Member Policy Management System

1. Business Requirement

Member Policy Management System is a system, which holds various Health Insurance Policies for Corporates. As the first phase of the work, they have intended to automate the process of maintaining the Member and Plan details.

1.1 Target Audience

Admin team will be the target audience of the Member Policy Management System.

2. Technical Specifications

Database Connectivity Details

Use MYSQL database for all the database activities. The script needed for the case study and the Connection details is given below:

Driver Class	com.mysql.jdbc.Driver
URI Name	jdbc:mysql://localhost/member_policy
UserName	root
Password	<PASSWORD IS BLANK>

Create the tables before proceeding with the source code. The detailed steps on how to execute the SQL file are available in the user manuals.

Code Skeleton for the Case Study

The code skeleton to be used for developing the case study will be given as a zip file. This will start as the base code for you to work on the solution.

3.1 Get Plan Details of the Member

Background

The member details are already recorded in the database in the table named **"MEMBER_DETAILS"**. Also the plan details are recorded in the table named **"PLAN_DETAILS"** (these tables would have been created by running the DB script). Based on the input, the relevant details need to be retrieved from the database.

Requirement Description:

This requirement is to retrieve the plan and the member details of the respective Member.
The functionality gets the member id, plan id and corporate name and retrieve the values from the database.

- If the member holds the given plan id, then retrieve member_id, plan_id, plan_value, tenure, corporate_name, monthly_due_date from the database.
- The retrieved values need to be set in MemberVO, which is already provided as part of the skeleton.
- If the member does not hold the given plan, then a user defined exception named "**MemberPolicyException**" should be thrown with the message "**Invalid Plan Id for the given Member**"

Design Rules:

i. MemberPolicyBO should be injected in MemberPolicyService and MemberPolicyDAO must be injected in MemberPolicyBO

ii. Do not change the package definitions or class names or method signatures in the code.

iii. You can use either Annotation or XML based configuration to inject the necessary beans, unless specified. If you are using XML based configuration, define your beans in applicationcontext.xml file ONLY. If you are using annotation based configuration, define your beans in 'MemberPolicyConfig' class only.

Method Specification:

Component Name	Method	Input	Output	Exceptional Case
MemberPolicyDAO	getPlanDetailsOfMember	String memberId, String planId, String corporateName	MemberVO	MemberPolicyException

3.2 Get Discount Details

Background

Each member can avail the discount percentage for the plan they hold. These discount percentages will be offered based on the Corporate Function Scheme.

Requirement Description:

This requirement is to retrieve the discount percentage for the respective Corporate Scheme.

- If the given corporate name is present in the Map, it returns the discount percentage
- If the given corporate name is not present in the Map, then a user defined exception named "**MemberPolicyException**" should be thrown with the message "**There is no discount for this Corporate Function**"

Design Rules:

i. MemberPolicyBO should have the method '**getDiscountToBePaid'** which must return the discount percentage

ii. Map should be configured with the below data. Please **DO NOT change** the corporate name or the discount percentage values while injecting the map in the Application Context

Corporate Name	Discount Percentage
IT	50
Automobiles	30
Manufacturing	20

iii. You can use either Annotation or XML based configuration to inject the necessary beans, unless specified. If you are using XML based configuration, define your beans in applicationcontext.xml file ONLY. If you are using annotation based configuration, define your beans in 'MemberPolicyConfig' class only.

Method Specification:

Component Name	Method	Input	Output	Exceptional Case
MemberPolicyBO	getDiscountDetails	String corporateName	Double discountPercentage	MemberPolicyException

3.3 Update Premium Amount of the Member

Background

The premium details of the member should be persisted in the database in the table named "**MEMBER_DETAILS**". (this table would have been created by running the DB script). Based on the discount amount retrieved from requirement 3.2, the premium amount should be calculated and should be persisted into the database.

Requirement Description:

This requirement is to update the premium amount of the member based on the discount percentage of the respective Member. **The functionality gets the plan value from the database and calculate the premium amount based on the discount percentage.**

- From Requirement 3.1, the memberVO would have been returned to MemberPolicyBO layer with the values retrieved from the database.
- Extract the plan value from the memberVO. The discount percentage would have been returned from the Map using requirement 3.2.
- Calculate the Premium amount as per the below formula in the method '**calculatePremiumAmount**'
  PREMIUM = (PLAN_VALUE * DISCOUNT_PERCENTAGE) /100;
- The calculated premium amount needs to be updated in the database for the given member_id.

Design Rules:

i. MemberPolicyBO should be injected in MemberPolicyService and MemberPolicyDAO must be injected in MemberPolicyBO

ii. Do not change the package definitions or class names or method signatures in the code.

iii. Do not change the property files provided in the application

iv. Use **ONLY DriverManagerDataSource for creating the datasource**. You MUST NOT change the database properties file (**mysql.properties**) provided as part of the Skeleton. Use **PropertyPlaceHolder** configuration to read the properties and create the datasource bean. **All the values from mysql.properties file should be read by using the PropertyPlaceHolder, except for driverclassname. The Driver Class name should be hard coded as it and should not be read using PropertyPlaceHolder**

v. Use **ONLY Declarative Transaction management** for updating the premium_amount into the database. We expect you to have this transaction in the 'MemberPolicyBO' class, for the method 'updatePremiumAmount'

vi. You can use either Annotation or XML based configuration to inject the necessary beans, unless specified. If you are using XML based configuration, define your beans in applicationcontext.xml file ONLY. If you are using annotation based configuration, define your beans in 'MemberPolicyConfig' class only.

Method Specification:

Component Name	Method	Input	Output	Exceptional Case
MemberPolicyService	UpdatePremiumAmount	String memberId, String planId, String corporateName	void	MemberPolicyException
MemberPolicyBO	UpdatePremiumAmount	String memberId, String planId, String corporateName	void	MemberPolicyException
MemberPolicyBO	calculatePremiumAmount	Double discountPercentage, MemberVO memberVO	Integer	MemberPolicyException
MemberPolicyDAO	updatePremiumAmount	String memberId, String planId, String corporateName	MemberVO	MemberPolicyException