# CAPSTONE REPORT

on

# AUTOMATED ATTENDANCE SYSTEM

Submitted By

**Vasu Dev Arora** (11401786)

**Simran Shaw** (11407238)

**Angad Singh Dhodi** (11400903)

**Bittu Baiju** (11402923)

**Project Group Number: CSERGC0042**

**Course Code : CSE445**

Under the Guidance of

**Mr. Deepak Prashar**

**School of Computer Science and Engineering**

## TOPIC APPROVAL PERFORMA

School of Computer Science and Engineering

**Program :** 1202D::B.Tech -M.Tech (Dual Degree) - CSE

**COURSE CODE :** CSE445 **REGULAR/BACKLOG :** Regular **GROUP NUMBER :** CSERGC0042

**Supervisor Name :** Deepak Prashar **UID :** 13897 **Designation :** Assistant Professor

**Qualification :** ________________________ **Research Experience :** ________________________

| SR.NO. | NAME OF STUDENT | REGISTRATION NO | BATCH | SECTION | CONTACT NUMBER |
|---|---|---|---|---|---|
| 1 | Simran Shaw | 11407238 | 2014 | K1407 | 09873950493 |
| 2 | Bittu Baiju | 11402923 | 2014 | K1406 | 9872383630 |
| 3 | Vasu Dev Arora | 11401786 | 2014 | K1408 | 9041415639 |
| 4 | Angad Singh Dhodi | 11400903 | 2014 | K1405 | 9646207754 |

**SPECIALIZATION AREA :** Networking and Security **Supervisor Signature:** ___________________

**PROPOSED TOPIC :** Automated Attendance Marking System

| Qualitative Assessment of Proposed Topic by PAC | | |
|---|---|---|
| **Sr.No.** | **Parameter** | **Rating (out of 10)** |
| 1 | Project Novelty: Potential of the project to create new knowledge | 7.60 |
| 2 | Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students. | 7.80 |
| 3 | Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program. | 8.00 |
| 4 | Project Supervision: Project supervisor’s is technically competent to guide students, resolve any issues, and impart necessary skills. | 8.20 |
| 5 | Social Applicability: Project work intends to solve a practical problem. | 7.80 |
| 6 | Future Scope: Project has potential to become basis of future research work, publication or patent. | 7.40 |

| PAC Committee Members | | |
|---|---|---|
| PAC Member 1 Name: Prateek Agrawal | UID: 13714 | Recommended (Y/N): NA |
| PAC Member 2 Name: Deepak Prashar | UID: 13897 | Recommended (Y/N): Yes |
| PAC Member 3 Name: Pushpendra Kumar Pateriya | UID: 14623 | Recommended (Y/N): Yes |
| PAC Member 4 Name: Balraj Singh | UID: 13075 | Recommended (Y/N): Yes |
| PAC Member 5 Name: Anupinder Singh | UID: 19385 | Recommended (Y/N): Yes |
| DAA Nominee Name: Kuldeep Kumar Kushwaha | UID: 17118 | Recommended (Y/N): Yes |

**Final Topic Approved by PAC:** **Automated Attendance Marking System**

**Overall Remarks:** Approved

**PAC CHAIRPERSON Name:** 11024::Amandeep Nagpal **Approval Date:** 15 Mar 2018

# DECLARATION

We hereby declare that the project work entitled ("Automated Attendance System") is an authentic record of our own work carried out as requirements of Capstone Project for the award of B. Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Mr. Deepak Prashar Sir, during January to April 2018. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Project Group Number: CSERGC0042

Name of Student 1: Vasu Dev Arora

Registration Number: 11401786

Name of Student 2: Simran Shaw

Registration Number: 11407238

Name of Student 3: Angad Singh Dhodi

Registration Number: 11400903

Name of Student 3: Bittu Baiju

Registration Number: 11402923

Signature of (Vasu Dev Arora)

Date:

Signature of (Simran Shaw)

Date:

Signature of (Angad Singh Dhodi)

Date:

Signature of (Bittu Baiju)

Date:

# CERTIFICATE

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Capstone Project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfillment of the conditions for the award of B. Tech degree in Computer Science and Engineering, from Lovely Professional University, Phagwara.

**Signature and Name of the Mentor**

**Designation:** Head of Department of Networking & Security

**School of Computer Science and Engineering,**
Lovely Professional University, Phagwara,
Punjab.

Date:

# ACKNOWLEDGEMENT

We humbly take this opportunity to present our votes of thanks to all those guideposts who really acted as lightening pillars to lighten our way throughout this project that has led to successful and satisfactory completion of this study.

We would like to express our deep sense of gratitude to Mr. Deepak Prashar Sir for his valuable support, encouragement, supervision and useful suggestions throughout the semester and for the completion of this project. His moral support and continuous guidance enabled us to complete our work successfully. We are very grateful for the cooperation and constant encouragement regarding to the whole process that we are able to explore our knowledge in Lovely Professional University.

We are also thankful to our parents and the various applications whose published work has been consulted and cited in our capstone and even I would like also like to express my thanks to my team members who has took a Jubilant part of individual work and who have been instrumental in creating proper, healthy and conductive environment including new and fresh innovative ideas for us to prepare the in a time bound framework which leaded to successful output.

# TABLE OF CONTENTS

PAC Form

Declaration

Certificate

Acknowledgement

**1. INTRODUCTION 1**
**2. PROFILE OF THE PROBLEM 2**
**3. EXISTING SYSTEM 3-5**
**3.1. INTRODUCTION**
**3.2. EXISTING SOFTWARE**
**4. PROBLEM ANALYSIS 6-7**
**4.1. PRODUCT DEFINITION**
**4.2. FEASIBILITY ANALYSIS**
**4.3. TECHNICAL FEASIBILITY**
**5. SOFTWARE REQUIREMENT ANALYSIS 8-10**
**5.1. FUNTIONAL REQUIREMENTS**
**5.2. NON-FUNCTIONAL REQUIREMENTS**
**6. DESIGN 11-23**
**6.1. SYSTEM DESIGN**
**6.2. DETAILED DESIGN**
**6.2.1. DATA FLOW DIAGRAMS**
**6.3. FLOW CHARTS**
**7. TESTING 24-27**
**7.1. STRUCTURAL TESTING**
**7.2. FUNTIONAL TESTING**
**7.3. TESTING LEVELS**
**7.4. TEST RESULTS**
**8. IMPLEMENTATION 28-29**
**8.1. PROJECT IMPLEMENTATION**
**8.2. CODING**
**8.3. POST IMPLEMENTATION MAINTAINENCE**

9. PROJECT LEGACY 30-32
9.1. CURRENT STATUS OF THE PROJECT
9.2. REMAINING AREAS OF CONCERN
9.3. TECHNICAL AND MANAGERIAL LESSONS LEARNT
10. USER MANUAL 33-38
10.1. APPLICATION
10.2. SOFTWARE FEATURES
10.3. FREQUENTLY ASKED QUESTIONS
11. SOURCE CODE 39-76

# 1.INTRODUCTION

In the era of automation, by replacing manual attendance and payroll system automated attendance system has captured a smart position. An automated attendance system includes almost all biometric attendance systems, smartcard attendance systems, swipe card attendance systems, web based attendance systems, etc. These automated attendance systems are now-a-days used in various organizations from police stations to banking to world business**.**

Usually the attendance **is** recorded manually by the tutor and therefore is prone to personal errors. There arises a need for a more efficient and effective method of solving this problem. A technology that can solve this problem and even do more is the RFID technology. Radio Frequency Identification combines radio frequency and microchip technologies to create a smart system that can be used to identify, monitor, secure and do object inventory. RFID is an automated identification and data collection technology, that ensures more accurate and timely data entry. RFID systems use tiny chips called tags that contain and transmit some piece of identifying information to an RFID reader, a device that in turn can interface with computers.

The main goal of the implementation of the attendance system being automated is to reduce the effort and input of the teacher. Using this system the teacher will directly get a compiled output of the result without any physical intervention. It also contributes to increasing the overall efficiency of the attendance marking process. The attendance system is created in such a way that it prevents any marking of unwanted attendance and also ensures complete verification and integrity of the same. It combines a mobile interface along with hardware implementation.

This combination also makes it capable to handle large strength of students while having extremely low overhead and operational costs.

# 2. PROFILE OF THE PROBLEM

Generally, regular attendance is done manually by the tutor which may lead to marking of false attendance (marking of students who are not present in the class) along with the time that is utilized in marking the attendance. There is also an input required by the tutor on his/her behalf to ensure that the attendance marking process is done without introducing errors of any sort.

By introducing an automatic system to mark the attendance it helps in saving the time of the teacher by directly introducing the students with an interface to mark their own attendance individually. It also ensures that the marking process is error-free. Usually, the Automated Attendance Systems that are being used focus mainly on the hardware side. This project makes use of a software application which complements the hardware part of the system. The software adds features like Integrity, Authentication and Availability to ensure that the marking process is smooth and streamlined.

# 3. EXISTING SYSTEM

## 3.1 INTRODUCTION

The Attendance Systems that are being traditionally used require an input from the teacher itself. These systems require the teacher to mark the attendance of the students individually which increases the time required to complete the attendance marking process.

Attendance is usually taken in two forms :-

1. Online
2. Offline

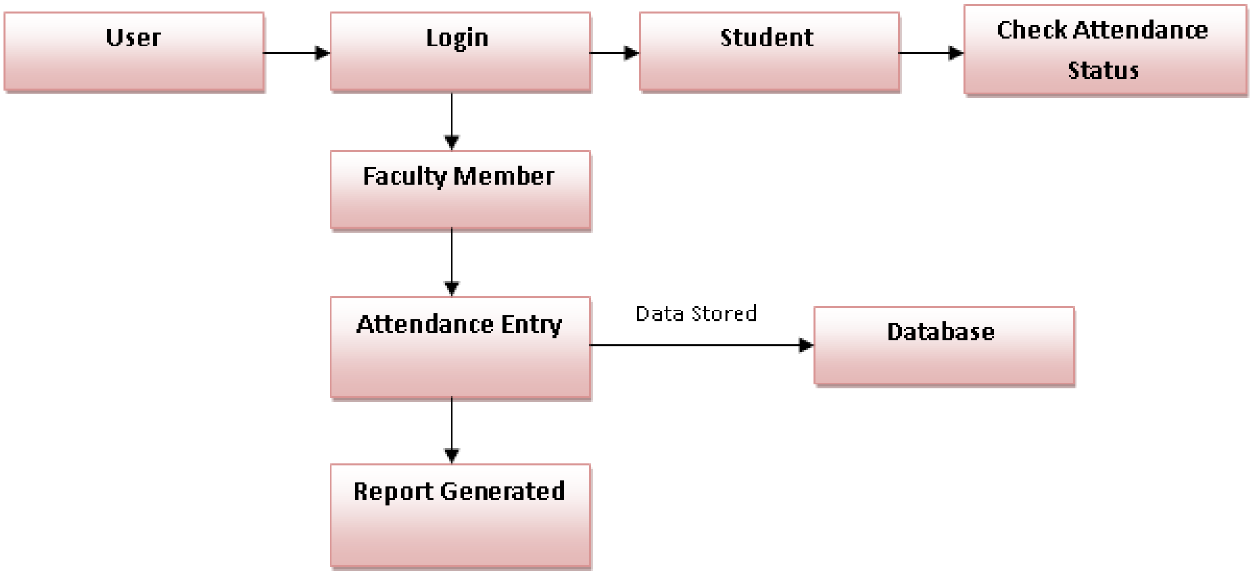
Offline Attendance is marked on paper. The attendance of the whole session is usually stored on a register using which the reports are generated. There are several issues with this kind of approach which are :-

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

Due to these several shortcomings the offline mode is not feasible for general use hence the Online Mode came into existence.

Online Attendance System enabled the whole process to go digital by storing all the data on remote servers which compute and calculate the attendance summary on their own. This removed the need for pen and paper by allowing the tutors to mark attendance using their computers or other smart devices. These systems make use of a Web Browser based interface to mark the attendance of the students and also enable mobility by giving the tutor and the student's access to their summary remotely. This also ensures protection against an data loss and also reduces the resources being utilized. During the marking phase there might be several errors that could occur such as marking of false attendance and sometimes attendance may be skipped accidently by the tutor. This contributes to a large wastage of time. The several stages involved in the attendance marking process are :-

- Accessing the Interface :- The tutor has to login to the online browser based interface using the credentials that are provided by the respective organization. This interface a basic list of the respective sections under the tutor.
- Marking The Attendance :- When the tutor selects a section he/she is then display with a list of all the students in that section and using this the attendance is marked.
- Storage and Synchronization :- All the data regarding the attendance is stored on a SQL based database on a remote server which is directly connected with the web interface.
- Summarized Results :- Once the attendance is marked the summary is displayed to the tutor on the same interface and the same is reflected on the students behalf.

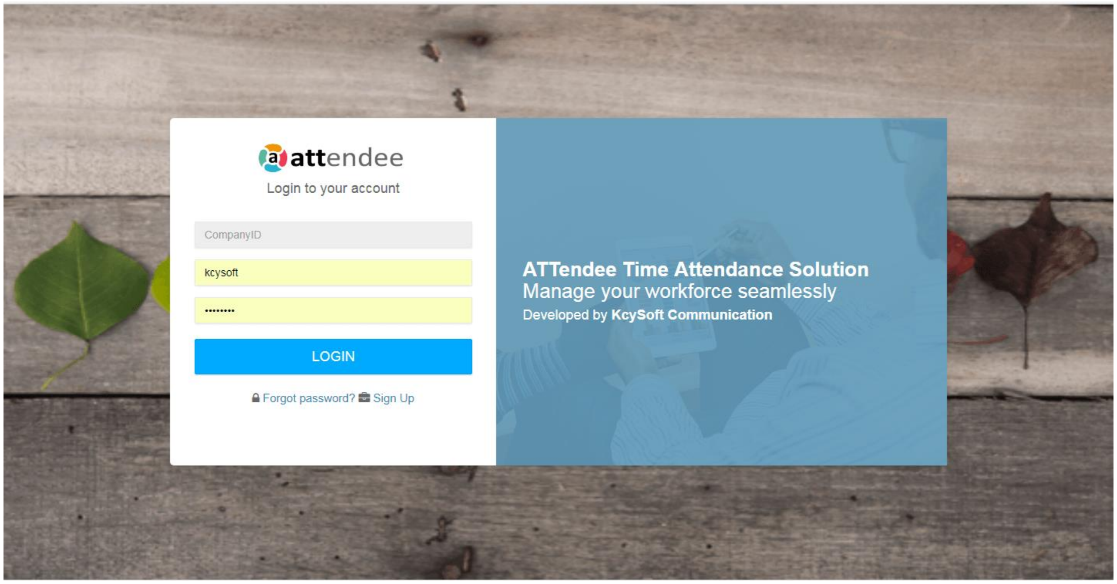


## 3.2 Existing Software

There are many existing software’s that are used to manage the attendance marking process which involve usage of all the above features. These software’s also sometimes make use of hardware technologies such as RFID, Biometric, Retina Scanners etc.  The sample software being used are given below.

- Biometric Attendance Solution

  With the use of an ATTendee biometric solution, students can marks their attendance  with the touch of a finger on biometric scanner. It also uses a web based interface to show the summarized result of the attendance.



attendee

Login to your account

CompanyID

kcysoft

••••••••

ATTendee Time Attendance Solution

Manage your workforce seamlessly

Developed by **KcySoft Communication**

- StratusTime

  Stratustime offers everything small businesses need in a time and attendance system. The cloud-based solution has a comprehensive assortment of time-tracking options, with employees able to clock in and out via traditional time clocks, web browsers, mobile devices, and telephones. The system also manages paid time off, generates employee schedules, monitors overtime hours, integrates with many payroll services and features a variety of mobile options.

# 4. PROBLEM ANALYSIS

In the last phase we discussed what was already available, in this phase we shall discuss what our project has to offer.

## 4.1 PRODUCT DEFINITION

The objective of this project is to improve the overall efficiency of the attendance marking process. This project empowers the teachers by automating the overall process of attendance marking by making use of several technologies together such as RFID, Raspberry Pi, Android Application to ensure that procedure is error-free and quick. As a part of the project we have developed an android application which is directly interacting with the hardware modules which are directly interacting with the user. This project allows the students to mark their respective attendance by using their own smartphones and hence removing any input from the teacher. The teacher is provided with the attendance details automatically and he/she can modify the attendance records if needed.

This project makes use of RFID which is extremely reliable and provides accurate results. This enhances the overall security related features when integrated with the mobile application.

## 4.2 FEASIBILITY ANALYSIS

The scope of this project is extremely vast but due to time constraint and limited resources we have incorporated a solution that is both cost-efficient and still provides high performance and accurate results. We have stated below the various problems in-hand and our solution for the same.

### 4.2.1 TECHNICAL FEASIBILITY

The major issues faced in terms of technicality are mentioned below.

- Transferring of RFID Data

- The primary issue we faced was facilitating the communication between the RFID module and the Android Application. This interfacing is necessary to ensure that authentication and integrity is maintained in the attendance marking process.

- Technologies Used

  For the application we made use of an Android App from which both the students as well as the tutors operate their accounts to view and mark their attendance. For storing the data an SQLite database is used. Also we have made use of a Raspberry Pi and hence made use of Python scripting to perform the interfacing. For retrieval of RFID data we made use of JSON (Javascript Object Notation) which is used to show the attendance data.
- Products

  Our Third problem was on how we were going to implement our project. We decided amongst two entities amongst an Android App or a Web based application. The Android App was a viable choice as it provided mobility and independence to the users.

# 5. SOFTWARE REQUIREMENT ANALYSIS

Now that we have arrived to the conclusion that the project is feasible. It is time to list all the requirements and functionalities that our project will perform.

## 5.1 Functional Requirements

### 5.1.1 Adding a New User

Description: The user will register on the Android Application by providing personal details so that he/she can perform activities such as view and mark attendance.

Input: Details of the User

Output: Proceed to Login Interface

### 5.1.2 Login

Description: This will verify whether the user is a registered user or not and then he can proceed to the further authentication modules.

Input: Username and Password

Output: Proceed to Fingerprint Authentication Module

### 5.1.2 Fingerprint Authentication

Description: The user will provide his/her finger impression in order to confirm his authentication and then he/she will proceed to the Attendance Module.

Input: Finger Impression

Output: Access to Attendance Module

### 5.1.3 Mark Attendance

Description: This will allow the user to view the details of the ongoing lecture and on proper validation he/she can mark their respective attendance.

Input: Mark attendance as present

Output: Confirmation Message

### 5.1.4 OTP Authentication

Description: For reflecting the attendance in the database, the student has to authenticate himself/herself by providing the One-Time Password (OTP) which is sent to the registered mobile number of the user.

Input: OTP sent to the Registered Mobile Number

Output: Confirmation Message

### 5.1.5 View Attendance

Description: The user gets a detailed report of their attendance history.

Input: Select “View Attendance” Button

Output: View Detailed Attendance Report

## 5.2 Non-Functional Requirements

### 5.2.1Security

Only authorized users can access the system with username and password.

### 5.2. 2 Performance

Easy Tracking of Records and Updating can be performed in the Attendance Module's.

### 5.2.3 User Friendly

The system is interactive in its nature and is extremely easy to use.

### 5.2.4 Reliability

The software will not be able to connect to the database in the event that there is no Internet Access or there may be any hardware failures.

### 5.2.5 Availability

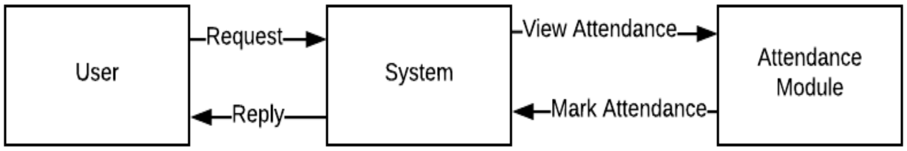
The software and all of its features will only be accessible to the authorized users of the organization like marking of attendance, viewing the attendance and updating the attendance.

# 6 . DESIGN

## 6.1 SYSTEM DESIGN

The proposed system must implement all the functional and non-functional requirements specified in the requirement analysis. Those requirements have a direct effect on the end result of the system to be developed.

Designing the front end has been kept simple, intuitive and easy to use since this will be the interface that the users will interact with.

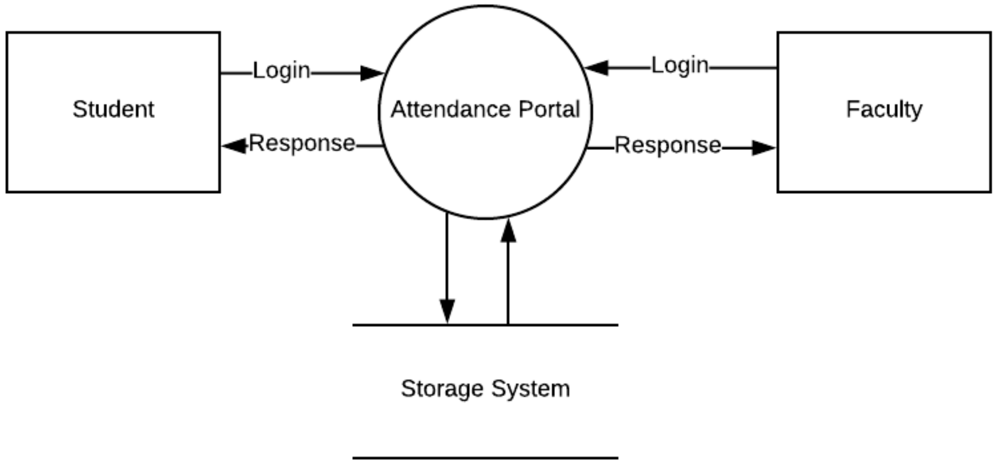


System Design

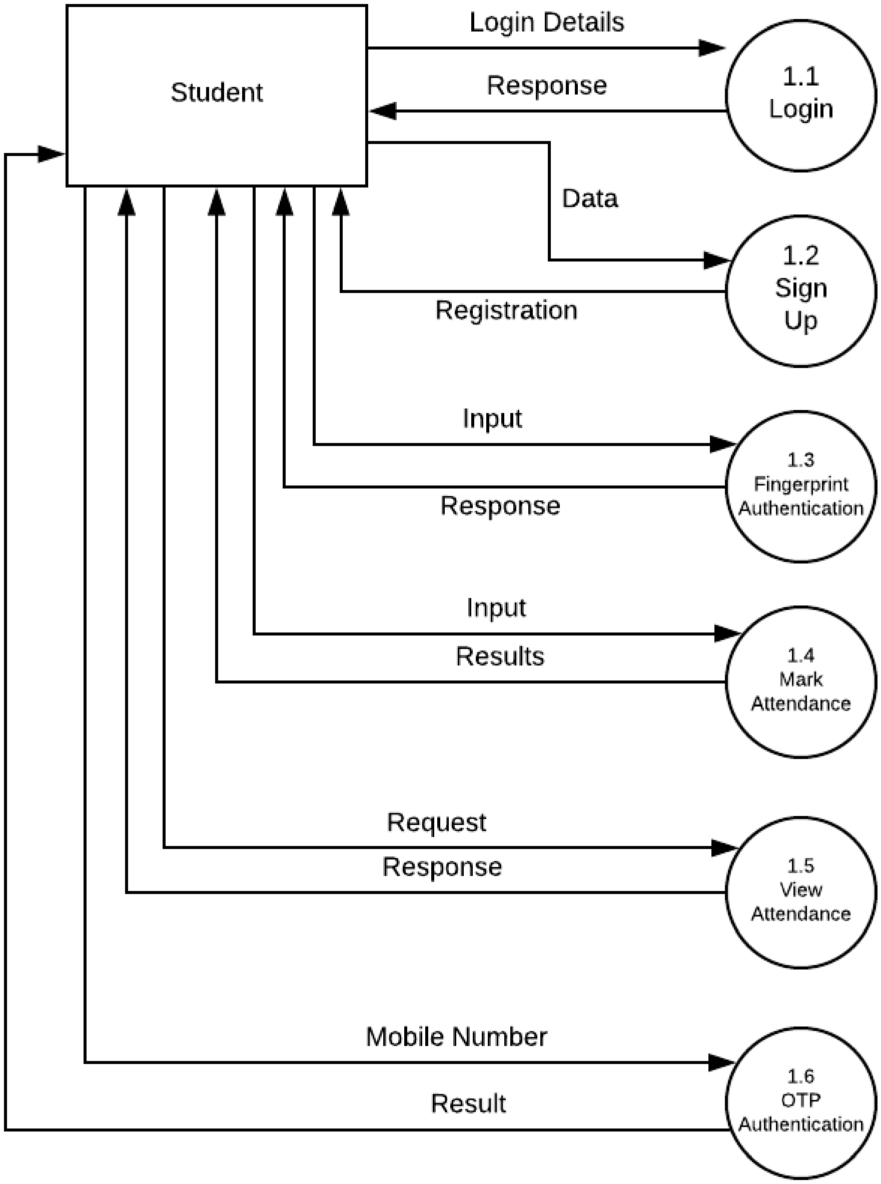
## 6.2 DETAILED DESIGN

The whole detailed design of the system can be illustrated using various diagrams representing the system.

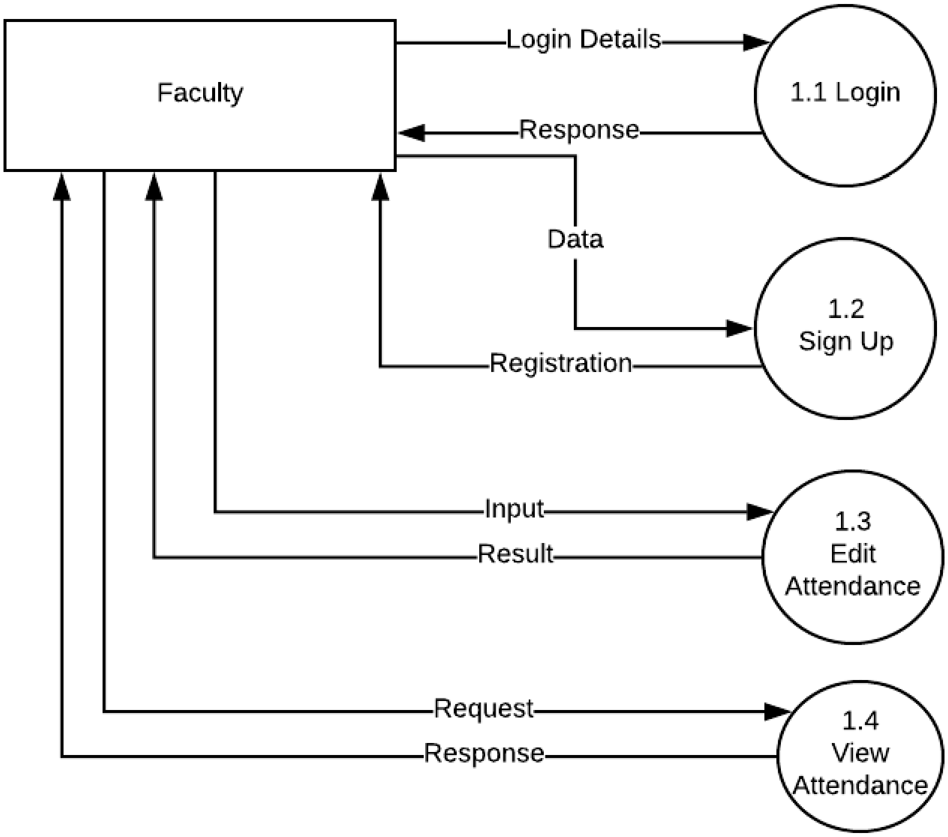
### 6.2.1 DATA FLOW DIAGRAM



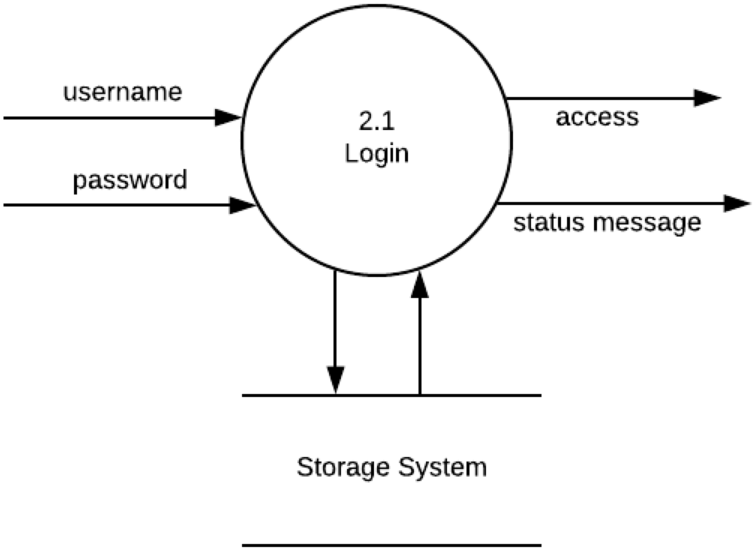
Data Flow Diagram- Level 0



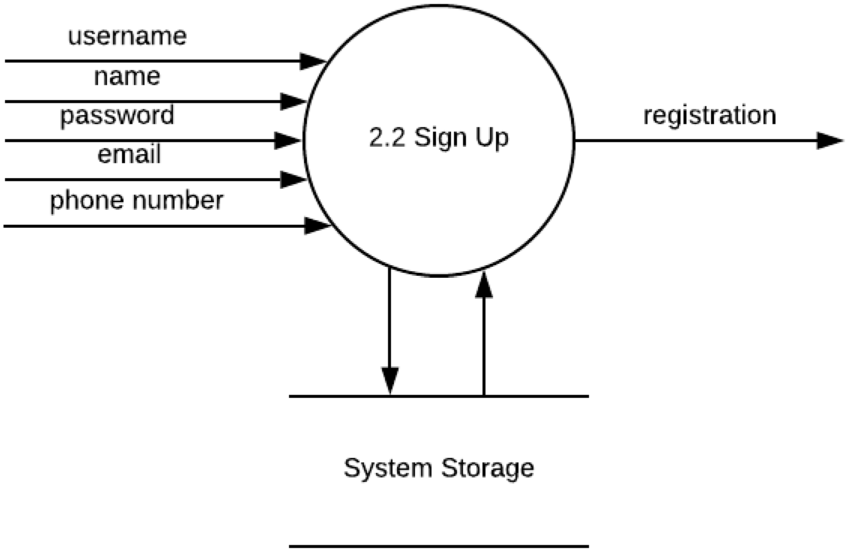
Level 1 - Student



Level 1 - Faculty

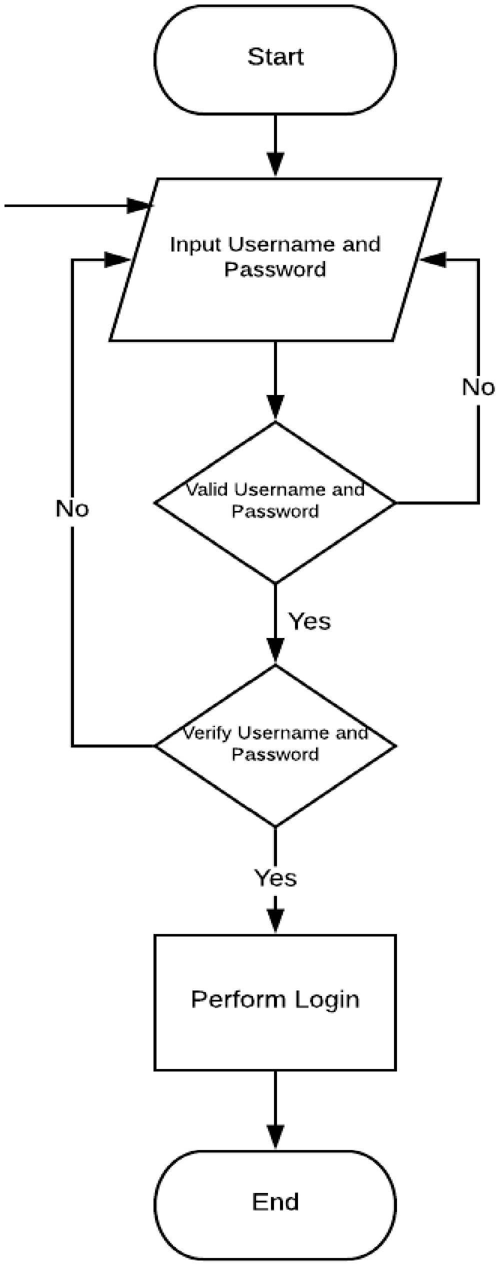


Level 2 – Login

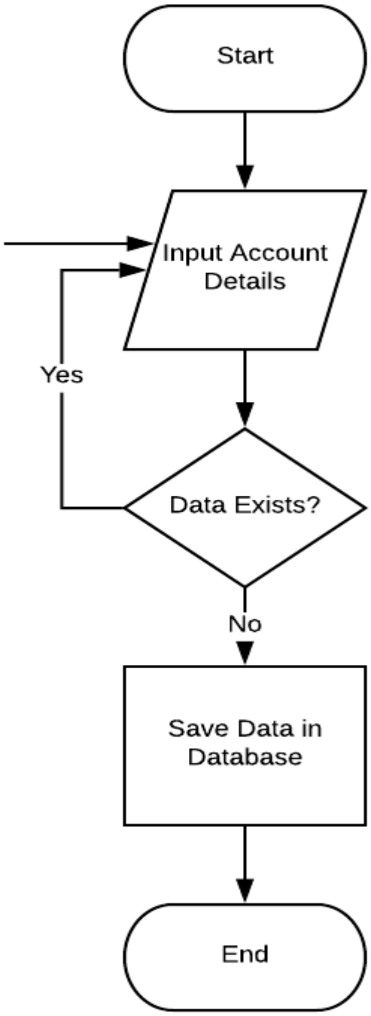


Level 2 – Sign Up

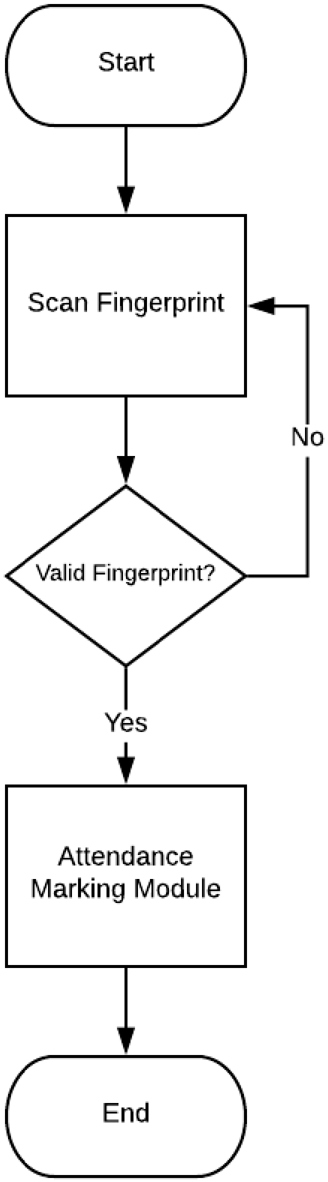
### 6.3 FLOW CHARTS


Start
Input Username and Password
No
Valid Username and Password
No
Yes
Verify Username and Password
Yes
Perform Login
End

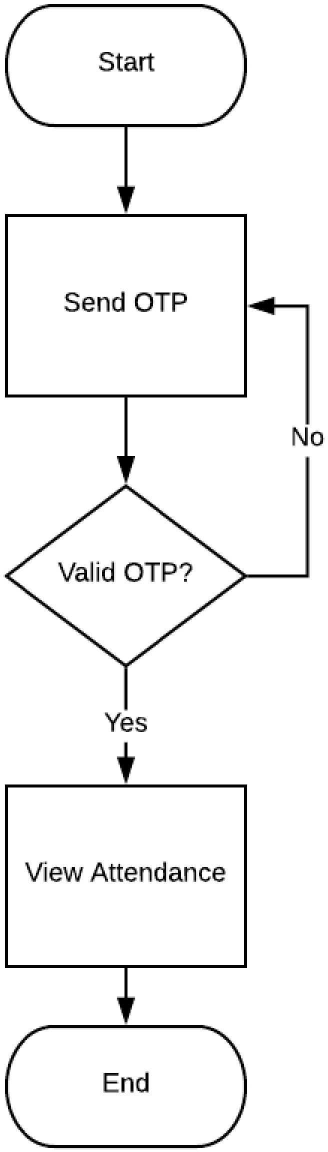
Login


Start
Input Account Details
Yes
Data Exists?
No
Save Data in Database
End

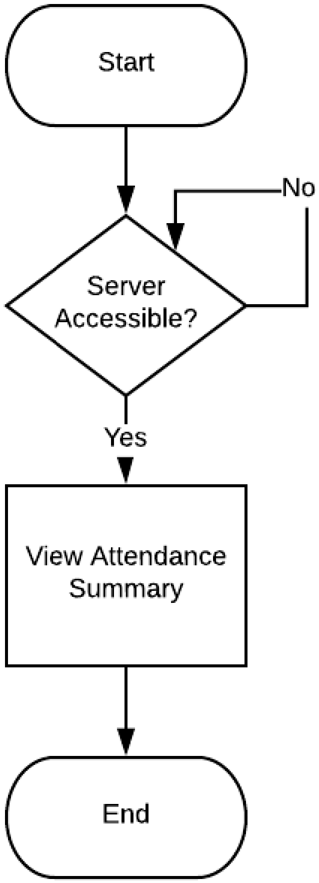
Sign Up



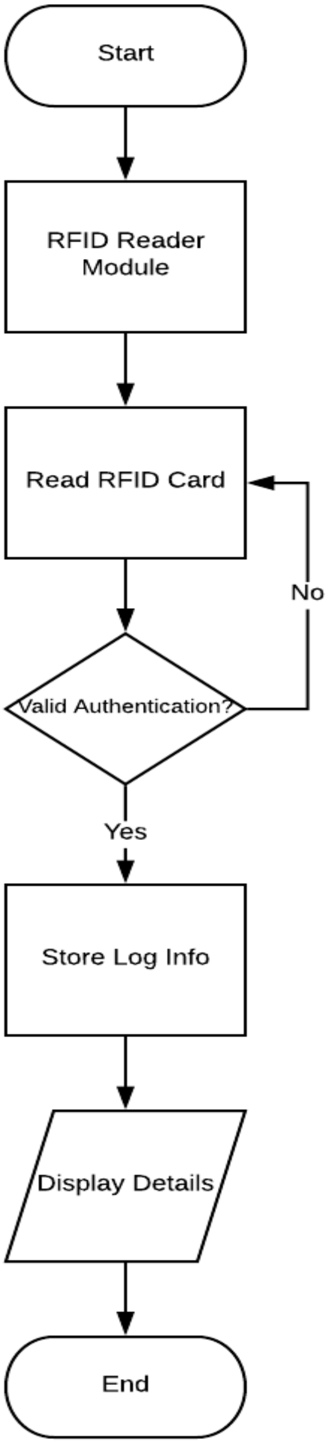
Fingerprint Module



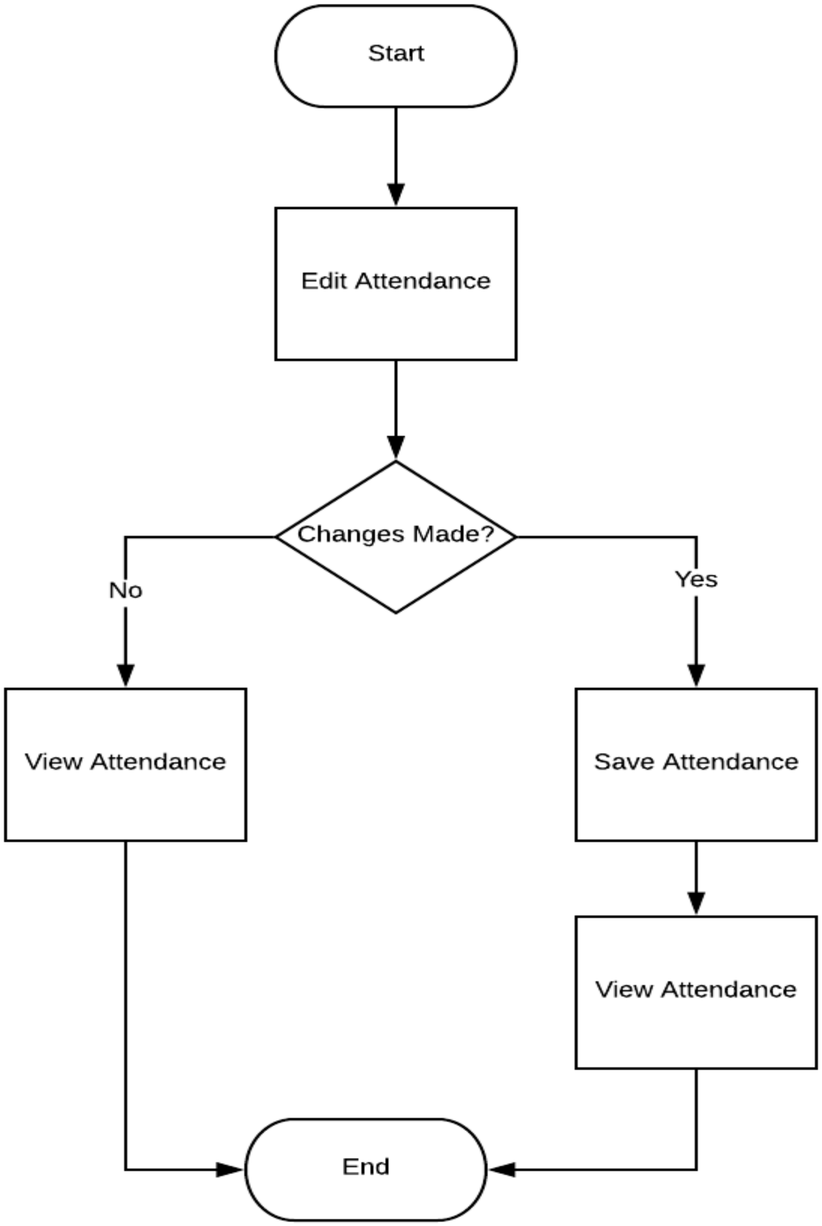
OTP Authentication



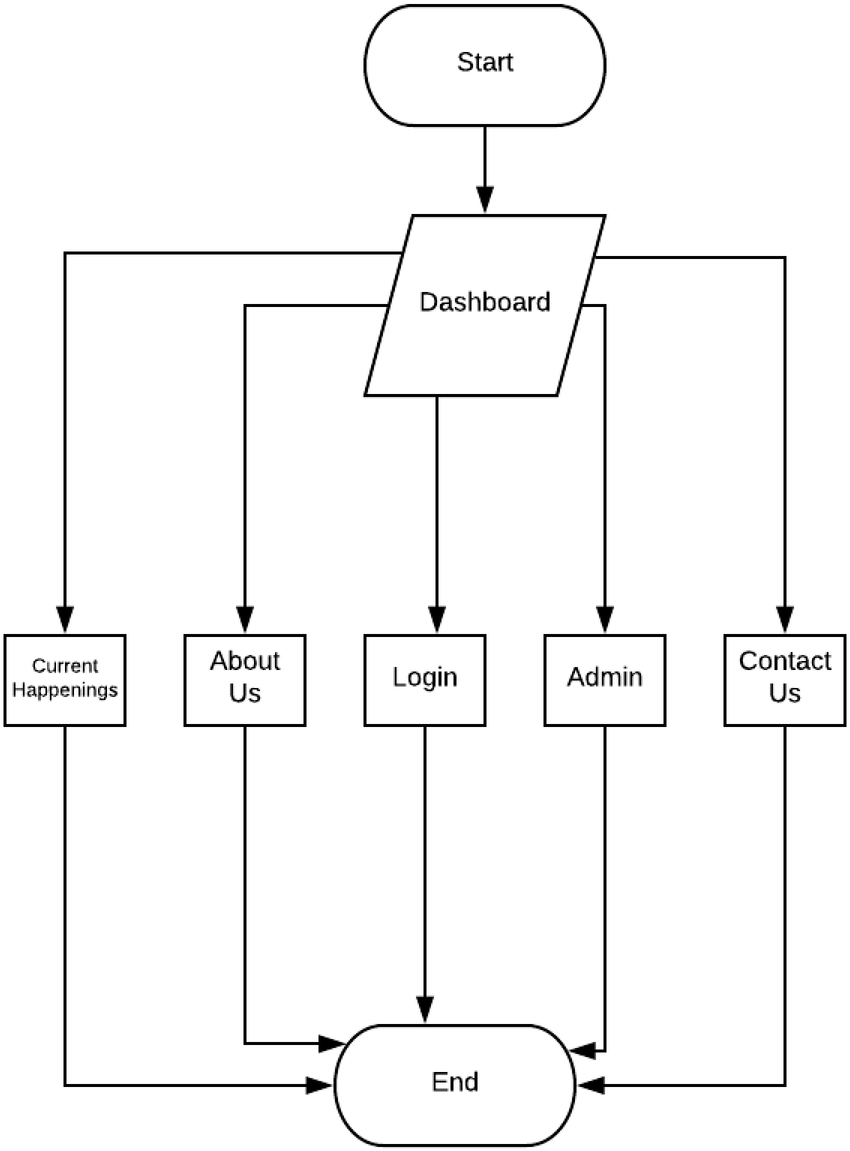
View Attendance



RFID Reading Process



Mark Attendance


Start
Dashboard
Current Happenings
About Us
Login
Admin
Contact Us
End

Dashboard

# 7. TESTING

This section elaborates the testing procedures we adopted. On a broader scene we tested the application using the following testing techniques.

### 1.Functional Dependency Testing

Functional dependency of software is the relationship between the different modules in the software. So for the entire software to function properly the interfacing between the modules and the data being entered and the output being obtained must be fine-tuned. So in functional dependency testing we perform tests using sets of input conditions so that all the behaviors are tested. For this purpose we used Black-Box testing.

### 2.Structural Dependency Testing

In Structural Testing we test the statements within the modules for errors and bugs. This is done in the early stage of testing in conjunction with the development phase. Structural testing provides data integrity to the interfaces of the modules. This greatly reduces the error propagation. For this process we performed White-Box Testing.

## 7.1 STRUCTURAL TESTING (White-Box Testing)

Due to the large number of modules and the enormous amount of possible outcomes and also due to the random nature of the outcomes, the best White Box testing method for our purpose was Path Testing.

### 7.1.1 Path Testing

In Path testing we derive a set of paths such that the maximum number of control conditions can be tested. This set of paths is also called the basis set is derived with the help of flow charts of the module. Due to the large number of modules and the paths in each individual module, we cannot show the analysis of each and every module or path. However we have compiled a defect log of the various type of errors and bugs that we came across while testing the module statements.

## 7.2 FUNCTIONAL TESTING

Once the independent modules were fully developed and the errors in the individual modules were removed we moved onto the next phase of testing and that is Black-Box testing. We chose the method of boundary value analysis for this process.

### 7.2.1 Boundary Value Analysis

Errors usually occur when the particular data passed is invalid for the module. Hence testing the module inter-dependency using boundary value greatly increases the chances of finding the error.

### 7.3 TESTING LEVELS

We divided the testing phase on the basis of the progress in the development phase.

### 7.3.1 STRUCTURAL TESTING

As we discussed earlier we performed white-box testing in conjunction with the development of the modules. This was done to detect and correct the types of errors as given below in the Defect Log.

| | Description | Defect Classification | Severity | Priority |
|---|---|---|---|---|
| 1 | Php file is missing | Missing File | H | H |
| 2 | Undefined Function | Syntactical | H | M |
| 3 | Array index out of Bounds | Logical | M | M |
| 4 | Undefined Variable | Syntactical | L | L |
| 5 | Time table array cant be fetched | Run-time | H | M |
| 6 | Execution time limit exceeded | Looping | L | M |
| 7 | Connectivity Error | Run-Time | L | L |
| 8 | Creating default object from empty value | Run-Time | M | M |
| 9 | Software Crash | Run-Time | H | H |
| 10 | Cannot Fetch data from RFID | Run-Time | H | H |
| 11 | Cannot send data to Android App | Run-Time | H | H |

### 7.3.2 FUNCTIONAL TESTING

Once all the inter-dependent modules are developed and free from errors, we started linking and integrating them to each other and performed Black-Box testing on them. In this we test the overall integrity of the modules.

| | Value | Expected Output | Obtained Output | Success/Failure |
|---|---|---|---|---|
| 1 | 4002ED8C26 | Attendance Marked for: Angad | Attendance Marked for: Angad | Success |
| 2 | 4002CB311D | Attendance Marked for: Simran | Attendance Marked for: Simran | Success |
| 3 | 900501A5FB | Attendance Marked for: Vasu | Card Un-Authorized | Failure |
| 4 | 40028A8B06 | Attendance Marked for: Tushar | Attendance Marked for: Tushar | Success |
| 5 | 8008948C81 | Attendance Marked for: Bittu | Attendance Marked for: Bittu | Success |

### 7.3.3 HARDWARE COMPATIBILTY TESTING

We tested the alpha version of the application on several different smartphones with different hardware and software configuration (all of them were running on Android). Our main objective behind this testing phase to find the minimum and recommended hardware requirements for running this application.

### 7.4 Test Results

After thoroughly testing the application we have eliminated all known errors and bugs in the system.

# 8. IMPLEMENTATION

In this section we are going to discuss the details about the implementation of the whole project.

## 8.1 PROJECT IMPLEMENTATION

The first task we faced during the project implementation phase was the division of the modules amongst the team members. This was done in a careful manner keeping in mind the various strength and weaknesses of the team members and the best utilization of the time and resources.

We divided the project into two modules:

1. Hardware
2. Software

Our team was divided into two sub teams where one sub team contained a person with high-end  Android Application based coding ability and another with creativity and design implementation which would be applied in the interfaces. The other team consisted of two members who were responsible for hardware related activities.

1. **Hardware**

   The hardware part of the project included the use of the Raspberry Pi along with the EM-18 RFID Reader Module. The task involved here was performing interfacing between these two modules. This also included fetching the RFID card data and then transferring and displaying it on the Raspberry Pi system. The main problem that arose during the interfacing of the modules was to perform Serial Communication between the modules to transfer data. For this the UART protocol was used which is mainly used to facilitate communication between two devices using a serial connection. This also includes configuring the baud rate which is responsible for flow control.

2. **Software**

   The Android App involved multiple interfaces which include the Dashboard, Login, Sign Up, Attendance Module. We made use of volley to enable communication between Raspberry Pi and Android Studio and interface between the two programming languages being used. We also made use of the Firebase API to provide OTP Authentication in the application. It also made use of a PHPMy Admin database to display the attendance summary in JSON (JavaScript Object Notation) readable format.

## 8.2 CODING

The most difficult part of the implementation was the coding phase. As the basic functionalities of the system were made separately. This however caused a lot of difficulties during the testing and the debugging due to the high level of coupling and large size of code. Once the code was debugged this micro this micro modulation saved several lines of code and the simplified the job of maintenance and improved reliability.

## 8.3 POST IMPLEMENTATION MAINTAINENCE

After implementation and testing the next major phase is maintenance. This phase is very important and requires lots of attention and resources because even after several waves of testing and debugging several bugs remained in the software and new unknown shortcomings and faults may be discovered. Fixing these issues helps in improving the scope of the project and the user’s confidence in the software. Since there was involvement of hardware in the form of the RFID Reader Module and the Raspberry Pi. Sometimes hardware faults could arise during run-time hence it was necessary to ensure the proper working of these hardware modules.

Maintenance of the software will be done in the form of upgrades and if any new shortcomings are discovered then this would entail a new form of development life cycle and would consume a lot of resources and time. The micro modules that were created helps increase the readability and reusability of the code, this brings down some overhead involved. In case of bugs, the entire code need to be inspected or changed. Instead just a small part of the code would be modified and tested to check if the bug has been fixed or not.

# 9. PROJECT LEGACY

During the course of this project we came across a lot of difficulties, which we had to surpass time and again in order to finish this project. We learned a lot from this project, from the various technical difficulties to the managerial aspects as well and the best part is that we found the solutions to the problems on our own. Hence in this section we will talk about the current status of our project, what the next stage of the project is and what we have gained from the project.

## 9.1 CURRENT STATUS OF THE PROJECT

We have created a full-fledged system which includes the Android Application along with the hardware which includes the Raspberry Pi and the EM-18 RFID Reader Module. Various features that complement the integration are also included so that the user can fully utilize the software. The Automated Attendance System provides two-way authentication including features like Fingerprint Recognition and an OTP (One Time Password) verification which enhances the overall security of the system

## 9.2 REMAINING AREAS OF CONCERN

Even though the Automated Attendance System is almost error-free but there are some areas of concern which include a scenario in which the tutor responsible for punching the final attendance is not present. Here the student might mark his/her attendance but it will only be partially-committed as the final input and confirmation has to be provided by the tutor.

### 9.3 TECHNICAL AND MANAGERIAL LESSONS LEARNT

During the course of the project development the technical and managerial lessons learned were immense. Much of this knowledge will help us in our future projects as well.

#### 9.3.1 TECHNICAL

The techniques involved in this project were completely new to us. Never before have we created a system that combines different technology stacks. Some of the important lessons learned are:

1.How to create an API in Android for performing back-end operation.

2.How to send and receive data between Android App and the Raspberry Pi.

3.Perform Database Management

4.How the Raspberry Pi works.

5.How the EM-18 RFID Reader Module Works

#### 9.3.2 MANAGERIAL

This project was an eye-opener for us. The sheer amount of planning and management that goes into projects is mind-boggling. Some of the managerial lessons learned are:

1. How to perform feasibility analysis. This activity helps identify the risks and the issues that we might face during the development of the project.

2. The designing phase must be fully thought out and all ambiguities must be removed before going to the coding segment.

3. Planning the testing phase increases the chance of finding bugs and critical faults in the system and helps in methodically covering all statements and decision trees.

# 10. USER MANUAL

## 10.1 APPLICATION

### 10.1.1 INTRODUCTION

The Automated Attendance System is system built to reduce effort and input needed in the attendance marking process. It also ensures total integrity in the attendance process by providing two-way authentication security using fingerprint recognition and OTP(One time password).

### 10.1.2 SYSTEM REQUIREMENT

Before installing our software please make sure that your smartphone meets the minimum specifications required to run the software.

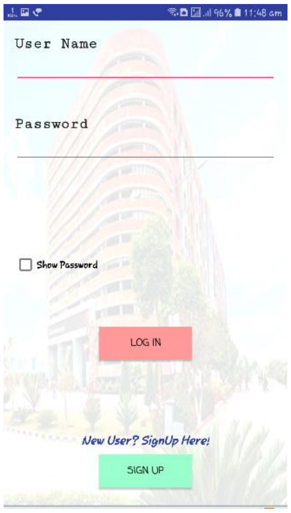
| Hardware | Minimum Specifications | Recommended Specifications |
|---|---|---|
| Process | Qualcomm Snapdragon 420 1.2 GHz or Samsung Exynos Dual core 1.2GHz | Qualcomm Snapdragon 620 1.4GHz or Samsung Exynos Core 1.4GHz |
| Display Resolution | 640x1136 | 720.1280 |
| RAM | 512 Mb | 1 GB |
| Internet Connectivity | 1 Mbps | 8 Mbps |

There is a necessary condition that the smartphone must have a fingerprint sensor to be able to run the software.

## 10.1.3 SOFTWARE FEATURES

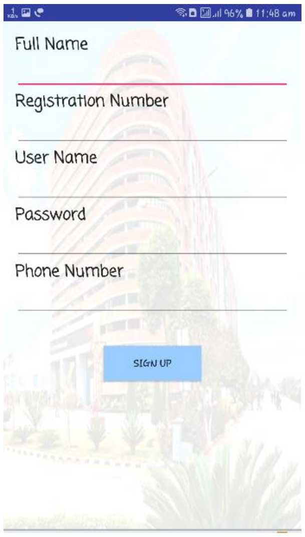
### 10.1.3.1 Log In

Used to Login to the Application to mark the attendance.



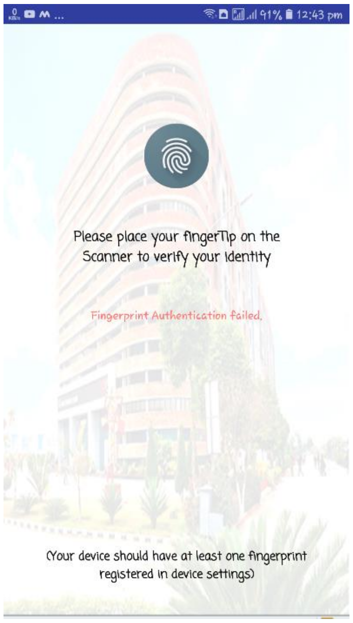
### 10.1.3.2 Sign Up

Used to register to the attendance system and become and authenticated user.



### 10.1.3.3 Fingerprint Scan

The first stage in the authentication of the student to mark the attendance.



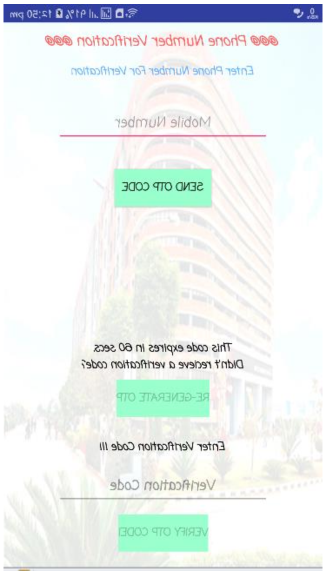
Please place your fingerTip on the Scanner to verify your identity

Fingerprint Authentication failed.

(Your device should have at least one fingerprint registered in device settings)

### 10.1.3.4 One Time Password (OTP)

The second stage in authentication to ensure attendance integrity.



Phone Number Verification

Enter Phone Number For Verification

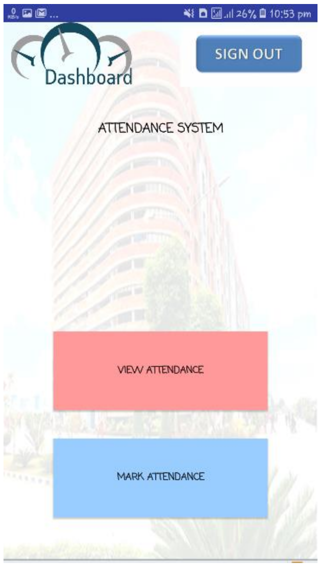
This code expires in 60 secs

Didn't recieve a verification code?

Enter Verification Code !!!

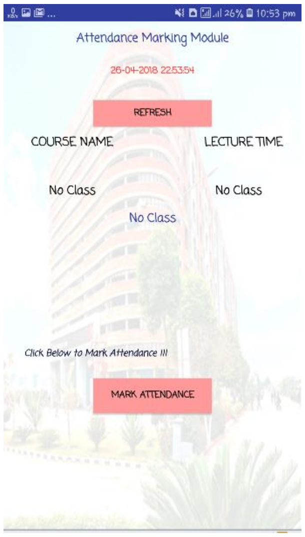
### 10.1.3.5 Attendance Module

This is used to choose between the action of viewing and marking the attendance.



### 10.1.3.6 Mark Attendance

This module can be used by the students to mark their own attendance.



Attendance Marking Module

26-04-2018 22:53:54

COURSE NAME LECTURE TIME

No Class No Class

No Class

Click Below to Mark Attendance !!!

## 10.2 API MODULE

## 10.2 API GUIDE:

We have used a Firebase API for Authentication Purposes. Using this API, we can send One Time Password(OTP) on registered Mobile Number and can easily authenticate the Student. The Details of this Firebase API is mentioned below:

### OTP Authentication:

| Data | Name | Value |
|---|---|---|
| Username of Developer | Vasudev Arora | vasudev.arora6120@gmail.com |
| Web API Key | attendancesystem-dbda5 | AlzaSyBh5MXgdM3FXSg7jqRU2cHpEa-eMXMhgtw |
| Public facing Name | project-755287360781 | [public_name] |

### POST Response Fields:

| Data | Name | Value |
|---|---|---|
| Success/Failure | Status | 1 |
| INPUT/Error Reason | ERROR | [Authenticated Text(OTP)] |

**Fingerprint Authentication:**

| Data | Name | Value |
|---|---|---|
| App NickName | Attendance System | Automated Attendance System |
| App ID | 1 : 755287360781 : android : 8617be46687f4f12 | [Unique ID] |
| Package Name | com.example.vasudev .attendancesystem | [Package Name] |
| Fingerprint HashKey | SHA Certificate fingerprint | DF:AB:93:A8:F1:7C:A4:68:C4:6C:E1:77:CB:3 E:63:BC:6D:3C:BB:E0 |

## 10.3 FREQUENTLY ASKED QUESTIONS

In this section we shall discuss all the FAQs which would help the users in case the same case arises.

- The application continues to load but nothing happens?
  This might be due to lack of connectivity between the app and the local network that has been created

- The application hangs up and suddenly becomes irresponsive?
  The application is busy sending and receiving data from the server. Please wait for sometime.

- Trying to view the attendance history but the output is not displayed?
  This is due to lack of internet connectivity for the Raspberry Pi hence no connection can be formed.

# 11.SOURCE CODE

## 1.1. LOGIN

```
class LoginActivity extends AppCompatActivity {


    public SQLiteDatabase db;

    EditText user,pass;

    String user1,pass1;


    @SuppressLint("WrongViewCast")

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_login);

        user=findViewById(R.id.user11);

        pass=findViewById(R.id.password11);

        db();




    }

    int x=0;

    @Override

    public void onBackPressed(){

        Toast.makeText(this, "Sorry, You Can't Go Back", Toast.LENGTH_SHORT).show();

    }
    public void showpassword(View view){
        x++;
        if (x%2==1) pass.setTransformationMethod(null);
        else pass.setTransformationMethod(new PasswordTransformationMethod());
    }

    public void db(){
        db=openOrCreateDatabase("AttendanceSystem", Context.MODE_PRIVATE, null);
        db.execSQL("CREATE TABLE IF NOT EXISTS Attendance(first VARCHAR,last
VARCHAR,username VARCHAR ,password VARCHAR,email VARCHAR);");    }

    public void login(View xyz){
        user1=user.getText().toString().trim();
        pass1=pass.getText().toString().trim();
        if (user1.equals("")||pass1.equals("")){
            show("Empty","Field is Empty");


        }
        else{
            try{
                Cursor c=db.rawQuery("SELECT * FROM Attendance WHERE username='"+user1 +
"'" , null);
                if (c.moveToFirst()){
                    if (pass1.equals(c.getString(3))){
                        show("Success","Logged In Successfully");
                        Handler h = new Handler(){
                            @Override
                            public void handleMessage(Message msg){
                                Intent i=new Intent(LoginActivity.this,MainActivity.class);
                                startActivity(i);
                            }
                        };
                        h.sendEmptyMessageDelayed(0,1500);
                    }
                    else{
                        show("Login Failed","Password is not matched");
                    }
                }
                else{
                    show("Login Failed","Entered Details are Incorrect");


                }
            }
            catch (Exception ee){
                show("Error","Cant Login");
            }
        }
    }
```

```
public void signup(View z){
    Intent k=new Intent(this,Signup.class);
    startActivity(k);
}

public void dashboard(View view){
    Toast.makeText(this, "Welcome To Dashboard !!!", Toast.LENGTH_SHORT).show();
    Intent intent = new Intent(this,Dashboard.class);
    startActivity(intent);
}

public void show(String title,String message){
    AlertDialog.Builder builder=new AlertDialog.Builder(this);
    builder.setCancelable(true);
    builder.setTitle(title);
    builder.setMessage(message);
    builder.show();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
   // getMenuInflater().inflate(R.menu.menu_login, menu);
    return true;
}
```

```
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement

        return super.onOptionsItemSelected(item);
    }
}
```

**11.2 SignUp**

```
public class Signup extends AppCompatActivity {

    public  SQLiteDatabase db;
    EditText first,last,email,user,pass;
    public String first1,last1,user1,email1,pass1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);
        first= findViewById(R.id.editText);
        last= findViewById(R.id.editText2);
        user= findViewById(R.id.editText3);
        email= findViewById(R.id.email11);
        pass= findViewById(R.id.editText9);
        db();
    }




    public void db(){
        db=openOrCreateDatabase("AttendanceSystem", Context.MODE_PRIVATE, null);
        db.execSQL("CREATE TABLE IF NOT EXISTS Attendance(first VARCHAR,last VARCHAR,username VARCHAR,password VARCHAR,email VARCHAR);");
    }


    public void sign(View v)
    {
        first1 = first.getText().toString().trim();
        last1 = last.getText().toString().trim();
        email1 = email.getText().toString().trim();
        pass1 = pass.getText().toString().trim();
        user1 = user.getText().toString().trim();
        if (first1.equals("") || last1.equals("") || email1.equals("") || pass1.equals("") || user1.equals("")) {
            show("Error", "Field is Empty");
        } else {
            try{
                db.execSQL("INSERT INTO Attendance VALUES('" + first1 + "','" + last1 + "','" + user1 + "','" + pass1 + "','" + email1 + "');");
                Intent ij=new Intent(Signup.this,LoginActivity.class);
                startActivity(ij);
                show("Success","Sign up Successfully");
            }
            catch (Exception e){
                show("Error","cant sign up");
            }

        }



    }
    public void show(String title,String message) {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(message);
        builder.show();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
       // getMenuInflater().inflate(R.menu.menu_signup, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement



        return super.onOptionsItemSelected(item);
    }
}
```

## 11.3 OTP AUTHENTICATION

```
public class OtpAuthentication extends AppCompatActivity {

    private static final String TAG = "PhoneAuth";

    private EditText phoneText;
    private EditText codeText;
    private Button verifyButton;
    private Button sendButton;
    private Button resendButton;

    private String phoneVerificationId;
    private PhoneAuthProvider.OnVerificationStateChangedCallbacks
            verificationCallbacks;
    private PhoneAuthProvider.ForceResendingToken resendToken;

    private FirebaseAuth fbAuth;

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

        phoneText =  findViewById(R.id.editText5);
    codeText =  findViewById(R.id.editText6);
    verifyButton =  findViewById(R.id.button6);
    sendButton = findViewById(R.id.button4);
    resendButton =  findViewById(R.id.button5);

    verifyButton.setEnabled(false);
    resendButton.setEnabled(false);



    fbAuth = FirebaseAuth.getInstance();



}

public void sendCode(View view) {

    String phoneNumber = "+91"+phoneText.getText().toString();

    setUpVerificatonCallbacks();

    PhoneAuthProvider.getInstance().verifyPhoneNumber(
            phoneNumber,        // Phone number to verify
            60,                 // Timeout duration
            TimeUnit.SECONDS,   // Unit of timeout
            this,               // Activity (for callback binding)
            verificationCallbacks);
}

private void setUpVerificatonCallbacks() {

    verificationCallbacks =
            new PhoneAuthProvider.OnVerificationStateChangedCallbacks() {

                @Override
                public void onVerificationCompleted(
                        PhoneAuthCredential credential) {

                    resendButton.setEnabled(false);
                    verifyButton.setEnabled(false);
                    codeText.setText("");
                    signInWithPhoneAuthCredential(credential);
                }

                @Override
                public void onVerificationFailed(FirebaseException e) {

                    if (e instanceof FirebaseAuthInvalidCredentialsException) {
                        // Invalid request
                        Log.d(TAG, "Invalid credential: "
                                + e.getLocalizedMessage());
                    } else if (e instanceof FirebaseTooManyRequestsException) {
                        // SMS quota exceeded
                        Log.d(TAG, "SMS Quota exceeded.");
                    }
                }

                @Override
                public void onCodeSent(String verificationId,
                                       PhoneAuthProvider.ForceResendingToken token) {

                    phoneVerificationId = verificationId;
                    resendToken = token;

                    verifyButton.setEnabled(true);
                    sendButton.setEnabled(false);
                    resendButton.setEnabled(true);
                }
            };
}

public void verifyCode(View view) {

    String code = codeText.getText().toString();

    PhoneAuthCredential credential =
                PhoneAuthProvider.getCredential(phoneVerificationId, code);
        signInWithPhoneAuthCredential(credential);


         if (phoneVerificationId.equalsIgnoreCase(code)) Toast.makeText(this, "OTP Verification Is
Done", Toast.LENGTH_SHORT).show();
    }


    private void signInWithPhoneAuthCredential(PhoneAuthCredential credential) {
        fbAuth.signInWithCredential(credential)
                .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful()) {
                            codeText.setText("");
                            resendButton.setEnabled(false);
                            verifyButton.setEnabled(false);
                            Toast.makeText(OtpAuthentication.this, "OTP Verification Done !!!",
Toast.LENGTH_SHORT).show();
                            Handler h = new Handler(){
                                @Override
                                public void handleMessage(Message msg){
                                    Intent intent = new Intent(OtpAuthentication.this,Marked.class);
                                    startActivity(intent);
                                }
                            };
                            h.sendEmptyMessageDelayed(0,1200);
                            FirebaseUser user = task.getResult().getUser();


                        } else {
                            if (task.getException() instanceof
                                    FirebaseAuthInvalidCredentialsException) {
                                // The verification code entered was invalid
                                Toast.makeText(OtpAuthentication.this, "Sorry, OTP Verification Failed !!!", Toast.LENGTH_SHORT).show();
                            }
                        }
                    }
                });
    }


    public void resendCode(View view) {


        String phoneNumber = phoneText.getText().toString();


        setUpVerificatonCallbacks();


        PhoneAuthProvider.getInstance().verifyPhoneNumber(
                phoneNumber,
                60,
                TimeUnit.SECONDS,
                this,
                verificationCallbacks,
                resendToken);
    }


}
```

FingerprintAuthenticationHandler.Java

```
public class FingerprintAuthenticationHandler extends FingerprintManager.AuthenticationCallback {



    private Context context;


    // Constructor
    public FingerprintAuthenticationHandler(Context mContext)
    {
        context = mContext;
    }
}

    public void startAuth(FingerprintManager manager, FingerprintManager.CryptoObject
cryptoObject) {
        CancellationSignal cancellationSignal = new CancellationSignal();
        if (ActivityCompat.checkSelfPermission(context,
Manifest.permission.USE_FINGERPRINT) != PackageManager.PERMISSION_GRANTED) {
            return;
        }
        manager.authenticate(cryptoObject, cancellationSignal, 0, this, null);
    }


    @Override
    public void onAuthenticationError(int errMsgId, CharSequence errString) {
        this.update("Fingerprint Authentication Error\n" + errString, false);
    }


    @Override
    public void onAuthenticationHelp(int helpMsgId, CharSequence helpString) {
        this.update("Fingerprint Authentication Help\n" + helpString, false);
    }
    @Override
    public void onAuthenticationFailed() {
        this.update("Fingerprint Authentication failed.", false);
    }


    @Override
    public void onAuthenticationSucceeded(FingerprintManager.AuthenticationResult result) {
        this.update("Fingerprint Authentication succeeded.", true);
    }


    public void update(String e, Boolean success){
        TextView textView = ((Activity)context).findViewById(R.id.errorText);
        textView.setText(e);
        if(success){
            textView.setTextColor(ContextCompat.getColor(context,R.color.colorPrimaryDark));
            Toast.makeText(context, "Fingerprint Matched Successfully !!!",
Toast.LENGTH_SHORT).show();

            Handler h = new Handler(){
                @Override
                public void handleMessage(Message msg){
                    Intent i = new Intent(context, Dashboard2.class);
                    context.startActivity(i);
                }
            };
            h.sendEmptyMessageDelayed(0,1500);


        }
```

activity_otp_authentication.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/lpu4"
    tools:context="com.example.vasudev.attendancesystem.OtpAuthentication">

    <TextView
        android:id="@+id/textView5"
        android:textColor="#66b2ff"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="52dp"
        android:fontFamily="casual"
        android:text="Enter Phone Number For Verification"
        android:textAlignment="center"
        android:textStyle="bold|italic" />

    <EditText
        android:id="@+id/editText5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="106dp"
        android:ems="10"
        android:fontFamily="casual"
        android:hint="Mobile Number"
        android:inputType="number"
        android:textAlignment="center" />

    <Button
        android:id="@+id/button4"
        android:layout_width="125dp"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"

        android:layout_centerHorizontal="true"

        android:layout_marginTop="189dp"

        android:background="#99ffcc"

        android:fontFamily="casual"

        android:onClick="sendCode"

        android:text="Send OTP Code"

        android:textStyle="bold" />


    <TextView

        android:id="@+id/textView6"

        android:textColor="#000000"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_above="@+id/textView7"

        android:layout_centerHorizontal="true"

        android:fontFamily="casual"

        android:text="This code expires in 60 secs."

        android:textStyle="bold|italic" />


    <TextView

        android:id="@+id/textView7"

        android:textColor="#000000"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="255dp"
        android:fontFamily="casual"
        android:text="Didn't recieve a verification code?"
        android:textStyle="bold|italic" />

    <Button
        android:id="@+id/button5"
        android:background="#99ffcc"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="199dp"
        android:fontFamily="casual"
        android:onClick="resendCode"
        android:text="Re-Generate OTP"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/textView8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"

        android:layout_marginBottom="149dp"

        android:fontFamily="casual"

        android:text="Enter Verification Code !!!"

        android:textColor="#000000"

        android:textStyle="bold|italic" />


    <EditText

        android:id="@+id/editText6"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_alignParentBottom="true"

        android:layout_centerHorizontal="true"

        android:layout_marginBottom="85dp"

        android:ems="10"

        android:fontFamily="casual"

        android:hint="Verification Code"

        android:inputType="textPersonName"

        android:textAlignment="center"

        android:textStyle="bold" />


    <Button

        android:id="@+id/button6"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"
```

```
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="25dp"
        android:background="#99ffcc"
        android:fontFamily="casual"
        android:onClick="verifyCode"
        android:text="Verify OTP Code!"
        android:textAlignment="center"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/textView9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="11dp"
        android:fontFamily="casual"
        android:text=" @@@ Phone Number Verification @@@ "
        android:textAlignment="center"
        android:textColor="#ff6666"
        android:textSize="18sp"
        android:textStyle="bold|italic" />
</RelativeLayout>
```

## 11.4 FINGERPRINT ACTIVITY

```
<?xml version="1.0" encoding="utf-8"?>
  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
  android:background="@drawable/lpu4"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.vasudev.attendancesystem.MainActivity">


    <LinearLayout
      android:layout_width="match_parent"
      android:id="@+id/headerLayout"
      android:orientation="vertical"
      android:gravity="center"
      android:layout_marginTop="100dp"
      android:layout_height="wrap_content">


      <ImageView
        android:id="@+id/icon"
        android:layout_width="86dp"

        android:layout_height="79dp"

        android:layout_marginBottom="30dp"

        android:paddingTop="2dp"

        android:src="@drawable/ic_fp" />



      <TextView

        android:id="@+id/desc"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:layout_margin="16dp"

        android:fontFamily="casual"

        android:gravity="center"

        android:paddingEnd="30dp"

        android:paddingStart="30dp"

        android:text="Please place your fingerTip on the Scanner to verify your identity"

        android:textAlignment="center"

        android:textColor="#000"

        android:textSize="16sp"

        android:textStyle="bold" />



      <TextView

        android:layout_width="match_parent"
```

```
            android:layout_height="wrap_content"
            android:textColor="@color/errorText"
            android:textSize="14sp"
            android:textAlignment="center"
            android:id="@+id/errorText"
            android:paddingEnd="30dp"
            android:paddingStart="30dp"
            android:layout_marginTop="30dp"
            android:gravity="center"/>


        </LinearLayout>


    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="26dp"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:fontFamily="casual"
        android:text="(Your device should have at least one fingerprint registered in device
settings)"
        android:textAlignment="center"
        android:textColor="#000"
        android:textSize="14sp"
        android:textStyle="bold" />
    </RelativeLayout>
```

### 11.5 SIGNUP ACTIVITY

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context="com.example.vasudev.attendancesystem.Signup"
    android:id="@+id/kll"
    android:background="@drawable/lpu4">

    <TextView
        android:id="@+id/first11"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:fontFamily="casual"
        android:text="Full Name"
        android:textAlignment="center"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textStyle="bold" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/editText"
        android:layout_below="@+id/first11"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />

    <TextView
        android:id="@+id/last11"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/editText"
        android:fontFamily="casual"
        android:text="Registration Number"
        android:textAlignment="center"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textStyle="bold" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/editText2"
        android:layout_below="@+id/last11"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignRight="@+id/editText"
        android:layout_alignEnd="@+id/editText" />

    <TextView
        android:id="@+id/user11"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/editText2"
        android:fontFamily="casual"
        android:text="User Name"
        android:textAlignment="center"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textStyle="bold" />
```

```
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/editText3"
    android:layout_below="@+id/user11"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignRight="@+id/editText2"
    android:layout_alignEnd="@+id/editText2" />

<TextView
    android:id="@+id/password11"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/editText3"
    android:fontFamily="casual"
    android:text="Password"
    android:textAlignment="center"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:textStyle="bold" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/editText4"
    android:layout_below="@+id/password11"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignRight="@+id/editText3"
    android:layout_alignEnd="@+id/editText3" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/editText4"
    android:fontFamily="casual"
    android:text="Phone Number"
    android:textAlignment="center"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:textStyle="bold" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/email11"
    android:layout_below="@+id/textView"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignRight="@+id/editText4"
    android:layout_alignEnd="@+id/editText4" />

<Button
    android:id="@+id/button2"
    android:layout_width="135dp"
    android:background="#99ccff"
    android:layout_height="wrap_content"
    android:layout_below="@+id/email11"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="40dp"
    android:onClick="sign"
    android:text="Sign Up" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:ems="10"
    android:id="@+id/editText9"
    android:layout_below="@+id/password11"
    android:layout_alignParentLeft="true"
```

```
        android:layout_alignParentStart="true"

        android:layout_alignRight="@+id/editText4"

        android:layout_alignEnd="@+id/editText4"

        android:password="true" />


</RelativeLayout>
```

## 11.6 LOGIN ACTIVITY

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

    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"

    android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"

    android:paddingRight="@dimen/activity_horizontal_margin"

    android:paddingTop="@dimen/activity_vertical_margin"

    android:paddingBottom="@dimen/activity_vertical_margin"

    tools:context="com.example.vasudev.attendancesystem.LoginActivity"

    android:background="@drawable/lpu4">


    <Button

        android:id="@+id/button16"

        android:layout_width="135dp"

        android:layout_height="wrap_content"

        android:layout_alignParentBottom="true"

        android:layout_centerHorizontal="true"
    android:layout_marginBottom="210dp"
    android:background="#ff9999"
    android:fontFamily="casual"
    android:onClick="login"
    android:text="Log In"
    android:textAlignment="center"
    android:textStyle="bold" />

<Button
    android:id="@+id/button15"
    android:layout_width="135dp"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="15dp"
    android:background="#99ffcc"
    android:fontFamily="casual"
    android:onClick="signup"
    android:text="Sign Up"
    android:textAlignment="center"
    android:textStyle="bold" />

<TextView
    android:id="@+id/user1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:fontFamily="serif-monospace"
    android:text="User Name"
    android:textAlignment="center"
    android:textAllCaps="false"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:textStyle="bold" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/user11"
    android:layout_below="@+id/user1"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />


<TextView
    android:id="@+id/password1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/user11"
    android:layout_marginTop="42dp"
    android:fontFamily="serif-monospace"
    android:text="Password"
    android:textAlignment="center"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:textStyle="bold" />

<CheckBox
    android:id="@+id/checkBox11"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignStart="@+id/user1"
    android:layout_marginTop="228dp"
    android:checked="false"
    android:onClick="showpassword"
    android:text="Show Password"
    android:textStyle="bold" />

<Button
    android:id="@+id/button"
    android:layout_width="135dp"
    android:layout_height="wrap_content"

    android:layout_alignParentBottom="true"

    android:layout_centerHorizontal="true"

    android:layout_marginBottom="129dp"

    android:background="#ff9999"

    android:fontFamily="casual"

    android:onClick="dashboard"

    android:text="Dashboard"

    android:textAlignment="center"

    android:textStyle="bold" />


<EditText

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:inputType="textPassword"

    android:ems="10"

    android:id="@+id/password11"

    android:layout_below="@+id/password1"

    android:layout_alignParentLeft="true"

    android:layout_alignParentStart="true"

    android:layout_alignRight="@+id/user11"

    android:layout_alignEnd="@+id/user11"

    android:password="true" />


<TextView
```

```
        android:id="@+id/textView10"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="72dp"
        android:text="New User? SignUp Here!"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="18sp"
        android:textStyle="bold|italic" />

</RelativeLayout>
```