



TWO MEN AND A WOMAN

TEST PLAN

Prepared by:
Daniela Shamatanova
Doychin Trendafilov
Yordan Nikolov



TABLE OF CONTENTS

Overview

1. Introductions

Objectives and Tasks

1. Objectives
2. Tasks
 - Planning and control
 - Analysis and design
 - Implementation and execution

Scope

1. Features To Be Tested
 - Public part features
 - Private part features (registered users only)
 - Newsfeed features for unregistered users
 - User connection feature
 - Administration feature
2. Features NOT To Be Tested
 - Public part features
 - Private part features (registered users only)

Testing Strategy

1. Testing types
 - Exploratory testing
 - Smoke testing
 - Functional testing
2. Testing techniques
 - Boundary Value Analysis
 - Equivalence partitioning
 - Classification trees
3. Entry Criteria
4. Exit Criteria

Environment

Responsibilities

Project Timeline

- Risks
- Mitigation Plan

Overview

1. Introductions

WEare is a social networking web application aimed at connecting professionals offering various services and exchanging experiences. This platform serves as a marketplace for skills and expertise, allowing users to network, collaborate, and engage in business activities. The application provides different functionalities tailored to various user roles:

Registered users:

- Connect with other users
- Create and publish posts
- Comment on posts
- Like posts from others
- Access a personalized feed featuring the newest or most relevant posts from their connections
- Manage their profile, including updating their name and uploading a profile picture
- Search for other people

Guest users:

- View a feed of the latest public posts
- Search for registered users

Admin users:

- Edit or delete user profiles
- Edit or delete posts
- Edit or delete comments

Objectives and Tasks

1. Objectives

The primary objective of testing is to ensure that the system fully meets its requirements, both functional and non-functional. The aim is to validate that the quality metrics for each requirement are satisfied, the use case scenarios are well-supported, and the overall product quality is maintained. By the end of the development cycle, the user should find that the project has met or exceeded all expectations as outlined in the requirements.

The secondary objectives involve identifying and exposing all issues and associated risks. This includes communicating known issues to the project team and ensuring that they are addressed appropriately before release. Achieving this objective requires thorough and methodical testing, with prompt and effective resolution of any discovered issues (bugs).

2. Tasks

The following tasks are to be completed by the entire QA team:

Planning and control:

- Conduct a detailed analysis of the Social Network Project Specifications and requirements.
- Distribute roles and responsibilities among team members.
- Establish channels of communication and schedule regular meetings.
- Define the scope of testing, including modules and features to be tested.
- Set entry and exit criteria for the entire project.

Analysis and design:

- Prioritize features and functionalities for testing based on their importance and impact.
- Select appropriate testing techniques and methodologies.
- Estimate the timeframe required for testing individual features.
- Allocate time estimates for different testing phases, such as exploratory testing, manual testing, and automated testing.

Implementation and execution:

- Develop detailed test cases based on the prioritized features.
- Conduct peer reviews of test cases and execute them.
- Identify test cases suitable for automation.
- Develop automated tests for both API and UI components.
- Execute automated tests and review results.
- Prepare comprehensive test reports and bug reports.

Scope

The primary focus of the QA team is to test the essential functionalities as outlined in the DEV team's final project document. Additional features developed by the DEV team will be tested as time permits.

1. Features To Be Tested:

Public part features:

- Registration form;
- Login form;
- Profile search form – searching users by first name, last name, ;
- Public feed displaying chronologically ordered public posts;

Private part features (registered users only):

- Change name;
- Change age;
- Change birthday;
- Upload profile picture;
- Set picture visibility (public or private);
- Change email;
- Update professional information;
- Modify services offered;

Newsfeed features for unregistered users:

- Viewing only public posts;

Newsfeed features for registered users:

- Create posts only contain text;
- Edit posts;
- Set post visibility;
- Like posts;
- Unlike posts (if already liked);
- Add comments to posts only contain text;
- Edit comments;
- Delete comments;
- Like comments;
- Dislike comments (if already liked);

User connection feature:

- View other users' profiles;
- Send and approve connection requests;
- Decline connection requests;
- Disconnect from other users;
- Disconnect from other users (No approval needed from the connected user);

Administration feature:

- Edit other users' posts;
- Delete other users' posts;
- Enable user accounts;
- Disable user accounts;
- Edit other users' comments;
- Delete profiles;
- Delete posts;
- Delete comments;

2. Features NOT To Be Tested:

Public part features:

- Registration form – email verification;
- Registration form – identify verification;
- Profile search form – searching users by email;

Private part features (registered users only):

- Set a song when creating a post;
- Set a video when creating a post;
- Set a location when creating a post;

Testing Strategy

1. Testing types

Exploratory testing:

Exploratory testing is an approach where testers leverage their skills and experience to assess the software. This method aims to identify functional and technical issues while optimizing the software. It is particularly useful for quickly learning about the Social Network and providing rapid feedback. This type of testing is useful for creating scenarios as we need to learn quickly about the Social Network and provide rapid feedback.

Smoke testing:

Also known as build verification testing, smoke testing focuses on the main functionalities of the software to ensure they work as intended before more detailed testing begins. This phase will involve both exploratory and manual tests on key features. The goal is to quickly identify severe failures that could halt further testing.

Functional testing:

Functional testing aims to validate that the software behaves according to its specified requirements. During manual testing, decisions will be made regarding which test cases to automate. Automated tests will be used for regression and smoke tests to ensure that no regression issues affect key functionalities. This allows for rapid feedback on the Social Network's status after each test cycle.

2. Testing techniques:

Boundary Value Analysis:

This technique focuses on testing the boundary values of valid and invalid partitions. It aims to identify defects that are more likely to occur at the edges of these partitions. Boundary Value Analysis tests input values close to the boundaries, as they have a higher likelihood of errors.

Equivalence partitioning:

In this technique, input data is divided into partitions of valid and invalid values. All partitions should exhibit the same behavior. Test cases are designed to cover each partition at least once, ensuring that if one condition in a partition is true, the same should hold for other equivalent partitions.

Classification trees:

Classification trees are used to create a set of if-then logical conditions for accurate prediction or classification of cases. This technique is particularly useful when the dataset needs to be divided into classes corresponding to the response variable, often binary classes like Yes or No.

3. Entry Criteria

Entry criteria serve as the prerequisites that must be met before initiating a specific task. The following conditions must be satisfied to officially commence testing:

- The software has been delivered.
- Test data for various test cases is prepared and available.
- Requirements have been clearly outlined and approved.
- High-level test cases have been developed and are prepared for execution.
- The test environment is configured, and all essential resources, including tools and devices, are in place and available to use by the QA team.

4. Exit Criteria

Exit criteria are the conditions that must be met to officially conclude a specific task. Meeting these criteria signifies the completion of the QA team's work:

- All high-priority functionalities have a 100% pass rate for both automated and manual test cases.
- At least 70% of test cases for medium-priority functionalities pass successfully in both automated and manual testing.
- The allocated time for testing has been exhausted.

Priority Scale and definition		
1	Highest	They are blockers. Critical service issues affecting all end users. Service unavailable or unusable with no workaround.
2	High	The service issue is critically impacting a significant proportion of end users or critically impacting collaboration among end users.
3	Medium	The service issue affects an individual, a small number of employees, a non-critical function or a workaround is easily available.
4	Low / Lowest	The service issue affects an individual or small group of Customer who requests a change to the functional specification. Product improvements that will not affect the software usage in any way and can be postponed, if there are tight deadlines.

Severity Scale and definition		
1	Blocking	Users are not able to perform a task or complete a scenario with the design as implemented. Important features are missing, broken, or behaving in a way that users won't be able to understand or remedy. Reasonable user action results in data loss. Spelling mistakes in main functionalities or names.
2	High	This issue may lead a customer to unknowingly put their site or system into an undesirable state. Any issue that erroneously reports system status. Significant interaction, layout, or other visual defects that most users will notice, and will slow or block users from completing important tasks. Clear negative impact to the perception of quality.
3	Medium	Minor loss of function or other issue where an easy workaround exists. Moderate interaction, layout, or visual defects that do not block or slow a user from completing a desired task, but are noticeable.
4	Low	Minor usability and formatting issues. Layout or visual defects that are likely to go unnoticed by users.

Environment

Given the absence of specific requirements for the test environment, the team has opted to conduct all testing under the following conditions:

- **Web Browser:** Latest version of Google Chrome (currently 117.0.5938.132 Official Build, 64-bit).
- **Operating System:** Windows 10, 64-bit.
- **Processor:** Intel® Core™ i7-6800K, clocked at 3.40GHz.
- **RAM:** 32GB.
- **OS Details:** Windows 10 Home, Version 22H2 (OS Build 19045.3448).
- **MariaDB:** Server 11.3.0 Alpha.

Responsibilities:

The responsibilities for the testing process will be divided among the three members of the "Two Men and a Woman" team as follows:

Test plan:

The test plan has been collaboratively created by all members of the team, with each contributing to identify the best approaches, methods, and solutions.

High-Level Test Cases:

The primary functionalities for testing have been allocated among the team members as follows:

Functions	Responsibilities
Login Page <ul style="list-style-type: none">As a registered userAs a non-registered user	Daniela Shamatanova
Friends Requests	Daniela Shamatanova
Footer links and texts	Daniela Shamatanova
Header links and buttons	Daniela Shamatanova
Like posts and like comments <ul style="list-style-type: none">As a registered user	Daniela Shamatanova
Search form <ul style="list-style-type: none">As a registered userAs a non-registered user	Daniela Shamatanova
Register Page <ul style="list-style-type: none">As a registered userAs a non-registered user	Doychin Trendafilov
Comments <ul style="list-style-type: none">As a registered userAs a non-registered user	Doychin Trendafilov
The feed of the newest posts <ul style="list-style-type: none">As a registered userAs a non-registered user	Doychin Trendafilov
Posts <ul style="list-style-type: none">As a registered userAs a non-registered user	Yordan Nikolov
Update Profile	Yordan Nikolov
Administrative part	Yordan Nikolov

Set up:

Task	Delegated to
Creation of Jira Xray Project	Yordan Nikolov – creator Doychin Trendafilov - review and edit Daniela Shamatanova – review and edit
Creation of GitHub Repository	Doychin Trendafilov – creator Yordan Nikolov - review and edit Daniela Shamatanova – review and edit
Database Deployment and Hosting	Doychin Trendafilov – creator Yordan Nikolov - review and edit Daniela Shamatanova – review and edit
Uploading application to Heroku	Doychin Trendafilov – creator Yordan Nikolov - review and edit Daniela Shamatanova – review and edit
Creation of Test Case template	Daniela Shamatanova – creator Yordan Nikolov - creator Doychin Trendafilov – creator
Creation of Bug report template	Daniela Shamatanova – creator Yordan Nikolov - creator Doychin Trendafilov – creator
Test Automation Framework	Yordan Nikolov – creator Doychin Trendafilov - creator Daniela Shamatanova – creator
Swagger API Documentation	Yordan Nikolov – creator Doychin Trendafilov - creator Daniela Shamatanova – creator

Resources:**Communication Channels:**

- Facebook private group
- Messenger chat
- Microsoft Teams
- Microsoft OneNote

Version Control:

- GitHub

Task Management Board:

- Jira

Manual Tools:

- Jira
- Microsoft Excel

Automation Tools:

- Postman v10.18.9
- IntelliJ IDEA 2023.2 (Ultimate Edition) Build #IU-232.8660.185, built on July 26, 2023
- Selenium WebDriver

Project Timeline:

The total duration allocated for completing all activities:

Task	Date
Requirement documents review	14.09 – 15.09
Test plan and documentation	15.09 – 17.09
Initial Set Up	15.09 – 18.09
Exploratory Testing	19.09 – 20.09
Manual Test Cases	20.09 – 25.09
API Tests Automation	26.09 – 01.10
Re-Testing	02.10 – 05.10
Test Report	06.10 – 08.10
UI Tests Automation	09.10 – 15.10
Bug Report	16.10 – 18.10

Risks:

Strategies to mitigate these risks:

- Tight Testing Schedule and Incorrect Time Estimates: Underestimation of the time required for testing activities.
- Unexpected Unavailability of QA Team Members: Sudden absence or unavailability of key team members.
- Unexpected Complexities in Feature Testing: Encountering unanticipated challenges while testing specific features.

Mitigation Plan:

Strategies to mitigate these risks:

- Regular Daily Communications: Maintain constant communication among team members to quickly identify and address issues.
- Overtime Work: The test team is prepared to work extended hours to meet deadlines.
- Scope Adjustment: The scope of the test plan may be modified to accommodate unforeseen challenges.