Nutritionist App Case Study

1	**Name of the Project**	**Nutritionist App**
2	**Objective/ Vision**	Build a system to search for a specific food to find it's nutrition details, show list of matching food, view the nutrition content for a selected food and bookmark favorite food for later reference. The application needs to search for food and find nutrition data for a selected food by registering with the following link and get API key required to call the APIs. - https://fdc.nal.usda.gov/api-key-signup.html The rest end point details are available - https://fdc.nal.usda.gov/api-guide.html
3	**Users of the System**	All Internet users
4	**Functional Requirements**	1. User Module On launching the application, the user should get the login page. The login page should have a link for registration using which the user should be able to register. On successful registration the user should be taken to the login page. Upon login, the user should be taken to the home page. Home page should have proper UI elements to search for food based on food description, food type, published date. Home page should have proper navigation links to logout, fetch recommended food etc., 2. Search Food Module User should able to search with different criteria (based on availability of rest end point) like a) Ingredients b) Brand Owner Sorting should be implemented (based on selected field - like description, food type, published date) 3. Recommended Food Module Fetch all the food items based on the brand of food which the user ordered mostly.
5	**Non-functional requirements**	a) App should be accessible from any location with access to the Internet.

		b) App should be responsive to display consistently across multiple device screens. c) App should have an intuitive UI that can be operated by novice-expert Internet users
6	**Tools and Technologies to be used**	**1.** VCS : Gitlab **2.** Middleware : Spring Boot **3.** Frond end : Angular/React **4.** Data Store : MongoDB / MySQL **5.** Testing : JUnit, Mocha, Chai, Jest, Protractor **6.** Container : Docker **7.** Bug Fix : Sonarlint **8.** CI : Gitlab

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