# Project: Employee Group Health Insurance Management System

## 1. Overview

The **Employee Group Health Insurance Management System** is a web-based application aimed at managing group health insurance policies for employees of various organizations. The system includes functionalities for employee registration, policy enrollment, claims management, premium computation, and employer-specific reports.

The system is developed using the **MVC architecture**, making it compatible with both **Java (Spring MVC)** and **.NET (ASP.NET Core MVC)** frameworks. It provides a seamless experience for employees, HR personnel, and insurance administrators.

## 2. Assumptions

- Role-based access for employees, HR managers, and insurance administrators.
- Each employer can offer multiple group insurance policies, and employees can choose policies based on their requirements.
- SQL Server or MySQL will be used for data storage.
- The system will provide report generation and export capabilities.

## 3. Module-Level Design

### 3.1 Employee Management Module

**Purpose:** Handles employee registration and profile management.

- **Controller:**
  - EmployeeController
    - registerEmployee()
    - getEmployeeDetails()
    - updateEmployeeProfile()
    - listAllEmployees()
- **Service:**
  - EmployeeService
    - Business logic for employee profile validation and updates.
- **Model:**
  - **Employee Entity**
    - Attributes:
      - employeeId (PK)
      - name
      - email

      - phone
      - address
      - designation
      - organizationId (FK)

### 3.2 Policy Enrollment Module

**Purpose:** Facilitates employee enrollment into group insurance policies.

- **Controller:**
  - EnrollmentController
    - enrollInPolicy()
    - getEnrolledPolicies()
    - cancelEnrollment()
- **Service:**
  - EnrollmentService
    - Handles business logic for policy selection and enrollment.
- **Model:**
  - **Enrollment Entity**
    - Attributes:
      - enrollmentId (PK)
      - employeeId (FK)
      - policyId (FK)
      - enrollmentDate
      - status (ACTIVE, CANCELLED)

### 3.3 Premium Calculator Module

**Purpose:** Calculates premiums based on policy type, employee age, and dependents.

- **Controller:**
  - PremiumCalculatorController
    - calculatePremium()
- **Service:**
  - PremiumCalculatorService
    - Implements algorithms for premium computation based on predefined rules.
- **Model:**
  - Operates directly on policy and enrollment data.

### 3.4 Claims Management Module

**Purpose:** Manages submission, processing, and approval/rejection of insurance claims.

- **Controller:**
  - ClaimController
    - submitClaim()

        - getClaimDetails()
        - updateClaimStatus()
        - listAllClaims()
- **Service:**
    - ClaimService
        - Validates claim submissions and processes approvals.
- **Model:**
    - **Claim Entity**
        - Attributes:
            - claimId (PK)
            - enrollmentId (FK)
            - claimAmount
            - claimReason
            - claimDate
            - claimStatus (SUBMITTED, APPROVED, REJECTED)

### 3.5 Reporting Module

**Purpose:** Generates reports for HR managers and administrators.

- **Controller:**
    - ReportController
        - generateEmployeeReport()
        - generatePolicyReport()
        - exportReport()
- **Service:**
    - ReportService
        - Prepares and exports reports in multiple formats (PDF, Excel).
- **Model:**
    - Operates on aggregated data from other modules.

## 4. Database Schema

### 4.1 Table Definitions

1. **Employee Table**

```
CREATE TABLE Employee (
  employeeId INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100),
  email VARCHAR(100),
  phone VARCHAR(15),
  address TEXT,
  designation VARCHAR(50),
  organizationId INT
);
```

2. **Policy Table**

```
CREATE TABLE Policy (
   policyId INT AUTO_INCREMENT PRIMARY KEY,
   policyName VARCHAR(100),
   coverageAmount DECIMAL(10, 2),
   premiumAmount DECIMAL(10, 2),
   policyType ENUM('INDIVIDUAL', 'FAMILY')
);
```

3. **Enrollment Table**

```
CREATE TABLE Enrollment (
   enrollmentId INT AUTO_INCREMENT PRIMARY KEY,
   employeeId INT,
   policyId INT,
   enrollmentDate DATE,
   status ENUM('ACTIVE', 'CANCELLED'),
   FOREIGN KEY (employeeId) REFERENCES Employee(employeeId),
   FOREIGN KEY (policyId) REFERENCES Policy(policyId)
);
```

4. **Claim Table**

```
CREATE TABLE Claim (
   claimId INT AUTO_INCREMENT PRIMARY KEY,
   enrollmentId INT,
   claimAmount DECIMAL(10, 2),
   claimReason TEXT,
   claimDate DATE,
   claimStatus ENUM('SUBMITTED', 'APPROVED', 'REJECTED'),
   FOREIGN KEY (enrollmentId) REFERENCES Enrollment(enrollmentId)
);
```

5. **Organization Table**

```
CREATE TABLE Organization (
   organizationId INT AUTO_INCREMENT PRIMARY KEY,
   organizationName VARCHAR(100),
   contactPerson VARCHAR(100),
   contactEmail VARCHAR(100)
);
```

## 5. Local Deployment Details

1. **Environment Setup:**
   - Install MySQL or SQL Server. Configure the database schema using provided SQL scripts.
   - Set up the development environment with JDK (Java) or .NET SDK (ASP.NET Core).
   - Configure application properties for database connections.
2. **Steps to Deploy Locally:**

- Clone the project repository.
- Build the project using Maven (Java) or Visual Studio (ASP.NET Core).
- Run the application server (Tomcat for Java or IIS/Kestrel for .NET).
- Access the application at http://localhost:8080 (Java) or http://localhost:5000 (.NET).

## 6. Conclusion

The **Employee Group Health Insurance Management System** provides an efficient solution for managing group health insurance policies. Its modular architecture ensures scalability, maintainability, and ease of integration. The system's features cater to diverse user roles, ensuring comprehensive insurance management for organizations and employees alike.