Java/JEE Mini Project

Online Claim Registration System

Document Control

|  |  |  |  |
| --- | --- | --- | --- |
| ***Reference No:*** |  | | |
| ***Security Classification:*** | **Internal** | | |
| ***Issue Date:*** |  | | |
| Author(s): | **Name** | | **Title** |
| Prabaharan Rajendran | | Insurance Practice |
|  | |  |
| Reviewer(s): | **Name** | | **Title** |
| Training Team | |  |
| Issuer(s): | **Name** | | **Title** |
|  | |  |
| Issuer Signature(s): |  | | |
| Distribution: | **Name** | | **Title** |
| Team | | All Designations in Training |
|  | |  |
|  | |  |
| Document History: | **Date** | **Revision** | **Change** |
| Oct 26, 2018 | 0.01D | Initial draft |
|  |  |  |
|  |  |  |

Table of Contents

[1 Introduction 4](#_Toc528348423)

[1.1 Setup Checklist for Mini Project 4](#_Toc528348424)

[1.2 Instructions 4](#_Toc528348425)

[2 Problem Statement 5](#_Toc528348426)

[2.1 Objective 5](#_Toc528348427)

[2.2 Abstract of the project 5](#_Toc528348428)

[2.3 Functional components of the project 5](#_Toc528348429)

[2.4 Technology used: 10](#_Toc528348430)

[3 Implementation in J2EE LOT 11](#_Toc528348431)

[3.1 Summary of the functionality to be built: 11](#_Toc528348432)

[3.2 Evaluation and assessment parameters: 12](#_Toc528348433)

[Additional Important Information: 13](#_Toc528348434)

[4 Glossary 14](#_Toc528348435)

# Introduction

This document outlines a mini project for the J2EE LOT. The project is to develop an

Insurance Claim Registration System. This document contains the work flow of the system and gives guidelines on how to build the functionality gradually in each of the course modules of the J2EE LOT.

## Setup Checklist for Mini Project

Minimum System Requirements

* Intel Pentium 90 or higher (P166 recommended)
* Microsoft Windows 95, 98, or NT 4.0, 2k, XP, Windows 7
* Memory: 32MB of RAM (64MB or more recommended)
* Internet Explorer 6.0 or higher
* Oracle 9i client and access to oracle 9i server
* JDK 8
* Eclipse Latest or Spring STS
* JUnit 4.0, Maven
* WildFly or Tomcat
* Oracle 9G or MySQL

## Instructions

* The code modules in the mini project should follow all the coding standards.
* Create a directory by your name in drive **<drive>**. In this directory, create a subdirectory **Mini Project**. Store your Project here.
* You can refer to your course material.
* You may also look up the help provided in the java docs and documentation provided with Wild-Fly.
* The total time required to complete this mini project is 45 hrs (5 Days).

# Problem Statement

## Objective

Development of online Insurance Claim Registration System for the small commercial business.

## Abstract of the project

This project is aimed at developing an Insurance Claim Registration System. This is a web-based application that can be accessed over the web. This system can be used for registering a claim for a small commercial insurance business. This system can also be used to search and view the existing insurance claims. This online insurance quote system will be used by three actors **1)** **Insured** - Small Commercial Business Owner **2) Insurance Claim Handler** and **3) Claim Adjuster** who has Admin access

## Functional components of the project

Following is a list of functionalities of the system. Wherever, the description of functionality is not adequate, you can make appropriate assumptions and proceed. Information, Warning or Error Messages, Validation check can be handled at your convenience. Innovation, Efficiency, Effectiveness would be evaluated.

The three type of users who would access the system viz. 1**1)** **Insured** - Small Commercial Business Owner **2) Insurance Claim Handler** and **3) Claim Adjuster** who has Admin access

**Users Roles and Privileges**

|  |  |
| --- | --- |
| **Roles** | **Privileges** |
| Insured | * They can create claim for their own policies * They can view the status of the claim |
| Agent | * They can create claims for their Customers * They can only view their customer's claims |
| Underwriters or Admin | * They can view all claims and generate report * They can create all 3 roles |

**Screen Flow and Prototype**

Create a Login Screen to login into the application through valid credentials. Generate a unique user-id for each of the registered users.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen** | **Field Name** | **Data Type** | **Property** | **Comments** |
| User Login | Login ID | Alpha Numeric 20 | Text Box | Unique |
| Password | Alpha Numeric 12 | Text Box | Validation |

**After a valid login**,

User can see the below buttons / options based on their user role. Disabled means the option will be invisible / disabled for the user to choose.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Button/User** | **New Profile Creation** | **Claim Creation** | **View Claim** | **Report Generation** |
| Insured | Disabled | Enabled (For their own policies Only) | Enabled | Disabled |
| Claim Handler | Disabled | Enabled (For their customer’s Only) | Enabled | Disabled |
| Claim Adjuster | Enabled | Enabled | Enabled | Enabled |

**Profile Creation Screen:**

User Creation Button will be enabled only for Admins once they login. Admins can create the profile of new users using the below prototyped screen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen** | **Field Name** | **Data Type** | **Property** | **Comments** |
| User Creation | User Name | Alpha Numeric 20 | Text Box | Unique – Validate against database |
| Password | Alpha Numeric 12 | Text Box | Should be meeting certain requirements, which you can assume |
| Role Code | Alpha Numeric 10 | Drop Down |  |

**Show the list of policies:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Policy List | View Only - Quick View |  |  |  |
| Account Number | Policy Number | Premium Amount | Create Claim (Link to create claim) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen** | **Field Name** | **Data Type** | **Property** | **Comments** |
| Claim Creation | Claim Reason | Alpha Numeric 30 | Text Box |  |
| Accident Location | Alpha Numeric 40 | Text Box |  |
| Accident City | Alpha Numeric 15 | Text Box |  |
| Accident State | Alpha Numeric 15 | Text Box |  |
| Accident Zip | Numeric 5 | Text Box | Add proper field validations |
| Claim Type | Alpha Numeric 30 | Drop Down |  |
|  |  |  |  |

Next button has to take user to the claim details screen

**Claim Details Screen:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen** | **Field Name** | **Data Type** | **Property** | **Comments** |
| Claim Details | Question 1 (Dynamic) | Alpha Numeric 80 | Radio Button | Questions will be dynamically populated from the Database tables depending upon the Line Of Businesses selected by the Insured.  Answers for questions will be captured by radio buttons and the value of each radio button also have to be dynamically retrieved from the database.  (Refer to database designs discussed in later section |
| Question 2 (Dynamic) | Alpha Numeric 80 | Radio Button |
| Question 3 (Dynamic) | Alpha Numeric 80 | Radio Button |
| Question 4 (Dynamic) | Alpha Numeric 80 | Radio Button |
| Question 5 (Dynamic) | Alpha Numeric 80 | Radio Button |
| Claim Number | Numeric 10 | Invisible | To be calculated by adding all the dynamically retrieved values of answers, given by the user, from the database |

**Report Generation Screen:**

Report generating screen created by using the below prototyped screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Claim List | View Only - Quick View |  |  |  |
| Policy Number | Claim Number | Claim Type | Claim Number |

Detailed report view can be created using below prototyped screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen** | **Field Name** | **Data Type** | **Property** | **Comments** |
| Policy Detailed Report View | View Only - Detail View after clicking View Button |  |  |  |
| Claim Reason | Alpha Numeric 30 | Label & Values |  |
| Accident Location Street | Alpha Numeric 40 | Label & Values |  |
| Accident City | Alpha Numeric 15 | Label & Values |  |
| Accident State | Alpha Numeric 15 | Label & Values |  |
| Accident Zip | Numeric 5 | Label & Values |  |
| Claim Type | Alpha Numeric 30 | Label & Values |  |
| Question 1 (Dynamic) | Alpha Numeric 80 | Label & Values |  |
| Question 2 (Dynamic) | Alpha Numeric 80 | Label & Values |  |
| Question 3 (Dynamic) | Alpha Numeric 80 | Label & Values |  |
| Question 4 (Dynamic) | Alpha Numeric 80 | Label & Values |  |
| Question 5 (Dynamic) | Alpha Numeric 80 | Label & Values |  |

**Data model and tables**

As per your planned designed system add the Data model classes and Database Tables. Data model and tables can be created as mentioned below. Any new approach to achieve the above screen’s functionalities are encouraged.

|  |  |  |  |
| --- | --- | --- | --- |
| **Transaction Tables** | | | |
| **Table Name** | **Field Name** | **Data Type** | **Comments** |
| User Role | User Name | Alpha Numeric 20 | Unique - Primary Key |
|  | Password | Alpha Numeric 12 | Validation |
|  | Role Code | Alpha Numeric 10 |  |
| Claim | Claim Number | Numeric 10 | Unique - Primary Key |
|  | Claim Reason | Alpha Numeric 30 |  |
|  | Accident Location Street | Alpha Numeric 40 |  |
|  | Accident City | Alpha Numeric 15 |  |
|  | Accident State | Alpha Numeric 15 |  |
|  | Accident Zip | Numeric 5 |  |
|  | Claim Type | Alpha Numeric 30 |  |
|  | Policy Number | Numeric 10 | Foreign Key to Policy |
| Policy | Policy Number | Numeric 10 | Unique - Primary Key |
|  | Policy Premium | Decimals 8 |  |
|  | Account Number | Numeric 10 | Foreign Key to Account |
| Policy Details | Policy Number | Numeric 10 | Foreign Key to Policy |
|  | Question ID | Alpha Numeric 15 |  |
|  | Answer | Alpha Numeric 30 |  |

Transactions are various operations performed by various users, you have to maintain all the transactions done by all the users in a transaction table.

## Technology used:

* + - *Front End & Web Components:–* 
      1. HTML / JavaScript
      2. Spring MVC (preferred), Servlet
      3. JSP
    - *Business Logic Components and Services :-* 
      1. Java Beans, MVC Architecture (Spring)
    - *Application Servers :-* 
      1. WildFly or Tomcat
    - *Databases:-*
      1. JPA with Hibernate Or JDBC
      2. Oracle 9i Or MySQL

# Implementation in J2EE LOT

## Summary of the functionality to be built:

The participants need to develop the Insurance Quote Generation System by building the functionality incrementally. Guidelines on the functionality to be built:

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **Course** | **Duration (in Hours)** | **Outcome and Evaluation** |
| 1 | Understanding the case sturdy | 2 | Clarification Document |
| 2 | Data model and Table Creation | 6 | Validate the tables |
| 3 | Screen Development (Parallel Activities) | 6 (Parallel Activities) | Review Screen |
| 4 | Reference Data Load | 2 | Review Tables along with Data |
| 5 | Project Directory Set Up | 2 | Workspace Review |
| 6 | Login Screen, Profile Creation & necessary business functionality Development | 12 Hours (Parallel Activities – 1 screen per team) | Login Screen, Profile Creation functionality validation |
| 7 | Account Screen & necessary business functionality Development | Account Screen Validation |
| 8 | Policy Screen & necessary business functionality Development | Policy Screen Validation |
| 9 | Search Screen & necessary business functionality Development | Search Screen Validation |
| 10 | Detail Report Screen & necessary business functionality Development | Reporting Screens Validation |
| 11 | Unit Test Case and Individual Flow Test | J-Unit Test Execution Result |
| 12 | Integration of all screens and functional flow test | 8 | Insurance Policy Creation Flow Quick Demo |
| 13 | Documentation and Presentation readiness | 2 | Project specs and learning PPTs |

## Evaluation and assessment parameters:

This mini project will be done in groups of five. Each group will identify a Team Lead who will decide which team member will code for which functionality. This project shall be evaluated at the end of spring module.

**Evaluation Criteria (out of 100):**

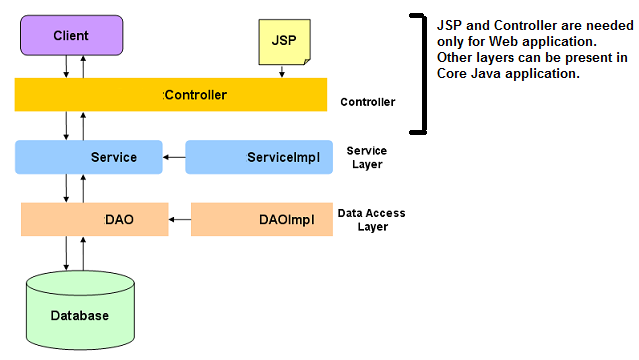
|  |  |
| --- | --- |
| Look and Feel of Web pages | **05** |
| Client-side and server-side validation | **10** |
| Code Documentation and using coding standards | **10** |
| Overall Business logic. This includes:   * Usage of Logging API (log4j) | **25** |
| Usage of Maven to build project | **5** |
| Good amount of appropriate dataset to showcase project completely | **5** |
| Appropriate test cases using J-Unit 4.0 | **5** |
| Using MVC architecture and clean encapsulation of business logic in appropriate components. Judicious use of java beans, cleaner looks to JSP | **35** |

## Additional Important Information:

**Application Architecture:**

Plan well before coding: A well planned design will save a lot of your coding time and will help your build a better application in terms of Scalability. Discuss your planned design / architecture of system i.e. Data model classes, Database tables, various layers and interfaces with your mentor and then get into coding.

Knowledge of Design Patterns such as Singleton, Builder and Factory Patterns will be helpful while designing your Model classes. Also, try to learn about SOLID principles of Software Design and try to implement those principles into your project this will keep your system Modular, Flexible, Testable, Fault Tolerant and Open to changes which are desired and expected from your once you are deployed into project.



**Coding Practices:**

Try to follow coding standards to write more readable code. Follow proper naming conventions for your variables, methods, classes and packages.

Write proper comments defining your classes, methods and their role inside the project structure.

Write Log statements to record the behavior of application during runtime, don’t solely depend upon the console.

# Glossary

1. **Insured:** A person who takes the Insurance to protect his assets against loss or damages.
2. **Agent:** A person who act as a middleman between the company providing the Insurance (Insurer) and the customer i.e. the person who is getting the Insurance (Insured).
3. **Underwriter:** A person inside the Insurance Company who checks, validates and then issues the Policy for that Insurance.
4. **Policy:** The terms and conditions of an Insurance that specify the coverage provided by the Insurance.
5. **Coverable:** Items that are covered / insured under the Insurance Policy.
6. **Coverage:** Scenarios under which a coverable will be insured.
7. **Claim Handler:** A person who creates claim and co-ordinate the claim processing
8. **Claim Adjuster :** A person who review and approve the claim.