Assignment: Recipe Management Application

1. Functional Requirements

This section outlines the functionality expected from the application, including API capabilities, database models, and documentation requirements.

1.1 Database Model

The application should include at least two models at the database level: **Recipe** and **Ingredient**.

- **Recipe** contains:
  - Name (string)
  - Description (string)
  - Cooking Instructions (string)
  - A list of Ingredients

- **Ingredient** contains:
  - Name (string)
  - Quantity (string, e.g., "2 cups" or "1 teaspoon")
  - Recipe (relation to Recipe)

Both models should include a timestamp for when the entity was created and a timestamp for the last update.

1.2 API

The application must provide a JSON API for users to perform CRUD operations on recipes and ingredients. In addition to CRUD operations, the API should allow users to:

- Add an ingredient to a recipe.
- Remove an ingredient from a recipe.

All lists of entities (recipes and ingredients within a recipe) should be sorted alphabetically (recipes by name, ingredients by name within each recipe). The developer is encouraged to implement the sorting logic, preferably using efficient data structures or algorithms for sorting.

The developer has the freedom to design the API endpoints and the JSON structure for requests and responses.

1.3 Documentation

Required documentation includes:

- Database configuration details
- Instructions for running application tests
- Instructions for starting the application, including any profile to be used

This documentation should be included in a README.md file in the application's root folder. Additional documentation should be provided as deemed necessary by the developer.

2. Technical Requirements

These requirements specify the technologies and architectural patterns the application should utilize.

2.1 Technology Stack

- The application must be compatible with the JVM.
- The developer may choose either Java or Kotlin as the programming language.
- Maven should be used as the build automation tool.
- Spring Boot (version 2 or higher) must be used as the application framework.
- PostgreSQL is required for the application's database. For testing purposes, an in-memory database may be used.
- Within the defined functional and technical boundaries, any libraries may be utilized.
- Using Spring Initializr to bootstrap the application is permitted.

2.2 Application Architecture

The application should follow a traditional 3-layered architecture, consisting of:

- **Web Layer:** Contains controllers, exception handlers, etc.
- **Service Layer:** Contains application and infrastructure services.
- **Repository Layer:** Contains repository interfaces and their implementations.

Ensure proper dependency management among these three layers for a clean, maintainable architecture.