

**Test Case Document**  
for  
**Supermarket Automation Software (SAS)**

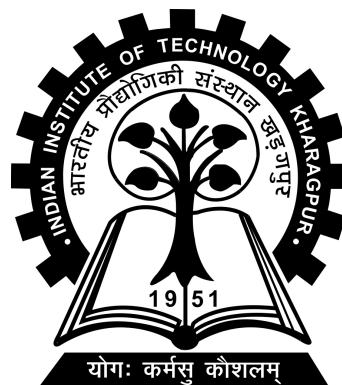
by

**Group-18**

Devendra Palod (*20CS10024*)

Subhajyoti Halder (*20CS10064*)

Deepiha S. (*20CS30015*)



Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

Spring Semester, 2021-22

# TEST CASES OUTLINE

Unit Test of the following features are conducted.

## login()

*Input: (username, password )*

If the registered credentials are (userID:20246405, password:asdfg), then

Case I: Both user Id and password entered are incorrect.

- Input: (“20”, “abc”)
- Output: Error

Case II: Entered user Id is correct but the password entered is incorrect.

- Input: (“20246405”, “abc”)
- Output: Error

Case III: User Id incorrect but password entered is correct.

- Input: (“2024”, “asdfg”)
- Output: Error

Case IV: Both user Id and password entered are correct.

- Input: (“20246405”, “asdfg”)
- Output: Logs in

## Adding a new item in the database

*Input = (Product name, Product ID, Brand, Type, Price, Quantity)*

Product details are written in input boxes. Output is obtained after pressing Add product Button.

Test Scenarios:

New Item with Product Id that does not exist and Quantity, Price are all positive:

- Input: (“AA battery cells”, “12345”, “Everyday”, “Electronics”, 15.0, 20)
- Output: Upon successful addition user is directed back to employee profile page.

New Item with Product Id that already exists:

- Input: (“AA battery cells”, “12345”, “Everyday”, “Electronics”, 15.0, 20)
- Output: error message

New Item with Product Name of an already existing product:.

- Input: (“AA battery cells”, “12345”, “Everyday”, “Electronics”, 15.0, 20)
- Output: error message

New Item with quantity set as a non-positive Integer:

- Input: (“AA battery cells”, “12345”, “Everyday”, “Electronics”, 15.0, -20)
- Output: error message

New Item with Price set as non-positive:

- Input: (“AA battery cells”, “12345”, “Everyday”, “Electronics”, -15.0, 20)

- Output: error message

If the any of the box are left empty:

- Input: (“AA battery cells”, “”, “Everyday”, “Electronics”, 15.0, 20)
- Output: error message

## Editing the quantity of an existing Item in the database:

*Input =(product id, new extra quantity)*

Item is accessed and output is obtained after pressing the Save Changes button.

Test Scenarios:

Entering the Product id, quantity:

- Input: (“12345”, 30)
- Output: Upon successful addition of extra stock, user is directed back to employee profile page.

Entering invalid product id (non-existent or negative):

- Input: (“123456”, 30) or (“-90678”, 30)
- Output: error message

Entering the Quantity as non-positive:

- Input: (“12345”, -30)
- Output: error message

## Editing the price of an existing Item in the database

*Input =(product id, new price)*

Item is accessed and the output is obtained after pressing the Save Changes button.

Test Scenarios:

Entering the Product id and valid price:

- Input: (“12345”, 30.00)
- Output: Upon successful updation of price, user is directed back to manager profile page.

Entering invalid product id (non-existent or negative):

- Input: (“123456”, 30.00) or (“-90678”, 30.00)
- Output: error message

Entering the price as non-positive:

- Input: (“12345”, -30.00)
- Output: error message

## reportSalesStats()

*Input = (start date, end date)*

Test Scenarios:

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().

## createItemTransaction()

*Input = (Product id, Quantity)*

Test Scenarios:

Case I: Quantity of a correctly chosen item is kept as zero and enter is pressed.

- Input: ("12345", 0)
- Output: Please Enter a Valid Quantity

Case II: No item is chosen and the quantity is kept as zero and enter 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 enter is pressed.

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

Case IV: Quantity of a correctly chosen item is kept as a positive integer and enter is pressed .

- Input: ("12345", 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.
- Sales Clerk Id
- Total Price

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

## addTransaction()

A valid transaction after generation of bill is added to the Transaction list.

*Input = (Bill Id, date, dictionary of Product ID, quantity, sales clerk id, Total price)*

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 = (Name, User name, Password)*

NOTE: The scenarios given below are for the case, when the user has filled all the asked details. If not, an error message will be shown and button to add user will not be activated.

Test Scenarios:

Case I: If the entered username does not exist in the database

- Input: (Kirubaraj, 123456, asdfghjkl)
- Output: User Created Successfully

Case II: If the entered username already exists in the database

- Input: (Ayush, 123456, zxcvbnm)
- Output: The username already exists