Company: Appsfactory
Author: Sameed Tariq
Assignment: Java Coding Task.

1. **The entities must be persisted in a database. Which database and database type (SQL/NoSQL) do you use? Justify your decision.**

The dataset of postcode is highly structured, and the structure of data don't change frequently. The standards are defined by ISO_3166 describes how data should be created and stored for each country.

Our normalized form of data needs the references of each other to in the data tables to store data. The RDMS support the fast transaction updating multiple rows across tables with complex integrity constraint.

The storing of annual premium insurance calculations data adhering to the principles of atomicity, consistency, isolation, and durability (ACID) are essential to support the workload of financial transaction.

The SQL database can scale up vertically by adding resources such as CPU and memory. However, NoSQL database can scale up horizontally by adding more nodes. In our case, we have selected different database of each domain. Hence, we can deploy the multiple nodes of read only transaction (Less update or write operation) to different Geographical locations.

The **H2 In-Memory (MySQL)** database is used for Insurance Premium Calculations.

2. **Implement a testing framework and explain your approach to ensuring software quality.**

The Test-Driven development is approach to applied to develop the modules of the code. It ensures the development of the functionality without affecting the other functions in the product. Secondly, the unit test is more reliable than the code without it. If any thing breaks in the future, the root of the problem will be identified right away rather than coming through unwieldy codebase to find the issue.
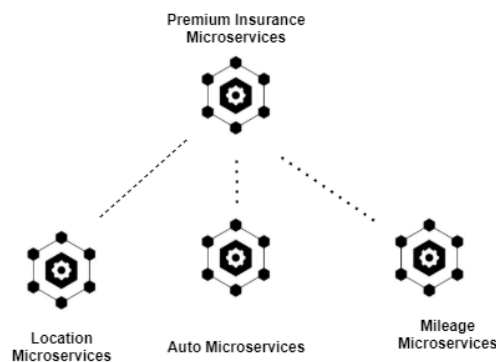
Junit, Mockito, SpringWebMvc and Integration test is applied in the project to ensure the quality of the software.

3. Which transactions/updates take place between the services?

a) Each microservices can be induvially access with the help of api-gateway
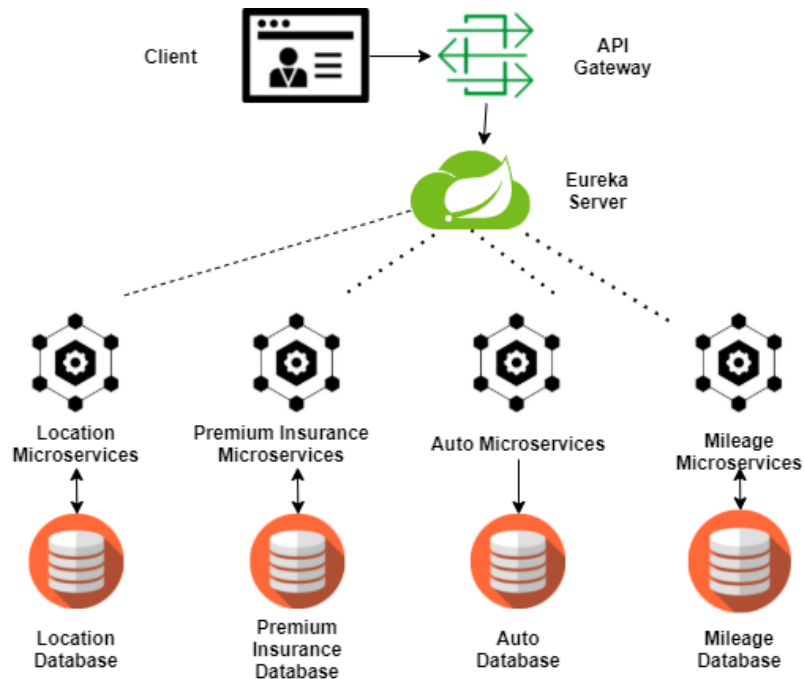
b) Microservice Premium Insurance can communicate with other microservices with the help of eureka server.

4. API Documentation
The API documentation of the Premium Insurance calculation is provided separately. You must find it in the attached documents.