Software Requirements Specification (SRS)

Project: Blog Submission and Notification System

1. Introduction

This Software Requirements Specification (SRS) outlines the functional and non-functional requirements for a Blog Submission and Notification System. The system allows users to log in, submit blog posts, and receive notifications via email and site alerts.

2. Overall Description

The system consists of the following components:

- Web Browser: Interface for user interactions.

- Account Service: Handles authentication.

- Blog Service: Manages post submissions and notifications.

- Mail Service: Sends notifications to subscribers.

- Database: Stores user credentials, blog posts, and notifications.

Users must be authenticated to submit blog posts. Upon successful login, they can submit a blog post, which triggers email and in-site notifications.

3. Functional Requirements

FR1: The system shall authenticate users using stored credentials.

FR2: The system shall reject access for invalid credentials.

FR3: The system shall allow authenticated users to submit blog posts.

FR4: The system shall store blog posts in the database.

FR5: The system shall send email notifications to subscribers after a new post.

FR6: The system shall create in-site notifications after a new post.

FR7: The system shall notify users of successful post submission.

Software Requirements Specification (SRS)

Project: Blog Submission and Notification System

4. Non-Functional Requirements

NFR1: The system shall respond to user actions within 2 seconds.

NFR2: The system shall be available 99.9% of the time.

NFR3: All user credentials must be securely hashed and stored.

NFR4: Email delivery must be completed within 5 seconds of post submission.

NFR5: The system shall support up to 1000 concurrent users.

5. System Architecture

The system follows a client-server architecture:

- Client: Web browser used by end-users.

- Backend: Microservices including Account Service, Blog Service, and Mail Service.

- Storage: Centralized database storing user, post, and notification data.

The services interact through RESTful APIs to maintain modularity and scalability.

6. Sequence Flow

1. User enters credentials.

2. Account Service queries the database.

3. If credentials are valid, login is confirmed.

4. User submits a blog post.

5. Blog Service stores the post and triggers:

   - Email notification to subscribers.

   - In-site notification storage.

6. User receives success confirmation.

Software Requirements Specification (SRS)

Project: Blog Submission and Notification System

7. Assumptions and Constraints

- Users must be registered before they can log in.

- Network connectivity is stable and secure.

- Mail Service has access to email addresses of subscribers.

- The system assumes subscribers have opted-in for email notifications.

8. Appendix

Actors:

- Web Browser (Client)

- Account Service (Auth API)

- Blog Service (Post API)

- Mail Service (SMTP or Email API)

- Database (SQL or NoSQL storage)