TERM PROJECT REPORT

İBRAHİM BANCAR
150220313

10.06.2024

Contents

DELİCİOUS RECİPES

İbrahim BANCAR

Istanbul Technical University
Faculty of Computer and
Informatics Engineering
Artifical Intelligence and Data
Engineering

Istanbul,Turkey

bancar22@itu.edu.tr

Abstract—**Delicious Recipes Project helps users effectively manage their recipe collection. Recipes can be added, displayed, edited, deleted, and searched for using keywords by users via methods in a class structure. The data is persistent while the program is running since the recipes are dumped in a JSON file and loaded from there. This report describes how this project will be used, how it receives data from the user, how the data is processed and this data is stored. It is supported by a conclusion and recommendations section and references are provided.**

Keywords—***JSON, recipe, class, dump, load, method, data***

I. INTRODUCTION

The Delicious Recipes project is a Python-based application designed to provide users with an efficient and easy-to-use platform for managing their recipes. The primary purpose of this project is to demonstrate the application of Python in creating an independent program which addresses everyday tasks, such as storing and organizing recipes. The creation of a command-line program with file handling, data management, and user interaction capabilities utilizing Python standard libraries is part of the project's scope. The project's goal is to construct an intuitive user-program interface that provides a reliable and simple user experience. In particular, the program lets users save and load data from a JSON file in addition to adding, editing, deleting, and searching for recipes. This report describes the project's development process, features that have been put into use, and possible improvements in the future. The project guarantees the accuracy and effective processing of user inputs by utilizing Python's built-in features, making it a dependable tool for recipe management.

II. METHODS

A. Initialization of the project

When the program is started, firstly, the "RecipeManager" class initializes by loading existing recipes from a "recipes.json" file or creating a new file if it does not exist.

B. Loading and Saving Recipes

The "load_recipes" method checks if the "recipes.json" file exists by utilizing the standard OS module of Python. If it does, it loads the recipes; if not, it creates a new file. The "save_recipes" method saves the current list of recipes to the "recipes.json" file in a dictionary format.

C. Adding a Recipe

The "add_recipe" method prompts the user for the recipe name, ingredients, and step-by-step instructions. It ensures the recipe name is unique before adding it to the list and saving it to the JSON file. If the recipe name has been already present then it warns the user by if-else blocks.

D. Displaying a Recipe

The "display_recipe" method allows the user to display a specific recipe by name. If no name is provided, it prompts the user to enter one. It searches for the recipe in the list and displays its details if found.

E. Deleting a Recipe

The "delete_recipe" method allows the user to delete a recipe by name. It prompts the user for the recipe name and removes it from the list if found.

F. Editing a Recipe

The "edit_recipe" method allows the user to edit the details of an existing recipe. It prompts the user for the recipe name and the new details, then updates the list and saves it to the JSON file. It proceeds by asking again for the recipe name, ingredients and instructions.

G. Searching Recipes by Keyword

The "search_recipe_by_keyword" method allows the user to search for recipes containing a specific keyword in the ingredients or instructions. It searches all

dictionaries that are included in the list and accesses the "ingredients" and "instructions" keys of dictionaries respectively finally displays all matching recipes.

H. Displaying User Manual

The "display_user_manual" method provides instructions on how to use the program, detailing the options available to the user.

I. Main Loop

The main loop of the program continuously prompts the user for their choice and calls the appropriate method based on the input. Whenever user inputs "exist", it breaks the while loop and finishes the running program.

III. DISCUSSION

The Delicious Recipes project was designed to efficiently manage a collection of recipes, providing users with the ability to add, display, edit, search, and delete recipes. The functionality of the program is built on a intense structure using Python's standard libraries, ensuring a seamless and efficient user experience.

A. User Interaction

1. Adding Recipes

- Users are guided through entering the recipe name, ingredients, and instructions.

- The program checks for duplicate names, ensuring each recipe is unique.

- For ease of retrieval and display, ingredients and instructions are processed and saved in a list format.

2. Displaying Recipes

- Users can display all recipes or search for a specific recipe by name.

- The program formats the recipe details clearly, listing ingredients and step-by-step instructions.

3. Editing Recipes

- Users can update the name, ingredients, and instructions of an existing recipe.

- The application makes it simpler for users to update by providing recent data as a reference.

4. Deleting Recipes

- Users can delete specific recipes or clear the entire data

- The program prompts for confirmation, preventing accidental deletions.

5. Searching by Keyword

- Users can search for recipes containing specific keywords in the ingredients or instructions.

- The user experience is improved by the effective search functionality that offers relevant outcomes.

B. Data Management

- **Persistence**: Recipes are saved between sessions, allowing users to access their data anytime.

- **Scalability**: The structure of JSON allows for easy expansion of the recipe database.

- **Portability:** JSON files are lightweight and easily transferable between different systems.

C. Figures and Tables

Figures and tables are not included in the project code itself, but the information is nevertheless communicated effectively thanks to the logical structure and understandable output formatting. For example, when a recipe is displayed, its ingredients and instructions are provided in an easy-to-read, well-organized format that acts as an implicit table of contents.

```
Recipe: pizza

Ingredients:
- yeast
- water
- flour
- oil
- salt

Instructions:
1. Heat the oven to 550°F or higher.
2. Divide the dough in half
3. Roll out the dough
4. Top the pizza
5. Bake the pizza
6. Service
Enter your choice:
```

D. Supporting Information

Additional supporting information, such as error handling processes and user feedback mechanisms, are included within the code comments and documentation.

IV. CONCLUSION

1. **Effective Data Management:** The Delicious Recipes project effectively manages recipe data using JSON files, ensuring persistence, portability, and scalability. This approach allows users to store and retrieve their recipes seamlessly between sessions.

2. **User-Friendly Interface**: The program provides a user-friendly interface with clear prompts and instructions, enabling users to add, display, edit, search, and delete recipes efficiently. This design ensures accessibility for users with a variety of levels of technical expertise.

3. **Error Handling and Feedback:** The program includes an error handling and user feedback mechanisms, ensuring smooth operation and preventing common user errors

4. **Strong Functionality:** The project offers strong functionality, including unique recipe naming, detailed ingredient and instruction management, and efficient search capabilities. These features enhance the overall user experience and usability of the program.

V. RECOMMENDATIONS

1. **Expand Recipe Categories:** Based on the conclusion that the current data management and user interface are effective, it is recommended to expand the functionality to include recipe categories. This could enhance the user experience by allowing users to organize recipes by type (e.g., desserts, main courses, appetizers).

2. **Implement User Accounts:** Considering the program's current efficiency in managing recipes, adding user account functionality could provide personalized experiences. Users could save their favorite recipes, create shopping lists, and share recipes with others.

3. **Add Advanced Search Options:** Building on the robust search capabilities, implementing advanced search options such as filtering by ingredients, preparation time, or dietary preferences could further improve usability and help users find specific recipes more efficiently.

4. **Mobile Application Development**: To reach a broader audience and enhance accessibility, developing a mobile application version of the Delicious Recipes project is recommended. This would allow users to access and manage their recipes easily.

REFERENCES

All bibliographic entries are listed alphabetically by the name of the author or the first major word of the title

[1] Vetterli, C. "Work Term Report Guidelines," Co-op Journal, Vol. 1 No. 1, August 1992. *(references)*

[2] Python Software Foundation, "Python Documentation," 2023. Available at: https://docs.python.org/3/

[3] McKinney, W. "Python for Data Analysis," O'Reilly Media, 2012. ISBN: 978-1-4493-1979-3.

[4] Lundh, F. "Python Standard Library," O'Reilly Media, 2001. ISBN: 978-0-596-00096-3.