

ANDROID ATTENDANCE MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering

By



Dindigalla Preethi Neha (37110192)
Yashasree Marram (37110869)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI – 600 119

MARCH - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **D.Preethi Neha (Reg. No. 37110192)** and **Yashasree Marram(RegnNo:37110869)** who carried out the project entitled "**ANDROID ATTENDANCE MANAGEMENT SYSTEM**" under my supervision from **November 2020** to **March 2021**.

Internal Guide
Dr.V.Nirmalrani, M.E.,Ph.D.,

Head of the Department

Submitted for Viva voce Examination held on _____

InternalExaminer

ExternalExaminer

DECLARATION

I **Dindigalla Preethi Neha** here by declare that the Project Report entitled “**ANDROID ATTENDANCE MANAGEMENT SYSTEM**” is done by meand under the guidance of **Dr.V.Nirmalrani, M.E.,Ph.D.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science andEngineering.

DATE:

PLACE:CHENNAI

SIGNATURE OFTHECANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D.,** and **Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide, **Dr.V.Nirmalrani, M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my projectwork.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The Attendance Management System is a mobile computing software application, which focusses on an activity or function, which is based on management information system of academic institutions. In comparison, with the other traditional attendance systems, the proposed system provides faster, cheaper and reachable system for online student attendance and generate the attendance report automatically. The portability, open source nature of smart phones and android development platform has made the development of application software for various environments as handy. Over the years the manual attendance management has been carried across most of educational institutions.

To overcome the problems of manual attendance, We have developed "Android Based Attendance Management System". Android Based Attendance Management System is based on Android, which can be implemented on any Android Phone. In This application, PHP is server side language, MySQL and PHP is used as back-end design and HTML, CSS and JavaScript are used as front-end tools. The system communicates with database residing on a remote server. It calculates automatically, the attendance percentage of students without any manual paper-based work. Managing the attendance using traditional approach is really a cumbersome process. The person has to be maintain the attendance record in registers and file using pen and paper.

The problem with this approach is that it requires lots of paper which are the part of our non-renewable natural resources. We are in the age, where we have to think about sustainable development. Managing the attendance using mobile phones, provide an alternative way in this direction. Communication between teachers and the parent is also an important issue that should also be considered, because parent can only get the information about their ward after the interaction with the faculty.

TABLE OF CONTENTS

	ABSTRACT	V
	LIST OF FIGURES	Viii
	LIST OF ABBREVIATIONS	IX
CHAPTER No.	TITLE	PAGE No.
1.	INTRODUCTION	1
	1.1 DEFINITION	2
	1.2 PURPOSE	2
	1.3 OVERVIEW	2
2.	LITERATURE SURVEY	4
3.	AIM AND SCOPE OF PRESENT INVESTIGATION	6
	3.1 SCOPE AND OBJECTIVE	6
	3.1.1 OBJECTIVE	6
	3.1.2 SCOPE	6
	3.2 SYSTEM REQUIREMENTS	7
	3.2.1 HARDWARE REQUIREMENTS	7
	3.2.2 SOFTWARE REQUIREMENTS	7
	3.3 LANGUAGE SPECIFICATION	7
	3.4 APPLICATION OF JAVA	7
	3.5 FEATURES OF JAVA	8
	3.5.1 SIMPLE	8
	3.5.2 OBJECT ORIENTED	8
	3.5.3 ROBUST	8
	3.5.4 PLATFORM INDEPENDENT	8
	3.5.5 SECURE	9
	3.5.6 MULTI TREADING	9
	3.5.7 ARCHITECTURAL NEUTRAL	9
	3.5.8 PORTABLE	9
	3.5.9 HIGH PERFORMANCE	10
	3.6 COLLECTION FRAMEWORK	10
	3.7 MYSQL	10
	3.8 FEASIBILITY STUDY	11
	3.8.1 ECONOMICAL FEASIBILITY	12

3.8.2 TECHNICAL FEASIBILITY	12
3.8.3 SOCIAL FEASIBILITY	12

4.	METHODOLOGY AND IMPLEMENTATION	13
	4.1 METHODOLOGY	13
	4.1.1 STEPS TO START WITH AND SET UP	14
	4.2 PROJECT IMPLEMENTATION	15
	4.2.1 ADMIN MODULE	15
	4.2.2 TEACHER MODULE	16
	4.2.3 TAKING ATTENDANCE	16
	4.3 SYSTEM ARCHITECTURE	17
		18
5.	RESULTS AND DISCUSSIONS	18
	OUTPUT SCREENSHOTS	18
6.	CONCLUSION & FUTURE ENHANCEMENT	21
	REFERENCES	22
	APPENDIX	23
	SOURCE CODE	

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
3.6.4	Platform Independent	9
3.7	Collection frame work	10
4.3	System Architecture	17

LIST OF ABBREVIATIONS

ABBREVIATIONS

DEFINITIONS

IDE

INTEGRATED DEVELOPMENT ENVIRONMENT

SDK

SOFTWARE DEVELOPMENT KIT

XML

EXTENSIBLE MARKUP LANGUAGE

CHAPTER 1 INTRODUCTION

Chapter 1: Introduction

Student Attendance Management System app deal with the maintenance of the student's attendance details. It is generates the attendance of the student on basis of presence in class. It is maintaining daily basis of attendance, the staff will be provide with the separate username and password to make student attendance. The staff handling the particular subject to responsible to make the attendance for all students. Only if the student presents the particular date, the attendance will be calculated. The student attendance report based on monthly and consolidate will be generated.

1.1 Definition:-

Attendance Management System app is software developed for daily student attendance in schools, colleges and institutes. It facilitates to access the attendance information of a particular student in a particular class. The information is sorted by the operators, which will be provided by the teacher for a particular class. This system will also help in evaluating attendance eligibility criteria of a student.

Student Attendance Management System app deal with the maintenance of the student's attendance details. It is generates the attendance of the student on basis of presence in class. It is maintaining daily basis of attendance, the staff will be provide with the separate username and password to make student attendance. The staff handling the particular subject to responsible to make the attendance for all students. Only if the student presents the particular date, the attendance will be calculated. The student attendance report based on monthly and consolidate will be generated.

1.2 Purpose:-

The purpose of developing Attendance Management System app is to

computerized the tradition way of taking attendance. Another purpose for developing this software is to generate the report automatically at the end of the session or in the between of the session. The traditional way of attendance

is, recording manually in a log book and then converting into desktop application. Such system can be tedious process and may often lead to errors while generation reports. Apart from this, sometimes the attendance sheets are lost, misplaced or information wrongly entered due to various reasons. In order to reduce the manual work and to achieve more efficiency in managing student's information, a smart phone based mobile application using android can be applied to make this process more easy and beneficial in every way. The proposed system will store, retrieve and sent the student information to the concerned person. In developed an android based mobile application for attendance system. Through this project, teacher can easily record the student attendance and report. Attendance Management System is software developed for daily student attendance in schools, colleges and institutes. It facilitates to access the attendance information of a particular student in a particular class.

1.3 Overview:-

Attendance Management System app basically has two main modules for proper functioning:-

First module is admin which has right for creating space for new batch. Any entry of new faculty, Updating in subject if necessary, and sending notice. Second module is handled by the user which can be a faculty or an operator. User has a right of making daily attendance, generating report. The information is sorted by the operators, which will be provided by the teacher for a particular class.

This system will also help in evaluating attendance eligibility criteria of a student. Android Based Attendance management system using android and Java technology. It consists of four modules such as user authentication process, calling of web service, attendance marking process and display information about students.

In, proposed a framework for maintaining student attendance and also it tracks the illegal use of mobile phones during class hours. Attendance is taken every

day on each working day whether it is college, school or any other institution

and in this process every institution wasted lots of paper just for keeping the record. With such small reasons, our natural resources are depleting at the enormous rate and it can be possible that our descendants don't able to get those resources.

CHAPTER 2 LITERATURE SURVEY

Literature Survey 1

Title	Web Based Student Information Management
Authors	S.R.Bharamagoudar
Year	2017
Description	this paper assist in automating the existing manual system. It can be monitored and controlled remotely. This paper provides accurate information always. All years together gathered information can be saved and can be accessed at any time. The purpose is to design a college website which contains upto date information of the college. That should improve efficiency of college record management.

Literature Survey 2

Title	Attandance Management System
Author	G.Gangagowr
Year	2017
Description	this system is used Way to SMS software. This software is used to send SMS easily to their parent's. This system can store their data about the students and those cares absent student details. It is an efficient method to store the attendance in the Web Site rather than wasting the paper. It also updates the student report directly on the server reducing the faculty's time on logging from the computer.

Literature Survey 3

Title	Online Student Attendance System
Author	P. N. Garad

Year	2016
------	------

Description	in this project, we gave access to three user i.e. Admin, Student, Others. This project is based on client-server. Here, the server is Tomcat and client is JSP. In this project teachers or the admin will be filling attendance and sending message to the student who is absent. They will have privilege to fill attendance form, update attendance form, send message to the guardian's account whose child is absent, also those attendance is less than 75%, and they also have privilege to send message to the students whose fees are pending. The staff can also view the message whenever they want and also can modify the details of students. Parents have privilege to view attendance and to view message sent by the teacher. Students also have their account with the privilege to view message sent by the subject teacher and to view the attendance.
-------------	--

Literature Survey 4

Title	Classroom Attendance Application
Author	Pranjul Kharel
Year	2016
Description	the scope of the project is the system on which the software is installed, i.e. the project is developed as an ANDROID application, and it will work for a particular institute. Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google with a user interface based on direct manipulation. RAD approaches to software development have put less emphasis on planning tasks and more emphasis on development. It has revealed that an online system for recording and reporting students' attendances is indeed a needed application in order to make the process more efficient and time-saving where more than 70% of the sample group agreed to that matter.

Literature Survey 5

Title	Web Based Coaching Institute Management System
Author	Mayuri Kamble
Year	2015

Description	“Coaching Institute Management System” software developed for an institute has been designed to achieve maximum efficiency and reduce the time taken to handle the storing activity. It is designed to replace an existing manual record system thereby reducing time taken for calculations and for storing data. The system is strong enough to withstand regressive daily operations under conditions where the database is maintained and cleared over a certain time of span. The implementation of the system in the organization will considerably reduce data entry, time and also provide readily calculated reports.
-------------	---

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

AIM OF THE PROJECT:

The Main Aim of the project to develop Android Project on Attendance Management System to provide android app on attendance management system to customer, from where user can see it from his mobile device. Android project on Attendance Management System is compatible with all android mobiles. So user can install and configure it on their android mobile.

3.1 SCOPE AND OBJECTIVE:

3.1.1 OBJECTIVE

“Attendance Management System” is software developed for maintaining the attendance of the student on the daily basis in the collage. Here the staffs, who are handling the subjects, will be responsible to mark the attendance of the students. Each staff will be given with a separate username and password based on the subject they handle. An accurate report based on the student attendance is generated here. This system will also help in evaluating attendance eligibility criteria of a student. Report of the student’s attendance on weekly and monthly basis is generated.

- The purpose to develop an android based attendance software is for registering students’ details, teachers and related data.
- The attendance report will be generated automatically without time consuming which is reliable without mistakes.
- To eliminate paper work and save time.
- To increase student academy performance.

3.1.2 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.

3.2 SYSTEM REQUIREMENTS

- Microsoft Windows 7/8/10 (32 or 64 bit).
- Mac OS X 10.8.5.
- GNOME or KDE or Unity desktop on Ubuntu or Fedora or GNU/Linux Debian.
- 2GB RAM.
- 4GB RAM recommended.
- 500 MB disk space
- 1 GB for Android SDK.
- Java Development Kit (JDK) 7.
- 1280x800 screen resolution.
- A faster processor (according to your budget).
- Download and Install Java Development Kit and Android Studio.
- Download the rest of SDK components inside SDK Manager in the Android Studio.

3.2.1. HARDWARE REQUIREMENTS:

System	:	Pentium IV 2.4 GHz.
Hard Disk	:	40 GB.
Floppy Drive	:	1.44 Mb.
Monitor	:	15 VGA Colour.
Mouse	:	Logitech.
Ram	:	512 Mb.
MOBILE	:	ANDROID

3.2.2. SOFTWARE REQUIREMENTS:

Operating system	:	ANDROID.
Coding Language	:	Java 1.6, PHP
Tool Kit	:	Android 2.2
IDE	:	Android Studio

3.3 LANGUAGE SPECIFICATION

Java is one of the world's generally significant and broadly utilized codes, and it has held this differentiation for a long time. Not at all like some other codes whose impact has wearied with entry of time, while Java's has developed.

3.4 APPLICATION OF JAVA

Java is generally utilized in each edge of world and of human existence. Java isn't just utilized in virtual products but at the same time is broadly utilized in planning equipment

controlling programming parts. There are in excess of 930 million JRE downloads every year and 3 billion cell phones run java. Following are some other usage of Java:

1. Developing Desktop Applications
2. Web Applications like Linkedin.com, Snapdeal.com etc
3. Mobile Operating System like Android
4. Embedded Systems
5. Robotics and games etc.

3.5 FEATURES OF JAVA

The excellent explanation for formation of Java was to bring transportability and security include into a scripting language. Close to these two significant highlights there were numerous different highlights that assumed a significant part in embellishment out the last type of this extraordinary language. Those highlights are:

3.5.1 Simple

Java is not difficult to learn and its punctuation is very basic, perfect and straightforward. The confounding and questionable ideas of C++ are either forgotten about in Java or they have been re-executed in a cleaner way. Eg: Pointers and Operator Overloading are not there in java yet were a significant piece of C++.

3.5.2 Object Oriented

In java everything is Object which has some data and behaviour. Java can be easily extended as it is based on Object Model.

3.5.3 Robust

Java puts forth an attempt to take out blunder inclined codes by underlining mostly on order time mistake checking and runtime checking. Yet, the primary territories which Java improved were Memory Management and misused Exceptions by presenting programmed Garbage Collector and Exception Handling

3.5.4 Platform Independent

Dissimilar to other programming dialects, for example, C, C++ and so on which are incorporated into stage explicit machines. Java is destined to be compose once, run-anyplace language. On aggregation Java program is incorporated into byte code. This byte

code is stage autonomous and can be run on any machine, in addition to this byte code design additionally give security. Any machine with Java Runtime Environment can run Java Program.

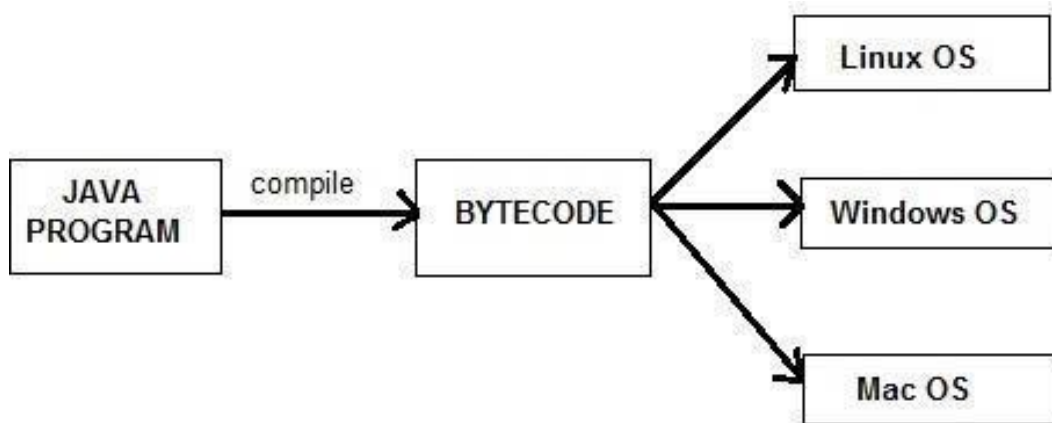


Fig.3.5.4 Platform Independent

3.5.5 Secure

With regards to security, Java is consistently the best option. With java secure highlights it empower us to create infection free, temper free framework. Java program consistently runs in Java runtime climate with practically invalid collaboration with framework OS, consequently it issafer.

3.5.6 Multi-Threading

Java multithreading highlight makes it conceivable to compose program that can do numerous assignments all the while. Advantage of multithreading is that it uses same memory and different assets to execute numerous strings simultaneously, as While composing, syntactic mistakes are checkedalong.

3.5.7 ArchitecturalNeutral

Compiler produces byte codes, which have nothing to do with a specific PC engineering, consequently a Java program is not difficult to decipher on any machine.

3.5.8 Portable

Java Byte code can be conveyed to any stage. No execution subordinate highlights. Everything identified with capacity is predefined, model: size of crude information types.

3.5.9 High Performance

Java is a deciphered language, so it won't ever be pretty much as quick as an incorporated language like C or C++. Be that as it may, Java empowers superior with the utilization of in the nick of time compiler.

3.6 COLLECTIONFRAMEWORK

Assortment system was not piece of unique Java discharge. Assortments was added to J2SE 1.2. Preceding Java 2, Java gave adhoc classes like Dictionary, Vector, Stack and Properties to store and control gatherings of items. Assortment structure gives numerous significant classes and interfaces to gather and sort out gathering of the same items

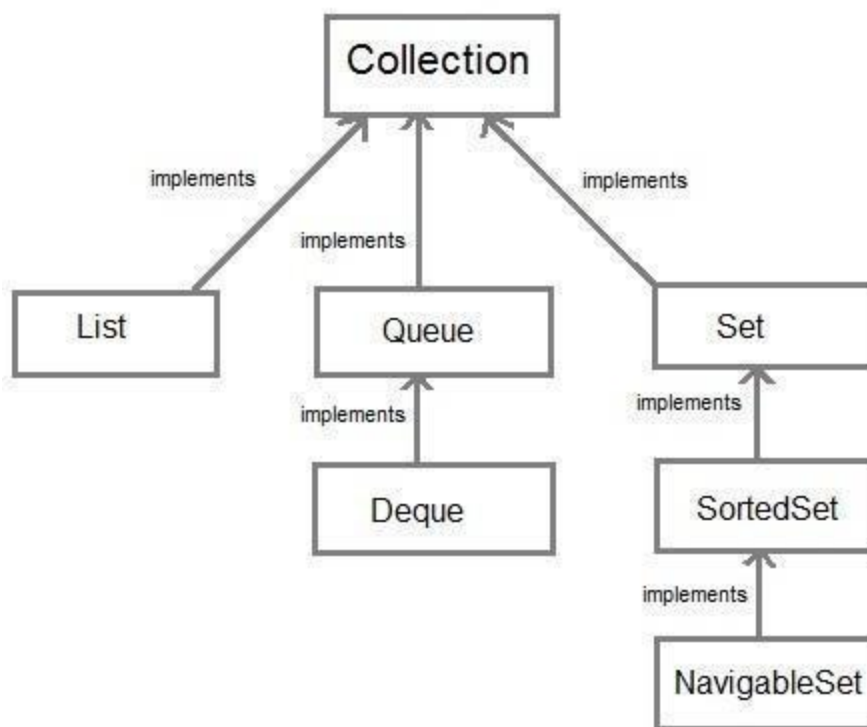


Fig.3.6 COLLECTION FRAMEWORK

3.7 MYSQL

MySQL, authoritatively, yet additionally called "My Sequel" is the world's most broadly utilized open-source social data set administration framework (RDBMS) that runs as a worker giving

multi-client admittance to various information bases, however

SQLite presumably has more complete implanted arrangements. The SQL expression represents Structured Query Language.

The MySQL improvement project has made its source code accessible under the provisions of the GNU General Public License, just as under an assortment of restrictive arrangements. MySQL was possessed and supported by a solitary revenue driven firm, the Swedish organization MySQL AB, presently claimed by Oracle Corporation. MySQL is a famous decision of data set for use in web applications, and is a focal segment of the broadly utilized LAMP open source web application programming stack (and other 'AMP' stacks).

Light is an abbreviation for "Linux, Apache, MySQL, Perl/PHP/Python." Freesoftware- open source projects that require a full-highlighted information base administration framework regularly use MySQL.

For business use, a few paid releases are accessible, and offer extra usefulness. Applications which use MySQL data sets include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal and other programming. MySQL is likewise utilized in some prominent, huge scope sites, including Wikipedia, Google (however not for look), Facebook, Twitter, Flickr and YouTube.

3.8 FEASIBILITY STUDY

The attainability of the undertaking is investigated in this stage and strategic agreement is advanced with an extremely broad arrangement for the task and some quotes. During framework investigation the attainability investigation of the proposed framework is to be completed. This is to guarantee that the proposed framework isn't a weight to the organization. For practicality examination, some comprehension of the significant prerequisites for the framework is fundamental.

The practicality study examines the issue and the data needs of the partners. It looks to decide the assets needed to give a data frameworks arrangement, the expense and advantages of such an answer, and the plausibility of such an answer.

The objective of the achievability study is to consider elective data frameworks arrangements, assess their possibility, and propose the option generally appropriate to the association. The plausibility of a proposed arrangement is assessed regarding its segments.

3.8.1 ECONOMICAL FEASIBILITY

This examination is completed to check the financial effect that the framework will have on the association. The measure of asset that the organization can fill the innovative work of the framework is restricted. The uses must advocated. Along these lines the created framework also inside the spending plan and this was accomplished in light of the fact that the majority of the innovations utilized are uninhibitedly accessible. Just the tweaked items must be bought.

3.8.2 TECHNICAL FEASIBILITY

This examination is done to check the specialized attainability, that is, the specialized prerequisites of the framework. Any framework created should not have an appeal on the accessible specialized assets. This will prompt high requests on the accessible specialized assets. This will prompt high requests being set on the customer. The created framework should have an unobtrusive prerequisite, as just negligible or invalid changes are needed for executing this framework.

3.8.3 SOCIAL FEASIBILITY

The part of study is to check the degree of acknowledgment of the framework by the client. This incorporates the way toward preparing the client to utilize the framework proficiently. The client should not feel undermined by the framework, rather should acknowledge it as a need.

CHAPTER 4

METHODOLOGY AND IMPLEMENTATION

4.1. METHODOLOGY

Our project on Attendance Management System is based on some concepts and methodologies that we devised during our initial study and visualization of the project. We have planned our project keeping in mind the Object Oriented Concepts that can be applied to the project. The project is strongly supported at the back end by Java-the most popular and successful object oriented programming language. The front end is developed using Extensible Mark-up Language (XML). For the database connectivity we are using SQLite database. It is a light weight database which is pre-installed with every android device and is capable of handling and managing various queries, cursors and more. The initial idea was to improve the attendance system which is being in use for the past decades. The idea was to develop an application which would be portable enough so as to carry it to the classrooms as an attendance register. This stuck us with an idea of developing an android application which can be easily installed in an Android enabled cell-phone and be used seamlessly by a teacher while taking the roll calls. In this process we designed a database for the application. It contains various tables like register, subjects, students and attendance for various purposes. These tables hold data for particular purposes which shall be dealt later. The flow includes the addition of subjects by the teacher and then adding the students to the respective subjects. When a subject becomes ready it is then visible in the selection area for giving attendance, checking the percentage and even checking the date wise attendance. The attendance can be checked periodically, date wise or as desired by the teacher. The final attendance can then be used for various purposes.

The initial step is to make another endeavor by any name then as the records get stacked and as we realize that Gradle is fabricated, we'll have `activity_main.xml` and `MainActivity.java` and here we have `MainActivity` which will be perceived as the main homepage screen with two-buttons.

The first activity contains Android Grid Layout which helps the user to easily find the content they are looking i.e we have to make the User Interface of our App. The User Interface will be exceptionally major and simple to utilize.

After UI is complete, move forward toward the coding part. In MainActivity.java class, Intent is utilized to explore users' decisions. An Intent is a class that is utilized to depict an activity to be performed. It is an essential way that assists engineers with beginning another movement inside an application. It can likewise be utilized as a device to impart between activities.

For all activities, we should declare the intent filters in the manifest file.

Next, it's time for setting up (here setting means coding, declare, initialize) all methods, listeners, and buttons needed in all the pages. Like for cpga.java, Scheduler.java, etc. , this means to get done with all the coding for each and every page.

Android is an O.S. which provides its own framework. Android SDK provides the tools and APIs necessary to develop applications on Android Platform using Java programming language.

This application is compatible with the Android Version of 8.0.0 Oreo.

4.1.1 STEPS TO START WITH AND SET UP

As it becomes impossible to write a sentence without being able to write words properly and with correct grammar and punctuation marks, hence a pre-requisite is essential in every scenario. Our project is no different when it comes to pre-requisites. It requires the user to create a csv file which would contain the student details in the following format (roll no., stream, year, sem, sec, name). The csv file can easily be created if we have an MS-Excel file of the same format which can easily be saved to csv format of comma separated delimiter. The saved csv file containing the student details can then be kept back to the Android device which would be used when we pick the students for the particular subject. This can be considered as one of the basic pre-requisite of our project. After the application has been installed in the android device the following steps are to be implemented: -

1. Navigation screen appears for the teacher registration and login, after the splash screen has been navigated to the present screen.
2. The teacher shall then register with a proper user name and password which they

must remember for further login.

3. Next the teacher needs to login from the login or register screen, by pressing the login button and providing the correct credentials without which they would not be able

to log-in to the application.

4. After successful login the teacher would be navigated to the “Select or Add SubjScreen”, from where they must first add the corresponding subjects which they desire to teach by clicking the “ADD SUBJECT” button at the bottom of the screen.

5. The “Add Subject” screen requires three parameters namely: - subject name, subject code and the corresponding section where they must teach. A “BROWSE” button is provided below to choose the students for the section and finally a ‘DONE” button to save the changes made.

6. On successful entry of the above task, the entered subject is seen in every screen namely –“Select or Add Subject”, “TotalAttendance” and “DatewiseAttendance”.

7. Now the attendance register can be accessed from the “Select or Add Subject Screen” for the previously added subjects.

8. Now accordingly attendance can be given for the present date or any previous date, or can be checked in the below two options of “Total Attendance” and “Date-wise Attendance”.

4.2. PROJECT IMPLEMENTATION

Modules of Student Attendance Management System

Login:

- After registration one can log in the system as the operator of the system on the behalf of user. The user will get only those privileges which are given to the user for which one has registered.
- For example, if a user has registered as a student then the user only has the privileges to view the data and cannot make any changes to the data that is shown.

4.2.1 ADMIN MODULE

- Add Student.
- Add Faculty.
- Add Teacher.

- View Student.
- View Teacher.
- View Student Attendance.
- View each student's attendance separately.

4.2.2 TEACHER MODULE

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

4.2.3 Taking Attendance:

- This module is only present in the teachers' interface but not in the students' interface. Teachers can mark attendance by selecting the section they want to take attendance for. Teachers can also rectify the attendance of a student if it is marked incorrect by chance of any student.

Our proposed system is divided into four distinct modules described as follows:

1. User authentication :

Initially, when the teacher runs the application for the first time, a login screen will be displayed that will prompt the teacher to enter the username and password required for authentication.

The teacher will be provided with a unique username which would be a combination of alphanumeric characters. Only when the teacher enters the correct username and password, a "success" message will be displayed and the teacher will get authenticated and directed to the next screen.

2. Calling of Web Service :

In this module, the teacher will need to select details such as the name of the subject for which the lecture is being taken, time of lecture and the particular semester for which the lecture is conducted. After doing so, the teacher needs to call the web service by clicking a button provided on the screen. The web service thus invoked would return the list of names of all the students belonging to a particular semester and branch as per the input provided.

3. Marking Attendance:

After the list of students has been displayed the teacher needs to begin the process of marking the attendance of students. For this purpose, our application would be

providing checkboxes against each student's name that will allow the teacher to mark the student either present or absent. Accordingly, the details of the student will be sent to the remote database and the attendance will be marked for that particular day.

4.Display information of student

: Once the attendance has been marked successfully, the teacher can anytime check the attendance record of a particular student by entering the unique roll number provided to every student. The information thus displayed would include the student's attendance percentage, number of lectures a student has attended for a particular subject, number of lectures missed, as well as the overall attendance.

4.3. SYSTEM ARCHITECTURE

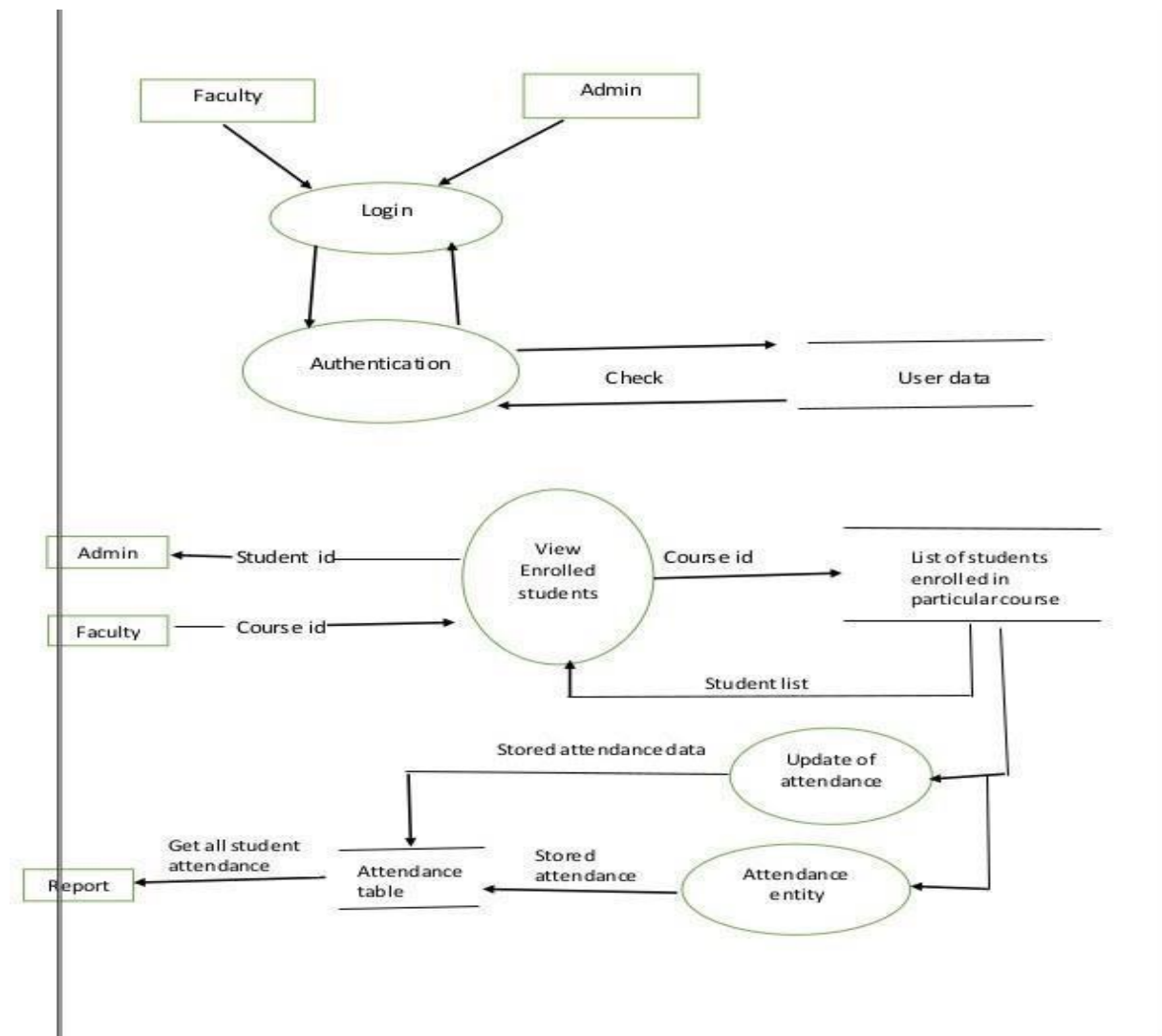
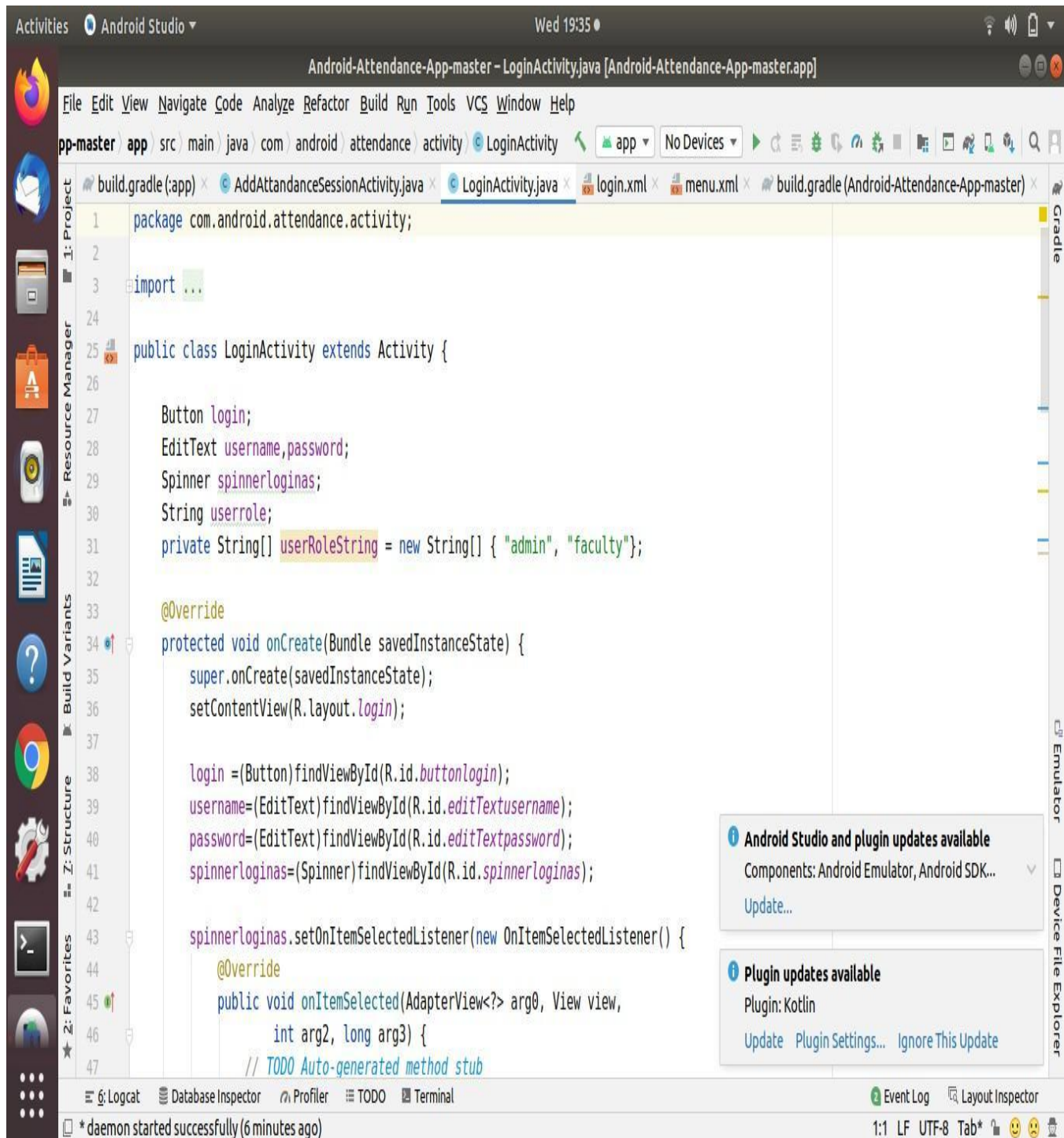


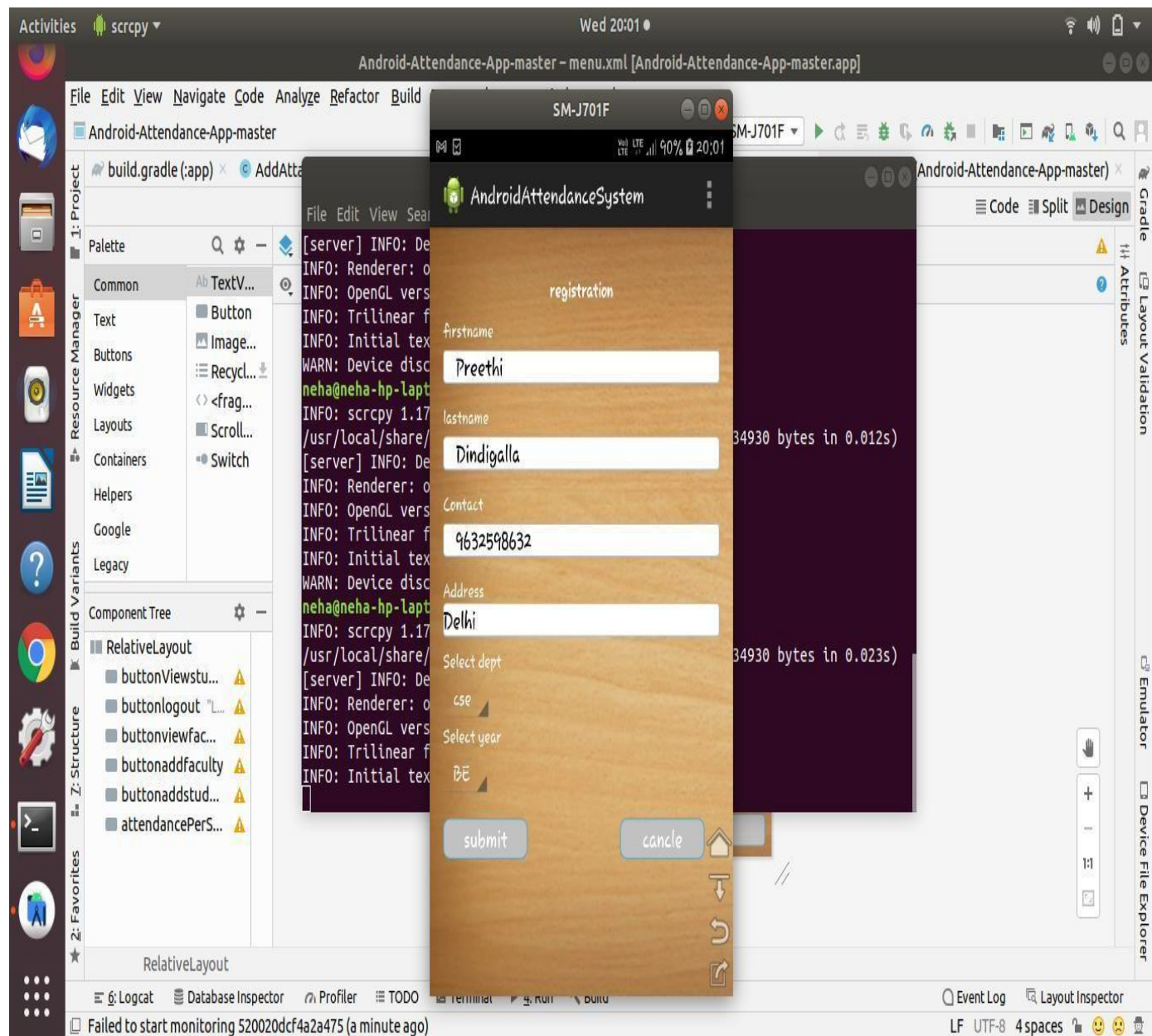
Fig.4.3. System Architecture

5.RESULTS AND DISCUSSIONS

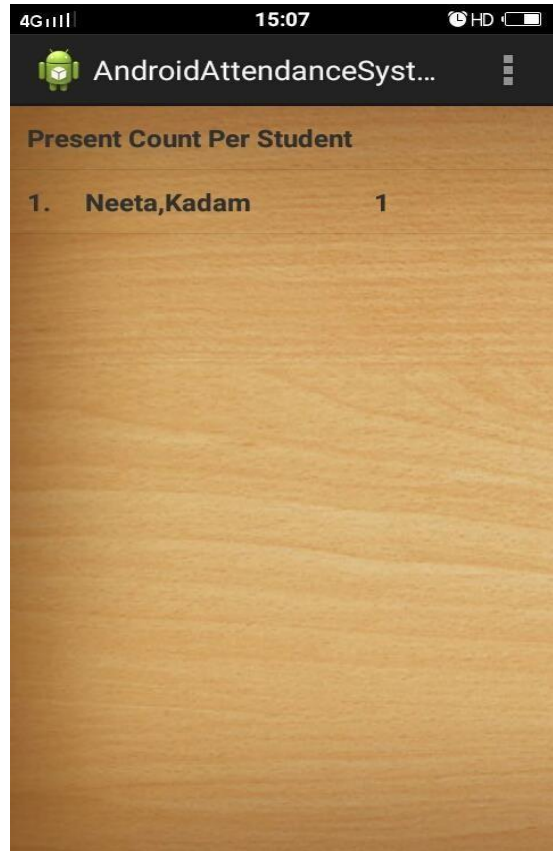
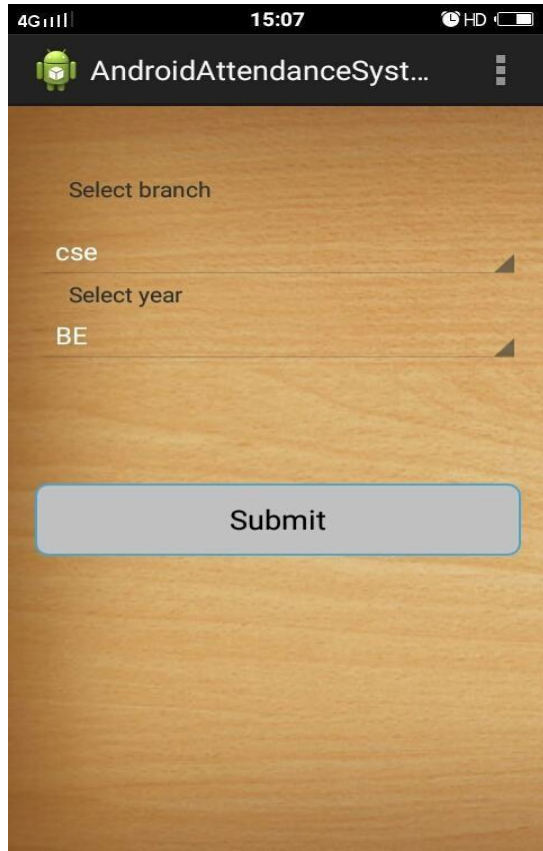
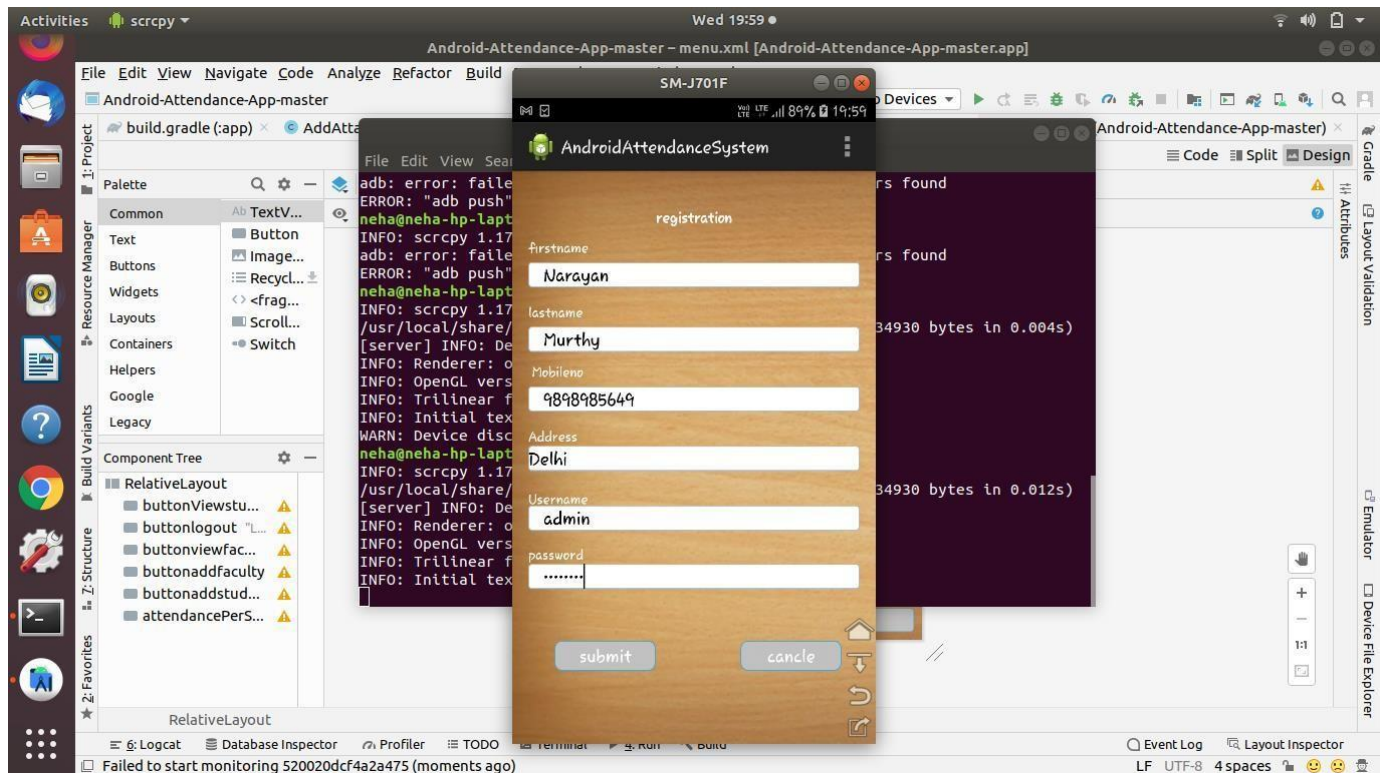
Code of mobile application for student attendance and mark management system



STUDENT ENTRY FORM



FACULTY ATTENDANCE ENTRY FORM



CHAPTER - 6

CONCLUSION& FUTURE ENHANCEMENT

CONCLUSION

This app helps the faculties to reduce their workstress by reducing the time and calculations required to update the attendance manually. The various levels of management personals are utilizing the app in variety of ways like, viewing the attendance, performance details of students. By using this optimized mobile app, 24/7 the management personals are accessing the information for various decision making analysis.

- Easy execution Environment
- Generate report Flexibly

6.1 FUTURE ENHANCEMENT

The task has an exceptionally immense extension in future. The undertaking can be executed on intranet in future. Task can be refreshed in not so distant future as and when necessity for the same emerges, as it is entirely adaptable regarding extension. With the proposed programming of data set Space Manager prepared and completely utilitarian the customer is currently ready to oversee and henceforth run the whole work in a greatly improved, precise and blunderfree way. The accompanying are the future extension for the undertaking.

REFERENCES

- [1] Somasundaram, V., Kannan. M, Sriram, V., 2016, "Mobile based Attendance Management System", Indian Journal of Science and Technology.
- [2] Rakhi Joshi, Shete, V., Somani, S. B., 2015, "Android Based Smart Learning and Attendance Management System", International Journal of Advanced Research in Computer and Communication Engineering.
- [3] Akhila, K., Prathyusha, B., PavanKumar, M., Amrutha, M., 2013, "A Novel Approach Of Mobile Based Student Attendance Tracking System Using Android Application", International Journal of Engineering Research & Technology (IJERT).

APPENDIX

A.SOURCE CODE

MainActivity.java

```
package com.android.attendance.activity;
import com.example.androidattendancesystem.R;

import android.os.Bundle;

import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class MainActivity extends Activity {

    Button start;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        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,LoginAc
```

```
tivity.class); startActivity(intent);
```

```
}
```

```
@Override
```

```

public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
}

```

LoginActivity.java

```

package com.android.attendance.activity;

```

```

import android.app.Activity;

```

```

import android.content.Intent;

```

```

import android.graphics.Color;

```

```

import android.os.Bundle; import

```

```

android.text.TextUtils; import

```

```

android.view.Menu; import

```

```

android.view.View;

```

```

import android.view.View.OnClickListener;

```

```

import android.widget.AdapterView;

```

```

import android.widget.AdapterView.OnItemClickListener;

```

```

import android.widget.ArrayAdapter;

```

```

import android.widget.Button;

```

```

import android.widget.EditText;

```

```

import android.widget.Spinner;

```

```

import android.widget.TextView;

```

```

import android.widget.Toast;

```

```

import com.android.attendance.bean.FacultyBean;

```

```

import com.android.attendance.context.ApplicationContext;

```

```

import com.android.attendance.db.DBAdapter;

```

```
import com.example.androidattendancesystem.R;
```

```
public class LoginActivity extends Activity {
```

```
    Button login;
```

```

EditText username,password;
Spinner spinnerloginas; String
userrole;
private String[] userRoleString = new String[] { "admin", "faculty"};

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

    login =(Button)findViewById(R.id.buttonlogin);
    username=(EditText)findViewById(R.id.editTextusername);
    password=(EditText)findViewById(R.id.editTextpassword);
    spinnerloginas=(Spinner)findViewById(R.id.spinnerloginas);

    spinnerloginas.setOnItemSelectedListener(new
OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> arg0, View view, int
            arg2, long arg3) {
            // TODO Auto-generated method stub

            ((TextView) arg0.getChildAt(0)).setTextColor(Color.WHITE);
            userrole =(String) spinnerloginas.getSelectedItem();

        }

        @Override
        public void onNothingSelected(AdapterView<?> arg0) {
            // TODO Auto-generated method stub

        }
    });

    ArrayAdapter<String> adapter_role = new ArrayAdapter<String>(this,
        android.R.layout.simple_spinner_item, userRoleString);

```

adapter_role

```

.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
;

spinnerloginas.setAdapter(adapter_role);

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("admin123")){

                    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();
```

```

}
}

}

else
{
    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");
    }
    DBAdapter dbAdapter = new
DBAdapter(LoginActivity.this);
    FacultyBean facultyBean =
dbAdapter.validateFaculty(user_name, pass_word);

    if(facultyBean!=null)
    {
        Intent intent = new
Intent(LoginActivity.this,AddAttendanceSessionActivity.class);
        startActivity(intent);

        ((ApplicationContext)LoginActivity.this.getApplicationContext()).setFacultyBea
n(facultyBean);

        Toast.makeText(getApplicationContext(), "Login
successful", Toast.LENGTH_SHORT).show();

```

```
    }  
    else  
    {  
        Toast.makeText(getApplicationContext(), "Login  
failed", Toast.LENGTH_SHORT).show();  
    }
```

```
}  
  
    }  
  
    }  
});
```

```
}
```

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

```
}
```

MenuActivity.java

```
package com.android.attendance.activity;
```

```
import java.util.ArrayList;
```

```
import com.android.attendance.bean.AttendanceBean; import  
com.android.attendance.context.ApplicationContext; import  
com.android.attendance.db.DBAdapter;  
import com.example.androidattendancesystem.R;
```

```
import android.os.Bundle;  
import android.app.Activity;  
import android.content.Intent;  
import android.view.Menu;  
import android.view.View;
```

```
import android.view.View.OnClickListener;
import android.widget.Button;
public class MenuActivity extends Activity {
```

```
    Button addStudent;
```

```
    Button addFaculty;
```

```
    Button viewStudent;
```

```
    Button viewFaculty;
```

```
    Button logout;
```

```
    Button attendancePerStudent;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.menu);
```

```
        addStudent =(Button)findViewById(R.id.buttonaddstudent);
```

```
        addFaculty =(Button)findViewById(R.id.buttonaddfaculty);
```

```
        viewStudent =(Button)findViewById(R.id.buttonViewstudent);
```

```
        viewFaculty =(Button)findViewById(R.id.buttonviewfaculty);
```

```
        logout =(Button)findViewById(R.id.buttonlogout);
```

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

```
            @Override
```

```
            public void onClick(View v) {
```

```
                // TODO Auto-generated method stub
```

```
        Intent intent =new  
Intent(MenuActivity.this,AddStudentActivity.class);  
        startActivity(intent);  
    }  
});
```

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

```

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

    Intent intent =new
Intent(MenuActivity.this,AddFacultyActivity.class);
    startActivity(intent);
}
});

viewFaculty.setOnClickListener(new OnClickListener() {

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

    Intent intent =new
Intent(MenuActivity.this,ViewFacultyActivity.class);
    startActivity(intent);
}
});

viewStudent.setOnClickListener(new OnClickListener() {

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

    Intent intent =new
Intent(MenuActivity.this,ViewStudentActivity.class);

```

```
        startActivity(intent);
    }
});
logout.setOnClickListener(new OnClickListener() {
```

```

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

            Intent intent =new
Intent(MenuActivity.this,MainActivity.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);
        }
    });

    attendancePerStudent=(Button)findViewById(R.id.attendancePerStudentButt on);
    attendancePerStudent.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View arg0) {

            DBAdapter dbAdapter = new DBAdapter(MenuActivity.this);
            ArrayList<AttendanceBean>
attendanceBeanList=dbAdapter.getAllAttendanceByStudent();

            ((ApplicationContext)MenuActivity.this.getApplicationContext()).setAttendance
BeanList(attendanceBeanList);

            Intent intent = new
Intent(MenuActivity.this,ViewAttendancePerStudentActivity.class);
            startActivity(intent);

        }
    });

}

```

@Override

public boolean onCreateOptionsMenu(Menu menu) {

```
// Inflate the menu; this adds items to the action bar if it is present.  
    getMenuInflater().inflate(R.menu.main, menu);  
    return true;  
}  
}
```