**testDeleteRecipe.py**

*Test Plan for* ***deleteRecipe***

1. **Test Case:** Deleting a Recipe by ID

* **Description:** This test verifies the basic functionality of the deleteRecipe method by deleting a recipe with an ID and checking if the recipe is deleted from the collection.
* **Expected Outcome:** The recipe with the specified ID should be deleted from the recipe manager's data.

1. **Test Case:** Deleting a Non-Existent ID

* **Description:** This test ensures that the deleteRecipe method handles the deletion of a non-existent recipe ID correctly.
* **Expected Outcome:** The method should not delete anything.

1. Test Case: Deleting a Recipe with a String ID

* **Description:** This test validates the behavior of the deleteRecipe method when a recipe with a string ID is attempted to be deleted.
* **Expected Outcome:** The method should not delete anything.

**testAddDelete.py**

*Test Plan for* ***testAddDelete***

1. Test Case: Adding then Deleting a Recipe

* **Description:** Verifies the basic functionalities of adding and deleting a recipe.
* **Expected Outcome:** The recipe is added to the data and then deleted right after.

1. Test Case: Adding a Recipe with a Recently Deleted ID

* **Description:** Checks whether the addRecipe method correctly adds a recipe with a previously deleted ID to the data.
* **Expected:** The recipe is added to the data.

**recipeProject.py**

1. deleteRecipe(self, id) – removes a recipe from the collection using the given id.

* *parameter* id: ID of the recipe to be deleted.
* **Returns:** None

1. exportRecipesGUI(self) – converts data into JSON format. An excerpt from exportRecipes, it is made solely for GUI implementation.

* **Returns:** List of dictionaries representing the recipes

1. importRecipesGUI(self) – reads JSON files and puts them into data. An excerpt from importRecipes, it is made solely for GUI implementation.

* **Returns:** Returns True if the file is imported successfully.

**rm\_shell.py**

1. do\_delete(self, arg) – command-line method that allows the user to delete a recipe from the memory. The method does not return any value.

* *parameter* arg: ID of the recipe to be deleted.
* **Usage:** delete <id>
* **Returns:** None

**app.py**

1. delete\_id(id) – prompts the user to confirm deletion of a specific recipe.

* **Route:** /delete/<id>
* **Returns:** Renders index.html template with recipes in the data.

1. deleteConfirm\_id(id) – confirms the deletion of a recipe with the specified ID.

* **Route:** /deleteConfirm/<id>
* **Returns:** Renders index.html template with recipes in the data.

1. export() – exports recipes to a JSON file, using the json library. Utilizes tkinter's filedialog library to allow users to select a file location and name.

* **Route:** /export
* **Returns:** Renders index.html template with recipes in the data.

1. import\_file() – imports recipes from a JSON file, using the json library. Utilizes tkinter's filedialog library to allow users to select a file from a dialog box.

* **Route:** /import
* **Returns:** Renders index.html template with recipes from the imported JSON.