Communities A Social Media Platform(SM02)

Test Plan

Harshit Pant CS21BTECH11021 Satpute Aniket Tukaram CS21BTECH11056

Mahin Bansal CS21BTECH11034 Burra Vishal Mathews CS21BTECH11010

Table of Contents

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
  - E2EE 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

```
Integrating testing
    The order in which you will integrate your modules and test cases for integrated modules.
System testing
Performance testing
```

Unit Tests

Authentication Module

Table 1: Authentication Module Unit Test

S.No	Module Name	Conditions to be tested	Inputs	Expected Output	Status
UT-1.1.a	Login Module	Incorrect Credentials	user: unknown-username **o** wrong-password, status: user-not-found **or** wrong-password (from User Module)	return: invalid-credentials	F
UT-1.1.b	Login Module	Incorrect Credentials (OAuth)	user: incorrect-email **or** wrong-password, status: user-not-registered **or** wrong-password (from OAuth)	return: invalid-credentials	F

Continued on next page

Table 1: Authentication Module Unit Test (Continued)

S.No	**Module Name**	**Conditions to be tested**	**Inputs**	**Expected Output**	**Status**
UT-1.1.c	Login Module	Correct Credentials	user: registered-username **and** correct-password, status: OK (from User Module or OAuth)	return: OK **and** JSON Web-token	P
UT-1.1.d	Token Module	Invalid Token			F
UT-1.1.e	Signup Module	Invalid Credentials			F
UT-1.1.f	Signup Module	Valid Credentials			P
UT-1.1.g	E2EE Module	Invalid Key			F

S.No	Module Name	Conditions to be tested	Inputs	Expected Output	Status
UT-1.1.a	Login Module	Incorrect Credentials	user: unknown-username **or** wrong-password, status: user-not-found **or** wrong-password (from User Module)	return: invalid-credentials	F
UT-1.1.b		Incorrect Credentials (OAuth)	user: incorrect-email **or** wrong-password, status: user-not-registered **or** wrong-password (from OAuth)	return: invalid-credentials	F
UT-1.1.c		Correct Credentials	user: registered-username **and** correct-password, status: OK (from User Module or OAuth)	return: OK **and** JSON Web-token	P
UT-1.1.d	Token Module	Invalid Token			F
UT-1.1.e	Signup Module	Invalid Credentials			F
UT-1.1.f		Valid Credentials			P
UT-1.1.g	E2EE Module	Invalid Key			F

User Module

S.No	Module Name	Conditions to be tested	Test Data	Expected Output	Status
UT-2.1.a					

System Module

S.No	Module Name	Conditions to be tested	Test Data	Expected Output	Status
UT-3.1.a					

DB Access Module

S.No	Module Name	Conditions to be tested	Test Data	Expected Output	Status
UT-4.1.a					

UI Module

S.No	Module Name	Conditions to be tested	Test Data	Expected Output	Status
UT-5.1.a					

Other Module

S.No	Module Name	Conditions to be tested	Test Data	Expected Output	Status
UT-6.1.a					

Integration Testing

```
Integrating different modules and testing
    Unit tested submodules combined and tested
    Use bottom-up approach for integration of modules
```

S.No	Modules Integrated	Condition to be tested	Test Data	Expected Output	Status
1.1					

System Testing

```
Testing of system as a whole
```

Performance Testing

```
Performance testing plan
```

Test Analysis

```
We discussed the following seven types of black box testing in the class: equivalence class partitioning, bou
```

```
Performance test plan/report
```

Test Statistics

- Number of classes:
- Number of methods:
- Number of modules tested:
- Number of test cases:
- Number of test cases failed:

Functional Test Report

```
Functional test summary in the form of a table
    Test case count for each module
    Type of testing method used
        equivalence class partitioning
        boundary value analysis
        cause-effect graphing
        pair-wise testing
        special cases
        error guessing
        state based testing
```

S.No	Module Name	Test Case Count	Testing Method

Performance Test Report