



## **UE22CS352B - Object Oriented Analysis & Design**

### **Mini Project Report**

#### **Title**

*Submitted by:*

*Name : SRN (Team Members)*

<b>Naman Rao</b>	<b>PES1UG22CS372</b>
<b>Naveen Nair</b>	<b>PES1UG22CS377</b>
<b>Nikhilesh Chowdary</b>	<b>PES1UG22CS389</b>
<b>Prakhyath Shetty</b>	<b>PES1UG22CS426</b>

Semester Section

**Assistant Professor Bhargavi Mokashi**

**January - May 2025**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
FACULTY OF ENGINEERING  
PES UNIVERSITY**

(Established under Karnataka Act No. 16 of 2013)  
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

## **Problem Statement:**

## **Key Features:**

### **1. User Authentication and Authorization**

- Implements secure login using JWT (JSON Web Tokens).
- Access to protected routes is restricted using Angular Route Guards.
- Only authenticated users can access and interact with the application beyond the login page.
- Role-based validation ensures that users can only edit blogs they have authored.

### **2. Blog Management Functionality**

- Full support for Create, Read, and Update blog operations.
- Blogs are automatically linked to the currently logged-in user as the author.
- Users can only modify their own blogs to maintain data integrity and security.

### **3. Navigation and Access Control**

- A clean navigation bar that includes links to Home, Create Blog, and Blog List.
- The navigation bar is only visible after the user has logged in, enhancing security.
- Unauthorized users are redirected to the login page when trying to access restricted routes.

### **4. Search Functionality**

- Includes a search bar to filter blog posts by title in real-time on the blog list page.
- Enhances usability by allowing users to quickly find relevant posts.

### **5. Backend Integration with Spring Boot and MySQL**

- The application uses Spring Boot to expose RESTful API endpoints for blog operations.
- MySQL is used as the database to persist user and blog data.
- The backend is responsible for handling business logic and interacting with the database.

### **6. Modular Angular Architecture**

- Utilizes standalone Angular components for login, blog creation, blog listing, blog details, and editing.
- Services are used for managing user authentication state and blog data operations.
- Promotes reusability and maintainability of the codebase.

### **7. Responsive and User-Friendly Interface**

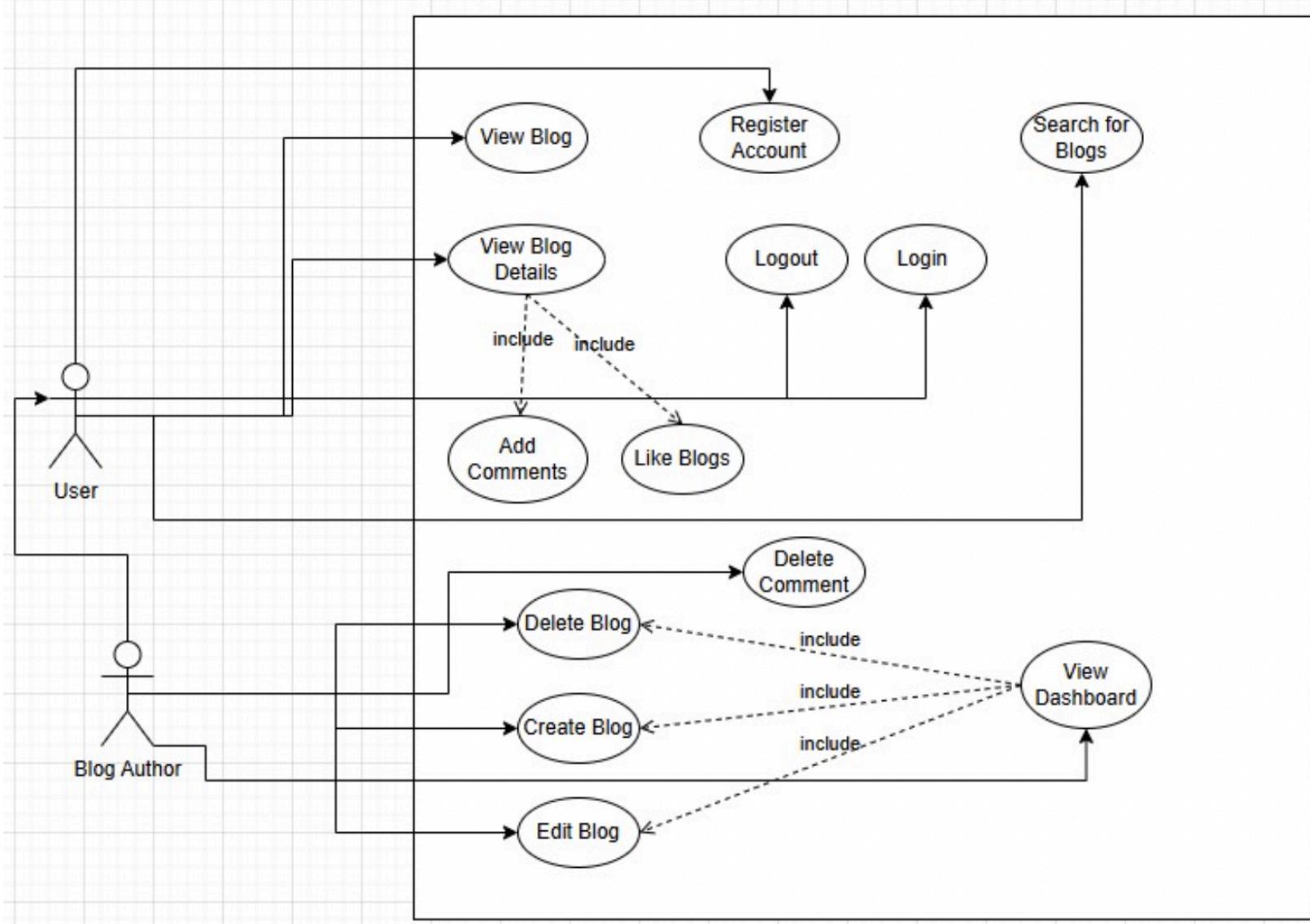
- Styled using modern CSS with clear feedback for user actions (success and error messages).
- User-friendly form layout with validations to ensure proper input.

## 8. Secure API Access

- Spring Security ensures that APIs are protected and only accessible with a valid JWT.
- Authentication is enforced on the backend for all sensitive operations.

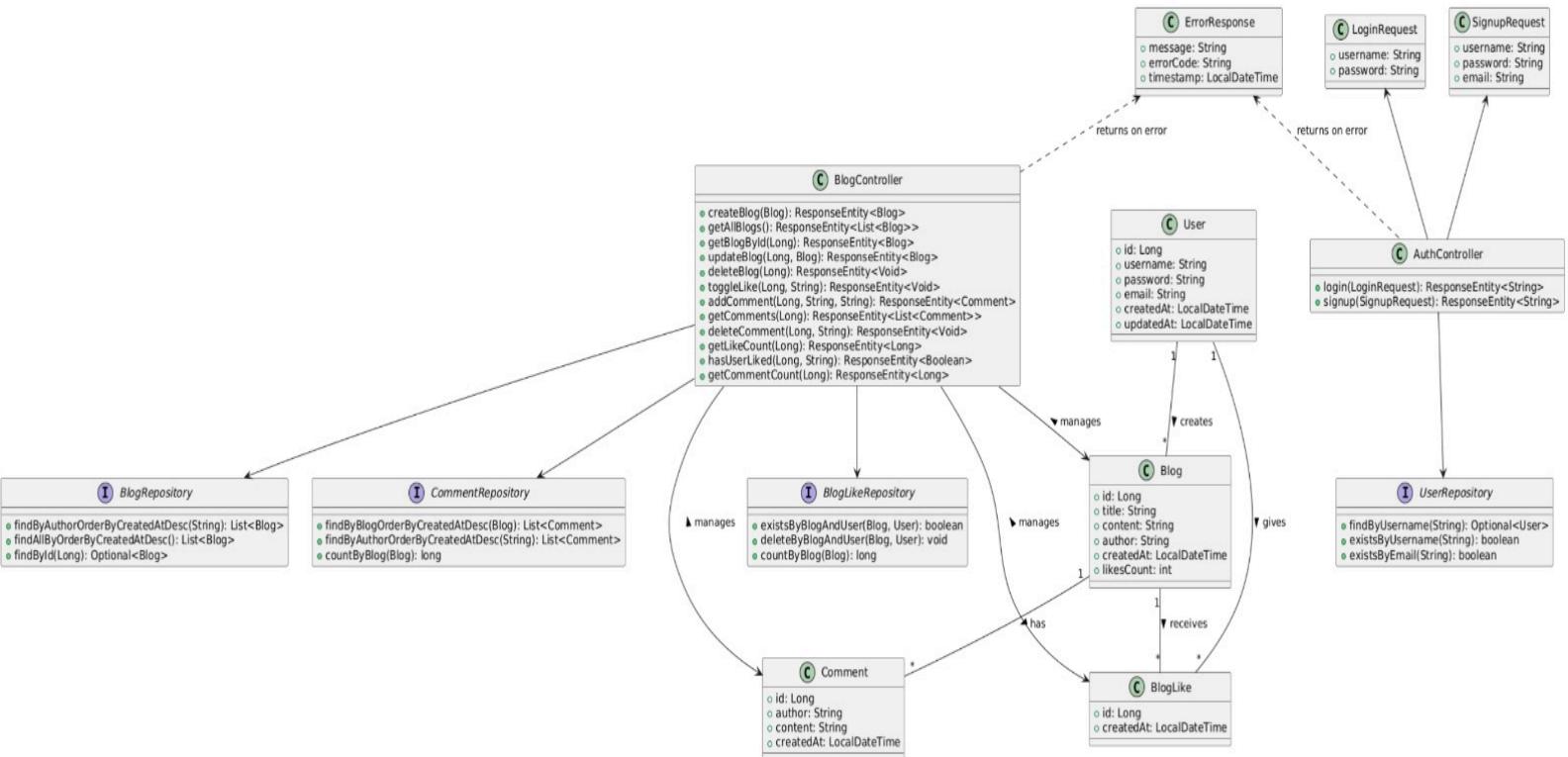
## **Models:**

## **Use Case Diagram:**

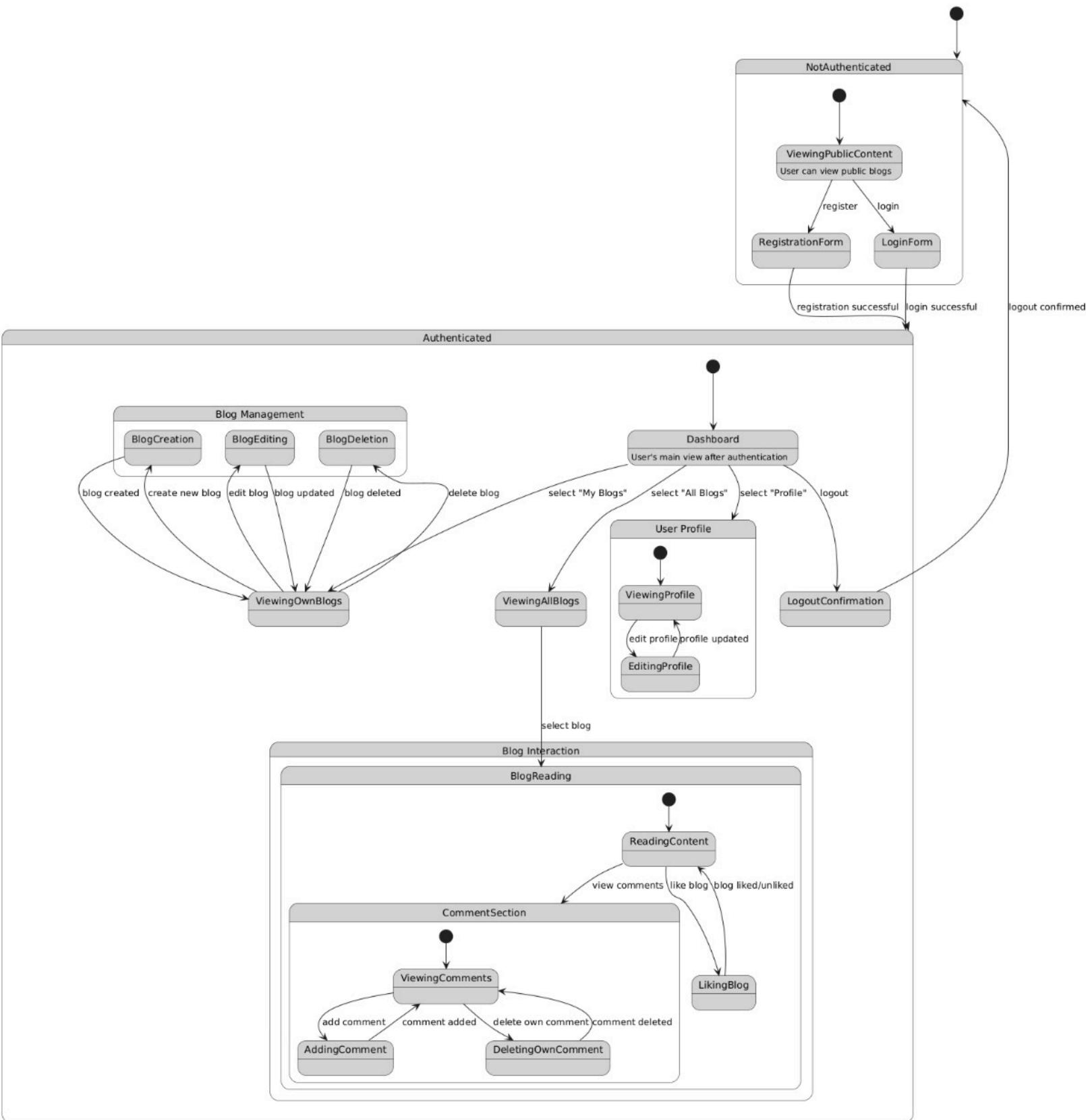


## Class Diagram:

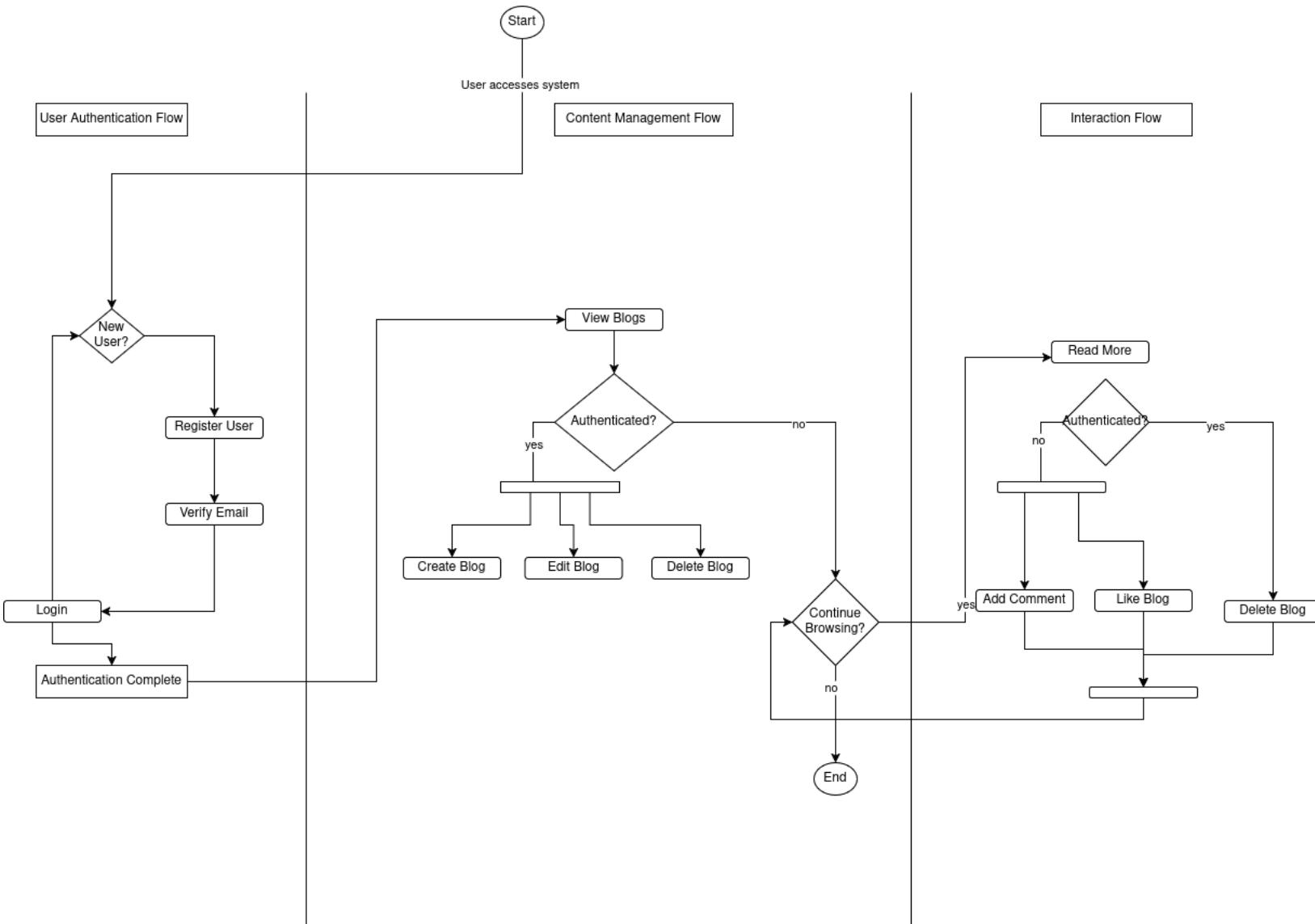
Complete Blogging Platform Class Diagram



## State Diagram:



## Activity Diagrams:



# **Architecture Patterns, Design Principles, and Design Patterns:**

## **Architecture Patterns**

Model – View – Controller Pattern (MVC)

## **Design Principles**

### **SOLID Principles:**

1. Single Responsibility Principle (SRP):
  - a. Each Class has a single responsibility(e.g., AuthService handles authentication, BlogService handles blog operations)
  - b. DAO interfaces (UserDao, BlogDao, CommentDao) are focused on data access operations
  - c. Controllers (AuthController) handle HTTP requests
2. Open Closed Principle
  - a. The use of interfaces (UserDao, BlogDao, CommentDao) allows for extension without modification
  - b. Service classes are open for extension through inheritance or composition
3. Liskov Substitution Principle(LSP):
  - a. Implementations (UserDaoImpl, BlogDaoImpl, CommentDaoImpl) properly implement their respective interfaces
  - b. Custom exceptions extend RuntimeException appropriately
4. Interface Segregation Principle(ISP):
  - a. DAO interfaces are small and focused on specific operations
  - b. Each interface has only the methods it needs

## **Design Patterns**

Multiple **object-oriented design patterns** were implemented to solve recurring problems in a systematic and reusable manner:

### **1. Factory Method Pattern**

Utilized through Spring's @Bean annotations in configuration classes to instantiate objects.

### **2. Service Layer Pattern**

Services like AuthService, BlogService, CommentService handle business logic. Services coordinate between controllers and DAOs

### **3. Data Access Object Pattern(DTO)**

Used extensively with interfaces like UserDao, BlogDao, CommentDao. Implementation classes like UserDaoImpl, BlogDaoImpl handle database operations

### **4. Repository Pattern**

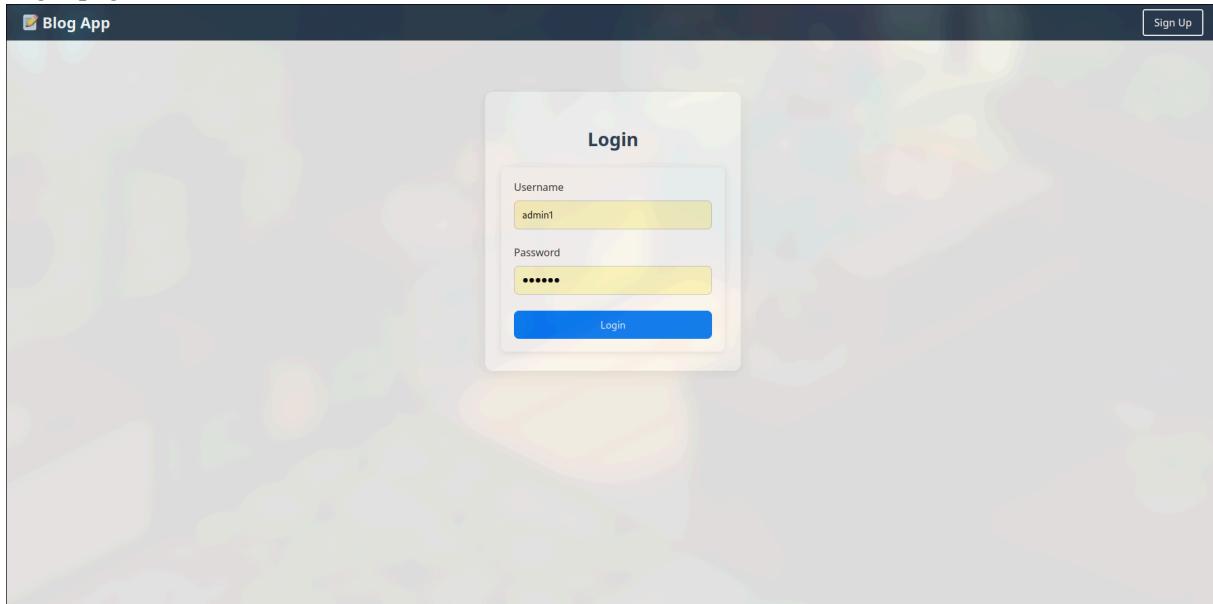
DAO implementations act as repositories for data access. Provides a clean separation between data access and business logic

**Github link to the Codebase: <https://github.com/NaveenNair04/BlogServer>**

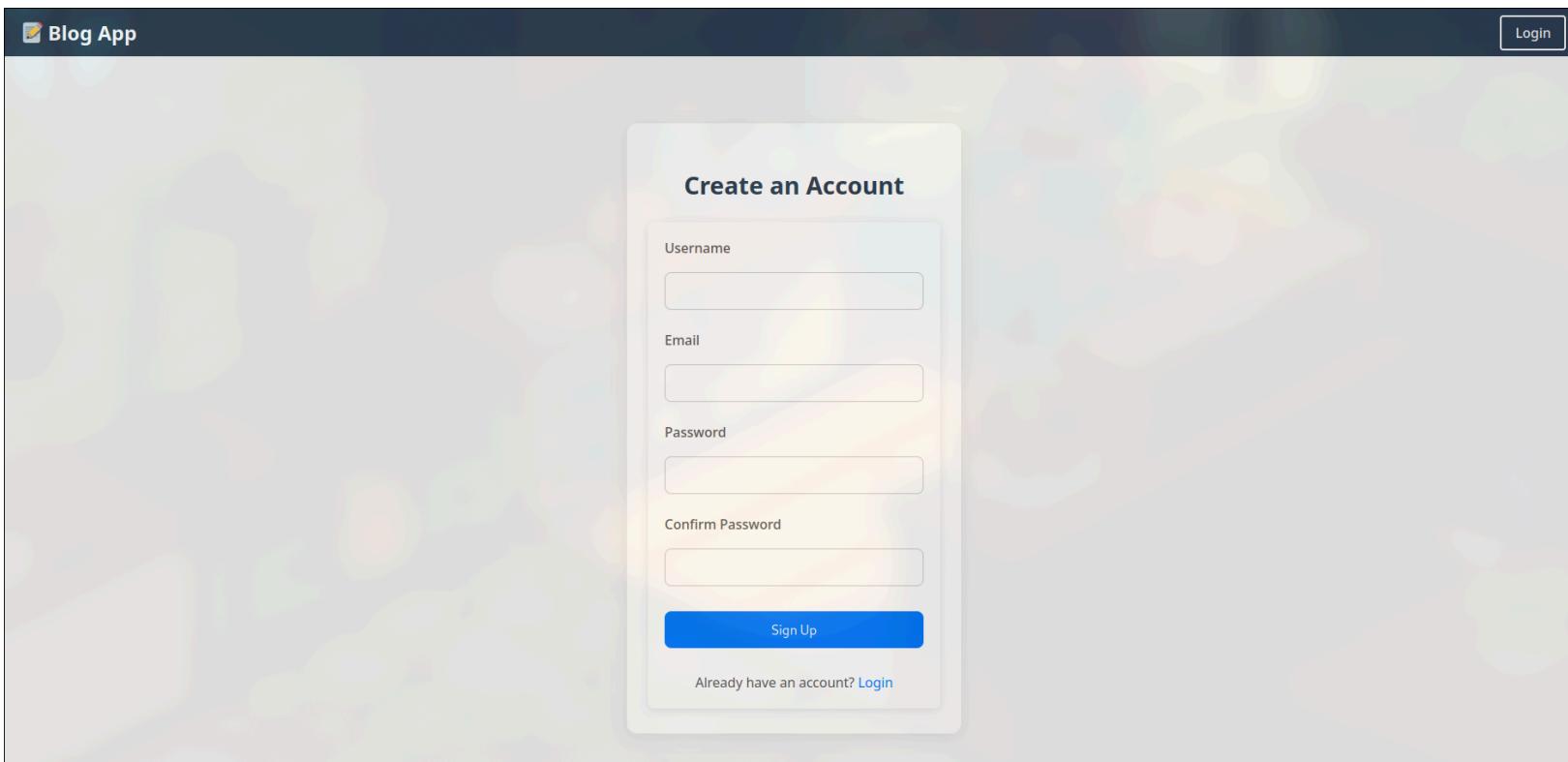
## **Screenshots:**

**UI:**

Login page:



Signup Page:



Dashboard:

# Blog App

Dashboard All Blogs Create Blog admin1 | Logout

## Welcome to Your Dashboard

Hello, admin1! Here's your blog overview.

### Quick Actions

[Create New Blog](#) [View All Blogs](#)

### Recent Blogs

**another one**  
another one?...  
By admin1 April 25, 2025, 11:44:49 AM [Read More](#)

**my blog 1**  
this my first blog...  
By admin1 April 25, 2025, 10:47:03 AM [Read More](#)

## All Blogs:

# Blog App

Dashboard All Blogs Create Blog admin1 | Logout

Search blogs by title...

**another one**  
by admin1 · Apr 25, 2025, 11:44:49 AM  
another one?...  
[Read More](#) [Edit](#) [Delete](#) [0](#) [0](#)

**my blog 1**  
by admin1 · Apr 25, 2025, 10:47:03 AM  
this my first blog...  
[Read More](#) [Edit](#) [Delete](#) [1](#) [1](#)

**not asdf**  
by admin · Apr 14, 2025, 5:01:34 PM  
not asdfas...  
[Read More](#) [1](#) [0](#)

**another test**  
by admin · Apr 14, 2025, 4:57:35 PM  
is my username admin?...  
[Read More](#) [1](#) [0](#)

**swastikas diet**  
by swastika · Apr 9, 2025, 3:11:27 PM  
nhi ho rha.....  
[Read More](#) [0](#) [1](#)

**chat bot**  
by nishita · Apr 8, 2025, 4:09:26 PM  
couldnt do it.....  
[Read More](#) [0](#) [0](#)

**test5**  
by test5 · Apr 7, 2025, 10:51:44 AM  
test5...  
[Read More](#) [0](#) [0](#)

**dsgffgh**  
by fhgf · Apr 7, 2025, 10:49:21 AM  
fgfj...  
[Read More](#) [0](#) [0](#)

**asdf**  
by asfd · Apr 7, 2025, 10:48:57 AM  
asdf...  
[Read More](#) [0](#) [0](#)

## Create Blog page:

### Create New Blog

Title

Content

[Back](#) [Create Blog](#)

Edit Blog:

[Update Blog](#)

Add Comment:

# my blog 1

by admin1 on Apr 25, 2025, 10:47:03 AM

this my first blog

1

## Comments (1)

[Post Comment](#)

admin1

Apr 25, 2025, 11:44:30 AM

yo

[Delete](#)

[« Back to Blogs](#)

**Individual contributions of the team members:**

<b>Name</b>	<b>Module worked on</b>
<b>Naveen Nair</b>	Database Connection and Request Handling
<b>Nikhilesh Chowdary</b>	Blog Modules
<b>Prakyath Shetty</b>	User Modules
<b>Naman Rao</b>	Like and Comment Modules