**Recipe App**

Welcome to the magical world of our Recipe App!

Have you ever found yourself in the kitchen, scratching your heard and wondering what to cook with the random ingredients you have? Or maybe you’re on a mission to eat healthier and need to keep those calories in check? Fear not, because our Recipe App is here to save the day!

Our delightful app lets you dive into a treasure trove of recipes, filtering out the perfect dish based on your whims and fancies. Whether you’re in the mood for something with a specific ingredient want to stick to a particular food group, or need to watch those calories, Recipe App has got your back.

So grab your chef’s hat, roll up your sleeves, and let’s cook up some magic!

**Getting Started**

**Follow these steps to compile and run the program: Requirements**

Before you start, make sure you have the following installed on your computer || Visual Studio || Git :

1. Clone the GitHub repository to your local machine using Git. Open your command-line terminal and run the following command: bashCopy code git clone
2. Open Visual Studio.
3. Click “File”> “Open” > “Project/Solution”.
4. Navigate to the directory where you cloned the repository.
5. Select the RecipeAppPart3.sIn solution file and click “Open”.
6. Once the project is open, build the solution by clicking “Build” > “Build Solution” in Visual Studio.

**Project Structure**

**MainWindow.xaml**

Defines the user interface for the application. It includes:

* Text boxes for entering filter criteria (ingredient name, food group, and maximum calories).
* Buttons for triggering the filter actions.
* A ListBox for displaying the filter list of recipes.

**MainWindow.xaml.cs**

Contains the interaction logic for the UI. It handles button clicks and displays the filtered recipes based on user input.

**AppMethods.cs**

Contains methods for managing the recipe data. It includes:

* Methods to add sample recipes for demonstration purposes.
* Methods to filter recipes by ingredient name, food group, and maximum calories.

**Recipe.cs**

Defines the “Recipe” class which include:

* Properties for the recipe name, ingredients, and steps.
* Methods to add ingredients and steps.
* A method to calculate the total calories.

**IngredientInfo.cs**

Defines the “IngredientInfo” class which include:

* Properties for the ingredient name, quantity, unit of measurement, number of calories, and food group.

**Brief Description**

In this project I refactored the RecipeApp to adhere to the principles of separation of concerns and single responsibility. Previously the “AppMethods” class was handling multiple responsibilities. Including managing recipes and handling user input/output. To address this, I made the following changes:

1. Separation of User Interface and Business Logic: this user interface logic is now contained within “Mainwindow.xaml.cs”, which handles user interactions and updates the UI. The business logic is encapsulated in separate classes (“AppMethods”, “Recipe”, and “IngredientInfo”).
2. Single Responsibility Principle:

* The “AppMethods” class now focuses solely on managing recipe operations, such as adding, filtering, and retrieving recipes.
* The “MainWindow” class handles user input and output, ensuring that the UI logic is distinct from the recipe management logic.

1. New Classes for Recipe Management:

* “Recipe”: manages individual recipe details, ingredients, and steps.
* “IngredientInfo”: encapsulates details about each ingredient, including name, quantity, unit, calories and food group.

By restricting the code in this manner, my application is now more modular, easier to maintain, and aligns with best practices for software design. Each class and method has a clear , responsibility, improving the overall readability and manageability of the code.