Solution 3.1

Overview

Solution 1 proposes building a basic Java desktop application for recipe management, where each recipe is stored as a separate plain text file (.txt) on the user's local computer. The application provides a minimal graphical user interface (GUI) for users to create, view, and delete recipes. All recipes are saved in a single folder, with each recipe occupying its own file.

How It Works

Recipe Creation

- The user clicks an "Add Recipe" button in the application.
- A form appears, allowing the user to enter the recipe title, ingredients, and instructions.
- When the user clicks "Save," the application concatenates all the entered information into a single string and writes it to a new .txt file.
- The file is typically named after the recipe title (e.g., ChocolateCake.txt). If a file with the same name exists, a timestamp or number may be appended to avoid overwriting.

Recipe Viewing

- The application scans the designated recipes folder for all .txt files.
- It displays a list of these files (usually by filename) to the user.
- When a user selects a recipe, the application reads the contents of the corresponding .txt file and displays it in a simple, scrollable text area.

Recipe Deletion

- The user selects a recipe from the list and clicks a "Delete" button.
- The application deletes the corresponding .txt file from the folder.

User Interface

- The GUI is typically composed of:
- A list or table showing all recipe files.
- Buttons for "Add Recipe," "View Recipe," and "Delete Recipe."
- A text area for displaying or editing recipe contents.

Example Recipe File Content

Recipe Title: Chocolate Cake

Ingredients:

```
- 2 cups flour

- 1 cup sugar

- 1/2 cup cocoa powder

- 2 eggs

Instructions:

1. Preheat oven to 350F.

2. Mix all ingredients.

3. Bake for 30 minutes.
```

Features and Limitations

Supported Features

- **Basic CRUD:** Create, Read, and Delete recipes.
- **Offline Access:** All data is local; no internet required.
- **Simplicity:** Easy to implement and understand.

Limitations

- **No Structure:** The data is just plain text, so there is no enforced format for ingredients, steps, or other fields.
- **No Organization:** No support for tags, categories, favorites, or sorting other than by filename.
- **No Media Support:** Cannot attach images or multimedia to recipes.
- **No Search or Filter:** Users cannot search for recipes by ingredient, tag, or keyword.
- **No Import/Export:** Recipes cannot be easily shared or backed up except by copying files manually.
- **No Validation:** There is no way to ensure users enter all necessary information or use a consistent format.

Testing and Validation Issues

- **Difficult to Test Data Integrity:** Since the file format is not enforced, automated tests cannot reliably check for missing or malformed fields.
- **Limited Test Automation:** Automated tests can only verify file creation, reading, and deletion—not the correctness of recipe content.
- **No Advanced Features to Test:** Search, filter, tagging, and image attachment cannot be tested because they do not exist.

- **No Input Validation:** The application cannot prevent invalid or incomplete recipes from being saved, making boundary value and equivalence class testing meaningless.

Why This Solution Is Not Selected

- **Poor Testability:** Most software testing techniques (e.g., boundary value, equivalence class, state transition, decision tables) cannot be applied to unstructured text files.
- **Lack of Features:** Does not meet the project requirements for organization, searchability, or extensibility.
- **Not Scalable:** As the number of recipes grows, managing them becomes cumbersome and error-prone.
- **Difficult Maintenance:** Any future improvements (like adding tags or images) would require a complete redesign.

In summary:
Solution 1a is a simple and easy-to-implement approach that works for the most basic use case of storing and viewing recipes. However, it lacks structure, organization, and extensibility, making it unsuitable for a robust, testable, and maintainable recipe management system.