1.Introduction

1.1. Purpose

The purpose of this document is to outline the requirements for the development of a Blog Application and Website. This project aims to provide a user-friendly platform for creating, managing, and sharing blog content. The document will detail both functional and nonfunctional requirements as proposed by the client.

1.2. Document convention

- ➢ Entire document should be justified.
- ➢ Convention for main title
  Font face:times new roman
  Font style: bold
  Font Size:14
- ➢ Convention for Sub title
  Font face:times new roman
  Font style: bold
  Font Size:12
- ➢ Convention for Body
  Font face:times new roman
  Font Size:12

1.3 Scope of development project

The project's primary aim is to develop an interactive and user-friendly web-based application tailored for bloggers, empowering them to effortlessly create, edit, and publish their content. It also aspires to create an engaging and informative website that prominently showcases blog posts, ensuring intuitive navigation for all users. Our overarching goal is to deliver a seamless experience across various devices and browsers, prioritizing data security, privacy, and scalability to accommodate future growth. Furthermore, we are dedicated to constructing a flexible architecture that readily accommodates future feature enhancements and integrations, ensuring the platform remains adaptive and relevant.

In essence, this blog web application will provide bloggers and readers alike with a robust platform for content creation and sharing. It will enable users to register, log in, and personalize their profiles, while also simplifying the composition, editing, and scheduling of blog posts through an intuitive text editor. Robust content management features will encompass categorization, tagging, and streamlined media uploads via an integrated media library.

Interactive features will foster user engagement, with mechanisms for comments, likes, and sharing. The application's design will place a strong emphasis on optimizing user experience, with responsive design principles, SEO enhancements, and support for multiple languages.

Additionally, monetization avenues like advertisements and premium content will be readily available.

Security, privacy, and accessibility will remain paramount, with comprehensive safeguards in place, including user data protection, content moderation tools, and adherence to accessibility guidelines. An admin dashboard will provide valuable insights and moderation capabilities, while continuous improvement will be facilitated through CI/CD pipelines.

Looking ahead, the platform's potential for expansion includes the integration of features such as podcasts, video streaming, or e-commerce, ensuring its adaptability and versatility to meet evolving user needs and market trends.

1.4 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

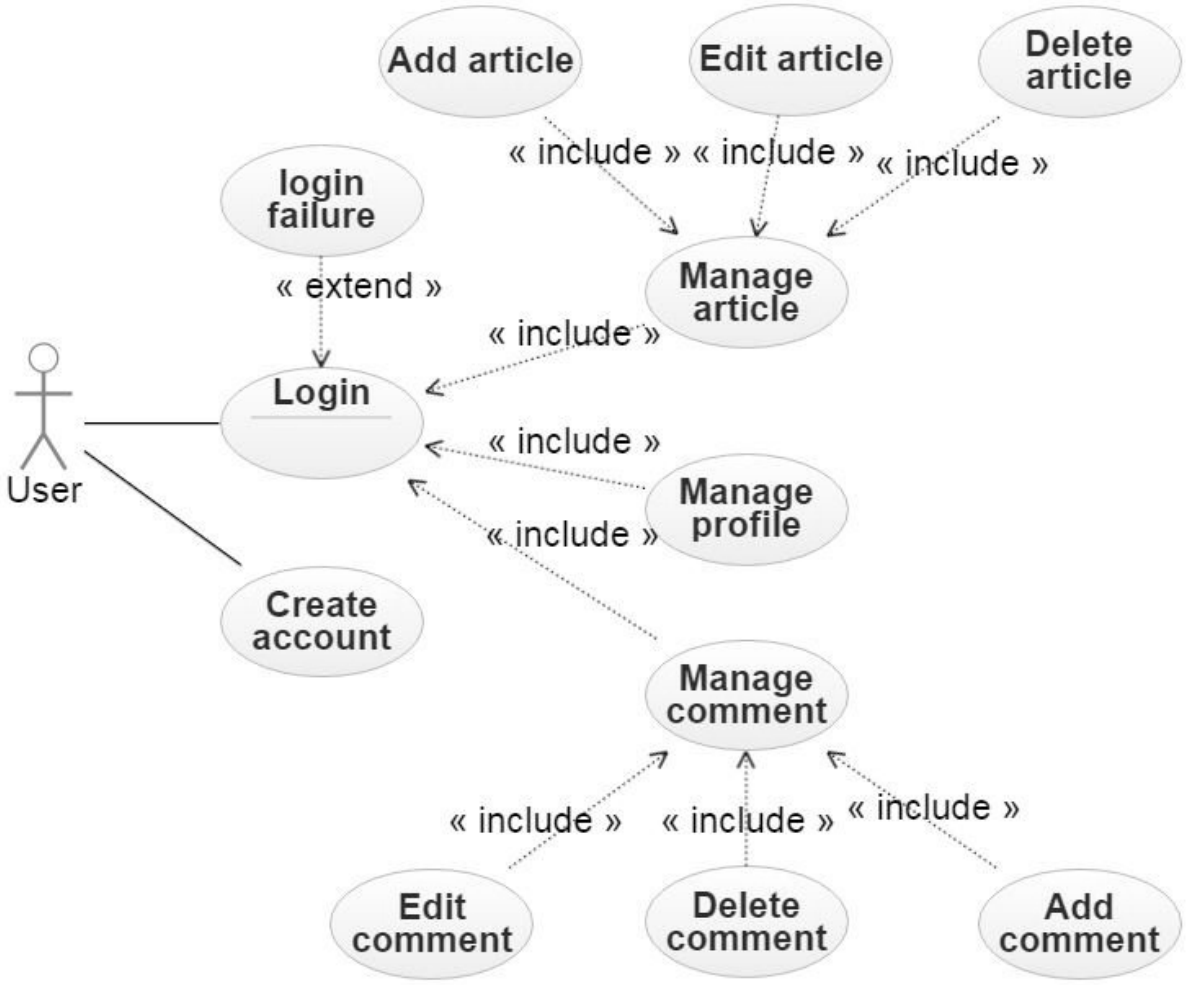
1.5 References

- 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://ebookily.net/doc/srs-library-management-system
  - Sample Blog SRS on GitHub
  - Blog Website Requirements Template on BlogTrek
  - Web Design and Development Guide by Smashing Magazine
  - W3Schools - Web Development Tutorials

2. Overall description

2.1 Product perspective

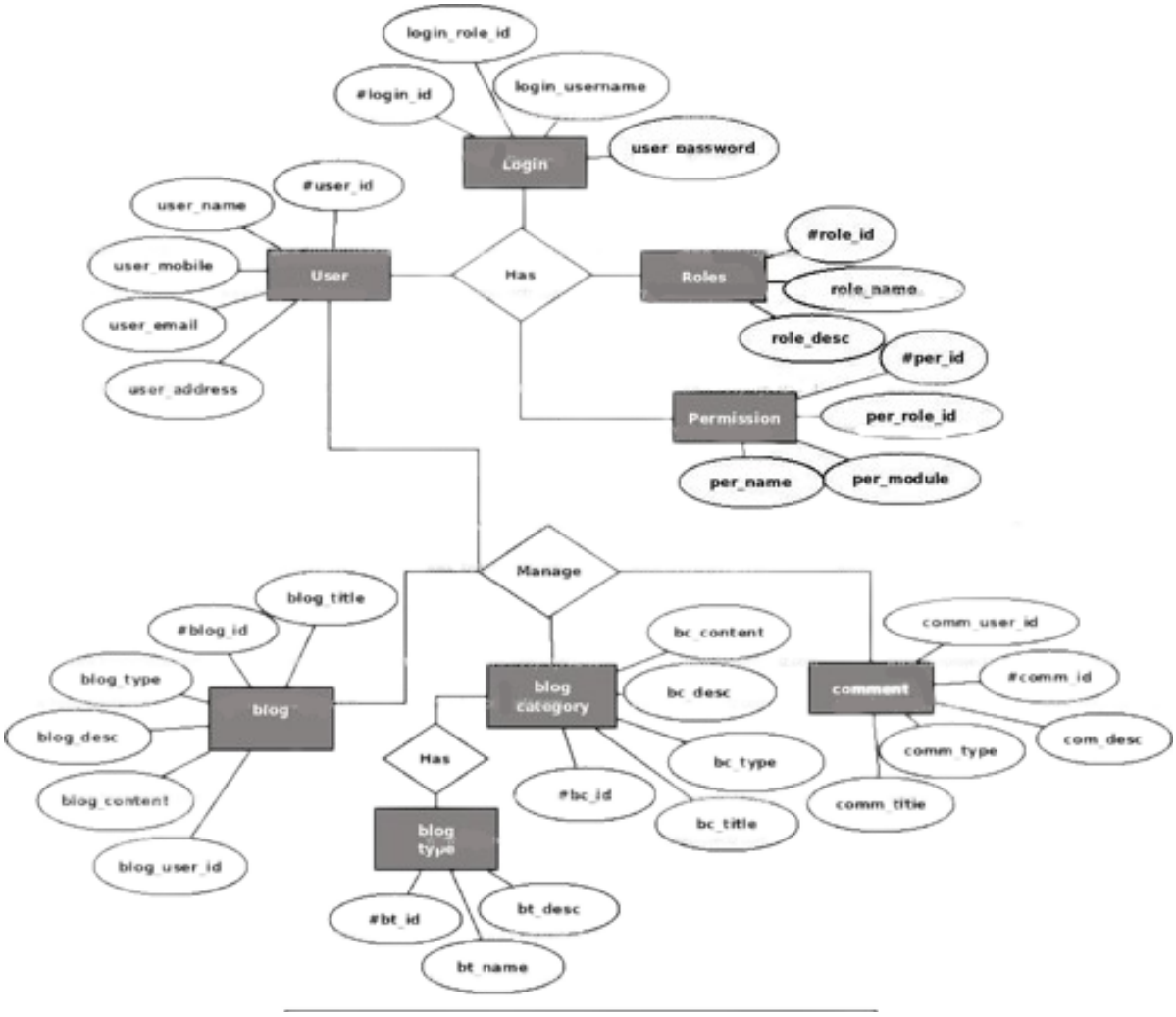
Use case diagram:



The use case diagram illustrates how users and administrators interact with the blog website/application, highlighting key actions and interactions within the system. Users can interact with multiple use cases, representing their engagement with different features of the platform like login,create account, add article, edit article etc.

2.2.Product Function

Entity relationship diagram of Blogging website and application



The blog website or application serves as a versatile platform where individuals and organizations can share a wide range of informative content on topics spanning from news and technology to travel and fashion. Its primary goal is to engage readers, encourage discussions, and build communities of like-minded individuals while also promoting personal branding and fostering credibility. Blogs are instrumental in content marketing, effectively promoting products and services while enhancing search engine visibility. Ultimately, a blog serves as a dynamic medium for sharing knowledge, experiences, and opinions with a global audience.

Introduction:

Our blog platform caters to various user roles, each with specific privileges and features. Understanding these user classes and their characteristics is crucial to making the most out of our platform.

2.3 User Classes and Characteristics

The blogging website/application offers distinct services and features based on user roles, primarily categorized as "Authors" and "Readers." Authors have elevated privileges, akin to

administrators, while Readers encompass a diverse user base that includes casual readers, subscribers, and commenters.

Author:

Authors are the content creators and hold a prominent role in the ecosystem. They can:

- Create and publish blog posts.
- Edit and update their published articles.
- Manage their blog profile and settings.
- Interact with readers through comments and responses.

Reader:

Readers encompass various user types, including casual readers, subscribers, and active participants in discussions. Their capabilities include:

- Browsing and reading published blog posts.
- Leaving comments and engaging in discussions.
- Saving favorite articles for later reading.
- Subscribing to authors for updates.
- Sharing posts on social media platforms.

This user classification ensures that both content creators (Authors) and content consumers (Readers) have tailored services and features to enhance their experience on the blogging website/application.

2.4 Operating Environment

The Blogging Website/Application is designed to operate in a Windows environment and is compatible with a wide range of modern web browsers, including but not limited to Microsoft Edge, Google Chrome, Mozilla Firefox, and Safari. While Internet Explorer 6.0 is not supported due to its outdated technology, most features are compatible with Mozilla Firefox, Google Chrome, and other modern browsers.

There is no specific hard disk size requirement as the application is web-based and does not rely on local storage. The basic input devices needed are a keyboard and a mouse, while the primary output device is a standard monitor. Additional output devices like printers are optional based on user preferences

2.5.Assumptions and Dependencies:

The Assumptions are:

- The development process assumes that coding will be meticulously done to ensure a reliable and error-free system.
- The application is designed to be user-friendly, prioritizing ease of use for all users.
- All user, post, and content information will be securely stored in a database accessible via the website.
- The system will provide ample storage capacity and ensure fast database access for efficient performance.

- The application will offer robust search functionality and support quick, seamless transactions.
- The Blogging Website/Application will be available 24 hours a day to accommodate users from different time zones.
- Users can access the platform from any computer with internet browsing capabilities and an internet connection.
- User access to their online accounts and actions will require correct usernames and passwords.

The dependencies are:

- The application's functionality depends on specific hardware and software configurations required for hosting and running the platform effectively.
- The development of the project relies on a clear listing of requirements and specifications, which will serve as the foundation for the development process.
- End users, including administrators, should have a solid understanding of the blogging website/application to effectively manage and utilize its features.
- The system is dependent on properly storing and managing general reports generated by the application.
- The integration and functionality of the system depend on maintaining accurate and accessible user and content information in a database.
- Any updates or changes related to content, such as blog posts, must be accurately recorded in the database to ensure data integrity and relevance.

2.6 Functional Requirements:

This highlights the key technologies and hardware recommendations for the Blogging Website/Application

Software Configuration:-

Front-end: HTML, CSS, JavaScript

Back-end: Python with Django framework

Database: PostgreSQL

Operating Systems: Windows 10, macOS, Linux

Development Environment: Visual Studio Code (IDE)

Web Server: Apache

Version Control: Git

Hardware Configuration:-

Processor: Modern multi-core processor (e.g., Intel Core i5, AMD Ryzen)

Hard Disk: Minimum 100GB storage

RAM: Minimum 8GB RAM

2.7 Data Requirement :

The blog website requires diverse data elements to function effectively. User data includes registration information, profile details, and preferences. Content data encompasses blog posts, categories, comments, and user interactions. Media files support multimedia content and user profile pictures. Analytics and tracking data monitor user activity, traffic, engagement, and performance. Security and authentication data ensure user account protection. Search data tracks queries and results. Notifications, emails, and configuration data enhance the user experience. Payment data may be needed for e-commerce. Legal compliance, backup, and error data maintain site integrity. Third-party integrations, content moderation, and localization elements expand functionality. This SRS establishes data management protocols to ensure security, compliance, scalability, and performance.

3. External Interface Requirement

3.1 GUI

The blog website's GUI is designed to provide an intuitive and user-friendly interface for both users and administrators. It encompasses the following features:

All modules within the software adhere to a standardized template, ensuring consistency in design and user experience across different sections of the website.

A part of the GUI is dedicated to user management, enabling administrators to create, update, and delete user accounts. It also includes login and logout functionalities.

Login Interface:

Users can register and create their accounts by providing necessary details.

To log in, users are required to enter their username and password. Incorrect entries trigger an error message.

Search Functionality:

The GUI includes a search feature that allows users to search for specific content, such as blog posts or topics, by entering keywords or titles.

Content Categories View:

The GUI provides a view of content categories or tags, enabling administrators to manage and organize blog content by adding, editing, or deleting categories.

4. System Features:

The users of the system should be provided the surety that their account is secure. This is possible by providing:

- Users will have their accounts secured through user authentication, requiring them to log in using their unique credentials (e.g., username and password).

- Proper accountability which includes not allowing a member to see another member's account.

5. Other non functional requirements:

5.1 Performance Requirements

- The blogging website/application will serve as the primary platform for interactions among users (authors and readers). Therefore, it should deliver fast and accurate performance.
- It should effectively handle both expected and unexpected errors, preventing information loss and minimizing downtime.
- The system must be capable of handling a large amount of data, accommodating a high number of blog posts, authors, and readers without errors or performance issues.

5.2 Safety Requirement

- Given the potential for database crashes due to unforeseen events like viruses or operating system failures, regular database backups will be implemented to prevent data loss.
- The system should be equipped with an uninterruptible power supply (UPS) or inverter to maintain operation in the event of power supply failures.

5.3. Security Requirement

- The system will utilize a secure database.
- User roles and access constraints will be implemented, allowing normal users to read information while preventing unauthorized modifications.
- Robust user authentication mechanisms will be in place to secure user accounts.
- Security measures will be implemented to prevent password hacking.
- Distinct accounts for administrators and regular users will ensure that only admins have rights to update the database.
- Data backup and recovery processes should be established to prevent data loss in case of unforeseen events, such as hardware failures or data corruption. Regular backups should be performed, and an efficient recovery process should be in place to restore the system's functionality.

5.4. Requirement Attributes

- Multiple admins will have the ability to make changes to the system, emphasizing collaboration and flexibility.
- The project will be open source, promoting transparency and community involvement.
- User-friendliness will be a priority in maintaining the quality of the system.
- The installation process should be straightforward and user-friendly.

5.5. Business Rules:

Business rules will encompass policies and practices for system users, including cost considerations, discount offers, and adherence to legal rules and protocols. Violations of rules and regulations should be avoided by both administrators and members.

5.6. User Requirements:

- Users will include both members (readers and authors) and administrators (admins) of the system.
- Members are expected to have basic computer and internet browsing knowledge.
- The system will provide user-friendly interfaces, user manuals, online help, and installation guides to ensure ease of use and maintenance.
- Users of the system will include both members (readers and authors) who interact with the platform.
- Members, whether readers or authors, are assumed to possess basic computer skills and familiarity with internet browsing.
- The system is designed to be user-friendly, providing intuitive interfaces, user manuals, online help, and installation guides to facilitate ease of use and maintenance.

5.7.Accessibility

Accessibility standards (e.g., WCAG) should be followed to ensure that the platform is usable by individuals with disabilities. Features like alternative text for images, keyboard navigation, and screen reader compatibility should be implemented to enhance accessibility.

5.8. Mobile Responsiveness

Mobile responsiveness ensures that the website/application functions seamlessly on various devices, including smartphones and tablets. Responsive design principles should be employed to adapt the layout and content presentation to different screen sizes and orientations.

5.9. SEO Optimization

SEO optimization involves implementing strategies to improve the website's visibility on search engines like Google. This includes optimizing content with relevant keywords, using proper metadata, creating SEO-friendly URLs, and enhancing overall site structure.

6. Other Requirements

6.1 Data and Category Requirement

The platform will have different user categories, such as authors, readers.Access rights and permissions will vary based on the user's category.The system will organize content, such as blog posts, into different categories.Each content category will have relevant data and metadata associated with it.The platform will ensure that content is coded and presented in a structured

and consistent format, making it easy for users to navigate and discover content within specific categories.These requirements emphasize the organization of users and content categories within the blogging website/application, ensuring that access rights and data presentation align with user roles and preferences.

6.2. Appendix

A: Admin,B: Blog,Blogcatalog,C: CMS (Content Management System),D: Database,GUI: Graphical User Interface,K: Key,L: Login, Localization,M: Monetization,N: Non-functional Requirement,O: Operating Environment,P: Performance,R: Requirement,S:Scope, Security, System features; U: User, User class and characteristics, User requirement.

6.3. Glossary

The following are the list of conventions and acronyms used in this document and the project as well:

Admin: An administrator or admin is a user role with elevated privileges, responsible for managing the system and its users.

Author: A user role that represents content creators, who can create, edit, and publish blog posts on the platform.

GUI (Graphical User Interface): The visual interface that users interact with to navigate and use the blogging website/application.

WCAG (Web Content Accessibility Guidelines): A set of guidelines and standards for web accessibility to ensure websites are usable by individuals with disabilities.

SEO (Search Engine Optimization): Strategies and techniques used to improve a website's visibility on search engines like Google.

Responsive Design: Design principles that ensure a website/application adapts to different screen sizes and devices, providing a consistent user experience.

Entity Relationship Diagram (ERD): A visual representation of the relationships between different data entities within a system.

Use Case Diagram: A visual representation of how users and administrators interact with the system, highlighting key actions and interactions.

Security Requirement: Features and protocols implemented to protect user data and ensure system security.

User Interface (UI): The graphical elements and layout that users interact with when using a software application.

User Experience (UX): The overall experience and satisfaction of users when interacting with a website or application.

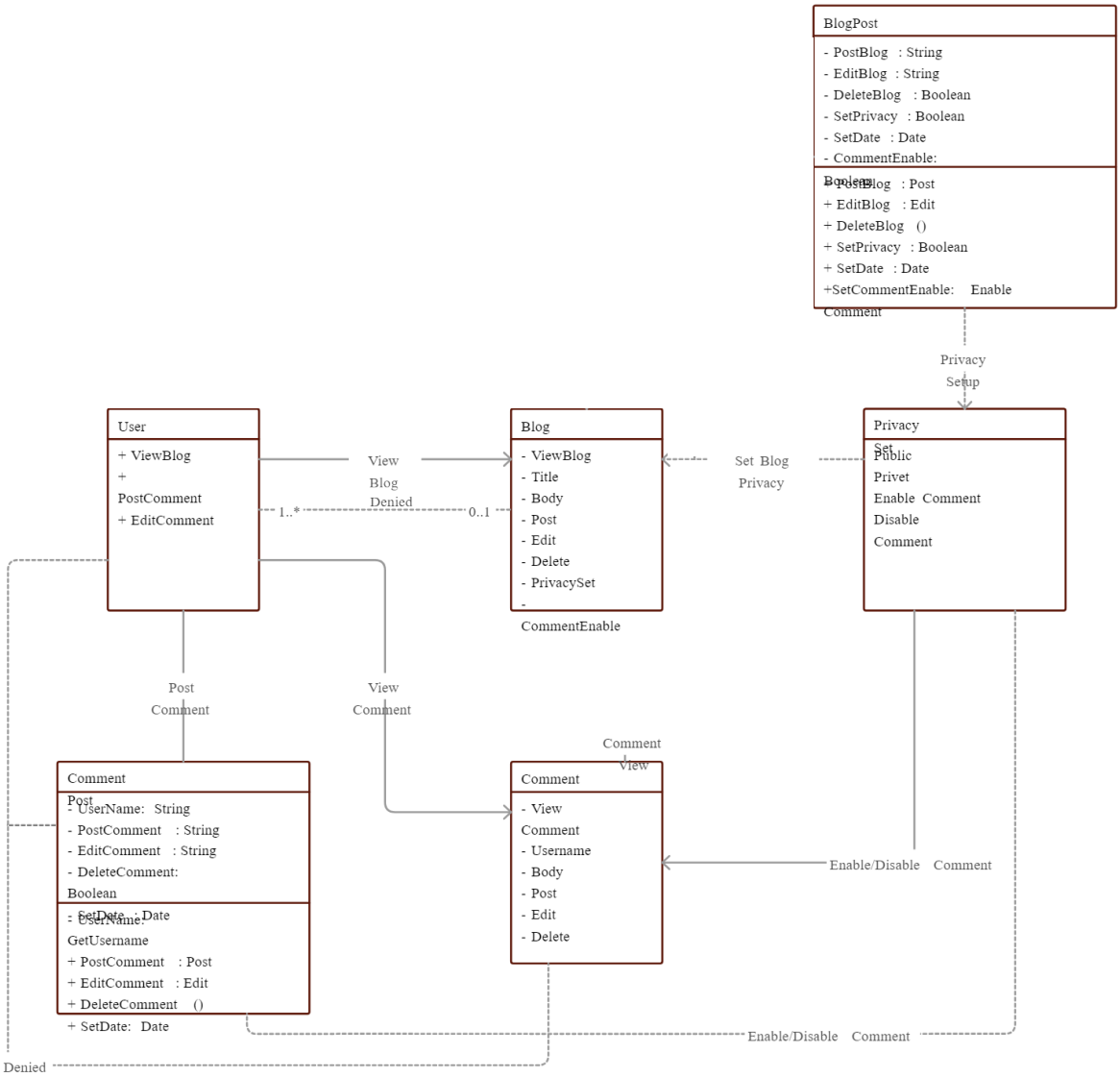
Web Server: Software that hosts and serves web content to users accessing the blogging website/application.

User Account: A user's personal account on the blogging website/application, containing their profile and settings.

Content Moderation: The process of reviewing and managing user-generated content to ensure it adheres to platform guidelines and policies.
Localization: Adapting the website/application to support multiple languages and regional settings.
Content Monetization: Strategies and features that allow users to generate revenue from their blog content, such as advertisements or premium subscriptions.
System Integrity: Ensuring the integrity and reliability of the system's operation and data.
System Scalability: The ability of the system to handle growth and increased user activity.
System Performance: The speed and efficiency of the system in responding to user requests and actions.
Accessibility Guidelines: Standards and guidelines for making web content accessible to individuals with disabilities.
Legal Compliance: Ensuring that the platform adheres to legal regulations and requirements.

6.4 Class Diagram

A class in the context of a blogging website/application represents a blueprint for a specific type of data or functionality within the system. Each class is defined by a name, a set of attributes that describe its properties, and a collection of methods that determine what actions can be performed with instances (objects) of that class. In this project, several key classes are essential for the system's operation, and they have various relationships with one another, including associations, aggregations, and generalizations, as depicted in the diagram. The central classes, such as 'User,' 'Author,' 'Reader,' 'Blog Post,' and 'Comment,' are fundamental to the system's functioning and interaction, forming the static model that outlines the structure and connections within the blogging platform.


BlogPost
- PostBlog : String
- EditBlog : String
- DeleteBlog : Boolean
- SetPrivacy : Boolean
- SetDate : Date
- CommentEnable:
Boolean
+ PostBlog : Post
+ EditBlog : Edit
+ DeleteBlog ()
+ SetPrivacy : Boolean
+ SetDate : Date
+SetCommentEnable: Enable
Comment
Privacy
Setup
User
+ ViewBlog
+
PostComment
+ EditComment
View
Blog
Denied
1..*
0..1
Blog
- ViewBlog
- Title
- Body
- Post
- Edit
- Delete
- PrivacySet
-
CommentEnable
Set Blog
Privacy
Privacy
Set
Public
Privet
Enable Comment
Disable
Comment
Post
Comment
View
Comment
Comment
View
Comment
Post
- UserName: String
- PostComment : String
- EditComment : String
- DeleteComment:
Boolean
- SetDate : Date
- UserName:
GetUsername
+ PostComment : Post
+ EditComment : Edit
+ DeleteComment ()
+ SetDate: Date
Comment
- View
Comment
- Username
- Body
- Post
- Edit
- Delete
Enable/Disable Comment
Enable/Disable Comment
Denied