



CMR UNIVERSITY

Private University Established in Karnataka State by Act No. 45 of 2013

SCHOOL OF ENGINEERING AND TECHNOLOGY

MAD PROJECT REPORT ON

“ANDROID ATTENDANCE MANAGEMENT SYSTEM”

For the requirement of 6th Semester (4BCS603 – Mobile Application Development)

Submitted By

Name	USN
TINA ALEX L	(18BBTCS135)

Submitted to

Dr. T. Parameswaran

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR University

Off Hennur - Bagalur Main Road,

Near Kempegowda International Airport, Chagalahatti,

Bengaluru, Karnataka-562149

Academic Year - 2021



CMR UNIVERSITY

Private University Established in Karnataka State by Act No. 45 of 2013

SCHOOL OF ENGINEERING AND TECHNOLOGY

Chagalhatti, Bengaluru, Karnataka-562149

Department of Computer Science and Engineering

CERTIFICATE

Certified that the MAD Project work entitled “ANDROID ATTENDANCE MANAGEMENT SYSTEM” carried out by **TINA ALEX L (18BBTCS135)**, bonafide student of **SCHOOL OF ENGINEERING AND TECHNOLOGY**, in partial fulfillment for the award of **BACHELOR OF TECHNOLOGY** in 6th semester Computer Science and Engineering of **CMR UNIVERSITY**, Bengaluru during the year 2021. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the report. The project has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Signature of Guide

.....

Dept. of CSE
SoET, CMRU, Bangalore

Signature of HOD

.....

Dept. of CSE
SoET, CMRU, Bangalore

Signature of Dean

.....

SoET, CMRU, Bangalore

Name of the Examiners:

1

2

Signature with Date:

.....

.....

DECLARATION

I, **TINA ALEX L** bearing USN **18BBTCS135**, student of Bachelor of Technology, Computer Science and Engineering, CMR University, Bengaluru, hereby declare that the project work entitled “**ANDROID ATTENDANCE MANAGEMENT SYSTEM**” submitted by me, for the award of the Bachelor’s degree in Computer Science and Engineering to CMR University is a record of bonafide work carried out independently by me under the supervision and guidance of Dr. T. Parameswaram, Associate Professor Dept. of CSE Department. CMR University.

I further declare that the work reported in this project work has not been submitted and will not be submitted, either in part or in full, for the award of any other degree in this university or any other institute or University.

Place: Bengaluru

(TINA ALEX L)

Date: 24-05-2021

(18BBTCS135)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of this project would be incomplete without the mention of the people who made it possible, without whose constant guidance and encouragement would have made efforts go in vain.

I express my sincere gratitude to my project guide **Dr. T. Parameswaran, Associate Professor Dept. of CSE Department. CMR University.**, without whose constant guidance and support the project would not be successful.

I also express my sincere thanks to my project co guide **Dr. R. Sathyaraj, Assistant Professor, Dept of CSE**, whose constant guidance and support for the successful completion of the project.

I would like to express my thanks to **Dr. Rubini P, Associate Professor and Head**, Department of Computer Science and Engineering, School of Engineering and Technology, CMR University, Bangalore, for his encouragement that motivated me for the successful completion of Project work.

I express my heartfelt sincere gratitude to **Dr. C. Prabhakar Reddy, Dean**, School of Engineering and Technology, CMR University for his support.

(TINA ALEX L)

TABLE OF CONTENTS

CHAPTER NO.	CONTENTS	PAGE NO.
	LIST OF FIGURES	V
	LIST OF TABLES	VI
	ABSTRACT	VII
1	INTRODUCTION	1-6
	1.1 Special Features of Language	2-6
	1.1.1. Front End	2
	1.1.2. XML Elements	2
	1.1.3. Java	2-4
	1.1.4. Back End	4
	1.1.5. Key Capabilities	4-6
2	LITERATURE SURVEY	7-8
	2.1 Existing System	7
	2.2 Disadvantage of Existing System	7
	2.3 Proposed System	7-8
	2.4 Advantage of Proposed System	8
	2.5 Problem Definition and Description	8
3	SOFTWARE AND HARDWARE REQUIREMENTS	9
	3.1 Hardware Requirements	9
	3.2 Software Requirement	9
4	SYSTEM DEVELOPMENT PROCESS	10-18
	4.1 System Analysis	10-13

	4.1.1 Package Selected	10
	4.1.2 Feasibility Study	10-12
	4.1.3 Data Flow Diagram	12-13
	4.2 System Design	14-18
	4.2.1 Architectural Design	14
	4.2.2 Input and output from design	15-16
	4.2.3 Tables	17-18
	4.2.4 ER Diagram	18
5	RESULTS / SNAPSHOTS	19-22
	5.1 Dashboard	19
	5.2 Admin	19
	5.3 Add or Remove Teacher	20
	5.4 Add or Remove Student	20
	5.5 Attendance Record	20
	5.6 Teacher Login	20-21
	5.7 Student Login	21-22
6	CONCLUSION	23
	6.1 Summary of the Report	23
	6.2 Future Possibilities	23
	REFERENCES	24

LIST OF FIGURES

Fig. No.	Name of the Figure	Page No
1.1	Project Overview	01
4.1.1	Level 0 DFD	12
4.1.2	Level 1 DFD	13
4.2.1	Architectural Design	14
4.2.2	Admin login	15
4.2.3	Staff login	15
4.2.4	Student login	16
4.2.5	Admin home page	16
5.1	Dashboard/Homepage	19
5.2.1	Admin login	19
5.2.2	Admin Dashboard	19
5.3	Admin- Add/Remove Teacher	20
5.4	Admin- Add/Remove Student	20
5.5	Admin- Attendance Record	20
5.6.1	Teacher Login	20
5.6.2	Teacher Dashboard	21
5.6.3	Teacher- Take Attendance	21
5.6.4	Teacher- Previous Records	21

5.7.1	Student Login	21
5.7.2	Student Dashboard	22
5.7.3	Student View Attendance	22

LIST OF TABLES

Table. No.	Name of the Table	Page No
4.1	Admin, Staff and Student login	17
4.2	Add/ Remove Staff	17
4.3	Add/Remove Student	17
4.4	Attendance	18

ABSTRACT

The Android Attendance Management System is developed in an Android Platform and with the use of Java programming language. This Attendance Management System is so simple and is designed with a user-friendly interface. The main purpose of this Attendance Management System in Android is to help schools or colleges in keeping track of the attendance of the students within the campus. The Attendance Management System in Android App contains three sections the admin section, Faculty section and the Student section. As you run the Attendance Management System in Android App on your computer or laptop, the emulator will appear then the app will be displayed on the screen. First the module is admin which has the right to add or remove student or faculty and the generation of attendance records. Second the module is faculty/teacher which has right to making daily attendance and generating the report. Third the module is a student who has the right to view their attendance record. Attendance can be taken based on class.

Chapter 1

INTRODUCTION

This project entitled as “ANDROID ATTENDANCE MANAGEMENT SYSTEM USING FIREBASE DATABASE”. The main objective of this project is to carry out some of the emerging technologies like mobile computing, Information and Communication Technology and advances in behavioral science studies to enhance and enriched the current educational system scenario, since the educational technology in India has been get modernizing in the recent past years due to the development and penetration of Information and Technology. The mobile based technology has been chosen for this study as well as for the project. The methodology of this work is to developed an android based mobile application attendance management system where attendance can be recorded via mobile devices using Google’s “FIREBASE” database a real-time server. This developed software stores, retrieves and deliver the information about the student information such as present or absent through mobile device in the provided server database. The final result of this project is very useful for the educational organizations to keep, track and maintain the database of students.



Fig 1.1 Project Overview

1.1 SPECIAL FEATURES OF LANGUAGE

1.1.1 Front End

XML is the abbreviation for Extensible Markup Language and is an established data exchange format. An XML document consists of elements, each element has a start tag, content and an end tag. An XML document must have exactly one root element (i.e., one tag which encloses the remaining tags). XML differentiates between capital and non-capital letters.

An XML file must be well-formed. This means that it must apply to the following conditions:

- An XML document always starts with a prolog (see below for an explanation of what a prolog is)
- Every opening tag has a closing tag.
- All tags are completely nested.

A valid XML file is well-formed and must contain a link to an XML schema and be valid according to that schema.

1.1.2 XML ELEMENTS

An XML document always starts with a prolog which describes the XML file. This prolog can be minimal, e.g. `<?xml version="1.0"?>`. It can also contain other information, e.g. the encoding `<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>`.

A tag which does not enclose any content is known as an "empty tag", for example `<flag/>`.

Comments in XML are defined as: `<!-- COMMENT -->`.

1.1.3 JAVA

The Java programming language contains several methods for processing and writing XML.

Older Java versions supported only the DOM API (Document Object Model) and the SAX (Simple API for XML) API.

In DOM you access the XML document over an object tree. DOM can be used to read and write XML files.

SAX (Simple API for XML) is a Java API for sequential reading of XML files. SAX can only read XML documents. SAX provides an event driven XML Processing following the Push-Parsing model. In this model you register listeners in the form of Handlers to the Parser. These are notified through call-back methods

Both DOM and Sax are older APIs and I recommend not using them anymore.

Stax (Streaming API for XML) is an API for reading and writing XML Documents. It was introduced in Java 6.0 and is considered superior to SAX and DOM.

Java Architecture for XML Binding (JAXB) is a Java standard that allows to convert Java objects to XML and vice versa. JAXB defines a programmer API for reading and writing Java objects to from XML documents. It also defines a service provider which allows the selection of the JAXB implementation. JAXB applies a lot of defaults thus making reading and writing of XML via Java very easy.

Android operating system manages the starting and shutting down the application process, whenever required. This means each android application runs in isolation with other, but they can certainly request access to hardware and other system resources. If you are familiar with mobile application development, may be in J2ME, then you may know about permissions. So when an android application is installed or started, it requests necessary permission required to connect the internet, phone book and other system resources. The user explicitly provides grant these permissions, or it may deny.

All these permissions are defined in the manifest file of Android application. Unlike Java Manifest file, Android manifest is an XML file, which lists all the components of apps, and settings for those components.

Four major components of Android application development are Activities, Services, Content Providers and Broadcast Receivers. Activity is most common of them, as it represents a single screen in Android Application. For example, in an Android Game, you can have multiple screens for login, high score, instructions and game screen. Each of this screen represents different Activities inside your app.

Similar to Java, a good thing about Android is that it manages certain task on behalf of the developer, one of them is creating activity object. Activities are managed by System, when you want to start an activity, you call `startActivity ()` method which takes an Intent object. In response to this call, System can either create new activity object or reuse an existing one.

Just like Garbage collection in Java, manages a critical task or reclaiming memory, Android manages the starting, stopping, creating and destroying of apps by themselves. You may think it's restrictive, but it's not. Android provides life-cycle events, which you can override to interact with this process.

1.1.4 BACKEND

Firebase – NoSQL Database

Store and sync data with our NoSQL cloud database. Data is synced across all clients in realtime, and remains available when your app goes offline.

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

1.1.5 Key Capabilities

Realtime - Instead of typical HTTP requests, the Firebase Realtime Database uses data synchronization—every time data changes, any connected device receives that update within milliseconds. Provide collaborative and immersive experiences without thinking about networking code.

Offline - Firebase apps remain responsive even when offline because the Firebase Realtime Database SDK persists your data to disk. Once connectivity is reestablished, the client device receives any changes it missed, synchronizing it with the current server state.

Accessible from Client Devices - The Firebase Realtime Database can be accessed directly from a mobile device or web browser; there's no need for an application server. Security and data validation are available through the Firebase Realtime Database Security Rules, expression-based rules that are executed when data is read or written.

Scale across multiple databases - With Firebase Realtime Database on the Blaze pricing plan, you can support your app's data needs at scale by splitting your data across multiple database instances in the same Firebase project. Streamline authentication with Firebase Authentication on your project and authenticate users across your database instances. Control access to the data in each database with custom Firebase Realtime Database Rules for each database instance.

How does it work?

The Firebase Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, real-time events continue to fire, giving the end user a responsive experience. When the device regains connection, the Realtime Database synchronizes the local data changes with the remote updates that occurred while the client was offline, merging any conflicts automatically.

The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules, to define how your data should be structured and when data can be read from or written to. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it.

The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database. The Realtime Database API is designed to only allow operations that can be executed quickly. This enables you to build a great realtime experience that can serve millions of users without compromising on responsiveness. Because of

this, it is important to think about how users need to access your data and then structure it accordingly.

Chapter 2

LITERATURE SURVEY

2.1. EXISTING SYSTEM

- Manually calculated maintenance of data.
- Suddenly searching for any type of related data is very difficult as well as it creates mess over there.
- It not provides the accurate data of attendance of students in the percentage calculation form.
- Portability.

2.2. DISADVANTAGE OF EXISTING SYSTEM

- High burden,
- Losing of data problem can be occurred easily.

2.3 PROPOSED SYSTEM

Our system is a user-friendly, flexible and full featured android attendance management tool, which allows controlling student attendance by automatically timekeeping and attendance tracking. Android attendance management system primarily focuses on building an efficient and user-friendly Android mobile application for an Attendance Monitoring using the Advanced database called Google's firebase database. The application will be installed on the professor's phone as well as student's phone which runs android OS. It intends to provide an interface to the professor who will require minimal details to input for marking of attendance of a particular class of students. Apart from that, the application would support strong user authentication and quick transmission of data.

- Security of data.
- Ensure data accuracies.
- Minima manual data entry.
- Greater efficiency.
- Better service.

- User friendliness and interactive.
- Minimum time required

2.4. ADVANTAGE OF PROPOSED SYSTEM

1. Less time consuming,
2. Access details through the system.

2.5. PROBLEM DEFINITION AND DESCRIPTION

This project entitled as “ANDROID ATTENDANCE MANAGEMENT SYSTEM USING FIREBASE DATABASE”. The aim of the project is to maintain the student and staff details. Attendance Management System is software developed for daily student attendance in schools, colleges and institutes. It facilitates to access the attendance information of a student in a class. This system will also help in evaluating attendance eligibility criteria of a student. By just a click on the mouse, the system will be able to produce the students' attendance report thus reducing the need for manual labour which is prone to human errors and time consuming. This application is built for automating the processing of attendance. It also enhances the speed of performing attendance task easily. The student and staff have unique user login id and password available. The student can only view the attendance record on weekly, monthly, and whole semester basis. The staff can view as well as modify the attendance record. Printing facility for attendance record is available for both students and staff.

Chapter 3

HARDWARE AND SOFTWARE REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

RAM	:	4 GB
MONITOR	:	LCD COLOR
HARD DISK	:	40 GB
KEYBOARD	:	Multimedia Keyboard
MOUSE	:	Optical Buttons

3.2 SOFTWARE REQUIREMENTS

OPERATING SYSTEM	:	Windows 10
FRONT END	:	XML & JAVA
BACK END	:	Firebase Database

Chapter 4

SYSTEM DEVELOPMENT PROCESS

4.1 System Analysis:

4.1.1 Package Selected

Front End: Designing - XML, Development - JAVA.

Back End: Google's Firebase Real Time Database.

Operating System: Windows Family.

4.1.2 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features:

TECHNICAL FEASIBILITY

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the

investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So, there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

4.1.3 DATA FLOW DIAGRAM

Data Flow Diagram (DFD) is one of the best ways if documenting the entire functionality of the system. For any system, which will have some data inflows, processing and data outflows from the system can be documented or represented effectively by the means of the data flow diagram. It is a diagrammatic representation of the system.

LEVEL 0 DFD

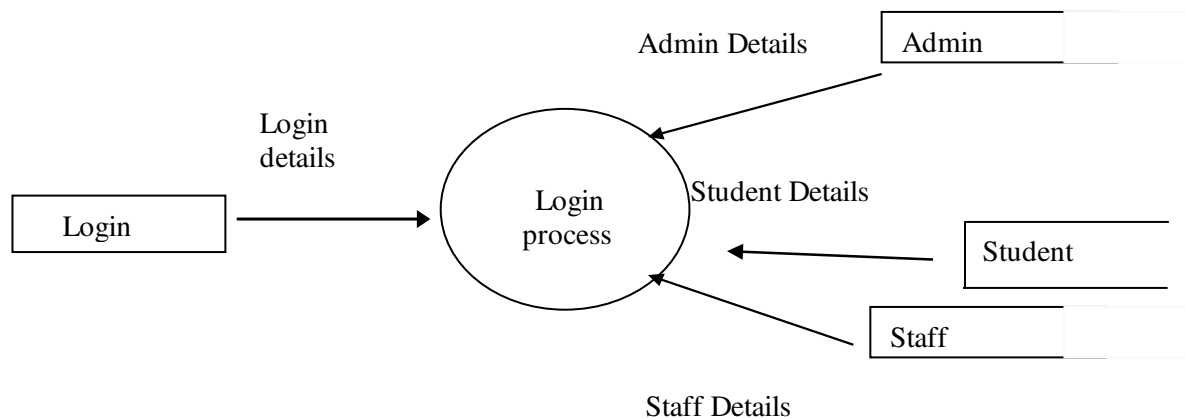


Fig 4.1.1. Level 0 DFD

LEVEL 1 DFD

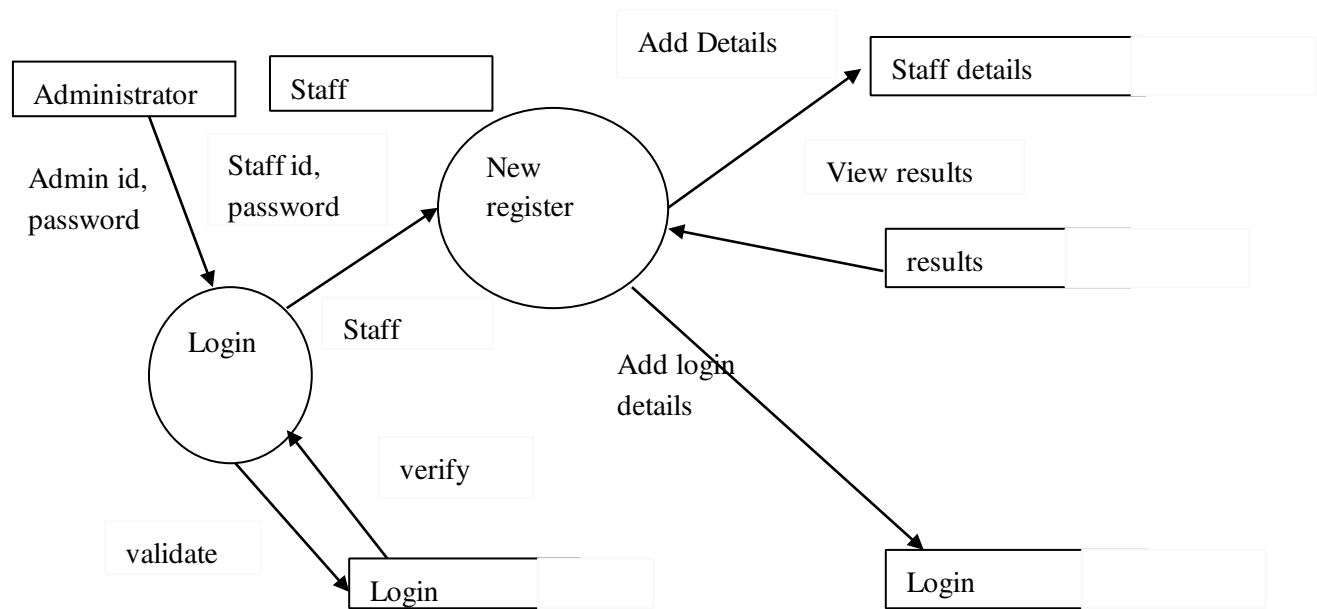


Fig 4.1.2. Level 1 DFD

4.2 SYSTEM DESIGN:

4.2.1 ARCHITECTURAL DESIGN

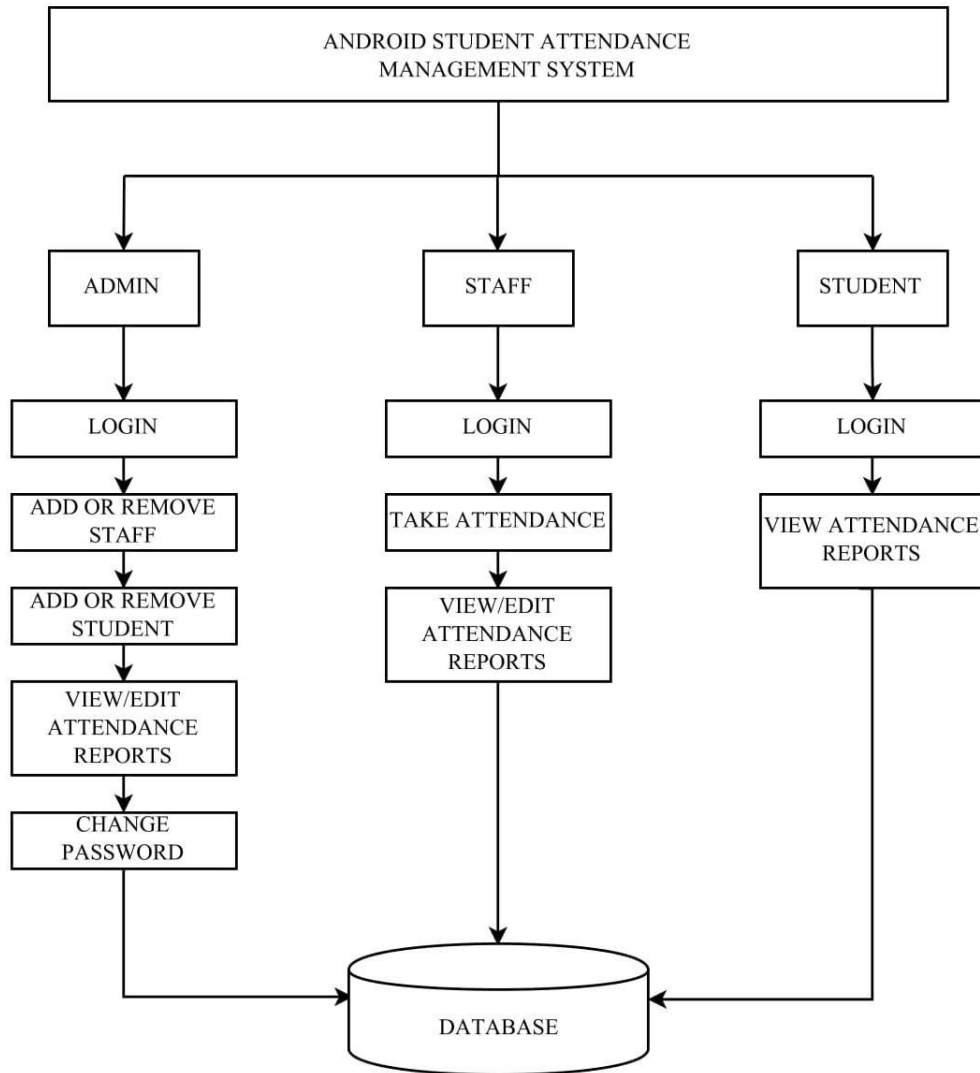
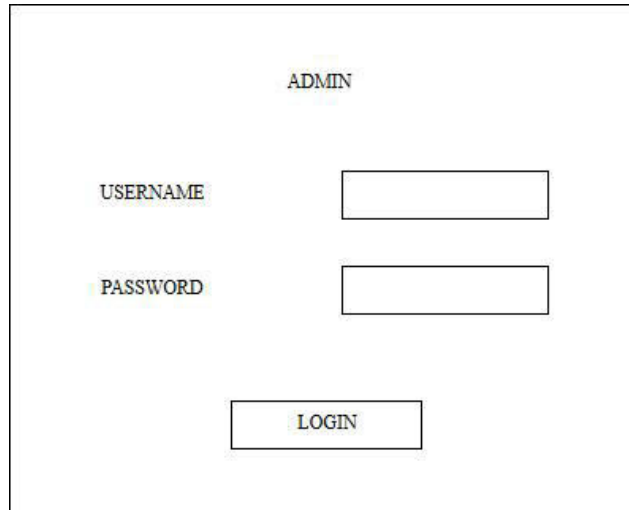


Fig 4.2.1. Architectural Design

4.2.2 INPUT AND OUTPUT FORM DESIGN

ADMIN LOGIN

Admin login helps to enter into the admin area where the admin can add/remove staff and student's details.



```

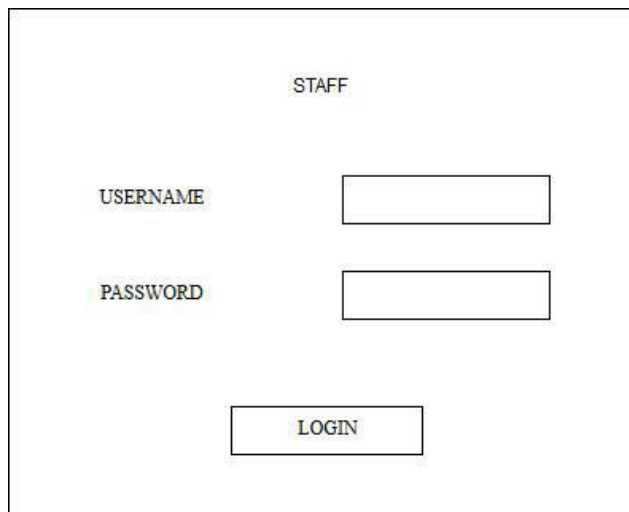
graph TD
    ADMIN[ADMIN]
    USERNAME[USERNAME] --- U_INPUT[ ]
    PASSWORD[PASSWORD] --- P_INPUT[ ]
    U_INPUT --- LOGIN[LOGIN]
    P_INPUT --- LOGIN
    
```

The diagram shows a rectangular box representing the Admin Login form. At the top center is the label 'ADMIN'. Below it, on the left, are the labels 'USERNAME' and 'PASSWORD'. To the right of 'USERNAME' is a rectangular input field. To the right of 'PASSWORD' is another rectangular input field. At the bottom center of the box is a rectangular button labeled 'LOGIN'.

Fig 4.2.2. Admin login

STAFF LOGIN

Through the staff login the staff can take attendance for various classes and students. They can also view the report of the student's attendance reports.



```

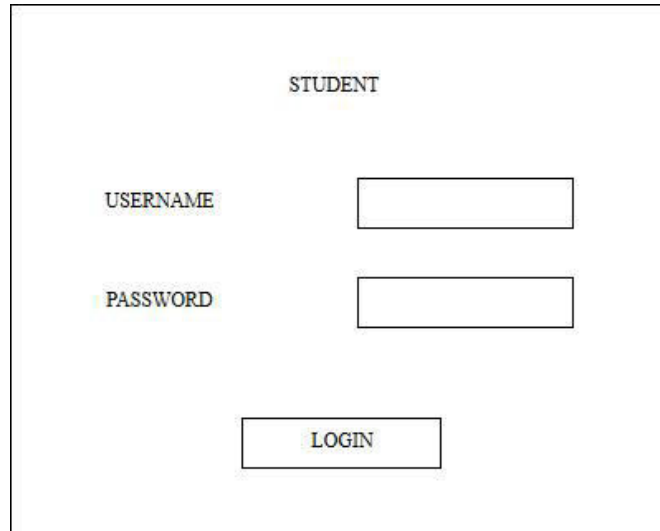
graph TD
    STAFF[STAFF]
    USERNAME[USERNAME] --- U_INPUT[ ]
    PASSWORD[PASSWORD] --- P_INPUT[ ]
    U_INPUT --- LOGIN[LOGIN]
    P_INPUT --- LOGIN
    
```

The diagram shows a rectangular box representing the Staff Login form. At the top center is the label 'STAFF'. Below it, on the left, are the labels 'USERNAME' and 'PASSWORD'. To the right of 'USERNAME' is a rectangular input field. To the right of 'PASSWORD' is another rectangular input field. At the bottom center of the box is a rectangular button labeled 'LOGIN'.

Fig 4.2.3. Staff login

STUDENT LOGIN

Students can view their attendance report details according to the date, class and period.

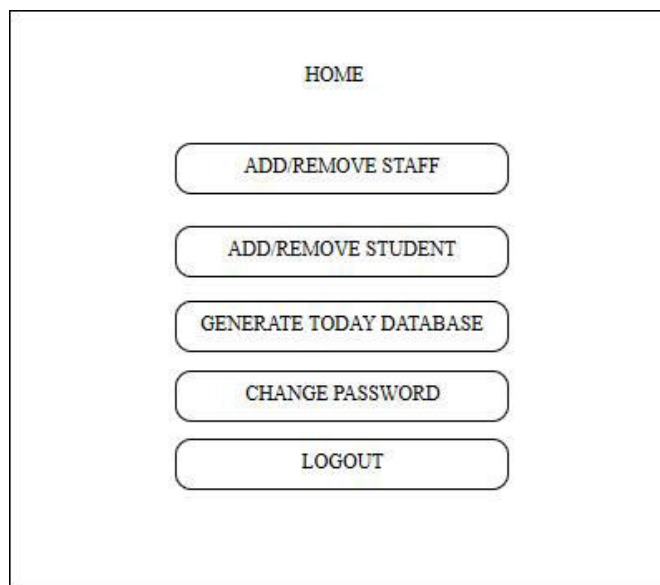


The figure shows a student login form. At the top, the word "STUDENT" is centered. Below it, there are two input fields: one for "USERNAME" and one for "PASSWORD". At the bottom, there is a "LOGIN" button.

Fig 4.2.4. Student login

ADMIN HOME PAGE

This is the admin home page where the admin can add/remove staff and students, generate databases for everyday and can change the password for him.



The figure shows the admin home page. At the top, the word "HOME" is centered. Below it, there are five buttons stacked vertically: "ADD/REMOVE STAFF", "ADD/REMOVE STUDENT", "GENERATE TODAY DATABASE", "CHANGE PASSWORD", and "LOGOUT".

Fig 4.2.5. Admin home page

4.2.3 TABLES

Table Design

Table: Admin, Staff, Student Login

Primary: Username

Field Name	Data Type	Constrains	Description
Username	Varchar(20)	Not Null	Primary Key
Password	Varchar(20)	Not Null	

Table 4.1 Admin, Staff and Student login

Table: Add/Remove Staff

Primary: Staff id

Field Name	Data Type	Constrains	Description
Staff Name	Text	Not Null	
Staff ID	Varchar (20)	Not Null	Primary key
Subject	NVarchar (50)	Not Null	
Class	NVarchar (50)	Not Null	
Password	NVarchar (50)	Not Null	

Table 4.2. Add/ Remove Staff

Table: Add/Remove Student

Primary: Student id

Field Name	Data Type	Constrains	Description
Student Name	Text	Not Null	
Student ID	Varchar (20)	Not Null	Primary key
Class	NVarchar (50)	Not Null	
Password	NVarchar (50)	Not Null	

Table 4.3 Add/Remove Student

Table: Attendance

Primary: Staff id

Field name	Data Type	Constrains	Description
Staff ID	Varchar (20)	Not Null	
Class	NVarchar (50)	Not Null	
Period No	NVarchar (50)	Not Null	

Table 4.4 Attendance

4.2.4 ER Diagram:

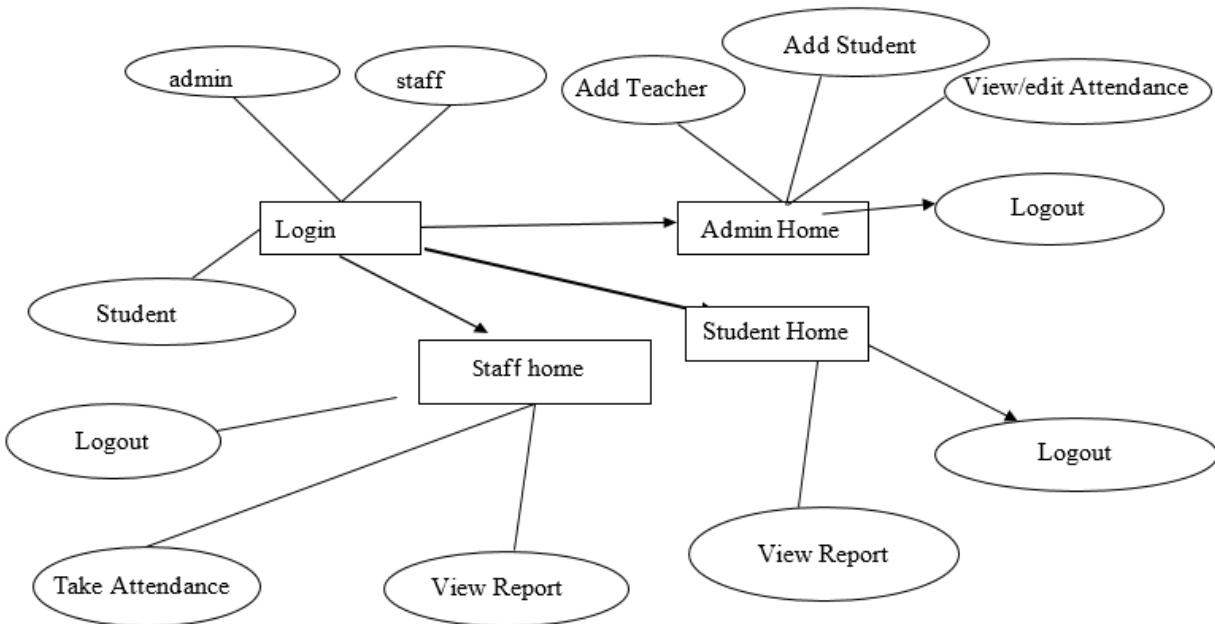


Fig 4.2.6. ER Diagram

Chapter 5

RESULTS / SNAPSHOTS

5.1. Dashboard

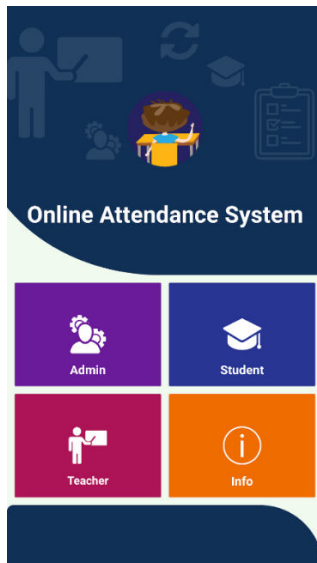


Fig 5.1. Dashboard/Homepage

5.2. Admin

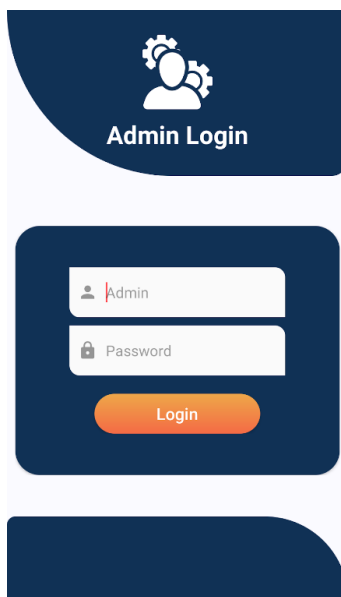


Fig 5.2.1. Admin login

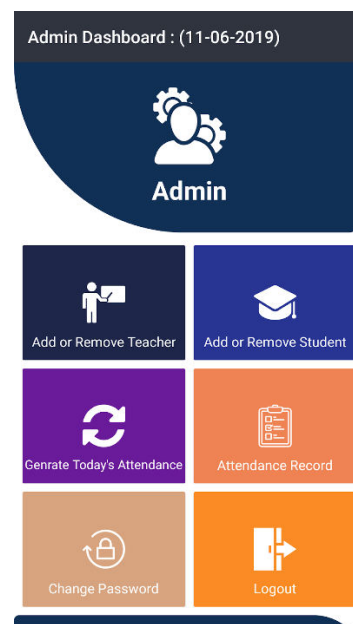


Fig 5.2.2. Admin Dashboard

5.3. Add or Remove Teacher

Fig 5.3. Admin- Add/ Remove Teacher

5.4. Add or Remove Student

Fig 5.4. Admin- Add/ Remove Student

5.5 Attendance Record

SID	Status	period
mukesh	P	p1
mukesh	A	p2
mukesh	A	p3
mukesh	A	p6
mukesh	A	p8
raju123	A	p1
raju123	P	p2
raju123	P	p3
raju123	A	p6
raju123	P	p8

Fig 5.5. Admin- Attendance Record

5.6. Teacher Login

Fig 5.6.1. Teacher login

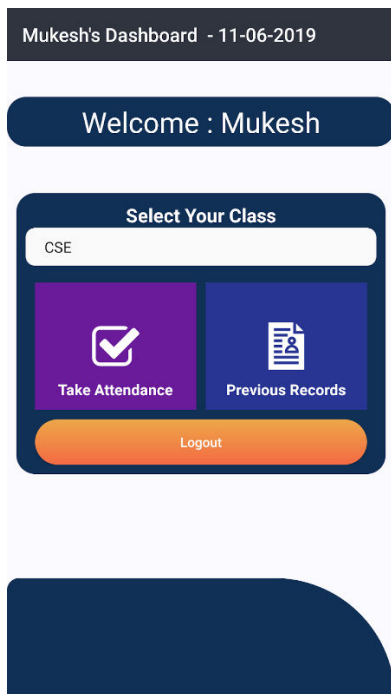


Fig 5.6.2. Teacher Dashboard

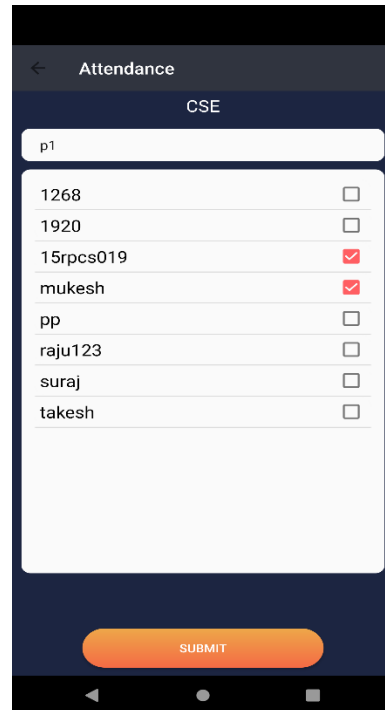


Fig 5.6.3. Teacher- Take Attendance

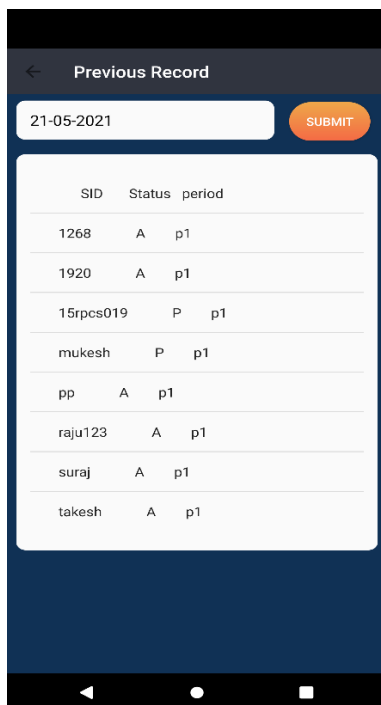


Fig 5.6.4. Teacher- Previous Record

5.7 Student Login

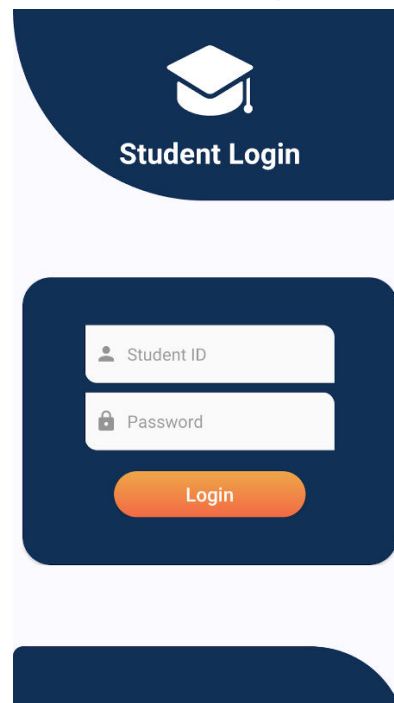


Fig 5.7.1. Student login

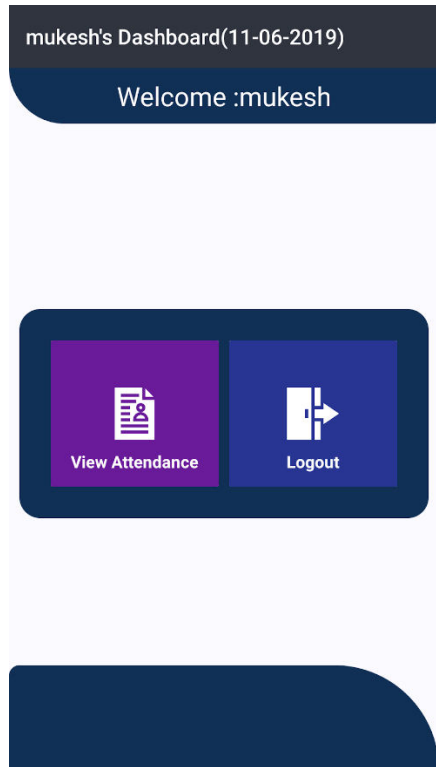


Fig 5.7.2 Student Dashboard

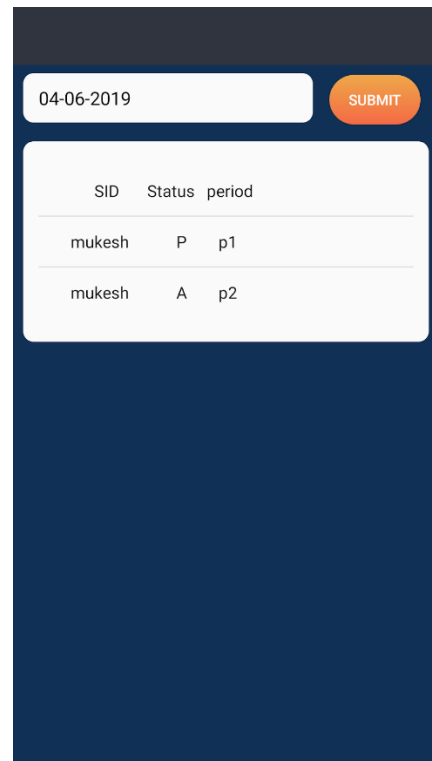


Fig 5.7.3 Student View Attendance

Chapter 6

CONCLUSION

6.1 Summary of the Project

This whole work is to access the details about the student attendance information and generate a final report. This project “Android Student Attendance Management System” is a collection of static and dynamic web based or mobile application-based pages. This project provides an offer to the user to enter the data through their respective registration forms. It is very helpful for the teachers and admin to keep and maintain the information about the students easily. In future this work can be expanded to store the internal marks, semester marks, college events and college placement activities of the students to get minimized all stuffs at one place in a systematic way to import and export the data through the admin and authorized persons whenever it will be needed in future by the educational organization.

6.2 Future Possibilities

When the database in offline mode we can't access even login. So, in future this can be avoided by using cloud storages like Firebase cloud, AWS.

REFERENCES

Book References

- "Mobile Phone Based Attendance System", by Shraddha S.Chawhan¹, Mangesh P. Girhale², Gunjan Mankar³, IOSR Journal of Computer Engineering.
- "Component-Based Software Engineering" by Ian Gorton, George T. Heineman, Ivica Crnković, Heinz W. Schmidt, Judith A. Stafford, Clemens Szyperski.
- "A Proposed Android Based Mobile Application to Monitor Works at Remote Sites", by S. Sivasubramanian¹.

Website Reference

1. www.android.com
2. firebase.google.com
3. www.androidhive.info