

## SQPT In Class Mock

### Instructions:

1. Along with this document file there is a zip project file, which needs to be downloaded and unzipped.
2. Based on the project source code, you must create possible functional requirements, traceability matrix and test cases in a word document file.
3. Inside of a project there is a TestProject2 under which you can find a file UnitTest1.cs, you are required to write your unit test on this file. To submit, you need to zip it back to a single file (0000XXStudentID.zip) and upload it back to LMS into in class assessment of SQPT Module.
4. Please, include your resulting word document into above 0000XXStudentID.zip file as well.

**Task #1:** Create **ten functional requirements** for a console-based Recipe Management System where users can create, edit, and delete recipes. Include features like search, detailed info. (see attached project source code for reference)

**Total: 10 points.**

FR-1	<b>User Authentication:</b> Users must be able to log in securely to access their recipes and perform actions such as creating, editing, and deleting recipes.
FR-2	<b>Recipe Creation:</b> Users should be able to create new recipes by providing details.
FR-3	<b>Recipe Editing:</b> Users should have the ability to edit existing recipes
FR-4	<b>Recipe Deletion:</b> Users should be able to delete recipes they no longer wish to keep in their collection.
FR-5	<b>Search Functionality:</b> The system should allow users to search for recipes based on keywords, ingredients, tags, or categories to quickly find relevant recipes.
FR-6	<b>View Detailed Recipe Information:</b> Users should be able to view detailed information about each recipe, including ingredients, instructions, and any additional notes or tips.
FR-7	<b>Sorting and Filtering:</b> Users should have options to sort and filter their recipe collection based on various criteria such as alphabetical order, date added, or category.

FR-8	<b>Recipe Rating and Reviews:</b> Users should have the ability to rate recipes and leave reviews, allowing for feedback from the community and helping others make informed decisions about which recipes to try.
FR-9	<b>Recipe Sharing:</b> Users should be able to share their favorite recipes with others via email, social media, or other communication channels.
FR-10	<b>Multilingual Support:</b> The system should support multiple languages to accommodate users from diverse backgrounds and regions.

***Task #2: Create possible **ten test cases** for created functional requirements, each marked as 3 points. Total: 30 Points***

Test Case ID	Purpose	Test Steps	Expected Result	Actual Result
TC_01_Login	To verify user login functionality	1. Launch the Recipe Management System. 2. Enter valid login credentials. 3. Click on the login button.	User is logged in and directed to the main menu.	User is successfully logged in and directed to the main menu.
TC_02_Create	To ensure creation of a new recipe	1. Access the "Create Recipe" option from the main menu. 2. Fill in all required fields with valid information. 3. Save the recipe.	Recipe is successfully created and added to the user's collection.	Recipe is successfully created and added to the user's collection.
TC_03_Edit	To verify editing of an	1. Navigate to the "Edit	Changes to the recipe	Changes to the recipe

	existing recipe	Recipe" option in the menu. 2. Select a recipe to edit. 3. Make changes to the recipe details. 4. Save the changes.	are saved successfully.	are saved successfully.
TC_04_Delete	To ensure deletion of a recipe	1. Go to the "Delete Recipe" option in the menu. 2. Choose a recipe to delete. 3. Confirm the deletion.	Selected recipe is removed from the user's collection.	Selected recipe is removed from the user's collection.
TC_05_Search	To test the search functionality	1. Access the search option in the menu. 2. Enter relevant keywords related to a recipe. 3. Initiate the search.	System displays a list of recipes matching the search criteria.	System displays a list of recipes matching the search criteria.
TC_06_View	To verify viewing detailed information of a recipe	1. Select a recipe from the user's collection. 2. Choose the option to view detailed information.	Detailed information about the selected recipe is displayed.	Detailed information about the selected recipe is displayed.
TC_07_Sort	To test sorting functionality	1. Access the sorting options in the menu. 2.	Recipes are displayed in	Recipes are displayed in

	by alphabetical order	Choose to sort recipes alphabetically.	alphabetical order.	alphabetical order.
TC_08_Import	To ensure successful import of recipes from external file	1. Select the option to import recipes. 2. Choose a valid file containing recipe data in supported format. 3. Confirm the import process.	Recipes from the file are successfully imported into the user's collection.	Recipes from the file are successfully imported into the user's collection.
TC_09_Multilingual_Support	To ensure the system supports multiple languages	1. Change the language setting of the system to a different supported language. 2. Navigate through the menu options and interface.	All menu options and interface elements are displayed in the selected language.	All menu options and interface elements are displayed in the selected language.
TC_10_Logout	To verify user logout functionality	1. Navigate to the logout option in the menu. 2. Click on the logout button.	User is logged out and returned to the login screen.	User is logged out and returned to the login screen.

**Task #3:** Create traceability matrix based on the above test cases and functional requirements. The traceability matrix must contain at least five rows, each marked as 1 points. **Total: 5 points**

Test Case ID	Requirement ID
FR-1 User Authentication	TC_01_Login
FR-2 Recipe Creation	TC_02_Create
FR-3 Recipe Editing	TC_03_Edit
FR-4 Recipe Deletion	TC_04_Delete
FR-5 Search Functionality	TC_05_Search
FR-6 View Detailed Recipe Information	TC_06_View
FR-7 Sorting and Filtering	TC_07_Sort
FR-8 Recipe Import	TC_08_Import
FR-10 Multilingual Support	TC_09_Multilingual_Support
FR-1 User Authentication	TC_10_Logout

**Task #4:** Create **ten MSTEST/NUNIT** unit tests for attached code based on your test cases, each marked as 3 points. **Total: 30 points**

**Task #5:** Use Moq Nuget.Package to Mock Actions log for actions such as Add, Update and Delete, use the necessary constructor for that. Create at least three different Unit Test with the usage of Mock<ActionLogger> to Verify each action method with invocations objects such as Times.Once or Never. (5 points each)

**Total: 15 Points**

**Task#6:** In this task you are required to automate the Yahoo.com search with keyword "WIUT". After updating the following code, the browser of your choice/version installed in a PC lab must be automatically started (1), browses into yahoo.com (2), puts the keywords in input box (3) and launches the search (4). At last (5), you must display the first found result title in VS console. Each working correct step updated in the below code is worth 2 points.

**Total: 10 Points**