# **Recipe Application**

## **Overview**

The Recipe Application is a console-based program that lets users manage recipes. Users can add ingredients and steps to create new recipes, as well as view existing ones.

**How to get started**

To use the Recipe Application, execute all of these steps:  
1. Clone the repository to your local machine.   
  
2. Navigate to the project directory.   
3. Compile and run the program with the given commands.   
4. Use the on-screen prompts to engage with the program.

Features: • Include new recipes with ingredients and processes.   
• See all recipes.   
• Find a specific recipe.   
• Warns users if a recipe's total calories surpass 300.

**Application: Adding a Recipe.**

5. Enter the recipe name when prompted.   
6. Enter the number and details of each ingredient.   
7. Enter the number of steps and describe each one.   
8. After adding the recipe, the application will display a success message.

**Viewing Recipes.**

1. Choose the option to see all recipes.   
2. The app will display all recipes with their details.   
3. You can display certain recipes by inputting their names.

**Exiting the Application**

4. Choose the option to exit the app.

**Dependencies**

The Recipe Application is built with.NET Core and has no external dependencies.

**Description:**

Part 2 code is more clearly arranged into separate classes for Ingredient, Step, and Recipe. This increases code readability and maintainability by adhering to the single responsibility principle.I updated the characteristics of the encapsulated Ingredient and Step classes, allowing for more control over data access.

The Recipe class in part 2 code initializes its Ingredients and Steps arrays during the constructor, ensuring that they are properly initialized when a Recipe instance is generated. The Recipe class in part 2 code defines an event (CaloriesExceeded) and subscribes to it within the Main method, demonstrating knowledge of event-driven programming.

The part 2 code improves input validation for user-entered quantities, ensuring that erroneous input is handled gracefully.

Part 2 code adds a menu-driven interface to interact with the Recipe Application, making it more user-friendly and intuitive.

Overall, the part 2 code shows stronger object-oriented design principles, improved encapsulation, and more robust error handling than the part 1 code. These improvements help to increase code quality and maintainability.

GitHub link:

https://github.com/ST10126814/PROG6221-PEO-Part2.git

**Screenshot:**

