SOFTWARE REQUIREMENTS SPECIFICATION

For

Blog Website
"The BlogSpot"

Prepared by:-
JOYCE MARIYA S J
KEERTHANAA B
SHARON NIVEDITA P
SHREE VARAA MANGAI V

Academic Year: *2023-2024*

Department of Information Technology

1. Introduction

1.1 Purpose

The purpose of The BlogSpot application is to provide individuals and organizations with a user-friendly platform for creating, managing, and publishing content in the form of articles, posts, or entries. Blogs serve as a medium for expressing ideas, sharing information, and engaging with an audience. These applications enable users to write, format, and organize their content, often incorporating multimedia elements like images and videos. Additionally, they offer features such as comment sections and social sharing options, fostering interaction and feedback from readers. Blog applications facilitate the dissemination of knowledge, personal stories, news, and opinions, making them a versatile tool for communication and expression in the digital age.

1.2 Document Conventions

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 BlogSpot Website project aims to transition from traditional blogging methods to a modern, internet-based application that allows users to create, manage, and access blog content efficiently. This project is primarily designed for bloggers and website visitors.

The product will serve as a comprehensive user interface for managing the blogging process and providing an engaging experience for both bloggers and readers. The Blogging Website can be adopted by individuals, businesses, or organizations looking to establish or enhance their online presence through blogs.

This project offers versatility and adaptability to suit various needs and scenarios. It allows for the easy incorporation of new features and modifications to cater to evolving requirements. The chosen development language for this project is JavaScript, leveraging its advantages in terms of web compatibility, extensive libraries and frameworks, cost-effectiveness, and robust development tools.

1.4 Definitions, Acronyms and Abbreviations

1. **UI: User Interface** - The visual and interactive elements of the website that users interact with.
2. **CMS: Content Management System** - A software application that allows users to create, edit, and manage digital content, such as blog posts.
3. **UI/UX: User Interface/User Experience** - UI refers to the design of the user interface, while UX refers to the overall user experience and how users interact with the website.
4. **URL: Uniform Resource Locator** - A web address used to access specific resources on the internet.
5. **SaaS: Software as a Service** - A software distribution model where applications are hosted by a third-party provider and made available to customers over the internet.
6. **DNS: Domain Name System** - The system that translates human-friendly domain names (e.g., www.example.com) into IP addresses used by computers to identify each other on the internet.
7. **React Js:** React, also known as React.js or ReactJS, is an open-source JavaScript library used for building interactive user interfaces (UIs) for web applications.

1.5 References

- **"Full-Stack React, TypeScript, and Node"** by David ChoiCovers the development of a full-stack web application using React, TypeScript, and Node.js.
- **"WordPress for Dummies"** by Lisa Sabin-Wilson
- **A beginner's guide to using WordPress**, a popular content management system (CMS).
- **"Learning React"** by Alex Banks and Eve Porcell Covers React.js, a popular JavaScript library for building user interfaces.
- **Mozilla Developer Network (MDN)** - https://developer.mozilla.org/
- **W3Schools** - https://www.w3schools.com/
- **Stack Overflow** - https://stackoverflow.com/

2. Overall Descriptions

2.1 Product Perspective
From a product perspective, our blog website focuses on:
- **User-Friendly Design:** Ensure an intuitive interface for bloggers and readers.
- **Content Management:** Provide an easy-to-use CMS for bloggers.
- **Engagement Features:** Include comments, likes, shares, and subscriptions.
- **Navigation:** Implement search, categories, and tags for easy content discovery.
- **Recommendations:** Suggest related articles to keep readers engaged.
- **Analytics:** Offer bloggers insights on content performance.
- **Monetization Options:** Consider ads, affiliate marketing, or subscriptions.
- **Security and Privacy:** Protect user data and adhere to privacy regulations.
- **SEO Optimization:** Optimize for search engines.

- **Social Integration:** Enable easy content sharing and social media presence.
- **Accessibility:** Ensure accessibility standards are met.
- **Regular Maintenance:** Plan for updates and performance improvements.
-
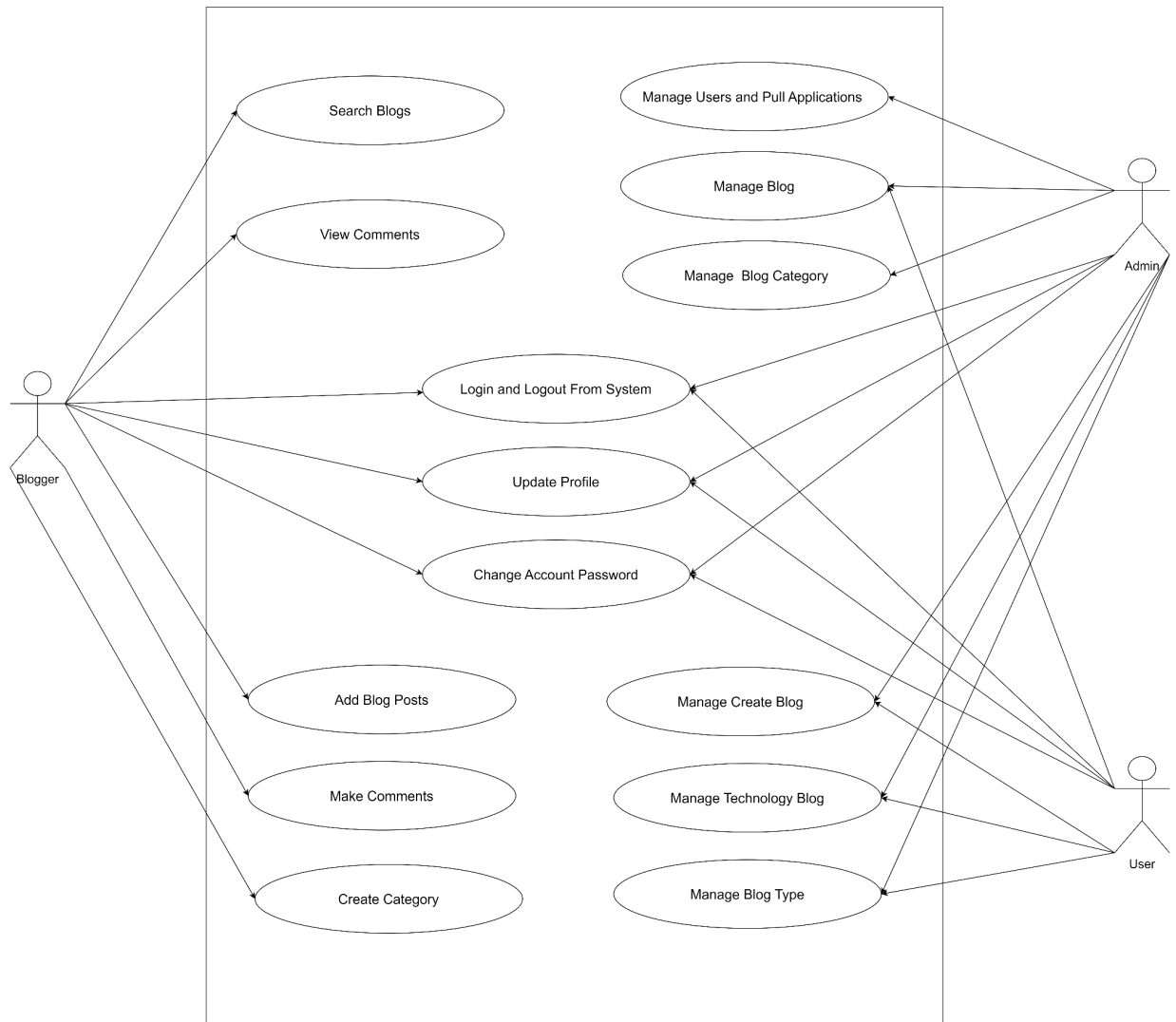
2.2 Product Function
Here are the essential product functions for a blog website:
- **Content Creation:** Allow bloggers to write, format, and publish posts.
- **User Registration:** Enable user profiles and interactions.
- **Search and Navigation**: Help users find content easily.
- **Engagement Tools**: Facilitate comments, likes, and sharing.

- **Subscription and Notifications**: Let users follow and receive updates.
- **Content Recommendations:** Suggest related posts to readers.
- **Analytics:** Provide insights into content performance.
- **Monetization Options:** Offer ad and subscription management.
- **Security and Privacy:** Protect user data and privacy.
- **SEO Features:** Optimize content for search engines.
- **Social Media Integration:** Enable easy sharing and following.
- **Accessibility:** Ensure inclusive design and usability.
- **Maintenance and Updates:** Keep the website running smoothly.
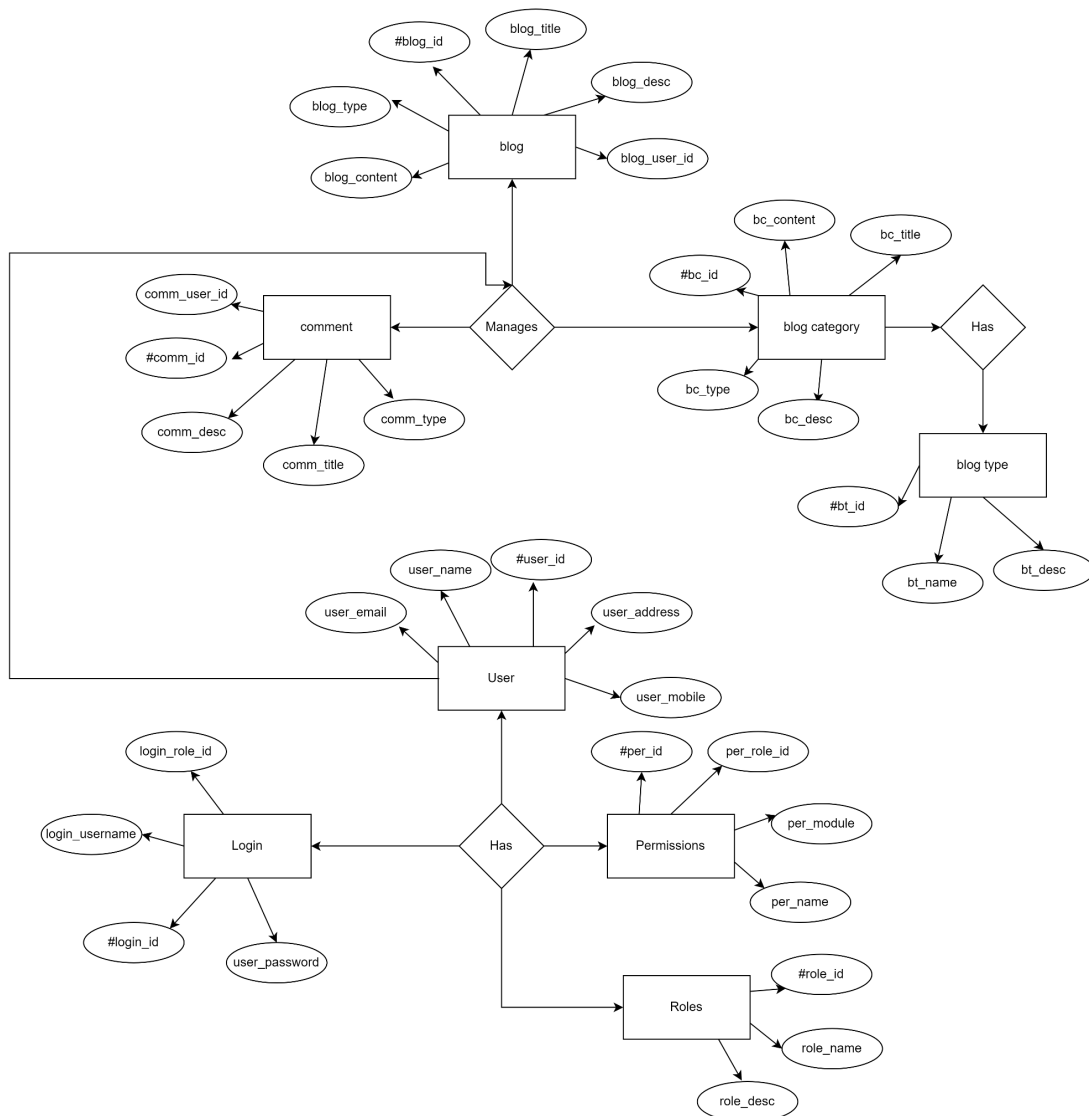
2.3 User Classes and Characteristics

Bloggers:

- Create and publish content.
- Manage their profiles.
- Access analytics on their posts.
- Moderate comments and interactions.

Readers:

- Browse and read blog posts.
- Comment on posts.
- Subscribe for updates.
- Share content on social media.

Admins/Moderators:

- Oversee content quality.
- Manage user accounts.
- Ensure website security.
- Handle technical maintenance.

2.4 Operating Environment

- **Web Server:** Hosted on a web server like Apache or Nginx.
- **Database:** Relational database (e.g., MySQL) for storing content and user data.
- **Programming Languages:** Backend in PHP, Python, Ruby, or Node.js; frontend in HTML, CSS, and JavaScript.
- **CMS Platform:** Content management system like WordPress, Drupal, or custom-built.
- **Security Tools:** Firewall, SSL/TLS encryption, and regular security updates.
- **Cloud Hosting (optional):** Utilize cloud services like AWS, Azure, or Google Cloud for scalability.
- **Monitoring and Analytics:** Tools for tracking website performance and user behavior.
- **Browser Compatibility:** Ensure compatibility with major browsers (Chrome)

2.5 Assumptions and Dependencies

Assumptions:

- **Content Availability:** Assumption of regular content creation.
- **User Interest:** Expectation of an interested user base.

- **Web Access:** Users have internet access and modern browsers.

Dependencies:

- **Hosting Provider:** Rely on a stable hosting service.
- **Database:** Depend on a functional database system.
- **Third-Party Components:** Reliance on plugins/extensions and external tools.
- **ISPs:** Users' internet service providers affect access.
- **Maintenance:** Ongoing updates and content moderation are necessary.
- **Security:** Compliance with data protection and user data security.

2.6 Requirement

- **User-Friendly Interface**: Intuitive design for easy navigation.
- **Content Management:** User-friendly CMS for writers.
- **Engagement Features:** Comments, likes, shares.
- **Navigation:** Search and content categorization.
- **Subscription & Notifications:** Email or push updates.
- **Analytics:** Post performance metrics for bloggers.
- **Security:** User data protection and spam prevention.
- **SEO Optimization:** Content discoverability.
- **Mobile Responsiveness:** Design for mobile devices.
- **Maintenance:** Regular updates and backups.

2.7 Data Requirement

- **User Data:** User profiles, including usernames, emails, and passwords.
- **Content Data:** Blog posts, including text, images, and multimedia.
- **Engagement Data:** User comments, likes, and shares.
- **Subscription Data:** User preferences for email or push notifications.
- **Analytics Data:** Metrics like page views, user demographics, and engagement statistics.
- **Security Data:** User authentication records and security logs.
- **Backup Data:** Regularly backed up content and user information.
- **Maintenance Data:** Logs of updates and maintenance activities.

3. External Interface Requirement

3.1 GUI
GUI design provides an intuitive and engaging user experience for both bloggers and readers of the website.
- **Homepage:** Clean and organized layout displaying featured posts and categories.

- **Navigation Bar:** Menu for easy access to categories, search, and user profile.
- **User Profile:** Personalized page for user details, posts, and settings.
- **Search Bar:** Prominent search functionality for finding specific content.
- **Blog Post Page:** Consistent template for reading and interacting with blog posts.
- **Comment Section:** User-friendly interface for leaving comments and reactions.
- **Subscription Page:** Manage subscriptions and notifications preferences.
- **Admin Dashboard :** Access for content creators to manage posts, analytics, and user interactions.

- **Accessibility Features:** Inclusion of accessibility options like text resizing and contrast settings.
- **Responsive Design:** Ensures usability on various devices and screen sizes.

4. System Features-sharon

- **User Management:** User registration, profiles, and authentication.
- **Content Handling:** Post creation, editing, and categorization.
- **User Engagement:** Comments, likes, shares, and subscriptions.
- **Notifications**: Email or push notifications for updates.
- **Recommendations**: Suggested posts based on user behavior.
- **Analytics**: Post performance tracking and user insights.
- **Monetization:** Ad management and subscription options.
- **Security:** User data protection and spam prevention.
- **SEO and Social:** SEO optimization and social media integration.
- **Accessibility**: Inclusive design and usability.
- **Maintenance:** Regular updates and backups.

5. Other Non-functional Requirements

5.1 Performance Requirement
Performance requirements are essential for ensuring a fast, responsive, and reliable blog website that can handle varying levels of user traffic.
- **Page Load Speed:** Pages should load quickly to provide an optimal user experience. Aim for load times under 3 seconds.
- **Scalability:** The website should handle increasing traffic gracefully, with the ability to scale resources as needed.
- **Concurrent Users:** Support a significant number of concurrent users without slowdowns or errors, especially during traffic spikes.
- **Caching:** Implement caching mechanisms to reduce server load and improve response times.
- **Database Optimization:** Optimize database queries and indexing to ensure efficient data

retrieval.

- **Content Delivery:** Utilize a Content Delivery Network (CDN) to serve static assets (images, CSS, JavaScript) from multiple geographic locations for faster content delivery.
- **Mobile Responsiveness:** Ensure the website is responsive and performs well on mobile devices.
- **Error Handling**: Provide informative error messages and graceful degradation in case of errors or downtime.
- **Load Testing**: Regularly conduct load testing to identify performance bottlenecks and address them proactively.
- **Security Performance:** Implement security measures without sacrificing website speed, such as through efficient encryption and authentication methods.
- **Database Backups:** Regularly backup the database to prevent data loss and ensure recovery in case of issues.
- **Content Distribution:** Use lazy loading for images and videos to improve initial page load times.
- **Database Caching:** Implement database query caching to reduce database load and enhance performance.
- **Content Minimization:** Minimize CSS, JavaScript, and HTML files to reduce load times.

5.2 Safety Requirement

Safety requirements ensure the protection of user data and the overall security of the blog website.

- **User Data Protection:** Secure storage and encryption of user information, including passwords.
- **Content Moderation:** Tools to detect and prevent spam, abusive content, and malware.
- **Regular Backups:** Scheduled backups to prevent data loss in case of incidents.
- **Firewall and DDoS Protection:** Protection against cyberattacks and DDoS attacks.
- **Secure Payment Processing:** If applicable, secure payment processing for subscriptions or purchases.
- **Privacy Compliance:** Adherence to data protection regulations, including GDPR or CCPA.
- **User Authentication:** Strong user authentication to prevent unauthorized access.
- **Software Updates:** Regularly update software and plugins to patch security vulnerabilities.
- **Incident Response Plan:** Preparedness and procedures for handling security incidents.
- **Access Control:** Limit access to sensitive areas and functions based on user roles.

5.3 Security Requirement

Security requirements are essential to protect user data and the integrity of the blog website.

- **User Authentication:** Strong authentication to protect user accounts.
- **Data Encryption**: Encrypt user data and sensitive information.
- **Firewall Protection:** Implement a firewall for defense against cyberattacks.
- **Regular Updates**: Keep software, plugins, and systems up to date for security patches.
- **Secure Hosting:** Choose a reputable and secure hosting provider.
- **Backup System:** Regularly backup data to prevent loss in case of incidents.
- **Access Control:** Limit access to sensitive areas based on user roles.
- **Security Audits:** Conduct regular security audits and vulnerability assessments.
- **DDoS Mitigation:** Implement measures to counter Distributed Denial of Service attacks.
- **Incident Response Plan:** Preparedness for handling security incidents effectively.

5.4 Requirement attributes

- **Clear and Specific:** Requirements should be unambiguous and precise, leaving no room for interpretation.
- **Measurable:** Define criteria to objectively measure compliance or completion of each requirement.
- **Prioritized:** Rank requirements by importance to focus on critical features first.
- **Testable:** Ensure each requirement can be tested or validated to confirm its implementation.
- **Traceable:** Establish clear links between requirements and project objectives.
- **Feasible:** Requirements should be realistic and achievable within the project's constraints.
- **Complete:** Ensure all necessary functionality and constraints are included.
- **Consistent:** Requirements should not conflict with each other and should align with project goals.
- **Modifiable:** Allow for changes and updates to accommodate evolving needs.
- **Aligned with Stakeholder Needs:** Ensure that requirements address the needs and expectations of all relevant stakeholders.

5.5 Business Rules

- **User Registration:** Users must register with valid information to create accounts.
- **Content Guidelines:** Enforce content guidelines to maintain quality and prevent inappropriate content.
- **Privacy Policy:** Comply with privacy regulations and protect user data.
- **Monetization Rules:** Implement rules for ads, subscriptions, or sponsored content.
- **Comment Moderation:** Moderate and filter comments to prevent spam and abusive content.
- **Copyright Protection:** Respect copyright laws and protect intellectual property.
- **Content Ownership:** Clarify ownership and rights of published content.
- **Security Measures:** Implement security protocols to safeguard against breaches and data theft.
- **User Conduct:** Set guidelines for user behavior, including respectful interactions.
- **Compliance:** Adhere to relevant laws and regulations, such as GDPR or COPPA.

5.6 User Requirement

- **User-Friendly Interface:** Easy navigation and intuitive design for bloggers and readers.
- **Content Accessibility:** Quick access to blog posts and efficient search features.
- **Interaction Features:** Ability to comment, like, share, and subscribe for updates.
- **Personalization:** Customizable profiles and content recommendations.
- **Security and Privacy:** Secure registration and protection of user data.
- **Mobile Responsiveness:** Accessibility and usability on mobile devices.
- **Feedback and Support:** Channels for user feedback and assistance.
- **Reliable Performance:** Fast page loading and minimal downtime.

6. Other Requirements

6.1 Data and Category Requirement
These requirements ensure effective content management and user engagement on the blog website.

Data Requirements:
- **User Data:** User profiles, usernames, emails, and passwords.
- **Content Data:** Blog posts, including text, images, and multimedia.
- **Engagement Data:** User comments, likes, shares, and subscriptions.
- **Analytics Data:** Post performance metrics, user demographics.
- **Security Data:** Authentication records and security logs.

Category Requirements:
- **Content Categories:** Categories for organizing and filtering blog posts.
- **Tagging System:** Tags to further categorize and label posts.
- **User Roles:** Differentiate between bloggers, readers, and moderators.
- **Subscription Categories:** Allow users to subscribe to specific content categories.

6.2 Appendix

These elements in the appendix provide transparency, legal clarity, and user support for the blog website.

- **User Guidelines:** Clear instructions for users on registration, posting, and engagement.
- **Privacy Policy:** Document outlining data handling and privacy practices.
- **Content Guidelines:** Rules and standards for acceptable content.
- **Terms of Service:** Legal terms and conditions for website usage.
- **Copyright Notice:** Information regarding content ownership and usage rights.
- **Contact Information:** Contact details for inquiries and support.
- **FAQ Section:** Frequently asked questions and answers for users.
- **Glossary:** Definitions of technical or industry-specific terms.
- **Feedback Form:** A form for users to provide feedback or report issues.
- **Advertising Policy:** Rules and guidelines for advertisements on the website.
- **Sitemap:** A visual representation of the website's structure for reference.
- **Accessibility Statement:** Information on website accessibility for users with disabilities.

6.3 Glossary

1. **Blog:** A regularly updated online publication containing articles, posts, or entries typically displayed in reverse chronological order, often focused on specific topics or themes.
2. **Blogger:** A person who creates and publishes content on a blog, often sharing their thoughts, expertise, or experiences with an online audience.
3. **User Interface (UI):** The graphical and interactive elements of a website or application through which users interact with and navigate the platform.
4. **User Experience (UX):** The overall experience and satisfaction a user has when interacting with a website or application, encompassing ease of use, accessibility, and overall enjoyment.
5. **JavaScript (JS):** A popular programming language commonly used for web development to add interactivity and dynamic functionality to websites.
6. **HTML (Hypertext Markup Language):** The standard markup language for creating and structuring content on web pages.
7. **CSS (Cascading Style Sheets):** A stylesheet language used to control the presentation and layout of HTML documents on a web page.

8. **Responsive Design:** Web design that ensures a website's layout and content adapt and display appropriately on various screen sizes and devices, including desktops, tablets, and smartphones.
9. **Comment Section:** A feature on blog posts that allows readers to leave comments, share feedback, and engage in discussions related to the content.
10. **Tags and Categories:** Organizational tools that help categorize and classify blog posts, making it easier for users to find content on specific topics.
11. **Archives:** A section of the website where older blog posts are stored and can be accessed by users, often organized by date or category.
12. **Dashboard:** The backend interface where bloggers can manage and create new content, view analytics, and make administrative changes to the website.
13. **Analytics:** Tools and data that provide insights into website traffic, user behavior, and other performance metrics, helping bloggers understand their audience and improve content.
14. **Hosting:** The service that stores and makes the website accessible on the internet; it typically involves renting server space from a hosting provider.
15. **Domain Name:** The unique web address that users type into their browsers to access a website (e.g., www.blogspot.com).
16. **HTTPS (Hypertext Transfer Protocol Secure):** A secure version of HTTP, providing encrypted data transfer between the user's browser and the website server.

6.4 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 describes 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 working. 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.

ROLE CLASS

+role_id: int
+role_title: string
+role_description: string

+addRole()
+editRole()
+delete Role()
+searchRole()
+assignRole()

PERMISSION CLASS

+permission_role id: int
+permission_title string
+permission_module: string
+permission_description: string

+addPermission()
+editPermission()
+delete permission ()
+searchPermission()

CREATE_BLOG CLASS

+create_blog_id: int
+create_blog_user_id: int
+create _blog_title:string
+create_blog_content:string
+create_blog_type: string
+create_blog_description: string

+ addCreateblog()
+ editCreateblog()
+ deleteCreateblog()
+ searchCreateblog()

USER CLASS

+user_id:int
+user_role_id:int
+user_name:string
+user_email:string
+user_dob:string
+user_address:string

+ addUser()
+ editUser()
+ deleteUser()
+ searchUser()

COMMENT CLASS

+comment_id: int
+comment_user_id: int
+comment_title:string
+comment_type: string
+comment_description: string

+ addComment()
+ editComment()
+ delete Comment()
+ searchComment()

BLOG CLASS

+blog_id: int
+blog_user_id: int
+blog_title:string
+blog_content:string
+blog_type: string
+blog_description: string

+ addBlog()
+ editBlog()
+ deleteBlog()
+ searchBlog()

BLOG CATEGORY CLASS

+blog category_id: int
+blog category_title:string
+blog category_content:string
+blog category_type: string
+blog category_description: string

+ addBlogCategory()
+ editBlogCategory()
+ deleteBlogCategory()
+ searchBlogCategory()

BLOG TYPE CLASS

+blog_id: int
+blog_title: string
+blog_description: string

+addBlogtype()
+editBlogtype()
+deleteBlogtype()
+searchBlogtype()