
No of students 5
Date : 05/07/2019 12:07:22
maths
BCA 6
No of students 31
Date : 05/04/2019 12:19:03
maths
BCA 6
No of students 31
Date : 05/03/2019 23:27:50
maths
BCA 6
No of students 31
Date : 05/03/2019 23:27:06
maths
BCA 6
No of students 31
Date : 05/03/2019 23:27:00


BitZone
ATTENDANCE
LECTURE
CLASS
BCA 6
50
BCA 4
15

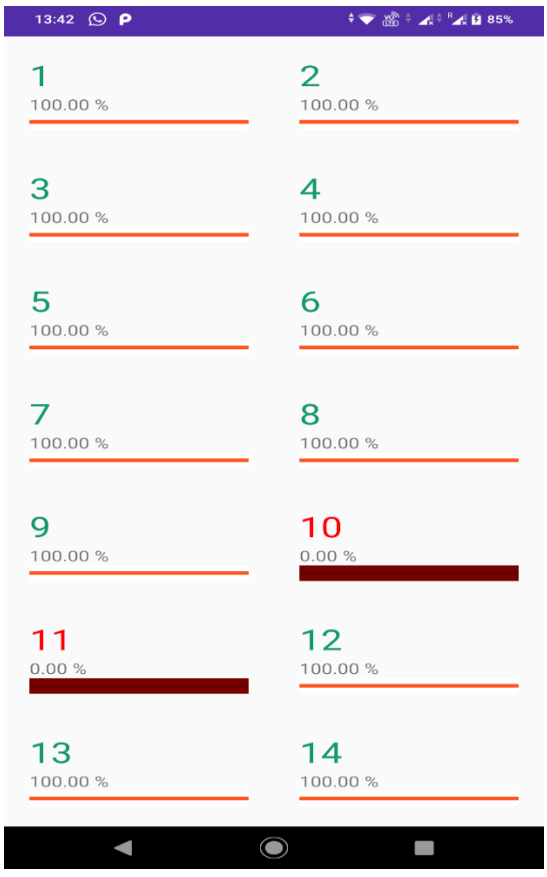

13:40
85%
Add New Class
Class Name
BCA 5
Students
30
ADD CLASS

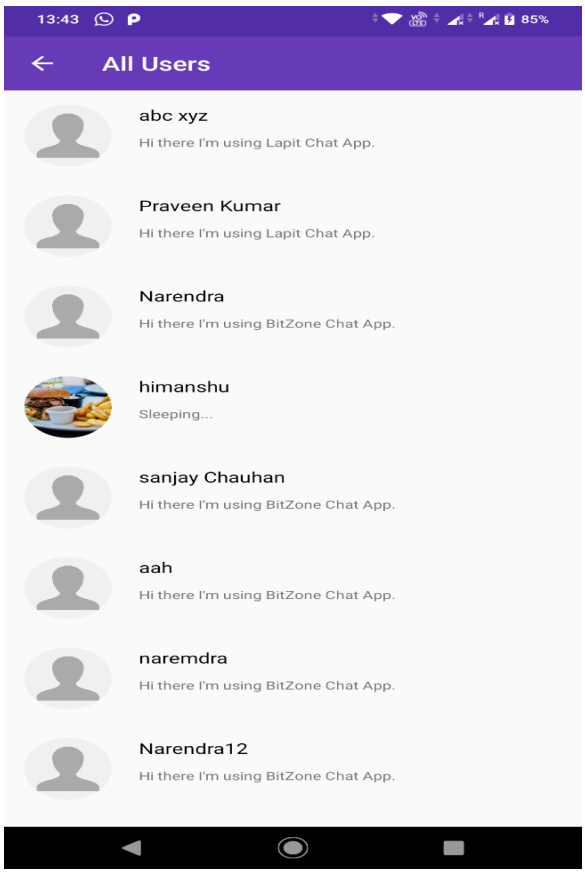

BitZone
ATTENDANCE
LECTURE
CLASS
maths
BCA 6
BCA 6
CS
BCA 4
BCA 4

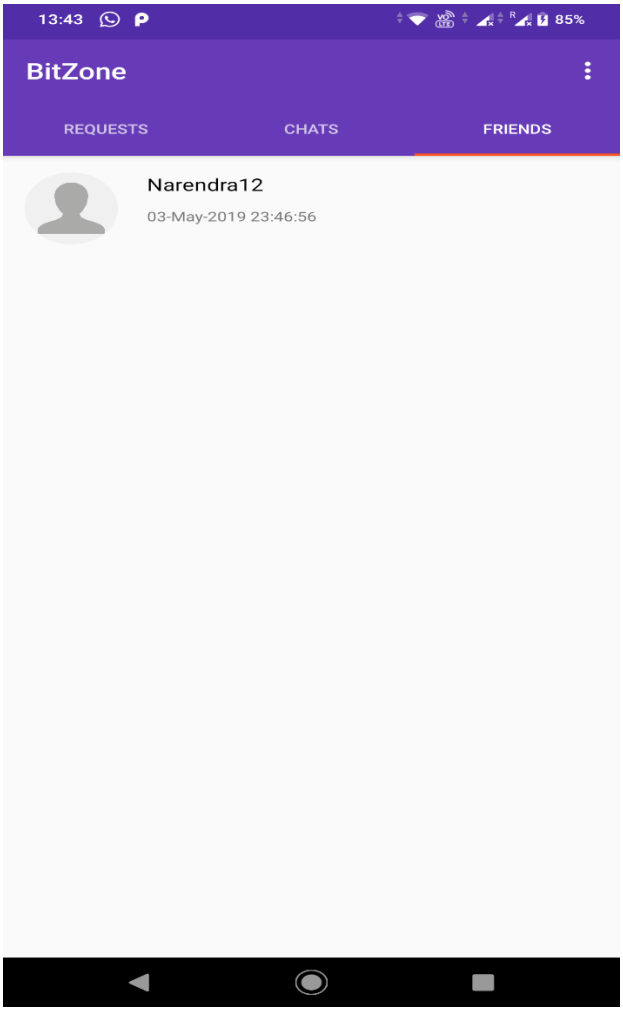

13:41
85%
Add New Lecture
Subject
Androida
Select Class
BCA 5
Students : 30
From
1
To
30
Extra info
roll 5 is rusticated
Add Lecture

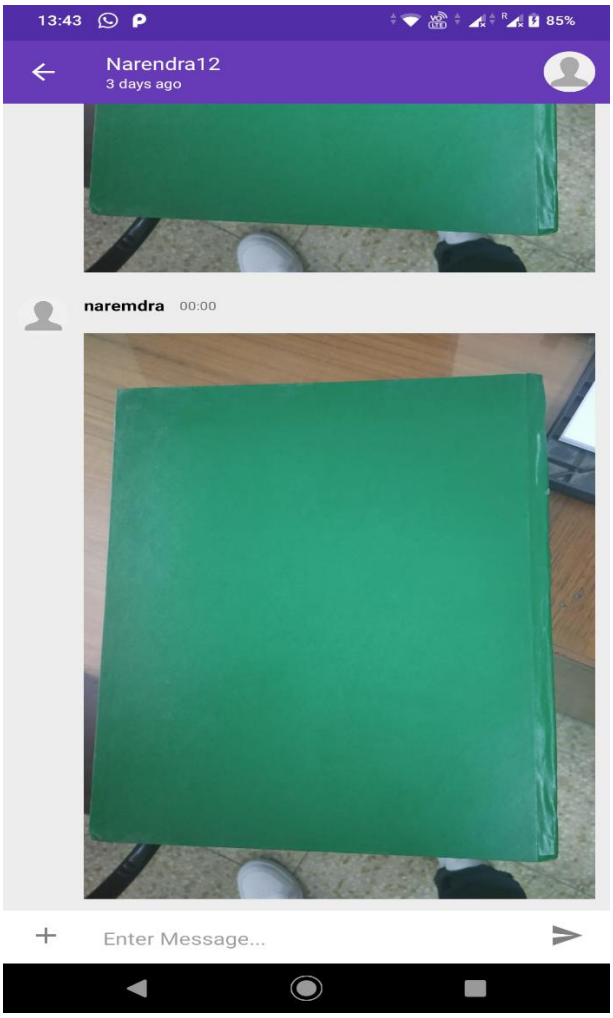

13:42
85%
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

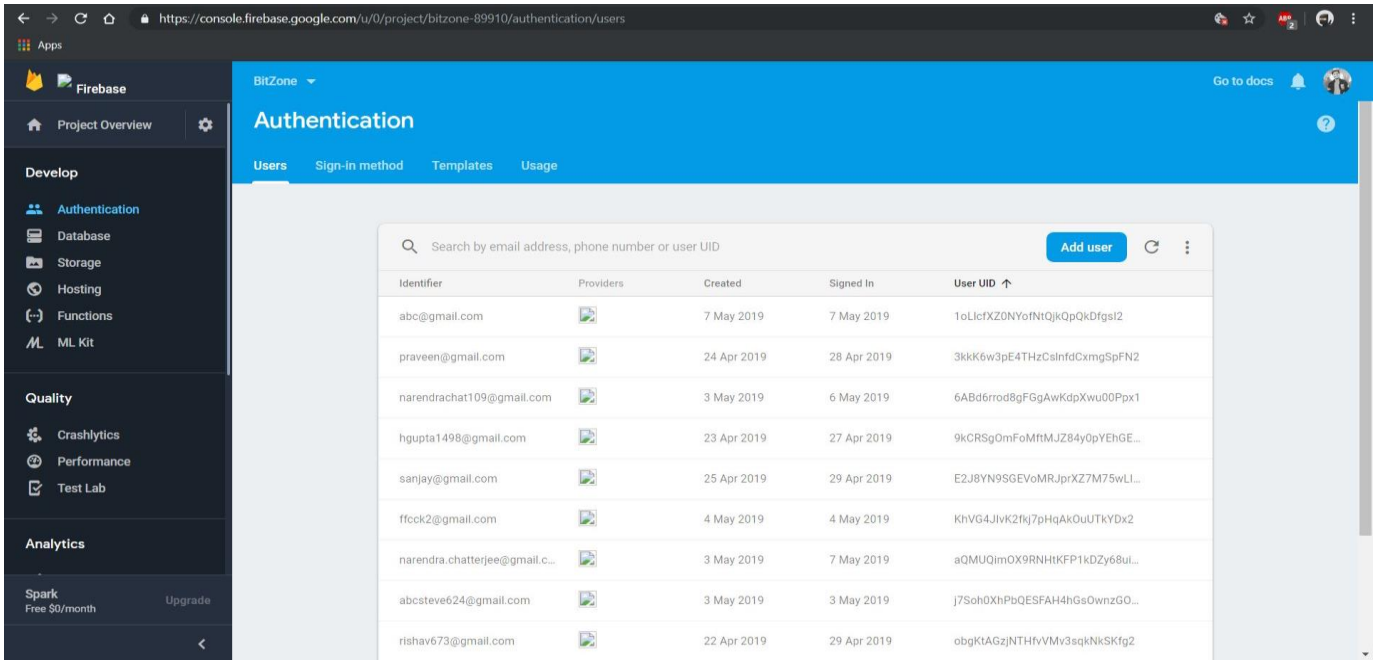

13:42
85%
Absents
10,11,17,23,
Presents
1,2,3,4,5,6,7,8,9,12,13,14,15,16,18,19,20
,21,22,24,25,26,27,28,29,30,

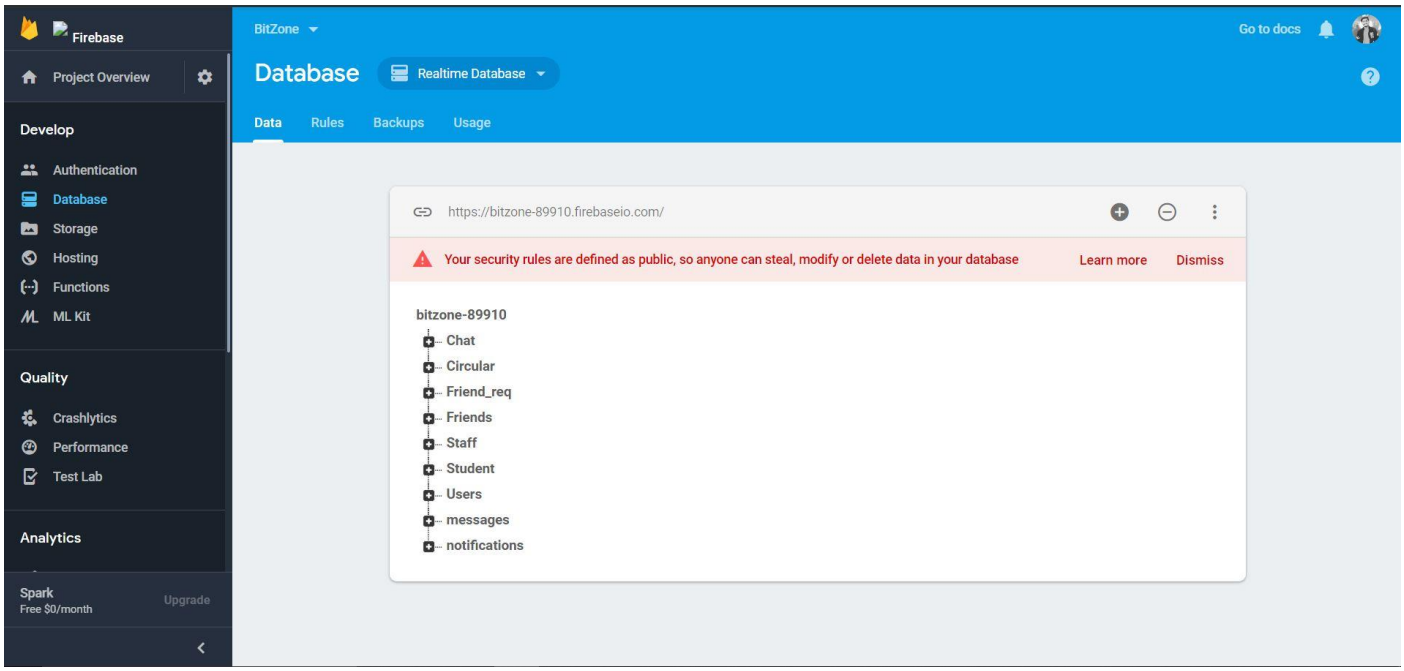

13:42
85%
1
100.00 %
2
100.00 %
3
100.00 %
4
100.00 %
5
100.00 %
6
100.00 %
7
100.00 %
8
100.00 %
9
100.00 %
10
0.00 %
11
0.00 %
12
100.00 %
13
100.00 %
14
100.00 %

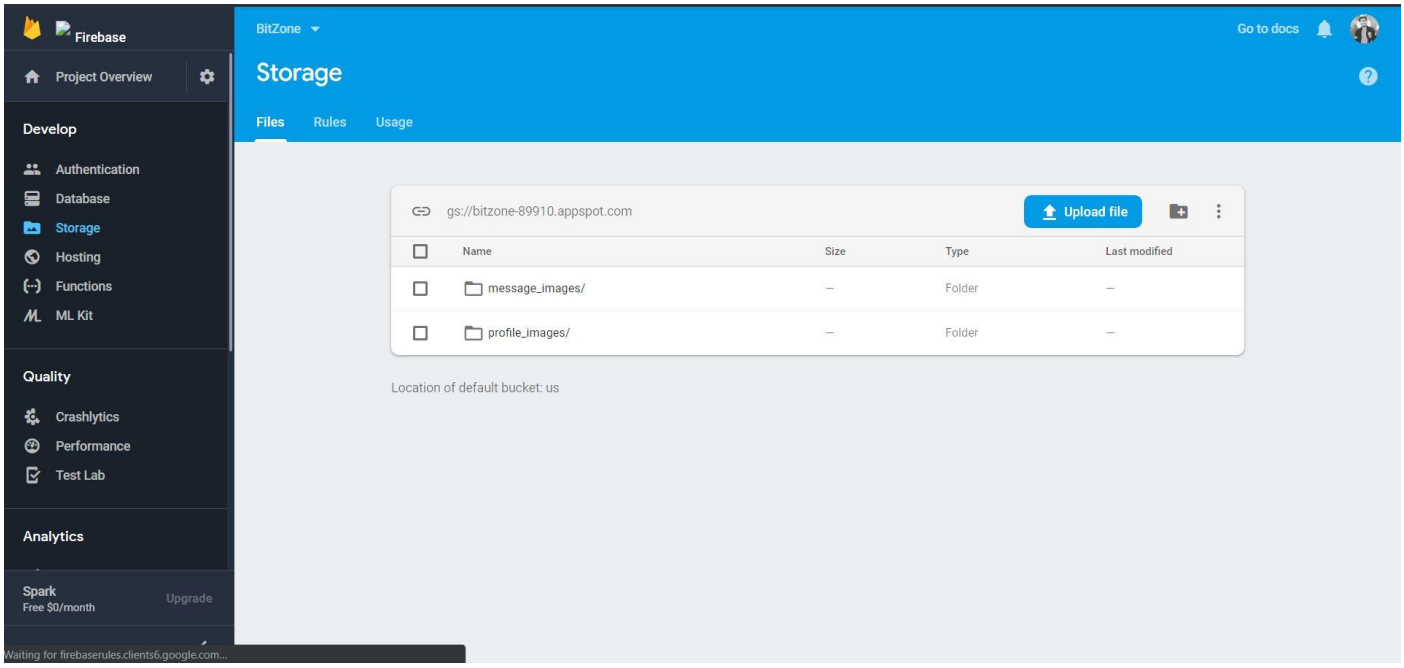

13:43
85%
All Users
abc xyz
Hi there I'm using Lapit Chat App.
Praveen Kumar
Hi there I'm using Lapit Chat App.
Narendra
Hi there I'm using BitZone Chat App.
himanshu
Sleeping...
sanjay Chauhan
Hi there I'm using BitZone Chat App.
aah
Hi there I'm using BitZone Chat App.
naremdra
Hi there I'm using BitZone Chat App.
Narendra12
Hi there I'm using BitZone Chat App.


13:43
85%
BitZone
REQUESTS
CHATS
FRIENDS
Narendra12
03-May-2019 23:46:56


13:43
85%
Narendra12
3 days ago
naremdra 00:00
Enter Message...


https://console.firebase.google.com/u/0/project/bitzone-89910/authentication/users
Apps
Firebase
Project Overview
Develop
Authentication
Database
Storage
Hosting
Functions
ML Kit
Quality
Crashlytics
Performance
Test Lab
Analytics
Spark
Free $0/month
Upgrade
BitZone
Go to docs
Authentication
Users
Sign-in method
Templates
Usage
Search by email address, phone number or user UID
Add user
Identifier
Providers
Created
Signed In
User UID
abc@gmail.com
7 May 2019
7 May 2019
1oLlcfXZ0NYofNtQjkQpQkDfgsI2
praveen@gmail.com
24 Apr 2019
28 Apr 2019
3kkK6w3pE4THzCslnfdCxmgSpFN2
narendrachat109@gmail.com
3 May 2019
6 May 2019
6ABd6rrod8gFGgAwKdpXwu00Ppx1
hgupta1498@gmail.com
23 Apr 2019
27 Apr 2019
9kCRSgOmFoMftMJZ84y0pYEhGE...
sanjay@gmail.com
25 Apr 2019
29 Apr 2019
E2J8YN9SGEVoMRJprXZ7M75wLI...
ffcck2@gmail.com
4 May 2019
4 May 2019
KhVG4JIvK2fkj7pHqAkOuUTkYDx2
narendra.chatterjee@gmail.c...
3 May 2019
7 May 2019
aQMUQimOX9RNHtKFP1kDZy68ui...
abcsteve624@gmail.com
3 May 2019
3 May 2019
j7Soh0XhPbQESFAH4hGsOwnzGO...
rishav673@gmail.com
22 Apr 2019
29 Apr 2019
obgKtAGzjNTHfvVMv3sqkNkSKfg2


Firebase
Project Overview
Develop
Authentication
Database
Storage
Hosting
Functions
ML Kit
Quality
Crashlytics
Performance
Test Lab
Analytics
Spark
Free $0/month
Upgrade
BitZone
Go to docs
Database
Realtime Database
Data
Rules
Backups
Usage
https://bitzone-89910.firebaseio.com/
Your security rules are defined as public, so anyone can steal, modify or delete data in your database
Learn more
Dismiss
bitzone-89910
Chat
Circular
Friend_req
Friends
Staff
Student
Users
messages
notifications


Firebase
Project Overview
Develop
Authentication
Database
Storage
Hosting
Functions
ML Kit
Quality
Crashlytics
Performance
Test Lab
Analytics
Spark
Free $0/month
Upgrade
Waiting for firebaserules.clients6.google.com...
BitZone
Go to docs
Storage
Files
Rules
Usage
gs://bitzone-89910.appspot.com
Upload file
Name
Size
Type
Last modified
message_images/
Folder
profile_images/
Folder
Location of default bucket: us

CHAPTER 7
TESTING

7.1 Testing

Testing is the only phase in the whole software development process that is regarded as a destructive process rather than a constructive one. During the testing phase, the engineer creates a series of test cases that are intended demolish the software that has been built. The basic testing principles that a software engineer must understand before applying methods to design effective test cases are as under:

1. All test should be traceable to customer requirement.

2. Tests should be planned long before testing begins.

3. Testing should begin in the small and progress towards testing in the large.

4. To be most effective, an independent third party should conduct testing. There are mainly two fundamental approaches to testing. There are:

7.1.1 Black box testing

Which checks for the fundamental aspects of the system with little regard for the internal logic structure of software? Black box tests are used to demonstrate that software functions are operational, that input is properly accepted and output is properly produced, and that the integrity of external information is maintained.

7.1.2 White box testing

White box testing is that testing which checks for the correctness of the procedural details at the grass root level of the code. White box testing is used to ensure that internal operations are performed according to specifications and all internal components have been adequately exercised. Whit box testing of software is predicted on close examination of procedural detail. Providing test cases, which exercise specific sets of conditions and or loops, tests logical paths through the software?

CHAPTER 8
SCOPE AND CONCLUSION

8.1 SCOPE

This project is aimed at how the institute can improve the efficiency of the services. BITZONE is one of the application to improve the interaction between a student and a teacher. This application involves almost all the features of the information system; the future implementations and upgradations would be done once the application is live.

8.2 CONCLUSION

Our project is only a humble venture to satisfy the needs in an Institution. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the organization.

The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

CHAPTER 9
REFERENCES

Bibliography

During the development of our system, I have taken the reference the Books, Internet, journals and some video lectures which I would like to mention in this section.

These books acted as my tutors during the system development.

1. Android Programming: The Big Nerd Ranch Guide
2. Head First Android Development: A Brain-Friendly Guide

I took some references from the following webpages:

- https://www.tutorialspoint.com/android/
- https://www.javatpoint.com/android-tutorial
- https://www.androidtutorialpoint.com/

Besides this I also took references from the following YouTube Channels:

- Channel name: Programming Knowledge
  https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E

- Channel name: The New Boston
  https://www.youtube.com/playlist?list=PL6gx4Cwl9DGBsvRxJJOzG4r4k_zLKrnxl