

Test Cases

Black-Box Test Cases

These test cases are defined based on the available documentation and execution of the program. The code was not inspected.

#	Test case (very brief description)	Preconditions (any required setup)	Test steps (steps executed during testing)	Expectation	Observation ("pass" or failure description)
1	Login for the existing trainer	Trainer should have an account	1)Launch app 2) Enter trainer username, password and press enter	Welcome menu should be visible and 2 options to select	pass
2	Login for the new trainer	Trainer doesn't have an account	1) Launch App Enter new trainer name and password to create an account and then login.	Welcome menu should be visible and 2 options to select.	pass
3	Validate Customer training plan	Trainer should have some customers assigned to him	After logging in, we need to select option-1	The customers list will be displayed.	pass
4	Add customer under the trainer	Customer details should be present.	After logging in, we need to select option-2	New customer will be added under the respective trainer.	pass
5	Efficiency of the customer	Customer details should be in the database/list.	1) After logging in, we need to select option-1 Enter the respective customer's tagId.	Customer's efficiency will be displayed.	pass
6	Further analysis of the customer	Customer details should be in the database/list.	1) After logging in, we need to select option-1 2) Enter the respective customer's tagId. 1. Press 1 or others to go back	Daily targeted activities and fitbit tag status of the customer will be displayed.	pass

7	Modification of the current training plan	Customer details should be in the database/list.	1) After logging in, we need to select option-1 2) Enter the respective customer's tagId. 3) Press 1 or others to go back 4) Press 1 or others to go back 1.	New training plan for selection will be displayed.	pass
8	Selection of the training plan	Customer details should be in the database/list.	1) After logging in, we need to select option-1 2) Enter the respective customer's tagId. 3) Press 1 or others to go back 4) Press 1 or others to go back 5) Enter any one from below: WL/WG (Weightloss/Weighgain) 1.	Modified training plan will be displayed	pass
9	Selection of new training activities under the modified plan	Customer details should be in the database/list.	1) After logging in, we need to select option-1 2) Enter the respective customer's tagId. 3) Press 1 or others to go back 4) Press 1 or others to go back 5) Enter any one from below: WL/WG (Weightloss/Weighgain) 6) Enter the new activities according to the modified plan.	The current/modified training plan and activities are displayed.	Pass
10	Verify the updated training program after re-login.	1. Account should exist in the database.	1. Launch the application 2. Enter any valid username and password that matches the existing users.	The updated Training Plan should reflect in the displayed message.	Fail (previous training program is still getting displayed in the message.

		A trainee should be assigned to the trainer.	3. Type 1 and press ENTER. 4. Enter the Tag Id shown in the recent message and press ENTER. 5. Type 1 and press ENTER 6. Type 1 and press ENTER 7. Provide necessary values for the next 2 questions and press ENTER after each input. 8. Login as the same user 7) Type 1 and press ENTER and verify the client details which we updated recently.		Because the update is happening only in individualDataFile.csv but not in dataFile.csv.
--	--	--	---	--	---

White-Box Test Cases

These additional test cases were defined during inspection of the code.

#	Test case (very brief description)	Preconditions (any required setup)	Test steps (steps executed during testing)	Expectation	Observation ("pass" or failure description)
1	New user login validation.	NA	1. Launch the application 2. Enter any random username and password which doesn't exist in the database and press ENTER.	Account doesn't exist message should be shown, and user should be asked to create a new account.	Pass
2	Existing user login validation.	An account by providing username and password should already be created	9. Launch the application 10. Enter any valid username and password that matches the existing users.	Welcome message along with the options should be shown.	Pass
3	Create duplicate account	Account should exist in the database.	1. Launch the application	Error message indicating account	Fail (Application crashes)

			<ol style="list-style-type: none"> 2. Enter any valid username and password that matches the existing users. 3. Enter same username and password. 	duplication should be shown.	
4	Evaluate input functionality of user options with non-integer values	Account should exist in the database.	<ol style="list-style-type: none"> 2. Launch the application 3. Enter any valid username and password that matches the existing users. 4. Type any String or float value and press ENTER 	Application should not crash, and user should be informed about valid input type.	Fail (Application crashes)
5	Evaluate input functionality of Target hours with non-integer values	Account should exist in the database.	<ol style="list-style-type: none"> 1. Launch the application 2. Enter any valid username and password that matches the existing users. 3. Type 2 and press ENTER 4. Type WL or WG and press ENTER 5. Type any String or float value and press ENTER 	Application should not crash, and user should be informed about valid input type.	Fail (Application crashes)
6	Evaluate check efficiency of the customer functionality with non-integer values	<ol style="list-style-type: none"> 1. Account should exist in the database. 2. A trainee should be assigned to the trainer. 	<ol style="list-style-type: none"> 1. Launch the application 2. Enter any valid username and password that matches the existing users. 3. Type 1 and press ENTER. 4. Type any String or float value and press ENTER. 	Application should not crash, and user should be informed about valid input type.	Fail (Application crashes)
7	Evaluate check efficiency of the customer functionality with non-matching integer value	<ol style="list-style-type: none"> 1. Account should exist in the database. 2. A trainee should be assigned to the trainer. 	<ol style="list-style-type: none"> 1. Launch the application 2. Enter any valid username and password that matches the existing users. 3. Type 1 and press ENTER. 	User should be informed about the value mismatch.	Fail (Application crashes)

			4. Type any integer value (non-float) other than the Tag Id value shown in the recent message.		
8	Default values not handled during user input after login	1. Account should exist in the database.	1. Launch the application 2. Enter any valid username and password that matches the existing users. 3. Type any integer value (non-float) other than 1, 2.	Default case should be handled, and appropriate message should be displayed.	Fail (Application exits without any message)
9	Verify the multiple updates made to the training program for the same user.	1. Account should exist in the database. 2. A trainee should be assigned to the trainer.	1. Execute steps in Test case#1. 2. Execute steps in Test case#1 again but change the values of training program and activities to different values. 3. Verify the dataFile.csv.	The most recently updated Training Plan should reflect in the dataFile.csv file.	Fail (Only the first update to the same client is getting updated in dataFile.csv).