Weather Forecasting Application

Title: - Weather Forecasting Application.

Software Requirements: -

1) JDK 11
2) Gitbash
3) GitHub
4) STS
5) mongoshell or mongo Compass or Mongo Atlas
6) Junit 5 & Mockito
7) Eureka
8) Postman
9) Chrome.

Minimum Hardware Requirements: -

1) CPU: 2 GHz Processor
2) RAM: 4 GB
3) HDD: 10 GB
4) Display Resolution: 1024 X 800

Introduction: -

Weather Forecasting Application is the application of Science & Technology to predict the conditions of atmosphere for a given location and time.

Weather Forecasting is very important since it helps to determine future climate changes.

On an everyday basis, many use of us uses weather forecast to determine what to were on a given day. Since outdoor activities are severely curtailed by heavy rain, snow, and wind chill, forecast can be used to plan activities around these events, and to plan and survive them.

User Stories: -

1. **Creating Spring boot Application**.
   As a Developer I will Create new Spring Boot Project Named as Weather Forecasting and Adding required Packages and Layers.
   Average Time for Completion: - ½ hours.

2. **Writing Model Class.**
   As a Developer I will Create POJO Class for All Document Entities. And Parameterised Constructor, Getters and Setters, ToString Method. It's a class to deal with Data.

3. **Writing Business Logic in Service layers**.
   As a Developer I will Write Business logic in service layer Classes.

   Service 1: - Register the User & Admin.

   Service 2: - Login the User and Admin.

   Service 4: - post Desired Location and Time from user store it into Database(mongodb).

   Service 5: - Return (Get) the Weather Forecast depending upon 'Users Live Location and Time' OR 'Users Desired Location and Time'.

4. **Connect MongoDB database to our application**.
   As a developer I will Create a Database and inside that I will Create Different Collection for Different Service and will perform Database connectivity in between them.

i. DB Name: - "User_Registration_Details" For Service 1.
ii. DB Name: - "User_Login_Details" For Service 2.
iii. DB Name: - "User_LiveLocation_Details" For Service 3.
iv. DB Name: - "User_DesiredLocation_Details" For Service 4.
v. DB Name: - "FinalWeather_Details" For Service 5.

5. **Controller Layer Implementation.**

As A developer I will Create a Controller layer to get the REST (Representational State Transfer) API Requests & Data from user into the Application and send the response back depends upon their requests.

6. **Perform Exception Handling**.
   As a developer I will handle abnormal behaviours of application if present.

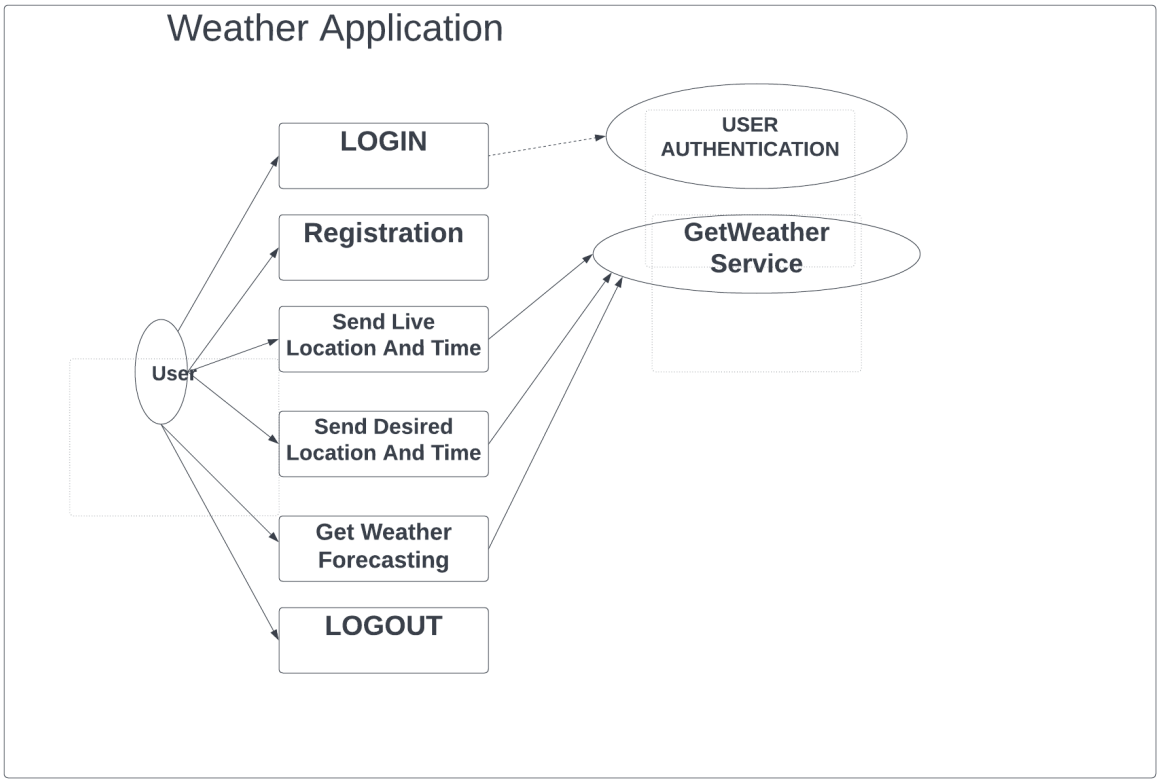
7. **Perform Testing.**
   As a developer I need to write test cases for the application using Junit5 and Mockito framework.

8. **Implement Eureka server – Client and API Gateway.**
   As a developer I will use it to access multiple microservices through single port instead of using different-different port numbers for different-different services.

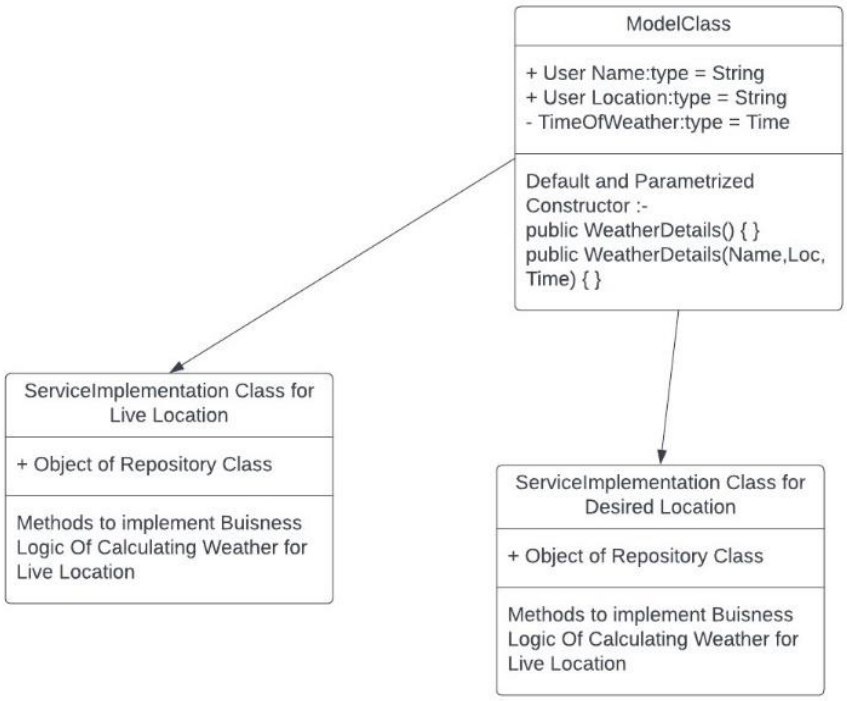
Features: -

1. Location.
2. Solar Radiation
3. Humidity
4. precipitation
5. Wind
6. Temperature
7. Atmospheric Pressure

Use Case Diagram: -

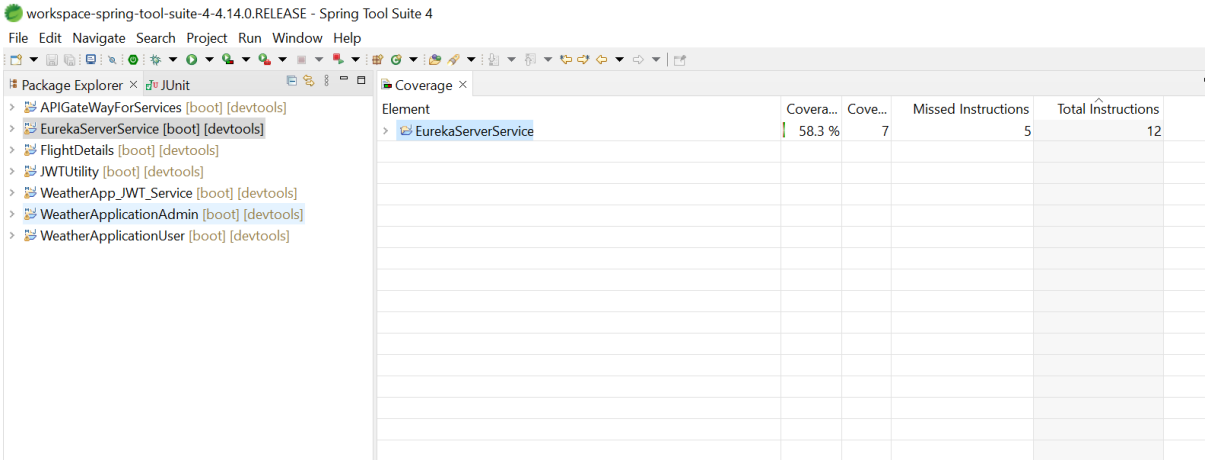


CLASS DIAGRAM :-

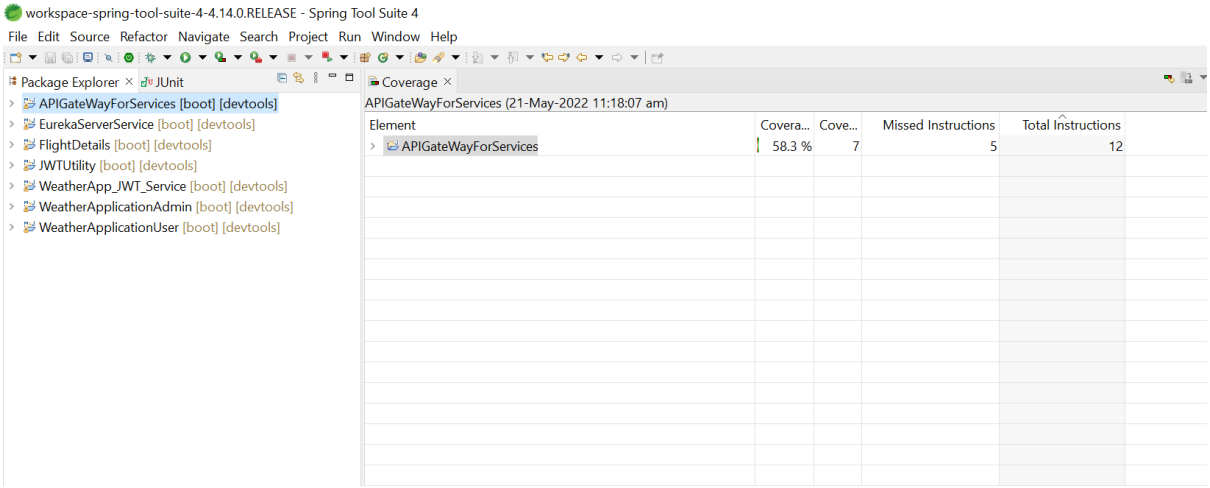


CODE Coverage Report:-

1) Eureka Server Service:-



2) API Gateway For Services:-



3) Weather Application Admin