* Architecture:

A screenshot of a computer screen

Description automatically generated

Description:

Presentation layer:

User Interface: Classes to provide a graphical user interface and display data on the web page. HTML, CSS and JS files

Routing logic: Classes to handle routing from the front-end layer to the backend layer. JavaScript files.

Business Logic Layer:

Controller:

 /\*\*

This package contains classes for handling HTTP requests related to various functionalities within the project.

These classes utilize Spring annotations and interact with services to perform actions on the system's data. \*/

Model:

/\*\*

This package defines various entities used within the application to represent data

stored in the database. Each class is annotated with JPA annotations to define the

persistence behaviour.

\*/

Services:

/\*\*

This package defines the services available to interact with data.

Each service method defines functionalities related to CRUD operations, searching,

sorting and data retrieval.

\*/

Implementation:

/\*\*

This package contains implementation classes for the services defined in the

cs2815.project.service package. These classes typically interact with the repository

layer and perform business logic related to the data entities.

\*/

Data access layer:

Repo:

/\*\*

This package defines repositories for interacting with the database to manage different

entities within the application. Each repository extends JpaRepository, providing basic

CRUD operations and the ability to define custom queries using JPQL.

/\*

Data access frameworks:

This includes frameworks such as JPA and Spring Data that allow us to access data sources

Data sources:

These are the tables that we are accessing throughout the system.