Test Case Template

Module: Home Menu

Test Strategy: Exploratory with positive/negative scenario

N	TC ID	Test Case Description	Test Steps	Expected Result	Actual Result	Pass/ Fail	Related Defects
1	TC_01	Validating welcome message is displayed	1. Run the program	“Welcome!” message is visible after running the program	“Welcome!” message is displayed		
2	TC_02	Validating the 3 options of the menu are displayed	1. Run the program	All the 3 options are displayed in numeric points (1, 2, 3).	All options are visible to the user		
3	TC_03	Inserting a valid option and validating that the expected screen is displayed	1. Run the program 2. Insert option ‘1’	“Data entry in the system” message is displayed, along with the customer ID input.	The correct screen was received, and the first input was prompted.		
4	TC_04	Inserting option 2 and validating that client cannot access it without having gone to option 1.	1. Run the program 2. Insert option ‘2’	An error message is displayed to the user that needs to access option 1 first.	“You must enter the information first” was received, and the menu was received again.		
5	TC_05	Insert option 3 to confirm that the system is closed correctly.	1. Run the program 2. Insert option ‘3’	A confirmation message that the system is closed should be received.	“Existing the system, see you soon!” after inserting option 3.		
6	TC_06	Inserting an invalid option: 0 and 4.	1. Run the program 2. Insert option ‘0’ 3. Reset and type ‘4’	An error message is received, and the menu is displayed again.	“Invalid option” error message is received, and the menu is prompted		
7	TC_07	Validating letters in lowercase and uppercase are not accepted as options	1. Run the program 2. Type “test string” 3. Reset and type “TEST STRING”	The invalid error message is received, and the menu is displayed again.	“Invalid option” error message is received, and the menu is prompted		
8	TC_08	Validating letters first with a number is not accepted	1. Run the program 2. Type ‘a1’, ‘a2’, ‘a3’	An error message is received, and the menu is displayed again.	“Invalid option” error message is received, and the menu is prompted		
9	TC_09	Validating numbers first with a letter is not accepted	1. Run the program 2. Type ‘1a’, ‘2a’, ‘3a’	An error message is received, and the menu is displayed again.	After entering the number at the beginning, it is ignoring all the next characters, so it is moving forward with no error.		IB-5

10	TC_10	Validating that “paste” functionality works with valid inputs	1. Run the program 2. Copy ‘1’ 3. Paste it on the input 4. Tap ‘enter’ 5. Repeat by copying ‘3’ instead	After passing number 1 and hitting enter, it should access the data entry section. For option 2 it should retrieve error message. For number 3, it should close the system.	Pasting number 1 + enter goes the data entry. Pasting number 2 + enter displays the expected error. Pasting number 3 + enter closes the system.		
11	TC_11	Validating that “paste” functionality works with invalid inputs	1. Run the program 2. Copy ‘TEST’ 3. Paste it on the input 4. Tap ‘enter’ 5. Repeat with a number 6. Repeat with special charact 7. Repeat with ‘a1’, ‘a2’, ‘a3’ 8. Repeat with ‘1a’, ‘2b’, ‘3c’.	An error message is received, and the menu is displayed again.	After entering the number at the beginning, it is ignoring all the next characters, so it is moving forward with no error.		IB-5
12	TC_12	Validating empty spaces retrieve an error message	1. Run the program 2. Hit enter without typing anything	An error message is received, and the menu is displayed again.	“Invalid option” error message is received, and the menu is prompted		
13	TC_13	Valid numbers with spaces are allowed	1. Run the program 2. Type ‘1 ‘ 3. Hit enter	Users should be able to proceed, because in this scenario we don’t want to block the customer to only insert 1 character	After inserting a valid option and spaces, the data entry section is displayed		
14	TC_14	Valid numbers with spaces and additional letters are not allowed	1. Run the program 2. Type ‘1 sdsd ‘ 3. Hit enter	An error message is received, and the menu is displayed again.	After entering the number at the beginning, it is ignoring all the next characters, so it is moving forward with no error.		IB-5
15	TC_15	Validating special characters are not allowed	1. Run the program 2. Type “[][]{}_+.,,.”	An error message is received, and the menu is displayed again.	“Invalid option” error message is received, and the menu is prompted		

Module: Option 1 – Data Entry Section

Test Strategy: Boundary Value Analysis | Exploratory: Negative

Pre-condition: 1. Run the program > 2. Access option 1.

N	TC ID	Test Case Description	Test Steps	Expected Result	Actual Result	Pass/Fail	Related Defects
16	TC_16	*Customer ID* input with BDA Data (in amount on numbers): negative, 8, 9, 10, 50	1. Insert the numbers mentioned in the data 2. Insert the 9-digit number	1. For negative, 8, 10, 50 should return “Invalid ID number, please try again!”. Re-enter menu is received. 2. For 9 numbers, should move forward to “full name” entry.	1. Expected error message is received. 2. Moved forward to second input		
17	TC_17	*Customer ID* input with negative entry	1. Submit with no data 2. Submit 9 numbers separated by spaces 3. Insert a letter and 9 numbers 4. Insert 9 numbers and a letter 5. Insert special characters	1. Error messages are triggered for all scenarios	For step 4, we are allowed to insert a letter at the end of the number and goes through.		IB-5 Empty data is not retrieving any error, returning an error after submitting empty data would be great to add.
18	TC_18	*Full name* input negative	1. Insert a number 2. A letter with a number 3. Submit empty input	1. Error messages are triggered for all scenarios	1. Submitting empty input is moving to the next section 2. Number is moving forward as valid name		IB-3
19	TC_19	*Full name* input positive	1. Insert a valid name, last name and second name 2. Localization ES: Submit: ñ and é	1. For both, the user is redirected to “invoice number” input.	1. Successfully redirected to the invoice number input.		
20	TC_20	*Invoice ID* input with BDA Data (in amount on numbers): negative, 5, 6, 7	1. Insert the numbers mentioned in the data 2. Insert the 6-digit number 3. Type 000012.	1. For negative, 5, 7 should return “Invalid ID number, please try again!”. Re-enter menu is received. 2, 3. For 6 numbers, should move forward to “hours watched” entry.	1 and 2 working as expected 3. The 0000 is not being considered as number, is retrieving error message		IB-4
21	TC_21	*Invoice ID* input with negative entry	1. Submit with no data 2. Submit 6 numbers separated by spaces	1. Error messages are triggered for all scenarios	Step 4 is moving forward as valid and affects the “hours watched” entry.		IB-5

			3. Insert a letter and 6 numbers 4. Insert 6 numbers and a letter 5. Insert special characters				
22	TC_22	*Hours Watched* input with BDA Data: -1, 0, 1, 9, 10, 11, 12, 39, 40, 41, 50, 51, 52, 99, 100, 101 Adding also: 719, 720, 721	1. Insert the numbers mentioned on the test description data.	1. For -1 an error should be received 2. For 0 it should allow user to proceed and select minutes instead 3. For 1, 9 and 10 > no additional charge is received 4. For 11, 12, 39, 40, 41, 50 > 10% is charged 5. For 51, 52, 99, 100 > 10% + 20% is charged 6. For 101, 719, 720 > 10% + 20% + 30% is charged 7. For 721, an error is received.	All entries are working as expected, however, we need to allow users to enter minutes watched if the hours watched are less than 1.		Suggestion: IB-1
23	TC_23	*Hours Watched* Negative input	1. Insert a letter 2. Type a special character 3. Type with ‘a1’, ‘a2’, ‘a3’ 4. Type with ‘1a’, ‘2b’, ‘3c’.	1. Should get an error message for all inputs	Step 4 allows you to move forward to the next input.		IB-5
24	TC_24	*Exempt from taxes*? Exploratory: Positive-Negative	1. Insert any letter != Y || N 2. Insert a number 3. Submit empty 4. Type ‘y y y’ or ‘n n n’ 5. Type ‘1y’ and ‘y1’ 6. Type a special character 7. Type ‘y’ or ‘Y’ 8. Type ‘n’ or ‘N’	1. Error messages are received from step 1 to 6. 7. No taxes are applied on the total price 8. Taxes are applied, 13% of the total amount to be paid	In step 4, typing “y y y” is saving the 2 additional “y” in the buffer and is being used for the next inputs. For “n n n”, is displaying the same menu 3 times.		IB-6
25	TC_25	Yes/No Menu Exploratory: Positive-Negative	1. Insert any letter != Y || N 2. Insert a number 3. Submit empty 4. Type ‘y y y’ or ‘n n n’ 5. Type ‘1y’ and ‘y1’ 6. Type a special character 7. Type ‘y’ or ‘Y’ 8. Type ‘n’ or ‘N’	1. Error messages are received from step 1 to 6. 2. Step 7 redirects to the main menu 3. Step 8 continue to show the same question until step 7 is triggered.	For “y y y” and “n n n”, is displaying the same menu 3 times.		IB-6

Module: Option 2 – Data Entry Section

Test Strategy: Exploratory: Positive – Negative

N	TC ID	Test Case Description	Test Steps	Expected Result	Actual Result	Pass/Fail	Related Defects
26	TC_26	Confirm structure of the receipt is the expected	1. Run the program 2. Complete TC Option 1 3. When back in menu, type option 2.	The format is the expected and no non-related value is received	The random Number generated to complete the payment is shown at the very top of the receipt		IB-2
27	TC_27	Retrieving information from TC Option 1.	1. Run the program 2. Complete TC Option 1 3. When back in menu, type option 2.	The results are displayed based on the selection of TC Option 1.	Data is presented as expected and calculations are correct.		
28	TC_28	Yes/No Menu Exploratory: Positive-Negative	1. Run the program 2. Insert any letter besides Y or N 3. Insert a number 4. Submit empty 5. Type ‘y y y’ or ‘n n n’ 6. Type ‘1y’ and ‘y1’ 7. Type a special character 8. Type ‘y’ or ‘Y’ 9. Type ‘n’ or ‘N’	1. Error messages are received from step 1 to 6. 2. Step 7 redirects to the main menu 3. Step 8 continue to show the same question until step 7 is triggered.	For “y y y” and “n n n”, is displaying the same menu 3 times.		IB-6

Module: Whole program
Test Strategy: E2E for price calculation.
Precondition: Run the program and access option 1.

N	TC ID	Test Case Description	Test Steps	Expected Result	Actual Result	Pass/Fail	Related Defects
29	TC_29	Scenario with no additional charge and no tax included: 10 Hours Watched	1. Insert valid information. 2. In hours watched insert 10 3. In taxes section type 'y'. 4. Type 'Y' to return to the main menu and access option 2.	1. No tax is applied, and the final price is broken down: 15$ fixed expense 10 x 2= $20 Without taxes: $35	$35 is the amount in the total pay section, and expected data is received as well.		
30	TC_30	Scenario with no additional charge and tax included: 10 Hours Watched	1. Insert valid information. 2. In hours watched insert 10 3. In taxes section type 'n'. 4. Type 'Y' to return to the main menu and access option 2.	1. Tax is applied, and the final price is broken down: 15$ fixed expense 10 x 2= $20 With taxes: $39,55	$39,55 is the amount in the total pay section, and expected data is received as well.		
31	TC_31	Scenario of additional charge of 10% and no tax included: 50 Hours Watched	1. Insert valid information. 2. In hours watched insert 50 3. In taxes section type 'y'. 4. Type 'Y' to return to the main menu and access option 2.	1. No tax is applied, and the final price is broken down: 15$ fixed expense 10 x 2= $20 40 x 2 * 1,10 = $88 Without taxes: $123	$123 is the amount in the total pay section, and expected data is received as well.		
32	TC_32	Scenario of additional charge of 10% and tax included: 50 Hours Watched	1. Insert valid information. 2. In hours watched insert 50 3. In taxes section type 'n'. 4. Type 'Y' to return to the main menu and access option 2.	1. Tax is applied, and the final price is broken down: 15$ fixed expense 10 x 2= $20 40 x 2 * 1,10 = $88 With taxes: $138,99	$138,99 is the amount in the total pay section, and expected data is received as well.		
33	TC_33	Scenario of additional charge of 20% and no tax included: 100 Hours Watched	1. Insert valid information. 2. In hours watched insert 100 3. In taxes section type 'y'. 4. Type 'Y' to return to the	1. No tax is applied, and the final price is broken down: 15$ fixed expense 10 x 2= $20	$243 is the amount in the total pay section, and expected data is received as well.		

			main menu and access option 2.	40 x 2 * 1,10 = $88 50 x 2 * 1,20 = $120 Without taxes: $243			
34	TC_34	Scenario of additional charge of 20% and tax included: 100 Hours Watched	1. Insert valid information. 2. In hours watched insert 100 3. In the taxes section type ‘n’. 4. Type ‘Y’ to return to the main menu and access option 2.	1. Tax is applied, and the final price is broken down: 15$ fixed expense 10 x 2= $20 40 x 2 * 1,10 = $88 50 x 2 * 1,20 = $120 With taxes: $274,59	$274,59 is the amount in the total pay section, and expected data is received as well.		
35	TC_35	Scenario of additional charge of 30% and no tax included: 720 Hours Watched	1. Insert valid information. 2. In hours watched insert 720 3. In taxes section type ‘y’. 4. Type ‘Y’ to return to the main menu and access option 2.	1. No tax is applied, and the final price is broken down: 15$ fixed expense 10 x 2= $20 40 x 2 * 1,10 = $88 50 x 2 * 1,20 = $120 620 x 2 * 1,30 = $1612 With no taxes: $1855	$1855 is the amount in the total pay section, and expected data is received as well.		
36	TC_36	Scenario of additional charge of 30% and tax included: 720 Hours Watched	1. Insert valid information. 2. In hours watched insert 720 3. In the taxes section type ‘n’. 4. Type ‘Y’ to return to the main menu and access option 2.	1. Tax is applied, and the final price is broken down: 15$ fixed expense 10 x 2= $20 40 x 2 * 1,10 = $88 50 x 2 * 1,20 = $120 620 x 2 * 1,30 = $1612 with taxes: $2096,15	$2096,15 is the amount in the total pay section, and the expected data is received as well.		

White Box

Most of the test cases I have used are a mix of white box and black box, by making sure that all the paths are executed, looking more for the internal structure of the code and the different decisions it has.

For example, Value Boundary Analysis is a great example of a black box covering white box, but from a user perspective.
I focused decision testing mainly on the calculation of the hours, since it is the only one having a specific decision-taking, for other features include a loop or a decision also.

Here's an example on how it should be tested:
Making sure to go through each of the decisions, based on the conditions in the code, and this specific scenario can be confirmed by TC: Module: Whole program E2E.

This TC covers some parts of the Value Boundary Analysis and goes through each of the categories seen in the diagram below, coming from the data entry first(hours), also if it is exempt from IVA, and then calculation which is displayed and visible on option 2 of the main menu.

This type of test is useful to understand how the application works and create strong test cases, however, black box also tackles this from a different perspective.