GroupE4 Requirements Specification

BuzzNet

Group E4

Group Members:

LI Chun Leung 155193164
PENG Minqi 1155191548
WONG Kwok Kam 1155192018
ZENG Bai Chuan 1155193167
ZHANG Ka Sing 1155194769

Department of Computer Science and Engineering

Version 1.6

11 May 2025

Document Revision History

Version	Revised by	Revision Date	Comments
0.1	WONG Kwok Kam	4 Feb 2025	Initial draft.
1.0	LI Chun Leung PENG Minqi WONG Kwok Kam ZENG Bai Chuan ZHANG Ka Sing	6 Feb 2025	Major updates to all sections after 1st general meeting. Especially on functional requirements, clarify the roles and their functions.
1.1	WONG Kwok Kam	7 Feb 2025	Commenting and modifying functional requirements and document layout after 2nd general meeting.
1.2	WONG Kwok Kam	9 Feb 2025	Major updates based on comments.
1.3	WONG Kwok Kam	27 Mar 2025	Major updates based on professor's suggestions.
1.4	PENG MINQI	2 April 2025	Some updates in requirements based on professor's suggestions. And updates on architecture design.
1.5	ZENG Bai Chuan	12 April 2025	Finalization
1.6	Wong Kwok Kam	11 May 2025	Finalization

1 Introduction

This Software Requirements Specification (SRS) document is intended to capture the complete software requirements for the Project ***BuzzNet***. This document is intended to capture the scope and requirements of the desired system to be approved by the client and to be used by the software development team as a way to understand what needs to be done, and to serve as a basis for software design, development, and testing.

In addition to system functionality, the SRS also describes the non-functional requirements, design constraints, and other factors necessary to provide a comprehensive view of the requirements for the software.

1.1 Overview

The ***BuzzNet*** web application is designed to provide a public social media platform for users to share text and image-based content. The system will support core social media functionalities including post creation and user interactions, including likes, comments and follow.

The system implements a role-based access control (RBAC) framework that distinguishes between standard user and administrator privileges. The architecture includes robust content moderation capabilities accessible to administrators. Standard users retain ownership-based permissions, allowing them to remove their own published content, including posts and associated comments.

The system aims to foster real-time public discourse while ensuring secure and moderated user interactions.

1.2 Definitions, acronyms, and abbreviations

Term, Acronym, or Abbreviation	Definition
UI	User Interface
UX	User Experience
2FA	Two-Factor Authentication
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart

1.3 References

The following is a list of all references used to produce this SRS.

Reference	Date	Published by	Source
Software Requirements Specification For (TMS)	January 2003	U.S. Agency for International Development	[Software Requirements Specification For (TMS)]

2 Assumptions

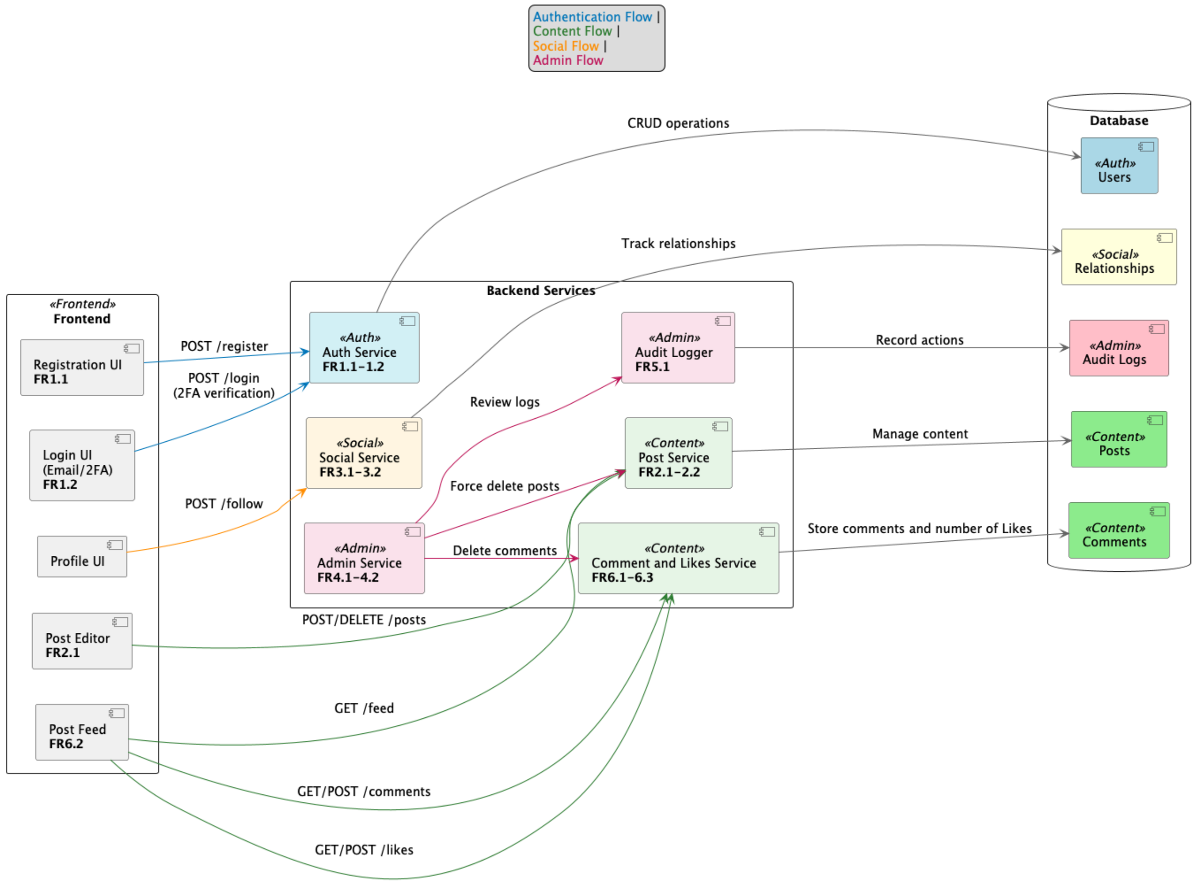
This document is based on the following assumptions.

Assumption No.	Title	Description
1	User Connectivity and Skilled Level	All users of this system are assumed to have the ability to connect to the Web application (via Internet or Intranet) and to have Web browsing capability.
2	Web Browser	Supported Web Browsers will be needed, including Chrome and Firefox.
3	Supported Operating Systems	Able to run on Linux, Windows, or Android released within 5 years.
4	Supported Language	Only English UI is supported in the early stage. But user can create posts in Chinese.
5	Runtime Environment	The runtime hardware, software, and networking environment will be supplied in fully operational condition by the client.
6	Screen Resolution	Optimal end user screen resolution will be 1920 * 1080.
7	Testing Prerequisite	Developers can directly take tests.
8	Change Request	The development of the system will proceed based on the client requirements. Any additional requirements and/or changes of existing requirements may result in changes to the development schedule and/or the project price. Change Requests must be submitted formally.
9	Resource Accessibility	The development group should keep the period within 2 months. Individuals in groups can take more than 20 hours per week. The project group can use the Linux server and other Cloud resources provided by the school.
10	Teamwork	Engineers in the group should be proficient in Git to make the project organized.
11	Report Technique	There is no specific reporting technology identified as standard. But the group should keep reporting regularly.

3 High-level System Architecture

The system will follow a client-server architecture with a clear separation between the frontend (client) and backend (server). Below is a high-level overview of the architecture:

3.1 Architecture Overview:



Ideal outcome *(Not all of the features will be implemented at early stage)*

3.2 System Flow:

3.2.1 User Interaction

The user interacts with the frontend by the web browser (e.g., writes a blog post, likes a post, etc).

3.2.2 API Request

The frontend sends an API request to the backend (e.g., retrieve post or comments, etc).

3.2.3 Backend Processing

The backend processes the request, performs necessary operations (e.g., database queries, etc), and returns a response.

3.2.4 Data Storage

The backend interacts with the database to store or retrieve data.

3.2.5 Response to User

The frontend updates the UI based on the backend’s response.

4 Functional Requirements

4.1 Functional Requirements

The following table summarises the functional requirements for the ***Buzznet.***

Req. No.	Title	Description
FR1.1	Registration	Users shall register using an username and password.
FR1.2	Login	Users shall log in with username/password.
FR2.1	Post Creation	Users shall create posts with text. All posts are considered non-anonymous and will display the username of the user who posted it.
FR2.2	Post Deletion	Users shall delete their own posts including all post comments.
FR3.1	Post Reaction	Users shall like and comment on posts.
FR4.1	Regulatory Control	Administrators shall delete any post.
FR4.2	Assessment	The system shall enforce role-based access control (user and admin).
FR5.1	Logging	The system shall log important actions for audit purposes.
FR6.1	User Comments Storage	The system shall allow users to submit comments on posts, which will be stored in the global database.
FR6.2	Display Comments	Users can view comments on posts in chronological order and the number of likes.
FR 7.1	User Profile	Users shall be able to see basic account information in the user profile.

4.2 Actors

An actor is anything that interacts with the data (e.g., viewing, modifying, etc). The following are identified actors of the software.

4.2.1 Standard User

Registered users can create content, interact with posts, and manage their account.

4.2.2 Administrator

Administrators can have privileges on top of standard users, such as, manage content moderation.

4.2.3 System

Automated processes (e.g., authentication, etc).

4.3 Use Cases

A use case defines a set of use-case instances, where each instance is a sequence of actions a system performs that yields an observable result of value to a particular actor. Following are the identified use cases.

4.3.01 Register Account for User

Initiator: Standard User

Description: User provides unique username and password to create an account.

Basic Flow of Events

User provides a username and password → System validates the input → If valid, the system creates the account.

Exceptional Flow of Events

E-1: Username inputed by the user is used. The user should use another username.

4.3.02 Register Account for Administrator

Initiator: Administrator

Description: Administrator creates the account in the account database directly.

Basic Flow of Events

Administrator provides a username, email, and password directly.

Exceptional Flow of Events

E-1: Username inputed by the user is used. The user should use another username.

4.3.03 Login

Initiator: Standard User, Administrator

Description: User / Administrator logs in with username and password.

Precondition:

The user has already finished the registration of the account and it is present in the database.

Basic Flow of Events

User inputs their username and password → System validates the credentials → System grants access to the user.

Exceptional Flow of Events

E-1: The credentials (username and password) of the user cannot be validated. Then the user cannot log in unless they provide the correct credentials.

4.3.04 Create Post

Initiator: Standard User, Administrator

Description: User / Administrator publishes text content.

Basic Flow of Events

User (logged in) inputs text in the post form→ User submits the post → System saves the post and displays it on the Feed View.

Exceptional Flow of Events

E-1: The user tries to submit an empty post. The system prompts the user to enter content.

4.3.05 Delete Post/Comment

Initiator: Standard User, Administrator

Description: User removes their own post/comment. Administrator removes inappropriate posts/comments.

Basic Flow of Events

For User (logged in): Navigates to their posts or comments → User selects the post/comment they want to delete → User confirms the deletion → System removes the post (including the comment associated with the post) or comment from the database.

For Administrator (logged in): Navigates to an inappropriate post/comment → Selects the post/comment to be deleted → Confirms the deletion → System deletes the post/comment from the database.

Exceptional Flow of Events

E-1: The user attempts to delete a post/comment that does not belong to them. The system denies the action.

4.3.06 User Comments

Initiator: Standard User, Administrator

Description: User submits comments on posts.

Basic Flow of Events:

User clicks into a post → User enters a comment in the comment section → User submits the comment → System stores the comment in the database and displays it under the post.

Exceptional Flow of Events:
E-1: The user submits an empty comment. The system warns the user that the comment cannot be empty.

5 Non-functional Requirements

5.1 System Requirements

Req. No.	Title	Description
SYS01	Web based	The application should be web-based, it must support major decent browsers, including Firefox and Chrome.
SYS02	Languages	The application currently only supports English UI, but user can create posts in Chinese.

5.2 Performance Requirements

Req. No.	Title	Description
PER01	Response Time	The average response time for user interface requests should be less than 3 seconds. For complex queries, the response time should be less than 6 seconds.
PER02	Concurrent Users	The system should support at least 50 concurrent users without impacting performance. During peak times, the system should handle up to 200 concurrent requests.
PER03	Resource Usage	CPU usage should remain below 70% to ensure stable operation under high load. Memory usage should stay below 80% to prevent application crashes or performance degradation.

5.3 Security Requirements

Req. No.	Title	Description
SEC01	Authorization	Role-based access control (RBAC) must be enforced to ensure users can only access resources relevant to their roles. The system should support fine-grained permissions to control access at the data level.
SEC02	Compliance	The system must comply with relevant security standards and regulations (GDPR, HIPAA) to protect user data and privacy.
SEC03	Logging and Monitoring	Security-related events (e.g., login attempts, data access) should be logged and monitored in real-time.

5.4 Documentation Requirements

Req. No.	Title	Description
DOC01	Requirements Document	A document specifying the functional requirements and non-functional requirements.
DOC02	Design And Implementation Document	A document that outlines the architecture, components, and processes involved in the development of the project.
DOC03	Testing Document	Testing documentation encompasses a set of documents that outline the testing process, strategy, and results for a software application.
DOC04	User Manual	Document to help users understand and utilize the application features.