**ASSIGNMENT -1**

**REPORT ON :-Insurance Broker Management System. Part I**

**Project Description:**

The **Insurance Broker Management System** allows brokers to manage customer data and insurance policies effectively. The system performs CRUD operations for both customer and policy entities and stores all data in JSON format for easy retrieval and persistence. The application is built using Java Servlets, which interact with the business logic through a well-structured service and repository layer.

**Design Decisions:**

* Architecture: The system follows a standard J2EE Web Application Architecture, with proper separation between Servlet, Service, and Repository layers. This architecture promotes maintainability and scalability.
* File Storage Format: The system uses JSON files for data storage. Both customer and policy data are persisted in JSON format, allowing for structured, human-readable data storage.
* Repository Design Pattern: The repository pattern is implemented to separate the logic responsible for accessing the database (or file system) from the business logic. This approach makes the system easier to maintain and test.

**Features:**

1. Customer Management:
   * Add, update, delete, and view customer data.
   * Customer data is stored in JSON format, and all CRUD operations are managed through the repository pattern.
2. Policy Management:
   * Add, view, and assign policies to customers and brokers.
   * Policies are also stored in JSON format, ensuring consistency with the customer data storage.
3. Thread Safety:
   * The application implements synchronized blocks in servlets to ensure thread safety during concurrent access to shared resources (e.g., customer and policy data).
4. Servlet Mappings:
   * All servlet mappings are done in the web.xml deployment descriptor, adhering to the guidelines of the assignment.

**UML Diagrams:**

1. Class Diagram:

A diagram of a string system

Description automatically generated

1. Sequence Diagram:

A diagram with text and arrows

Description automatically generated with medium confidenceA screenshot of a diagram

Description automatically generatedA diagram of a policy

Description automatically generated

**Screenshots:**

Home page

**A screenshot of a computer

Description automatically generated**

About page

**A screenshot of a computer

Description automatically generated**

Contact Us page:-

**A screenshot of a computer

Description automatically generated**

Register page

**A screenshot of a computer

Description automatically generated**

Login page

**A computer screen shot of a login page

Description automatically generated**

Dashboard page

**A screenshot of a computer

Description automatically generated**

Customer Management page

**A screenshot of a computer

Description automatically generated**

Add new customer functionality

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

View functionality that displays the policy name ,broker Id and Premium Amount for the customer in Assigned policies table

**A screenshot of a computer

Description automatically generated**

Assign Policy to customer functionality by clicking on Assign Policy button

A screenshot of a computer

Description automatically generated

Edit functionality for customer

**A screenshot of a computer

Description automatically generated**

Delete functionality for customer

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Policy Management page

**A screenshot of a computer

Description automatically generated**

Add new policy functionality

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Edit functionality

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Delete functionality

**A screenshot of a computer

Description automatically generated**

A short video on concurrent access handling without data corruption displaying thread-safe resource management

[Double click to play]



**Collaboration Details:**

The team collaborated effectively using GitHub. All members contributed to different aspects of the project .

Team members and the tasks assigned to each of the member:-

**Back-end:-**

CRUD operations for the Customer management : Samruddhi Chavan

Assign policies to customers : Samruddhi Chavan

CRUD operations for the policies management : Sruthi Jayaprakash Pandiath

**Front-end:-**

Home , About , Contact , Login, Register and Dashboard page :- Rajat Sachdeva

Add new customer, Add new policies and delete functionality in both pages :- Rajat Sachdeva

View , Assigned policies ,Assign policy to customer and Edit functionality in both pages:-Shrabani Sagareeka

**UML Diagrams (Class and Sequence):-** Manpreet Kaur Gulati

**Report and Testing:-** Shrabani Sagareeka

We used GitHub to track issues, assign tasks, and manage the project’s progress.

Conflicts were resolved by fetching the latest changes and resolving merge conflicts locally. The final version of the project was pushed to the repository after thorough testing.