**Recipe Management System Assignment**

**Objective**

In this assignment, you will build a **Recipe Management System** using Python that allows users to manage and display recipes from an **Excel file**. You will demonstrate your understanding of **object-oriented programming (OOP)** concepts such as **classes, inheritance, file handling, exception handling, and string processing**.

**Requirements**

1. **Create the Recipe Class and Subclasses**
   * Implement a base class Recipe with the following:
     + **Attributes**:
       - name: Name of the recipe
       - ingredients: A list of ingredients
       - steps: Instructions for preparing the recipe
       - is\_vegetarian (boolean).
     + **Methods**:
       - display\_recipe(): Prints the recipe details (name, ingredients, and steps).
       - Getters and setters for attributes
   * Implement the following subclasses, inheriting from Recipe:
     + **DessertRecipe**
       - Additional attribute: sweetness\_level (integer from 1 to 10).
       - Override display\_recipe() to include the sweetness level.
     + **MainCourseRecipe**
       - Additional attribute: spice\_level (integer from 1 to 5).
       - Override display\_recipe() to include the spice level.
2. **All attributes MUST be protected.**
3. **Read Data from an Excel Sheet**
   * Load recipe data from an **Excel file (recipe\_data.xlsx)**
   * Using the pandas library to read the data is optional
4. **Create Instances from the Excel Sheet**
   * Use the data from the file to create instances of Recipe and its subclasses based on the **Type** column.
   * Ensure each instance correctly stores its attributes.
5. **Implement Required Methods for subclasses**
   * Each sub class must have the following methods that performs a meaningful action:
     + Formatting and displaying recipe details.
     + Adjusting spice or sweetness levels.
6. **Make program Menu Driven**
   * Implement a text-based menu that allows users to:
     + **1) View all recipes**
     + **2) Search for a recipe by name**
     + **3) Filter recipes by type (Dessert, Main Course, Vegetarian)**
     + **4) Exit the program**
7. **Implement Exception Handling**
   * Handle errors such as:
     + File not found (FileNotFoundError)
     + Invalid user input (ValueError)
     + Missing data in the Excel sheet (KeyError)
8. **String Processing**
   * Format and display recipe details neatly.
   * There should be no restrictions on searching recipes. If a recipe exists , the program is to retrieve it regardless of how the name of the recipe was entered.
   * Process string data such as capitalizing names, splitting ingredients into lists, and ensuring readable output.
9. **UML Diagram**
   * Create a UML Diagram for the classes created. Make sure inheritance is depicted.