

Software Requirements Specification

An Online Social Media Platform

(SM02)

Group 9

Harshit Pant
CS21BTECH11021

Satpute Aniket Tukaram
CS21TEBCH11056

Mahin Bansal
CS21BTECH11034

Burra Vishal Mathews
CS21BTECH11010

Contents

		3.4.9	Chat Screen	22
		3.4.10	Profile Screen-Own	22
		3.4.11	Profile Screen-Other	22
1	Introduction			1
1.1	Purpose			1
1.2	Scope			1
1.3	Overview			2
2	Overall Description			2
2.1	Product Perspective			2
2.2	Product Functions			2
2.3	User Characteristics			3
2.4	Principal Actors			3
2.5	Assumptions and Dependencies . . .			4
3	Specific Requirements			4
3.1	Functional Requirements			4
3.1.1	<u>Authentication:</u>			4
3.1.2	<u>User Profiles:</u>			5
3.1.3	<u>Communities:</u>			8
3.1.4	<u>Posts:</u>			11
3.1.5	<u>Comments:</u>			13
3.1.6	<u>Notifications:</u>			14
3.1.7	<u>Admin privileges:</u>			15
3.1.8	<u>Moderator privileges:</u>			15
3.1.9	<u>Search:</u>			18
3.1.10	<u>Recommendations/User Feed:</u>			18
3.1.11	<u>Trending Page:</u>			19
3.1.12	<u>Direct-messages:</u>			19
3.1.13	<u>Superuser privileges:</u>			20
3.2	Performance Requirements			21
3.3	Design Constraints			21
3.4	External Interface Requirements . . .			21
3.4.1	Login Screen			21
3.4.2	Signup Screen			21
3.4.3	Home Screen-Registered User			21
3.4.4	Home Screen-Guest User			22
3.4.5	Notification Screen			22
3.4.6	Community Screen			22
3.4.7	Community Creation Screen			22
3.4.8	Post Creation Screen			22
4	Further Extensions			23
1.	Introduction			
1.1.	Purpose			
	The purpose of is to provide a platform where users can create communities of their choice and indulge in communities of similar interests.			
	This document is meant to delineate the features of and serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.			
1.2.	Scope			
	We describe what features are in-scope nad what are not in-scope for the software product.			
	In-scope:			
	<ul style="list-style-type: none"> • Guest Users can view public communities, posts and other registered users with public profiles. • Guest Users can view the Trending page. • User authentication including login and signup and a referral system. • Registered users can create communities with different visibilities and can join communities. • Users can create posts in communities and perform other CRUD operations. They can comment on posts. • Users can upvote and downvote posts and comments. • Users can report posts and comments. • Users can search for communities, users and posts. • Users can follow other users. • Users can view their user-specific-feed and their notifications. 			

- Users can use a direct messaging system to communicate with other users
- **Community moderators** can moderate posts and comments in their communities.
- Community moderators can ban users from their communities.
- Community moderators can issue warnings to members of their communities.
- **Community admins** can add and remove moderators from their communities.
- Community admins can delete their communities.
- Community admins can revoke user-bans.
- **Platform admins/superusers** can delete users and communities.

Out-of-scope:

- Interface specific for group chats. Group chats can be indirectly achieved by creating a community with visibility set to private.
- Any real-time features like live chat, live feed, etc.

1.3. Overview

The rest of this SRS document is organized as follows:

- Section 2 describes the overall description of the product.
- Section 3 describes the specific requirements of the product and the functional requirements of the product using use-case analysis.
- Section 4 gives some further extensions.
- Section 5 is the Appendix for the document.

2. Overall Description

2.1. Product Perspective

This product intends to be a social media platform where users can create communities of their choice with different visibilities and indulge in communities of similar interests. Users can also interact with other users through direct messaging.

Inorder to prevent malicious users, the platform also employs a moderation system where users can report posts and comments and moderators can moderate posts and comments in their communities.

The product also intends to employ a recommender system to recommend communities and create a user-specific feed for each user.

The product is a web application and hence is intended to be independent of any underlying operating system.

2.2. Product Functions

Class of use case	Use case	Description of use case
Authentication	Signup	Signup a user using username and password
	Signup using referral code. Join via link	Signup a user using referral code.
	Login via username and password	Login a user using username and password
	Login via Google OAuth	Login a user using Google OAuth
	Forgot password	Reset password of a user by sending an email to registered email id.
	Change Password	Change password of a user
	Logout	Logout a user
Guest Users	Signup	Signup a user using one of the above methods
	Search for public or request only communities, users and posts	Search for communities, users and posts
	Fuzzy search	Search for communities, users and posts using fuzzy search
	View communities	View public community pages and posts
	View posts	View public posts
	View users	View public user profiles
	View trending page	View trending posts
Registered Users	Create community	Create a community based on provided community settings.
	Join community	Join a public community
	Send join request	Send a join request to a community which is set to request only
	Accept a community invite	Accept a private community invite sent by a community admin/moderator.
	Search for communities, users and posts	Applicable for public communities, request only community, public user profiles, follower profiles and posts in public communities and posts in communities joined by the user.
	Create posts	Create a post in a community joined by the user if the user has the privilege.
	Comment on posts	Comment on posts in communities joined by the user if the post has comments on.
	Reply to comment	Reply to comments on posts in communities joined by the user

Registered Users	Receive comment notifications	Receive notifications when a user replies to a comment made by the user or a comment is made on a post made by the user.	Moderators	Moderate posts	Moderate posts in communities where the user is a moderator
	Upvote/Downvote	Upvote/Downvote posts and comments in communities joined by the user.		Moderate comments	Moderate comments in communities where the user is a moderator
	Edit Post	Edit post created by the same user.		Ban user	Ban a user from a community where the user is a moderator
	Edit comment	Edit comment created by the same user.		Issue warning	Issue warning to a user in a community where the user is a moderator
	Delete post	Delete post created by the same user.		Give/revoke post/comment privileges	Give/revoke post/comment privileges to a user in a community where the user is a moderator.
	Delete comment	Delete comment created by the same user.	Admins	Delete community	Delete a community where the user is an admin
	Report	Report posts and comments in public communities or communities joined by the user.		Appoint/Remove moderators	Appoint/Remove moderators in a community where the user is an admin
	Follow/Unfollow profile	Follow/Unfollow a user profile.		Revoke user-ban	Revoke user-ban in a community where the user is an admin
	Block user	Block all direct-messages and posts by a user		Give/revoke moderator privileges	Give/revoke moderator privileges to a user in a community where the user is an admin.
	Edit Profile	Edit user profile	Superuser	Ban user	Ban a user from the platform.
	Edit profile visibility	Edit user profile visibility. Toggle between public and private.		Delete community	Delete a community from the platform.
	View Followers	View followers of a public user profile or your own profile.		Accept community reports	Accept community reports and delete the community.
	Send Chat request	Send chat request to a private profile which is does not follows the user		Revoke user-ban	Revoke user-ban from the platform.
	Accept Chat request	Accept chat request from a user profile			
	Chat	Chat with mutual followers directly			
	Send direct message	send message to mutual follower and profiles which have accepted chat request			
	Receive direct message	Receive direct message from mutual follower and profiles from which the user has accepted the chat request			
	View user-specific-feed	View posts from communities joined by the user and posts from profiles followed by the user in any public community or mutula community.			
	Schedule post	Schedule a post to be posted at a later time.			
	Save Posts	Save posts to view later			
	View saved posts	View saved posts			
	Send unban request	Send unban request to a community admin if the user is banned from the community.			

2.3. User Characteristics

- Users are advised to be familiar with the basic features of a social media platform.

2.4. Principal Actors

The principal actors in the system are:

- Guest Users: Users which have not created an account on the platform.
- Registered Users: Users which have created an account on the platform.
- Community Admins: Users which have created a community on the platform.
- Community Moderators: Users which have been appointed as moderators in a community.
- Superuser: Users which have the highest privileges on the platform.
- System: The system is also an actor in the system.

2.5. Assumptions and Dependencies

- The platform assumes that the user has a stable internet connection.
- The platform assumes that the user has a device with an updated web browser.
- The platform assumes that a registered user has a valid email id.
- The platform depends on Google OAuth services for authentication in some cases.

3. Specific Requirements

3.1. Functional Requirements

3.1.1. Authentication:

Use-case 1: Signup

Primary Actor: Guest User

PreCondition: Stable internet Connection and a valid email Id

Main Scenario:

- 1) Guest User clicks on the signup button.
- 2) Guest User enters the username, email id and password.
- 3) Guest User re-enters the password in confirm password field.
- 4) Guest User clicks on create account.
- 5) System checks if the username is unique.
- 6) System checks if the email id is valid by sending a confirmation email.
- 7) System checks if the email id is already registered.
- 8) User is logged into platform.
- 9) User information is stored in the database.
- 10) A default user profile is created.

Alternate Scenario:

- 3a. Guest User enters the password and confirm password fields differently.
- 5a. System displays an error message that the username is already taken.
- 6a. System displays an error message that the email id is invalid.
- 7a. System displays an error message that the email id is already registered.

Use-case 2: Join Via Link:

Primary Actor: Guest User

PreCondition: Stable internet connection, valid email Id and a join invite email from a registered user.

Main Scenario:

- 1) Guest User receives a join invite email from a registered user.

- 2) Guest User clicks on the join button in the email.
- 3) Guest User is redirected to the signup page.
- 4) Guest User enters the username and password. The email field is already filled.
- 5) Guest User re-enters the password in confirm password field.
- 6) Guest User clicks on create account.
- 7) System checks if the username is unique.
- 8) Guest User is logged into platform.
- 9) User information is stored in the database.
- 10) A default user profile is created.
- 11) Referral points are added to the user who sent the invite.

Alternate Scenario:

- 5a. Guest User enters the password and confirm password fields differently.
- 7a. System displays an error message that the username is already taken.

Use-case 3: Login via Google OAuth:

Primary Actor: Guest User or Registered User

PreCondition: Stable internet connection and a valid Google account.

Main Scenario:

- 1) User clicks on the login button.
- 2) User clicks on the Google login button.
- 3) User is redirected to the Google login page.
- 4) User enters the Google account credentials and verified by Google OAuth API
- 5) User is redirected to the platform.
- 6) If the email id is not already registered, the user is redirected to the signup page.
- 7) After successful signup, the user is logged into the platform.
- 8) User information is stored in the database.
- 9) A default user profile is created.
- 10) If the email id is already registered, the user is logged into the platform.

Alternate Scenario:

- 4a. User enters the wrong Google account credentials. User is redirected to the login page.
- 7a. User enters the wrong username or different passwords. User is redirected to the signup page.

Use-case 4: Login via username and password:

Primary Actor: Registered User

PreCondition: Stable internet connection

Main Scenario:

- 1) User clicks on the login button.
- 2) User enters the username and password.
- 3) User clicks on the login button.
- 4) User is logged into the platform.

Alternate Scenario:

- 2a. User enters the wrong username or password. User is redirected to the login page.

Use-case 5: Forgot Password:

Primary Actor: Registered User

PreCondition: Stable internet connection and a registered email ID

Main Scenario:

- 1) User clicks on the forgot password button.
- 2) User enters the registered email id.
- 3) User clicks on the send email button.
- 4) User may click on resend email after 30 seconds.
- 5) System sends an email to the registered email id with a link to reset password.
- 6) User clicks on the link.
- 7) User is redirected to the reset password page.
- 8) User enters the new password and confirm password.
- 9) User clicks on the reset password button.
- 10) User is redirected to the login page.

Alternate Scenario:

- 3a. User enters an unregistered email id. System displays an error message that the email id is not registered.
- 8a. User enters the password and confirm password fields differently.

Use-case 6: Change Password:

Primary Actor: Registered User

PreCondition: Stable internet connection and the user is logged in

Main Scenario:

- 1) User clicks on the profile button.
- 2) User clicks on the change password button.
- 3) User enters the old password, new password and confirm password.
- 4) User clicks on the change password button.
- 5) User is redirected to the login page.
- 6) User logs in with the new password.
- 7) User is redirected to the home page.

Alternate Scenario:

- 3a. User enters the new password and confirm password fields differently.

- 5b. User enters the wrong old password. System displays an error message that the old password is wrong.

Use-case 7: Logout:

Primary Actor: Registered User

PreCondition: Stable internet connection and the user is logged in

Main Scenario:

- 1) User clicks on the logout button in the dashboard.
- 2) User is redirected to the login page.

3.1.2. User Profiles:

Use-case 8: Edit Profile:

Primary Actor: Registered User

PreCondition: Stable internet connection and the user is logged in

Main Scenario:

- 1) User clicks on the profile button.
- 2) User clicks on the edit profile button.
- 3) User can edit the following fields
 - Username
 - Name
 - Gender
 - Email Id
 - Bio
 - Profile Picture
 - Preferences:
 - Privacy Settings (Public/Private)
 - Notification settings (notification on comments/replies/messages/mentions/follows)
 - Chat Settings (Allow direct chat from all users/only following/only mutual followers/no chat requests)
- 4) User clicks on the save button.
- 5) User is redirected to the profile page.
- 6) User profile information is updated in the database.

Alternate Scenario:

- 3a. User enters the username which is already taken. System displays an error message that the username is already taken.
- 3b. User enters the email id which is already registered or an invalid email id. System displays an error message that the email id is already registered.
- 3g1. User toggles privacy settings. Confirmation message is displayed.

Use-case 9: Edit Profile Visibility

Primary Actor: Registered User

PreCondition: Stable internet connection and the user is editing his/her profile

Main Scenario:

- 1) User toggles the privacy settings.
 - **Public:**
 - a) User profile is visible to all users.
 - b) User profile is visible to all users in search results.
 - c) Any user can view the user's following/follower list.
 - d) Posts made by the user on public communities and mutual communities are visible to all users on the user profile.
 - **Private:**
 - a) User profile is visible to only mutual followers/following.
 - b) User profile is visible in search results.
 - c) Only profiles that the user follows can view the user's following/follower list.
 - d) Posts made by the user on public communities and mutual communities are visible to only mutual followers on the user profile.
- 2) User clicks on the save button.
- 3) User is redirected to the profile page.
- 4) User profile information is updated in the database.

Alternate Scenario:

- 1a. User toggles privacy settings. Confirmation message is displayed.

Use Case 10: Chat Settings

Primary Actor: Registered User

PreCondition: Stable internet connection and the user is editing his/her profile

Main Scenario:

- 1) User toggles the chat settings.
 - **Allow direct chat from all users:**
 - a) Any user can send a direct message to the user.
 - **Allow direct chat from only following:**
 - a) Only profiles that the user follows can send a direct message to the user.
 - b) Other profiles can send a chat request to the user.
 - **Allow direct chat from only mutual followers:**

- a) Only mutual followers can send a direct message to the user.
- b) Other profiles can send a chat request to the user.

- **No chat requests:**

- a) Only mutual followers can send a direct message to the user.
 - b) Other profiles cannot send a chat request/direct message to the user.
- 2) User clicks on the save button.
 - 3) User is redirected to the profile page.
 - 4) User profile information is updated in the database.

Alternate Scenario:

- 1a. User toggles chat settings. Confirmation message is displayed.

Use Case 11: Notification Settings

Primary Actor: Registered User

PreCondition: Stable internet connection and the user is editing his/her profile

Main Scenario:

- 1) User toggles the notification settings. The user can turn on/off the following notifications:
 - **Notification on comments/replies:**
 - a) User receives notifications when a user comments on a post made by the user.
 - b) User receives notifications when a user replies to a comment made by the user.
 - **Notification on messages:**
 - a) User receives notifications when a user sends a direct message to the user.
 - **Notification on mentions:**
 - a) User receives notifications when a user mentions the user in a post or comment in a public/mutual community.
 - **Notification on follows:**
 - a) User receives notifications when a user follows the user.
- 2) User clicks on the save button.
- 3) User is redirected to the profile page.
- 4) User profile information is updated in the database.

Alternate Scenario:

- 1a. User toggles notification settings. Confirmation message is displayed.

Use Case 12: View Followers

Primary Actor: Registered User

PreCondition: Stable internet connection and the user is logged in

Main Scenario:

- 1) User clicks on the followers button.
- 2) User is redirected to the followers page.
- 3) User can view a list of followers.
- 4) User can click on the profile of a follower to view the profile.

Use Case 13: View Following

Primary Actor: Registered User

PreCondition: Stable internet connection and the user is logged in

Main Scenario:

- 1) User clicks on the following button.
- 2) User is redirected to the following page.
- 3) User can view a list of following.
- 4) User can click on the profile of a following to view the profile according to the visibility set by the other user.

Alternate Scenario:

- 4a. User clicks on the profile of a following which is set to private. User can only view the contents of the profile if the user is a mutual follower.
- 4b. User clicks on the profile of a following which is set to public. User can view the profile.

Use Case 14: Follow/Unfollow Profile

Primary Actor: Registered User

PreCondition: Stable internet connection and the user is viewing a profile

Main Scenario:

- 1) User clicks on the follow/unfollow button.
- 2) User profile information is updated in the database.

Use Case 15: View Other User-Profiles

Primary Actor: Registered User

PreCondition: Stable internet connection and the user is logged in

Main Scenario:

- 1) User clicks on a profile
- 2) User can view the profile of the user according to the visibility set by the other user.
- 3) Irrespective of the visibility set by the user, anyone can see the reports that were accepted against the user on their profile.

Alternate Scenario:

- 2a. User clicks on a profile which is set to private. User can only view the contents of the profile if the user is a mutual follower.
- 2b. User clicks on a profile which is set to public. User can view the profile.

Use Case 16: Block User

Primary Actor: Registered User

PreCondition: Stable internet connection and the user is viewing a profile

Main Scenario:

- 1) User clicks on the block button.
- 2) User can no longer
 - View posts made by the blocked user.
 - View comments made by the blocked user.
 - Send/Receive direct messages to the blocked user.
 - Receive chat requests from the blocked user.
 - Receive notifications from the blocked user.
- 3) User profile information is updated in the database.
- 4) User is redirected to the profile page.

Use Case 17: Report Profile

Primary Actor: Registered User

PreCondition: Stable internet connection and the user is viewing a profile

Main Scenario:

- 1) User clicks on the report button.
- 2) User can select the reason for reporting the profile.
- 3) User can enter a description for reporting the profile.
- 4) User clicks on the report button.
- 5) Report is sent to the superuser.
- 6) Report information is updated in the database.
- 7) User is redirected to the profile page.

Alternate Scenario:

- 4a. User clicks on the report button. System displays an error message that the user has already reported the profile.
- 0a. User can select one/multiple messages to report.
- 0b. User can select one/multiple comments to report.
- 0c. User can select one/multiple posts to report.
- 0d. User can select communities to report the community admin.

Use Case 18: Sending Join via link

Primary Actor: Registered User

PreCondition: Stable internet connection and the user is logged in

Main Scenario:

- 1) User clicks on the invite button.
- 2) User can select the following options:
 - Copy link
 - Share link via email
 - Share link via other social media platforms
- 3) User clicks on the copy link button.
- 4) Link is copied to the clipboard.
- 5) User can paste the link to share.

Alternate Scenario:

- 2a. User clicks on the share link via email button. User is redirected to the email page.
- 2b. User clicks on the share link via other social media platforms button. User is redirected to the social media platform.

3.1.3. Communities:

Use Case 19: Create Community

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in

Main Scenario:

- 1) User clicks on the create community button.
- 2) User enters the following community details:
 - Community Name
 - Community description
 - Community type (public/request only/invite only)
 - **Public**
 - * Anyone can join the community.
 - * Anyone can view the community page.
 - * Anyone can view the posts in the community.
 - * Anyone can view the comments in the community.
 - * Community is visible in search results.
 - **Request only**
 - * Anyone can send a join request to the community.
 - * Requests can be accepted/rejected by the community admin/moderator.
 - * Only community members can view the community page.

- * Only community members can view the posts in the community.

- * Only community members can view the comments in the community.

- * Community is visible in search results.

– **Invite only**

- * Only community members can view the community page.

- * Only community members can view the posts in the community.

- * Only community members can view the comments in the community.

- * Community is not visible in search results.

- * Only community admin/moderator(s) can invite other users to the community.

• Allowed post types

- Text
- Image
- Video
- Link
- Poll

• Community Rules

• Community Profile Picture

- Post and comment privileges for community members.

- Community categories and other fields to help with search results and recommendations.

- 3) User clicks on the create community button.

- 4) Community is created.

- 5) Community information is stored in the database.

- 6) User is redirected to the community page.

- 7) User becomes the community admin.

Alternate Scenario:

- 2a. User enters the community name which is already taken. System displays an error message that the community name is already taken.

- 1a. User has been banned from creating communities. System displays an error message that the user has been banned from creating communities.

Use Case 20: Edit Community

Primary Actor: Community Admin

PreCondition: Stable internet connection, the user is logged in and the user is the community admin

Main Scenario:

- 1) User clicks on the edit community button.

- 2) User can edit the following community details:
 - Community description
 - Community type (public/request only/invite only)
 - Community Rules
 - Community Profile Picture
 - Community categories and other fields to help with search results and recommendations.
- 3) User clicks on the save button.
- 4) Community information is updated in the database.
- 5) User is redirected to the community page.

Use Case 21: Delete Community

Primary Actor: Community Admin

PreCondition: Stable internet connection, the user is logged in and the user is the community admin

Main Scenario:

- 1) User clicks on the delete community button.
- 2) User clicks on the confirm delete button.
- 3) Community is deleted.
- 4) Community information is marked as deleted in the database.
- 5) Community posts are marked as deleted in the database.
- 6) Community comments are marked as deleted in the database.
- 7) Community members are removed from the community.
- 8) User information is updated in the database.
- 9) User information of the community members is updated in the database.
- 10) User is redirected to the home page.

Use Case 22: Join Community

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the community is public

Main Scenario:

- 1) User clicks on the join community button.
- 2) User is added to the community.
- 3) User information is updated in the database.
- 4) Community information is updated in the database.
- 5) User is redirected to the community page.

Alternate Scenario:

- 1a. User has been banned from this community. System displays an error message that the user has been banned from this community.

Use Case 23: Leave Community

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the user is a member of the community

Main Scenario:

- 1) User clicks on the leave community button.
- 2) User is removed from the community.
- 3) User information is updated in the database.
- 4) Community information is updated in the database.
- 5) User is redirected to the home page.

Use Case 24: View Public Community

Primary Actor: General User

PreCondition: Stable internet connection

Main Scenario:

- 1) User clicks on the Community
- 2) User is redirected to the community page.
- 3) User can view the community details.
- 4) User can view the community posts.
- 5) User can view the community comments.

Use Case 25: Sending Join Request

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the community is request only

Main Scenario:

- 1) User clicks on the join community button.
- 2) User is added to the join request list.
- 3) Any community admin/moderator can accept/reject the join request.
- 4) User information is updated in the database.
- 5) Community information is updated in the database.
- 6) A notification is sent to the community admin/moderator.

Alternate Scenario:

- 1a. User has been banned from this community. System displays an error message that the user has been banned from this community.
- 1b. User has already sent a join request. System displays an error message that the user has already sent a join request.

Use Case 26: Accept/Reject Join Request

Primary Actor: Community Admin/Community Moderator

PreCondition: Stable internet connection, the user is logged in and the community is request only

Main Scenario:

- 1) User visits the community request page.
- 2) System checks for privileges.
- 3) User can view the list of join requests.
- 4) User can accept/reject the join request.
- 5) Community-members information is updated in the database.
- 6) Community information is updated in the database.
- 7) A notification is sent to the user.

Use Case 27: Invite User to Community

Primary Actor: Community Admin/Community Moderator

PreCondition: Stable internet connection, the user is logged in and the community is invite only

Main Scenario:

- 1) User clicks on the invite user button.
- 2) System checks for privileges.
- 3) User can enter the username of the user to invite.
- 4) User clicks on the invite button.
- 5) User is added to the invite list.
- 6) Community information is updated in the database.
- 7) A notification is sent to the user.

Alternate Scenario:

- 4a. User enters the username which is not registered. System displays an error message that the username is not registered.
- 4b. User enters the username which is already a member of the community. System displays an error message that the username is already a member of the community.
- 4c. User enters the username which is already invited to the community. System displays an error message that the username is already invited to the community.
- 4d. User enters the username which is already banned from the community. System displays an error message that the username is already banned from the community.

Use Case 28: View Community Members

Primary Actor: Community Admin/Moderators

PreCondition: Stable internet connection, the user is logged in and the user is a admin/moderator of the community

Main Scenario:

- 1) User clicks on the community members button.
- 2) System checks for privileges.
- 3) User is redirected to the community members page.
- 4) User can view a list of community members.
- 5) User can click on the profile of a community member to view the profile.

Use Case 29: Ban User from Community

Primary Actor: Community Admin/Moderators

PreCondition: Stable internet connection, the user is logged in and the user is a admin/moderator of the community

Main Scenario:

- 1) User selects a user from the community members list and selects ban user option.
- 2) System checks for privileges.
- 3) User is added to the ban list.
- 4) Community information is updated in the database.
- 5) User information is updated in the database.
- 6) A notification is sent to the user.

Use Case 30: Send unban-request from Community

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the user is banned from the community

Main Scenario:

- 1) User clicks on the unban-request button.
- 2) User adds a description for the unban-request.
- 3) User clicks on the send unban-request button.
- 4) Unban-request is sent to the community admin.
- 5) A notification is sent to the community admin/moderator.

Alternate Scenario

- 3a. User has already sent 3 unban requests which have been rejected. System displays an error message that the user has already sent 3 unban requests which have been rejected.

Use Case 31: Unban User from Community

Primary Actor: Community Admin

PreCondition: Stable internet connection, the user is logged in and the user is a admin of the community

Main Scenario:

- 1) User selects a user from the ban list and selects unban user option.
- 2) System checks for privileges.
- 3) User is removed from the ban list.
- 4) Community information is updated in the database.
- 5) User information is updated in the database.
- 6) A notification is sent to the user.

Alternate Scenario:

- 2a. User accepts the unban request in the community-admin notification page.

Use Case 32: Report Community

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the user is viewing a community

Main Scenario:

- 1) User clicks on the report button.
- 2) User can select the reason for reporting the community.
- 3) User can enter a description for reporting the community.
- 4) User clicks on the report button.
- 5) Report is sent to the superuser.
- 6) Report information is updated in the database.

Alternate Scenario:

- 4a. User has already reported the community 3 times all of which have been rejected.

3.1.4. Posts:

Use Case 33: Create Post

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the user is a member of the community

Main Scenario:

- 1) User clicks on the create post button.
- 2) User enters the following post details:
 - Post Title
 - Post Description
 - The post can be one of the following.
 - Text
 - Image
 - Video
 - URLs
 - Polls
 - Post Categories and other fields to help with search results and recommendations.
- 3) System checks for privileges.
- 4) System checks for allowed post types.
- 5) User can add spoiler tags to portions of the post.
- 6) User can select schedule post option and set the date and time.
- 7) User can toggle comments on/off.
- 8) User previews the post.
- 9) User clicks on the create post button.
- 10) Post is created.
- 11) Post information is stored in the database.
- 12) User is redirected to the post page.

Alternate Scenario:

- 2a. User has been banned from creating posts. System displays an error message that the user has been banned from creating posts.
- 2b. The post privileges of community is set to only selected users. System displays an error message that the user does not have the privileges to create posts.
- 3a. User selects a post type which is not allowed. System displays an error message that the post type is not allowed.

Use Case 34: Edit Post

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the user is the creator of the post

Main Scenario:

- 1) User clicks on the edit post button.
- 2) User can edit the following post details:
 - Post Title
 - Post Description
 - The post can be one of the following.
 - Text
 - Image
 - Video
 - URLs
 - Polls
 - Post Categories and other fields to help with search results and recommendations.
- 3) System checks for privileges.
- 4) System checks for allowed post types.
- 5) User can add spoiler tags to portions of the post.
- 6) User can deselect schedule post option or change the date and time if the post is already scheduled and has not been posted.
- 7) User can toggle comments on/off.
- 8) User previews the post.
- 9) User clicks on the save button.
- 10) Post information is updated in the database.
- 11) User is redirected to the post page.

Alternate Scenario:

- 3a. User selects a post type which is not allowed. System displays an error message that the post type is not allowed.

Use Case 35: Delete Post

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the user is the creator of the post

Main Scenario:

- 1) User clicks on the delete post button.
- 2) User clicks on the confirm delete button.
- 3) Post is deleted.
- 4) Post information is marked as deleted in the database.
- 5) Post comments are marked as deleted in the database.
- 6) User information is updated in the database.
- 7) User is redirected to the home page.

Use Case 36: Report Post

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the user is viewing a post

Main Scenario:

- 1) User selects the post.
- 2) User can select the reason for reporting the post.
- 3) User can enter a description for reporting the post.
- 4) User clicks on the report button.
- 5) Report is sent to the community admin.
- 6) Report information is updated in the database.

Alternate Scenario:

- 4a. User has already reported the post 3 times all of which have been rejected.

Use Case 37: View Post

Primary Actor: General User

PreCondition: Stable internet connection

Main Scenario:

- 1) User clicks on the post.
- 2) User is redirected to the post page.
- 3) If the post has spoiler tags then the user can double click on the spoiler tags to view the content.
- 4) User can view the post details.
- 5) User can view the post comments.
- 6) User can upvote the post.
- 7) User can downvote the post.
- 8) Post information is updated in the database.
- 9) User can save the post.

Use Case 38: Save Post

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and is a member of the community

Main Scenario:

- 1) User clicks on the save post button.

- 2) Post is added to the user-saved posts list.
- 3) Post information is updated in the database.

Use Case 39: View Saved Posts

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in

Main Scenario:

- 1) In the user profile, the user clicks on the saved posts button.
- 2) User is redirected to the saved posts page.
- 3) User can view a list of saved posts.
- 4) User can click on the post to view the post.

Use Case 40: Upvote Post

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and is a member of the community

Main Scenario:

- 1) User clicks on the upvote post button.
- 2) If the user has previously downvoted the post, the downvotes decrease by 1.
- 3) Post upvote count is incremented.
- 4) Post information is updated in the database.

Use Case 41: Downvote Post

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and is a member of the community

Main Scenario:

- 1) User clicks on the downvote post button.
- 2) If the user has previously upvoted the post, the upvotes decrease by 1.
- 3) Post downvote count is incremented.
- 4) Post information is updated in the database.

Use Case 42: Mention a user in post/comment

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in

Main Scenario:

- 1) User enters the @ symbol followed by the username in the post/comment.
- 2) User clicks on the send button.

- 3) User is redirected to the post/comment page.
- 4) The mentioned user is notified that the user has been mentioned in the post/comment.
- 5) User can click on the notification to view the post/comment.

3.1.5. Comments:

Use Case 43: Create new comment

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and is a member of the community and the comments are turned on for that post

Main Scenario:

- 1) User enters the comment in the comment box.
- 2) User clicks on the send button.
- 3) A new discussion thread is created.
- 4) The new comment is added to the discussion thread.
- 5) If the user has mentioned another user in the comment, the mentioned user is notified that the user has been mentioned in the comment.
- 6) If the comment is of size greater than a certain threshold then the comment is truncated and a read more button is displayed.
- 7) Comment information is stored in the database.

Alternate Scenario:

- 2a. User has been banned from creating comments. System displays an error message that the user has been banned from creating comments.
- 2b. Comments are turned off for the post. System does not display the comment box.
 - 1) If some comments were previously added when the comments were turned on, then those comments are displayed.
 - 2) Network error. The system discards the comment.

Use Case 44: Edit comment

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the user has already commented

Main Scenario:

- 1) User clicks on the edit comment button.
- 2) User can edit the comment.
- 3) User clicks on the save button.
- 4) Edited is displayed in the discussion thread next to the comment.
- 5) If the edited comment contains a new mention then a notification is sent to the mentioned user.
- 6) Comment information is updated in the database.

Alternate Scenario:

- 2a. User has been banned from creating comments. The user may have created a comment when he had the privileges but currently does not have the permission.
- 2b. Comments are turned off for the post. System does not display the edit comment button.
 - 1) Network error. The system discards the edit to the comment.

Use Case 45: Reply to a comment

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the user is a member of the community and there exists some comment for that post.

Main Scenario:

- 1) User selects some comment to reply to.
- 2) User enters the reply in the comment box.
- 3) Check if user has the privileges to comment.
- 4) User clicks on the send button.
- 5) A new discussion thread is created under the selected comment.
- 6) The new comment is added to the discussion thread.
- 7) If the user has mentioned another user in the comment, the mentioned user is notified that the user has been mentioned in the comment.
- 8) A notification is sent to the user who is being replied to.
- 9) If the comment is of size greater than a certain threshold then the comment is truncated and a read more button is displayed.
- 10) Comment information is stored in the database.

Alternate Scenario:

- 4a. User has been banned from creating comments. System displays an error message that the user has been banned from creating comments.
- 4b. Comments are turned off for the post. System does not display the comment box.
 - 1) If some comments were previously added when the comments were turned on, then those comments are displayed.
 - 2) Network error. The system discards the comment.

Use Case 46: Delete comment

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the user has already commented

Main Scenario:

- 1) User clicks on the delete comment button.

- 2) User clicks on the confirm delete button.
- 3) Comment is deleted.
- 4) If no reply exists for the comment then the discussion thread is deleted.
- 5) If the comment has replies then the comment is viewed as deleted and the replies remain visible.
- 6) Comment information is marked as deleted in the database.
- 7) User information is updated in the database.

Alternate Scenario:

- 2a. User has been banned from creating comments. The user may have created a comment when he had the privileges but currently does not have the permission.
- 2b. Comments are turned off for the post. System does not display the delete comment button.
 - 1) Network error. The system discards the delete to the comment.

Use Case 47: Report comment

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the user is viewing a comment

Main Scenario:

- 1) User selects the comment.
- 2) User can select the reason for reporting the comment.
- 3) User can enter a description for reporting the comment.
- 4) User clicks on the report button.
- 5) Report is sent to the community admin.
- 6) A notification is sent to the community admin/moderator.
- 7) A notification is sent to the user who is being reported.
- 8) Report information is updated in the database.

Alternate Scenario:

- 4a. User has already reported the comment 3 times all of which have been rejected.
 - 1) Network error. The system discards the report to the comment.

3.1.6. Notifications:

Use Case 48: View Notifications

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in

Main Scenario:

- 1) User clicks on the notification button.
- 2) User is redirected to the notification page.

- 3) User can view a list of notifications. Following are the list of notifications a user can receive.

- New post in a community the user is a member of.
- New comment in a post the user has created.
- New reply to a comment the user has created.
- New mention in a post/comment.
- New community invite from a community the user is not a member of.
- New join request accepted by a community the user is a member of.
- Referral points earned.
- New chat request.
- New direct message.
- New follower.
- Reports on posts/comments.
- Unban request accepted by the community admin/moderator.
- Unban request accepted by the superuser.
- Warnings from the superuser.
- Warnings from the community admin/moderator.
- Post ban, comment ban, community bans, platform bans from the superuser/community admin/moderator.

- 4) User can click on the notification to view the contents.

Use Case 49: Notification Settings

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in

Main Scenario:

- 1) User clicks on the notification settings button.
- 2) User can select the following options:
 - Turn on/off notifications for posts.
 - Turn on/off notifications for comments.
 - Turn on/off notifications for replies.
 - Turn on/off notifications for referral points.
 - Turn on/off notifications for chat requests.
 - Turn on/off notifications for direct messages.
 - Turn on/off notifications for followers.
- 3) User clicks on the save button.
- 4) User information is updated in the database.

Use Case 50: Community Notification

Primary Actor: Community Admin

PreCondition: Stable internet connection, the user is logged in and the user is a admin of the community

Main Scenario:

- 1) User clicks on the notification button.
- 2) User is redirected to the notification page.
- 3) User can view a list of notifications. Following are the list of notifications a user can receive.
 - Reports on posts/comments in the community.
 - Unban requests from the community members.
 - Reports on community.
- 4) User can click on the notification to view the contents.

Use Case 51: Superuser Notification

Primary Actor: Superuser

PreCondition: Stable internet connection, the user is logged in and the user is a superuser

Main Scenario:

- 1) User clicks on the notification button.
- 2) User is redirected to the notification page.
- 3) User can view a list of notifications. Following are the list of notifications specific to a superuser.
 - Reports on posts/comments.
 - Reports on community.
 - Unban requests from the platform users.
- 4) User can click on the notification to view the contents.

Use Case 52: Community Moderator Notifications

Primary Actor: Community Moderator

PreCondition: Stable internet connection, the user is logged in and the user is a moderator of the community

Main Scenario:

- 1) User clicks on the notification button.
- 2) User is redirected to the notification page.
- 3) User can view a list of notifications. Following are the list of notifications specific to a community moderator.
 - Reports on posts/comments in the community.
 - Unban requests from the community members.
 - Notifications from the community admin regarding
 - Delete post privileges.
 - Delete comment privileges.
 - Ban user privileges.
 - Unban user privileges.
 - Warning privileges.
 - Issuing post privilege to community users.

- Stopping comments on a post.
- Removing post privileges from community users.
- Removing comment privileges from community users.
- Appointment as moderator.
- Removal as moderator.

- 4) User can click on the notification to view the contents.

3.1.7. Admin privileges:

Use Case 53: Community privileges

Primary Actor: Community Admin

PreCondition: Stable internet connection, the user is logged in and the user is a admin of the community

Main Scenario:

- 1) Admin has the following community privileges:
 - Change community settings.
 - Appointing/removing moderators.
 - Granting post privileges to community members.
 - Granting select moderator privileges.

Use Case 54: Cascading privileges to moderators

Primary Actor: Community Admin

PreCondition: Stable internet connection, the user is logged in and the user is a admin of the community

Main Scenario:

- 1) Admin can grant the following privileges to moderators:
 - Delete post privileges.
 - Delete comment privileges.
 - Ban user privileges.
 - Unban user privileges.
 - Warning privileges.
 - Issuing post privilege to community users.
 - Stopping comments on a post.
 - Removing post privileges from community users.
 - Removing comment privileges from community users.
- 2) Admin can also remove the above privileges from moderators.

3.1.8. Moderator privileges:

Use Case 55: Community privileges

Primary Actor: Community Moderator

PreCondition: Stable internet connection, the user is logged in and the user is a moderator of the community

Main Scenario:

- 1) Moderator can receive the following community privileges:
 - Delete post privileges.
 - Delete comment privileges.
 - Ban user privileges.
 - Unban user privileges.
 - Warning privileges.
 - Issuing post privilege to community users.
 - Stopping comments on a post.
 - Removing post privileges from community users.
 - Removing comment privileges from community users.

Use Case 56: Delete Post (Moderator)

Primary Actor: Community Moderator

PreCondition: Stable internet connection, the user is logged in and the user is a moderator of the community

Main Scenario:

- 1) Moderator clicks on the delete post button.
- 2) Moderator clicks on the confirm delete button.
- 3) Notification is sent to the post creator that the post has been deleted.
- 4) Post is deleted.
- 5) Post information is marked as deleted in the database.
- 6) Post comments are marked as deleted in the database.
- 7) User information is updated in the database.

Use Case 57: Delete Comment (Moderator)

Primary Actor: Community Moderator

PreCondition: Stable internet connection, the user is logged in and the user is a moderator of the community

Main Scenario:

- 1) Moderator clicks on the delete comment button.
- 2) Moderator clicks on the confirm delete button.
- 3) Notification is sent to the comment creator that the comment has been deleted.
- 4) Comment is deleted.
- 5) If the comment has replies then the comment is viewed as deleted and the replies remain visible.
- 6) Comment information is marked as deleted in the database.

- 7) User information is updated in the database.

Use Case 58: Ban User (Moderator)

Primary Actor: Community Moderator

PreCondition: Stable internet connection, the user is logged in and the user is a moderator of the community

Main Scenario:

- 1) Moderator views the community members list.
- 2) Moderator selects a user from the community members list and selects ban user option.
- 3) Moderator selects the ban duration and enters the reason for banning the user.
- 4) Moderator clicks on the confirm ban button.
- 5) Notification is sent to the user that the user has been banned from the community.
- 6) User is added to the ban list.
- 7) Community information is updated in the database.
- 8) User information is updated in the database.

Use Case 59: Unban User (Moderator)

Primary Actor: Community Moderator

PreCondition: Stable internet connection, the user is logged in and the user is a moderator of the community

Main Scenario:

- 1) Moderator views the community members list.
- 2) Moderator selects a user from the ban list and selects unban user option.
- 3) Moderator clicks on the confirm unban button.
- 4) Notification is sent to the user that the user has been unbanned from the community.
- 5) User is removed from the ban list.
- 6) Community information is updated in the database.
- 7) User information is updated in the database.

Use Case 60: Warning User (Moderator)

Primary Actor: Community Moderator

PreCondition: Stable internet connection, the user is logged in and the user is a moderator of the community

Main Scenario:

- 1) Moderator selects a post/comment.
- 2) Moderator clicks on the warn button.
- 3) Moderator enters the reason for warning the user.
- 4) Moderator clicks on the confirm send button.
- 5) Notification is sent to the user that the user has been warned.
- 6) User information is updated in the database.
- 7) User is added to the warning list.
- 8) Community information is updated in the database.

Alternate Scenario

- 4a. Moderator can view the members list and select a user to warn.

Use Case 61: Remove Post Privileges (Moderator)

Primary Actor: Community Moderator

PreCondition: Stable internet connection, the user is logged in and the user is a moderator of the community and the user has post privileges

Main Scenario:

- 1) Moderator views the community members list.
- 2) Moderator selects a user from the community members list and selects remove post privileges option.
- 3) Moderator clicks on the confirm button.
- 4) Notification is sent to the user that the user has been removed from the post privileges list.
- 5) User is removed from the post privileges list.
- 6) Community information is updated in the database.
- 7) User information is updated in the database.

Use Case 62: Remove Comment Privileges (Moderator)

Primary Actor: Community Moderator

PreCondition: Stable internet connection, the user is logged in and the user is a moderator of the community and the user has comment privileges

Main Scenario:

- 1) Moderator views the community members list.
- 2) Moderator selects a user from the community members list and selects remove comment privileges option.
- 3) Moderator clicks on the confirm button.
- 4) Notification is sent to the user that the user has been removed from the comment privileges list.
- 5) Community information is updated in the database.
- 6) User information is updated in the database.

Use Case 63: Issuing post privileges

Primary Actor: Community Moderator

PreCondition: Stable internet connection, the user is logged in and the user is a moderator of the community and the user does not have post privileges

Main Scenario:

- 1) Moderator views the community members list.
- 2) Moderator selects a user from the community members list and selects issue post privileges option.
- 3) Moderator clicks on the confirm button.
- 4) Notification is sent to the user that the user has been issued post privileges.

- 5) User is added to the post privileges list.
- 6) Community information is updated in the database.
- 7) User information is updated in the database.

Use Case 64: Pause comments on a post

Primary Actor: Community Moderator

PreCondition: Stable internet connection, the user is logged in and the user is a moderator of the community and the comments are turned on for the post

Main Scenario:

- 1) Moderator views the post.
- 2) Moderator clicks on the pause comments button.
- 3) Moderator clicks on the confirm button.
- 4) Notification is sent to the post creator that the comments have been paused.
- 5) Comments are turned off for the post.
- 6) Post information is updated in the database.

Use Case 65: Resume comments on a post

Primary Actor: Community Moderator

PreCondition: Stable internet connection, the user is logged in and the user is a moderator of the community and the comments are turned off for the post

Main Scenario:

- 1) Moderator views the post.
- 2) Moderator clicks on the resume comments button.
- 3) Moderator clicks on the confirm button.
- 4) Notification is sent to the post creator that the comments have been resumed.
- 5) Comments are turned on for the post.
- 6) Post information is updated in the database.

Alternate Scenario:

- 2a. Comments are already turned off for the post. System displays an error message that the comments are already turned off for the post.

Use Case 66: Remove comment privileges from users

Primary Actor: Community Moderator

PreCondition: Stable internet connection, the user is logged in and the user is a moderator of the community and the user has comment privileges

Main Scenario:

- 1) Moderator views the community members list.
- 2) Moderator selects a user from the community members list and selects remove comment privileges option.
- 3) Moderator clicks on the confirm button.

- 4) Notification is sent to the user that the user has been removed from the comment privileges list.
- 5) Community information is updated in the database.
- 6) User information is updated in the database.

3.1.9. Search:

Use Case 67: Search for a community

Primary Actor: General User

PreCondition: Stable internet connection

Main Scenario:

- 1) User enters the 'c/' followed by the community name in the search bar.
- 2) User clicks on the search button.
- 3) User is redirected to the search results page.
- 4) User can view a list of communities.
- 5) The results are communities which are either
 - a) Public.
 - b) Request only.
 - c) Private and the user is a member of the community.
- 6) The list contains the following communities:
 - Communities with the same name as the search query.
 - Communities with names close to the search query.
- 7) Results can be sorted and filtered.
- 8) User can click on the community to view the community.

Use Case 68: Search for a user

Primary Actor: General User

PreCondition: Stable internet connection

Main Scenario:

- 1) User enters the '@/' followed by the username in the search bar.
- 2) User clicks on the search button.
- 3) User is redirected to the search results page.
- 4) User can view a list of users.
- 5) The list contains the following users:
 - Users with the same username as the search query.
 - Users with usernames close to the search query.
- 6) Results can be sorted and filtered.
- 7) User can click on the user to view the user profile.

Use Case 69: Search for a post

Primary Actor: General User

PreCondition: Stable internet connection

Main Scenario:

- 1) User enters the post title in the search bar.
- 2) User clicks on the search button.
- 3) User is redirected to the search results page.
- 4) User can view a list of posts.
- 5) The posts belong to the following communities.
 - Public.
 - Communities which the user is a member of.
- 6) The list contains the following posts:
 - Posts with the same title as the search query.
 - Posts with titles close to the search query.
- 7) Results can be sorted and filtered.
- 8) User can click on the post to view the post.

Use Case 70: Search for a combination of above search types

Primary Actor: General User

PreCondition: Stable internet connection

Main Scenario:

- 1) User enters the search query in the search bar.
- 2) A general search query is of the form 'c/community_name @/username post_title'.
- 3) Any of the above search types can be used/omitted in the search query.
 - a) 'c/community_name @/username post_title' searches for posts with the title 'post_title' in the community 'community_name' by the user 'username'.
 - b) 'c/community_name @/username' searches for posts in the community 'community_name' by the user 'username'.
 - c) 'c/community_name post_title' searches for posts with the title 'post_title' in the community 'community_name'.
 - d) '@/username post_title' searches for posts with the title 'post_title' by the user 'username' in public/mutual communities.
- 4) Results are fuzzy matched.
- 5) Results can be sorted and filtered.
- 6) User can click on the search result to view the search result.

3.1.10. Recommendations/User Feed:

Use Case 71: View User Feed

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in

Main Scenario:

- 1) User clicks on the user feed button.

- 2) User is redirected to the user feed page which is specific to the user.
- 3) User can view a list of posts.
- 4) User can also view a list of 'recommended' communities (request-only/public).
- 5) User can also view a list of 'recommended' users.
- 6) The posts belong to the following communities.
 - Public.
 - Communities which the user is a member of.
- 7) The posts/communities/users are suggested by some recommender system
- 8) Results can be sorted and filtered.
- 9) User can click on the post to view the post.

3.1.11. Trending Page:

Use Case 72: View Trending Page

Primary Actor: General User

PreCondition: Stable internet connection

Main Scenario:

- 1) User clicks on the trending button.
- 2) User is redirected to the trending page which is common for all users.
- 3) User can view a list of posts.
- 4) The posts belong to public communities.
- 5) Posts are based on different parameters.
- 6) Results can be sorted and filtered.
- 7) User can click on the post to view the post.

3.1.12. Direct-messages:

Use Case 73: Send Chat Request

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in

Main Scenario:

- 1) User views the user profile.
- 2) Send Chat request is visible only if the user is not already a mutual follower/ followed by the other user or the profile is private.
- 3) User clicks on the send chat request button.
- 4) User adds the reason for sending the chat request.
- 5) Notification is sent to the user that the user has received a chat request.
- 6) User information is updated in the database.

Use Case 74: Accept Chat Request

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the user has received a chat request

Main Scenario:

- 1) User clicks on the notification button.
- 2) The use can view the profile of the user who has sent the chat request.
- 3) User clicks on the accept chat request button.

Alternate Scenario:

- 3a. User clicks on the reject chat request button. Notification is sent to the user that the user has rejected the chat request.
- 3b. User clicks on the block user button. Notification is sent to the user that the user has blocked the user.

Use Case 75: Send Direct Message

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the user has accepted the chat request and the user is in the chat interface

Main Scenario:

- 1) User enters the message in the message box.
- 2) User clicks on the send button.
- 3) Message is sent to the user.
- 4) Message information is stored in the database.

Alternate Scenario

- 2a. Network error. The system discards the message.
- 2b. User has blocked the user. The system discards the message.
- 2c. Message is empty. The system displays an error message that the message is empty.

Use Case 76: Reply to a message

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the user has accepted the chat request and the user is in the chat interface

Main Scenario:

- 1) User selects the message to reply to.
- 2) User enters the reply in the message box.
- 3) User clicks on the send button.
- 4) Message is sent to the user.
- 5) Message information is stored in the database.

Alternate Scenario

- 3a. Network error. The system discards the message.
- 3b. User has blocked the user. The system discards the message.

3c. Message is empty. The system displays an error message that the message is empty.

Use Case 77: Delete a message

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the user has accepted the chat request and the user is in the chat interface

Main Scenario:

- 1) User selects the message to delete.
- 2) User clicks on the delete message button.
- 3) The message can be deleted in two ways, either only for the user or for both the users.
- 4) Message is deleted.
- 5) Message information is marked as deleted in the database.

Alternate Scenario

2a. Network error. The system discards the delete to the message.

Use Case 78: Report a message

Primary Actor: Registered User

PreCondition: Stable internet connection, the user is logged in and the user has accepted the chat request and the user is in the chat interface

Main Scenario:

- 1) User selects the message to report.
- 2) User clicks on the report message button.
- 3) User can select the reason for reporting the message.
- 4) User can enter a description for reporting the message.
- 5) User clicks on the report button.
- 6) Report is sent to the superuser.
- 7) A notification is sent to the superuser.
- 8) A notification is sent to the user who is being reported.
- 9) Report information is updated in the database.

Alternate Scenario:

5a. User has already reported the message 3 times all of which have been rejected.

- 1) Network error. The system discards the report to the message.

3.1.13. Superuser privileges:

Use Case 79: Make a user a superuser

Primary Actor: Superuser

PreCondition: Stable internet connection, the user is logged in and the user is a superuser

Main Scenario:

- 1) Superuser views the user profile.
- 2) Superuser clicks on the make superuser button.
- 3) Superuser clicks on the confirm make superuser button.
- 4) Notification is sent to the user that the user has been made a superuser.
- 5) User is made a superuser.
- 6) User information is updated in the database.

Use Case 80: Ban User (Superuser)

Primary Actor: Superuser

PreCondition: Stable internet connection, the user is logged in and the user is a superuser

Main Scenario:

- 1) Superuser views the user profile.
- 2) Superuser clicks on the ban user button.
- 3) Superuser selects the ban duration and enters the reason for banning the user.
- 4) Superuser clicks on the confirm ban button.
- 5) Notification is sent to the user that the user has been banned from the platform.
- 6) User is added to the ban list.
- 7) User information is updated in the database.

Use Case 81: Unban User (Superuser)

Primary Actor: Superuser

PreCondition: Stable internet connection, the user is logged in and the user is a superuser

Main Scenario:

- 1) Superuser views the ban list.
- 2) Superuser selects a user from the ban list and selects unban user option.
- 3) Superuser clicks on the confirm unban button.
- 4) Notification is sent to the user that the user has been unbanned from the platform.
- 5) User is removed from the ban list.
- 6) User information is updated in the database.

Use Case 82: Warning User (Superuser)

Primary Actor: Superuser

PreCondition: Stable internet connection, the user is logged in and the user is a superuser

Main Scenario:

- 1) Superuser views the user profile.
- 2) Superuser clicks on the warn button.
- 3) Superuser enters the reason for warning the user.
- 4) Superuser clicks on the confirm send button.

- 5) Notification is sent to the user that the user has been warned.
- 6) User information is updated in the database.
- 7) User is added to the warning list.

Use Case 83: Delete Community (Superuser)

Primary Actor: Superuser

PreCondition: Stable internet connection, the user is logged in and the user is a superuser

Main Scenario:

- 1) Superuser views the community.
- 2) Superuser clicks on the delete community button.
- 3) Superuser clicks on the confirm delete button.
- 4) Notification is sent to the community admin that the community has been deleted.
- 5) Community is deleted.
- 6) Community information is marked as deleted in the database.
- 7) Community posts are marked as deleted in the database.
- 8) Community comments are marked as deleted in the database.
- 9) User information is updated in the database.

3.2. Performance Requirements

- The system should be able to handle concurrent access.
- The user feed and search results should be displayed within 2 seconds.
- The system should have a user friendly interface.

3.3. Design Constraints

- Security
 - The system should be secure against SQL injection,XSS,CSRF attacks,etc.
 - The system should keep the user data in an encrypted format.
 - The system should exchange data over a secure connection.
 - The system should have a secure login system.
 - The system should have a secure password reset system.
 - The system should have a secure session management system.
 - The system should have a secure file upload system.
- Fault Tolerance
 - The system should be able to handle network errors.
 - The system should be able to handle server errors.
 - The system should be able to handle database errors.
 - The system should be able to handle file upload errors.

- Maintainability
 - The system should be easy to maintain.
 - The system should be easy to update.
 - The system should be easy to scale.

3.4. External Interface Requirements

3.4.1. Login Screen

- The login screen has two buttons: **login via username** and **login via Google**
- The default login method is **login via username**
- The user can switch to **login via Google** by clicking on the **login via Google** button.
- Login via username has the following fields:
 - Username
 - Password
- Login via username also has a **forgot password** button and the **Create an account** link which takes the user to the signup screen.
- Login via Google launches the Google login page.
- The user can login via Google by entering the Google credentials.

3.4.2. Signup Screen

- The signup screen has the following fields:
 - Username
 - Email
 - Password
 - Confirm Password
- The signup screen also has a **login** link which takes the user to the login screen.

3.4.3. Home Screen-Registered User

- The home screen has the following panes:
- A top-navbar with the following fields:
 - Home
 - Communities-joined/admin/moderator
 - Chat
 - Notifications
 - Profile
 - Logout
 - Search bar
- A left-sidebar with the following fields:
 - Home
 - Joined community list
 - Create community
 - Trending

- The central pane has the following fields:
 - User feed
 - Community recommendations
 - User recommendations

3.4.4. Home Screen-Guest User

- The home screen has the following panes:
- A top-navbar with the following fields:
 - Home
 - Login
 - Signup
 - Search bar
- A left-sidebar with the following fields:
 - Home
 - Trending
- The central pane has the following fields:
 - User feed

3.4.5. Notification Screen

The notification screen has the following panes:

- A top-navbar
- A left-sidebar
- The central pane lists notifications related to the user/admin/moderator/superuser:

3.4.6. Community Screen

The community screen has the following panes:

- A top-navbar
- A left-sidebar
- The central pane lists the posts in the community
- The right-sidebar lists the community information

3.4.7. Community Creation Screen

Along with the usual navbar and left-sidebar, the community creation screen has the following fields:

- Community name
- Community description
- Community rules
- Community tags
- Community profile picture
- Community cover picture
- Community settings including
 - Community type
 - Allowed post types
 - Post and comment privileges

3.4.8. Post Creation Screen

Along with the usual navbar and left-sidebar, the post creation screen has the following fields:

- Post title
- Post description
- Post tags
- Post content
- Spoiler tags
- Scheduling information
- Comments on/off

3.4.9. Chat Screen

The chat screen has the following panes:

- A top-navbar
- A left-sidebar
- The central pane lists the chats from different users in minimized form.
- On clicking the chat, the central pane displays the chat interface.
- The right-sidebar lists the chat requests.
- The right-sidebar also lists the blocked users.
- On expanding a chat, the right-sidebar lists the chat options.
- The chat interface has the following fields:
 - Message box
 - Send button
 - Chat options for censoring.
- The chat layout is similar to a messaging app with the user's messages on the right and the other user's messages on the left each of which are color coded.
- The chat interface has options to edit/delete/reply-to/report messages.

3.4.10. Profile Screen-Own

The profile screen has the following panes:

- A top-navbar
- A left-sidebar
- The central pane has two buttons, posts and comments. Pressing posts lists the user's posts and pressing comments lists all comments and replies.
- The right-sidebar lists the user information.
- Along with user information other fields includes all the reports that were accepted by the superuser or any community admin.

3.4.11. Profile Screen-Other

The profile screen has the following panes:

- A top-navbar
- A left-sidebar
- The central pane has two buttons, posts and comments. Pressing posts lists the user's posts and pressing comments lists all comments and replies.
- Contents of central pane are subject to the user's privacy settings.
- The right-sidebar lists the user information, followers and following.
- Along with user information other fields includes all the reports that were accepted by the superuser or any community admin.
- The right-sidebar also lists the chat request button.

4. Further Extensions

Some of the possible future extensions are:

- **User profile badges:** Users can earn badges based on their activity on the platform.
- **User profile achievements:** Users can earn achievements based on their activity on the platform.
- **User profile levels:** Users can level up based on their activity on the platform.
- **User profile reputation:** Users can earn reputation based on their activity on the platform.
- **Video/Voice chats:** Users can video/voice chat with other users.
- **Custom layouts for communities:** Community admins can customize the layout of the community.
- **Integration with Machine Learning:** The system can use machine learning to recommend communities/posts/users to the user, as well as moderation.
- **Events:** Users can create events and invite other users to the event.