TEST SUITE

for

Supermarket Automation Software

Prepared by -

Kaushal Banthia (19CS10039)

Parth Tusham (19CS30034)

Shashwat Shukla (19CS10056)

March 25, 2021

Contents

Unit Test of the following features are conducted:

- **login()**

Input: (<userID>, <password>)
If the registered credentials are (userID: kaushal, password: abcd), then

Case I: Both user Id and password entered are incorrect
- Input: ("kau", "abc")
- Output: Error

Case II: Entered user Id is correct but password entered is incorrect
- Input: ("kaushal", "abc")
- Output: Error

Case III: User Id incorrect but password entered is correct
- Input: ("kau", "abcd")
- Output: Error

Case IV: Both user Id and password entered are correct
- Input: ("kaushal", "abcd")
- Output: Logs in

From the Manager's Window

- **editInventory()**

Error Message("Please enter a Valid/New Product details"): EM1
Error Message("Please Enter a Valid Quantity/Cost/Price"): EM2

1. **Adding a new item in the database:**

Input = (<Product Name>, <Product ID>, <Product Quantity>, <Product Cost>, <Product Price>)
Product details are written in input boxes.
Output is obtained after pressing Add Items Button.

Test Scenarios:

New Item with Product Id that does not exist and Quantity, Price and Cost are all positive:
- Input: ("1234", "Shampoo", "10", "20.0", "30.0")
- Output: Addition Successful New item has been added

New Item with Product Id that already exists:

- Input: (“1234”, “Shampoo”, “10”, “20.0”, “30.0”)
- Output: EM1

New Item with Product Name of an already existing product:

- Input: (“1234”, “Shampoo”, “10”, “20.0”, “30.0”)
- Output: EM1

New Item with quantity set as a nonpositive Integer:

- Input: (“1234”, “Shampoo”, “-5”, “20.0”, “30.0”)
- Output: EM1

New Item with Cost set as non-positive:

- Input: (“1234”, “Shampoo”, “10”, “-5”, “30.0”)
- Output: EM1

New Item with Price set as less than cost:

- Input: (“1234”, “Shampoo”, “10”, “20.0”, “10.0”)
- Output: EM1

If the any of the box are left empty:

- Input: (“”, “Shampoo”, “19”, “10.0”, “20.0”)
- Output: EM1

2. Editing an existing Item in the database:

Input =(<Product Quantity>, <Product Cost>, <Product Price>)

An item is selected from the list and is edited.
Output is obtained after pressing the Save Changes button.

Test Scenarios:

Entering the Quantity, Cost and Price as positive with Cost less than Price:

- Input: (“10”, “20.0”, “30.0”)
- Output: Addition Successful New item has been added.

Entering the Quantity as non-positive:

- Input: (“-1”, “20.0”, “30.0”)
- Output: EM2

Entering the cost as non-positive:

- Input: (“5”, “-3.0”, “30.0”)
- Output: EM2

Entering the Price as non-positive or less than cost:

- Input: (“5”, “15.0”, “10.0”)
- Output: EM2

3. **Removing an existing Item in the database:**

   Press the remove button and Click on save changes. When viewed in the Database the product will not appear in the inventory.

- **removeUser()**

  In the User Edit page in the Manager Window if the Remove button is pressed, followed by the Save Changes button, then the selected clerk is removed, which could be verified by seeing the list of clerks again.

- **reportStats()**

  ```
  Input = (<Start Date>, <End Date>)
  ```

  Case I: The start date is after the end date
  - `Input: (26-03-2021, 25-03-2021)`
  - `Output: Error`

  Case II: The start date is the same as the end date
  - `Input: (26-03-2021, 26-03-2021)`
  - `Output: Accepted (A single day)`

  Case III: The start date is before the end date
  - `Input: (26-03-2021, 27-03-2021)`
  - `Output: Accepted (2 days)`

- **printStats()**

  From Transaction database, it takes the input, which are all valid transactions.
  Shows the graph for various metrics on the basis of the timeframe obtained from **reportStats()**.

From the Clerk's Window

- **createItemTransaction()**

  ```
  Input = (<Product Name selected from drop-down menu>, <Product Quantity>)
  ```

  Case I: Quantity of a correctly chosen item is kept as zero and the "Add to Cart" button is pressed
  - `Input: ("Shampoo", 0)`
  - `Output: Please Enter a Valid Quantity.`

Case II: No item is chosen and the quantity is kept as zero and the "Add to Cart" button is pressed

- `Input: (None, 0)`
- `Output: Please select a Valid Product and Enter a Valid Quantity.`

Case III: No item is chosen but the quantity is kept as a positive integer and the "Add to Cart" button is pressed

- `Input: (None, 1)`
- `Output: Error`

Case IV: Quantity of a correctly chosen item is kept as a positive integer (greater than the quantity of that product available) and the "Add to Cart" button is pressed (Let the quantity of "Shampoo" available in cart be 5)

- `Input: ("Shampoo", 8)`
- `Output: Chosen item(s) not available in required quantity.`

Case V: Quantity of a correctly chosen item is kept as a positive integer (lesser than or equal to the quantity of that product available) and the "Add to Cart" button is pressed

- `Input: ("Shampoo", 1)`
- `Output: Product(s) added to Cart.`

- **printBill()**

  When called it will show a bill only if the cart is non-empty. The following details will be printed in the bill:

  - Bill ID
  - Date
  - Item name /ID, quantity purchased.
  - Clerk User ID
  - Total Price

  It takes input from the already made **createItemTransaction()**, which has previously verified all the items so need to check this function.

Backend Functions

- **addTransaction()**

  A valid transaction is taken from a Transaction list. Full Cost is calculated using Price and Quantity, Profit is calculated using `(Price - Cost) * Quantity.`

  `Input = (<List of valid Item Transactions>)`

  Wrong transactions can never be entered in this case because the input is taken from **createItemTransaction()** and **printBill()** which have already been validated.

- **register()**

  `Input = (<Username>, <Email ID>, <Password>)`

  NOTE: The scenarios given below are for the case, when the user has filled all the asked details. If not, an error message - “Please fill the required details” will be shown.

  Case I: If the entered username does not exist in the database
  - `Input: (Chomksy,Chomksy@gmail.com , FlatisLife123)`
  - `Output: User Created Successfully.`

  Case II: If the entered username already exists in the database
  - `Input: (Chomksy,Chomksy@gmail.com , FlatisLife123)`
  - `Output: The username already exists.`

- **createUser()**

  A valid user is created from the **register()**. The **register()** function is already validated and hence, there is no need to test this function.