Software Design Document
Version 1.0
October 22, 2021

"**Apna Classroom**"
Online Classroom

Group 2
Aaryak Shah (2019IMT-001)
Ashes Mondal (2019IMT-021)
Subodh Rajpopat (2019IMT-103)

Submitted in partial fulfillment
Of the requirements of
ITIT-3103 Software Engineering

Table Of Contents

Table Of Figures

1. Introduction

The Apna Classroom system provides a secure platform for faculty and students to manage their classrooms. It allows teachers to create virtual classrooms for announcements, quizzes and assignments. Students are able to subscribe to classrooms to remain up-to-date on their class work and performance. Classrooms are paired with their individual video conferencing rooms for convenience of the users. Many present technologies can be leveraged for this project. The client application will make use of ***React JS*** as the GUI framework to let us construct a good design. ***SCSS*** will be used for generating style sheets. The system's backend will utilise technologies like ***Node.js*** and ***Express.js*** to handle the API endpoints, as well as a ***NoSQL*** database like ***MongoDB*** to store data. The project aims to cover all the basic features for a classroom management system. It will deliver a backend system that is capable of handling aforementioned features, as well as a GUI client application to facilitate easy interaction with the system. The product is limited to cater to only users within the client's domain.

2. Design Considerations

The design considerations describe some of the issues that need to be considered and discussed before preparing the complete system design. This will refer to the SRS document to resolve ambiguities and problems.

2.1. Assumptions

This system makes a few assumptions regarding the software's requirements that need to be taken into consideration.

- The login credentials of the students are already made available by the administration, and no registration is required.
- Each batch of students can receive their individual classrooms and no combined batches are made.
- Only one faculty is required to be assigned to any given classroom.
- Student roll numbers are always unique and can be used to derive their email id.
- Roll numbers always follow a fixed format.
- Student attendance is taken manually.

2.2. Constraints

The online classroom portal requires a sufficient level of security. User authentication and authorization needs to be done by means of a secure JWT system.

The UI needs to display a modern and helpful design that makes using the software convenient.

The system should be resistant to injection attacks and so should use a NoSQL solution.

2.3. Design Methodology

The Apna Classroom software is being developed under an Agile framework. The software should thus be designed keeping in mind the possibility of changes to the software.

Designs should be made in a modular fashion, which will allow for rapid deployment and modification.

The database should be designed as a NoSQL database which will allow for easy changes to the schema and reduce time spent on database maintenance.

The API endpoints should be designed as independent controller functions that can operate on the database. A RESTful approach will be used.

2.4. System Environment

The system needs to be hosted on the institute server and hence must be lightweight and flexible for easy deployment. The system will be used only within the institute and so scalability demands are low.

[P.T.O]

3. Architecture

3.1. System Design

The Apna Classroom system should follow the following design in terms of the Component-Connection structure as well as mapping.

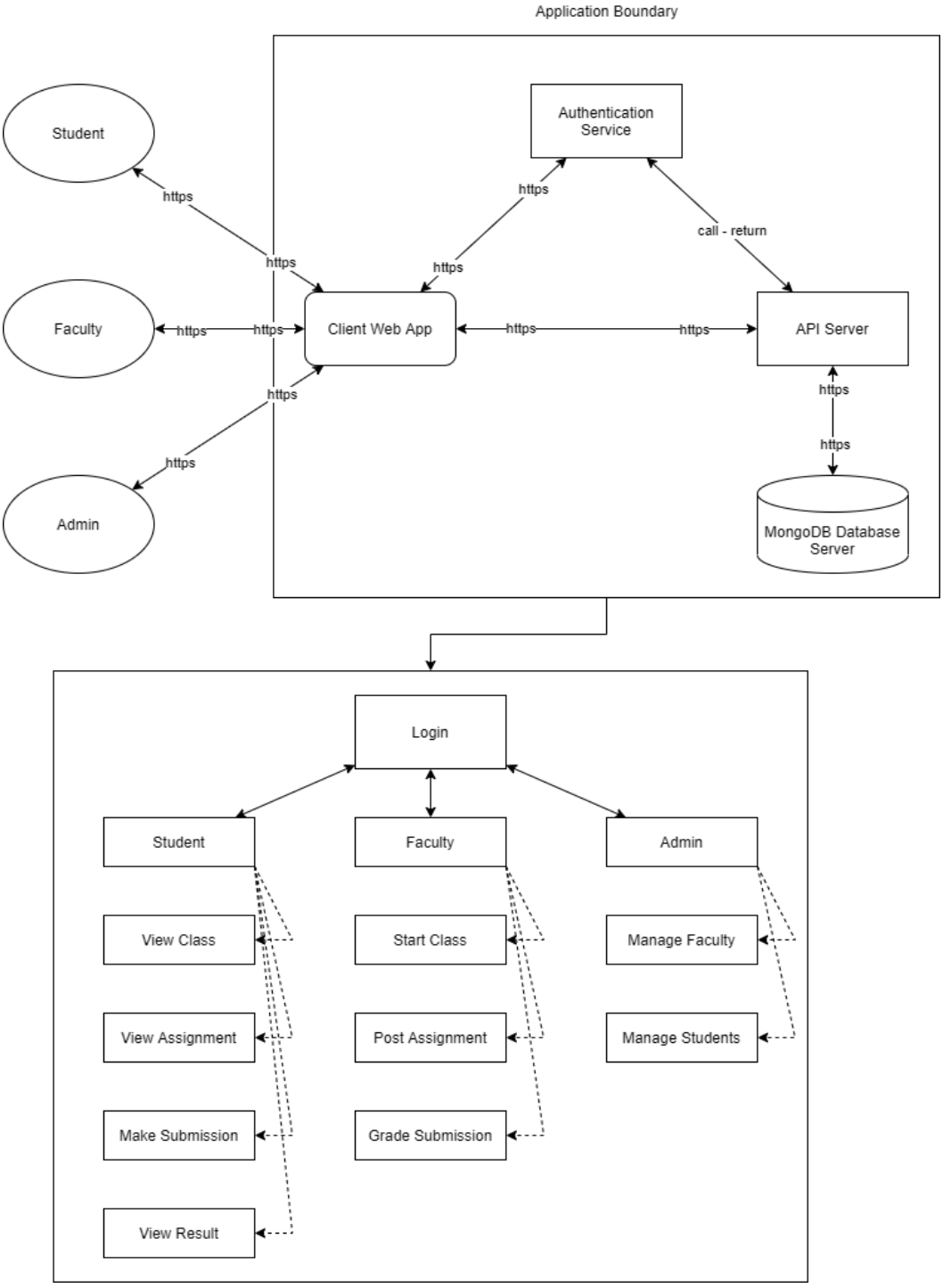

Figure 1. System Design

3.2. Functional Decomposition Tree

The major functional modules of the software as described by the system design are divided into multiple independent sub-functions. This will describe the organization of the code in its implementation. It should be ensured that there is high cohesion and low coupling between the modules.

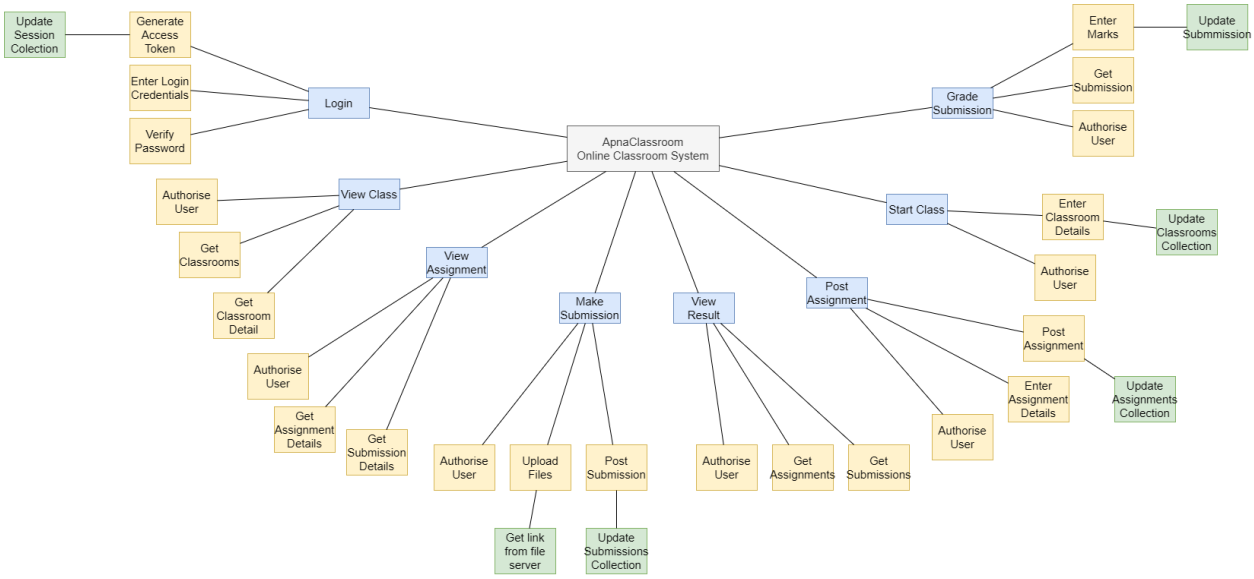

Figure 2. Functional Decomposition Tree

The modules described by the tree are as follows:

1. Login: Authenticate user and assign an access token

- ***1.1. Generate Access Token:*** Create a JWT for user
  - ***1.1.1 Update Session Collection:*** Add user session to database
- ***1.2. Enter Login Credentials:*** Enter email and password in UI
- ***1.3. Verify Password:*** Check if password hash matches DB

2. View Class: Display current classroomin UI

- ***2.1. Authorize User:*** Ensure JWT is valid
- ***2.2. Get Classrooms:*** Find classrooms this user is enrolled in from DB
- ***2.3. Get Classroom Detail:*** Obtain details of current class from classrooms

3. View Assignment: Display current assignment in UI

- ***3.1. Authorize User:*** Ensure JWT is valid

3.2. Get Assignment Details: Find assignment details from DB

3.3. Get Submission Details: Find submission data for user for assignment

4. Make Submission: Make a new submission for an assignment

4.1. Authorize User: Ensure JWT is valid

4.2. Upload Files: Get submission files from user and send to file server

4.2.1 Get Link from File Server: Obtain access link to uploaded file

4.3. Post Submission: Make submission with current date and details

4.3.1 Update Submissions Collection: Add submission to DB

5. View Result: Make a new submission for an assignment

5.1. Authorize User: Ensure JWT is valid

5.2. Get Assignments: Get assignments for this user for this classroom

5.3. Get Submissions: Get submission details for each assignment

6. Post Assignment: Make a new assignment for a classroom

6.1. Authorize User: Ensure JWT is valid

6.2. Enter Assignment Details: Enter assignment data in UI

6.3. Post Assignment: Make assignment with due date and details

6.3.1 Update Assignments Collection: Add assignment to DB

7. Start Class: Create a new classroom

7.1. Authorize User: Ensure JWT is valid

7.2. Enter Classroom Details: Enter classroom data in UI

7.2.1 Update Classrooms Collection: Add classroom to DB

8. Grade Submission: Assign marks to a submission

8.1. Authorize User: Ensure JWT is valid

8.2. Get Submission: Get submission details from DB

8.3. Enter Marks: Enter marks for submission in UI

8.3.1 Update Submissions Collection: Update marks in DB

3.3. Context Dlagram

The context diagram describes the major actors for the system and how they interact with it.

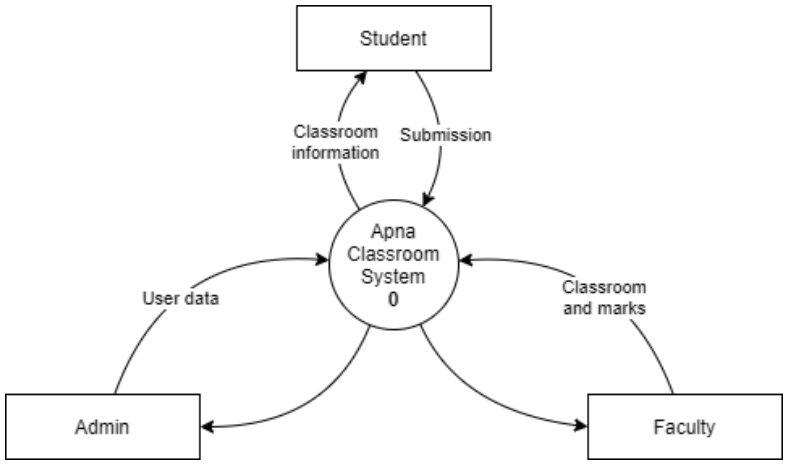

Figure 3. Context Diagram

3.4. Data Flow Diagram

The data flow diagram describes the subfunctions of the system as shown in the context diagram. The level 1 DFD is as shown below.

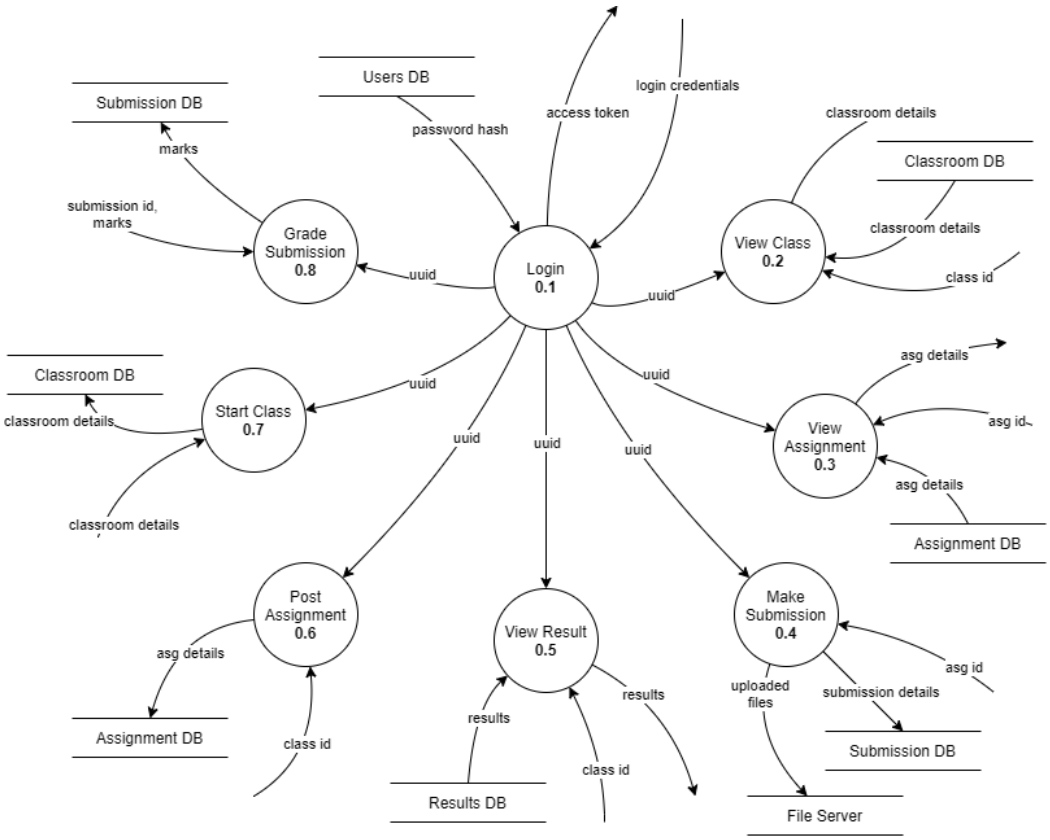

Figure 4. Level 1 Data Flow Diagram

3.5. Data Dictionary

Table 1: User

Field	Type	Null	Default
uuid	string/uuid	no	random uuid
name	string	no	no
email	string/email	no	no
password	string	no	no
classroomIDs	string[]	yes	no
role	string	no	“student”

Table 2: Classroom

Field	Type	Null	Default
facultyID	string/uuid	no	no
subjectName	string	no	no
batchCode	string	no	no
semester	string	no	no
description	string	yes	no
studentIDs	string[]	yes	no
meetingID	string/uuid	no	random uuid
theme	hex/color	no	random color

Table 3: Submission

Field	Type	Null	Default
assignmentID	string/uuid	no	random uuid
studentID	string/uuid	no	no
marks	int	yes	no
fileIDs	string/uuid[]	yes	no
submissionDate	date	yes	no

Table 4: Assignment

Field	Type	Null	Default
classroomID	string/uuid	no	no
facultyID	string/uuid	no	no
title	string	no	no
body	string	yes	no
dueDate	date	no	no
maxMarks	int	no	no
commentIDs	string/uuid[]	yes	no
fileIDs	string/uuid[]	yes	no

[P.T.O]

Table 5: Announcement

Field	Type	Null	Default
classroomID	string/uuid	no	no
userID	string/uuid	no	no
title	string	no	no
body	string	yes	no
commentIDs	string/uuid[]	yes	no
fileIDs	string/uuid[]	yes	no

Table 6: Comment

Field	Type	Null	Default
commentID	string/uuid	no	random uuid
userID	string/uuid	no	no
body	string	no	no
date	date	no	no

Table 7: Result

Field	Type	Null	Default
assignmentID	string/uuid	no	no
totalMarks	int	no	0
correctedSubmissions	int	no	0
highestMarks	int	yes	no
lowestMarks	int	yes	no

4. Component Design

4.1. Activity Diagram

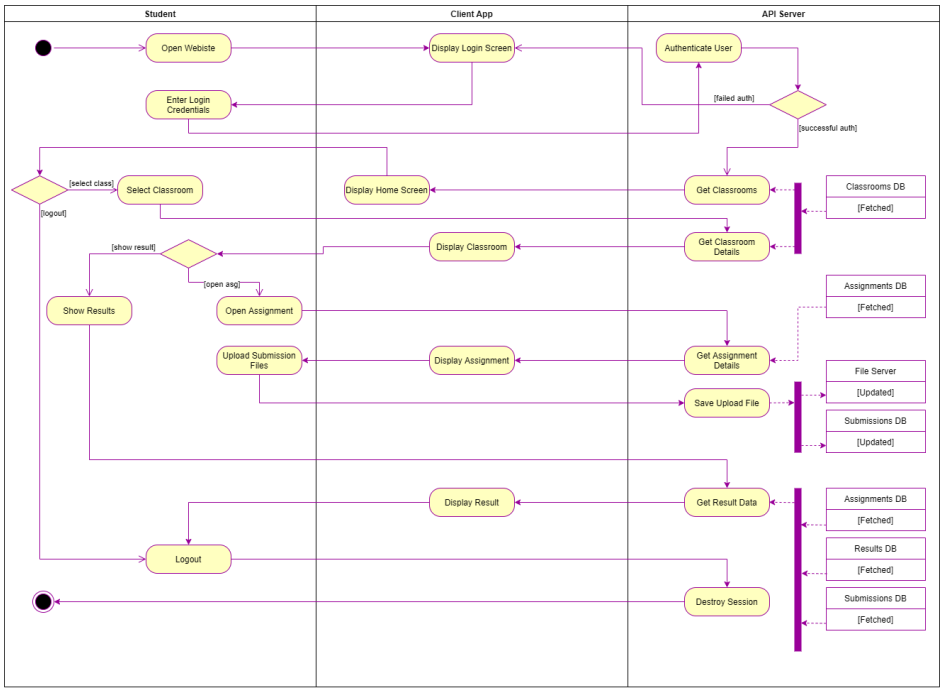

Figure 5. Activity Diagram (Student)

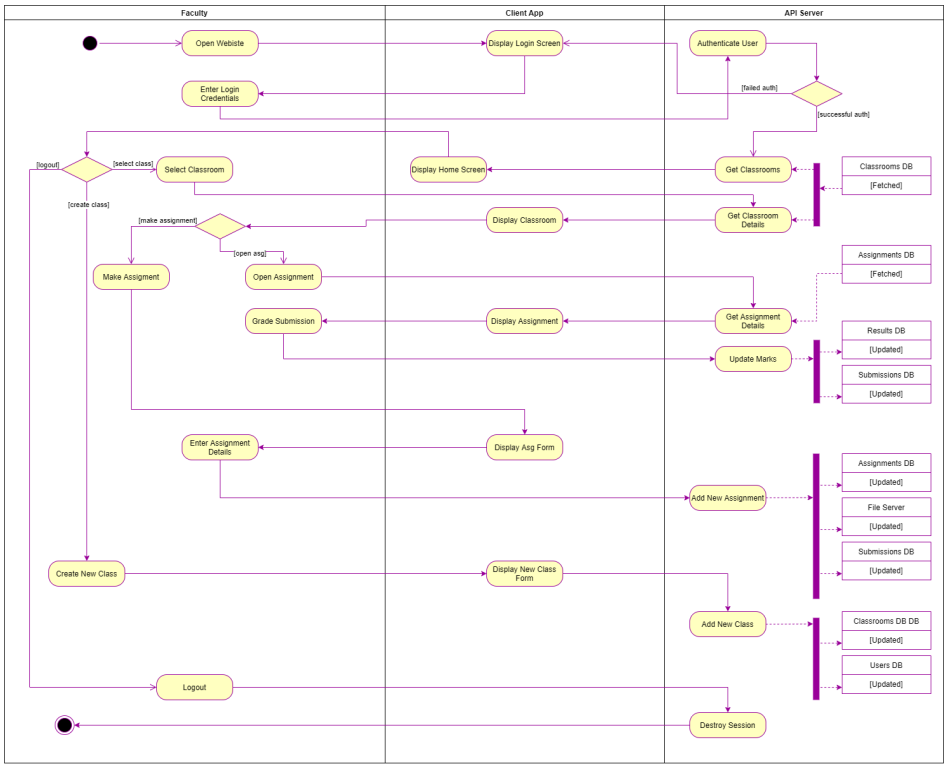

Figure 6. Activity Diagram (Faculty)

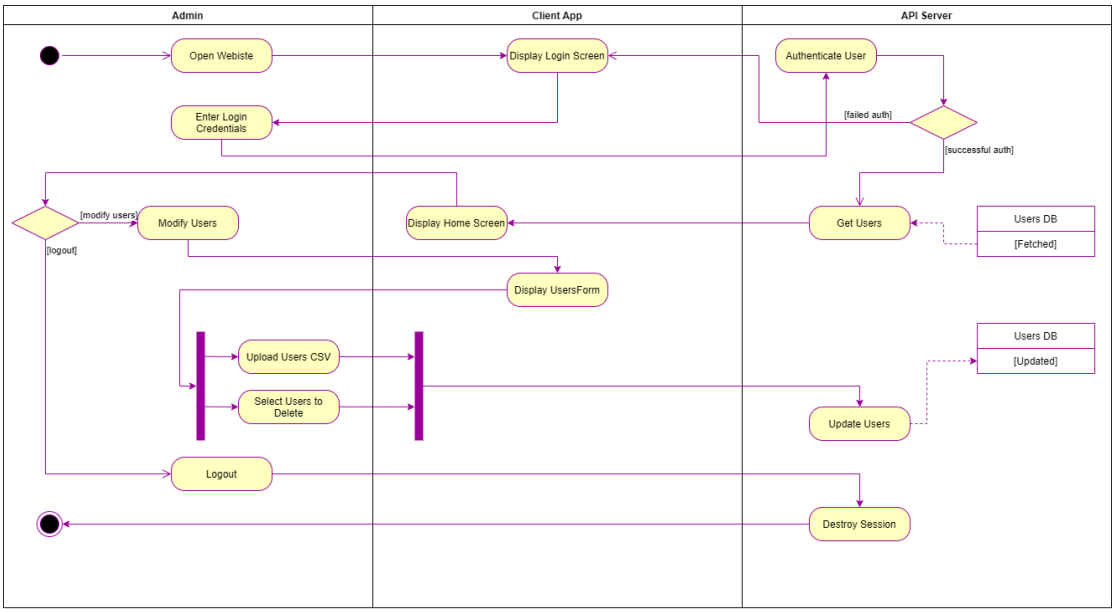

Figure 7. Activity Diagram (Admin)

5. User Interface Design

We have adhered to the following principles to design UI:

- **Structure:** All UI objects are structured in such a way that associated things are combined together and unassociated things are separated.
- **Simplicity:** Our UI is easy to navigate through, with consistent interfaces and preventing errors as much as possible.
- **Visibility:** The design is clear in appearance with elements of different sizes and comprehensible fonts.
- **Feedback:** The interactive UI provides informative feedback to the user through responses for the user's actions.
- **Reusability:** Elements with similar functions have similar designs to avoid vagueness in understanding the design.

The UI of our website will contain the main screens as shown below:

1. **Login Screen:** UI for users to enter their login credentials.
2. **Home Screen:** Frontpage of the site from where users can access all the functionality. Also displays a user's enrolled classrooms.
3. **Classroom Screen:** Displays details for selected class. Buttons to navigate to associated meet and results. Feed of posts and assignments are shown.
4. **Assignment Screen:** Shows the details for the selected assignment. Students can see a section to upload a submission. Faculty can see submissions and grade them.
5. **Meeting Screen:** Shows the video conference UI for a given classroom.

Some screens are visible only to certain types of users, as shown below

Student:

1. **Results Screen:** Displays the results of student for a particular subject
2. **ToDos Screen:** Displays upcoming/due assignments of the student

Faculty:

1. **Create Classroom Screen:** Displays form to add a new classroom.

Admin:

1. **Modify Users Screen:** Admin utility page to add or update faculty and students

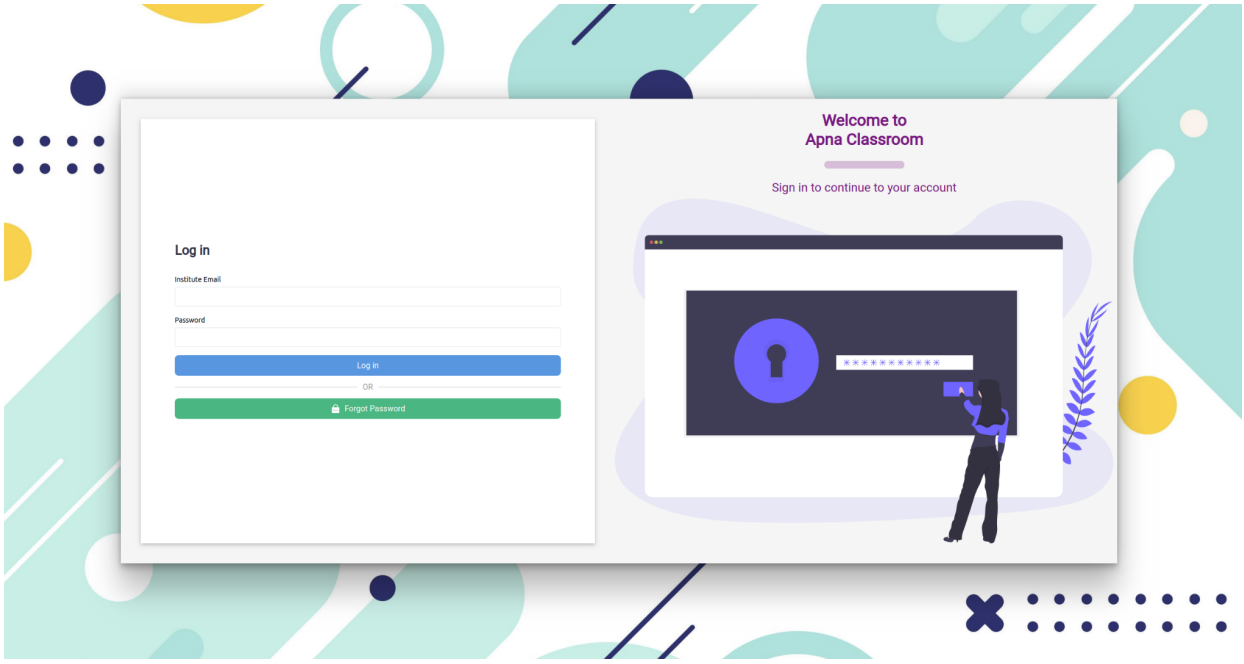

Figure 8. Login Screen

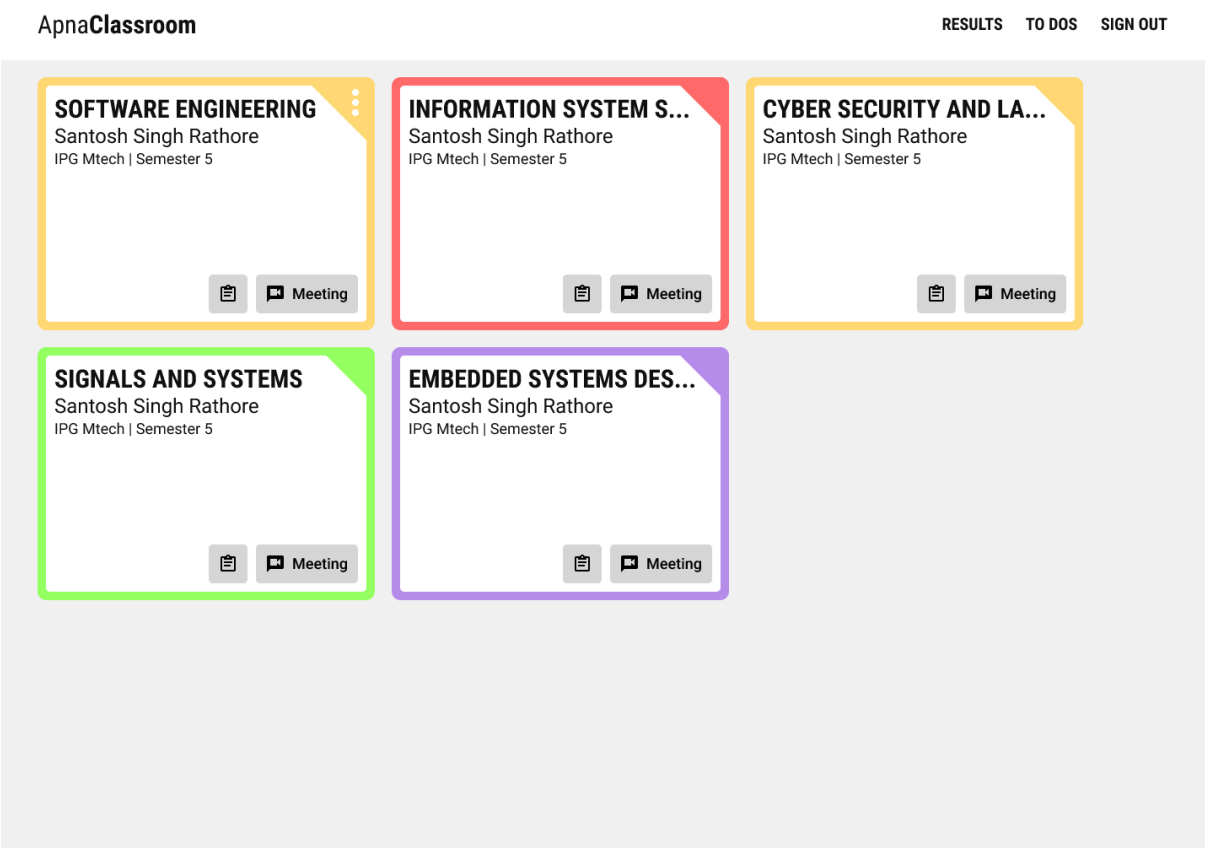

Figure 9. Home Screen

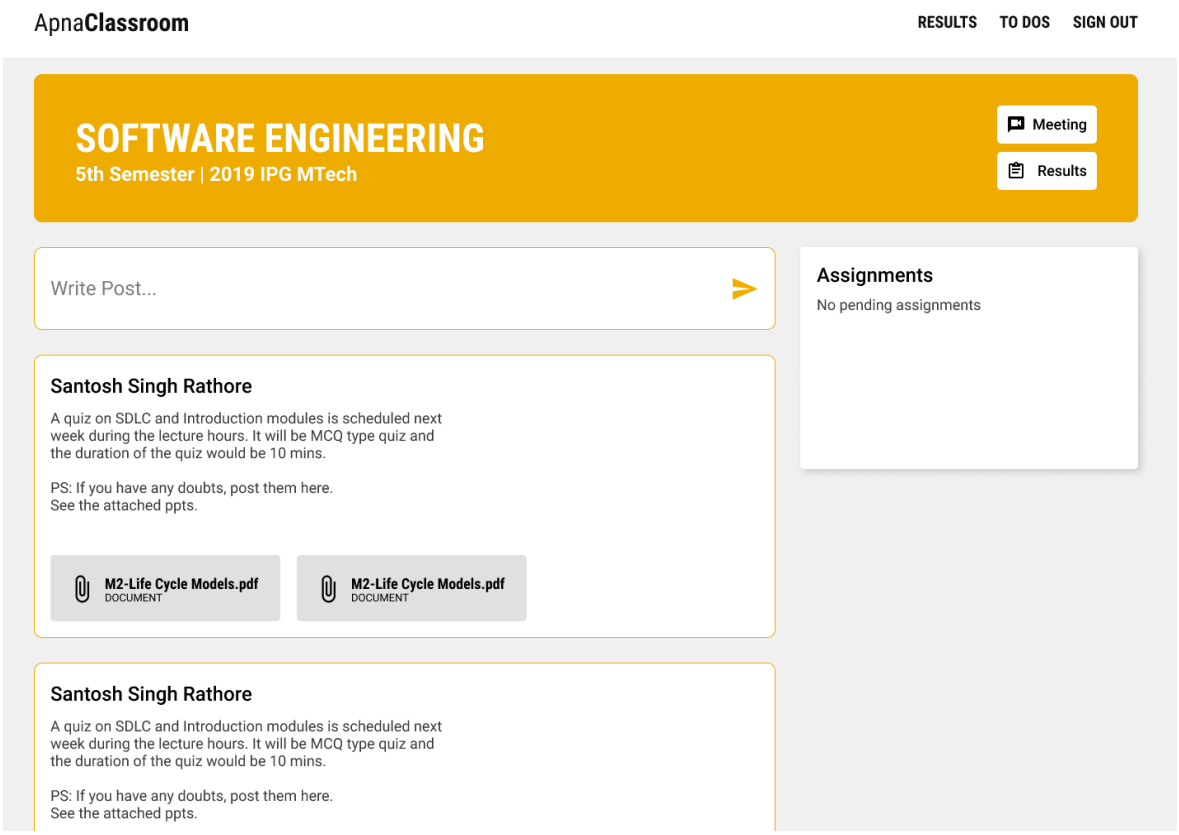

Figure 10. Classroom Screen

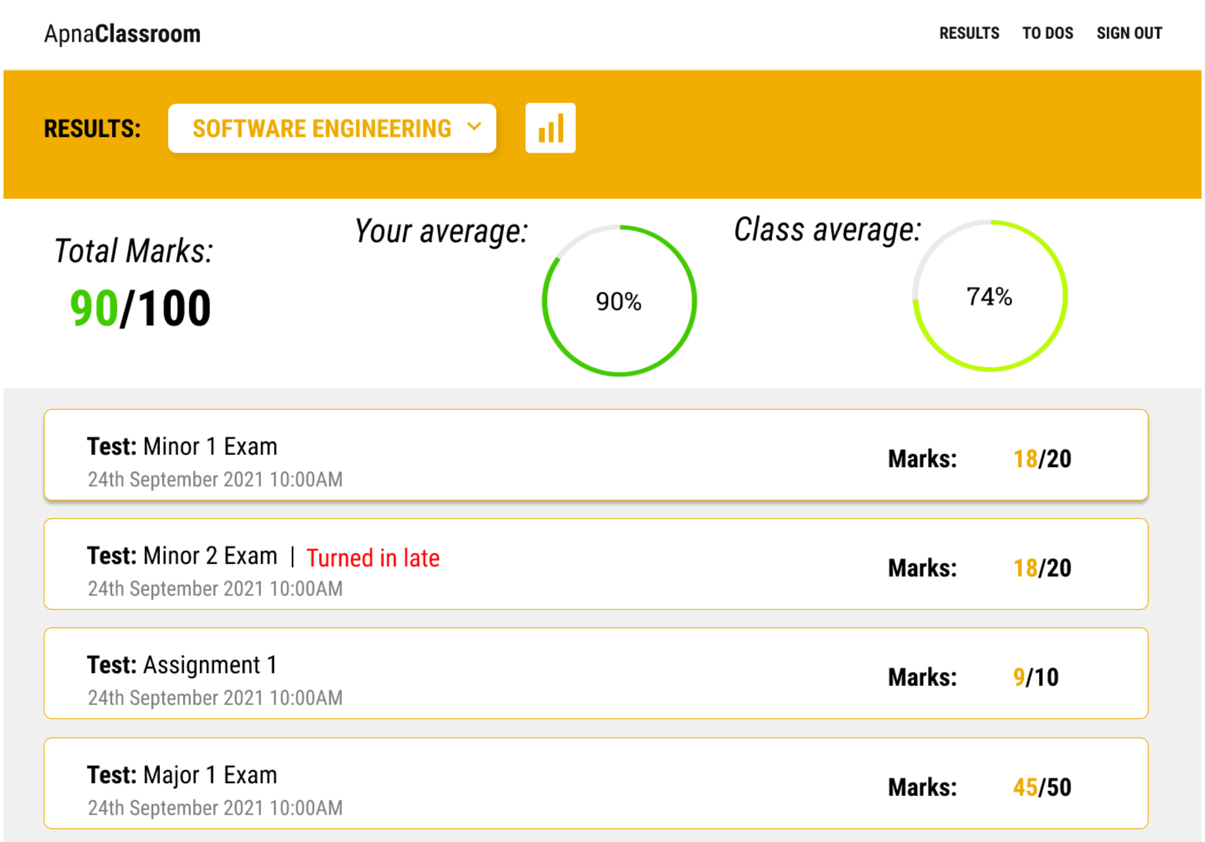

Figure 11. Results Screen

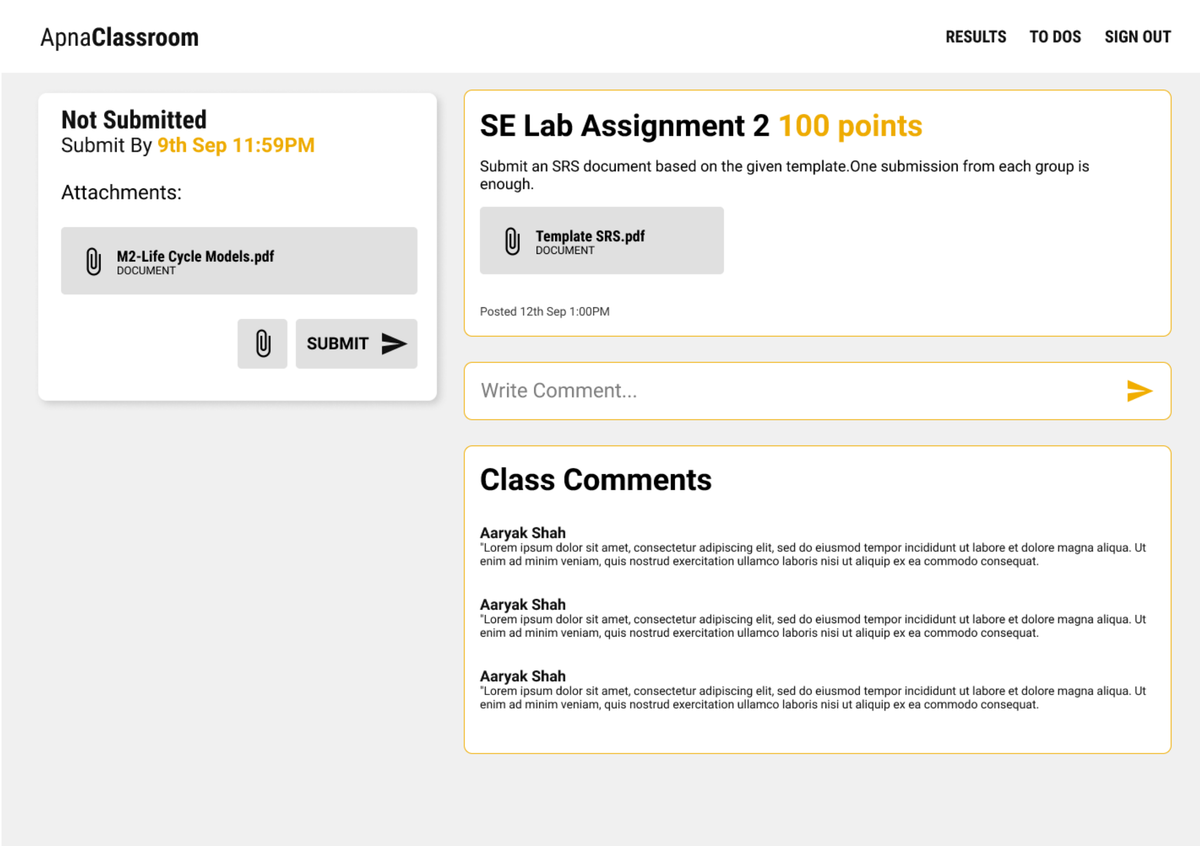

Figure 12. Assignment Screen

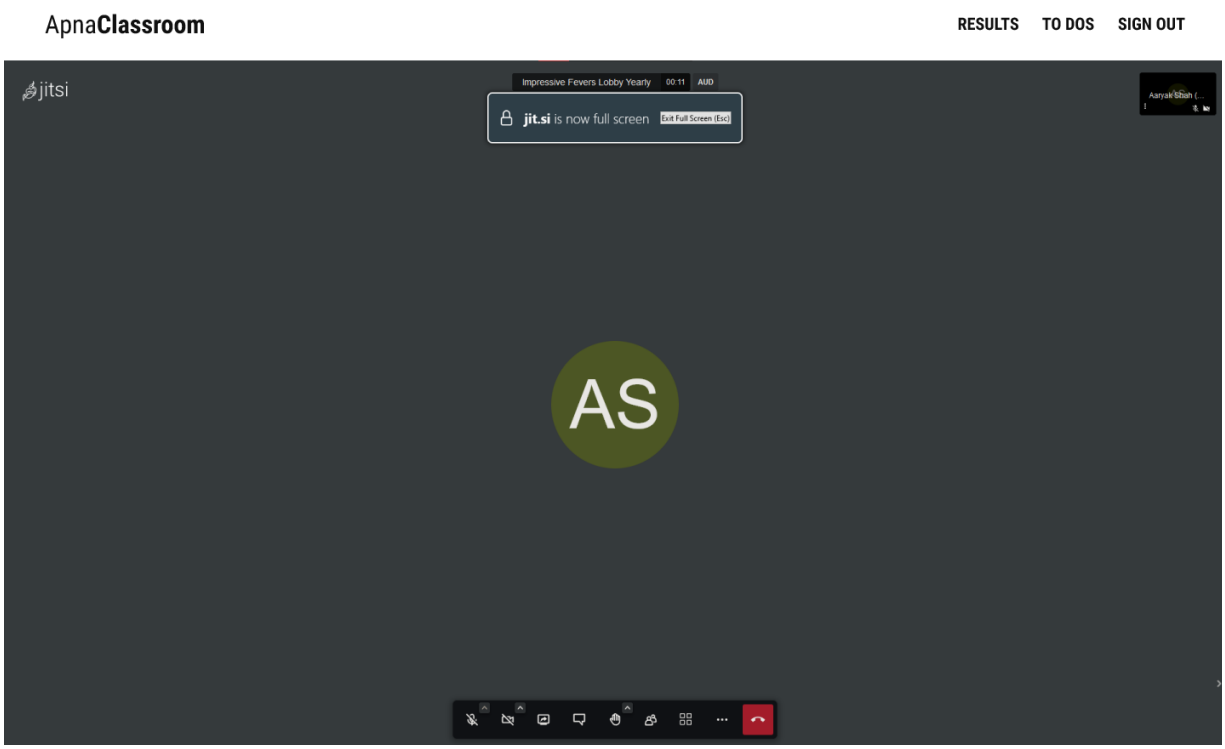

Figure 13. Meeting Screen

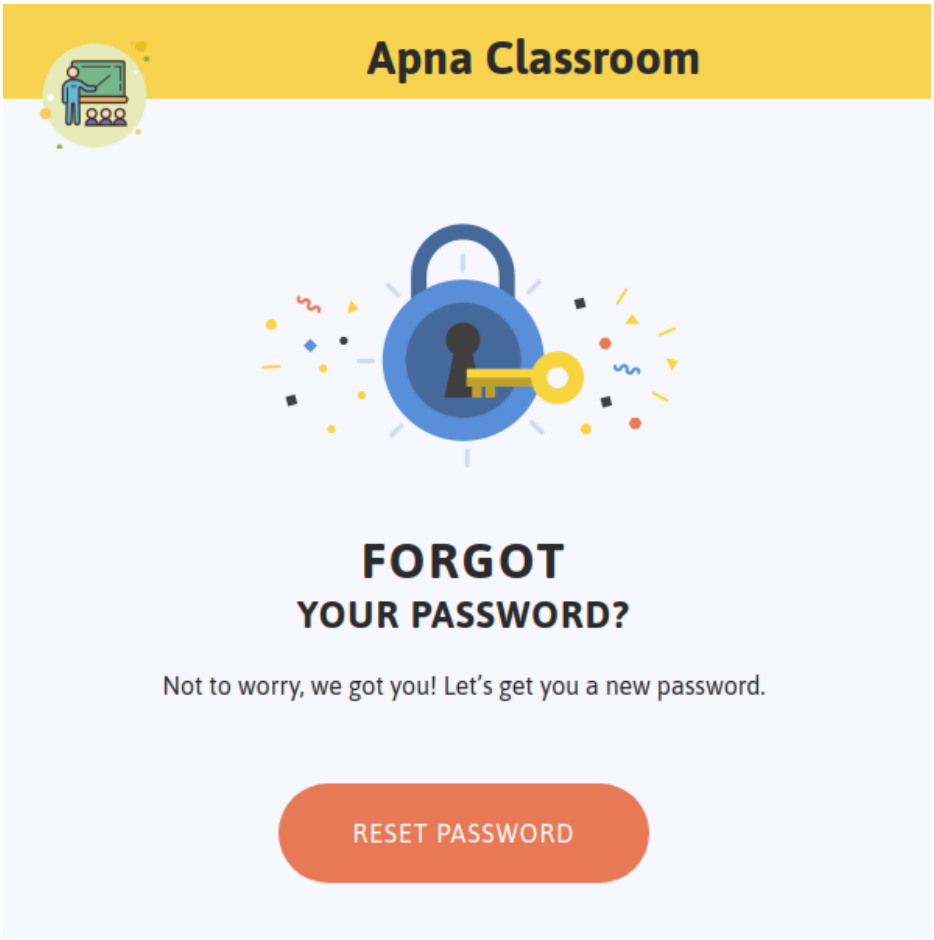

Figure 14. Forgot Password Screen