**Title: Insurance Management System**

**Name:**

Contents

[**Introduction** 2](#_Toc152964694)

[**System Overview** 2](#_Toc152964695)

[**Project Structure** 3](#_Toc152964696)

[**Getting Started** 3](#_Toc152964697)

[**System Features** 4](#_Toc152964698)

[**Screenshots** 5](#_Toc152964699)

[**Conclusion** 7](#_Toc152964700)

[**Appendices** 7](#_Toc152964701)

# **Introduction**

The Insurance Management System is a JavaFX application designed to efficiently and effectively manage insurance policies and user information. It employs a Model-View-Controller (MVC) architecture, ensuring modularity and clarity in structure and functionality.

# **System Overview**

The system is segmented into three primary sections: Models, Controllers, and Views. Models handle data and database operations. Controllers serve as intermediaries between Models and Views. Views are concerned with the graphical user interface.

This application supports distinct roles, such as agents and customers. Agents have extended capabilities, including managing policies and viewing customer information, whereas customers primarily interact with their policy data.

# **Project Structure**

The project is organized as follows:

1. **src/controllers/**
   * **AgentController.java**: Manages agent interactions, like policy management.
   * **CustomerController.java**: Handles customer-specific actions.
   * **CustomerPolicyController.java**: Manages policy operations from a customer's perspective.
   * **LoginController.java**: Handles login functionality.
   * **PolicyController.java**: Oversees all policy-related actions.
   * **SignupController.java**: Manages user registration process.
2. **src/models/**
   * **Agent.java**: Represents the agent entity.
   * **Customer.java**: Represents the customer entity.
   * **Dao.java**: Facilitates database interactions.
   * **DBConnect.java**: Manages database connectivity.
   * **Policy.java**: Represents insurance policies.
3. **src/views/**
   * **AgentView.fxml**, **CustomerPolicyView.fxml**, **CustomerView.fxml**, etc.: These FXML files define the layout and elements of the user interface for various views in the application.

# **Getting Started**

To launch the application, run **Main.java**. The user will be greeted with a login screen. Sample credentials for different user roles are:

* **Agent Access:**
  + Username: agent
  + Password: agent
* **Customer Access:**
  + Username: customer
  + Password: customer

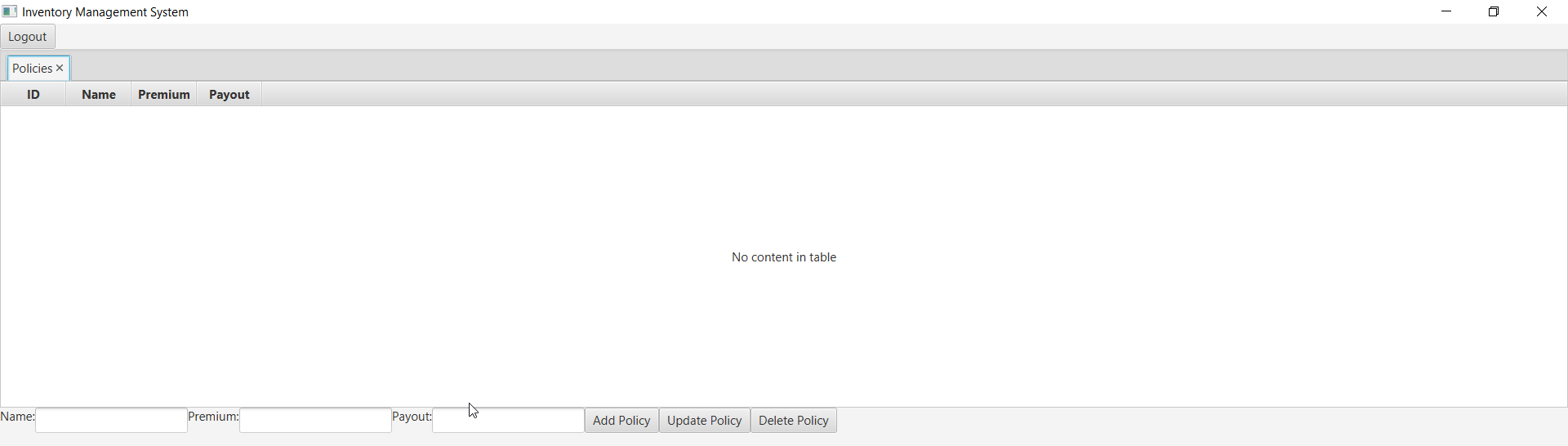
# **System Features**

* **User Authentication:** Secure login mechanism for agents and customers.
* **Role-Based Functionalities:** Distinct functionalities based on user roles.
* **Policy Management:** Agents can add, update, and delete policies.
* **Customer Interaction:** Customers can view and interact with their policies.
* **User-Friendly Interface:** Intuitive and responsive GUI for all user actions.

# **Screenshots**

A screenshot of a login box

Description automatically generated



Successful login for a customer

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

Successful login for an agent user

A screenshot of a computer

Description automatically generated

All 10 records for Policies added to database.

A screenshot of a computer

Description automatically generated

Update performed by the agent for the first record

Deletion of the last record by the agent

A screenshot of a computer

Description automatically generated

# **Conclusion**

The Insurance Management System is a robust and user-friendly JavaFX application tailored for managing insurance policies and user interactions. Adhering to the MVC architecture, it offers clear separation of concerns, making it modular and scalable. The system effectively caters to different user roles, providing a comprehensive solution for insurance management needs.

# **Appendices**

**Class Diagram**

A screenshot of a computer

Description automatically generated