Blog - RestAPI

- Description
- Objectives
- Scope
- Diagrams

Description

- This is an **admin-central blog**, in which the admin has the major authorities.
- Admin can **create** new posts, categories and **update and delete** them also. He also can see and delete his registered readers.
- The project is secured with **Jwt(json web token) and Spring security,** which forbids normal people to access the admin panel. A normal person is only given access like reading to blogs and making some filters.
- Also the entities of the project are linked through **foreign key-constraints** and relationship concepts like **one to one and many to one** is also applied in the project.
- Further, **pagination** feature is also implemented in the show all blog route, **swagger-documentation**, **hazelcast-cache**, **junit-5 test** for registration route is also there.

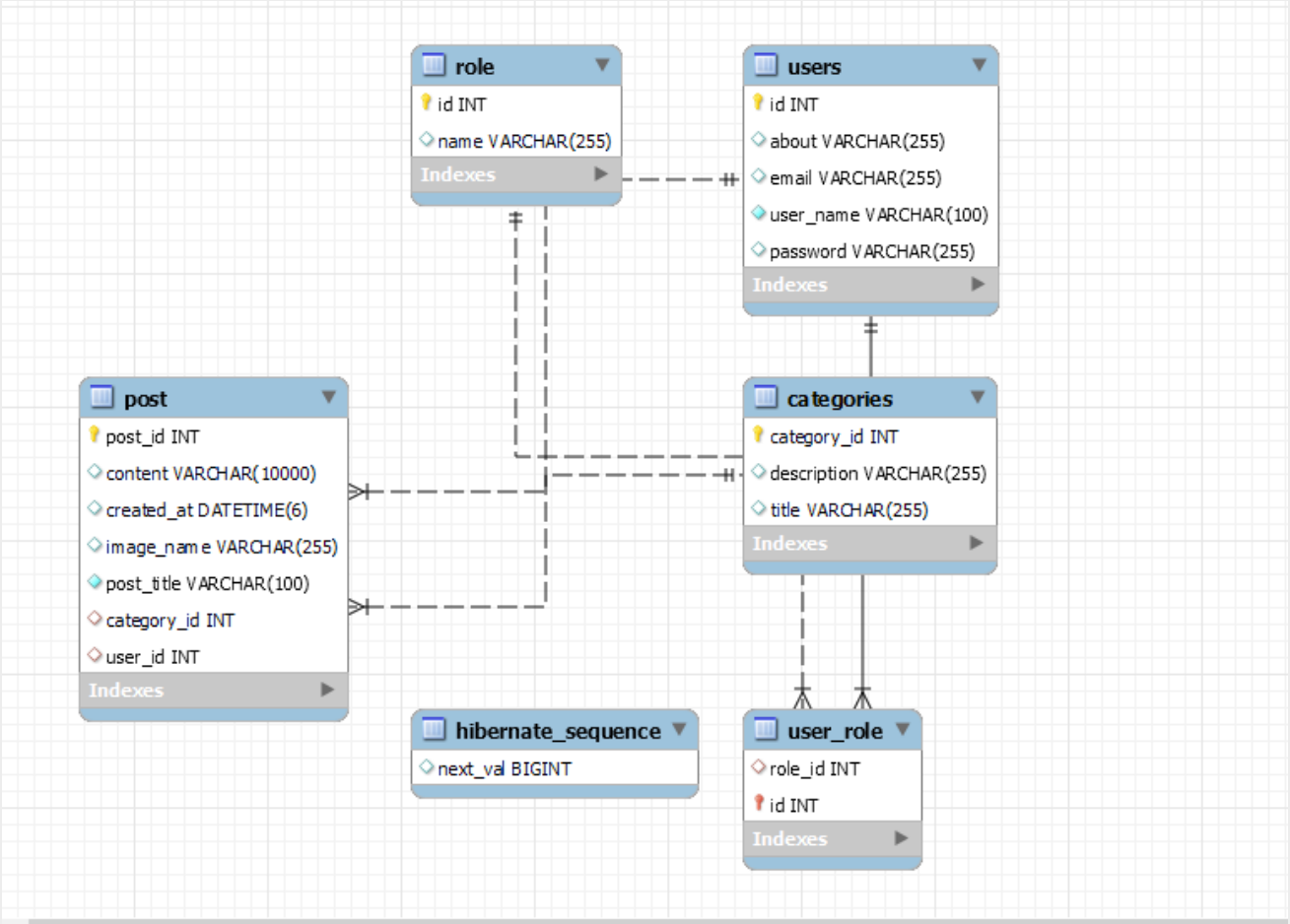
Objectives

- The objective of the project is to build a platform in which authors can come and open their **blogging profiles**. On the other hand reader can come and filter on the basis of author and categories and make best use of the available content.

Scope

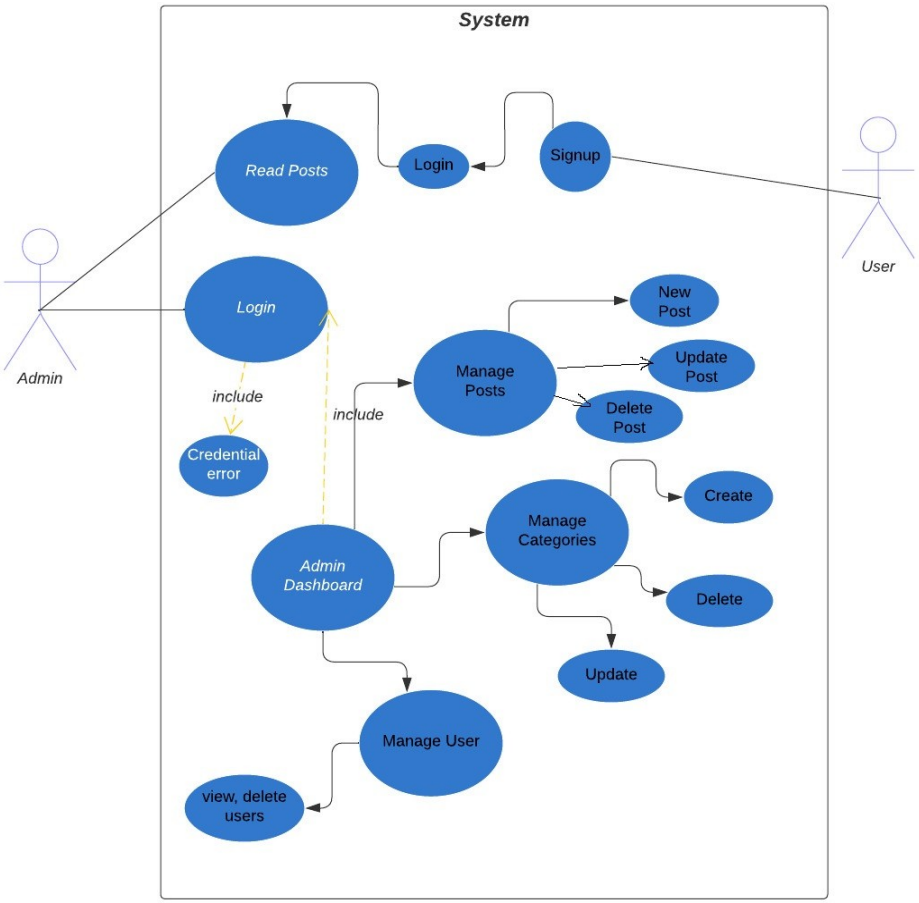
- Let's first discuss the scope of admin. The admin panel consists of three major action buttons, managing post, categories and user.
- **Manage Posts**- Inside manage post admin can create new post, update or delete old posts.
- **Manage Categories**- Inside manage categories, admin can create new category or update/delete old ones.
- **Manage User**- Admin can't update user profile but he can see the list of registered users and also delete them if he wishes to.
- A normal user can on the other hand read posts and filter them based on categories and author.
- **Read posts**- Read a specific posts or cal view all the posts.
- **Filter by Category**- Search posts by categories
- **Filter by Author**- Search posts by author.

Database

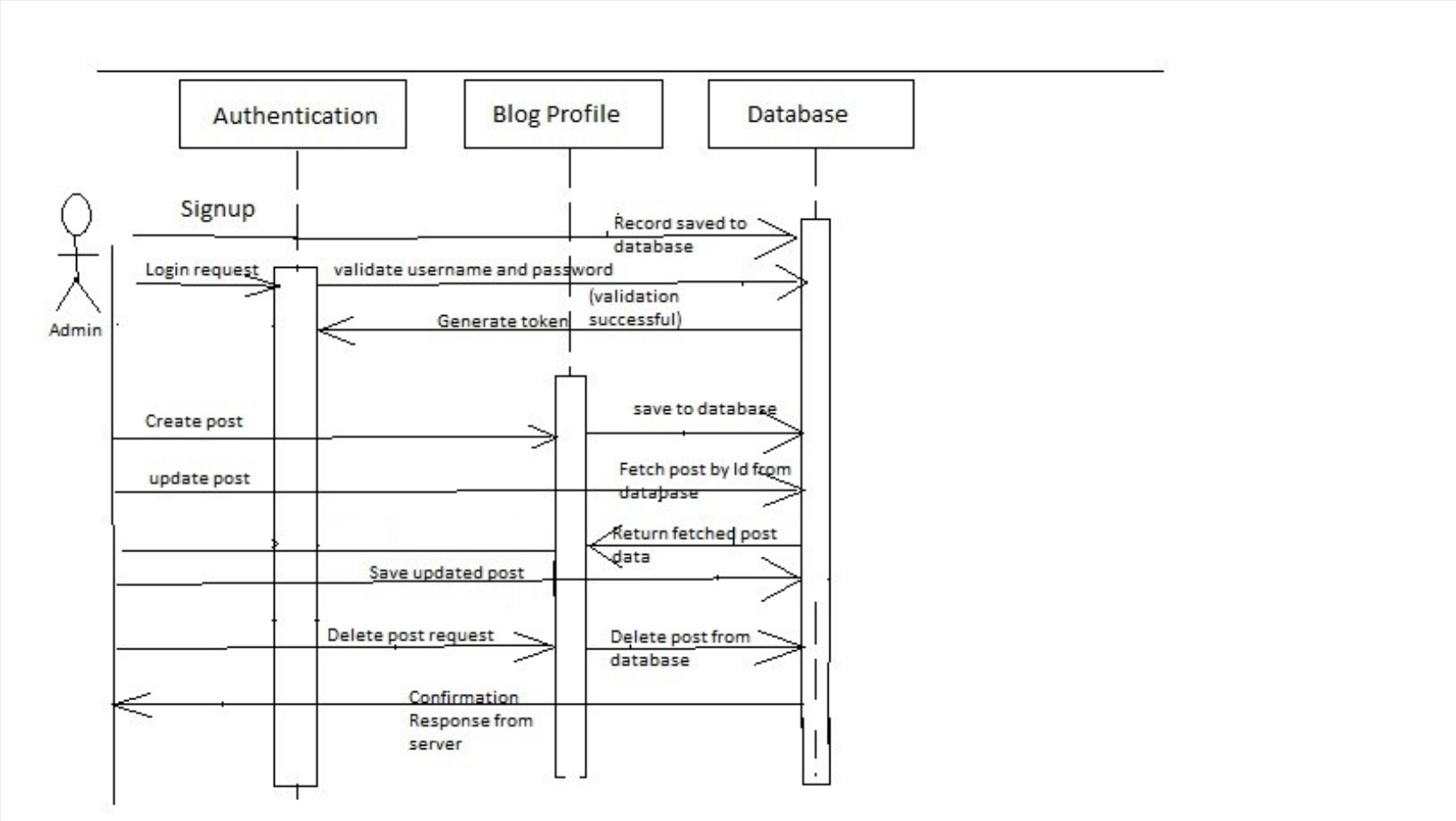

role
id INT
name VARCHAR(255)
Indexes
users
id INT
about VARCHAR(255)
email VARCHAR(255)
user_name VARCHAR(100)
password VARCHAR(255)
Indexes
post
post_id INT
content VARCHAR(10000)
created_at DATETIME(6)
image_name VARCHAR(255)
post_title VARCHAR(100)
category_id INT
user_id INT
Indexes
categories
category_id INT
description VARCHAR(255)
title VARCHAR(255)
Indexes
hibernate_sequence
next_val BIGINT
user_role
role_id INT
id INT
Indexes

Use case diagram-Blog

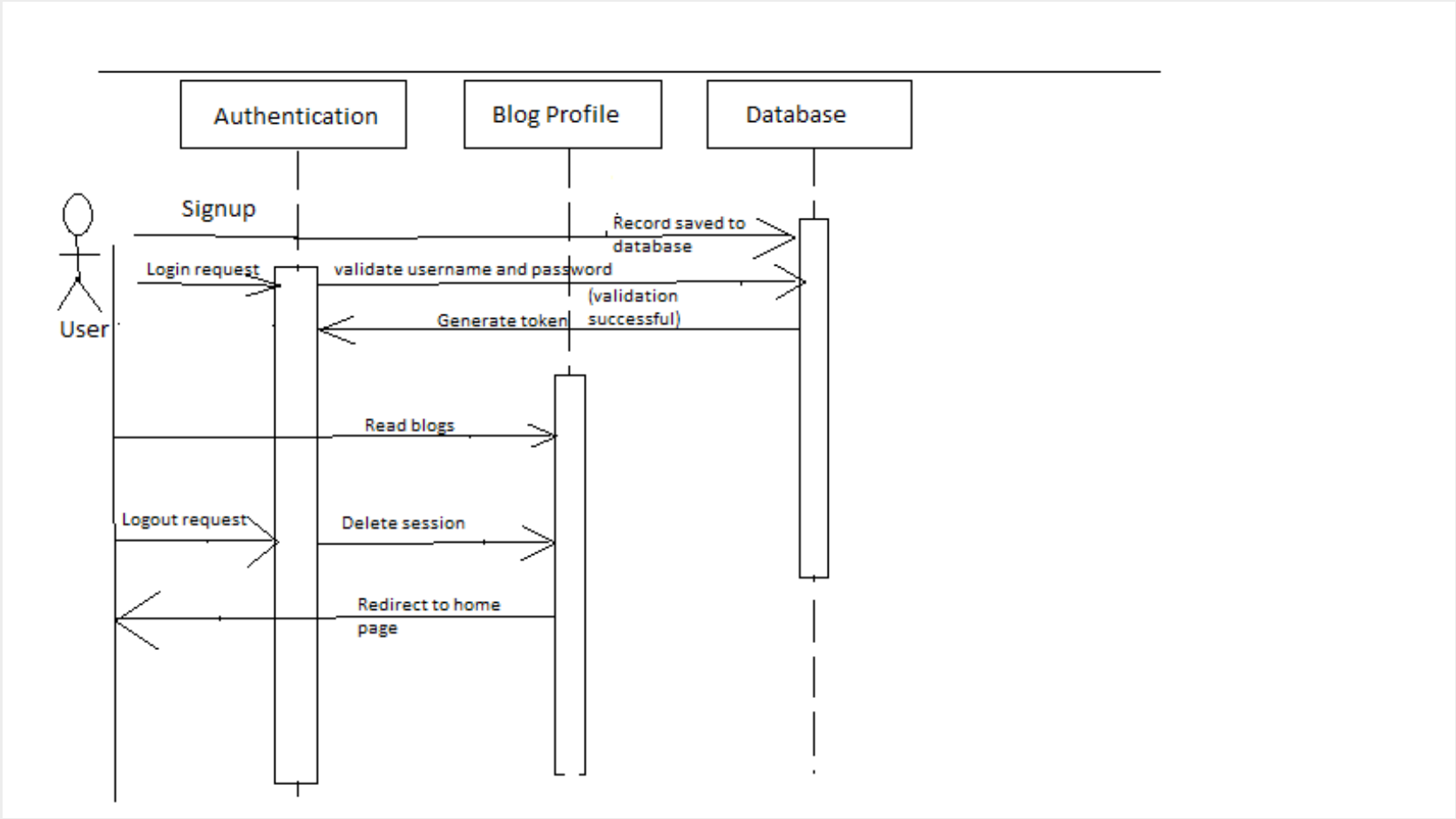
Archana kumari | September 27, 2022

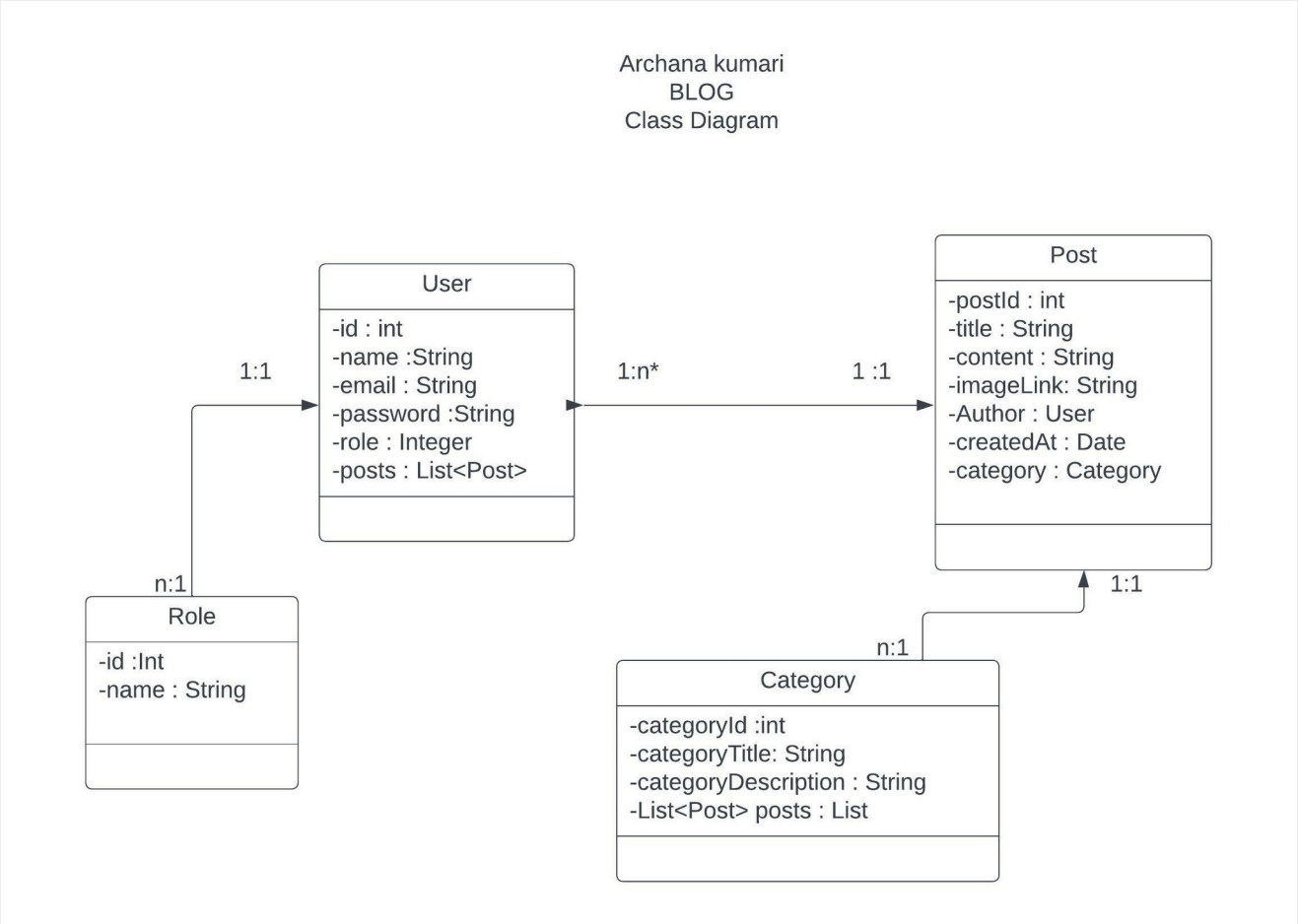


Sequence Diagram



Sequence Diagram




Archana kumari
BLOG
Class Diagram
User
-id : int
-name :String
-email : String
-password :String
-role : Integer
-posts : List<Post>
Post
-postId : int
-title : String
-content : String
-imageLink: String
-Author : User
-createdAt : Date
-category : Category
Role
-id :Int
-name : String
Category
-categoryId :int
-categoryTitle: String
-categoryDescription : String
-List<Post> posts : List
1:1
1:n*
1 :1
n:1
1:1
n:1

Thank-you