

1. Test Plan

Before running the tests, define the key areas to test and the expected outcomes.

Key Areas to Test:

1. **GUI Interaction:**
 - Ensure all buttons, input fields, and tabs function correctly.
 - Validate the responsiveness of the inventory table (automatic updates).
 2. **Items Management:**
 - Adding items with valid and invalid inputs.
 - Checking if the inventory table updates immediately.
 3. **Category Management:**
 - Adding categories and associating items with categories.
 4. **Transactions:**
 - Recording sales with sufficient stock.
 - Handling sales with insufficient stock.
 - Recording restocks and updating the inventory.
 5. **Data Persistence:**
 - Verify that data is saved to `inventory.json`.
 - Reload the program and confirm the data is correctly restored.
 6. **Error Handling:**
 - Validate the system gracefully handles invalid inputs.
-

2. Test Cases

Define test cases to systematically verify the system's behavior.

Test Case Table

Test Case ID	Scenario	Steps to Execute	Expected Outcome
TC-01	Add a valid item	Fill in all fields and press "Add Item".	The item is added to the table and appears in the JSON file.
TC-02	Add an item with missing fields	Leave some fields blank and press "Add Item".	An error message is displayed: "All fields are required."
TC-03	Record a valid sale	Select a valid SKU, enter a quantity within the stock limit, and press "Record Transaction".	Stock decreases by the entered quantity, and the table updates automatically.
TC-04	Record a sale with insufficient stock	Select a valid SKU, enter a quantity greater than the available stock, and press "Record Transaction".	An error message is displayed: "Not enough stock for SKU '...'"
TC-05	Record a valid restock	Select a valid SKU, enter a quantity, and press "Record Transaction".	Stock increases by the entered quantity, and the table updates automatically.
TC-06	Add a valid category	Enter a new category name and press "Add Category".	The category is added and available for selection.
TC-07	Load saved inventory	Close the program, reopen it, and view the inventory.	Inventory data is restored from <code>inventory.json</code> and displayed correctly.
TC-08	Automatic inventory update	Perform a sale or restock and wait for 3 seconds.	The inventory table updates automatically without user intervention.

3. Test Execution

Perform the test cases step by step and record the results.

Example Execution Steps:

1. **Start the Program:**
 - Run `main.py` and ensure all tabs load without errors.
2. **Add Items:**
 - Add a valid item and verify it appears in the table.
 - Attempt to add an item with incomplete details and confirm the error message.
3. **Perform Transactions:**
 - Test a sale and verify the stock decreases.
 - Attempt a sale with insufficient stock and confirm the error message.
 - Test a restock and verify the stock increases.
4. **Category Management:**
 - Add a category and confirm it is available for selection when adding new items.
5. **Check Data Persistence:**
 - Restart the program and confirm all data is loaded from `inventory.json`.
6. **Automatic Updates:**
 - Wait for 3 seconds after performing a transaction and confirm the table updates automatically.

4. Test Results

Create a table to record the outcomes of each test case.

Test Case ID	Result	Notes
TC-01	Pass	Item added successfully and appeared in the table.
TC-02	Pass	Error message displayed correctly for missing fields.
TC-03	Pass	Stock decreased correctly, and the table updated automatically.
TC-04	Pass	Error message displayed for insufficient stock, and no changes to inventory.
TC-05	Pass	Stock increased correctly, and the table updated automatically.
TC-06	Pass	New category added and available for item association.
TC-07	Pass	Inventory loaded successfully from <code>inventory.json</code> .
TC-08	Pass	Table updated automatically after 3 seconds.

5. Report and Observations

Write a brief summary of your observations:

- **Overall Behavior:** The system worked as expected, and all test cases passed successfully.
 - **Key Highlights:** Real-time updates and error handling worked seamlessly.
 - **Improvements:** Could include additional features like exporting inventory data to a file or adding user authentication for enhanced security.
-

6. Documentation

In the next point I add picture of my testing examples.

7. Testing Images

TC-01 Add a valid Item

The screenshot shows the 'Item Management' window of the 'Inventory Management System'. The window has three tabs: 'Items Management' (selected), 'Category Management', and 'Transactions'. The 'Item Management' section contains a form with the following fields: 'Item Name:', 'SKU:', 'Category:', 'Price:', and 'Quantity:'. Below the form, a message states 'Item 'Siemens' added successfully!'. A table displays the current inventory:

#	Item Name	SKU	Category	Price	Quantity
1	Siemens	1abc	PLC	\$2500.00	3
2	AB	123	PLC	\$2500.00	14
3	ABB	456	PLC	\$1000.00	9
4	AB	765	VFD	\$10000.00	4
5	Siemens	777	Circuit Breaker	\$500.00	10

Below the table, there is an 'Add Item' button and a status 'Total Items: 5'.

TC-02 Add an item with missing fields

The screenshot shows the 'Item Management' window of the 'Inventory Management System'. The window has three tabs: 'Items Management' (selected), 'Category Management', and 'Transactions'. The 'Item Management' section contains a form with the following fields: 'Item Name:', 'SKU:', 'Category:', 'Price:', and 'Quantity:'. The fields are populated with: '4445' for SKU, 'PLC' for Category, '4000' for Price, and '3' for Quantity. Below the form, a message states 'All fields are required.'. A table displays the current inventory:

#	Item Name	SKU	Category	Price	Quantity
1	Siemens	1abc	PLC	\$2500.00	3
2	AB	123	PLC	\$2500.00	14
3	ABB	456	PLC	\$1000.00	5
4	errr	4444	PLC	\$3000.00	2

Below the table, there is an 'Add Item' button and a status 'Total Items: 4'.

TC-03

Record a valid sale

The screenshot shows the 'Inventory Management System' window with the 'Transactions' tab selected. The left sidebar contains 'Items Management', 'Category Management', and 'Transactions'. The main area is titled 'Transactions' and contains the following form fields and controls:

- SKU: 4444
- Quantity: 4
- Transaction Type: Sale (selected from a dropdown menu)
- Record Transaction button
- Feedback message: Sold 4 units of '4444'.

TC-04

Record a valid sale

The screenshot shows the 'Inventory Management System' window with the 'Transactions' tab selected. The left sidebar contains 'Items Management', 'Category Management', and 'Transactions'. The main area is titled 'Transactions' and contains the following form fields and controls:

- SKU: 456
- Quantity: 8
- Transaction Type: Sale (selected from a dropdown menu)
- Record Transaction button
- Feedback message: Not enough stock for SKU '456'.

TC-05 Record a valid restock

Inventory Management System

Items ManagementCategory ManagementTransactions

Transactions

SKU:

123

Quantity:

5

Transaction Type:

Restock

Record Transaction

Restocked 5 units of '123'.

TC-06 Add a valid category

Inventory Management System

Items ManagementCategory ManagementTransactions

Category Management

Category Name:

Add Category

Category 'Circuit Breaker' added successfully!

Inventory Management System

Items ManagementCategory ManagementTransactions

Item Management

Item Name:

SKU:

Category:

Price:

Quantity:

Item 'AB' added successfully!

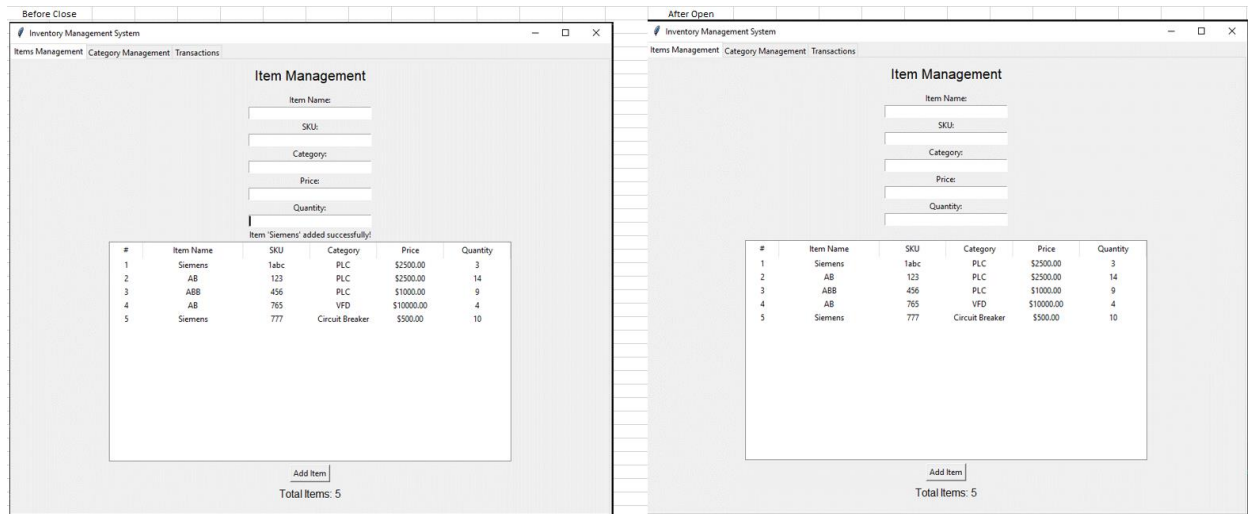
#	Item Name	SKU	Category	Price	Quantity
1	Siemens	1abc	PLC	\$2500.00	3
2	AB	123	PLC	\$2500.00	14
3	ABB	456	PLC	\$1000.00	9
4	AB	765	VFD	\$10000.00	4

Add Item

Total Items: 4

TC-07

Load saved inventory



TC-07

Load saved inventory

This is happening every time we do a transaction or every time we add an Item.