**Design:**

**Domain Layer**

The domain layer is a collection of entity objects and related business logic that is designed to represent the business model.

**Controller Layer**

This layer is responsible for processing the user’s input and returning the correct response back to the user.

**Service Layer**

This layer is the middleware between Controller and Repository.

The service gathers data from controller, performs validation and business logic, and calling repositories for data manipulation.

**Repository Layer**

**Spring Data PagingAndSortingRepository** allows implementing Paging and Sorting -based repositories

We don't need to write any DAO code, but Spring Data **PagingAndSortingRepository** will generate an implementation automatically at runtime.

**Architecture:**

**Architecture Diagram:**

Diagram

Description automatically generated

**Browser:**

User request normally comes from the browser.

**Controller:**

Based on the browser request the Controller decide to which service method to call and delegate the request according.

**Service:**

Perform the necessary validation and business logic and calls the repository for data manipulation.

**Repository:**

Perform the necessary operation on Database and return the result.

**Database:**

Persists the data.