1.Introduction

In this new era of Science and Technology, computer is one of the most important components in our life. Works can be done in a better way by the help of computer.

The Main aim of our project is to automate the teaching staff reporting system in the form of Android Mobile Application. Our intention is to store notes and update attendance,save schedules in a single application which is easy to work in offline .We have seen over the years that the process of manual attendance is being carried out across almost all educational institutions. The process is not only time consuming but also sometimes yield inefficient results in the false marking and calculation of attendance. We need not maintain pen and paper based attendance registers. Following this thought, we have proposed a Native application for the Teaching staff android application .This Android application will give the students attendance information and Saved schedule or notes notification feature whereby every teaching staff will be periodically notified regarding his/her class periods. Our system primarily focuses on building an efficient and user friendly Android mobile application. The application will be installed on the Mentor phone which runs android OS. It intends to provide an interface to the professor who will require a user id and password to carry out the task. Apart from that, the application would support strong user authentication and quick transmission of data.

Native apps are considered to be the most reliable, showing extraordinary performance when compared to hybrid and web apps.Native apps are developed for a specific mobile platform using particular programming languages and technologies. IOS apps, for example, are written in Objective-C and Swift, Android apps in Java or Kotlin. C++ is used for native Windows and Blackberry apps. Native apps can be used for access to built in feature of the device. Native applications are available from app stores.A hybrid app is, in a way, a compromise between web and native app development. It combines the advantages of both types of app development.A hybrid application is technically a web app packed in a native app container. Like the web app, it is written in HTML, CSS and JavaScript. Unlike the web app, it is distributed through the app store or Over-the-Air for iOS devices.

1.1 PROBLEM IDENTIFICATION

The prototype was designed based on the Pomodoro technique. Pomodoro is a time management technique which helps users to manage the time in an efficient manner. The main goal of this thesis is to design and develop the prototype in order to measure the product quality according to quality attributes chosen based on ISO/IEC 9126 quality standard.

The following tasks were initially formulated for this thesis project:

1. Literature study in the domain of smart phone technology and software product quality
2. Designing and implementing of a prototype supported by Android platform.
3. Defining a quality model for the prototype.
4. Designing an experiment to evaluate usability of the prototype.

1.2 EXISTING SYSTEM: Existing system of android application is a manual one in which each and every faculty member has their own register in which they mark attendance

Disadvantages of Existing System :

- Faculty generally make mistake sometimes
- Calculation is a great task for every 15 days

1.3 PROPOSED SYSTEM :

Our proposal on PCR is to automate the teaching staff reporting system in the form of android mobile app NATIVE APPLICATION FOR THE TEACHING STAFF installed in faculty's mobile which runs on Android OS.

Advantages of proposed system :

- It automates the attendance for every day
- Sends automatic notification alerts
- It's a very smart way since we don't need to use pen and paper

2.LITERATURE REVIEW

2.1 LITERATURE REVIEW :

Android is a relatively new platform. It is a product of Google, Inc., and its first release was presented in 2007 (Meier, 2010). It has been released by google in many versions with different specifications for each version the starting version is named as Android 1.0 was installed on many different mobile devices and its users can download Android apps and other content through Google Play service, which replaced the old Android Market (Bishop, 2012). As the official Android website describes this platform, "Android is a software stack for mobile devices that includes an operating system, middleware and key applications". Android provides the "core set of applications including an email client, SMS program, calendar, maps, browser, contacts, and others" while additional applications can be downloaded through Google Play service. Google claims that "Android powers millions of phones, tablets and other devices." Phones and tablets are mobile devices that can have Android applications installed on them. These applications are written in Java programming language and they are called mobile device applications or apps. Development techniques for apps are structured sets of Java code focused on implementing particular task that provides content for a mobile device application. Although Java programming language includes a broad variety of topics, this thesis focuses on development techniques required for successful implementation of Android Mobile EMU Portal.

2.2 Prior Research :

Contributions of prior research efforts provide useful information for successful implementation of Android Mobile EMU Portal. This thesis analyzes how to determine which development technique to use for a particular feature, what are the steps to implement each technique, and whether they can be applied for EMU-related data. Testing of the official Android EMU app has shown that it provides some of the pre-selected features for Android Mobile EMU portal like Twitter Updates, Online Directory, Athletics News, and Campus Map. Implementation steps for development techniques required are explored in various Android development books and Internet tutorials. Since a majority of Android

Mobile EMU Portal content is obtained from the Web, this section reflects existing development techniques that enable gathering online data.

2.3 Feasibility Study

The feasibility study is an evolution of proposed system regarding its workability, organizational ability to meet the user needs and effective use of resources. When a new application is proposed, it should go through the feasibility study before it is approved for the development.

There are three aspects of feasibility study.

1. Economic Feasibility
2. Technical Feasibility
3. Behavioural Feasibility

2.3.1 Economic Feasibility :

The project is economically feasible as it only requires a mobile phone with Android operating system. The application is free to download once released into Android market. The users should be able to connect to internet through mobile phone and this would be the only cost incurred on the project.

2.3.2 Technical Feasibility:

To develop this application, a high speed internet connection, a database server, a web server and software are required. The current project is technically feasible as the application was successfully deployed on Android Emulator.Android studio is the used software for this project mostly in order to build this android application.

2.3.3 Behavioral Feasibility:

The application is behaviorally feasible since it requires no technical guidance, all the modules are users friendly and execute in a manner they were designed to. The user must know about the basic use of mobile phone and this application provide an environment of user friendly.

3. SDLC METHODOLOGIES

The software industry includes many different processes, for example, analysis, development, maintenance and publication of software. This industry also includes software services, such as training, documentation, and consulting. Our focus here about software development life cycle (SDLC). So, due to that different types of projects have different requirements. Therefore, it may be required to choose the SDLC phases according to the specific needs of the project. These different requirements and needs give us various software development approaches to choose from during software implementation.

Types of Software developing life cycles (SDLC)

3.1 V-Shaped Mode Description:

It is an extension for waterfall model, Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical V shape. The major difference between v-shaped model and waterfall model is the early test planning in v-shaped model.

The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly-disciplined model and the next phase starts only after completion of the previous phase.

Once you have the clear and detailed product requirements, it is time to design the complete system. The system design will have the understanding and detailing the complete hardware and communication setup for the product under development. The system test plan is developed based on the system design. Doing this at an earlier stage leaves more time for the actual test execution later.

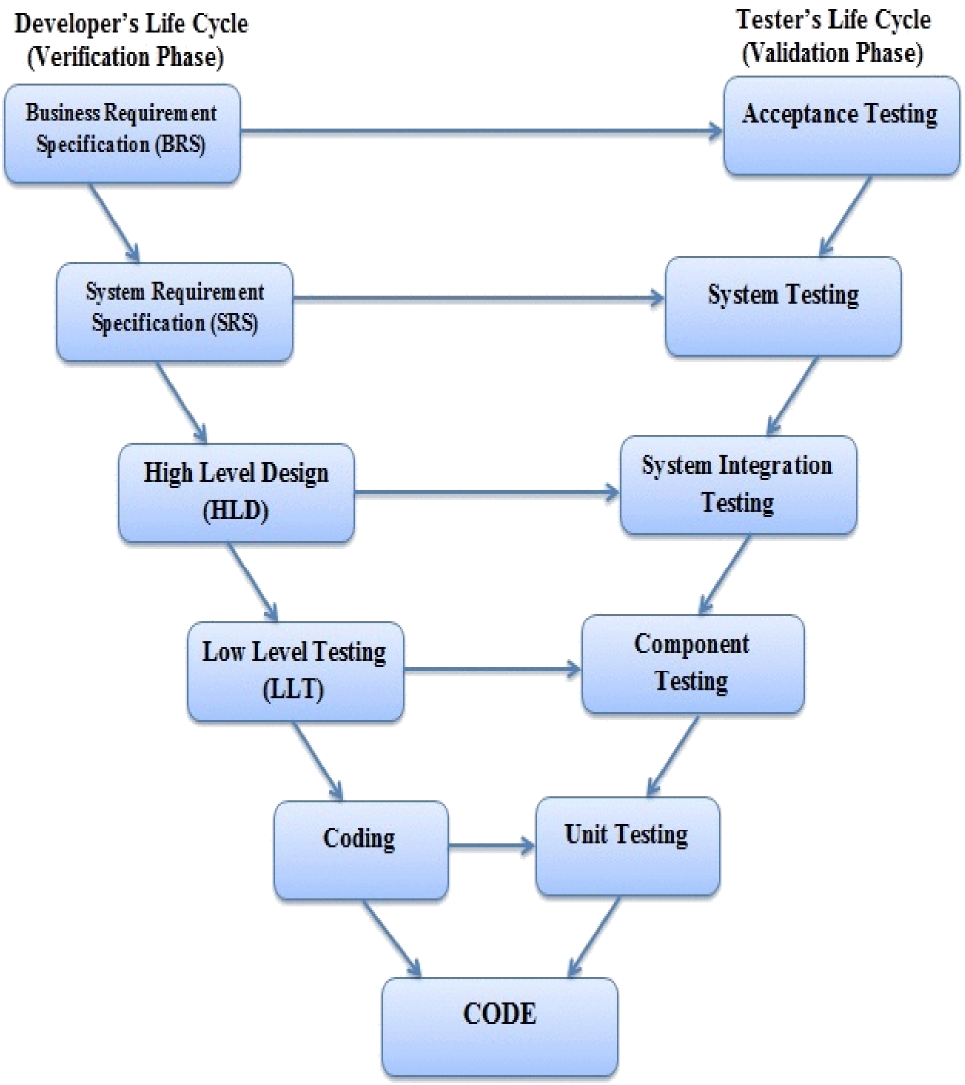


Figure 3.1: V-Shaped Diagram of SDLC

3.2 Evolutionary Prototyping Model Description:

It refers to the activity of creating prototypes of software applications, for example, incomplete versions of the software program being developed. It is an activity that can occur in software development. It used to visualize some component of the software to limit the gap of misunderstanding the customer requirements by the development team. This also will reduce the iterations may occur in waterfall approach and hard to be implemented due to inflexibility of the waterfall approach. So, when the final prototype is developed, the requirement is considered to be frozen. It has some types, such as·Throwaway prototyping Prototypes that are eventually discarded rather than becoming a part of the finally delivered software.

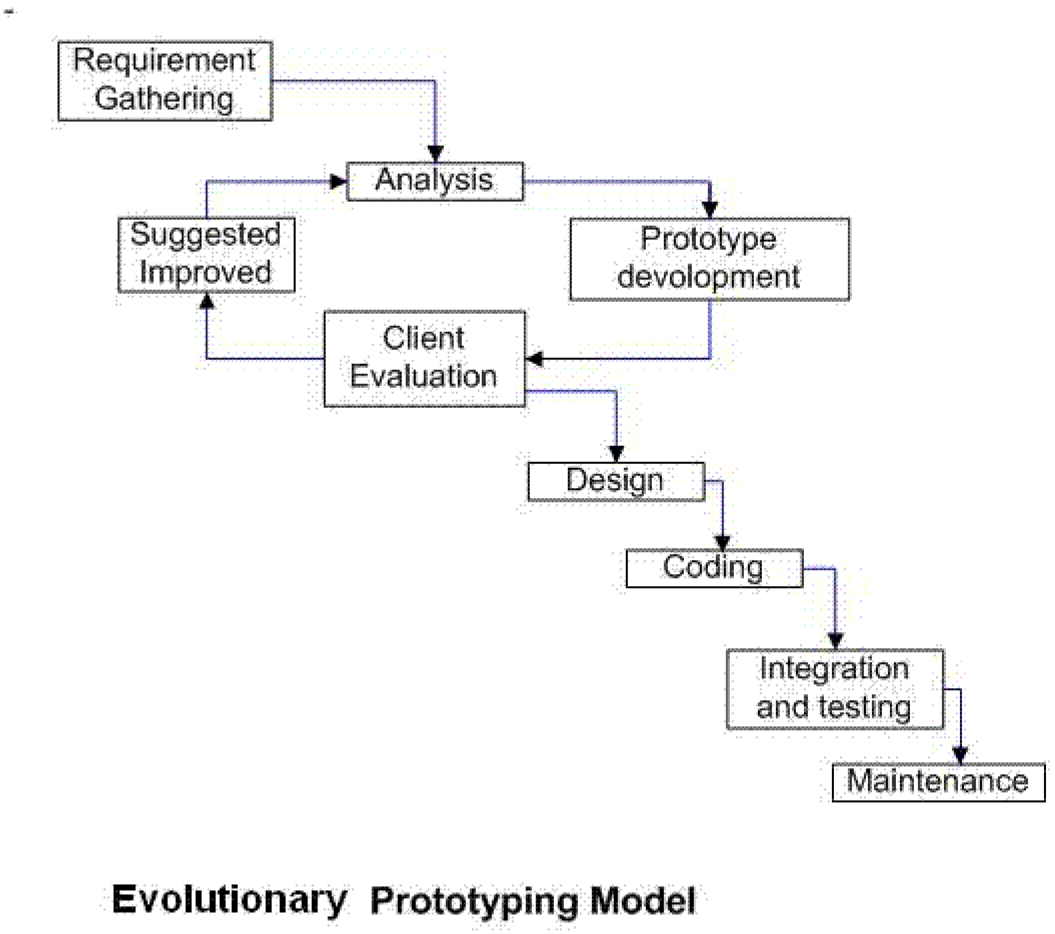


Figure 3.2: Evoluationary Prototyping model

3.3 Spiral Method (SDM)

Description:

It is combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts. This model of development combines the features of the prototyping model and the waterfall model. The spiral model is favored for large, expensive, and complicated projects. This model uses many of the same phases as the waterfall model, in essentially the same order, separated by planning, risk assessment, and the building of prototypes and simulations.Steps for spiral model can be generalized as follows: The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system. A first prototype of the new system is constructed from the preliminary design it is usually a scaled-down represents an approximation of the characteristics of the final product.

- A second prototype is evolved by the fourfold procedure.
- Evaluating the first prototype in terms of its strengths, weakness and risks.
- Defining the requirements of the second prototype.
- Planning and designing the second prototype.
- Constructing and testing the second prototype.

At the customer option the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgement, result in a less-than-satisfactory final product.The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is the developed from it according to the fourfold procedure outlined above.The preceding steps are iterated until the customer is satisfied that the refine prototype represents the final product desired.The final system is constructed, based on the refined prototype. thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

Estimates (i.e. budget, schedule, final product.· Needs special etc.) become more realistic as work skills to progresses, because important issues are discovered earlier.· Early limiting re-usability involvement of developers· Manages risks and develops system into phases.

High cost and time to reach the Estimates (i.e. budget, schedule, final product.· Needs special etc.) become more realistic as work skills to evaluate the risks and progresses, because important assumptions.Highly customized issues are discovered earlier.·

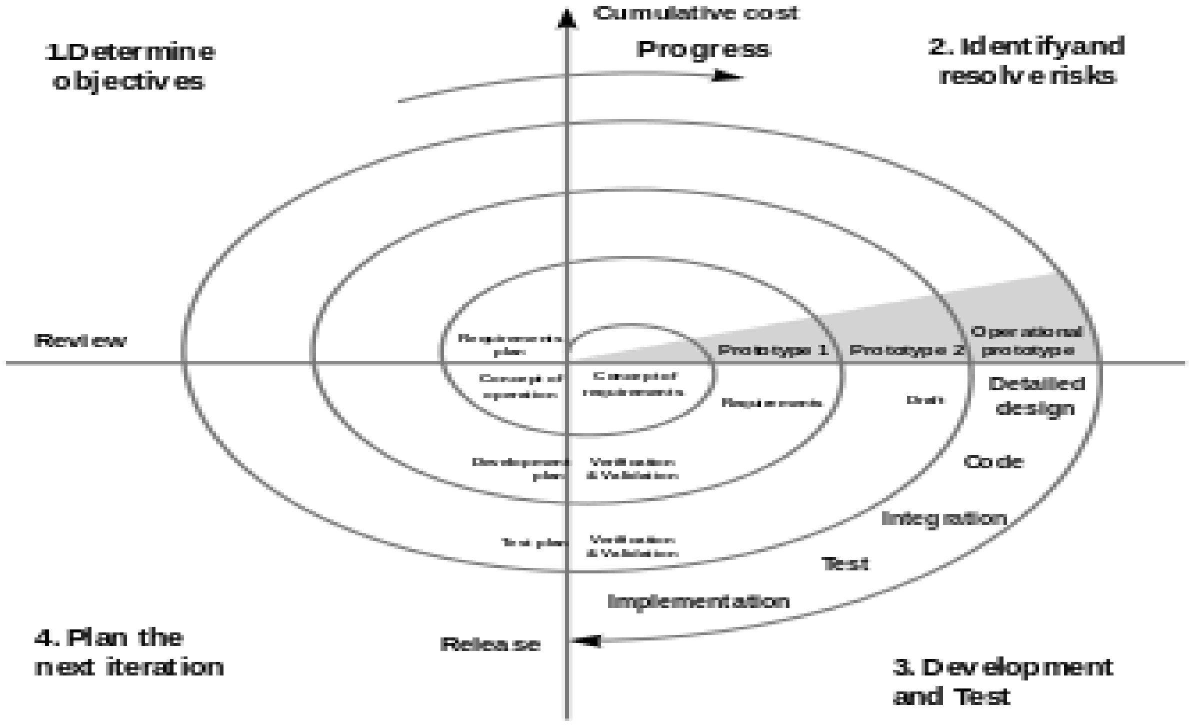


Figure 3.3: Spiral Model Diagram

3.3 Spiral model diagram Usage:It is used in shrink-wrap application and large system which built-in small phases or segments.

3.4 Iterative and Incremental Method

Description:

It is developed to overcome the weaknesses of the waterfall model. It starts with an initial planning and ends with deployment with the cyclic interactions in between. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system.

Usage:It is used in shrink-wrap application and large system which built-in small phases or segments. Also can be used in system has separated components, for example, ERP system. Which we can start with budget module as first iteration and then we can start with inventory module and so forth.

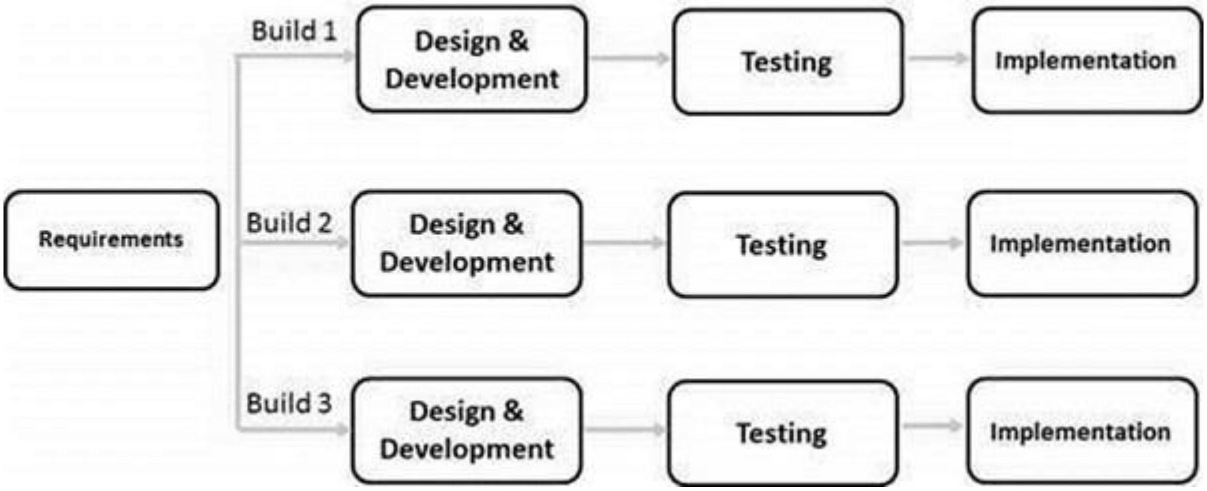


Figure 3.4: Iterative and Incremental Model

Advantages and Disadvantages :

Advantages	**Disadvantages**
Produces business value early in the development life cycle· Better use of scarce resources through proper increment in the definition.Can accommodate some change requests Partitioning the functions and features between increments. More focused on customer value than the linear approaches· Problems can be detected earlier	Requires heavy documentation .Follows a defined set of processes.Defines increments based in function and feature dependencies. Requiress more customer involvement than the linear approaches.Partioning the functions and features might be problematic. Integration between iteration can be issue if this is not considered during the development.

Table 3.4.: Advantages and Disadvantages of Iterative and Incremental model

3.5 PERFORMANCE REQUIREMENTS:

Factors such as Power, Memory, and Graphical User Interface (GUI) affect the performance of an application. During application development the developer must be very aware and anticipate these concerns so that the code is written in such a way that the resource utilization in system is balanced and distributed optimally to all the required applications.Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement.Specification is properly given, it is possible to design a system, which will fit into required environment. It is very difficult to change the system once it as been designed and on the other hand designing a system, which doesn’t cater to the requirements of the user, is of no use. The requirement specification for any system can be broadly stated as given below :

- The system should be able to interface with the existing system.
- The system should be accurate.
- The system should be better than the existing system.
- The existing system is completely dependent on the user to perform all the duties.

One of the trickiest problems you'll face when micro-optimizing an Android app is that your app is certain to be running on multiple types of hardware. Different versions of the VM running on different processors running at different speeds. It's not even generally the case that you can simply say "device X is a factor F faster/slower than device Y", and scale your results from one device to others. In particular, measurement on the emulator tells you very little about performance on any device. There are also huge differences between devices with and without a JIT: the best code for a device with a JIT is not always the best code for a device without.

To ensure your app performs well across a wide variety of devices, ensure your code is efficient at all levels and aggressively optimize your performance.

Hardware and Software Requirements :

Software Requirements :

Languages : Java,JavaScript,Php,Html

Front End : Android Studio

Back End : Sql Database

Hardware Requirements :

Android operating system

Upto 8GB Ram of any android mobile

Extendable Storage

4.ANDROID FUNDAMENTALS

4.1 Android Programming Languages:

In Android, basically programming is done in two languages **JAVA or C++** and **XML(Extension Markup Language)**. Nowadays KOTLIN is also preferred. The XML file deals with the design, presentation, layouts, blueprint,etc (as a front-end) while the JAVA or KOTLIN deals with working of buttons, variables, storing, etc (as a back-end).

Android Components:

The App components are the building blocks of Android. Each components have their own role and life-cycles i.e from launching of an app till the end. Some of these components depend upon other also. Each component have a definite purpose.

The four major app components are:

Activities: It deals with the UI and the User interactions to the screen. In other words, it is a User Interface that contains activities. These can be one or more depending upon the App. It starts when the application is launched. Atleast one activity is always present which is known as MainActivity.

The activity is implemented through the following Syntax:

```
Public class MainActivity extends Activity {
        //Processes
   }
```

Services: Services are the background actions performed by the app, these might be long running operations like a user playing music while surfing the Internet. A service might need another sub-services so as to perform a specific tasks. The main purpose of the Services is to provide non-stop working of the app without breaking any interaction with the user.

Syntax:

```
   Public class MainServices extends Services {
        //Code for the services
   }
```

Broadcast Receivers : A Broadcast is used to respond to the messages from other applications or from the System. For example, when the Battery of the phone is low, then the Android OS fires a Broadcasting message to launch Battery Saver function or app, after receiving the message the appropriate action is taken by the

app. Broadcast Receiver is the subclass of BroadcastReceiver class and each object is represented by an Intent objects.
Syntax:

```
Public class MyReceiver extends BroadcastReceiver{

        Public void onReceive(context,Intent)

}
```

Content Provider : Content Provider is used to transfer the data from one application to the others on request of the other application. These are handled by the class ContentResolver class. These class implements a set of APIs(Application Programming Interface) that enables the other applications to perform the transactions. Any Content Provider must implement the Parent Class of ContentProvider class.
Syntax:

```
Public class MycontentProvider extends ContentProvider{

        Public void onCreate( ) { }

}
```

4.2 Structure Layout of Android Application:

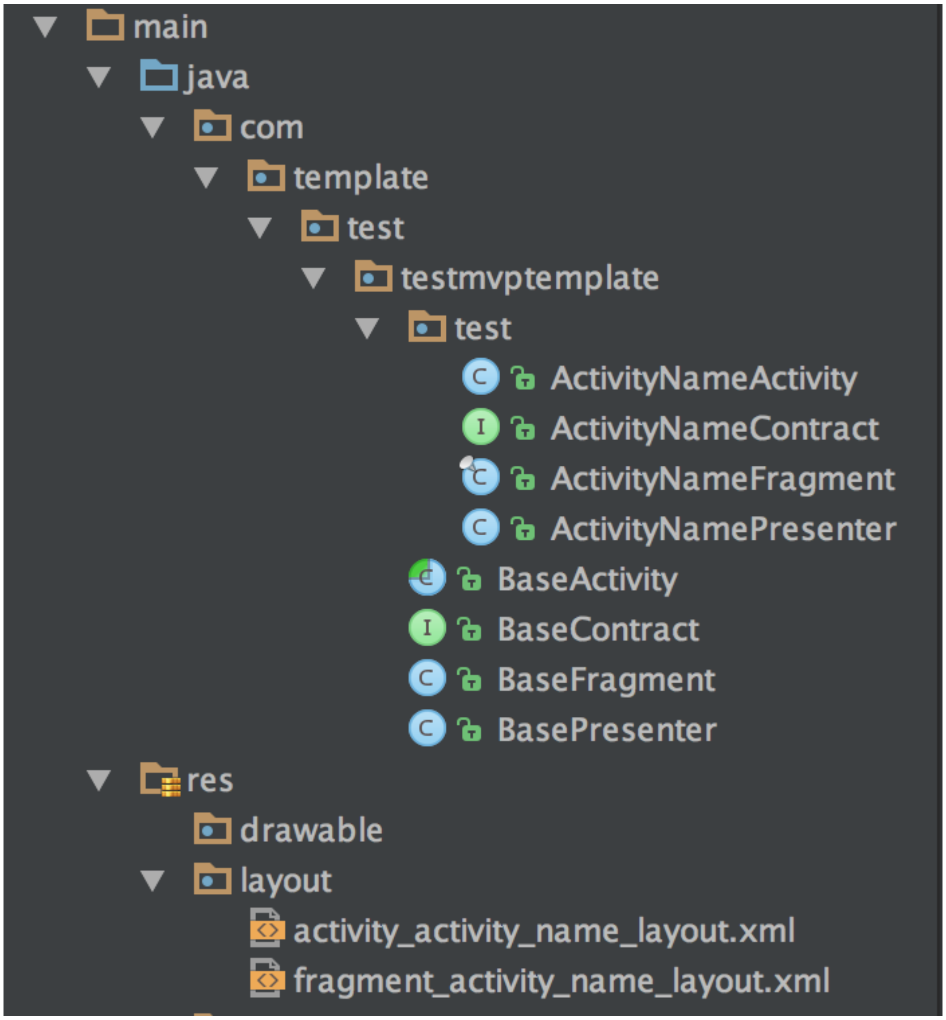
Android Mainfest :

It is a XML file which is the root of the project source set. It describes the essential information about the app and the Android build tools, the Android Operating System and the Google Play. It contains the permission that an app might need in order to perform the specific task. It also contains the Hardware and the Software features of the app, which determines the compatibility of an app on Play Store. It also includes the special activities like services, broadcast receiver, content providers, package name,etc.

The **JAVA folder** consist of the java files that are required to perform the background task of the app. It consist of the functionality of the buttons, calculation, storing, variables, toast(small popup message) , programming function, etc. The number of these files depends upon the type of activities created.

Res or Resource folder consist of the various resources that are used in the app. This consist of sub-folders like drawable, layout, mipmap, raw and values. The drawable consist of the images. The layout consist of the XML files that defines the user interface layout. These are stored in res.layout and are accessed as R.layout class. The raw consist of the Resources files like audio file or music file,etc. These are accessed through R.raw.filename. values are used to store the hardcoded strings(considered safe to store string values) values, integers and colors. It consist of various other directories like:-

Gradle: Gradle is an advance toolkit, which is used to manage the build process, that allows to define the flexible custom build configurations. Each build configuration can define its own set of code and resources, while reusing the parts common to all versions of your app. The Android plugin for Gradle works with the build toolkit to provide processes and configurable settings that are specific to building and testing Android applications. Gradle and the Android plugin run independent of Android Studio.



```
▼ main
    ▼ java
        ▼ com
            ▼ template
                ▼ test
                    ▼ testmvptemplate
                        ▼ test
                            C ActivityNameActivity
                            I ActivityNameContract
                            C ActivityNameFragment
                            C ActivityNamePresenter
                          C BaseActivity
                          I BaseContract
                          C BaseFragment
                          C BasePresenter
    ▼ res
          drawable
        ▼ layout
              activity_activity_name_layout.xml
              fragment_activity_name_layout.xml
```

Figure 4.2: Structure and layout of android manifest file

4.3 Life Cycle of Android Application :

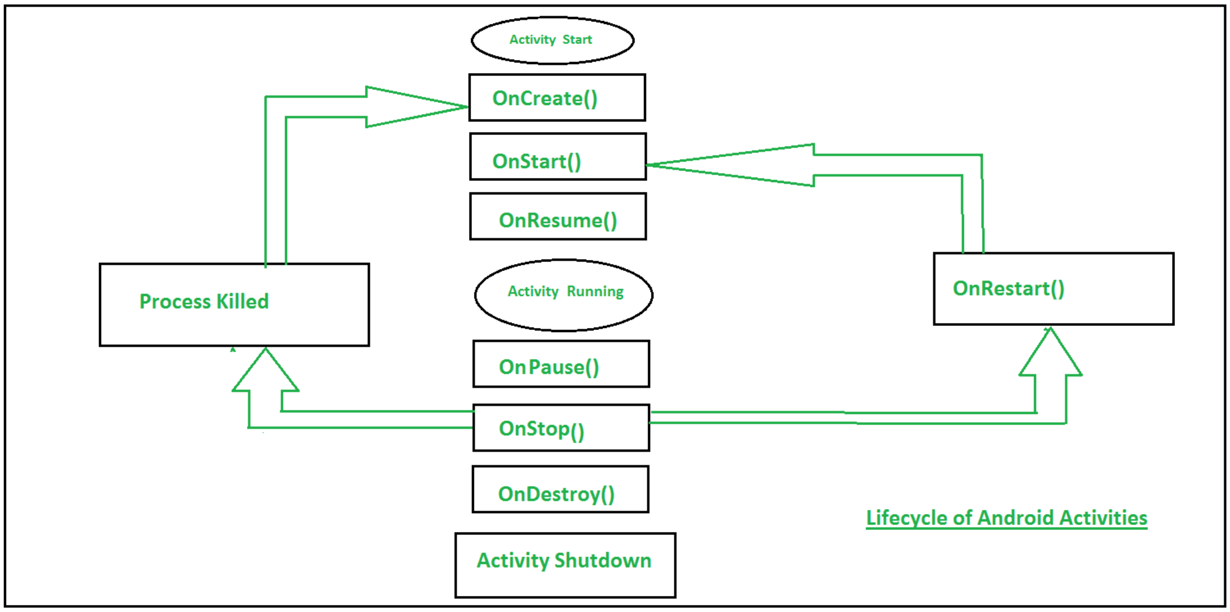


Figure 4.3: Life Cycle of Android Application

States of Android Lifecycle

1. OnCreate: This is called when activity is first created.
2. OnStart : This is called when activity become visible to the user.
3. OnResume : This is called when activity starts to interact with the user.
4. OnPause : This is called when activity is not visible to the user.
5. OnStop : This is called when activity is no longer visible.
6. OnRestart : This is called when activity is stopped, and restarted again.
7. OnDestroy **:** This is called when activity is to be closed or destroyed.

4.4. Types of Android Applications :

4.4.1 Native Application:

Native apps are considered to be the most reliable, showing extraordinary performance when compared to hybrid and web apps.

Native apps are developed for a specific mobile platform using particular programming languages and technologies. IOS apps, for example, are written in Objective-C and Swift, Android apps in Java or Kotlin. C++ is used for native Windows and Blackberry apps.

Native app development is preferred for: high-performance apps

Advantages of native applications

- Access to built-in features of the device
- Native UI/UX
- Available from app stores
- SDK for developers

Disadvantages of native app development

- High price and long development time
- Complicated and expensive maintenance and support
- Content not seen by search engines
- Support of multiple versions of the application

4.4.2 Web Application:

Web app development is the cheapest option for delivering a mobile user experience. But this doesn’t mean that web apps are a low-cost solution with limited quality. Sometimes they are sufficient to satisfy users’ expectations.

A web app, just like any website, is built with HTML, CSS and JavaScript. There are also web app frameworks - sets of tools which make the development process faster and more efficient. MeteorJS, React and Angular are the most popular JavaScript web development frameworks and one of the trends of 2018 on the web, just as Laravel for PHP.

Web apps are good for: news, weather, e-commerce, social apps, currency converters, expense managers, etc

Progressive web apps are breakthrough in modern web development. They allow building, cross – platform applications without significant disadvantages for the end user. If you are future project is not a complex business app with extensive features complex front end and back end you should consider building a PWA.

The time, when web applications were seen only as a cheap solution for new or small companies is already gone. Web apps are becoming a powerful business tool.

Advantages of web applications

- Build the development team fast
- Support every device, every platform and every version of OS
- Fast deployment of new features
- No need to support multiple versions of the software
- No app store approval
- Visible to search engines

Disadvantages of web apps

- Internet connection required to function properly
- Not available in the app store
- Not appropriate for apps with a complex frontend
- Not native experience
- Limited access to smartphone's features
- Ad blockers

4.4.3 Hybrid App:

A hybrid app is, in a way, a compromise between web and native app development. It combines the advantages of both types of app development.A hybrid application is technically a web app packed in a native app container. Like the web app, it is written in HTML, CSS and JavaScript. Unlike the web app, it is distributed through the app store or Over-the-Air for iOS devices. Another difference is that the hybrid app is reliant on the WebView and not the mobile browser.

For many it is surprising to learn that some of the worlds most popular apps turn out to be hybrid! This includes amazon (for IOS and android), evernote (for IOS and android) ,Netflix (for IOS and android). You can check out both versions of these apps to make sure that the user experience offered is on a high level,in most cases indistinguishable from the native applications.

Advantages of hybrid mobile apps

- Cross-platform
- Web development technology
- Lower price
- Reaching a wider audience
- Downloadable from the app store
- No ad blockers
- Good performance

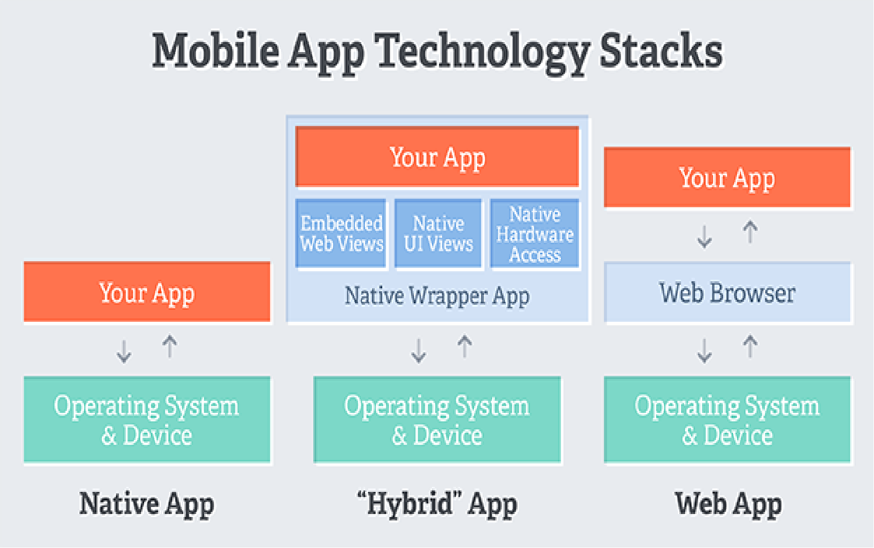


Figure 4.4: Mobile App Technology Stacks

5.SYSTEM DESIGN

5.1 Data Source Design :

A factory for connections to the physical data source that this datasource object represents. An alternative to the drivermanager facility, a datasource object is the preferred means of getting a connection. An object that implements the datasource interface will typically be registered with a naming service based on the javatmnaming and Directory (JNDI) API.

The datasource interface is implemented by a driver vendor. There are three types of implementations:

Basic implementation -- produces a standard Connection object

Connection pooling implementation -- produces a Connection object that will automatically participate in connection pooling. This implementation works with a middle-tier connection pooling manager.

Distributed transaction implementation -- produces a Connection object that may be used for distributed transactions and almost always participates in connection pooling. This implementation works with a middle-tier transaction manager and almost always with a connection pooling manager.

A datasource object has properties that can be modified when necessary. For example, if the data source is moved to a different server, the property for the server can be changed. The benefit is that because the data source's properties can be changed, any code accessing that data source does not need to be changed.

A driver that is accessed via a datasource object does not register itself with the drivermanager. Rather, adatasource object is retrieved though a lookup operation and then used to create a Connection object. With a basic implementation, the connection obtained through a datasource object is identical to a connection obtained through the drivermanager facility.

Public and Inherited Methods :

Method Name	Description
getConnection()	Attempts to establish a connection with the data source that this DataSource object represents.
getConnection(String Username, String Password)	Attempts to establish a connection with the data source that this DataSource object represents.
getLogWriter()	Retrives the log writter for this DataSource object
getLoginTimeout()	Gets the maximum time in second that this datsource can wait while attempting to connect to a database
getParentLogger()	Return the parent Logger of all the Loggers used by this data source.

Table 5.1: Public and Inherited methods used for data source design

5.2 Description about User Interface Design :

There are number of Layouts provided by Android which you will use in almost all the Android applications to provide different view, look and feel.

Sr.No	Layout & Description
1	**Linear Layout** Linear Layout is a view group that aligns all children in a single direction, vertically or horizontally.
2	**Relative Layout** Relative Layout is a view group that displays child views in relative positions.
3	**Table Layout** Table Layout is a view that groups views into rows and columns.
4	**Absolute Layout** Absolute Layout enables you to specify the exact location of its children.
5	**Frame Layout** The Frame Layout is a placeholder on screen that you can use to display a single view.
6	**List View** List View is a view group that displays a list of scrollable items.
7	**Grid View** Grid View is a View Group that displays items in a two-dimensional, scrollable grid.

Table 5.2: Different types of Android Layout Managers

5.2.1.Layout Attributes :

Each layout has a set of attributes which define the visual properties of that layout. There are few common attributes among all the layouts and their are other attributes which are specific to that layout. Following are common attributes and will be applied to all the layouts:

Sr .No	Attribute & Description
1	**android:id** This is the ID which uniquely identifies the view.
2	**android:layout_width** This is the width of the layout.
3	**android:layout_height** This is the height of the layout
4	**android:layout_marginTop** This is the extra space on the top side of the layout.
5	**android:layout_marginBottom** This is the extra space on the bottom side of the layout.
6	**android:layout_marginLeft** This is the extra space on the left side of the layout.

7	**android:layout_marginRight** This is the extra space on the right side of the layout.
8	**android:layout_gravity** This specifies how child Views are positioned.
9	**android:layout_weight** This specifies how much of the extra space in the layout should be allocated to the View.
10	**android:layout_x** This specifies the x-coordinate of the layout.
11	**android:layout_y** This specifies the y-coordinate of the layout.
12	**android:layout_width** This is the width of the layout.
13	**android:layout_width** This is the width of the layout.
14	**android:paddingLeft** This is the left padding filled for the layout.
15	**android:paddingRight** This is the right padding filled for the layout.

16	**android:paddingTop** This is the top padding filled for the layout.
17	**android:paddingBottom** This is the bottom padding filled for the layout.

Table 5.2.1: Different types of Android Layout Attributes

Here width and height are the dimension of the layout/view which can be specified in terms of dp (Density-independent Pixels), sp (Scale-independent Pixels), pt (Points which is 1/72 of an inch), px(Pixels), mm (Millimeters) and finally in (inches).

You can specify width and height with exact measurements but more often, you will use one of these constants to set the width or height −

- **android:layout_width=wrap_content** tells your view to size itself to the dimensions required by its content.
- **android:layout_width=fill_parent** tells your view to become as big as its parent view.

5.2.2.Gravity Attribute Values Description :

Gravity attribute plays important role in positioning the view object and it can take one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its size.
fill_vertical	0x70	Grow the vertical size of the object if needed so it completely fills its container.
center_horizontal	0x01	Place object in the horizontal center of its container, not changing its size.
fill_horizontal	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
center	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing

		its size.
fill	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
clip_vertical	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
clip_horizontal	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.
start	0x00800003	Push object to the beginning of its container, not changing its size.
end	0x00800005	Push object to the end of its container, not changing its size.

Table 5.2.2: Gravity value attributes description

View Identification :

A view object may have a unique ID assigned to it which will identify the View uniquely within the tree. The syntax for an ID, inside an XML tag is –

Android:id=”@+id/my_button”

Following is a brief description of @ and + signs −

- The at-symbol (@) at the beginning of the string indicates that the XML parser should parse and expand the rest of the ID string and identify it as an ID resource.
- The plus-symbol (+) means that this is a new resource name that must be created and added to our resources. To create an instance of the view object and capture it from the layout, use the following :

  Button myButton = (Button)findviewbyId(R.Id.Button);

5.3 Brief explaination about layout :

5.3.1.Linear Layout :

Linear Layout is a view group that aligns all children in a single direction, vertically or horizontally.

Sample Code Used for Linear Layout:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="vertical" >

  <TextView android:id="@+id/text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="This is a TextView" />

  <Button android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="This is a Button" />

  <!-- More GUI components go here  -->
```

```
</LinearLayout>
```

5.3.2 Relative Layout : Relative Layout is a view group that displays child views in relative positions.

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

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

            android:layout_height="fill_parent"
            android:layout_width=”fill_parent”
            android:orientation="vertical" >
            <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="This is a TextView" />

            <Button android:id="@+id/button"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="This is a Button" />

  </RelativeLayout>
```

6.PROCESS DESIGN

6.1.Android Software Stack:

Android is structured in the form of a software stack comprising applications, an operating system, run-time environment, middleware, services and libraries. This architecture can, perhaps, best be represented visually as outlined in Each layer of the stack, and the corresponding elements within each layer, are tightly integrated and carefully tuned to provide the optimal application development and execution environment for mobile devices.

Linux Kernal :

Positioned at the bottom of the Android software stack, the Linux Kernel provides a level of abstraction between the device hardware and the upper layers of the Android software stack. Based on Linux version 2.6, the kernel provides preemptive multitasking, low-level core system services such as memory, process and power management in addition to providing a network stack and device drivers for hardware such as the device display, Wi-Fi and audio.

The original Linux kernel was developed in 1991 by Linus Torvalds and was combined with a set of tools, utilities and compilers developed by Richard Stallman at the Free Software Foundation to create a full operating system referred to as GNU/Linux. Various Linux distributions have been derived from these basic underpinnings such as Ubuntu and Red Hat Enterprise Linux.

It is important to note, however, that Android only uses the Linux kernel. That said, it is worth noting that the Linux kernel was originally developed for use in traditional computers in the form of desktops and servers. In fact, Linux is now most widely deployed in mission critical enterprise server environments. It is a testament to both the power of today’s mobile devices and the efficiency and performance of the Linux kernel that we find this software at the heart of the Android software stack.

Android Run Time :

When an Android app is built within Android Studio it is compiled into an intermediate bytecode format (referred to as DEX format). When the application is subsequently loaded onto the device, the Android Runtime (ART) uses a process referred to as Ahead-of-Time (AOT) compilation to translate the bytecode down to the native instructions required by the device processor. This format is known as Executable and Linkable Format (ELF).

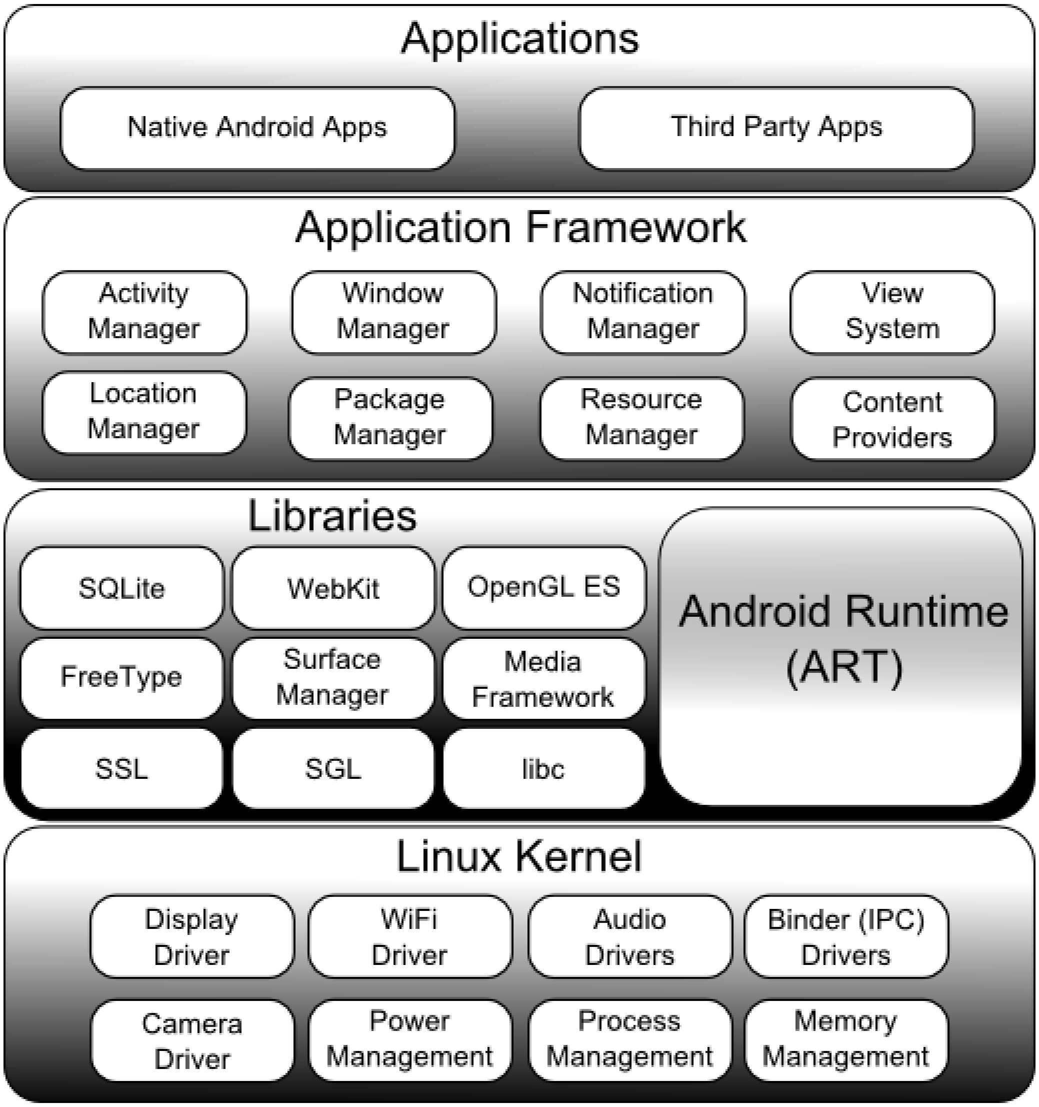


Figure 6.1: Android Software Stack

Each time the application is subsequently launched, the ELF executable version is run, resulting in faster application performance and improved battery life.

This contrasts with the Just-in-Time (JIT) compilation approach used in older Android implementations whereby the bytecode was translated within a virtual machine (VM) each time the application was launched.

Android Libraries :

In addition to a set of standard Java development libraries (providing support for such general purpose tasks as string handling, networking and file manipulation), the Android development environment also includes the Android Libraries. These are a set of Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access.

A summary of some key core Android libraries available to the Android developer is as follows:

- **android.app** – Provides access to the application model and is the cornerstone of all Android applications.
- **android.content** – Facilitates content access, publishing and messaging between applications and application components.
- **android.database** – Used to access data published by content providers and includes SQLite database management classes.
- **android.graphics** – A low-level 2D graphics drawing API including colors, points, filters, rectangles and canvases.
- **android.hardware** – Presents an API providing access to hardware such as the accelerometer and light sensor.
- **android.opengl** – A Java interface to the OpenGL ES 3D graphics rendering API.
- **android.os** – Provides applications with access to standard operating system services including messages, system services and inter-process communication.
- **android.media** – Provides classes to enable playback of audio and video.
- **android.net** – A set of APIs providing access to the network stack. Includes android.net.wifi, which provides access to the device’s wireless stack.

The Application Framework :

The Application Framework is a set of services that collectively form the environment in which Android applications run and are managed. This framework implements the concept that Android applications are constructed from reusable, interchangeable and replaceable components. This concept is taken a step further in that an application is also able to publish its capabilities along with any corresponding data so that they can be found and reused by other applications.

The Android framework includes the following key services:

- **Activity Manager** – Controls all aspects of the application lifecycle and activity stack.
- **Content Providers** – Allows applications to publish and share data with other applications.
- **Resource Manager** – Provides access to non-code embedded resources such as strings, color settings and user interface layouts.
- **Notifications Manager** – Allows applications to display alerts and notifications to the user.
- **View System** – An extensible set of views used to create application user interfaces.
- **Package Manager** – The system by which applications are able to find out information about other applications currently installed on the device.
- **Telephony Manager** – Provides information to the application about the telelphony services available on the device such as status and subscriber information.
- **Location Manager** – Provides access to the location services allowing an application to receive updates about location changes.

6.2 Modules Description :

- Login Activity
- Add Students using Profile Activity
- Take Attendance
- CGPA Calculator
- Notes Activity
- View Schedules
- Settings Activity

Login Activity : Login Activity allows user to enter into the application.In order to login user needs to register by clicking on the register button on the application.On clicking register on the application it shows the user to enter “Username” , “Password” and “Confirm Password”.After filling this by clicking on register button user can be registered.User details about name and password can be stored in sqllite package that the developer used in android studio.If the details entered by the user are not matching then the application will show a Toast message

Add Students Using profile Activity : Aftter entering into the application,it will give some options.Selecting the option profile,the user can add students.The user enter the information of student Name,Email-Id,Roll Number and contact Number.After clicking save button the details can be saved at sqllite package that is used in android studio for prearing the application.The user can also edit the student details by clicking on edit Student button.

Take Attendance: The application shows attendance button along with its icon.After clicking the attendance button it will ask user to select date and time to note the attendance.After Selecting time and date it will load students with a checkbox.If the student is present click once on the checkbox else if the student is absent double click on the checkbox it will be updated on the database.Student details along with attendance can be seen in profile Activity.

CGPA Calculator : It is used to calculate the total marks of student in gpa format.It consist of Seven subjects with edit text widget.After entering the subject marks in it the student can click on the button .After that the application will show the calculated total gpa of a student

Notes Activity: It will ask the student to enter the notes of any subject.The notes that is entered by the student can be saved and edited.Notes Activity will also give notification to the user.It will remind the student about the saved notes.The user can add any number of notes in the application along with subject title.Notes activity is prepared by using android studio with xml layouts and java files along with sql database

Schedules Activity :The student can add the any number of schedules.Schedules can be taken with the time and date.When the scheduled time is near, the mobile application will show the notification to the user.The Schedule can also be edited and viewed using the refresh button.

Settings Activity: It is used to clear all the student data at once**.**Used to clear any data saved at once in the application.

6.3 Uml Diagrams

6.3.1 Usecase diagram: Usecase diagram represents set of actors interacting with the system.

In this project the application user interacts with the system for performing operations adding students,cgpa calculation and saving notes,schedules using the android application

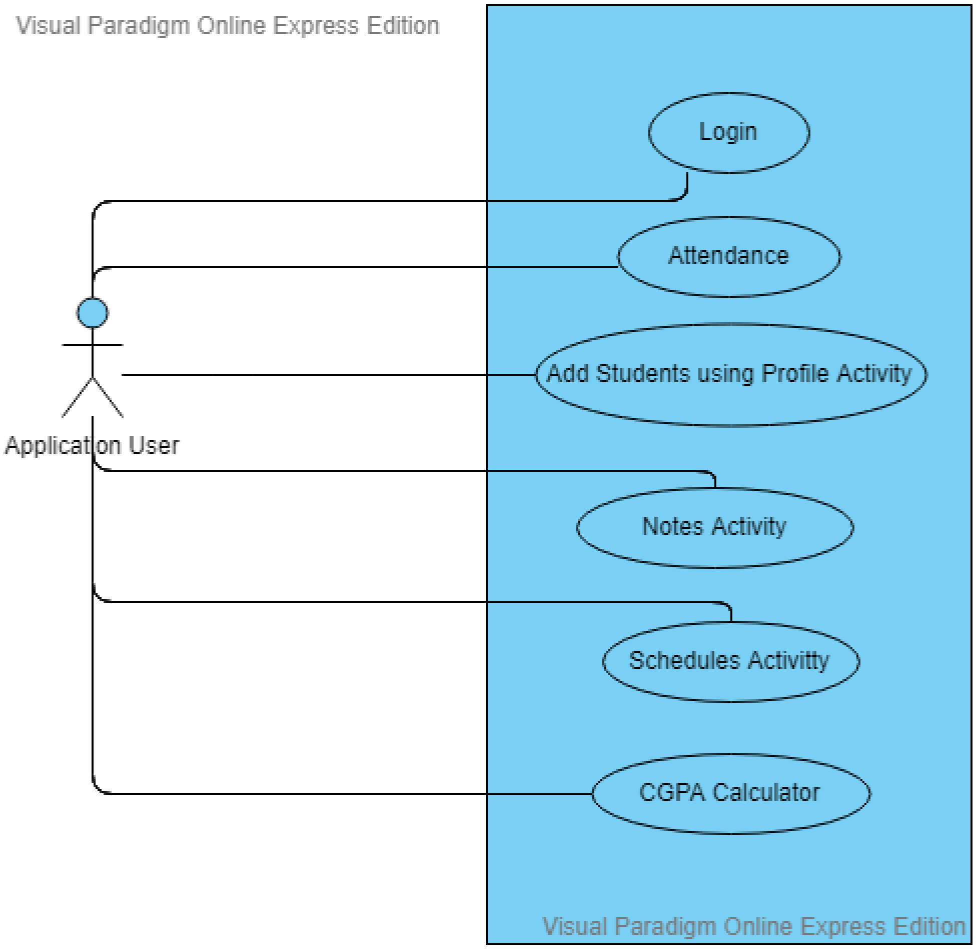

Visual Paradigm Online Express Edition
Login
Attendance
Add Students using Profile Activity
Application User
Notes Activity
Schedules Activitty
CGPA Calculator
Visual Paradigm Online Express Edition

Figure 6.3.1 :Usecase diagram

6.3.2 Class diagram :

Class diagram represents the static behavior of the system. Class diagram represents set of classes and relation between them.

In this project class diagram contains set of classes user, Profile and attendance for adding students using profile activity in the application and to save attendance and view attendance and graphical representation to edit and show

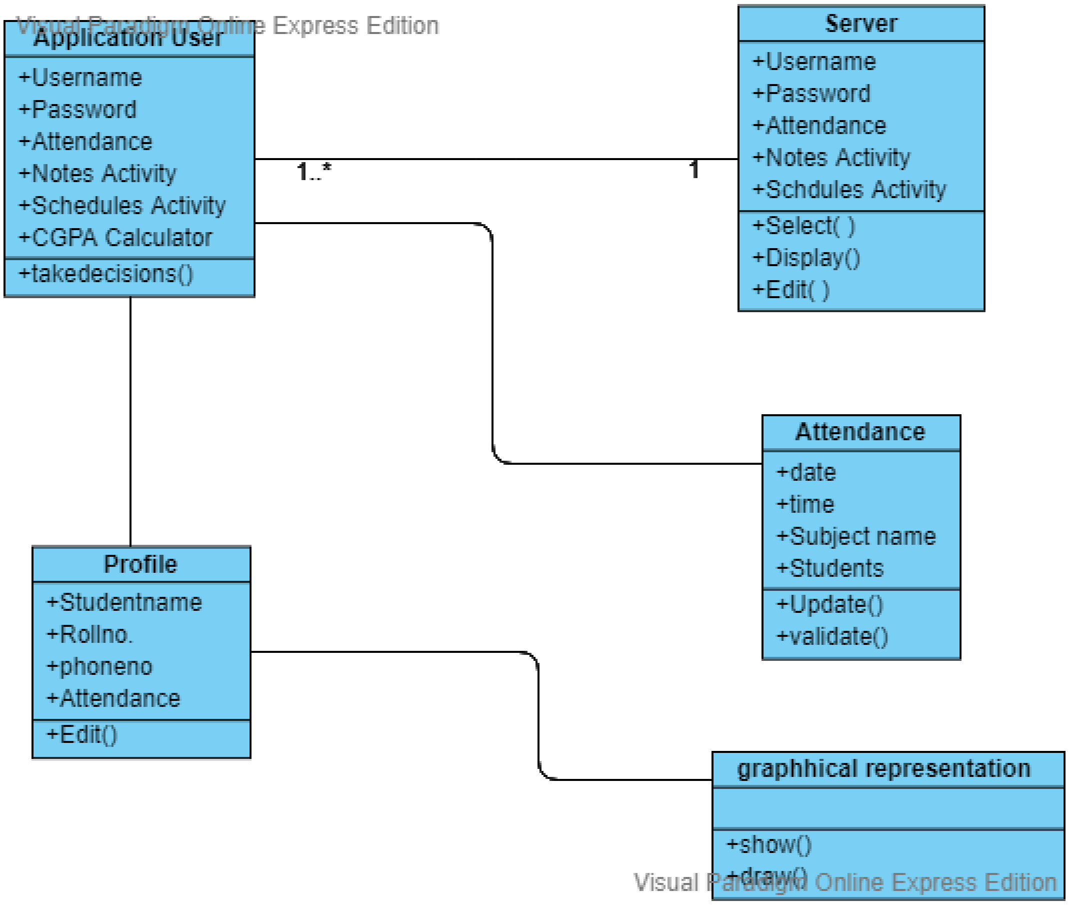


Figure 6.3.2: Class diagram

6.3.3 Sequence diagram:

Sequence diagram represents the sequence of operations performed between different objects.

In this project sequence diagram represents the sequence of operations between objects like user, user interface, server, profile and attendance.

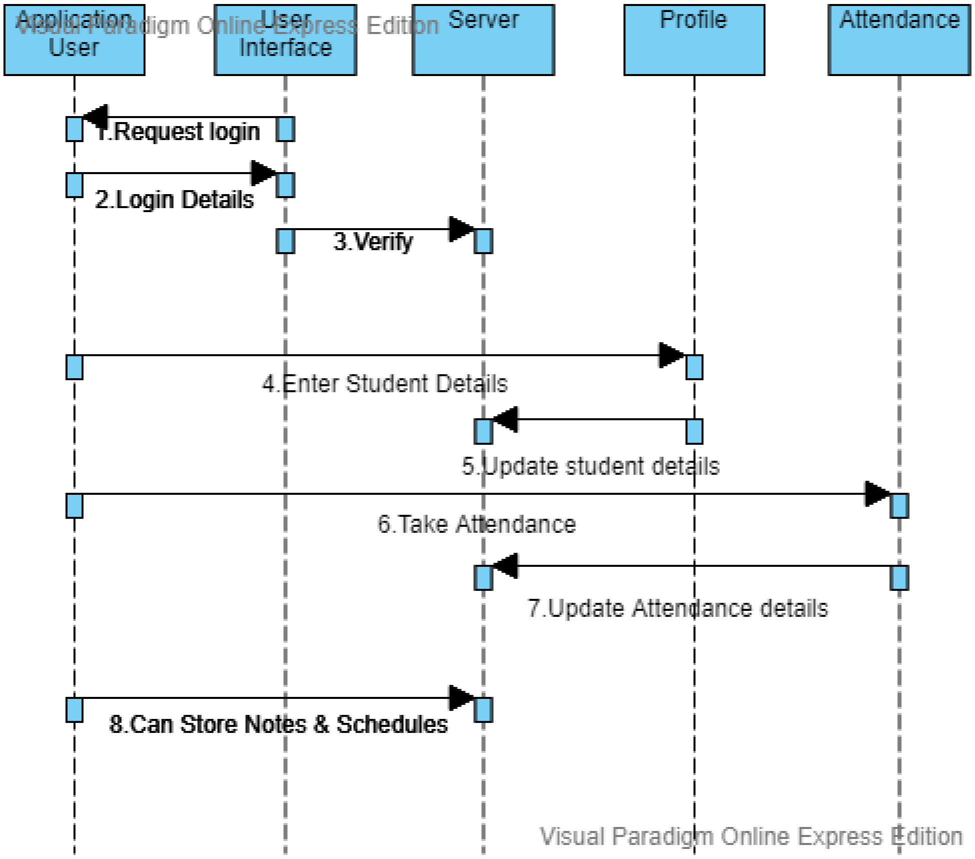


Figure 6.3.3:sequence diagram

6.3.4 Activity diagram:

Activity diagram represents the different actions performed with in an object.

In this project activity diagram represents the activities performed like attendance and other application user controls

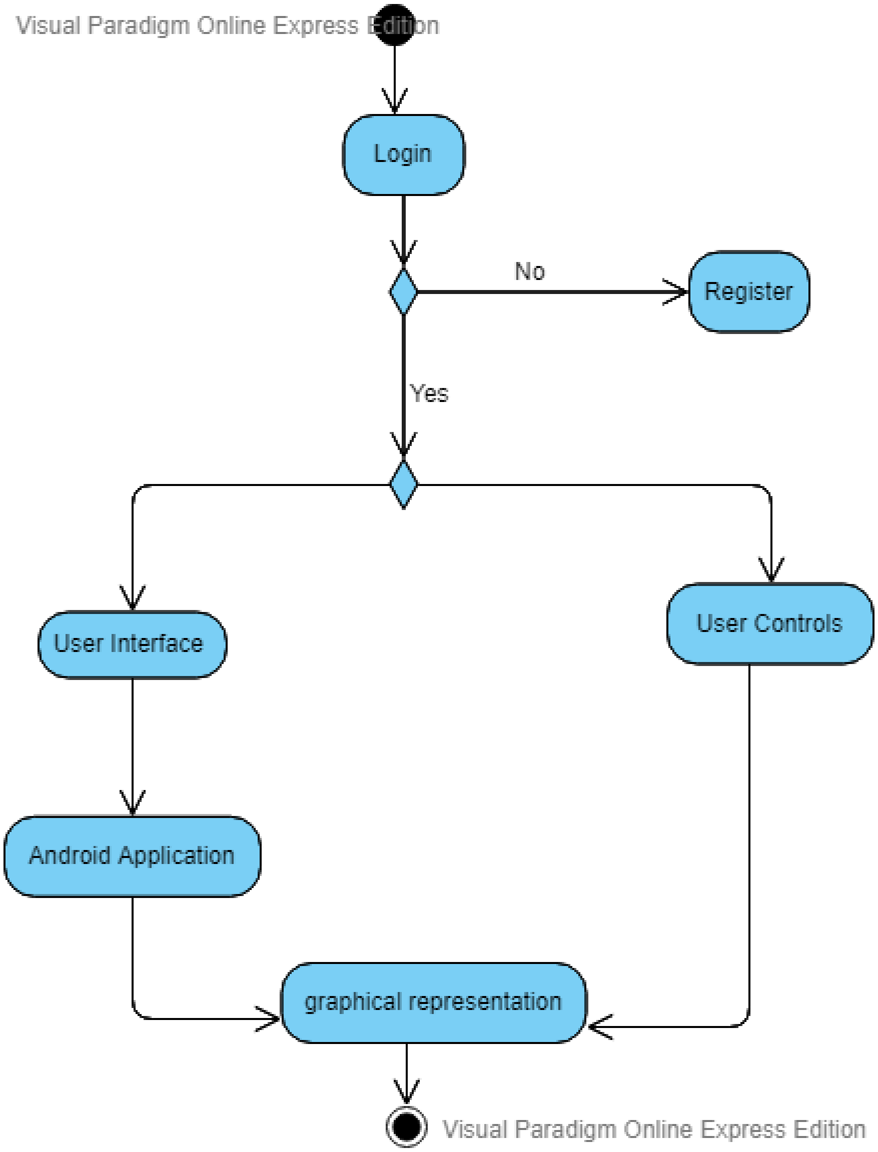


Figure 6.3.4 :activity diagram

6.3.5 Collaboration diagram :

Colloboration diagram represents interaction between any two objects is called collaboration diagram.

In this project the user can interact students by adding the student details and can save notes and other schedules, similarly the user interface can interact with the server as well as user.

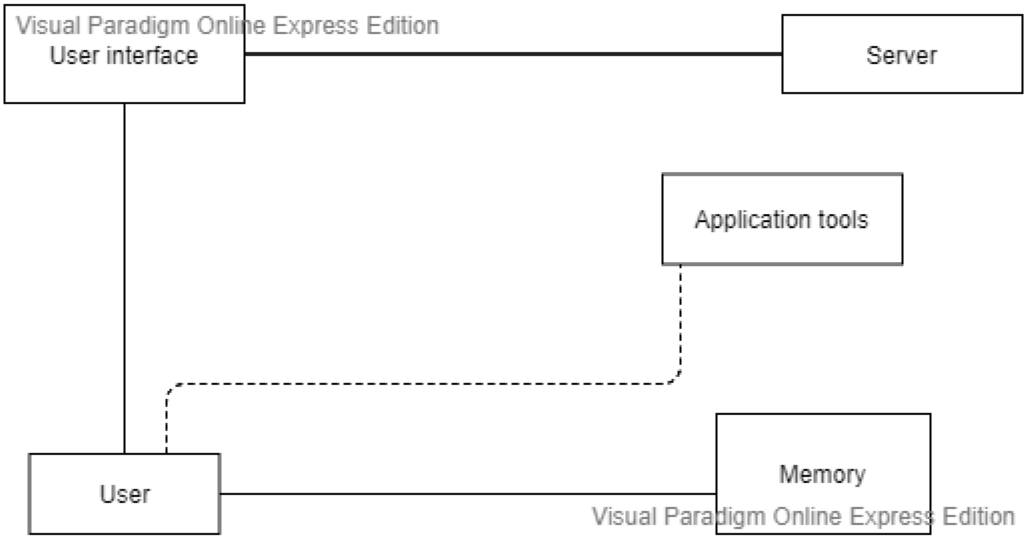

Visual Paradigm Online Express Edition
User interface
Server
Application tools
User
Memory
Visual Paradigm Online Express Edition

Figure 6.3.5 :Collaboration Diagram

7.CODING

7.1 Code Description:

Login:

Login page is used for authenticating whether the user is authorized or not.If the user valid the control transfer to home page.If not valid the message displayed as login error.

Register:

Register page is used for registering the user it takes the user basic details as input and stores in database for future use.

Home

Home page contains five different types of Icons.Each Icon performs different types of activity. When the user click on any one Icon the control transfers to the particular activity.

Attendance

Attendance page can be used to take attendance for students by clicking on load button for particular time and date of each subject.

Profile

Profile page can be used to add students with their registration number, name and contact can also be saved. A particular student details can also be edited using the edit student button.

Schedules

Schedules page can be used to set a particular schedule at a particular date and time and also mentioning the name of the subject.The application will notify the user at notification bar of the mobile device.

Notes

Notes page can be used to note some important notes and saved in the application.The available notes can also be shown as notification in the notification bar of the mobile device.

CGPA Calculator

Cgpa Calculator can also be known as Cumulative grade points average(cgpa). The application contains 8 text widgets which is used to calculate total of the grade points average.

7.2 Technology used :

Java:

Java is a programming language and a platform.

Java is a high level, robust, object-oriented and secure programming language.

platform: Any hardware or software environment in which a program runs, is known as a platform. Since Java has a runtime environment (JRE) and API, it is called a platform.

According to Sun, 3 billion devices run Java. There are many devices where Java is currently used. Some of them are as follows:

Desktop Applications such as acrobat reader, media player, antivirus, etc.

Web Applications such as irctc.co.in, javatpoint.com, etc.

Enterprise Applications such as banking applications.

Mobile

Embedded System

Smart Card

Robotics

Games, etc.

Types of Java Applications

There are mainly 4 types of applications that can be created using Java programming:

1) Standalone Application

Standalone applications are also known as desktop applications or window-based applications. These are traditional software that we need to install on every machine. Examples of standalone application are Media player, antivirus, etc. AWT and Swing are used in Java for creating standalone applications.

An application that runs on the server side and creates a dynamic page is called a web application. Currently, Servlet, JSP, Struts, Spring, Hibernate, JSF, etc. technologies are used for creating web applications in Java.

3) Enterprise Application

An application that is distributed in nature, such as banking applications, etc. is called enterprise application. It has advantages of the high-level security, load balancing, and clustering. In Java, EJB is used for creating enterprise applications.

4) Mobile Application

An application which is created for mobile devices is called a mobile application. Currently, Android and Java ME are used for creating mobile applications.

Java Platforms / Editions

There are 4 platforms or editions of Java:

1) Java SE (Java Standard Edition)

It is a Java programming platform. It includes Java programming APIs such as java.lang, java.io, java.net, java.util, java.sql, java.math etc. It includes core topics like OOPs, String, Regex, Exception, Inner classes, Multithreading, I/O Stream, Networking, AWT, Swing, Reflection, Collection, etc.

2) Java EE (Java Enterprise Edition)

It is an enterprise platform which is mainly used to develop web and enterprise applications. It is built on the top of the Java SE platform. It includes topics like Servlet, JSP, Web Services, EJB, JPA, etc.

3) Java ME (Java Micro Edition)

It is a micro platform which is mainly used to develop mobile applications.

4) JavaFX

It is used to develop rich internet applications. It uses a light-weight user interface API.

Features of Java:

The primary objective of Java programming language creation was to make it portable, simple and secure programming language. Apart from this, there are also some excellent features which play an important role in the popularity of this language. The features of Java are also known as java buzzwords.

A list of most important features of Java language is given below.

Simple:

Java is very easy to learn, and its syntax is simple, clean and easy to understand. According to Sun, Java language is a simple programming language because:

- Java syntax is based on C++ (so easier for programmers to learn it after C++).
- Java has removed many complicated and rarely-used features, for example, explicit pointers, operator overloading, etc.
- There is no need to remove unreferenced objects because there is an Automatic Garbage Collection in Java.

Object-oriented:

Java is an object-oriented programming language. Everything in Java is an object. Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behavior.

Object-oriented programming (OOPs) is a methodology that simplifies software development and maintenance by providing some rules.

Java is platform independent because it is different from other languages like C, C++, etc. which are compiled into platform specific machines while Java is a write once, run anywhere language. A platform is the hardware or software environment in which a program runs.

There are two types of platforms software-based and hardware-based. Java provides a software-based platform.

The Java platform differs from most other platforms in the sense that it is a software-based platform that runs on the top of other hardware-based platforms. It has two components:

1. Runtime Environment
2. API(Application Programming Interface)

Secured:

Java is best known for its security. With Java, we can develop virus-free systems. Java is secured because:

No explicit pointer

Java Programs run inside a virtual machine sandbox

Classloader: Classloader in Java is a part of the Java Runtime Environment(JRE) which is used to load Java classes into the Java Virtual Machine dynamically. It adds security by separating the package for the classes of the local file system from those that are imported from network sources.

Bytecode Verifier: It checks the code fragments for illegal code that can violate access right to objects.

Security Manager: It determines what resources a class can access such as reading and writing to the local disk.

Java language provides these securities by default. Some security can also be provided by an application developer explicitly through SSL, JAAS, Cryptography, etc.

Robust

Robust simply means strong. Java is robust because:

- It uses strong memory management.
- There is a lack of pointers that avoids security problems.
- There is automatic garbage collection in java which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore.
- There are exception handling and the type checking mechanism in Java. All these points make Java robust.

Architecture-neutral

Java is architecture neutral because there are no implementation dependent features, for example, the size of primitive types is fixed.

In C programming, int data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture. However, it occupies 4 bytes of memory for both 32 and 64-bit architectures in Java.

Portable

Java is portable because it facilitates you to carry the Java bytecode to any platform. It doesn't require any implementation.

Dynamic:

Java is considered to be more dynamic then c or c++ since it is designed to adopt to an evolving environment. Java programs can carry an extensive amount of run time information that can be used to verify and resolve accesses to objects at run time.

XML :

- XML stands for eXtensible Markup Language
- XML is a markup language much like HTML
- XML was designed to store and transport data
- XML was designed to be self-descriptive
- XML is a W3C Recommendation
- It has sender information.
- It has receiver information
- It has a heading
- It has a message body.

But still, the XML above does not DO anything. XML is just information wrapped in tags.

The Difference Between XML and HTML

XML and HTML were designed with different goals:

- XML was designed to carry data - with focus on what data is
- HTML was designed to display data - with focus on how data looks
- XML tags are not predefined like HTML tags are

- XML Does Not Use Predefined Tags
- The XML language has no predefined tags.
- The tags in the example above (like <to> and <from>) are not defined in any XML standard. These tags are "invented" by the author of the XML document.
- HTML works with predefined tags like <p>, <h1>, <table>, etc.
- With XML, the author must define both the tags and the document structure
- XML is Extensible
- Most XML applications will work as expected even if new data is added (or removed).
- Imagine an application designed to display the original version of note.xml (<to> <from> <heading> <body>).
- Then imagine a newer version of note.xml with added <date> and <hour> elements, and a removed <heading>.
- The way XML is constructed, older version of the application can still work:

XML Simplifies Things

- It simplifies data sharing
- It simplifies data transport
- It simplifies platform changes
- It simplifies data availability

Many computer systems contain data in incompatible formats. Exchanging data between incompatible systems (or upgraded systems) is a time-consuming task for web developers. Large amounts of data must be converted, and incompatible data is often lost.

XML stores data in plain text format. This provides a software- and hardware-independent way of storing, transporting, and sharing data.XML also makes it easier to expand or upgrade to new operating systems, new applications, or new browsers, without losing data.With XML, data can be available to all kinds of "reading machines" like people, computers, voice machines, news feeds, etc.XML is used in many aspects of web development.XML is often used to separate data from presentation.

XML Separates Data from Presentation

XML does not carry any information about how to be displayed.

The same XML data can be used in many different presentation scenarios.

Because of this, with XML, there is a full separation between data and presentation.

XML is Often a Complement to HTML

In many HTML applications, XML is used to store or transport data, while HTML is used to format and display the same data.

XML Separates Data from HTML

When displaying data in HTML, you should not have to edit the HTML file when the data changes.

With XML, the data can be stored in separate XML files.

With a few lines of JavaScript code, you can read an XML file and update the data content of any HTML page.

Name Conflicts :

In XML, element names are defined by the developer. This often results in a conflict when trying to mix XML documents from different XML applications.

XML Namespaces - The xmlns Attribute :

When using prefixes in XML, a **namespace** for the prefix must be defined.The namespace can be defined by an **xmlns** attribute in the start tag of an element.

The xmlns attribute in the first <table> element gives the h: prefix a qualified namespace.

The xmlns attribute in the second <table> element gives the f: prefix a qualified namespace.When a namespace is defined for an element, all child elements with the same prefix are associated with the same namespace.

XML Naming RulesXML elements must follow these naming rules:

- Element names are case-sensitive
- Element names must start with a letter or underscore
- Element names cannot start with the letters xml (or XML, or Xml, etc)
- Element names can contain letters, digits, hyphens, underscores, and periods
- Element names cannot contain spaces.

7.3 Sample Code:

AppBase.Java

```
package com.example.hp.logactivity;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.GridView;

import java.util.ArrayList;

public class AppBase extends AppCompatActivity {

    public static ArrayList<String> divisions;
    public static DatabaseHandler handler;
    public static Activity activity;
    ArrayList<String> basicFields;
    GridAdapter adapter;
    GridView gridView;

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        super.onCreateOptionsMenu(menu);
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.mai_menu, menu);
        return true;
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.base_layout);
        basicFields = new ArrayList<>();
        handler = new DatabaseHandler(this);
        activity = this;

        getSupportActionBar().show();
        divisions = new ArrayList<>();
        divisions.add("S1 COMPUTER SCIENCE");
        divisions.add("S2 COMPUTER SCIENCE");
        divisions.add("S3 COMPUTER SCIENCE");
        divisions.add("S4 COMPUTER SCIENCE");
        divisions.add("S5 COMPUTER SCIENCE");
        divisions.add("S6 COMPUTER SCIENCE");
        divisions.add("S7 COMPUTER SCIENCE");
        gridView = (GridView) findViewById(R.id.grid);
        basicFields.add("ATTENDANCE");
        basicFields.add("SCHEDULER");
        basicFields.add("NOTES");
        basicFields.add("PROFILE");
        basicFields.add("CGPA CALCULATOR");
        adapter = new GridAdapter(this, basicFields);
        gridView.setAdapter(adapter);
    }
    public void loadSettings(MenuItem item) {
        Intent launchIntent = new Intent(this, SettingsActivity.class);
        startActivity(launchIntent);
    }

    public void loadAbout(MenuItem item) {
        Intent launchIntent = new Intent(this, About.class);
        startActivity(launchIntent);
    }
}
```

LOGIN.Java

```
package com.example.hp.logactivity;

import android.app.Dialog;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

public class LoginActivity extends AppCompatActivity {
 EditText mTextUsername;
 EditText mTextPassword;
 Button mButtonLogin;
 TextView mTextViewRegister;
 DatabaseHelper db;
 ViewGroup progressView;
    protected boolean isProgressShowing = false;

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

        Dialog dialog = new
Dialog(this,android.R.style.Theme_Translucent_NoTitleBar);
        View v = this.getLayoutInflater().inflate(R.layout.progress_bar,null);
        dialog.setContentView(v);
        dialog.show();

        db = new DatabaseHelper(this);
        mTextUsername = (EditText)findViewById(R.id.edittext_username);
        mTextPassword = (EditText)findViewById(R.id.edittext_password);
        mButtonLogin = (Button)findViewById(R.id.button_login);
        mTextViewRegister = (TextView)findViewById(R.id.textview_register);
        mTextViewRegister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent registerIntent = new Intent(LoginActivity.this,
RegisterActivity.class);
                startActivity(registerIntent);
            }
        });

        mButtonLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String user = mTextUsername.getText().toString().trim();
                String pwd = mTextPassword.getText().toString().trim();
                Boolean res = db.checkUser(user, pwd);
                if(res == true)
                {
                    Intent HomePage = new Intent(LoginActivity.this,AppBase.class);
                    startActivity(HomePage);
                }
                else
                {
                    Toast.makeText(LoginActivity.this,"Login
Error",Toast.LENGTH_SHORT).show();
                }
            }
        });
```

```
    }

    public void showProgressingView()
    {

        if (!isProgressShowing) {
            View view;
            view = findViewById(R.id.progressBar1);
            view.bringToFront();
        }
    }

    public void hideProgressingView() {
        View v = this.findViewById(android.R.id.content).getRootView();
        ViewGroup viewGroup = (ViewGroup) v;
        viewGroup.removeView(progressView);
        isProgressShowing = false;
    }
}
```

Attendance.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:weightSum="10"
        android:id="@+id/abcd"
        android:orientation="horizontal">
        <Spinner
```

```
            android:id="@+id/attendanceSpinner"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:textSize="20sp">
        </Spinner>
        <Button
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@color/colorPrimaryDark"
            android:textColor="#ffffff"
            android:textStyle="bold"
            android:id="@+id/loadButton"
            android:layout_marginTop="3dp"
            android:layout_weight="8"
            android:text="Load" />

    </LinearLayout>
    <ListView
        android:layout_below="@id/abcd"
        android:id="@+id/attendanceListViwe"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginTop="10dp"
        android:layout_above="@+id/buttonSaveAttendance">
    </ListView>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="SAVE"
        android:background="@color/colorPrimary"
        android:textColor="#ffffff"
        android:textStyle="bold"
        android:textSize="16sp"
        android:id="@+id/buttonSaveAttendance"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />

</RelativeLayout>
```

Login.xml

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

          xmlns:app="http://schemas.android.com/apk/res-auto"
          xmlns:tools="http://schemas.android.com/tools"
          android:layout_width="match_parent"
            android:layout_height="match_parent"
            tools:context="com.example.hp.LoginActivity"
            android:orientation="vertical"
            android:id="@+id/container"
            android:background="@drawable/img_background"
            android:gravity="center_horizontal">

            <ImageView
                android:id="@+id/imageView"
                android:layout_width="120dp"
                android:layout_height="150dp"
                android:layout_marginTop="10dp"
                android:foreground="@drawable/logo"
                android:layout_alignParentTop="true"
                android:layout_centerHorizontal="true" />
            <LinearLayout
```

```
    android:gravity="center_horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">


<EditText
        android:id="@+id/edittext_username"
        android:layout_width="190dp"
        android:layout_height="40dp"
        android:layout_marginTop="50dp"
        android:paddingLeft="2dp"
        android:background="#ffffff"
        android:drawableLeft="@drawable/username"
        android:hint="@string/username" />

    <EditText
        android:id="@+id/edittext_password"
        android:layout_width="190dp"
        android:layout_height="40dp"
        android:layout_marginTop="20dp"
        android:background="#ffffff"
        android:drawableLeft="@drawable/password"
        android:hint="@string/password"/>
</LinearLayout>

<Button
    android:id="@+id/button_login"
    android:layout_width="190dp"
    android:layout_height="40dp"
    android:layout_marginTop="20dp"
    android:background="#04ea00"
    android:textColor="#ffffff"
    android:text="@string/login"/>
```

```
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="10dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="35dp"
        android:textColor="#ffffff"
        android:text="@string/not_registered"/>

    <TextView
        android:id="@+id/textview_register"
        android:layout_width="wrap_content"
        android:layout_height="35dp"
        android:paddingLeft="10dp"
        android:textStyle="bold"
        android:textColor="#04ea00"
        android:text="@string/register"/>

</LinearLayout>
</LinearLayout>
```

8.System Testing

A level of software testing is a process where every unit or component of a software/system is tested. The main goal of system testing is to evaluate the system's compliance with the specified needs.

There are many different testing levels which help to check behavior and performance for software testing. These testing levels are designed to recognize missing areas and reconciliation between the development lifecycle states. In SDLC models there are characterized phases such as requirement gathering, analysis, design, coding or execution, testing, and deployment.

Testing is important because software bugs could be expensive or even dangerous. Software bugs can potentially cause monetary and human loss, and history is full of such examples.

In April 2015, Bloomberg terminal in London crashed due to software glitch affected more than 300,000 traders on financial markets. It forced the government to postpone a 3bn pound debt sale.Nissan cars have to recall over 1 million cars from the market due to software failure in the airbag sensory detectors. There has been reported two accident due to this software failure.Starbucks was forced to close about 60 percent of stores in the U.S and Canada due to software failure in its POS system. At one point store served coffee for free as they unable to process the transaction.Some of the Amazon’s third party retailers saw their product price is reduced to 1p due to a software glitch. They were left with heavy losses.Vulnerability in Window 10. This bug enables users to escape from security sandboxes through a flaw in the win32k system.In 2015 fighter plane F-35 fell victim to a software bug, making it unable to detect targets correctly.China Airlines Airbus A300 crashed due to a software bug on April 26, 1994, killing 264 innocent liveIn 1985, Canada's Therac-25 radiation therapy machine malfunctioned due to software bug and delivered lethal radiation doses to patients, leaving 3 people dead and critically injuring 3 others.In April of 1999, a software bug caused the failure of a $1.2 billion military satellite launch, the costliest accident in history.All these phases go through the process of software testing levels. There are mainly four testing levels are:

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing

1) **Unit testing:**

A Unit is a smallest testable portion of system or application which can be compiled, liked, loaded, and executed. This kind of testing helps to test each module separately.

The aim is to test each part of the software by separating it. It checks that component are fulfilling functionalities or not. This kind of testing is performed by developers.

2) **Integration testing:**

Integration means combining. For Example, In this testing phase, different software modules are combined and tested as a group to make sure that integrated system is ready for system testing.

Integrating testing checks the data flow from one module to other modules. This kind of testing is performed by testers.

3) **System testing:**

System testing is performed on a complete, integrated system. It allows checking system's compliance as per the requirements. It tests the overall interaction of components. It involves load, performance, reliability and security testing.

System testing most often the final test to verify that the system meets the specification. It evaluates both functional and non-functional need for the testing.

4) **Acceptance testing:**

Acceptance testing is a test conducted to find if the requirements of a specification or contract are met as per its delivery. Acceptance testing is basically done by the user or customer. However, other stockholders can be involved in this process.A level of software testing is a process where every unit or component of a software/system is tested.The primary goal of system testing is to evaluate the system's compliance with the specified needs.In Software Engineering, four main levels of testing are Unit Testing, Integration Testing, System Testing and Acceptance Testing.

8.1.1 Test Cases

Login

Test Case ID	Description	Expected Resullt	Actual Result	Testing Phase	Seviour	Priority	Pass/Fail
LG000	To check login page without entering username and password	Login error message should be displayed		Design	High	High	
LG001	To check login page with entering username and without password	Login error message should be displayed		Design	High	High	
LG002	To check login page with entering username and 1 character password and click on login button	Login error message should be displayed		Design	High	High	
LG003	To check login page with entering username and 2 character password and click on login button	Login error message should be displayed		Design	High	High	
LG004	To check login page with entering	Login error message		Design	High	High	

	username and 3 character password and click on login button	should be displayed					
LG005	To check login page with entering username and 4 character password and click on login button	Login error message should be displayed		Design	High	High	
LG006	To check login page with entering username and 5 character password and click on login button	Login error message should be displayed		Design	High	High	
LG007	To check login page with correct username and password and click on login button	Home page should be displayed		Design	High	High	

Table 8.1.1:LOGIN TEST CASES

Registration

Test Case ID	Description	Expected Resullt	Actual Result	Testing Phase	Seviour	Priority	Pass/Fail
RG000	To check registration page without entering any data	Registration error message should be displayed		Design	High	High	
RG001	To check registration page with 1 character as username and click on submit button	Registration error message should be displayed		Design	High	High	
RG002	To check registration page with 2 character as username and click on submit button	Registration error message should be displayed		Design	High	High	
RG003	To check registration page with 3 character as username and click on submit button	Registration error message should be displayed		Design	High	High	

RG004	To check registration page with 4 character as username and click on submit button	Registration error message should be displayed		Design	High	High	
RG005	To check registration page with 5 character as username and click on submit button	Registration error message should be displayed		Design	High	High	
RG006	To check registration page with correct username and click on submit button	You are registered message should be displayed		Design	High	High	

TABLE 8.1.2: REGISTRATION PAGE TEST CASES

8.2 Test Report/Result:

Login:

Test Case ID	Description	Expected Resullt	Actual Result	Testing Phase	Seviour	Priority	Pass/Fail
LG000	To check login page without entering username and password	Login error message should be displayed	Error message displayed showing incorrect login details	Execution	High	High	pass
LG001	To check login page with entering username and without password	Login error message should be displayed	Error message displayed showing incorrect login details	Execution	High	High	pass
LG002	To check login page with entering username and 1 character password and click on login button	Login error message should be displayed	Error message displayed showing incorrect login details	Execution	High	High	pass
LG003	To check login page with entering username and 2 character password	Login error message should be displayed	Error message displayed showing incorrect login details	Execution	High	High	pass

LG004	To check login page with entering username and 3 character password and click on login button	Login error message should be displayed	Error message displayed showing incorrect login details	Execution	High	High	pass
LG005	To check login page with entering username and 4 character password and click on login button	Login error message should be displayed	Error message displayed showing incorrect login details	Execution	High	High	pass
LG006	To check login page with entering username and 5 character password and click on login button	Login error message should be displayed	Error message displayed showing incorrect login details	Execution	High	High	pass
LG007	To check login page with correct username and password and click on login button	Home page should be displayed	Home page opened	Execution	High	High	pass

TABLE 8.2.1: LOGIN TEST CASES REPORT

Registration

Test Case ID	Description	Expected Resullt	Actual Result	Testing Phase	Seviour	Priority	Pass/Fail
RG000	To check registration page without entering any data	Registration error message should be displayed	Error message displayed showing registration error	Execution	High	High	pass
RG001	To check registration page with 1 character as username and click on submit button	Registration error message should be displayed	Error message displayed showing registration error	Execution	High	High	pass
RG002	To check registration page with 2 character as username and click on submit button	Registration error message should be displayed	Error message displayed showing registration error	Execution	High	High	pass
RG003	To check registration page with 3 character as username and click on submit button	Registration error message should be displayed	Error message displayed showing registration error	Execution	High	High	pass

RG004	To check registration page with 4 character as username and click on submit button	Registration error message should be displayed	Error message displayed showing registration error	Execution	High	High	pass
RG005	To check registration page with 5 character as username and click on submit button	Registration error message should be displayed	Error message displayed showing registration error	Execution	High	High	pass
RG006	To check registration page with correct username and click on submit button	You are registered message should be displayed	Error message displayed showing registration error	Execution	High	High	pass

TABLE 8.2.2: REGISTRATION PAGE TEST CASES REPORT

9. Output Screens

Login Screen

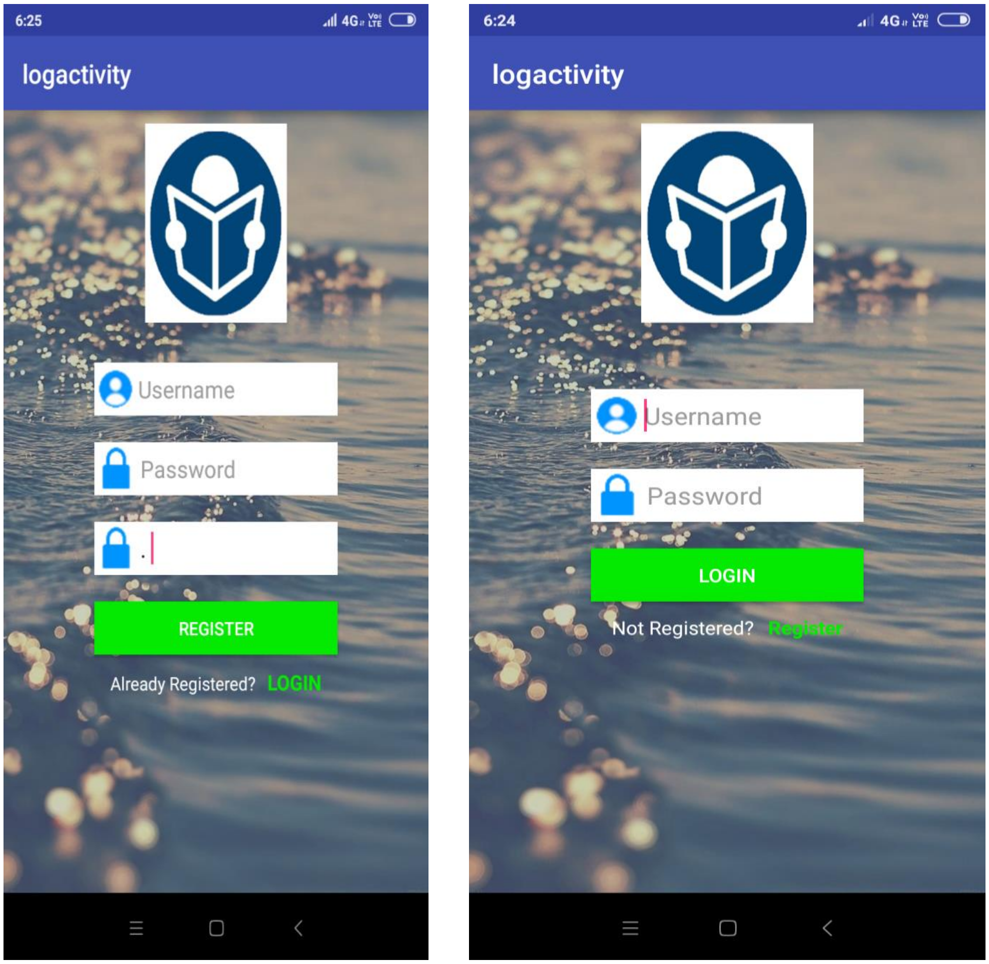
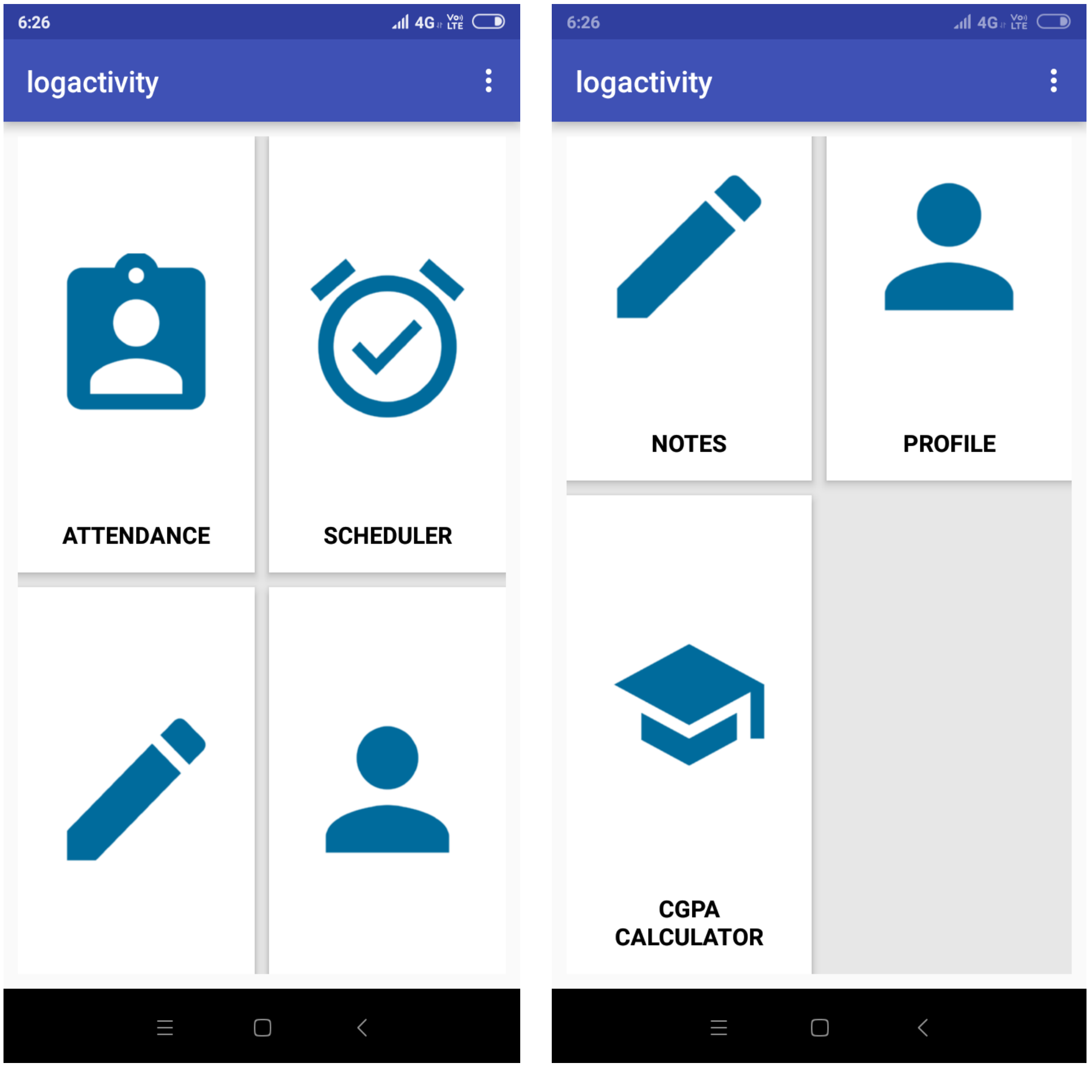


Figure 9.1:Login Screen and Registration Screen

Description: This page will show the user to register and enter into the android application

Home Page:

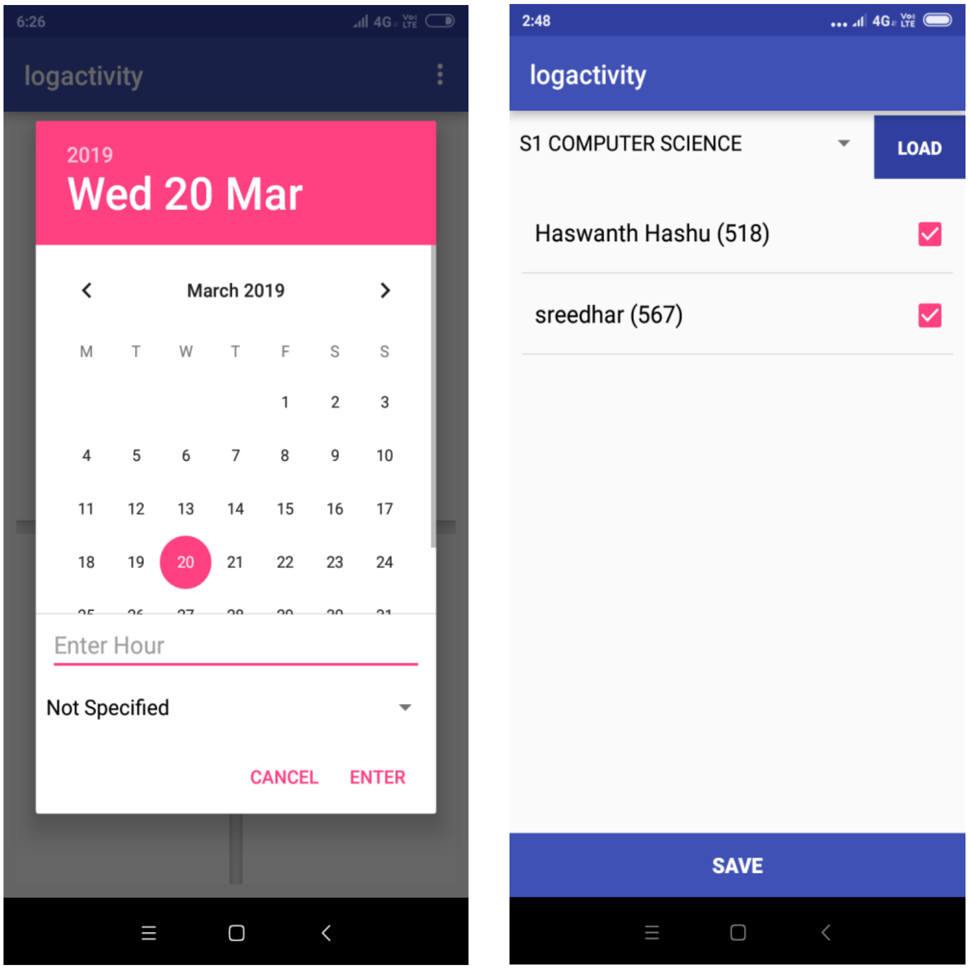


Home Screen -top view **Home Screen-bottom view**

Figure 9.2: Home Screen

Description: This page will show the user to select any option in the android application.

Attendance:



2019
Wed 20 Mar

Not Specified

S1 COMPUTER SCIENCE

Haswanth Hashu (518) ☑

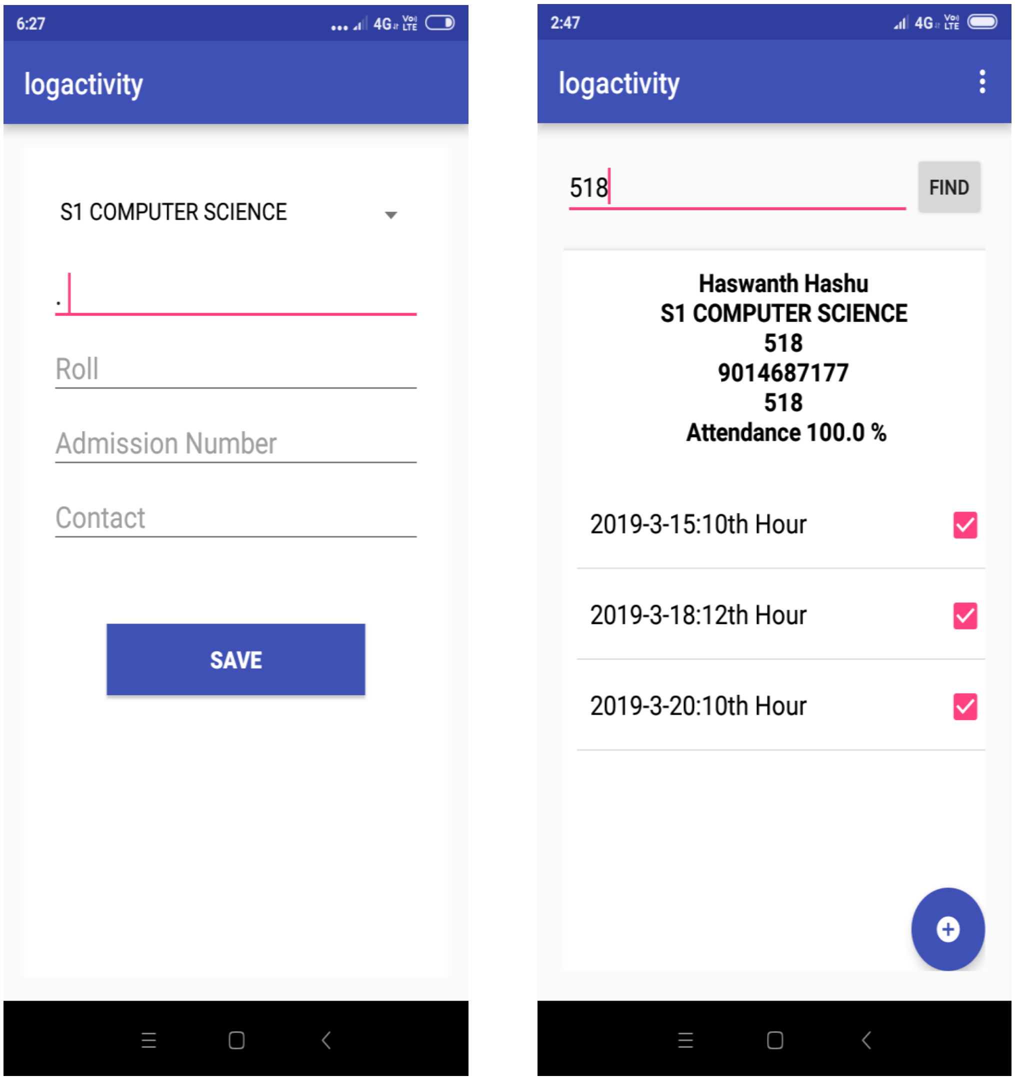
sreedhar (567) ☑

Date picker **Take attendance**

Figure 9.3: Attendance Screen

Description: This page wil show the user to select time and date for taking the attendance of saved students in the application.

Profile Activity :


6:27

logactivity

S1 COMPUTER SCIENCE

Roll

Admission Number

Contact

SAVE

2:47

logactivity

518 FIND

Haswanth Hashu
S1 COMPUTER SCIENCE
518
9014687177
518
Attendance 100.0 %

2019-3-15:10th Hour ☑

2019-3-18:12th Hour ☑

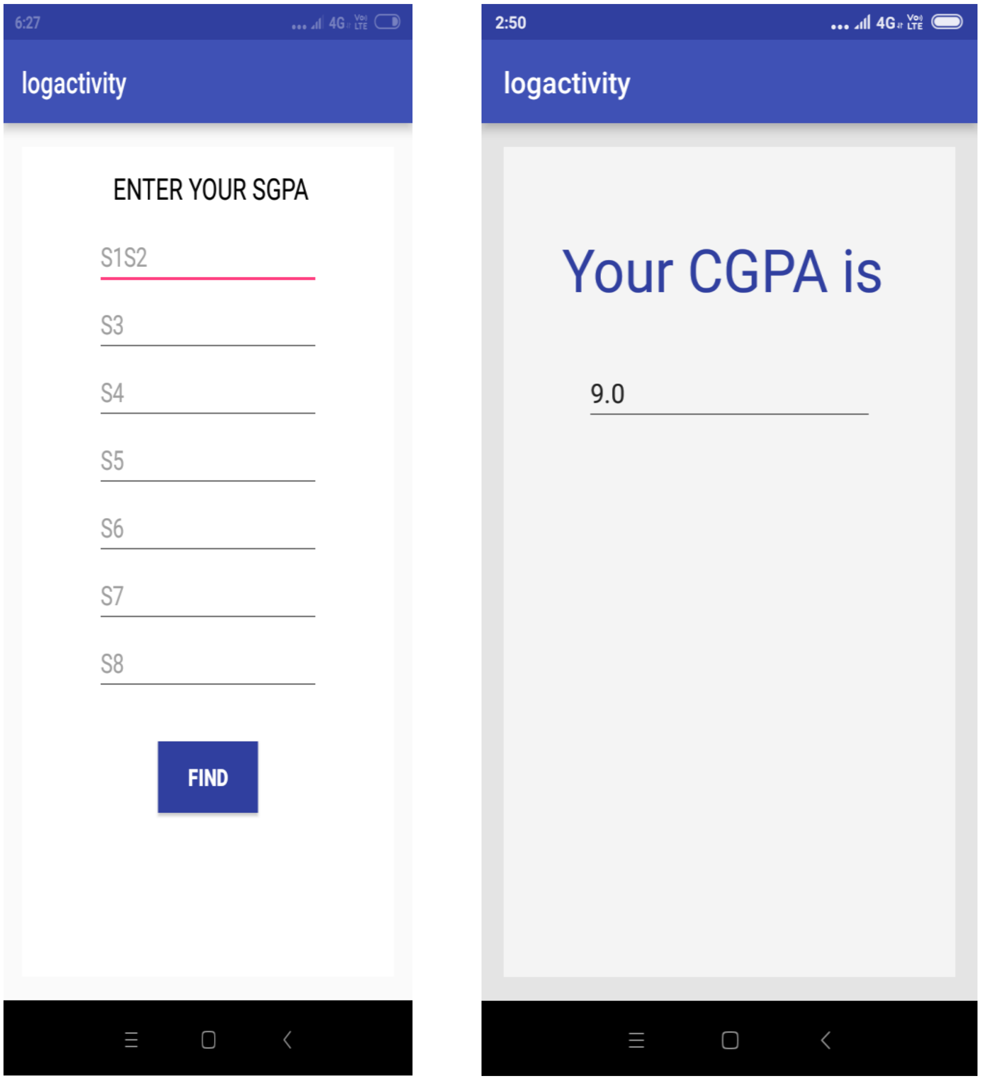
2019-3-20:10th Hour ☑

Student Details Page **View Details**

Figure 9.4: Profile Activity

Description: This page will be used to save the details of student and the user can also know the details of saved student by searching with his/her roll no.

CGPA Calculator:



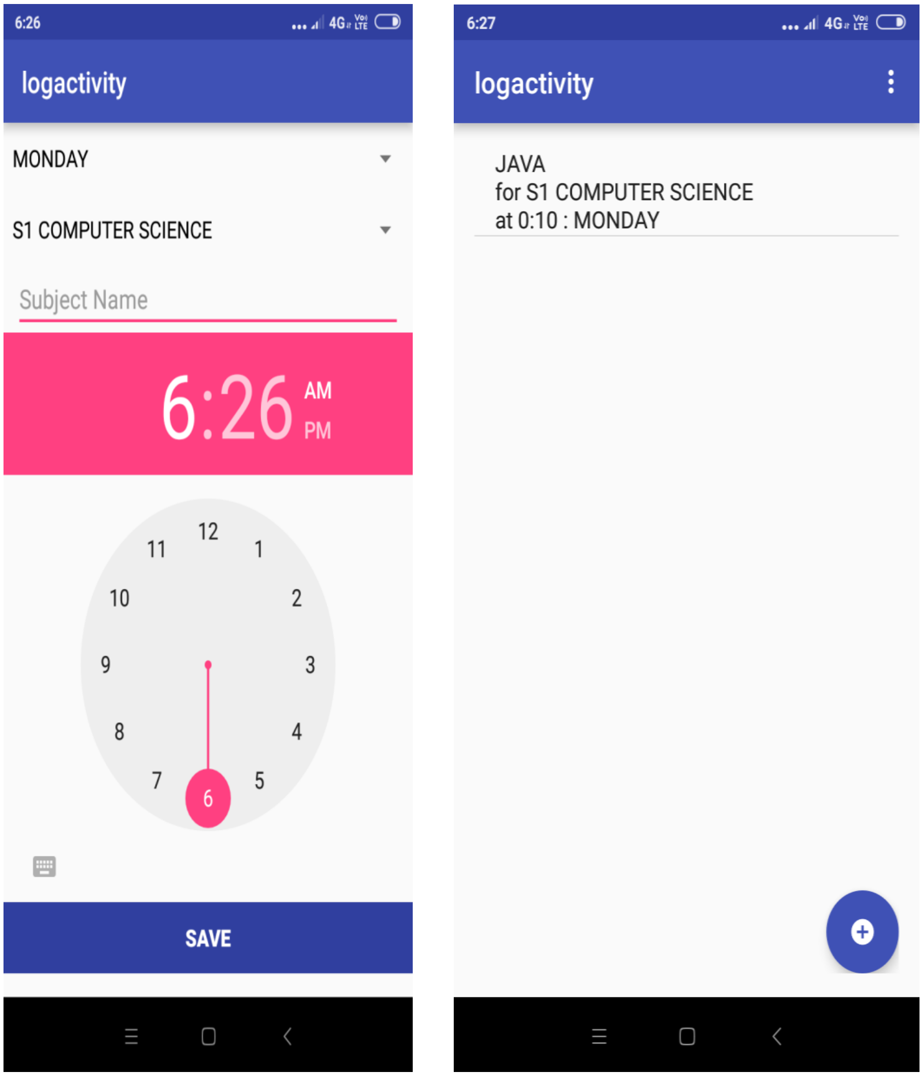
Your CGPA is

9.0

Figure 9.5: CGPA Calculator

Description: CGPA Calculator can be defined as cumulative grade points average. It can be used to calculate aggregate of a individual students grade points.

Scheduler :



MONDAY

S1 COMPUTER SCIENCE

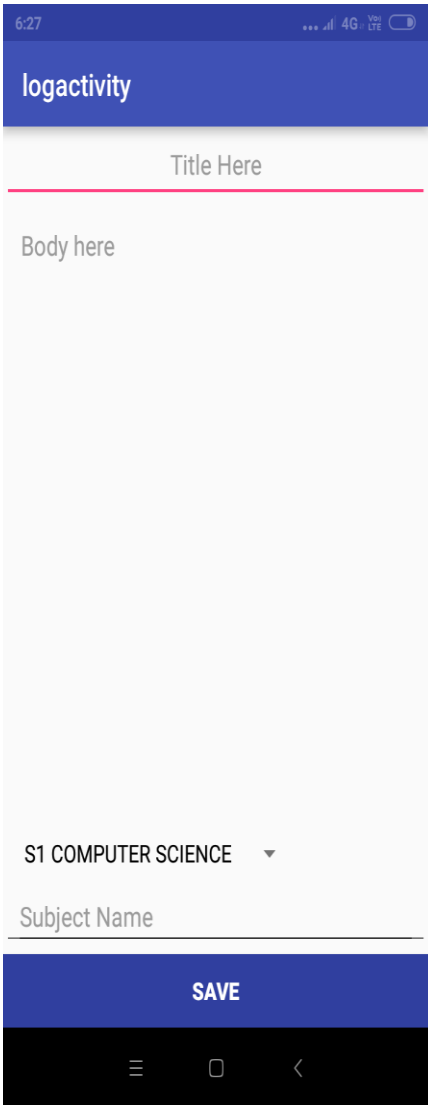
JAVA
for S1 COMPUTER SCIENCE
at 0:10 : MONDAY

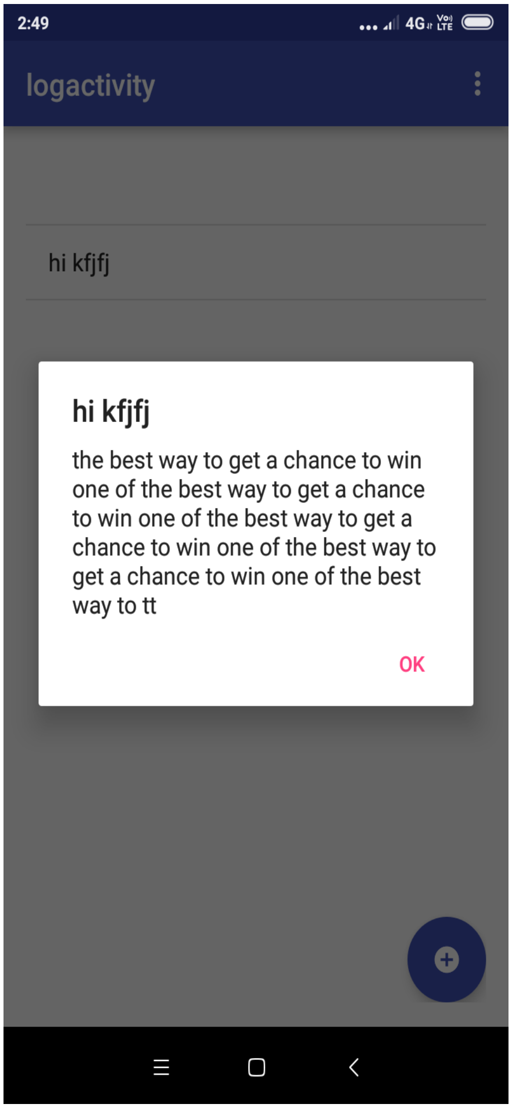
Time Picker **Saved Schedules**

Figure 9.6: Schedules

Description: This page will be used to select time and save schedules and notify the application user at the selected time

Notes :





hi kfjfj

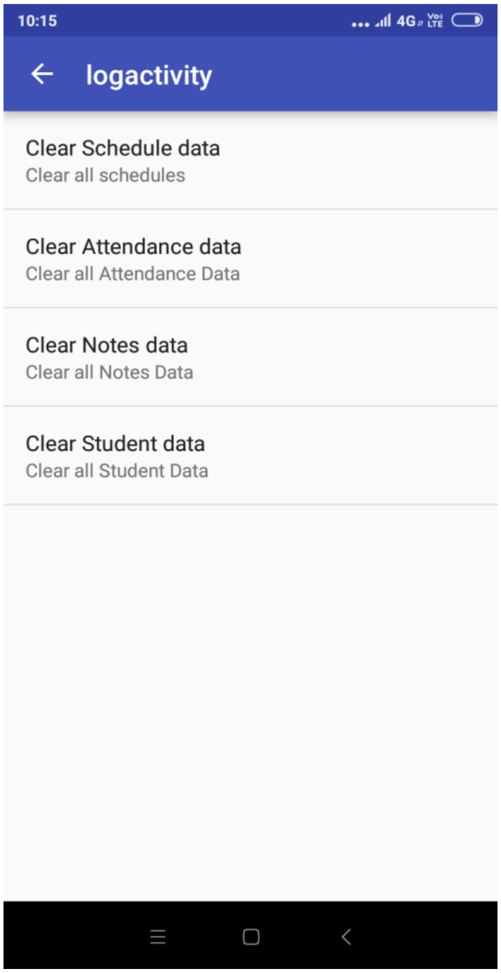
the best way to get a chance to win one of the best way to get a chance to win one of the best way to get a chance to win one of the best way to get a chance to win one of the best way to tt

Notes Page **Display of notes saved**

Figure 9.7: Notes Screen

Description: This page will be used to save notes and the user can get a pop-up message to read the saved notes.

Settings :



Clear Schedule data
Clear all schedules

Clear Attendance data
Clear all Attendance Data

Clear Notes data
Clear all Notes Data

Clear Student data
Clear all Student Data

Figure 9.8: Settings

Description : This page will show the user to clear all the schedules and attendance of any saved data.

10. Future Scope of the project

I have developed this Android Application as a project. The project is about automating the pen and paper work of a teacher. In this project it contains the features of taking attendance of students,can store complete student information and also can store notes which can be associated with a subject and class. When the teacher takes attendance for a period, the note database is searched for Available notes and if found, Notification will be given.It also have option to Schedules class periods and CGPA calculator, student information, Material Design etc..

We can further explore the android application to note and saves fees balance of a particular student.we can connect the teacher with parent by using connection to contacts directly through the android application.

11. Conclusion

Reference system through mobile devices is a very effective tool which can be used to a great extent. The system is portable and can be easily installed and used on any mobile phones supporting Android OS. The use of this system can result in a reduction of number of hours spent in feeding attendance details in the server database. It also provides an interface which is easy to understand by the users and greatly helps in adapting to the use of this system. The android application is about automating the pen and paper work of a teacher. Most of the process included in this application have been tested and concluded that the application is user friendly.

12. BIBILOGRAPHY

[1] "Google Launches Android Studio And New Features For Developer Console, Including Beta Releases And Staged Rollout". VentureBeat. December 8, 2014. Retrieved December 9, 2014.

[2] "Android Studio Release Notes". Android Developers Official Website. 17 August 2018. Retrieved 5 September 2018.

[3] Honig, Zach (May 15, 2013). "Google intros Android Studio, an IDE for building apps". Engadget. AOL. Retrieved May 16, 2013.

[4] Dobie, Alex (May 15, 2013). Android Studio unveiled at Google I/O keynote". Android Central. Mobile Nations. Retrieved May 16, 2013.

[5] Olanoff, Drew (May 15, 2013). "Google Launches Android Studio And New Features For Developer Console, Including Beta Releases And Staged Rollout". TechCrunch. AOL. Retrieved May 16, 2013.

[6] "Android Studio BETA". Google. Google. May 15, 2013. Retrieved August 15, 2014.

[7] Google Go language IDE built using the IntelliJ Platform: go-lang-plugin-org/go-lang-idea-plugin, Go Language support for IDEA based IDEs, 2019-02-23, retrieved 2019-02-23, Supported IDEs [..] Android Studio 1.2.1

[8] "android-retroflow: Backport of Java 9 (JEP 266) reactive-streams Flow and SubmissionPublisher API for Android Studio 3.0 desugar toolchain, forked from [..]". retrostreams. 2017-10-22. Retrieved 2017-10-25.

[9] "Android's Java 9, 10, 11, and 12 Support". Jake Wharton. 2018-11-27. Retrieved 2019-02-23. Hopefully by the time Java 12 is actually released D8 will have implemented desugaring for Java 11's nestmates. Otherwise the pain of being stuck on Java 10 will go up quite a bit!

[10] "Refer 'System Requirements' section". Android Studio Official Website. Retrieved October 26, 2017.

[11]."Download Android Studio". Android Developers. Retrieved June 13, 2015.

[12]. "Google Launches Android Studio And New Features For Developer Console, Including Beta Releases And Staged Rollout". VentureBeat. December 8, 2014. Retrieved December 9, 2014.

[13].Jump up to:a b "Android Studio Release Notes". Android Developers Official Website. 17 August 2018. Retrieved 5 September2018.

[14]. Honig, Zach (May 15, 2013). "Google intros Android Studio, an IDE for building apps". Engadget. AOL. Retrieved May 16, 2013.

[15].Dobie, Alex (May 15, 2013). "Android Studio unveiled at Google I/O keynote". Android Central. Mobile Nations. Retrieved May 16, 2013.

[16]. Olanoff, Drew (May 15, 2013). "Google Launches Android Studio And New Features For Developer Console, Including Beta Releases And Staged Rollout". TechCrunch. AOL. Retrieved May 16, 2013.

[17]. "Android Studio BETA". Google. Google. May 15, 2013. Retrieved August 15, 2014.

[18]. Google Go language IDE built using the IntelliJ Platform: go-lang-plugin-org/go-lang-idea-plugin, Go Language support for IDEA based IDEs, 2019-02-23, retrieved 2019-02-23, Supported IDEs [..] Android Studio 1.2.1+

[19]. "android-retroflow: Backport of Java 9 (JEP 266) reactive-streams Flow and SubmissionPublisher API for Android Studio 3.0 desugar toolchain, forked from [..]". retrostreams. 2017-10-22. Retrieved 2017-10-25.

[20]. "Android's Java 9, 10, 11, and 12 Support". Jake Wharton. 2018-11-27. Retrieved 2019-02-23. Hopefully by the time Java 12 is actually released D8 will have implemented desugaring for Java 11's nestmates. Otherwise the pain of being stuck on Java 10 will go up quite a bit!

Appendix

1. User manual

The step by step procedure for using the application as follows

Configuring the environment

To use this android application the user should contain a smart phone which any android operating system. The application is about automating the pen and paper work of a teacher.

The android application is built using the android studio software therefore the application will support any type of android operating system.

Step 1:

Open the application and the user will see a login page to enter into the application

Step 2:

The user need to register by the user name and password and needs to click to submit button. So that user can enter login details to enter into the application.

Step 3:

The user can see five differen options. The user can add the students by clicking on the profile and can save the student details in profile option.

The student details which are saved can also be edited by clicking on the edit student button

Step 4:

The user can take attendance of saved students in the application by clicking on the attendance option. It will ask the user to select date and time and will load the saved students. After that the user can take attendance of students and save it int the application.

Step 5:

The attendance details can be updated and seen in the profile option by searching the student with his admission number.

Step 6:

The application user can make schedules in the application by clicking on the schedules option. The mobile phone will notify the user at the scheduled time.

Step 7:

The user can write important notes in the application and save it by clicking on the notes button.

Step 8:

The user can calculate the grade points of a student by clicking on the CGPA Calculator.

2.ABBREVIATIONS

SDLC	**:**	Software Development Life Cycle
CGPA	**:**	Cumulative Grade points average
XML	**:**	Extensible markup Language
SDK	**:**	Software Development Kit
DEX	:	Dalvik Executable files