Test Plan

WEare

**Client**: WEare

**Project**: WEare Social Network

**Document title**: Test Plan

**Version**: 1.0

**Date**: 18/09/2023

**Authors**: Katerina Nedyalkova, Miriam Kyoseva and Nikolay Grozev

**GitHub Repository:** https://github.com/A50Team7/Final-Project

**Jira Project URL:** https://team7a50.atlassian.net/browse/FP

Table of Contents

1. **Introduction**
   1. Overview
   2. Purpose of the Application
2. **System Test**
   1. Test cases
   2. Test Data
   3. Scope
      1. In Scope
3. **Approach**
   1. Functional Testing
   2. Security Testing
   3. API Testing
   4. UI/UX Testing
4. **Reporting** 
   1. Reporting defects
   2. Test reports
5. **Exit Criteria**
6. **Schedule**
7. **Resources**
8. **Risks**
9. **Assumptions**
10. **Test Case Template**
11. **Bug Template**
12. **Priority Definitions**
13. **Severity Definitions**

# 1. Introduction

## 1. Overview

### Purpose of this document

This document describes the test plan for WEare 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 construction of WEare. It describes what will be tested, by whom and their responsibilities.

### Objective of testing

The main purpose of the test process is to verify if the product meets defined requirements, make sure bugs are discovered and reported and to report deviations to stakeholders.

### Test base

Testing will be executed based on written specifications. They are stored and shared in electronic format and will be added and modified if needed, during the test process.

### Pre-conditions before starting system test

The software development based on the approved specifications has been finished

Relevant interfaces to third party applications are connected to a system tested environment

## 2. Purpose of the application

The WEare social network is a web application designed to help users connect, share, and offer professional services. With functionalities spanning from a public-facing platform to private interactions among registered users, and administrative capabilities, this platform provides a comprehensive social networking experience.

# 2. System Test

## 1. Test cases

Detailed test cases will be written for at least high risk and critical functionality, such as register, login, posting, commenting, etc.

Detailed test cases will not be written for low risk and non-critical functionality, such as changing content of webpages. Tests executed will be logged and reported upon.

Detailed test cases might not be written for non-functional tests, such as compatibility and security tests. In such cases the test approach will be described, and tests executed are logged and reported upon.

Xray for Jira will be used to store, organize and run manual tests / test cases. The test cases will cover not only the “Happy paths” but also alternative paths to validate that the delivered functionality is build according to the requirements.

## 2. Test data

Critical to successful test execution is to have correct and sample test data. Without test data that fits test cases, tests cannot be executed. During execution of this project, participants will communicate about required test data to be created.

If needed, additional agreements will be made in order test data to be created and the timelines for this. Required and available test data will be shared in confluence.

## 3. Scope

### In Scope:

**Public Part:**

* + Homepage display and navigation.
  + Login and registration functionality.
  + Public profile display and chronological ordering of posts.
  + Profile search based on name or professional category.
  + Public feed viewing.

**Private Part (For Registered Users):**

* + Personal and login information editing.
  + Profile picture upload and visibility settings.
  + Profile details modifications (e.g., gender, location, review).
  + Professional profile with detailed service descriptions.
  + Connection request functionality.
  + Post creation with text, publicity settings, and images.
  + Personalized post feed generation based on a multifaceted algorithm.
  + Interactions with posts, including liking, disliking, and commenting.

**Administration Part:**

* + Profile editing and status changes.
  + Post and comment modifications.

**REST API:**

* + CRUD operations for posts and comments.
  + User connection requests and approvals.
  + User data retrieval and updates.

# 3. Approach

1. **Functional Testing:** Validate all listed functionalities.
2. **API Testing:** All CRUD and user operations in the REST API.
3. **UI/UX Testing:** Compatibility across browsers and devices.
4. **Database Testing:** Data integrity and relationship in MySQL/MariaDB.

# 4. Reporting

## 1. Reporting defects

All defects that are related to development of “WEare” must be submitted in or imported into the defect tracking system Jira.

Defects are classified as bugs if only:

* They are clearly described with steps to reproduce, expected result and actual result
* Have an impact/severity assigned
* Issues related to graphical design should be provided with a screenshot
* They can be reproduced
* They are deviations from agreed specifications or concern overly clear errors

## 2. Test report

* Total number of identified bugs, per severity

# 5. Exit Criteria

1. 100% of the test cases must be executed
2. 50% or more of the test cases are automated
3. API tests in Postman and REST-Assured are 100% complete and executable
4. Identified critical and major defects must be documented in Jira
5. All the test deliverables must be completed and submitted
6. Planned time has run out

# 6. Schedule

* **Preparation:** 18.09.2023-26.09.2023
* **Functional and UI Testing:** 27.09.2023-19.10.2023
* **API Testing:** 02.10.2023-19.10.2023
* **Closure:** 20.10.2023

# 7. Resources

1. **Test Environment:**
   1. Clone of the production environment.
   2. Chromium based browsers. (Chrome)
2. **Testing Tools:** Selenium, JUnit, REST-Assured, and Postman.
3. **Team:** 3 QA Engineers

Role N

|  |  |  |
| --- | --- | --- |
| **Role** | **Name and email** | **Responsibilities for test** |
| **QA Engineer (to be)** | **Myriam Kyoseva** | **Manual Testing, Test Automation, API Testing** |
| **QA Engineer (to be)** | **Nikolay Grozev** | **Manual Testing, Test Automation, API Testing** |
| **QA Engineer (to be)** | **Katerina Nedyalkova** | **Manual Testing, Test Automation, API Testing** |

# 8. Risks

1. Requirement changes during testing.
2. Unavailability of tools or environment.
3. Limited test data availability.

# 9. Assumptions

1. Test environments and tools will be ready.
2. Necessary test data will be made available.
3. The application has been developed according to specifications.

# 10. Test Case Template

* Description/Narrative:
  + Brief explanation what functionality is being tested (Whenever applicable)
* Preconditions:
  + Noted every precondition for the successful execution of the test case
* Test Steps:
  + Detailed steps to execute the test case
* Expected Result

The issue type that will be used in Jira is: XRay – Test. The preconditions are a different issue type that will be linked to the test case.

Every test case will be linked to a Test Set, grouped by functionality being tested.

Every test set will be linked to a Test Plan.

# 11. Bug Template

* Description/Narrative (Whenever applicable)
* Preconditions
* Steps to Reproduce
* Expected Result
* Actual Result
* Screenshot/Screen recording (Whenever applicable)

Bugs detected by an existing test case will be linked to the test case.

# 12. Priority Definitions

| **Priority value** | **Jira severity** | **Definition** |
| --- | --- | --- |
| Priority - 1 | Highest | Must be fixed immediately / in the next build. |
| Priority -2 | High | Must be fixed in any of the upcoming builds but should be included in the release. |
| Priority -3 | Low | May be fixed after the release / in the next release. |
| Priority -4 | Lowest | May or may not be fixed at all. |

# 13. Severity Definitions

| **Severity value** | **Jira severity** | **Definition** |
| --- | --- | --- |
| Severity - 1 | Critical | A condition with a high impact to multiple end users of the Website and there is no mutually agreed workaround implemented in a such a way that the issue can be adequately mitigated. |
| Severity-2 | High | A condition where a 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 | Medium | A condition where 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 | Low | A condition where a minor cosmetic Website or documentation error exists, and end user use of the Website is not impacted. |