



**IT314**

**SOFTWARE ENGINEERING PROJECT**

**MENTOR : PROF SAURABH TIWARI**

**Software Requirements Specification**

**Group ID 3 Topic: Virtual Classroom**

SR	Student ID	Name
1.	202201005	Akhil Rachhadiya
2.	202201006	Kavan Patel
3.	202201027	Harsh Rajwani
4.	202201033	Sagar Surti
5.	202201049	Aditya Sable
6.	202201061	Patel Nipurnakumari
7.	202201065	Joshi Meet
8.	202201076	Dhanani Vidhi
9.	202201084	Manavadariya Sujalkumar
10.	202201091	Kathan Khuman

# **INDEX:**

## **1. Introduction**

1.1 Introduction

1.2 Technologies used

## **2. Overall Description**

2.1 Users and Stakeholders

2.2 Constraints

2.3 Assumptions and Dependencies

2.4 Software Model Used

## **3. Requirement Analysis**

3.1 Functional Requirements

3.2 Non-functional Requirements

## **4. Use Cases**

4.1 General use cases

4.2 Logged in user

4.3 Premium User

4.4 Admin

## **5. Use Case Description**

## **6. User Stories**

## **7. Sprints**

## **8. Future Work**

# **1. INTRODUCTION:**

The Software Requirements Specification (SRS) covers its purpose, scope, definitions, abbreviations, references, and a general summary. This document analyzes and offer a comprehensive understanding of the Virtual Classroom by clearly defining the problem statement. It highlights the functionalities required by stakeholders and their needs while outlining the high-level features of the platform.

This project is a combination of Google Classroom and YouTube, offering a platform for managing and delivering educational content. It provides functionality similar to Google Classroom, including student tracking, individual classes, and the ability to monitor progress. Additionally, it allows educators to upload lectures along with accompanying materials, much like YouTube.

The goal is to offer a seamless experience for educators to manage classes and for students to access lectures, assignments, and resources in one integrated platform.

## **Technologies Used :**

- **Frontend:** React+Vite, CSS/TailwindCSS
- **Backend:** Node.js, Express.js
- **Storage:** Firebase(Video lectures & materials)
- **Database:** MongoDB (via Mongoose)
- **Authentication:** JWT

## **2. OVERALL DESCRIPTION:**

This section gives you the gist of our whole project, including details of the users and stakeholders involved in this and also the constraints under which we have made this project. We have also made some assumptions, which are defined below.

### **2.1 Users and Stakeholders**

#### **Users:**

- Students : who use this platform for learning new topics.
- Teachers : uses this for delivering courses.

#### **Stakeholders:**

- Developers: responsible for building and maintaining the platform.

### **2.2 Constraints**

- **Performance Constraints:** The platform should be able to handle high traffic and large numbers of concurrent users efficiently.
- **Security:** Protect user data and prevent unauthorized access.
- **Compliance:** Adhere to education-related regulations and data protection laws.
- **Budget:** Limited resources for development and maintenance
- **Storage :** Limited storage space for user data (such as videos, documents, and progress tracking).

## 2.3 Assumptions and Dependencies

- **User Engagement:** Users will regularly access the platform for learning and content delivery.
- **Internet Access:** Learners and instructors have access to the internet for cloud-based functionalities.
- **Technology Familiarity:** Users have basic tech proficiency to navigate the platform.
- **Third-party Services:** Dependence on external APIs for features like payment processing, authentication, or video streaming.
- **Cloud Infrastructure:** Reliance on cloud providers for storage and hosting.
- **Regulations:** Compliance with data privacy laws and educational standards.

## 2.4 Proposed Model:

1. **User-Centric Design:** A platform tailored to students and educators with personalized dashboards.
2. **Modular Content Delivery:** A flexible system where instructors can easily upload and manage content (e.g., videos, quizzes).
3. **Cloud Infrastructure:** Using scalable cloud storage and computing for data and traffic management.
4. **Learning Analytics:** Integration of performance tracking to provide feedback to both learners and teachers.
5. **Integration with External Tools:** API connections for third-party content, and tools.

- 6. Repeat Sprint Cycles Until Completion:** Gather feedbacks and Improve after repetitive cycles.
- 7. Final Review and Launch Preparation :** After multiple rounds of testing, security checks and reviews.

## **Benefits of Agile for Learnify Project**

**Adaptability:** Agile enables continuous feedback from students and educators, ensuring the platform is regularly updated to meet evolving user needs.

**Incremental Releases:** New features and improvements are released in phases, allowing early user testing, minimizing risks, and facilitating quick adjustments.

**Collaboration:** Regular collaboration with stakeholders, including instructors and administrators, helps create a platform that is aligned with user expectations, fostering greater engagement and satisfaction.

### **3. REQUIREMENT ANALYSIS:**

#### **3.1 Functional Requirements:**

- **User Authentication:** Secure login for students, instructors with roles and permissions.
- **Course Creation and Management:** Instructors can create, update, and organize course content (e.g., videos, quizzes).
- **Progress Tracking:** Students can track their learning progress and performance.
- **Search and Filtering:** Users can search for courses based on topics, difficulty, and ratings.
- **Collaboration Tools:** Forums, discussions, and peer reviews to enhance learning.
- **Notifications:** Alerts for new content, deadlines, and updates.

#### **3.2 Non-functional requirements:**

- **Performance:** The platform must handle multiple of concurrent users with minimal latency and fast response times.
- **Scalability:** Should scale efficiently to accommodate an increasing number of courses, users, and data as the platform grows.
- **Security:** Ensure data protection with secure user authentication, encryption, and compliance with data privacy regulations (e.g., GDPR).
- **Availability:** The platform must have high uptime (99.9%) to ensure consistent access for users.
- **Usability:** Intuitive and user-friendly interface for students, and instructors.
- **Compatibility:** Accessible on multiple devices and browsers (desktop, mobile, tablet).
- **Maintainability:** Easy to maintain and update, with clear documentation for future developers.

## 4. USE CASES:

### 4.1 General Use Cases for Learnify :

**Sign up and login securely:** Students, instructors can register and log in with secure authentication methods.

**Browse and search courses:** Users can view courses, search by topics, difficulty, and ratings.

**View learning resources:** Students can access course materials such as videos, documents, and assignments.

### 4.2 Logged-In User Use Cases:

**Track progress:** Students can monitor their learning progress and grades.

**Enroll in courses:** Students can sign up for courses and start learning.

**Interact with content:** Students can participate in discussion forums and provide feedback.

### 4.3 Instructor Use Cases:

**Create and manage courses:** Instructors can create new courses, upload materials, and update course content.

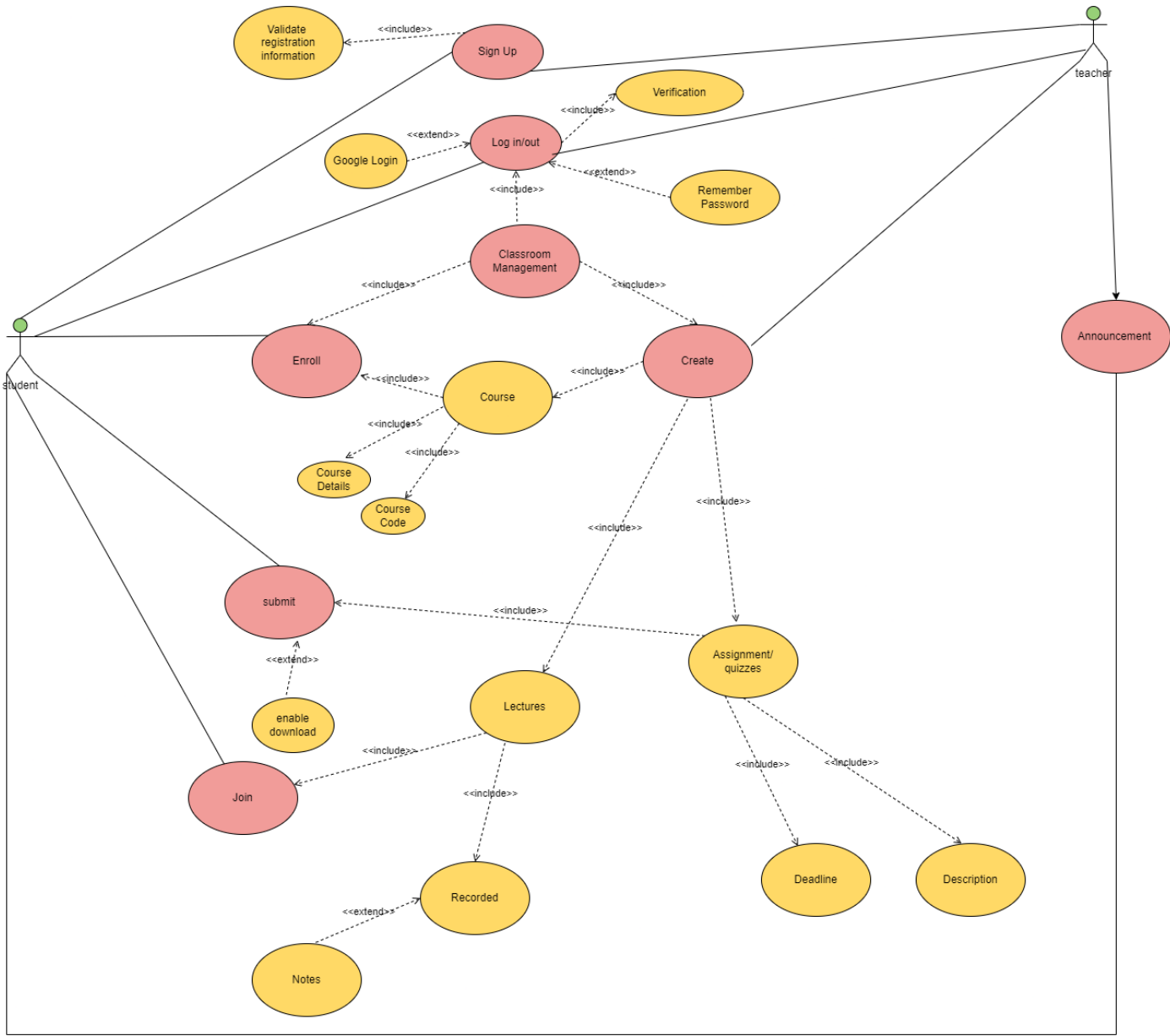
**Monitor student progress:** Instructors can track the progress and performance of students.

**Provide feedback:** Instructors can comment on students' work and assign grades.



5. USE CASE DESCRIPTION:

VIRTUAL CLASSROOM



## 5.1 Use Case: Sign Up Securely

- **Description:**

Allows users to create an account on the platform securely using email/password or social login.

**Actors:** Student, Teacher

### **General User**

#### **Preconditions:**

- The user has access to the platform.
- The user does not already have an account.

#### **Main Flow:**

- The user clicks the "Sign Up" button.
- The system displays the sign-up form.
- The user enters required details (email, password, etc.).
- Optionally, the user clicks "Sign Up with Google."
- The system validates the input.
- The system checks for existing emails.
- Account is created and verified via email.
- User is redirected to the dashboard.

**Invalid fields:** System prompts corrections.

Email exists: Prompt user to login.

#### **Postconditions:**

User account is created and email verification sent.

## 5.2 Use Case: Login

**Description:**

Allows users to log into the platform using email/password or social login.

**Actors:** Student, Teacher

**General User****Preconditions:**

- User has an existing account.

**Main Flow:**

- User clicks the "Login" button.
- The system displays the login form.
- The user enters login credentials.
- The system authenticates and grants access.
- User is redirected to the dashboard.

**Alternative Flow:**

**Invalid credentials:** System displays an error and prompts retry.

**Postconditions:**

User is logged in and granted access to the platform.

## 5.3 Use Case: Announcement

**Description:**

Enables administrators to post announcements visible to all users.

**Actors:** Developer

**Main Flow:**

- Admin accesses the "Announcement" section.
- Admin enters announcement content.
- Admin clicks "Post."
- System displays the announcement to users.

Alternative Flow:

- Invalid content: System displays an error.

**Postconditions:**

- Announcement is posted and visible to all users.

## 5.4 Use Case: Add Class

**Description:**

Instructors can create and manage new classes or courses.

**Actors:** Instructor

**Preconditions:**

Instructor is logged in.

**Main Flow:**

- Instructor clicks on "Add Class" button.
- Instructor enters class details (name, description, etc.).
- Instructor uploads course material.
- Instructor submits to create the class.

- System saves and lists the new class.  
Alternative Flow:
- Missing details: System prompts for corrections.

**Postconditions:**

New class is created and visible to users.

## **5.5 Use Case: Join Class**

**Description:**

Allows students to enroll in courses they are interested in.

**Actors:** Student

**Preconditions:**

Student is logged in.

Class is available for enrollment.

**Main Flow:**

- Student browses available classes.
- Student clicks "Join or plus sign" button and enter course code.
- System enrolls the student and updates class list.

**Alternative Flow:**

- Class full: System notifies user and prompts to choose another.

**Postconditions:**

- Student is enrolled in the class and can access course materials.

## **5.6 Use Case: Upload Course Content**

**Description:**

Instructors upload educational content for their courses.

**Actors:** Instructor

**Preconditions:**

Instructor is logged in.

Course has been created.

**Main Flow:**

- Instructor clicks on "Upload" button.
- Instructor selects the content (videos, PDFs, etc.).
- Instructor submits the content.
- System stores and displays the content in the course.

Alternative Flow:

- Invalid format: System shows an error message.

**Postconditions:**

Content is uploaded and available for students.

**5.7 Use Case: Send Email****Description:**

System sends email to selected users.

**Actors:** Teacher

**Preconditions:**

User has an registered email address.

## **Main Flow:**

- The system generates an email (e.g., confirmation, notification).
- The system sends the email to the selected user's registered address.
- User receives the email in their inbox.

## **Alternative Flow:**

Email fails: System retries or alerts the admin.

## **Postconditions:**

Email is sent successfully, or failure is logged for further action.

## 6. USER STORIES :

Front of the Card	Back of the Card (Acceptance Criteria)
<p>1) As a student, I want to register myself on the application, so that I can create an account and join classes.</p> <p>Priority: High</p>	<ol style="list-style-type: none"><li>1. The student should be able to create an account after providing valid credentials.</li><li>2. The system should verify the credentials before account creation.</li></ol>
<p>2) As a user, I want to log in/out from my account, so that I can access my classroom features.</p> <p>Priority: High</p>	<ol style="list-style-type: none"><li>1. The user can log in and out using valid credentials.</li><li>2. The system should remember the user's login information for quicker access</li><li>3. The user should have the option to reset their password if forgotten.</li></ol>



<p>3) As a teacher/admin, I want to create a classroom, so that students can join and participate in the course.</p> <p>Priority: High</p>	<ol style="list-style-type: none"> <li>1. The user can create a classroom by specifying details like course name, description, schedule, and participants.</li> <li>2. The system sends notifications to the invited students once the classroom is created.</li> </ol>
--------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>4) As a student, I want to enroll in courses, so that I can access the classroom content and attend lectures.</p> <p>Priority: High</p>	<ol style="list-style-type: none"> <li>1. The student can enroll in courses based on course availability and eligibility.</li> <li>2. The system confirms the enrollment and grants access to the course content and schedule.</li> </ol>
--------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>5) As a teacher, I want to schedule lectures, so that students can attend live sessions.</p> <p>Priority: High</p>	<ol style="list-style-type: none"> <li>1. The teacher can schedule lectures by selecting dates and times.</li> <li>2. The system should allow the teacher to telecast live lectures.</li> </ol>
<p>6) As a student, I want to attend live lectures, so that I can participate in real time or review materials at my convenience.</p> <p>Priority: High</p>	<ol style="list-style-type: none"> <li>1. The student can join live lectures through the application.</li> </ol>
<p>7) As a teacher, I want to create assignments, so that students can check their progress in the course.</p> <p>Priority: High</p>	<ol style="list-style-type: none"> <li>1. The teacher can create assignments, with various formats.</li> </ol>

<p>8) As a student, I want to submit assignments online, so that I can complete my coursework within the platform.</p> <p>Priority: High</p>	<ol style="list-style-type: none"> <li>1. The student can upload files or write text to submit assignments.</li> <li>2. The system should confirm successful submission and notify the student.</li> <li>3. Late submissions should be flagged for teacher review.</li> </ol>
<p>9) As a student/teacher, I want to participate in class discussions, so that I can engage with course content and other participants.</p> <p>Priority: Medium</p>	<ol style="list-style-type: none"> <li>1. Users should be able to post and reply to messages in discussion forums specific to their classes.</li> <li>2. The forum should support text, images, and links, and allow moderation by teachers or admins.</li> </ol>

<p>10) As a teacher, I want to manage course materials, so that students have access to the necessary resources.</p> <p>Priority: Medium</p>	<ol style="list-style-type: none"> <li>1. The teacher can upload, update, and organize course materials such as PDFs, slides, and videos.</li> <li>2. Students should be able to download or view these materials within the platform.</li> </ol>
--------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>11) As a student/teacher/admin, I want to receive and send announcements, so that I stay informed about important updates like class schedules, assignments, and system changes.</p> <p>Priority: Medium</p>	<ol style="list-style-type: none"> <li>1. Users can receive notifications for announcements via the platform and email.</li> <li>2. Admins and teachers should be able to create and broadcast announcements to all relevant users.</li> </ol>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## **7. SPRINTS:**

### **1. User Management and Basic Authentication**

- Student/Teacher - Create Account , Log In/Out, Signup, Forgot and reset password

#### **Details :**

This sprint focuses on implementing core user management features for Learnify, including user account creation, login/logout functionality, and user signup. It also covers the "forgot password" and password reset functionalities to allow users to recover or change their credentials securely. These essential features are critical for setting up a solid foundation for user interactions and will provide the necessary infrastructure for future additions, ensuring a smooth user experience from the start.

### **2. Classroom and Course Management**

- Teacher - Classroom Creation
- Student - Course Enrollment
- Teacher - Manages course (Lessons, Course materials)

#### **Details :**

This sprint focuses on enabling teachers to create classrooms and manage courses, including adding lessons and course materials such as books. It also allows students to enroll in these courses, ensuring they can access the content provided. These features establish the foundation for organizing and distributing course content while enabling student participation and engagement. By the

end of this sprint, both teachers and students will have the essential tools for course and classroom management within the platform.

### **3. Assignment Management**

- Teacher – Create Assignments
- Student – Submit Assignments

#### **Details :**

This sprint focuses on the creation and management of assignments within the platform. Teachers can create assignments and set deadlines, while students can submit their completed assignments through the system. This feature streamlines the process of distributing and collecting work, allowing for better organization and tracking of student progress. By the end of this sprint, both teachers and students will have a clear, efficient process for managing assignments.

### **4. Discussion Forums and Announcements**

- Student, Teacher - Participate in Discussion Forum
- Student, Teacher - Announcements

#### **Details :**

This sprint focuses on implementing discussion forums and announcement features to facilitate communication between students and teachers. Students and teachers can engage in discussions, ask questions, and provide answers through the forum. Additionally, instructors or admins can post announcements to notify users about important updates, deadlines,

or events. These features improve interaction and keep all users informed, promoting a collaborative and transparent learning environment.

## **8. FUTURE WORK:**

The remaining work for the project includes several important features aimed at improving both user experience and platform functionality. Live Lecture Sessions will allow students to join live classes, facilitating real-time interaction between instructors and learners. This feature aims to bridge the gap between virtual and traditional classrooms, enhancing the learning experience.

Additionally, Feedback Management will enable students to provide feedback on courses and individual lectures. This will help instructors improve their teaching methods and adapt content to better meet student needs. Another key feature to be added is the Live Project Demonstration section, where students can showcase their projects in real-time. This will provide an interactive platform for peer and instructor review, fostering a collaborative and innovative learning environment.

These updates will significantly enhance the platform's educational value, ensuring a more engaging and interactive virtual classroom experience.