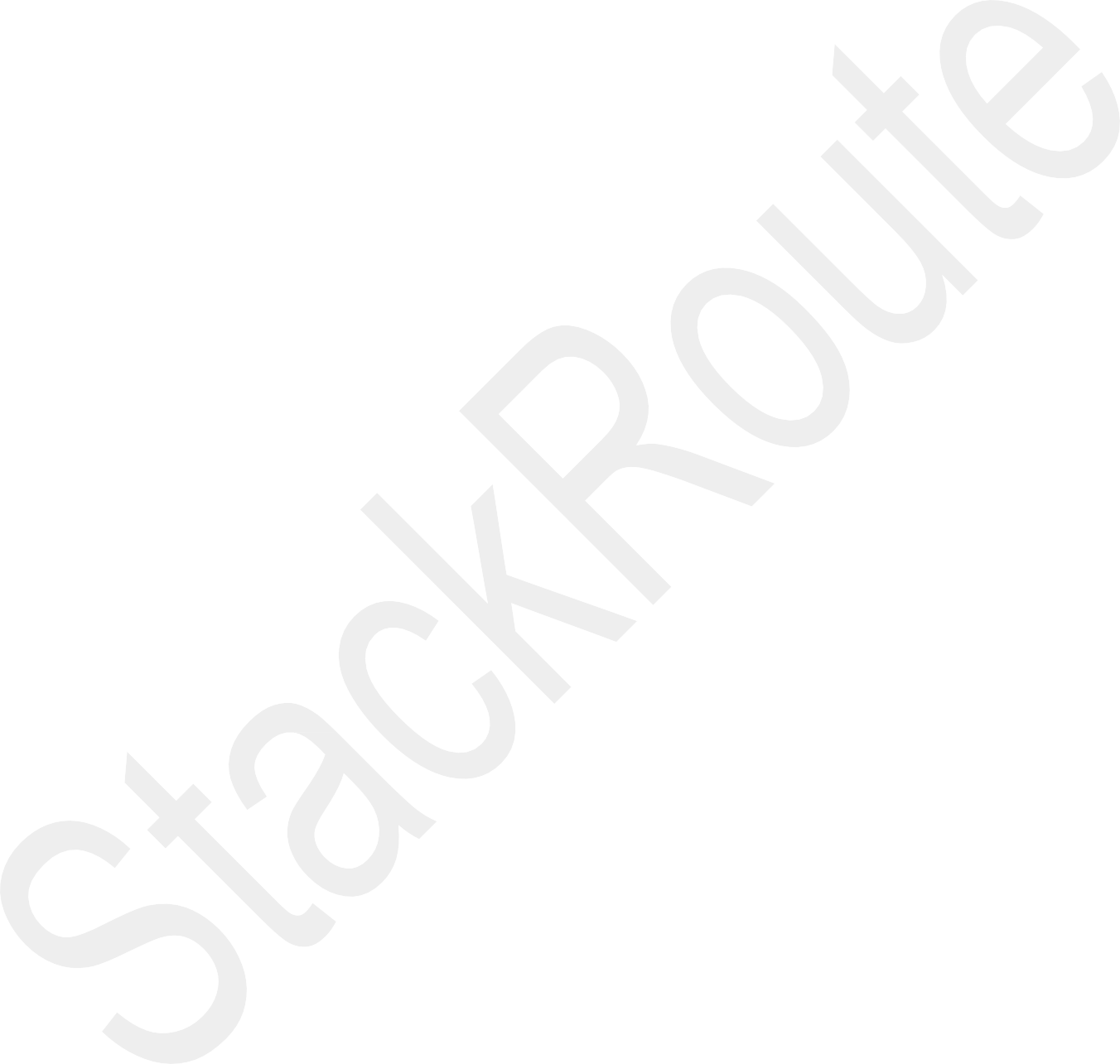
ChefUp APP CASE STUDY

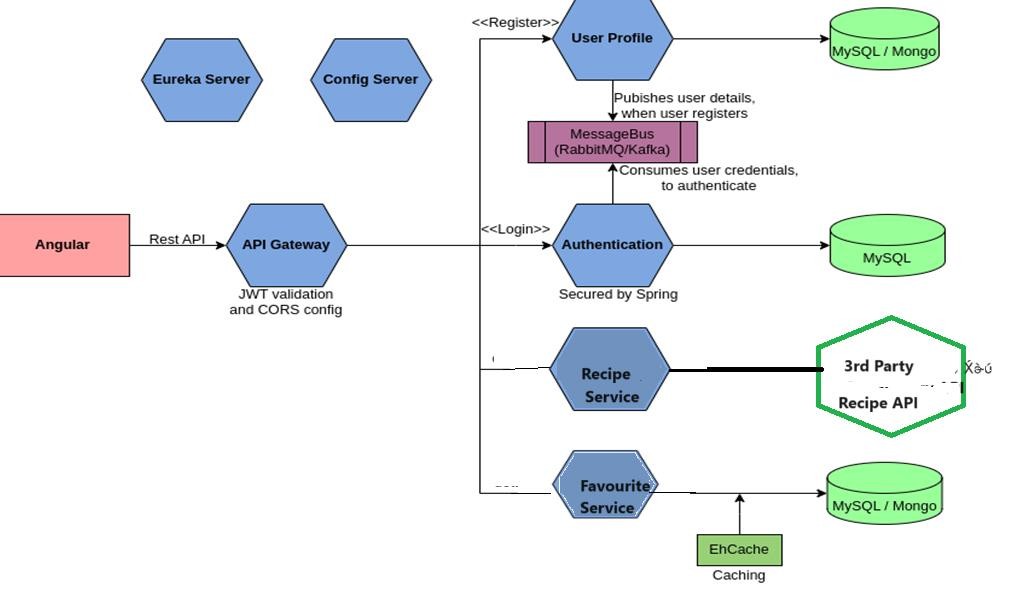
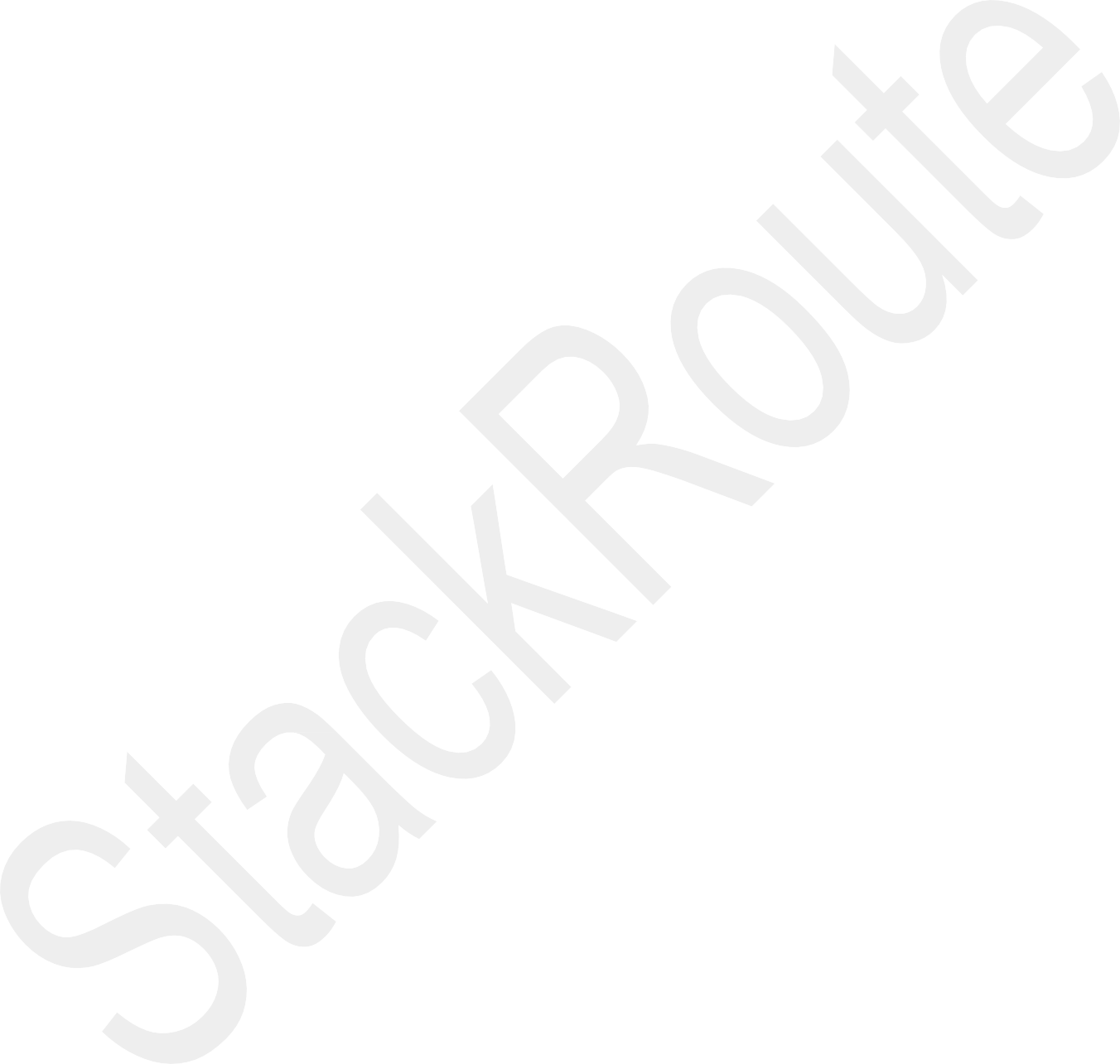
|  |  |  |
| --- | --- | --- |
| **1** | **Name of the Project** | **ChefUp App** |
| **2** | **Objective** | Develop an Application that allows users to view the recipe of food item on their choice and add as favorite  The application needs to fetch details by registering with the following API  **Sample API:**  **Search API response**  **{**  **"results": [**  **{**  **"id": 716429,**  **"title": "Pasta with Garlic, Scallions, Cauliflower & Breadcrumbs",**  **"image": "https://img.spoonacular.com/recipes/716429-312x231.jpg",**  **"imageType": "jpg",**  **},**  **{**  **"id": 715538,**  **"title": "What to make for dinner tonight?? Bruschetta Style Pork & Pasta",**  **"image": "https://img.spoonacular.com/recipes/715538-312x231.jpg",**  **"imageType": "jpg",**  **}**  **],**  **"totalResults": 86**  **}** |
| **3** | **Functional Requirements** | 1. User Interface (UI) should achieve the following:    1. User Registration    2. User Login    3. Search and view the recipe of any food item    4. Add the recipe details into favorite list    5. View favorite.    6. UI should be responsive which can run smoothly on various devices.    7. The UI should be appealing and user friendly |
| **4** | **Non-functional Requirements** | 1. The app should be able to load recipe details quickly and smoothly, even on low-end devices. 2. The app should be able to handle many users without slowing down or crashing. 3. The app should be easy to use and navigate, even for users with no prior experience with apps. 4. The app should protect user data from unauthorized access, modification, or deletion. |
| **5** | **Technical Requirements** | 1. Application should be developed using Microservices in the Backend. JWT tokens to be used for securing the Backend. 2. Frontend should be developed using Angular 3. Microservice patterns like API Gateway, Service Discovery 4. Comprehensive Unit tests and integration tests with coverage should be implemented to validate the functionality of the Application. 5. Application should be integrated with databases 6. SCM like GitHub to be used for regularly committing the source code. 7. Implement Documentation of API using Swagger/Open API. |
| **6** | **Tools and Technologies to be used** | SCM : Github  Middleware : Nest.js  Frontend : React  Data Store : MongoDB  Testing : jasmine  CodeQuality : Sonar qube  CI : Jenkins  API Documentation : Swagger |

|  |  |  |
| --- | --- | --- |
|  |  |  |

**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 username and password to access the application's functionalities.** |
| **3** | **As a user, I should be able to view all s based on country name using Third Party**  **API.** |
| **4** | **As a user, I should be able to save details to a wishlist/favourite so that I can access them later.** |
| **5** | **As a user, I should be able to access details saved to my wishlist/favourite.** |
| **6** | **As a user, I should be able to delete details saved to my wishlist/favourite.** |

**High Level Architecture Diagram**



**The responsibilities of the microservices in the above figure are as follows:**

* **User Profile Service**: This Service is responsible for storing user registration details. The Service publishes the user credentials sent as part of registration to the message bus and stores the remaining user profile information in the database.
* **Authentication Service**: This Service is responsible for consuming user credential from the message bus and storing it in the database. When a user logs in, this service validates the login credentials against the credentials stored in the database. If the credentials matches, this service generates a JWT token and sends back as response, else an error message is sent.
* **Recipe Service**: This Service is responsible for accessing an external spoonacular API to fetch recipe details based on the food details
* **Favourite Service**: This Service is responsible for storing recipe details bookmarked by users in the database.
* **API Gateway**: This Service acts as the entry point of the system. It intercepts all the requests and validates the JWT Token before routing it to the appropriate microservices.
* **Eureka Server**: This Service acts as a service registry where all the other microservices registers during startup for discoverability.
* **Config Server**: This Service acts as a centralized location to store the configuration of the other microservices of the system.

**Recommended Steps to complete the Case Study**

**Step 1**: Understand the Case Study

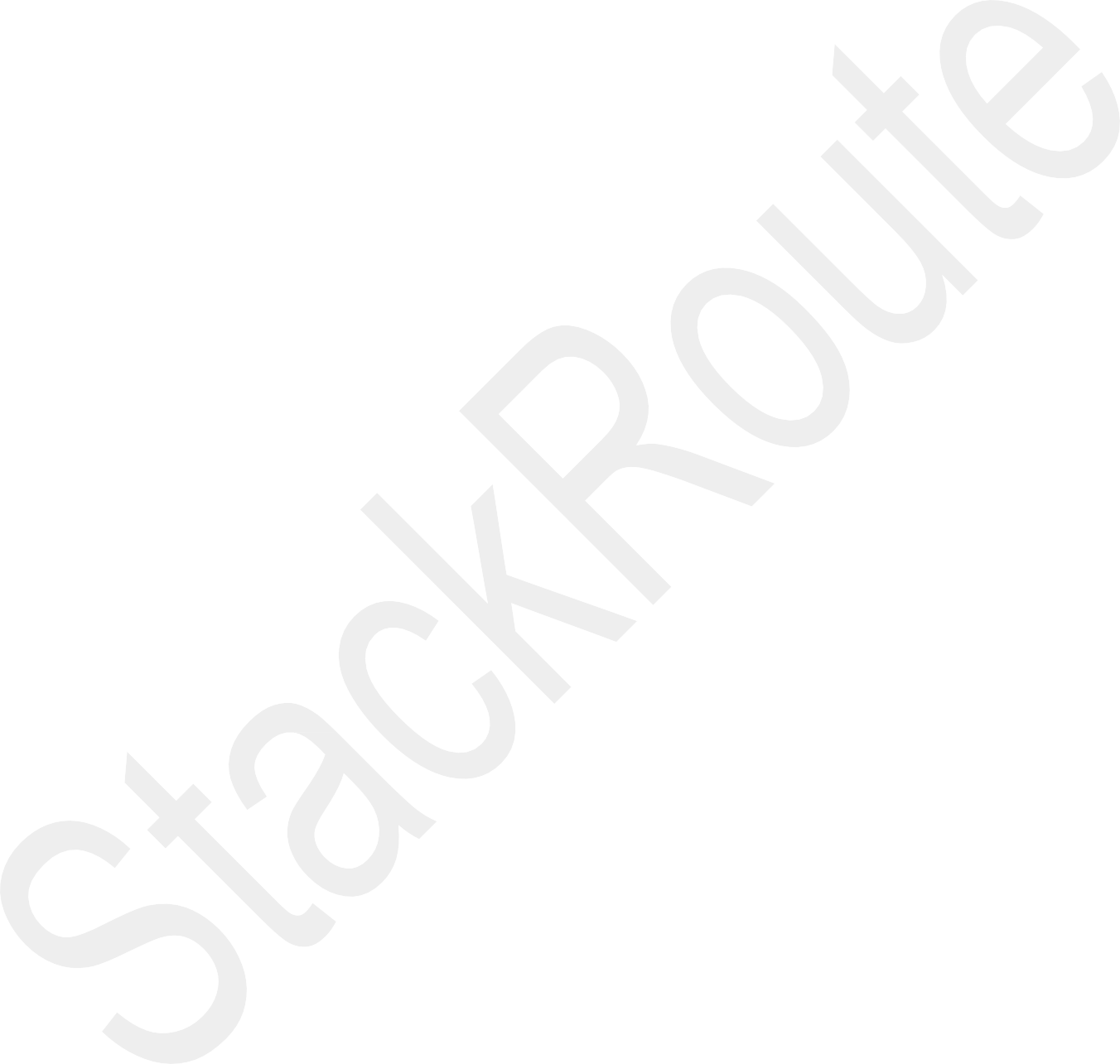
**Step 2:** Identify the Data Model and draw the data flow diagram

**Step 3:** Draw the UI Wireframes

**Step 4:** Create the Boilerplate

**Step 5:** Implement and write test cases for the backend

**Step 6:** Implement and write test cases for the frontend



**Step 7:** Integrate the frontend with the backend