

## Overall Test Plan

Our approach to testing is going to involve two forms of testing. The first component of the test involves testing system components individually. We are going to simulate data, covering any abnormal conditions and edge cases. We are going to create example data using unit test and the API puppeteer. Our second set are integration tests that test the system as a whole.

## Test Case Descriptions

### CS1.1      **Item Tests With Barcode that Exist 1**

CS1.3      This test will ensure that inserting an item into the database works when that item already exists.

CS1.4      Input: Item barcode number followed by item quantity.

CS1.5      Outputs: The existing item quantity is incremented within the database. The correct item information will be displayed on the website.

CS1.6      Normal

CS1.7      Whitebox

CS1.8      Functional

CS1.9      Unit Test

CS1.10     Results: The record will be verified against the expected outcome. A picture of the result will be displayed to the tester.

### CS2.1      **Item Tests With Barcode and Item Name that Doesn't Exist 2**

CS2.3      This test will ensure that inserting an item into the database works correctly when the item does not exist.

CS2.4      Input: Barcode number and all item information (name, weight/cost, image, quantity)

CS2.5      Outputs: All item information for the created item will be processed into the database. The correct item information for the new item will be displayed on the website.

CS2.6      Normal

CS2.7      Whitebox

CS2.8      Functional

CS2.9      Unit Test

CS2.10     Results: All new records will be verified along with a screenshot of the new item displayed to the tester.

**CS3.1            Item Tests With Barcode that Doesn't Exist and Item Name that Does 3**

CS3.3            This test will ensure that inserting an item into the database works correctly when the item name exists but barcode does not.

CS3.4            Input: The tester will input a barcode that does not exist in the database, but an item name that does.

CS3.5            Outputs: Barcode will be added to the existing item and item quantity will be incremented. The correct item information for the updated item will be displayed on the website.

CS3.6            Normal

CS3.7            Whitebox

CS3.8            Functional

CS3.9            Unit Test

CS3.10           Results: The record will be verified against the expected outcome. A picture of the result will be displayed to the tester.

**CS4.1            Item Tests Without Barcode Item Exists 4**

CS4.3            This test will ensure that inserting an item into the database works correctly when it does not contain a barcode but the item name exists in the database.

CS4.4            Input: Item name of existing item followed by the quantity to add to that item.

CS4.5            Outputs: Item quantity will be incremented. The correct item information for the updated item will be displayed on the website.

CS4.6            Normal

CS4.7            Whitebox

CS4.8            Functional

CS4.9            Unit Test

CS4.10           Results: The record will be verified against the expected outcome. A picture of the result will be displayed to the tester.

**CS5.1            Item Tests Without Barcode Item Doesn't Exist 5**

CS5.3            This test will ensure that inserting an item into the database works correctly when it does not contain a barcode and item name does not exist in the database.

CS5.4            Input: All item information (name, weight/cost, image, quantity)

CS5.5            Outputs: All item information for the created item will be processed into the database. The correct item information for the new item will be displayed on the website.

CS5.6            Normal

CS5.7            Whitebox

CS5.8            Functional

CS5.9            Unit Test

CS5.10           Results: All new records will be verified along with a screenshot of the new item displayed to the tester.

CS6.1            **Search Tests 1**

CS6.3            This test ensures that item search functionality works as expected when browsing for an item.

CS6.4            Input: An existing item name will be entered into the text box on our website used to query our database.

CS6.5            Outputs: All items with that item name will be returned by the search.

CS6.6            Normal

CS6.7            Whitebox

CS6.8            Functional

CS6.9            Unit Test

CS6.10           Results: The record will be verified against the expected outcome. A picture of the result will be displayed to the tester.

CS7.1            **Delete Tests 1**

CS7.3            This test ensures that we can delete an item record in the database.

CS7.4            Input: The automated test system will navigate to our inventory system administrator page and press the delete button on an item record.

CS7.5            Outputs: The item will be deleted from the database.

CS7.6            Normal

CS7.7            Whitebox

CS7.8            Functional

CS7.9            Unit Test

CS7.10           Results: The record will be verified against the expected outcome. A picture of the result will be displayed to the tester.

CS8.1            **Shopping Cart Tests 1**

CS8.3            This test will ensure that when the student adds an item to the shopping cart that this functionality works as expected.

CS8.4            Input: An item in the inventory management page is added to the shopping cart.

CS8.5            Outputs: The correct item data will be displayed on the students shopping cart page.

CS8.6            Normal

CS8.7            Whitebox

CS8.8            Functional

CS8.9            Unit Test

CS8.10           Results: Cart information will be verified against the expected outcome. A picture of the result will be displayed to the tester.

**CS9.1            Users Login Tests 1**

CS9.3            This test will ensure that the user is able to login to the system without errors. The login verification will be against UC's authentication system using 6+2. This will be verified using an authentication token.

CS9.4            Input: User credentials will be entered into the login page.

CS9.5            Outputs: The user will be redirected to the browse tab and a login verification message.

CS9.6            Normal

CS9.7            Whitebox

CS9.8            Functional

CS9.9            Unit Test

CS9.10          Results: The login verification message and authentication token will be checked. A screenshot will also be returned showing login verification.

**CS10.1          Administrator Login Tests 1**

CS10.3          This test will ensure that an administrator can log-in to the system without errors. The login for the administrator will be verified against our database.

CS10.4          Input: Administrator information will be entered into the login page and the administrator will press the login button.

CS10.5          Outputs: The admin to be redirected to the manage items tab and receive the admin login verification message. All administrator options will be visible.

CS10.6          Normal

CS10.7          Whitebox

CS10.8          Functional

CS10.9          Unit Test

CS10.10        Results: The admin login verification message and authentication token will be checked as well as the visibility of admin only options. A screenshot will also be returned showing login verification.

#### CS11.1        **Updating Records Tests 1**

CS11.3        This test will ensure that an administrator can update item records. The administrator will be taken to the manage items tab where the admin will be able to edit existing item data.

CS11.4        Input: The administrator will input updated item data information.

CS11.5        Outputs: The updated item data information will be displayed on the manage pantry items screen.

CS11.6        Normal

CS11.7        Whitebox

CS11.8        Functional

CS11.9        Unit Test

CS11.10       Results: The new data will be verified against the old data and a screen shot will be returned to the tester to verify the correctly updated item data.

#### CS12.1        **Barcode Scanner Tests 1**

CS12.3        This test will verify that the barcode scanner works correctly. A barcode will be scanned and the output will be verified to determine correctness.

CS12.4        Input: A barcode will be scanned on a popup.

CS12.5        Outputs: A barcode number will be outputted to the textbox which will be verified for correctness.

CS12.6        Normal

CS12.7        Whitebox

CS12.8        Functional

CS12.9        Manual Test

CS12.10       Results: A barcode will be outputted to the textbox and verified against expected result.

#### CS13.1        **Updating Shopping Cart Tests 1**

CS13.3        This test will ensure that we can update a shopping cart once a student adds data to his shopping cart.

CS13.4        Input: The student will change the order information such as item quantity.

CS13.5        Outputs: Shopping cart data will be updated.

CS13.6        Normal

CS13.7        Whitebox

CS13.8        Functional

CS13.9        Unit Test

CS13.10       Results: New shopping cart data will verified against expected result. A screenshot will be sent to the tester.

CS14.1        **Checkout Tests 1**

CS14.3        This test will ensure that order information is sent to the bearcat pantry email address when a cart is checked out and an order is created.

CS14.4        Input: A shopping cart is filled with item's and the checkout button is pressed.

CS14.5        Outputs: The order information will be sent to bearcat pantry email address.

CS14.6        Normal

CS14.7        Whitebox

CS14.8        Functional

CS14.9        Unit Test

CS14.10      Results: Existence of pantry email and its contents will be verified against expected result. Order information will be screenshot and displayed to the tester.