Bearcat Pantry Stocker App - Test Plan

Overall Test Plan

Our approach to testing will be a combination of focusing on individual modules of the application followed by tests of the entire application with multiple modules included. The tests will mostly be functionality tests as performance is not a critical factor in the application at the moment. They will be mostly whitebox tests in normal, abnormal and boundary conditions. Finally, we will also conduct some tests in the production environment to check whether the application runs consistently. The test plan as a whole should provide a consistent and secure application for UC students to use in the future.

Test Case Descriptions

1.1 Table Schema Modification Testing
1.2 Purpose: To test whether the schema modifications of tables worked properly
1.3 Description: This test will check the individual tables and see the modifications to the schema (addition, deletion and update of columns) applied both work properly and the data integrity is maintained
1.4 Inputs: The Table
1.5 Expected Output: The data integrity is maintained and modification is proper
1.6 Normal/abnormal/boundary: Normal
1.7 Blackbox/whitebox: Whitebox
1.8 Functional/performance: Functional
1.9 Unit/integration: Unit

2.1 Database Modification Testing
2.2 Purpose: To test whether the modifications to the database work properly
2.3 Description: This test will check whether the modifications to the database (addition and removal of tables) works correctly.
2.4 Inputs: The database
2.5 Expected Output: The table to be created is created properly
2.6 Normal/abnormal/boundary: Normal
2.7 Blackbox/whitebox: Whitebox
2.8 Functional/performance: Functional

2.9 Unit/integration: Both

3.1 End to End Testing - Weight Field
3.2 Purpose: To test whether the addition of weight field works properly
3.3 Description: The app was modified to include the weight value with the item. So, this test will check whether the addition of the field works properly throughout the app.
3.4 Inputs: User input when adding data
3.5 Expected Output: Weight value updated properly in database and ability to see weight value in item description in frontend
3.6 Normal/abnormal/boundary: All
3.7 Blackbox/whitebox: Whitebox
3.8 Functional/performance: Functional
3.9 Unit/integration: Both

4.1 End to End Testing - Transaction Table
4.2 Purpose: To test whether the transaction table works properly
4.3 Description: When checking in and out, the app now also updates the transaction table for data analysis purposes later on
4.4 Inputs: User input
4.5 Expected Output: Modification of transaction table
4.6 Normal/abnormal/boundary: All
4.7 Blackbox/whitebox: Whitebox
4.8 Functional/performance: Functional
4.9 Unit/integration: Both

5.1 Feature Testing - Filter
5.2 Purpose: To test whether the filter works properly
5.3 Description: This tests checks whether the filter implemented to filter the items works properly
5.4 Inputs: User Input
5.5 Expected Output: Only valid items are shown
5.6 Normal/abnormal/boundary: Normal and Boundary
5.7 Blackbox/whitebox: Whitebox
5.8 Functional/performance: Functional
5.9 Unit/integration: Unit

6.1 Feature Testing - Data Analysis - Total Transactions
6.2 Purpose: To test whether the query and analysis for total transactions work

6.3 Description: This test whether the implementation of analysis is correct. It would be done in 2 parts: whether the query retains correct data and whether the summarization/analysis is correct
6.4 Inputs: None
6.5 Expected Output: Proper summary and analysis values
6.6 Normal/abnormal/boundary: Normal and boundary
6.7 Blackbox/whitebox: Whitebox
6.8 Functional/performance: Functional
6.9 Unit/integration: Unit

7.1 Feature Testing - Data Analysis - Transactions by Category
7.2 Purpose: To test whether the query and analysis for transactions by category work
7.3 Description: This test whether the implementation of analysis is correct. It would be done in 2 parts: whether the query retains correct data and whether the summarization/analysis is correct
7.4 Inputs: None
7.5 Expected Output: Proper summary and analysis values
7.6 Normal/abnormal/boundary: Normal and Boundary
7.7 Blackbox/whitebox: Whitebox
7.8 Functional/performance: Functional
7.9 Unit/integration: Unit

8.1 Feature Testing - Data Visualization - Total Transactions
8.2 Purpose: To test whether the data visualization for total transactions work
8.3 Description: This tests whether the visualization of the data received from backend is correct and aesthetically pleasing on frontend
8.4 Inputs: User Inputs
8.5 Expected Output: Aesthetically pleasing and correct visualization
8.6 Normal/abnormal/boundary: Normal and Boundary
8.7 Blackbox/whitebox: Whitebox
8.8 Functional/performance: Functional
8.9 Unit/integration: Unit

9.1 Feature Testing - Data Visualization - Transactions by Category
9.2 Purpose: To test whether the data visualization for transactions by category work
9.3 Description: This tests whether the visualization of the data received from backend is correct and aesthetically pleasing on frontend
9.4 Inputs: User Input
9.5 Expected Output: Aesthetically pleasing and correct visualization
9.6 Normal/abnormal/boundary: Normal and Boundary

9.7 Blackbox/whitebox: Whitebox
9.8 Functional/performance: Functional
9.9 Unit/integration: Unit

10.1 Integration Testing - Data Dashboard - Total Transactions
10.2 Purpose: To test whether the frontend and backend interact properly for total transactions
10.3 Description: This tests checks the interaction between frontend and backend and ensures that they are safe and correct
10.4 Inputs: User Input
10.5 Expected Output: Correct data is passed between the components
10.6 Normal/abnormal/boundary: All
10.7 Blackbox/whitebox: Whitebox
10.8 Functional/performance: Functional
10.9 Unit/integration: Integration

11.1 Integration Testing - Data Dashboard - Transactions by Category
11.2 Purpose: To test whether the frontend and backend interact properly for transactions by category
11.3 Description: This tests checks the interaction between frontend and backend and ensures that they are safe and correct
11.4 Inputs: User input
11.5 Expected Output: Correct data is passed between the components
11.6 Normal/abnormal/boundary: All
11.7 Blackbox/whitebox: Whitebox
11.8 Functional/performance: Functional
11.9 Unit/integration: Integration

12.1 Feature Testing - Data Export - Total Transactions
12.2 Purpose: To test whether the data export work properly for total transactions
12.3 Description: This tests checks whether the feature to export the data to excel file works properly - aka that data is both well preserved and presented
12.4 Inputs: User Input
12.5 Expected Output: Correct excel file
12.6 Normal/abnormal/boundary: Normal and Boundary
12.7 Blackbox/whitebox: Whitebox
12.8 Functional/performance: Functional
12.9 Unit/integration: Unit

13.1 Feature Testing - Data Export - Transactions by Category

13.2 Purpose: To test whether the data export works properly for transactions by category
13.3 Description: This tests checks whether the feature to export the data to excel file works properly - aka that data is both well preserved and presented
13.4 Inputs: User Input
13.5 Expected Output: Correct excel file
13.6 Normal/abnormal/boundary: Normal and Boundary
13.7 Blackbox/whitebox: Whitebox
13.8 Functional/performance: Functional
13.9 Unit/integration: Unit

14.1 End to End Testing - Inventory to Zero Data Integrity
14.2 Purpose: Tests the change of behavior when the inventory reaches zero for a product
14.3 Description: When the inventory reaches zero, instead of deleting the record, just hide it from the normal user whereas show out of stock for admin
14.4 Inputs: User input
14.5 Expected Output: Hidden from users and available as out of stock for admins
14.6 Normal/abnormal/boundary: All
14.7 Blackbox/whitebox: Whitebox
14.8 Functional/performance: Functional
14.9 Unit/integration: Both

Test Case Matrix

Test Case	Normal / Abnormal / Boundary	Whitebox / Blackbox	Functional / Performance	Unit / Integration
1	Normal	Whitebox	Functional	Unit
2	Normal	Whitebox	Functional	Both
3	All	Whitebox	Functional	Both
4	All	Whitebox	Functional	Both
5	Normal and Boundary	Whitebox	Functional	Unit
6	Normal and	Whitebox	Functional	Unit

	Boundary			
7	Normal and Boundary	Whitebox	Functional	Unit
8	Normal and Boundary	Whitebox	Functional	Unit
9	Normal and Boundary	Whitebox	Functional	Unit
10	All	Whitebox	Functional	Integration
11	All	Whitebox	Functional	Integration
12	Normal and Boundary	Whitebox	Functional	Unit
13	Normal and Boundary	Whitebox	Functional	Unit
14	All	Whitebox	Functional	Both