Process MeNtOR 3.0

Uni-SEP

Crypto Trading Platform
Implementation Document

Version:	1.0
Print Date:	April 4, 2022
Release Date:	April 4, 2022
Release State:	Final
Approval State:	Approved
Approved by:	
Prepared by:	Roberto Armena, Karen Zhan, Zoe Desouza, Nicole Moshkovitch
Reviewed by:	
Path Name:	
File Name:	
Document No:	

1.1 Test Cases

Test ID	TC-1
Category	Evaluation of user credentials stored on file
Requirements Coverage	UC1-User-Login-Success
Initial Condition	System has been initiated and run. Database containing user and password accessed. MainUI has provided an interface to enter username and password.
Procedure	The list of steps required for this test case 1. The user opens the program 2. User enters username in username entry box 3. User enters password in password entry box 4. User enters presses submit button 5. System logs in successfully
Expected Outcome	Login window closes, trading options shown to User.

Test ID	TC-2
Category	Evaluation of user credentials stored on file
Requirements Coverage	UC1-User-Login-Failure
Initial Condition	System has been initiated and run. Database containing user and password accessed. MainUI has provided an interface to enter username and password.
Procedure	The list of steps required for this test case 1. The user selects login 2. User enters username in username entry box 3. User enters password in password entry box 4. User enters presses submit button 5. Login information incorrect, incorrect filled in (i.e. username left blank), or non-existent in database
Expected Outcome	Notify User that log-in failed, system terminates.

Test ID	TC-3
Category	Evaluation of add trade client function on main UI
Requirements Coverage	UC2-Add-Client-Success
Initial Condition	System has been initiated and run. Account has been validated with passcode (UC1) and the Main UI successfully launches.
Procedure	The list of steps required for this test case 1. User clicks ‘Add Row’ on the Main UI. 2. User enters the client name, coin list, and trading strategy in the new row’s boxes.
Expected Outcome	Main UI is able to call UserSelection Class to call Broker Class to create a new trade object with the information entered by the user in the new row added.
Notes	User cannot add the name of a trade client that already exists on another trade client. Coin list must include only valid cryptocurrency acronyms.

Test ID	TC-4
Category	Evaluation of add trade client function on main UI
Requirements Coverage	UC2-Add-Client-Failure
Initial Condition	System has been initiated and run. Account has been validated with passcode (UC1) and the Main UI successfully launches.
Procedure	The list of steps required for this test case 1. User clicks ‘Add Row’ on the Main UI. 2. User enters the client name in a new row that matches a previous row. 3. Error message window appears with error message, user is prompted to enter another name
Expected Outcome	Notify User that Add Row failed, clear row from table on Main UI and output message to enter a valid name.
Notes	May run into issues identifying when the User is done typing a name.

Test ID	TC-5
Category	Evaluation of delete row function on Main UI
Requirements Coverage	UC2-Delete-Row-Success
Initial Condition	System has been initiated and run. Account has been validated with passcode (UC1) and the Main UI successfully launches. At least one row exists on the table.
Procedure	The list of steps required for this test case

	1. The User clicks on the row to delete, then selects ‘Delete Row’. 2. MainUI calls UserSelection Class. 3. UserSelection Class object deletes the Broker object from the list of Trade objects it stores. 4. Main UI deletes the row and its contents from the table.
Expected Outcome	Selected row disappears from the Main UI table, and the UserSelection object removes the Broker from its list.
Notes	Uses the name of the trade client to locate the client in the list in the UserSelection class.

Test ID	TC-6
Category	Evaluation of delete function on Main UI
Requirements Coverage	UC2-Delete-Row-Failure
Initial Condition	System has been initiated and run. Account has been validated with passcode (UC1) and the Main UI successfully launches.
Procedure	The list of steps required for this test case 1. User selects ‘Delete Row’ on the Main UI without clicking on a row to delete. 2. Main UI sends an empty call to UserSelection Class. UserSelection Class does not remove anyBroker objects because the information was not selected. 3. Nothing is deleted on Main UI.
Expected Outcome	No row is deleted on Main UI.

Test ID	TC-7
Category	Evaluation of Perform Trade function on Main UI
Requirements Coverage	UC2-Perform-Trade-Failure
Initial Condition	System has been initiated and run. Account has been validated with passcode (UC1) and Main UI successfully launches. No client trade rows exist on the table.
Procedure	The list of steps required for this test case 1. User clicks on ‘Perform Trade’. 2. Main UI displays error message for User to add Rows.
Expected Outcome	Notify User that Perform Trade failed, remain on Main UI interface.
Notes	Trade would go on successfully as long as table is non-empty.

Test ID	TC-8
Category	Evaluation of Crypto Prices Fetched from CoinGecko
Requirements Coverage	UC3-Get-Coin-Values-Success
Initial Condition	System has been initiated and run. Account has been validated with passcode (UC1) and Main UI successfully launches. At least one client trade row exists on the table (UC2).
Procedure	The list of steps required for this test case 1. User clicks on ‘Perform Trade’ 2. Main UI stores all current brokers in UserSelection object 3. ExecuteTrade is enacted on all brokers, calling TradeStrategy on each broker object 4. TradeStrategy attempts to fetch each coin price from CoinGecko API 5. Each coin price is distributed back to the respective broker(s) 6. Each broker’s respective trading strategies is applied
Expected Outcome	Add trading action (buy/sell) to list storing cumulative trading history and outputted to table ad histogram

Test ID	TC-9
Category	Evaluation of Crypto Prices Fetched from CoinGecko
Requirements Coverage	UC3-Get-Coin-Values-Failure
Initial Condition	System has been initiated and run. Account has been validated with passcode (UC1) and Main UI successfully launches. At least one client trade row exists on the table (UC2).
Procedure	The list of steps required for this test case 1. User clicks on ‘Perform Trade’ 2. Main UI stores all current brokers in UserSelection object 3. ExecuteTrade is enacted on all brokers, calling TradeStrategy on each broker object 4. TradeStrategy attempts to fetch each coin price from CoinGecko API 5. TradeStrategy sends a coin not in CoinGecko’s database
Expected Outcome	Display error message indicating the coin is invalid and trade is terminated
Notes	Will continue displaying error message until all user coin values are valid (i.e. in coinGecko’s database)

1.2 Group Meeting Logs

Present Group Members	**Meeting Date**	**Issues Discussed / Resolved**
All	March 22, 2022	Finalized classes and relevant design patterns to implement and delegated coding tasks
All	March 25, 2022	Finalized and implemented trading strategies and connected program to coinGecko API
All	April 1, 2022	Finalized and implemented code to render trading actions onto the table and histogram
All	April 3, 2022	Ran through all test cases and debugged program, ensured all relevant design patterns have been implemented