



Software Engineering and Testing. BSC Year 2, 2024/2025

Assessment 2: Requirements Document

**Submitted by: Danyil Tymchuk (B00167321) &
Artem Surzhenko (B00163362)**

22/02/2025

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ordinary Degree in Computing in the Institute of Technology Blanchardstown, is entirely my own work except where otherwise stated.

Author: Danyil Tymchuk

Dated: 22/02/2025

Author: Artem Surzhenko

Dated: 22/02/2025

Table of Contents

.....

”QWERTY” Social Network

Client

1. Project Overview

“QWERTY” is a simple web-based social platform where users can share text posts and interact with others. The software lets users create accounts, write posts, comment on others' posts, and search for content they're interested in. The platform works through a web browser, making it easy to access from any device. The main parts of the system include basic login/signup, post creation, and commenting. It's designed to be straightforward and easy to use, perfect for small groups who want to share thoughts and connect with each other.

2. Document Revision

Rev. 1.0 date – initial version

3. Scope

This software will be designed and developed as a basic social media platform focused on text-based content sharing. It will be implemented as a web application accessible through standard web browsers (Chrome, Firefox, Safari) and will not include mobile app versions. The system will only handle text-based posts with a maximum length of 500 characters and basic user accounts with username/password authentication. A simple graphical interface will be provided with essential functionality exposed through a minimalist design, as agreed with the client.

The user authentication system will be limited to basic registration and login features, without support for third-party authentication services or advanced security measures. The commenting system will only allow text comments, with no support for media attachments or advanced formatting. The platform will operate with a basic database setup suitable for small-scale usage and will not include advanced data analytics or reporting features. As specified by the client, the focus

will be on core functionality rather than enhanced features or complex user interactions.

4. Walkthrough Scenarios

Regular User:

Users should be able to register for a new account using a username and password. Once logged in, users can create text posts that appear on their profile and the main feed. Users can view posts from other users on the main page and interact with them through comments. If users want to find specific content, they can use the search feature by entering keywords. At any point, users should be able to log out of their account. If they make a mistake in their post, they should be able to edit or delete it. Users can also view their own profile to see all their posts in one place.

Main actions:

- Create/login to account
- Write and post text content
- Comment on posts
- View their profile
- Edit/delete their posts
- Log out

Site Administrator:

The administrator should be able to access the basic admin features using special login credentials. They should be able to remove inappropriate posts or comments when necessary. The admin can view all posts on the platform and should be able to reset user passwords if needed. They should also have the ability to disable user accounts that violate platform rules. In case of system issues, the admin should be able to perform basic system resets.

Main actions:

- Monitor platform content
- Remove inappropriate posts/comments
- Disable problematic accounts
- Perform system maintenance

5. Software Requirements Analysis:

Functional Requirements:

End User Requirements

1. Account Management

- Users must be able to create an account with username and password
- Users must be able to log in and out of their account
- System should verify login credentials before granting access

2. Post Management

- Users must be able to create text posts up to 500 characters
- Users must be able to view their posts on their profile
- Users must be able to edit or delete their own posts
- System should display posts in chronological order

3. Comment System

- Users must be able to comment on any visible post
- Users must be able to view all comments on a post
- System should display comments in chronological order

Administrator Requirements

1. Content Management

- Admin must be able to remove inappropriate posts or comments
- Admin must be able to disable user accounts when necessary
- System should log all admin actions

2. User Management

- Admin must be able to reset user passwords
- Admin must be able to view all user accounts
- System should maintain user account status records

System Requirements

1. Data Management

- System must maintain user account data in persistent storage
- System must store all posts and comments in a database
- System must ensure data consistency during operations

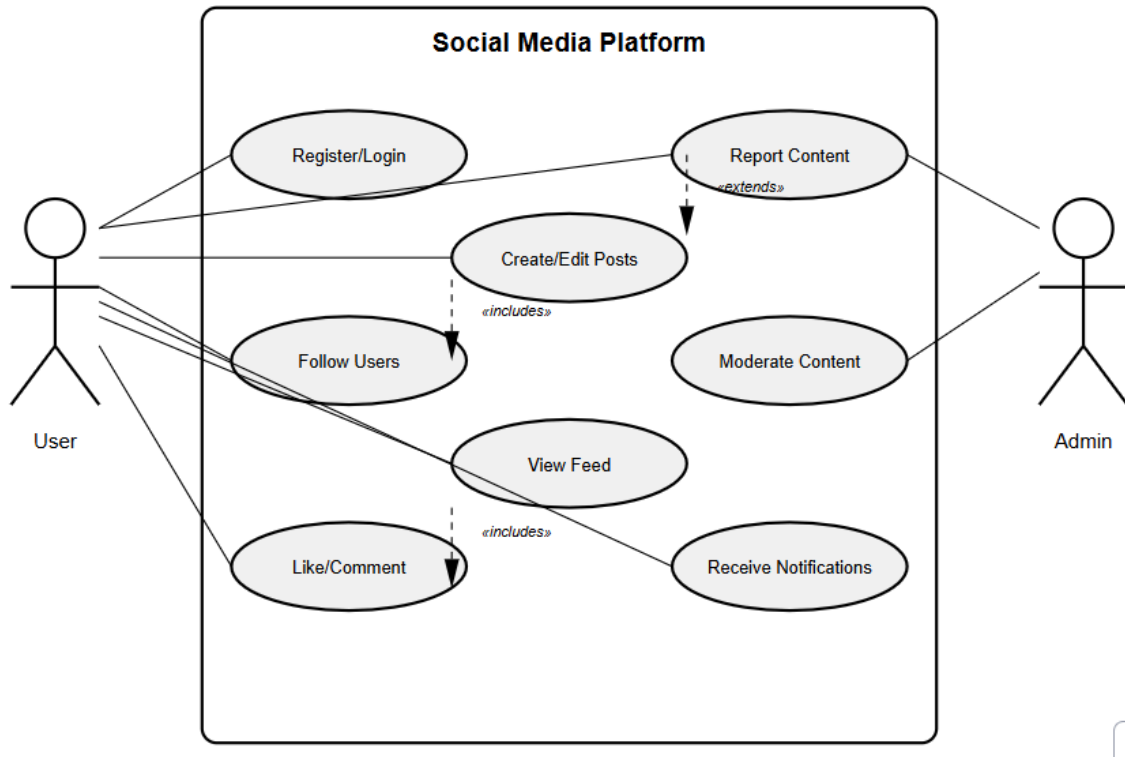
2. Authentication

- System must validate user credentials during login
- System must maintain session security
- System must handle multiple concurrent users

3. Error Handling

- System must display appropriate error messages
- System must prevent unauthorized access
- System must handle invalid inputs appropriately

Use case diagram



Use Case Specifications

Use Case: Create Post

Actor: Registered User **Description:** User creates a new text post **Flow:**

1. User selects "Create Post"
2. System displays post creation form
3. User enters post text
4. System validates input
5. System saves post
6. System displays success message

Use Case: User Authentication

Actor: User **Description:** User logs into account **Flow:**

1. User enters credentials
2. System validates input
3. System verifies credentials

4. System grants access
5. System redirects to home page

Undecided Features:

- Multi-language support
- Post categorization
- User profile customization
- Post sharing functionality

5.3 Non-functional Requirements:

Usability Requirements

- New users should be able to create an account in under 5 minutes
- Interface should be simple enough for non-technical users
- All major functions should be accessible within 2 clicks
- Error messages should be clear and understandable

Reliability Requirements

- System should handle common user errors without crashing
- Lost connections should not result in data loss

Security Requirements

- Passwords must be at least 8 characters long
- Database must be protected from basic SQL injection attacks
- User passwords must be stored in hashed format

Browser Compatibility

- Basic functionality must work on mobile browsers
- Text should be readable at all screen sizes

6. Graphical User Interface Design

The interface will be built using HTML, CSS, and basic JavaScript for a clean, minimalist design:

- Login/Register page with simple forms
- Main feed with scrollable post list
- Post creation page with text input
- Profile page showing user's posts

- Simple search bar in header

7. Technical Requirements and Feasibility:

System Models

UML Diagrams:

- Use Case Diagram (already provided)
- Basic Class Diagram:
 - User Class
 - Post Class
 - Comment Class

Technology Stack

- Frontend: HTML5, CSS3, JavaScript
- Backend: PHP (simple and widely supported)
- Database: MySQL (free, reliable, easy to set up)

Infrastructure Requirements

- Web hosting with PHP support
- MySQL database

API Requirements

- No external APIs needed for core functionality
- Simple REST endpoints for:
 - User authentication
 - Post management
 - Comment system

8. Conclusion

The proposed social media platform is feasible to implement with basic web technologies. The simplified scope, focusing on core features like text posts and comments, makes it achievable within a student project timeframe. The chosen technology stack (PHP/MySQL) is well-documented and has low setup requirements, making it suitable for development and testing.

We recommend proceeding with the project as specified, with potential for future enhancements after core functionality is established. The system's basic requirements can be met with standard web technologies and minimal infrastructure costs.

GitHub repo: <https://github.com/DanyilT/WebDev-Project>

Checklist: Is your document complete and correct?

Content:

- Do the requirements state the customers' needs
- Are you satisfied with all parts of the document
- Do you believe all parts are possible to implement
- Is each part of the document in agreement with all other parts
- Do the requirements avoid specifying a solution
- Do the requirements avoid specifying a design

Completeness:

- Are all the necessary interfaces specified – this includes input and output
- Are the specifications precise enough
- Are all sections from the document template included – if changed, why?

Clarity:

- Are all requirements reasonable?
- Is the level of details for each requirements appropriate?
- Are the requirements written in a language appropriate to the reader?
- Are all items clear and unambiguous?