



Dhirubhai Ambani Institute of Information and Communication Technology

Gandhinagar, Gujarat

IT-314

## Software Engineering

(Prof. Saurabh Tiwari)

### Online Forum

#### Software Requirements Specification

#### Group – 21

202201260 – Dishant Patel  
202201262 – Meet Chauhan  
202201268 – Jaydatt Patel  
202201289 – Adit Shah  
202201294 – Vivek Chaudhari  
202201310 – Manan Patel  
202201311 – Dharmik Godhani  
202201332 – Khushi Vora  
202201340 – Jimit Mehta  
202201358 – Jayesh Padiya  
202201370 – Sunay Revad

# 1 Index

## Contents

|           |  |           |
|-----------|--|-----------|
| <b>1</b>  | <b>Index</b>   | <b>1</b>  |
| <b>2</b>  | <b>Introduction</b>                                    | <b>2</b>  |
| <b>3</b>  | <b>Problem Description</b>                             | <b>2</b>  |
| <b>4</b>  | <b>Scope of the Project</b>                            | <b>2</b>  |
| 4.1       | 1. Admin Features . . . . .                            | 3         |
| 4.2       | 2. User Features . . . . .                             | 3         |
| 4.3       | 3. Moderator Features . . . . .                        | 3         |
| 4.4       | 4. Common Features . . . . .                           | 3         |
| 4.5       | 5. Technical Scope . . . . .                           | 4         |
| <b>5</b>  | <b>Tools and Technologies used Used</b>                | <b>4</b>  |
| <b>6</b>  | <b>Overall Description of the Online Forum Project</b> | <b>4</b>  |
| 6.1       | Users and Stakeholders . . . . .                       | 4         |
| 6.2       | Constraints . . . . .                                  | 4         |
| 6.3       | Assumptions and Dependencies . . . . .                 | 5         |
| <b>7</b>  | <b>Requirement Analysis</b>                            | <b>5</b>  |
| 7.1       | Functional Requirements . . . . .                      | 5         |
| 7.2       | Non-Functional Requirements . . . . .                  | 6         |
| <b>8</b>  | <b>Use Cases</b>                                       | <b>6</b>  |
| <b>9</b>  | <b>Use Case Description</b>                            | <b>8</b>  |
| <b>10</b> | <b>User Stories</b>                                    | <b>15</b> |

## 2 Introduction

SRS (Software Requirements Specification) is a document that serves as a blueprint for software development. It defines the purpose, functionalities, and constraints of the system being developed. For the project of a website for an online forum, the SRS document provides a comprehensive understanding of the requirements, ensuring alignment and clarity throughout the software development lifecycle.

## 3 Problem Description

In the current digital ecosystem, engaging in meaningful discussions and accessing expert knowledge involves challenges such as finding relevant communities, participating in productive conversations, and obtaining accurate and helpful information. Traditional forums and platforms often lack effective organization, making it difficult for users to find like-minded individuals or experts to engage with. Furthermore, there is no centralized system that supports seamless interaction between users, experts, and communities.

To address these issues, the online forum provides a streamlined, user-friendly platform that connects users with experts and diverse perspectives. This system enhances the user experience by organizing discussions, making it easier for users to engage with others, share knowledge, and seek support.

The platform caters to four types of users:

- **Admin:** Manages users and content, oversees the platform, enforces community guidelines, and ensures the smooth operation of the forum.
- **Users:** Participate in discussions, share knowledge, ask questions, and seek support from others in the community.
- **Experts:** Provide valuable insights, contribute to discussions, and offer guidance on specific topics.
- **Sponsors:** Organizations or individuals who support the platform through funding or promoting content, enabling the forum to remain sustainable and enhance user experience.

By centralizing and organizing these functionalities, the online forum aims to reduce complexity, foster productive discussions, and provide a seamless experience for all users, while ensuring the platform remains supported and sustainable.

## 4 Scope of the Project

The Online Forum project is designed to provide a flexible, reliable, and scalable platform for users to interact, share content, and engage in discussions. The scope includes the following:

**4.1 1. Admin Features**

- Manage user accounts (create, update, delete).
- Moderate posts and comments (approve, delete, or report).
- Monitor the overall system functionality and user activity.
- Manage privacy settings and user roles.
- Generate reports on user activity and content moderation.

**4.2 2. User Features**

- Register, log in, and manage their own profile.
- Create and interact with posts (like, comment, share).
- Follow other users to prioritize their posts in the feed.
- Bookmark posts for future reference.
- Search for posts, spaces, or users using keywords and tags.
- Report inappropriate content (posts, comments) to moderators.

**4.3 3. Moderator Features**

- Monitor and moderate user-generated content.
- Approve or delete flagged content.
- Manage reported posts and comments.
- Ensure compliance with community guidelines and privacy policies.

**4.4 4. Common Features**

- User authentication and role-based access control (Admin, User, Moderator).
- Responsive and intuitive user interface for all users (Admin, User, Moderator).
- Notifications for activities like new posts, comments, and followers.

## 4.5 5. Technical Scope

- The system will be a web-based application, accessible via any modern browser.
- Secure storage and transmission of sensitive data, including user details, posts, and comments.
- Scalable infrastructure to handle a large number of users and posts concurrently.
- Integration with third-party services like social media sharing and email notifications.

## 5 Tools and Technologies used Used

**Front-End** - React.js, Bootstrap, Tailwind CSS, Figma

**Back-End** - Node.js, Express.js, Git, GitHub

**Database** - MongoDB

**GUI Testing** - Selenium

**Load Testing** - Apache, Load Tester

## 6 Overall Description of the Online Forum Project

### 6.1 Users and Stakeholders

#### 1. Users:

- **Admin:** Manages user accounts, moderates posts, monitors content, and ensures system functionality.
- **User:** Registers, creates and interacts with posts, follows other users, comments, bookmarks content, and engages with the community.
- **Moderator:** Monitors posts and comments for inappropriate content, manages reports, and enforces community guidelines.

#### 2. Stakeholders:

- Educational institutions, community leaders, students, users, moderators, IT administrators, and the development team.

### 6.2 Constraints

- **System:** Must handle multiple users simultaneously, ensure secure access, and be scalable for future growth.
- **Technology:** Web-based platform, accessible via standard browsers, with a robust and scalable database.

- **Regulations:** Must comply with data protection laws (such as GDPR), institutional policies, and user privacy standards.

### 6.3 Assumptions and Dependencies

- Users will have basic technical skills, a stable internet connection, and access to standard browsers.
- Input data (posts, user details) will be accurate and consistent.
- Third-party services (email notifications, social media integrations, etc.) will function as expected.
- User authentication and privacy settings will be implemented effectively to ensure secure user interactions.

## 7 Requirement Analysis

### 7.1 Functional Requirements

1. **User Registration & Authentication:** New users can create an account, and old users can log into their existing accounts to personalize their experience.
2. **Search Space/Post:** Users can look up their favorite spaces to discuss what is on their mind.
3. **Create Space:** Users can create spaces and tag those spaces to build a community based on their interests.
4. **Post a Question:** Users can post questions within a space to start a discussion.
5. **Post an Answer:** Users can post answers to questions within a space.
6. **Bookmarks:** Users can bookmark their favorite posts to look at them whenever they want.
7. **Post/Interaction History:** Users can view their posts and interactions, as well as those of others.
8. **Upvote/Downvote:** Users can upvote or downvote any post or comment based on their perception.
9. **View Answers:** Users can view the answers to posted questions.
10. **Creator Analytics:** Users can check data about their content's impressions and engagements.
11. **Edit Profile:** Users can edit their profile details, including username and bio.

12. **Change Password:** Users can change their password to maintain account security.
13. **Logout:** Users can log out of their accounts when they are done using the platform.
14. **Password Reset:** Users can reset their password if forgotten, via email verification.
15. **View Spaces:** Users can browse and view existing spaces and join discussions.
16. **View Comments:** Users can view comments on posts and engage with other users.
17. **Notification for Posts:** Users will receive notifications for new posts in spaces they follow.

## 7.2 Non-Functional Requirements

1. **Scalability:** The website should be able to handle large traffic without crashing.
2. **User-Friendly (UI/UX):** Users should be able to interact easily with the website.
3. **Easy Maintenance:** It should be easy to maintain, update, and modify the website for developers.
4. **Performance (Low Latency):** The website should handle large numbers of critical requests with minimal delay.
5. **Security & Privacy:** Private user data must be protected from data leaks and attacks.
6. **Reliability:** The website should maintain optimal performance over a stated period.
7. **Availability:** Refers to the proportion of time the system is operational and accessible.

## 8 Use Cases

1. **Admin - Login:** Admin can log in to the platform using valid credentials.
2. **Admin - Create Student Accounts:** Admin can create new student accounts by entering user details.
3. **Admin - Manage Student Accounts:** Admin can add, update, or delete student accounts.

4. **Admin - Create Space:** Admin can create new spaces for users to interact based on interests.
5. **Admin - Reset Password:** Admin can reset their password using old password verification.
6. **Admin - View System Performance Reports:** Admin can view reports on system usage and performance.
7. **Admin - Send Notifications:** Admin can send notifications to users regarding important updates.
8. **User - Login:** Users can log in to the platform using valid credentials.
9. **User - Sign Up:** New users can create an account by providing necessary details.
10. **User - Search Posts/Spaces:** Users can search for specific posts or spaces to find discussions of interest.
11. **User - Create Post:** Users can create a new post in a space to ask questions or start discussions.
12. **User - Post an Answer:** Users can answer questions posted by others in a space.
13. **User - View Answers:** Users can view answers posted in response to their questions.
14. **User - Edit Profile:** Users can edit their profile, including details like username, email, and bio.
15. **User - Change Password:** Users can change their password to maintain account security.
16. **User - Logout:** Users can log out of their account when done.
17. **User - Bookmark Posts:** Users can bookmark posts they want to view later.
18. **User - Upvote/Downvote:** Users can upvote or downvote posts and comments based on their preference.
19. **User - View Post History:** Users can view their past posts and interactions on the platform.
20. **User - Receive Notifications:** Users can receive notifications about new posts, answers, and updates in followed spaces.
21. **User - Reset Password:** Users can reset their password via email verification if forgotten.



22. **User - View Previous Post/Interaction History:** Users can view a history of their posts and interactions with others.
23. **User - View Analytics:** Users can view analytics of their posts' engagement, such as views and interactions.

## 9 Use Case Description

1. **Admin - Login:** Admin can log in to the platform using valid credentials.

- **Actors:** Admin
- **Preconditions:**
  - Admin has a valid account.
  - Admin has access to the platform.
- **Main Flow:**
  - (a) Admin navigates to the admin login page.
  - (b) The system displays the login form.
  - (c) Admin enters email and password.
  - (d) The system validates the credentials:
    - Check if email exists.
    - Verify if the password is correct.
  - (e) If credentials are valid, the system redirects the admin to the admin dashboard.
- **Alternative Flow:**
  - If any required fields are missing, the system prompts the admin to complete the form.
  - If credentials are invalid, the system displays an error message prompting retry.
- **Postconditions:** Admin is logged in and redirected to the admin dashboard.

2. **Admin - Create Student Accounts:** Admin can create new student accounts by entering user details.

- **Actors:** Admin
- **Preconditions:**
  - Admin is logged in.
  - Admin has permissions to create accounts.
- **Main Flow:**
  - (a) Admin navigates to the "Create Student Account" page.
  - (b) Admin enters student details (name, email, password, etc.).

- (c) The system validates the input.
  - (d) The system creates a new student account and displays a success message.
  - **Alternative Flow:**
    - If any fields are missing or invalid, the system prompts the admin to correct them.
  - **Postconditions:** A new student account is created.
3. **Admin - Manage Student Accounts:** Admin can add, update, or delete student accounts.
- **Actors:** Admin
  - **Preconditions:**
    - Admin is logged in.
    - Admin has permissions to manage student accounts.
  - **Main Flow:**
    - (a) Admin navigates to the "Manage Student Accounts" page.
    - (b) Admin selects a student account to modify (add, update, or delete).
    - (c) The system displays the student's details.
    - (d) Admin updates the information or deletes the account.
  - **Alternative Flow:**
    - If the student account does not exist, the system displays an error message.
  - **Postconditions:** The student account is modified or deleted.
4. **Admin - Create Space:** Admin can create new spaces for users to interact based on interests.
- **Actors:** Admin
  - **Preconditions:**
    - Admin is logged in.
    - Admin has permissions to create spaces.
  - **Main Flow:**
    - (a) Admin navigates to the "Create Space" page.
    - (b) Admin enters space details (name, description, rules, etc.).
    - (c) The system creates a new space and displays a success message.
  - **Alternative Flow:**
    - If any required fields are missing, the system prompts the admin to complete them.
  - **Postconditions:** A new space is created for user interactions.

5. **Admin - Reset Password:** Admin can reset their password using old password verification.
  - **Actors:** Admin
  - **Preconditions:**
    - Admin is logged in.
    - Admin knows their old password.
  - **Main Flow:**
    - (a) Admin navigates to the "Reset Password" page.
    - (b) Admin enters their old password and new password.
    - (c) The system validates the old password and updates the new password.
  - **Alternative Flow:**
    - If the old password is incorrect, the system prompts for a retry.
  - **Postconditions:** Admin's password is successfully reset.
6. **Admin - View System Performance Reports:** Admin can view reports on system usage and performance.
  - **Actors:** Admin
  - **Preconditions:**
    - Admin is logged in.
  - **Main Flow:**
    - (a) Admin navigates to the "System Performance Reports" page.
    - (b) The system displays reports on system usage, performance, and statistics.
  - **Postconditions:** Admin can view the system reports.
7. **Admin - Send Notifications:** Admin can send notifications to users regarding important updates.
  - **Actors:** Admin
  - **Preconditions:**
    - Admin is logged in.
  - **Main Flow:**
    - (a) Admin navigates to the "Send Notifications" page.
    - (b) Admin enters the notification content and selects users or groups to send it to.
    - (c) The system sends the notification to the selected users.
  - **Postconditions:** Notification is sent to the users.
8. **User - Login:** Users can log in to the platform using valid credentials.

- **Actors:** User
  - **Preconditions:**
    - User has a valid account.
  - **Main Flow:**
    - (a) User navigates to the login page.
    - (b) User enters email and password.
    - (c) The system validates the credentials and logs the user in.
  - **Postconditions:** User is logged in to the platform.
9. **User - Sign Up:** New users can create an account by providing necessary details.
- **Actors:** User
  - **Preconditions:**
    - User is not registered on the platform.
  - **Main Flow:**
    - (a) User navigates to the sign-up page.
    - (b) User enters required details (username, email, password, etc.).
    - (c) The system validates the input.
    - (d) The system creates a new account for the user and logs them in.
  - **Alternative Flow:**
    - If any required fields are missing or invalid, the system prompts the user to correct them.
  - **Postconditions:** A new user account is created and user is logged in.
10. **User - Search Posts/Spaces:** Users can search for specific posts or spaces to find discussions of interest.
- **Actors:** User
  - **Preconditions:**
    - User is logged in.
  - **Main Flow:**
    - (a) User enters search keywords in the search bar.
    - (b) The system displays relevant posts and spaces based on the search query.
  - **Postconditions:** User is presented with search results.
11. **User - Create Post:** Users can create a new post in a space to ask questions or start discussions.
- **Actors:** User

- **Preconditions:**
    - User is logged in.
    - User has selected a space to post in.
  - **Main Flow:**
    - (a) User clicks on "Create Post" within a space.
    - (b) User enters the post title and content.
    - (c) The system validates the content and creates the post in the selected space.
  - **Postconditions:** The post is created and displayed within the space.
12. **User - Post an Answer:** Users can answer questions posted by others in a space.
- **Actors:** User
  - **Preconditions:**
    - User is logged in.
    - User is viewing a post with a question.
  - **Main Flow:**
    - (a) User clicks on the "Answer" button for a post.
    - (b) User enters their answer in the provided field.
    - (c) The system saves and displays the answer under the post.
  - **Postconditions:** The user's answer is visible on the post.
13. **User - View Answers:** Users can view answers posted in response to their questions.
- **Actors:** User
  - **Preconditions:**
    - User is logged in.
    - User has posted a question in a space.
  - **Main Flow:**
    - (a) User navigates to the post they created.
    - (b) The system displays the answers posted by other users.
  - **Postconditions:** User can view the answers to their post.
14. **User - Edit Profile:** Users can edit their profile, including details like username, email, and bio.
- **Actors:** User
  - **Preconditions:**
    - User is logged in.

- **Main Flow:**
    - (a) User navigates to the "Edit Profile" page.
    - (b) User updates details (username, email, bio, etc.).
    - (c) The system saves the updated profile.
  - **Postconditions:** User's profile is updated.
15. **User - Change Password:** Users can change their password to maintain account security.
- **Actors:** User
  - **Preconditions:**
    - User is logged in.
    - User knows their current password.
  - **Main Flow:**
    - (a) User navigates to the "Change Password" page.
    - (b) User enters current password, then the new password.
    - (c) The system validates the old password and updates to the new password.
  - **Postconditions:** User's password is successfully changed.
16. **User - Logout:** Users can log out of their account when done.
- **Actors:** User
  - **Preconditions:**
    - User is logged in.
  - **Main Flow:**
    - (a) User clicks on the "Logout" button.
    - (b) The system logs the user out and redirects them to the login page.
  - **Postconditions:** User is logged out and redirected to the login page.
17. **User - Bookmark Posts:** Users can bookmark posts they want to view later.
- **Actors:** User
  - **Preconditions:**
    - User is logged in.
  - **Main Flow:**
    - (a) User clicks on the "Bookmark" icon on a post.
    - (b) The system saves the post in the user's bookmarked posts.
  - **Postconditions:** The post is saved in the user's bookmarks.

18. **User - Upvote/Downvote:** Users can upvote or downvote posts and comments based on their preference.
- **Actors:** User
  - **Preconditions:**
    - User is logged in.
    - User is viewing a post or comment.
  - **Main Flow:**
    - (a) User clicks the upvote or downvote button.
    - (b) The system updates the vote count for the post or comment.
  - **Postconditions:** Post or comment receives an updated vote count.
19. **User - View Post History:** Users can view their past posts and interactions on the platform.
- **Actors:** User
  - **Preconditions:**
    - User is logged in.
  - **Main Flow:**
    - (a) User navigates to their profile.
    - (b) User clicks on the "Post History" tab.
    - (c) The system displays a list of the user's past posts and interactions.
  - **Postconditions:** User can view their post history.
20. **User - Receive Notifications:** Users can receive notifications about new posts, answers, and updates in followed spaces.
- **Actors:** User
  - **Preconditions:**
    - User is logged in.
    - User has followed spaces and/or posts.
  - **Main Flow:**
    - (a) The system sends notifications about new posts, answers, and updates.
    - (b) User receives the notifications via email or in-app.
  - **Postconditions:** User is notified about relevant activity.
21. **User - Reset Password:** Users can reset their password via email verification if forgotten.
- **Actors:** User

- **Preconditions:**
    - User has forgotten their password.
  - **Main Flow:**
    - (a) User clicks on the "Forgot Password" link.
    - (b) User enters their email address.
    - (c) The system sends a password reset link to the user's email.
    - (d) User clicks the reset link and sets a new password.
  - **Postconditions:** User's password is reset.
22. **User - View Previous Post/Interaction History:** Users can view a history of their posts and interactions with others.
- **Actors:** User
  - **Preconditions:**
    - User is logged in.
  - **Main Flow:**
    - (a) User navigates to their profile.
    - (b) User clicks on the "Previous Interactions" tab.
    - (c) The system displays a history of the user's past posts and interactions.
  - **Postconditions:** User can view their past interactions.
23. **User - View Analytics:** Users can view analytics of their posts' engagement, such as views and interactions.
- **Actors:** User
  - **Preconditions:**
    - User is logged in.
  - **Main Flow:**
    - (a) User navigates to the "Analytics" section of their profile.
    - (b) The system displays engagement statistics for their posts (views, upvotes, comments, etc.).
  - **Postconditions:** User can view the engagement statistics for their posts.

## 10 User Stories



## Actor – USER

| User Story  | Acceptance Criteria  | Priority    |
|---|--|-------------|
| As a user, I want to be able to view another user's profile so that I learn more about their contributions. | <ul style="list-style-type: none"> <li>● If the profile is private, then the other user cannot view the content posted by them.</li> <li>● The profile should display user's profile picture, bio, number of followers, number of users they are following, recent posts (if profile is public), comments, and followed spaces.</li> </ul>   | Must Have   |
| As a user I want to login to my account so that I can access the complete range of features.                | <ul style="list-style-type: none"> <li>● A user must enter a valid email address and password.</li> <li>● If the user has forgotten password, then they should get an email to their registered email id to reset their password.</li> </ul>   | Must Have   |
| As a user, I want to be able to follow other users so that I can view their latest posts on priority basis. | <ul style="list-style-type: none"> <li>● If a user's profile is private, others can follow them only if they permit them.</li> <li>● User's feed prioritizes the post made by the users they follow.</li> <li>● (Based on discussion) User is notified when their followed user posts.</li> </ul>  | Should Have |
| As a user, I want to view and edit my own profile so that I can update my personal details.                 | <ul style="list-style-type: none"> <li>● If the user is logged in, then only he/she can view and edit his/her profile.</li> <li>● The system should save the entered data with a success message.</li> <li>● There should be a reset option besides a save changes button.</li> <li>● The user should be able to update the profile picture, bio and contact information.</li> <li>● There should be privacy settings for making the profile private or public, private meaning only followers can see the profile details.</li> </ul> | Must Have   |
| As a user, I want to view a post so that I can remain updated about topics which interest me.               | <ul style="list-style-type: none"> <li>● If the post is private, only the users which follow the user can interact with it.</li> <li>● The post should display the option to view the author's profile by clicking on their name or avatar.</li> <li>● The author's posts should include an "Edit" or "Delete" option if they are viewing their own post.</li> <li>● Users should be able to share the post via social media, direct link, or email.</li> </ul>  | Must Have   |

|  |  |             |
|--|--|-------------|
| As a user I want to be able to comment on another user's post so that I can engage in a discussion with them.                        | <ul style="list-style-type: none"> <li>● A user must be logged in to be able to comment on another user's post.</li> <li>● A user can comment on a private post if and only if they follow the user.</li> <li>● A user should be able to edit/delete their comment.</li> </ul>   | Must Have   |
| As a user, I want to search for posts or spaces so that I can find relevant information easily.                                      | <ul style="list-style-type: none"> <li>● The search function should allow users to search by keyword, tags, or author.</li> <li>● Search results should include both posts and spaces.</li> </ul>  | Must Have   |
| As a user, I want to create posts so that I can share my thoughts and engage with the community.                                     | <ul style="list-style-type: none"> <li>● The post creation page should allow users to write text, upload images, and attach files.</li> <li>● Users should be able to assign tags and select spaces where the post will be shared.</li> <li>● A draft should be saved automatically, and a confirmation message should appear after posting.</li> </ul>  | Must Have   |
| As a user, I want to bookmark posts so that I can save them for future reference.  | <ul style="list-style-type: none"> <li>● Users should be able to bookmark any post and access it from their profile.</li> <li>● Bookmarked posts should be categorized by date or tags for easy retrieval.</li> <li>● A notification should confirm when a post is successfully bookmarked.</li> </ul>   | Could Have  |
| As a user, I want to report any inappropriate comment or post so that I can help moderator address harmful content in the community. | <ul style="list-style-type: none"> <li>● A "Report" button should be visible next to every post and comment.</li> <li>● If a user is found to abuse the report feature, moderators should have the ability to restrict their reporting capabilities.</li> <li>● The system should automatically notify moderators when a post or comment is reported, along with the selected reason and reporter's username.</li> </ul> | Should Have |
| As a user, I want to upvote or downvote posts to express my opinion on their quality or relevance.                                   | <ul style="list-style-type: none"> <li>● Users should be able to click an upvote or downvote button on each post.</li> <li>● The system should display the total number of upvotes and downvotes.</li> </ul>   | Should Have |

|   |   |             |
|---|---|-------------|
| As a user I want to block other users so that I can restrict some users from interacting with me.                             | <ul style="list-style-type: none"> <li>• User should be logged in to be able to block someone.</li> <li>• If a user is blocked by someone, they cannot view their ID or posts.</li> </ul>   | Should Have |
| As a user I want to apply filters so that I can find my preferred spaces easily.  | <ul style="list-style-type: none"> <li>• Spaces with that tag assigned to them should come up.</li> </ul>   | Should Have |
| As a user I want to join a space so that I can get involved in discussions which interest me.                                 | <ul style="list-style-type: none"> <li>• A user must be logged in to join a space.</li> <li>• If a space is private, then the user can join the space only if the admin permits them.</li> </ul>  | Must Have   |
| As a user I want to assign tags to the spaces I create so that I can attract people with like interests to join my space.     | <ul style="list-style-type: none"> <li>• Users can assign multiple tags restricted to a limit to their space.</li> <li>• Users can add custom tags following a certain character limit.</li> <li>• System should prevent users from adding duplicate tags.</li> </ul> | Should Have |
| As a user I want to add description to my space so that an interested user can get to know what my space is about.            | <ul style="list-style-type: none"> <li>• A user must have logged.</li> <li>• User must have created space to add the description.</li> </ul>  | Could Have  |
| As a user I want to assign moderator role to others so that they can help me keep a positive and well-maintained environment. | <ul style="list-style-type: none"> <li>• Only the space creator can assign/revoke moderator role.</li> <li>• The user to be assigned moderator should be member of the space.</li> <li>• Moderators should be separately marked in the space.</li> </ul>              | Should Have |

### Actor – MODERATOR

|   |   |             |
|---|---|-------------|
| As a moderator, I want to issue warnings to users so that I can maintain community standards. | <ul style="list-style-type: none"> <li>• Moderators should be able to issue warnings through a user's profile or post.</li> <li>• Warnings should include a predefined reason and optional custom message.</li> </ul> | Should Have |
|---|---|-------------|

|  |   |             |
|--|---|-------------|
|  | <ul style="list-style-type: none"> <li>• Users should be notified of the warning and have the ability to respond or appeal.</li> </ul>  |             |
| As a moderator, I want to delete inappropriate posts so that I can keep the space safe and clean.  | <ul style="list-style-type: none"> <li>• Moderators should be able to delete any post in the community.</li> <li>• Deleted posts should be logged, with reasons provided for deletion.</li> <li>• The system should notify the post creator about the deletion and provide a link to community guidelines.</li> </ul> | Must Have   |
| As a moderator, I want to ban users who violate rules so that I can protect the community.   | <ul style="list-style-type: none"> <li>• Moderators should have the ability to suspend a user permanently or temporarily from a space.</li> <li>• The system should record the suspension, the reason and notify the user.</li> </ul>   | Must Have   |
| As a moderator, I want to be able to modify the space settings so that I can determine the format of content which can be uploaded on the space.                             | <ul style="list-style-type: none"> <li>• Link should be from secure sites.</li> <li>• Who can comment on posts.</li> <li>• Determine type of posts the permitter users can make on the space.</li> </ul>  | Should Have |
| As a space moderator, I want to temporarily mute users who are being disruptive in discussions, so that I can allow conversations to continue without further interruptions. | <ul style="list-style-type: none"> <li>• The moderator can mute a user for a predefined period (e.g., 1 hour, 24 hours).</li> <li>• Muted users cannot post or comment during the mute period but can still read content.</li> </ul>  | Could Have  |
| As a moderator I want to review user appeals for ban or suspension so I can reconsider or revoke the actions if necessary.   | <ul style="list-style-type: none"> <li>• System for user to submit appeals.</li> <li>• Moderator can review accept or reject appeals and provide a reason.</li> </ul>   | Must Have   |

#### Actor – ADMINISTRATOR

|   |   |           |
|---|---|-----------|
| As an admin, I want to view analytics to track platform activity and engagement so that I | <ul style="list-style-type: none"> <li>• The system should display statistics on user activity, post engagement, and space growth.</li> </ul> | Must Have |
|---|---|-----------|

|  |  |             |
|--|--|-------------|
| can improve user retention.  | <ul style="list-style-type: none"> <li>● Analytics should include filters for specific time periods and spaces.</li> <li>● Admins should be able to export analytics reports for further analysis.</li> </ul>  |             |
| As an admin, I want to manage sponsors so that I can facilitate funding and support for the platform.  | <ul style="list-style-type: none"> <li>● Admins should be able to add, remove or edit sponsors.</li> <li>● The system should display sponsor logos or ads on specific pages or spaces.</li> <li>● Admins should be able to track engagement with sponsor content.</li> </ul> | Should Have |
| As an admin, I want to review reports and delete the accounts of users who violate community guidelines so that I can keep the platform healthy for informative discussions. | <ul style="list-style-type: none"> <li>● Admins should be able to permanently delete users, including their posts and comments.</li> <li>● Deletion should trigger a notification to the user and generate a log for future reference.</li> </ul>                            | Must Have   |
| As an admin, I want to review new user account, so I can prevent bots and spammers from joining the forum.   | <ul style="list-style-type: none"> <li>● Captcha to identify bots.</li> <li>● Automated flagging for suspicious activities.</li> </ul>   | Could Have  |
| As an admin, I want to temporarily suspend a user's account for a specific time(suspend), so that I can prevent further violations without permanently banning the user.     | <ul style="list-style-type: none"> <li>● The admin can lock user accounts for predefined periods (e.g., 24 hours, 7 days).</li> <li>● The system automatically unlocks the user's account after the set period.</li> </ul>   | Could Have  |