# User Service Documentation

**Overview**

This documentation provides a comprehensive overview of a Spring Boot-based User Service application. The service handles user registration and retrieval, using MongoDB for storage. It includes:

* Input validation
* Logging with Aspect-Oriented Programming (AOP)
* Exception handling
* Unit testing using JUnit 5

REST Endpoints

1. **Register a User** URL: POST /api/users

Query Parameter: notify=true (optional) Request Body:

{

"id": "6",

"name": "Mohini S", "age": 22,

"email": ["abhft@example.com",](mailto:abhft@example.com) "country": "France",

"phone": "564568839"

}

Validations:

* age must be > 18
* country must be "France"

1. **Retrieve User by ID** URL: GET /api/users/{id}

Description: Fetches user details by unique MongoDB ID.

**Technologies Used**

* + Spring Boot 3.1
  + Spring Data MongoDB
  + Hibernate Validator
  + AspectJ
  + Lombok
  + JUnit 5

**AOP (Aspect-Oriented Programming)**

Purpose: To separate cross-cutting concerns like logging from the core business logic. Implementation: Logs method calls in the controller layer using AspectJ annotations.

Example Pointcut:

@Before("execution(\* com.user.demo.controller.\*.\*(..))")

**MongoDB Configuration** Database Name: userdb Collection Name: users Connection String:

spring.data.mongodb.uri=mongodb://localhost:27017/userdb

View Data:

Use MongoDB Compass or the CLI:

db.users.find()

**API Testing with Postman**

**A)Register the User**

1. URL: <http://localhost:9192/api/users?notify=true>
2. Method :POST

3)Body :Json User data :

{

"id": "6",

"name": "Mohini S",

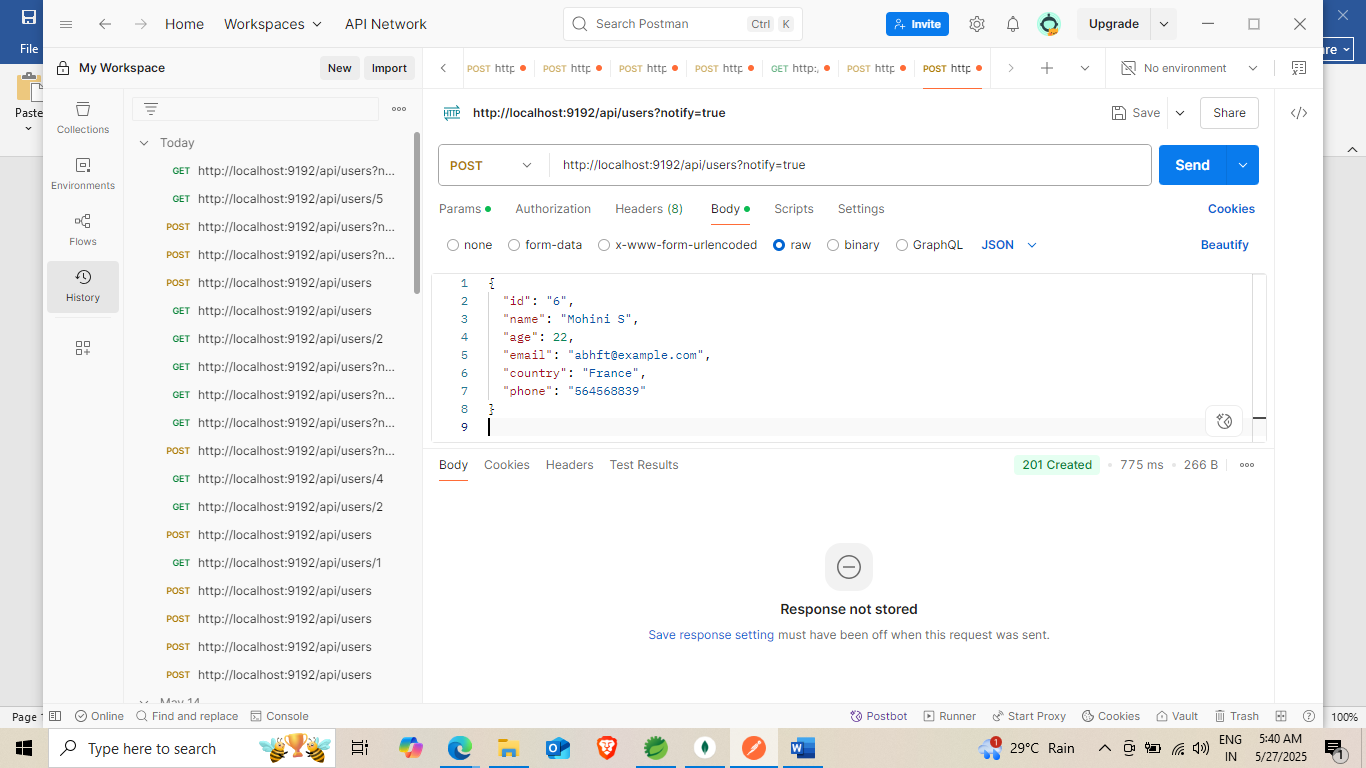
"age": 22,

"email": "abhft@example.com",

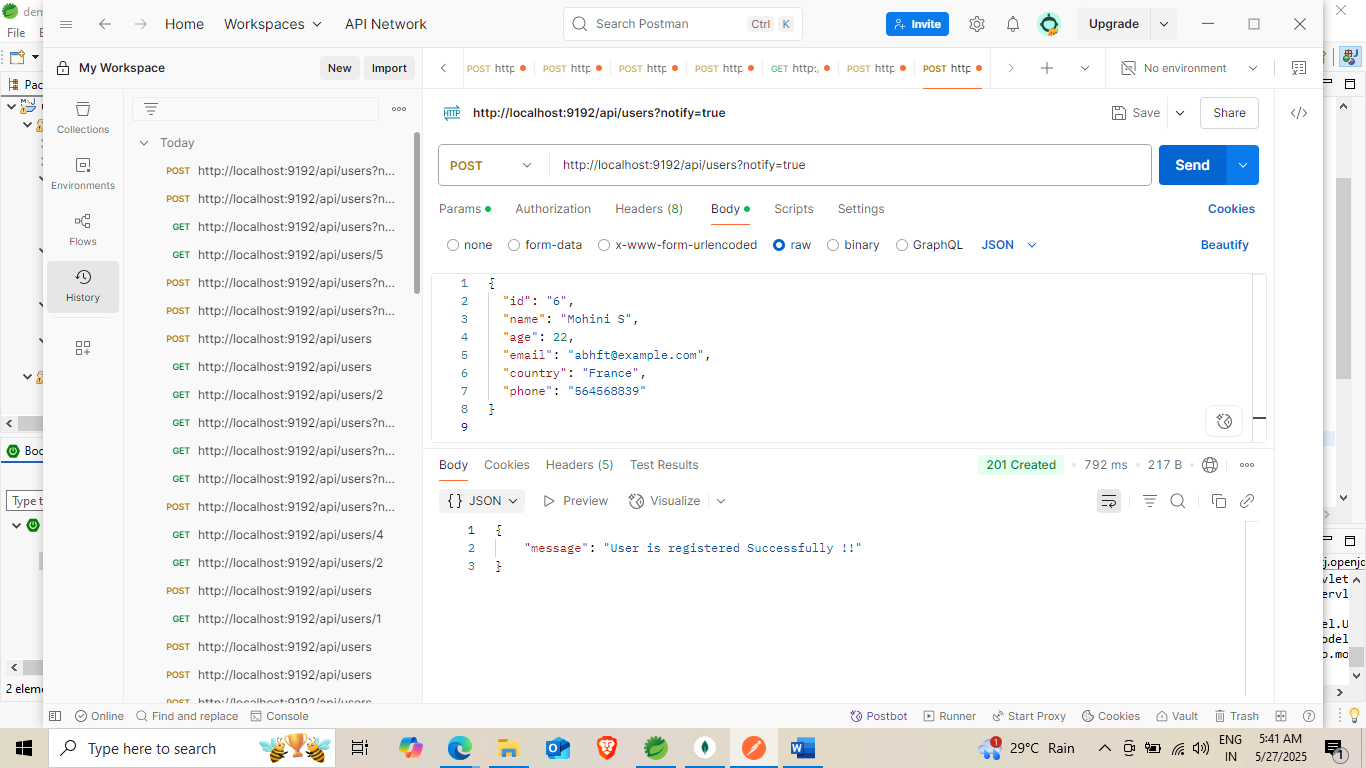
"country": "France",

"phone": "564568839"

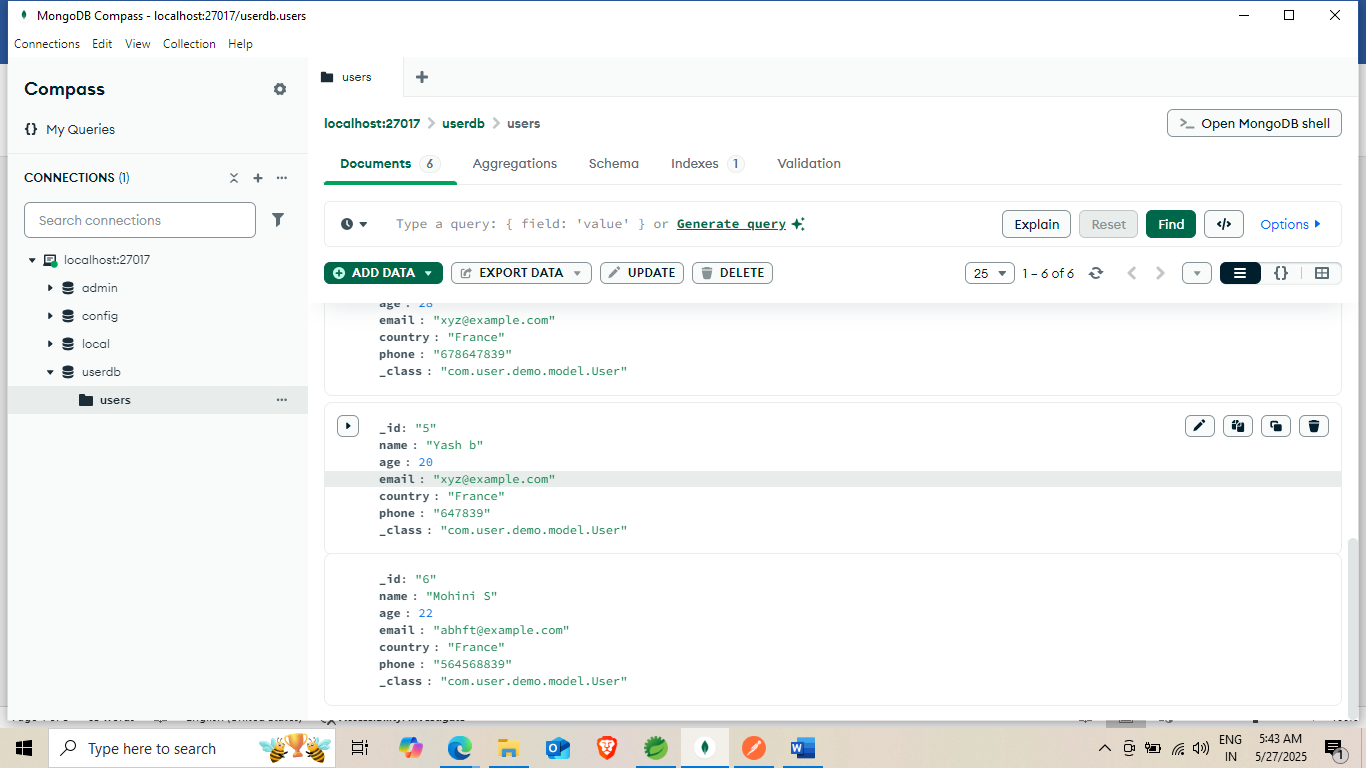
}



**By hitting the Rest API :Output**



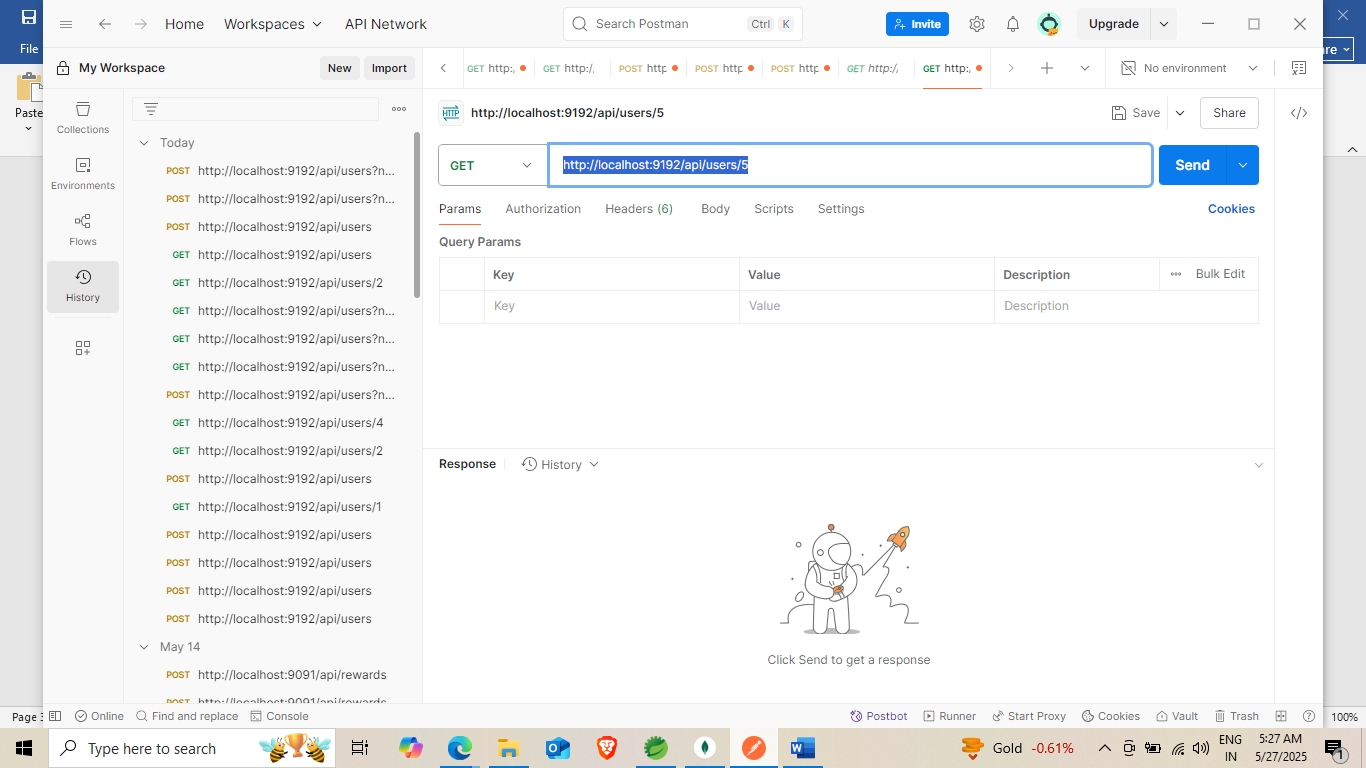
**MongoDB View:**



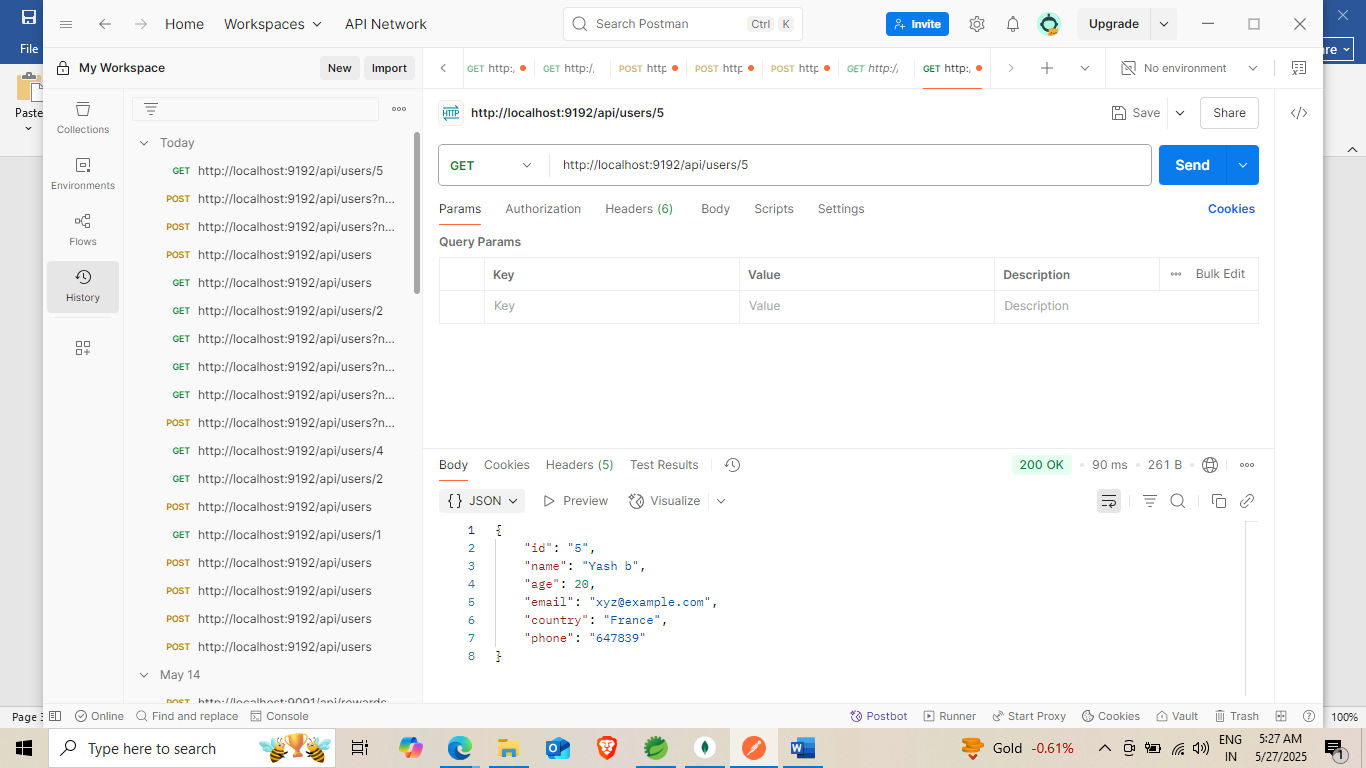
**B) Display the Details of Registered User**

URL: <http://localhost:9192/api/users/5>

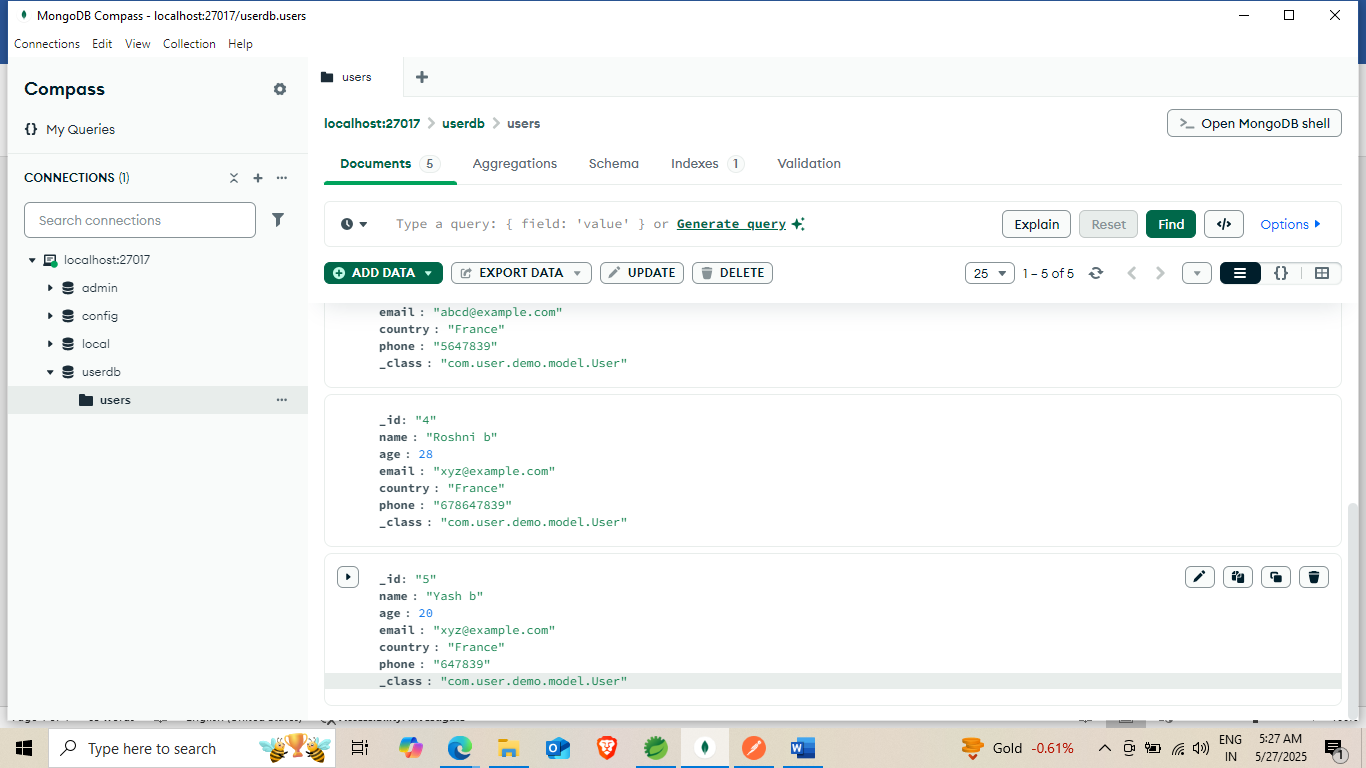
Method:GET



**Output :**

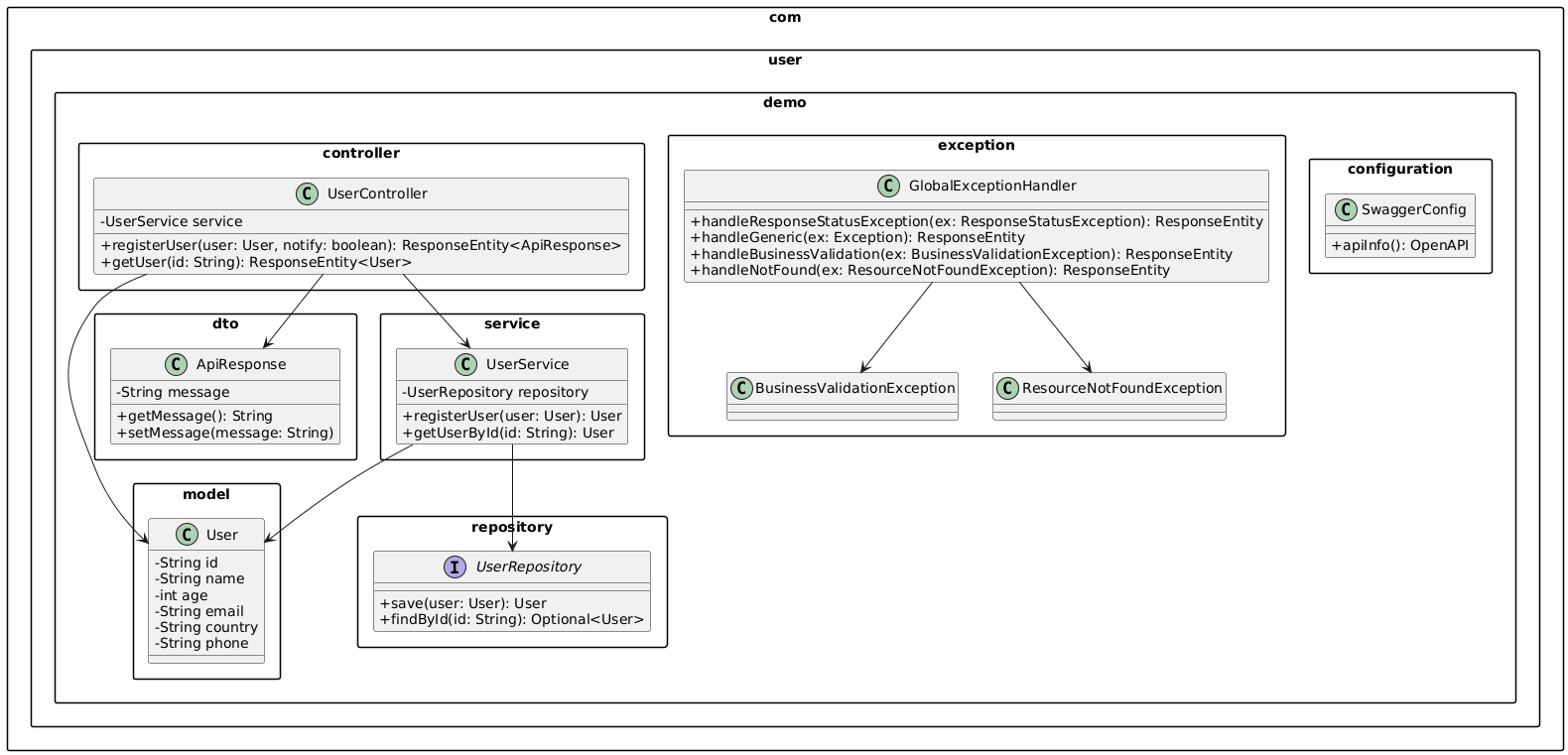


**MongoDB View:**



**UML Diagram**

A UML diagram for the User Service

****