



Awais Abdullah

01-134141-020

Muhammad Umer

01-134141-090

Smart Attendance System

Bachelor of Science in Computer Science

Supervisor: Dr. Faisal Bashir

Department of Computer Sciences

Bahria University, Islamabad

Nov,2017

Certificate

We accept the work contained in the report titled "Smart Attendance System", written by Awais Abdullah AND Muhammad Umer as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by . . .

Supervisor: Dr. Faisal Bashir

Internal Examiner:

External Examiner:

Project Coordinator: Dr. Sumaira Kausar

Head of the Department: Dr. Faisal Bashir

Nov, 2017

Abstract

In the last decade we have witnessed an enormous increase in the end user acceptance of mobile communications. The appearance of mobile platforms based on the open source software has rapidly increased the interest into mobile applications development. Development of software should be more real, user friendly, compatible with system and cost effective. The use of software by user must be simple and should not require much training to use software. Currently student attendance is marked verbally by the teacher who calls out roll numbers or names of the students, that is time-consuming tedious work. So we have used the android platform for our project which is open source, scales to every device and it offers a unified approach to application development. We had an innate interest in developing class room applications from the start of our degree. This is an attempt to make an application that will be beneficial to both teachers and students. Our application facilitates this by letting the students mark their own attendance using an Android application. The teacher will then only verify them and then voila attendance is marked. Teachers can also view their student's attendance record on daily basis. They can view their courses and see the attendance from that course. This application is designed to be easy to use by both teachers and students.

Acknowledgments

First of all we are thankful to Al-Mighty Allah, the Most Beneficent and the Most Merciful, Who gave us knowledge, determination and strength to achieve our goal. We have done our best but it would not have been possible without the kind support and help of our families and teachers. They supported and encouraged us throughout our studies and project. We are highly thankful to our respected supervisor Dr. Faisal Bashir who helped and guided us in completing this project. From start he was source of support and guidance on every problem.

Table of Contents

Chapter 01	11
Introduction	11
1.1 Overview	11
1.2 Objective	11
1.3 Problem Description	12
1.4 Scope	12
Chapter 02	13
Literature Review	13
2.1 Introduction	13
2.2 Android Base Smart Learning and Attendance System	15
2.3 Attendance Register	15
2.4 Attendance Tracker	15
2.5 Smart Attendance	16
2.6 Easy Attendance School College	16
2.7 Mobile Based Attendance System	16
Chapter 3	18
Requirements Specification	18
3.1 Existing System	18
3.2 Proposed System	19
3.3 Requirements Specification	19
3.3.1 Functional Requirements	19
3.3.2 Non-Functional Requirements	20
3.4 Use case Specification	21
3.4.1 Use Case 1 Modules Choice	21
3.4.2 Use Case 2 Sign In/Sign Up	22
3.4.3 Use Case 3 User fills Credentials	23
3.4.4 Use Case 4 Sign In	24
3.4.5 Use Case 5 Selection of Courses	25
3.4.6 Use Case 6 Teacher Specifies time for attendance	26
3.4.7 Use Case 7 List of present students	27
3.4.8 Use Case 8 Teacher edits attendance	29
3.4.9 Use Case 9 Student chooses semester and courses	30

3.4.10 Use Case 10 Student Marks Attendance	31
3.4.11 Use Case 11 User Logs Out.....	32
Chapter: 04	33
Design	33
4.1 Introduction.....	34
4.2 System Architecture	34
4.2.1 Context Diagram	34
4.2 Design Constraints.....	35
4.3 Activity Diagram	36
4.5 Sequence Diagram.....	37
Chapter: 05	39
Implementation	39
5.1 Introduction.....	39
5.1.1 Tools and Technologies.....	39
5.1.1.1 Android Studio	39
5.1.1.1.1 Interface	40
5.1.1.2 Java	42
5.1.1.3 Netbeans Eclipse	42
5.2 Software Development Model	42
5.2.1 Star Model	42
5.3 Modules.....	43
5.4 Getting SSID of Router	45
5.5 GUI	46
Chapter: 06	53
System Testing and Evaluation.....	53
6.1 Introduction.....	53
6.2 Testing Methods.....	53
6.2.1 Unit Testing	53
6.2.2 Integration Testing	53
6.2.3 System Testing	53
Test Case 1: Installing Application	54
Test Case 2: Running the application	54
Test Case 3: Running application on Incompatible OS version	55

Test Case 4: Menu Navigation is Hierarchical	56
Test Case 5: Sign In Sends Verification Code via SMS	56
Test Case 6: Verifies correct authentication Code	57
Test Case 7: Error at Incorrect Verification Code	58
Test Case 8: User Signs in using Correct Credentials.....	59
Test Case 9: User is not allowed to sign in using Incorrect Credentials	59
Chapter: 07	61
Conclusion.....	61
7.1 Introduction.....	61
7.2 Problem Description.....	61
7.3 Solution advantages.....	62
7.4 Limitations	62
7.5 Achievements/Learning	62
7.6 Future Plan	63
References	63

List of Figures

Figure 3.1 Menu Choice	22
Figure 3.2 Sign In/Sign Up	23
Figure 3.3 User fills credentials for sign up	24
Figure 3.4 User sign in with correct credentials	25
Figure 3.5 User selects Semester and Courses	26
Figure 3.6 Teacher Specify Time Frame for attendance	27
Figure 3.7 List of Present Students	28
Figure 3.8 Teacher Edit attendance	29
Figure 3.9 Student selects semester and courses.....	30
Figure 3.10 Student marks attendance.....	32
Figure 3.11 User Log Out	33
Figure 4.1 Context Diagram	35
Figure 4.2 Activity Diagram.....	1
Figure 5.1 Android Studio Project Menu	40
Figure 5.3 Life Cycle of Star Model	43
Figure 5.4 Working of application Modules.....	44
Figure 5.4.1 SSID of Router	45
Figure 5.5.1 Main Screen	46
Figure 5.5.2 Teacher’s Portal	47
Figure 5.5.3 Timer Running.....	48
Figure 5.5.4 Attendance View.....	49
Figure 5.5.5 Student’s Portal.....	50
Figure 5.5.6 Marking of attendance.....	51
Figure 5.5.7 Attendance report	52

List of Tables

Table 3.1 Functional Requirements	20
Table 3.2 Non-Functional Requirements	20
Table 3.3 Use Case 1 Modules Choice	21
Table 3.4 Use Case 2 Sign In/Sign Up	22
Table 3.5 Use Case 3 User fills Credentials	23
Table 3.6 Use case 4 Sign In	24
Table 3.7 Use Case 5 Teacher selects courses	25
Table 3.8 Use case 6 Teacher Specifies time	27
Table 3.9 Use Case 7 List of Present Students	28
Table 3.10 Use Case 8 Teacher edits attendance	29
Table 3.11 Use Case 9 Student chooses Semester and Courses.....	30
Table 3.12 Use Case 10 Student Marks Attendance.....	31
Table 3.13 Use Case 11 User logs out of the system	32
Table 6.1 Installing Application	54
Table 6.2 Running the application	54
Table 6.3 Running Application on Incompatible OS version.....	55
Table 6.4 Menu Navigation.....	56
Table 6.5 Verification Code via SMS	57
Table 6.6 Verifies Correct Authentication Code	57
Table 6.7 Error at Incorrect Verification Code	58
Table 6.8 User Sign In Using Correct Credentials.....	59
Table 6.9 User with Incorrect Credentials	59

Acronyms and Abbreviations

SAS	Smart Attendance System
EASC	Easy Attendance School System
FR	Functional Requirements
NFR	Non Functional Requirements

Chapter 01

Introduction

1.1 Overview

Technology advancement is rapidly growing day by day. Manual tasks are now done automatically in almost every field. People want to go with time, and time saving is now main factor of success. Keeping these all things in mind we are going to make a Smart Attendance Systems. Currently attendance systems are implemented manually, where teacher manually marked attendance and saved the record. Teachers have to call the names of all the students one by one. This is difficult to maintain and time consuming. SAS facilitate Teacher and Student to mark and view attendance on daily bases by few clicks. Teacher will login and select desired course and allow students for few minutes to mark their presence in the class. Student can login select the course and mark their presence in the allowed time and after that time is over students will not be able to mark their attendance, attendance will be disabled. The main point here is that teacher and student must be connected to same router only then students will be able to mark their attendance.

1.2 Objective

The main objective of making Smart attendance system is to provide a platform for teachers and students to mark their attendance easily. With some clicks teacher can allow the attendance and students can mark their attendance. We want students to mark their own attendance by using their android based cellphone and teacher don't have to call the names of students to mark attendance. We achieved our objective by making SAS application for android platform. We used java programming language. We have used the firebase real time database where data is processed on real time. SAS application is easy to use for both teachers and students as well.

Currently there are few applications available for attendance marking and management. After a quick search one can see that there is still a lot of improved applications could be

built for attendance marking and other functionalities that could be given to teachers and students.

1.3 Problem Description

Attendance system is manually implemented in almost everywhere. Where teachers and students have to mark and view attendance manually by using some web application. In these attendance systems teachers have to call the names of every student one by one to mark their presence in the class, which is time consuming for teachers. With new mobile technologies this problem could be solved. Now a days there are so many platforms like android. We can built applications by using these tools and platforms and make life easier for teachers and students. Teachers and students need an easier way to mark the attendance where they can fulfill their requirements by some clicks. Our application provides that to both teacher and as well as students. Teacher and student just have to install the application on their android cellphone and sign up by giving some credentials. After signing up

1.4 Scope

Smart attendance system is an Android application, which facilitates the teachers by marking the attendance of students on daily bases. Teacher sign Up and sign In into the application with few credentials and select the subject of which he wants to mark the attendance. Teacher will allow the specific time like 1 minute it depends on teacher, for attendance and students have to mark their attendance in that specified time. Student must connect to the same WIFI to which teacher is connected in order to mark their attendance. We included this functionality in our project in order to prevent the proxy. Students have to be in the class room and have to connect to the same WIFI then they will be able to mark their attendance.

Chapter 02

Literature Review

2.1 Introduction

This section reviews the research works carried out by different researchers that are related to the proposed work. In general, the mobile application is developed using any one of the languages such as Java using Software Development Kit (SDK). The data used for the application or processed by the application are stored in the data bases. The following mobile application developers succeed in developing the student attendance management system with the structured query language (SQL) data bases.

V. Somasundaram et al. presented a mobile-based attendance system using visual basic .Net (VB.NET) and SQL server. This system is used to store, organize, find and manage the information of the students and helps to generate the reports of the student information.

K. Akhila et al. proposed an android-based mobile application for student attendance tracking system. It offers reliability, time saving, and it is easy to control and to take the attendance using android mobile phones. It can reduce the efforts of the staff members towards attendance maintenance. It is an efficient and user friendly android mobile application for attendance monitoring.

Rakhi Joshi et al. developed an android-based attendance management with smart learning system. The web-based mobile application is developed with a SQL server. This system is used to mark attendance through smart phone and gives a prior intimation to student as soon as their attendance goes below the specified level through SMS. Moreover, Amita Dhale et al. presented a survey on "smart connect", android and web based application for college management system. It is developed using SQL server. It is mainly used to store the details required for the institutions.

The mobile operating system (MOS) place a key role in the development of mobile application since the application for one MOS is not compatible with other MOS. Therefore, before developing the mobile application for a particular application the MOS must be

considered and the application must be developed for the same. Thus, the student attendance management and monitoring systems are developed for the Android MOS.

Akshay A. Kumbhar et al presented an automated attendance monitoring system using android platform. It is then used to maintain the attendance of the student regularly. Jessenth Ebenezer et al presented an android-based student activity register system. It is used to mark the attendance and to store the details of the students so that the professors or higher officials can view the attendance of the students and regulate them if they are not regular to the classes.

The mobile application-based attendance management system is also employed in the organizations to mark the attendance of the employees. S.P. Avinaash Ram and J. Albert Mayan presented a mobile application for employee registration and mobile attendance. It is used to update the employee attendance regularly and track their attendance. Moreover, it is helpful to the staff and the authorities to take the attendance. This system is also used to know the number of employees easily and to monitor whether they are regular to the organization. This system also provides the details of every employee.

In addition to providing the authentication in organizations, the location-based attendance management system is also practiced. Mohammad Salah et al presented a mobile application for time and attendance system based on the location. This application is used to take the attendance of the employees based on whether they are in the same location of an organization or not. This application is developed using android. The wireless technologies such as Bluetooth and Wi-Fi are used with 428 Dr. D. Asir Antony Gnana Singh, Dr. E. Jebamalar Leavline & P. Meera Vijayan mobile application for attendance management system. Riya Lodha et al developed an application for attendance management system using Bluetooth enabled devices. This application functions with the wireless technology using Bluetooth to mark the attendance. Hence, it reduces the time taken for the attendance marking. Freya. J. Vora proposed a framework of android-based mobile attendance system. It uses Wi-Fi technology to mark the attendance in android based phones. It is allows to store and edit the attendance.

From the literature review, it is observed that the mobile application-based attendance management system plays a central role in the educational institution and the commercial sectors to regulate the attendance of the students and the employees. Moreover, the mobile application-based mark management system enhances the quality of education through easy

access and analysis of marks. However, this mobile application is developed based on the MOS. This paper presents a Java-based mobile application for attendance and mark management system for educational institution.

Following are the systems and applications that are available right now.

2.2 Android Base Smart Learning and Attendance System

Main purpose of research project is to implement Android based application for attendance management system for schools. The proposed project will be implemented in applications such as online studymaterial, notices, academic calendar and online reminders of examination, online attendance record, performance record, and parent intimation system using Android applications. This system helps teacher to take attendance throughsmartphone and keep record of students for their progressive assessment. This system gives a prior intimation tostudent as soon as their attendance goes below the specified attendance threshold in the form of an SMS.

2.3 Attendance Register

Attendance register is developed for students in a simple and efficient way, using which they can view their attendance record on daily basis. Students can select the desired subjects and date, a list of whole attendance record is display along with statistics, percentage, absent and present. Student can select the desired subject and marked his/her attendance.

2.4Attendance Tracker

Attendance tracker is attendance tracking application of students and teachers. Teacher can mark attendance by selecting class. Teacher can edit attendance by changing status present or absent. Student can view his/her attendance record along with statistic bar. Student can download attendance record in Excel file.

2.5 Smart Attendance

Smart attendance is an android application featuring the attendance system. User can Sign up into application with few credentials. Sign In into app to use the features. Teacher is the main actor who can add students, add courses, take attendance, View attendance, and View students. Teacher can view the student list and mark the attendance. Teacher can view the previous record of attendance and students.

2.6 Easy Attendance School College

Easy Attendance School College is an android application useful for teachers. Teacher can use this app for maintainability of attendance record. It has fast interface of marking attendance. Teacher select his/her desired class, a list of student's displays and check box displaying. Teachers check or uncheck box for saving attendance status of student. Teacher can edit the attendance record. Export, import the record. Teacher can also facilitate with Note feature.

2.7 Mobile Based Attendance System

This mobile based attendance system is built to eliminate the time and effort wasted in taking attendances in schools, colleges and universities. This is greatly reduces the amount of paper work required in attendance marking and management. This is an Android application. This application is built for faculty to mark attendance on cellphone.

This system is divided into the following modules:

- **Student Attendance list creation:** Once this application is installed on the phone, it allows the user to create a student sheet consisting of name, roll number, date, absent/present mark and subject. He has to fill the student names along with associated roll numbers.
- **Attendance Marking:** The faculty has the list on his phone now. He can mark the attendance according to that list.
- **Attendance Storage:** This data is now stored on faculty mobile phone. Faculty can view it anytime.

- **Attendance Sheet Transfer:** Faculty can transfer this attendance sheet to a server via bluetooth connection.

Chapter 3

Requirements Specification

3.1 Existing System

There are quite a few existing systems for marking attendance. All systems have their own pros and cons. In most of the existing systems that are currently being used teachers have to call the name of every student to mark their attendance which is quite hectic and time consuming for teachers.

Another existing system that is developed for teachers and students. Core features of SAS system are to maintain the attendance system automatically. Teachers can mark attendance and save the attendance record. Select his/her class and subject, a list of students relevant to subject is display. Mark the attendance by check or uncheck the box. On the other side student can view attendance record by selecting subject. Select date and view the attendance record. This is beneficial for both actors that they fulfil their task with some clicks rather than manually tasking. In this system still teacher have to mark the attendance manually.

RFID based Attendance System uses RFID reader to get the student information through student matrix card. After getting the student information, it will send it to the computer in that class or lab. After that, the individual in charge (professor, staff, and student) must connect to the PC using Bluetooth to make his/her see the student attendant in that class.

A research on Near Field Communication based systems for attendance marking has been carried out, but there were certain limitations in it. There are some shortcomings in this system. The attendance marking process is time consuming and tedious. To avoid proxy of student, manual head count also needs to be performed.

To overcome this problem and to minimize number of proxies where teacher have to call the names and mark attendance manually we developed the Smart Attendance System.

3.2 Proposed System

We are going to develop an Android Smart Attendance System which facilitates the teachers and students to mark attendance easily. Teacher can sign up and sign in to use the application. Whenever teacher will sign in he will get a verification code, teacher will enter that code to sign in. In this way no one can sign in by using stolen Password of teacher. Teacher can select the course and number of hours of attendance. After selecting the course and no of hours teacher will select the time in which student can mark their attendance. When teacher selects the time now he will click on allow button to allow students to mark their attendance in that time. There will be a time running to show the remaining time. After this time is finished attendance will be disabled and teacher will save the attendance.

On students side first of all student have to sign up by giving few credentials. After signing up now student will select the semester and then the course of which he wants to mark his attendance. After selecting course student will mark himself present. By default the student will be absent student have to mark himself present manually. An important thing here is that student must be connected to the same router to which teacher is connected. If student is connected to some other internet connection he will not be able to see if the attendance is open or not.

Currently there are few applications and systems are available for attendance marking for mobile devices but there is so much room for improvement in these applications. On focusing this thing we tried to contribute in this area.

3.3 Requirements Specification

3.3.1 Functional Requirements

In this section we are going to describe the core functionalities of our system.

In table 3.1 functional requirements of the system are explained. We have described the six functional requirements of our system.

Table 3.1 Functional Requirements

Requirement ID	Requirement Statement	Must/Want	Comments
FR001	The application will have a user authentication system	Must	
FR002	Students will be able to mark their attendance	Must	Teacher will allow them
FR003	System will allow the teachers to verify the attendance	Must	
FR004	Admin will be provided DBMS to manage users	Must	
FR005	System will register new users using Two-Step authentication	Must	
FR006	Application will use CAPTCHA for protection	Want	

3.3.2 Non-Functional Requirements

In this section we are going to describe the Non Functional Requirements

In table 3.2 the Non Functional requirements of the system are explained. In our system we described three Non Functional requirements.

Table 3.2 Non-Functional Requirements

Requirement ID	Requirement Statement	Must/Want	Comments
NFR001	The application shall be available on Android	Must	
NFR002	The Application must load within 5 seconds	Must	

NFR003	The application will have touch screen interface which will be intuitive for users	Must	
--------	--	------	--

3.4 Use case Specification

3.4.1 Use Case 1 Modules Choice

Table 3.3 describes the use case 1 which is related to the modules choice. User will select the appropriate module to go to the next interface.

Table 3.3 Use Case 1 Modules Choice

Use Case ID	1
Brief Description	User opens the application and modules choice is displayed.
Actors	Teacher/Student
Pre-conditions	Application must be open.
Post-conditions	The appropriate module is displayed.
Basic Flow	<ol style="list-style-type: none"> 1. This begins with the user starting the application. 2. Modules choice is being displayed. 3. User chooses the module.
Special Requirements	The user must be connected to the internet.

In Figure 3.1 the main menu of the application is explained. From main menu the user will select the appropriate module.

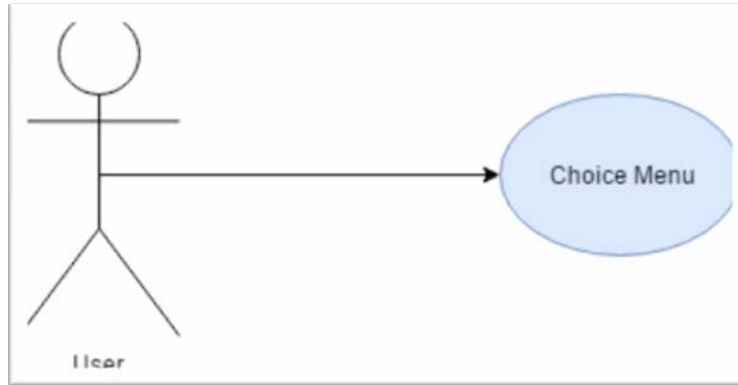


Figure 3.1 Menu Choice

3.4.2 Use Case 2 Sign In/Sign Up

Table 3.4 describes the use case 2 in which user chooses the sign in or sign up. User will go to the relevant interface depending on their choice.

Table 3.4 Use Case 2 Sign In/Sign Up

Use Case ID	2
Brief Description	User will choose between sign in and sign up
Actors	Teacher/Student
Pre-conditions	Sign in and Sign up menu must be open.
Post-conditions	The appropriate menu is displayed.
Basic Flow	<ol style="list-style-type: none"> 1. User will be displayed the buttons to sign in or sign up. 2. User will choose either sign in or sign up
Special Requirements	None

Figure 3.2 comprises the sign in/sign up choices for the user. User will select the sign in or sign up option from the menu.

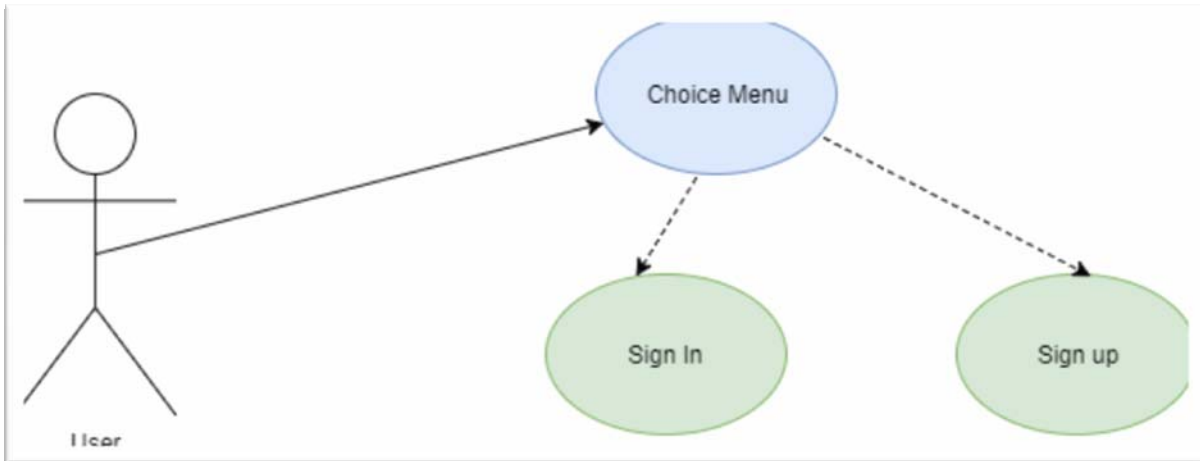


Figure 3.2 Sign In/Sign Up

3.4.3 Use Case 3 User fills Credentials

Table 3.5 describes the use case 3. User fills the credentials for sign up and application sends them the verification code.

Table 3.5 Use Case 3 User fills Credentials

Use Case ID	3
Brief Description	User fills in information for sign up, Application sends verification code for authentication.
Actors	Teacher/Student
Pre-conditions	Sign up form must be opened.
Post-conditions	User registered message pops up.
Basic Flow	<ol style="list-style-type: none"> 1. User will be displayed text boxes to fill credentials. 2. User will get the verification code. 3. After authentication user is now signed up.
Special Requirements	Valid phone number.

In Figure 3.3 the sign up credentials are explained. Figure describes the whole sign up procedure and the required credentials of the user.

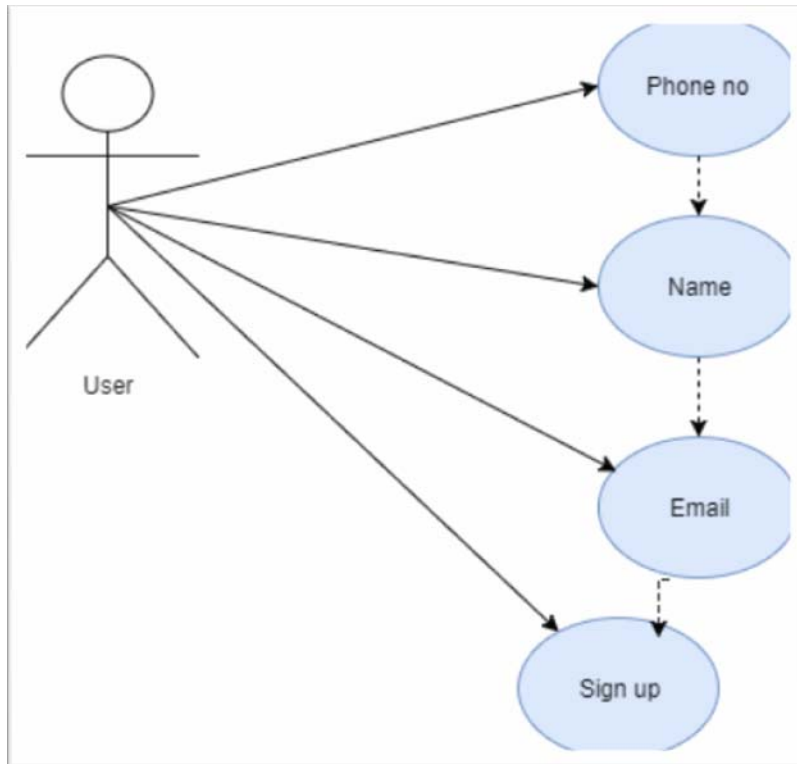


Figure 3.3 User fills credentials for sign up

3.4.4 Use Case 4 Sign In

Table 3.6 describes the sign in user case. User fills the credentials for sign in.

Table 3.6 Use case 4 Sign In

Use Case ID	4
Brief Description	User sign in by giving credentials..
Actors	Teacher/Student
Pre-conditions	Sign in menu must be open.
Post-conditions	user gets signed in.
Basic Flow	1. User will be displayed text boxes to input

	credentials 2. If user is registered, he is logged in. 3. If user is not registered, he is displayed error message.
Special Requirements	User must be registered

Figure 3.4 describes the sign in credentials of the user. If user wants to sign in into the application these credentials must be given.

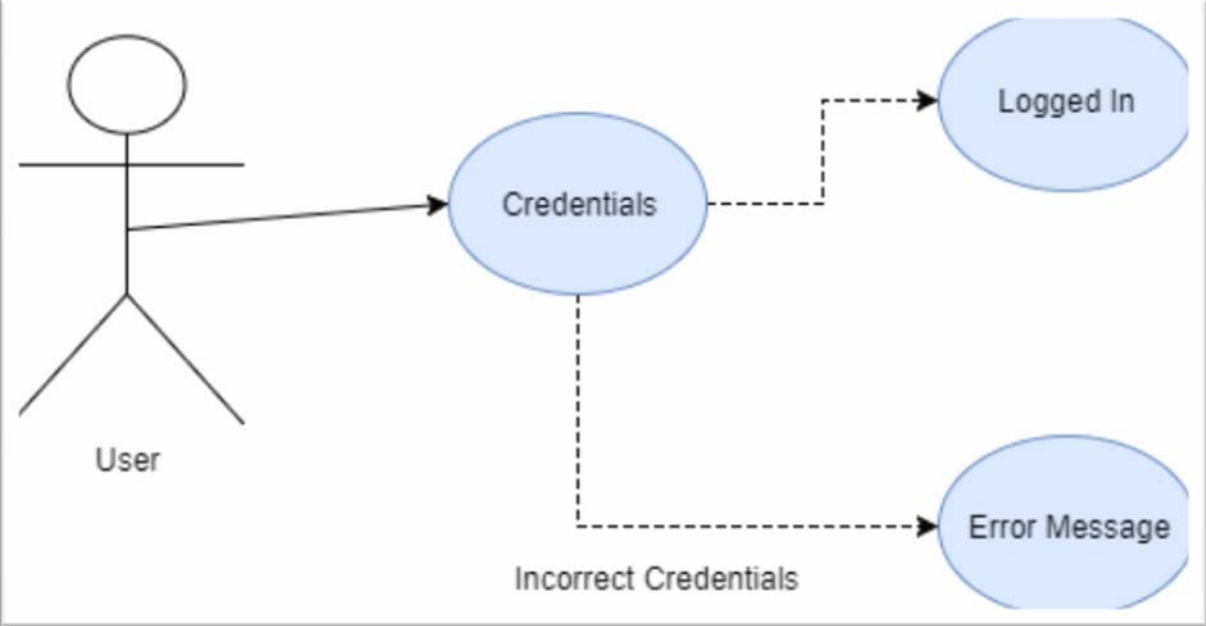


Figure 3.4 User sign in with correct credentials

3.4.5 Use Case 5 Selection of Courses

Table 3.7 describes the use case 5. Teacher selects the courses from the dropdown list and clicks the add button.

Table 3.7 Use Case 5 Teacher selects courses

Use Case ID	5
--------------------	---

Brief Description	Teacher selects the courses from drop down list.
Actors	Teacher
Pre-conditions	Teacher is signed in.
Post-conditions	Courses are selected.
Basic Flow	<ol style="list-style-type: none"> 1. Teacher selects the semester. 2. After semester teacher will select the course.
Special Requirements	None

In Figure 3.5 the semester and course selection is explained. User will select the semester and course.

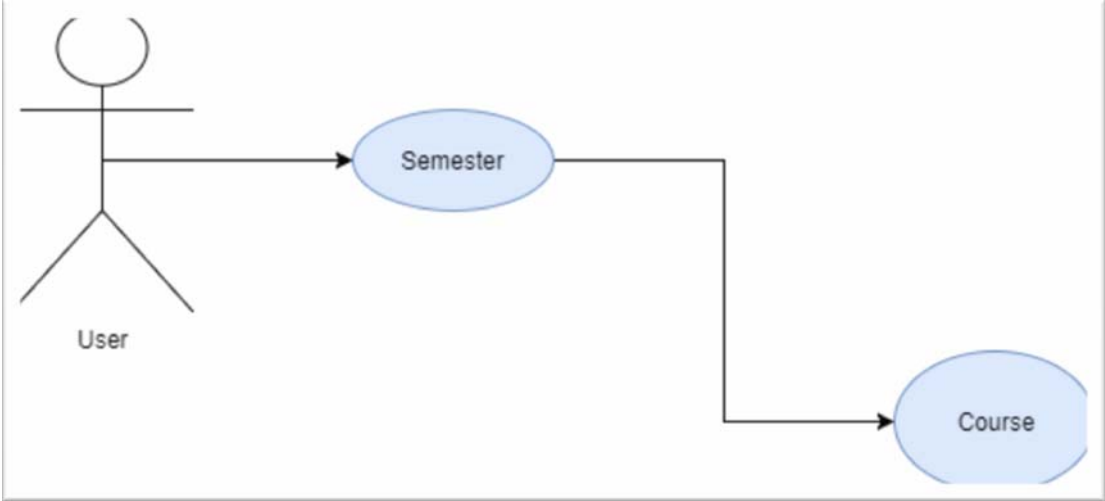


Figure 3.5 User selects Semester and Courses

3.4.6 Use Case 6 Teacher Specifies time for attendance

Table 3.8 describes the use case 6 in which teacher specifies the time frame for attendance. Teacher enters time in minutes and clicks the allow button to allow the attendance.

Table 3.8 Use case 6 Teacher Specifies time

Use Case ID	6
Brief Description	Teacher specifies time frame.
Actors	Teacher
Pre-conditions	appropriate credit hours select
Post-conditions	Timer starts
Basic Flow	<ol style="list-style-type: none"> 1. Teacher enters the time in minutes. 2. Teacher clicks the allow button and timer starts
Special Requirements	None

In Figure 3.6 the time which will be set by the teacher to mark the attendance is explained. Timer will start when the teacher will set the time and in this time students will be able to mark their attendance.

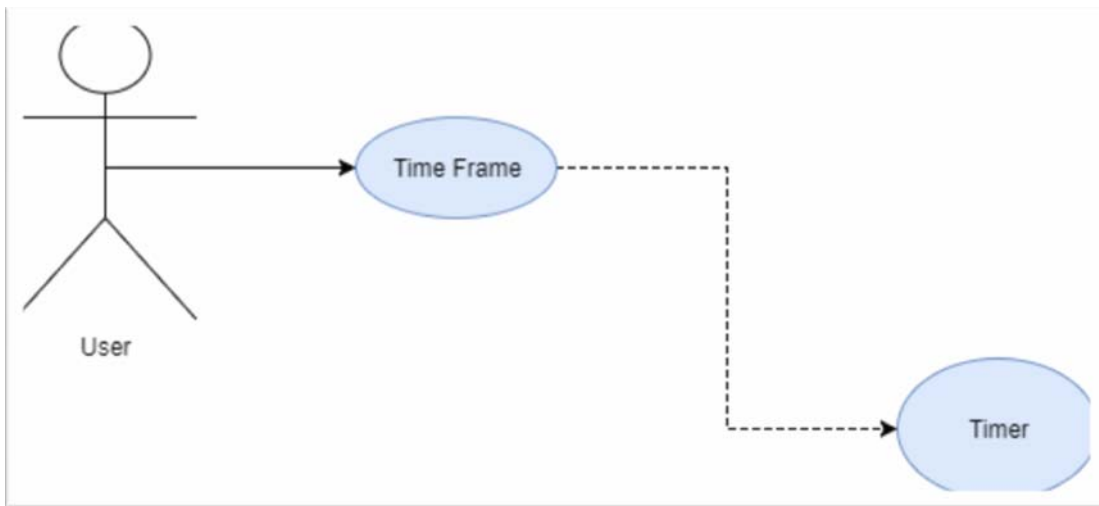


Figure 3.6 Teacher Specify Time Frame for attendance

3.4.7 Use Case 7 List of present students

Table 3.9 describes the use case 7 in which list of present students is shown to the teachers. Teacher sees the list of present students and edits the attendance if teacher want to edit the attendance.

Table 3.9 Use Case 7 List of Present Students

Use Case ID	7
Brief Description	List of present students displayed, teacher save or chooses to edit
Actors	Teacher
Pre-conditions	Students have marked their attendance
Post-conditions	DataBase Updated
Basic Flow	<ol style="list-style-type: none"> 1. Teacher press the view button to view the attendance list. 2. Teacher chooses to edit or save the attendance.
Special Requirements	Teacher and students must be connected to same router

Figure 3.7 shows the list of the present students to the teacher. Teacher will be able to edit the attendance if he wants to.

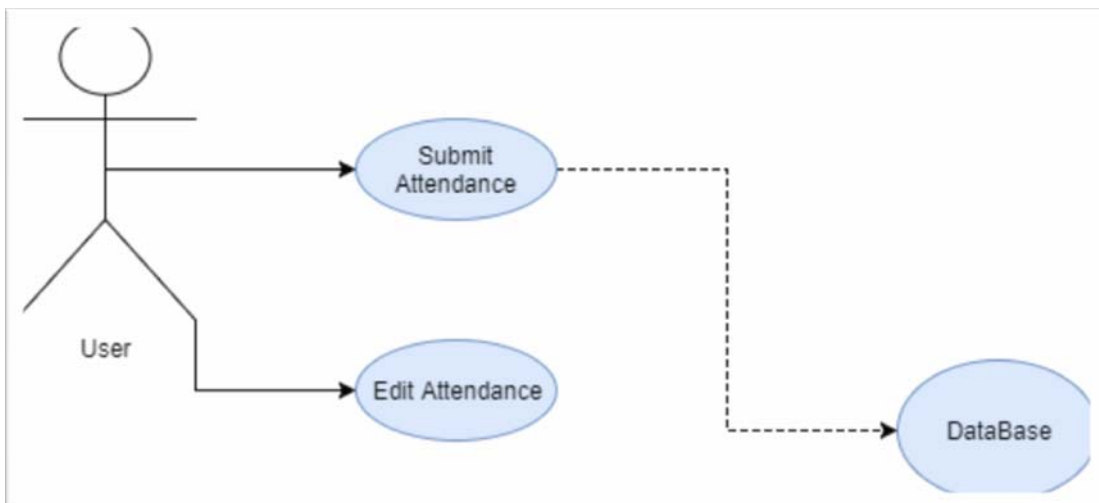


Figure 3.7 List of Present Students

3.4.8 Use Case 8 Teacher edits attendance

Table 3.10 describes the use case 8. Teacher edits the attendance and updated attendance is automatically updated in the database.

Table 3.10 Use Case 8 Teacher edits attendance

Use Case ID	8
Brief Description	Teacher will edit the attendance and save it.
Actors	Teacher
Pre-conditions	Students have marked their attendance
Post-conditions	Attendance is locked
Basic Flow	1. Teacher press the view button to view the attendance list. 2. Teacher edit the attendance and submit it
Special Requirements	Teacher and students must be connected to same router

In Figure 3.8 the editing of the attendance is shown. Teacher edits the attendance and updated attendance will be saved in the database.

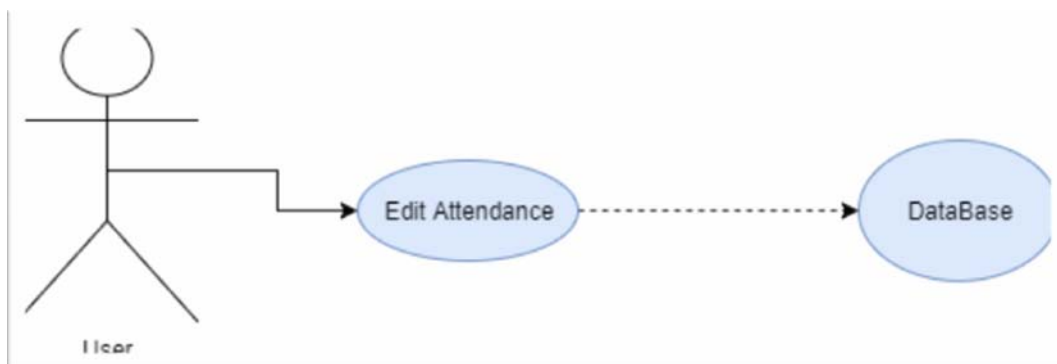


Figure 3.8 Teacher Edit attendance

3.4.9 Use Case 9 Student chooses semester and courses

Table 3.11 describes the use case 9. Student selects the semester and courses from the dropdown list.

Table 3.11 Use Case 9 Student chooses Semester and Courses

Use Case ID	9
Brief Description	Student chooses the semester and course.
Actors	Student
Pre-conditions	Student must be signed in
Post-conditions	Appropriate menu is displayed
Basic Flow	<ol style="list-style-type: none">1. Students select the semester from the dropdown list2. After selecting the semester student will select the course
Special Requirements	None

In Figure 3.9 the selection of semester and courses of the students are explained. Student selects the semester and course.

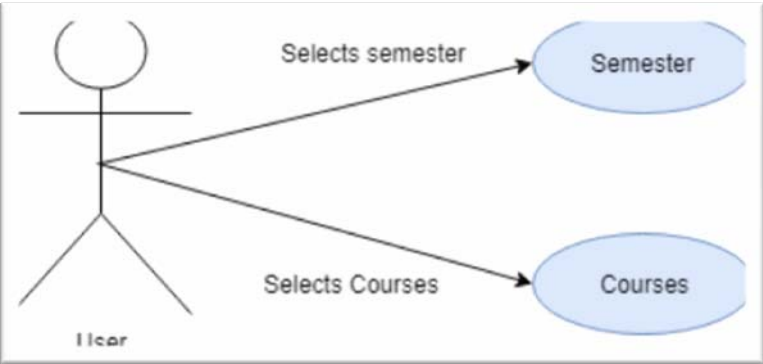


Figure 3.9 Student selects semester and courses

3.4.10 Use Case 10 Student Marks Attendance

Table 3.12 describes the use case 10. Students mark their attendance in the specified time and attendance is shown to teacher.

Table 3.12 Use Case 10 Student Marks Attendance

Use Case ID	10
Brief Description	Student is displayed the menu to mark attendance
Actors	Student
Pre-conditions	Teacher have allowed the attendance and timer is still running
Post-conditions	Attendance is saved in the database
Basic Flow	<ol style="list-style-type: none">1. Teacher press the view button to view the attendance list.2. Teacher edit the attendance and submit it
Special Requirements	Teacher and students must be connected to same router

In Figure 3.10 marking of the attendance is shown. Students mark their own attendance in the given time.

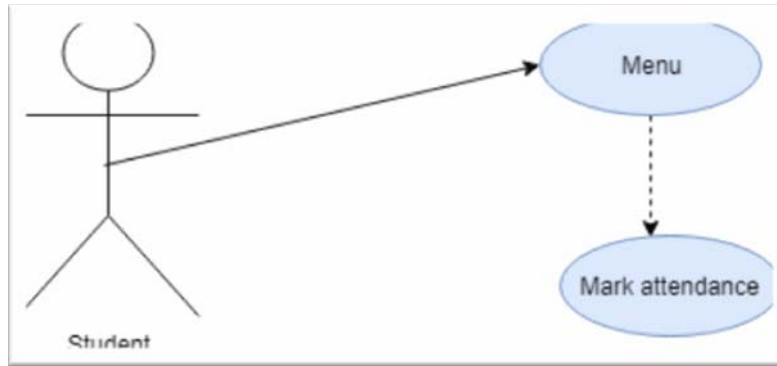


Figure 3.10 Student marks attendance

3.4.11 Use Case 11 User Logs Out

Table 3.13 describes the use case 11. User clicks the logout button and system logs the user out.

Table 3.13 Use Case 11 User logs out of the system

Use Case ID	11
Brief Description	User logs out of the system.
Actors	Teacher/Student
Pre-conditions	User must be logged in
Post-conditions	User should be logged out
Basic Flow	<ol style="list-style-type: none"> 1. User clicks the logout button 2. The system logs the user out
Special Requirements	Teacher and students must be connected to same router

In Figure 3.11 the logout of the user is shown. User clicks the logout button and system log them out.



Figure 3.11 User Log Out

Chapter: 04

Design

4.1 Introduction

This chapter describes the design of the developed system. The design of the system includes the System Architecture, Design Constraints, Design Methodology, High Level Design, Low level Design. This will give an overview of the whole system.

4.2 System Architecture

The system developed is a mobile application, specifically for an Android Platform. Below is the context diagram which show the interactions of external entities with the developed system. External entities include teacher and the student. These are the external entities who are interacting with the system. Teacher is basically allowing the attendance to the students and students are marking their attendance. After students marked their attendance teacher verifies that attendance by viewing the attendance list and if teacher wants to edit the attendance he can edit and then save the attendance to the database.

4.2.1 Context Diagram

Following is the context diagram showing all the entities interacting with the system

In Figure 4.1 the interaction of the entities of the system is explained. Figure shows how the different entities of the system interacting with each other.

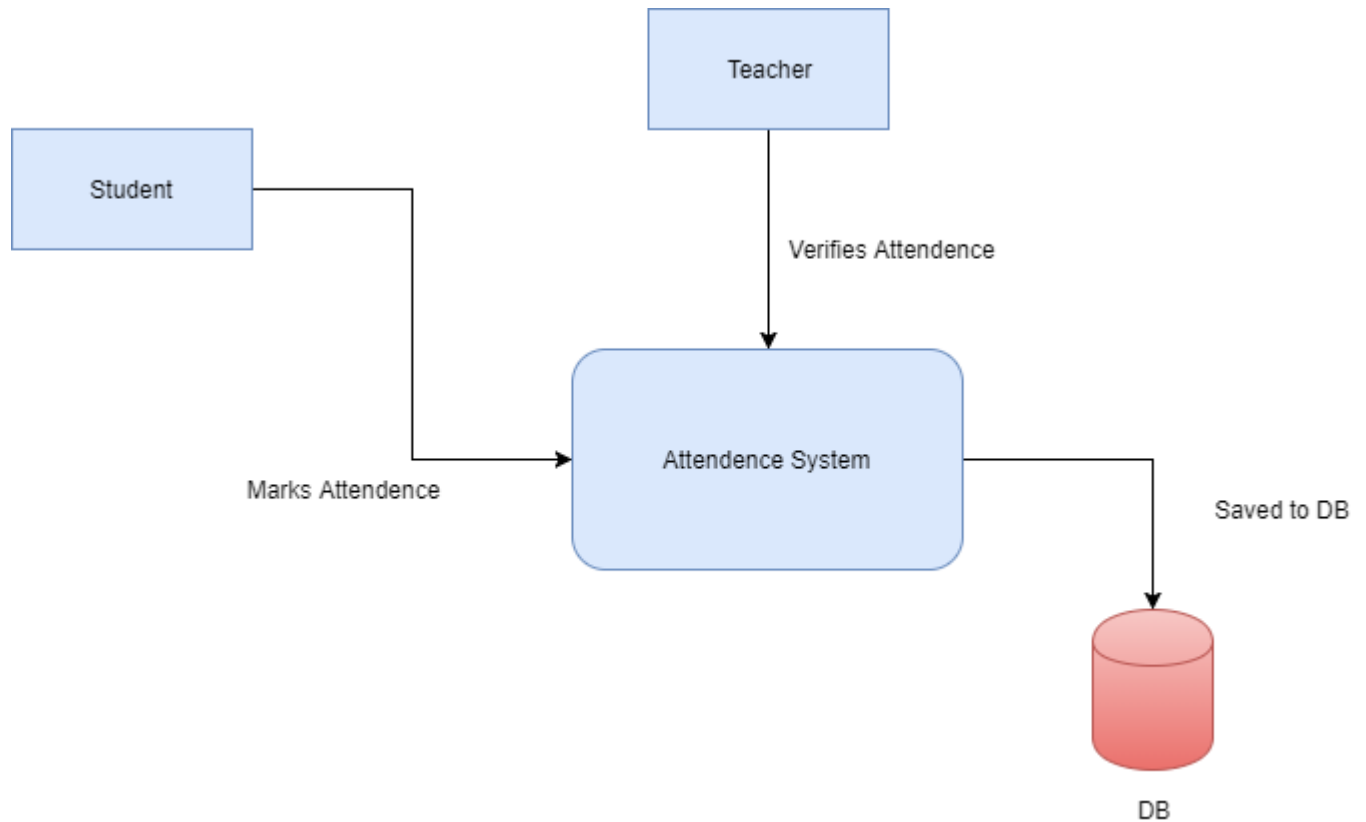


Figure 4.1 Context Diagram

4.2 Design Constraints

One of the most important factors in deciding on a design process is to be fully aware of design constraints and limitations. The chosen design process for this project aims to minimize the impact of design constraints. This application is based on Android Platform. There are compatibility constraints in this application. Application compatible with few versions of Android. When we develop an Android application we use SDK version. The API's used doesn't support the older versions of Android. Users of the application must be connected to the WiFi in order to use the application. Application doesn't run on the Mobile data. With a timeframe of 4 to 5 months to complete literature studies and software design and synthesis and write a project report, the design process must be optimized to maximize productive time. Design processes may take place in parallel if properly coordinated.

4.3 Activity Diagram

This diagram explains the user activities that take place in his interaction with the system.

In Figure 4.2 the user activities of the user are explained in detail. Figure shows the different activities of the user with the system.

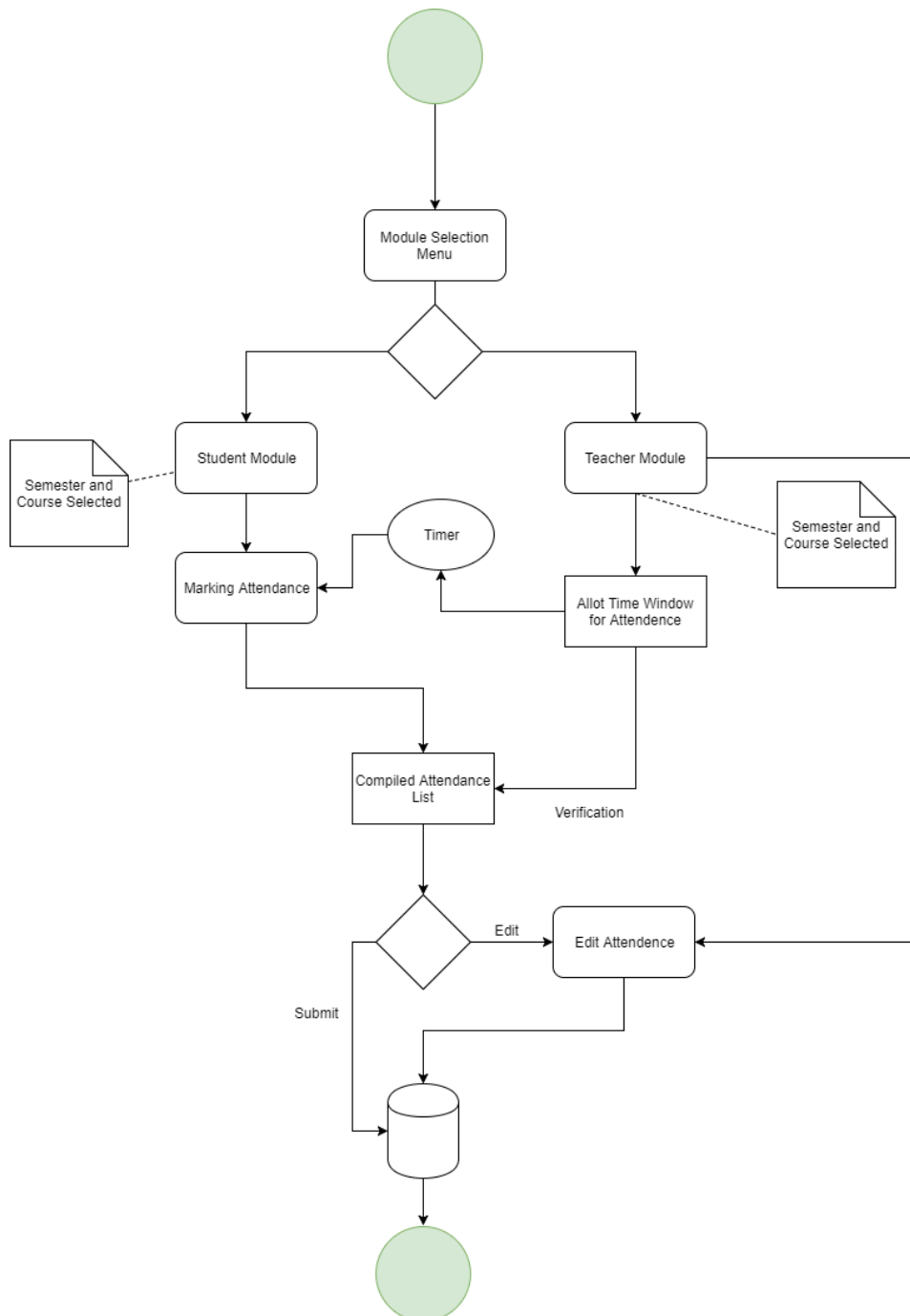


Figure 4.2 Activity Diagram

This application includes teacher and student module. The user interface allows the user to select the appropriate module. It presents the user ability to sign in or sign up to the application after user authentication. Teacher's view is different and student view is different. The application allows the user to navigate through the interface easily and mark attendance.

4.5 Sequence Diagram

The sequence of interactions that happen within the system are explained in the following sequence diagram.

In Figures 4.3 the sequence of the actions of the user with the system is explained. All the actions of the user are shown in this sequence diagram.

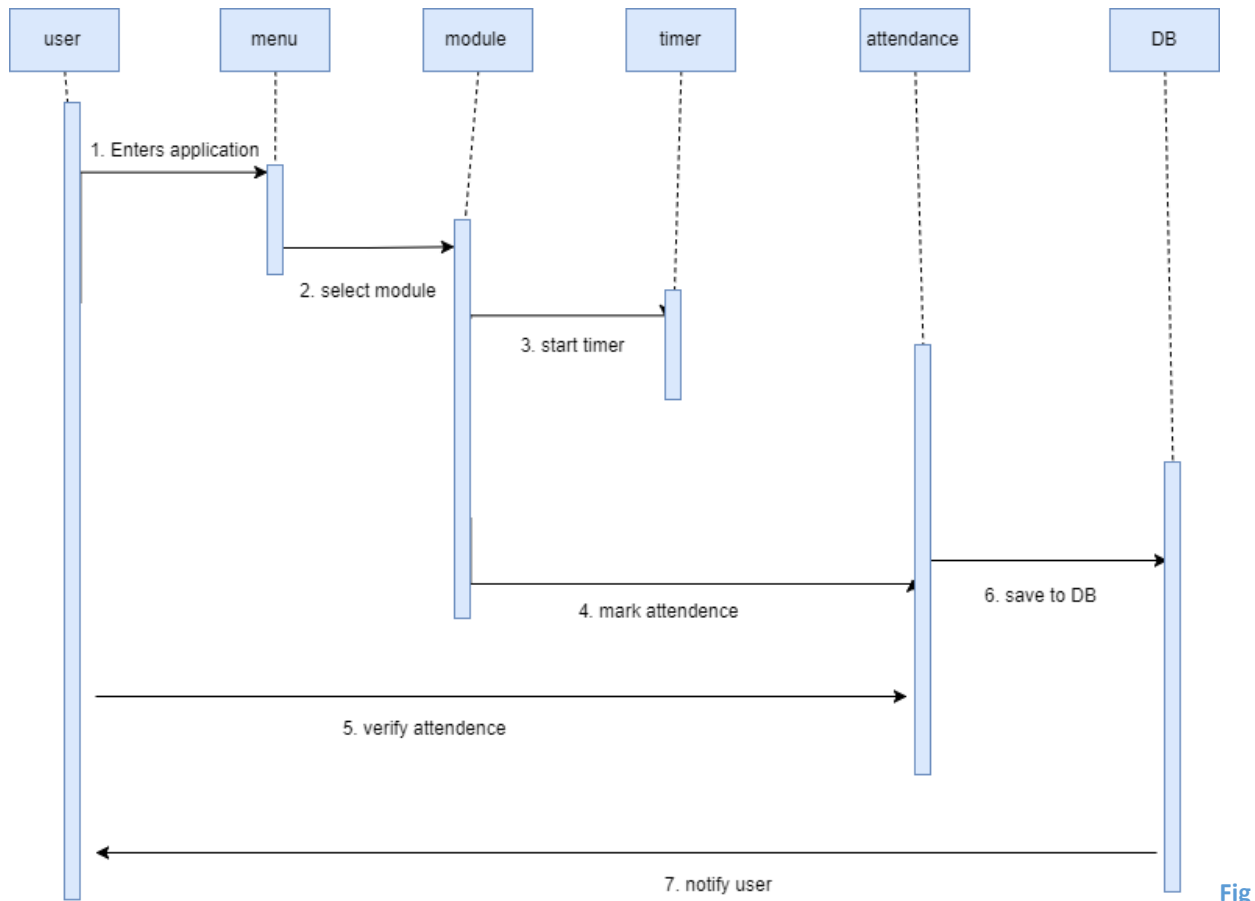


Figure 4.3 Sequence Diagram

It starts with the user selecting the correct module. Teacher will select the teachers module and student will select the students module. After user selected the correct module user will go to his interface. Teacher will select the semester and courses and will allow the attendance to the students. Similarly students will also select the semester and courses and will mark their present. After attendance time is over teacher will view the attendance and save it.

Chapter: 05

Implementation

5.1 Introduction

This chapter focuses on tools and technology that we are implemented for development of our system.

5.1.1 Tools and Technologies

Following are the tools and technologies which were used to develop this application.

5.1.1.1 Android Studio

We used the Android Studio to build our application. Android Studio is the official integrated development environment for android operating system, built on JetBrains intellij IDEA software and designed specifically for Android Development. It is available for windows, mac

and linux based operating system. It is a replacement for Eclipse Android Development tools for Android Development. It has gradle based build support specific refactoring and quick fixes. Lint tools to catch performance, usability, version compatibility and other problems. Template based wizards to create common android designs and components. A layout editor that allows user to drag and drop UI components. It has built in support for Google cloud platform which enables integration with firebase cloud messaging. It has Android virtual device to run and test the applications.

Figure 5.1 shows the main project menu of the Android Studio. The main menu of the Android Studio and it's working is shown in the below Figure 5.1.

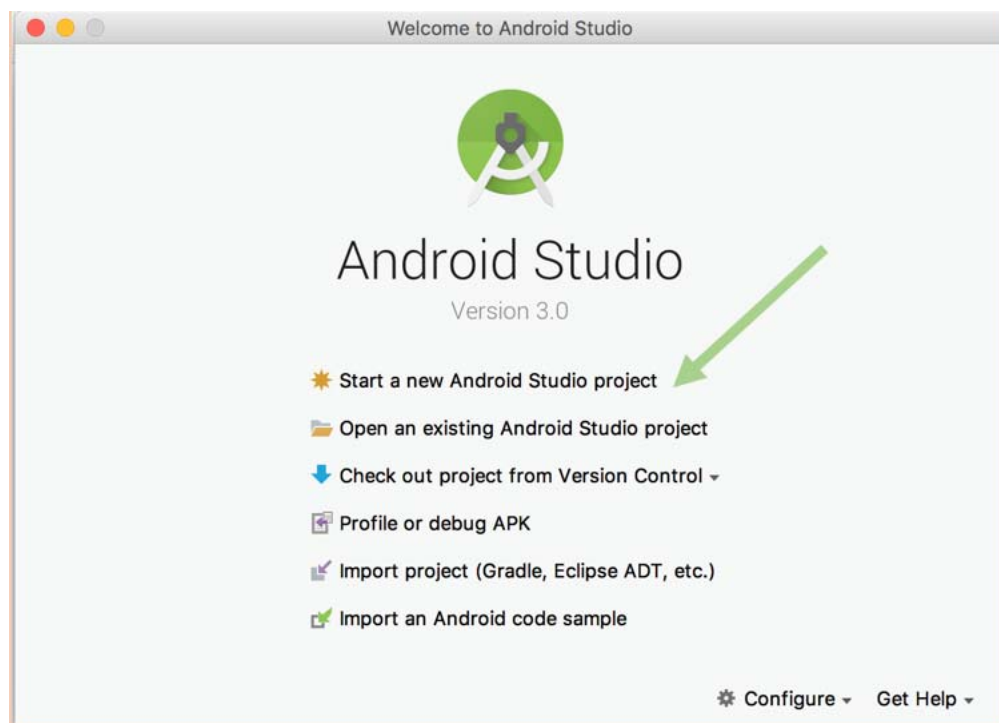


Figure 5.1 Android Studio Project Menu

5.1.1.1.1 Interface

The Android Studio main window is made up of several logical areas identified as shown in the figure below.

Figure 5.2 shows the main interface of the Android Studio. Figure shows all the detail of the main interface and how it works.

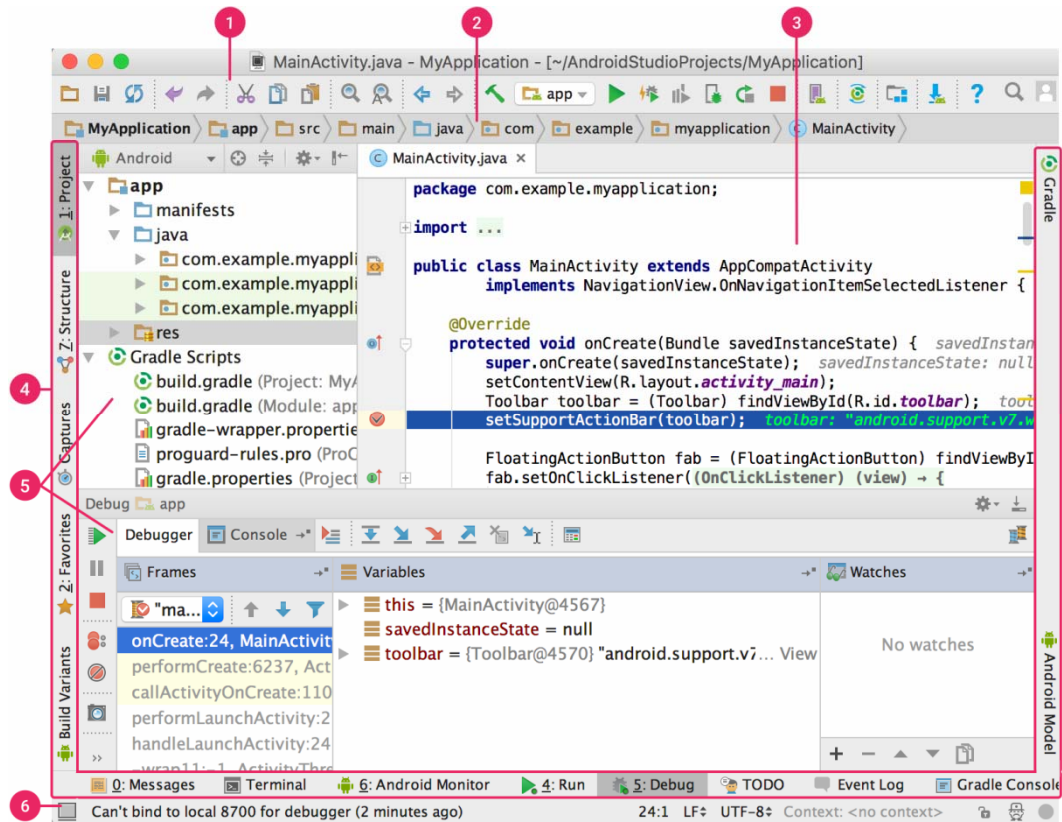


Figure 5.2 Main Interface of Android Studio

1. The toolbar lets you carry out wide range of actions, including running your app and launching Android tools.
2. The navigation bar helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the Project window.
3. The editor window is where you create and modify code. Depending on the current file type, the editor can change. For example, when viewing a layout file, the editor displays the Layout Editor.
4. The tool window bar runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows.
5. The tool windows give you access to specific tasks like project management, search, version control, and more. You can expand them and collapse them.
6. The status bar displays the status of your project and the IDE itself, as well as any warnings or messages.

5.1.1.2 Java

We used java programming language to develop our application. Java is a programming language that is class based, object oriented specifically designed to have a few implementation dependencies as possible. It is intended to let the developers write once and run anywhere meaning compiled java code can run on any platform.

5.1.1.3 Netbeans Eclipse

Eclipse is an Integrated Development Environment designed for application development in Java and J2EE. Eclipse is written in Java and is most widely used to develop Java applications. We used it to write classes for our application wherever immediate testing was not a concern. Eclipse was used in the development of our project in conjunction with Android Studio because we felt that bug fixing is easier done in Eclipse.

5.2 Software Development Model

5.2.1 Star Model

Figure 5.3 shows the star model. Star Model is suggested by Hartson and Hix in 1989. Everything is evaluated in this model and implemented at the same time.

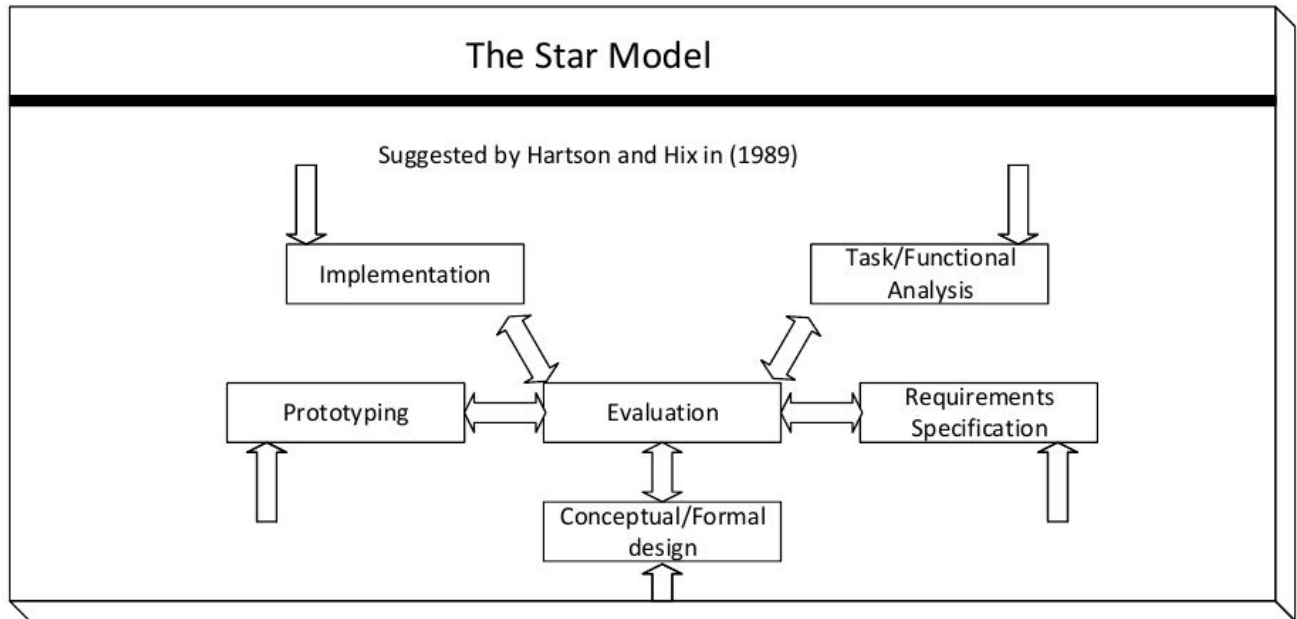


Figure 5.3 Life Cycle of Star Model

Important Features

- Derived from empirical studies of interface designer.
- Emphasizes fast prototyping.
- Evaluation at the center of activities.
- We can start design in any phase.
- Two modes of activates:
 - Analytic (Top down, Organizing, Formal working form system view to user view).
 - Synthetic (Bottom up, Thinking, Creative, Adhoc working from user view to system view).

5.3 Modules

The application was developed with keeping modularity in mind. This helps to improve and update individual components without causing disturbance to the whole system.

In Figure 5.4 all the modules of the application and their working is explained. Figure shows how modules are working together in this application.

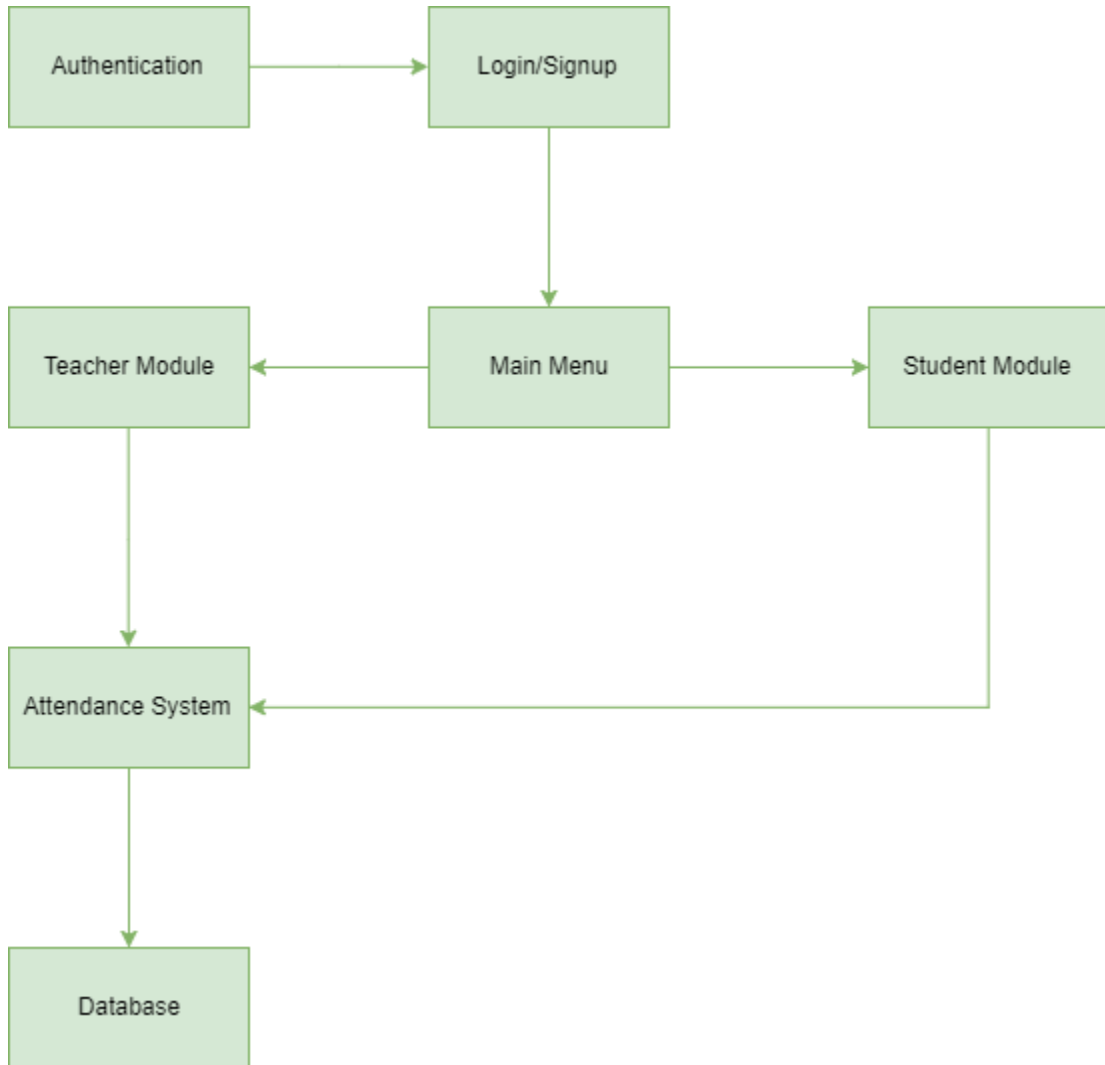


Figure 5.4 Working of application Modules

Authentication module handles all User Sign-In and Sign-Ups. For Sign-Ins it uses a cloud based database system with 2-Factor authentication system for added security. Therefore it only asks for the phone number of the user and then using the authentication code as password logs them in to the system. For new Sign-ups it follows the same basic practice, it asks for new user information and then it uses their phone number to successfully create their ID.

It uses a simple and elegant menu UI for good user navigation within the application. Through this menu system the user can access one of the two further attendance modules: Teacher module, student module.

The teacher module is only accessible by those users who have logged in as a teacher. This module contains a timing system and a WiFi regulation system. It only allows a fixed amount of time to be set in the timer which is then used by the student module. The WiFi regulation system uses Service Set Identifier (SSID) to allow a single connection to be used by all users in an instance of attendance module.

The student module first connects with the attendance module instance used by the teacher module to match the internet connection and then verifies that it is within the timeframe set for the attendance.

The database we used is Google Firebase. It is a cloud based DB which uses NoSQL. It allows data to be available in real-time for all our client devices and remains available even if the app is offline.

5.4 Getting SSID of Router

In Figure 5.4.1 the piece of code is shown. This code is used to get the SSID of the router.

```
WifiManager wifiManager = (WifiManager) getApplicationContext().getSystemService(Context.WIFI_SERVICE);  
WifiInfo wInfo = wifiManager.getConnectionInfo();  
String ssid = wInfo.getSSID();
```

Figure 5.4.1 SSID of Router

5.5 GUI

In Figure 5.5.1 the main interface of the application is shown. There are two buttons to navigate to the next activity of the application. Teacher will click on the teacher button and student will click on the student button to get to the next interface of the application.

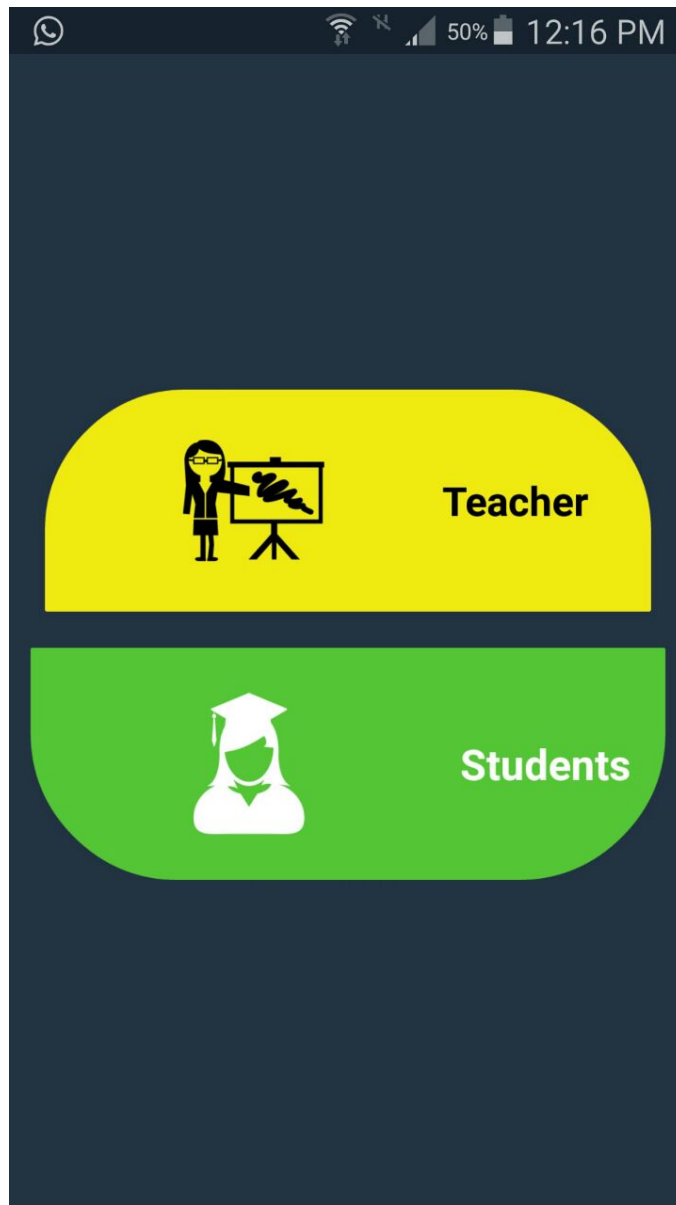
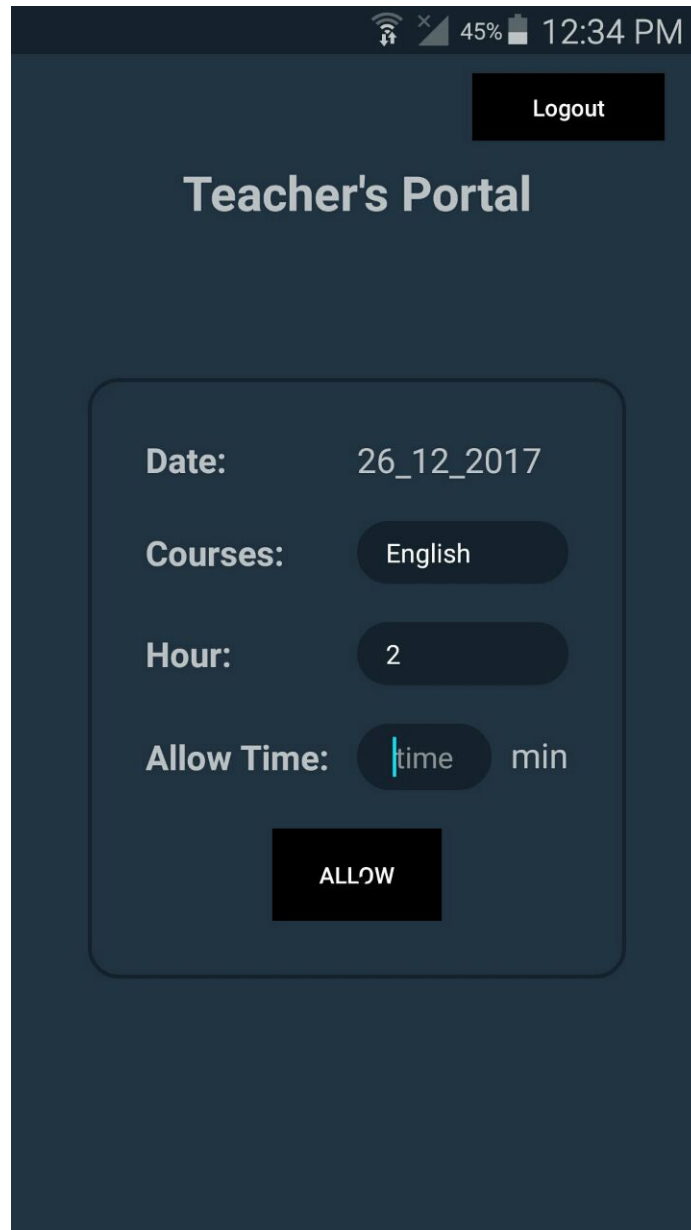


Figure 5.5.1 Main Screen

In Figure 5.5.2 the main teacher's portal is shown. Teacher will select the course and no of hours for attendance. After selecting course and hours teacher will enter the allowed time in minutes and then teacher will click the allow button to allow the attendance.



The screenshot shows a mobile application interface for a teacher's portal. At the top, there is a status bar with icons for Wi-Fi, signal strength, 45% battery, and the time 12:34 PM. Below the status bar is a dark blue header with a "Logout" button in the top right corner. The main title "Teacher's Portal" is centered in a large, white font. Below the title is a rounded rectangular form containing the following fields: "Date:" with the value "26_12_2017"; "Courses:" with a dropdown menu showing "English"; "Hour:" with a dropdown menu showing "2"; and "Allow Time:" with a text input field containing "time" and a "min" label to its right. At the bottom of the form is a large black button with the text "ALLOW" in white capital letters.

Figure 5.5.2 Teacher's Portal

In Figure 5.5.3 the running timer is shown. Teacher has selected the courses and no of hours for attendance. Teacher allowed the attendance for ten minutes. Students can mark the attendance within this given time. Timer is running to show the remaining time of attendance.

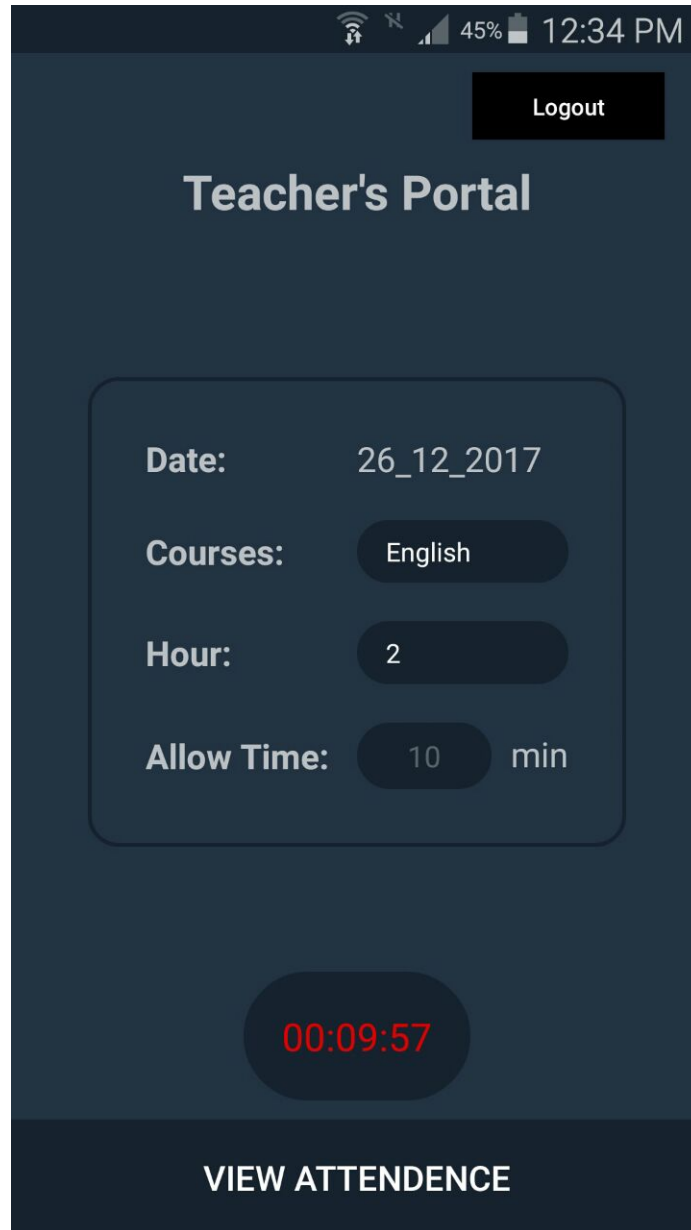


Figure 5.5.3 Timer Running

In Figure 5.5.4 the attendance view of the teacher is shown. Teacher can view the attendance and edit the attendance as well.

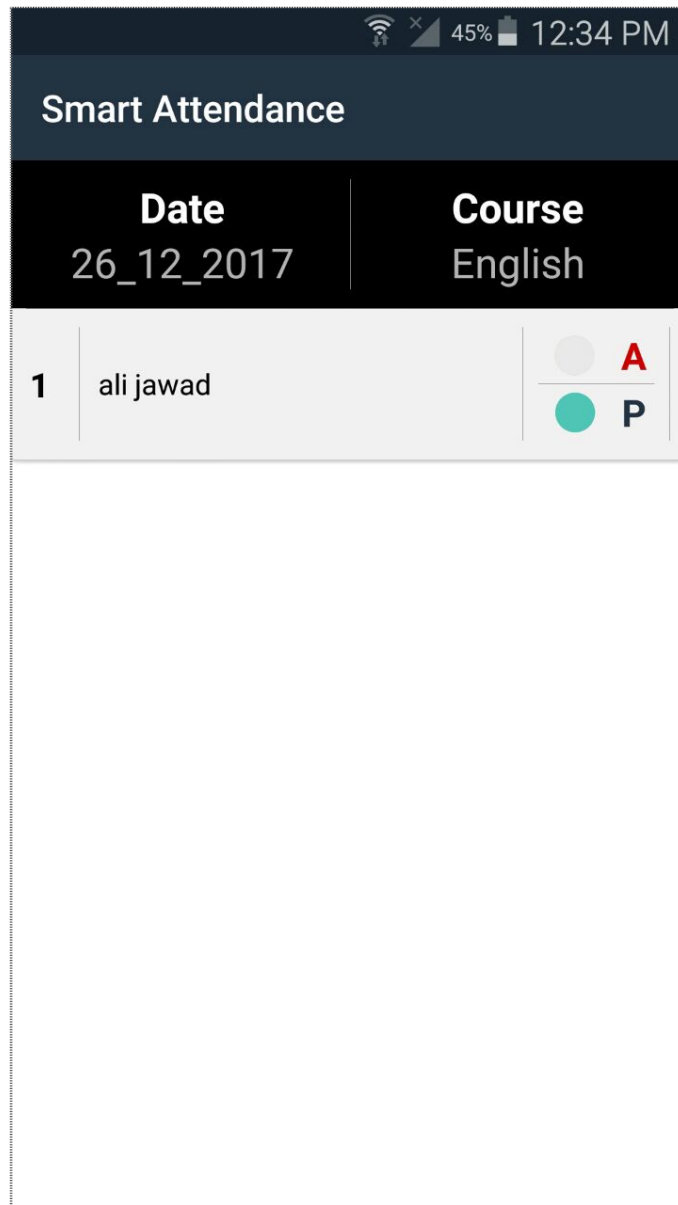


Figure 5.5.4 Attendance View

In Figure 5.5.5 the view of student's portal is shown. Remaining time for attendance is shown through the timer. As attendance is allowed by the teacher student can mark their attendance now.

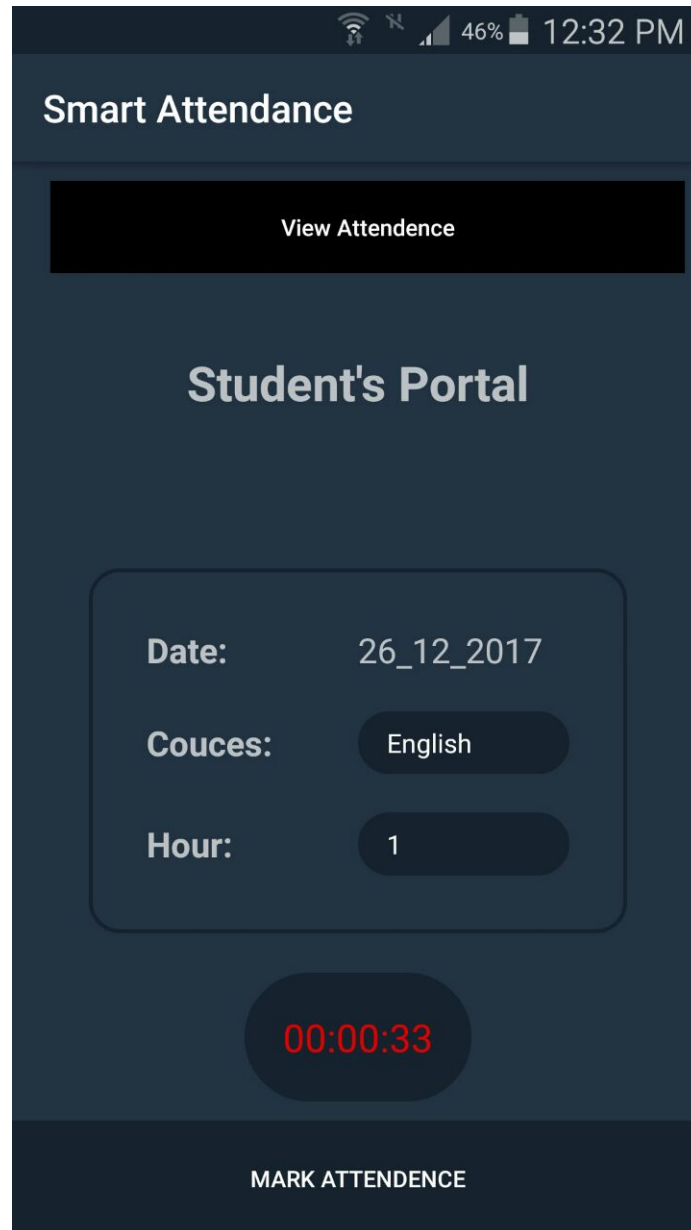


Figure 5.5.5 Student's Portal

In Figure 5.5.6 student marking attendance is shown. By default student is absent student have to mark themselves present. Student will select the present radio button and click the submit button to submit the attendance.

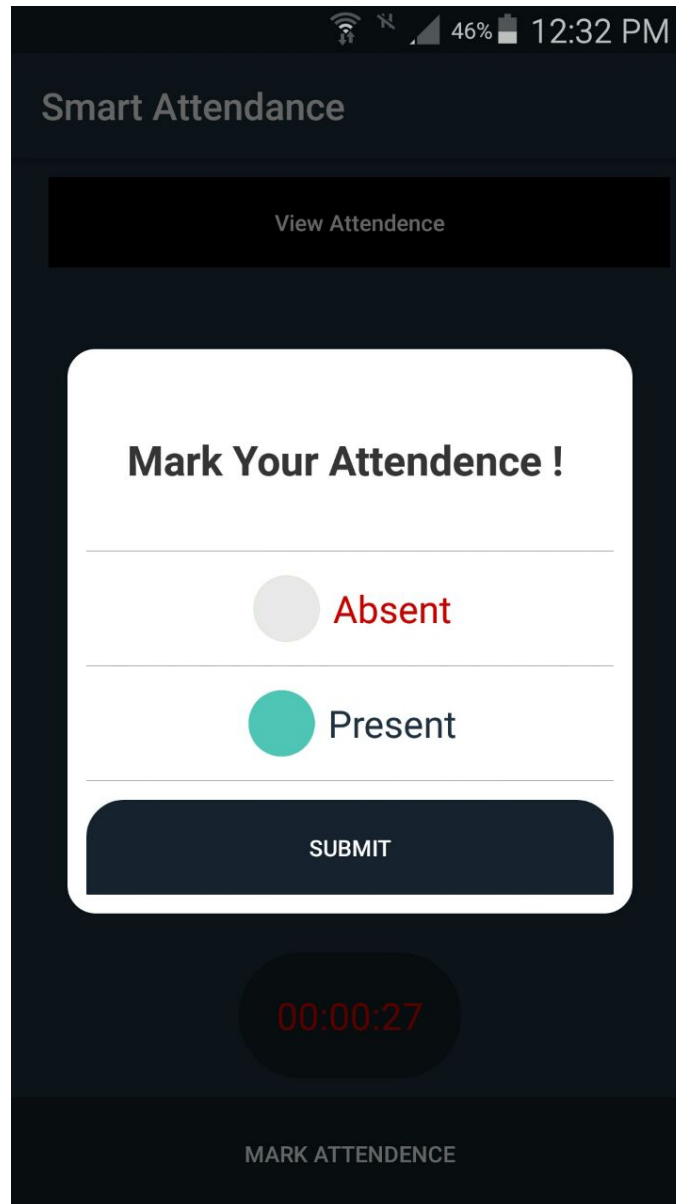
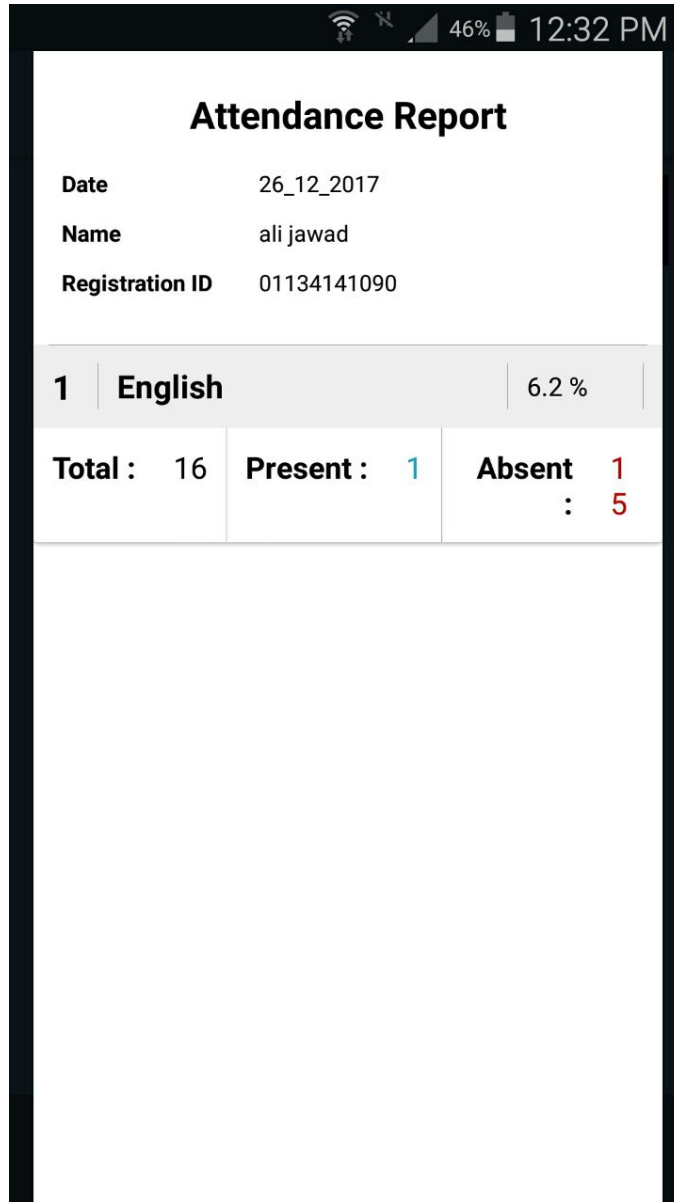


Figure 5.5.6 Marking of attendance

In Figure 5.5.7 the attendance report of student is shown. Student can view the report of every course. Total credit hours, present hours and absent hours are shown in the attendance report. Percentage is also calculated of every course.

A screenshot of a mobile application showing an attendance report. The status bar at the top shows a Wi-Fi icon, a signal strength icon, a battery icon at 46%, and the time 12:32 PM. The main content area has a white background with a black border. At the top, the title "Attendance Report" is centered in bold black text. Below the title, there are three lines of text: "Date 26_12_2017", "Name ali jawad", and "Registration ID 01134141090". Below this is a table with a grey header row. The header row contains "1", "English", and "6.2 %". Below the header is a white row with three columns: "Total : 16", "Present : 1", and "Absent : 1". The "Absent" value is further broken down as "5" in red text below the "1".

1		English	6.2 %
Total :	16	Present :	1
		Absent :	1
			5

Figure 5.5.7 Attendance report

Chapter: 06

System Testing and Evaluation

6.1 Introduction

The purpose of testing and evaluation is to ensure that our system is meeting with our actual requirements. Core functionalities are completed and well-functioning.

6.2 Testing Methods

There are multiple methods that can be used for software testing ranging from unit testing, integrating testing, system testing, white box testing and black box testing.

6.2.1 Unit Testing

Unit testing was applied as the application was being developed by evaluating every function after it was written. The function was compiled, run and then its effects were examined thoroughly. Unit testing as the application was being developed helped us save a lot of time as the number of functions in the application increased which if left un tested would have left us with long list of functions that needed testing. It revealed that everything is in order and working as intended.

6.2.2 Integration Testing

This testing was done after completing a single module to test if every small component is working as intended. Like after implementing the teacher module it was tested and then when selecting semesters and courses module added, they were both tested again to confirm if they work in harmony and don't disturb each other. This eventually helped to the whole system being tested in our next testing phase.

6.2.3 System Testing

After completing all of the modules of the application whole application was tested under different conditions and results were examined. Every part of the application was examined and satisfactory results were produced.

Test Case 1: Installing Application

Table 6.1 comprises the use case 1 which is related to the installing of the application. It explains the whole use case and whether the test case is passed or failed.

Table 6.1 Installing Application

Test Case ID	01
Description	Tests if application installs correctly
Platform	Android
Requirements	Application should install successfully on compatible devices
Steps to be taken	<ol style="list-style-type: none">1. Copy application .apk file to phone2. Install using Android phone's file manager
Expected Result	Application should install successfully
Actual Result	Application installs successfully
Status	Success
Remarks	N/A

Test Case 2: Running the application

Table 6.2 comprises the test case 2 running the application. This test case is about the running of the application on a compatible device.

Table 6.2 Running the application

Test Case ID	02
Description	Tests to see if application runs fine

Platform	Android
Requirements	Application should run on all compatible Android Devices
Steps to be taken	<ol style="list-style-type: none"> 1. Find application in phone menu 2. Run application
Expected Result	Application should run successfully
Actual Result	Application runs successfully
Status	Success
Remarks	N/A

Test Case 3: Running application on Incompatible OS version

Table 6.3 comprises of the test case 3. Test Case 3 is running the application on the incompatible devices.

Table 6.3 Running Application on Incompatible OS version

Test Case ID	03
Description	Tests to see if application doesn't allow incompatible OS version
Platform	Android
Requirements	Application should display an error and close if the OS version is incompatible
Steps to be taken	<ol style="list-style-type: none"> 1. Run application on incompatible OS
Expected Result	Application should display error message and close
Actual Result	Application closes after displaying an error message

Status	Success
Remarks	N/A

Test Case 4: Menu Navigation is Hierarchical

Table 6.4 contains the test case 4 of the system. In this test case the main menu of the application is tested whether it is working properly or not.

Table 6.4 Menu Navigation

Test Case ID	04
Description	Application has correct menu hierarchy in place
Platform	Android
Requirements	All menus are displayed to the user in a proper sequence
Steps to be taken	<ol style="list-style-type: none"> 1. Run application 2. Select module 3. Select sign-in or sign-up
Expected Result	The navigation system is in check with correct sequence of user actions
Actual Result	The navigation system is in check with correct sequence of user actions
Status	Success
Remarks	N/A

Test Case 5: Sign In Sends Verification Code via SMS

Table 6.5 comprises the test case 5 of the system. In test case it is tested whether system send the verification code to the user for authentication or not.

Table 6.5 Verification Code via SMS

Test Case ID	05
Description	Application sets up 2-Step authentication at SignUp
Platform	Android
Requirements	Application should send an SMS to the user's phone number on sign up.
Steps to be taken	<ol style="list-style-type: none"> 1. Fill in credentials 2. Hit submit to sign up
Expected Result	The application sends an SMS to the user's phone
Actual Result	Authentication code is sent to the user via an SMS
Status	Success
Remarks	N/A

Test Case 6: Verifies correct authentication Code

Table 6.6 contains the test case no 6. In this test case it is been checked whether system authenticates the correct verification code or not.

Table 6.6 Verifies Correct Authentication Code

Test Case ID	06
Description	Application completes 2-Step authentication
Platform	Android
Requirements	The application verifies the 2-Step authentication code sent by SMS
Steps to be	<ol style="list-style-type: none"> 1. Submit the authentication code received through SMS

taken	
Expected Result	Successful code results in completion of authentication process
Actual Result	Application allows the user to SignUp after authenticating correct code
Status	Success
Remarks	N/A

Test Case 7: Error at Incorrect Verification Code

Table 6.7 comprises of the test case no 7. In this test case we checked if system shows an error to user on an incorrect verification code.

Table 6.7 Error at Incorrect Verification Code

Test Case ID	07
Description	Application rejects incorrect attempts of authentication
Platform	Android
Requirements	The application verifies the 2-Step authentication code sent by SMS
Steps to be taken	1. Submit the authentication code received through SMS
Expected Result	Application displays error message on invalid code
Actual Result	Error message is displayed after entering incorrect code.
Status	Success
Remarks	N/A

Test Case 8: User Signs in using Correct Credentials

Table 6.8 comprises of the test case no 8. This test case is performed to check if system signs in the user when user give the correct credentials.

Table 6.8 User Sign In Using Correct Credentials

Test Case ID	08
Description	Application allows user to sign in with correct credentials
Platform	Android
Requirements	Users can only sign in using their registered credentials
Steps to be taken	<ol style="list-style-type: none">1. Fill in username2. Enter password3. Hit Sign in
Expected Result	User is logged in
Actual Result	User is logged in
Status	Success
Remarks	N/A

Test Case 9: User is not allowed to sign in using Incorrect Credentials

Table 6.9 contains the test case no 9. In this test case it is been tested if user gives the incorrect credentials what is the response of the system.

Table 6.9 User with Incorrect Credentials

Test Case ID	09
---------------------	----

Description	Application rejects incorrect attempts at signing in
Platform	Android
Requirements	Users can only sign in using their registered credentials
Steps to be taken	<ol style="list-style-type: none">1. Fill in username2. Enter password3. Hit Sign in
Expected Result	Application displays error message on invalid credentials
Actual Result	Error message is displayed after entering incorrect username/password
Status	Success
Remarks	N/A

Chapter: 07

Conclusion

7.1 Introduction

Smart Attendance system is a platform which can facilitate the user with attendance system. There are two main actor involve, Teacher and Student. Teacher can use the system to mark the attendance with respect to courses and save the attendance record on daily bases. Once teacher marked and upload the attendance he/she cannot change the attendance, attendance will be locked. Student can also facilitate with system by marking their attendance with permission of teacher. They can view their attendance record on daily bases. They select the course which he wants to see the attendance. This system provides feasibility, security and maintainability to both actors. The hosted database is used to save the whole record of attendance. Admin panel is managed with web portal and application is developed in android, Java.

7.2 Problem Description

Attendance system is manually implemented in almost everywhere. Where teachers and students have to mark and view attendance manually by using some web application. In these attendance systems teachers have to call the names of every student one by one or in some systems a scanner or some hardware devices were moved to every student to mark his/her attendance through thumb print verification to mark their presence in the class, which is costly to buy such hardware devices, quite difficult and time consuming for teachers. Sometime in some web based CMS, teacher's password hacked and students misuse it to mark their proxies. Now a days there are so many platforms like android. We can built applications by using these tools and platforms and make life easier for teachers and students. Teachers and students need an easier way to mark the attendance where they can

fulfill their requirements by some clicks. Our application provides that to both teacher and as well as students. Teacher and student just have to install the application on their android cell phone and sign up by giving some credentials.

7.3 Solution advantages

Smart attendance system is cost effective and safe. Whenever teacher will sign in, a verification code will be send to his cell phone and SAS will automatically get that code and then teacher will get sign in. In this system there are very less chances of proxies and this system save much time of teacher and students.

7.4 Limitations

Though the routers available in market don't have capability to adjust or control wifi range so if a student is not in class but he/she is in the range of wifi he can also mark their attendance. To deal with such cases, teacher will use random time to mark attendance. Teacher can allow attendance at any time during the class there is no fixed time.

7.5 Achievements/Learning

During the project our learning process is increase by tools and technology we used in our project. Following are the major learning achieved by us:

- Android Studio
- Java
- Firebase Real Time Database

7.6 Future Plan

We Aim to make this app more secure and efficient. We want to include more functionalities in our application. The application will have more features like Attendance summary and sending notifications to students if their attendance is short in a particular course. We also have future plan to build a web portal for admin.

References

1. Java. Developers <https://developer.android.com/training/index.html?q=Java>. Cited on Page. [25](#)
2. Attendance Register. Rudra Nivran <https://play.google.com/store/apps/details?id=com.rudra.attendanceRegister&hl=en> Cited on Page. [11](#)
3. Attendance Tracker. PeterMan App <https://play.google.com/store/apps/details?id=peterman.apps.attendance&hl=en> Cited on Page. [11](#)
4. Smart Attendance. Alienart <https://play.google.com/store/apps/details?id=com.alienartsoftwares.smartattendance.free&hl=en> Cited on Page. [11](#)
5. Easy Attendance School College. Operose Technologies <https://play.google.com/store/apps/details?id=com.attendance.school&hl=en> Cited on Page. [11](#)
6. Black box testing. Gurru99 <https://www.guru99.com/black-box-testing.html>. Cited on Page. [26](#)

