TEST PLAN WEARE APP

Prepared by

DENITSA PETROVA
VLADISLAV GANCHEV
ELENA ZLATEVA

1. INTRODUCTION

This document describes the test plan for a **WEare web application project.** The purpose of this document is to define the strategy, approach, roles and responsibilities of the parties involved in the test process during the implementation of a **WEare web application project**. It describes what will be tested, what tests will be done and by whom and their responsibilities.

The social media platform plays a crucial role in connecting people, fostering communities, and facilitating communication on a global scale. The success of the web application depends on its ability to provide a smooth and efficient user experience for the users.

In order to function effectively, testing of the application is required. This process is integral to maintaining a high-quality user experience and addressing any potential issues before they impact a large user base.

2. OBJECTIVES AND TASKS

2.1 Objectives

The main purpose of the test process is to verify if the Social Media web application meets defined requirements. This means that the system works to ensure that it functions reliably, securely, and effectively for its users. This involves a range of testing activities to identify and rectify potential issues before the platform is released to a wider audience.

By conducting thorough testing, a Social Media web application enhances user satisfaction, trust, and adoption, leading to a more successful and sustainable platform in the long run. In addition, the main purpose of the test is to make sure reported bugs during the process are properly solved and to report deviations to stakeholders.

2.2 Tasks

The tasks that are identified by this Test Plan of WEare app include:

2.2.1. Planning and control

- set a group channel in Teams in Microsoft which will be the official channel for communication and update
- set a Daily meetings at 10:30 am every day from Monday to Sunday, at to discuss project advancements and challenges
- set a meeting once per week to discuss and revise the test plan on Wednesday at 5:00 pm
- set a meeting once per week with an assigned group mentor on Tuesday at 7 pm

2.2.2. Analysis and design

- the technical requirements will be analyzed and static analysis will be performed on the basis of which the high level test cases will be developed and created in JIRA.
- for priority of test cases will be used the default Jira scale, including the following five priority levels: Highest, High, Medium, Low, Lowest
- for severity of defects will be used custom scale, including the following five levels, corresponding to the priority levels of the failed test cases which indicated the presence of the defect: Critical, Major, Moderate, Low, Cosmetic
-

2.2.3 Implementation and execution

- the tests cases will be implemented in JIRA with real test data and will be manually executed grouped by functionality
- the manually executed tests will be analyzed and appropriate tests for automation with Selenium will be chosen
- the manual test cases for which an API alternative is present will be automated with Postman and Rest Assured

2.2.4 Exit criteria and reports

- after analyzing and prioritizing the functionalities and requirements the exit criteria will be set in a way to guarantee that the most crucial functionalities will be covered
- after the testing of each functionality a report will be provided including pass/fail rate and number of bugs reported

2.2.5 Closure

- after the reports are sent to the clients, a team meeting will be held to discuss what was achieved, what went wrong and what is left for development

3. SCOPE

The test cases will cover not only the “Happy paths” but also alternative paths to validate that the delivered functionality is built according to the requirement.

3.1 Features to be Tested:

- registration
- login

- personal profile
- professional profile
- administrative profile
- public feed
- private feed
- create post
- create comment
- like post/comment
- connect/disconnect with other users
- REST API functionalities

*The full list of all high level test cases is provided in High Level Test Cases document.

3.2 Features Not to be Tested

- Footer menu, namely the sections “WEare social network”, “Useful Links”, “Account”, “Have a Question” and social media links
- “Register” option on the Personal Profile page - unnecessary and not implemented
- Database
- Latest Users Section on Homepage
- Types of accepted images and min/max image size when creating posts and comments

4. TESTING STRATEGY

System Testing

Definition: System testing will include testing of the complete and fully integrated software product WEare in order to verify that the product to be delivered meets the requirements provided in the Social Network Project Specification document, as well as in the WEare application README file. It will investigate both functional and non-functional requirements. The functionality of the software will be tested from end-to-end by a separate testing team than the development team, before the product is pushed into production.

Participants:

- Denitsa Petrova - Manual and Automation QA
- Vladislav Ganchev - Manual and Automation QA

- Elena Zlateva - Manual and Automation QA

Methodology:

Several approaches and techniques will be used:

4.1 **Requirements-based testing** - Functional Testing

The requirements provided in the Social Network Project Specification document will be carefully examined and a scheme of the corresponding functionalities will be created. On the basis of the scheme test cases will be derived, in order to exercise the system, test the implementation of the required functionalities and the degree to which it follows the technical requirements. This will be achieved through user-case testing.

User-case tests will be developed on the basis of the application's specifications, in order to guarantee full coverage of the required functionalities and interactions. Then the test cases will be divided and executed by functionality, starting with the most crucial ones, and going down the list. That way we can guarantee that the most important functionalities will be well tested, even in case of unexpected delays.

For crucial functionalities, the following additional techniques will be applied:

- Equivalence Partitioning and Boundary Value Analysis:

  For crucial functionalities with specific requirements (password, username, creation of post, creation of comment) additional Equivalence partitioning and Boundary value analysis will be implemented in order to ensure the criteria are strictly followed and well implemented.

- Decision Table

  For crucial functionalities, such as creating a profile(as user or administrator), additional Decision table technique will be implemented.

4.2 **Exploratory Testing - Functional Testing**

The exploratory testing includes exploring the software program to find defects by interacting with the system as a regular user, not strictly following predefined test steps and linear interactions, jumping back and forth through the system, thus exposing potential bugs or gaps in the code, not anticipated and not covered by the Requirements-based testing and the Static Analysis.

4.3 REST API Testing - Functional Testing

The already executed manual tests, for which an API alternative is present, will be automated with Postman and Rest Assured.

Full list of the API tests will be provided in the API Coverage Document.

4.4. Static Analysis - Functional Testing

The code will be analyzed in order to find the actual implementation of some key features, such as number of symbols for username and password, what kind of input they accept (alphabetical, numerical, special symbols), etc. In case of discrepancies between the specifications and the actual implementation, additional test cases will be derived and implemented.

5. HARDWARE REQUIREMENTS

It mentions the minimum hardware requirements that will be used to test the web application:

- Type:
  - Laptop
  - Computer

- Processor/RAM:
  - Windows:
    - Processor: 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 1.69 GHz
    - Installed RAM: 16.0 GB (15.7 GB usable)
    - System type: 64-bit operating system, x64-based processor

  - MAC:
    - Intel Processor: 1,4 GHz Quad-Core Intel Core i5 (MAC)
    - Memory: 8 GB RAM
    - System type: 64-bit operating system, x64-based processor

6. ENVIRONMENT REQUIREMENTS

Following software is required in addition to client-specific software.

- Operating system:
    - Windows 10
    - MacOs Monterey or higher with a minimum of 8GB free space.

- Browser:
    - Google Chrome 70 or higher
    - Safari 10+
    - Firefox 10+
    - Internet Explorer 5+ (IE)

7. TEST SCHEDULE

The test schedule will start on 14th of September 2023 and will end on 19th of October 2023.

The detailed schedule is provided in the Detailed Schedule document, an inseparable part of this Test Plan.

8. ENTRY AND EXIT CRITERIA

8.1 Entry Criteria

Before the testing phase can begin, the following conditions must be met:

1. Requirement Analysis: All the requirements mentioned in the project description are clearly understood and documented.
2. The fully developed application is provided, after Unit and Integration testing performed by the Developer team.
3. Test Environment Setup:

3.1 Local Environment

a.Database Setup: The database is set up and configured according to the project requirements.

b.Docker Setup: The Docker environment for the app is set up and functioning correctly.

c.Application Setup: The application (project) is set up and ready for testing.

3.2 Global Environment/Test

    a. Database Setup: The database is set up and configured according to the project requirements.
    b. Docker Setup: The Docker environment for the app is set up and functioning correctly.
    c. Application Setup: The application (project) is set up and ready for testing.
4. Test Data Preparation: Necessary test data, based on the project requirements.
5. Test Case Preparation: Test cases covering all the functional requirements are prepared.
6. Test Tools and Resources: Necessary test tools are available, and resources are allocated. The tools include:
    a. Jira: For bug reporting and tracking the progress of testing.
    b. X-Ray
    c. Swagger UI documentation
    d. Intellij
    e. Postman: For automating REST API testing.
    f. Selenium WebDriver Framework: For automating web application testing.
    g. Node.js
    h. Apache JMeter
7. The QA Team is familiar with the tools that need to be used.

8.2 Exit Criteria

Before the testing phase can be considered complete, the following conditions must be met:

1. Functional Testing: all critical features including public part, private part, and administrative part, have been tested and passed.
    11.1.1. 100% of Highest and High Priority Tests have been executed
    11.1.2 80% of Medium Priority Tests have been executed
    11.1.3. 100% of High severity and Blocker defects are reported
    11.1.4. 50% of Medium severity defects have been reported
    11.1.5. Planed time has run out

Criteria	T1 14.09-30.09	T2 01.10-07.10	T3 08.10.15.10	T4 16.10-19.10
11.1.1	❌	❌	❌	✅
11.1.2	❌	❌	❌	✅
11.1.3	❌	❌	✅	✅

11.1.4	❌	❌	✅	✅
11.1.5	❌	❌	❌	✅

9. TEST DELIVERABLES

- Test plan
- Test report
- Automated tests code/repository link
- Script for execution of automated tests
- A document with the required prerequisites to run the tests
- A document with a concise description of how to run the tests, filter them, etc.
- High-level test cases

10. RESOURCES/ROLES & RESPONSIBILITIES

Task	Members
- Planning and control	Denitsa Petrova Elena Zlateva Vladislav Ganchev
- Analysis and design	Denitsa Petrova Elena Zlateva Vladislav Ganchev
- Implementation and execution	Denitsa Petrova Elena Zlateva Vladislav Ganchev
- Exit criteria and reports	Denitsa Petrova Elena Zlateva Vladislav Ganchev
- Closure	Denitsa Petrova Elena Zlateva Vladislav Ganchev

11. RISKS/ASSUMPTIONS

Risk: The project schedule is too tight; it's hard to complete the project on time.

Mitigation: To address this challenge, the team will implement a prioritized testing approach. By setting specific test priorities for each test run, it will be able to ensure that critical components are thoroughly examined first. This strategic allocation of resources will help us maximize the testing efforts within a constrained schedule, ultimately increasing the success of on-time project completion.

Risk: Project delay due to technical issues

Mitigation: To address this challenge, the team will prioritize the resolution of technical issues, allocating the necessary resources and expertise to address them promptly. This could include a cross-team communication and escalating the issue to the trainers and technical team to support the resolution of the technical blockers.

Risk: Lack of experience with new tools including Docker, SQL database, Heidi and others

Mitigation: To address this challenge, the team will foster an open communication with the cross-team members to share their experience, learnings and insights related to the new tools. In addition, a drive will be created to compile useful resources, including tutorials, best practices, and troubleshooting guides for the new tools.

12. TOOLS

The tools that will be used include:

- JIRA
- GitHub
- Xray via JIRA
- IntelliJ
- Docker
- Heidi(or relevant program for iOS devices)
- Google Drive
- MariaDB
- FreeMySQLHosting
- Other tools required for the automation testing

13. APPROVALS

The Directors of QA should approve the test plan.

Name:

1. Diana Gospodinova, Director of QA

2. Nikolay Avramov, Director of QA

Signature Date:

End.