**NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES**



**SOFTWARE QUALITY ENGINEERING**

**FAST UNIVERSITY FOOD ORDERING APPLICATION**

|  |  |
| --- | --- |
| **Team Members** | 1. Faaiz Kadiwal (21K-3830) 2. Ammar Asdaque (21K-3923) 3. Anser Tayyab (21K-3909) |
| **Course Instructor** | Ms. Syeda Rubab Jaffar |
| **Submission Date** | 03-December-2023 |

| **Test Case Name** | **Requirement** | **Preconditions** | **Steps** | **Expected Results** |
| --- | --- | --- | --- | --- |
| TC\_Login\_001 | User login functionality | None | 1. Open the Login window. | 1. Login window is displayed. |
|  |  |  | 2. Enter a valid username and password. | 2. Username and password are entered successfully. |
|  |  |  | 3. Click on the "LOGIN" button. | 3. The user is successfully logged in. |
| TC\_Login\_002 | Incorrect username format | None | 1. Open the Login window. | 1. Login window is displayed. |
|  |  |  | 2. Enter an invalid username format. | 2. "Incorrect Format for Username" message is shown. |
|  |  |  | 3. Click on the "LOGIN" button. | 3. The login attempt fails with an error message. |
| TC\_Login\_003 | Incorrect password format | None | 1. Open the Login window. | 1. Login window is displayed. |
|  |  |  | 2. Enter an invalid password format. | 2. "Password must be or under 8 characters" message is shown. |
|  |  |  | 3. Click on the "LOGIN" button. | 3. The login attempt fails with an error message. |
| TC\_Login\_004 | Successful login - User RD | Database | 1. Open the Login window. | 1. Login window is displayed. |
|  |  | Entry in DB | 2. Enter the correct username and password for RD. | 2. User is successfully logged in as RD. |
|  |  |  | 3. Click on the "LOGIN" button. | 3. The user is redirected to the Canteen window for RD. |
| TC\_Login\_005 | Successful login - User BS | Database | 1. Open the Login window. | 1. Login window is displayed. |
|  |  | Entry in DB | 2. Enter the correct username and password for BS. | 2. User is successfully logged in as BS. |
|  |  |  | 3. Click on the "LOGIN" button. | 3. The user is redirected to the Canteen window for BS. |
| TC\_Login\_006 | Successful login - User CF | Database | 1. Open the Login window. | 1. Login window is displayed. |
|  |  | Entry in DB | 2. Enter the correct username and password for CF. | 2. User is successfully logged in as CF. |
|  |  |  | 3. Click on the "LOGIN" button. | 3. The user is redirected to the Canteen window for CF. |
| TC\_Login\_007 | Successful login - User FF | Database | 1. Open the Login window. | 1. Login window is displayed. |
|  |  | Entry in DB | 2. Enter the correct username and password for FF. | 2. User is successfully logged in as FF. |
|  |  |  | 3. Click on the "LOGIN" button. | 3. The user is redirected to the Canteen window for FF. |

| **est Case Name** | **Requirement** | **Preconditions** | **Steps** | **Expected Results** |
| --- | --- | --- | --- | --- |
| Valid Input | User should be able to create an account with valid input. | - None | 1. Enter a valid username (**21K-1234**).<br>2. Enter a valid password (**password**).<br>3. Confirm the password (**password**). | The account should be created successfully. |
| Invalid Username Format | Username should follow the specified format (**[0-9][0-9]K-[0-9][0-9][0-9][0-9]**). | - None | 1. Enter an invalid username (**abc123**).<br>2. Enter a valid password (**password**).<br>3. Confirm the password (**password**). | An error message should be displayed indicating an incorrect format for the username. |
| Invalid Password Length | Password should be 8 characters or under. | - None | 1. Enter a valid username (**21K-5678**).<br>2. Enter a short password (**pass**).<br>3. Confirm the password (**pass**). | An error message should be displayed indicating that the password must be 8 characters or under. |
| Mismatched Passwords | Password and Confirm Password should match. | - None | 1. Enter a valid username (**21K-9101**).<br>2. Enter a valid password (**pass123**).<br>3. Confirm a different password (**pass456**). | An error message should be displayed indicating that passwords must match. |
| Username Already Exists | User should not be able to create an account if the username already exists. | Username already exists in the database | 1. Enter an existing username.<br>2. Enter a valid password (**password**).<br>3. Confirm the password (**password**). | An error message should be displayed indicating that the account already exists. |
| Undo Button | User should be able to undo the signup action. | - None | 1. Perform some actions in the SignUp window.<br>2. Press the Undo button. | The SignUp window should be closed, and the Login window should be displayed. |

| **Test Case Name** | **Requirement** | **Preconditions** | **Steps** | **Expected Results** |
| --- | --- | --- | --- | --- |
| Valid Input | User should be able to navigate to the login page when "NEXT" is pressed. | - None | 1. Open the MainMenu window.<br>2. Press the "NEXT" button. | The MainMenu window should be closed, and the Login window should be displayed. |
| Undo Button | User should be able to undo the navigation action. | - None | 1. Open the MainMenu window.<br>2. Perform some actions in the MainMenu window.<br>3. Press the "Cancel" button. | The MainMenu window should be closed without navigating to the Login window. |

| **Test Case Name** | **Requirement** | **Preconditions** | **Steps** | **Expected Results** |
| --- | --- | --- | --- | --- |
| Back to SignIn | User should be able to navigate back to the SignIn page when the "Back to SignIn" button is pressed. | - None | 1. Open the Canteen window.<br>2. Press the "Back to SignIn" button. | The Canteen window should be closed, and the SignIn window should be displayed. |
| Confirm Order Delivery | User should be able to confirm the order delivery and navigate to the View page. | - None | 1. Open the Canteen window.<br>2. Press the "Confirm Order Delivery" button. | The Canteen window should be closed, and the View window with the specified ID should be displayed. |
| Accept Order | User should be able to accept the order and navigate to the Confirmation page. | - None | 1. Open the Canteen window.<br>2. Press the "Accept Order" button. | The Canteen window should be closed, and the Confirmation window with the specified ID should be displayed. |
| View Sales | User should be able to view sales and navigate to the Sales page. | - None | 1. Open the Canteen window.<br>2. Press the "View Sales" button. | The Canteen window should be closed, and the Sales window with the specified ID should be displayed. |
| Delete Item | User should be able to delete an item and navigate to the Item page. | - None | 1. Open the Canteen window.<br>2. Press the "Delete Item" button. | The Canteen window should be closed, and the Item window with the specified ID and operation type should be displayed. |
| Modify Item Price | User should be able to modify an item's price and navigate to the Item page. | - None | 1. Open the Canteen window.<br>2. Press the "Modify Item Price" button. | The Canteen window should be closed, and the Item window with the specified ID and operation type should be displayed. |
| Add Items to the Menu | User should be able to add items to the menu and navigate to the Item page. | - None | 1. Open the Canteen window.<br>2. Press the "Add Items to the Menu" button. | The Canteen window should be closed, and the Item window with the specified ID and operation type should be displayed. |
| Cancel | User should be able to cancel and close the Canteen window. | - None | 1. Open the Canteen window.<br>2. Press the "Cancel" button. | The Canteen window should be closed without navigating to any other window. |

| **Test Case Name** | **Requirement** | **Preconditions** | **Steps** | **Expected Results** |
| --- | --- | --- | --- | --- |
| Verify Icon Selection and Navigation | The system should display icons representing different options. Clicking on each icon should navigate to the corresponding screen. | The Main frame is open. | 1. Move the mouse over each icon and check if the tooltip text is displayed correctly. 2. Click on each icon and verify the corresponding action. | All icons are displayed correctly, and clicking on each icon leads to the expected navigation. |
| Verify Receipts Icon | Clicking on the "Show Receipts" icon should navigate to the Receipts screen. | The Main frame is open. | 1. Click on the "Show Receipts" icon. | The application should navigate to the Receipts screen. |
| Verify Logout Icon | Clicking on the "Logout" icon should navigate to the login screen. | The Main frame is open. | 1. Click on the "Logout" icon. | The application should navigate to the login screen. |
| Verify Close Icon | Clicking on the close (cancel) icon should close the application. | The Main frame is open. | 1. Click on the close (cancel) icon. | The application should close. |
| Verify Tooltip Text | Tooltip text should be displayed correctly when hovering over the icons. | The Main frame is open. | 1. Move the mouse over each icon and check if the tooltip text is displayed correctly. | The tooltip text should be displayed for each icon with the correct information. |

| **Test Case Name** | **Requirement** | **Preconditions** | **Steps** | **Expected Results** |
| --- | --- | --- | --- | --- |
| Verify Item Addition to Cart | Users should be able to add items to the cart. | The Ordering frame is open. | 1. Enter a valid item name in the "Enter Item Name" field. 2. Enter a valid quantity in the "Enter Quantity" field. 3. Click on the "Add to Cart" icon. | The selected item should be added to the cart, and the cart display and total price should be updated accordingly. |
| Verify Item Addition to Cart - Invalid Item | Users should not be able to add invalid items to the cart. | The Ordering frame is open. | 1. Enter an invalid item name in the "Enter Item Name" field. 2. Enter a valid quantity in the "Enter Quantity" field. 3. Click on the "Add to Cart" icon. | An error message should be displayed, indicating that the item is not valid and prompting the user to enter a valid item name. |
| Verify Item Addition to Cart - Invalid Quantity | Users should not be able to add items with an invalid quantity to the cart. | The Ordering frame is open. | 1. Enter a valid item name in the "Enter Item Name" field. 2. Enter an invalid quantity in the "Enter Quantity" field. 3. Click on the "Add to Cart" icon. | An error message should be displayed, indicating that the quantity is not valid and prompting the user to enter a valid quantity. |
| Verify Cart Clearing | Users should be able to clear the items in the cart. | The Ordering frame is open, and items are added to the cart. | 1. Click on the "Clear Cart" icon. | The cart display and total price should be cleared. |
| Verify Order Placement | Users should be able to place an order successfully. | The Ordering frame is open, and items are added to the cart. | 1. Click on the "Place Order" icon. | The order should be placed successfully, and the items should be added to the database. |
| Verify Order Placement - Empty Cart | Users should not be able to place an order with an empty cart. | The Ordering frame is open, and the cart is empty. | 1. Click on the "Place Order" icon. | An error message should be displayed, indicating that the cart is empty, and the order cannot be placed. |
| Verify Item Addition to Cart - Quantity Limit | Users should not be able to order more than 10 items of the same type at a time. | The Ordering frame is open. | 1. Enter a valid item name in the "Enter Item Name" field. 2. Enter a quantity greater than 10 in the "Enter Quantity" field. 3. Click on the "Add to Cart" icon. | An error message should be displayed, indicating that the quantity limit is exceeded, and the user should enter a valid quantity. |
| Verify Item Addition to Cart - Database Connectivity | The system should be able to fetch item details from the database when adding items to the cart. | The Ordering frame is open. | 1. Enter a valid item name in the "Enter Item Name" field. 2. Enter a valid quantity in the "Enter Quantity" field. 3. Click on the "Add to Cart" icon. | The selected item details should be fetched from the database, and the cart display and total price should be updated accordingly. |
| Verify Item Deletion from Cart | Users should be able to delete items from the cart. | The Ordering frame is open, and items are added to the cart. | 1. Select an item in the cart display. 2. Click on the "Delete Item" icon. | The selected item should be removed from the cart, and the cart display and total price should be updated accordingly. |

| **Test Case Name** | **Requirement** | **Preconditions** | **Steps** | **Expected Results** |
| --- | --- | --- | --- | --- |
| **View Orders** | The system should display a list of pending orders in the **Confirmation** window. | The **Confirmation** window is open. | 1. Run the application. <br> 2. Open the **Confirmation** window. | The list of pending orders is displayed correctly, including Order ID, Item ID, Item, Quantity, and Bill. |
| **Accept Order** | The system should confirm the specified order with the entered End Time. | The **Confirmation** window is open. | 1. Enter a valid Order ID. <br> 2. Enter a valid End Time. <br> 3. Click the "CONFIRM" button. | The order is confirmed, and the list is updated with the specified End Time. |
| **Accept Order with Invalid Order ID** | The system should display an error message for an invalid Order ID during the confirmation process. | The **Confirmation** window is open. | 1. Enter an invalid Order ID (e.g., non-numeric). <br> 2. Enter a valid End Time. <br> 3. Click the "CONFIRM" button. | An error message is displayed, indicating the need for a valid Order ID. |
| **Accept Order with Invalid End Time** | The system should display an error message for an invalid End Time during the confirmation process. | The **Confirmation** window is open. | 1. Enter a valid Order ID. <br> 2. Enter an invalid End Time (e.g., invalid format). <br> 3. Click the "CONFIRM" button. | An error message is displayed, indicating the correct format for the End Time. |
| **Delete Order** | The system should delete the specified order from pending orders. | The **Confirmation** window is open. | 1. Enter a valid Order ID. <br> 2. Click the "DELETE" button. | The order is deleted from the list of pending orders. |
| **Delete Order with Invalid Order ID** | The system should display an error message for an invalid Order ID during the deletion process. | The **Confirmation** window is open. | 1. Enter an invalid Order ID (e.g., non-numeric). <br> 2. Click the "DELETE" button. | An error message is displayed, indicating the need for a valid Order ID. |

| **Test Case Name** | **Requirement** | **Preconditions** | **Steps** | **Expected Results** |
| --- | --- | --- | --- | --- |
| **Enqueue from DB** | The system should add orders from the database to the queue. | Database contains orders. | 1. Create a **Queue** object. <br> 2. Call the **EnqueueFromDB** method with a valid roll number. | The queue is populated with orders from the database for the specified roll number. |
| **Display Queue** | The system should display the orders in the queue. | The queue is not empty. | 1. Create a **Queue** object. <br> 2. Add orders to the queue. <br> 3. Call the **display** method. | The orders in the queue are correctly displayed, including Order ID, Item, Quantity, and Bill. |
| **Get Money Spent** | The system should calculate the total money spent on the orders in the queue. | The queue is not empty. | 1. Create a **Queue** object. <br> 2. Add orders to the queue. <br> 3. Call the **getMoneySpent** method. | The total money spent on the orders in the queue is correctly calculated. |
| **Insert at Head (Linked List)** | The system should insert a new node at the head of the linked list. | Linked list is initialized. | 1. Create a **LL** object. <br> 2. Call the **insertAtHead** method with valid arguments. | The new node is inserted at the head of the linked list with the specified data. |
| **Display Linked List** | The system should display the contents of the linked list. | The linked list is not empty. | 1. Create a **LL** object. <br> 2. Insert nodes into the linked list. <br> 3. Call the **display** method. | The contents of the linked list are correctly displayed, including Item, Quantity, and Bill. |
| **Insert Into DB (Linked List)** | The system should insert the head node of the linked list into the database. | The linked list is not empty. | 1. Create a **LL** object. <br> 2. Insert nodes into the linked list. <br> 3. Call the **insertIntoDB** method. | The head node of the linked list is successfully inserted into the database. |
| **Get Total Price (Linked List)** | The system should calculate the total price of items in the linked list. | The linked list is not empty. | 1. Create a **LL** object. <br> 2. Insert nodes into the linked list. <br> 3. Call the **getTotalPrice** method. | The total price of items in the linked list is correctly calculated. |
| **Clear Cart (Linked List)** | The system should clear all nodes from the linked list. | The linked list is not empty. | 1. Create a **LL** object. <br> 2. Insert nodes into the linked list. <br> 3. Call the **clearCart** method. | The linked list becomes empty after clearing all nodes. |
| **Insert from DB (MinHeap)** | The system should add orders from the database to the min heap. | Database contains orders. | 1. Create a **MinHeap** object. <br> 2. Call the **insertFromDB** method with a valid container name. | The min heap is populated with orders from the database for the specified container. |
| **Display Min Heap** | The system should display the orders in the min heap. | The min heap is not empty. | 1. Create a **MinHeap** object. <br> 2. Insert orders into the min heap. <br> 3. Call the **display** method. | The orders in the min heap are correctly displayed, including Order ID, Item, Quantity, Bill, and End Time. |
| **Extract from Order ID (MinHeap)** | The system should extract an order from the min heap based on Order ID. | The min heap is not empty. | 1. Create a **MinHeap** object. <br> 2. Insert orders into the min heap. <br> 3. Call the **extractFromOrderID** method with a valid Order ID. | The order with the specified Order ID is successfully extracted from the min heap. |

| **Test Case Name** | **Requirement** | **Preconditions** | **Steps** | **Expected Results** |
| --- | --- | --- | --- | --- |
| **View Menu** | The system should display the menu items and their prices for a specific canteen. | Database contains menu items and prices. | 1. Create an **Item** object. <br> 2. Call the **viewMenu** method with a valid canteen code. | The menu items and their prices are displayed in the text area. |
| **Add Item (Add Mode)** | The system should add a new item to the menu. | Database contains the specified canteen. | 1. Create an **Item** object in add mode. <br> 2. Enter a valid item name and price. <br> 3. Click the "Add Item" button. | The new item is added to the menu, and the menu is updated. |
| **Add Item (Item Already Exists)** | The system should display an error if the item to be added already exists. | Database contains the specified canteen with existing items. | 1. Create an **Item** object in add mode. <br> 2. Enter the name of an existing item and a valid price. <br> 3. Click the "Add Item" button. | An error message is displayed, indicating that the item already exists. |
| **Delete Item (Delete Mode)** | The system should delete an item from the menu. | Database contains the specified canteen with existing items. | 1. Create an **Item** object in delete mode. <br> 2. Enter the item ID to be deleted. <br> 3. Click the "Add Item" button. | The specified item is deleted from the menu, and the menu is updated. |
| **Delete Item (Item Not Found)** | The system should display an error if the item to be deleted is not found. | Database contains the specified canteen with existing items. | 1. Create an **Item** object in delete mode. <br> 2. Enter a non-existent item ID. <br> 3. Click the "Add Item" button. | An error message is displayed, indicating that the item is not found. |
| **Modify Item (Modify Mode)** | The system should modify the price of an existing item. | Database contains the specified canteen with existing items. | 1. Create an **Item** object in modify mode. <br> 2. Enter the item ID and a new valid price. <br> 3. Click the "Add Item" button. | The price of the specified item is updated, and the menu is updated. |
| **Modify Item (Item Not Found)** | The system should display an error if the item to be modified is not found. | Database contains the specified canteen with existing items. | 1. Create an **Item** object in modify mode. <br> 2. Enter a non-existent item ID. <br> 3. Click the "Add Item" button. | An error message is displayed, indicating that the item is not found. |
| **Modify Item (Invalid Price)** | The system should display an error if an invalid price is entered during modification. | Database contains the specified canteen with existing items. | 1. Create an **Item** object in modify mode. <br> 2. Enter the item ID and an invalid price (non-numeric). <br> 3. Click the "Add Item" button. | An error message is displayed, indicating that an invalid price is entered. |

| **Test Case Name** | **Requirement** | **Preconditions** | **Steps** | **Expected Results** |
| --- | --- | --- | --- | --- |
| **View Orders** | The system should display the pending orders for the logged-in user. | Database contains order records for the logged-in user. | 1. Create an **Orders** object. | Pending orders for the logged-in user are displayed in the text area. |
| **Navigate Back** | The user should be able to navigate back to the main menu. | The **Orders** window is open. | 1. Move the mouse over the "Back" button. <br> 2. Click the "Back" button. | The **Orders** window is closed, and the main menu window is displayed. |

| **Test Case Name** | **Requirement** | **Preconditions** | **Steps** | **Expected Results** |
| --- | --- | --- | --- | --- |
| **View Orders - Constructor without Parameters** | The system should display a view of orders from the database. | Database contains order records. | 1. Create a **View** object without parameters. | The **View** window is displayed with a list of orders. |
| **View Orders - Constructor with Parameter** | The system should display a view of orders for a specific category. | Database contains order records for the specified category. | 1. Create a **View** object with a category parameter. | The **View** window is displayed with a list of orders for the specified category. |
| **Navigate Back** | The user should be able to navigate back to the main menu. | The **View** window is open. | 1. Move the mouse over the "Back" button. <br> 2. Click the "Back" button. | The **View** window is closed, and the main menu window is displayed. |
| **Confirm Delivery - Minimum Time Remaining** | The system should confirm the delivery with the minimum time remaining. | There are orders in the system. | 1. Move the mouse over the "Confirm Delivery by Minimum Time Remaining" button. <br> 2. Click the "Confirm Delivery by Minimum Time Remaining" button. | The order with the minimum time remaining is confirmed, and the list is updated. |
| **Confirm Delivery - By Order ID** | The system should confirm the delivery based on the entered Order ID. | There are orders in the system. | 1. Move the mouse over the "Confirm Delivery by Order ID" button. <br> 2. Enter a valid Order ID. <br> 3. Click the "Confirm Delivery by Order ID" button. | The order with the specified Order ID is confirmed, and the list is updated. |
| **Invalid Order ID Entry** | The system should display an error message for an invalid Order ID entry. | There are orders in the system. | 1. Move the mouse over the "Confirm Delivery by Order ID" button. <br> 2. Enter an invalid Order ID (length > 4 or non-numeric). <br> 3. Click the "Confirm Delivery by Order ID" button. | An error message is displayed, and the order list remains unchanged. |

| **Test Case Name** | **Requirement** | **Preconditions** | **Steps** | **Expected Results** |
| --- | --- | --- | --- | --- |
| **View Sales - Constructor without Parameters** | The system should display the sales information for a canteen. | Database contains sales records for the canteen. | 1. Create a **Sales** object without parameters. | The **Sales** window is displayed with the list of sales and total sales. |
| **View Sales - Constructor with Parameter** | The system should display the sales information for a specific category. | Database contains sales records for the specified category. | 1. Create a **Sales** object with a category parameter. | The **Sales** window is displayed with the list of sales and total sales for the specified category. |
| **Navigate Back** | The user should be able to navigate back to the main menu. | The **Sales** window is open. | 1. Move the mouse over the "Back" button. <br> 2. Click the "Back" button. | The **Sales** window is closed, and the main menu window is displayed. |
| **Clear Sales - Confirm** | The user should be able to clear sales records for delivered and confirmed orders. | There are delivered and confirmed orders in the system. | 1. Move the mouse over the "Clear Sales" button. <br> 2. Click the "Clear Sales" button. | The sales records for delivered and confirmed orders are cleared. The sales list and total sales are updated. |
| **Clear Sales - No Sales to Clear** | The user should be notified if there are no sales to clear. | There are no delivered and confirmed orders in the system. | 1. Move the mouse over the "Clear Sales" button. <br> 2. Click the "Clear Sales" button. | A message is displayed indicating that there are no sales to clear. The sales list and total sales remain unchanged. |

| **Test Case Name** | **Requirement** | **Preconditions** | **Steps** | **Expected Results** |
| --- | --- | --- | --- | --- |
| **View Receipts - Default Constructor** | The system should display the receipts for a user. | Database contains receipt records for the user. | 1. Create a **Receipts** object without parameters. | The **Receipts** window is displayed with the list of receipts and total money spent. |
| **Navigate Back** | The user should be able to navigate back to the main menu. | The **Receipts** window is open. | 1. Move the mouse over the "Back" button. <br> 2. Click the "Back" button. | The **Receipts** window is closed, and the main menu window is displayed. |