

## Weather App Case Study

1	Name of the Project	<b>Weather App</b>
2	Objective/ Vision	<p>Build a system to view Weather forecast details for a particular city. User can search for a city to know its weather and based on forecast he should plan his journey. Application should alert user on rain, storm. The application should contain header section, search section and result display section.</p> <p>The application needs to fetch details by using the following API link.  <a href="https://openweathermap.org/api">https://openweathermap.org/api</a></p> <p>Sample reference APIs:  <a href="https://api.openweathermap.org/data/2.5/weather?id={city id}&amp;appid={API key}">api.openweathermap.org/data/2.5/weather?id={city id}&amp;appid={API key}</a></p>
3	Users of the System	All Internet users
4	Functional Requirements	<ol style="list-style-type: none"> <li>1) Home Page should consist of Register page link through which a user can register himself. Upon registration, the user able to login into his account.</li> <li>2) User home page should also have options for to edit his profile and changing his password.</li> <li>3) Weather Forecast - View to weather forecast details of a selected city.</li> <li>4) User should able to choose forecast period in days.</li> <li>5) Search on basis of climate – rain, sunny, light shower</li> <li>6) Logged in user can put cities in watch list.</li> </ol>
5	Non-functional requirements	<ol style="list-style-type: none"> <li>a) App should be accessible from any location with access to the Internet.</li> <li>b) App should be responsive to display consistently across multiple device screens.</li> <li>c) App should have an intuitive UI that can be operated by novice-expert Internet users</li> </ol>
6	Tools and Technologies to be used	<ol style="list-style-type: none"> <li>1. VCS : Gitlab</li> <li>2. Middleware : Spring Boot</li> <li>3. Frond end : Angular/React</li> <li>4. Data Store : MongoDB / MySQL</li> <li>5. Testing : JUnit, Mocha, Chai, Jest, Protractor</li> <li>6. Container : Docker</li> <li>7. Bug Fix : Sonarlint</li> <li>8. CI : Gitlab</li> </ol>

# User Stories

1	As a user I should be able to register with the application so that I can login and use the functionalities of the application.
2	As a user, I should be able to login with my user name and password in order to access the functionalities of the application.
3	As a user, I should be able to login with my Gmail account in order to access the functionalities of the application.(Optional requirement)
4	As a user I should be able to search resources to view their details
5	As a user, I should be able to save resources to a wishlist/favourite so that I can access them later
6	As a user, I should be able to access items saved to my wishlist/favourite

Notes:

- The application should be based on microservices architecture
- API Gateway pattern should be implemented using Spring Cloud Gateway
- Services should register themselves with Eureka Service Discovery server.
- All layers of microservices should be covered with automated unit and integration tests
- All microservice endpoints should have API documentation

## High Level Architecture Diagram

