

Low Level Design

Authentication System



Prerequisites

Before diving into the problem, let's understand what authentication and authorization is and how asp.net core handles authentication and authorization.

Authentication

Authentication is the process of checking whether the user is part of the system or not.

Authentication checks the user present in the system database or not.

Authorization

Authorization is the process of checking whether the authenticated user has access to a particular resource or not.

A system contains many operations on resources such as updating blogs ,adding blogs and updating profile settings.

Authentication is the first step verification for Authorization.

How this System Handles Authentication

I am using Json Web Token (JWT) based authentication that creates a token with different roles such as Admin,User and sending it as a cookie to the client. The token will be appended to every request to that domain from that client.

Design Problem

Design a Blog system where students can register themselves to see the blogs and give ratings to the blogs. The admin can add blogs to the system and remove existing blogs if it has any invalid data.

Solution

We follow the below steps to solve the problem.

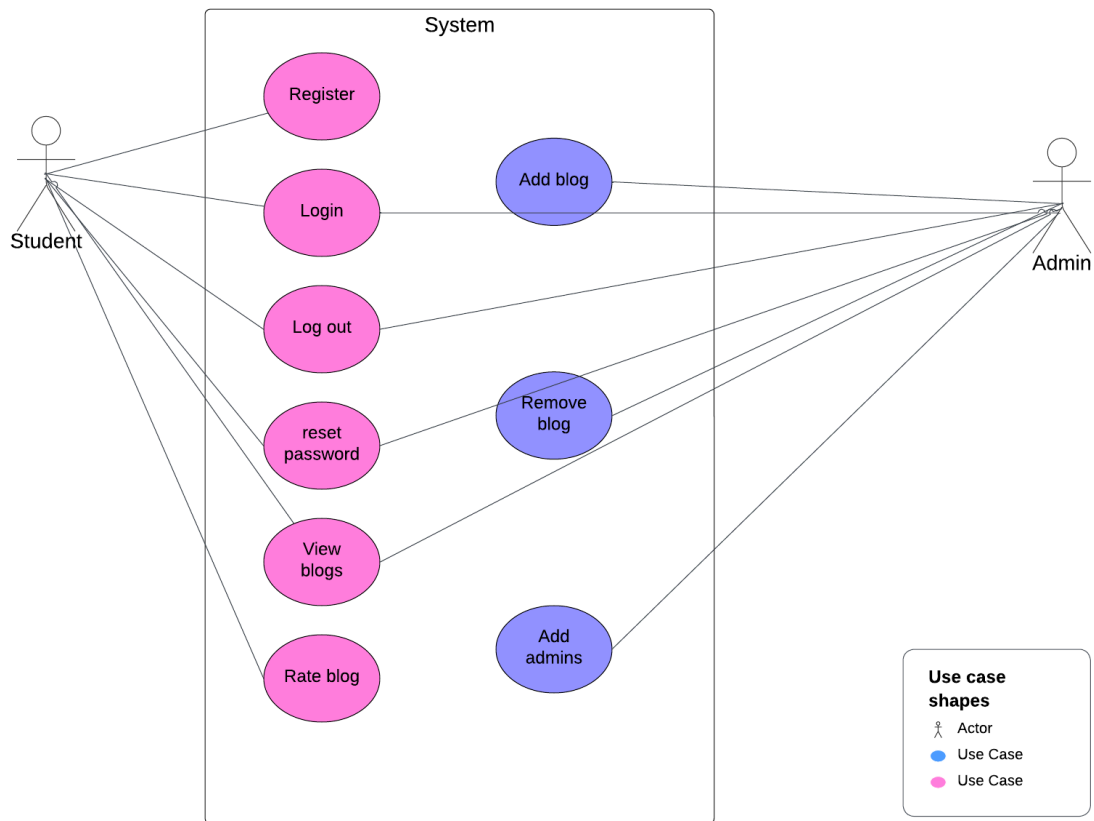
1. List down the requirements.
2. Make a use case diagram.
3. Make a class diagram.
4. Find the patterns and code.

List Down requirements

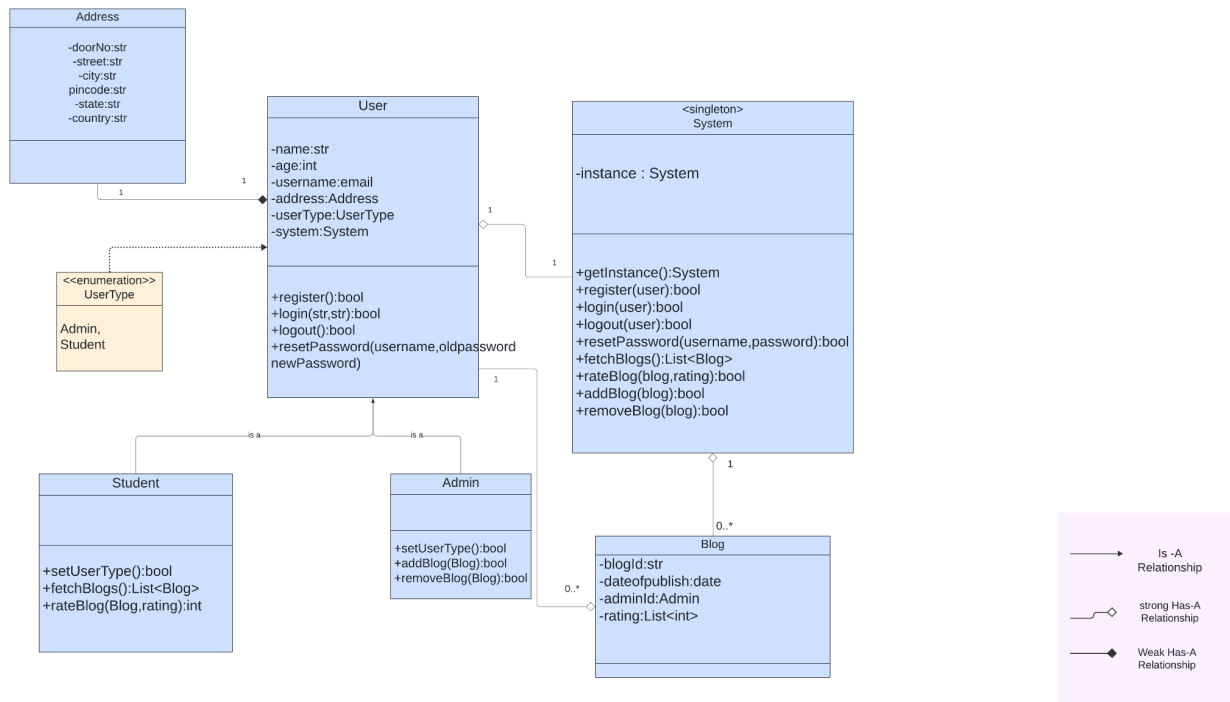
- Students should register themselves through their Email.
- Make sure one user per Email.
- Students should be able to reset his password.
- Students can see the blogs present in the system.
- Students can't remove the blogs.
- Admin can add and remove the blogs.
- Students can give a rating out of 5 to a blog.
- Admin can create accounts for admins.

Note: In this design, I am mainly concerned with how an authentication will be implemented for users with different roles. I am using JWT for authentication and authorization purposes.

Use Case Diagram



Class Diagram



Finding Patterns and Code

- In the above diagram, I make use of inheritance to create student and admin classes.
- I also used singleton design patterns to get a single instance of the system to perform the API calls.
- As I told earlier, I mainly focused on the authentication system rather than the functional requirements.
- You can find the Web API code in the repository with testing from the postman client.

Conclusion

- Low level design problems are open ended. There is no perfect solution to it.
- 80% percent of solutions for the problems are acceptable.

Resources and References

(extended from MakeMyTrip repository resources. (<https://github.com/vamsi-krishna-mishran/MakeMyTrip/blob/main/MakeMyTripLLD.pdf>))

<https://www.lucidchart.com/pages/>

<https://textbooks.cs.ksu.edu/cc410/i-oop/05-uml/06-associations/index.html>