**Test plan for Social networking web application**

Table of Contents

[1 Introduction](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524393)

[1.1 Purpose](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524394)

[1.2 Project Overview](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524395)

[2 Scope](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524396)

[2.1 In-Scope](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524397)

[2.2 Out-of-Scope](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524398)

[3 Testing Strategy](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524399)

[3.1 Test Objectives](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524400)

[3.2 Data Approach](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524402)

[3.3 Level of Testing](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524403)

[3.4 Functional Testing](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524405)

[4 Execution Strategy](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524408)

[4.1 Entry Criteria](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524409)

[4.2 Exit criteria](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524410)

[4.3 Validation and Defect Management](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524411)

[5 Environment Requirements](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524412)

[5.1 Test Environments](file:///C:\Users\dsingla\Downloads\Test%20plan%20&amp;%20strategy%20for%20Network%20Monitoring%20system%201.docx#_Toc515524413)

## 

1. **Introduction**
   1. **Purpose**

The purpose of a test plan for the social networking web application is to test the application to make sure that every functionality is working properly and as expected.

**1.2 Project Overview**

A social networking web application is like a digital community where people can create profiles, connect with friends, share updates, photos, and videos, and interact with each other through messages, comments, and likes. It's a place where users can discover new people, interests, and content, and stay connected with others online.

1. **Scope**

**2.1 In-scope**

This test plan covers testing activities for all key features of the social networking web application, including user registration, authentication, profile management, content creation, social interactions, notifications, and overall system performance.

**2.2 Out-of-Scope**

**3.** **Testing Strategy**

**3.1 Test Objectives**

The objective of this test plan is to ensure the functionality, usability, performance, and security of the social networking web application.

**3.2 Data Approach:**

* Valid test data
* Invalid test data
* New data entry
* Real time update of data over the application

**3.3 Levels of testing:**

* **Smoke Testing:** Smoke testing is done to determine whether the most important features work as expected and that there are no blockers in the build that can potentially lead to blocking the entire testing team.
* **Unit Testing:** In this testing each individual component of the application will be tested whether they are working as expected or not.
* **Functional Testing**: In this testing tester tests whether application is working as expected or not as per functionalities defined.
* **Non-Functional Testing:** This is type of testing where software UI is tested to check whether UI is as per defined condition or not.
* **Regression Testing**: Type of testing to test whether application is working fine or not due to changes in code.

## 3.4 Functional Testing

Features to be tested.

* Login functionality
* Profile Management
* Content Creation
* Social Interaction
* Application settings

**4 Execution strategy**

**4.1 Entry criteria**

* All the requirements must be ready.
* Application must be prepared to test.
* Test Data must be prepared.

**4.2 Exit criteria**

* All test cases have been executed and passed.
* Defects found during testing have been resolved or mitigated.
* Application performance meets acceptable criteria.
* Security vulnerabilities have been addressed.
* Compatibility testing has been successfully completed on all target browsers and devices.

**4.3 Validation and Defect Management**

**Validation:**

·      Ensuring that the software meets specified requirements

·       Verifying functionality, performance, and security

·       Conducting user acceptance testing (UAT)

·       Validating against use cases and user stories

·       Confirming compatibility with browsers and devices

**Defect Management:**

·       Logging defects in a centralized system (e.g., Jira)

·       Assigning severity and priority to defects (High, Medium, Low).

·       Providing detailed steps to reproduce issues

·       Tracking the status of defects (e.g., open, in progress, resolved)

·       Collaborating with developers to resolve defects

·       Conducting regression testing after defect fixes

·        Analyzing defect trends for process improvement

# 

# **5. Environment Requirements:**

**5.1 Test Environments:**

1. **Development Environment:**
   * Purpose: To develop and unit test software components.
   * Requirements: Development IDEs, compilers, debuggers, version control systems.
   * Configuration: Developers' workstations or virtual development servers.
2. **Integration Environment:**
   * Purpose: To integrate and test software components together.
   * Requirements: Middleware, integration tools, stubs/mocks, shared databases.
   * Configuration: Dedicated integration servers or virtual machines.
3. **Staging Environment:**
   * Purpose: To conduct pre-production testing in an environment similar to production.
   * Requirements: Same as production environment, but with synthetic data.
   * Configuration: Separate servers or virtual machines in a staging network.
4. **Production Environment:**
   * Purpose: To host the live/production version of the application.
   * Requirements: High availability, scalability, redundancy, security measures.
   * Configuration: Production servers, load balancers, databases, and network infrastructure.
5. **Testing Tools and Software:**
   * Purpose: To support various testing activities such as automated testing, performance testing, and monitoring.
   * Requirements: Testing frameworks, automation tools, performance testing tools, monitoring tools.
   * Configuration: Installed on designated testing servers or virtual machines.
6. **Relevant Data Sets:**
   * Purpose: To provide realistic test data for testing scenarios.
   * Requirements: Sample data, anonymized production data, synthetic data.
   * Configuration: Stored in databases or file systems accessible to testing environments.
7. **Network Configuration:**
   * Purpose: To configure network settings for communication between different components.
   * Requirements: IP addressing, routing, firewall rules, VPN connections.
   * Configuration: Configured on network devices such as routers, switches, and firewalls.
8. **Hardware and Software Configuration:**
   * Purpose: To configure hardware and software components according to testing requirements.
   * Requirements: Hardware specifications, operating systems, middleware, databases.
   * Configuration: Installed and configured on respective servers or virtual machines.
9. **Isolation:**
   * Purpose: To ensure that each environment is isolated from others to prevent interference.
   * Requirements: Segmented networks, virtualization, containerization.
   * Configuration: Isolated VLANs or virtual networks for each environment.
10. **Access Control:**
    * Purpose: To control access to testing environments based on roles and responsibilities.
    * Requirements: Role-based access control (RBAC), permissions management.
    * Configuration: Access control lists (ACLs), authentication mechanisms, user roles and permissions.
11. **Version Control:**
    * Purpose: To manage and track changes to software configurations and test artifacts.
    * Requirements: Version control system (e.g., Git, SVN).
    * Configuration: Repositories for storing code, scripts, configurations, and documentation.
12. **Monitoring and Logging:**

* Purpose: To monitor system performance, detect issues, and capture log data for analysis.
* Requirements: Monitoring tools, logging frameworks.
* Configuration: Configured to collect metrics, logs, and alerts from servers, applications, and infrastructure components.

1. **Backup and Restore Procedures:**
   * Purpose: To ensure data integrity and recoverability in case of failures.
   * Requirements: Backup solutions, disaster recovery plans.
   * Configuration: Scheduled backups, automated restore procedures, offsite backups for critical data.