Smart Classroom powered by IoT

By

Ahmad Mukhtar

Hamid Munir

Ibrahim Ali Khan

Submitted to Faculty of Department of Computer Software Engineering National University of Sciences and Technology, Islamabad in partial fulfilment for the requirements of a B.E Degree in Computer Software Engineering, May 2017

In the name of Allah, the Most Beneficent, the Most Merciful

ABSTRACT

Smart Classroom powered by IoT

"Internet of Things" is on rise these days in this modern era of technology. Any institution or department without using modern technology is considered useless, now-a-days. Similarly, there is a strong need for modernization and automation.

Smart Classroom powered by IoT is also an advancement towards modern age technology. With the help of this system, users will be able to share and download files including lectures or other necessary documents over the IoT based network. It will also provide them the ability to conduct quiz, mark attendance using biometric attendance. They will also be able to send notifications over the network regarding activities in the institution and to provide feedback about a class.

This system facilitates teachers to share files with students, conduct quiz in an easy-to-use interface. It facilitates students to download files, take quiz, mark biometric attendance, view attendance and to provide feedback about a certain class. It also facilitates users to send and receive important notifications about activities related to the institute. Integration with Firebase provides reliability of data stored in the database and delivery of notifications. System is efficient enough to be supported by the minimum hardware technology available.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

It is certified that work contained in the thesis – Smart Classroom powered by IoT carried out by Ahmad Mukhtar, Hamid Munir, Ibrahim Ali Khan under supervision of Dr. Naima Iltaf for partial fulfilment of Degree of Bachelor of Software Engineering is correct and approved.

Approved By

Dr. Naima Iltaf

Department of CSE, MCS

Dated: ________________________

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

To our parents, without whose support and cooperation, a work of this magnitude would not have been possible. To our supervisor, Dr. Naima Iltaf who has given us great support and valuable suggestions throughout the implementation process.

ACKNOWLEDGEMENTS

There is no success without the will of ALLAH Almighty. We are grateful to ALLAH, who has given us guidance, strength and enabled us to accomplish this task. Whatever we have achieved, we owe it to Him, in totality. We are also grateful to our parents and family and well-wishers for their admirable support and their critical reviews. We would like to thank our supervisor. Dr. Naima Iltaf, for her continuous guidance and motivation throughout the course of our project. Without their help, we would have not been able to accomplish anything.

Table of Contents

List of Figures

Chapter 1. Introduction

1.1. Overview

This system facilitates teachers to share files with students, conduct quiz in an easy-to-use interface. It facilitates students to download files, take quiz, mark biometric attendance, view attendance and to provide feedback about a certain class. It also facilitates users to send and receive important notifications about activities related to the institute. Integration with Firebase provides reliability of data stored in the database and delivery of notifications. System is efficient enough to be supported by the minimum hardware technology available.

1.2. Problem Statement

Conventionally, attendance is marked manually which takes about 5 or even more than 5 minutes in a class and if these minutes add up, they lead to wastage of a lot of time. Moreover, delayed delivery of important notification through email or notice board may cause many losses to students and to teachers as well. Tests based on MCQs are still conducted on paper which also becomes hectic for teachers to first create and print many papers and then calculate the result at the end. No timely feedback can also cause problems to a teacher and a student as well.

1.3. Approach

The project involves implementation of an android application with the help of firebase or raspberry pi. Firebase will be utilized while using the application outside of class and raspberry pi will be utilized inside the class. It is a IoT based project which involves interaction between different roles based in the application.

1.4. Scope

The project is intended to automate the daily tasks in classrooms using the Internet of Things. It will automate the attendance system using the fingerprint of students. In case the student is absent a notification will be sent in the form of an email. It will integrate a cloud and android platform for sending notifications and making the lectures available for download. It will also allow a teacher to take quiz(test) or feedback from students anonymously on smartphone.

1.5. Objectives

The main objective of this system is to provide a facility to users which include teachers and students for easy, secure and reliable communication/service in the. All required types of communication are possible by the software system developed i.e. file sharing, feedback, quizzes.

During this project, all the aspects of software engineering are covered i.e. survey and feasibility analysis, requirement gathering, architectural and detailed design, implementation and testing along with documentation (SRS, SDS, Test Document, Final Report and User manual). Students are also expected to develop extensive knowledge and technical skills in the following fields:

1.1. Firebase
1.2. Socket Programming
1.3. Push Notifications
1.4. Android and Python Programming
1.5. NoSQL Database

1.6. Deliverables

Sr	Tasks	Deliverables
1	Literature Review	Literature Survey
2	Requirements Specification	Software Requirements Specification document (SRS)
3	Detailed Design	Software Design Specification document (SDS)
4	Implementation	Project demonstration
5	Testing	Evaluation plan and test document

6	Training	Deployment plan
7	Deployment	Complete application with necessary documentation

1.7. Overview of document

Purpose

This document includes software requirements for **Smart Classroom powered by IoT**, release number 1.0. This document specifies the detailed requirements and features of **Smart Classroom** which is being developed. This document will act as a guideline for developers and all the other stakeholders throughout the development.

Document Conventions

- Heading are prioritized in a numbered fashion, the highest priority heading having a single digit and subsequent headings having more numbers, per their level.
- Font used is Segoe UI and Calibri.
- All the main headings are of size 22 and bold.
- All the second level sub-headings are of size 16 and bold.
- All the further sub-headings are of size 14 and bold.
- All references in this document are provided where necessary, however where not present, the meaning is self-explanatory. All ambiguous terms have been clarified in the glossary at the end of this document.

Intended Audience and Reading Suggestions

This document is primarily intended for the developers, the project supervisor, and the evaluators. Furthermore, any person who needs to understand the system can found this document helpful.

For better understanding, the document is divided into sections:

- In chapter 1 and 2 an overall description of Application is provided. First product perspective is presented with product features and main functions. Then follow user classes and characteristics, operating environments that Application supports as well as design and implementation constraints. After all that, user documentation is presented and will provide you with more details about each feature's technology.

- In chapter 3 and 4 most important features are presented with detailed description, use cases and requirements.
- In chapter 5 testing and evaluation of the system is documented with detailed test cases for unit and integration testing.

This document is intended for:

- **Developers:** (Project Group)

  To be sure that they are developing the right project that fulfills the requirements provided in this document.
- **Testers:** (Project Group, Supervisor)

  To have an exact list of the features and functions that must respond per requirements.
- **Users:**

  To get familiar with the idea of the project and how to use/respond in failure situations and suggest other features that would make it even more functional.
- **Documentation writers:** (Project Group)

  To know what features and in what way they should explain. What technologies are required, how the system will respond in each user's action, what possible system failures may happen and what are the solutions to all those failures etc.
- **Project Supervisor:** (Dr. Naima Iltaf)

  This document will be used by the project supervisor to check whether all the requirements have been understood and in the end whether the requirements have been implemented properly and completely.
- **Project Evaluators:** (CSE Dept. MCS)

  To know the scope of the project and evaluate the project throughout the development for grading.

Project Scope

The project is intended to automate the daily tasks in classrooms using the Internet of Things. It will automate the attendance system using the fingerprint of students. In case the student is absent a notification will be sent in the form of an email. It will integrate a cloud and android platform for sending notifications and making the lectures available for download. It will also allow a teacher to take quiz(test) or feedback from students anonymously on smartphone.

Chapter 2. Literature Review

2.1. Internet of Things

IoT has emerged as a new network paradigm, which allows various physical entities in the world to connect with each other. The observed or generated information of these entities have a great potential to provide useful knowledge across different service domains, such as building management, energy-saving systems, surveillance services, smart homes, smart cities, etc. IoT was first proposed in 1999 by Kevin Ashton, who is the co-founder of Auto-ID center at the Massachusetts Institute of Technology (MIT). One foundational technology of IoT is the Radio-Frequency IDentification (RFID) technology, which allows microchips transmit the identification number of the objects to a reader through wireless communication. Through RFID technology, the physical objects can be identified, tracked, and monitored automatically. Nowadays, RFID technology has been widely adopted in logistics, pharmaceutical production, retailing, and supply chain management.

Another foundational technology of IoT is Wireless Sensor Networks (WSNs), which adopt interconnected intelligent sensors to periodically sense the monitored environment and send the information to the sink (or base station), at which the gathered/collected information can be further processed for end-user queries. The applications include disaster control, environment and habitat monitoring, battlefield surveillance, traffic control, and health care applications. Additionally, many other technologies and devices such as Near Field Communication (NFC), short-range wireless communication (*i.e.*, ZigBee and Bluetooth), universal mobile accessibility (*i.e.*, Wi-Fi hotspots, and cellular networks), social networking and cloud computing support internet of things to compose an extensive network infrastructure.

2.2. IoT in Pakistan

All over the world, people have been putting in effort to automate classroom activities. In Pakistan, we see today that every university has a portal/website for students to download lectures and other notes. Although our classrooms today include projectors, a good sound system, but these are mainly the two only changes that have been incorporated in our centuries old educational systems.

Technology is changing the world so rapidly, but even today, we are reluctant about incorporating it in our classrooms. We reject the use of smartphones in the class simply because the students might not learn from it, while it is inevitable, that if someone has already decided to waste his time in activities other than studies, he doesn’t need a smartphone to not study.

Moreover, in developing countries like Pakistan, where although 3G and 4G are available almost everywhere, still rates are not affordable for a student. In situation like these a solution like a local server is needed that doesn’t require Internet but makes sort of an intranet that works without connecting to the internet and still behaves like an internet inside its coverage area.

For achieving this, embedded systems using Raspberry Pi, Arduino etcetera are the best solutions and people have been suing them all over the world to come up with working solutions of smart class rooms. For example, in MIT, devices are installed within the chair themselves where they are being used to transfer lectures and digital representation of a concept or a theory.

We have incorporated a raspberry pi to act as the local server and help in lecture notes provisioning, conducting quizzes, real-time feedbacks, notifications and attendance marking and attendance viewing.

2.3. Existing Systems

The system of biometric attendance was implemented in Military College of Signals but it was limited to one device for all the students which would cause a chaotic situation as many students arrived at same time for attendance. Another flaw was that the attendance was marked only while entering the department and when leaving the department whether the student take any class or not, he/she was considered present by that device.

Some other applications also exist but they are either only allowing biometric attendance, we are offering a bigger application which includes not only biometric attendance but other significant features like lecture/file sharing, feedback sharing, push notifications, attendance notifications, and quizzes.

Chapter 3. Software Req. Specification (SRS)

3.1. Overall Description

Product Perspective

The project is intended to automate the daily tasks in classrooms using the Internet of Things. It will automate the attendance system using the fingerprint of students when they are in the proximity of class, once verified they will get the lectures related to current class which will be available for download on the android application. They will also be able to see their attendance on the application. A teacher will be able to take tests on the app and get feedback from students regarding the class anonymously. If a student is absent, an email will be sent to his/her guardian and administration.

Product Function:

Main features of the product are given below:

- Automated attendance by using raspberry pi and fingerprint sensor.
- Ability to see your attendance from your smartphone.
- Uploading Lectures or relevant materials.
- Downloading Lectures or relevant materials.
- Official notifications from administration to students.
- Feedback from students.
- Quiz/Test on the application.
- Email notification to guardian and administration if the student is absent.

User Classes and Characteristics

Following are user classes and their brief description.

Tester (occasional user)

Tester will use this project to check for bug finding. They will also use the project to check if it’s in accordance to the Software Requirements Specification document.

Project Supervisor (occasional user)

Project supervisor will also use the product to evaluate. They will use this product to find the accuracy and error in the output.

Students (Regular user)

Students will use the product to download lectures or relevant material, to see their attendance and to receive official notifications from the administration.

Teachers (Regular user)

Teachers will use the product to upload the lectures or relevant material, or to verify the attendance of students.

Administration

They will use the product to send official notifications related to the university and add or remove students.

Operating environment

Required operating environment for the application is listed below.

Hardware Requirements:

- **Android device:** To install software for communication
- **Network Infrastructure:** To provide the network connection to the software.
- **Raspberry PI:** To ensure the presence of student in class during attendance and local server.
- **Fingerprint enabled smartphone:** To obtain the fingerprint of students while marking attendance.

Software Requirements:

- **Operating System:** Android
- **Firebase:** The cloud database for collecting fingerprints from Raspberry PI and sending it to the android application.

Design and Implementation Constraints

Constraints of the product are given below:

- App will not work without network connection.
- The storage on cloud will be limited for storing lectures and fingerprints for the sake of project.
- Communication and speed will be dependent on network infrastructure's specifications.
- The number of devices connected will be limited for the sake of project as we will be using a free plan of Firebase.

User Documentation

A user manual will be provided to the users in which separate instructions will be given per user i.e. Regular user and the admin, developers and testers. It will include the details of the system's working. Help documents will also be a part of the system.

The project report will also be available for the users which will highlight the system features, working and procedures.

Assumptions and Dependencies

- Overall performance of the product will depend on the hardware infrastructure and network speed.
- User must know about the Interface for the better performance of the product.
- Limitations of the product must be kept in mind by the user.

3.2. System Features

System features are organized by use cases and functional hierarchy so that the main functions of the system will be understandable. In the description of system features there are several references in various system interfaces.

Biometric Attendance

This feature allows a student to mark his/her student.

Description

When a class will be about to start, students will use their smartphone to mark their attendance using the fingerprint sensor of their smartphone which will then be uploaded to the cloud.

Stimulus/Response Sequences

Data flow:

Basic Data Flow

1. User will be connected to the raspberry pi device installed in the classroom.
2. User gives his/her thumb impression for marking the attendance.
3. User's thumb impression is checked with old one for verification.
4. User's attendance gets marked after the verification.

Alternative Data Flows

Alternative Data Flow 1

1. User opens application and is not connected to any network.
2. Error message is displayed to user.

Alternative Data Flow 2

1. User's thumb impression doesn't match or isn't already in the system.
2. User is displayed with an error message.

Alternative Data Flow 2

1. User's device is not connected with raspberry pi.
2. User is displayed with an error message.

Functional Requirements

1. Application must be installed and connected to a network.
2. Raspberry pi and user's smartphone should be visible to each other for marking the attendance.
3. Raspberry pi and user's smartphone shall be able to connect with each other.
4. Student's id should be in the database for accessing the application.

Attendance Display

This feature provides students the ability to see their attendance from the android application.

Description

Students will be able to view their attendance of a specific class using this option.

Stimulus/Response Sequences

Data flow:

Basic Data Flow

1. User opens the attendance section in the android application.
2. Application retrieves his/her attendance from the cloud.
3. Application displays the attendance of the user.

Alternative Data Flows

Alternative Data Flow 1

1. User opens application and is not connected to any network.
2. Error message is displayed to user.

Functional Requirements

1. Application must be installed and connected to a network.
2. User shall be able to view attendance from the cloud anytime.
3. Student's id should be in the database for accessing the application.

Uploading Lectures or Relevant Material

This feature will provide teachers the ability to upload lectures or relevant material to the cloud or raspberry pi for the availability in the android application.

Description

User must be able to upload the lecture or relevant material for the class to the cloud.

Stimulus/Response Sequences

Data flow:

Basic Data Flow

1. User selects the upload option.
2. User uploads the lecture or relevant material.
3. Lecture or relevant material is uploaded.

Alternative Data Flows

Alternative Data Flow 1

1. User opens the app but isn’t connected to the network.
2. Error message is displayed.

Alternative Data Flow 2

1. User uploads the file and upload fails.
2. Error is displayed to the user.

Functional Requirements

1. Application must be installed and connected to the network.
2. User shall be able to upload lectures anytime.
3. Teacher’s id should be in the database for accessing the application.

Download Lectures or Relevant Material

This feature allows students to download lectures or relevant material.

Description

User must also be able to download files from the android application.

Stimulus/Response Sequences

Data flow:

Basic Data Flow

1. User taps on available lecture resource.
2. The resource starts downloading.

Alternative Data Flows

Alternative Data Flow 1

1. User taps on available lecture resource.
2. The resource isn't available.
3. Error message is displayed.

Alternative Data Flow 2

1. User opens the application and isn't connected to the network.
2. Error is displayed to the user.

Functional Requirements

1. Application must be installed and connected to the network.
2. User shall be able to download lectures anytime.
3. Student's id should be in the database for accessing the application.
4. Raspberry pi and user's smartphone shall be able to connect with each other.

Official Notifications

This feature allows administration to send official notifications.

Description

User must also be able to send the official notifications related to university.

Stimulus/Response Sequences

Data flow:

Basic Data Flow

1. User taps on the notification section.
2. User creates/types a new notification message and taps on send.
3. The notification is sent.

Alternative Data Flows

Alternative Data Flow 1

1. User opens the application and isn't connected to the network.
2. Error is displayed to the user.

Functional Requirements

1. Application must be installed and connected to the network.
2. User's id should be in the database for accessing the application.

Feedback

This feature allows students to give feedback about a teacher/class anonymously.

Description

User must also be able to give feedback about a teacher or class anonymously using the android application.

Stimulus/Response Sequences

Data flow:

Basic Data Flow

1. User gets a feedback option during the lecture.
2. User writes the feedback and taps on send.
3. The feedback is sent.

Alternative Data Flows

Alternative Data Flow 1

1. User opens the application and isn't connected to the network.
2. Error is displayed to the user.

Functional Requirements

1. Application must be installed and connected to the network.
2. User shall be able to give feedback during the lecture.
3. User's id should be in the database for accessing the application.
4. Raspberry pi and user's smartphone shall be able to connect with each other.

Test/Quiz

This feature allows teachers to take test/quiz from the class.

Description

A teacher can create the quiz for a class which can be started on a specified time and then the students can submit the quiz at the specified time and after the submission their result is shown.

Stimulus/Response Sequences

Data flow:

Basic Data Flow

1. User will create a new quiz/test from the quiz section before the class.

2. User will select the type of quiz.
3. User will write questions per type of quiz.
4. Quiz will be created.

Alternative Data Flows

Alternative Data Flow 1

1. User opens the application and isn't connected to the network.
2. Error is displayed to the user.

Functional Requirements

1. Application must be installed and connected to the network.
2. Teacher should be able to hide the quiz from the students before the quiz starts.
3. User's id should be in the database for accessing the application.

Attendance Notification

If the student is absent, an email notification will be sent.

Description

If the student is absent from the class, an email will be sent to his/her guardian and administration.

Stimulus/Response Sequences

Data flow:

Basic Data Flow

1. At the end of allocated attendance time, raspberry pi will check for the absent students from the cloud.
2. An email will then be sent to the guardian of the absent students and administration as well.

Alternative Data Flows

Alternative Data Flow 1

1. User opens the application and isn't connected to the network.
2. Error is displayed to the user.

Functional Requirements

1. Raspberry pi should be able to get the attendance of students from the cloud.
2. Raspberry pi should be able to send emails.

Add/Remove User

Admin will be able to add or delete users in the cloud database.

Description

Administration will be able to add/delete users in the database of application so that they can access the app through their login ids.

Stimulus/Response Sequences

Data flow:

Basic Data Flow

1. User will tap on the Add/Remove User.
2. User will select add/delete user per situation.
3. User will fill all the necessary details about the user or just delete user's data if that's the case.

Alternative Data Flows

Alternative Data Flow 1

1. User opens the application and isn't connected to the network.
2. Error is displayed to the user.

Functional Requirements

1. Application must be installed and connected to network.
1. Cloud should be accessible for adding or deleting the users from the database.

3.3. External Interface Requirements

User Interfaces

Responsive graphical user interfaces must be provided to user to work with the application. Here are few dummy screenshots of the application:

Login Screen:

This screen will allow user to login using Raspberry PI (in class) or Cloud (outside class).

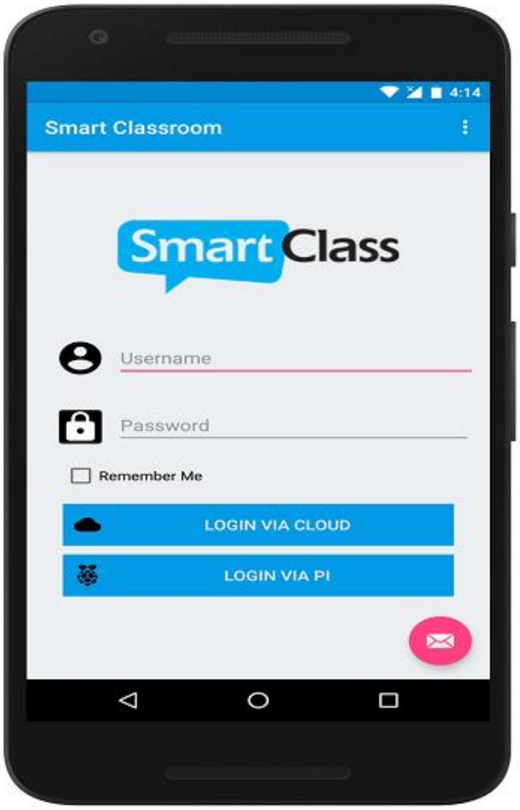

Figure 3.3.1 Login Screen

Student screen:

Students will be able to mark attendance, view attendance, download files, take quizzes, provide feedback and view notifications using this screen.

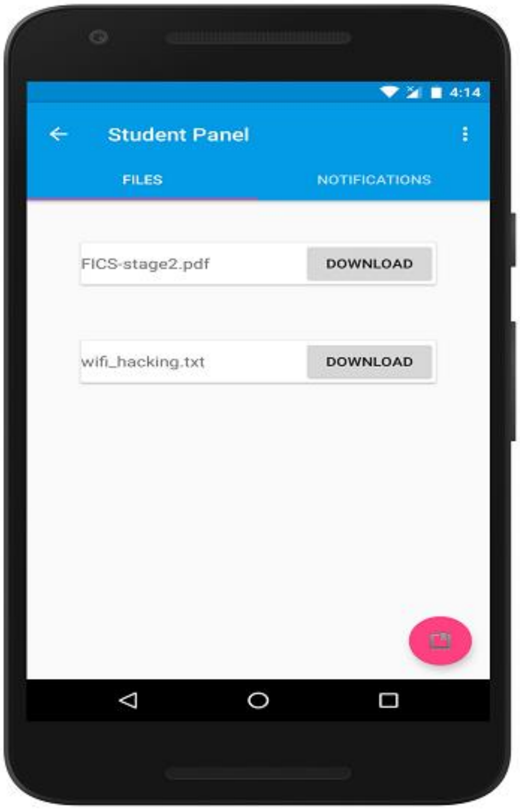

Figure 3.3.2 Student Screen

Teacher Screen:

Teachers will be able to upload files, view feedback, and conduct quiz using this screen.

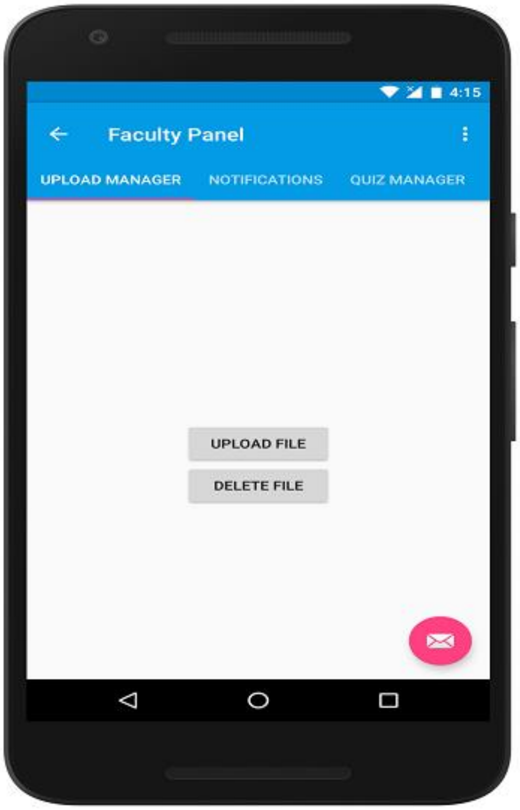

Figure 3.3.3 Teacher Screen

Admin Screen:

Admin(s) will be able to update the database for students, courses, and schedule accordingly using this screen.

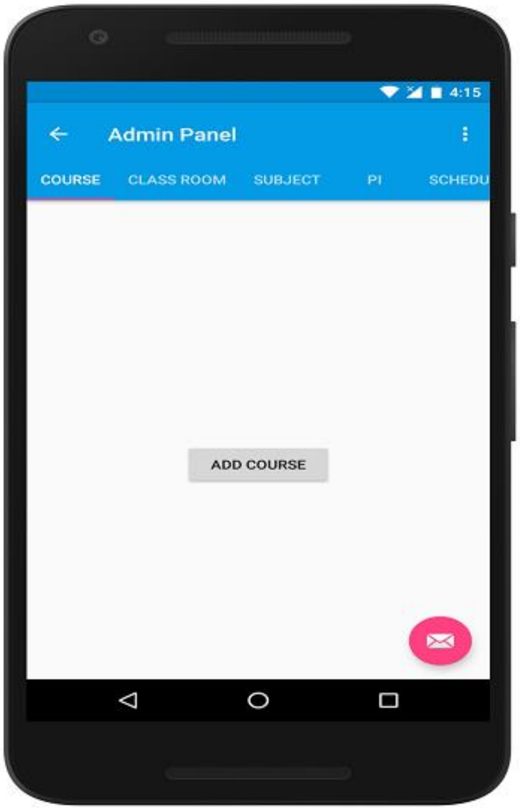

Figure 3.3.4 Admin Screen

Hardware Interfaces

- Attendance will be marked using fingerprint enabled smartphone.
- Thumb impression will be collected using fingerprint sensor.
- Android device for uploading/download lectures, sending/receiving notifications, and for checking attendance.

Software Interfaces

- Android OS for running the android application.
- Raspberry PI OS which will be installed on the Raspberry PI.

Communications Interfaces

- Firebase will be used for communication outside the class.
- Raspberry PI will act as local server for communication inside the class.

3.4. Other Nonfunctional Requirements

Performance Requirements

Application shall run on a minimal amount of memory and take up a small amount of disk space after install. Depending on the performance of the user's device or amount of data, the communication might slow down the Application.

Safety Requirements

This application is fast and responsive. However as mentioned in Performance Requirements, working with large data may lead Application to become unresponsive or even crash. Network crash will also waste a lot of time and user may lose information.

Security Requirements

Application should not need any additional information other than the collected data from the user or already present data. There are no connections to third party devices or severs so no data will be sent or received or used in any way.

Software Quality Attributes

- **Reliability**

  The application should run perfectly with all the features mentioned above available. It should be tested and debugged completely. All exceptions should be well handled.

- **User Friendliness/Simplicity**

  Application should have a graphical user interface with user friendly menu and options.

- **Availability**

  Application will be available through a suggested and well known platform. The cloud is expected to be available 99.9% of the time depending on the network connection.

- **Ease of Installation**
  The user shall be able to install the application with just one click/tap from the designated website.

Chapter 4. Design and Development

4.1. Introduction

This chapter includes software design for **Smart Classroom powered by IoT**, release number 1.0. This document specifies the detailed architectural design of **Smart Classroom** which is being developed. This document will act as a guideline for developers and all the other stakeholders throughout the development. Document includes classes and their inter-relationships, use cases with detailed descriptions, sequence diagrams, activity diagrams and various others.

4.2. System Architecture Description

This section provides detailed system architecture of Smart Classroom powered by IoT. Overview of system modules, their structure and relationships are described in this section. User interfaces and related issues are also discussed.

Overview of Modules

This Smart Classroom powered by IoT Application has following required modules. Here we give a brief overview of all these modules.

1. **Biometric Attendance:**

   This module allows the users to mark their attendance using their smartphone fingerprint sensor within the proximity of classroom.

2. **Attendance Display:**

   This module allows the users to see their attendance on their smartphone from the cloud application to remain updated with their attendance information.

3. **Uploading Lectures:**

   This module enables a teacher to upload lecture slides or relevant notes for a specific lecture on the cloud through the application which will then be available on student side for download.

4. **Download Lectures:**

   This module enables the student to download the lecture slides or relevant notes which will be made available by the teacher from the cloud application.

5. **Official Notifications:**

   This module will be used for sending the important notifications related to the academia or events to a specific group of students/faculty or to the whole institute through the cloud platform.

6. **Feedback:**

   This module will be used by students for giving feedback about a lecture anonymously or otherwise if they have any query or suggestion.

7. **Quiz/Test Module:**

   This module will be used by both teacher and students. Teachers will use it to create quiz for students and students will use it for taking the quiz. These quizzes will be MCQ based.

8. **Attendance Notification:**

   This module will start working after the attendance time has ended and it will send the emails about the absent students to someone in faculty and the guardians of the students.

9. **Add/Remove User:**

   This module will be used by the administration to add or remove users in their respective courses in the cloud database.

10. **User Interface:**

    User interface is one of the ways to interact with application. It packages all those screens, dialogs and forms that are visible to user.

Structure and Relationships

This section covers the overall technical description of Smart Classroom powered by IoT. It shows the working of application in perspective of different point-of-views and shows relationship between different components.

System Block Diagram

This diagram shows the higher-level description of the application. It shows all the modules of the system and their associations and flow of data between modules.

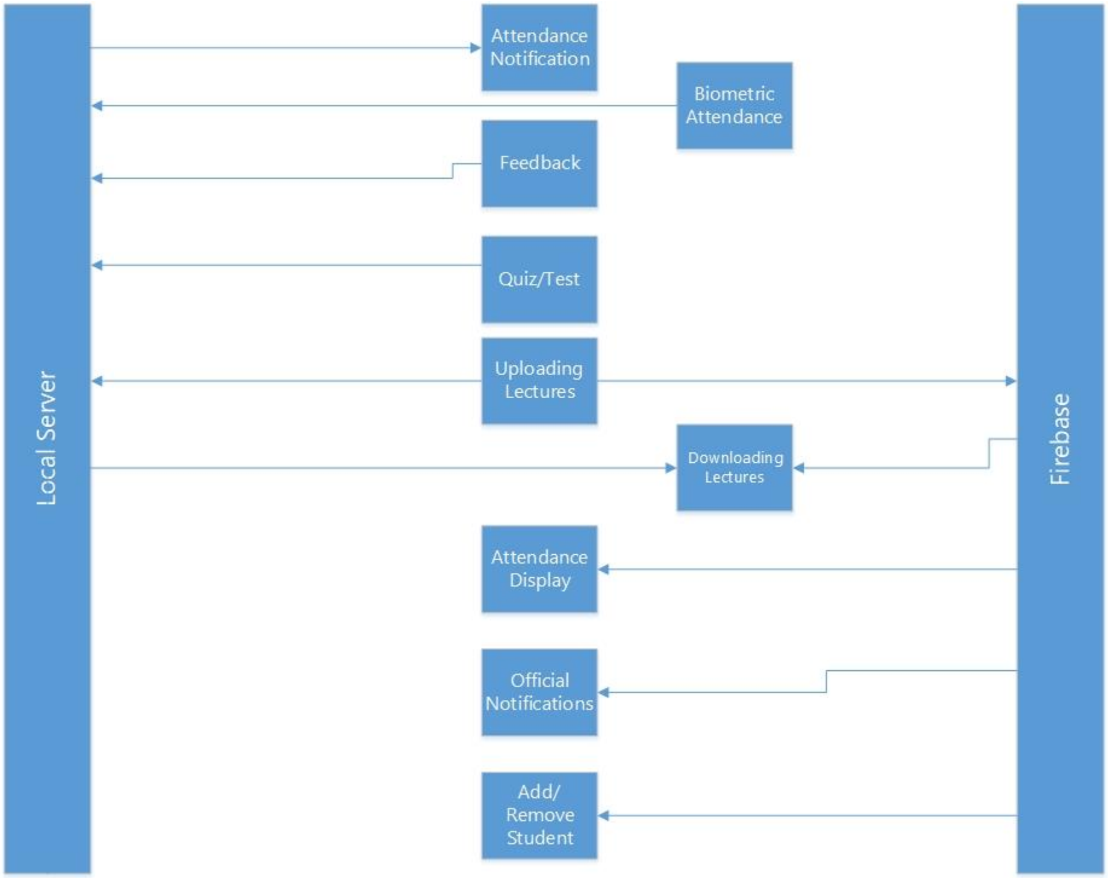

Figure 4.2.1 System Block Diagram

In class, the local network will be in use and in any other place like home cloud network will be used after connecting to the internet. To mark the attendance, Biometric Attendance will get the fingerprint of the user and send it to the local network server (raspberry pi) to verify and mark the attendance. Feedback will also be used in class to suggest or ask something from teacher anonymously or otherwise. Just like Feedback, Quiz module will also be working in class as test can only happen in the class. The teacher can create it for the class and the students should take the quiz. Uploading Lectures will be used by teacher in or outside the class to upload the lectures while Downloading Lectures will be working to download the lectures in or outside the class. Attendance Notification will be working in class after the attendance. Official Notifications, Attendance Display and Add/Remove Users will only work on cloud.

User View (Use case diagram)

Following diagram shows course of events that take place when an actor (user and other allowed interactions) interact with the system.

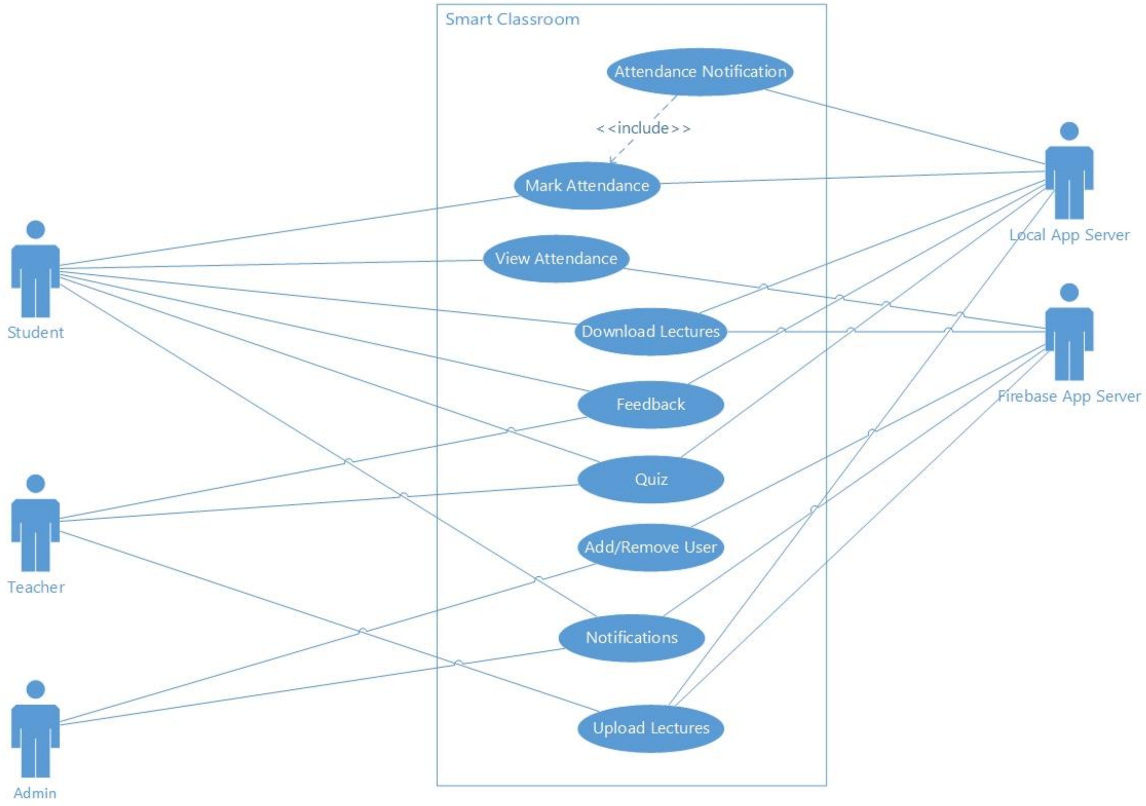

Figure 4.2.2 System Use Case Diagram

Use cases shown in the figure above are described below.

Use Case 1: Mark Attendance

Use Case Requirement: Students will verify their identity in smartphone application, once the student is in proximity of class, raspberry pi (local network) will then mark the attendance of the user on the cloud.
Use Case Paths: • Normal: Attendance marked. • Exceptional: Attendance not marked.

Normal Path: Attendance marked
Preconditions: The students verify their identity from fingerprint sensor of smartphone.
Interactions: A notification is sent to the raspberry pi (local network) for processing.
Post conditions: Attendance of the student is marked.
Exceptional Path: Attendance is not marked
Preconditions: The student is not connected to network or the identity of student is not verified.
Interaction: An error notification is sent to the raspberry pi (local network).
Post conditions: Student is notified about attendance not being marked.

Use Case 2: Attendance Notification

Use Case Requirement: Once the attendance has been marked, local app server will get the list of absent students and then send an email to the administration and guardians of all absent students.
Use Case Paths: • Normal: Email sent. • Exceptional: Email not sent.
Normal Path: Email sent
Preconditions: The application gets the list of absent students.
Interactions: An email is generated for administration and every student’s guardian to be sent.

Post conditions: Email is sent to all the recipients.
Exceptional Path: Email not sent
Preconditions: The local app server is not connected to network.
Post conditions: Email is not sent.
Include: Mark Attendance

Use Case 3: Download Lectures

Use Case Requirement: Students will download the lectures or relevant materials shared by the teacher during or outside the class.
Use Case Paths: • Normal: Lecture Downloaded. • Exceptional: Lecture Not Downloaded.
Normal Path: Lecture downloaded
Preconditions: The student will be connected to local or cloud network.
Interactions: Student will tap on the relevant lecture or notes from the Resource section to download the available material.
Post conditions: Lecture will be downloaded on student’s smartphone.
Exceptional Path: Lecture not downloaded
Preconditions: The student is not connected to network or there is no lecture available for download.

Post conditions: Student gets a message about no lecture available for download.

Use Case 4: Feedback

Use Case Requirement: Students will be giving feedback during or after the class in case of any query or suggestion.
Use Case Paths: • Normal: Feedback submitted. • Exceptional: Feedback not submitted.
Normal Path: Feedback submitted
Preconditions: The student will be connected to local or cloud network.
Interactions: Student will tap on the Feedback section to give any suggestion or query and tap on submit button.
Post conditions: Feedback will be submitted to relevant teacher.
Exceptional Path: Feedback not submitted
Preconditions: The student is not connected to network.
Post conditions: Student gets an error message.

Use Case 5: Quiz

Use Case Requirement: Teachers will be taking MCQs based quizzes from students within the classroom.
Use Case Paths:

• Normal (teacher): Quiz created. • Exceptional (teacher): Quiz not created. • Normal (student): Quiz submitted. • Exceptional (student): Quiz not submitted.
Normal Path (teacher): Quiz created
Preconditions: The teacher will create a quiz for the class.
Interactions: The teacher will make it public (if in class) or hidden (if not in class) and submit it to cloud.
Post conditions: Quiz will be created.
Exceptional Path (teacher): Quiz not created
Preconditions: Teacher is not connected to network or all the required fields are not completed.
Post conditions: Teacher gets an error message.
Normal Path (student): Quiz submitted
Preconditions: The student will attempt the quiz.
Interactions: After completing the quiz, student will tap on submit button to submit it to cloud for processing.
Post conditions: Quiz will be submitted and result will be displayed.
Exceptional Path (teacher): Quiz not created
Preconditions: Student is not connected to network or is not within premises of the classroom.

Post conditions: Student gets an error message.

Use Case 6: Add/Remove User

Use Case Requirement: Admin will be able to add or remove users from courses in the cloud database.
Use Case Paths: • Normal: User added or removed. • Exceptional: User addition or removal fails.
Normal Path: User added or removed
Preconditions: Admin will add or remove the user in the database by filling the details of user.
Interactions: System will check for the records and process accordingly.
Post conditions: User will be added or removed from the database.
Exceptional Path: User addition or removal fails
Preconditions: Admin is not connected to network or the record is already present in the database (in case of addition) or there is no record (in case of removal).
Post conditions: Admin gets an error message.

Use Case 7: Notifications

Use Case Requirement: Admin or teachers will be able to send important notifications or announcements related to exams or class or other important things to the students.

Use Case Paths: • Normal: Notification Sent. • Exceptional: Notification Failed.
Normal Path: Notification Sent
Preconditions: The user will be connected to local or cloud network.
Interactions: Admin or teacher will login at firebase portal to write and send push notifications.
Post conditions: Push notifications will be delivered to the respective persons.
Exceptional Path: Notification Failed
Preconditions: The user is not connected to network.
Post conditions: User gets an error message.

Use Case 8: Upload Lectures

Use Case Requirement: Teacher will upload the lectures or relevant materials for the students during or outside the class.
Use Case Paths: • Normal: Lecture Uploaded. • Exceptional: Lecture Not Uploaded.
Normal Path: Lecture uploaded
Preconditions: Teacher will be connected to local or cloud network.

Interactions: Teacher will tap on the upload lecture option and choose the file to upload it.
Post conditions: Lecture will be uploaded to the cloud app.
Exceptional Path: Lecture not uploaded
Preconditions: Teacher is not connected to network or no file was selected for uploading.
Post conditions: Teacher gets an error message.

Sequence Diagrams

Following sequence diagrams show the sequence of activities performed in application.

Mark Attendance

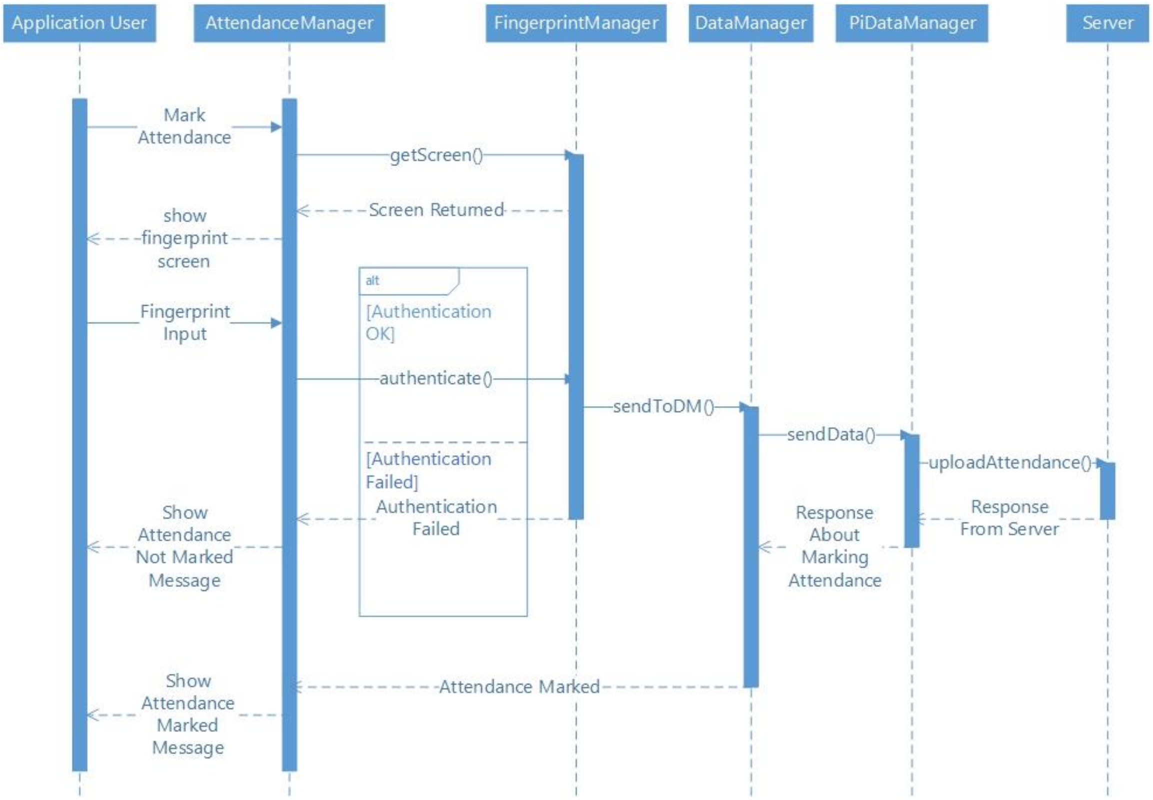

Figure 4.2.3 Sequence – Mark Attendance

View Attendance

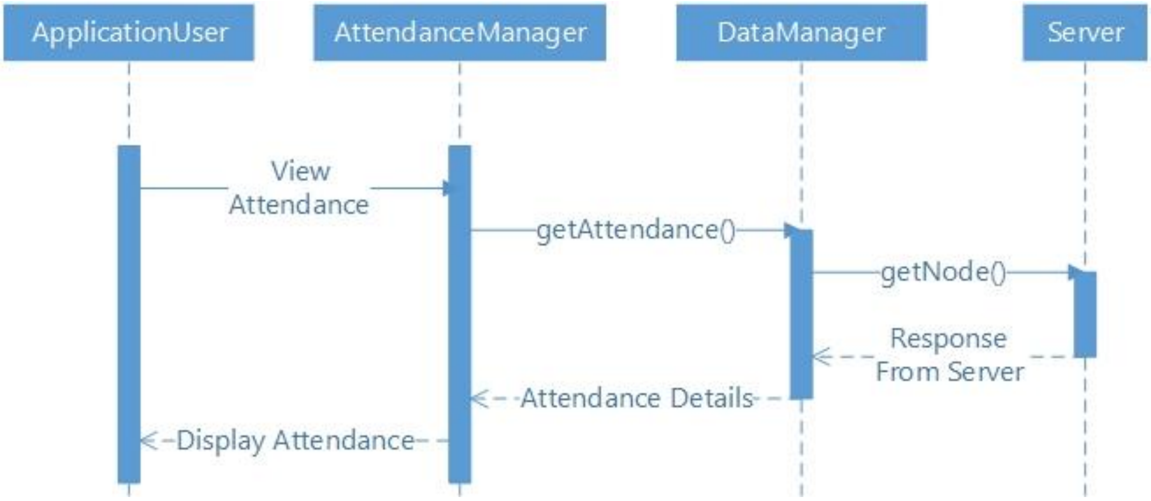

Figure 4.2.4 Sequence – View Attendance

Download Lectures

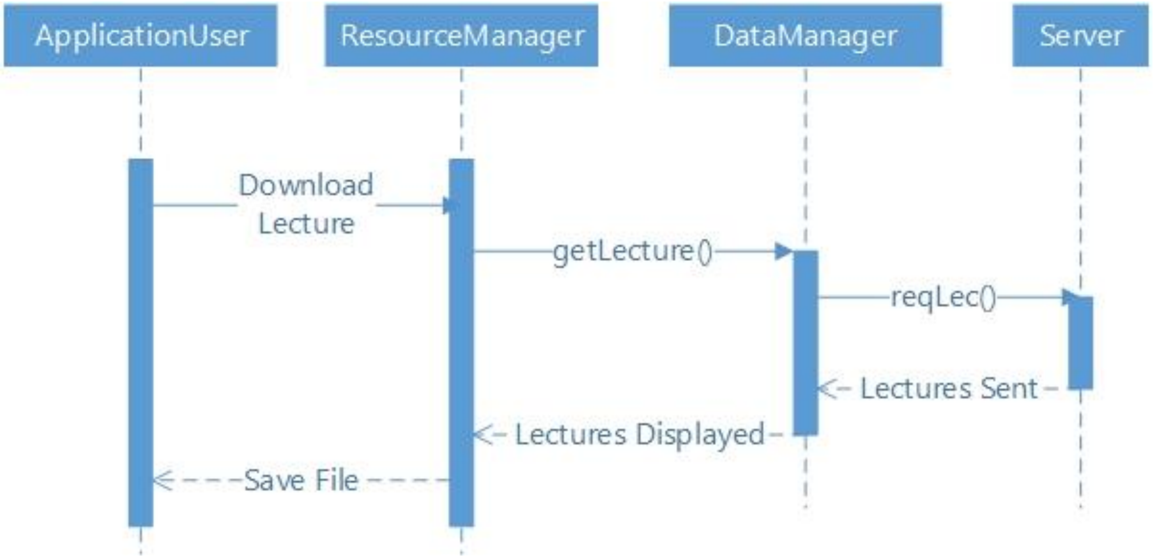

Figure 4.2.5 Sequence – Downoad Lectures

Upload Lectures

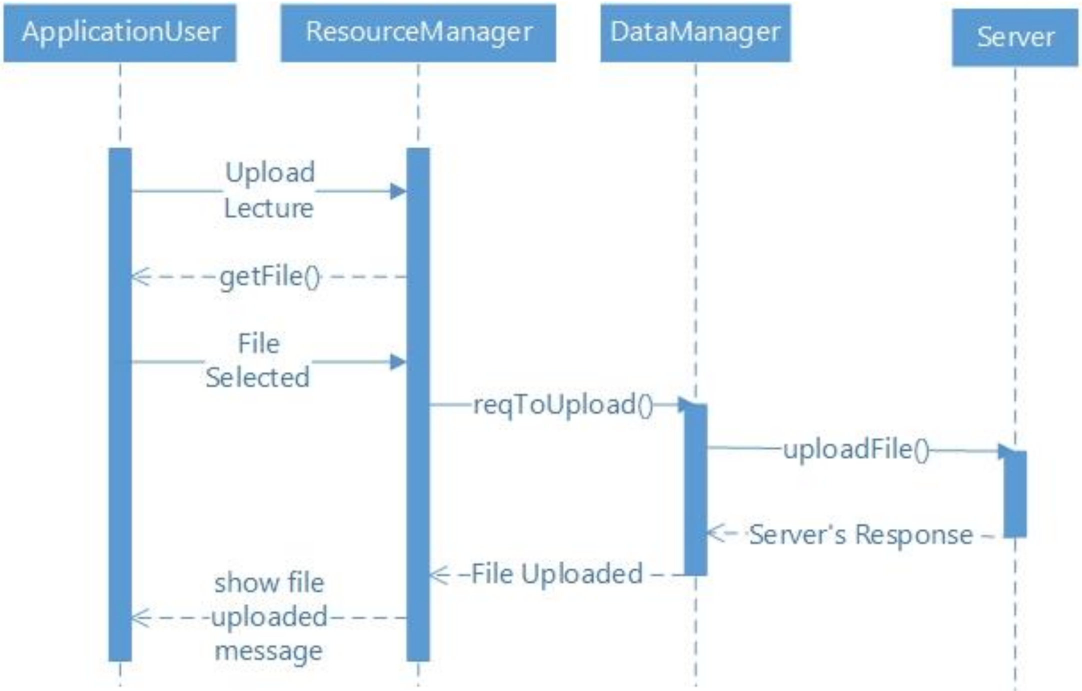

Figure 4.2.6 Sequence – Upload Lectures

Feedback

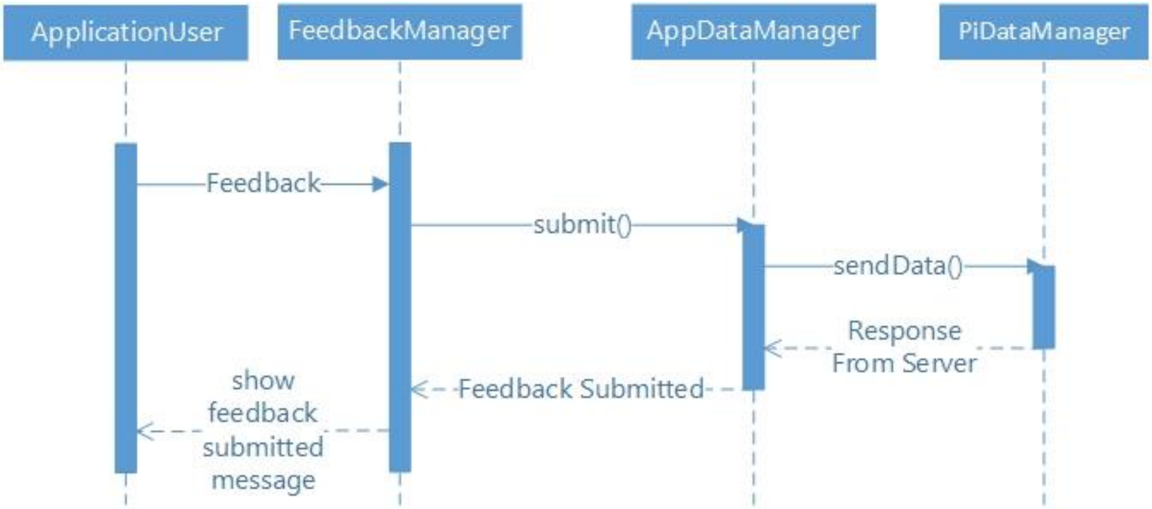

Figure 4.2.7 Sequence – Feedback

Quiz (Teacher End)

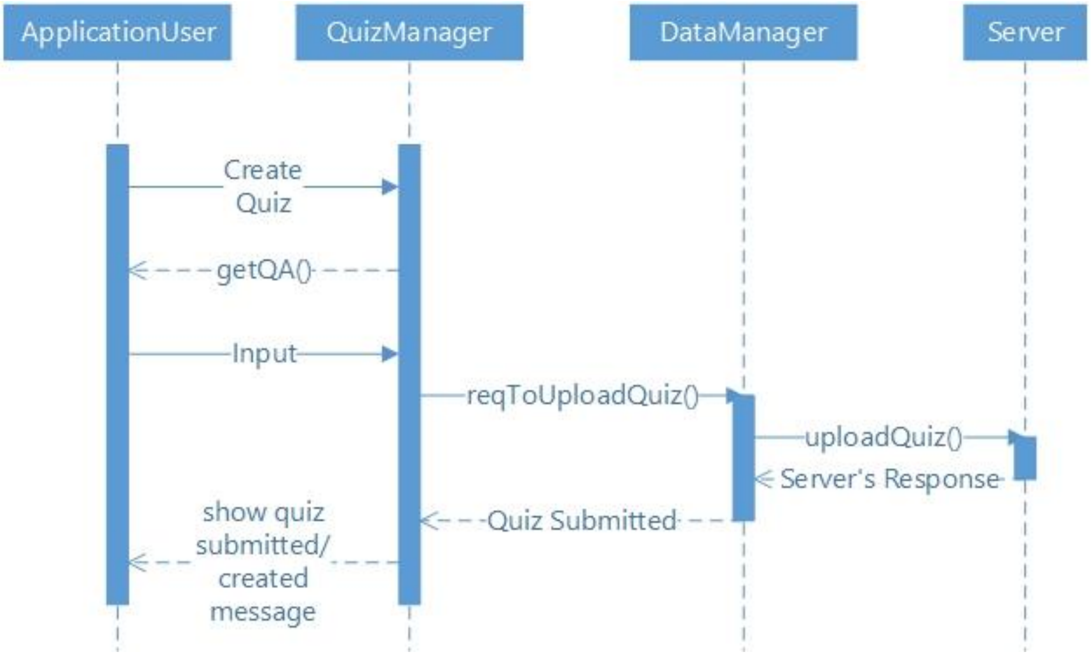

Figure 4.2.8 Sequence – Quiz (Teacher End)

Quiz (Student End)

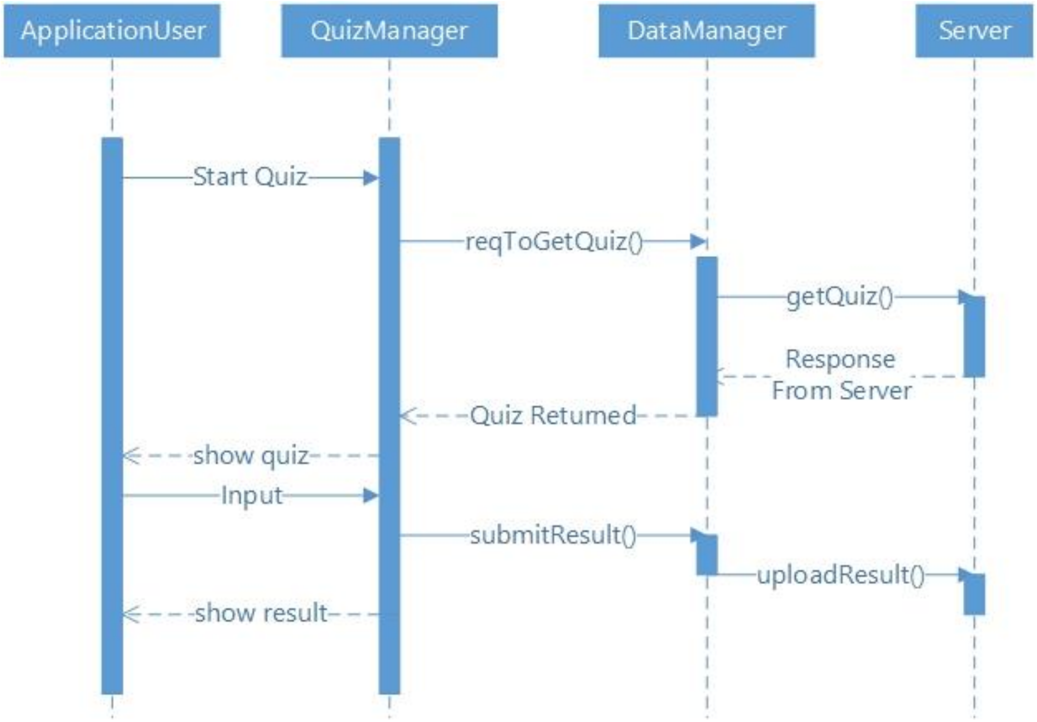

Figure 4.2.9 Sequence – Quiz (Student End)

Notifications

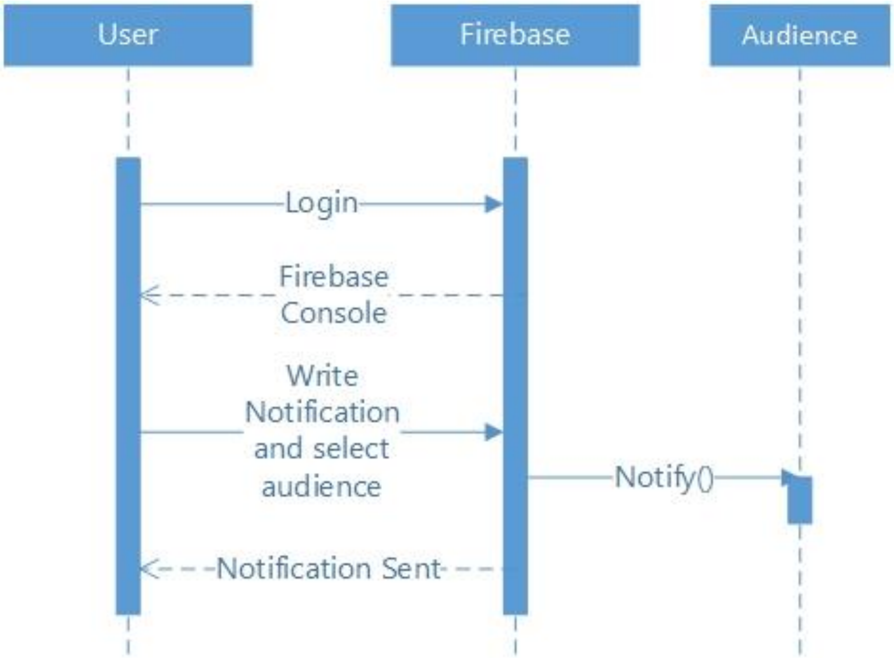

Figure 4.2.10 Sequence – Notifications

Add Users

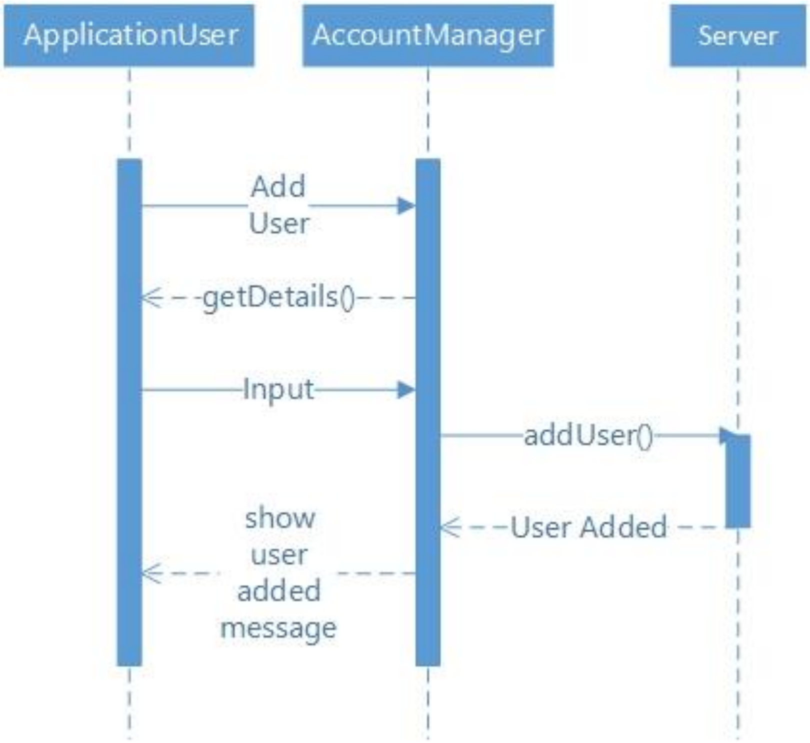

Figure 4.2.11 Sequence – Add Users

Remove Users

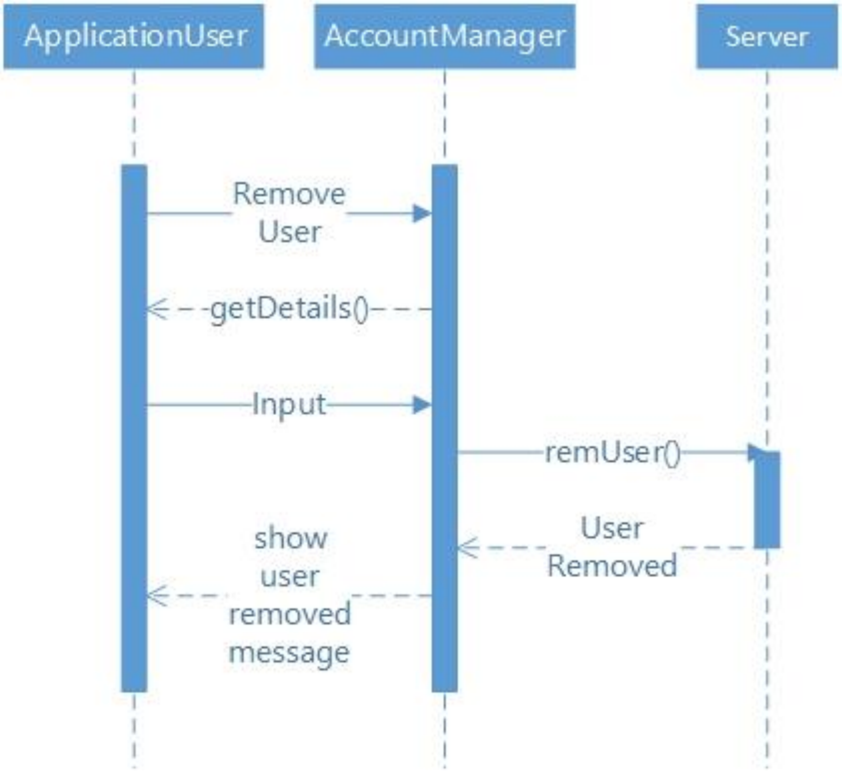

Figure 4.2.12 Sequence – Remove Users

Implementation View (Class Diagram)

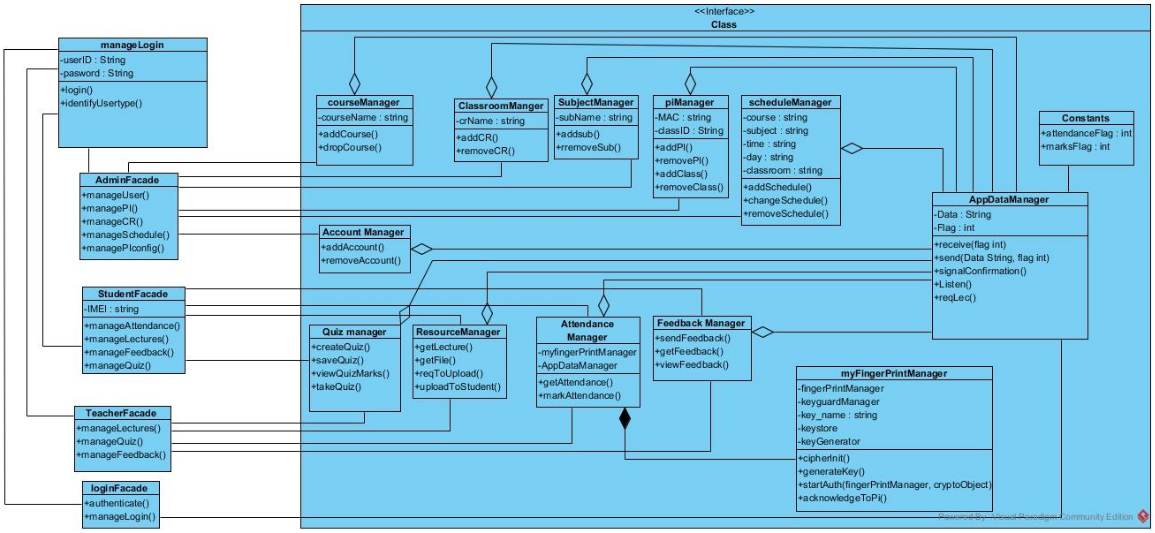

Figure 4.2.13 System Class Diagram

Smart Application Class name	**Description**
ManageLogin	This is the class that handles login for a user.
StudentFacade	This is a facade class that exposes all the methods of the complex subsystem that student can use to carry out all his/her required functionalities.
AdminFacade	This is a facade class that exposes all the methods of the complex subsystem that admin can use to carry out all his/her required functionalities.
TeacherFacade	This is a façade class that exposes all the methods of the complex subsystem that Teacher can use to carry out all his/her required functionalities.
LoginFacade	This is the facade class that enables the user to login by exposing only the required class of the subsystem.
Attendance Manager	The attendanceManager will authenticate the user first with myFingerPrintManager and then mark the attendance of student using appDataManager.
AppDataManager	The role of this class is to enable communication between android app and raspberry pi.
FeedbackManager	The role of this class is to let the student to give the Feedback to teacher and enable the teacher to view Feedback.

QuizManager	quizManager class allows the teacher to create quiz and view quiz while a student can take quiz and view quiz marks.
AccountManager	The role of this class is to manage user accounts. A user can create accounts and remove accounts with this class.
myFingerPrintManager	This class takes finger print from user and authenticate it and takes a new fingerprint from user.
resourceManager	This class manages the content that is uploaded by the teachers and allows the student to download lectures.
ClassRoomManager	It manages adding and deleting class room entries like cr-35 in the firebase database using AppDataManager.
courseManager	It manages entering and deleting course entries/nodes like 19b in Firebase database using AppDataManager.
scheduleManager	It manages entering and deleting course schedule entries/nodes for week in Firebase database using AppDataManager.
subjectManager	It manages entering and deleting course subject entries/nodes like SE-381 SQE in Firebase database using AppDataManager.
constants	It is the class that contains all the constants that are needed within the application as well as those that are needed to make

	raspberry pi understand what type of data should it expect, over local network.
PiManager	It is the class that helps admin to add a raspberry pi entry in firebase or remove an existing one. Also, it helps admin to define which class schedule should a raspberry pi maintain.
User	It denotes the generic user who will eventually interact with the application. It contains generic information about the user to help identify not only the user but device.
Student	This class extends user class. The role of this class is to show the activities performed by the student.
Admin	Admin class extends user class. The role of Admin class is to Add/Remove Users from the Database.
Teacher	Teacher class also extends user class. Teacher class can upload lectures, create quiz and view Feedback by calling its methods.

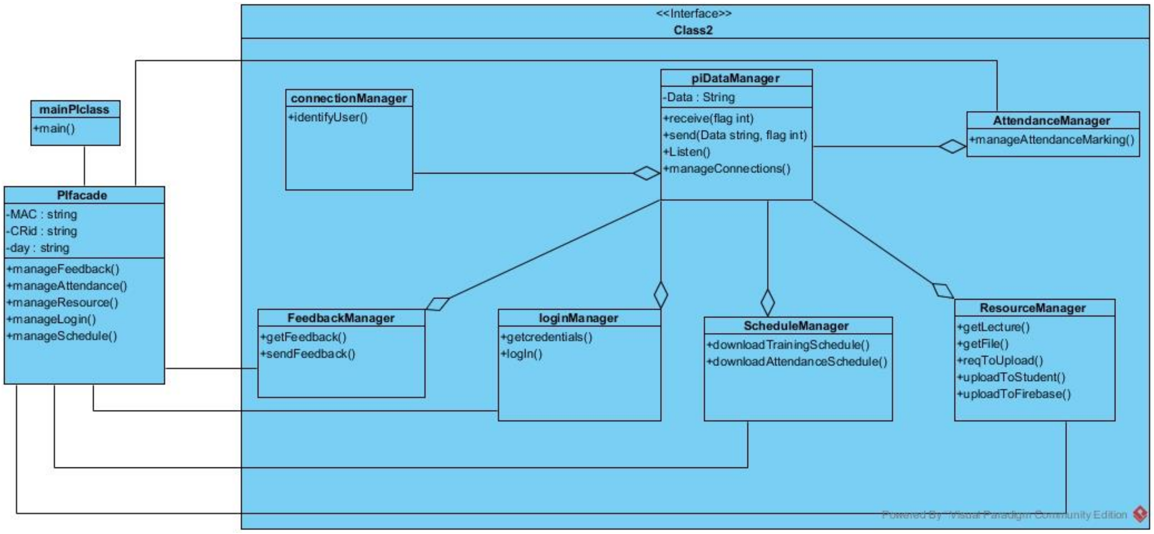

Figure 4.2.14 System Class Diagram 2

PI class name	**Description**
mainPIclass	It is the main class that starts when the raspberry pi is turned on
PiFacade	This is the facade class that enables the raspberry pi to act as a server and perform all the required tasks automatically by exposing certain parts of the subsystem
AttendanceManager	The class that helps students in marking their attendance using raspberry pi
loginManager	It assists in logging in users (students and teachers) via raspberry pi
scheduleManager	This is the class that helps in downloading and maintaining schedule of pi of a classroom throughout the day
resourceManager	This class assists teachers in sending lectures to students via raspberry pi. It

	also uploads lecture to cloud as well as it assists students in downloading lecture.
connectionManager	It identifies all established connections and differentiates between connections from teachers and students.

Dynamic View (Activity Diagram)

In activity diagram, the dynamic view of the system is shown. All the activities are shown.

Student View

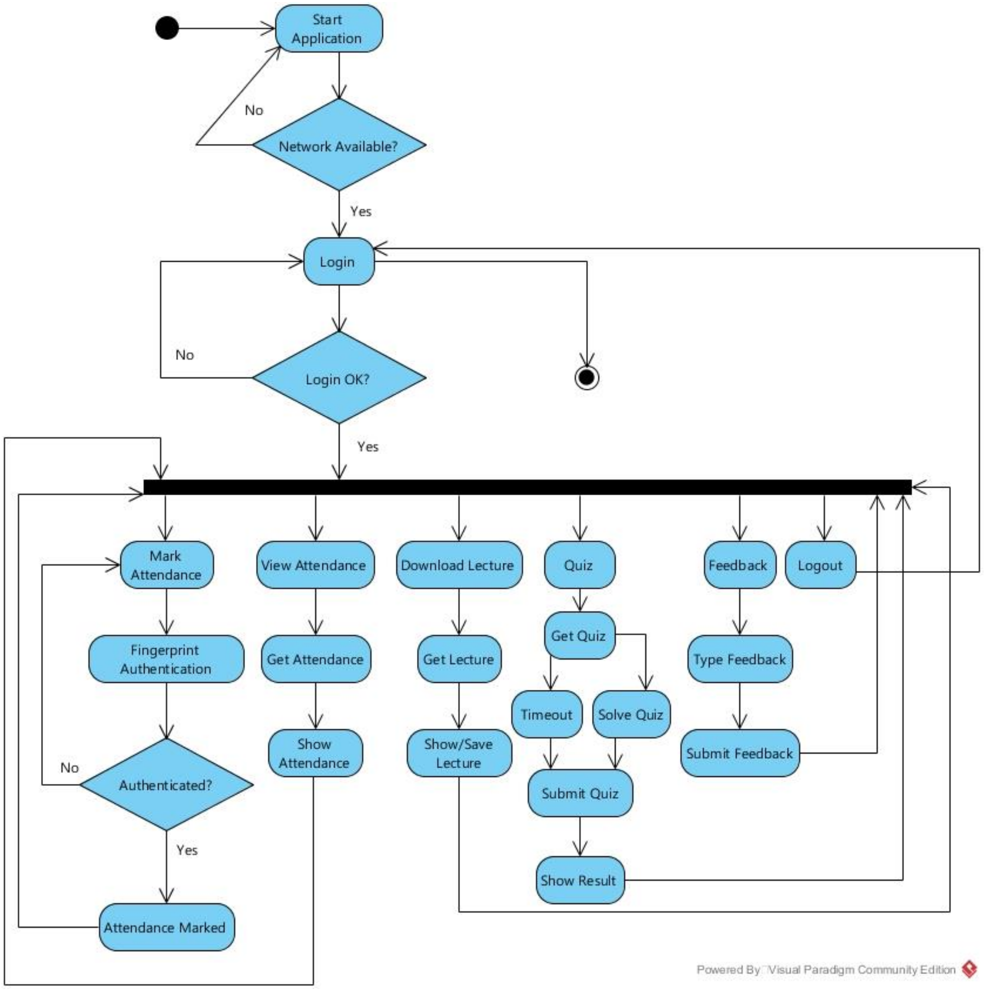

Figure 4.2.15 Activity – Student

Teacher View

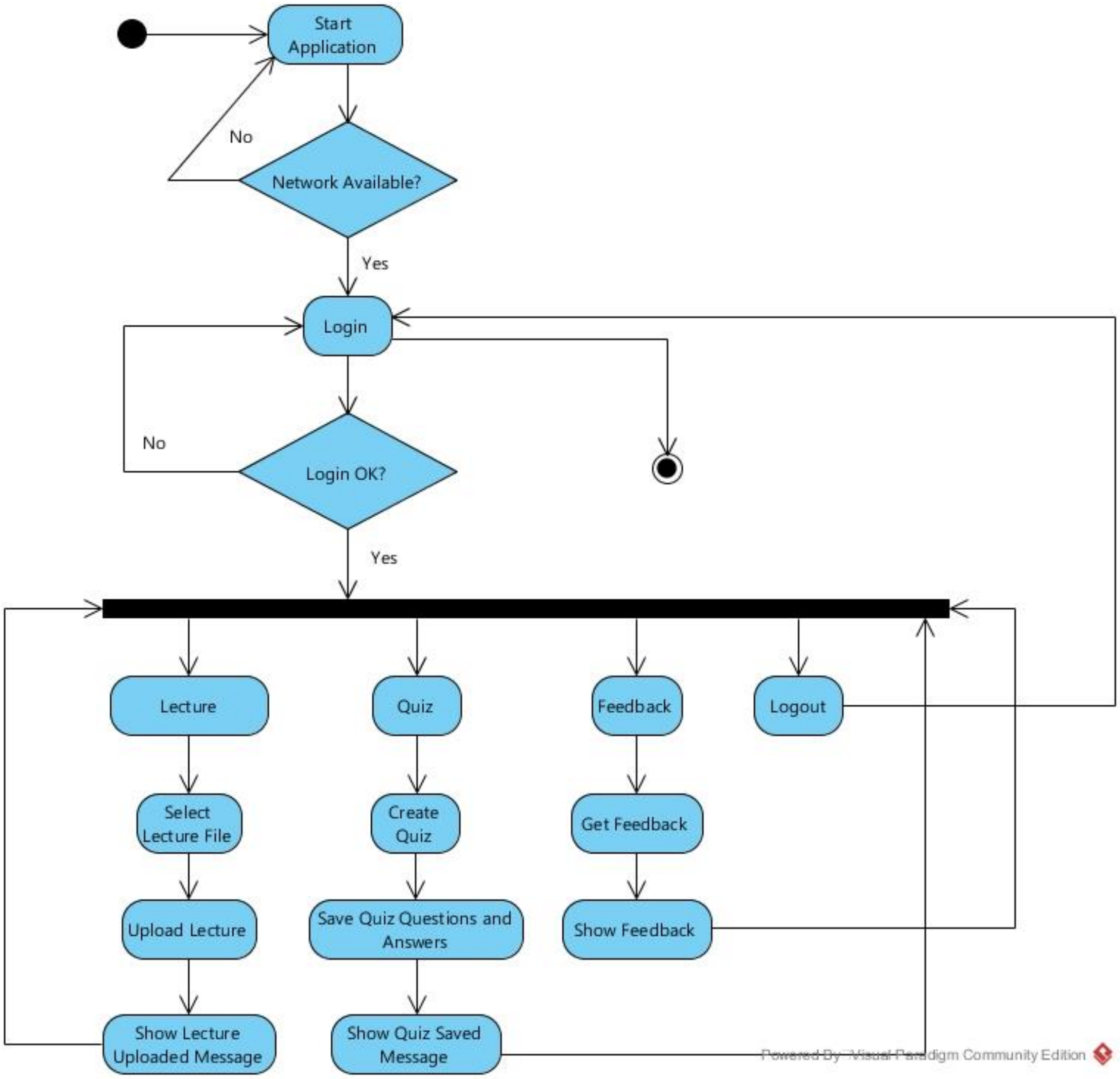

Figure 4.2.16 Activity – Teacher

Admin View

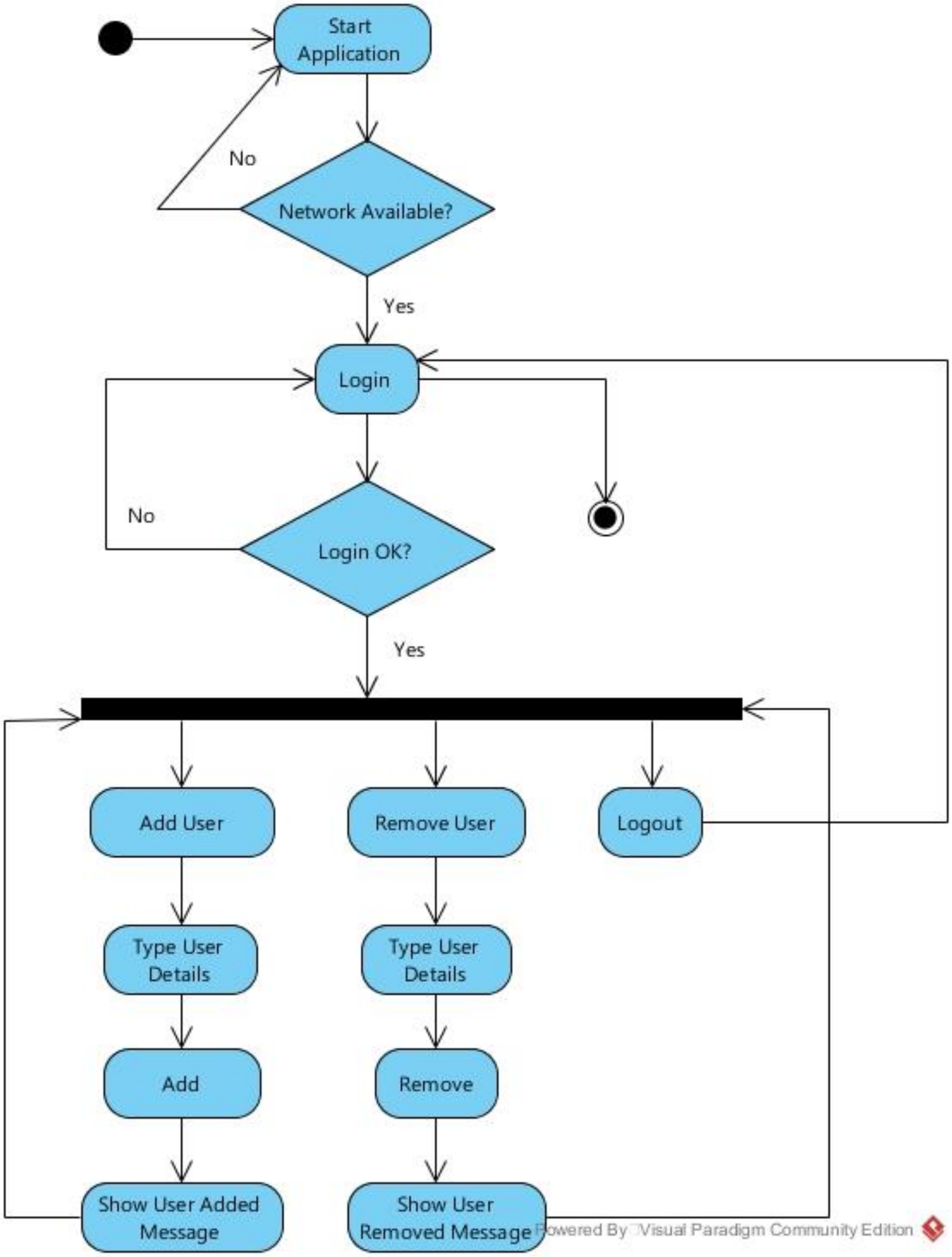

Figure 4.2.17 Activity – Admin

State Transition Diagrams

In this section, state transition of application is shown how it changes to another state.

Student View

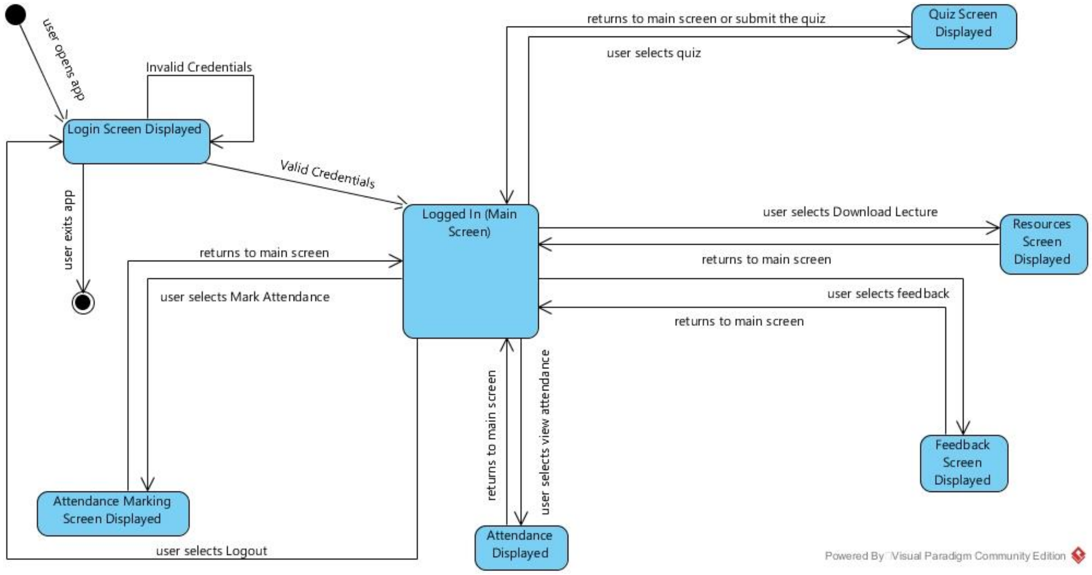

Figure 4.2.18 State Transition – Student

Teacher View

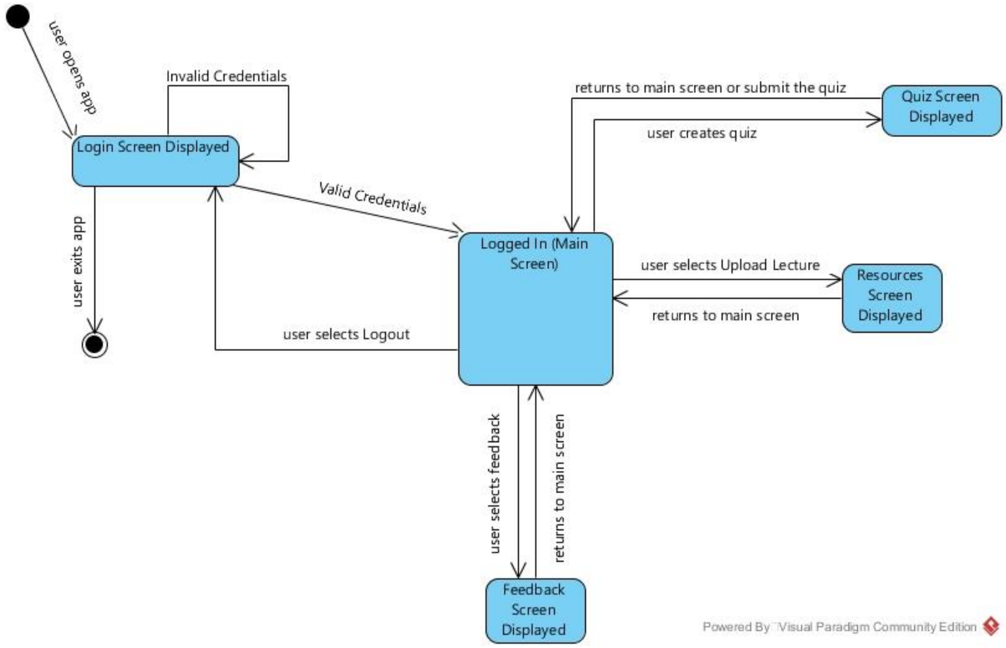

Figure 4.2.19 State Transition – Teacher

Admin View

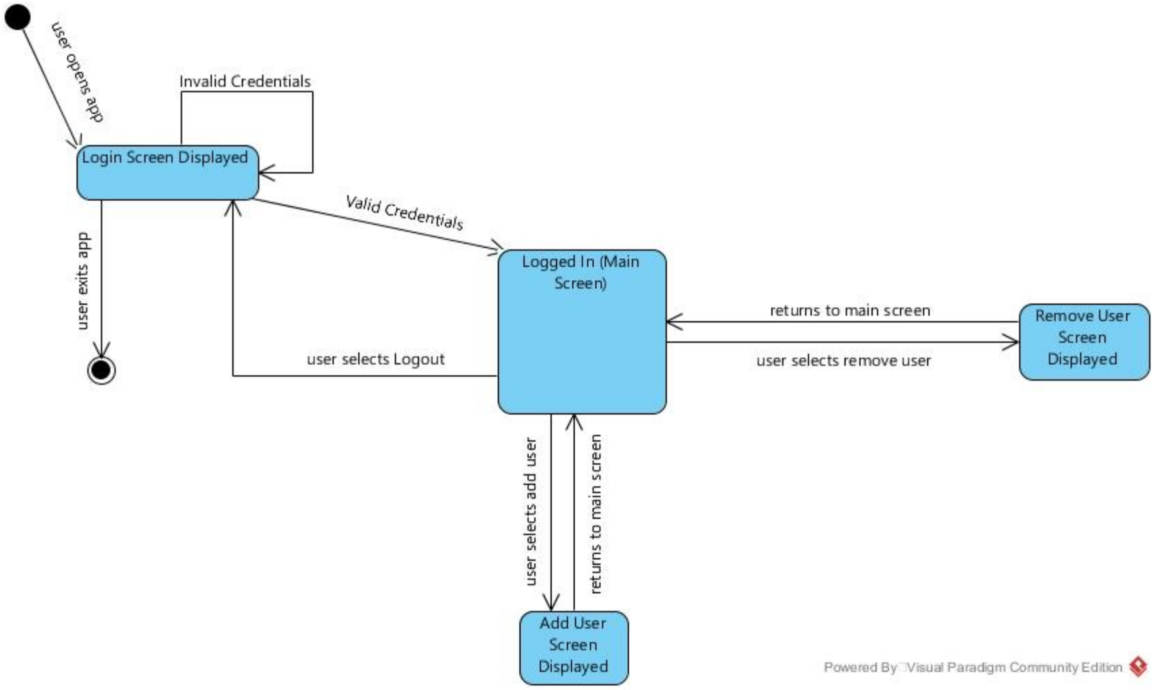

Figure 4.2.20 State Transition – Admin

4.3. Detailed Description of Components

This section describes in detail all components of Smart Classroom powered by IoT application.

Biometric Attendance

Identification	Name: Biometric Attendance
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **3.1 Biometric Attendance** This feature allows a student to mark his/her student.

	Description: When a class will be about to start, students will use their smartphone to mark their attendance using the fingerprint sensor of their smartphone which will then be uploaded to the cloud.
Function	This module marks the attendance of a student using biometric verification on smartphone and then uploads the attendance on the network.
Subordinates	It has one subordinate: 1. Biometric Attendance: Requirement 3.1 in SRS
Dependencies	This component is independent module and runs in parallel to entire application.
Interfaces	This component has following interfaces: **1. Network Interface:** To interact with raspberry pi to update attendance and confirm the presence of student in class. **2. Application UI Interface:** To get fingerprint verification and inform about the status of attendance.
Resources	**Hardware:** Fingerprint enabled smartphone, raspberry pi **Software:** Open source android libraries, Android SDKs, Python libraries, Firebase SDK
Processing	Biometric Attendance runs when the students give their thumb impression on the smartphone and after authentication it requests raspberry pi to update the attendance on cloud.
Data	This component uses following information of the application: 1. Device Info

	2. Student Info 3. Data Flags

Attendance Display

Identification	Name: Attendance Display
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **3.2 Attendance Display** This feature provides students the ability to see their attendance from the android application. **Description:** When a student will go to the attendance section of the application, he/she will be able to see his/her attendance of a specific class.
Function	This module shows the attendance of a student after getting it from cloud, once a student requests to view it.
Subordinates	It has one subordinate: 1. Attendance Display: Requirement 3.2 in SRS
Dependencies	This component is independent module and runs in parallel to entire application.
Interfaces	This component has following interfaces: **1. Network Interface:** To interact with cloud for getting student's attendance details.

	2. **Application UI Interface:** To display the attendance of student.
Resources	**Hardware:** Smartphone **Software:** Open source android libraries, Android SDKs, Firebase SDK
Processing	Students login from the application and tap on Attendance to view their attendance which is acquired from the cloud.
Data	This component uses following information of the application: 1. Student Info

Uploading Lectures

Identification	Name: Uploading Lectures
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **3.3 Uploading Lectures or Relevant Material** This feature will provide teachers the ability to upload lectures or relevant material to the cloud or raspberry pi for the availability in the android application. **Description:** User must be able to upload the lecture or relevant material for the class to the cloud or raspberry pi.
Function	This module allows the teacher to upload any lecture or relevant file for a class to cloud or raspberry pi for sharing it with students.

Subordinates	It has one subordinate: 1. Uploading Lectures or Relevant Material: Requirement 3.3 in SRS
Dependencies	This component is independent module and runs in parallel to entire application.
Interfaces	This component has following interfaces: **1. Network Interface:** To interact with cloud or raspberry pi for sending the files over network. **2. Application UI Interface:** To upload the relevant lecture file.
Resources	**Hardware:** Smartphone, Raspberry pi **Software:** Open source android libraries, Android SDKs, Python libraries, Firebase
Processing	Teachers login from the application and tap on Upload Resource section after which they select the relevant files and it is uploaded to raspberry pi or firebase cloud depending on teacher’s login which will either be via raspberry pi or cloud. If the file is sent to raspberry pi, raspberry pi will upload it to the cloud and makes it available for students to download.
Data	This component uses following information of the application: 1. Teacher Info 2. Class Info 3. Lecture File

Download Lectures

Identification	Name: Download Lectures

Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **3.4 Download Lectures or Relevant Material** This feature allows students to download lectures or relevant material. **Description:** User must also be able to download files from the android application.
Function	This module allows the student to download any lecture or relevant file for a class from the application which is uploaded by teacher.
Subordinates	It has one subordinate: 1. Download Lectures or Relevant Material: Requirement 3.4 in SRS
Dependencies	This component is independent module and runs in parallel to entire application.
Interfaces	This component has following interfaces: **1. Network Interface:** To interact with cloud or raspberry pi for receiving the files over network. **2. Application UI Interface:** To download the relevant lecture file.
Resources	**Hardware:** Smartphone, Raspberry pi **Software:** Open source android libraries, Android SDKs, Python libraries, Firebase

Processing	Students login from the application and tap on Download Resource section after which they select the relevant files which are available for download and it is downloaded to their smartphone device.
Data	This component uses following information of the application: 1. Student Info 2. Class Info

Official Notifications

Identification	Name: Official Notifications
Type	Service
Purpose	This service fulfils following requirement from Software Requirements Specification Document: **3.5 Official Notifications** This feature allows administration to send official notifications. **Description:** User must also be able to send the official notifications related to university.
Function	This module allows the administration to send any official notification or announcement related to classes or university through the Firebase platform.
Subordinates	It has one subordinate: 1. Official Notifications: Requirement 3.5 in SRS

Dependencies	This service is independent module and runs in parallel to entire application.
Interfaces	This service has following interfaces: **1. UI Interface:** To send notifications to users.
Resources	**Hardware:** Computer **Software:** Firebase
Processing	Admin logs in at Firebase platform, navigates to notifications section and writes and sends a notification to all or specific users.
Data	This component uses following information: 1. Student Info 2. Class Info 3. Teacher Info 4. Admin Info

Feedback

Identification	Name: Feedback
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document:

	3.6 Feedback This feature allows students to give feedback about a teacher/class anonymously. **Description:** User must also be able to give feedback about a teacher or class anonymously using the android application.
Function	This module allows the student to provide feedback or any suggestion about a class to the teacher anonymously or otherwise.
Subordinates	It has one subordinate: 1. Feedback: Requirement 3.6 in SRS
Dependencies	This component is independent module and runs in parallel to entire application.
Interfaces	This component has following interfaces: **1. Network Interface:** To interact with cloud or raspberry pi for sending feedback over network. **2. Application UI Interface:** To type and submit feedback.
Resources	**Hardware:** Smartphone, Raspberry pi **Software:** Open source android libraries, Android SDKs, Python libraries
Processing	Students tap on Feedback to type any suggestion or query and then submit it to raspberry pi and raspberry pi sends it to the teacher.
Data	This component uses following information of the application: 1. Student Info

	2. Class Info 3. Teacher Info

Quiz

Identification	Name: Quiz
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **3.7 Test/Quiz** This feature allows teachers to take quiz from the class and students to submit quiz. **Description:** A teacher can create the quiz for a class which can be started on a specified time and then the students can submit the quiz at the specified time and after the submission their result is shown.
Function	This module allows the teacher to create a MCQs based quiz for a class and save it (hide it) before the specified time of class and then at the time of class, teacher can publish the quiz and students can start quiz from their application. At the end of submission, result will be shown to the student
Subordinates	It has one subordinate: 1. Test/Quiz: Requirement 3.7 in SRS
Dependencies	This component is independent module and runs in parallel to entire application.

Interfaces	This component has following interfaces: 1. **Network Interface:** To interact with raspberry pi for sending and receiving quiz over network. 2. **Application UI Interface:** To create quiz (for teachers) and to submit quiz and show results (for students).
Resources	**Hardware:** Smartphone, Raspberry pi **Software:** Open source android libraries, Android SDKs, Python libraries, Firebase
Processing	This module works when teacher creates a quiz. The quiz can be saved on teacher's device for future usage in class. Student can start quiz when teacher makes it available and once completed, student will submit the quiz and the result will be displayed.
Data	This component uses following information of the application: 1. Student Info 2. Class Info 3. Teacher Info 4. Quiz Info

Attendance Notification

Identification	Name: Attendance Notification
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **3.8 Attendance Notification**

	If the student is absent, an email notification will be sent. **Description:** If the student is absent from the class, an email will be sent to his/her guardian and administration.
Function	This module sends an email to administration and guardian of absent students after the attendance of a class is marked.
Subordinates	It has two subordinate: 1. Biometric Attendance: Requirement 3.1 in SRS 2. Attendance Notification: Requirement 3.8 in SRS
Dependencies	This component is dependent on Biometric Attendance component as it collects the info about absent students after the attendance has been marked.
Interfaces	This component has following interfaces: **1. Network Interface:** To interact with raspberry pi to get info about absent students and then email to respective people.
Resources	**Hardware:** Raspberry pi **Software:** Open source android libraries, Android SDKs, Python libraries, Firebase SDK
Processing	This module starts working after the Biometric Attendance of a class is completed. It collects information of the absent students and then send emails to their guardians and administration.
Data	This component uses following information of the application: 1. Attendance Info 2. Student Info

	3. Admin and Guardian Info

Add/Remove User

Identification	Name: Add/Remove User
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **3.9 Add/Remove User** Admin will be able to add or delete users in the cloud database. **Description:** Admin will be able to add/delete users in the cloud database of so that they can access the app through their login ids.
Function	This module allows admin to add or remove users (students and teachers) in respective courses for accessing the application and its features.
Subordinates	It has one subordinate: 1. Add/Remove User: Requirement 3.9 in SRS
Dependencies	This component is independent module and runs in parallel to entire application.
Interfaces	This component has following interfaces: **1. Application UI Interface:** To add/remove users in cloud database.

Resources	**Hardware:** Smartphone **Software:** Open source android libraries, Android SDKs, Firebase
Processing	Admin logs in from application, navigates to add/remove user. Admin will be removing or adding user depending on user type such as student or teacher by filling their respective details in their respective courses
Data	This component uses following information of the application: 1. Student Info 2. Course Info 3. Teacher Info

User Interface

Identification	Name: User Interface
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **3.1 Biometric Attendance** This feature allows a student to mark his/her student. **Description:** When a class will be about to start, students will use their smartphone to mark their attendance using the fingerprint sensor of their smartphone which will then be uploaded to the cloud. **3.2 Attendance Display**

This feature provides students the ability to see their attendance from the android application.

Description: When a student will go to the attendance section of the application, he/she will be able to see his/her attendance of a specific class.

3.3 Uploading Lectures or Relevant Material

This feature will provide teachers the ability to upload lectures or relevant material to the cloud or raspberry pi for the availability in the android application.

Description: User must be able to upload the lecture or relevant material for the class to the cloud or raspberry pi.

3.4 Download Lectures or Relevant Material

This feature allows students to download lectures or relevant material.

Description: User must also be able to download files from the android application.

3.5 Official Notifications

This feature allows administration to send official notifications.

Description: User must also be able to send the official notifications related to university.

3.6 Feedback

This feature allows students to give feedback about a teacher/class anonymously.

Description: User must also be able to give feedback about a teacher or class anonymously using the android application.

3.7 Test/Quiz

	This feature allows teachers to take quiz from the class and students to submit quiz. **Description:** A teacher can create the quiz for a class which can be started on a specified time and then the students can submit the quiz at the specified time and after the submission their result is shown. **3.9 Add/Remove User** Admin will be able to add or delete users in the cloud database. **Description:** Admin will be able to add/delete users in the cloud database of so that they can access the app through their login ids
Function	User interface is one of the ways to interact with application. It packages all those screens, dialogs and forms that are visible to user. User interface is user friendly and easy to understand.
Subordinates	This component has following subordinates: 1. Biometric Attendance: Requirement 3.1 in SRS 2. View Attendance: Requirement 3.2 in SRS 3. Uploading Lectures or Relevant Material: Requirement 3.3 in SRS 4. Download Lectures or Relevant Material: Requirement 3.4 in SRS 5. Official Notifications: Requirement 3.5 in SRS 6. Feedback: Requirement 3.6 in SRS 7. Test/Quiz: Requirement 3.7 in SRS 8. Add/Remove User: Requirement 3.9 in SRS
Dependencies	Working of this component is dependent on integration of all other components.

Interfaces	**N/A**
Resources	**Hardware:** Smartphone, Computer **Software:** Open Source Android libraries, Android SDKs, Firebase
Processing	User Interface display info, notifications and messages to user passed by other components.
Data	This component uses following information of the application: 1. Device Info 2. Network Info 3. Internet Status 4. User Info

4.4. Reuse and Relationships to other Products

Smart Classroom powered by IoT is not based on any previous systems neither it's an extension of any other applications at any level, there are some applications with the name Smart Classroom but they differ in many ways like some apps only offer notes taking, some only offer video streaming. It can be evolved into a bigger and more complex system with more features and functionality. Developers can also reuse some of the modules of the system. The practical usage of the system can be increased by adding more and more services to system.

The application can also be used by changing in some of its modules like biometric attendance through fingerprint devices instead of smartphones, advanced quiz system like descriptive tests instead of MCQs.

4.5. Design Decisions and Tradeoffs

The project requires client-server architecture. The smartphone app acts as a client and raspberry pi (local server) and firebase (internet/cloud server) are acting as servers. There are two scenarios being followed in the project. One is inside the class, where the smartphone app acts as a client and raspberry pi acts as a server, both are connected through LAN. In the other scenario, which is outside the class, smartphone app acts as a client and firebase acts as a server which are connected through internet.

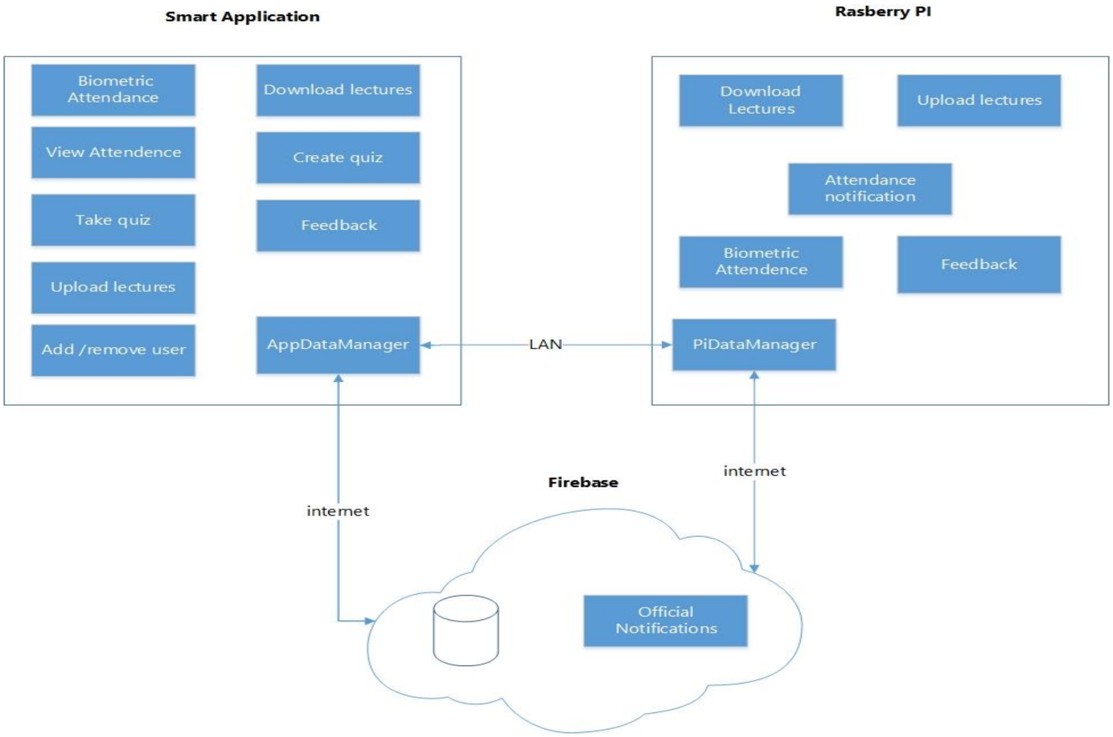

Figure 4.5.1 Architecture Diagram

We are using the Façade design pattern as our system was becoming very complex at the implementation level because of the three roles which use either different or shared parts of the subsystem. That's why a façade class has been introduced for each of these, i.e. student, teacher and an admin which exposes only the required functionalities of the underlying subsystem.

The constraints of our project involve:

- The schedule provided to us will be fixed and without any conflict
- Number of connections over WiFi or Bluetooth are limited
- One raspberry pi is limited to only one class (fixed)
- Every student should have fingerprint enabled smartphone

Chapter 5. Project Test and Evaluation

5.1. Introduction

This test plan document describes the appropriate process and methods used to plan testing of the Smart Classroom powered by IoT project. The test plan will ensure that Smart Classroom powered by IoT works as intended without any failure.

We are performing manual testing of the project i.e.; no tool or script is being utilized for testing purpose. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. First of all, each unit (module) will be tested separately and then whole project will be tested after integration of all the units.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed.

This document includes the plan, scope, approach and procedure of Smart Classroom powered by IoT test. The pass/fail criteria of the test items are also defined. The Test Plan document documents and tracks the necessary information required to effectively define the approach to be used in the testing of the product.

5.2. Test Items

Based on the requirements of Smart Classroom powered by IoT, design description, modules of android application, python application, and non-functional scenario will be tested. The requirements defined in Software Requirements Specification and the design entities as explained in Software Design Document will be tested.

5.3. Features to Be Tested

Following features are being tested:

1. Ability to add/remove courses, accounts, subjects and schedule in the cloud database through android application.
2. Ability to send push notifications from the firebase console.
3. Ability to mark attendance via raspberry pi using biometric authentication on the android application.

4. Ability to view attendance from the cloud database on the android application.
5. Ability to upload lectures using cloud database or raspberry pi from android application.
6. Ability to download lectures using cloud database or raspberry pi on android application.
7. Ability to share feedback through android application over raspberry pi.

Ability to create and take quiz using the android application over raspberry pi.

5.4. Test Approach

The project is using waterfall approach, with producing modules and integrating them. All the modules are tested individually and then integrated with system and integration tests are applied.

Black Box testing technique will be used for testing functionality of each module.

Unit Testing

Unit testing is that part of testing which requires a thorough check of each module of the project. In our project, there are 8 modules which we must check if they are functioning normally or not. For this, we will start from a unit which is least dependent on other modules for its function and then work our way through to the module which requires all the rest to function and test.

Integration Testing

Integration testing is the part where we will test all the previous tested modules in a way that they are functioning normally when they are combined.

System Testing

In the end, system testing will ensure that all the modules are working, separately and together combined. Then only the outcome of the program will decide the correctness of whole system.

5.5. Item Pass/Fail Criteria

Details of the test cases are specified in the section Test Deliverables. Following the principles outlined below, a test item would be judged as pass or fail.

- Preconditions are met
- Inputs are carried out as specified

- The result works as what specified in output => Pass

The system doesn’t work or not the same as output specification => Fail

5.6. Suspension Criteria and Resumption Requirements

Testing will be suspended when a defect is introduced/found that cannot allow any further testing. Testing will be resumed after defect removal.

5.7. Test Deliverables

Test case name	**Login to Respective Account**
Test Case Number	1
Description	The actors who can login are student, faculty, and admin.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network. 2) Account must exist in the cloud database.
Input	User name and Password.
Steps	1. Enter Username as username-fm for faculty member, username-admin for admin, and username-course (amukhtar-19b) for student. 2. Enter Password 3. Tap on ‘LOGIN VIA CLOUD’ or ‘LOGIN VIA PI’ button.
Expected output	User should be logged in respective account and their respective screen should be displayed. Admin should not be able to login using Raspberry Pi

Actual output	Logged In

Test case name	**Add Accounts**
Test Case Number	2
Description	The admin is the only actor that can access the privilege to add accounts.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network. 2) Actor must be admin or have all the access that an admin has and account shouldn’t already exist. 3) Admin should be logged in via cloud.
Input	Account Type and course if any, User name, Name, and Password.
Steps	4. Select Type and Course if any. 5. Enter Username 6. Enter Name 7. Enter Password 8. Tap on ‘CONFIRM’ button.
Expected output	Account should be added with confirmation message
Actual output	Confirmed

Test case name	**Modify Accounts**
Test Case Number	3
Description	The admin is the only actor that can access the privilege to modify accounts.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network. 2) Actor must be admin or have all the access that an admin has and the account must exist already. 3) Admin should be logged in via cloud.
Input	Account Type and course if any, User name, Name, and Password.
Steps	1. Select Type and Course if any. 2. Enter Username of the user to be modified 3. Enter New Name 4. Enter New Password 5. Tap on ‘CONFIRM’ button.
Expected output	Account should be modified with confirmation message
Actual output	Confirmed

Test case name	**Delete Accounts**
Test Case Number	4

Description	The admin is the only actor that can access the privilege to delete accounts.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network. 2) Actor must be admin or have all the access that an admin has and the account must exist already. 3) Admin should be logged in via cloud.
Input	Account Type and course if any, and User name.
Steps	1. Select Type and Course if any. 2. Enter Username 3. Tap on ‘CONFIRM’ button.
Expected output	Account should be deleted with confirmation message
Actual output	Confirmed

Test case name	**Add Courses**
Test Case Number	5
Description	The admin is the only actor that can access the privilege to add courses.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network.

	2) Actor must be admin or have all the access that an admin has and the course shouldn't exist already. 3) Admin should be logged in via cloud.
Input	Course Name
Steps	1. Enter Name 2. Tap on 'ADD' button.
Expected output	Course should be added with confirmation message
Actual output	Added

Test case name	**Add Classrooms**
Test Case Number	6
Description	The admin is the only actor that can access the privilege to add classrooms.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network. 2) Actor must be admin or have all the access that an admin has and the classroom shouldn't exist already. 3) Admin should be logged in via cloud.
Input	Classroom Name
Steps	1. Enter Name 2. Tap on 'ADD' button.

Expected output	Classroom should be added with confirmation message
Actual output	Added

Test case name	**Add Subjects**
Test Case Number	7
Description	The admin is the only actor that can access the privilege to add subjects.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network. 2) Actor must be admin or have all the access that an admin has and the subject shouldn't exist already. 3) Admin should be logged in via cloud.
Input	Course, Subject Name, and ID.
Steps	1. Select Course 2. Enter Name 3. Enter Course ID 4. Tap on 'ADD SUBJECT' button.
Expected output	Subject should be added with confirmation message
Actual output	Confirmed

Test case name	**Add Raspberry Pis**
Test Case Number	8
Description	The admin is the only actor that can access the privilege to add raspberry pis.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network. 2) Actor must be admin or have all the access that an admin has and the raspberry pi entry shouldn’t exist already. 3) Admin should be logged in via cloud.
Input	MAC of Raspberry PI and Classroom.
Steps	1. Enter MAC 2. Select Classroom 3. Tap on ‘ADD’ button.
Expected output	Raspberry PI should be added with confirmation message
Actual output	Added

Test case name	**Add Schedule**
Test Case Number	9
Description	The admin is the only actor that can access the privilege to add schedule.

Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network. 2) Actor must be admin or have all the access that an admin has and the schedule shouldn't exist already. 3) Admin should be logged in via cloud.
Input	Classroom, Day, Course, Start Time, End Time, Subject, and Faculty.
Steps	1. Select Classroom 2. Select Day 3. Select Course 4. Enter Start Time 5. Enter End Time 6. Select Subject 7. Select Faculty 8. Tap on 'CONFIRM' button.
Expected output	Schedule should be added with confirmation message
Actual output	Done

Test case name	**Delete Schedule**
Test Case Number	10

Description	The admin is the only actor that can access the privilege to delete schedule.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network. 2) Actor must be admin or have all the access that an admin has and the schedule should exist already. 3) Admin should be logged in via cloud.
Input	Classroom and Day.
Steps	1. Select Classroom 2. Select Day 3. Tap on 'CONFIRM' button.
Expected output	Schedule should be deleted with confirmation message
Actual output	Done

Test case name	**Send Notifications**
Test Case Number	11
Description	The admin is the only actor that can access the privilege to send notifications.
Testing Technique used	Black Box Testing
Preconditions	1) Device should be connected to a network.

	2) Actor must be admin or have all the access that an admin has and should have access to Firebase.
Input	Message, Delivery Date, App, Audience, Version, Language, Title, Targeted user, and Notification to be sent.
Steps	1. Select delivery date 2. Select application 3. Select Audience 4. Select app version 5. Select language 6. Enter Title of Notification 7. Enter Targeted User 8. Enter Notification Message 9. Select Priority 10. Select Expiry 11. Click on 'SEND MESSAGE' button.
Expected output	Notification should be sent with confirmation message
Actual output	Confirmed

Test case name	**Upload Files**
Test Case Number	12
Description	The teacher/faculty member is the only actor that can access the privilege to upload files.

Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network. 2) Actor must be teacher/faculty member or have all the access that a teacher has and a course and subject must exist for sharing a file. 3) Teacher should be logged in via cloud or raspberry pi.
Input	Course, Subject, and File
Steps	1. Select Course 2. Select Subject 3. Select File 4. Tap on ‘CONFIRM’ button.
Expected output	File should be added with confirmation message
Actual output	Database Entry Added

Test case name	**Delete Files**
Test Case Number	13
Description	The teacher/faculty member is the only actor that can access the privilege to delete files.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network.

	2) Actor must be teacher/faculty member or have all the access that a teacher has and file must exist which is to be deleted. 3) Teacher should be logged in via cloud or raspberry pi.
Input	Course, Subject, and File
Steps	1. Select Course 2. Select Subject 3. Select File 4. Tap on ‘CONFIRM’ button.
Expected output	File should be deleted with confirmation message
Actual output	Delete Successful

Test case name	**Download Files**
Test Case Number	14
Description	The student is the only actor that can access the privilege to download files.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network. 2) Actor must be student or have all the access that a student has and a file should be available for download. 3) Student should be logged in via cloud or raspberry pi.
Input	Tap the ‘DOWNLOAD’ button.

Steps	1. Tap on ‘DOWNLOAD’ button of the required file.
Expected output	File should be downloaded with confirmation message
Actual output	Download Complete

Test case name	**Mark Attendance**
Test Case Number	15
Description	The student is the only actor that can access the privilege to mark attendance.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network. 2) Actor must be student and his/her fingerprint should be authenticated along with the IMEI of the smartphone. 3) It should be in proximity of Raspberry Pi. 4) The attendance module should be enabled at the time of class for marking the attendance. 3) Student should be logged in via raspberry pi.
Input	Identity Authentication from Fingerprint sensor.
Steps	1. Place finger or thumb on fingerprint sensor.
Expected output	Attendance should be marked with confirmation message

Test case name	**View Attendance**
Test Case Number	16
Description	The student is the only actor that can access the privilege to view attendance.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network. 2) Actor must be student and attendance should be marked of the respective subject before viewing. 3) Student should be logged in via cloud.
Input	Tap on ‘VIEW ATTENDANCE’ Button.
Steps	1. Select Course
Expected output	Attendance should be displayed.

Test case name	**Create Quiz**
Test Case Number	17
Description	The teacher is the only actor that can access the privilege to create a quiz.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network.

	2) Actor must be teacher or have all the access that a teacher has. 3) Teacher should be logged in via raspberry pi.
Input	Course, Subject, Time, MCQs.
Steps	1. Select Course 2. Select Subject 3. Select Time 4. Type MCQs (Questions and their options) 5. Type Correct Answers 6. Tap on “CONFIRM” button.
Expected output	Quiz should be created with confirmation message

Test case name	**Take Quiz**
Test Case Number	18
Description	The student is the only actor that can access the privilege to take a quiz.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network. 2) Actor must be student or have all the access that a student has. 3) Student should be logged in via raspberry pi.

Input	Select the options given for MCQs
Steps	1. Take Quiz 2. Select Answers 3. Tap on 'SUBMIT' button.
Expected output	Quiz should be submitted with confirmation message after tapping on SUBMIT button, or in case of timeout, or in case of shifting away from the application. And result will be shown.

Test case name	**Give Feedback**
Test Case Number	19
Description	The student is the only actor that can access the privilege to give feedback.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network. 2) Actor must be student or have all the access that a student has. 3) Student should be logged in via raspberry pi.
Input	Subject, and Typed Feedback.
Steps	1. Type Feedback. 2. Tap on "CONFIRM" button.
Expected output	Feedback should be submitted with confirmation message

Test case name	**View Feedback**
Test Case Number	20
Description	The teacher is the only actor that can access the privilege to view feedback.
Testing Technique used	Black Box Testing
Preconditions	1) Smartphone should be connected to a network. 2) Actor must be teacher or have all the access that a teacher has and a feedback should exist for a subject. 3) Teacher should be logged in via raspberry pi.
Input	Subject
Steps	1. Select Subject 2. Tap on "CONFIRM" button.
Expected output	Feedback should be displayed

5.8. Responsibilities, Staffing and Training Needs

Responsibilities

All developers of the project are responsible for the completion of all units testing and integration testing tasks.

Staffing and Training Needs

Basic knowledge of testing strategies and techniques is needed for the testing of project.

Techniques such as Black Box testing, integration testing should be known to developers.

All the developers will be testing each other’s work and will be actively participating in the development and testing of the project simultaneously.

5.9. Risk and Contingencies

Efforts have been made to remove all and every chance of failure but there are certain unpredictable factors such as network issues, corrupt input data, or system failure that may lead to some issues. Error handling will be applied more deeply to cover all these issues but unforeseen circumstances may happen.

Schedule Risk

The project might get behind schedule. So, to complete the project on time, we will need to increase the hours/day.

Budget Risk

The budget will be compensated by using less costly alternatives to fit the budget requirements.

Chapter 6. Future Work

This project can be used as a basis to understand and add features to make it into an even bigger and complex system, which can be modified per an institution's needs. Or it could continue growing as a generic software for universities, it can be commercialized, but first, it would need to be deployed at least at the department level here at MCS or in any other university who is willing to incorporate technology in the classrooms and reviewed via the Teachers and Students. Additional features like notes making, conducting examination, aid for real-time visualization of a concept or a theory can be added in the future work if any.

Bibliography

- Firebase Database REST API | Firebase. (n.d.). Retrieved from https://firebase.google.com/docs/reference/rest/database/
- Firebase Cloud Messaging HTTP Protocol | Firebase. (n.d.). Retrieved from https://firebase.google.com/docs/cloud-messaging/http-server-ref
- IEEE. (1998, December 29). 1233-1998 - IEEE Guide for Developing System Requirements Specifications. Retrieved from
- http://ieeexplore.ieee.org/document/741940/?tp=&isnumber=16016&arnumber=741940&punumber=5982
- Wikipedia. (2005, April 28). Sequence Diagram. Retrieved from https://en.wikipedia.org/wiki/Sequence_diagram
- Wikipedia. (2004, November 16). Block Diagram. Retrieved from https://en.wikipedia.org/wiki/Block_diagram
- Wikipedia. (2007, July 2). Internet of things. Retrieved from https://en.wikipedia.org/wiki/Internet_of_things
- Wikipedia. (2002, July 26). Facade pattern. Retrieved from https://en.wikipedia.org/wiki/Facade_pattern
- Socket Programming HOWTO. (n.d.). Retrieved from https://docs.python.org/2/howto/sockets.html

Appendix A. Glossary

Firebase:	The cloud platform being used
API:	Application Programming Interface
OS:	Operating System

Appendix B: Issues/Limitations

All possible issues have already been mentioned where required in the SRS. Any remaining ones are listed below:

1. The group shall try to match the features and NFRs as best as possible, however, like all software projects, any discrepancies are apologized for at this stage.
2. Feedback on requirements is expected from the users to help the group in improving the design and implementation of the project.