Software Requirements Specification
Version 1.0
September 30, 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. Purpose

1.1. Introduction

This SRS document aims to provide a detailed understanding of the requirements and specifications for the Online Classroom portal. The project aims to provide a complete system to handle online classrooms and maintain communications between the students and faculty.

The product is a web application, designed for use by both students and prof.s to make the classroom experience smoother and more transparent.

1.2. Scope

The Apna Classroom system is designed to work within the domain of the Indian Institute of Information Technology and Management (IIITM). It will be limited to users within this domain and will utilise the existing databases of the institute for authentication. It will allow these users to access the various online classroom features.

1.3. Glossary

Term	**Definition**
API	Application Programming Interface. An interface which lets the web application communicate with the server features.
Btn	Button used by the user in the application
CTA	Call To Action. A UI feature that provides a way to trigger some functionality.
Database	A server that is used for storing a large quantity of user data

GUI	Graphical User Interface for user interaction
Prof	A member of the faculty of the IIITM institution
Server	A system that runs backend programs and communicates with applications
SRS	Software Requirements Specification
Web Application	An application that is accessed through a browser
Web Site	A place on the world wide web

1.4. References

SRS Sample Document Template
Use Case and Activity Diagrams ppt

1.5. Document overview

The following two chapters of the document describe the details and specifications of the software project. The second section gives a thorough description of the functionality of the project and it’s many use cases. The third section presents a specification of the software requirements in a form that is developer-friendly. These chapters are scribed after discussion between all involved parties to provide a full picture of the system.

2. Overall description

Apna Classroom is a software application on the web platform. It involves an assortment of data from the existing IIITM database and provides it's feature set alongside. The software is divided into two subsystems, a client-side rendered web application and a backend API which can be hosted on the IIITM web server. Users can access the portal through any browser.

2.1. System environment

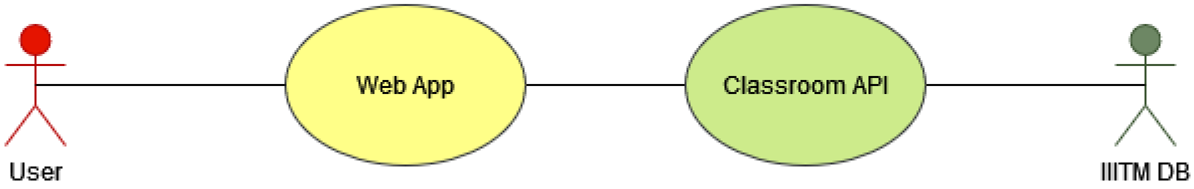

Figure 1. System Design

The Apna Classroom online classroom portal, as mentioned, will be divided into two parts. Both the client application and the API service will be hosted on the institute provided web server. The client application will communicate with the API backend, and the API backend will further interact with the IIITM database.

2.2. Functional requirements definitions

The functional requirements are the software requirements which pertain to the functionalities that the project must incorporate as requested by the customer. This is as opposed to the nonfunctional requirements which are more peripheral requirements needed for the functioning of the system.

2.3. Use cases

The system is divided into two parts. The frontend application is a GUI web application that services both students and faculty. The users can access the site through a sign in portal. The GUI is composed initially of a home screen that displays the user's associated classrooms. The user can access a classroom's page through here. The classroom page will show the announcements and assignments in a feed, along with CTAs to the features such as video meeting and post form.

The assignment page will allow students to make submissions, and provide a way for a prof to mark these submissions.

The home screen also allows students to access the results page. This screen shows the user their subject-wise results and the marks they received for their submissions.

The backend is an API that communicates with the web application. It is responsible for accepting requests to create new classrooms, announcements, assignments and submissions. The API stores this data in a structured way in the database. It also responds to the web application with relevant data whenever required.

2.3.1. Use Case: User Authentication with IIITM Database

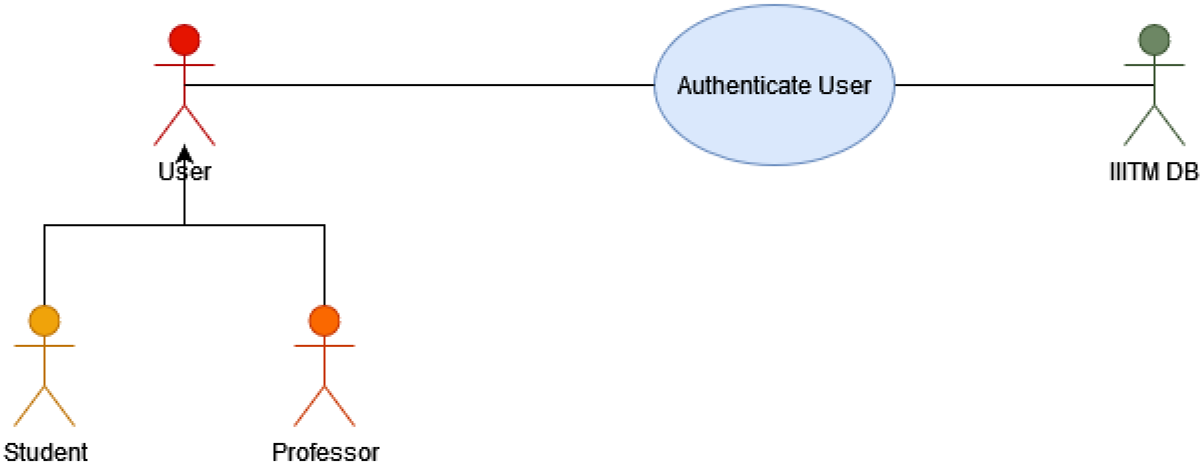

Figure 2. Authenticate user using IIITM database

Brief Description

The user (student or faculty) attempts to sign in to the application and is subsequently authenticated.

Initial step-by-step description

This use case requires the user to have a iiitm domain email id.

1. The user connects to the API service
2. The user enters their login credentials
3. The API authenticates the user
4. The user proceeds to the main application

Refer to: SRS 3.2.1

2.3.2. Use Case: Making a Virtual Classroom

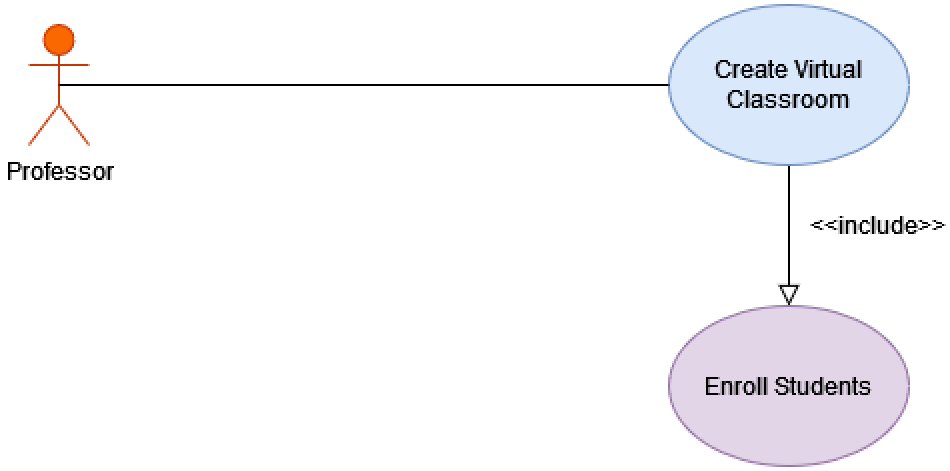

Figure 3. Add new virtual classroom

Brief Description:
The faculty chooses to create a virtual classroom.

Initial step-by-step description:
This use case requires the user to be authenticated as a faculty member of IIITM.

1. The prof. clicks on the new classroom button
2. The prof. must fill a form describing the details of the classroom
3. The prof. details the batch of students who will attend the class
4. The API generates a meeting link
5. The API creates a classroom and automatically enrolls the relevant students
6. Students who were enrolled receive an email

Refer to: SRS 3.2.2

2.3.3. Use Case: Make an Announcement in a Classroom

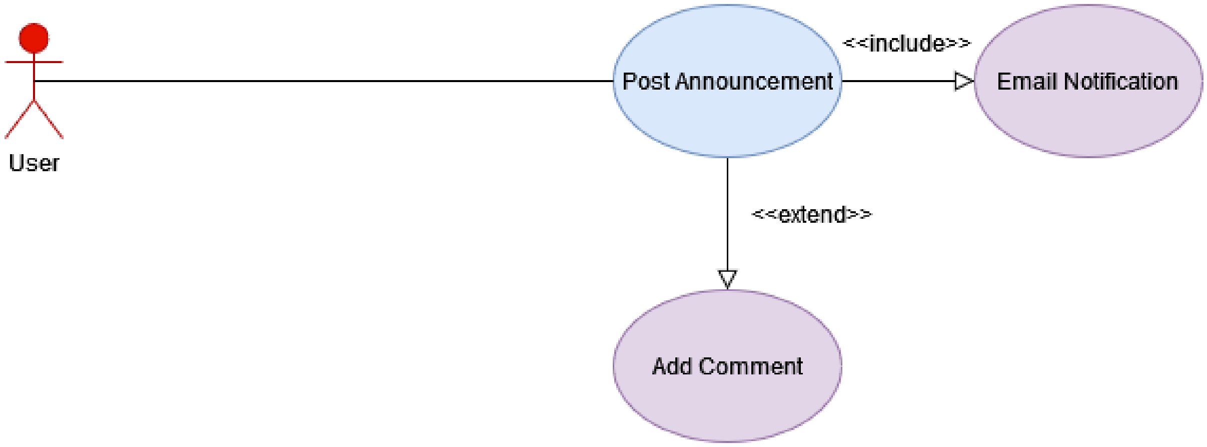

Figure 4. Post new announcement

Brief Description:
The user announces something to a classroom

Initial step-by-step description.
This use case requires the user to be authenticated as either a faculty member or

a student, and must be a member of the specified classroom

1. The user clicks on New Post
2. The user adds a title and announcement body
3. The post is processed by the API and added to the content feed for the classroom
4. The users in the classroom receive an email notification

Refer to: SRS 3.2.3

2.3.4. Use Case: Making an Assignment for a Classroom

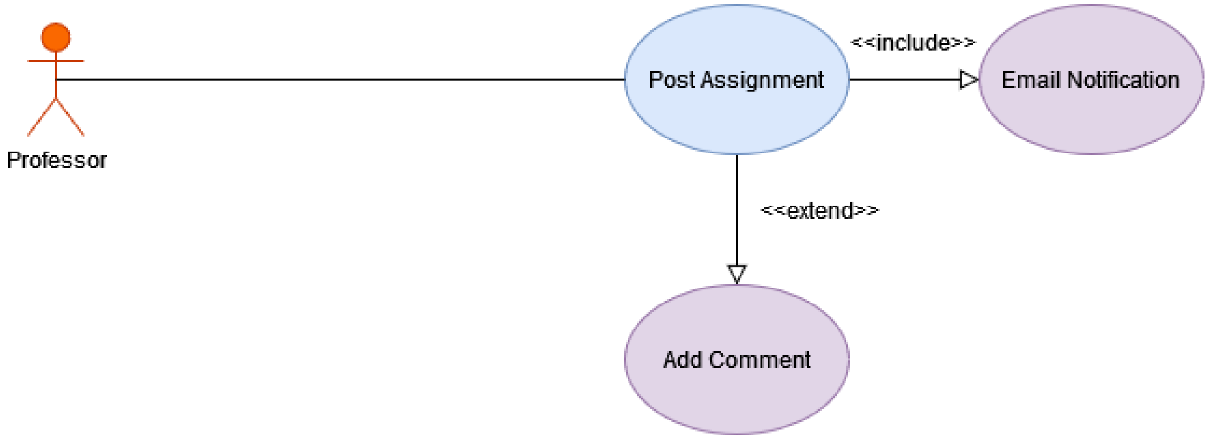

Figure 5. Post new assignment

Brief Description:

The prof. chooses to make an assignment to evaluate the classroom members

Initial step-by-step description:

This use case requires the user to be authenticated as a faculty member, and

must be the teacher for the specified classroom

1. The prof. chooses to make a new assignment
2. The prof. adds a title and description of the assignment
3. The prof. allots the maximum marks and due date for the assignment
4. The API registers the assignment and adds it to the classroom feed
5. Members of the classroom get notified by email

Refer to: SRS 3.2.4

2.3.5. Use Case: Grading a Submission

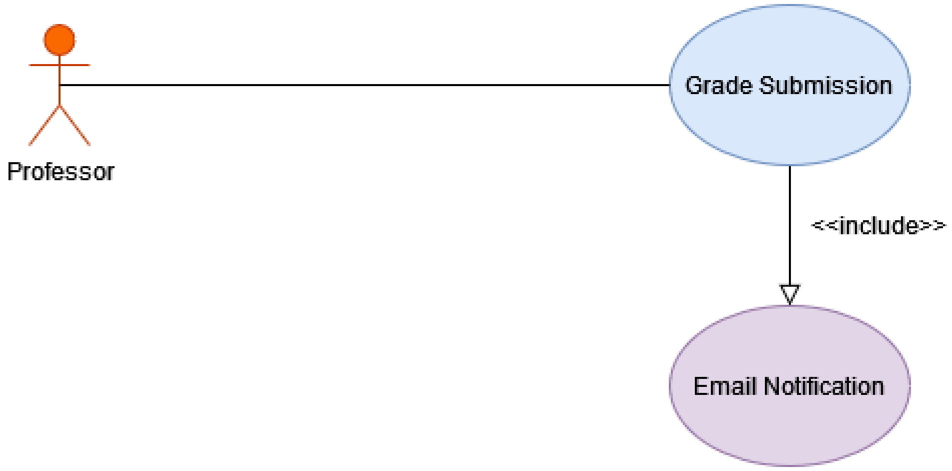

Figure 6 Grade a submission

Brief description:
The prof. grades a student’s assignment submission

Initial step-by-step description:
This use case requires the user to be authenticated as a faculty member and

must be the teacher of the specified classroom

1. The prof. views the submissions on an assignment
2. The prof. assigns some marks out of the total on the submission
3. The API registers the grading and updates the results database

Refer to: SRS 3.2.5

2.3.6. Use Case: Making a Submission

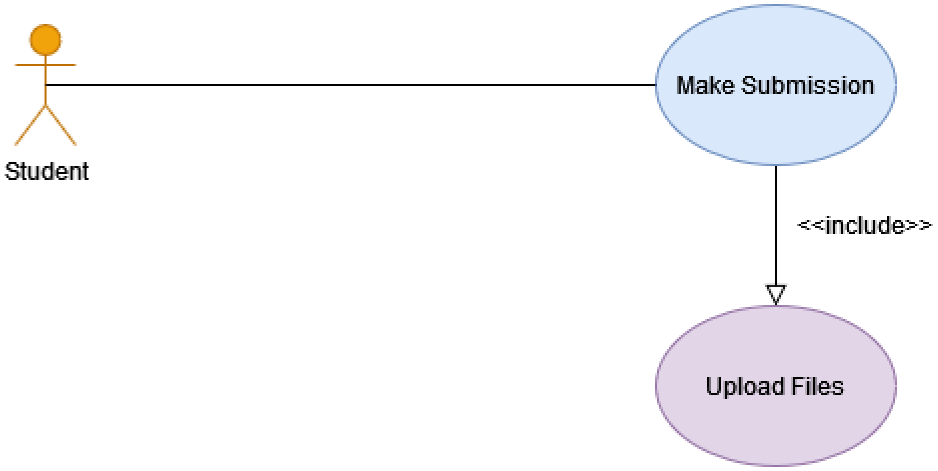

Figure 7. Make a new submission

Brief description:

The student decides to make a submission for an assignment

Initial step-by-step description:

This use case requires the user to be authenticated as a student and must be a

member of the specified classroom

1. The student views and attempts an assignment
2. The student uploads his submission file
3. The student clicks on submit
4. The API registers the submission

Refer to: SRS 3.2.6

2.3.7. Use Case: Viewing results

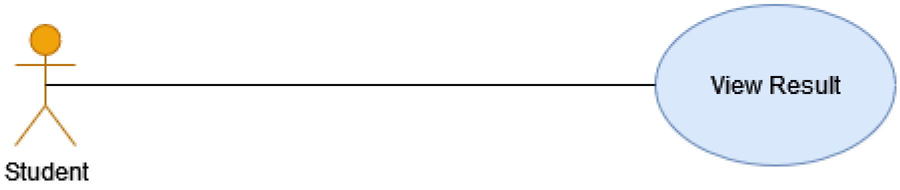

Figure 8. View results

Brief description:

The student decides to view their results

Initial step-by-step description:

This use case requires the user to be authenticated as a student

1. The student accesses the results page
2. The API compiles a result report from the results database
3. The client displays the result as a cumulative performance as well as submission wise.

Refer to: SRS 3.2.7

2.4. Non-functional requirements

These are the software requirements which do not directly relate to the project's functionality. These supplementary requirements relate to the other information required for the correct functioning of the system.

This web application must be domain-limited and work on all modern web browsers.

3. Requirement specifications

3.1. External interface specifications

This specification does not involve any external interfaces.

3.2. Functional Requirements

3.2.1. Authentication System

Use Case Name:	User Authentication with IIITM Database
Priority	Essential
Trigger	Access application
Precondition	User has a IIITM email
Basic Path	1. Web app sends auth request to API 2. API authenticates using IIITM Database
Alternate Path	N/A
Postcondition	User is logged in and can use features
Exception Path	If the user does not have a iiitm domain account the user is not logged in
Reference	SRS 2.3.1

3.2.2. Virtual Classroom

Use Case Name:	Making a Virtual Classroom
Priority	Essential
Trigger	New classroom button
Precondition	The user is authenticated as a faculty member
Basic Path	1. The user clicks on the button 2. The user fills a form detailing the classroom's subject, description and the expected batch 3. The API generates a meeting link and creates virtual classroom
Alternate Path	N/A
Postcondition	The classroom is registered and students are automatically enrolled
Exception Path	If the user's session has timed out, the user is redirected to log in
Reference:	SRS 2.3.2

3.2.3. Classroom Announcement Posts

Use Case Name:	Make an Announcement in a Classroom
Priority	Essential
Trigger	Make an announcement CTA in a classroom
Precondition	The user must be a member of the specified classroom
Basic Path	1. The user clicks on CTA 2. The user details the title and announcement body in a form 3. The user submits the announcement 4. The API adds announcement to the classroom's feed
Alternate Path	N/A
Postcondition	The posts data collection is updated and the classroom members are notified
Exception Path	If the session is timed out the user is redirected to log in
Reference:	SRS 2.3.3

3.2.4. Classroom Assignment

Use Case Name:	Making an Assignment for a Classroom
Priority	Essential
Trigger	Make a new assignment CTA in a classroom
Precondition	The user must be the prof. of the specified classroom
Basic Path	1. The prof. clicks on CTA 2. The prof. adds details of the assignment, including title, description, the due date and the maximum marks for the assignment 3. The API registers the assignment
Alternate Path	N/A

Postcondition	The assignment is added to the database and students in the classroom are specified
Exception Path	If the user is a student the CTA is not available. If the user session is timed out the user is redirected to login
Reference:	SRS 2.3.4

3.2.5. Submission Grading

Use Case Name:	Grading a Submission
Priority	Essential
Trigger	Add marks CTA
Precondition	The user must be authenticated as a prof.
Basic Path	1. The prof. clicks on CTA on an assignment post 2. The prof. views the submissions 3. The prof. assigns marks to the submissions 4. The API registers the grading
Alternate Path	N/A
Postcondition	The marks are added to the database and the results collection is updated.
Exception Path	If the user is a student the CTA is not available, instead the user’s marks are displayed if available. If the user session is timed out the user is redirected to login
Reference:	SRS 2.3.5

3.2.6. Assignment Submission

Use Case Name:	Making a Submission
Priority	Essential
Trigger	Assignment submit Btn
Precondition	The user must be a student of the specified classroom
Basic Path	1. The user views the assignment 2. The user uploads submission files, if any 3. The user clicks the submit btn

	4. The API registers the submission and stores the files
Alternate Path	N/A
Postcondition	The submission is added to the database
Exception Path	If the user session is timed out the user is redirected to login
Reference:	SRS 2.3.6

3.2.7. Results Screen

Use Case Name:	Viewing Results
Priority	Essential
Trigger	View results Btn
Precondition	The user must be authenticated
Basic Path	1. The user clicks on the view results btn 2. The web app communicates with the API to obtain results 3. The API loads the results from the database and relays them to the web app
Alternate Path	N/A
Postcondition	The results are displayed
Exception Path	If the user session is timed out the user is redirected to login
Reference:	SRS 2.3.7

3.3. Detailed non-functional requirements

User Data Collection

Data Item	**Type**	**Description**
Name	Text	Name of user
Email Address	Email	Email address
User Type	Text	Student/Faculty
Associated Classrooms	List<ID>	The classrooms that the user is part of

Classroom Data Collection

Data Item	**Type**	**Description**
Subject	Text	Subject name of the Class
Description	Text	Description of the subject
Meeting ID	Text	Meeting ID for the classroom
Batch Code	Text	Student batch descriptor
Teacher	ID	ID of the teacher
Associated Students	List<ID>	List of student IDs who are enrolled to the classroom

Assignment Data Entity

Data Item	**Type**	**Description**
Title	Text	Title of the Assignment
Description #	Text	Description of the assignment
Attachments #	File	Files associated with the assignment
Due Date	Date	Deadline for the assignment
Max Marks	Number	Maximum marks for the assignment
Classroom	ID	Classroom where assignment is posted

Announcement Data Entity

Data Item	**Type**	**Description**
Title	Text	Title of assignment
Body #	Text	Announcement body
Attachments #	File	Files associated with the announcement
User	ID	User who posts the announcement
Classroom	ID	Classroom where announcement is posted

Submission Data Entity

Data Item	**Type**	**Description**
Attachments #	File	Submission files

Submission Time	Date	Time at which submission is made
Marks Assigned #	Number	Marks allotted to submission by a professor
User	ID	User who makes the submission
Assignment	ID	Assignment on which submission is made

Fields marked with a “#” symbols are optional or not always available.

These data entities are stored as collections on the database. The classroom, it’s students, and it’s various posts and assignments are all related through their IDs as references.

Hardware	Institute Server
Operation System	Windows, Linux, MacOS, any other operating system with browser
Internet Connection	Required
Code Standard	The frontend will use React.js The backend will utilise Node.js and MongoDB The code will be fully documented
Performance	The software product should reliably register and store classrooms, posts, assignments and submissions

3.4. System Evolution

The system should be updatable and leave room for adding features such as attendance, result statistics, re-evaluation of grading and an integrated quiz builder.

4. Index