Software Requirements Specification

For

ConnectU

Version 1.0

Prepared by

Maha Shoaib Khan (CS-21011)
Rafay Baig (CS-21060)
Manahil Siddiqui (CS-21087)

Instructor: Sir Kashif Asrar

Date: 06/12/2023

Table Of Contents

1 Introduction

A description of the scope and an outline of all the contents of this SRS document are provided in this section. A list of terminology and acronyms, along with the purpose of this document is made available. Readers will obtain significant insights into the software requirements. This section serves as the foundation for the later comprehensive requirements presented in the Software Requirements Specification (SRS) document.

1.1 Document Purpose

The purpose of this document is to specify the software requirements for a social media application *ConnectU*. It provides a clear explanation of the system's development goals and mission, the technologies used in the development, the functional and non functional requirement of this project and more details. The document elucidate the objectives and characteristics of the system, as well as its interfaces, functionalities, operational limitations, and response to external factors. This SRS covers the entire *ConnectU* application capabilities, giving broad coverage to the details. Together with the front-end user interface components that describe the design elements and user interactions, it also describes the back-end logic and real-time communication capabilities.

1.2 Product Scope

ConnectU is a social media online platform where users are able to interact with one another, can add friends, exchange images and videos, leave comments, and more. The social media application described in this software specification document, has a wide variety of features and functionalities designed to enhance communication and collaboration in the digital communication space.

The project includes, a user-friendly front-end interface created using ReactJS and its different libraries along with HTML and CSS, and a well-integrated back-end built using ExpressJS framework and NodeJS. All the user's data is stored in the database i.e. Mongoose database. Main feaures of the project is the real-time sharing of comments, images, videos and other resources. The main objective of developing this application is to collaborate to the world of social media, developing interactive and user-friendly application, allowing people around the globe to connect with each other.

1.3 Intended Audience

This Software Requirements Specification (SRS) document is intended for the individuals involved in the development of this project that includes the Development team, the Project Manager, the SQA team and the Supervisor along with the proof-reading and documentation team. Additionally, the document is designed to be visible to the professors who would be interested in the features and functions of the proposed system, as well as users of the application, or anyone between the ages of 18 to 50.

1.4 Document Overview

Following details are included in this document;

1. **System Overview:** Explains the front-end, back-end, and database, among other key components of the system, in a clear and comprehensive manner.
2. **System Features:** Provides information about the system's specific features, such as real-time commenting, images and videos uploading, reacting to others post etc.
3. **Non-Functional Requirements:** Addresses the concerns of system architects and developers by outlining performance and security requirements.
4. **System Constraints:** Explains restrictions and limitation within the project.

1.5 Definitions, Acronyms and Abbreviations

- **HTML:** Hyper Text Markup Language user to structure the website.
- **CSS:** Cascading Style Sheet use for desiginig purpose on the front-end.
- **SRS:** is an abbreviation for Software Requirements and Specification, which is a document that outlines the needs of a software system.
- **User:** an entity registered on the application database using the features of the app.
- **NodeJS:** is a cross-platform, open-source JavaScript run-time environment that runs JavaScript code outside a web browser.
- **ExpressJS:** is a back end web application framework for building RESTful APIs with NodeJS.
- **ReactJS:** is a free and open-source front-end JavaScript library for building user interfaces based on components.
- **JavaScript:** is a scripting language for creating dynamic web page content.
- **Mongoose:** provides a straight-forward, schema-based solution to model your application data
- **SQA:** Software Quality Assurance, is a means and practice of monitoring all software engineering processes, methods, and work products to ensure compliance against defined standards.

1.6 References and Acknowledgments

Not Applicable

2 Overall Description

2.1 Product Perspective

ConnectU aims to redefine online interactions, emphasizing user-centric design, personalized profiles, and interactive features. With a focus on seamless content sharing and responsiveness across devices, our goal is to establish a vibrant and engaging digital community for users to express themselves through text, images, and videos.

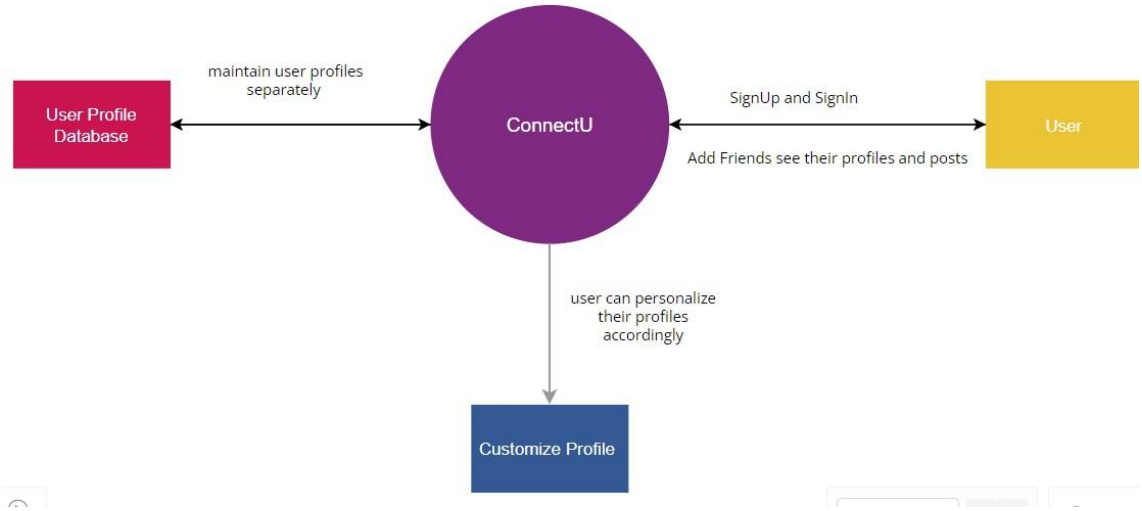

Figure 1: Context Diagram of ConnectU

2.2 Product Functionality

2.2.1 User Management

- Facilitate user registration with validation mechanisms.
- Implement authentication and login functionality.
- Manage user profiles, allowing users to update their details.

2.2.2 Social Interaction

- Efficient creation and display of posts enhance user engagement.
- Like and comment features incorporate real-time updates, fostering dynamic interaction.
- The friend management system ensures secure and swift addition or removal of connections.

2.2.3 Customization

- Integrate dark mode switching for a personalized and visually comfortable user experience.

2.2.4 Feed Display

- Present a curated feed on the home page, showcasing posts from users' friends.

2.3 Users and Characteristics

2.3.1 Target Audience

- General users seeking an intuitive and feature-rich social media platform.
- Users familiar with web and mobile applications, ensuring accessibility across various devices.

2.3.2 User Characteristics

- Users capable of managing profiles and navigating social media platforms.
- Individuals interested in real-time social interaction and dynamic content sharing.
- Users looking for personalized experiences through features like dark mode.

2.4 Operating Environment

ConnectU ensures universal accessibility, supporting all browsers and devices. Enjoy unrestricted access to the website from any location at any time. Connect seamlessly from your PC, laptop, or mobile device. No limitations, just a user-friendly experience. ConnectU empowers you to stay connected effortlessly.

2.5 Design and Implementation Constraints

2.5.1 Technology Stack

- Requires MongoDB for data storage.
- Utilizes the MERN stack (MongoDB, Express.js, React.js, Node.js).

2.5.2 Responsive Design

Design constraints imposed by the need for responsiveness across different screen sizes.

2.6 User Documentation

No specific expertise is needed to operate this system, as it is designed as a straightforward website. Basic familiarity with web browsers is sufficient; users should know how to access the website, register, and log in. The user interface is not only visually appealing but also intuitive and easy to navigate.

2.7 Assumptions and Dependencies

2.7.1 Assumptions

- Users have a basic understanding of web and mobile applications.
- MongoDB is configured securely and remains accessible for consistent data transactions.

2.7.2 Constraints

- An internet connection is required for real-time interaction and data synchronization.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The easy and visually appealing *ConnectU* user interface improves the experience of using the platform for the users. The primary elements of the user interface are as follows:

- **Login/Authentication** For authentication, users will be welcomed with a login screen. This screen will collect your credentials like *Login id* and *Password*. After entering correct credentials user will be redirected to the main feed.
- **Registration Page** For user who are new to the app would be given facility to register themselves first. Registration would involve collecting credentials from the new user like *Name*, *Login id*. *Password* etc. Once registered the user will then be redirected to the login page where entering correct credentials will lead him/her to the main feed.
- **Home Page:** After successful login we can enter our home page. On home page we have a very clean looking website, having a number of different widgets, information about the current user,option of creating new post, current user's previous posts and other user's posts. User can like or dislike the posts, can comment on them and can view previous comments. User can also add or remove friends from the same home page. User can also view other users profile clicking on their profile

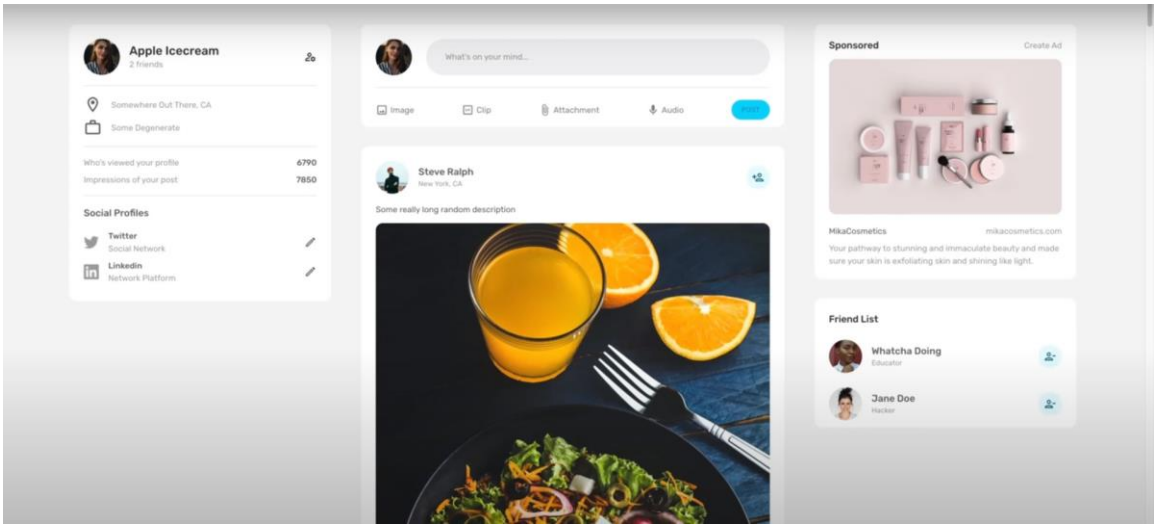

Figure 2: Home page sample of ConnectU

- **User Profile:** Clicking the profile of other person, redirects the user to tat person's profile where he/she can view their profile, their posts, their friends etc. Th user can also like or dislike that person's post from here.

3.1.2 Hardware Interfaces

The system's architecture prioritises accessibility across several devices, guaranteeing a consistent and seamless user experience regardless of the hardware platform.

- Computers, laptops, mobile phones, and other devices with a strong internet connection are all supported device types for the ConnectU app.
- An internet connection is necessary for uploading, viewing feeds and accessing other app functionalities.
- The system is compatible with a wide range of devices running different operating systems, including as Windows, macOS, Linux, iOS, and Android, and can be accessed by standard web browsers.
- The system mostly communicates with hardware to display outputs (such as images and messages) and receive user inputs (such as keyboard, mouse, and touch motions).

3.1.3 Software Interfaces

The applications communicates with the data base in order to retrieve user feeds and information to be displayed. The communication between the database and the mobile application consists of only reading operations.

3.1.4 Communication Interfaces

Given that each component of the system depends on the others, communication between them is crucial. The underlying operating systems for the web manage the communication because the method of accomplishing it is irrelevant to the system.

3.2 Functional Requirements

The functional needs of this application may be categorised into a number of functional domains. Each functional area represents a collection of related tasks or functions that the system needs to do.

- Users should be able to safely access the system by entering their correct login credentials.
- The entered credentials must be authenticated to confirm if the user is valid.
- The registration process involves the input of new users information that is to be stored to check when ever that user logs in.
- User can also set their profile pictures uploading the picture from the device they are using for the app.
- Users is able to create posts, add comments to the posts,like or dislike the post etc, and all of this is done in real-time.
- Creating post includes uploading images adding text etc.
- User can visit the profile of other person and view their posts and add comments etc.
- User can also add or remove friends for every post so that the post will only be visible to that person who is included in the friends list.

Collectively, these functional criteria delineate the roles and responsibilities that the ConnectU is expected to perform. This provides a thorough implementation roadmap for the required functionality to the development team.

3.3 Behaviour Requirements

3.3.1 Use Case View

The use case diagram that follows lists the main participants and the specific use cases that they are working on, and it illustrates the many features and interactions that make up ConnectU.

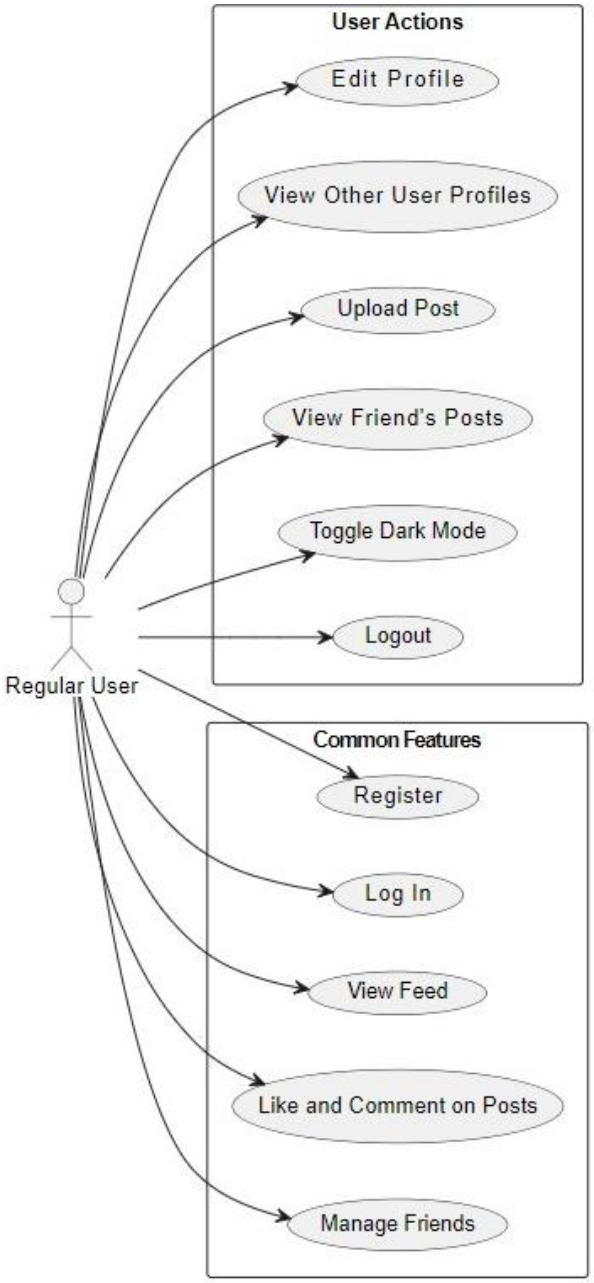

Figure3*: Use* Case *Diagram*

4 Other Non-Functional Requirements

4.1 Performance Requirements

- The application should load the home page and user profiles within 3 seconds.
- The backend server must handle a minimum of 1000 simultaneous user connections without significant performance degradation.
- Data retrieval and display of posts in the feed should occur within 2 seconds.

4.2 Safety And Security Requirements

- All user data, including passwords, must be encrypted during transmission and storage.
- The app should implement secure authentication practices, including session management and token-based validation.
- Measures should be in place to prevent common security threats, such as SQL injection and cross-site scripting.

4.3 Software Quality Attributes

4.3.1 Reliability

- In the event of a system failure, the procedure will be reversed, and the registration process will be halted to ensure the database remains free from any inaccurate information. Users will need to re-enter their information in such instances.

4.3.2 Usability

- The user interface should be intuitive, providing a seamless experience for users of all skill levels.

4.3.3 Flexibility

- The system should be flexible to accommodate future enhancements and technological changes.

4.3.4 Scalability

- Ability to handle increased user loads in the future.

5 Other Requirements

Not Applicable