IndusTree- Factory Management Platform
Final Report

Team Members: Nestor Lobo, Shreya Verma, Varsha Teratipally

Project Title: IndusTree (Factory Management Platform)

Project Summary: A platform to aid in the management of daily workings in a factory. It enables a centralized management system of tasks such as tracking machine stock levels, request stock. It also provides a management system for the Human Resources Department for creating employee and managing employee information and enabling employees to apply for leaves and claims, search for employees and edit personal information.

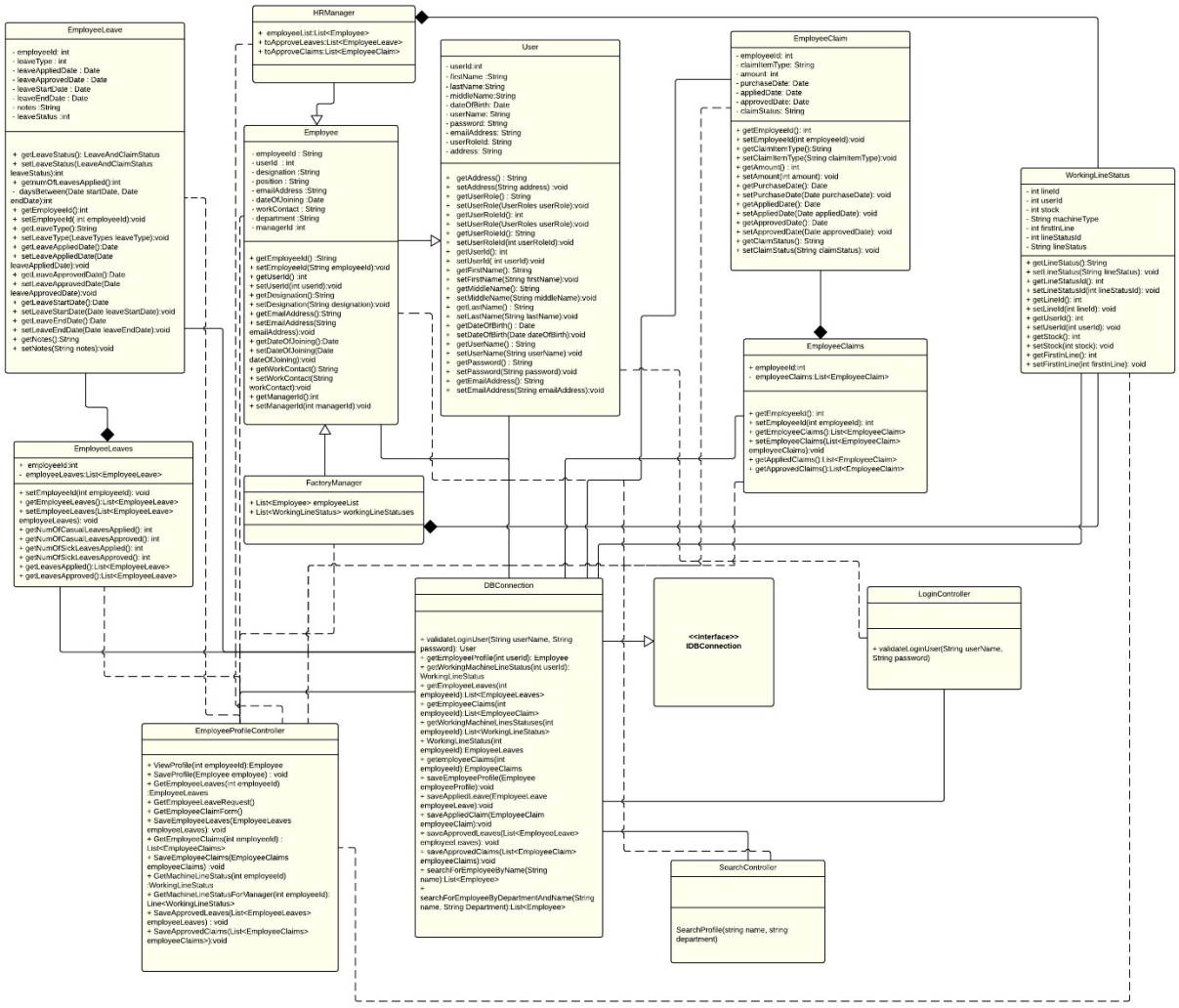
Features that are implemented:

ID	Title	Topic Area
UR – 001	Users logs into the application	Authentication
UR – 002	Creation of user's profile	Authentication
UR – 003	Users can view their profile	Profile
UR – 004	Users can edit their personal information	Profile
UR – 005(a)	Users can search for fellow employees for contact information based on department	Search
UR – 005(b)	Users can search for fellow employees for contact information based on first name, last name and department	Search
UR – 006(a)	Users can click on the displayed search results and can navigate to the user's public profile	Search
UR – 006(b)	Users can click on the displayed search results and access public and classified information of all employees	Search
UR – 007(a)	Users can view their machine line status	Line Status
UR – 008	Users can apply for leave and claims	Leaves and Claims
UR – 009	Users can approve the leaves and claims applied	Approvals Central
UR – 010	Users can request for stock	Line Status
UR – 011	Users can logout	Authentication
UR – 012	Users can reset their passwords	Authentication
UR – 013	User can deactivate the employee profiles	Authorization
UR – 014	User can provide access rights to the employee based on the request	Authorization

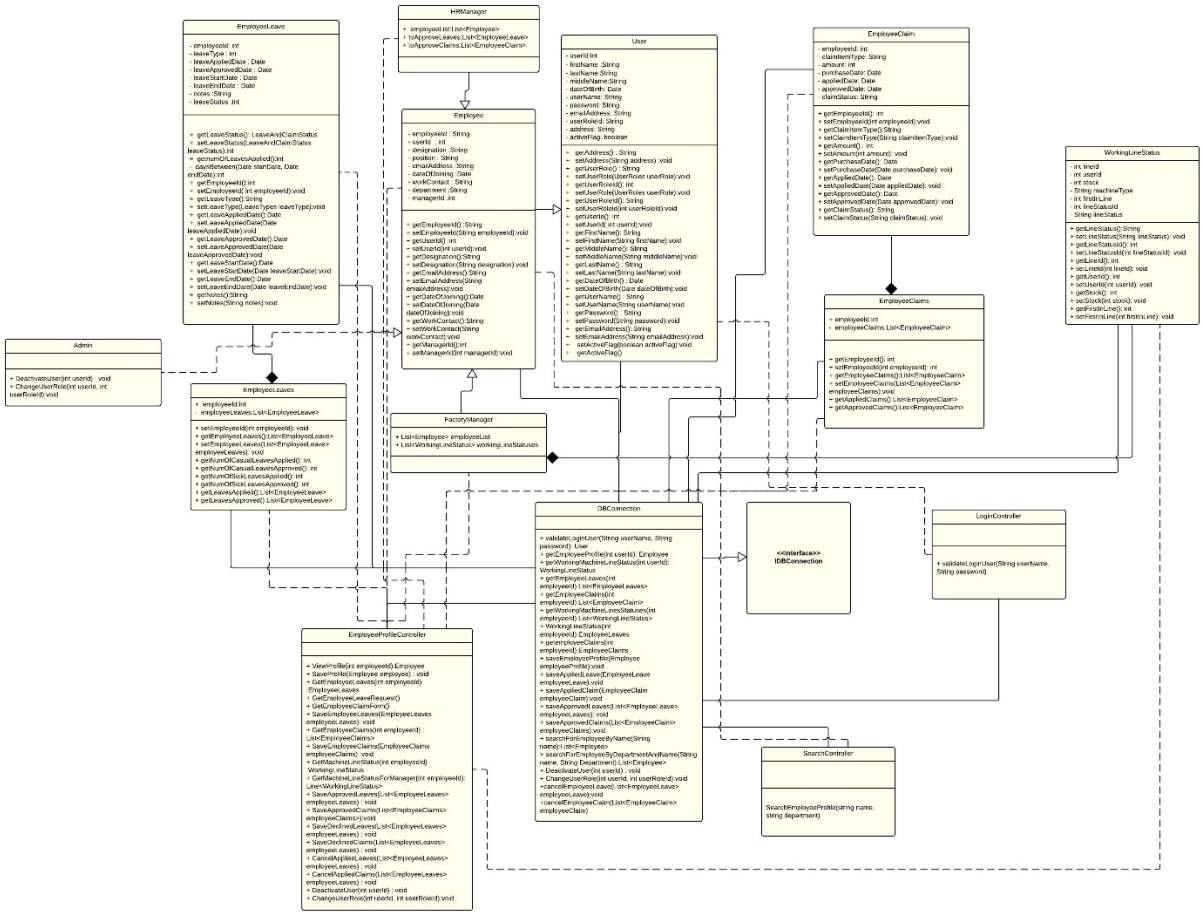
Features that are not implemented:

ID	Title	Topic Area
UR – 007(b)	Users can view machine line status of all lines	Line Status

Part-2 Class Diagram:



Final Class Diagram:



The changes made in the final class diagram from Part-2 are minimal and are as follows:

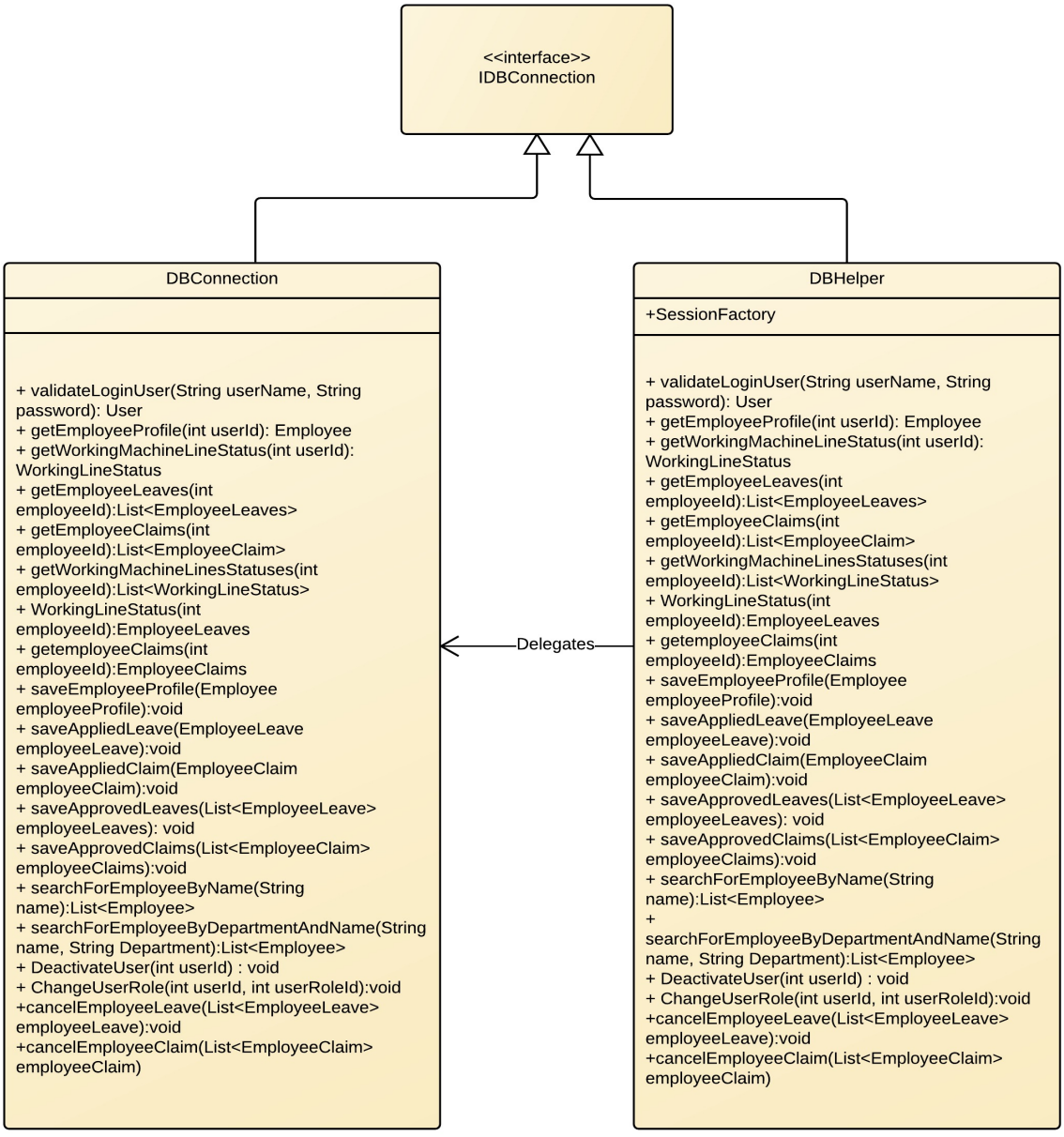
- The functionality of UR-007(b): User can view machine line status of all lines was not implemented. The dependency node connecting the HR Manager and the working line status is removed from the new class diagram. Rest all is same.

How did the design help in the development?

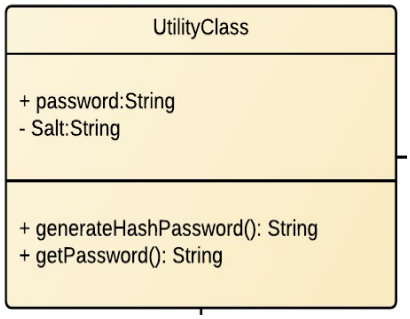
- With the help of the class diagram, we could visualise a static view of the application. With the illustrations of relationships and dependencies among all classes in the diagram, it was easier to conceptualize the model under consideration. This aided in simplifying the implementation.
- With the help of Use-Case Diagrams, we could understand the requirements comprehensively, the step-by-step flow of events, possibilities for error, the system dependency and the interaction of user with the system and handling authorizations and validations associated with each case.
- With the help of sequence diagrams, we could model the system interactions over time. We could easily identify how objects work and operate with each other along with its sequence.

Design Patterns in the final prototype:

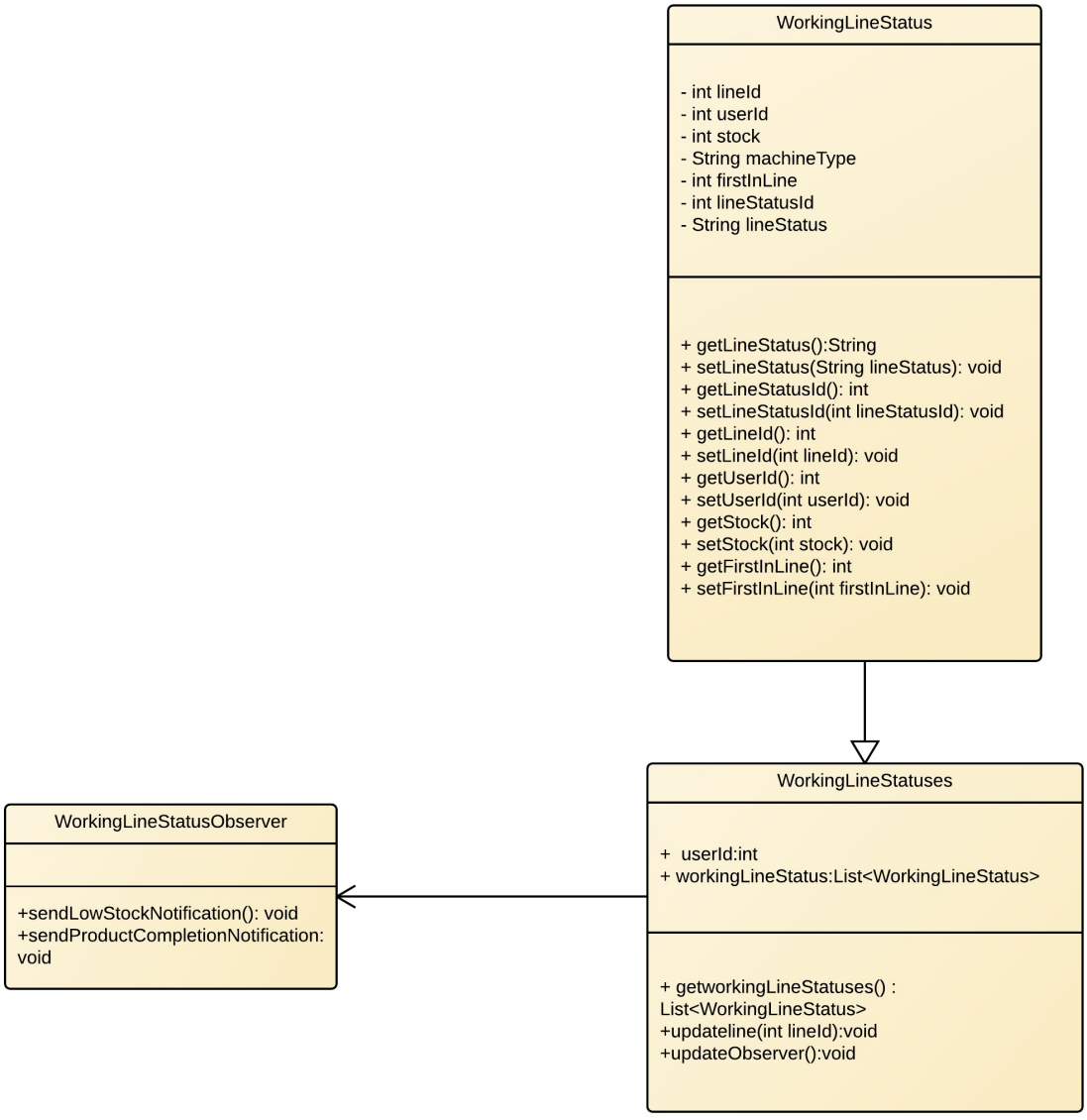
- **Proxy Pattern**
  The application connects to the database using a proxy class called DBHelper and the actual implementation is hidden in the DBConnection class. These classes implement the interface IDBConnection where all the method signatures are declared.



- **Strategy Design Pattern**
  The Salt and Hash methods have been implemented to aid in securing user's passwords. Additional data is added to the password and then hashed to encrypt the passwords.

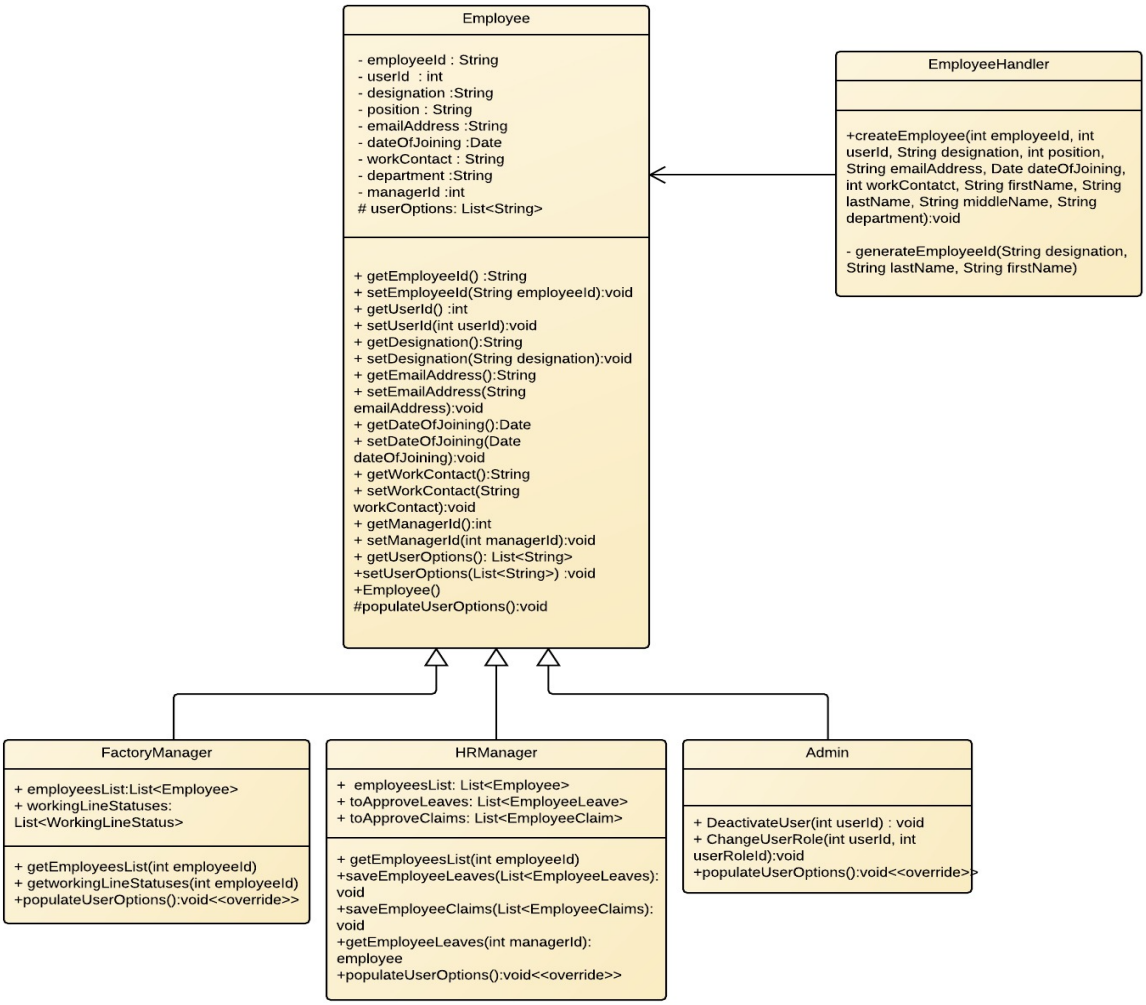


- **Observer Pattern**
  For all the machine lines in the manufacturing company, the application displays real-time stock. This gets updated every five minutes. When the stock is low, the Factory Manager is notified, enabling him to order further stock. The implemented Subscriber class then sends a notification to the Factory Manager.



- **Factory Design Pattern**

  In the application, employees of every designation are provided with their own specific functionalities. The Employee class specifies the generic behavior common to all employees, and the FactoryManager, HRManager and Admin classes specify the behavior of the employees based on their specific requirements. The EmployeeHandler class is used to create employee objects based on the designation.

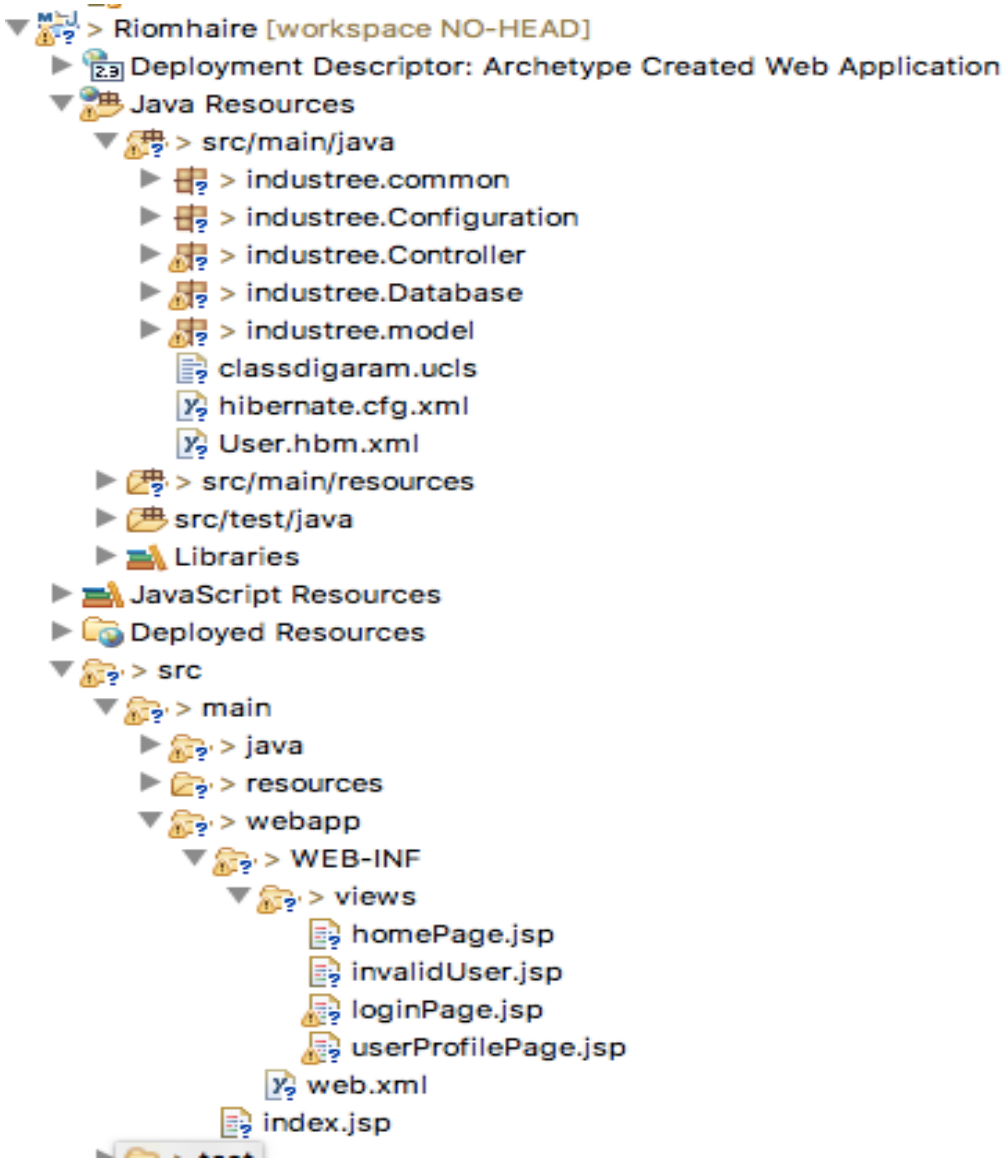


- **MVC Design Pattern**

  MVC design pattern is made up of the Model responsible for handling the data, the Controller responsible for responding to the user input and performing the data operations on the Model objects, and the View is responsible for the presentation of the data.

  In the IndusTree application,

1. Model contains the java bean classes
2. Controller is the HomePageController
3. Views are the jsp files



Overall Learning Process:

The pre-implementation design phase is perhaps the most crucial stage in the development process because ideas and issues noticed during this phase shape most decisions that will be taken in the development.
Here, having the aid of design tools like the UML and Class Diagrams can help in the visualisation of the final implementation and flow of the system. After the design, periodic introspection of the implementation helps keep the project on the right track.
The use of Hibernate made data retrieval from the database simple. During the implementation, Design Patterns aid in streamlining the code to the best possible design and knowledge of Anti-Patterns help in refactoring code.