SOFTWARE REQUIREMENTS SPECIFICATION

**For**

**Blog Platform**

**Prepared by:-**

*Tamilarasan.M*

*Thirumurugan.Rs*

*Yogesh kannan.V*

# Introduction

## Purpose

The main objective of this document is to illustrate the requirements of the project Blog platform. The document gives a detailed description of the both functional and non-functional requirements proposed by the client. The primary purpose of blogging platforms is to enable authors to publish content (whether articles, product reviews, or opinions) swiftly and simply through stand-alone websites, social networking sites, or email. The main purpose of this project is to maintain an easy circulation system using computers and to provide different reports. This project describes the hardware and software interface requirements using ER diagrams and UML diagrams.

## Document Conventions

* + - The entire document should be justified.
    - Convention for the Main title

Font face: Times New Roman Font Style: Bold

Font Size: 14

* + - Convention for Subtitle

Font face: Times New Roman Font Style: Bold

Font Size: 12

* + - Convention for body

Font face: Times New Roman Font Size: 12

## Scope of Development Project

A project scope is a document that outlines the objectives, deliverables, timelines, and resources required to complete a project. In the context of a blog platform, the scope of a development project would include defining the features and functionalities of the platform, such as the ability to create, manage, and publish content, user authentication, search functionality, and more, In a software development environment, project scope is a document that outlines the objectives and deliverables of a software project. Project scopes provide a roadmap for every stakeholder involved in a project to follow, providing a clear path to ensure the successful completion of the project. Defining project scope is a critical first step that will help your team avoid scope creep, reduce and mitigate risks and uncertainties, and make sure the project stays on track. In this tutorial, we explore what project scope is, why it is important to developers, and how to define project scope. We will also offer some tips on avoiding scope creep and best practices.

.

## Definitions, Acronyms and Abbreviations

JAVA -> platform independence SQL-> Structured query Language ER-> Entity Relationship

UML -> Unified Modeling Language

IDE-> Integrated Development Environment SRS-> Software Requirement Specification

## References

* + - Books

 Software Requirements and Specifications: A Lexicon of Practice, Principles, and Prejudices (ACM Press) by Michael Jackson

Software Requirements (Microsoft) Second EditionBy Karl E. Wiegers

Software Engineering: A Practitioner’s Approach Fifth Edition By Roger S. Pressman

* + - Websites

[**http://www.slideshare.net/**](http://www.slideshare.net/)

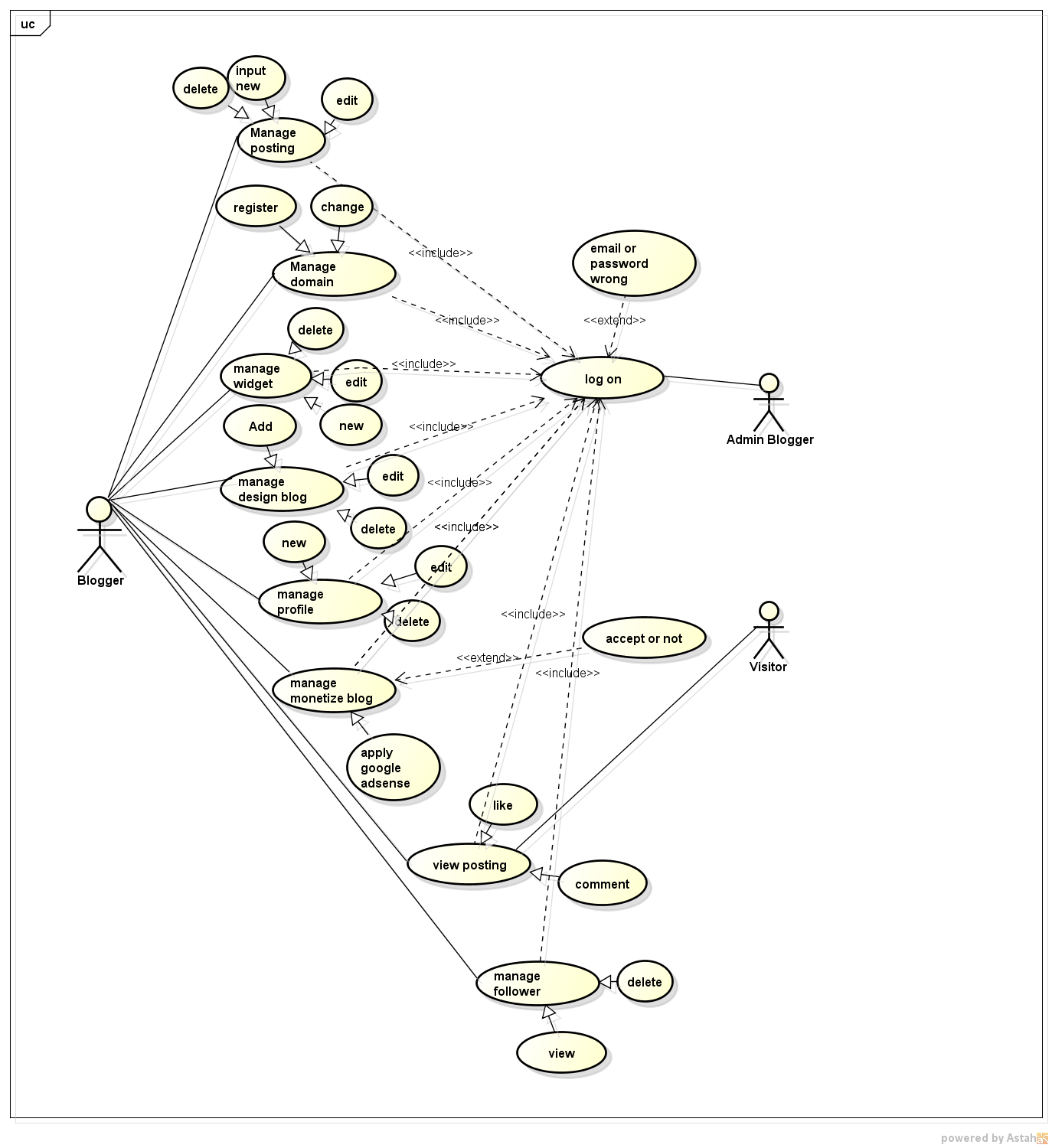
[**http://ebookily.net/doc/srs-library-management-system**](http://ebookily.net/doc/srs-library-management-system)

# Overall Descriptions

## Product Perspective

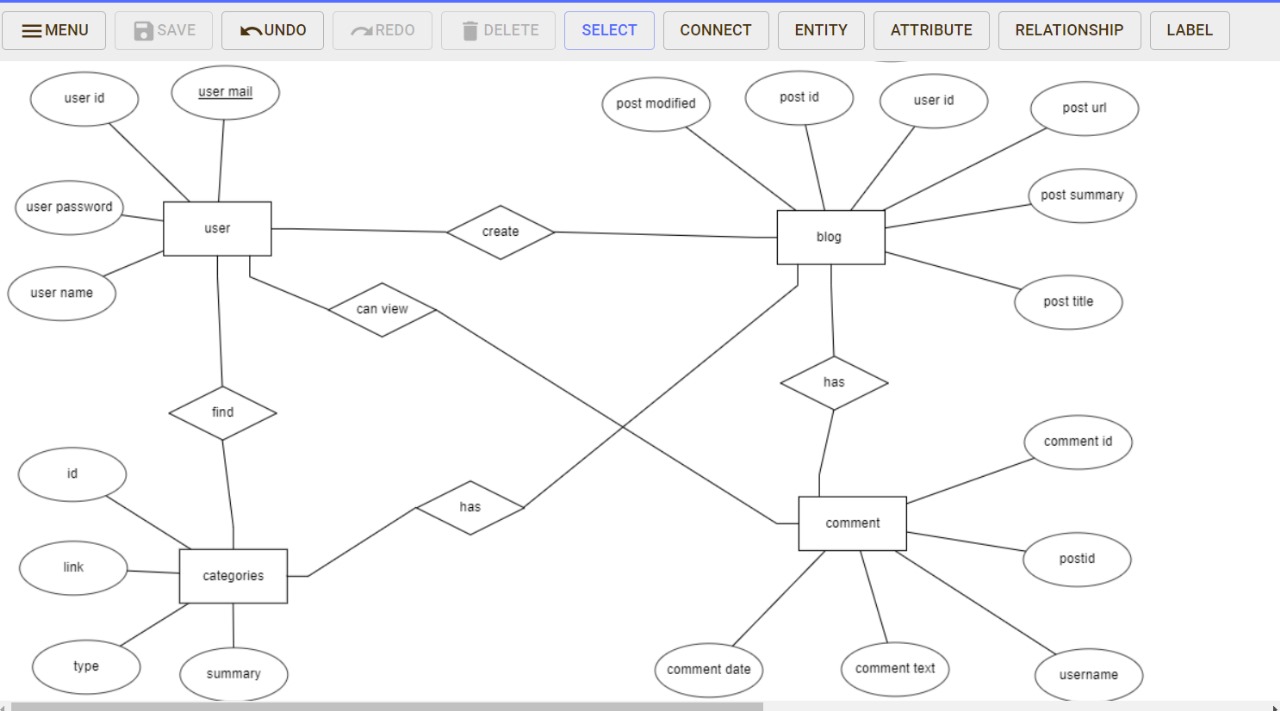
This is a broad-level diagram of the project showing a basic overview. The users can be either staff or students. This system will provide a search functionality to facilitate the search of resources. This search will be based on various categories viz. book name or the ISBN. Further, the library staff personnel can add/update the resources and the resource users from the system. The users of the system can request the issue/renewal/return of books for which they would have to follow certain criteria.

Use Case Diagram of BLOG PLATFORM



## Product Function

Entity Relationship Diagram of Bolg platform



The blog platform provides online real-time information about the posters available in the blog platform and the user information. The main purpose of this project is to reduce the manual work. This software is capable of managing blog Issues and generating various Reports for Record-Keeping according to end-user requirements. The member’s status of issue/return is maintained in the database. The member’s details can be fetched by the user from the database as and when required. The valid members are also allowed to view their account information.

## User Classes and Characteristics

The system provides different types of services based on the type of users [blog admin]

user\_id: Unique identifier for each user.

username: User's chosen display name.

email: User's email address.

password: Encrypted/hashed password for user authentication.

posts: List to store posts created by the user.

comments: List to store comments made by the user.

Additionally, the User class includes methods to create posts and add comments to existing posts, establishing relationships between users, posts, and comments within the blog platform. This is a basic representation and can be expanded upon to include more functionalities and user attributes as needed.

## Operating Environment

The product will be operating in a Windows environment. The Blog platform is a website and shall operate in all famous browsers, for a model we are taking Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox. Also, it will be compatible with the IE 6.0. Most of the features will be compatible with Mozilla Firefox & and Opera 7.0 or higher versions. The only requirement to use this online product would be an internet connection.

The hardware configuration includes Hard Disk: 30 GB, Monitor: 15” Color monitor, Keyboard: 122 keys. The basic input devices required are keyboard, and mouse and output devices are monitor, printer, etc.

## Assumptions and Dependencies

The assumptions are:-

* + - The coding should be error-free
    - The system should be user-friendly so that it is easy to use for the users
    - The information of all users and posters must be stored in a database that is accessible by the website
    - The system should have more storage capacity and provide fast access to the database
    - The system should provide a search facility and support quick transactions
    - The Blog platform is running 24 hours a day
    - Users may access from any computer that has Internet browsing capabilities and an

Internet connection

* + - Users must have their correct usernames and passwords to enter into their online accounts and do actions

The dependencies are:-

* + - The specific hardware and software due to which the product will be run
    - Based on listing requirements and specifications, the project will be developed and run
    - The end users (admin) should have a proper understanding of the product
    - The system should have the general report stored
    - The information of all the users must be stored in a database that is accessible by the blogger

## Requirement

Software Configuration:-

This software package is developed using Java as the front end which is supported by the sun micro system. Microsoft SQL Server as the back end to store the database.

Operating System: Windows NT, Windows 98, Windows XP Language: Java Runtime Environment, Net beans 7.0.1 (front end) Database: MS SQL Server (back end)

Hardware Configuration:- Processor: Pentium(R)Dual-core CPU Hard Disk: 20GB

RAM: 256 MB or more

## Data Requirement

The inputs consist of the query to the database and the output consists of the solutions for the query. The output also includes the user receiving the details of their accounts. In this project, the inputs will be the queries as fired by the users like creating an account, selecting a poster, and putting them into account. Now the output will be visible when the user requests the server to get details of their account in the form of time, date and which poster is currently in the account.

# External Interface Requirement

## User Authentication:

## Interface: OAuth, OpenID, or other secure authentication services.

## Content Delivery Networks (CDN):

## Interface: Integration with CDN providers for efficient media delivery.

## Social Media Integration:

## Interface: APIs for major platforms, facilitating content sharing and engagement.

## Payment Gateways:

## Interface: Integration with services like Stripe for secure transactions.

## SEO Tools:

## Interface: Integration with SEO tools for content optimization and discoverability.

## Analytics and Tracking:

## Interface: Integration with platforms like Google Analytics for user behavior insights.

## Email Services:

## Interface: Integration for reliable email notifications and communication.

## Commenting Systems:

## Interface: Integration with systems like Disqus for user-friendly commenting.

## Responsive Design:

## Interface: Compatibility across browsers and devices for a consistent user experience.

## Language Translation Services:

## Interface: Integration with translation APIs for multilingual support.

# System Features

User Registration and Authentication:

Secure account creation and login.

Profile Management:

Customizable user profiles.

Content Creation and Editing:

User-friendly post editor with multimedia support.

Content Management:

Categorization and tagging for organized content.

Commenting System:

Encourage user engagement with comments.

Social Media Integration:

Seamless content sharing on major platforms.

Search Functionality:

Robust search engine for efficient content discovery.

Subscription and Notification System:

User subscriptions and notifications for new content.

Responsive Design:

Accessible and visually appealing across devices.

User Interaction Analytics:

Basic analytics for post views and engagement.

# Other Non-functional Requirements

\Performance:

Requirement: The platform should load pages within 2 seconds to ensure a responsive user experience.

Scalability:

Requirement: The system should handle a growing user base and increased content without significant degradation in performance.

Reliability:

Requirement: The platform should have a 99.9% uptime, minimizing downtime for maintenance and updates.

Security:

Requirement: Implement encryption for user data, secure authentication mechanisms, and protection against common web vulnerabilities (e.g., SQL injection, cross-site scripting).

Data Backup and Recovery:

Requirement: Regularly back up user data and implement a reliable system for data recovery in case of unexpected incidents.

Compliance:

Requirement: Adhere to legal and regulatory requirements for user data protection, copyright, and accessibility standards.

Compatibility:

Requirement: Ensure compatibility with popular web browsers (e.g., Chrome, Firefox, Safari) and devices (desktops, tablets, mobile phones).

Usability and Accessibility:

Requirement: The platform should comply with accessibility standards (e.g., WCAG) to ensure usability for individuals with disabilities.

Maintainability:

Requirement: Develop clean and modular code to facilitate easy maintenance and updates.

Monitoring and Analytics:

Requirement: Implement tools for real-time monitoring of user interactions, system performance, and potential issues.

Backup Power and Redundancy:

Requirement: Have backup power sources and redundancy measures to prevent data loss in the event of power outages or hardware failures.

Data Privacy:

Requirement: Adhere to privacy regulations and ensure user data is handled with the utmost confidentiality.

Load Balancing:

Requirement: Distribute incoming traffic evenly across servers to prevent overload on any single server and maintain performance.

Cross-Browser Testing:

Requirement: Regularly conduct cross-browser testing to ensure consistent functionality and appearance across different web browsers.

Content Delivery Network (CDN) Integration:

Requirement: Integrate with a CDN to optimize content delivery and reduce latency for users globally.

Cost-Effective Hosting:

Requirement: Choose a hosting solution that balances performance with cost-effectiveness to ensure the platform's sustainability.

Internationalization:

Requirement: Support internationalization by allowing the platform to be adapted for different languages and regions.

Feedback Mechanism:

Requirement: Implement a feedback mechanism for users to report issues or provide suggestions, contributing to continuous improvement.

## Class Diagram

A class is an abstract, user-defined description of a type of data. It identifies the attributes of the data and the operations that can be performed on instances (i.e. objects) of the data. A class of data has a name, a set of attributes that describe its characteristics, and a set of operations that can be performed on the objects of that class. The classes’ structure and their relationships to each other frozen in time represent the static model. In this project, there are certain main classes

which are related to other classes required for their work. There are different kinds of relationships between the classes as shown in the diagram like normal association, aggregation, and generalization. The relationships are depicted using a role name and multiplicities. Here ‘Librarian’, ‘Member’, and ‘Books’ are the most important classes which are related to other classes.

