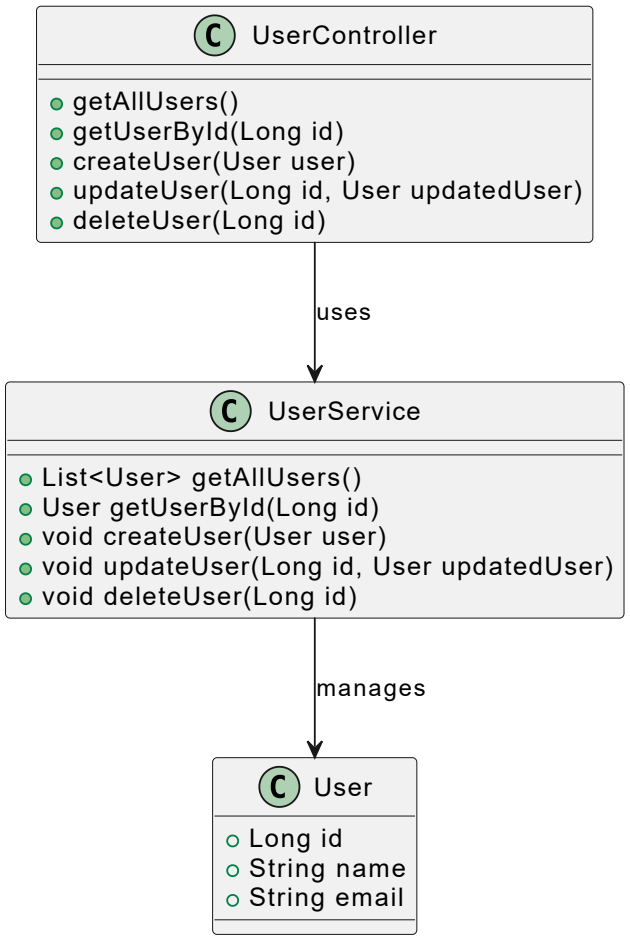
SASVA Project Scan Report

Git URL: https://github.com/rahulshirsat30/task-manager.git | Branch: main

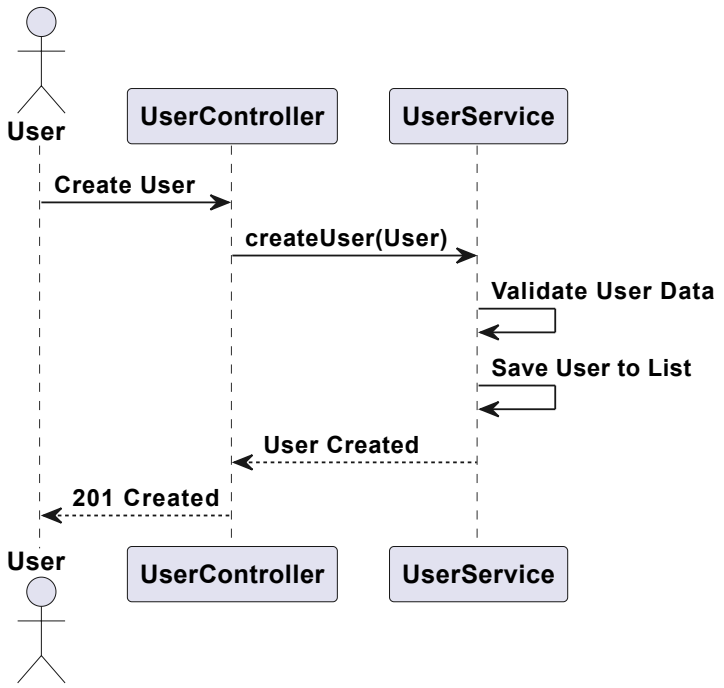
1) Class Diagram:

This class diagram represents the core components of the Task Manager application. It includes the User class, which defines the user entity with attributes like id, name, and email. The UserController class handles HTTP requests related to user management and interacts with the UserService class, which contains the business logic for managing user data. The relationships indicate that the UserController uses the UserService to perform operations on User entities.



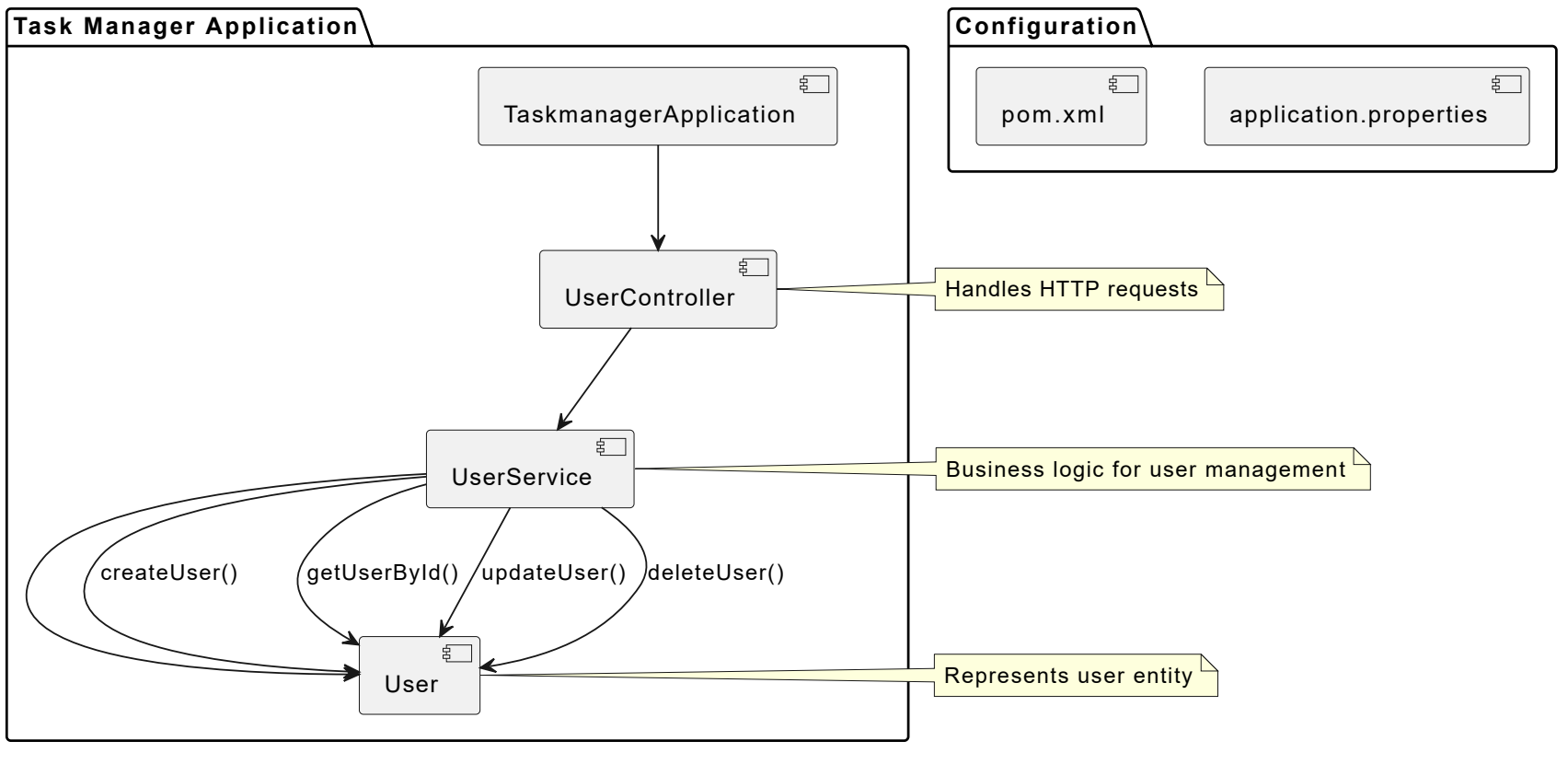
2) Sequence Diagram:

This sequence diagram illustrates the process of creating a user in the Task Manager application. The User initiates a request to the UserController to create a new user. The UserController then calls the UserService to handle the creation logic, which includes validating the user data and saving the user to an in-memory list. Once the user is successfully created, the UserService returns a confirmation to the UserController, which responds to the User with a '201 Created' status.



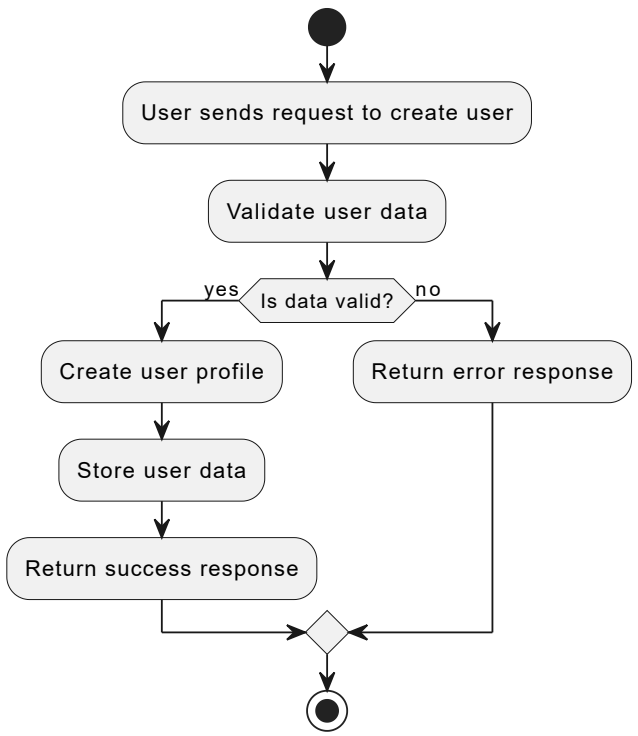
3) Logical Diagram:

This logical diagram represents the architecture of the Task Manager application. It shows the main components: the TaskmanagerApplication as the entry point, UserController for handling HTTP requests, UserService for business logic, and User as the data model. The diagram also includes configuration files like application.properties and pom.xml, indicating their role in the application setup.



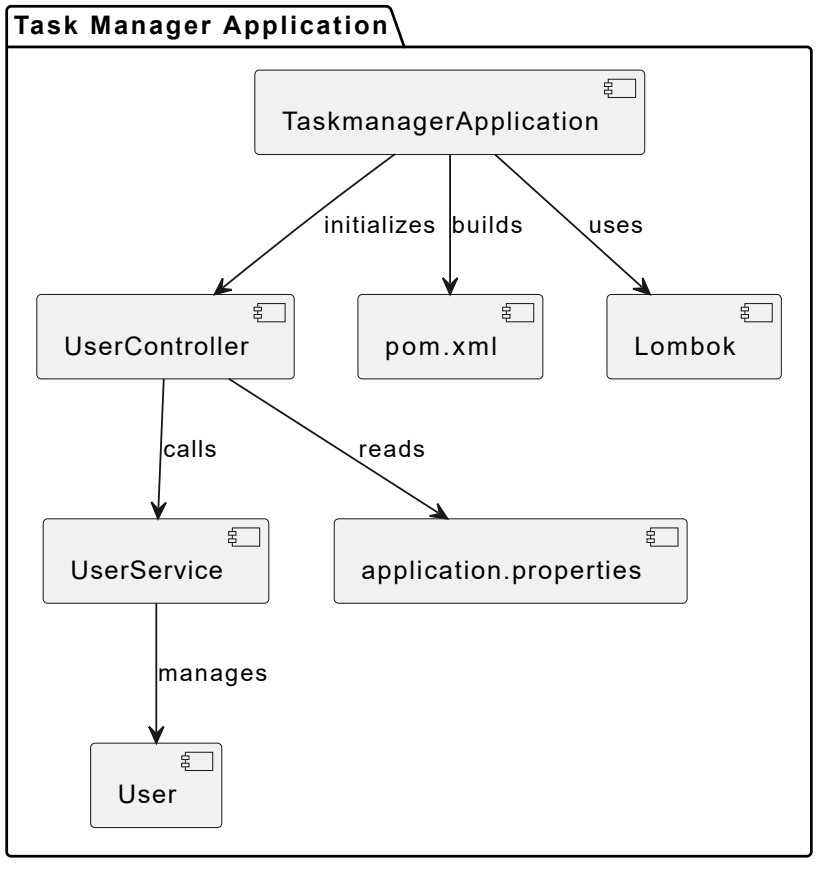
4) Activity Diagram:

This activity diagram illustrates the process of creating a user in the Task Manager application. It starts with the user sending a request to create a user profile. The application then validates the user data. If the data is valid, the user profile is created and stored, followed by a success response. If the data is invalid, an error response is returned.



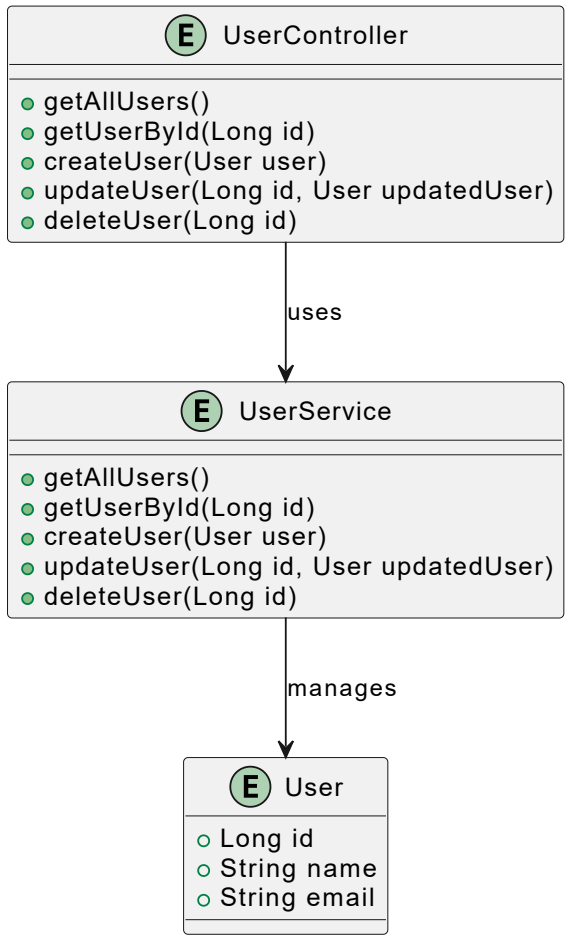
5) Component Diagram:

This component diagram illustrates the architecture of the Task Manager application. It shows the main components including the application entry point (TaskmanagerApplication), the UserController for handling HTTP requests, the UserService for business logic, and the User model representing user data. The application reads configuration from application.properties and is built using Maven (pom.xml). Lombok is utilized to reduce boilerplate code, enhancing code maintainability.



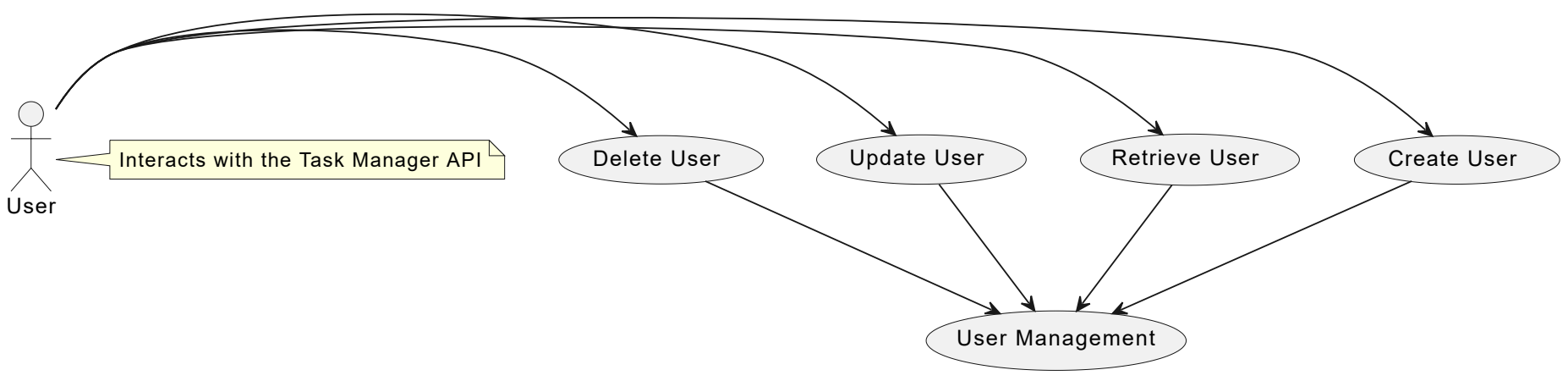
6) ER (Entity-Relationship) Diagram:

This ER diagram represents the core entities of the Task Manager application. The 'User' entity includes attributes for user identification and contact information. The 'UserController' entity defines methods for handling user-related HTTP requests, while the 'UserService' entity encapsulates the business logic for managing user data. The relationships indicate that the UserController uses the UserService to perform operations on User entities.



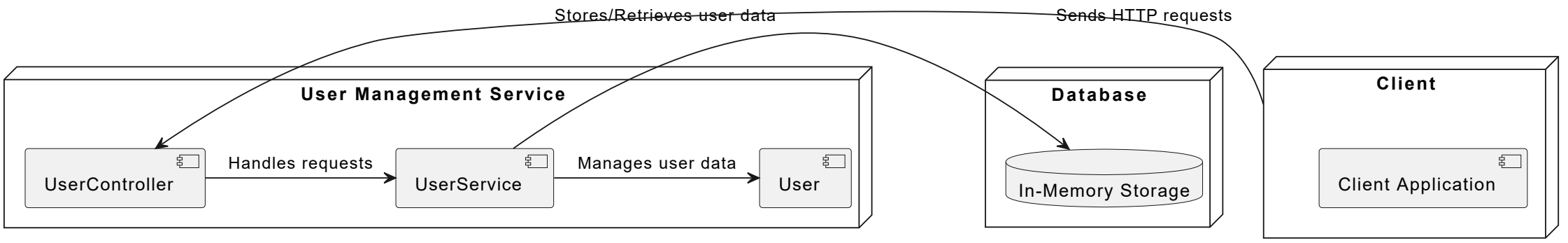
7) Use Case Diagram:

This Use Case Diagram illustrates the interactions between the User actor and the Task Manager application. The User can perform four main operations: Create User, Retrieve User, Update User, and Delete User, all of which are part of the User Management feature. The diagram emphasizes the user's role in managing user profiles through the application's RESTful API.



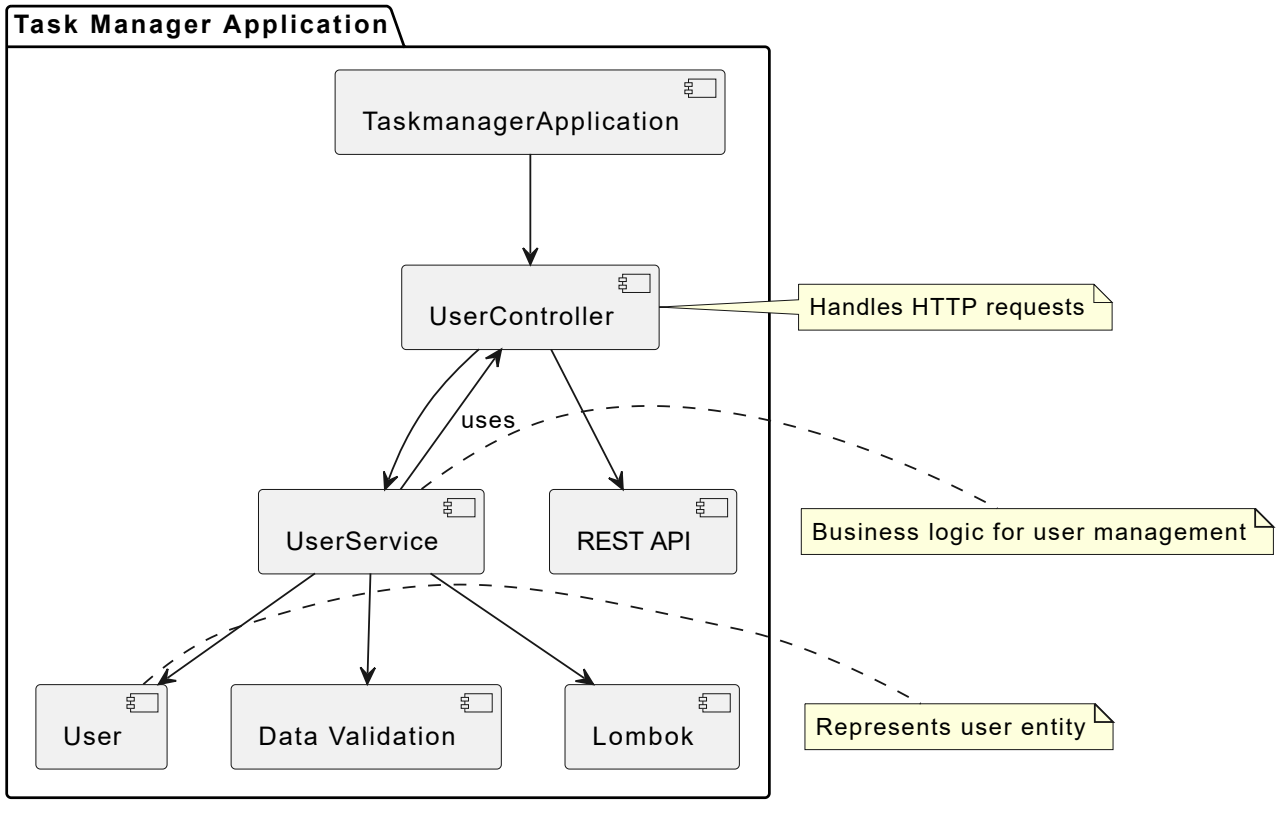
8) Deployment Diagram:

This deployment diagram illustrates the architecture of the Task Manager application. It shows the User Management Service, which consists of the UserController, UserService, and User model. The UserService interacts with an in-memory database for storing user data. The Client Application communicates with the UserController to send HTTP requests, enabling user management operations.



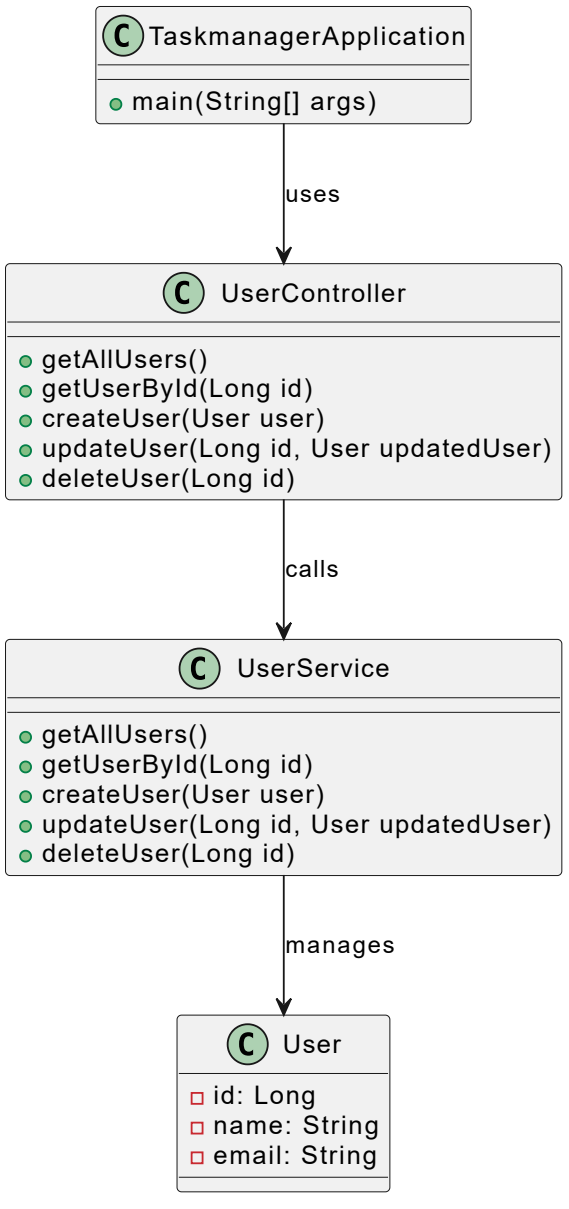
9) Composite Structure Diagram:

This Composite Structure Diagram illustrates the internal structure of the Task Manager application. It shows the main components: TaskmanagerApplication, UserController, UserService, and User. The UserController handles HTTP requests and interacts with the UserService, which contains the business logic for managing user data. The User entity represents the user data model. Additionally, it highlights the use of REST API for communication and the integration of data validation and Lombok for reducing boilerplate code.



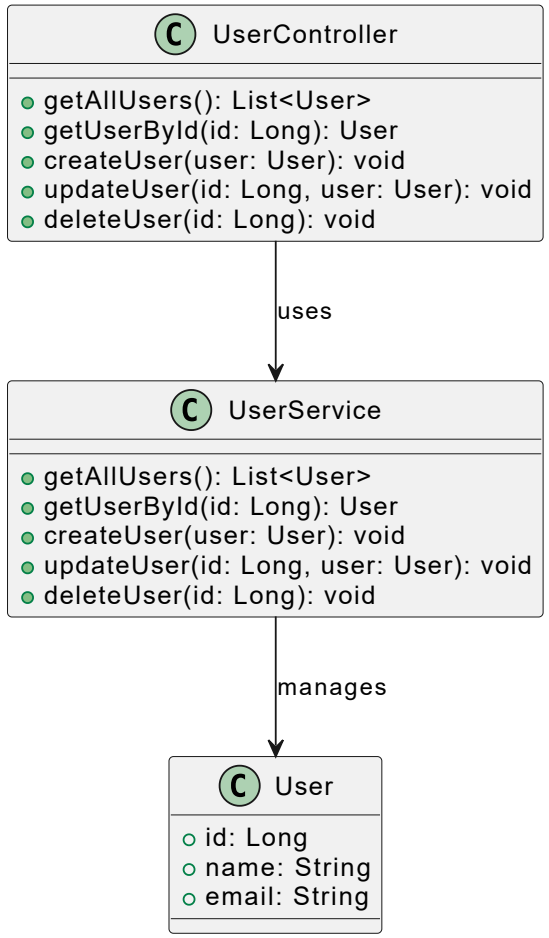
10) Object Diagram:

This object diagram represents the core components of the Task Manager application. It includes the main application class (TaskmanagerApplication), the UserController which handles HTTP requests, the UserService that contains business logic for user management, and the User class that represents the user entity. The relationships indicate that the application uses the controller, which in turn calls the service to manage user data.



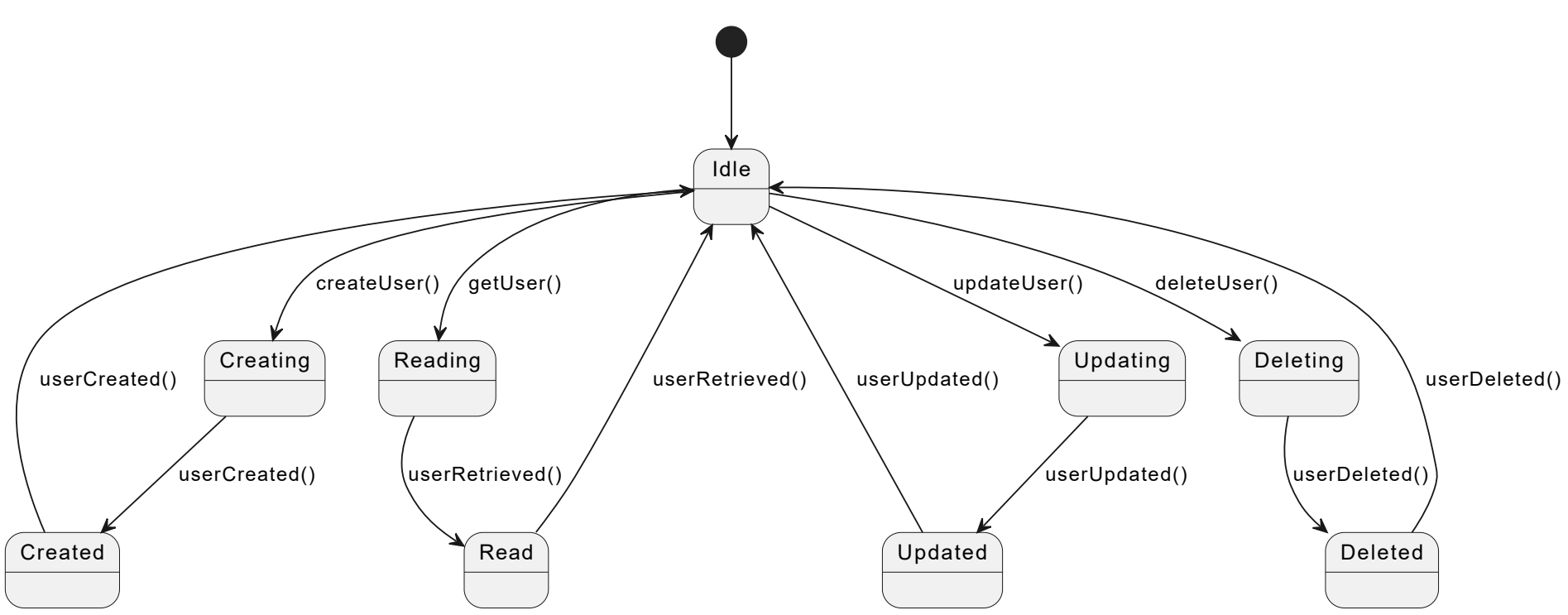
11) User Journey Map:

This User Journey Map illustrates the interactions between the User, UserController, and UserService components in the Task Manager application. The User entity represents the user data being managed, while the UserController handles HTTP requests related to user operations. The UserService contains the business logic for managing user data, and it is utilized by the UserController to perform actions such as creating, retrieving, updating, and deleting users.



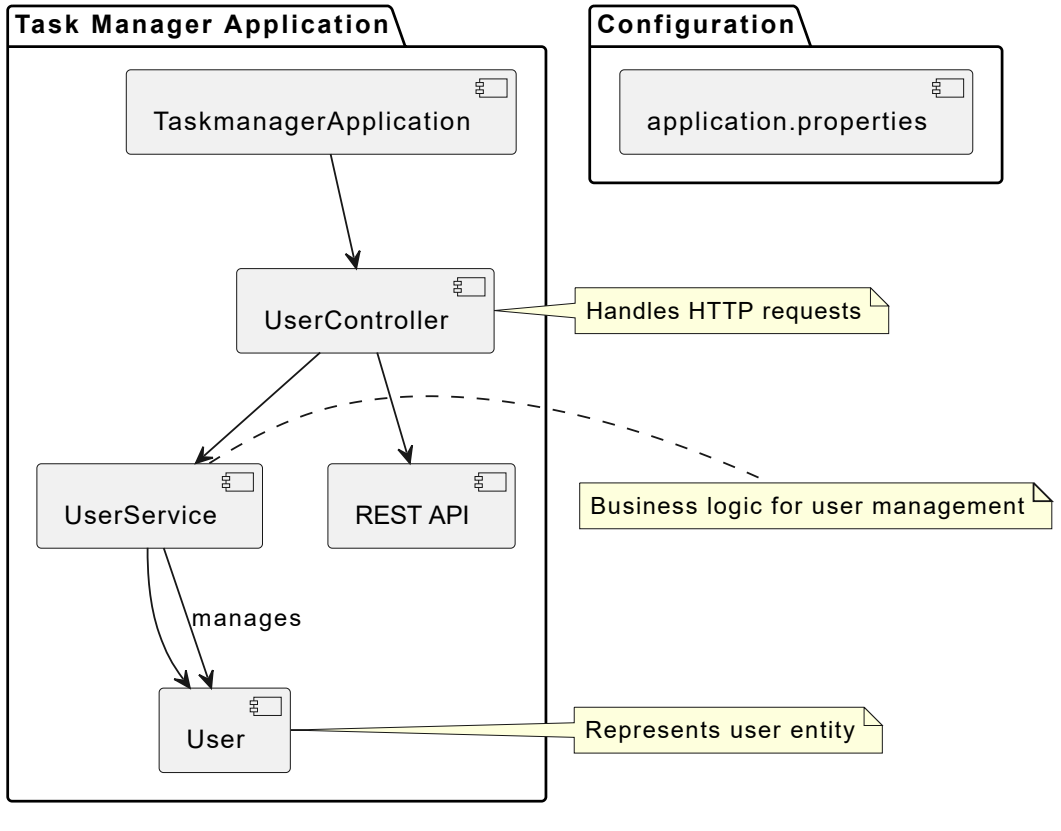
12) State Diagram:

This state diagram represents the various states of the Task Manager application related to user management. It starts from the Idle state, transitioning to Creating, Reading, Updating, and Deleting states based on user operations. Each operation leads to a corresponding state indicating completion, after which it returns to the Idle state, ready for the next operation.



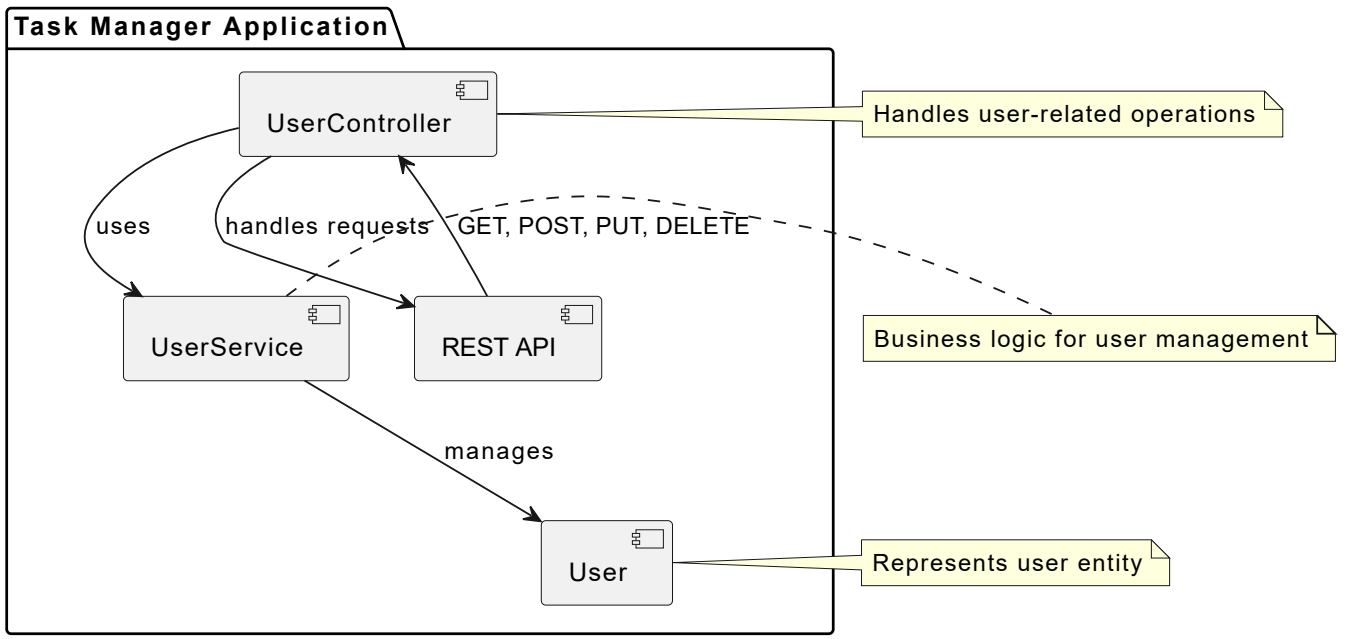
13) Architecture Diagram:

This architecture diagram illustrates the structure of the Task Manager application. It shows the main components: the TaskmanagerApplication as the entry point, UserController for handling HTTP requests, UserService for business logic, and User as the data model. The application is configured through application.properties. The arrows indicate the flow of control and data between these components.



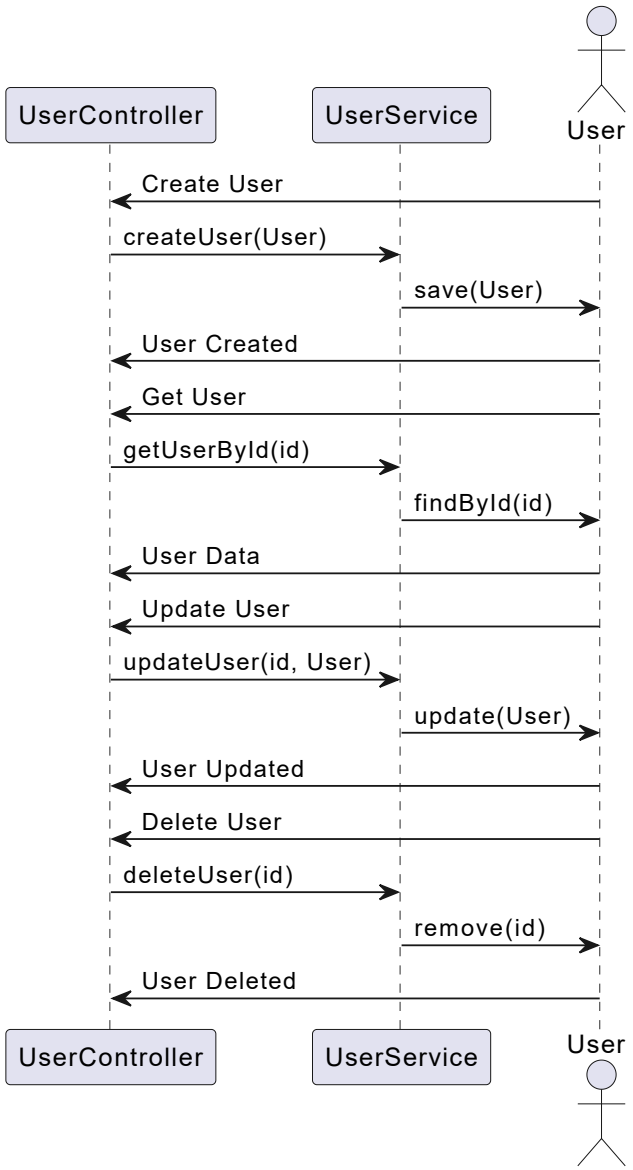
14) Workflow Diagram:

This workflow diagram illustrates the interactions within the Task Manager application. The UserController handles HTTP requests and interacts with the UserService to manage user data. The UserService contains the business logic for user management, while the User class represents the user entity. The REST API facilitates communication between clients and the UserController using standard HTTP methods.



15) Communication Diagram:

This communication diagram illustrates the interactions between the User, UserController, UserService, and User entities in the Task Manager application. It shows how a User can create, retrieve, update, and delete user profiles through the UserController, which communicates with the UserService to perform the necessary operations on the User entity.


UserController
UserService
User
Create User
createUser(User)
save(User)
User Created
Get User
getUserById(id)
findById(id)
User Data
Update User
updateUser(id, User)
update(User)
User Updated
Delete User
deleteUser(id)
remove(id)
User Deleted
UserController
UserService
User