

# **Software Requirements Specification (SRS)**

## **Blog & Content Publishing Platform**

---

### **1. Introduction**

#### **1.1 Purpose**

The purpose of this Software Requirements Specification (SRS) document is to define the functional and non-functional requirements of the **Blog & Content Publishing Platform**. This document serves as an academic and technical reference for system understanding, design validation, implementation guidance, and evaluation during project development.

#### **1.2 Scope**

The Blog & Content Publishing Platform is a web-based application designed to enable users to create, publish, manage, and interact with blog content in a structured and secure environment. The system supports multiple user roles, content lifecycle management, and user engagement features such as likes and comments. The platform emphasizes usability, data consistency, security, and scalability while maintaining a clean separation between presentation, business logic, and data storage.

#### **1.3 Definitions, Acronyms, and Abbreviations**

- **SRS** – Software Requirements Specification
  - **CMS** – Content Management System
  - **UI** – User Interface
  - **API** – Application Programming Interface
  - **DBMS** – Database Management System
  - **CRUD** – Create, Read, Update, Delete
-

## **2. Overall Description**

### **2.1 Product Perspective**

The Blog & Content Publishing Platform is a full-stack web application consisting of:

- A browser-based frontend interface
- A backend application handling business logic and APIs
- A relational database for persistent storage

The system follows a layered architecture where the user interface, application logic, and data access layers are logically separated to improve maintainability and scalability.

### **2.2 Product Functions**

Major functions of the system include:

- User registration and authentication
- Role-based access control (Admin and User)
- Blog creation, editing, publishing, and deletion
- Viewing and browsing blog posts
- Like and comment functionality
- Blog moderation and management by administrators
- Secure storage and retrieval of content data

### **2.3 User Classes and Characteristics**

- **Content Authors (Primary Users)**

Users who register, create blog posts, edit content, and interact with other blogs.

- **Administrators (Secondary Users)**

Users responsible for managing users, moderating content, and maintaining system integrity.

## 2.4 Operating Environment

- **Frontend:** Web Browser (Chrome, Firefox, Edge)
- **Backend:** Java (JDBC-based application)
- **Database:** MySQL / MariaDB (via XAMPP)
- **Web Technologies:** HTML, CSS, JavaScript
- **Server:** Apache / Tomcat

## 2.5 Constraints

- Only registered users can create or interact with blogs
- Blog content must be stored in the database
- Role-based restrictions must be enforced
- System must function without advanced frameworks if required
- Data consistency must be maintained

## 2.6 Assumptions

- Users have basic knowledge of web applications
  - Users provide valid and appropriate content
  - Database server is available during system operation
- 

## 3. Functional Requirements (FR)

### FR-01 — User Registration and Login

**Description:** Allows users to register and securely log into the system.

**Actors:** User, System

**Inputs:** Username, email, password

**Outputs:** Authenticated user session

**Priority:** High

**Acceptance Criteria:** Only registered users can access the system; invalid credentials are rejected.

## **FR-02 — Role-Based Access Control**

**Description:** Differentiates access permissions for Admin and User roles.

**Actors:** System

**Priority:** High

**Acceptance Criteria:** Admin and User functionalities are strictly separated.

## **FR-03 — Blog Creation**

**Description:** Enables users to create and publish blog posts.

**Actors:** User

**Inputs:** Blog title, content

**Outputs:** Published blog stored in database

**Priority:** High

**Acceptance Criteria:** Blog is saved and visible to other users.

## **FR-04 — Blog Editing and Deletion**

**Description:** Allows users to edit or delete their own blogs.

**Actors:** User

**Priority:** Medium

**Acceptance Criteria:** Only the blog owner or admin can modify or delete content.

## **FR-05 — Blog Viewing and Browsing**

**Description:** Allows users to view all published blogs.

**Actors:** User, Visitor

**Priority:** High

**Acceptance Criteria:** Blogs are displayed in a readable format.

## **FR-06 — Like Functionality**

**Description:** Enables users to like blog posts.

**Actors:** User

**Outputs:** Updated like count

**Priority:** Medium

**Acceptance Criteria:** A user can like a blog only once.

### **FR-07 — Comment Functionality**

**Description:** Allows users to comment on blogs.

**Actors:** User

**Inputs:** Comment text

**Priority:** Medium

**Acceptance Criteria:** Comments are stored and displayed under the blog.

### **FR-08 — Blog Moderation**

**Description:** Allows admins to remove inappropriate content.

**Actors:** Admin

**Priority:** High

**Acceptance Criteria:** Admin actions are reflected immediately.

---

## **4. Non-Functional Requirements (NFR)**

### **NFR-01 — Performance**

**Specification:** Blog pages should load within 2 seconds under normal load.

### **NFR-02 — Security**

**Specification:** Passwords must be encrypted; SQL injection must be prevented using prepared statements.

### **NFR-03 — Reliability**

**Specification:** System must ensure no data loss during CRUD operations.

### **NFR-04 — Scalability**

**Specification:** System should support increasing number of users and blogs.

### **NFR-05 — Usability**

**Specification:** User interface should be simple and intuitive.

### **NFR-06 — Maintainability**

**Specification:** Code should be modular and easy to enhance.

---

## **5. External Interface Requirements**

### **5.1 User Interface**

- Web-based interface
- Forms for login, registration, blog creation
- Blog listing and detail views

### **5.2 Software Interfaces**

- JDBC for database connectivity
  - MySQL DBMS
-

## **6. Data Requirements**

### **6.1 Database Requirements**

- Users table
- Blogs table
- Likes table
- Comments table

### **6.2 Logging**

- User activity logging
  - Error logging
- 

## **7. System Architecture Overview**

The system follows a layered architecture:

- Presentation Layer (UI)
  - Application Layer (Business Logic)
  - Data Access Layer (JDBC)
  - Database Layer (MySQL)
- 

## **8. Constraints and Dependencies**

- Depends on database availability
  - Requires web server for deployment
-

## **9. Future Enhancements**

- Blog categories and tags
  - Search and filter functionality
  - User profile pages
  - Rich text editor
- 

## **10. Conclusion**

This Software Requirements Specification outlines the complete functional and non-functional requirements of the Blog & Content Publishing Platform. The system provides a structured, secure, and scalable solution for online content publishing and serves as a strong foundation for academic evaluation and future development.