ABSTRACT

The project is implemented using Android Studio, Java and SQLite. This project is an online portal between employee and company. This innovative system allows companies to share important data as well as notifications with their employees. It consists of an attender login along with employee login. Since companies operate through pc and document uploading is simpler through a pc, the attender login is to be performed through a computer.

Attender may upload documents of meeting schedules, timetable document, notifications, reports etc. through their provided login. The documents are uploaded by attender to different corresponding departments. We propose to build this system on an online server that allows attender to upload data and employee may view search and download required documents through the android device. Here employees only see and download data of their particular team. Rest data is hidden. Attender may access and upload/edit documents to any teams or add any notice as desired.

INDEX

CHAPTER1 INTRODUCTION

1.1 Brief Introduction

Employee and Attendance Management app is software developed for daily employee attendance in companies. It facilitates to access the attendance information of a particular employee in a particular team/project. The information is sorted by the operators, which will be provided by the attender/manager for a particular team/project. This system will also help in evaluating attendance eligibility criteria of an employee.

1.2 PURPOSE

The purpose of developing Employee and Attendance Management 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 day or in the between of the day.

1.3 OVERVIEW

Employee and Attendance Management basically has two main modules for proper functioning: - First module is admin which has right for creating space for new employee. Any entry of new attender/manager, Updating in project if necessary, and sending notice. Second module is handled by the user which can be an attender or an operator. User has a right of making daily attendance, generating report.

CHAPTER 2

REQUIREMENT ANALYSIS

2.1 Requirement Gathering

Requirement analysis is the first and important step in the Software Development activity for building gro bustand user friendly applications ,Iha vest art ed working on determining the functionalities that the application should provide. I have done a good amount of research on existing systems and the disadvantages of those.

2.2 Requirement Specifications

Below are the technical requirements to develop student result management system,

Software Requirements Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application.

The following are the software requirements for the application:

- ✓ OperatingSystem:Windows10
- ✓ Development Environment:AndroidStudio4.2
- ✓ API:JavaDevelopmentKit(JDK)7
- ✓ CoreLanguage:Java, XMLforFront-end.

Hardware Requirements the most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware.

- ✓ CPU:Intel processor with support for Intel VT-x(Intel64)
- ✓ Cores:Dual-Core(Quad-Core recommended)

- ✓ RAM:minimum4GB(>4GBrecommended)
- ✓ SecondaryStorage:250GBharddiskspaceplusatleast1GBforAndroidSDK,
- ✓ Emulator System images and cashes.
- ✓ Screen resolution: 1366 x800.

2.3 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 companies. But later on the project can be modified to operate it online.

CHAPTER 3

MODULES

The application comprises the following major modules.

3.1 Admin Module

Admin is responsible for adding attender and employee. He canal so view each employee attend an nceseparatel after reach session.

3.2 Attender Module

Once admin adds attender/manager, they can access the application and assign the attendance to a particular employee at particular day.

3.3 Features

Employee and Attendance Management divided into two modules i.e. Admin and Attender.

3.3.1 Features of Admin are:

- ✓ Add Employee.
- ✓ Add Attender.
- ✓ View Employee.
- ✓ View Attender.
- ✓ View Employee Attendance.
- ✓ View each Employee’s attendance separately

3.3.2 Features of Attender are:

- ✓ Take attendance and keep them department/team wise
- ✓ Add New Employee
- ✓ View each employee’s attendance separately
- ✓ Edit Employee/Attendance later
- ✓ Save project wise

CHAPTER 4

SYSTEM DESIGN

4.1 XML

XML (Extensible Markup Language) is a markup language similar to HTML, but without predefined tags to use. Instead, you define your own tags designed specifically for your needs. This is a powerful way to store data in a format that can be stored, searched, and shared. Most importantly, since the fundamental format of XML is standardized, if you share or transmit XML across systems or platforms, either locally or over the internet, the recipient can still parse the data due to the standardized XML syntax. There are many languages based on XML, including XHTML, MathML, SVG, XUL, XBL, RSS, and RDF. You can also define your own.

4.2. DESCRIPTION

- Scroll View

A view group that allows the view hierarchy placed within it to be scrolled. Scroll view may have only one direct child placed within it. To add multiple views within the scroll view, make the direct child you add a view group, for example Linear Layout, and place additional views within that Linear Layout. Scroll view supports vertical scrolling only. For horizontal scrolling, use Horizontal Scroll View instead. Never add a Recycler View or List View to a scroll view. Doing so results in poor user interface performance and a poor user experience

- Card View

Card View uses elevation property on Lollipop for shadows and falls back to a custom emulated shadow implementation on older platforms. Due to expensive nature of rounded corner clipping, on platforms before Lollipop, Card View does not clip its children that intersect with rounded corners. Instead, it adds padding to avoid such intersection

- Circle Image View

As this is just a custom Image View and not a custom Drawable or a combination of both, it can be used with all kinds of drawables, i.e. a Picasso Drawable from Picasso or other non-standard drawables (needs some testing though).

- Text View

A user interface element that displays text to the user. To provide user-editable text, see EditText.

- Image view

Displays image resources, for example Bitmap or Drawable resources. Image View is also commonly used to apply tints to an image and handle image scaling

- Edit text

A user interface element for entering and modifying text. When you define an edit text widget, you must specify the R.styleable .TextView_ input Type attribute.

- Recycler View

A flexible view for providing a limited window into a large data set.

- Nested Scroll View

Nested Scroll View is just like ScrollView, but it supports acting as both a nested scrolling parent and child on both new and old versions of Android. Nested scrolling is enabled by default.

- Eaze Graph

Eaze Graph is an Android library for creating beautiful and fancy charts. Its main goal was to create a light weight library which is easy to use and highly customize able with an "up-to-date"-look. Currently 4 different chart types are available

- View

An XML view is one of the predefined view types that are available in OpenUI5. The XML view type is defined in an XML file. xml or as an XML string.

CHAPTER 5

IMPLEMENTATION

5.1. JAVA

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended tolet application developers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low- level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub,particularly for client-server web applications, with a reported 9 million developers.

5.2. DESCRIPTION

- Void on Create(Bundle saved Instance State)

When an Activity first call or launched then on Create(Bundle saved Instance State) method is responsible to create the activity.When ever orientation(i.e. from horizontal to vertical or vertical to horizontal) of activity gets changed or when an Activity gets forcefully terminated by any Operating System then saved Instane State i.e. object of Bundle Class will save the state of an Activity. After Orientation changed then onCreate(Bundle saved Instance State) will call and recreate the activity and load all data from saved Instance State.

- Void Window.open()

he Window interface's open() method loads the specified resource into the new or existing browsing context (window, <iframe> or tab) with the specified name. If the name doesn't exist, then a new browsing context is opened in a new tab or a new window, and the specified resource is loaded into it.

- public static final

A public static final variable is a compile-time constant, but a public final is just a final variable, i.e. you cannot reassign value to it but it's not a compile-time constant. This may look puzzling, but the actual difference allows how the compiler treats those two variables.

- private Recycler 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. As the name implies, Recycler View recycles those individual elements. When an item scrolls off the screen, Recycler View doesn't destroy its view. Instead, Recycler View reuses the view for new items that have scrolled onscreen. This reuse vastly improves performance, improving your app's responsiveness and reducing power consumption

- private ArrayList

An Array List class is a resizable array, which is present in the java. util package. While built-in arrays have a fixed size, Array Lists can change their size dynamically. Elements can be added and removed from an Array List whenever there is a need, helping the user with memory management.

- Private Swipe Refresh Layout.

In Android, the common "pull to refresh" UX concept is not built in to a ListView/RecyclerView. However, many Android applications would like to make use of this concept for their feeds. This is useful for all sorts of feeds such as a Twitter timeline. This effect can be achieved using the Swipe Refresh Layout class

- private EditText

A user interface element for entering and modifying text. When you define an edit text widget, you must specify the android.R.styleable#TextView_inputType attribute. For example, for plain text input set inputType to "text"

- et_ search.add Text Changed Listener(new Text Watcher())

Android Edit Text is a subclass of Text View. Edit Text is used for entering and modifying text. While using Edit Text width, we must specify its input type in input Type property of Edit Text which configures the keyboard according to input. Edit Text uses Text Watcher interface to watch change made overEdit Text. For doing this, Edit Text calls the add Text Changed Listener() method.

- public abstract void after TextChanged (Editable s)

This method is called to notify you that, somewhere within s, the text has been changed. It is legitimate to make further changes to s from this callback, but be careful not to get yourself into an infinite loop, because any changes you make will cause this method to be called again recursively. (You are not told where the change took place because other after TextChanged() methods may already have made other changes and invalidated the offsets. But if you need to know here, you can use Spannable#setSpan in onTextChanged(CharSequence, int, int, int) to mark your place and then look up from here where the span ended up.

- Void before TextChanged

This method is called to notify you that, within s, the count characters beginning at start are about to be replaced by new text with length after. It is an error to attempt to make changes to s from this callback.

- void on TextChanged

This method is called to notify you that, within s, the count characters beginning at start have just replaced old text that had length before. It is an error to attempt to make changes to s from this callback.

CHAPTER 6

SOURCE CODE

Activity main

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingBottom="@dimen/activity_vertical_margin"
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   android:background="@drawable/clg"
   tools:context=".MainActivity" >
   <TextView
      android:id="@+id/textView1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentLeft="true"
      android:layout_alignParentTop="true"
      android:layout_marginLeft="78dp"
      android:layout_marginTop="86dp"
      android:text="WEL-COME"
      android:textColor="#ffff00"
      android:textStyle="italic"
      android:textAppearance="?android:attr/textAppearanceLarge" />
   <Button
```

```
    android:id="@+id/buttonstart"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:background="@drawable/roundedbutton"
    android:text="Start" />
</RelativeLayout>
```

Login page

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:paddingBottom="@dimen/activity_vertical_margin"
  android:paddingLeft="@dimen/activity_horizontal_margin"
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"
  android:background="@drawable/clg"
  tools:context=".MainActivity" >
  <TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginLeft="82dp"
    android:layout_marginTop="43dp"
    android:text="Login here.."
    android:textAppearance="?android:attr/textAppearanceLarge" />
  <EditText
    android:id="@+id/editTextpassword"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView3"
    android:layout_below="@+id/textView3"
    android:layout_marginTop="15dp"
    android:background="@drawable/roundedtextview"
    android:ems="10"
    android:inputType="textPassword" />
  <TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/editTextusername"
    android:layout_below="@+id/editTextusername"
    android:layout_marginTop="14dp"
    android:text="Password"
    android:textAppearance="?android:attr/textAppearanceMedium" />
  <TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/editTextusername"
    android:layout_below="@+id/textView1"
    android:layout_marginTop="110dp"
    android:text="Username"
    android:textAppearance="?android:attr/textAppearanceMedium" />
  <EditText
android:id="@+id/editTextusername"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView2"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:background="@drawable/roundedtextview"
    android:ems="10" >
    <requestFocus />
  </EditText>
  <Spinner
    android:id="@+id/spinnerloginas"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView2"
    android:layout_alignBottom="@+id/textView2"
    android:layout_alignParentRight="true"
    android:layout_marginBottom="32dp" />
  <Button
```

```
        android:id="@+id/buttonlogin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignRight="@+id/spinnerloginas"
        android:layout_marginBottom="34dp"
        android:text="Login" />
</RelativeLayout>
```

Add attendance

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/clg"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_marginLeft="57dp"
        android:layout_marginTop="43dp"
        android:text="Select Technology"
        android:textAppearance="?android:attr/textAppearanceSmall" />
    <Spinner
        android:id="@+id/spinner1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="56dp"
        android:layout_marginTop="34dp"
        android:layout_toRightOf="@+id/textView1" />
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/spinner1"
        android:layout_marginTop="10dp"
        android:layout_toLeftOf="@+id/spinner1"
        android:text="Select Building"
        android:textAppearance="?android:attr/textAppearanceSmall" />
    <Spinner
        android:id="@+id/spinnerSE"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/spinneryear"
        android:layout_below="@+id/spinneryear" />
    <Spinner
        android:id="@+id/spinneryear"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/spinner1"
  android:layout_alignLeft="@+id/spinner1" />
    <EditText
        android:id="@+id/DateEditText"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/DateImageButton"
        android:layout_toLeftOf="@+id/spinnerSE"
        android:background="#ffffff"
        android:editable="false"
        android:ems="10" >
        <requestFocus />
    </EditText>
    <ImageButton
        android:id="@+id/DateImageButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/spinnerSE"
        android:layout_below="@+id/spinnerSE"
        android:layout_marginLeft="14dp"
        android:layout_marginTop="22dp"
        android:cropToPadding="true"
```

```
    android:src="@drawable/calendar_icon" />
<Button
    android:id="@+id/buttonsubmit"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/DateImageButton"
    android:layout_marginTop="22dp"
    android:background="@drawable/roundedbutton"
    android:text="Add Attendance" />
<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView2"
    android:layout_alignTop="@+id/spinnerSE"
    android:layout_marginLeft="17dp"
    android:layout_marginTop="13dp"
    android:text="Project"
    android:textAppearance="?android:attr/textAppearanceSmall" />
<Button
    android:id="@+id/viewTotalAttendanceButton"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:background="@drawable/roundedbutton"
        android:text="View Total Attendance" />
</RelativeLayout>
```

View employee

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:background="@drawable/clg"
    tools:context=".MainActivity" >
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="29dp"
        android:text="Select Technology"
        android:textAppearance="?android:attr/textAppearanceSmall" />
    <Spinner
        android:id="@+id/spinnerbranchView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/textView1"
        android:layout_marginTop="14dp" />
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView1"
        android:layout_below="@+id/spinnerbranchView"
        android:text="Select Buidling"
        android:textAppearance="?android:attr/textAppearanceSmall" />
    <Spinner
        android:id="@+id/spinneryearView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView2"
        android:layout_centerHorizontal="true" />
    <Button
        android:id="@+id/submitButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

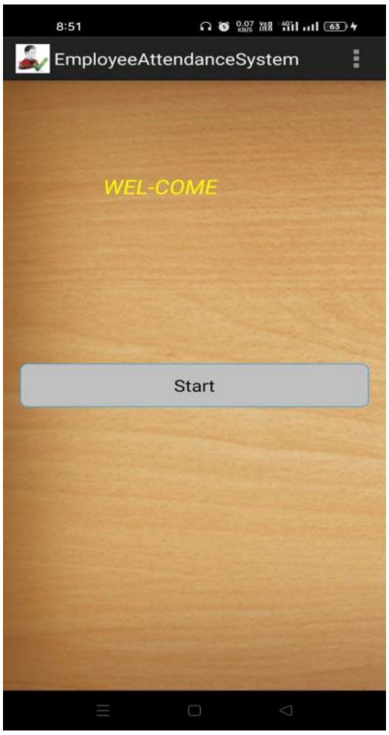
```
        android:layout_below="@+id/spinneryearView"
        android:layout_marginTop="80dp"
        android:background="@drawable/roundedbutton"
        android:text="Submit" />
</RelativeLayout>
```

View attender

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/labelF"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:padding="10dip"
        android:textSize="16dip"
        android:textStyle="bold" >
</TextView>
```

CHPTER 7

SNAPSHOTS



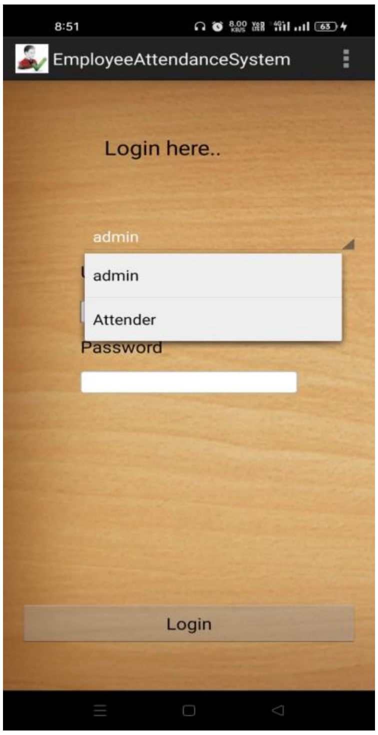
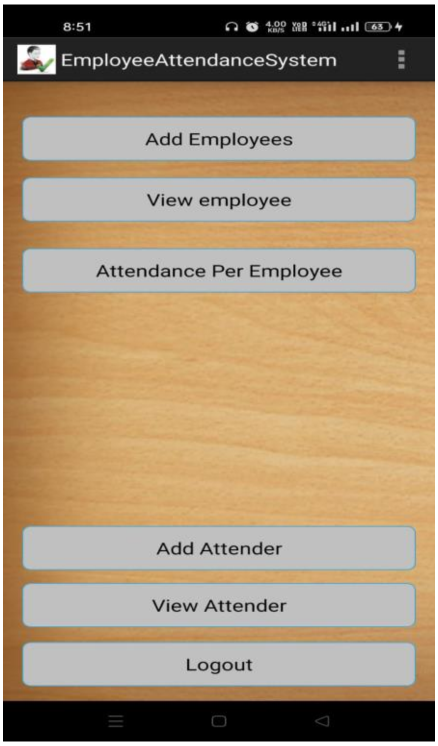


Fig1: Design interface

Fig2: Login page



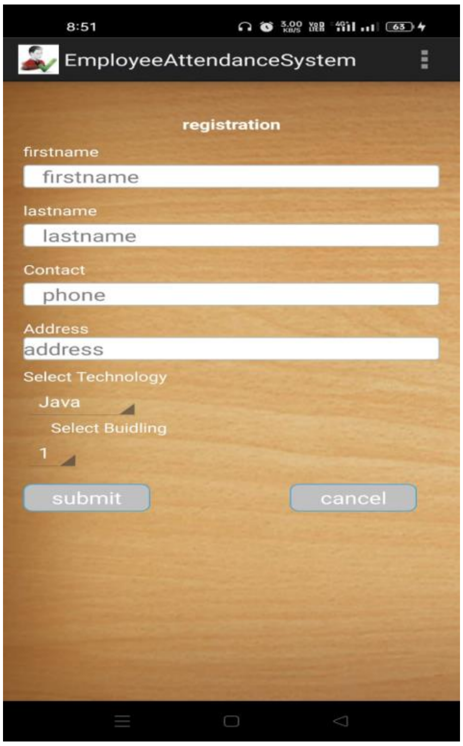
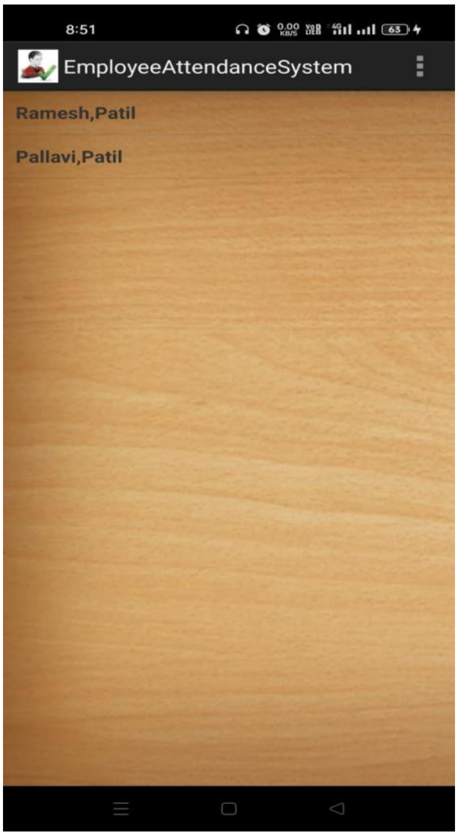


Fig3: Add employee page

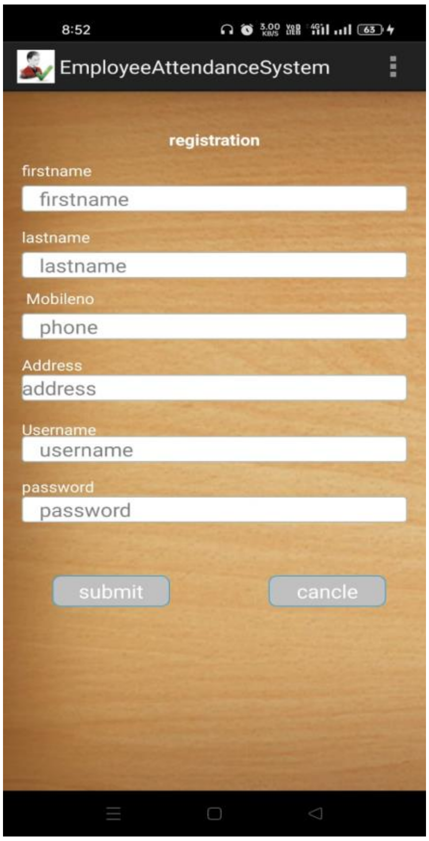
Fig4: Adding employee



Ramesh,Patil

Pallavi,Patil

Fig5: Employee list



registration

firstname: firstname

lastname: lastname

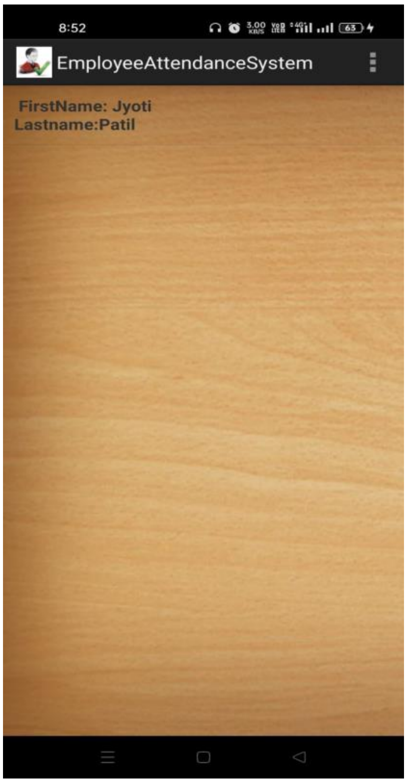
Mobileno: phone

Address: address

Username: username

password: password

Fig6: Add attender page

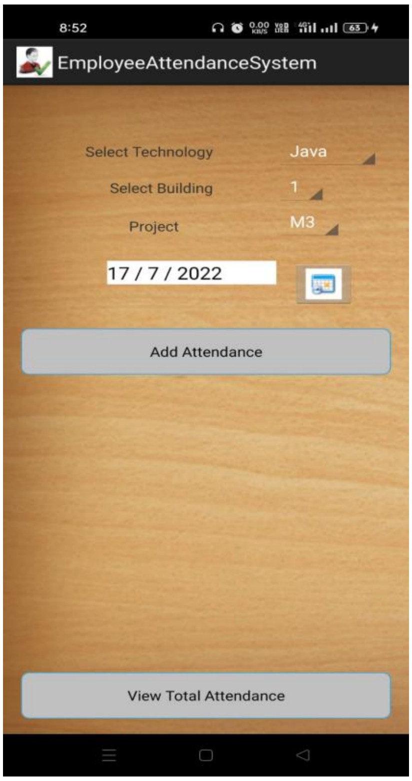


8:52

EmployeeAttendanceSystem

FirstName: Jyoti
Lastname:Patil

Fig7: Attender list



8:52

EmployeeAttendanceSystem

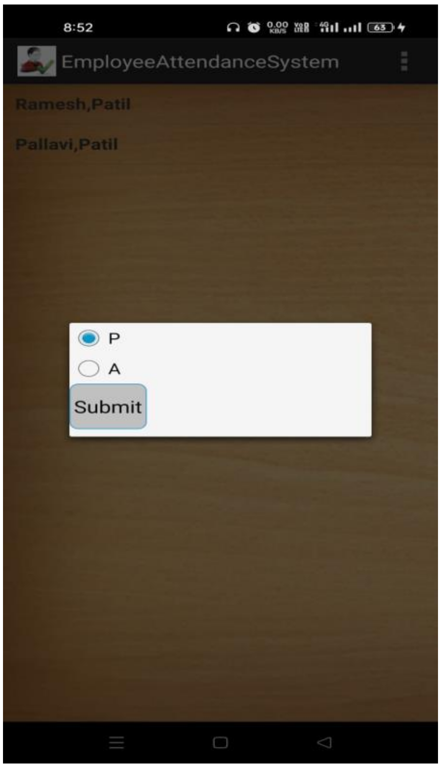
Select Technology Java

Select Building 1

Project M3

17 / 7 / 2022

Fig8: Taking attendance



8:52

EmployeeAttendanceSystem

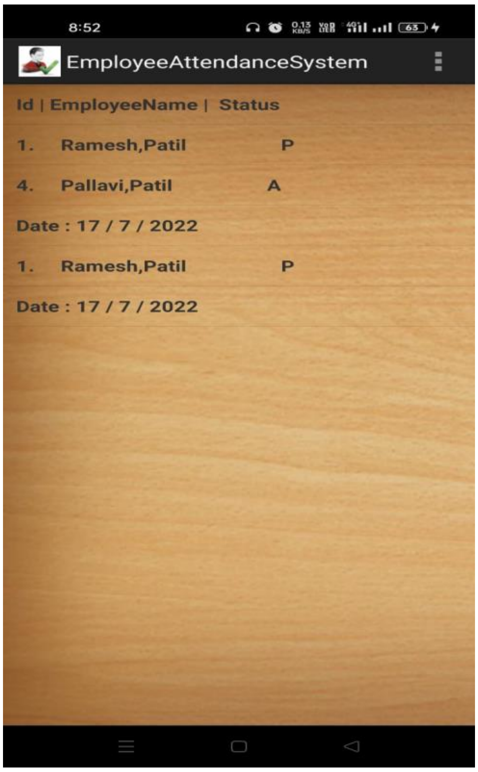
Ramesh,Patil

Pallavi,Patil

● P

○ A

Submit



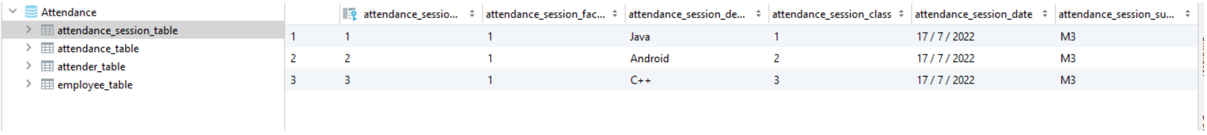
8:52

EmployeeAttendanceSystem

```
Id | EmployeeName |  Status
1.   Ramesh,Patil          P
4.   Pallavi,Patil         A
Date : 17 / 7 / 2022
1.   Ramesh,Patil          P
Date : 17 / 7 / 2022
```

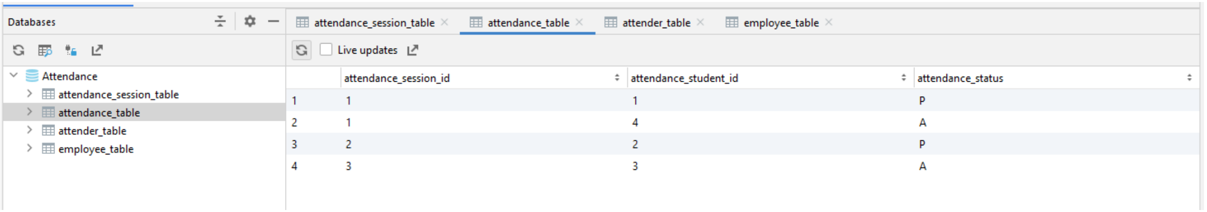
Fig9: Marking attendance

Fig10: Attendance list



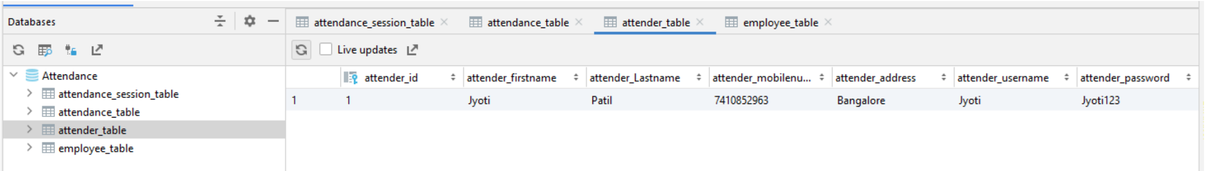
	attendance_sessio...	attendance_session_fac...	attendance_session_de...	attendance_session_class	attendance_session_date	attendance_session_su...
1	1	1	Java	1	17 / 7 / 2022	M3
2	2	1	Android	2	17 / 7 / 2022	M3
3	3	1	C++	3	17 / 7 / 2022	M3

Table: Attendance session table



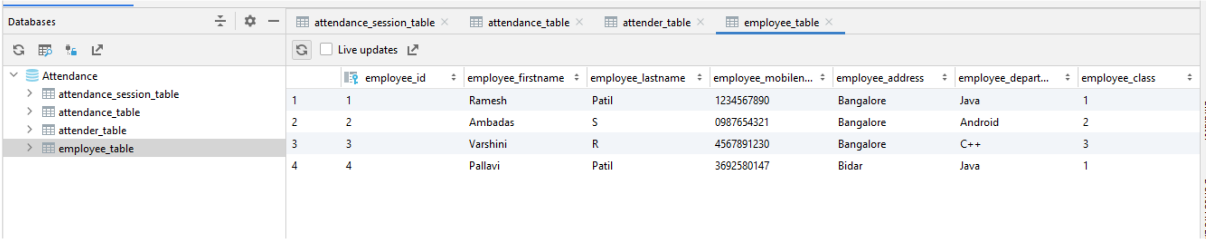
	attendance_session_id	attendance_student_id	attendance_status
1	1	1	P
2	1	4	A
3	2	2	P
4	3	3	A

Table: Attendance table



	attender_id	attender_firstname	attender_Lastname	attender_mobilenu...	attender_address	attender_username	attender_password
1	1	Jyoti	Patil	7410852963	Bangalore	Jyoti	Jyoti123

Table: Attender table



	employee_id	employee_firstname	employee_lastname	employee_mobilen...	employee_address	employee_depart...	employee_class
1	1	Ramesh	Patil	1234567890	Bangalore	Java	1
2	2	Ambadas	S	0987654321	Bangalore	Android	2
3	3	Varshini	R	4567891230	Bangalore	C++	3
4	4	Pallavi	Patil	3692580147	Bidar	Java	1

Table: Employee table

CONCLUSION

Employee and Attendance manager developed using Java fully meets the objectives of the system foe which it was developed. The application has reached a steady-state where all bugs have been eliminated. The application is operated at a high level of efficiency and the teachers and users associated with the system understand its advantage. The system this solves the problems it was intended to solve.

REFERANCES

WEBSITES

- World Health Organization. 2021. Coronavirus disease 2019 (COVID-19): Situation report. Retrieved from https://www.worldometers.info/coronavirus/?utm_campaign=homeAdvegas1 ?%20
- The Government of India. 2020. Live updates from World meter. Retrieved from https://www.worldometers.info/coronavirus/country/india/.
- covid19india.orgfor India specific API.
- . corona.Imao.ninja for Global API Flat icon for social media icons.
- http://google.com
- .in.searchley.com/Mobile Develop

TEXTBOOKS

- Android Developer Fundamentals Course – By Google.
- Android Programming – Pushing The Limits – By Erik Hellman.
- Head First Android Development – By Dawn Griffiths and David Griffiths