VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgaum-590018

A Mini Project Report On

ATTENDANCE MANAGEMENT APP

SUBMITTED IN PARTIAL FULFILMENT FOR 6TH SEMESTER

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

SUMAN PRIYA 1JB19CS140

ULLAS D KANCHAN 1JB19CS151

UNDER THE GUIDANCE OF

Dr. Gopalkrishna M T
Professor
Dept. of CSE, SJBIT

Mrs. Srinidhi K S
Assistant Professor
Dept. of CSE, SJBIT

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SJB INSTITUTE OF TECHNOLOGY

#67, BGS HEALTH & EDUCATION CITY, DR. VISHNUVARDHAN ROAD, KENGERI, BENGALURU-560060, KARNATAKA, INDIA.

2021 - 2022

|| Jai Sri Gurudev ||

Sri Adichunchanagiri Shikshana Trust ®

SJB INSTITUTE OF TECHNOLOGY

#67, BGS Health & Education city, Dr. Vishnuvardhan road, Kengeri, Bengaluru-560060, Karnataka, India.

Department of Computer Science & Engineering

SJBIT

CERTIFICATE

Certified that the Mobile Application Development Mini project work entitled "**ATTENDANCE MANAGEMENT APP**" is a bonafide work carried out by Suman Priya & Ullas D Kanchan and bearing USN **1JB19CS140 & 1JB19CS151** respectively of **SJB Institute of Technology** in partial fulfillment for 6th semester in **COMPUTER SCIENCE AND ENGINEERING** of the **Visvesvaraya Technological University**, **Belagavi** during the academic year **2021-22.** It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The Mini project report has been approved as it satisfies the academic requirements in respect of Mini Project prescribed by the institution.

(Signature of Guide)	(Signature of Guide)	(Signature of HOD)
Dr. Gopalkrishna M T	**Mrs. Srinidhi K S**	**Dr. Krishna A N**
Professor	Assistant Professor	Professor and Head
Dept. of CSE, SJBIT	Dept. of CSE, SJBIT	Dept. of CSE, SJBIT

1. **Internal Examiner:** ______________________

2. **External Examiner:** ______________________

ACKNOWLEDGEMENT

We would like to express our profound thanks to His Divine Soul **Padmabhushan Sri Sri Sri Dr. Balagangadharanatha MahaSwamiji** and His Holiness **Jagadguru Sri Sri Sri Dr. Nirmalanandanatha MahaSwamiji** for providing us an opportunity to pursue our academics in this esteemed institution.

We would also like to express our profound thanks to **Reverend Sri Sri Sri Dr. Prakashnath Swamiji**, Managing Director, SJB Institute of Technology, for his continuous support in providing amenities to carry out this mini project in this admired institution.

We express our gratitude to **Dr. K V Mahendra Prashanth**, Principal, SJB Institute of Technology, for providing an excellent facilities and academic ambience; which have helped us in satisfactory completion of mini project work.

We extend our sincere thanks to **Dr. Krishna A N,** Head of the Department, Computer Science and engineering; for providing an invaluable support throughout the period of mini project work.

We wish to express heartfelt gratitude to our guides, **Dr. Gopalkrishna M T,** Professor**,** Dept of CSE and **Mrs. Srinidhi K S**, Assistant Professor, Dept of CSE for their valuable guidance, suggestions and cheerful encouragement during the entire period of this work.

Finally, We take this opportunity to extend our earnest gratitude and respect to our parents, Teaching & Non-teaching staffs of the department, the library staff and all our friends, who have directly or indirectly supported us during the period of this mini project work.

Regards,

Suman Priya [1JB19CS140]

Ullas D Kanchan [1JB19CS151]

ABSTRACT

Student attendance system is the system of tacking the attendance of the student on basis of presence in class. Successful industries, schools, universities begin by engaging students and making sure that they will come regularly so the attendance rate become very important. The attendance is important because students are more likely to succeed in academics when they attend class consistently. It’s difficult for the lecturer to build students' skills and progress if a large number of students are frequently absent. Because of the advancement of technology today has immersed itself towards education. The presence of technology has reached its maximum of providing sustainable technology towards quality education through delivery and effective learning and smart devices have become a way of life especially in higher education academic fields be able to develop their system into smart attendance.

TABLE OF CONTENTS

List Of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 PROBLEM STATEMENT / AIM

The main aim of developing this Android application is to provide a student attendance management system that helps to track and store the attendance data in real-time. Besides student attendance, as help in tracking attendance of the whole month at once. The system makes daily attendance hassle-free and frees up the teacher's classroom time so that the only focus is on teaching and imparting knowledge to the students.

1.2 SCOPE

The scope of the project is the system on which the software is installed, i.e. the project is developed as a desktop application, and it will work for a particular institute. But later on the project can be modified to operate it online. Online attendance management system enables school administrators to record, manage & compile daily student attendance data. Along with student attendance, this software also allows teachers to generate 100% accurate student attendance reports.

1.3 PROJECT DESCRIPTION

The Application has Six Functions: - Add new class, add new student, view student list, change date, view attendance, view total attendance.

- **Add new list of class:** After opening app teachers can add class with full details like class name and subject name.
- **Add new list of students:** After creating the class the teachers can add full student details like name, roll no
- **View students list:** After adding students' full information. We can view all Student details in the form of list.
- **Change Date:** Teachers can change the date of the attendance to take attendance of previous days or next coming days or update previous attendance.
- **View Attendance:** In view attendance faculty can take attendance who all are present and absent for his classes and update the attendance and date also mentioned in each class.
- **View Total Attendance:** Teachers can see the total number of students present and absent in each class throughout the month.

Chapter 2

HARDWARE AND SOFTWARE REQUIREMENTS

2.1 Hardware Requirements

- Processor: Intel CORE i5 or higher
- Ram: 8GB or higher
- Hard Disk: 1TB or higher

2.2 Software Requirements

- Android Studio
- Internet Connection
- Java
- SQLite

Chapter 3

DESIGN

3.1 Command

In computing, a command is a directive to a computer program to perform a specific task. It may be issued via a command-line interface, such as a shell, or as input to a network service as part of a network protocol, or as an event in a graphical user interface triggered by the user selecting an option in a menu.

3.1.1 Files Used

- AndroidMainfest.xml: - main xml file.
- ClassAdapter.java: - class item recycler view adapter.
- ClassItem.java: - contains single class item.
- DbHelper.java: - contains all database queries.
- MainActivity.java: - main activity launched on start of app.
- MyCalendar.java: - change date.
- MyDialog.java: - dialog box codes.
- SheetActivity.java: - stores attendance of whole month.
- StudentActivity.java: - adding student details.
- StudentAdapter.java: - student item recycler view adapter.
- StudentItem.java: - contains single student item.
- activity_main.xml: - layout of main activity.
- activity_sheet.xml: - layout of sheet list.
- student_item.xml: - layout of student item.
- class_dialog.xml: - layout of a class item.
- toolbar.xml: - top bar of the app.

3.1.2 Description of Function

Sl.No	Functions	Description
1.	TextView	A TextView displays text to the user and optionally allows them to edit it. A TextView is a complete text editor, however the basic class is configured to not allow editing.
2.	EditText	A EditText is an overlay over TextView that configures itself to be editable. The predefined subclass of TextView that includes rich editing capabilities.
3.	Button	In Android, Button represents a push button. A Push buttons can be clicked, or pressed by the user to perform an action.
4.	RadioButton	Radio buttons allow the user to select one option from a set. You should use radio buttons for optional sets that are mutually exclusive if you think that the user needs to see all available options side-by-side. If it's not necessary to show all options side-by-side, use a spinner instead.
5.	Widgets	widget is a small gadget or control of the android application placed on home screen. Widgets can be very handy as they allow you to put your favoriteapplications on your home screen in order to quickly access them.
6.	ImageView	Displays image resources, for example Bitmap or Drawable resources. ImageView is also commonly used to apply tints to an image and handle image scaling.
7.	Hint	android: hint is more like a placeholder that sort of explains what type of input the EditText is asking for. i.e. If an EditText is asking for posting a status on socialmedia, the hint like What's on your mind? will be suitable.
8.	Toast	A toast provides simple feedback about an operation in a small popup. It only fills the amount of space required for the message and the current activity remains visible and interactive. Toasts automatically disappear after a timeout.
9.	Recycle View	Recycler View makes it easy to efficiently display large sets of data. You supply the data and define how each item looks, and the Recycler View library dynamically creates the elements when they're needed.

10.	Scroll View	A Scroll View is a view group that is used to make vertically scrollable views. A scroll view contains a single direct child only.

Chapter 4

IMPLEMENTATION

4.1 CODE:activity_main.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"

  tools:context=".MainActivity">
  <include
    android:id="@+id/toolbarMain"
    layout="@layout/toolbar"/>
  <com.google.android.material.floatingactionbutton.FloatingActionButton

    android:id="@+id/fab_main"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_margin="24dp"
    android:src="@drawable/ic_baseline_add_24"/>

  <androidx.recyclerview.widget.RecyclerView
    android:layout_below="@+id/toolbarMain"
    android:id="@+id/RecyclerView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
</RelativeLayout>
```

4.2 CODE: activity_ main

```
public class MainActivity extends AppCompatActivity { FloatingActionButton fab;
  RecyclerView recyclerView;
  ClassAdapter classAdapter;
  RecyclerView.LayoutManager layoutManager;
  ArrayList<ClassItem> classItems = new ArrayList<>();
```

```
Toolbar toolbar;
DbHelper dbHelper;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    dbHelper = new DbHelper(this);
    fab = findViewById(R.id.fab_main);
    fab.setOnClickListener(v -> showDialog());
    loadData();
    recyclerView = findViewById(R.id.RecyclerView);
    recyclerView.setHasFixedSize(true);
    layoutManager = new LinearLayoutManager(this);
    recyclerView.setLayoutManager(layoutManager);
    classAdapter = new ClassAdapter(this, classItems);
    recyclerView.setAdapter(classAdapter);

    classAdapter.setOnItemClickListener(position -> gotoItemActivity(position));
    setToolBar();

}
private void loadData() {

    Cursor cursor = dbHelper.getClassTable();
    classItems.clear();

    while(cursor.moveToNext()) {

        @SuppressLint("Range") int id = cursor.getInt(cursor.getColumnIndex(DbHelper.C_ID));

        @SuppressLint("Range") String className =
         cursor.getString(cursor.getColumnIndex(DbHelper.CLASS_NAME_KEY));

        @SuppressLint("Range") String subjectName =
         cursor.getString(cursor.getColumnIndex(DbHelper.SUBJECT_NAME_KEY));
        classItems.add(new ClassItem(id, className, subjectName));
    }
}
private void setToolBar() {
    toolbar = findViewById(R.id.toolbar);
    TextView title = toolbar.findViewById(R.id.title_toolbar);
    TextView subtitle = toolbar.findViewById(R.id.subtitle_toolbar);
    ImageButton back = toolbar.findViewById(R.id.back);
```

```
        ImageButton save = toolbar.findViewById(R.id.save);
        title.setText("Attendance App");
        subtitle.setVisibility(View.GONE);
        back.setVisibility(View.INVISIBLE);
        save.setVisibility(View.INVISIBLE);

    }
    private void gotoItemActivity(int position) {
        Intent intent = new Intent(this, StudentActivity.class);
        intent.putExtra("className", classItems.get(position).getClassName());
        intent.putExtra("subjectName", classItems.get(position).getClassName());
        intent.putExtra("position", position);

        intent.putExtra("cid", classItems.get(position).getCid()); startActivity(intent);
    }
    private void showDialog() {
        MyDialog dialog= new MyDialog();

        dialog.show(getSupportFragmentManager(), MyDialog.CLASS_ADD_DIALOG);
        dialog.setListener((className, subjectName)->addClass(className, subjectName));

    }
    private void addClass(String className, String subjectName) {

        long cid = dbHelper.addClass(className, subjectName);
        ClassItem classItem = new ClassItem(cid, className, subjectName);
        classItems.add(classItem);
        classAdapter.notifyDataSetChanged();
    }
    @Override
    public boolean onContextItemSelected(@NonNull MenuItem item) { switch(item.getItemId()) {
            case 0:
                showUpdateDialog(item.getGroupId());
                break;
            case 1:
                deleteClass(item.getGroupId());
        }
        return super.onContextItemSelected(item);
    }
    private void showUpdateDialog(int position) {
        MyDialog dialog = new MyDialog();
        dialog.show(getSupportFragmentManager(), MyDialog.CLASS_UPDATE_DIALOG);
        dialog.setListener((className,subjectName)->   updateClass(position, className,
  subjectName));
```

```
    }

    private void updateClass(int position, String className, String subjectName) {
        dbHelper.updateClass(classItems.get(position).getCid(), className, subjectName);
        classItems.get(position).setClassName(className);
        classItems.get(position).setSubjectName(subjectName);
        classAdapter.notifyItemChanged(position);

    }
    private void deleteClass(int position) {
        dbHelper.deleteClass(classItems.get(position).getCid());
        classItems.remove(position);
        classAdapter.notifyItemRemoved(position);
    }
}
```

4.3 Code: activity_sheet.xml

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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:orientation="vertical"
    tools:context=".SheetActivity">

    <include
        layout="@layout/toolbar"/>
    <ScrollView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:scrollbars="none">

        <HorizontalScrollView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:scrollbars="none">
            <LinearLayout
                android:padding="16dp"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content">
                <TableLayout
                android:id="@+id/tableLayout"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:background="#ffffff"
                android:divider="#000000"
                android:orientation="horizontal"
                android:showDividers="beginning|middle|end"
                android:stretchColumns="*"/>
            </LinearLayout>
          </HorizontalScrollView>
        </ScrollView>
    </LinearLayout>
```

4.4 Code: DbHelper.java

```
    public class DbHelper extends SQLiteOpenHelper
      private static final int VERSION = 1    ;
      //class table
      public static final String CLASS_TABLE_NAME = "CLASS_TABLE"; public static final String
      C_ID = "_CID";
      public static final String CLASS_NAME_KEY = "CLASS_NAME"; public static final String
      SUBJECT_NAME_KEY = "SUBJECT_NAME";
      private static final String CREATE_CLASS_TABLE =
           "CREATE TABLE " + CLASS_TABLE_NAME + "( " + C_ID +
           " INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, " + CLASS_NAME_KEY
           + " TEXT NOT NULL, " + SUBJECT_NAME_KEY + " TEXT NOT NULL, " +
           "UNIQUE (" + CLASS_NAME_KEY + "," + SUBJECT_NAME_KEY + ")"");";
      private static final String DROP_CLASS_TABLE = "DROP TABLE IF EXISTS " +
    CLASS_TABLE_NAME;
      private static final String SELECT_CLASS_TABLE = "SELECT * FROM " +
    CLASS_TABLE_NAME;

      private static final String STUDENT_TABLE_NAME = "STUDENT_TABLE"; public static final
      String S_ID = "_SID";

      public static final String STUDENT_NAME_KEY = "STUDENT_NAME"; public static final
      String STUDENT_ROLL_KEY = "ROLL";


      private static final String CREATE_STUDENT_TABLE = "CREATE TABLE " +
           STUDENT_TABLE_NAME + "( " + S_ID + " INTEGER PRIMARY KEY
    AUTOINCREMENT NOT NULL, " +C_ID + " INTEGER NOT NULL, "
    +STUDENT_NAME_KEY + " TEXT NOT NULL," + STUDENT_ROLL_KEY + " INTEGER, "
```

```
+FOREIGN KEY ( " + C_ID + ") REFERENCES " + CLASS_TABLE_NAME + "(" + C_ID
+ ")" + ");";

   private static final String DROP_STUDENT_TABLE = "DROP TABLE IF EXISTS " +
STUDENT_TABLE_NAME;

   private static final String SELECT_STUDENT_TABLE = "SELECT * FROM " +
STUDENT_TABLE_NAME;

   private static final String STATUS_TABLE_NAME = "STATUS_TABLE"; public static final
   String STATUS_ID = "_STATUS_ID";

   public static final String DATE_KEY = "STATUS_DATE"; public static final String
   STATUS_KEY = "STATUS";
   private static final String CREATE_STATUS_TABLE = "CREATE TABLE " +
STATUS_TABLE_NAME +
         "(" + STATUS_ID + " INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, " +
         S_ID + " INTEGER NOT NULL, " + C_ID + " INTEGER NOT NULL, " + DATE_KEY + "
         DATE NOT NULL, " + STATUS_KEY + " TEXT NOT NULL, " + "UNIQUE (" + S_ID +
         "," + DATE_KEY + ")," +FOREIGN KEY (" + S_ID + ") REFERENCES " +
         zTUDENT_TABLE_NAME + "( " + S_ID + ")," +
         " FOREIGN KEY (" + C_ID + ") REFERENCES " + CLASS_TABLE_NAME + "( " + C_ID
         + ")" + ");";

   private static final String DROP_STATUS_TABLE = "DROP TABLE IF EXISTS
" + STATUS_TABLE_NAME;
   private static final String SELECT_STATUS_TABLE ="SELECT * FROM " +
STATUS_TABLE_NAME;

   public DbHelper(@Nullable Context context) { super(context, "Attendance.db", null, VERSION);
   }
   @Override

   public void onCreate(SQLiteDatabase db) { db.execSQL(CREATE_CLASS_TABLE);
      db.execSQL(CREATE_STUDENT_TABLE); db.execSQL(CREATE_STATUS_TABLE);


   }
   @Override

   public void onUpgrade(SQLiteDatabase db, int i, int i1)
   { try{
         db.execSQL(DROP_CLASS_TABLE);
         db.execSQL(DROP_STUDENT_TABLE);
         db.execSQL(DROP_STATUS_TABLE);
```

```
    }catch(SQLException e) {
      e.printStackTrace();
    }
  }
  long addClass(String className, String subjectName) { SQLiteDatabase database =
    this.getWritableDatabase(); ContentValues values = new ContentValues();
    values.put(CLASS_NAME_KEY, className); values.put(SUBJECT_NAME_KEY,
    subjectName);
    return database.insert(CLASS_TABLE_NAME, null, values);
  }


  Cursor getClassTable() {
    SQLiteDatabase database = this.getReadableDatabase();
    return database.rawQuery(SELECT_CLASS_TABLE, null);
  }
  int deleteClass(long cid) {
    SQLiteDatabase database = this.getReadableDatabase();
    return database.delete(CLASS_TABLE_NAME, C_ID+"=?", new
String[]{String.valueOf(cid)});
  }
  long updateClass(long cid, String className, String subjectName) { SQLiteDatabase database =
    this.getWritableDatabase(); ContentValues values = new ContentValues();
    values.put(CLASS_NAME_KEY, className); values.put(SUBJECT_NAME_KEY,
    subjectName);
    return database.update(CLASS_TABLE_NAME, values, C_ID+"=?", new
String[]{String.valueOf(cid)});
  }
  long addStudent(long cid, int roll, String name)
    SQLiteDatabase database = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(C_ID, cid);
    values.put(STUDENT_NAME_KEY, name);
    values.put(STUDENT_ROLL_KEY, roll);
    return database.insert(STUDENT_TABLE_NAME, null, values);
  }
  Cursor getStudentTable(long cid) {
    SQLiteDatabase database = this.getReadableDatabase();

    return database.query(STUDENT_TABLE_NAME, null, C_ID+"=?", new
String[]{String.valueOf(cid)}, null, null, STUDENT_ROLL_KEY);
  }
  int deleteStudent(long sid) {
    SQLiteDatabase database = this.getReadableDatabase();
```

```
        return database.delete(STUDENT_TABLE_NAME, S_ID+"=?", new
String[]{String.valueOf(sid)});
    }

    long updateStudent(long sid, String name) { SQLiteDatabase database =
        this.getWritableDatabase(); ContentValues values = new ContentValues();
        values.put(STUDENT_NAME_KEY, name);

        return database.update(STUDENT_TABLE_NAME, values, S_ID+"=?", new
String[]{String.valueOf(sid)});

    }
    long addStatus(long sid, long cid, String date, String status) { SQLiteDatabase database =
        this.getWritableDatabase(); ContentValues values = new ContentValues(); values.put(S_ID,sid);

        values.put(C_ID, cid);

        values.put(DATE_KEY,date);

        values.put(STATUS_KEY, status);

        return database.insert(STATUS_TABLE_NAME,null, values);
    }
    long updateStatus(long sid, String date, String status) { SQLiteDatabase database =
        this.getWritableDatabase(); ContentValues values = new ContentValues();
        values.put(STATUS_KEY, status);

        String whereClause = DATE_KEY + "='" + date+ "' AND " + S_ID + "=" + sid; return
        database.update(STATUS_TABLE_NAME, values, whereClause, null);
    }
    @SuppressLint("Range"
    String getStatus(long sid, String date) {
        String status = null;
        SQLiteDatabase database = this.getWritableDatabase();
        String whereClause = DATE_KEY + "='" + date+ "' AND " + S_ID + "=" +sid; Cursor cursor =
        database.query(STATUS_TABLE_NAME, null, whereClause,
      null, null, null,null);
        if(cursor.moveToFirst())
            status = cursor.getString(cursor.getColumnIndex(STATUS_KEY)); return status;
    }
    Cursor getDistinctMonths(long cid) {
        SQLiteDatabase database = this.getReadableDatabase(); return
        database.query(STATUS_TABLE_NAME,new
String[]{DATE_KEY},C_ID+"="+cid, null, "substr("+DATE_KEY+",4,7)", null, null);//01-04-2020

    }

}
```

4.5 Code: ClassAdapter.java

```
package com.example.attend;

import android.content.Context;
import android.view.ContextMenu;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

public class ClassAdapter extends RecyclerView.Adapter<ClassAdapter.ClassViewHolder> {
    ArrayList<ClassItem> classItems;
    Context context;

    private OnItemClickListener onItemClickListener;

    public interface OnItemClickListener {
        void onClick(int position);
    }

    public void setOnItemClickListener(OnItemClickListener onItemClickListener) {
        this.onItemClickListener = onItemClickListener;
    }

    public ClassAdapter(Context context, ArrayList<ClassItem> classItems) {
        this.classItems = classItems;
        this.context = context;
    }

    public static class ClassViewHolder extends RecyclerView.ViewHolder implements
View.OnCreateContextMenuListener{
        TextView className;
        TextView subjectName;
        public ClassViewHolder(@NonNull View itemView, OnItemClickListener onItemClickListener) {
            super(itemView);
            className = itemView.findViewById(R.id.class_tv);
            subjectName = itemView.findViewById(R.id.subject_tv);
            itemView.setOnClickListener(v->onItemClickListener.onClick(getAdapterPosition()));
            itemView.setOnCreateContextMenuListener(this);
        }

        @Override
        public void onCreateContextMenu(ContextMenu menu, View view, ContextMenu.ContextMenuInfo menuInfo) {
            menu.add(getAdapterPosition(), 0, 0, "EDIT");
            menu.add(getAdapterPosition(), 1, 0, "DELETE");
        }
    }

    @NonNull
    @Override
    public ClassAdapter.ClassViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View itemView = LayoutInflater.from(parent.getContext()).inflate(R.layout.class_item, parent, false);
        return new ClassViewHolder(itemView,onItemClickListener);
    }

    @Override
    public void onBindViewHolder(@NonNull ClassAdapter.ClassViewHolder holder, int position) {
        holder.className.setText(classItems.get(position).getClassName());
        holder.subjectName.setText(classItems.get(position).getSubjectName());
    }

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

4.6 Code: SheetActivity.java

```
package com.example.attend;

import androidx.appcompat.app.AppCompatActivity;
import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.util.Log;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;

import java.lang.reflect.Type;
import java.util.Calendar;

public class SheetActivity extends AppCompatActivity {


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

        showTable();

    }

    private void showTable() {
        DbHelper dbHelper = new DbHelper(this);
        TableLayout tableLayout = findViewById(R.id.tableLayout);
        long[] idArray= getIntent().getLongArrayExtra("idArray");
        int[] rollArray = getIntent().getIntArrayExtra("rollArray");
        String[] nameArray = getIntent().getStringArrayExtra("nameArray");
        String month = getIntent().getStringExtra("month");

        int DAY_IN_MONTH = getDayInMonth(month);

        //row setup
        int rowSize = idArray.length + 1;

        TableRow[] rows = new TableRow[rowSize];
        TextView[] roll_tvs = new TextView[rowSize];
        TextView[] name_tvs = new TextView[rowSize];
        TextView[][] status_tvs = new TextView[rowSize][DAY_IN_MONTH+1];

        for(int i = 0; i < rowSize; i++) {
            roll_tvs[i] = new TextView(this);
            name_tvs[i] = new TextView(this);
            for(int j = 1; j <= DAY_IN_MONTH; j++) {
                status_tvs[i][j] = new TextView(this);
            }
        }

        //header
        roll_tvs[0].setText("Roll");
        roll_tvs[0].setTypeface(roll_tvs[0].getTypeface(), Typeface.BOLD);
        name_tvs[0].setText("Name");
        name_tvs[0].setTypeface(name_tvs[0].getTypeface(), Typeface.BOLD);
        for(int i = 1; i <=  DAY_IN_MONTH; i++) {
            status_tvs[0][i].setText(String.valueOf(i));
            status_tvs[0][i].setTypeface(status_tvs[0][i].getTypeface(), Typeface.BOLD);
        }

        for(int i = 1; i < rowSize; i++) {
            roll_tvs[i].setText(String.valueOf(rollArray[i-1]));
            name_tvs[i].setText(nameArray[i-1]);

            for(int j = 1; j <= DAY_IN_MONTH; j++) {
                String day = String.valueOf(j);
                if(day.length() == 1) day = "0"  + day;
                String date = day + "." + month;
                String status = dbHelper.getStatus(idArray[i-1],date);
                status_tvs[i][j].setText(status);
            }
        }

        for(int i = 0; i < rowSize; i++) {
            rows[i] = new TableRow(this);

            if(i%2 == 0)
                rows[i].setBackgroundColor(Color.parseColor("#EEEEEE"));
            else
                rows[i].setBackgroundColor(Color.parseColor("#E4E4E4"));

            roll_tvs[i].setPadding(16, 16, 16, 16);
            name_tvs[i].setPadding(16, 16, 16, 16);

            rows[i].addView(roll_tvs[i]);
            rows[i].addView(name_tvs[i]);

            for(int j = 1; j <= DAY_IN_MONTH; j++) {
                status_tvs[i][j].setPadding(16, 16, 16, 16);
```

```
                rows[i].addView(status_tvs[i][j]);
            }

            tableLayout.addView(rows[i]);
        }
        tableLayout.setShowDividers(TableLayout.SHOW_DIVIDER_MIDDLE);

    }

    private int getDayInMonth(String month) {
        int monthIndex = Integer.valueOf(month.substring(0, 1));
        int year = Integer.valueOf(month.substring(4));

        Calendar calendar = Calendar.getInstance();
        calendar.set(Calendar.MONTH, monthIndex);
        calendar.set(Calendar.YEAR, year);
        return calendar.getActualMaximum(Calendar.DAY_OF_MONTH);
    }
}
```

Chapter 5

Results and Snapshots

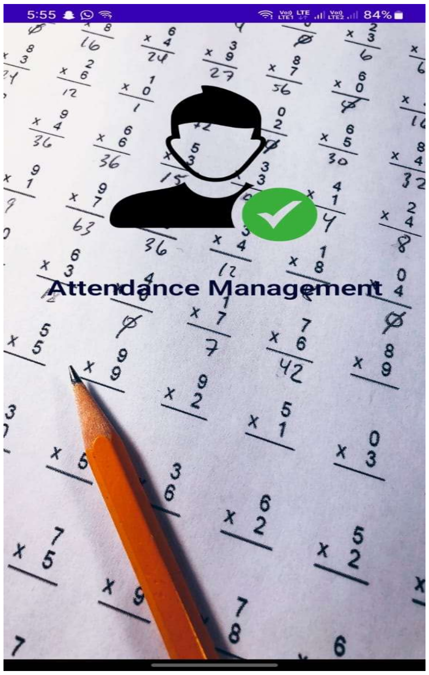
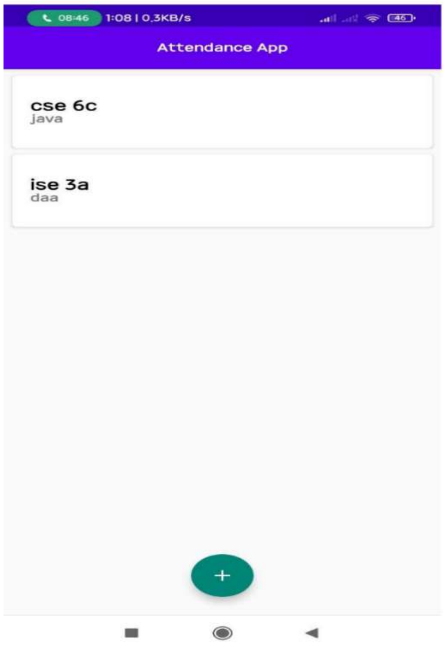


Fig 5.1: Splash Page



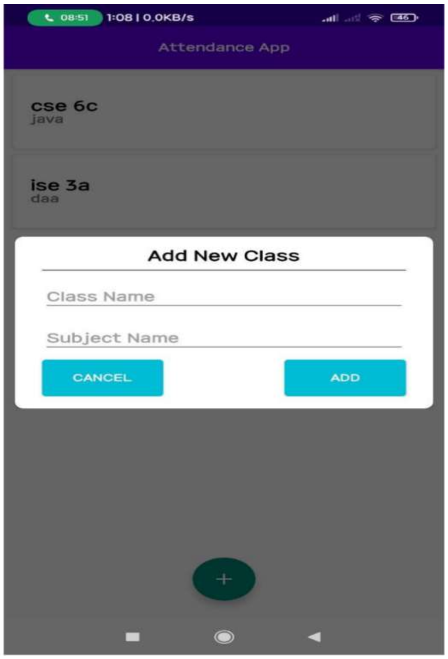
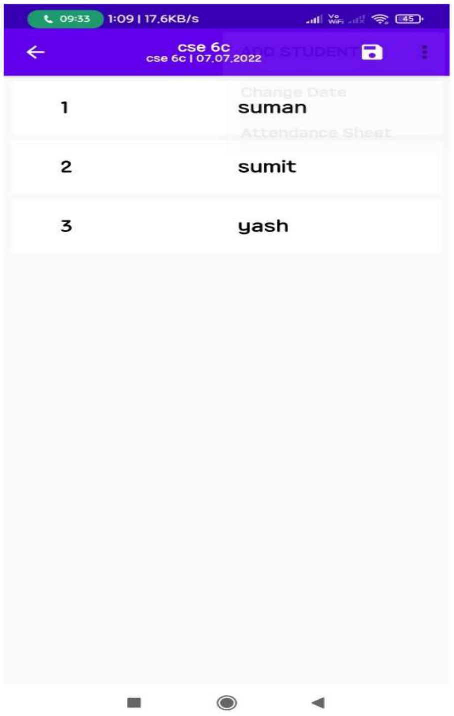


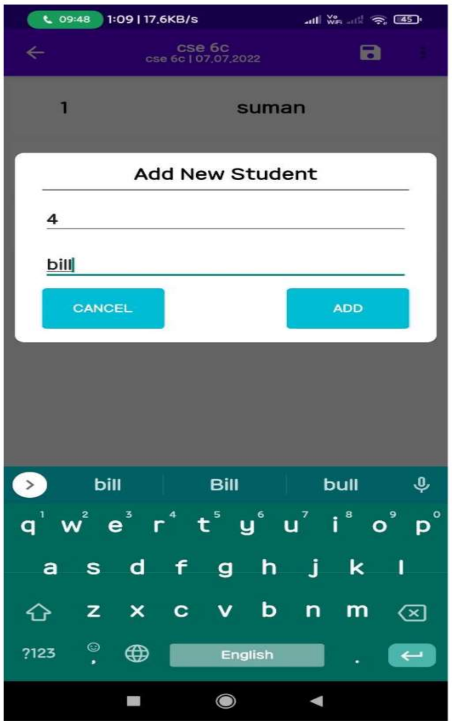
Fig 5.2: Class list page

Fig 5.3: Add class dialog



1	suman
2	sumit
3	yash

Fig 5.4: student list page

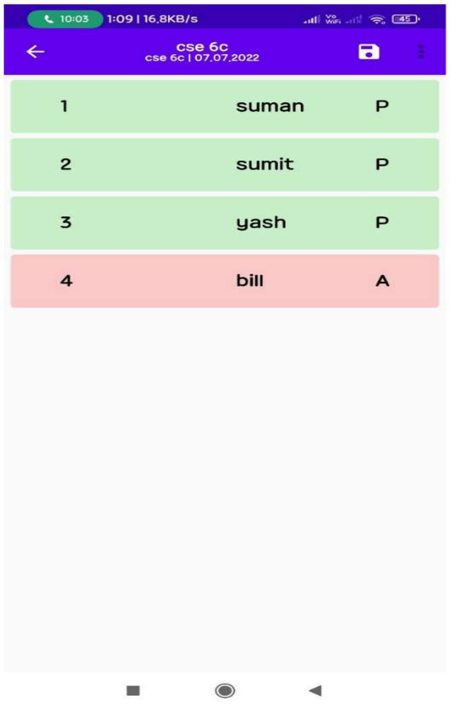


Add New Student

4

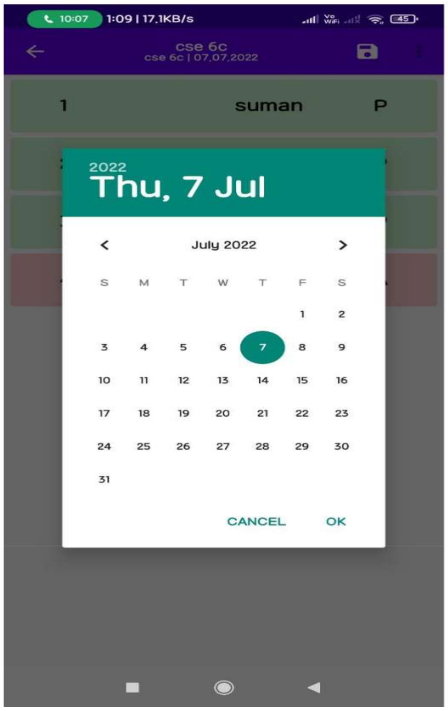
bill

Fig 5.5: add student dialog



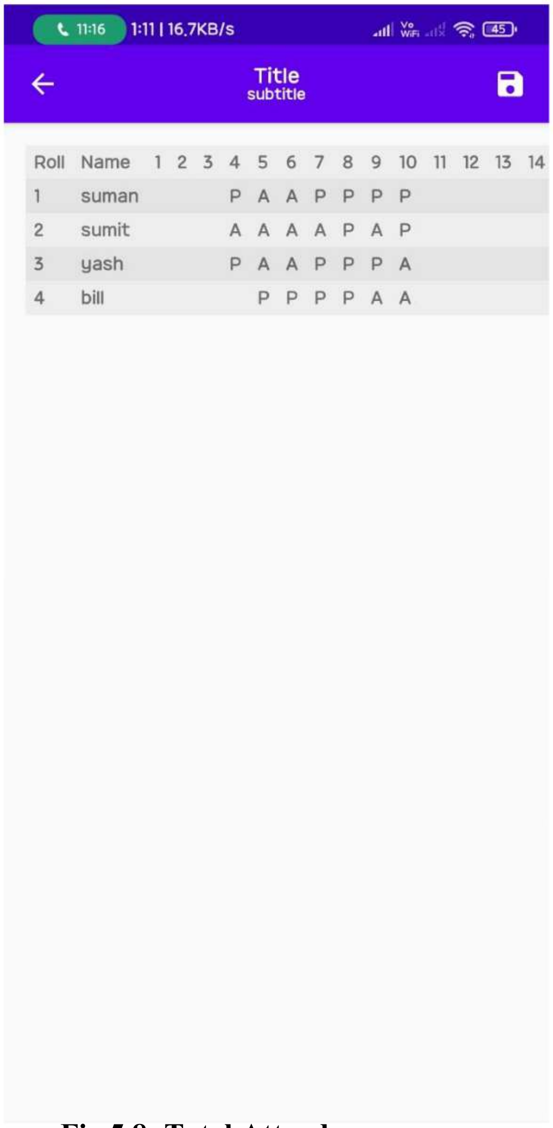
1	suman	P
2	sumit	P
3	yash	P
4	bill	A

Fig 5.6: attendance list page



S	M	T	W	T	F	S
					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	29	30
31						

Fig 5.7: date change page



Roll	Name	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	suman				P	A	A	P	P	P	P				
2	sumit				A	A	A	A	P	A	P				
3	yash				P	A	A	P	P	P	A				
4	bill					P	P	P	P	A	A				

Fig 5.8: Total Attendance page

Chapter 6

CONCLUSION AND FUTURE ENHANCEMENTS

6.1 CONCLUSION

Having identified the process of the existing attendance system and the advantages of the newly developed mobile system; the system in this project proves to be of great use. In conclusion, this project has studied the existing attendance and provided a mobile system to enhance attendance at school and college efficiency. These solutions provide benefits to both the teachers, as well as the students. The numerous advantages would also have a positive impact on the education system of the country.

6.2 FUTURE ENHANCEMENTS

The project has a very vast scope in future. The project can be implemented on internet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of database, the teachers are now able to manage and edit attendance on app can work in a much better, accurate and error free manner. The following are the future scope for the project.

- Adding students record at one place..
- Add student login.
- Adding AI powered GUI.
- Provide more secure authentication

BIBLIOGRAPHY

REFERENCE BOOKS

[1].“Programming Android Java Programming for the New Generation of Mobile Devices” by Zigurd Mennieks

[2].“Android Cookbook” by Ian F Darwin

[3].“Headfirst Android Development: A Brain-Friendly Guide” by David Griffiths

WEBSITE LINKS

[1]. https://www.youtube.com

[2]. https://www.udemy.com

[3]. https://developer.android.com/docs

[4]. https://app.diagrams.net