iRobot video interview

4/19/2019

Omkar Desai

Assignment

Build a Python application to search for recipes online and suggest a recipe to the user based on the ingredients available to him

Tasks

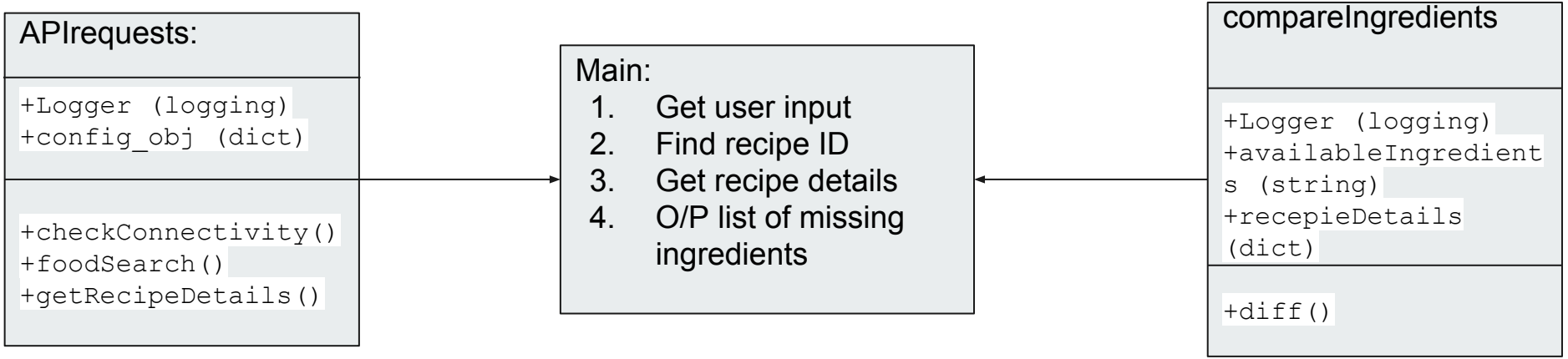
1. Use the food2fork API to get recipes using the available ingredients
2. Suggest the highest rated recipe that uses all the available ingredients
3. Compare the available ingredients with the ingredients required to make the suggested recipe
4. Provide the user with the list of missing ingredients

Application screenshots- Successful execution

```
Please provide all the ingredients available with you seperated by comas:
chicken, tomatoes
```

```
Here are the ingredients you will need to make Chicken with Herb-Roasted Tomatoes and Pan Sauce :
6 tablespoons olive oil, divided
2 tablespoons herbes de Provence
1 teaspoon kosher salt plus more
Freshly ground black pepper
1 tablespoons Worcestershire sauce
1 small shallot, minced
2 tablespoons red wine vinegar
3 tablespoons flat-leaf parsley leaves
3 tablespoons fresh tarragon leaves
Total execution time including API calls: 3.9238905906677246 s
Total execution time excluding API calls: 0.024112939834594727 s
```

UML


APIrequests:
+Logger (logging)
+config_obj (dict)
+checkConnectivity()
+foodSearch()
+getRecipeDetails()
Main:
1. Get user input
2. Find recipe ID
3. Get recipe details
4. O/P list of missing ingredients
compareIngredients
+Logger (logging)
+availableIngredients (string)
+recepieDetails (dict)
+diff()

Application architecture

The application is divided into 2 classes:

- APIrequests.py
  - Ensuring network connectivity
  - Making all API calls to food2fork's endpoint
  - User key management

- compareIngredients.py
  - Comparing the available ingredients with the required ingredients for the recipe
  - Can be further extended to support a lot of post processing. Eg: Providing cooking times or detailed instructions.

Result and potential improvements

1. Application is divided into classes with specific responsibilities so it is maintainable
2. Error messages, error handling and instructions are easy to understand
3. Logging of events and errors helps with future maintenance and debugging
4. Unit testing and integration testing ensure that changes don't break old code

Improvements:

1. API key should not be stored, it should be taken as input
2. A GUI using the 'tkinter' package would make the app more user friendly
3. Using a documentation tool like Sphinx or readme.io