

Communities A Social Media Platform(SM02)

Test Plan

Harshit Pant CS21BTECH11021

Satpute Aniket Tukaram CS21BTECH11056

Mahin Bansal CS21BTECH11034

Burra Vishal Mathews CS21BTECH11010

- Test Overview
- Test Details
 - Unit Tests
 - * Authentication Module
 - * User Module
 - Guest User Module
 - Registered User Module
 - Admin User Module
 - Moderator User Module
 - Superuser Module
 - * System Module
 - Cache Module
 - Recommendation Module
 - * Database Access Module
 - User Record Module
 - Post Record Module
 - Comment Record Module
 - Vote Record Module
 - Chat(Private Messages) Record Module
 - Message(Group and Private) Record Module
 - Group Record Module
 - User_chat Record Module
 - User_group Record Module
 - Community Record Module
 - Joined Community Record Module
 - Blocked User Record Module
 - Reported User Record Module
 - Roles Record Module
 - * UI Module
 - Integration Testing
 - System Testing
 - Performance Testing
 - * Latency
 - * Scalability
 - * Load Handling/Load Testing
 - * Cache Performance
 - * Concurrent Access
- Test Analysis
 - Functional Test Report

Test Overview

Communities is an online social media platform that allows users to create and join communities based on their interests. Users can create communities, post content, comment on posts, and interact with other users. The platform also includes other features such as chat, search, and recommendations and certain level of moderation.

This document outlines the test plan for Communities, including the modules to be tested, the types of tests to be conducted, and the test analysis.

The test plan includes the following operations/modules that will be tested:

- Authentication Modules
 - Login Module
 - Signup Module
 - Token Authentication Module
- User Modules
 - Guest User Module
 - Registered User Module
 - Admin User Module
 - Moderator User Module
 - Superuser Module
- System Modules
 - Cache Manager Module
 - Recommender System Module
 - Job Scheduler Module
 - Services Module
 - * Feed Service
 - * Comments Service
 - * Post Service
 - * Search Service
 - * Votes Service
 - * Chat Service
 - * Connections Service
 - * Scoring Service
- DB Access Modules
 - User Module
 - Post Module
 - Comment Module
 - Vote Module
 - Chat Module
 - Community Module
- UI Modules
 - Listing Module
- Other Modules
 - Notification Module
 - Reporting Module
 - Moderation Module

The test plan includes the following sections:

- Test Overview: A brief overview of the test plan which also lists the operations/modules that will be tested.
- Test Details: Detailed information about the unit tests, integration testing, system testing, and performance testing.
- Test Analysis: A summary of the test statistics, functional test report, and performance test report.
 - Test Statistics: The number of classes, methods, modules tested, test cases, and test cases failed.
 - Functional Test Report: A summary of the functional test cases for each module, including the type of testing method used.
 - Performance Test Report: A summary of the performance test plan and report.

Test Details

Unit Tests

Authentication Module

Table 1: Authentication Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Inputs	Expected Output	Status
UT-1.1.a	Login Module	Login user using Interanl sign-in	Incorrect Credentials	user: unknown-username or wrong-password, DB-status: user-not-found or wrong-password (from User Module)	return: invalid-credentials	F
UT-1.1.b	Login Module	Login User using Google OAuth	Incorrect Credentials (OAuth)	user: incorrect-email or wrong-password, OAuth-status: user-not-registered or wrong-password (from OAuth)	return: invalid-credentials	F
UT-1.1.c	Login Module	Login User	Correct Credentials	user: registered-username and correct-password, DB-status: OK (from User Module or OAuth)	return: OK and JSON Web-token	P
UT-1.2.a	Token Module	Validate JWT	Invalid Token	cookie: invalid-token	The module extracts the header and the payload from the request and recalculates the signature and matches it with the token, in this case it does not match and hence returns returns: invalid-token	F
UT-1.3.a	Signup Module	Signup User	Username already taken	user: already-taken-username and password and email-id, DB-status: username-taken	return: username-taken	F
UT-1.3.b	Signup Module	Signup User	Unverified email id	user: username and password and unverified-email-id, email-service-status: email-verification-timeout	return: email-unverified	F
UT-1.3.c	Signup Module	Signup User	Passwords Don't Match	user: username and invalid-password and email-id	return: password-mismatch	F
UT-1.3.d	Signup Module	Signup User	Valid Credentials	user: username and password and email-id, DB-status: user-created	return: user-created and JWT and redirects user to home page	P

User Module

Guest User Module

Table 2: Guest User Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.a	Guest User Module	get Trending	Invalid Request	no-input	returns: list of post-records, community-records, user-records	P
UT-2.2.b	Guest User Module	view Community	Invalid CommunityID	community: unknown-community-id and user-token: no-token, get Community: community-not-found	return: community-not-found	F
UT-2.2.c	Guest User Module	view Community	Community not public	community: non-public-community-id and user-token: no-token, get Community: OK	return: community-not-public	F
UT-2.2.d	Guest User Module	view Community	Valid Community	community: known-community-id and user-token: no-token, get Community: OK	return: OK and community-record	P
UT-2.2.e	Guest User Module	view Profile	Invalid UserID	user: unknown-user-id and user-token: no-token, get User: user-not-found	return: user-not-found	F
UT-2.2.f	Guest User Module	view Profile	User Profile is Not Public	user: non-public-user-id and user-token: no-token, get User: OK	return: user-profile-not-public	F
UT-2.2.g	Guest User Module	view Profile	Valid User	user: known-user-id and user-token: no-token, get User: OK	return: OK and user-record	P
UT-2.2.h	Guest User Module	view Post	Invalid PostID	post: unknown-post-id and user-token: no-token, get Post: post-not-found	return: post-not-found	F

Continued on next page

Table 2: Guest User Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.i	Guest User Module	view Post	Post is Not From Public Community	post: non-public-post-id and user-token: no-token, get Post: OK	return: post-not-public	F
UT-2.2.j	Guest User Module	view Post	Valid Post	post: known-post-id and user-token: no-token, get Post: OK	return: OK and post-record	P
UT-2.2.k	Guest User Module	view Comment	Invalid CommentID	comment: unknown-comment-id and user-token: no-token, get Comment: comment-not-found	return: comment-not-found	F
UT-2.2.l	Guest User Module	view Comment	Comment is Not From Public Post	comment: non-public-comment-id and user-token: no-token, get Comment: OK	return: comment-not-public	F
UT-2.2.m	Guest User Module	view Comment	Valid Comment	comment: known-comment-id and user-token: no-token, get Comment: OK	return: OK and comment-record	P
UT-2.2.n	Guest User Module	search	Invalid Search Query	search-query: invalid-search-query and user-token: no-token	return: search-string-invalid	F
UT-2.2.o	Guest User Module	search	Valid Search Query	search-query: valid-search-query and user-token: no-token, search-service: search-results	return: OK and search-results	P

Continued on next page

Table 2: Guest User Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.p	Guest User Module	signup	Invalid Credentials	user: invalid-username or invalid-password or invalid-email-id, authentication-service: invalid-credentials	return: invalid-credentials	F
UT-2.2.q	Guest User Module	signup	Valid Credentials	user: valid-username and valid-password and valid-email-id, authentication-service: user-created	return: user-created and redirects to home page	P

Registered User Module

Table 3: Registered User Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.a	User Module	get Home Feed	Invalid user-token	user: invalid-token, authentication-service: invalid-token	return: invalid-token	F
UT-2.2.b	User Module	get Home Feed	Valid user-token	user: valid-token, authentication-service: valid-token	return: OK and feed	P
UT-2.2.c	User Module	view Community	Invalid CommunityID	community: unknown-community-id and user-token: valid-token, get Community: community-not-found	return: community-not-found	F

Continued on next page

Table 3: Registered User Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.d	User Module	view Community	Community not public and user is not a member	community: non-public-community-id and user-token: valid-token, get Community: OK	return: community-not-public	F
UT-2.2.e	User Module	view Community	Valid Community	community: known-community-id and user-token: valid-token, get Community: OK	return: OK and community	P
UT-2.2.f	User Module	view Profile	Invalid UserID	user: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.2.g	User Module	view Profile	User Profile(user2) is Not Public and user2 is not following user1	user: non-public-user-id and user-token: valid-token, get User: OK	return: user-profile-not-public and user2-reports	F
UT-2.2.h	User Module	view Profile	Valid User	user: known-user-id and user-token: valid-token, get User: OK	return: OK and user-record	P
UT-2.2.i	User Module	view own followers	Invalid user-token	user-token: invalid-token	return: invalid-token	F
UT-2.2.j	User Module	view own followers	Valid user-token	user-token: valid-token , get Following: OK	return: OK and followers	P
UT-2.2.k	User Module	view own following	Invalid user-token	user-token: invalid-token	return: invalid-token	F
UT-2.2.l	User Module	view own following	Valid user-token	user-token: valid-token , get Following: OK	return: OK and following	P
UT-2.2.m	User Module	view Post	Invalid PostID	post: unknown-post-id and user-token: valid-token, get Post: post-not-found	return: post-not-found	F

Continued on next page

Table 3: Registered User Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.n	User Module	view Post	Post is Not From Public Community and community not joined by user	post: non-public-post-id and user-token: valid-token, get Post: OK	return: post-not-public	F
UT-2.2.o	User Module	view Post	Valid Post	post: known-post-id and user-token: valid-token, get Post: OK	return: OK and post-record	P
UT-2.2.p	User Module	view Comment	Invalid CommentID	comment: unknown-comment-id and user-token: valid-token, get Comment: comment-not-found	return: comment-not-found	F
UT-2.2.q	User Module	view Comment	Comment is Not From Public Post and post not joined by user	comment: non-public-comment-id and user-token: valid-token, get Comment: OK	return: comment-not-public	F
UT-2.2.r	User Module	view Comment	Valid Comment	comment: known-comment-id and user-token: valid-token, get Comment: OK	return: OK and comment-record	P
UT-2.2.s	User Module	make comment	Invalid PostID	post: unknown-post-id and user-token: valid-token, parnet-comment-id: comment-id, get Post: post-not-found	return: post-not-found	F
UT-2.2.t	User Module	make comment	Post is Not From Public Community and community not joined by user	post: non-public-post-id and user-token: valid-token, parnet-comment-id: comment-id, get Post: OK	return: post-not-public	F

Continued on next page

Table 3: Registered User Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.u	User Module	make comment	Invalid Parent CommentID	post: known-post-id and user-token: valid-token, parnet-comment-id: unknown-comment-id, get Post: OK	return: parent-comment-not-found	F
UT-2.2.v	User Module	make comment	Valid Comment	post: known-post-id and user-token: valid-token, parnet-comment-id: known-comment-id, get Post: OK	return: OK	P
UT-2.2.w	User Module	make post	Invalid CommunityID	community: unknown-community-id and user-token: valid-token, get Community: community-not-found	return: community-not-found	F
UT-2.2.x	User Module	make post	Community not public and user is not a member	community: non-public-community-id and user-token: valid-token, get Community: OK	return: community-not-public	F
UT-2.2.y	User Module	make post	Post Privilege not present	community: known-community-id and user-token: valid-token, validate Privileges: False OK	return: no-post-privilege	F
UT-2.2.y	User Module	make post	Valid Post	community: known-community-id and user-token: valid-token, get Community: OK	return: OK	P
UT-2.2.z	User Module	make post	post type not allowed	community: known-community-id and user-token: valid-token, get Community: OK	return: post-type-not-allowed	F

Continued on next page

Table 3: Registered User Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.aa	User Module	make community	Invalid Community Name	community: invalid-community-name and user-token: valid-token	return: invalid-community	F
UT-2.2.ab	User Module	make community	Valid Community Name	community: valid-community-name and user-token: valid-token , get Community: community-not-found	return: OK	P
UT-2.2.ac	User Module	join community	Invalid CommunityID	community: unknown-community-id and user-token: valid-token, get Community: community-not-found	return: community-not-found	F
UT-2.2.ad	User Module	join community	Community not public	community: non-public-community-id and user-token: valid-token, get Community: OK	return: community-not-public	F
UT-2.2.ae	User Module	request to join community	Valid Community	community: known-community-id and user-token: valid-token, get Community: OK	return: OK	P
UT-2.2.af	User Module	request to join community	Community not public	community: non-public-community-id and user-token: valid-token, get Community: OK	return: community-not-public	F
UT-2.2.ag	User Module	request to join community	User already a member	community: known-community-id and user-token: valid-token, get Community: OK	return: user-already-member	F

Continued on next page

Table 3: Registered User Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.ah	User Module	request to join community	User already requested to join	community: known-community-id and user-token: valid-token, get Community: OK	return: user-already-requested	F
UT-2.2.ai	User Module	accept invite	Invalid CommunityID	community: unknown-community-id and user-token: valid-token, get Community: community-not-found	return: community-not-found	F
UT-2.2.ai	User Module	accept invite	Invalid CommunityID	community: unknown-community-id and user-token: valid-token, get Community: community-not-found	return: community-not-found	F
UT-2.2.aj	User Module	accept invite	Community not invite-only	community: non-request-community-id and user-token: valid-token, get Community: OK	return: community-not-invite-only	F
UT-2.2.ak	User Module	accept invite	No Invite Found To UserID	request: unknown-request-id and user-token: valid-token, Joined.get User-community Status: not-invited	return: invite-not-found	F
UT-2.2.al	User Module	accept invite	Valid Request	request: known-request-id and user-token: valid-token, Joined.get User-community Status: invited	return: OK	P

Continued on next page

Table 3: Registered User Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.am	User Module	reject invite	Invalid CommunityID	community: unknown-community-id and user-token: valid-token, get Community: community-not-found	return: community-not-found	F
UT-2.2.an	User Module	reject invite	Community not invite-only	community: non-request-community-id and user-token: valid-token, get Community: OK	return: community-not-invite-only	F
UT-2.2.ao	User Module	reject invite	No Invite Found To UserID	request: unknown-request-id and user-token: valid-token, Joined.get User-community Status: not-invited	return: invite-not-found	F
UT-2.2.ap	User Module	reject invite	Valid Request	request: known-request-id and user-token: valid-token, Joined.get User-community Status: invited	return: OK	P
UT-2.2.aq	User Module	block user	Invalid UserID	user: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.2.ar	User Module	block user	Valid User	user: known-user-id and user-token: valid-token, get User: OK , blocked.get Blocked user-id by valid-token: OK	return: user-already-blocked	F

Continued on next page

Table 3: Registered User Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.as	User Module	block user	Valid User	user: known-user-id and user-token: valid-token, get User: OK , blocked.get Blocked user-id by valid-token: user-not-blocked	return: OK	P
UT-2.2.at	User Module	unblock user	Invalid UserID	user: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.2.au	User Module	unblock user	Valid User	user: known-user-id and user-token: valid-token, get User: OK , blocked.get Blocked user-id by valid-token: user-not-blocked	return: user-not-blocked	F
UT-2.2.av	User Module	unblock user	Valid User	user: known-user-id and user-token: valid-token, get User: OK , blocked.get Blocked user-id by valid-token: OK	return: OK	P
UT-2.2.aw	User Module	follow user	Invalid UserID	user: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.2.ax	User Module	follow user	Valid User	user: known-user-id and user-token: valid-token, get User: OK , followed.get Followed user-id by valid-token: OK	return: user-already-followed	F

Continued on next page

Table 3: Registered User Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.ay	User Module	follow user	Valid User	user: known-user-id and user-token: valid-token, get User: OK , followed.get Followed user-id by valid-token: user-not-followed	return: OK	P
UT-2.2.az	User Module	unfollow user	Invalid UserID	user: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.2.ba	User Module	unfollow user	Valid User	user: known-user-id and user-token: valid-token, get User: OK , followed.get Followed user-id by valid-token: user-not-followed	return: user-not-followed	F
UT-2.2.bb	User Module	unfollow user	Valid User	user: known-user-id and user-token: valid-token, get User: OK , followed.get Followed user-id by valid-token: OK	return: OK	P
UT-2.2.bc	User Module	create community	Invalid Community Name	community: invalid-community-name and user-token: valid-token	return: invalid-community	F
UT-2.2.bd	User Module	create community	Valid Community Name	community: valid-community-name and user-token: valid-token , get Community: community-not-found	return: OK	P

Continued on next page

Table 3: Registered User Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.be	User Module	report user By Post	Invalid UserID	user: unknown-user-id and user-token: valid-token post: post-id, get User: user-not-found	return: user-not-found	F
UT-2.2.bf	User Module	report user By Post	Invalid PostID	post: unknown-post-id and user-token: valid-token, get Post: post-not-found	return: post-not-found	F
UT-2.2.bg	User Module	report user By Post	reporting user not in community	user: known-user-id and user-token: valid-token, post: known-post-id, get User: OK, get Post: OK, Joined.get User-community Status: user-not-in-community	return: user-not-in-community	F
UT-2.2.bh	User Module	report user By Post	Valid User	user: known-user-id and user-token: valid-token, post: known-post-id, get User: OK, get Post: OK, Joined.get User-community Status: joined	return: OK	P
UT-2.2.bi	User Module	report user By Comment	Invalid UserID	user: unknown-user-id and user-token: valid-token comment: comment-id, get User: user-not-found	return: user-not-found	F

Continued on next page

Table 3: Registered User Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.bj	User Module	report user By Comment	Invalid CommentID	comment: unknown-comment-id and user-token: valid-token, get Comment: comment-not-found	return: comment-not-found	F
UT-2.2.bk	User Module	report user By Comment	reporting user not in community	user: known-user-id and user-token: valid-token, comment: known-comment-id, get User: OK, get Comment: OK, Joined.get User-community Status: user-not-in-community	return: user-not-in-community	F
UT-2.2.bl	User Module	report user By Comment	Valid User	user: known-user-id and user-token: valid-token, comment: known-comment-id, get User: OK, get Comment: OK, Joined.get User-community Status: joined	return: OK	P
ut-2.2.bm	User Module	report community	Invalid CommunityID	community: unknown-community-id and user-token: valid-token, get Community: community-not-found	return: community-not-found	F
UT-2.2.bn	User Module	report community	Valid Community	community: known-community-id and user-token: valid-token, get Community: OK	return: OK	P

Continued on next page

Table 3: Registered User Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.bo	User Module	report to admin by community	Invalid CommunityID	community: unknown-community-id and user-token: valid-token, get Community: community-not-found	return: community-not-found	F
UT-2.2.bp	User Module	report to admin by community	Valid Community	community: known-community-id and user-token: valid-token, get Community: OK	return: OK	P
UT-2.2.bq	User Module	send chat request	Invalid UserID	user: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.2.br	User Module	send chat request	Receiver has direct chat from all users on	user: known-user-id and user-token: valid-token, get User(receiver): OK and chat-setting: all-direct-chat-from-users-on, get User(sender): OK, blocked.get User(1) By User(2): user(1)-not-blocked-by-user(2)	return: user-can-receive-all-messages	P
UT-2.2.bs	User Module	send chat request	Receiver has direct chat from all users on	user: known-user-id and user-token: valid-token, get User(receiver): OK and chat-setting: all-direct-chat-from-users-on, get User(sender): OK, blocked.get User(1) By User(2): user(1)-blocked-by-user(2)	return: user-blocked	F

Continued on next page

Table 3: Registered User Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.bt	User Module	send chat request	Receiver has no-chat-requests-on	user: known-user-id and user-token: valid-token, get User(receiver): OK and chat-setting: no-chat-requests-on, get User(sender): OK, blocked.get User(1) By User(2): user(1)-not-blocked-by-user(2)	return: user-cannot-receive-chat-requests	F
UT-2.2.bu	User Module	send chat request	Receiver has no-chat-requests-on	user: known-user-id and user-token: valid-token, get User(receiver): OK and chat-setting: no-chat-requests-on, get User(sender): OK, blocked.get User(1) By User(2): user(1)-blocked-by-user(2)	return: user-blocked	F
UT-2.2.bv	User Module	send chat request	Other chat-settings	user: known-user-id and user-token: valid-token, get User(receiver): OK and chat-setting: chat-requests-from-following, get User(sender): OK, blocked.get User(1) By User(2): user(1)-not-blocked-by-user(2)	return: user-can-receive-chat-requests	P

Continued on next page

Table 3: Registered User Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.bw	User Module	send chat request	Other chat-settings	user: known-user-id and user-token: valid-token, get User(receiver): OK and chat-setting: chat-requests-from-following, get User(sender): OK, blocked.get User(1) By User(2): user(1)-blocked-by-user(2)	return: user-blocked	F
UT-2.2.bx	User Module	send chat	Invalid UserID	user: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.2.by	User Module	send chat	Unaccepted chat request	user: known-user-id and user-token: valid-token, get User(receiver): OK and chat-setting: chat-requests-from-following, get User(sender): OK, blocked.get User(1) By User(2): user(1)-not-blocked-by-user(2)	return: chat-request-not-accepted	F
UT-2.2.bz	User Module	send chat	User is blocked by receiver	user: known-user-id and user-token: valid-token, get User(receiver): OK and chat-setting: chat-requests-from-following, get User(sender): OK, blocked.get User(1) By User(2): user(1)-blocked-by-user(2)	return: user-blocked	F

Continued on next page

Table 3: Registered User Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.2.ca	User Module	send chat	Valid User	user: known-user-id and user-token: valid-token, get User(receiver): OK and chat-setting: chat-requests- from-following, get User(sender): OK, blocked.get User(1) By User(2): user(1)-not- blocked-by-user(2)	return: OK	P

Admin User Module

Table 4: Admin Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.3.a	Admin Module	Change Community Settings	Invalid CommunityID	communityID: unknown-community-id and user-token: valid-token, get Community: community-not-found	return: community-not-found	F
UT-2.3.b	Admin Module	Change Community Settings	Valid Community	communityID: known-community-id and user-token: valid-token, get Community: OK	return: OK	P
UT-2.3.c	Admin Module	Appointing Moderator	Invalid UserID	userID: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.3.d	Admin Module	Appointing Moderator	User not in Community	userID: non-community-user-id and user-token: valid-token, get User: OK	return: user-not-in-community	F

Continued on next page

Table 4: Admin Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.3.e	Admin Module	Appointing Moderator	Valid User	userID: known-user-id and user-token: valid-token, get User: OK	return: OK	P
UT-2.3.f	Admin Module	Remove Moderator	Invalid UserID	userID: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.3.g	Admin Module	Remove Moderator	User not Moderator	userID: non-moderator-user-id and user-token: valid-token, get User: OK	return: user-not-moderator	F
UT-2.3.h	Admin Module	Remove Moderator	Valid User	userID: known-user-id and user-token: valid-token, get User: OK	return: OK	P
UT-2.3.i	Admin Module	Granting Post Privilege	Invalid UserID	userID: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.3.j	Admin Module	Granting Post Privilege	User not in Community	userID: non-community-user-id and user-token: valid-token, get User: OK	return: user-not-in-community	F
UT-2.3.k	Admin Module	Granting Post Privilege	Valid User	userID: known-user-id and user-token: valid-token, get User: OK	return: OK	P
UT-2.3.l	Admin Module	Revoking Post Privilege	Invalid UserID	userID: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F

Continued on next page

Table 4: Admin Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.3.m	Admin Module	Revoking Post Privilege	User do not have privilege	userID: non-privilege-user-id and user-token: valid-token, get User: OK	return: user-not-have-privilege	F
UT-2.3.n	Admin Module	Revoking Post Privilege	Valid User	userID: known-user-id and user-token: valid-token, get User: OK	return: OK	P

Moderator User Module

Table 5: Moderator Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.4.a	Moderator Module	Remove Post	Invalid PostID	post: unknown-post-id and user-token: valid-token, get Post: post-not-found	return: post-not-found	F
UT-2.4.b	Moderator Module	Remove Post	Post is Not From Community	post: non-community-post-id and user-token: valid-token, get Post: OK	return: post-not-from-community	F
UT-2.4.c	Moderator Module	Remove Post	Moderator does not have privilege	post: known-post-id and user-token: valid-token, get Post: OK, role.getUserCommunity.no-post-removal-privilege	return: user-cannot-remove-post	F
UT-2.4.d	Moderator Module	Remove Post	Valid Post	post: known-post-id and user-token: valid-token, get Post: OK, role.getUserCommunity.privileges: post-removal-privilege	return: OK	P

Continued on next page

Table 5: Moderator Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.4.e	Moderator Module	Remove Comment	Invalid CommentID	comment: unknown-comment-id and user-token: valid-token, get Comment: comment-not-found	return: comment-not-found	F
UT-2.4.f	Moderator Module	Remove Comment	Comment is Not From Post	comment: non-post-comment-id and user-token: valid-token, get Comment: OK	return: comment-not-from-post	F
UT-2.4.g	Moderator Module	Remove Comment	Moderator does not have privilege	comment: known-comment-id and user-token: valid-token, get Comment: OK, role.getGetUserCom no-comment-removal-privilege	return: user-cannot-remove-comment	F
UT-2.4.h	Moderator Module	Remove Comment	Valid Comment	comment: known-comment-id and user-token: valid-token, get Comment: OK	return: OK	P
UT-2.4.i	Moderator Module	Ban User	Invalid UserID	userID: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.4.j	Moderator Module	Ban User	User is not in Community	userID: non-community-user-id and user-token: valid-token, get User: OK	return: user-not-in-community	F
UT-2.4.k	Moderator Module	Ban User	Moderator does not have privilege	userID: known-user-id and user-token: valid-token, get User: OK, role.getGetUserCom no-user-ban-privilege	return: user-cannot-ban-user	F

Continued on next page

Table 5: Moderator Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.4.l	Moderator Module	Ban User	Valid User	userID: known-user-id and user-token: valid-token, get User: OK	return: OK	P
UT-2.4.m	Moderator Module	Unban User	Invalid UserID	userID: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.4.n	Moderator Module	Unban User	User is not banned	userID: non-banned-user-id and user-token: valid-token, get User: OK	return: user-not-banned	F
UT-2.4.o	Moderator Module	Unban User	Moderator does not have privilege	userID: known-user-id and user-token: valid-token, get User: OK, role.getGetUserComm no-user-unban-privilege	return: user-cannot-unban-user	F
UT-2.4.p	Moderator Module	Unban User	Valid User	userID: known-user-id and user-token: valid-token, get User: OK	return: OK	P
UT-2.4.q	Moderator Module	Grant Post Privilege	Invalid UserID	userID: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.4.r	Moderator Module	Grant Post Privilege	User is not in Community	userID: non-community-user-id and user-token: valid-token, get User: OK	return: user-not-in-community	F

Continued on next page

Table 5: Moderator Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.4.s	Moderator Module	Grant Post Privilege	Moderator does not have privilege	userID: known-user-id and user-token: valid-token, get User: OK, role.getGetUserCom no-post-privilege	return: user-cannot-grant-post	F
UT-2.4.t	Moderator Module	Grant Post Privilege	Valid User	userID: known-user-id and user-token: valid-token, get User: OK	return: OK	P
UT-2.4.u	Moderator Module	Revoke Post Privilege	Invalid UserID	userID: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.4.v	Moderator Module	Revoke Post Privilege	User do not have privilege	userID: non- privilege-user-id and user-token: valid-token, get User: OK	return: user-not-have-privilege	F
UT-2.4.w	Moderator Module	Revoke Post Privilege	Moderator does not have privilege	userID: known-user-id and user-token: valid-token, get User: OK, role.getGetUserCom no-post-privilege	return: user-cannot-revoke-post	F
UT-2.4.x	Moderator Module	Revoke Post Privilege	Valid User	userID: known-user-id and user-token: valid-token, get User: OK	return: OK	P
UT-2.4.y	Moderator Module	Remove Comment Privilege	Invalid UserID	userID: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.4.z	Moderator Module	Remove Comment Privilege	User do not have privilege	userID: non- privilege-user-id and user-token: valid-token, get User: OK	return: user-not-have-privilege	F

Continued on next page

Table 5: Moderator Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.4.aa	Moderator Module	Remove Comment Privilege	Moderator does not have privilege	userID: known-user-id and user-token: valid-token, get User: OK, role.getGetUserComm no-comment- privilege	return: user-cannot-revoke-comment	F
UT-2.4.ab	Moderator Module	Remove Comment Privilege	Valid User	userID: known-user-id and user-token: valid-token, get User: OK	return: OK	P
UT-2.4.ac	Moderator Module	Grant Comment Privilege	Invalid UserID	userID: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.4.ad	Moderator Module	Grant Comment Privilege	User is not in Community	userID: non-community-user-id and user-token: valid-token, get User: OK	return: user-not-in-community	F
UT-2.4.ae	Moderator Module	Grant Comment Privilege	Moderator does not have privilege	userID: known-user-id and user-token: valid-token, get User: OK, role.getGetUserComm no-comment- privilege	return: user-cannot-grant-comment	F
UT-2.4.af	Moderator Module	Grant Comment Privilege	Valid User	userID: known-user-id and user-token: valid-token, get User: OK	return: OK	P
UT-2.4.ag	Moderator Module	Comment Disable	Invalid PostID	post: unknown-post-id and user-token: valid-token, get Post: post-not-found	return: post-not-found	F

Continued on next page

Table 5: Moderator Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.4.ah	Moderator Module	Comment Disable	Post is Not From Community	post: non-community-post-id and user-token: valid-token, get Post: OK	return: post-not-from-community	F
UT-2.4.ai	Moderator Module	Comment Disable	Moderator does not have privilege	post: known-post-id and user-token: valid-token, get Post: OK, role.getUserCom no-comment-privilege	return: user-cannot-disable-comment	F
UT-2.4.aj	Moderator Module	Comment Disable	Valid Post	post: known-post-id and user-token: valid-token, get Post: OK	return: OK	P
UT-2.4.ak	Moderator Module	Comment Enable	Invalid PostID	post: unknown-post-id and user-token: valid-token, get Post: post-not-found	return: post-not-found	F
UT-2.4.al	Moderator Module	Comment Enable	Post is Not From Community	post: non-community-post-id and user-token: valid-token, get Post: OK	return: post-not-from-community	F
UT-2.4.am	Moderator Module	Comment Enable	Moderator does not have privilege	post: known-post-id and user-token: valid-token, get Post: OK, role.getUserCom no-comment-privilege	return: user-cannot-enable-comment	F
UT-2.4.an	Moderator Module	Comment Enable	Valid Post	post: known-post-id and user-token: valid-token, get Post: OK	return: OK	P

Continued on next page

Table 5: Moderator Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.4.ao	Moderator Module	View Community Reports	Invalid CommunityID	communityID: unknown-community-id and user-token: valid-token, get Community: community-not-found	return: community-not-found	F
UT-2.4.ap	Moderator Module	View Community Reports	Valid Community	communityID: known-community-id and user-token: valid-token, get Community: OK	return: OK	P

Superuser Module

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-2.5.a	Superuser Module	make superuser	Invalid UserID	userID: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.5.b	Superuser Module	make superuser	Valid User	userID: known-user-id and user-token: valid-token, get User: OK	return: OK	P
UT-2.5.c	Superuser Module	ban_c user	Invalid UserID	userID: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.5.d	Superuser Module	ban_c user	Valid User	userID: known-user-id and user-token: valid-token, get User: OK	return: OK	P
UT-2.5.e	Superuser Module	unban_c user	Invalid UserID	userID: unknown-user-id and user-token: valid-token, get User: user-not-found	return: user-not-found	F
UT-2.5.f	Superuser Module	unban_c user	Valid User	userID: known-user-id and user-token: valid-token, get User: OK	return: OK	P
UT-2.5.g	Superuser Module	Delete Community	Invalid CommunityID	communityID: unknown-community-id and user-token: valid-token, get Community: community-not-found	return: community-not-found	F
UT-2.5.h	Superuser Module	Delete Community	Valid Community	communityID: known-community-id and user-token: valid-token, get Community: OK	return: OK	P

System Module

Cache Module

Table 7: Cache Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-3.1.a	Cache Module	get Item From Cache	Item Not In Cache	item:item-id	returns:item-not-in-cache	F
UT-3.1.b	Cache Module	get Item From Cache	Item TTL Expired	item:item-id	returns:item-ttl-expired	F
UT-3.1.c	Cache Module	get Item From Cache	Valid Item	item:item-id	returns:OK and item	P
UT-3.1.d	Cache Module	set Item In Cache	Invalid Item	item:item-id	returns:item-id-already-in-use	F
UT-3.1.e	Cache Module	set Item In Cache	Valid Item	item:item-id	returns:OK	P
UT-3.1.f	Cache Module	delete Item From Cache	Invalid Item	item:item-id	returns:item-not-in-cache	F
UT-3.1.g	Cache Module	delete Item From Cache	Valid Item	item:item-id	returns:OK	P
UT-3.1.h	Cache Module	update Item In Cache	Invalid Item	item:item-id	returns:item-not-in-cache	F
UT-3.1.i	Cache Module	update Item In Cache	Valid Item	item:item-id	returns:OK	P
UT-3.1.j	Cache Module	write Back To DB	Invalid Item	item:item-id	returns:item-not-in-cache	F
UT-3.1.k	Cache Module	write Back To DB	Valid Item	item:item-id	returns:OK	P
UT-3.1.l	Cache Module	update Item In Cache From DB	Invalid Item	item:item-id	returns:item-not-in-cache	F
UT-3.1.m	Cache Module	update Item In Cache From DB	Item Not Present in DB	item:item-id	returns:item-not-in-DB	P
UT-3.1.n	Cache Module	update Item In Cache From DB	Valid Item	item:item-id	returns:OK	P

Recommendation Module

Table 8: Recommendation Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-3.2.a	Recommendation Module	get Trending	Trending Content	no-input	returns:OK and trending-content	P
UT-3.2.b	Recommendation Module	get User Feed	Invalid UserID	user:invalid-user-id, get User:user-not-found	returns:invalid-user-id	F
UT-3.2.c	Recommendation Module	get User Feed	Valid UserID	user:user-id, get User:OK	returns:OK and user-feed	P

Database Access Module

User Record Module

Table 9: User Record Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.1.a	User Module	getUser	Invalid Username	user: invalid-user-name	return: invalid-user-name	F
UT-4.1.b	User Module	getUser	No User by this Username	user: unknown-user, DB-status: no-such-user	return: no-such-user	F
UT-4.1.c	User Module	getUser	Existing username	user: username, DB-status: OK	return: OK and user-record	P
UT-4.1.d	User Module	updateUser	Invalid Username	user: invalid-user-name	return: invalid-user-name	F
UT-4.1.e	User Module	updateUser	No User by this Username	user: username, DB-status: OK	return: OK	F
UT-4.1.f	User Module	updateUser	Edit User Settings (User setting is a 6-tuple) - Invalid Tuple - Invalid Profile Visibility	user: new-user-record (invalid integer for profile visibility)	return: invalid-profile-visibility	F
UT-4.1.g	User Module	updateUser	Edit User Settings (User setting is a 6-tuple) - Invalid Tuple - Invalid Chat Request Setting	user: new-user-record (invalid integer for chat request)	return: invalid-chat-request-setting	F

Continued on next page

Table 9: User Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.1.h	User Module	updateUser	Edit User Settings (User setting is a 6-tuple) - Invalid Tuple - Invalid Notification Setting - remaining 4 values	user: new-user-record (invalid 4-tuple for notification)	return: invalid-notification-setting	F
UT-4.1.i	User Module	updateUser	Edit User Settings (User setting is a 6-tuple) - Valid Tuple	user: username	return: OK	P
UT-4.1.j	User Module	updateUser	Edit email-id - Invalid email-id	user: new-user-record (email-id: invalid)	return: invalid-email	F
UT-4.1.k	User Module	updateUser	Edit email-id - Valid email-id	user: valid-new-user-record	return: OK	P
UT-4.1.l	User Module	updateUser	Edit user bio	user: username, DB-status: OK	return: OK	P
UT-4.1.m	User Module	deleteUser	Invalid Username	user: invalid-user, DB-status: user-not-found	return: invalid-user-name	F
UT-4.1.n	User Module	deleteUser	No User by this Username	user: username, DB-status: no-such-user	return: no-such-user	F
UT-4.1.o	User Module	deleteUser	Exisiting username	user: username, DB-status: OK	return: user-deleted	P

Post Record Module

Table 10: Post Record Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.2.a	Post Module	get Post	Invalid PostID/Deleted PostID	post: unknown-post-id, DB-status: post-id-not-found	return: post-not-found	F

Continued on next page

Table 10: Post Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.2.b	Post Module	get Post By Community	Invalid CommunityID/ Deleted Community	community: unknown-community-id and sort: sort-by and filter: filter-by, DB-status: community-not-found	return: community-not-found	F
UT-4.2.c	Post Module	get Post By Community	Invalid Sort-By	community: community-id and sort: invalid-sort-by and filter: filter-by	return: invalid-sort	F
UT-4.2.d	Post Module	get Post By Community	Invalid Filter-by	community: community-id and sort: sort-by and filter: invalid-filter-by	return: invalid-filter-by	F
UT-4.2.e	Post Module	get Post By Community	Valid Request	community: community-id and sort: sort-by and filter: filter-by, DB-status: OK	return: OK and post-list	F
UT-4.2.f	Post Module	get Post By User	Invalid UserID	post: unknown-user-id and sort: sort-by and filter: filter-by, DB-status: user-not-found	return: user-not-found	F
UT-4.2.g	Post Module	get Post By User	Valid Request	post: user-id and sort: sort-by and filter: filter-by, DB-status: OK	return: OK and post-list	P
UT-4.2.h	Post Module	get Post By User	Invalid Sort-By	post: user-id and sort: invalid-sort-by and filter: filter-by	return: invalid-sort	F
UT-4.2.i	Post Module	get Post By User	Invalid Filter-by	post: user-id and sort: sort-by and filter: invalid-filter-by	return: invalid-filter-by	F
UT-4.2.j	Post Module	get Post By Score User	Invalid UserID (Used for user-feed generation)	post: unknown-user-id, DB-status: user-not-found	return: user-not-found	F

Continued on next page

Table 10: Post Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.2.k	Post Module	get Post By Score Trending	Valid Request	no-input	return: post-list	P
UT-4.2.l	Post Module	get Post By Tag	Invalid TagID	tag: unknown-tag-id and sort: sort-by and filter: filter-by, DB-status: tag-not-found	return: tag-not-found	F
UT-4.2.m	Post Module	get Post By Tag	Invalid Sort-By	tag: tag-id and sort: invalid-sort-by and filter: filter-by	return: invalid-sort	F
UT-4.2.n	Post Module	get Post By Tag	Invalid Filter-By	tag: tag-id and sort: sort-by and filter: invalid-filter-by	return: invalid-filter-by	F
UT-4.2.o	Post Module	get Post By Tag	Valid Request	tag: tag-id and sort: sort-by and filter: filter-by, DB-status: OK	return: OK and post-list	P
UT-4.2.p	Post Module	get Post By Query String	Invalid Sort-By	query-string: search-string and sort: invalid-sort-by and filter: filter-by	return: invalid-sort-by	F
UT-4.2.q	Post Module	get Post By Query String	Invalid Filter-By	query-string: search-string and sort: sort-by and filter: invalid-filter-by	return: invalid-filter-by	F
UT-4.2.r	Post Module	get Post By Query String	Valid Request	query-string: search-string and sort: sort-by and filter: filter-by, DB-status: OK	return: OK and post-list	P
UT-4.2.s	Post Module	update Post	Invalid PostID	post: unknown-post-id, DB-status: post-not-found	return: post-not-found	F
UT-4.2.t	Post Module	delete Post	Invalid PostID	post: post-id, DB-status: post-not-found	return: post-not-found	F

Continued on next page

Table 10: Post Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.2.u	Post Module	delete Post By User	invalid UserID	post: user-id, DB-status: user-not-found	return: user-not-found	F
UT-4.2.v	Post Module	delete Post By User	valid UserID	post: user-id, DB-status: OK	return: OK	P
UT-4.2.w	Post Module	delete Post By Community	invalid CommunityID	post: community-id, DB-status : community-not-found	return: community-not-found	F
UT-4.2.x	Post Module	delete Post By Community	valid CommunityID	post: community-id, DB-status: OK	return: OK	P

Comment Record Module

Table 11: Comment Record Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.3.a	Comment Module	get Comment	Invalid CommentID	comment: unknown-comment-id, DB-status: comment-not-found	return: comment-not-found	F
UT-4.3.b	Comment Module	get Comment	Valid Comment	comment: known-comment, DB-status: OK	return: OK and comment	P
UT-4.3.c	Comment Module	get Comment By Post	Invalid PostID	post: unknown-post-id, DB-status: post-not-found	return: post-not-found	F
UT-4.3.d	Comment Module	get Comment By Parent	Invalid Parent CommentID	comment: unknown-comment-id, DB-status: comment-not-found	return: comment-not-found	F
UT-4.3.e	Comment Module	get Comment By Parent	Valid Comment	comment: known-comment, status: OK	return: OK and comment	P

Continued on next page

Table 11: Comment Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.3.f	Comment Module	get Comments By Ancestor	Invalid Ancestor CommentID	comment: unknown-comment-id, DB-status: comment-not-found	return: comment-not-found	F
UT-4.3.g	Comment Module	get Comments By Ancestor	Valid Ancestor CommentID	comment: known-comment, status: OK	return: OK and comment	P
UT-4.3.h	Comment Module	update Comment	Invalid CommentID	comment: unknown-comment-id, DB-status: comment-not-found	return: comment-not-found	F
UT-4.3.i	Comment Module	update Comment	Valid Comment	comment: known-comment, status: OK	return: OK	P
UT-4.3.j	Comment Module	delete Comment	Invalid CommentID	comment: unknown-comment-id, DB-status: comment-not-found	return: comment-not-found	F
UT-4.3.k	Comment Module	delete Comment	Valid Comment	comment: known-comment, status: OK	return: OK	P
UT-4.3.l	Comment Module	delete Comment By Post	Invalid PostID	post: unknown-post-id, DB-status: post-not-found	return: post-not-found	F
UT-4.3.m	Comment Module	delete Comment By Post	Valid PostID	post: known-post-id, DB-status: OK	return: OK	P

Vote Record Module

Table 12: Vote Record Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.4.a	Vote Module	get Vote	Invalid VoteID	vote: unknown-vote-id, DB-status: vote-not-found	return: vote-not-found	F
UT-4.4.b	Vote Module	get Vote	Valid Vote	vote: known-vote, DB-status: OK	return: OK and vote	P
UT-4.4.c	Vote Module	get Vote By Post	Invalid PostID	post: unknown-post-id, DB-status: post-not-found	return: post-not-found	F
UT-4.4.d	Vote Module	get Vote-count By Post	Invalid PostID	post: unknown-post-id, DB-status: post-not-found	return: vote-not-found	F
UT-4.4.e	Vote Module	get Vote-count By Post	Valid PostID	post: known-post-id, DB-status: OK	return: OK and vote-count	P
UT-4.4.f	Vote Module	get Vote By Post	Valid PostID	post: known-post-id, DB-status: OK	return: OK and vote-list	P
UT-4.4.g	Vote Module	get Vote By Post-User	Invalid PostID	post: unknown-post-id and user: user-id, DB-status: post-not-found	return: post-not-found	F
UT-4.4.h	Vote Module	get Vote By Post-User	Invalid UserID	post: post-id and user: unknown-user-id, DB-status: user-not-found	return: user-not-found	F
UT-4.4.i	Vote Module	get Vote By Post-User	Valid Request	post: post-id and user: user-id, DB-status: OK	return: OK and vote	P
UT-4.4.j	Vote Module	update Vote	Invalid VoteID	vote: unknown-vote-id, DB-status: vote-not-found	return: vote-not-found	F
UT-4.4.k	Vote Module	update Vote	Valid Vote	vote: known-vote, DB-status: OK	return: OK	P
UT-4.4.l	Vote Module	delete Vote	Invalid VoteID	vote: unknown-vote-id, DB-status: vote-not-found	return: vote-not-found	F

Continued on next page

Table 12: Vote Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.4.m	Vote Module	delete Vote	Valid Vote	vote: known-vote, DB-status: OK	return: OK	P
UT-4.4.n	Vote Module	delete Vote By Post	Invalid PostID	post: unknown-post-id, DB-status: post-not-found	return: post-not-found	F
UT-4.4.o	Vote Module	delete Vote By Post	Valid PostID	post: known-post-id, DB-status: OK	return: OK	P
UT-4.4.p	Vote Module	delete Vote By Post-User	Invalid PostID	post: unknown-post-id and user: user-id, DB-status: post-not-found	return: post-not-found	F
UT-4.4.q	Vote Module	delete Vote By Post-User	Invalid UserID	post: post-id and user: unknown-user-id, DB-status: user-not-found	return: user-not-found	F
UT-4.4.r	Vote Module	update Vote By Post-User	Invalid PostID	post: unknown-post-id and user: user-id, DB-status: post-not-found	return: post-not-found	F
UT-4.4.s	Vote Module	update Vote By Post-User	Invalid UserID	post: post-id and user: unknown-user-id, DB-status: user-not-found	return: user-not-found	F
UT-4.4.t	Vote Module	update Vote By Post-User	Valid Request	post: post-id and user: user-id, DB-status: OK	return: OK	P

Chat(Private Messages) Record Module

Table 13: Chat Record Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.5.a	Chat Module	get Chat	Invalid ChatID	chat: unknown-chat-id, DB-status: chat-not-found	return: chat-not-found	F

Continued on next page

Table 13: Chat Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.5.b	Chat Module	get Chat	Valid Chat	chat: known-chat, DB-status: OK	return: OK	P
UT-4.5.c	Chat Module	delete Chat	Invalid ChatID	chat: unknown-chat-id, DB-status: chat-not-found	return: chat-not-found	F
UT-4.5.d	Chat Module	delete Chat	Valid Chat	chat: known-chat, DB-status: OK	return: OK	P
UT-4.5.e	Chat Module	delete Message-Chat	Invalid MessageID	message: unknown-message-id and chat: chat-id, DB-status: message-not-found	return: message-not-found	F
UT-4.5.f	Chat Module	delete Message-Chat	Invalid ChatID	message: message-id and chat: invalid-chat-id, DB-status: OK	return: invalid-chat-id	F
UT-4.5.f	Chat Module	delete Message-Chat	Valid Message	message: message-id and chat: chat-id, DB-status: OK	return: OK	P

Message(Group and Private) Record Module

Table 14: Message Record Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.6.a	Message Module	get Message	Invalid MessageID	message: unknown-message, DB-status: message-not-found	return: message-not-found	F
UT-4.6.b	Message Module	get Message	Valid Message	message: known-message, DB-status: OK	return: OK and message	P
UT-4.6.c	Message Module	get Messages By Chat	Invalid ChatID	chat: unknown-chat-id, DB-status: chat-not-found	return: chat-not-found	F

Continued on next page

Table 14: Message Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.6.d	Message Module	get Messages By Chat	Valid Chat	chat: known-chat-id, DB-status: OK	return: OK and message-list	P
UT-4.6.e	Message Module	get Messages By Group	Invalid GroupID	group: unknown-group-id, DB-status: group-not-found	return: group-not-found	F
UT-4.6.f	Message Module	get Messages By Group	Valid Group	group: known-group-id, DB-status: OK	return: OK and message-list	P
UT-4.6.g	Message Module	update Message	Invalid MessageID	message: unknown-message, DB-status: message-not-found	return: message-not-found	F
UT-4.6.h	Message Module	update Message	Valid Message	message: known-message, DB-status: OK	return: OK	P
UT-4.6.i	Message Module	delete Message	Invalid MessageID	message: unknown-message, DB-status: message-not-found	return: message-not-found	F
UT-4.6.j	Message Module	delete Message	Valid Message	message: known-message, DB-status: OK	return: OK	P
UT-4.6.k	Message Module	delete Messages By Chat	Invalid ChatID	chat: unknown-chat, DB-status: chat-not-found	return: chat-not-found	F
UT-4.6.l	Message Module	delete Messages By Chat	Valid Chat	chat: known-chat, DB-status: OK	return: OK	P
UT-4.6.m	Message Module	delete Messages By Group	Invalid GroupID	group: unknown-group, DB-status: group-not-found	return: group-not-found	F
UT-4.6.n	Message Module	delete Messages By Group	Valid Group	group: known-group, DB-status: OK	return: OK	P

Group Record Module

Table 15: Group Record Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.7.a	Group Module	get Group	Invalid GroupID	group: unknown-group, DB-status: group-not-found	return: group-not-found	F
UT-4.7.b	Group Module	get Group	Valid Group	group: known-group, DB-status: OK	return: OK and group	P
UT-4.7.c	Group Module	delete Group	Invalid GroupID	group: unknown-group, DB-status: group-not-found	return: group-not-found	F
UT-4.7.d	Group Module	delete Group	Valid Group	group: known-group, DB-status: OK	return: OK	P

User_chat Record Module

Table 16: User-Chat Record Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.8.a	User-Chat Module	get User and Chat	Invalid UserID	user: unknown-user and chat-id, DB-status: user-not-found	return: user-not-found	F
UT-4.8.b	User-Chat Module	get User and Chat	Invalid ChatID	user: user-id and chat: unknown-chat, DB-status: chat-not-found	return: chat-not-found	F
UT-4.8.c	User-Chat Module	get User and Chat	Invalid UserID and ChatID tuple	user: known-user, chat: known-chat, DB-status: unknown-user-chat-tuple	return: user-not-in-chat	F
UT-4.8.d	User-Chat Module	get User and Chat	Valid Request	user: known-user, chat: known-chat, status: OK	return: OK	P
UT-4.8.e	User-Chat Module	delete User and Chat	Invalid UserID	user: unknown-user and chat-id, DB-status: user-not-found	return: user-not-found	F

Continued on next page

Table 16: User-Chat Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.8.f	User-Chat Module	delete User and Chat	Invalid ChatID	user: user-id and chat: unknown-chat, DB-status: chat-not-found	return: chat-not-found	F
UT-4.8.g	User-Chat Module	delete User and Chat	Invalid UserID and ChatID tuple	user: known-user, chat: known-chat, DB-status: unknown-user-chat-tuple	return: user-not-in-chat	F
UT-4.8.h	User-Chat Module	delete User and Chat	Valid Request	user: known-user, chat: known-chat, status: OK	return: OK	P

User_group Record Module

Table 17: User-Group Record Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.9.a	User-Group Module	get User and Group	Invalid UserID	user: unknown-user and group-id, DB-status: user-not-found	return: user-not-found	F
UT-4.9.b	User-Group Module	get User and Group	Invalid GroupID	user: user-id and group: unknown-group, DB-status: group-not-found	return: group-not-found	F
UT-4.9.c	User-Group Module	get User and Group	Invalid UserID and GroupID tuple	user: known-user, group: known-group, DB-status: unknown-user-group-tuple	return: user-not-in-group	F
UT-4.9.d	User-Group Module	get User and Group	Valid Request	user: known-user, group: known-group, status: OK	return: OK	P

Continued on next page

Table 17: User-Group Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.9.e	User-Group Module	delete User and Group	Invalid UserID	user: unknown-user and group-id, DB-status: user-not-found	return: user-not-found	F
UT-4.9.f	User-Group Module	delete User and Group	Invalid GroupID	user: user-id and group: unknown-group, DB-status: group-not-found	return: group-not-found	F
UT-4.9.g	User-Group Module	delete User and Group	Invalid UserID and GroupID tuple	user: known-user, group: known-group, DB-status: unknown-user-group-tuple	return: user-not-in-group	F
UT-4.9.h	User-Group Module	delete User and Group	Valid Request	user: known-user, group: known-group, status: OK	return: OK	P

Community Record Module

Table 18: Community Record Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.10.a	Community Module	get Community	Invalid CommunityID	community: unknown-community, DB-status: community-not-found	return: community-not-found	F
UT-4.10.b	Community Module	get Community	Valid Community	community: known-community, DB-status: OK	return: OK	P
UT-4.10.c	Community Module	delete Community	Invalid CommunityID	community: unknown-community, DB-status: community-not-found	return: community-not-found	F

Continued on next page

Table 18: Community Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.10.d	Community Module	delete Community	Valid Community	community: known-community, DB-status: OK	return: OK	P
UT-4.10.e	Community Module	update Community Bio	Invalid CommunityID	community: unknown-community and bio: community-bio, DB-status: community-not-found	return: community-not-found	F
UT-4.10.f	Community Module	update Community Bio	Valid Community	community: known-community and bio: community-bio, DB-status: OK	return: OK	P
UT-4.10.g	Community Module	update Community Settings	Invalid CommunityID	community: unknown-community and settings: setting-tuple, DB-status: community-not-found	return: community-not-found	F
UT-4.10.h	Community Module	update Community Settings	Invalid SettingTuple (Community type, post type, member privileges)	community: known-community and settings: invalid-setting-tuple	return: Invalid-setting-tuple	F
UT-4.10.i	Community Module	update Community Settings	Valid Community	community: known-community and settings: setting-tuple, DB-status: OK	return: OK	P
UT-4.10.j	Community Module	update Community Image	Invalid CommunityID	community: unknown-community and image: community-image, DB-status: community-not-found	return: community-not-found	F

Continued on next page

Table 18: Community Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.10.k	Community Module	update Community Image	Valid Community	community: known-community and image: community-image, DB-status: OK	return: OK	P

Joined Community Record Module

Table 19: Joined Community Record Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.11.a	Joined Community Module	get User-Community	Invalid UserID	user: unknown-user and community: community-id, DB-status: user-not-found	return: user-not-found	F
UT-4.11.b	Joined Community Module	get User-Community	Invalid CommunityID	user: user-id and community: unknown-community, DB-status: community-not-found	return: community-not-found	F
UT-4.11.c	Joined Community Module	get User-Community	Invalid UserID and CommunityID tuple	user: known-user, community: known-community, DB-status: unknown-user-community-tuple	return: user-not-in-community	F
UT-4.11.d	Joined Community Module	get User-Community	Valid Request	user: known-user, community: known-community, status: OK	return: OK	P
UT-4.11.e	Joined Community Module	get User-Community Status (requested, invited, joined, blocked)	Invalid UserID	user: unknown-user and community: community-id, DB-status: user-not-found	return: user-not-found	F

Continued on next page

Table 19: Joined Community Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.11.f	Joined Community Module	get User-Community Status (requested, invited, joined, blocked)	Invalid CommunityID	user: user-id and community: unknown-community, DB-status: community-not-found	return: community-not-found	F
UT-4.11.g	Joined Community Module	get User-Community Status (requested, invited, joined, blocked)	Invalid UserID and CommunityID tuple	user: known-user, community: known-community, DB-status: unknown-user-community-tuple	return: user-not-in-community	F
UT-4.11.h	Joined Community Module	get User-Community Status (requested, invited, joined, blocked)	Valid Request	user: known-user, community: known-community, status: OK	return: OK and user-status	P
UT-4.11.i	Joined Community Module	delete User-Community	Invalid UserID	user: unknown-user and community: community-id, DB-status: user-not-found	return: user-not-found	F
UT-4.11.j	Joined Community Module	delete User-Community	Invalid CommunityID	user: user-id and community: unknown-community, DB-status: community-not-found	return: community-not-found	F
UT-4.11.k	Joined Community Module	delete User-Community	Invalid UserID and CommunityID tuple	user: known-user, community: known-community, DB-status: unknown-user-community-tuple	return: user-not-in-community	F
UT-4.11.l	Joined Community Module	delete User-Community	Valid Request	user: known-user, community: known-community, status: OK	return: OK	P

Continued on next page

Table 19: Joined Community Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.11.m	Joined Community Module	update User-Community Status (requested, invited, joined, blocked)	Invalid UserID	user: unknown-user and community: community-id, DB-status: user-not-found	return: user-not-found	F
UT-4.11.n	Joined Community Module	update User-Community Status (requested, invited, joined, blocked)	Invalid CommunityID	user: user-id and community: unknown-community, DB-status: community-not-found	return: community-not-found	F
UT-4.11.o	Joined Community Module	update User-Community Status (requested, invited, joined, blocked)	Invalid UserID and CommunityID tuple	user: known-user, community: known-community, DB-status: unknown-user-community-tuple	return: user-not-in-community	F
UT-4.11.p	Joined Community Module	update User-Community Status (requested, invited, joined, blocked)	Valid Request	user: known-user, community: known-community, status: (joined / invited/ requested)	return: OK	P
UT-4.11.q	Joined Community Module	update User-Community Status (requested, invited, joined, blocked)	Invalid Status	user: known-user, community: known-community, status: invalid-status	return: invalid-status	F
UT-4.11.r	Joined Community Module	update User-Community privileges	Invalid UserID	user: unknown-user and community: community-id, DB-status: user-not-found	return: user-not-found	F

Continued on next page

Table 19: Joined Community Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.11.s	Joined Community Module	update User-Community privileges	Invalid CommunityID	user: user-id and community: unknown-community, DB-status: community-not-found	return: community-not-found	F
UT-4.11.t	Joined Community Module	update User-Community privileges	Invalid UserID and CommunityID tuple	user: known-user, community: known-community, DB-status: unknown-user-community-tuple	return: user-not-in-community	F
UT-4.11.u	Joined Community Module	update User-Community privileges	Invalid Privileges	user: known-user, community: known-community, privileges: invalid-privileges	return: invalid-privileges	F
UT-4.11.v	Joined Community Module	update User-Community privileges	Valid Request	user: known-user, community: known-community, privileges: valid-privileges	return: OK	P
UT-4.11.w	Joined Community Module	update User-Community privileges	Valid Request	user: known-user, community: known-community, privileges: valid-privileges	return: OK	P
UT-4.11.x	Joined Community Module	update User-Community privileges	Valid Request	user: known-user, community: known-community, privileges: valid-privileges	return: OK	P

Blocked User Record Module

Table 20: Blocked User Record Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.12.a	Blocked Module	get Blocked Users(1) By UserID(2)	Invalid UserID	user: unknown-user, DB-status: user-not-found	return: user-not-found	F

Continued on next page

Table 20: Blocked User Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.12.b	Blocked Module	get Blocked Users(1) By UserID(2)	Valid Request	user: known-user, DB-status: OK	return: OK and blocked-list	P
UT-4.12.c	Blocked Module	delete Blocked User(1) By UserID(2)	Invalid UserID	user1: unknown-user and user2: user-id, DB-status: user-not-found	return: user-not-found	F
UT-4.12.d	Blocked Module	delete Blocked User(1) By UserID(2)	Invalid UserID	user1: user-id and user2: unknown-user, DB-status: user-not-found	return: user-not-found	F
UT-4.12.e	Blocked Module	delete Blocked User(1) By UserID(2)	Tuple Not Present	user1: user-id and user2: known-user, DB-status: tuple-not-found	return: tuple-not-found	F
UT-4.12.f	Blocked Module	delete Blocked User(1) By UserID(2)	Valid Request	user1: known-user and user2: known-user, DB-status: OK	return: OK	P
UT-4.12.g	Blocked Module	get User(1) By UserID(2)	Invalid UserID	user: unknown-user, DB-status: user-not-found	return: user-not-found	F
UT-4.12.h	Blocked Module	get User(1) By UserID(2)	Valid Request	user: known-user, DB-status: OK	return: OK and user	P

Reported User Record Module

Table 21: Reported User Record Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.13.a	Reported Module	get By UserID	Invalid UserID	user: unknown-user, DB-status: user-not-found	return: user-not-found	F
UT-4.13.b	Reported Module	get By UserID	Valid Request	user: known-user, DB-status: OK	return: OK and report-list	P

Continued on next page

Table 21: Reported User Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.13.c	Reported Module	get By ReportID	Invalid ReportID	report: unknown-report, DB-status: report-not-found	return: report-not-found	F
UT-4.13.d	Reported Module	get By ReportID	Valid Request	report: known-report, DB-status: OK	return: OK and report	P
UT-4.13.e	Reported Module	delete By UserID	Invalid UserID	user: unknown-user, DB-status: user-not-found	return: user-not-found	F
UT-4.13.f	Reported Module	delete By UserID	Valid Request	user: known-user, DB-status: OK	return: OK	P
UT-4.13.g	Reported Module	delete By ReportID	Invalid ReportID	report: unknown-report, DB-status: report-not-found	return: report-not-found	F
UT-4.13.h	Reported Module	delete By ReportID	Valid Request	report: known-report, DB-status: OK	return: OK	P
UT-4.13.i	Reported Module	update Report Status By ReportID	Invalid ReportID	report: unknown-report, status: report-status, DB-status: report-not-found	return: report-not-found	F
UT-4.13.j	Reported Module	update Report Status By ReportID	Invalid Report Status	report: known-report, status: invalid-status, DB-status: OK	return: invalid-status	F
UT-4.13.k	Reported Module	update Report Status By ReportID	Valid Request	report: known-report, status: report-status, DB-status: OK	return: OK	P

Roles Record Module

Table 22: Roles Record Module Unit Test

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.14.a	Roles Module	get Role By User-Community	Invalid UserID	user: unknown-user, community: community-id	return: user-not-found	F
UT-4.14.b	Roles Module	get Role By User-Community	Invalid CommunityID	user: user-id, community: unknown-community	return: community-not-found	F
UT-4.14.c	Roles Module	get Role By User-Community	Invalid UserID and CommunityID tuple	user: known-user, community: known-community, DB-status: unknown-user-community-tuple	return: user-not-in-community	F
UT-4.14.d	Roles Module	get Role By User-Community	Valid Request	user: known-user, community: known-community, DB-status: OK	return: OK and role	P
UT-4.14.e	Roles Module	update Role By User-Community	Invalid UserID	user: unknown-user, community: community-id	return: user-not-found	F
UT-4.14.f	Roles Module	update Role By User-Community	Invalid CommunityID	user: user-id, community: unknown-community	return: community-not-found	F
UT-4.14.g	Roles Module	update Role By User-Community	Invalid UserID and CommunityID tuple	user: known-user, community: known-community, DB-status: unknown-user-community-tuple	return: user-not-in-community	F
UT-4.14.h	Roles Module	update Role By User-Community	Invalid Role	user: known-user, community: known-community, role: invalid-role	return: invalid-role	F
UT-4.14.i	Roles Module	update Role By User-Community	Valid Request	user: known-user, community: known-community, role: valid-role	return: OK	P
UT-4.14.j	Roles Module	delete Role By User-Community	Invalid UserID	user: unknown-user, community: community-id	return: user-not-found	F

Continued on next page

Table 22: Roles Record Module Unit Test (Continued)

S.No	Module Name	Function	Conditions to be tested	Test Data	Expected Output	Status
UT-4.14.k	Roles Module	delete Role By User-Community	Invalid CommunityID	user: user-id, community: unknown-community	return: community-not-found	F
UT-4.14.l	Roles Module	delete Role By User-Community	Invalid UserID and CommunityID tuple	user: known-user, community: known-community, DB-status: unknown-user-community-tuple	return: user-not-in-community	F
UT-4.14.m	Roles Module	delete Role By User-Community	Valid Request	user: known-user, community: known-community, DB-status: OK	return: OK	P

UI Module

- One of the UI Module that will be tested is the Listing Service Module which is a module which is present at the client side of the application. This module is responsible for listing the communities, posts, comments and users. The module will be tested for the following functionalities:
 - List Community (Search Results and Feed)
 - List Posts (Search Results and Community and Feed)
 - List Comments
 - List Users (Search Results and Feed)
 - List Reports (Admin, Moderator, Superuser(platform related), User(All reports on a User))
 - List Community Requests (Admin, Moderator)
 - List Notifications
 - List User-Community (Admin, Moderator)
 - List Followers
 - List Following
 - List Blocked Users (Admin, Moderator, User(One User bloked by other), Superuser(blocked-from-platform))

Table 23: Listing Service Module Unit Test

S.No	Module Name	Functionality	Conditions to be tested	Test Data	Expected Output	Status
UT-5.1.a	Listing Service Module	List Community	Some Community Record in the List of Records is Invalid	list: list of community records in JSON	return: invalid-list	F
UT-5.1.b	Listing Service Module	List Community	Valid List of Community Records	list: list of community records in JSON	return: returns HTML	P

Continued on next page

Table 23: Listing Service Module Unit Test (Continued)

S.No	Module Name	Functionality	Conditions to be tested	Test Data	Expected Output	Status
UT-5.1.c	Listing Service Module	List Posts	Some Post Record in the List of Records is Invalid	list: list of post records in JSON	return: invalid-list	F
UT-5.1.d	Listing Service Module	List Posts	Valid List of Post Records	list: list of post records in JSON	return: returns HTML	P
UT-5.1.e	Listing Service Module	List Comments	Some Comment Record in the List of Records is Invalid	list: list of comment-tree records in JSON	return: invalid-list	F
UT-5.1.f	Listing Service Module	List Comments	Valid List of Comment Records	list: list of comment-tree records in JSON	return: returns HTML	P
UT-5.1.g	Listing Service Module	List Users	Some User Record in the List of Records is Invalid	list: list of user records in JSON	return: invalid-list	F
UT-5.1.h	Listing Service Module	List Users	Valid List of User Records	list: list of user records in JSON	return: returns HTML	P
UT-5.1.i	Listing Service Module	List Reports	Some Report Record in the List of Records is Invalid	list: list of report records in JSON	return: invalid-list	F
UT-5.1.j	Listing Service Module	List Reports	Valid List of Report Records	list: list of report records in JSON	return: returns HTML	P
UT-5.1.k	Listing Service Module	List Requests	Some Request Record in the List of Records is Invalid	list: list of request records in JSON	return: invalid-list	F
UT-5.1.l	Listing Service Module	List Requests	Valid List of Request Records	list: list of request records in JSON	return: returns HTML	P
UT-5.1.m	Listing Service Module	List Notifications	Some Notification Record in the List of Records is Invalid	list: list of notification records in JSON	return: invalid-list	F
UT-5.1.n	Listing Service Module	List Notifications	Valid List of Notification Records	list: list of notification records in JSON	return: returns HTML	P
UT-5.1.o	Listing Service Module	List User-Community	Some User-Community Record in the List of Records is Invalid	list: list of user-community records in JSON	return: invalid-list	F

Continued on next page

Table 23: Listing Service Module Unit Test (Continued)

S.No	Module Name	Functionality	Conditions to be tested	Test Data	Expected Output	Status
UT-5.1.p	Listing Service Module	List User-Community	Valid List of User-Community Records	list: list of user-community records in JSON	return: returns HTML	P
UT-5.1.q	Listing Service Module	List Followers	Some Follower Record in the List of Records is Invalid	list: list of follower records in JSON	return: invalid-list	F
UT-5.1.r	Listing Service Module	List Followers	Valid List of Follower Records	list: list of follower records in JSON	return: returns HTML	P
UT-5.1.s	Listing Service Module	List Following	Some Following Record in the List of Records is Invalid	list: list of following records in JSON	return: invalid-list	F
UT-5.1.t	Listing Service Module	List Following	Valid List of Following Records	list: list of following records in JSON	return: returns HTML	P
UT-5.1.u	Listing Service Module	List Blocked Users	Some Blocked User Record in the List of Records is Invalid	list: list of blocked user records in JSON	return: invalid-list	F
UT-5.1.v	Listing Service Module	List Blocked Users	Valid List of Blocked User Records	list: list of blocked user records in JSON	return: returns HTML	P

Integration Testing

Table 24: Integration Testing

S.No	Modules Integrated	Condition to be tested	Test Data	Expected Output	Status
IT-1.1	View Profile Module & Update Profile Module	Invalid User ID	user: unknown-user, DB-status: user-not-found	return: user-not-found	F
IT-1.2	View Profile Module & Update Profile Module	Invalid Update Data	user: known-user, update-data: invalid-data, DB-status: OK	return: invalid-data	F
IT-1.3	View Profile Module & Update Profile Module	Valid Request	user: known-user, update-data: valid-data, DB-status: OK	return: OK	P

Continued on next page

Table 24: Integration Testing (Continued)

S.No	Modules Integrated	Condition to be tested	Test Data	Expected Output	Status
IT-2.1	View Community Module & Update Community Module	Invalid Community ID	community: unknown-community, DB-status: community-not-found	return: community-not-found	F
IT-2.2	View Community Module & Update Community Module	Invalid Update Data	community: known-community, update-data: invalid-data, DB-status: OK	return: invalid-data	F
IT-2.3	View Community Module & Update Community Module	No Privilege	community: known-community, update-data: valid-data, DB-status: no-privilege	return: no-privilege	F
IT-2.4	View Community Module & Update Community Module	Valid Request	community: known-community, update-data: valid-data, DB-status: OK	return: OK	P
IT-3.1	View Post Module & Update Post Module	Invalid Post ID	post: unknown-post, DB-status: post-not-found	return: post-not-found	F
IT-3.2	View Post Module & Update Post Module	Invalid Update Data	post: known-post, update-data: invalid-data, DB-status: OK	return: invalid-data	F
IT-3.3	View Post Module & Update Post Module	No Privilege	post: known-post, update-data: valid-data, DB-status: no-privilege	return: no-privilege	F
IT-3.4	View Post Module & Update Post Module	Valid Request	post: known-post, update-data: valid-data, DB-status: OK	return: OK	P
IT-4.1	View Comment Module & Update Comment Module	Invalid Comment ID	comment: unknown-comment, DB-status: comment-not-found	return: comment-not-found	F

Continued on next page

Table 24: Integration Testing (Continued)

S.No	Modules Integrated	Condition to be tested	Test Data	Expected Output	Status
IT-4.2	View Comment Module & Update Comment Module	Invalid Update Data	comment: known-comment, update-data: invalid-data, DB-status: OK	return: invalid-data	F
IT-4.3	View Comment Module & Update Comment Module	No Privilege	comment: known-comment, update-data: valid-data, DB-status: no-privilege	return: no-privilege	F
IT-4.4	View Comment Module & Update Comment Module	Valid Request	comment: known-comment, update-data: valid-data, DB-status: OK	return: OK	P
IT-5.1	View User Profile Module & Chat Module	Invalid User ID	user: unknown-user, DB-status: user-not-found	return: user-not-found	F
IT-5.2	View User Profile Module & Chat Module	Invalid Chat Data	user: known-user, chat-data: invalid-data, DB-status: OK	return: invalid-data	F
IT-5.3	View User Profile Module & Chat Module	Blocked User	user: known-user, chat-data: valid-data, DB-status: blocked-user	return: blocked-user	F
IT-5.4	View User Profile Module & Chat Module	Valid Request	user: known-user, chat-data: valid-data, DB-status: OK	return: OK	P
IT-6.1	View Home Module & view Post Module	Invalid User ID	user: unknown-user, DB-status: user-not-found	return: user-not-found	F
IT-6.2	View Home Module & view Post Module	Invalid Post ID	user: known-user, post: unknown-post, DB-status: post-not-found	return: post-not-found	F
IT-6.3	View Home Module & view Post Module	Valid Request	user: known-user, post: known-post, DB-status: OK	return: OK	P
IT-7.1	View Community Module & View Post Module	Invalid Community ID	community: unknown-community, DB-status: community-not-found	return: community-not-found	F

Table 25: System Testing

S.No	Class	Test Case	Description	Expected Output
ST-1	Authenticati	Login	User login	The user is able to successfully login to the portal, in case of a superuser a separate UI is loaded
ST-2		Logout	User logout	The user successfully logs out of the system
ST-3		Register	User register	The user successfully registers with a unique user name on the platform with a registered email-id for usage in case of forgot-password
ST-4		Forgot Password	User forgot password	The user successfully resets the password using the registered email-id
ST-5	Guest User	View Profile	The user tries to view another persons user profile	Profile displayed if it is public otherwise only the reports on the profile are displayed
ST-6		Get Trending Posts	Get trending posts	Trending posts displayed
ST-7		Search	User searches for a community, post, comment, user	The search results are displayed based on the search query (as per the visibility of the search results for the user)
ST-8		View Community	The user tries to view a community	Community displayed if it is public
ST-9	Registered User	View Post	The user tries to view a post	Post displayed if it is public or from a community that the user has joined
ST-10		Join Community	The user tries to join a community	The user successfully joins the community if it public or the request is accepted in-case of request only or is invited to the community
ST-11		Leave Community	The user tries to leave a community	The user successfully leaves the community which he/she had previously joined
ST-12		Follow User	The user tries to follow another user	The user successfully follows the user
ST-13		Block User	The user tries to block another user	The user successfully blocks the user
ST-14		Report User	The user tries to report another user	The user successfully reports the user
ST-15		Report Post	The user tries to report a post	The user successfully reports the post
ST-16		Report Comment	The user tries to report a comment	The user successfully reports the comment

Continued on next page

Table 25: System Testing (Continued)

S.No	Class	Test Case	Description	Expected Output
ST-17		Chat	The user tries to chat with another user	The user successfully chats with the user depending on the status of his/her chat-request or the chat setting of the receiver
ST-18		View Home	The user tries to view the home feed	The user successfully views the home feed based on the posts from the communities and users he/she has joined/followed or according to the taste of the user
ST-19		Update Profile	Update user profile	Profile updated
ST-20		Create Community	Create community	Community created
ST-21		Create Post	Create post	Post created if the user has post privileges
ST-22		Create Comment	Create comment	Comment created if the user has comment privileges
ST-23		Reply Comment	Reply to a comment	Comment replied
ST-24		update Post	Update post	Post updated
ST-25		update Comment	Update comment	Comment updated
ST-26		Delete Post	Delete post	Post deleted
ST-27	Community Admin	Update Community Guidelines	Update community guidelines	Guidelines updated
ST-28		Update Community Settings	Update community settings	Community updated
ST-29		Update Community Roles	Update community roles	Roles updated
ST-30		Appoint Moderator	Appoint a moderator	Moderator appointed
ST-31		Grant Privileges to Moderator	Grant privileges to a moderator	Privileges granted
ST-32		Remove Privileges from Moderator	Remove privileges from a moderator	Privileges removed
ST-33		Remove Moderator	Remove a moderator	Moderator removed
ST-34		Delete Community	Delete a community	Community deleted

Continued on next page

Table 25: System Testing (Continued)

S.No	Class	Test Case	Description	Expected Output
ST-35	Community Moderator	View Reports	View reports	Reports displayed
ST-36		View Requests	View requests	Requests displayed
ST-37		Ban User	Ban a user	User banned
ST-38		Remove Post	Remove a post	Post removed
ST-39		Remove Comment	Remove a comment	Comment removed
ST-40		Grant Post/Comment Priv	Grant Post/Comment Priv	Post/Comment Priv granted
ST-41		Remove Post/Comment Priv	Remove Post/Comment Priv	Post/Comment Priv removed
ST-42		Unban User	Unban a user	User unbanned
ST-43	Superuser	View User-ban-from-platform Reports	View reports	Reports displayed
ST-44		Ban User from Platform	Ban a user from the platform	User banned from the platform
ST-45		Unban User from Platform	Unban a user from the platform	User unbanned from the platform
ST-46		Delete Community	Delete a community	Community deleted
ST-47		View Community-reports	View reports	Reports displayed

Performance Testing

A social media platform like Communities need to be tested on the following performance metrics:

- Response Time
- Scalability
- Handling of Load
- Cache Performance
- Concurrent Access

Latency

- The latency of such a system should ideally be very less.
- In-order to test the latency of the APIs and the client-side rendering, we can measure the time taken by the APIs to respond to the client requests under different amount of load on the API server.
- Generally some of the modules which require low latency are the Chat Module, the Listing Service Module(client-side rendering), Search Module, Comments Module, Login Module, etc.

Scalability

- The system should be able to scale with increasing number of users as well as concurrent requests by multiple users.
- We can try to test the scalability of the system by increasing the number of users and the number of requests made by each user.
- We can measure the response time under on changing the above parameters.

Load Handling/Load Testing

- Ideally the system should be able to handle a large number of requests at the same time as well as prevent the system from crashing.
- In order to prevent the system from crashing under large amount of requests, the system must be able to prioritize the requests and handle them accordingly.
- The system should be able to downscale the requests which are not important and handle the requests which are important under very high load.
- This is the role of the Job Queue included in the system architecture and it's behaviour can be tested under various amount/ type of load.

Cache Performance

- The system should be able to cache the data which is frequently accessed and reduce the load on the database.
- However including cache in the system should not increase the latency of the system because each request is now being handled by the cache server and the database server and hence takes more time in case of cache-miss.
- The cache performance can be tested by measuring the percentage of cache-hits and cache-misses under different amount of load and under random order of requests.

Concurrent Access

- The system should be able to handle multiple requests from the same user at the same time.
- Some modules that require concurrent access are the Chat Module, the Comment Module, the Notification Module, the Report Module(multiple moderators can access the same report at the same time), etc.
- Efficiently serving responses for same item from the database to multiple users at the same time is a challenge and the system should be able to handle this.
- In-order to serve the same item to multiple users at the same time, the system should ideally use the cache server to serve the same item to multiple users at the same time.
- In-order to test the concurrent access, we can simulate multiple users accessing the same item at the same time and measure the time taken by the system to respond to each user.

Test Analysis

Functional Test Report

- Some methodologies used for the test-cases are as follows:
 - **Equivalence class partitioning** - The inputs in the test cases are divided into different equivalence classes and the test cases are written for each equivalence class. (for example in case of login - invalid username and valid username, valid and invalid emails, etc.)
 - **Pair-wise testing** - Different inputs for a test case are combined in different test cases using the technique of pair-wise testing.(for example in case of login - username and password are the two fields)
 - **Boundary value analysis** - The test cases are written for the boundary values of the inputs.
 - **State-based testing** - For certain test cases (for example cache testing, the cache is present in multiple states - TTL-expired, in-cache, not-in-cache, etc.)

S.No	Module	Test Case Count
1	Authentication	9
2	Guest User	17
3	Registered User	79

S.No	Module	Test Case Count
4	Community Admin	14
5	Community Mod	42
6	Superuser	8
7	Cache	14
8	Recommendation	3
9	User Record	15
10	Post Record	24
11	Comment Record	13
12	Vote Record	20
13	Chat Record	7
14	Message Record	14
15	Group Record	5
16	User_chat Record	8
17	User_group	8
18	community record	11
19	Joined Table	24
20	Blocked Table	8
21	Reports Table	11
22	Roles Table	13
23	UI Modules	22

Total Test Cases: 389