

A MINI PROJECT REPORT

On

Attendance based Android Application

Submitted by

**Gunjika
Roll No:178420003**

**Shweta Agarwal
Roll No:188429047**

**Raksha Goswami
Roll No:188429037**

**Kishor Chaturvedi
Roll No:188429024**

**Department of Computer Engineering & Applications
Institute of Engineering & Technology**



**GLA University
Mathura- 281406, INDIA
May, 2019**



Department of Computer Engineering and Applications

GLA University, Mathura

17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,
Mathura – 281406

Declaration

We hereby declare that the work which is being presented in the Mini Project "**Attendance based Android Application**", in partial fulfillment of the requirements for Mini-Project LAB, is an authentic record of our own work carried under the supervision of **Mr. Divyansh Bhardwaj, Technical Trainer, GLA University, Mathura.**

Gunjika
(178420003)

Shweta Agarwal
(188429047)

Raksha Goswami
(188429037)

Kishor Chaturvedi
(188429024)



Department of Computer Engineering and Applications
GLA University, Mathura

17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,
Mathura – 281406

CERTIFICATE

This is to certify that the project entitled "**Attendance based Android Application**" carried out in Mini Project – II Lab is a bonafide work done by **Gunjika (178420003)** **Raksha Goswami (188429037)**, **Shweta Agarwal (188429047)** and **Kishor Chaturvedi (1884290)** and is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

Signature with date

Name of Supervisor: Mr. Divyansh Bhardwaj

Signature with date

Mr. Rakesh Kumar

(Mini Project Coordinator)

Date:

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the MCA Mini Project undertaken during MCA Second Year. This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by many individuals. This project would never have seen the light of the day without the help and guidance that we have received.

*Our heartiest thanks to **Prof. (Dr.) Anand Singh Jalal**, Head of Dept., Department of CEA for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.*

*We owe special debt of gratitude to **Mr. Divyansh Bhardwaj**, Technical Trainer Department of CEA, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. He has showered us with all his extensively experienced ideas and insightful comments at virtually all stages of the project & has also taught us about the latest industry-oriented technologies.*

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Gunjika

Raksha Goswami

Shweta Agarwal

Kishor Chaturvedi

Abstract

The main objective of this paper is to carry out some of the emerging technologies like mobile computing, Information and Communication Technology and advances in behavioral science studies to enhance and enriched the current educational system scenario, since the educational technology in India has been get modernizing in the recent past years due to the development and penetration of Information and Technology. The mobile based technology has been chosen for this study as well as for the project. The methodology of this work is to developed an android based mobile application attendance management system where attendance can be recorded via mobile devices using “java” and “Firebase”. This developed software stores, retrieves and deliver the information about the student information such as present or absent through mobile device in the provided server database. The final result of this project are very useful for the educational organizations to keep, track and maintain the database of students

Table of Contents

Declaration	ii
Certificate	iii
Acknowledgments	iv
Abstract	V
Table of Contents	Vi
1. Introduction (This chapter must describe introduction about your project)	1
1.1 Motivation and Overview	1
1.3 Objective	2
2. Software Requirement Analysis	7
2.1 Define the problem	7
2.2 Define the modules and their functionalities (SRS)	7
3. Software Design	16
3.1	16
3.2	17
4. Testing	22
4.1	22
4.2	22
5. Implementation and User Interface	27
5.1	27
5.2	27
References/Bibliography	28
6. Appendices	

CHAPTER NO 1
INTRODUCTION

Nowadays, mobile devices have become a way of life for students especially in higher education. Computers are now replaced by compact smart phones that can fit into pocket and can be carried anywhere. Attendance Management System is an application for taking daily attendance in schools and colleges. It provides minimal error in report generation of a particular student's attendance. The main motive behind this software is to reduce proxy. Another reason for developing this software is to generate the report automatically at the end of the session or in the between of the session. Smart phones are based on operating systems like blackberry, IOS and Android. To design this project, smart phones with Android operating system are chosen because penetration rate of Android OS is 75 percent. It is an open source and free ware operating system. The application is compatible with all Android versions starting from 4.2 Jelly Bean.

Android-based attendance application is an application available for daily student attendance in colleges and institutes. It facilitates to access the attendance information of a particular student in a particular class. The information is sorted by the operators, which will be provided by the teacher for a particular class. This system will also help in evaluating attendance eligibility criteria of a student. Everyone can access the system by providing there valid id, password, and fingerprint. They can access modules such as viewing their own profile, student's profiles, view time table, attendance of students, and notification of absence.

In this application there are two types of user one is a student's application and another one is the teacher's application. In teacher's module teacher will make a connection using hotspot and student will join this connection. Those students who join this connection, attendance of that student will mark automatically. Teacher gets a notification about those students who do not present in the class and student gets a notification about whom attendance will not mark by mistake and teacher have some time to modify the attendance of that student.

OVERVIEW & MOTIVATION

1.1 OVERVIEW

Android-based attendance application is an application available for daily student attendance in colleges and institutes. It facilitates to access the attendance information of a particular student in a particular class. The information is sorted by the operators, which will be provided by the teacher for a particular class. This system will also help in evaluating attendance eligibility criteria of a student. Everyone can access the system by providing their valid id, password, and fingerprint. They can access modules such as viewing their own profile, student's profiles, view time table, attendance of students, and notification of absence.

In this application there are two types of user one is a student and another one is the teacher. In teacher's module teacher will make a connection using hotspot and student will join this connection. Those students who join this connection, attendance of that student will mark automatically. Teacher gets a notification about those students who do not present in the class and student gets a notification about whom attendance will not mark by mistake and teacher have some time to modify the attendance of that student.

1.2 MOTIVATION

- The attendance system is one of the most important system used in every organization to keep the track of attendance
- The previous attendance system used in colleges needed the faculty to give the attendance details to be uploaded in the server.

- We got the motivation to make this project from college and also from already given attendance management system.
- This whole project was based on this concept, that application provides teachers to take attendance easily and save their operational time.

1.3 OBJECTIVE

Design a Student attendance system to keep track of attendance in mobile devices for easy and proper evaluation of attendance.

- To developed and design the android-based mobile attendance application for the management of attendance records in the educational organization.
- To create an Android mobile application to provide a User Interface to interact with the system
- To implement the new technology development system to make it digitalized ,authorized, secured one in the given web server to kept it records as many years we want to kept it for its future use as per our need.
- To Increase security.

CHAPTER NO 2
SOFTWARE REQUIREMENT ANALYSIS**2.1 DEFINE THE PROBLEM**

In making this application first we have to make, we make a plan to develop this application. we create Wi-Fi through P2P tool of android studio. The option what we want to put in the application such as student name, id, branch, timetable, profile, attendance, teacher name, id, department profile. We use spinner for list and then we create a new project and new activity for the application. In application teacher take attendance through Wi-Fi and student connected Wi-Fi automatically and get present .if any student marked absent so they got a notification and teacher modify the attendance.

2.2 Define the modules and their functionalities (SRS)**2.2.1 Methodology**

Our project on Attendance Management System is based on some concepts and methodologies that we devised during our initial study and visualization of the project. We have planned our project keeping in mind the Object Oriented Concepts that can be applied to the project. The project is strongly supported at the back end by Java-the most popular and successful object oriented programming language. The front end is developed using Extensible Mark-up Language (XML). For the database connectivity we are using Fair base database. It is a light weight database which is pre-installed with every android studio and is capable of handling and managing various queries, cursors and more.

In the web based module is for the admin panel to control its all over activates like updating, creation, deletion or any other database changes in the whole project by the admin and it's also contains a teacher's registration phase on website only through which they can generate a user id and password to login in the mobile based application which will be connected through this admin panel server. on the other hand, the mobile based application will be totally used by the teacher to get logged in and by choosing their authorized register class id and the subject id they are able to display the digital register book on their smart mobiles onwards and submit it finally by cross checking it to the main authorized provided admin server this process we designed a

database for the application. It contains various tables like register, subjects, students and attendance for various purposes. These tables hold data for particular purposes which shall be dealt later.

2.2.2 Purpose

The main characteristics of my developed system is that it is web based, fully responsive and flexible. It can be accessed from any computer no matter where you are. Its purpose is to make a web based attendance software for ICIT department to register the student details; their subjects, teachers, and related field. The daily attendance of students is taken automatically by selecting student name and program, If the student was present then the present check box is clicked similarly if the student was absent then the absent check Box is clicked instead of the present check box, and by clicking the save button information will be stored in database. The attendance report will be generated automatically without time consuming which is reliable and there will be not any mistakes.

2.2.3 Scope

The system can be further enhanced and several other functionalities can be added. The system can be made login independent. The present system logs in using Internet all the time. We can enhance the system by implementing offline mode login feature. The feature to update the attendance at a later stage can also be implemented. The system can also be enhanced by using voice recognition feature of the Android.

2.2.4 MATERIALS

Hardware:

- An Android Smartphone with Wi-Fi capabilities.
- RAM: 1 GB or above

Software:

- Android Based OS
- Min. Version: Android Marshmallow 5.0.1

2.2.5 TOOLS USED

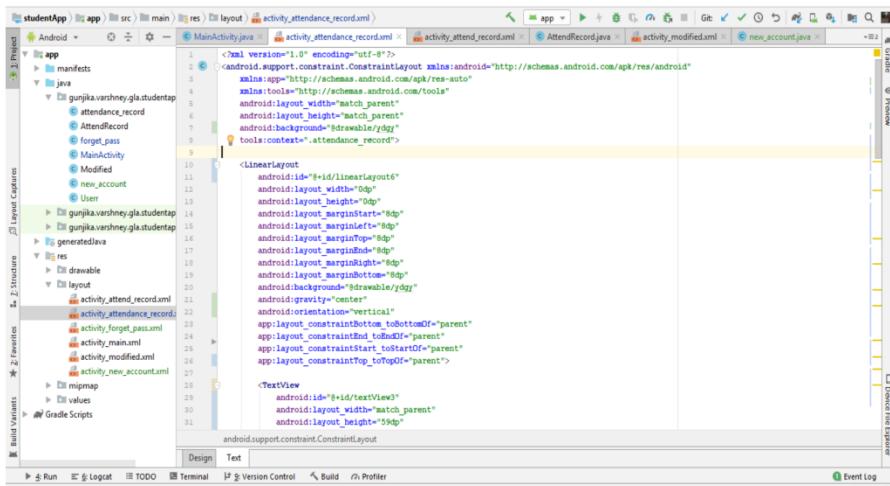
Firebase:

The Assistant tool window in **Android Studio**. **Firebase** is a mobile platform that helps you quickly develop high-quality apps, grow your user base, and earn more money. **Firebase** is made up of complementary features that you can mix-and-match to fit your needs, with Google Analytics for **Firebase** at the core



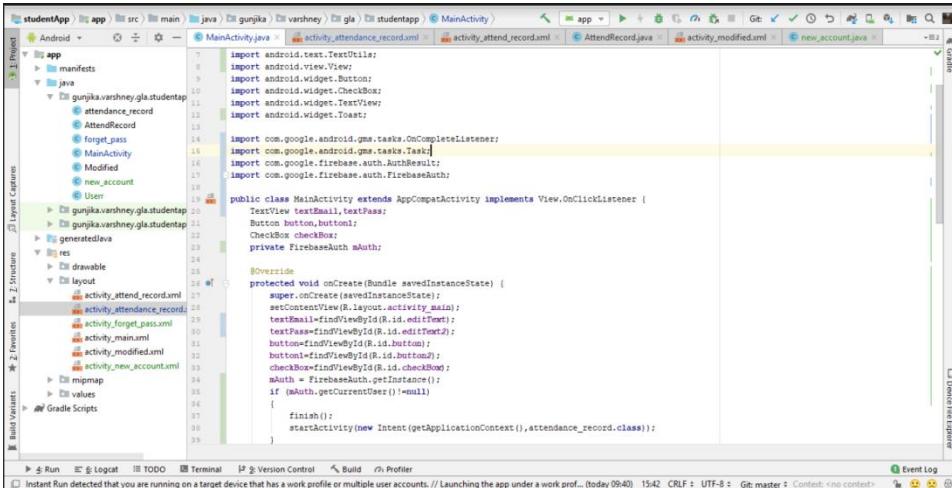
XML:

XML stands for Extensible Markup Language. **XML** is a markup language much like **HTML** used to describe data. ... In this article we will go through the basic concepts of **xml in Android** and different **XML** files used for different purpose in **Android**. This will help you in writing a UI code to design your desired user interface.



JAVA:

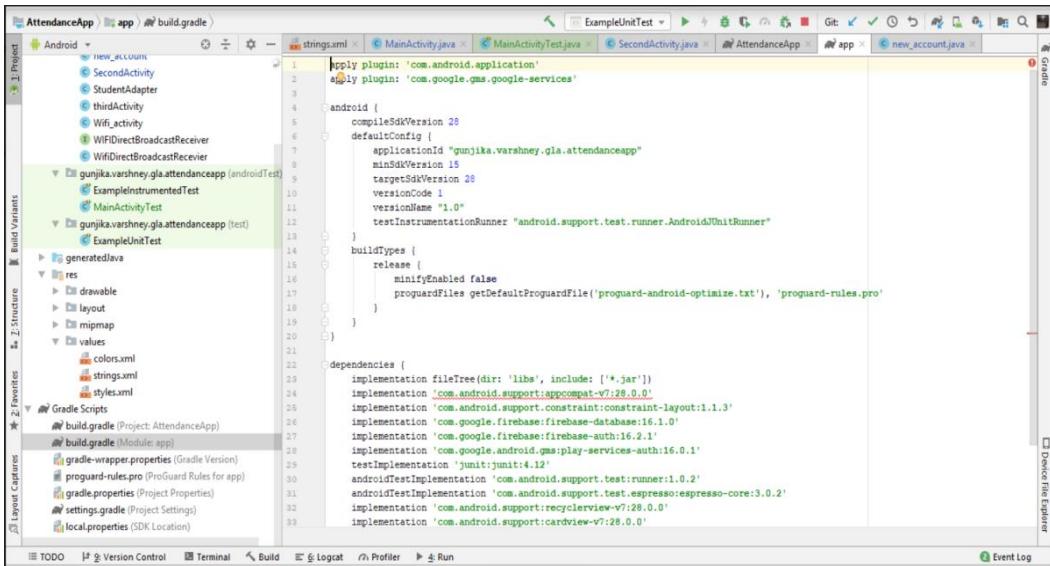
Android Studio is the official integrated development environment (IDE) for Google's **Android** operating system, built on Jet Brains' IntelliJ IDEA software and designed specifically for **Android** development. It is available for download on Windows, macos and Linux based operating systems.



GRADLE:

In **Android Studio**, **Gradle** is a custom build tool used to build **android** packages (apk files) by managing dependencies and providing custom build logic. APK file

(Android Application package) is a specially formatted zip file which contains. Byte code. Resources (images, UI, xml etc.)



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

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "gunjika.varshney.gla.attendanceapp"
        minSdkVersion 15
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    implementation 'com.google.firebaseio:firebase-database:16.1.0'
    implementation 'com.google.firebaseio:firebase-auth:16.2.1'
    implementation 'com.google.android.gms:play-services-auth:16.0.1'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'com.android.support:recyclerview-v7:28.0.0'
    implementation 'com.android.support:cardview-v7:28.0.0'
}
```

2.2.6 MODULE

- **STUDENT MODULE**
- **TEACHER MODULE**

Student Module:

Student Module is new version of server based attendance management system. In this we use Wi-Fi to take attendance. In the student module, Student have a unique id and password for login. After login student can access the attendance time table and his profile. If any student marked absent mistakenly so student get a notification of his absent and he apply to modify the attendance.

Teacher Module:

SOFTWARE REQUIREMENT ANALYSIS CHAPTER TITLE

Teacher module is a nice way to take attendance. In this teacher did not upload the attendance. Attendance will automatically have uploaded through Wi-Fi; teacher use Wi-Fi to take attendance in which student automatically connected with teacher module and marked present. Teacher can add branch, add student, add time table, add attendance. This is easy way to take attendance.

CHAPTER NO 3
SOFTWARE DESIGN

The design phase emphasizes on the transformation of customer requirements as defined in the SRS document, into a form that is suitable for coding.

The design phase can be broadly classified in two levels.

- Preliminary or high level design
- Detailed design
- The preliminary design can be further divided into two sub categories
- Function Oriented Software Design
- Object Oriented Software Design

3.1 FUNCTION ORIENTED SOFTWARE DESIGN

This design model can be represented by drawing the DFDs (Data Flow Diagrams) for the given SRS document. A data flow diagram is a graphical representation of the data flow through an information system which is used to model the process aspects of the system. DFD is the preliminary step used to create an overview of the system. DFD is used for structured design.

3.1.1 LEVEL 0 DFD DAIGRAM

The context-level DFD is then exploded to produce a Level 0 DFD which models the details of the system. The Level 0 DFD shows how the system is divided into sub-systems (processes), and how each processes deals with one or more of the data flows to or from an external entity, and how the processes together provide all of the functionality of the system. The level 0 DFD also identifies the internal data stores which must be there for the system to do its job, and shows the data flow between the various parts of the system.

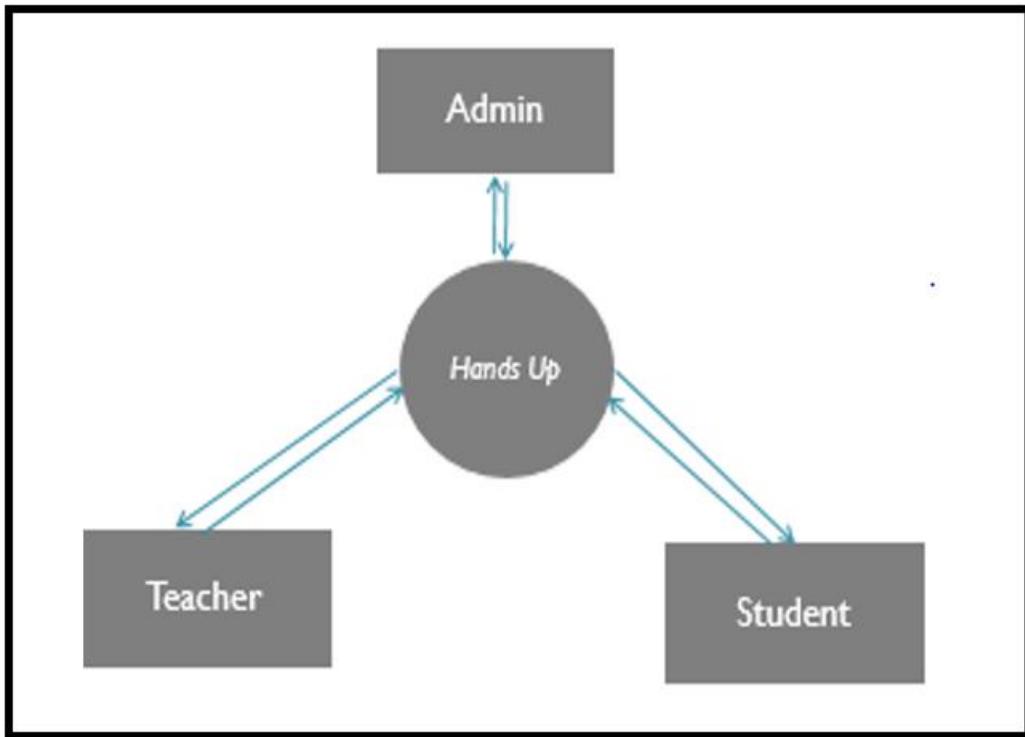


Figure 3.1.1: Level 0 Diagram

3.1.2 LEVEL 1 DFD DAIGRAM

The Level 1 DFD is the further decomposition of Level 0 processes into sub-processes (sub-systems) which give detailed description of the data flow in each processes. Here we have decomposed the process View enrolled student lists in the sub-process Take attendance. The faculty selects the course ID and get the details of the enrolled student and then he can take attendance for that particular course . The data storage used in this level is the Attendance list.

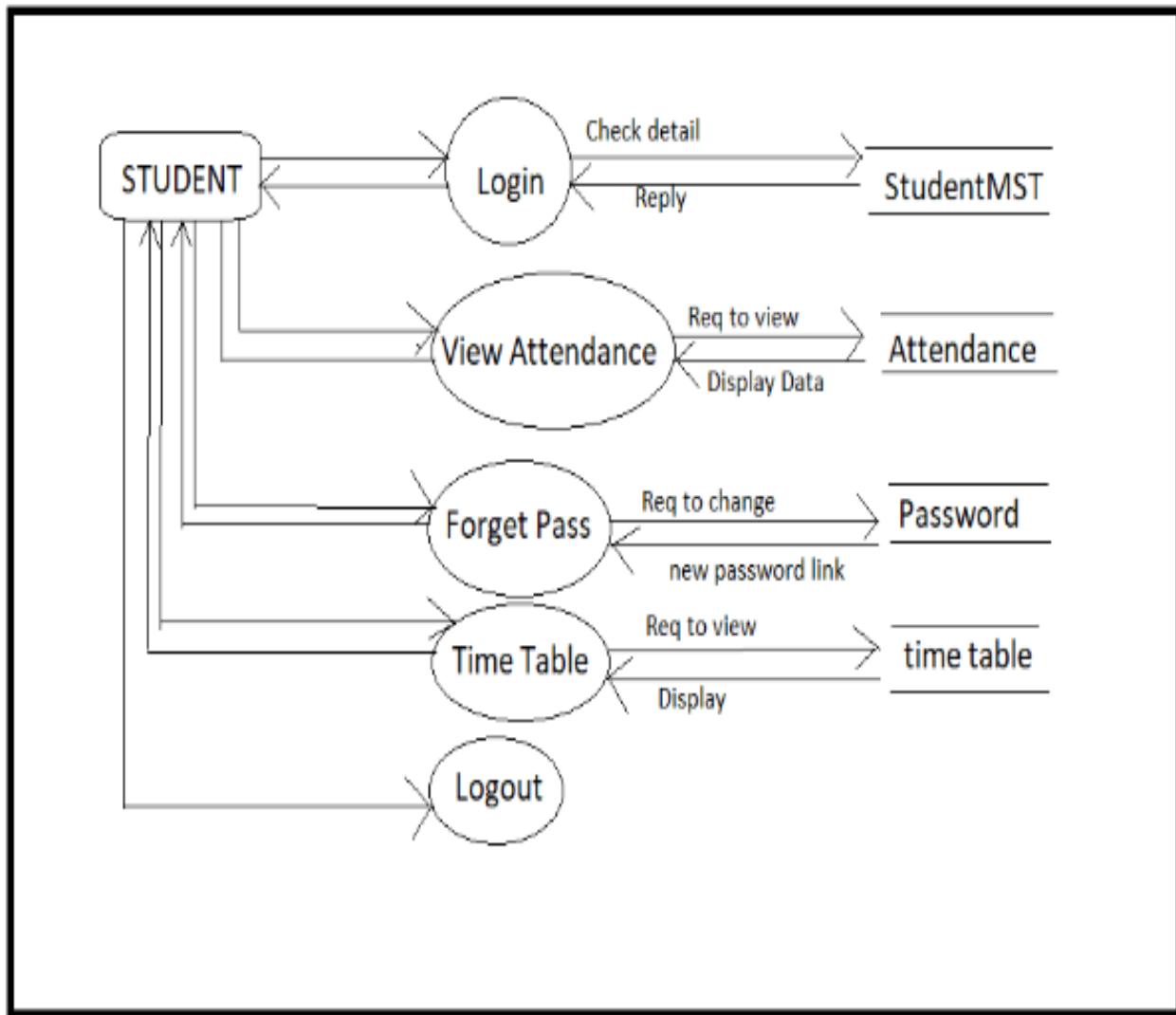


Figure 3.1.2: Level 1 Diagram of Student Application

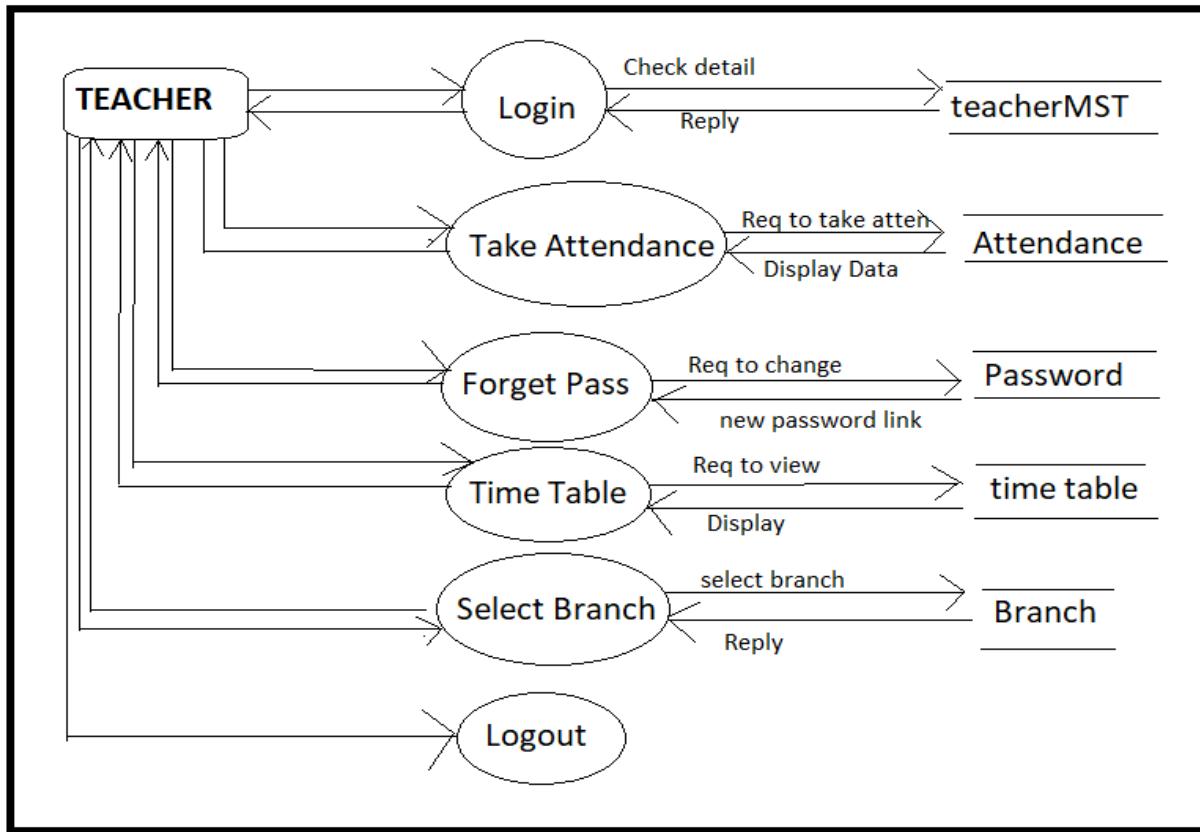


Figure 3.1.2: Level 1 Diagram of Teacher Application

3.2 OBJECT ORIENTED SOFTWARE DESIGN

In this design approach the system is viewed as a collection of entities (objects). Each object manages its own properties (attributes) and state.

3.2.1 ER DIAGRAM

The entity-relationship diagram is a data modeling technique that graphically represents an information systems entities and the relationships between those entities. An ER diagram is a conceptual and representational model of data which is used to represent the system framework infrastructure. The ER diagram contains following elements:

- Entities

- Relationships
- Attributes

In designing the ER diagram, we identify and define all the entities, determine the interactions between the entities and determine the cardinality of the relationship. The ER diagram for the attendance system represents all the entities namely staff, Course, Enrolled Student and Attendance and the relationship between these entities used in the system. The student entity contains the attributes roll no, enroll no, name, branch, address where the roll no is its primary key. The staff entity contains the attributes Sub code, Surname and Tbname where Sub code is its primary key.

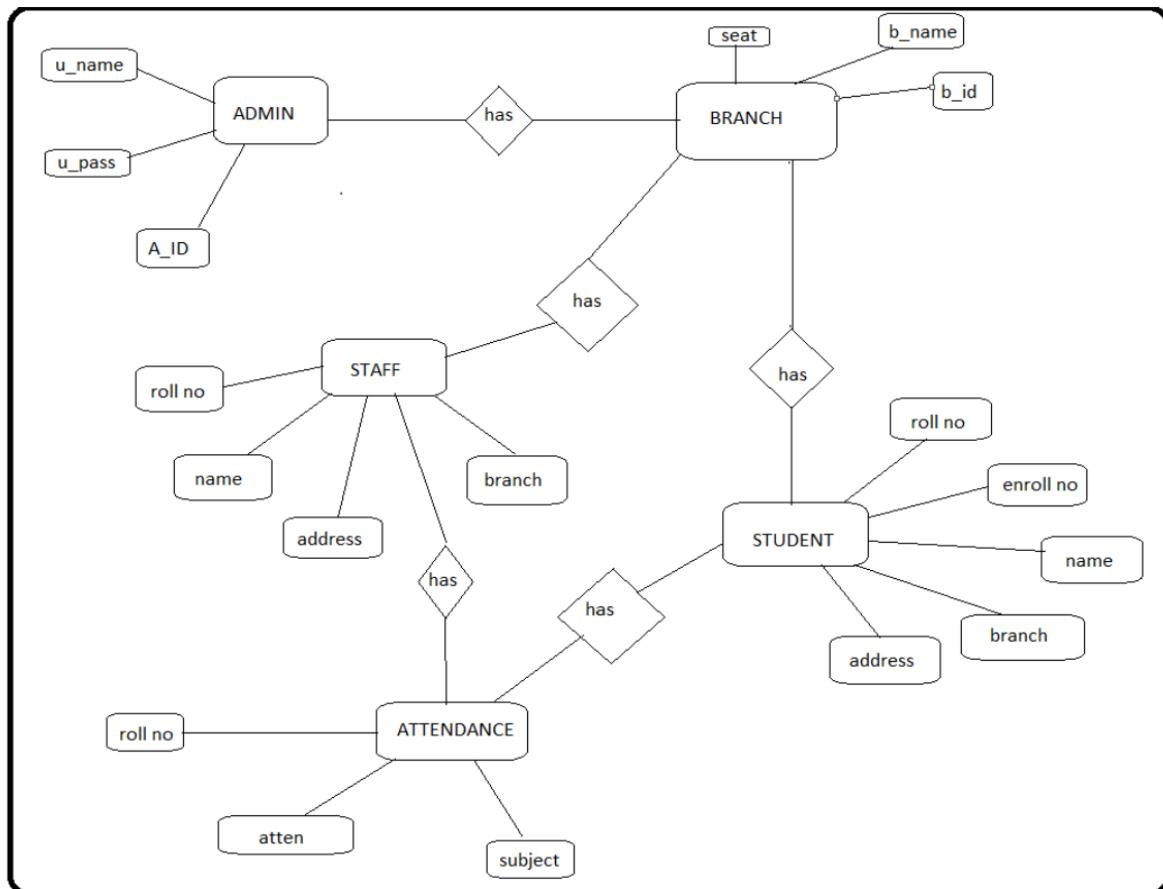


Figure 3.2.1: ER DIAGRAM

3.3 USE CASE DIAGRAM

Use case diagrams are the diagrammatic representation depicting users interactions with the system. This diagram shows different types of users and various ways in which these users interact with the system.

The use case diagram for a faculty. It shows all the different possible ways in which a faculty can use the attendance system. Every faculty can use the system through their mobile phones to take the attendance. The faculty after logging into the system can view the courses taken. Then he can view the list of enrolled students in a particular courses and can take attendance for that subjects. The user can also view the attendance at a later stage. He can directly upload the attendance details in the server through his mobile phone

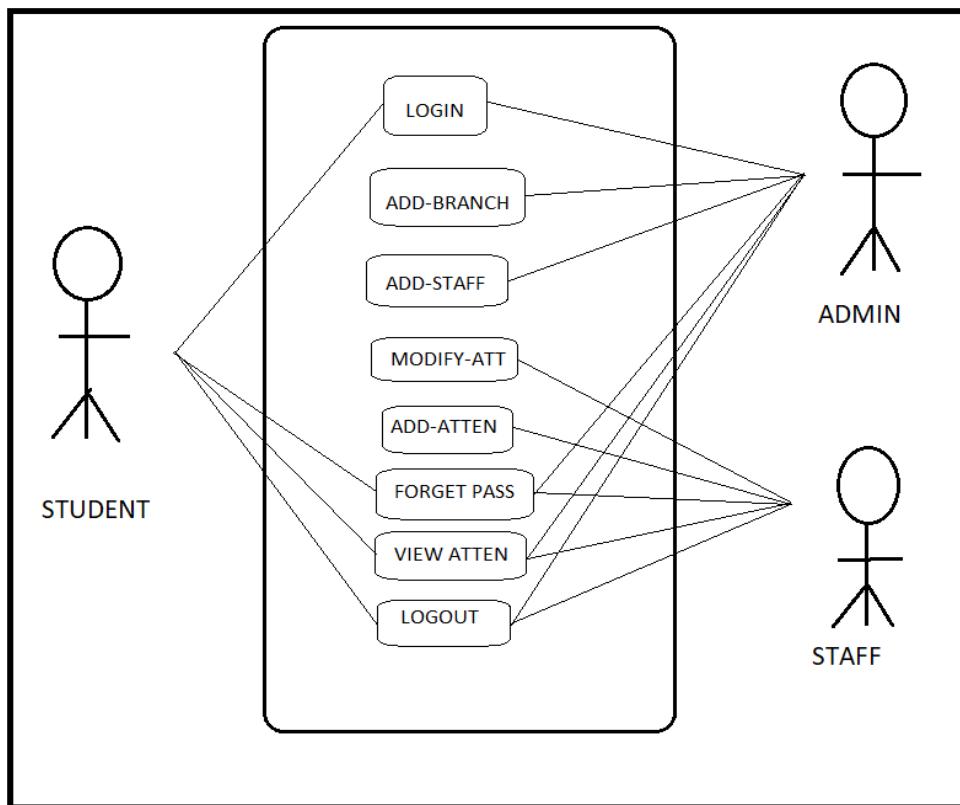


Figure 3.3: Use Case Diagram

3.4 SEQUENCE DIAGRAM

Sequence diagram is an interaction diagram which shows how the processes interact with one another and in what order. It shows the object interactions arranged in time sequence. It represents the objects and classes involved in the scenario. It also shows the sequence of messages exchanged between those objects which is needed to perform different functionality of the scenario. Sequence diagrams are associated with use case realizations of the Logical View of the system.

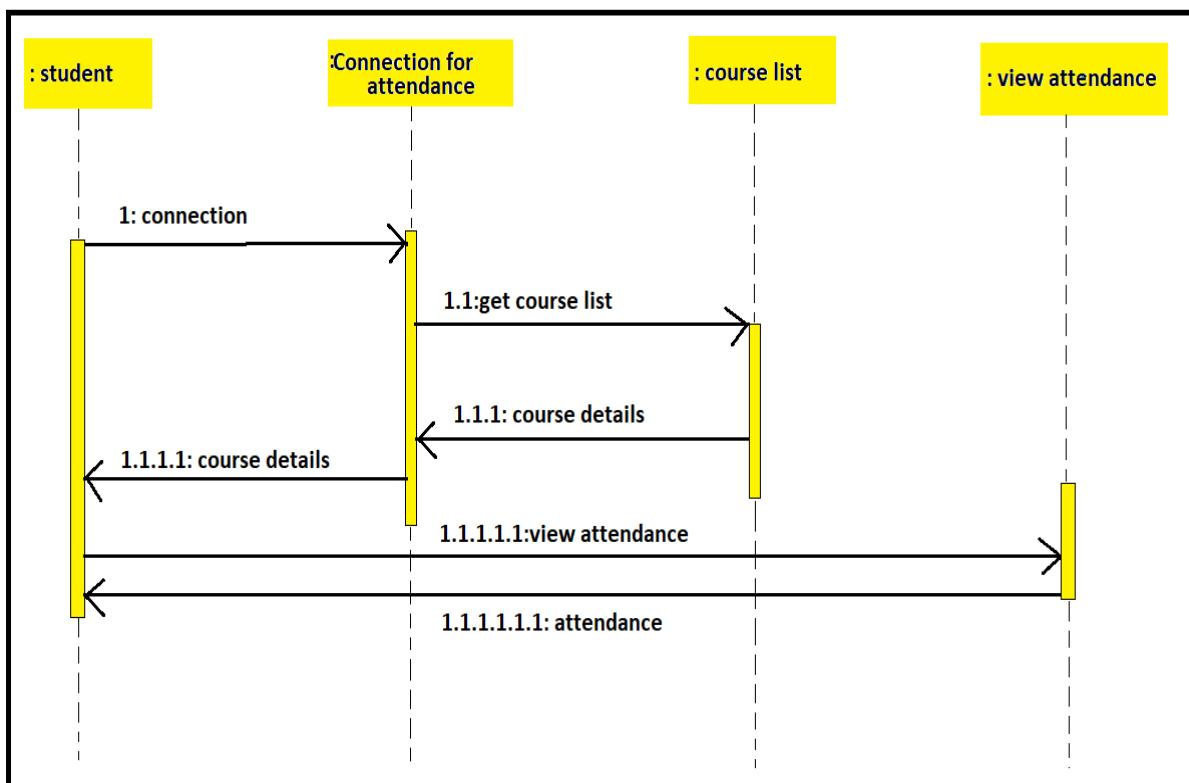


Figure 3.4.1: Sequence Diagram of Student Application

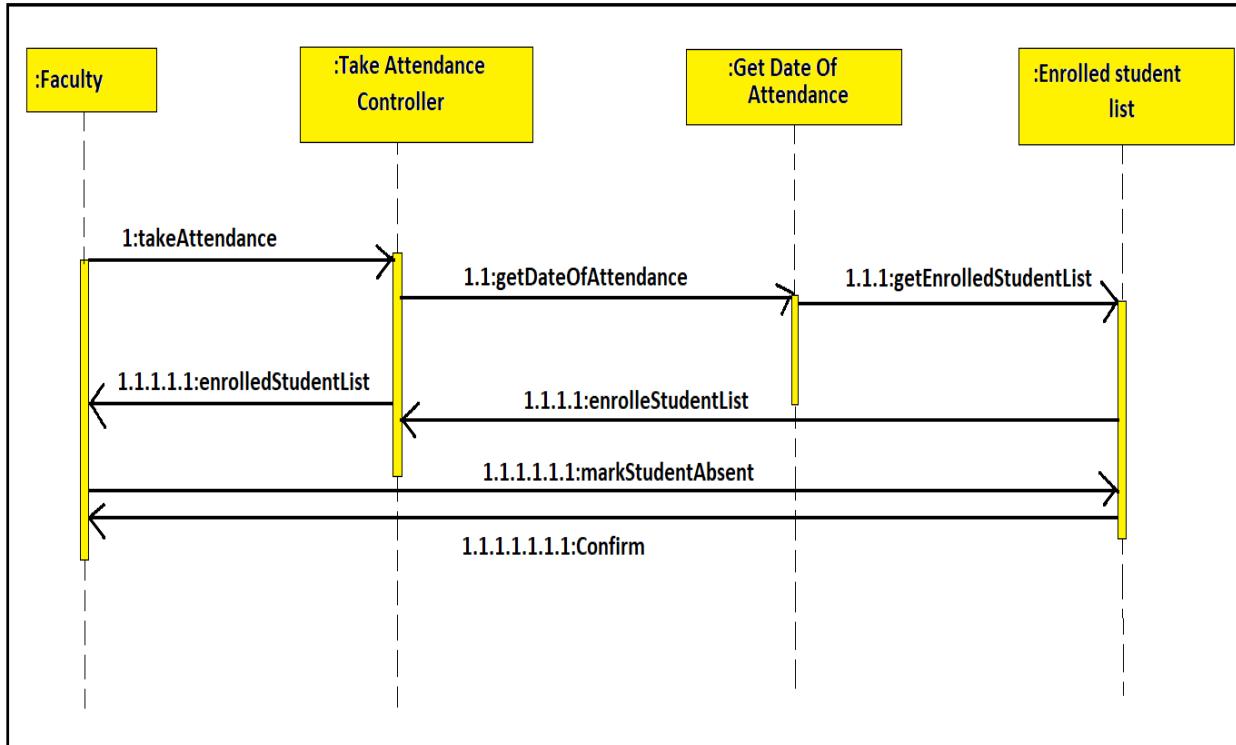


Figure 3.4.2: Sequence Diagram of Teacher Application

3.5 CLASS DIAGRAM

Class diagram is a type of static structure diagram which describes the structure of a system by representing the classes of the system, their attributes, operations and the relationships among these classes.

The class diagram for the attendance system. In this system, we have five classes namely Faculty, Student, Course, Course list and Attendance. The class Faculty can search the Course List, get the Enrolled Student List and take the Attendance. The class Faculty has many-to-many relationship with class Course List and Student and one-to-one relationship with class Attendance. The class Course List contains the details of the Course and has one-to-many

relationship with class Course. Each class Student is enrolled in different courses. The class Student has many-to-many relationship with the class Course.

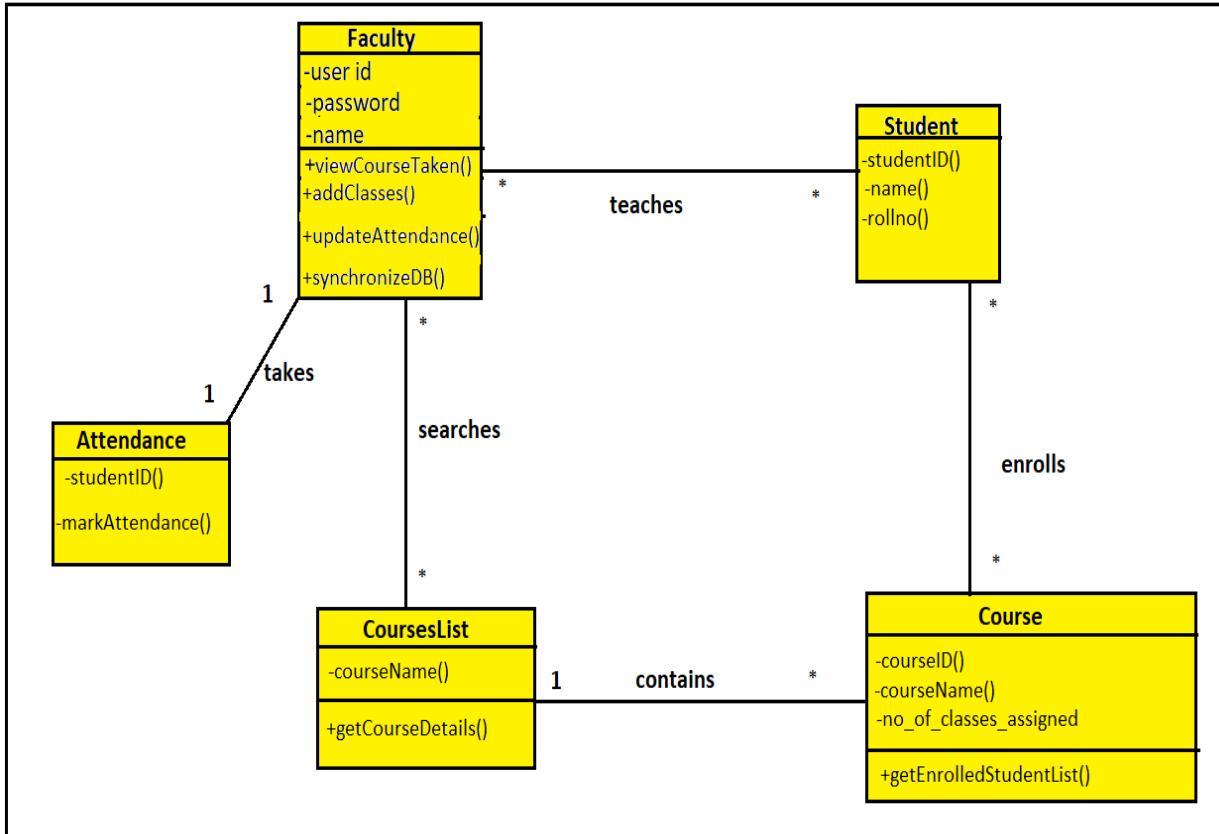


Figure 3.5: Class Diagram

CHAPTER NO 4

TESTING

4.1 BLACK BOX TESTING:

It is carried out to test functionality of the program, it is also called 'Behavioral' testing. The tester in this case has a set of input values and respective desired results. On providing input, if the output matches with the desired results, on providing input. If the output matches with the desired results the programs if tested 'ok', and problematic. In this testing method, the design and structure of the code are not known to the tester.

The checklist is split into five different fields:

- Device specific characteristics, these are characteristics that we related to the device on which the application is installed.
- Network specific checks.
- Application checks. these are things to check that have to do with functionality that is frequently used in the application.
- Application user interfaces checks.

DEVICE SPECIFIC CHECKS

#	DESCRIPTION	YES/ NO ?	REMARKS
1.1	CAN THE APPLICATION BE INSTALLED? ON THE DEVICE?	YES	ALSO CAN BE AVAILABLE ON PLAYSTORE.
1.2	DOES THE APP BEHAVE AS DESIGNED/DESIRED IF THERE IS AN INCOMING CALL ?	YES	WORKING FINE
1.3	DOES THE APP BEHAVE AS DESIGNED/DESIRED IF THERE IS AN INCOMING SMS?	YES	WORKING FINE
1.4	DOES THE APP BEHAVE AS DESIGNED/DESIRED IF THE CHARGER IS CONNECTED ?	YES	WORKING FINE
1.5	DOES THE APP BEHAVE AS DESIGNED/DESIRED IF THE CHARGER IS DISCONNECTED ?	YES	WORKING FINE

1.6	DOES THE APP BEHAVE AS DESIGNED/DESIRED IF THE DEVICE IS TILTED ?	YES	WORKING FINE
1.7	IN CASE THERE'S A TRUE "BACK" BUTTON AVAILABLE ON THE DEVICE DOES THE "BACK" BUTTON TAKE THE USER TO THE PREVIOUS SCREEN?	YES	IT WILL
1.8	IN CASE THERE'S A TRUE "MENU" BUTTON AVAILABLE ON THE DEVICE DOES THE "MENU" BUTTON SHOW THE APP'S MENU?	YES	IT WILL
1.9	IN CASE THERE'S A TRUE "HOME" BUTTON AVAILABLE ON THE DEVICE DOES THE "HOME" BUTTON GET THE USER BACK TO THE HOME SCREEN OF THE DEVICE?	YES	IT WILL
1.10	DOES THE APP BEHAVE AS DESIGNED/DESIRED IF THE "BATTERY LOW" MESSAGE IS PUSHED?	YES	WORKING FINE
1.11	DOES THE APP BEHAVE AS DESIGNED/DESIRED IF THE SOUND ON THE DEVICE IS TURNED OFF?	YES	WORKING FINE
1.12	DOES THE APP BEHAVE AS DESIGNED/DESIRED IF THE IS IN THE AIRPLANE MODE ?	YES	WORKING FINE
1.13	CAN THE APP BE DE-INSTALLED FROM THE DEVICE?	YES	CAN BE UNINSTALL
1.14	CAN THE APP BE FOUND ON THE APP STORE?	NO	STILL NOT AVAILABLE

WHITE BOX TESTING:

It is conducted to test program and its implementation, in order to improve code efficiency or structure. It is also known as 'structural' testing. In this testing method, the design and structure of the code are known to the tester. Programmers of the code conduct this test on the code.

4.2 MEMORY LEAK:

leak canary from square is a good tool for detecting memory leaks in your application. It creates weak references to activities in your application. (you can also customize it by adding watches to any other objects) it then checks if the reference is cleared after coif not, it dumps heap into a.

hprof file and analysis it to confirm if there is a leak. if there is one, it shows a notification and in a separate application, it shows reference tree of how the leak happens. You can find more about leak canary in this article: [leak canary: detect all memory leaks.](#)

CHAPTER NO5

Implementation and User Interface

WORKING OF THE APPLICATION:

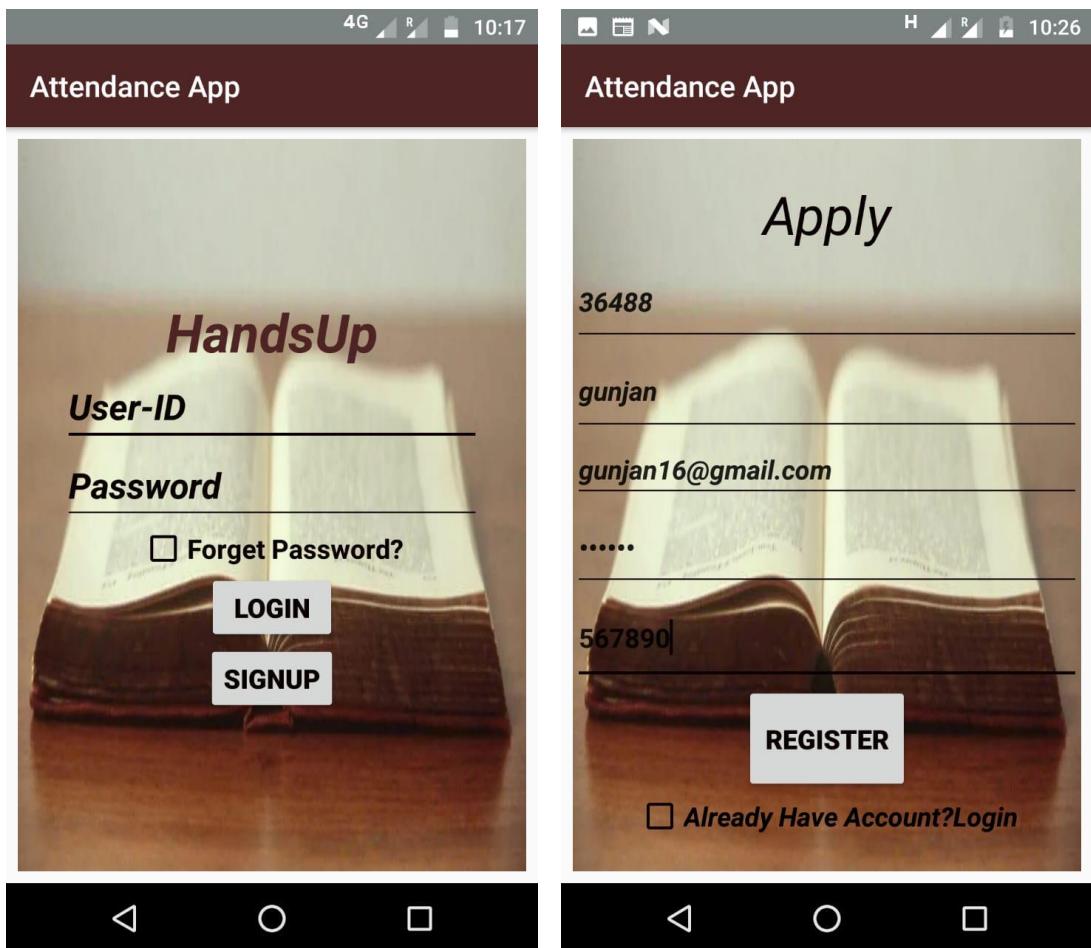


Figure 5.1 is Login Page and Figure 5.2 is Signup page

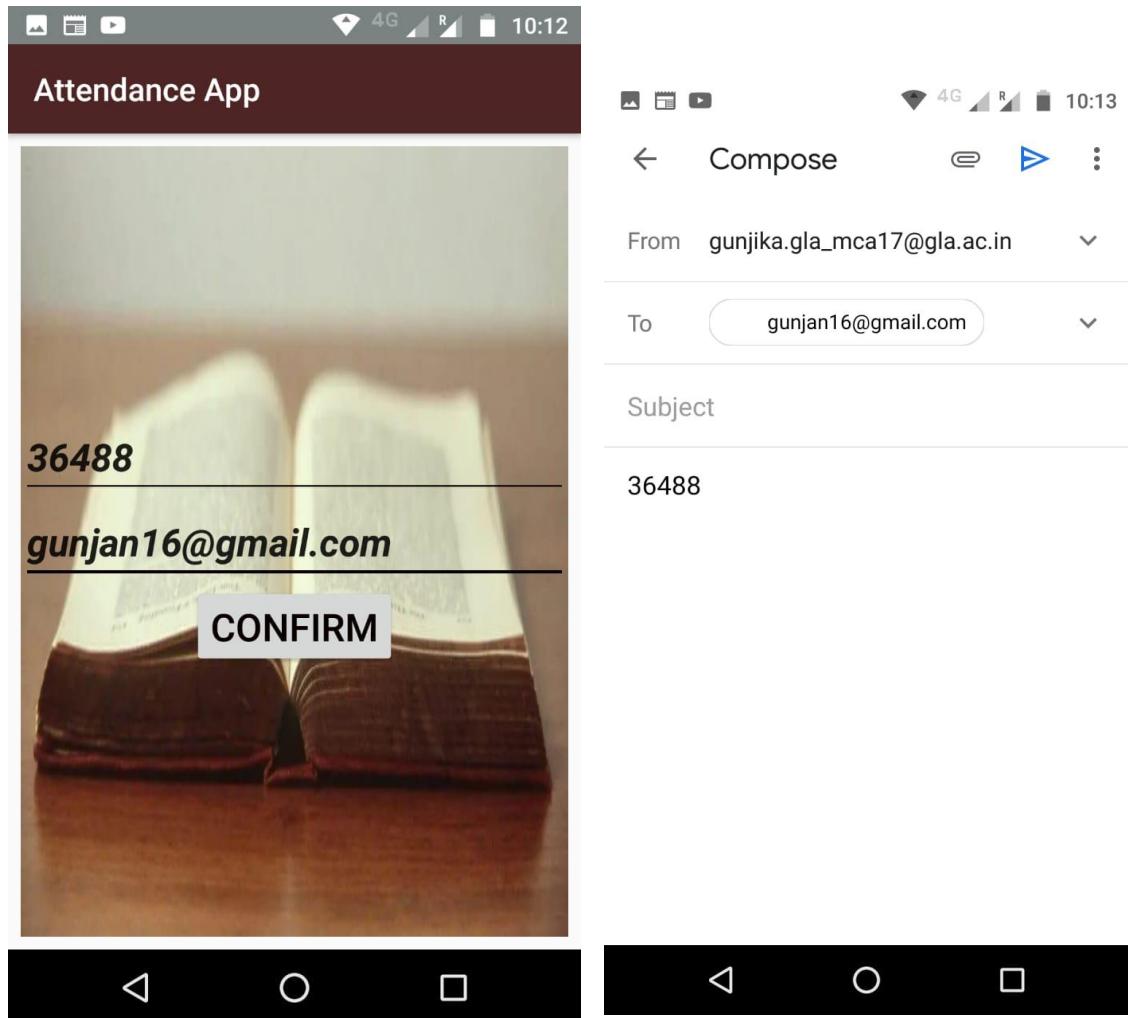


Figure 5.3 is Forget Password Page and Figure 5.4 is Link Page

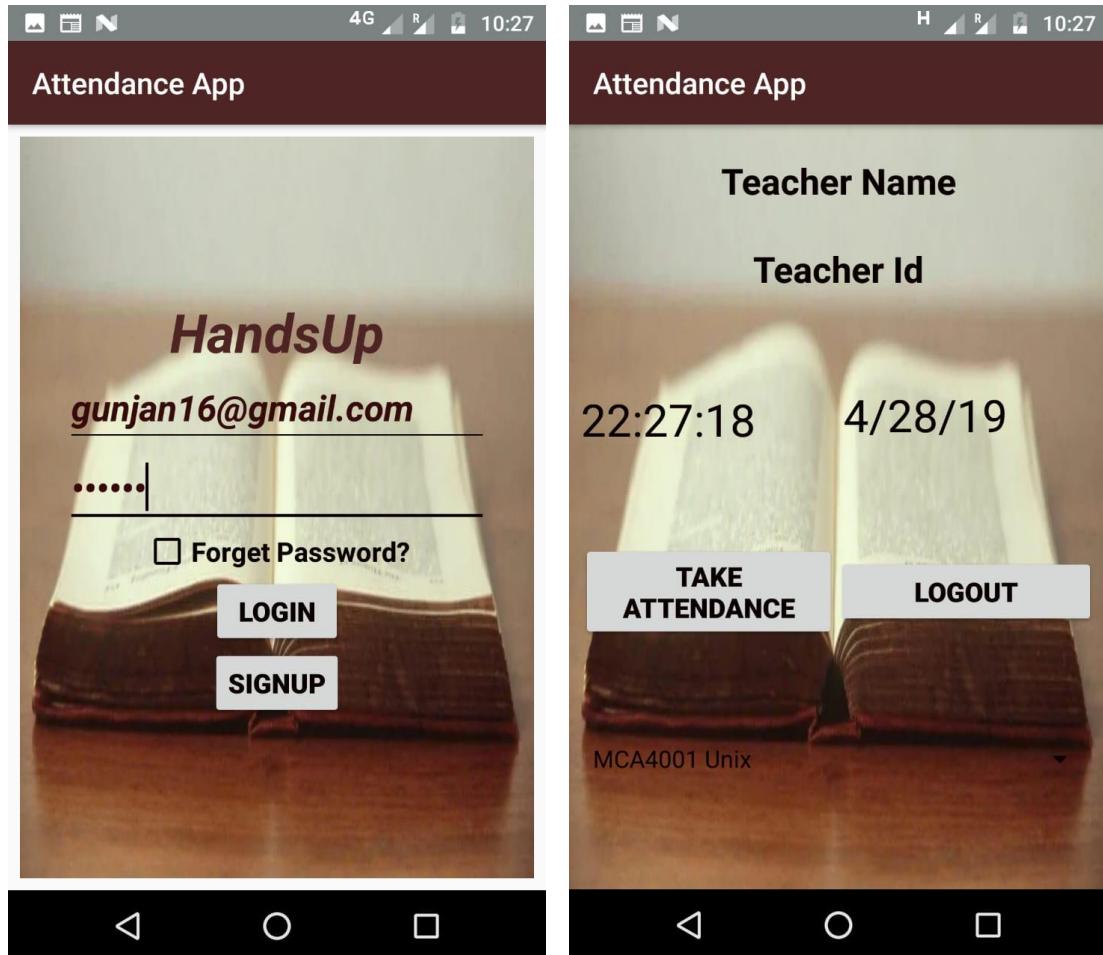


Figure 5.5 is Teacher Login Page and 5.6 is Teacher Profile Page

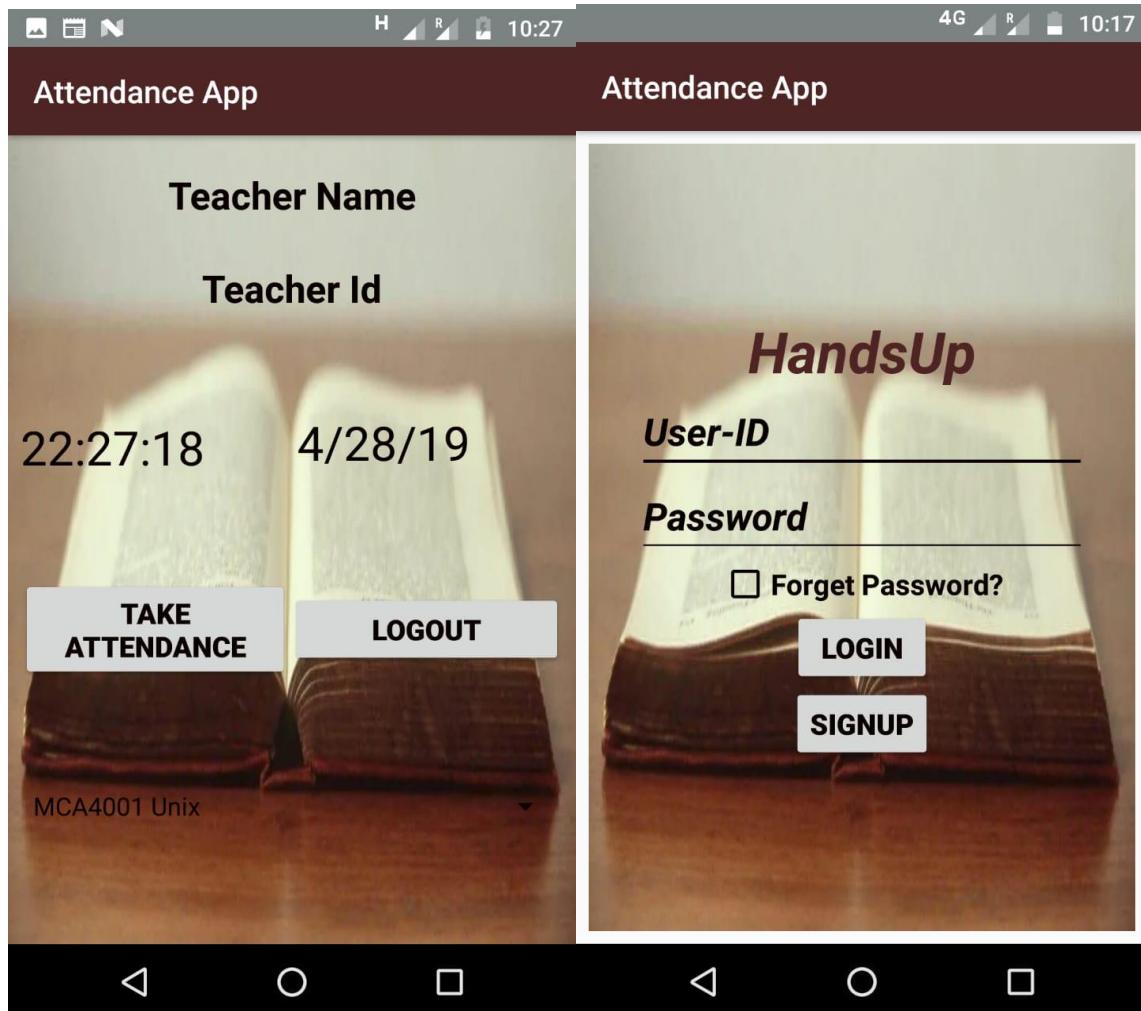


Figure 5.7 is Logout Page and 5.8 is Home page

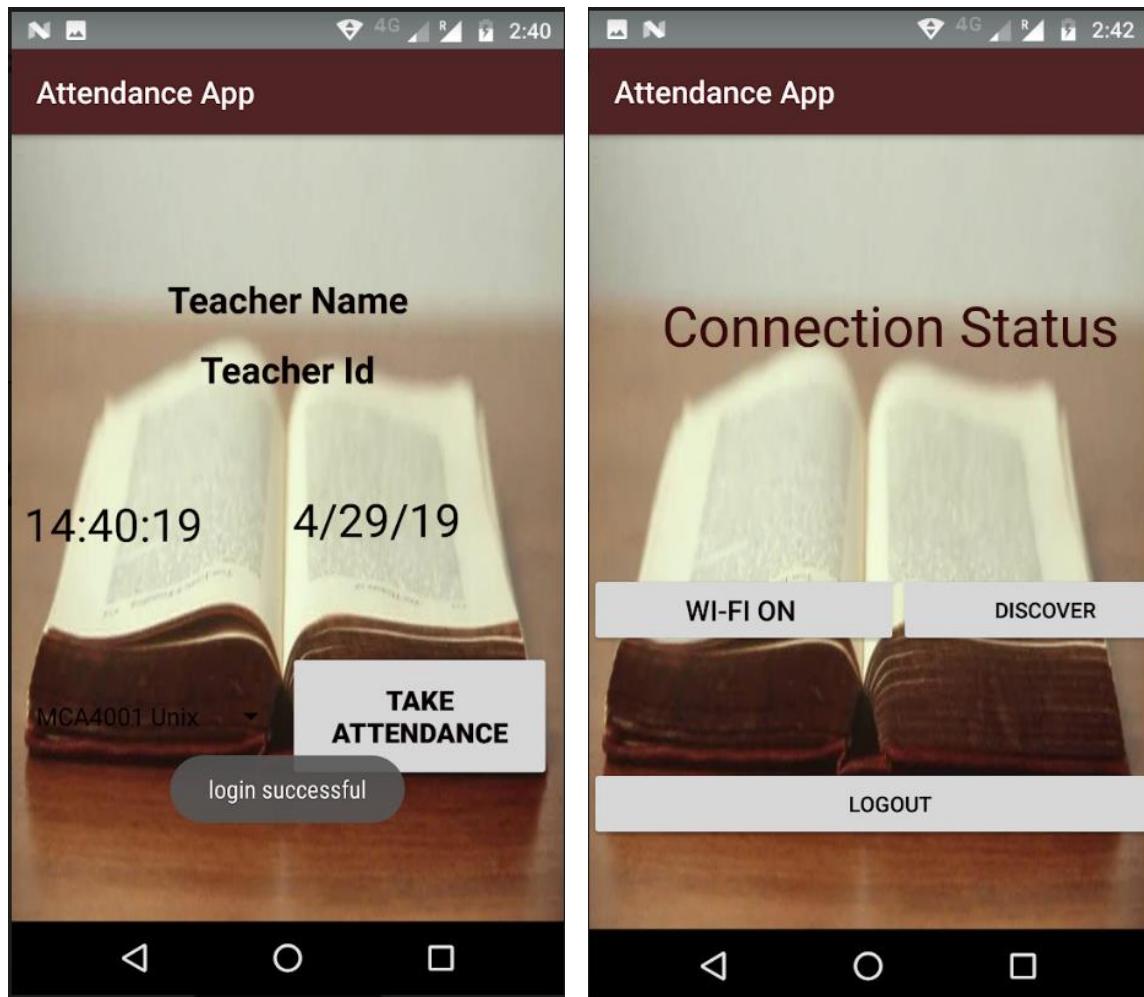


Figure 5.9 is Teacher Profile Page and Figure is Wi-Fi connection page

CHAPTER NO 6
REFERENCES/BIBLIOGRAPHY

6.1 REFERENCES/BIBLIOGRAPHY:

To develop this project on Android Based Attendance Application. We take some knowledge towards android studio and firebase from some books and internet that are given below.

- Android eBook for MCA4044 GLAU
- <https://www.youtube.com/playlist?list=PLGCjwl1RrtcTXrWuRTa59RyRmQ4OedWrt>
- <https://www.youtube.com/playlist?list=PLGCjwl1RrtcTXrWuRTa59RyRmQ4OedWrt>
- <https://www.youtube.com/watch?v=7Yc3Pt37coM&t=601s>
- https://www.youtube.com/watch?v=EM2x33g4syY&list=PLk7v1Z2rk4hj6SDHf_YybDeVhUT9MXaj1&index=1
- https://www.youtube.com/results?search_query=android+developer+tutorial+for+beginners
- <https://www.youtube.com/watch?v=nw627o-8Fok&list=PLFh8wpMiEi88SIJ-PnJjDxktry4lgBtN3>

**CHAPTER NO 7
APPENDICES****7.1 TEACHER MODULE:****7.1.1 MainActivity.java (Home Page):**

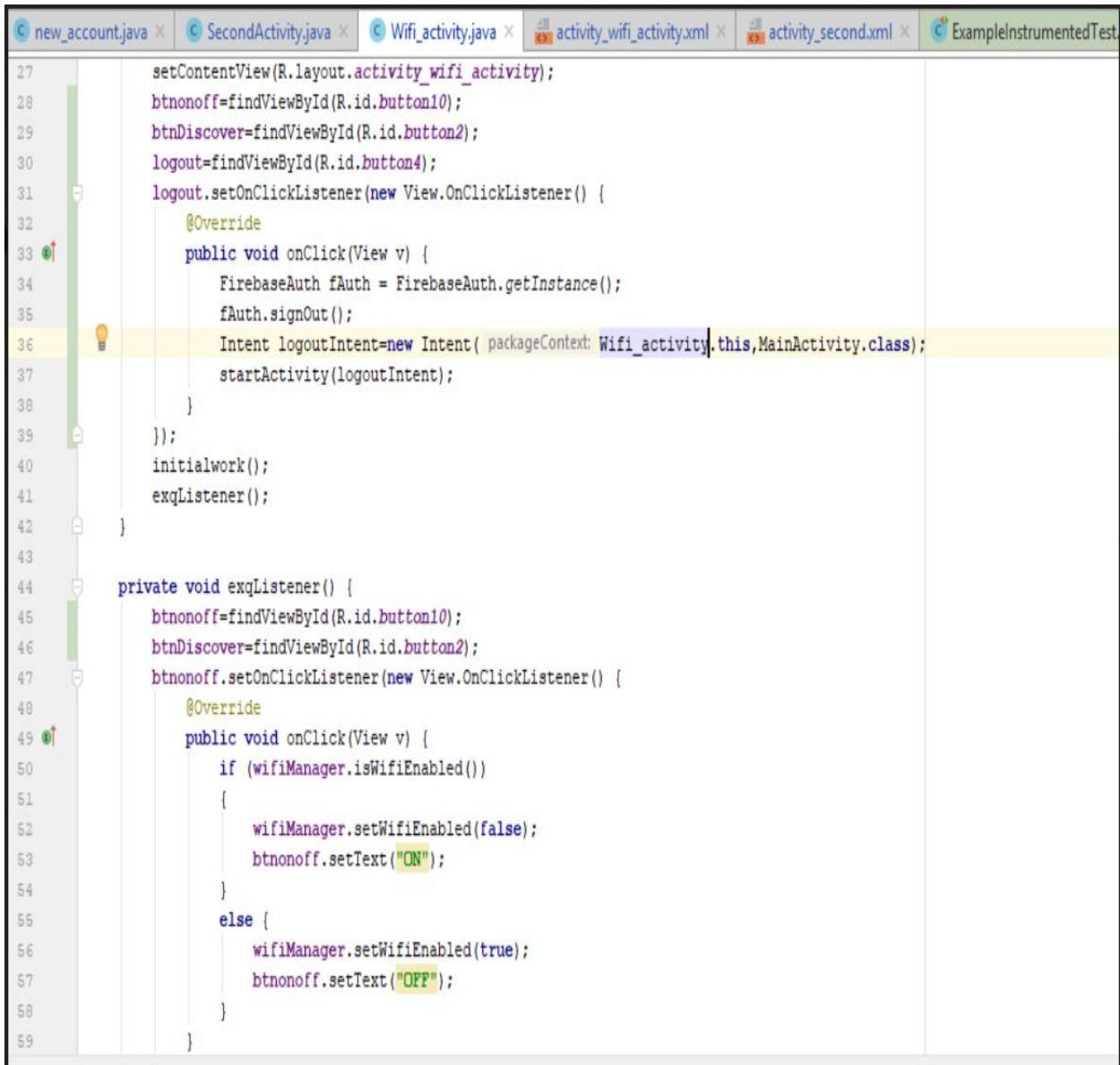
The screenshot shows the Android Studio code editor with the MainActivity.java file open. The code implements a login functionality using Firebase Authentication. It includes imports for FirebaseAuth and AuthResult, declares UI components (textEmail, textPass, button, button1, checkBox), initializes them via findViewById, and sets up click listeners. It also checks if a user is already signed in and starts the attendance record activity if they are. The code editor has a status bar at the bottom indicating it's an IDE and plugin for Android Studio.

```
16 import com.google.firebase.auth.AuthResult;
17 import com.google.firebase.auth.FirebaseAuth;
18
19 public class MainActivity extends AppCompatActivity implements View.OnClickListener {
20     TextView textEmail, textPass;
21     Button button, button1;
22     CheckBox checkBox;
23     private FirebaseAuth mAuth;
24
25     @Override
26     protected void onCreate(Bundle savedInstanceState) {
27         super.onCreate(savedInstanceState);
28         setContentView(R.layout.activity_main);
29         textEmail = findViewById(R.id.editText);
30         textPass = findViewById(R.id.editText2);
31         button = findViewById(R.id.button);
32         button1 = findViewById(R.id.button4);
33         checkBox = findViewById(R.id.checkBox);
34         mAuth = FirebaseAuth.getInstance();
35         if (mAuth.getCurrentUser() != null)
36         {
37             finish();
38             startActivity(new Intent(getApplicationContext(), attendance_record.class));
39         }
40         button.setOnClickListener(this);
41         button1.setOnClickListener(this);
42         checkBox.setOnClickListener(this);
43     }
44     private void userLogin()
45     {
46         String email = textEmail.getText().toString().trim();
47         String pass = textPass.getText().toString().trim();
```

7.1.2 SecondActivity.java (Profile Of Teacher)

```
3 import ...
23
24 public class SecondActivity extends AppCompatActivity {
25     TextView date, time1;
26     Button takeAttend, logout;
27     Spinner spinner;
28     private FirebaseAuth mAuth;
29     DatabaseReference databaseReference;
30     User userr;
31
32     @Override
33     protected void onCreate(Bundle savedInstanceState) {
34         super.onCreate(savedInstanceState);
35         setContentView(R.layout.activity_second);
36         time1=findViewById(R.id.textView9);
37         takeAttend=findViewById(R.id.button12);
38         date=findViewById(R.id.textView12);
39         //logout=findViewById(R.id.button13);
40         spinner=findViewById(R.id.spinner);
41         mAuth=FirebaseAuth.getInstance();
42         databaseReference= FirebaseDatabase.getInstance().getReference();
43
44         Calendar calendar=Calendar.getInstance();
45         String currentdate= DateFormat.getDateInstance(DateFormat.SHORT).format(calendar.getTime());
46         SimpleDateFormat format=new SimpleDateFormat("HH:mm:ss");
47         String time=format.format(calendar.getTime());
48         time1.setText((CharSequence) time);
49         date.setText(currentdate);
50
51
52
53         databaseReference.addValueEventListener(new ValueEventListener() {
54             @Override
```

7.1.3 wifi_activity.java (Wi-Fi connection)



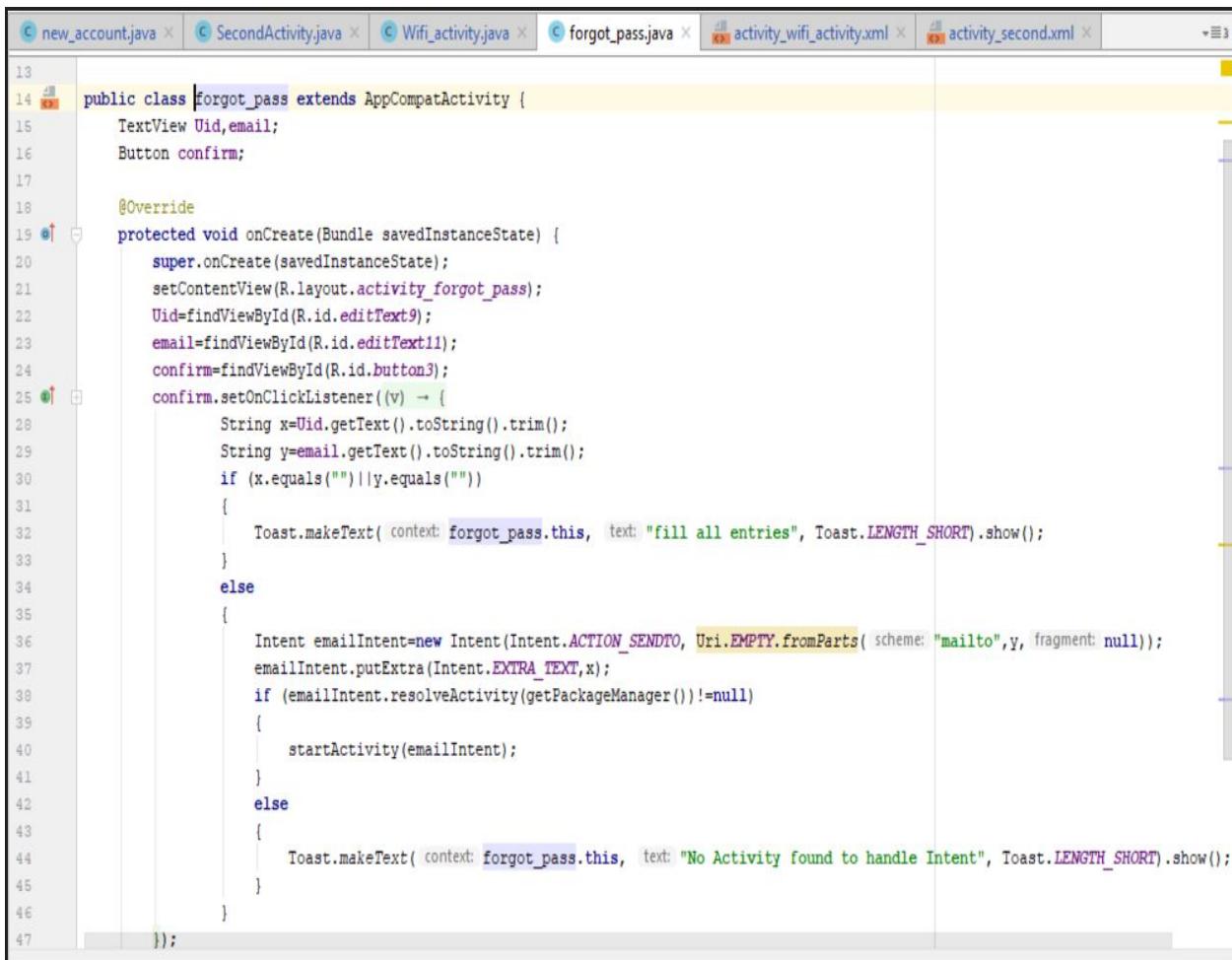
```
27     setContentView(R.layout.activity_wifi_activity);
28     btnonoff=findViewById(R.id.button10);
29     btnDiscover=findViewById(R.id.button2);
30     logout=findViewById(R.id.button4);
31     logout.setOnClickListener(new View.OnClickListener() {
32         @Override
33         public void onClick(View v) {
34             FirebaseAuth fAuth = FirebaseAuth.getInstance();
35             fAuth.signOut();
36             Intent logoutIntent=new Intent( mContext,Wifi_activity.this,MainActivity.class);
37             startActivity(logoutIntent);
38         }
39     });
40     initialwork();
41     exqListener();
42 }
43
44 private void exqListener() {
45     btnonoff=findViewById(R.id.button10);
46     btnDiscover=findViewById(R.id.button2);
47     btnonoff.setOnClickListener(new View.OnClickListener() {
48         @Override
49         public void onClick(View v) {
50             if (wifiManager.isWifiEnabled())
51             {
52                 wifiManager.setWifiEnabled(false);
53                 btnonoff.setText("ON");
54             }
55             else {
56                 wifiManager.setWifiEnabled(true);
57                 btnonoff.setText("OFF");
58             }
59         }
60     });
61 }
```

7.2 STUDENT MODULE:

7.2.1 new_account.java (New Account Signup):

```
71     }
72 }
73 }
74
75 private void register(final String s1, final String s2, final String s3, final String s4, String s5) {
76
77     mAuth.createUserWithEmailAndPassword(s3, s4)
78         .addOnCompleteListener(task) -> {
79             if(task.isSuccessful()) {
80                 Toast.makeText( context: new_account.this, text: "Registered Successfully!!", Toast.LENGTH_SHORT).show()
81             }
82             else
83                 if (task.getException() instanceof FirebaseAuthWeakPasswordException)
84                     Toast.makeText( context: new_account.this, text: "PassWord Too Weak.\nTry Again!!", Toast.LENGTH_SHORT).show()
85                 else if (task.getException() instanceof FirebaseAuthUserCollisionException)
86                     Toast.makeText( context: new_account.this, text: "User Already Exists!! \nTry Again!!", Toast.LENGTH_SHORT).show()
87                 else if (task.getException() instanceof FirebaseNetworkException)
88                     Toast.makeText( context: new_account.this, text: "Internet Not Available!\nRetry... ", Toast.LENGTH_SHORT).show()
89                 else
90                     Toast.makeText( context: new_account.this, task.getException().toString(), Toast.LENGTH_SHORT).show()
91
92     });
93
94
95 databaseReference.addValueEventListener(new ValueEventListener() {
96     @Override
97     public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
98         User user=new User(s1,s2,s3,s4);
99         databaseReference.setValue(user);
100
101
102        /* User usri=dataSnapshot.getValue(User.class);
103        String temp=usri.getName();
104        USid.setText(temp); */
105    }
106 }
```

7.2.2 forgot_pass.java (Forget Password Page):



```
13
14 public class forgot_pass extends AppCompatActivity {
15     TextView Uid,email;
16     Button confirm;
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_forgot_pass);
22         Uid=findViewById(R.id.editText9);
23         email=findViewById(R.id.editText11);
24         confirm=findViewById(R.id.button3);
25         confirm.setOnClickListener((v) -> {
26             String x=Uid.getText().toString().trim();
27             String y=email.getText().toString().trim();
28             if (x.equals("")||y.equals(""))
29             {
30                 Toast.makeText( context: forgot_pass.this, text: "fill all entries", Toast.LENGTH_SHORT).show();
31             }
32             else
33             {
34                 Intent emailIntent=new Intent(Intent.ACTION_SENDTO, Uri.EMPTY.fromParts( scheme: "mailto",y, fragment: null));
35                 emailIntent.putExtra(Intent.EXTRA_TEXT,x);
36                 if (emailIntent.resolveActivity(getApplicationContext())!=null)
37                 {
38                     startActivity(emailIntent);
39                 }
40                 else
41                 {
42                     Toast.makeText( context: forgot_pass.this, text: "No Activity found to handle Intent", Toast.LENGTH_SHORT).show();
43                 }
44             }
45         });
46     });
47 }
```

