**INSURANCE BROKER MANAGEMENT SYSTEM REPORT**

**J2EE Web Components**

**ITE-5332-0NA**

**Table of Contents**

Contents

[**Table of Contents** 2](#_Toc184550225)

[1. Introduction 2](#_Toc184550226)

[1.1 Project Overview 2](#_Toc184550227)

[1.2 Objectives 2](#_Toc184550228)

[2. Problem Statement 3](#_Toc184550229)

[3. Proposed Solution 3](#_Toc184550230)

[4. Technologies Used 3](#_Toc184550231)

[5. Database Schema 3](#_Toc184550232)

[6. Database Structure 3](#_Toc184550233)

[6.1 ER Diagram 3](#_Toc184550234)

[6.2 Relational Model 4](#_Toc184550235)

[7. Features 5](#_Toc184550236)

[8. UML Diagrams 5](#_Toc184550237)

[8.1 Class Diagram 5](#_Toc184550238)

[8.2 Sequence Diagram 5](#_Toc184550239)

[9. Screenshots 5](#_Toc184550240)

[10. Conclusion 13](#_Toc184550241)

# 1. Introduction

## 1.1 Project Overview

The **Insurance Broker System** is a web-based solution designed to streamline insurance management tasks for brokers. It includes customer, policy, and claims management, while leveraging scalable architecture and efficient persistence mechanisms.

## 1.2 Objectives

* Simplify the management of insurance data.
* Automate claims processing and tracking.
* Provide robust reporting features.
* Ensure secure and concurrent data handling.

# **2. Problem Statement**

Traditional methods for managing insurance records are prone to errors, inefficiencies, and limited scalability. This project addresses these issues by implementing an automated and scalable system for brokers.

# 3. Proposed Solution

The system provides:

* Centralized management of customer, policy, and claim data.
* Automated reporting and analytics for brokers.
* Thread-safe data operations using a Repository pattern.

# 4. Technologies Used

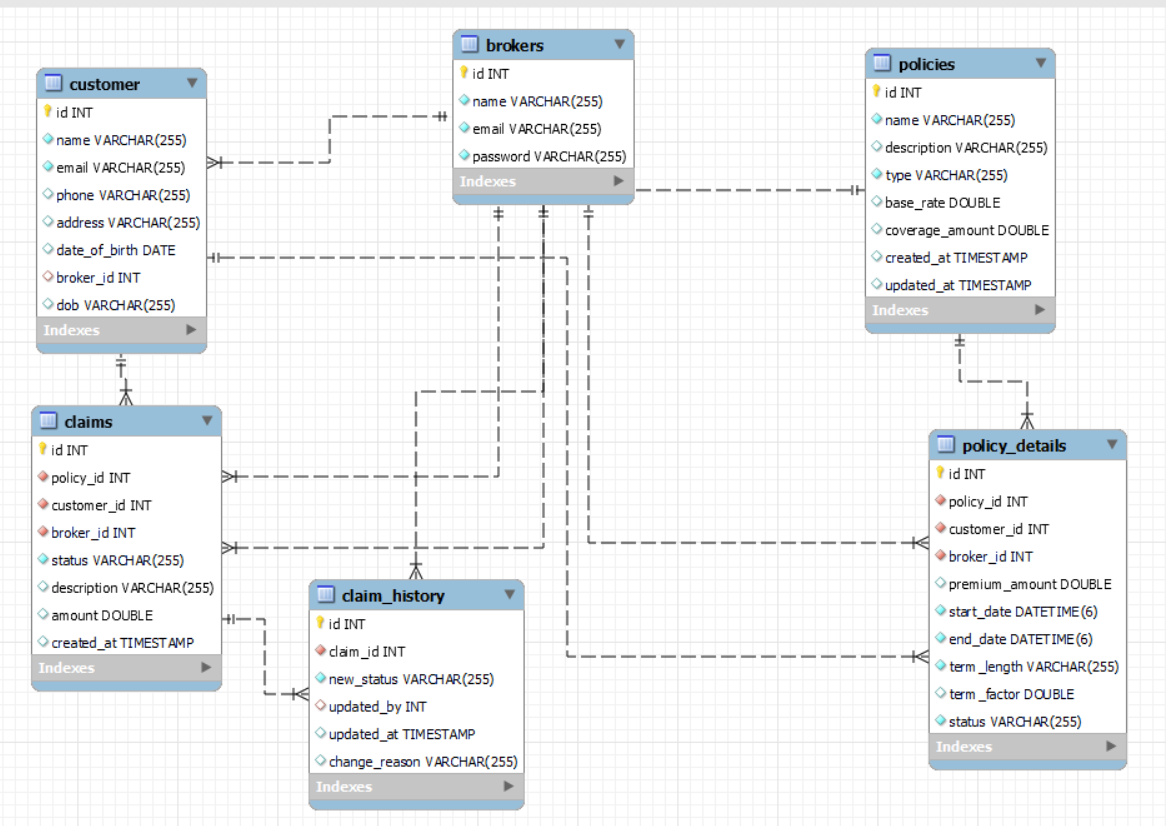
* **Frontend:** HTML, CSS, JavaScript
* **Backend:** Java Servlets, J2EE Architecture
* **Database:** MySQL with Hibernate
* **Containerization:** Podman
* **Patterns:** Repository Design Pattern

# **5. Database Schema**

* **Brokers Table**: Stores broker data, including name, email, and encrypted passwords.
* **Customers Table**: Manages customer data, such as personal details, contact information, and associated brokers.
* **Policies Table**: Defines insurance policies with details like type, base rate, and coverage amount.
* **PolicyDetails Table**: Links policies to customers and brokers, including term length, premium amount, and status.
* **Claims Table**: Tracks insurance claims with details like status, description, and claim amount.
* **ClaimHistory Table**: Logs updates for claims, including new status, updated by whom, and reasons for changes.

# 6. Database Structure

## 6.1 ER Diagram



## 6.2 Relational Model

The relational model includes the following tables:

**Brokers Table**

* **Columns**: id, name, email, password
* **Description**: Stores broker credentials

**Customers Table**

* **Columns**: id, name, email, phone, address, date\_of\_birth, broker\_id
* **Description**: Manages customer details..

**Policies Table**

* **Columns**: id, name, description, type, base\_rate, coverage\_amount, created\_at, updated\_at
* **Description**: Defines policy information.

**PoliciesDetails Table**

* **Columns**: id, policy\_id, customer\_id, broker\_id, premium\_amount, start\_date, end\_date, term\_length, term\_factor, status
* **Description**: Links policies to customers.

**Claims Table**

* **Columns**: id, policy\_id, customer\_id, broker\_id, status, description, amount, created\_at
* **Description**: Tracks customer claims.

**ClaimHistory Table**

* **Columns**: application\_id, job\_id, candidate\_id, application\_date, status.
* **Description**: Logs claim updates.

# 7. Features

* **Customer Management:** CRUD operations for customer data.
* **Policy Management:** Managing policies and linking them to customers.
* **Claims Management:** Submitting, updating, approving/rejecting, and tracking claims.

# **8. UML Diagrams**

## ****8.1 Class Diagram****

## A diagram of a computer Description automatically generated

## ****8.2 Sequence Diagram****

## A screenshot of a computer Description automatically generated

# 9. Screenshots

**Broker registration**

**A computer screen shot of a computer screen

Description automatically generated**

**Broker login**

**A computer screen shot of a login page

Description automatically generated**

**Broker Dashboard**

**A screenshot of a dashboard

Description automatically generated**

**Add new customer**

**A screenshot of a computer

Description automatically generated**

**Assign Policy**

**A screenshot of a computer

Description automatically generated**

**Calculated premium amount for the customer**

**A screenshot of a computer

Description automatically generated**

**Submit claim**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Claim history**

**A screenshot of a computer

Description automatically generated**

**Admin claim approval/reject**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Claim history for broker**

**A screenshot of a computer

Description automatically generated**

**Policy Renewal**

**Expired policy**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Policy renewed for customer**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

# **10. Conclusion**

The **Insurance Broker System** provides a modern, scalable solution for managing insurance operations. By integrating advanced technologies and patterns, the project ensures data integrity, ease of use, and scalability.