

CHAPTER 1

INTRODUCTION

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

This project presents the Attendance system through mobile devices, which is developed using Android application. The application is installed in every student's mobile phones and can be used to take attendance in offline mode (that is even when there is no network connectivity as the application is a stand-alone application)..

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

The scope of the project is the system on which the software is installed, i.e. the project is developed as an android mobile application, and it will work for a particular user. But later on the project can be modified to operate it online.

1.1 FEASIBILITY STUDY

1.1.1 Economically Feasible

The system being developed is economic with respect to School or College's point of view. It is cost effective in the sense that has eliminated the paper work completely.

The system is also time effective because the calculations are automated which are made at the end of the month or as per the user requirement.

The result obtained contains minimum errors and are highly accurate as the data is required.

1.1.2 Technically Feasible

The technical requirement for the system is economic and it does not use any other additional Hardware and software. The Android applications are developed using the Android software development kit (SDK). The SDK includes a comprehensive set of development tools which includes a debugger, software libraries, a handset based emulator which is based on QEMU (Quick EMUlator) and tutorials. The integrated development environment (IDE) which is officially supported for Android apps development is Eclipse which uses the Android Development Tools (ADT) plugin.

1.1.3 Behavioral Feasibility

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system.

1.2 Working of Present System

In the present system all work is done on paper. The whole session attendance is stored in register and at the end of the session the reports are generated. We are not interested in generating report in the middle of the session or as per the requirement because it takes more time in calculation. At the end of session the students who don't have 75% attendance get a notice.

1.2.1 DISADVANTAGES OF PRESENT WORKING SYSTEM

- **Not User Friendly:** The existing system is not user friendly because the retrieval of data is very slow and data is not maintained efficiently.
- **Difficulty in report generating:** We require more calculations to generate the report so it is generated at the end of the session. And the student not get a single chance to improve their attendance
- **Manual control:** All calculations to generate report is done manually so there is greater chance of errors.
- **Lots of paperwork:** Existing system requires lot of paper work. Loss of even a single register/record led to difficult situation because all the papers are needed to generate the reports.
- **Time consuming:** Every work is done manually so we cannot generate report in the middle of the session or as per the requirement because it is very time consuming.

1.2.2 CHAREACTERSTIC OF THE PROPOSED SYSTEM

- **User Friendly:-** The proposed system is user friendly because the retrieval and storing of data is fast and data is maintained efficiently. Moreover the graphical user interface is provided in the proposed system, which provides user to deal with the system very easily.

- **Reports are easily generated:** reports can be easily generated in the proposed system so user can generate the report as per the requirement (monthly) or in the middle of the session. User can give the notice to the students so he/she become regular.
- **Very less paper work:** The proposed system requires very less paper work. All the data is feted into the computer immediately and reports can be generated through computers. Moreover work become very easy because there is no need to keep data on papers.

1.3 Motivation

The attendance Management System is one of the most important system used in every organization to keep the track of attendance. The conventional methods being followed for taking attendance requires a lot of paper works. It is also very hectic and very tedious task to do and thus handling and maintaining the previous system is very tough as it is not automated. Thus, there is a need to automate the system by building the android application to reduce the manual efforts needed in storing the records and maintaining it. The students can mark themselves and the records are being stored in the SQLite databases. The attendance system through mobile devices is fully automated andis very easy to use the system and does not need any external efforts to store and manage the attendance.

CHAPTER 2

PROJECT REQUIREMENTS

2.1 Add/Edit Course

As a user professor can add course that he plans to teach for a particular semester. To do that, user will click on “Manage Courses” “Add a course” which would open up “Add New Course” form. User enters details like course id, course name and semester for which the course is enrolled. On adding a course, all the details are added to MASTER_COURSE_TABLE.

User can also edit the details of already added course. To do that, clicking on “Manage Courses” will list all the courses that have been added. Click on pencil icon of the course that user wants to edit. It will open up edit form where user can update the name/description and save it, which will do an update to MASTER_COURSE_TABLE.

2.2 Take Attendance

The attendance will be taken by manual inputs.

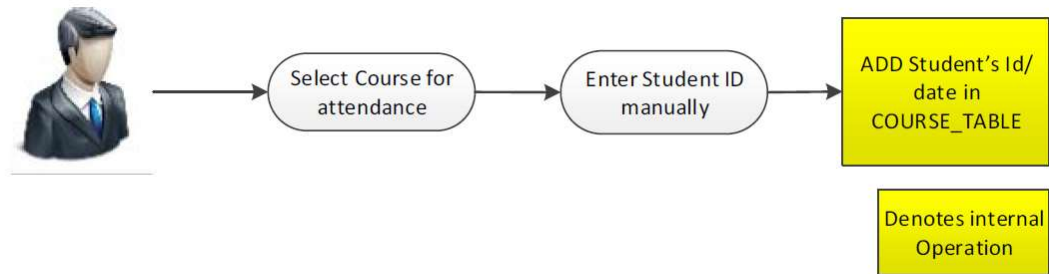


Fig 1: Mark Attendance Manually

2.3 View Attendance

Viewing attendance basically allows user to look at attendance record of student – i.e. of total number of days classes were conducted, how many days was a student absent; and if absent, dates on which he was marked absent.

CHAPTER 3

ANDROID DEVELOPMENT BASICS

Android is a Linux based Operating System that was developed by Android Inc., which was bought by Google later on. Android OS is designed mainly for smart phones and tablets that have touch screens/motion detectors. Despite being primarily designed for touchscreen input, it also has been used in game consoles, digital cameras, and other electronics. To give competition to Apple's iOS, Google made Android as open source and releases code under Apache License. In last few years, Android has become the most widely used operating system in the market and having open source has just increased its popularity many folds.

Android code is written primarily in Java programming language, and is compiled with the help of Android SDK tools. On compiling, it generates an Android package (commonly known as .apk) file which is used to install the application on android device.

App Components:

To aid in android development, there are mainly four types of components. Each component serves a distinct purpose and allows system to interact with your application in different ways. Broadly speaking, there are four types of components [1].

- Activity
- Services
- Content provider
- Broadcast receivers

3.1 Types of Application Components

Android helps to create applications using these components. This will help to understand how one can build the components which will be the building blocks of the application.

3.1.1 Activities

An activity represents a single screen with a user interface. For example, in this Attendance Tracker application, activities include adding a course, scanning student to register presence or viewing student's attendance record. "Main" activity is the activity that is launched when any application starts. Different activities can start from one

activity, keeping the previous activity in stack although the activity is stopped. Activities follow LIFO (Last In First Out) mechanism.

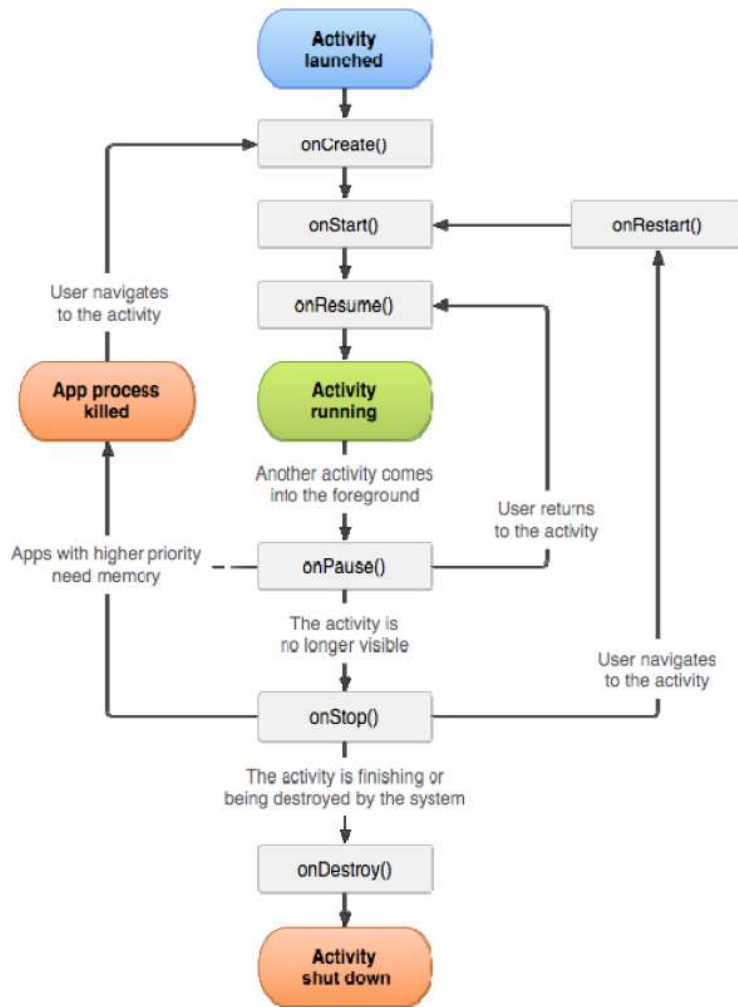


Fig 2: Android Lifecycle for Activity

3.1.2 Services

Services are the component in Android Operating System which runs in background. Hence, all the background tasks are done using the services component. This component is useful especially when a task is to be performed without impacting the user of its operation. This component does not provide any User Interface. For example, downloading a file from internet or loading image. All the services that are created in the application have to inherit the Service class provided in Android Operating System.

Service component has two forms :

- (1) **Started:** When an activity starts `startService()`, then a service is "started". A service can run in the background indefinitely once started, even if the component that started it is destroyed. For example, it might download a image over the network. When the download is completed, the service should stop itself.

- (2) **Bound:** When an application component calls `bindService()`, a service is "bound". A bound service provides a client-server interface that allows components to interact with the service, send requests and get results.

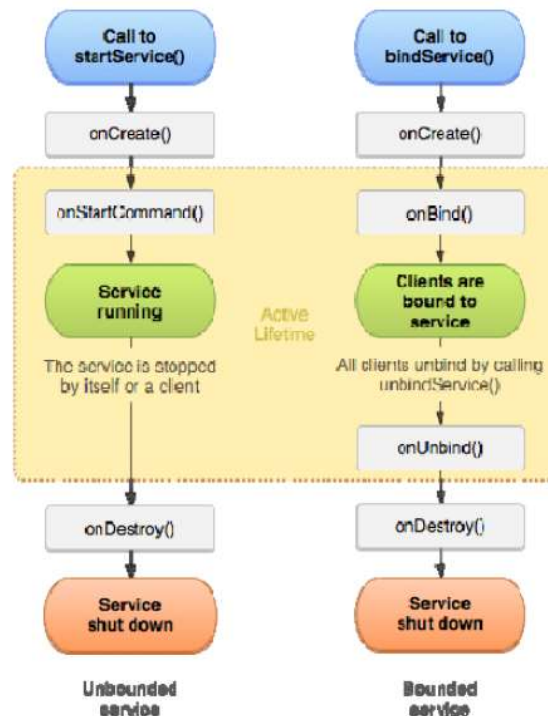


Fig 3: Android Lifecycle for Services

3.1.3 Content Providers

This component manages the shared set of application data i.e. it provides a mechanism through which the data stored by one android application can be made accessible to other application. The application can store the data anywhere from SQLite Database to web to any persistent storage location. Through the content providers in the android operating system, the data stored can be modified, queried or even store the newly entered data in the application. This component is useful to write or read the data where the user cannot access (the protected storage on the data).

3.1.4 Broadcast Receivers

A broadcast receiver allows registering for system or application events. When any registered event occurs, receivers for an event will be notified by the Android runtime. System broadcast include screen turning off, the battery is low, or a picture was captured. Applications broadcast would include letting other applications know that some data has been downloaded to the device and is available for them to use.

3.2 Android Application Files

Files can be broadly divided into three categories [1].

- ☐ Java file
- ☐ Layout file
- ☐ Manifest file

3.2.1 Java Files

These files are the files where all the processing of the events happens, and allows user to interact with the system. These files are the heart of the android application. This is the place where onCreate(), onStart(), onPause(), onStop() etc. methods are defined. This part is responsible for getting the user inputs and processes the activities accordingly. Through these files, the layouts can be added dynamically, the user entered values in the text boxes or other input can be obtained and stored.

3.2.2 Layout Files

A layout defines the visual structure for a user interface, such as the UI for any activity. These files are responsible for defining the user Input. The Android framework gives you the flexibility to use either or both of these methods for declaring and managing your application's UI. The user has the luxury of seeing how the designed layout will look like in the Graphical Mode and they have the option of selecting the device that they desire to view the layout. There are multiple interfaces that are possible in android. Few of them are enlisted below –

a. Linear Layout – Using this layout option, the developer can align all the components on the screen vertically or horizontally. This layout can be only in one direction. This layout is usually used when the position of the components are static and will not change during the course of the application,

b. Relative Layout – Through this layout, the position of the components can be described in relation to other components. This is widely used as the position of the components may vary based on the screen resolution on different devices. This is recommended for the developers to ideally use for a large pool of devices due to varying screen size.

c. List View – When there are multiple items to be displayed on the screen and they may not fit the size of the screen, then the List View is used. The items in the list are filled in the adapter from various sources like Database query, array etc.

d. Scroll View – When the components are not fitting into physical display, then the scroll view layout is used to insert more components in the screen. A scroll view be can

scrolled vertically only. If the developer wants to use the horizontal scroll, then there is another layout for android operating system – `HorizontalScrollView`. It is not recommended to use the `ListView` with the scroll view as the `List View` takes care of scrolling on its own.

3.2.3 Manifest File

It is the main part of the android application. This file contains all the information about the application – what android operating system components are present in the application, what permissions are required by the application etc. For using any component, it has to be present in the manifest file otherwise that component or functionality will not be executed when the application is installed in the device. The manifest file defines what version of the Android Operating System, the application is compatible with. The manifest file also defines what devices that the application can support. It can be defined as:

```
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="18" />
```

The developer can define all the activities, their theme etc. in the manifest files as –

```
<supports-screens
    android:anyDensity="false"
    android:largeScreens="true"
    android:normalScreens="true"
    android:resizeable="true"
    android:smallScreens="true"
    android:xlargeScreens="false" />
<activity
    android:name=".CaptureActivity"
    android:configChanges="orientation|keyboardHidden|screenSize"
    android:screenOrientation="landscape"
    android:theme="@android:style/Theme.NoTitleBar.Fullscreen"
    android:windowSoftInputMode="stateAlwaysHidden" >
</activity>
```

The permission that the application requires for the proper execution of the tasks/functionalities that are used in the application is declared in the manifest file. The sample of that can be found as:

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

The broadcast receivers also have to be declared in the manifest file for the operating system recognition. They are declared in the manifest file as:

```
<receiver
android:name="android.com.example.attendancetracker"
android:label="@string/app_name" >
<intent-filter>
<action android:name="android.appwidget.action.APPWIDGET_UPDATE"
/>
</intent-filter>
<meta-data
android:name="android.appwidget.provider"
android:resource="@xml/attendance_widget" />
</receiver>
```

3.3 Storage Options in Android Operating System

While working with Android Operating System, there are two options where the user data can be stored. The options are [1] –

- Shared Preferences
- SQLite Database
- Internal Storage
- External Storage

The user has the option of using any of the options depending on the need of the application as well as the criticality of the data.

3.3.1 Shared Preferences

This mechanism stores and retrieves the data as key value pairs of the primitive data types like String, Integer (int) or Booleans. The data stored by this mechanism will be persisted

across multiple sessions of the application. The data will remain as it is even if the application is killed several times.

3.3.2 SQLite Database

The operating system allows the developer to use the local databases. The databases created using the SQLite databases will be accessible to any class present in the application. These databases have to be accessed by their names. The data will be available across the application but not outside the application. For using this facility, the developer has to create a subclass of SQLiteOpenHelper and override the onCreate method.

3.3.3 Internal Storage

The operating system allows the data to be directly stored on the internal storage of the device. The data that is stored in the file is private to the application and the other applications that are present in the device cannot access the data stored. The user (owner) of the device cannot also access the data of the application. When the application is uninstalled, the files will be removed automatically from the internal storage of the device.

3.3.4 External Storage

External storage can be any removable media like USB, SD Card etc. The data stored in this storage are public and can be accessed by anyone in the application. The user/application can read the data and have the option of modifying the data that is present in the file.

3.4 Tools Required

Android applications can be developed across multiple IDE's and there are several plugins available for making the current IDE capable to write the android programs. Out of several IDE's, the popular IDE's are enlisted under –

- **Eclipse** – Any version of Eclipse can be made capable of writing the Android Applications after installing the Android Developer Tools plugin from Eclipse 22 Market. Once, this is installed, the user can go the File Menu and select Android Application as the new project.
- **Microsoft Visual Studio** (For mono-droid programs) – This is currently not very popular for cross platform IDE. This IDE is currently capable of writing android programs that are based on Mono-Droid.

- **Android Development Tool** – This IDE is provided by Google Inc. that has been configured by the developers at Google to write the Android applications in Eclipse. The primary IDE here is Eclipse.
- **Android Studio** – This is the IDE that is developed by Google Inc. Currently, it is still in the development mode and it was announced in April 2013. It is currently available for free. It is based on IntelliJ software. It is available for download across various operating systems like Windows, Mac, and Linux.

CHAPTER 4

SPECIFICATION REQUIREMENTS

4.1 Mobile Requirements

- Processor:Processor Z2520 1.2GHz,or faster processor
- RAM :3GB (Recommended) or more
- Mobile :Android Mobile
- Version :Android Jelly Bean or above

4.2 System Requirements

- IDE:Android Studio
- OS :Windows7 or above/Mac OS
- Hard disk:2 GB & 4 GB (500 MB for IDE + 1.5 GB for SDK and emulator)
- DB Browser for SQLite Database

CHAPTER 5

Design

The design phase emphasizes on the transformation of customer requirements as defined in the SRS document, into a form that is suitable for coding.

The design phase can be broadly classified in two levels.

- Preliminary or high level design

- Detailed design

The preliminary design can be further divided into two sub categories

- Function Oriented Software Design

- Object Oriented Software Design

5.1 Function Oriented Software Design

This design model can be represented by drawing the DFDs (Data Flow Diagrams) for the given SRS document.

A data flow diagram is a graphical representation of the data flow through an information system which is used to model the process aspects of the system. DFD is the preliminary step used to create an overview of the system. DFD is used for structured design.

5.1.1 Context diagram

A context diagram is a structured graphical tool used for identifying the functional areas and the processes which are performed within and between the system and outside the system. Context diagram supports a data-oriented approach for designing system. It helps in investigating the output and the process requirement of the system. It helps in defining the boundaries of the proposed system. The symbols used in the context diagrams are for external entities, data storage and data flows and process.

Here in the following context diagram4.1, we represent our attendance system which has one external entity i.e. user of the system namely Faculty and the data flowing in and out of the system is the attendance details.

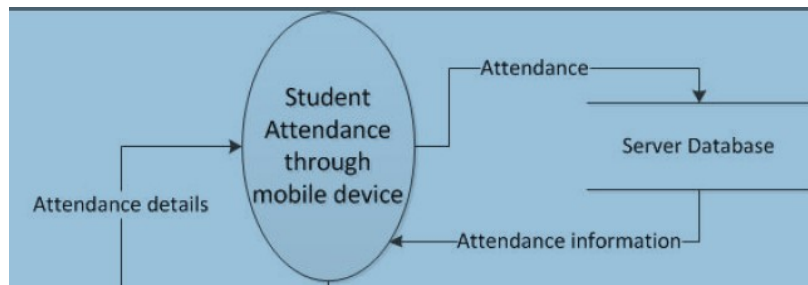


Fig 4: Context diagram

5.1.2 Level 1 diagram

The context-level DFD is then exploded to produce a Level 1 DFD which models the details of the system. The Level 1 DFD shows how the system is divided into sub systems (processes), and how each processes deals with one or more of the data flows to or from an external entity, and how the processes together provide all of the functionality of the system. The level 1 DFD also identifies the internal data stores which must be there for the system to do its job, and shows the data flow between the various parts of the system.

In the below Level 1 DFD 4.2, the attendance system has been decomposed further into 5 processes which are namely Login, View courses taken, View enrolled student list, View attendance, Upload attendance.

5.1.3 Level 2 diagram

The Level 2 DFD is the further decomposition of Level 1 processes into sub-processes (sub-systems) which give detailed description of the data flow in each processes. Here we have decomposed the process View enrolled student lists (Fig 4.2) in the sub-process Take attendance. The faculty selects the course ID and get the details of the enrolled student and then he can take attendance for that particular course . The data storage used in this level is the Attendance list.

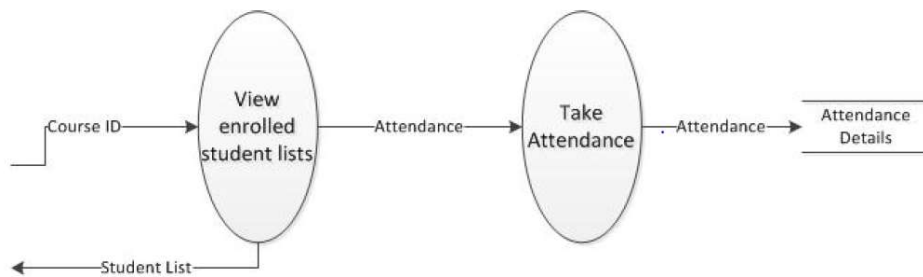


Fig 5: Level 2 Diagram

5.2 ER Diagram

The entity-relationship diagram is a data modeling technique that graphically represents an information systems entities and the relationships between those entities. An ER diagram is a conceptual and representational model of data which is used to represent the system framework infrastructure.

The ER diagram contains following elements:

- Entities
- Relationships
- Attributes

In designing the ER diagram, we identify and define all the entities, determine the interactions between the entities and determine the cardinality of the relationship. The ER diagram for the attendance system represents all the entities namely User, Course, Enrolled Student and Attendance and the relationship between these entities used in the system. The User entity contains the attributes Uid, Name, Tname and Pwd where the Uid is its primary key. The Course entity contains the attributes Subcode, Subname and Tname where Subcode is its primary key.

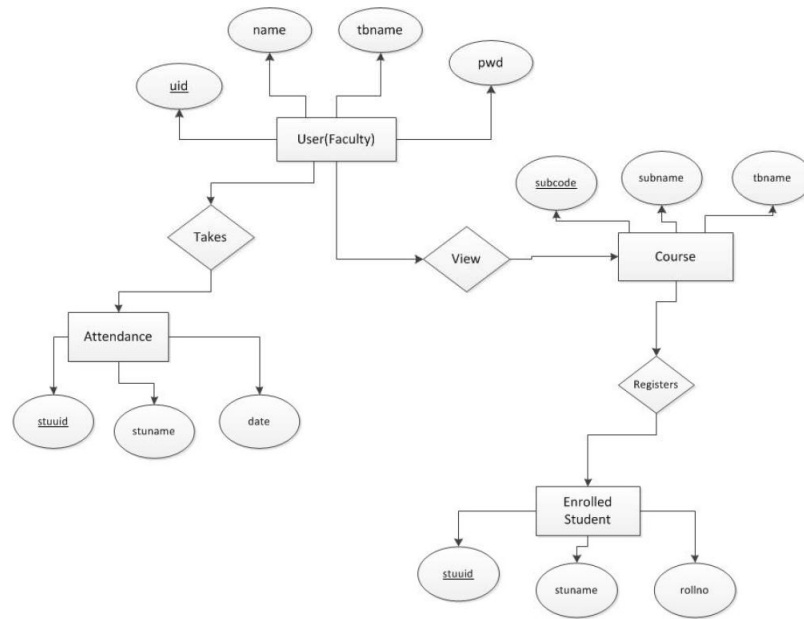


Fig 6:E-R Diagram

CHAPTER 6

IMPLEMENTATION

6.1. Splash Activity

This is the first screen of the application. This screen will be visible for 3-5 seconds as soon as the user clicks on the App icon in the mobile. It consists of the App icon with the Title which can be called as splash screen. This is build using the relative layout from the XML file of this activity. Splash screen contains an image element and it will be displayed for few seconds.

Splash Screen is used to display the logo of the app and is generally used for introduction of the application.

6.2 Menu/Homepage Activity

The second screen will be the Home page/Dashboard of the App. Dashboard consists of four image Buttons and one Floating Button each button takes to different activity. Dashboard contains different sections in a Linear layout and Text Fields and each section has different functionality. The four buttons are Calendar, Subjects, Timetable and Predictor. The Floating activity lets the user add the different subjects. The Text Field shows the current system date and time and also display the notification to the user about the upcoming events on that specific date.

It has the navigation drawer which is built above the dashboard activity to choose three options i.e. one option is of rating the app according to the users liking and the second option is settings activity of and app and the third activity is of sharing the app to the multiple users with the help of different platforms. The button on the toolbar lets user jump to setting activity, If back button is pressed in home screen, it takes the user outside of the application.

6.3 Subjects Activity

This activity consists a list view of all the subjects. On clicking the particular subjects, the user is directed to a new screen/activity which shows the user his overall attendance percentage in the form of pie graph and also the necessary information's. The onclick function is being implemented by on click listener.

This activity also consists of a little help button at the top of the toolbar to which lets user know how to interact with each item.

6.4 Navigation Drawer Activity

The Navigation Drawer activity is present on the top of Dashboard activity.

This activity is opened when the user clicks on the Hamburger icon on the top of the toolbar or sliding across the dashboard items.

The Navigation Drawer contains a section called header which displays the image contained in the ImageView attribute. The header section also shows the name of the app. There are total three items in the Navigation Drawer: “Rate Us!”, “Settings” and “Share”.

Each of these items perform different actions. The first one asks for the feedback from the user. The second one takes user to the new activity and the third one lets user share the app across different users through different sharing platforms.

6.5 Settings Activity

The settings activity can be launched either from the Navigation Drawer activity or by clicking the ActionBar (three-dots) options present in the toolbar in the Dashboard activity.

The activity contains the two toggle buttons and the sliding bar. The first toggle button lets the user set a reminder for the particular class/event. The second toggle button lets the user set a class notification.

There are two text fields with SET TIME which is enabled on clicking the corresponding buttons and lets user select the time to remind or notify him/her. The sliding bar present in the activity allows user to set the minimum attendance criteria for a subject.

CHAPTER 7

CONCLUSION

Registering attendance during a class can be very time consuming process, which requires manual steps and is error prone. This application solves all the downfalls of registering attendance manually. The attendance 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 the attendance details in the SQLite database. Additionally, student has the luxury to look at attendance record of students whenever he wants. Just a few clicks away!, It also provides an interface which is easy to understand by the users and greatly helps in adapting to the use of this system.

7.1 Future Scope

The system can be further enhanced and several other functionalities can be added. The system can be made login dependent. The present system can be enhanced by adding faculty characteristics also.

We can enhance the system by implementing online mode login feature using Firebase. The feature to update the attendance at a later stage can also be implemented. The system can also, be enhanced by using voice recognition feature of the Android.

CHAPTER 8

BIBLIOGRAPHY

- [1] Android Developer guide
<http://developer.android.com/>
- [2] Research on Tesseract, most accurate OCR engine available.
<https://code.google.com/p/tesseract-ocr/>
- [3] An overview of the Tesseract OCR engine by Ray Smith
http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/pubs/archive/33418.pdf
- [4] “Android Cookbook” by Ian F. Darwin (April 2012), *O’Reilly Media, Inc.*
<http://androidcookbook.com/home.seam>
- [5] Android Developers Blog
<http://android-developers.blogspot.com/>
- [6] Stack Overflow
<http://stackoverflow.com/>