VISVESWARAYA TECHNOLOGICAL UNIVERSITY
Jnana Sangama, Belagavi-590018

MOBILE APPLICATION DEVELOPMENT (18CSMP68)
MINI PROJECT REPORT ON

"EASY ATTENDANCE APP"

Submitted in partial fulfillment of the requirements for the VI semester of Degree of

BACHELOR OF ENGINEERING
In
COMPUTER SCIENCE AND ENGINEERING
Submitted By

APEKSHA	**(1BY20CS029)**
BHAVANA N S	**(1BY20CS038)**
BRUNDAJA D N	**(1BY20CS041)**

Under The Guidance Of

Prof. BHARATHI R
Associate Professor
Department of CSE
BMSIT&M

Dr. NAGABHUSHAN S V
Associate Professor
Department of CSE
BMSIT&M

BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Avalahalli, Yelahanka, Bengaluru – 560064.

2022-2023

VISVESWARAYA TECHNOLOGICAL UNIVERSITY
Jnana Sangama, Belagavi-590018

BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Avalahalli, Yelahanka, Bengaluru – 560064.

CERTIFICATE

This is to certify that the Mobile Application Development(18CSMP68) Mini Project work entitled **"Easy Attendance App"** has been carried out by **Ms. Apeksha (1BY20CS029)**, **Ms. Bhavana N S (1BY20CS038)**, **Ms. Brundaja D N (1BY20CS041)**, are bonafide students of **BMS Institute of Technology and Management** in partial fulfillment for the award of **Bachelor of Engineering Degree in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi** during the year **2022-2023**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in this report. The Mini Project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the said 6th Semester.

_____	_____
Signature of the Guide	**Signature of the HOD**
Prof. Bharathi R, Associate Professor /	**Dr. Thippeswamy G**
Dr. Nagabhushan S V, Associate Professor	**Professor & Head of Department**
Department of CSE, BMSIT&M	**Department of CSE, BMSIT&M**

EXTERNAL VIVA – VOCE

Name of the Examiners	Signature with Date
1. _____	_____
2. _____	_____

INSTITUTE VISION

To emerge as one of the finest technical institutions of higher learning, to develop engineering professionals who are technically competent, ethical and environment friendly for betterment of the society.

INSTITUTE MISSION

Accomplish stimulating learning environment through high quality academic instruction, innovation and industry-institute interface.

DEPARTMENT VISION

To develop technical professionals acquainted with recent trends and technologies of computer science to serve as valuable resource for the nation/society.

DEPARTMENT MISSION

Facilitating and exposing the students to various learning opportunities through dedicated academic teaching, guidance and monitoring.

PROGRAM EDUCATIONAL OBJECTIVES

1. Lead a successful career by designing, analyzing and solving various problems in the field of Computer Science & Engineering.
2. Pursue higher studies for enduring edification.
3. Exhibit professional and team building attitude along with effective communication.
4. Identify and provide solutions for sustainable environmental development.

PROGRAM SPECIFIC OUTCOMES

1. Analyze the problem and identify computing requirements appropriate to its solution.

2. Apply design and development principles in the construction of software systems of varying complexity.

ACKNOWLEDGEMENT

We are happy to present this Mobile Application Development (18CSMP68) Mini Project after completing it successfully. This Mini project would not have been possible without the guidance, assistance, and suggestions of many individuals. We would like to express our deep sense of gratitude and indebtedness to each and every one who has helped us make this Mini project a success.

We heartily thank our **Principal, Dr. MOHAN BABU G N, BMS Institute of Technology & Management,** for his constant encouragement and inspiration in taking up this Mini project.

We heartily thank our **Head of the Department, Dr. THIPPESWAMY G, Department of Computer Science and Engineering, BMS Institute of Technology &Management,** for his constant encouragement and inspiration in taking up this Mini project.

We gracefully thank our Mini Project Guide, **Prof. BHARATHI R Associate Professor** & **Dr. NAGABHUSHAN S V, Associate Professor, Department of Computer Science and Engineering,** for their guidance, support and advice.

Special thanks to all the staff members of Computer Science Department for their help and kind co-operation.

Lastly, we thank our parents and friends for the support and encouragement given to us in completing this precious work successfully.

Apeksha (1BY20CS029)
Bhavana N S (1BY20CS038)
Brundaja D N (1BY20CS041)

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. With advent of smartphones and tablets which are very handy to use, The Easy Attendance app addresses these challenges by leveraging the power of mobile technology to automate and digitize the entire attendance management process. The development process involved utilizing modern software development tools and technologies such as Android Studio, Java programming language, and database integration. The app follows best practices for user interface design and incorporates responsive layouts to ensure a seamless user experience across different devices.

TABLE OF CONTENTS

ACKNOWLEDGEMENT		**i**
ABSTRACT		**ii**
TABLE OF CONTENTS		**iii**
LIST OF FIGURES		**iv**
CHAPTER NO.	**CHAPTER TITLE**	**PAGE NO.**
CHAPTER 1:	**INTRODUCTION**	**1**
	1.1 Brief Introduction	1
	1.2 Objectives	2
	1.3 Scope	3
	1.4 Problem Statement	3
	1.5 Limitations of Existing System	4
	1.6 Proposed System	5
CHAPTER 2:	**LITERATURE SURVEY**	**6**
CHAPTER 3:	**SYSTEM REQUIRIMENT SPECIFICATIONS**	**8**
	3.1 Software Requirements	8
	3.2 Hardware Requirements	8
	3.3 Functional Requirements	8
	3.4 Non-Functional Requirements	9
CHAPTER 4:	**SYSTEM ANALYSIS**	**11**
	4.1 Implementation Requirement	11
	4.1.1 Front-end	11
	4.1.2 Back-end	13
CHAPTER 5:	**SYSTEM IMPLEMENTATION**	**14**
	5.1 Project Modules	14
	5.2 Project Implementation	15
CHAPTER 6:	**INTERPRETATION OF RESULTS**	**16**
	CONCLUSION	**20**
	FUTURE WORK	**20**
	REFERENCES	**21**

LIST OF FIGURES

Sl. no	Fig. no	Name	Page no
1	1.1	Easy Attendance App Logo	1
2	4.1	Android Studio Logo	11
3	4.2	XML Icon	12
4	4.3	Java Logo	13
5	5.1	System Architecture	14
6	6.1	Home Page	23
7	6.2	Class Creation Page	23
8	6.3	Updated Class	24
9	6.4	Specific Class Page	24
10	6.5	Add Student Page	24
11	6.6	Attendance Marking Page	25
12	6.7	Updated Attendance	25
13	6.8	Daily Attendance Reports	25
14	6.9	Edit Student Details Page	26
15	6.10	Call Student	26
16	6.11	Save Student Deatils	26

CHAPTER 1

INTRODUCTION

1.1 Brief Introduction:

The "EASY ATTENDANCE APP" is a mobile application that is designed to automate the process of tracking and managing attendance on an Android device. Attendance management is an essential function for any educational institution, as it helps to keep track of student attendance and manage their academic schedules. The traditional manual system of attendance management is often time-consuming, prone to errors, and can cause inconvenience to the teachers or faculty members.

To address these challenges, the EASY ATTENDANCE APP is developed, a mobile application that enables teachers to mark their student's attendance using their Android devices. The application is designed to be user-friendly and efficient, with features such as real-time monitoring of attendance, and automated reporting.

The EASY ATTENDANCE APP project aims to develop a mobile application that helps educational institutions to manage their attendance data effectively. The application enables teachers to mark their student's attendance using their Android devices, and the attendance data will be stored securely on the educational institution's machines that are collected from the application. The application also provides real-time monitoring of attendance, automated reporting, and customizable attendance policies.

The EASY ATTENDANCE APP project is developed using modern Android development tools and technologies, including Android Studio, SQLite database, and Java programming language.

The EASY ATTENDANCE APP project is beneficial for educational institutions of all sizes, especially those that require a mobile solution to manage attendance data. The application helps educational institutions to streamline their attendance management process, reduce errors, and improve overall efficiency.

Fig:1.1 Easy Attendance App Logo

1.2 Objectives:

- To develop a mobile application that automates the process of tracking and managing attendance on an Android device, enabling faculties or teaches to mark student's attendance using their Android devices.
- To provide a user-friendly and efficient attendance management system that streamlines the attendance tracking process, reducing errors and improving overall productivity.
- To provide real-time monitoring of attendance and automate the reporting of attendance data, enabling faculty members to make data-driven decisions.
- To provide customizable attendance policies, allowing educational institutions to set attendance rules and parameters that meet their specific requirements.
- To ensure the security and privacy of attendance data by storing it securely on the institution's server.
- To provide a modern and intuitive user interface that is easy to navigate.
- To develop the application using modern Android development tools and technologies, ensuring that the application is compatible with the latest Android devices and operating systems.
- To provide support and maintenance for the application, ensuring that it continues to function correctly and remains up-to-date with the latest technological advancements.

Overall, the objectives of the EASY ATTENDANCE APP project are to provide a valuable tool for educational institutions to manage attendance data efficiently and effectively, improving overall productivity and reducing the associated time and costs.

1.3 Scope:

The scope of easy attendance app includes the following:

- ✓ **Teachers Management:** The system will mange the data of teachers and give access to the teachers as admin.
- ✓ **Students Management:** The system manages the details of students including their name, class, academic year, register number, contact details etc.
- ✓ **Status Management:** The system manages the status of student whether he or she is active or inactive to the classes.
- ✓ **Attendance Management:** The system manages the daily attendance of students and also store this data in database.
- ✓ **Records Management:** The system displays and stores the attendance data of the students in the database securely.
- ✓ **Integration with Student Information Systems (SIS):** Depending on the requirements, the app may integrate with the institution's existing student information system to synchronize attendance data, student records, and other relevant information.
- ✓ **User Interface and User Experience:** The app has a user-friendly interface that is intuitive and easy to navigate for all user roles. It provides a seamless user experience and support multiple devices and screen sizes.
- ✓ **Customization and Configuration:** The app allows customization options to adapt to the specific needs of educational institutions. This can include configuring attendance rules, setting up class schedules, or defining attendance thresholds.
- ✓ **Offline Support:** In environments where internet connectivity may be limited or unreliable, the app may have offline support capabilities, allowing users to take attendance and synchronize data when a connection is available.

1.4 Problem Statement:

The inefficiency and inconvenience associated with traditional methods of manual attendance tracking in educational institutions. Paper-based attendance systems or manual record-keeping methods are prone to errors, time-consuming, and can lead to discrepancies in attendance records. These limitations can hinder accurate attendance monitoring, compromise data integrity, and create challenges in generating timely attendance reports.

Additionally, manual attendance tracking methods often require substantial administrative effort, as teachers need to manually record attendance for each student in every class. This not only consumes valuable class time but also poses challenges in maintaining consistent and reliable attendance records across different teachers and classes.

Furthermore, manual attendance tracking does not provide real-time visibility into attendance data, making it difficult for administrators, teachers, and students to monitor attendance patterns, identify attendance-related issues, and take proactive measures to address them. Lack of timely and accurate attendance information can lead to ineffective decision-making, hamper student engagement, and create barriers in promoting accountability within the educational institution.

1.5 Limitations of Existing System:

- **Time-consuming:** The traditional manual system of attendance management is a time-consuming process, requiring teachers or faculty members to take attendance manually and compile the data into a spreadsheet or database.
- **Prone to errors:** The manual system of attendance management is prone to errors, with mistakes often occurring during the recording of attendance data or the manual entry of data into a spreadsheet or database.
- **Inefficient reporting:** The existing system lacks real-time monitoring and automated reporting capabilities, making it difficult for managers or faculty members to make data-driven decisions based on attendance data.
- **Limited accessibility:** The manual system of attendance management is limited to a physical location, making it difficult for employees or students to mark attendance remotely.
- **Lack of security:** The manual system of attendance management poses security and privacy risks, as attendance sheets or registers can be lost, damaged, or tampered with, compromising the integrity of the attendance data.
- **Difficult to customize:** The existing system lacks customizable attendance policies, making it difficult for organizations or educational institutions to set attendance rules and parameters that meet their specific requirements.

1.6 Proposed System:

The proposed system will have a user-friendly interface that will allow teachers to take attendance by simply tapping on the student's name in the app. It will also have a feature to allow teacher to mark their attendance if they are attending online classes.

The system will generate daily and monthly attendance reports, and that can be stored on excel sheets making it easier for teachers and administrators to track attendance records and identify students with poor attendance. The Easy Attendance App will be a cost-effective and efficient way for educational institutions to manage attendance records and improve student performance. It will help institutions to identify and address attendance issues early on and improve student engagement and academic success.

Advantages of Proposed system:

- **Accuracy:** The proposed system as it is automatic system it uses computerized system and programming which ensures that the system is accurate and reliable.
- **Efficiency:** The system automates the attendance tracking process, reducing the workload for teachers and administrators and allowing them to focus on other important tasks.
- **Accessibility**: The Android-based system can be accessed from an android phone which is now available to approx. every person, making it easy for teachers and administrators to manage attendance records remotely.
- **Time-saving:** The system generates daily and monthly attendance reports, saving time for teachers and administrators and enabling them to quickly identify students with poor attendance.
- **Environment Friendly:** This system is environment friendly as it reduces the use of Pen and Paper.

CHAPTER 2

LITERATURE SURVEY

The development of an Android student attendance app involves incorporating various technologies, methodologies, and best practices from both the mobile application development and educational domains. A literature survey helps to gather insights from existing research and related projects to inform the design and implementation of the app.

Mobile Application Development for Attendance Tracking:

Numerous studies have explored the use of mobile applications for attendance tracking in educational settings. These apps offer advantages such as portability, convenience, and real-time data access.

- Research by Sharma et al. (2016) discussed the development of an Android-based attendance management system using QR codes, which simplified the attendance tracking process.
- Gutiérrez-Velasco et al. (2017) proposed a mobile app that utilized near field communication (NFC) technology to track student attendance. The app offered reliable and secure attendance monitoring.

Attendance Tracking Technologies:

Different technologies and techniques have been employed for attendance tracking. These include QR codes, NFC, Bluetooth, biometric identification, and geolocation.

- Zhang et al. (2016) introduced an attendance tracking system based on Bluetooth Low Energy (BLE) beacons, which facilitated automated and accurate attendance recording.
- In a study by Sánchez-Gordon et al. (2018), a mobile app integrated with fingerprint recognition was developed for student attendance monitoring, enhancing accuracy and security.

User Experience and Interface Design:

User experience (UX) and interface design are crucial for ensuring the app's usability and adoption.

- Research by Kortum and Sorber (2015) highlighted the importance of simplicity, intuitiveness, and efficiency in mobile app interfaces to enhance user satisfaction and engagement.
- Studies on educational app design emphasized the need for clear navigation, visual cues, and user-friendly features to facilitate easy attendance tracking and data access.

Integration with Existing Systems:

Integration with the institution's existing systems, such as student information systems (SIS), can streamline data management and provide a holistic view of student information.

- Studies by Awang Nor and Ahmad (2016) and Dlamini et al. (2019) emphasized the need for seamless integration between attendance tracking apps and SIS to ensure accurate recordkeeping and data synchronization.

By incorporating the findings from existing research, the app can be tailored to meet the specific needs and challenges of attendance tracking in educational institutions.

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATIONS

System requirements are intended to communicate in a precise way, the functions that the system must provide. To reduce ambiguity, they may be written in a structured form of natural language supplemented by tables and system models.

3.1 Software Requirements:

Programming language	:	JAVA and XML
Operating system	:	Windows 10
Front end	:	XML
Application required	:	Android Studio

3.2 Hardware Requirements:

C.P. U	:	Pentium IV 2.4 GHz or above
Memory (Primary)	:	512 MB, 1 GB or above
Output Devices	:	Android smartphone, USB cable
Input Devices	:	Keyboard
Hard Disk	:	40 GB, 80GB, 160GB or above
Monitor	:	15 VGA colour

3.3 Functional Requirements:

Attendance Tracking:

- Provide a user-friendly interface for teachers to create classes.
- Provide a user-friendly interface for teachers to add students to the respective classes
- Allow the teacher or admin to customize the classes.
- Allow the teacher to mark attendance for individual students.
- Allow for easy selection of class sections, subjects, and dates.
- Implement options to mark students as present or absent.

Attendance Reporting:

- Generate attendance reports for teachers, administrators, and students.
- Provide reports for individual students, classes, or specified time periods.
- Include attendance statistics, trends, and summaries.

3.4 Non-Functional Requirements:

Performance:

- Responsiveness: The app should have fast response times to user actions, ensuring a smooth and seamless user experience.
- Scalability: The app should handle a large number of users and attendance records without significant performance degradation.
- Efficient Resource Usage: Optimize memory and battery usage to ensure the app operates efficiently on mobile devices.

Usability:

- Intuitive User Interface: Design the app with a user-friendly interface that is easy to navigate and understand, minimizing the learning curve for users.
- Accessibility: Ensure the app adheres to accessibility standards, allowing users with disabilities to access and use the app effectively.
- Multilingual Support: Provide support for multiple languages to cater to users from diverse linguistic backgrounds.

Security and Privacy:

- Data Protection: Implement encryption techniques to secure sensitive data such as user credentials and attendance records.
- Secure Authentication: Employ robust authentication mechanisms to prevent unauthorized access to the app and protect user accounts.
- Privacy Compliance: Adhere to privacy regulations and best practices to ensure the confidentiality and privacy of user data.

Reliability:

- Availability: Ensure the app is available and accessible to users whenever needed, minimizing downtime or service disruptions.
- Error Handling: Implement effective error handling mechanisms to handle exceptions and provide meaningful error messages to users.
- Data Integrity: Maintain the accuracy and integrity of attendance data by implementing proper data validation and error-checking mechanisms.

CHAPTER 4

SYSTEM ANALYSIS

4.1 Implementation Requirements:

The system implementation involves both front-end and back-end

4.1.1 Front-end:

Android Studio:

Fig:4.1 Android Studio Tool Logo

Android Studio is the official integrated development environment (IDE) for developing Android applications. It is an open-source software created by Google and it offers a range of tools and features for developers to build Android applications efficiently.

Key features of Android Studio:

1. **User Interface Designer:** Android Studio includes a graphical user interface (GUI) designer that allows developers to create UI components such as buttons, text fields, and layouts by dragging and dropping them onto the screen.
2. **Emulator:** Android Studio comes with an emulator that allows developers to test their applications on different virtual devices without needing a physical device. This is especially useful during development and testing phases.
3. **Gradle Build System**: Android Studio uses the Gradle build system to compile and package applications. This system makes it easy for developers to manage dependencies and configure their projects.

4. **Android SDK Manager:** The Android SDK Manager in Android Studio allows developers to download and install the necessary Android SDKs, tools, and libraries for building Android applications. 6. Performance Analysis Tools: Android Studio includes tools for performance analysis, such as the Android Profiler, that help developers identify and fix performance issues in their applications.

XML:

Fig:4.2 XML Document Icon

XML is also widely used in Android development to define user interface layouts and to store application data. In Android, XML files are used to describe the layout and content of user interface elements, such as buttons, text views, and images. These XML files are typically stored in the "res" (resources) directory of an Android application project.

One of the main advantages of using XML for Android UI design is that it separates the presentation of the user interface from the code that controls its behaviour. This makes it easier to modify the user interface without affecting the underlying code, and also allows designers and developers to work on different parts of the application independently.

For example, an Android XML layout file might define a layout with a text view and a button. The XML file specifies the position, size, and appearance of these elements, as well as any text or images that they display. The code that controls the behaviour of the button, such as what happens when it is clicked, is typically implemented in a separate Java file.

4.1.2 Back-end:

Java:

Fig:4.3 Java Logo

Java is one of the primary programming languages used for developing Android applications. Android applications are built using the Java programming language and run on the Android operating system, which is based on the Linux kernel.

Key features of Java for Android development:

1. **Android SDK:** Java developers use the Android SDK (Software Development Kit) to develop Android applications. The SDK provides a set of tools and APIs for building Android applications, including the Android Studio IDE, which is the primary development environment for Android app development.
2. **Java Libraries:** Java developers can use a wide range of Java libraries to build Android applications. These libraries provide a range of functionality for tasks such as network communication, database access, and user interface design.
3. **Android API:** Java developers can use the Android API to interact with the Android operating system and access system-level features such as camera, GPS, and accelerometer.
4. **Android Runtime Environment:** The Android runtime environment, also known as ART, is the virtual machine that runs Java code on Android devices. ART is designed to provide high performance and efficient memory management.
5. **Android Debug Bridge:** The Android Debug Bridge (ADB) is a command-line tool that Java developers can use to communicate with Android devices during development and testing. ADB provides features such as device emulation, locate logging, and package management.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 Project Modules:

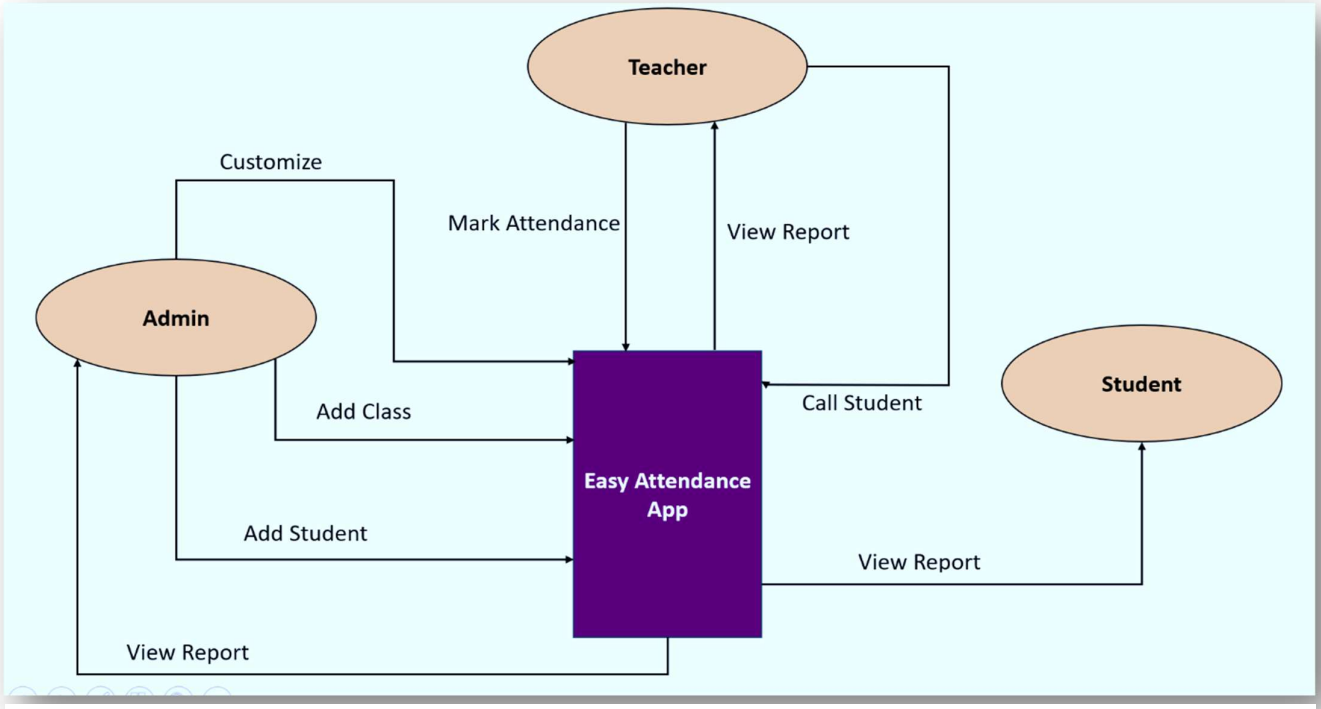

Fig:5.1 Proposed System Architechture Diagram

Admin:

- The admin can first create as many classes as required by entering subject name and the class name (section) and can select the theme as well.
- He can then add the students to the respective classes by giving the details: student name, register number, mobile number.
- He can also view the student attendance reports.

Teacher:

- The teacher can mark the student attendance present or absent by selecting the class.
- Teacher can call the student regarding his/her performance and attendance status based on the daily subject wise attendance report.

Student:

- Student can view the daily attendance status and report.

5.2 Project Implementation:

The proposed system is based on Android Operating system. Android is a Linux-based operating system designed primarily for touch screen mobile devices such as smart phones and tablet computers, developed by Google in conjunction with the Open Handset Alliance. Android was built from the ground-up to enable developers to create compelling mobile applications that take full advantage of all a handset has to offer.

The system is specified on android operating system only because the market share of Android is high. Android also comes with an application development framework, which provides an API for application development and includes services for building Graphical User Interface applications, data access, and other component types. The framework is designed to simplify the reuse and integration of components.

Android apps are built using a mandatory XML manifest file. The manifest file values are bound to the application at compile time. This file provides essential information to an Android platform for managing the life cycle of an application.

Examples of the kinds of information included in a manifest file are descriptions of the app's components among other architectural and configuration properties. Components can be one of the following types: Activities, Services, Broadcast Receivers, and Content Providers.

Code snippets:

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android=http://schemas.android.com/apk/res/android
package="com.ajstudios.easyattendance">
  <application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Easy Attendance"
    android:roundIcon="@mipmap/ic_launcher_round"
```

```
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name="com.ajstudios.easyattendance.Reports_Detail_Activity"
            android:theme="@style/AppTheme2" />
        <activity android:name="com.ajstudios.easyattendance.Reports_Activity" />
        <activity
            android:name="com.ajstudios.easyattendance.ClassDetail_Activity"
            android:theme="@style/FullScreenTheme" />
        <activity
            android:name="com.ajstudios.easyattendance.Insert_class_Activity"
            android:theme="@style/AppTheme2" />
        <activity android:name="com.ajstudios.easyattendance.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

The **manifest file** is an important part of our app because it defines the structure and metadata of our application, its components, and its requirements. This file includes nodes for each of the Activities, Services, Content Providers, and Broadcast Receivers that make the application, and using Intent Filters and Permissions determines how they coordinate with each other.

Insert_class_activity.java

```
public class Insert_class_Activity extends AppCompatActivity {
        Button create_button;
        EditText _className;
        EditText _subjectName;
        Realm realm;
        RealmAsyncTask transaction;
        private String position_bg = "0";
```

```
@SuppressLint("UseCompatLoadingForDrawables")
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_insert_class_);
 Toolbar toolbar = findViewById(R.id.toolbar_insert_class);
 setSupportActionBar(toolbar);
Objects.requireNonNull(getSupportActionBar()).setDisplayHomeAsUpEnabled(true);
 create_button = findViewById(R.id.button_createClass);
 _className = findViewById(R.id.className_createClass);
 _subjectName = findViewById(R.id.subjectName_createClass);
 Realm.init(this);
 realm = Realm.getDefaultInstance();
 final RadioRealButton button1 = (RadioRealButton) findViewById(R.id.button1);
 final RadioRealButton button2 = (RadioRealButton) findViewById(R.id.button2);
 final RadioRealButton button3 = (RadioRealButton) findViewById(R.id.button3);
 final RadioRealButton button4 = (RadioRealButton) findViewById(R.id.button4);
 final RadioRealButton button5 = (RadioRealButton) findViewById(R.id.button5);
 final RadioRealButton button6 = (RadioRealButton) findViewById(R.id.button6);
 RadioRealButtonGroup group = (RadioRealButtonGroup)
findViewById(R.id.group);
 group.setOnClickedButtonPosition(new
RadioRealButtonGroup.OnClickedButtonPosition() {
 @Override
 public void onClickedButtonPosition(int position) {
 position_bg = String.valueOf(position);
 }
 });
 create_button.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View view) {
 if (isValid()) {
 final ProgressDialog progressDialog = new
ProgressDialog(Insert_class_Activity.this);
```

```
progressDialog.setMessage("Creating class..");
 progressDialog.show();
 transaction = realm.executeTransactionAsync(new Realm.Transaction() {
 @Override
 public void execute(Realm realm) {
 Class_Names class_name = realm.createObject(Class_Names.class);
 String id = _className.getText().toString() +
_subjectName.getText().toString();
 class_name.setId(id);
 class_name.setName_class(_className.getText().toString());
 class_name.setName_subject(_subjectName.getText().toString());
 class_name.setPosition_bg(position_bg);
 }
 }, new Realm.Transaction.OnSuccess() {
 @Override
 public void onSuccess() {
 progressDialog.dismiss();
 Toast.makeText(Insert_class_Activity.this, "Successfully created",
Toast.LENGTH_SHORT).show();
 finish();
 }
 }, new Realm.Transaction.OnError() {
 @Override
 public void onError(Throwable error) {
 progressDialog.dismiss();
 Toast.makeText(Insert_class_Activity.this, "Error!",
Toast.LENGTH_SHORT).show();
}
});
 }else{
 Toast.makeText(Insert_class_Activity.this, "Fill all details",
Toast.LENGTH_SHORT).show();
 }
```

```
});
 }
```

The **Insert_class_activity** class plays a vital role in the Easy Attendance App which is responsible for handling the insertion or addition of a new class activity or session into the app's database.

Student_list.java

```
public class Students_List extends RealmObject {
 String id;
 String name_student;
 String regNo_student;
 String mobileNo_student;
 String class_id;
 public String getId() {
 return id;
 }
 public void setId(String id) {
 this.id = id;
 }
 public String getName_student() {
 return name_student;
 }
 public void setName_student(String name_student) {
 this.name_student = name_student;
 }
 public String getRegNo_student() {
 return regNo_student;
 }

 public void setRegNo_student(String regNo_student) {
 this.regNo_student = regNo_student;
}
 public String getClass_id() {
 return class_id;
```

```
 }
 public void setClass_id(String class_id) {
 this.class_id = class_id;
 }
 public String getMobileNo_student() {
 return mobileNo_student;
 }
 public void setMobileNo_student(String mobileNo_student) {
 this.mobileNo_student = mobileNo_student;
 }
}
```

The **Students_List** class is used to store and organize information about the students enrolled in a particular class.

Attendence_report.java

```
public class Attendance_Reports extends RealmObject {
 String date;
 String monthOnly;
 String dateOnly;
 String classId;
 String date_and_classID;
 String classname;
 String subjName;
 RealmList<Attendance_Students_List> attendance_students_lists;
 public String getDate() {
 return date;
 }
 public void setDate(String date) {
this.date = date;
 }
 public String getClassId() {
 return classId;
 }
```

```
 public void setClassId(String classId) {
 this.classId = classId;
 }
 public RealmList<Attendance_Students_List> getAttendance_students_lists() {
 return attendance_students_lists;
 }
 public void setAttendance_students_lists(RealmList<Attendance_Students_List>
attendance_students_lists) {
 this.attendance_students_lists = attendance_students_lists;
 }
 public String getDate_and_classID() {
 return date_and_classID;
 }
 public void setDate_and_classID(String date_and_classID) {
 this.date_and_classID = date_and_classID;
 }
 public String getMonthOnly() {
 return monthOnly;
 }
 public void setMonthOnly(String monthOnly) {
 this.monthOnly = monthOnly;
 }

 public String getDateOnly() {
 return dateOnly;
 }
 public void setDateOnly(String dateOnly) {
 this.dateOnly = dateOnly;
 }
public String getClassname() {
 return classname;
 }
 public void setClassname(String classname) {
 this.classname = classname;
```

```
 }
 public String getSubjName () {
 return subjName;
 }
 public void setSubjName (String subjName) {
 this. subjName = subjName;
 }
}
```

The purpose of the **Attendance_Reports** class is to provide functionalities related to generating attendance reports for the classes within the app. This class interacts with the app's attendance data and student records

CHAPTER 6

INTERPRETATION OF RESULTS

The source code declared for the Easy Attendance App has been tested and it has been found that the source code is correct.

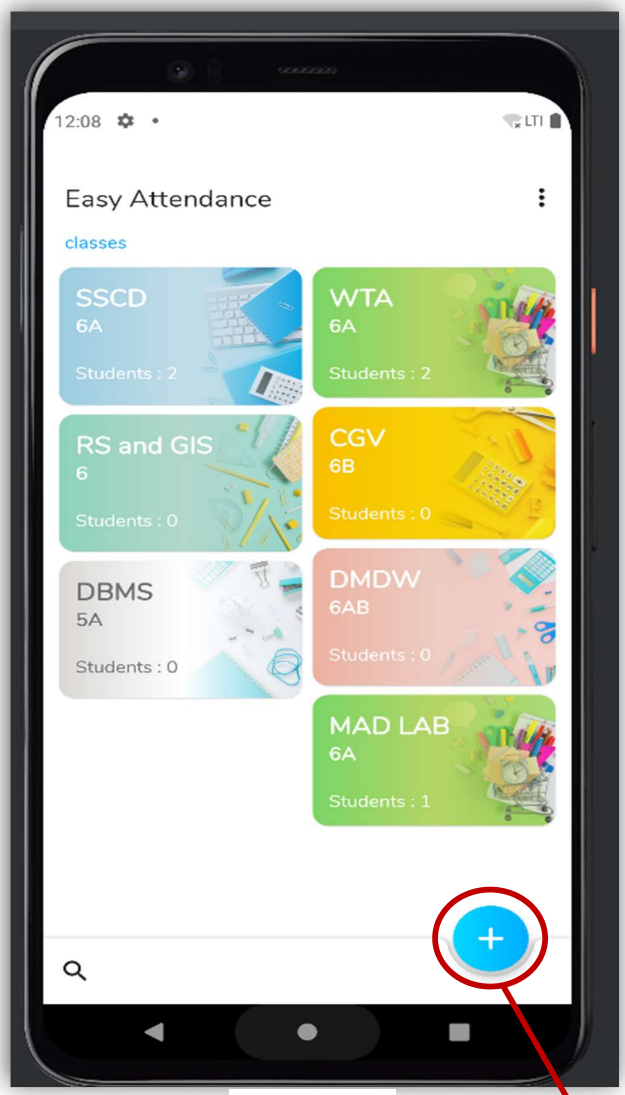

Fig:6.1

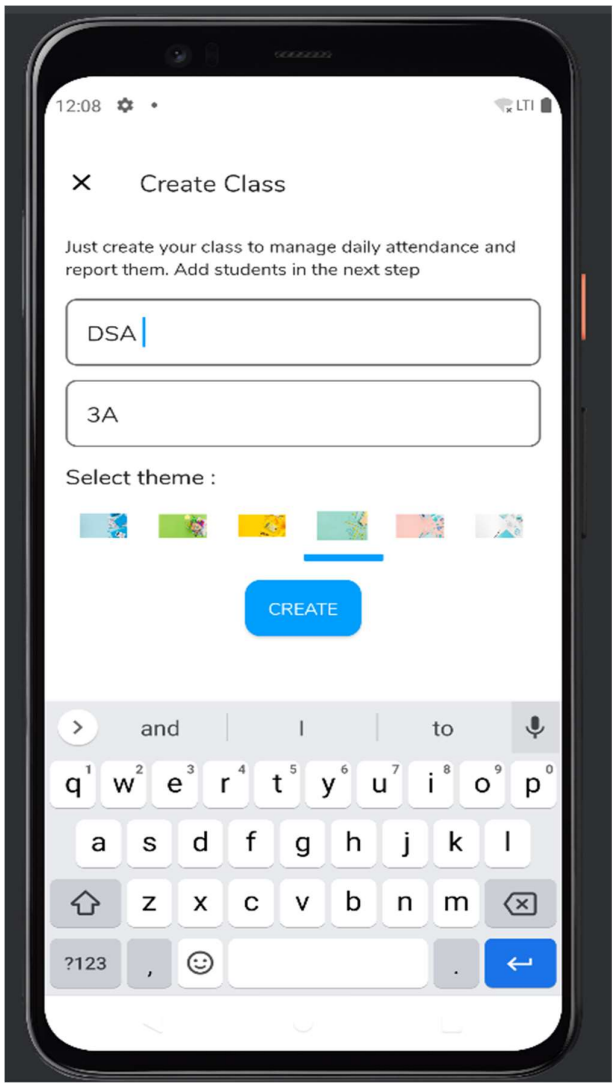

Fig:6.2

The Fig:6.1 is the home page of our project. New class can be created by clicking on the '+' icon on the bottom right corner which takes to the class creation page which is Fig:6.2

Fig:6.3

We can see in the above snapshot Fig:6.3 that the class DSA 3A has been created

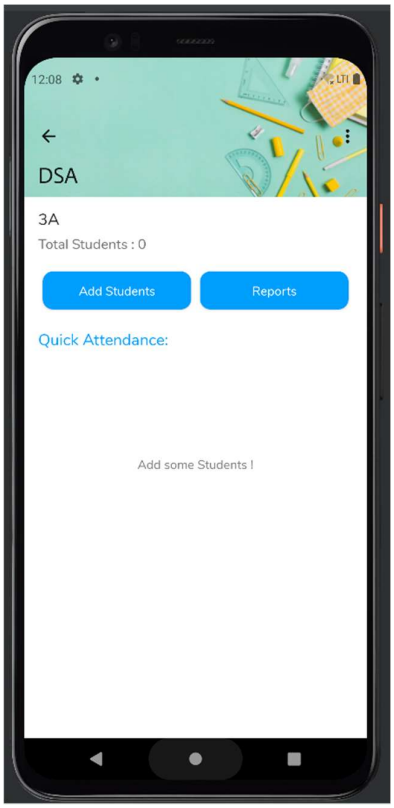

Fig:6.4

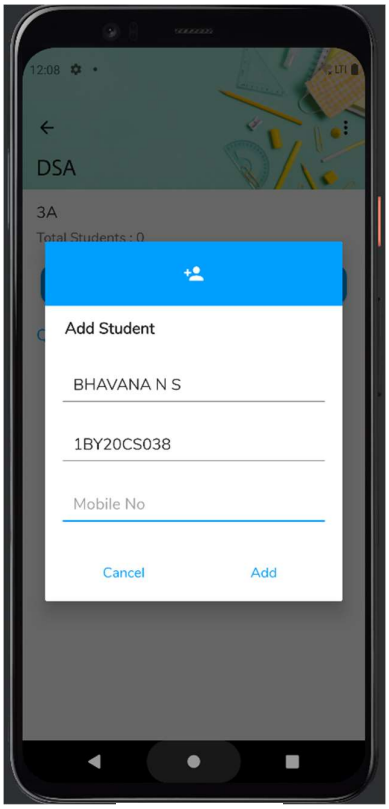

Fig:6.5

We can now navigate into the specific class and can add students by giving the student details: student name, register number, mobile number.

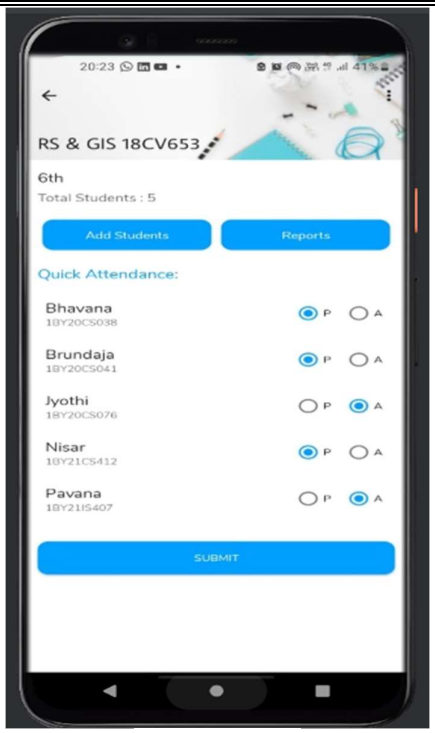

Fig:6.6

Attendance marking page where teacher can take the attendance as P-Present or A-Absent

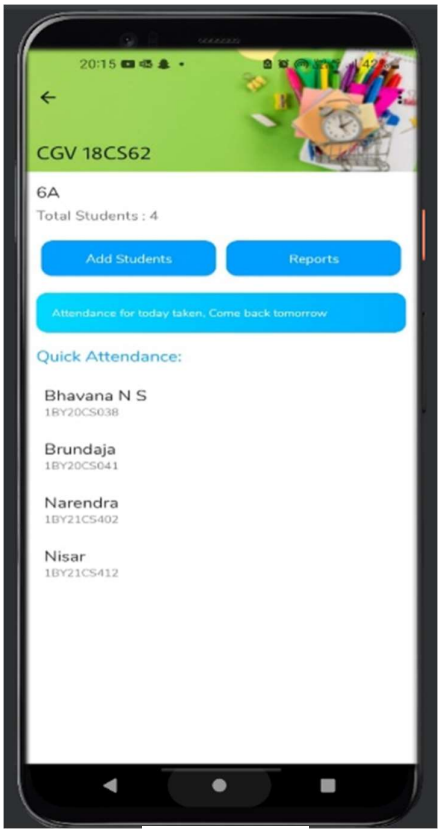

Fig:6.7

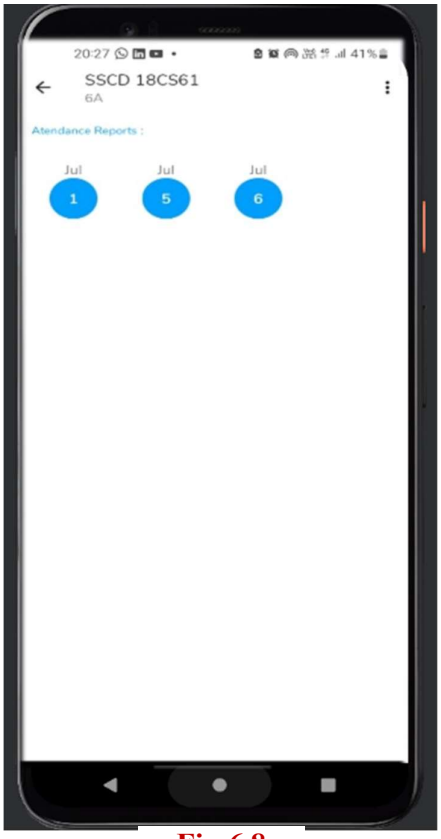

Fig:6.8

The Fig:6.7 shows that the attendance for the class is blocked one saved no changes can be made and the Fig:6.8 shows the day wise attendance taken by the teacher

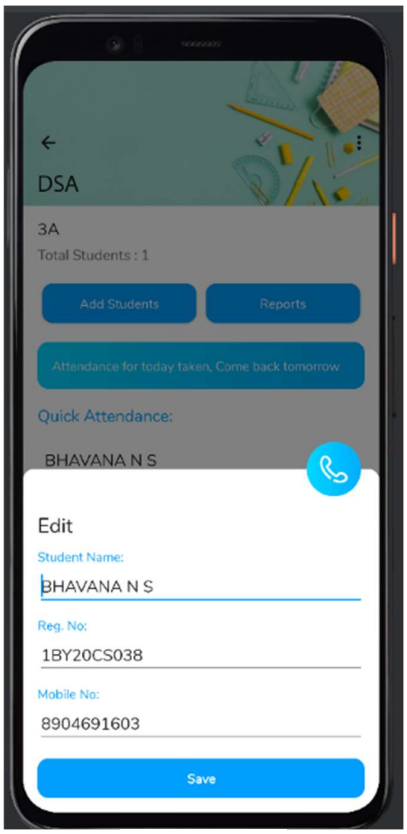

Fig:6.9

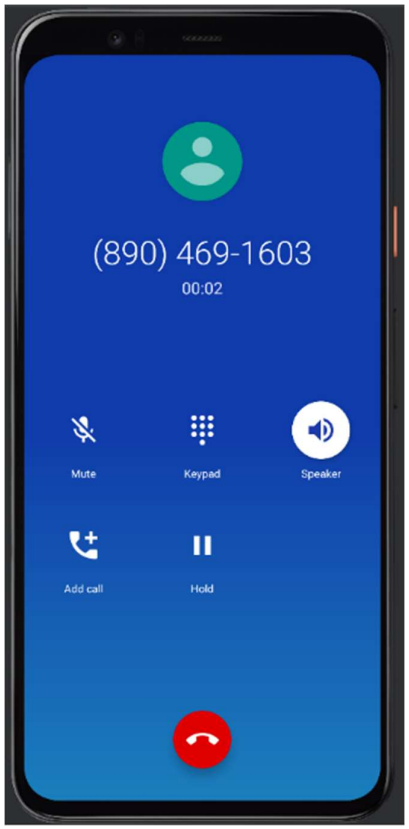

Fig:6.10

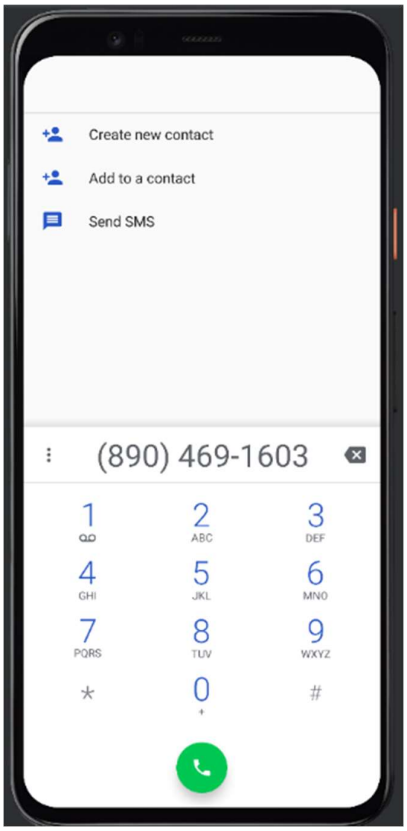

Fig:6.11

The above figures show the calling feature added in our project where the teacher can call the student based on his/her academic performance and attendance status to ensure that they maintain average attendance required.

CONCLUSION

This system can provide more efficient, effective way to manage attendance data and provide more compressive way to store attendance data as it provides a facility to store data in excel sheet. As it has been so much hectic work to be take attendance manually with pen and paper mode and also it causes to be error prone system.

Our system can solve this problem by making it automatic and because of this our system is less error prone compare to the existing system. This application is made using android development tool so that it can be portable and because of it will be used by any teacher who has an android device.

The system provides easy and fast way to take attendance and stores data digitally which make it more reliable, efficient and secure and also it successfully solves the drawbacks of existing system. And it also makes attendance easy to maintain and due to having an option to export attendance data to excel sheet is will make is easy to manage and store the data. And also, it is more reliable, secure and user friendly.

FUTURE ENHANCEMENTS

Our system can be improved in future and it can have some future enhancement that can cause to have more efficient system:

Following are some future enhancements that can be added in future:

- Our system has for now one access i.e., Teacher but in future we can add one more access for student. The student will have access to check its attendance, its dashboard so that he or she can improve their attendance.
- One module can be added that is a dashboard which will display the individual student's attendance percentage and how he or she is improving attendance.
- For now, there is no option for importing and exporting excel sheet to students and attendance data which can be implemented in future.
- In future AI can also be implemented in our project so that teachers can get more effective insights from the student's attendance.
- The SMS service can be added which will send message to the parents if the attendance of their child's drops below the cap of attendance.

REFERENCES

[1] "Mobile Application Development on Android Platform: A Review" by S. Harsoor, H. G. Virani, and S. L. Tadeppalli. (2018)

[2] "Android-Based Mobile Application Development Framework for Energy Efficiency" by F. Umar, A. A. A. Abdullah, and A. A. Fadzil. (2016)

[3] "Mobile Application Development for Field Data Collection Using Android Platform" by M. A. Islam, M. K. Sarker, and M. A. Hossain. (2017)

[4] "Mobile Application Development for E-Learning Based on Android" by R. Kadhi and Z. Bouzid. (2017)

[5] "Challenges in Developing Mobile Applications: A Comparative Study of Android and iOS" by N. K. Kumar and P. K. Verma. (2017)

[6] "User Experience in Mobile Application Development: A Systematic Review" by A. Montano, E. Rodríguez, and M. Á. Redondo. (2020)

[7] https://scholarworks.calstate.edu/downloads/cn69m4196

[8] https://www.researchgate.net/3576_OnlineStudentAttendanceSystemUsingAndroid