# **Test Plan** **“WЕАre Social Network”**

# TABLE OF CONTENTS:

## **OVERVIEW:**

* 1. *INTRODUCTION*
  2. *TEST PURPOSE*
  3. *OBJECTIVES*
  4. *TASKS*

## **SCOPE OF TESTING:**

* 1. *FUNCTIONALITIES TO BE TESTED*
  2. *FUNCTIONALITIES NOT TO BE TESTED*

## **ENTRY CRITERIA / EXIT CRITERIA:**

1. *ENTRY CRITERIA*
2. *EXIT CRITERIA*

## **TEST ENVIRONMENT SETUP:**

* 1. *HARDWARE*
  2. *OPERATING SYSTEMS*
  3. *DESKTOP BROWSERS*

## **SCHEDULES AND TIMELINES**

## **ROLES & RESPONSIBILITIES**

## **MAJOR DELIVERABLES**

## **DEPENDENCIES**

## **TOOLS**

## **Defect Severity Definitions**

## **USEFUL LINKS TO REPORTS & RESOURCES**

# **I. OVERVIEW**

1.1 INTRODUCTION

The Weare Social Networking Platform enables users to connect, share, and engage in social activities online.

Our Testing Strategy outlines the plan to evaluate the platform comprehensively. It defines the scope, methods, resources, and timeline. Our goal is to ensure the platform meets functional and performance criteria, providing a seamless user experience, and minimizing defects before its public launch.

## 1.2 TEST PURPOSE

This Testing Strategy is the guiding compass for our final project, representing a culmination of our learning journey in the boot camp course. It fulfills the following essential roles:

* **Clear Project Scope** - it articulates the project's scope comprehensively, encompassing all aspects from core functionality to performance and security.
* **Optimized Resource Utilization** - specifying testing methodologies and resource requirements ensures efficient allocation of our newly acquired skills and knowledge.
* **Timely Achievement** - the strategy establishes a well-defined timeline, facilitating the prompt identification and resolution of any challenges we may encounter.
* **Quality Assurance** - our goal is to ensure that the final project not only meets but exceeds the expectations set by the boot camp course. We strive to provide an exceptional user experience and deliver a polished project;

## 1.3 OBJECTIVES

### **1.3.1 Comprehensive Testing Plan:**

* Develop a detailed testing plan that covers critical aspects such as user registration, profile management, post creation, liking, commenting, and connectivity features.
* Clearly define roles and responsibilities within the testing team for effective coordination.

### **1.3.2 User Experience Validation:**

* Ensure the Weare platform delivers a superior and user-friendly experience, focusing on functionality and reliability.

### **1.3.3 Testing Approach:**

* Implement a structured testing process that addresses functionality ultimately enhancing the overall quality of the user experience.

1. Exploratory Testing
2. Manual Testing
3. Black-Box Techniques
4. Static technique testing
5. Smoke Testing
6. Integration Testing (API)
7. Automated UI Testing
8. Performance Testing (Non-Functional)
9. Security Testing (Non-Functional)

## 1.4 TASKS

* **Testing Activities** - conduct comprehensive testing of WEAre social networking platform functionalities, including user registration, profile management, post creation, liking, commenting, and connectivity features;
* **Test Documentation** - create detailed test cases, test scripts, and test data to ensure thorough test coverage;
* **Test Execution -** execute the defined test cases on the Weare platform, monitoring for any deviations from expected behavior.
* **Issue Identification** - identify and document any defects, discrepancies, or issues encountered during testing.
* **Issue Reporting** - report identified issues promptly using the designated reporting system or tools;
* **Post-Testing Activities** - summarize and analyze test results, producing comprehensive test reports.
* **Communication** - maintain open communication channels with team members and trainers to provide updates on testing progress and issue status;
* **Documentation Maintenance** - keep test documentation up to date to reflect any changes in the testing process.
* **Test Closure** - prepare for the formal closure of testing activities, including final reports and sign-off procedures.

# **II.** **SCOPE OF TESTING**

The scope of testing for WEAre Social Network application encompasses a comprehensive evaluation of various functionalities to ensure the quality and reliability of the system. This section outlines the specific functionalities that will be tested and those that will not be tested during the testing phase.

## 2.1 FUNCTIONALITIES TO BE TESTED

* + User Registration and Login Functionalities:
    - Validating Happy path for User Registration.
    - Validating Happy path for User Login.
  + Functionality Requirements for Posts:
    - Post Content Functionalities.
    - Like/Unlike Button Functionalities.
    - Commenting on Post Functionalities.
  + Non-Authenticated User Actions & Visible Pages:
    - Visibility of Landing Page Elements with Links.
    - Registration Form.
    - Sign-in Form.
    - Profile Search.
    - Public Feed Content.
    - Private Feed Content.
  + Authenticated User Actions:
    - User Authentication and Logout.
    - User Profile Update.
    - User Connection Requests.
    - Post Creation and Visibility.
    - Personalized Post Feed.
    - Interactions With Posts.
  + Administrative Actions for Admin Users:
    - Admin Profile Management.
    - Admin Post Management.
    - Admin Comment Management.

## 2.2 FUNCTIONALITIES NOT TO BE TESTED

* + User Optional Features.
  + User Email Verification.
  + User Identity Verification.
  + Footer Functionality
  + Mobile Devices - testing on mobile devices and platforms, including iOS and Android, is not covered by this test plan.
  + Browser Compatibility - extensive cross-browser compatibility testing for older or less common browsers such as Internet Explorer 11 or Safari 9 is not included in this plan. We will focus on the latest versions of popular browsers like Chrome and Firefox.
  + Third-Party Integrations - This plan does not cover verification of third-party integrations, including payment gateways, and social media sharing APIs.

# **III.ENTRY CRITERIA / EXIT CRITERIA**

## **3.1 ENTRY CRITERIA:**

1. Test data, including user accounts, posts, comments, and administrative profiles, is available and accurately represents the scenarios to be tested.
2. Test cases, test plans, and other testing documentation are prepared and reviewed.
3. Testers have access to the application and can log in using valid test user accounts.
4. Test Environment Ready: Ensure the test environment (servers, tools) is configured and ready for testing.
5. Test Data Availability: Confirm the availability of accurate test data, including user accounts and sample content.
6. Testing Documentation Prepared: Ensure that test cases, test plans, and scripts are prepared and reviewed.
7. Access to the Application: Verify that testers have access to the application using valid test accounts.
8. Resource Allocation: Ensure that testing resources, including hardware and personnel, are available.
9. Defect Reporting Setup: Set up the defect reporting tool and ensure testers are trained to use it.

## **3.2 EXIT CRITERIA:**

1. **Test Case Execution:** Execute all test cases with 100% coverage and document pass/fail results.
2. **50%** of Highest/High Priority cases are automated.
3. **Reporting after JIRA Testing Phase:** Provide a summary report after the JIRA testing phase, highlighting the results and findings specific to JIRA testing.
4. **Reporting after API Testing Phase:** Provide a summary report after the API testing phase, highlighting the results and findings specific to API testing.
5. **Reporting after REST Testing Phase:** Provide a summary report after the REST testing phase, highlighting the results and findings specific to REST testing.
6. **Reporting after Selenium Testing Phase:** Provide a summary report after the Selenium testing phase, highlighting the results and findings specific to Selenium testing.
7. **Reporting after Jmeter Performance Testing Phase:** Provide a summary report after the Jmeter Load,Stress, Endurance and Spike tests. Highlighting the results and findings specific to Performance testing.
8. **Reporting after API Security Testing Phase:** Provide a summary report after the Security testing with ZAP. Highlighting the results and findings specific to Security testing.
9. **Documentation Updates:** Update testing documentation to reflect test results and changes.
10. **Test Closure Report:** Prepare a test closure report summarizing testing activities and outcomes.
11. **Approval:** Obtain approval of test results and the test closure report.

# **IV.TEST ENVIRONMENT SETUP**

The test environment setup outlines the hardware, operating systems, and desktop browsers that will be used for conducting the testing of the WEAre Social Network application. This section provides a clear understanding of the testing environment to ensure consistent and accurate test results.

## 4.1 HARDWARE

## The hardware specifications for the test environment are as follows:

* Minimum RAM: 8GB;
* Minimum Free Disk Space: 50GB;
* Processor:x64-based processor.

## 4.2 OPERATING SYSTEMS

The following operating systems will be used in the test environment:

* Windows 10 Pro: 64-bit operating system version 22H2;
* Windows 11 Home: 64-bit operating system version 22H2.

## 4.3 DESKTOP BROWSERS

The following desktop browsers will be used for testing:

* Google Chrome: Version: 117.0.5938.92 (Official Build) (64-bit);
* Mozilla Firefox: Version: 117.0.1 (64-bit).

# **V. SCHEDULES AND TIMELINES**

## 1.Project Requirement Review (14/09/2023 – 16/09/2023)

* Defining the application and client requirements.
* Identifying the functional requirements of the application.
* Identifying the required test tools and environment.
* Identifying deadlines and constraints.

## 2.Environmental/Tool Setup (16/09/2023 – 18/09/2023)

* Hardware and software setup
* Database configuration
* Server and network configuration
* Configuration of communication tools
* Test tools installation and configurations

## 3.Test Planning (18/09/2023 – 21/09/2023)

* Defining scope and constraints of the project.
* Prioritizing the functional requirements of the application for MUST, SHOULD and COULD keywords from the given documentation.
* Specifying test objectives
* Allocating resources
* Identifying and mitigating risks
* Listing deliverables
* Setting exit criteria
* Managing test data and environment
* Creating execution schedule
* Handling defects
* Reporting progress and metrics
* Controlling changes
* Communicating effectively

## 4.Test Case Development, Exploratory Testing & Manual Testing (20/09/2023 – 28/09/2023)

* Preparing test tool and structuring project management tool and issue types.
* Еxploratory testing of the application
* Preparing templates for issue types.
* Preparing test data for various scenarios.
* Populating the test database with sample data.
* Begin manual system testing with test cases.

## 5.Test Executions and Reporting (26/09/2023 – 28/09/2023)

* Analyze and extract reports from the results from the manual tests.

## 6.REST API Testing (29/09/2023 – 08/10/2023)

* Reviewing the API documentation.
* Creating HTTP requests and scenarios according to the given API documentation.
* Performing REST API testing through Postman.
* Performing RestAssured API tests.
* REST API test reporting.

## 7.Automated UI Testing (09/10/2023 – 19/10/2023)

### a. Functional Testing

* Smoke testing for the highest priority test cases
* Integration Testing
* Regression testing (Including Selenium Web Drive):
* Developing Selenium automation scripts for the highest priority test cases.
* Execute automated test scripts using Selenium.
* Analyze and report results from automated tests.

### b. Non-Functional Testing (18/10/2023 – 19/10/2023)

* Performing load and stress tests.
* Performing security tests for vulnerabilities.

8. Test Closure (19/10/2023)

* Compile test results, defects, and test summary reports. Hold a test closure meeting to review testing efforts.

# VI. ROLES & RESPONSIBILITIES

|  |  |
| --- | --- |
| ROLE | RESPONSIBILITIES |
| Hristo Hristov | Test Planning/Manual Testing/Functional and Non-Functional Testing |
| Timur Naumov | Test & Tool Planning/Schedule & Task Management/Manual Testing/Functional and Non-Functional Testing |
| Krasen Kochev | Test Planning/Manual Testing/Functional and Non-Functional Testing |

# VII. MAJOR DELIVERABLES

* + - 1. Test Plan.
      2. Test report.
      3. Automated tests code/repository link.
      4. Script for execution of automated tests.
      5. A document with the required prerequisites to run the tests.
      6. A document with a concise description of to run the tests, filter them, etc.
      7. High-level test cases.

# VIII. DEPENDENCIES

1. Test Environment Availability -testing relies on a stable and accessible test environment.
2. Resource Availability - testing resources, including hardware, software, and personnel, must be available as per the testing schedule.
3. Documentation – testing often depends on accurate and up-to-date documentation, including requirements and test cases.

# IX. TOOLS

## Project Management Tool: **Jira(XRAY)** for logging, tracking, and managing software defects and issues.

## Containerization: **Docker** for creating and managing containers for application deployment and testing.

## Database Management Tool: **Heidi SQL** to interact with the database for data-related testing.

## Version Control System: **Git and GitHub** for source code management and collaboration.

## API Testing Tools**: Postman** for testing API endpoints and services and Rest-Assured. **Rest-Assured** is a Java-based library that is used to test RESTful Web Services.

## Development Environment: **IntelliJ IDEA** integrated development environment for coding and scripting.

## Automation Testing Tool: **Selenium WebDriver** for automated testing of web-based applications.

## Load & Stress Testing Tool: **Apache JMeter** for simulating high user loads and stress testing.

## Collaboration and Communication: **Microsoft Teams & SharePoint** for team communication and collaboration.

# X. Severity and Priority definitions

| **Severity Value** | **Jira Severity** | **Definition** |
| --- | --- | --- |
| Severity - 1 | Blocker | A condition whereby the Website is substantially inoperable or impaired in a post-Go Live production instance with a high impact on multiple end users of the Website and a mutually agreed-upon workaround has not been implemented in such a way that the issue has been adequately mitigated. |
| Severity - 2 | Critical | A condition whereby the Website feature is not working, or a substantial performance problem exists, which causes the platform to perform poorly, impacting multiple end users of the Website. A workaround is available and can be implemented. |
| Severity - 3 | Major | A condition whereby the Website malfunctions, but end users' use of the Website is not substantially impacted. Also appropriate for "How-to Questions" and other low-impact product inquiries. |
| Severity - 4 | Minor | A condition whereby the Website, or documentation error exists, and end-user use of the Website is not impacted. Also appropriate for product enhancement requests. |

| **Priority Value** | **Jira Priority** | **Definition** |
| --- | --- | --- |
| Priority – 1 | Highest | A condition whereby a critical issue severely impairs the core functionality of the system, rendering it inoperable or significantly impacting multiple end users. No viable workaround is available, and immediate attention is required to restore normal system operation. |
| Priority - 2 | High | A condition whereby a major issue disrupts system functionality or performance, affecting multiple end users. While a workaround exists, addressing the issue promptly is essential to minimize disruption and ensure a satisfactory user experience. |
| Priority - 3 | Medium | A condition where system malfunctions or non-critical issues occur, causing some inconvenience to end users. Workaround options are available, and the problem does not significantly hinder day-to-day operations. |
| Priority - 4 | Low | A condition involving non-critical system errors, minor documentation issues, or low-impact product inquiries. End-user functionality remains largely unaffected, and the problem may not be urgent. |
| Priority - 5 | Lowest | A condition primarily related to suggestions for product improvements, enhancements, or documentation enhancements. These items do not represent critical issues and can be scheduled for future development or documentation updates. |

# USEFUL LINKS TO REPORTS & RESOURCES

## **Team SharePoint** : <https://telerikacademy.sharepoint.com/sites/FINALPROJECTTEAM/Shared%20Documents/Forms/AllItems.aspx?csf=1&web=1&e=gfrZwx&cid=86b4d6e8%2D9c22%2D4295%2D8468%2D7629869a7550&RootFolder=%2Fsites%2FFINALPROJECTTEAM%2FShared%20Documents%2FGeneral&FolderCTID=0x012000BAE2F28163C50E4B8F78E8BB327BE01C>

## **GitHub Team Repository:** <https://github.com/FINAL-PROJECT-T-H-K/Social-Network-Project>

## **Team & Test Schedule** : <https://github.com/FINAL-PROJECT-T-H-K/Social-Network-Project/tree/main/02TeamAndTestSchedule>

## **JIRA Test Coverage Report**: <https://t-h-k-qa50-final-project.atlassian.net/plugins/servlet/ac/com.xpandit.plugins.xray/test-coverage-report-page?project.key=FHKT&project.id=10002&ac.reportId=652dbc494d3e7358e1cac684>

## **JIRA Test Plan Report**: <https://t-h-k-qa50-final-project.atlassian.net/plugins/servlet/ac/com.xpandit.plugins.xray/xray-testplans-metrics-report?project.key=FHKT&project.id=10002>

## **POSTMAN HTML Report**: <https://github.com/FINAL-PROJECT-T-H-K/Social-Network-Project/tree/main/04RestApiTestingPostmanAndNewmanReport>

## JMETER TESTING AND REPORT: <https://github.com/FINAL-PROJECT-T-H-K/Social-Network-Project/tree/main/05JMeterPerformanceTestingAndReport>

## REST ASSURED API & UI Testing : <https://github.com/FINAL-PROJECT-T-H-K/Social-Network-Project/tree/main/TestAutomationFramework>

1. API SECURITY TESTING : <https://github.com/FINAL-PROJECT-T-H-K/Social-Network-Project/tree/main/07ZAPSecurityTestingReport>