# ATTENDANCE TRACKER

*A*

*Project Report*

*Submitted in partial fulfilment of the*
*Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

By

**K.Vivek(1602-17-737-060)**

**P.Sai Teja(1602-17-737-040)**

**K.Naveen(1602-17-737-310)**

*Under the guidance of*

**Mrs.DRL Prasanna**

**Assistant Professor**

**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

**2019-20**

## Vasavi College of Engineering (Autonomous)
## (Affiliated to Osmania University)
## Hyderabad-500 031
## Department of Information Technology

# DECLARATION BY THE CANDIDATE

We, **K.Vivek, P.Sai Teja, K.Naveen** bearing hall ticket numbers, **1602-17-737-060**,**1602-17-737-040**,**1602-17-737-310** respectively hereby declare that the project report entitled **"Attendance Tracker"** under the guidance of **Mrs.DRL Prasanna,** Assistant Professor, Department of Information Technology, Vasavi College of Engineering, Hyderabad, submitted in partial fulfilment of the requirement of THEME BASED PROJECT of VI Semester of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by us and the results embodied in this project.

**K.Vivek**
**1602-17-737-060**

**P.Sai Teja**
**1602-17-737-040**

**K.Naveen**
**1602-17-737-310**

## Vasavi College of Engineering (Autonomous)

## (Affiliated to Osmania University)

## Hyderabad-500 031

## Department of Information Technology

### BONAFIDE CERTIFICATE

This is to certify that the project entitled **“ATTENDANCE TRACKER”** being submitted by **K.Vivek, P.Sai Teja, K.Naveen** bearing **H.T.NO:1602-17-737-060 ,1602-17-737-040 ,1602-17-737-310** respectively in partial fulfilment of the requirements for the completion of THEME BASED PROJECT of Bachelor of Engineering, VI Semester, in Information Technology is a record of bonafide work carried out by him/her under my guidance.

**Mrs.DRL Prasanna**
**Assistant Professor (IT)**
**Internal Guide**

**Dr. K. Ram Mohan Rao**
**Professor & HOD**
**Department of IT**

# ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them. We would like to take the opportunity to express our humble gratitude to **Mrs. DRL Prasanna**, our internal guide, under whose guidance we have finished this project successfully. Her constant guidance and willingness to share her vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the work. We would like to thank all faculty members and staff of the Department of Information Technology, Vasavi College of Engineering for their generous help in various ways for the completion of this project. Finally, yet importantly, We would like to express our heartfelt thanks to our respected **Dr.K.Ram Mohan Rao(HOD,IT)** who was very helpful in providing resources for the project and also our project coordinators **Mrs.Leela Pallava(Asst.Prof**,IT)**,Mr.Srinivasa Chakravarthy(Asst.Prof,IT)** whose efforts were constant in monitoring our project.

K.Vivek

1602-17-737-060

P.Sai Teja

1602-17-737-040

K.Naveen

1602-17-737-310

# Abstract

Taking attendance in a class can often be a time consuming and a manual process, which is prone to human error(s) and hence recording incorrect data. Also, querying the data per student can be the tedious process since it involves some sort of manual tracking/counting of days attended. For this project, we are creating an android based application that can be used for taking attendance which is easy to use and free from any manual tasks. Admin can add faculty and students. Faculty (Professor/Course Instructor) just have to login with the username and password and take attendance for a particular course in the application. Faculty can view student's day wise attendance and also cumulative attendance i.e total number of classes a particular student attended till the date. Admin can view the total number of classes present by each student enrolled. It helps the teacher to take attendance through their smart phone and to keep the record of the attendance in their pocket for any time use.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# Chapter 1
# Introduction

Nowadays, mobile devices have become a way of life for everyone. Computers are now replaced by compact smart phones that can fit into pocket and can be carried anywhere. Attendance Tracker is an application for taking daily attendance in schools and colleges.. The main motive behind this software is to replace the traditional pen and register system.. 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 70 percent. It is an open source and free ware operating system.

In the attendance system through mobile devices, the faculty takes the attendance which is stored in the mobile database. This project presents the Attendance system through mobile devices, which is developed using Android application. The application is installed in every faculty's mobile phones and can be used to take attendance in offline mode (that is even when there is no network connectivity as the application is a stand-alone application). This application stores the attendance in the mobile internal database and the faculty can view the student's attendance whenever required.

## 1.1 Features of the Attendance Tracker App

1.It has 3 Actors mainly involved

- Admin
- Faculty
- Student

2.Admin adds faculty and students through registration forms

3.Faculty will be given a unique username and password

4.Faculty can login and take student's attendance of a particular day in a particular course.

# Chapter 2

# System Analysis

In the current scenario of various educational institutions to manage and maintaining student information is very tough task for any one. The traditional way of taking attendance through the lecturers is a manually in a register book in which they are used to do a manual calculation to maintain monthly attendance database of the students. Such system takes so much time to generate record and many times it creates an error also as it follows manual traditional system. Apart from this error problems some time many more problems are generated like lost of attendance register book or manually fake data inserted through unknown recourses which are really not authorized for that work. In order to reduce all these problems and to overcome through this problem, a smart mobile based application could be proposed and developed to increase its efficiency. It is a beneficial step in every way for our future E-schools.

## 2.1 Previous System

Following traditional systems are used to mark attendance in the teaching process.

**Manual Attendance System**-It is the conventional method of taking attendance by calling names or signing on paper but it is inefficient due to more chances of malfunctioning and more paper work as well.

**RFID with Object Counter**-Radio Frequency Identification (RFID) based attendance system is one of the solutions to address this problem, but that is time consuming and unsafe. Anyone can carry others card to mark proxy attendance.

**Bluetooth Based Attendance System**-In this, attendance is being taken using instructor’s mobile phone. Application software is installed in instructor’s mobile telephone, enables it to query student’s mobile via Bluetooth. It transfers student’s mobile Media Access Control (MAC) addresses to the instructor’s mobile phone and presence of the student can be confirmed. The problem of this proposed system is student’s phone is required for attendance. In case if the student is absent and if his mobile is with his friend keeping it in coverage area then also his presence would be marked.

## 2.2 Drawbacks or Cons of the Existing System

- Manually calculated maintenance of data
- Suddenly searching for any type of related data is very difficult as well as it create mess over there.
- Portability.
- Losing of data problem can be occurred easily.

## 2.3 Overview of the Proposed System (Android based Attendance)

In the proposed project Android based attendance system is designed which is less time consuming, safe and easy to implement because

**User Friendly**: - This software is user friendly as it is simple to use and the user doesn’t need any special training to use this software. Data evaluation, data storing and retrieval is easy and doesn’t need any heavy calculation or method. The UI is simple and easy to understand.

**Easy Tracking of Sudent’s attendance**: It's easy for the faculty to keep a track on the students attendance record and hence notify the students with minimum attendance to attend classes.

**Minimal paper work**: There is no paper work required. Data are stored automatically in the system. Evaluation are done automatically. Hence it is cost effective too.

**Time Saving**: Data storing, data retrieval, data evaluation is done at minimum time hence it is time saving producing data with minimal errors.

## 2.4 Feasibility Study

Attendance Tracker(Android Based Attendance System) is feasible because of the following reasons:

**Economically Feasibility**: This software is economic from school or colleges point of view. It is cost effective as use of paper has been eliminated .It is time effective since evaluation of attendance is easy.

**Technical feasibility**: This software is technically feasible since there are no extra hardware requirements. The only requirement is an android smartphone with minimum version 4.2 jelly Bean.

**Behavioral Feasibility**: This software is very simple to use. The user doesn't need any special training to use this software. The software has been designed keeping the users point of view

## 2.5 Features of the Attendance Tracker App

- Home screen
- Login screen

Options on Home screen when logged in as Admin:

- Add Faculty
- Add Student
- View Student
- View Faculty
- View Attendance count of students
- Remove Student
- Remove Faculty

Options on Home screen when logged in as Faculty

- TakeAttendance

- View Attendance
- View Total Attendance
- Remove Student

## 2.6 Android and its Overview

Android is a Linux-based operating system developed for smart phones or tablet computers. It is a stack of software that includes operating system, middleware and libraries and APIs written in C.

## 2.7 Features of Android

### 2.7.1 Application Framework

Android application framework is supported by number of open source libraries like OpenSSL, SQLite, and Libc. The application framework is also supported by the Android core libraries.

Android code is written primarily in Java programming language, and is compiled with the help of Android SDK tools. On compiling, it generates an Android package (commonly known as .apk) file which is used to install the application on android device.

### 2.7.2 App Components

To aid in android development, there are mainly four types of components. Each component serves a distinct purpose and allows system to interact with your application in different ways. Broadly speaking, there are four types of components.

- **Activity**- An activity represents a single screen with a user interface. For example, in this Attendance Tracker application, activities include adding a student, register or viewing student's attendance record.
  Life cycle of Activity is as shown below

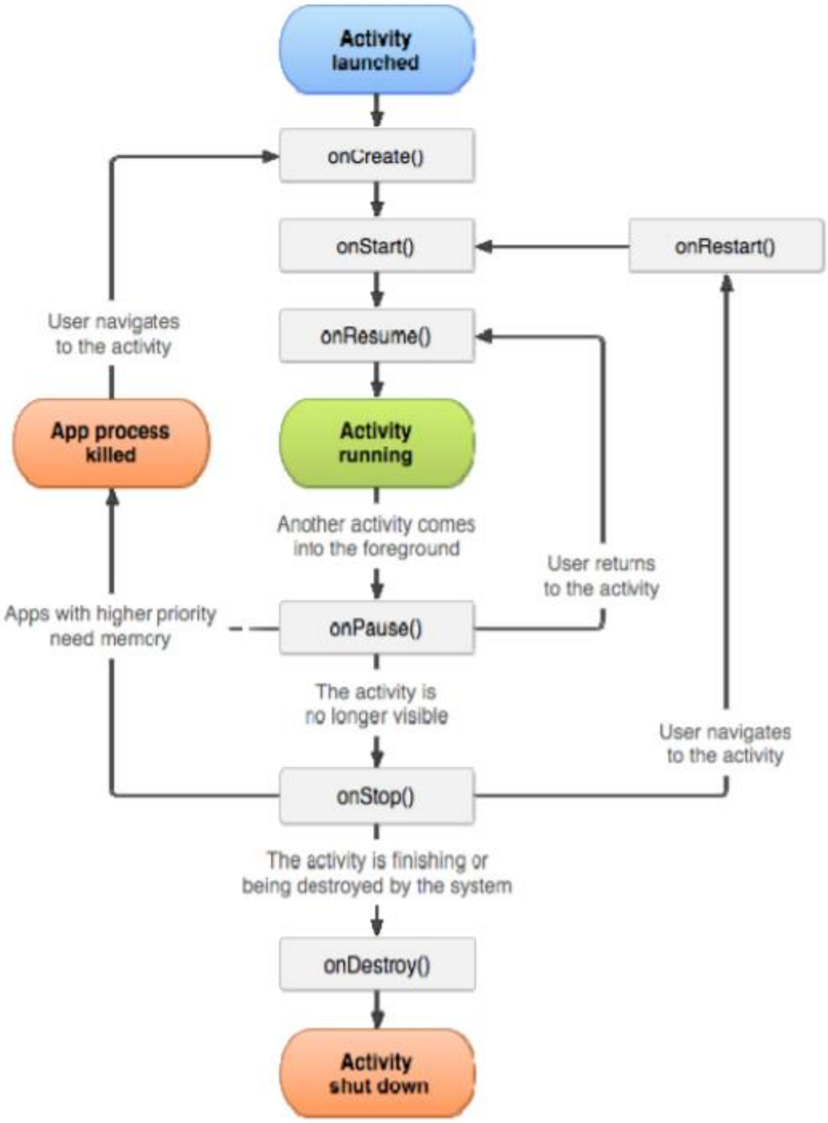

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Fig.2.1 Activity Life Cycle

- **Services-** A Sevice is an application component that can perform long-running operations in the background, and it doesn't provide a user interface. Another application component can start a service, and it continues to run in the background even if the user switches to another application.
- **Content provider** - A content provider manages access to a central repository of data. A provider is part of an Android application, which often provides its own UI for working with the data.
- **Broadcast receivers-** Broadcast receivers are components in your Android application that listen in on broadcast messages

### 2.7.3 Android Application files

Files can be broadly divided into three categories

- Java file
- Layout file
- Manifest file

**Java Files**

These files are the files where all the processing of the events happens, and allows user to interact with the system. Through these files, the layouts can be added dynamically, the user entered values in the text boxes or other input can be obtained and stored.

**Layout Files**

A layout defines the visual structure for a user interface, such as the UI for any activity. These files are responsible for defining the user Input. The Android framework gives you the flexibility to use either or both of these methods for declaring and managing your application's UI.

**Manifest Files**

It is the main part of the android application. This file contains all the information about the application – what android operating system components are present in the application, what permissions are required by the application etc.

## 2.7.4 SQLite Database

Android OS contains the SQLite database management classes which is used by an application to maintain its own private database. SQLite is a relational database management system contained in C programming library

## 2.8 Technology Used for the Project

- The IDE used for the project is Android Studio which is based on IntelliJ software.
- Front End – XML(Extended Markup Language)
- Back End- Java
- Database – SQLite

# Chapter 3

# System Design

## 3.1 USE CASES

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 following are the use cases for Attendance Tracker App:

**1.Admin:** Admin plays the authoritative position. Admin can add faculty & students .

**2.Faculty:** Faculty records attendance of the students**.**

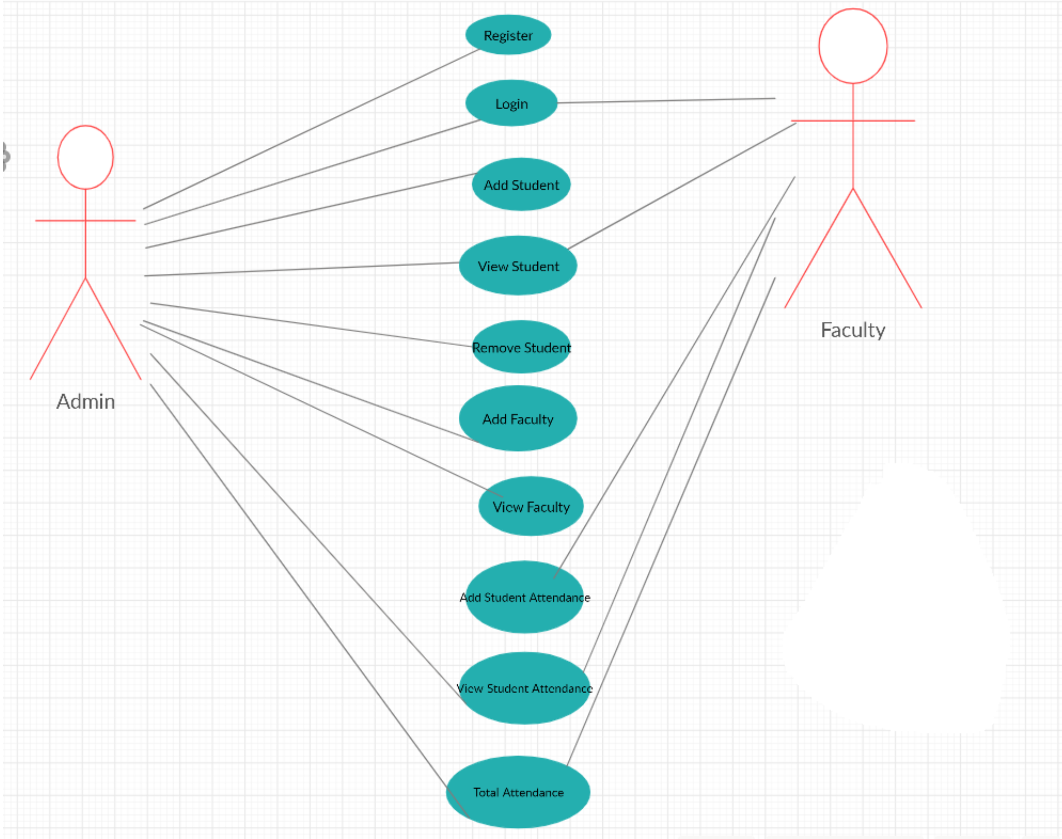


**Fig.3.1 Use Case Diagram**

### 3.2 Class Diagram(Static diagram)

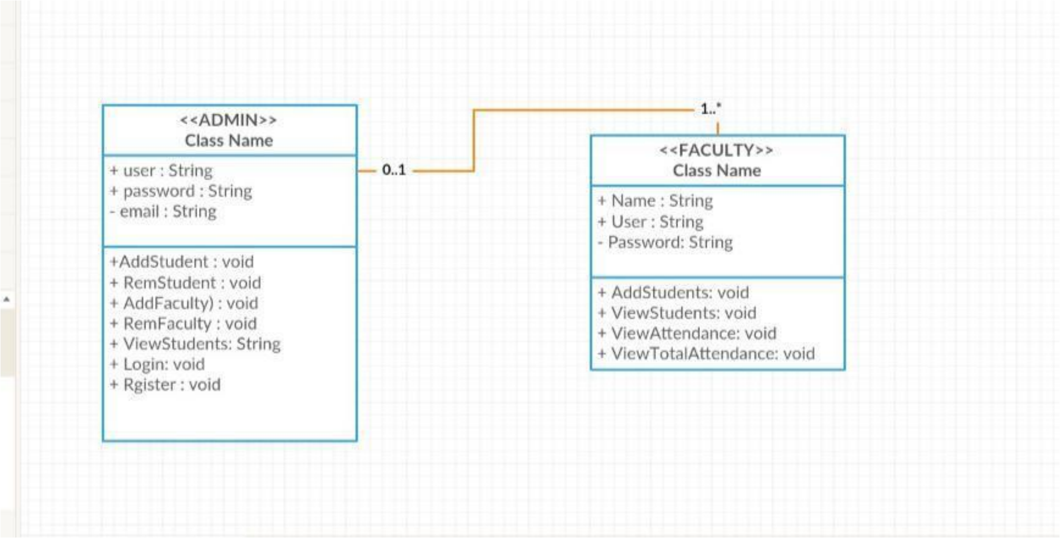


**Fig.3.2 Class Diagram**

### 3.3 Sequence Diagram(Run time diagram)

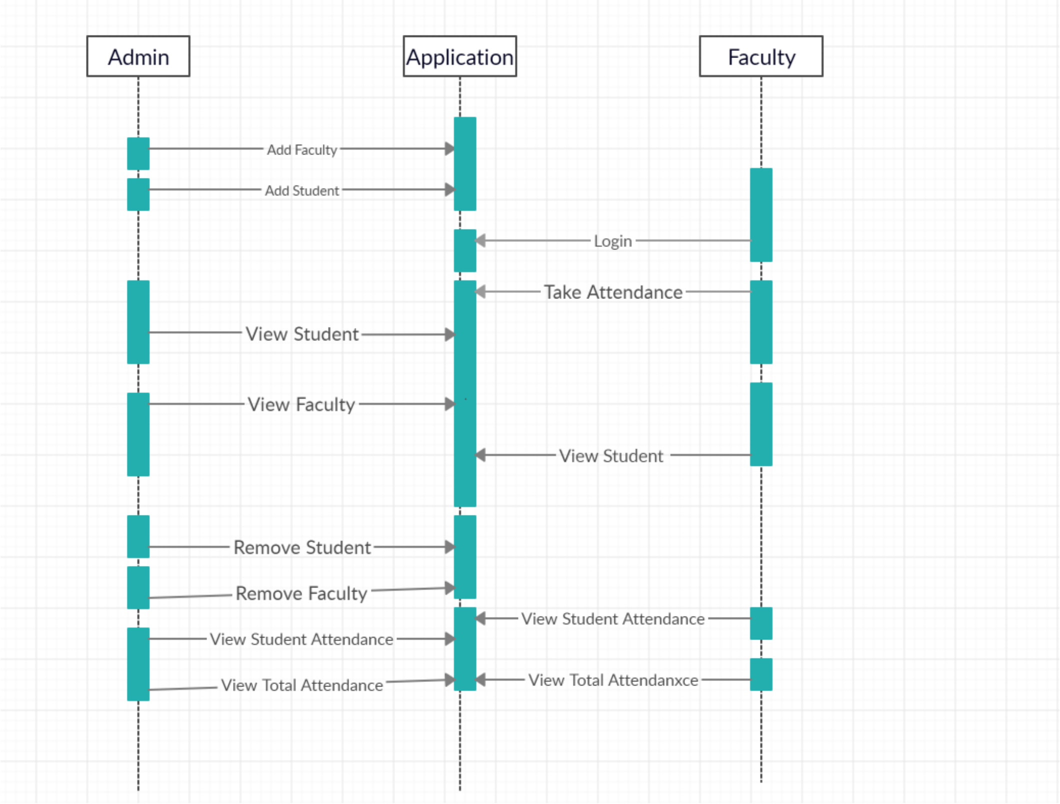


**Fig.3.3 Sequence Diagram**

# Chapter 4

# Implementation

## 4.1 Modules Implemented

After the application has been installed in the android device the following modules are implemented: -

### 4.1.1 Admin Module

- Add Student.
- Add Faculty.
- Add Teacher.
- View Student.
- View Teacher.
- View Student Attendance count.
- View each student’s attendance separately

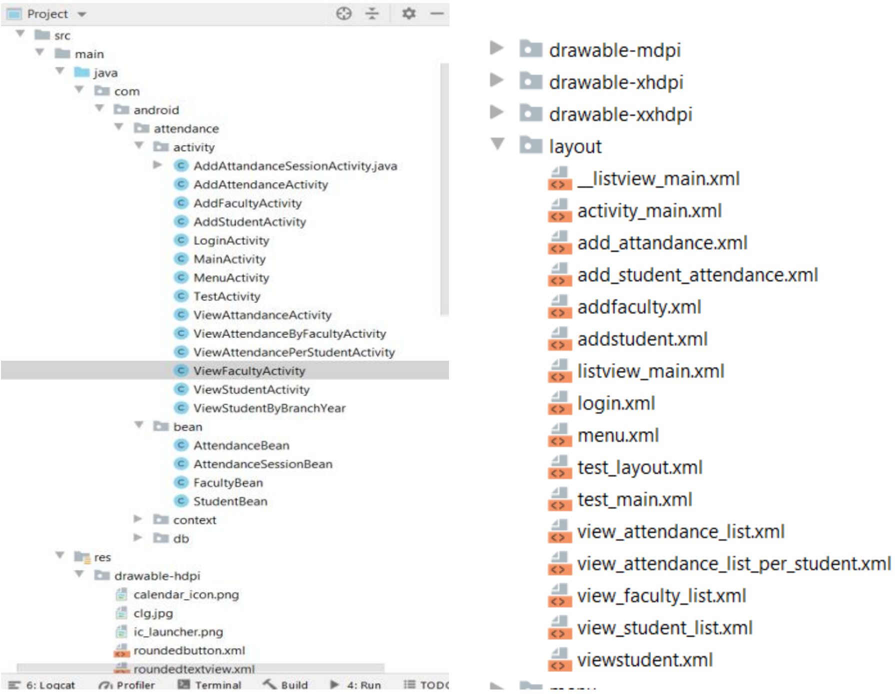
### 4.1.2 Faculty Module

- Take attendance and keep them class wise
- Add New student. View each student’s attendance separately.
- Edit Student/Attendance later.
- Simple designed interface.

### 4.1.3 Four tables created are

**"faculty_table"**;
**"student_table"**;
**"attendance_session_table"**;
**"attendance_table"**;

## 4.2 Activities and Layout files of the Project



```
src
  main
    java
      com
        android
          attendance
            activity
              AddAttandanceSessionActivity.java
              AddAttendanceActivity
              AddFacultyActivity
              AddStudentActivity
              LoginActivity
              MainActivity
              MenuActivity
              TestActivity
              ViewAttandanceActivity
              ViewAttendanceByFacultyActivity
              ViewAttendancePerStudentActivity
              ViewFacultyActivity
              ViewStudentActivity
              ViewStudentByBranchYear
            bean
              AttendanceBean
              AttendanceSessionBean
              FacultyBean
              StudentBean
            context
            db
    res
      drawable-hdpi
        calendar_icon.png
        clg.jpg
        ic_launcher.png
        roundedbutton.xml
        roundedtextview.xml
```

```
drawable-mdpi
drawable-xhdpi
drawable-xxhdpi
layout
  __listview_main.xml
  activity_main.xml
  add_attandance.xml
  add_student_attendance.xml
  addfaculty.xml
  addstudent.xml
  listview_main.xml
  login.xml
  menu.xml
  test_layout.xml
  test_main.xml
  view_attendance_list.xml
  view_attendance_list_per_student.xml
  view_faculty_list.xml
  view_student_list.xml
  viewstudent.xml
```

**Fig.4.1 Activity and Layout Files**

## 4.3 Description of Main Modules and Code

**1. Home screen** appears as the starting screen with a "start" button and can be navigated to the login screen

### 4.3.1 Main Activity

```
super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      start =(Button)findViewById(R.id.buttonstart);
      start.setOnClickListener(new OnClickListener() {

         @Override
         public void onClick(View v) {
            // TODO Auto-generated method stub

            Intent intent =new Intent(MainActivity.this,LoginActivity.class);
            startActivity(intent);
         }
      });

 }
```

**2. Login Screen**: Login for Admin & Faculty is created with LoginActivity & its corresponding UI is done with “login ”Layout file.

- For Admin username is “admin” and password is “vasavi”.
- Faculty needs to login with username and password provided by admin on adding faculty to database.

### 4.3.2 Login Activity(Code Snippet)

```
    login.setOnClickListener(new OnClickListener() {

      @Override
      public void onClick(View v) {
         // TODO Auto-generated method stub

         if(userrole.equals("admin"))
         {
           String user_name = username.getText().toString();

                String pass_word = password.getText().toString();

               if (TextUtils.isEmpty(user_name))
               {
                  username.setError("Invalid User Name");
               }
               else if(TextUtils.isEmpty(pass_word))
               {
                  password.setError("enter password");
               }
               else
               {
                  if(user_name.equals("admin") &
pass_word.equals("vasavi")){
                  Intent intent =new
Intent(LoginActivity.this,MenuActivity.class);
                  startActivity(intent);
                  Toast.makeText(getApplicationContext(), "Login
successful", Toast.LENGTH_SHORT).show();
                  }else{
                     Toast.makeText(getApplicationContext(), "Login
failed", Toast.LENGTH_SHORT).show();
                  }
               }
            })
```

3. Next the Admin needs to add faculty and students through validated registration forms.

4. On successful entry of the above task, the faculty can login and take attendance.

### 4.3.3 Add Attendance Activity(Code Snippet)

```
for(StudentBean studentBean : studentBeanList)
      {
 String users =
"Name:"+studentBean.getStudent_firstname()+""+studentBean.getStudent_las
tname()+"\nRoll No:"+studentBean.getStudent_id();

         studentList.add(users);
         Log.d("users: ", users);

      }public void onCheckedChanged(RadioGroup group, int checkedId) {
                  if(checkedId == R.id.PresentradioButton) {
                     status = "Present";
                  } else if(checkedId == R.id.AbsentradioButton) {
                     status = "Absent";
                  } else {
                  }
               }
            });
```

5. Now the attendance register can be accessed through the available buttons after faculty logins. View Attendance displays the attendance of all students in the course particular to that day. View Total attendance cumulative attendance of the students in the course.

### 4.3.4 View Attendance Activity

```
for(AttendanceBean attendanceBean : attendanceBeanList)
{
   String users = "";
   if(attendanceBean.getAttendance_session_id() != 0)
   {
      DBAdapter dbAdapter = new
DBAdapter(ViewAttendanceByFacultyActivity.this);
      StudentBean studentBean
=dbAdapter.getStudentById(attendanceBean.getAttendance_student_id());
      users = attendanceBean.getAttendance_student_id()+".
"+studentBean.getStudent_firstname()+"
"+studentBean.getStudent_lastname()+"
"+attendanceBean.getAttendance_status();
   }
   else
   {
      users = attendanceBean.getAttendance_status();
   }

   attendanceList.add(users);
   Log.d("users: ", users);
}
```

6.Admin can view the details of faculty and students and can also delete the data.

### 4.3.5 Database Adapter (code snippet)

- **The database is SQlite Database**

#### Creation of Tables

```
   private static final String DATABASE_NAME = "Attendance";
   private static final String FACULTY_INFO_TABLE = "faculty_table";
   private static final String STUDENT_INFO_TABLE = "student_table";
   private static final String ATTENDANCE_SESSION_TABLE =
   "attendance_session_table";
   private static final String ATTENDANCE_TABLE = "attendance_table";
    public void onCreate(SQLiteDatabase db) {
         String queryFaculty="CREATE TABLE "+ FACULTY_INFO_TABLE +" (" +
         KEY_FACULTY_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
         KEY_FACULTY_FIRSTNAME + " TEXT, " +
         KEY_FACULTY_LASTNAME + " TEXT, " +
         KEY_FACULTY_MO_NO + " TEXT, " +
         KEY_FACULTY_ADDRESS + " TEXT," +
         KEY_FACULTY_USERNAME + " TEXT," +
         KEY_FACULTY_PASSWORD + " TEXT " + ")";
   Log.d("queryFaculty",queryFaculty);
   String queryStudent="CREATE TABLE "+ STUDENT_INFO_TABLE +" (" +
         KEY_STUDENT_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
         KEY_STUDENT_FIRSTNAME + " TEXT, " +
         KEY_STUDENT_LASTNAME + " TEXT, " +
         KEY_STUDENT_MO_NO + " TEXT, " +
         KEY_STUDENT_ADDRESS + " TEXT," +
         KEY_STUDENT_DEPARTMENT + " TEXT," +
         KEY_STUDENT_CLASS + " TEXT " + ")";
   Log.d("queryStudent",queryStudent );
  String queryAttendanceSession="CREATE TABLE "+
ATTENDANCE_SESSION_TABLE +" (" +
         KEY_ATTENDANCE_SESSION_ID + " INTEGER PRIMARY KEY
AUTOINCREMENT, " +
         KEY_ATTENDANCE_SESSION_FACULTY_ID + " INTEGER, " +
         KEY_ATTENDANCE_SESSION_DEPARTMENT + " TEXT, " +
         KEY_ATTENDANCE_SESSION_CLASS + " TEXT, " +
         KEY_ATTENDANCE_SESSION_DATE + " DATE," +
         KEY_ATTENDANCE_SESSION_SUBJECT + " TEXT" + ")";
   Log.d("queryAttendanceSession",queryAttendanceSession );


   String queryAttendance="CREATE TABLE "+ ATTENDANCE_TABLE +" (" +
         KEY_SESSION_ID + " INTEGER, " +
         KEY_ATTENDANCE_STUDENT_ID + " INTEGER, " +
         KEY_ATTENDANCE_STATUS + " TEXT " + ")";
   Log.d("queryAttendance",queryAttendance );


   try
   {
      db.execSQL(queryFaculty);
      db.execSQL(queryStudent);
      db.execSQL(queryAttendanceSession);
      db.execSQL(queryAttendance);
   }
   catch (Exception e) {
      e.printStackTrace();
      Log.e("Exception", e.getMessage());
   }

}
```

### 4.3.6 Code for retrieving students by branch and year

```
public ArrayList<StudentBean> getAllStudentByBranchYear(String
branch,String year)
{
   ArrayList<StudentBean> list = new ArrayList<StudentBean>();

   SQLiteDatabase db = this.getWritableDatabase();
   String query = "SELECT * FROM student_table where
student_department='"+branch+"' and student_class='"+year+"'";
   Cursor cursor = db.rawQuery(query, null);

   if(cursor.moveToFirst())
   {
      do{
         StudentBean studentBean = new StudentBean();

studentBean.setStudent_id(Integer.parseInt(cursor.getString(0)));
         studentBean.setStudent_firstname(cursor.getString(1));
         studentBean.setStudent_lastname(cursor.getString(2));
         studentBean.setStudent_mobilenumber(cursor.getString(3));
         studentBean.setStudent_address(cursor.getString(4));
         studentBean.setStudent_department(cursor.getString(5));
         studentBean.setStudent_class(cursor.getString(6));
         list.add(studentBean);
      }while(cursor.moveToNext());
   }
   return list;
}
```

### 4.3.7 Attendance Session Code

```
public int addAttendanceSession(AttendanceSessionBean
attendanceSessionBean) {
   SQLiteDatabase db = this.getWritableDatabase();

   String query = "INSERT INTO attendance_session_table
(attendance_session_faculty_id,attendance_session_department,attendance_
session_class,attendance_session_date,attendance_session_subject) values
('"+
         attendanceSessionBean.getAttendance_session_faculty_id()+"',
'"+
         attendanceSessionBean.getAttendance_session_department()+"','"+
         attendanceSessionBean.getAttendance_session_class()+"', '"+
         attendanceSessionBean.getAttendance_session_date()+"', '"+
         attendanceSessionBean.getAttendance_session_subject()+"')";
   Log.d("query", query);
   db.execSQL(query);

   String query1= "select max(attendance_session_id) from
attendance_session_table";
```

```
   Cursor cursor = db.rawQuery(query1, null);

   if(cursor.moveToFirst())
   {
      int sessionId = Integer.parseInt(cursor.getString(0));

      return sessionId;
   }


   db.close();
   return 0;
}
```

### 4.3.8 Add Attendance(Code snippet)

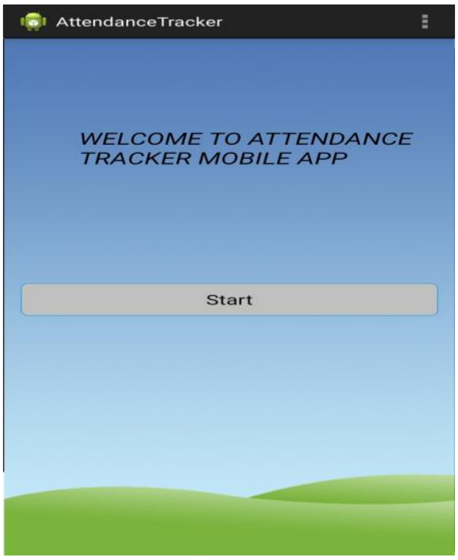
```
public void addNewAttendance(AttendanceBean attendanceBean) {
   SQLiteDatabase db = this.getWritableDatabase();

   String query = "INSERT INTO attendance_table values ("+
         attendanceBean.getAttendance_session_id()+", "+
         attendanceBean.getAttendance_student_id()+", '"+
         attendanceBean.getAttendance_status()+"')";
   Log.d("query", query);
   db.execSQL(query);
   db.close();
}
```

### 4.3.9 GitHub Link

[https://github.com/vivek0006/AttendanceTracker.git](https://github.com/vivek0006/AttendanceTracker.git)

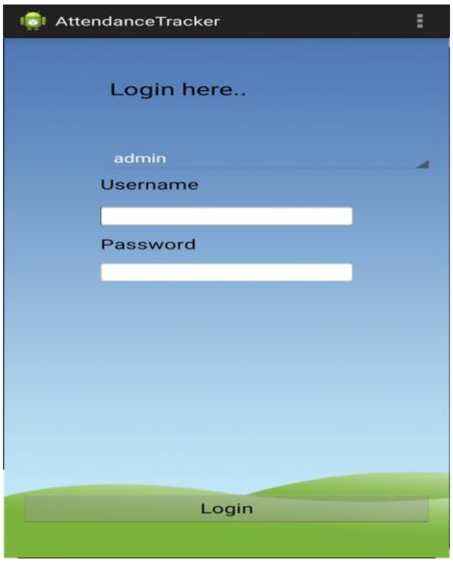
# Chapter 5
# Results

## 5.1 UI SCREENS(OUTPUTS)



WELCOME TO ATTENDANCE TRACKER MOBILE APP

Fig.5.1.HOME SCREEN



Login here..

admin

Username

Password

Fig.5.2.ADMIN LOGIN SCREEN

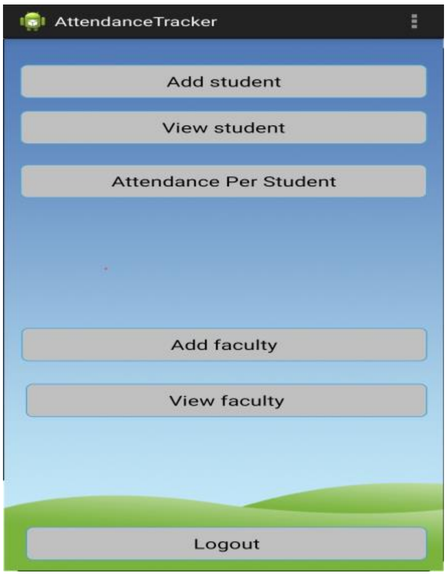
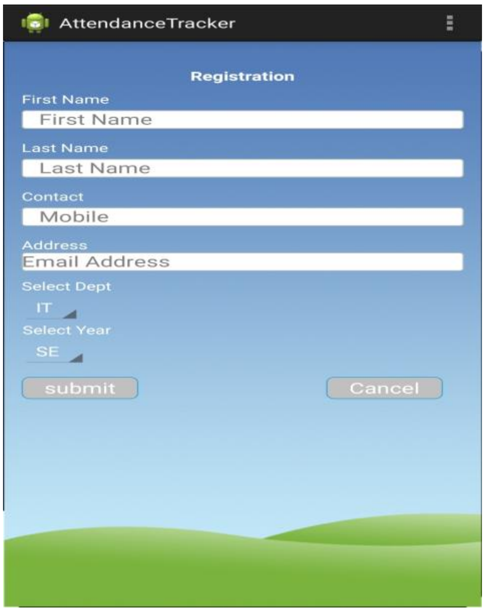


Fig5.3.Admin Login Screen



AttendanceTracker

Registration

First Name
First Name

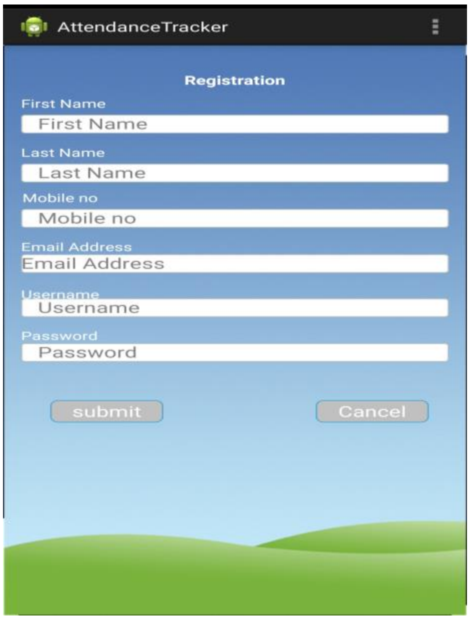
Last Name
Last Name

Contact
Mobile

Address
Email Address

Select Dept
IT

Select Year
SE



AttendanceTracker

Registration

First Name
First Name

Last Name
Last Name

Mobile no
Mobile no

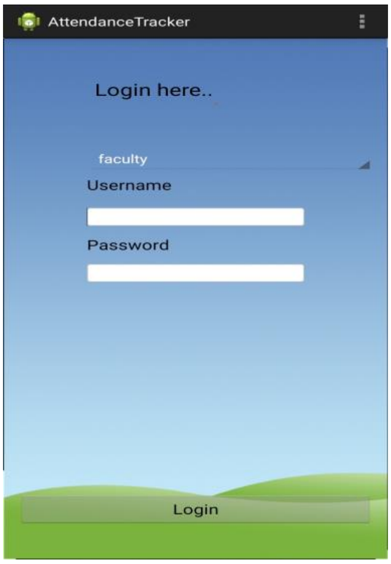
Email Address
Email Address

Username
Username

Password
Password

Fig.5.4.Add Student

Fig.5.5 Add Faculty



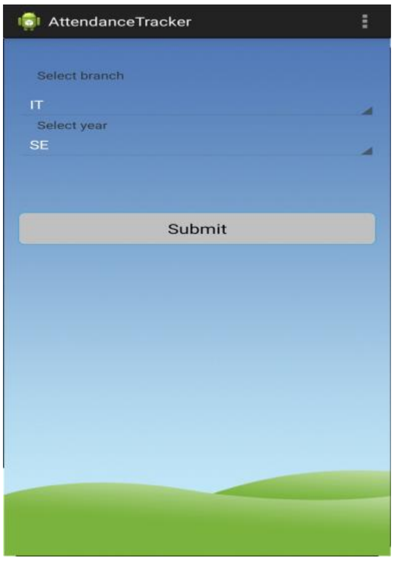
AttendanceTracker

Login here..

faculty

Username

Password



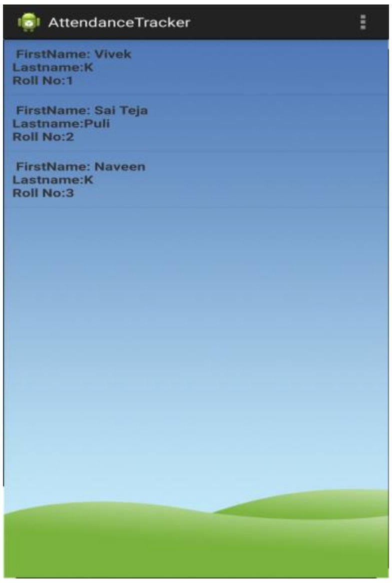
AttendanceTracker

Select branch
IT

Select year
SE

Fig 5.6.FACULTY LOGIN

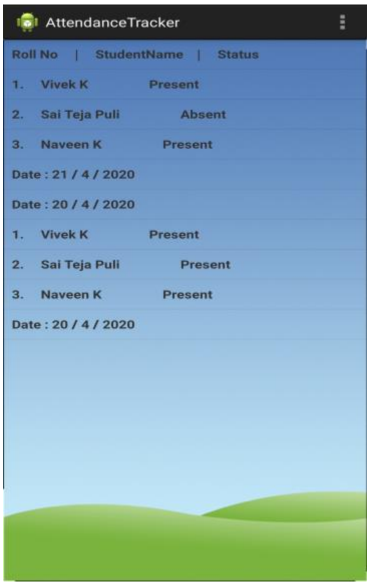
Fig5.7.TAKING ATTENDANCE



FirstName: Vivek
Lastname:K
Roll No:1

FirstName: Sai Teja
Lastname:Puli
Roll No:2

FirstName: Naveen
Lastname:K
Roll No:3



```
Roll No  |  StudentName  |  Status
1.  Vivek K          Present
2.  Sai Teja Puli        Absent
3.  Naveen K          Present
Date : 21 / 4 / 2020
Date : 20 / 4 / 2020
1.  Vivek K          Present
2.  Sai Teja Puli        Present
3.  Naveen K          Present
Date : 20 / 4 / 2020
```

Fig.5.8.View Attendance Fig.5.9 View total Attendance

### 5.2 Results

- Attendance Traker app can be used for taking attendance instead of traditional methods .
- It is efficient as data evaluation, data storing and retrieval is easy and doesn’t need any heavy calculation or method. The UI is simple and easy to understand.
- Attendance recording through the app ensures least error whereas manual traditional system of taking attendance is more prone to error.
- Database is used efficiently.
- Using this app inserting fake data through unknown sources is avoided .
- Mainly as the App is more faculty friendly .Hence proxies can be avoided.
- As the app requires only a smart phone, it is economically feasible and doen’t require other hardware requirements.
- The app has undergone android unit testing and is ready to use
- User doesn’t require any special training to use the app.
- Attendance Tracker App gives the best experience to users.
- Also the app is scalable in almost all android versions.

# Chapter 6

# Testing

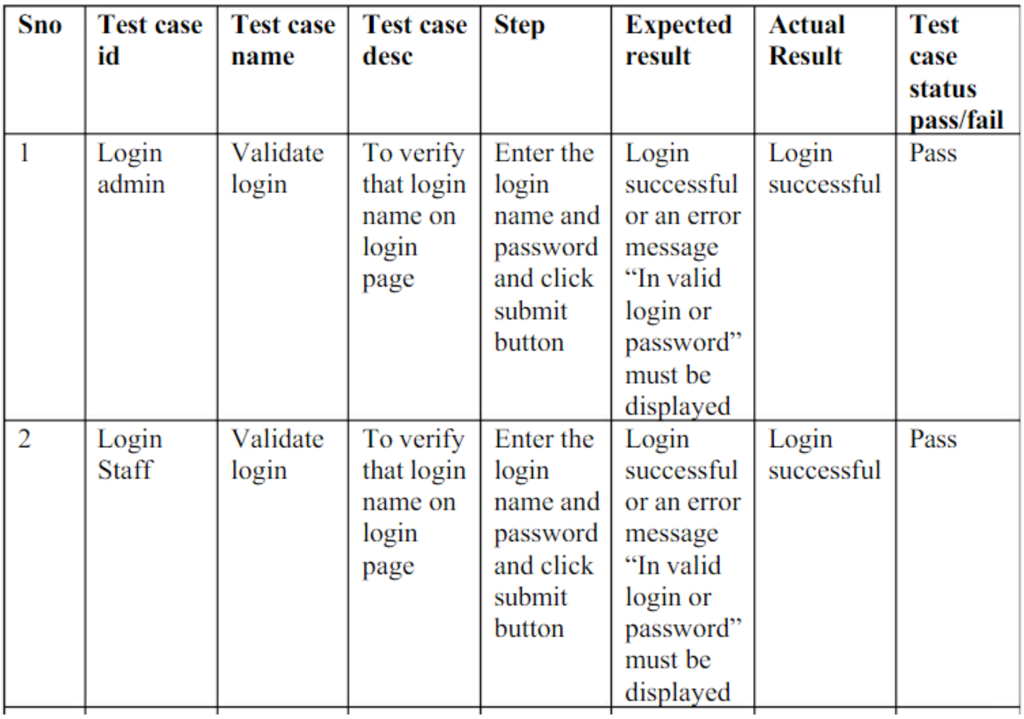
Testing has been done by creating a separate TestActivity.java file and running it as android unit test.

```
public void onClick(View arg0) {
   DBAdapter dbAdapter = new DBAdapter(TestActivity.this);
   AttendanceSessionBean attendanceSessionBean = new
AttendanceSessionBean();


   attendanceSessionBean.setAttendance_session_faculty_id(1);
   attendanceSessionBean.setAttendance_session_department("IT");
   attendanceSessionBean.setAttendance_session_class("FE");
   attendanceSessionBean.setAttendance_session_date("06/04/2020");
   attendanceSessionBean.setAttendance_session_subject("DataBase");

   dbAdapter.addAttendanceSession(attendanceSessionBean);
   Log.d("add", "inserted");
       for(AttendanceSessionBean sessionBean :
attendanceSessionBeanList)
      {
         Log.d("for", "in for loop");
         int aid = sessionBean.getAttendance_session_id();
         int fid = sessionBean.getAttendance_session_faculty_id();
         String sclass = sessionBean.getAttendance_session_class();
         String dept = sessionBean.getAttendance_session_department();
         String date=  sessionBean.getAttendance_session_date();
         String sub= sessionBean.getAttendance_session_subject();
         Log.d("id", aid+"");
         Log.d("fid", fid+"");
         Log.d("sclass", sclass);
         Log.d("dept",dept);
         Log.d("date", date);
         Log.d("sub", sub);
      }
   }
});
```

## 6.1 Test Case

### 6.1.1 Admin and Faculty Login Validation

- **For Admin username is “admin” and password is “vasavi”**



| Sno | Test case id | Test case name | Test case desc | Step | Expected result | Actual Result | Test case status pass/fail |
|---|---|---|---|---|---|---|---|
| 1 | Login admin | Validate login | To verify that login name on login page | Enter the login name and password and click submit button | Login successful or an error message “In valid login or password” must be displayed | Login successful | Pass |
| 2 | Login Staff | Validate login | To verify that login name on login page | Enter the login name and password and click submit button | Login successful or an error message “In valid login or password” must be displayed | Login successful | Pass |

**Table 6.1**

# Chapter 7

# Conclusion & Future Scope

## 7.1 Conclusion

The Attendance Tracker App through mobile devices is a very effective tool which can be used to a great extent. The system is portable and can be easily installed and used on any mobile phones supporting Android OS. The use of this system can result in a reduction of number of hours spent in feeding the attendance details in the attendance registers. It also provides an interface which is easy to understand by the users and greatly helps in adapting to the use of this system.

By this App Attendance marking becomes easy. Less chances of malfunctioning are there. The system has reached a steady state where all bugs have been eliminated. The system is operated at a high level of efficiency and all the teachers and user associated with the system understands its advantage.

## 7.2 Future Scope

In future this system can be implemented to automate most of the educational systems and it can be designed for cross platform. This project is intended to replace the age old system of attendance register with a digital register which can show its worthiness by its features and ease of use. With due course of time we intend it to establish a connection with the college server so as to access and update the attendance over the college LAN. Features like sending warning messages to the students with low attendance, and a student portal to check their attendance are planning to get added. The system can also be enhanced by using voice recognition feature of the Android.

# Chapter 8

## References

1. **http://developer.android.com/**
2. **http://android-developers.blogspot.com/**
3. **Stack Overflow http://stackoverflow.com/**
4. **Android course for beginners https://www.youtube.com/watch?v=aS__9RbCyHg**
5. **Tutorials for learning Android Operating System - http://www.youtube.com/user/thenewboston?feature=watch**
6. **Learn Android SDK from Scratch http://code.tutsplus.com/series/learn-android-sdk-from-scratch--mobile-21677**