**Testing for SuperPrice App (Group P1-03):**

|  |  |  |
| --- | --- | --- |
| Edited by | Edited Date | Edit Info |
| Michael Xie (s3943224) | 15/09/2023 and 17/09/2023 | Added testing criteria's for the backend, Tests 18 to  Removed Test 2 for Test 22 |

*Test 1:*

**Scenario:** A customer/tester searches for a specific item by name

**User story:** As a customer who is using the webapp for the first time, I want to search for products by only name and without filters, so I can easily locate the products I am looking for without having to learn how the filtering system works

Acceptance criteria:

Criterion 1:

**Given:** the user has entered their target item name in the search bar (in navigation bar) and there are matches for that item name

**When:** the user submits with the enter key or the search button (right of the search bar)

**Then:** the user is redirected to the "Item List” page with all the results displayed

Criterion 2:

**Given:** the user has entered their target item name in the search bar but there are NO matches for that item name

**When:** the user submits with the enter key or the search button

**Then:** the user is redirected to the “Item List” page with the displayed message ‘Sorry! We could not find results for “<insert target item name>” with the applied filters!’

Criterion 3:

**Given:** the user has entered the URL "/api/items/browse?term=" followed by a product search term into their client

**When:** the user submits this URL

**Then:** the user's client will receive a JSON object which represents the list of all products which have either its name or brand containing the search term (http ok response)

Criterion 4:

**Given:** the user has entered the URL "/api/items/browse" not followed by the query parameter of "term" into their client

**When:** the user submits this URL

**Then:** the user's client will receive no JSON object and a HTTP bad request response

*~~Test 2:~~ (Test 22, includes acceptance criteria for API calls)*

**~~Scenario:~~** ~~A customer/tester filters a searched item term~~

~~Acceptance criteria:~~

**~~Given:~~** ~~the user is in the “Item List” page, has clicked on one of the filter buttons (Brand, Category, Supermarket, Price) to reveal a modal, has selected their filter options (with checkboxes or radio buttons) and there are matches for the item name with the filters~~

**~~When:~~** ~~the user presses the “Apply” button at the bottom of the modal~~

**~~Then:~~** ~~the modal is closed, and the item list is updated to show only the items which satisfy those filters.~~

**~~Given:~~** ~~the user is in the “Item List” page, has clicked on one of the filter buttons to reveal a modal, has selected their filter options and there are NO matches for the item name with the filters~~

**~~When:~~** ~~the user presses the “Apply” button at the bottom of the modal~~

**~~Then:~~** ~~the modal is closed, and the message ‘Sorry! We could not find results for “<insert target item name>” with the applied filters!’ is displayed~~

*Test 3:*

**Scenario:** A customer/tester adds an item to cart

Acceptance criteria:

**Given:** the user is in the “Home” page

**When:** the user clicks the “Add” button for one of the item cards under the title “Special Deals” to reveal a modal, has selected the quantity and pressed the "Confirm” button  
**Then:** the modal is closed, the item is added to the cart and cart icon in the navigation bar is updated to show how many items are now in the cart

**Given:** the user is in the “Product” page

**When:** the user clicks the “Add” button for one of the item cards under the title “Available At” OR underneath the image for the target product to reveal a modal, has selected the quantity and pressed the "Confirm” button  
**Then:** the modal is closed, the item is added to the cart and cart icon in the navigation bar is updated to show how many items are now in the cart

**Given:** the user is in the “Item List” page

**When:** the user clicks the “Add” button for one of the item cards to reveal a modal, has selected the quantity and pressed the "Confirm” button  
**Then:** the modal is closed, the item is added to the cart and cart icon in the navigation bar is updated to show how many items are now in the cart

*Test 4:*

**Scenario:** A customer/tester views their cart

Acceptance criteria:

**Given:** the user has items in their cart  
**When:** the user clicks the cart icon in the navigation bar

**Then:** the user is redirected to the “Checkout” page (1st step) where the list of all products and their quantity is displayed

**Given:** the user has NO items in their cart

**When:** the user clicks the cart icon in the navigation bar

**Then:** the user is redirected to the “Checkout” page with the message “Empty Cart! Please add items to the cart to proceed!”

*Test 5:*

**Scenario:** A customer/tester removes an item from their cart

Acceptance criteria:

**Given:** the user is in the “Checkout” page (1st step), has items in the cart, has pressed the “X” button to the very right of the target item box, to reveal a modal

**When:** the user presses the “Confirm” button

**Then:** the modal is closed, the item is removed from the cart and the chart icon in the navigation bar is updated to show how many items are now in the cart

**Given:** the user is in the “Checkout” page (1st step), has items in the cart, has set the quantity counter value to 0 (left of the “X” button) to reveal a modal  
**When:** the user presses the “Confirm” button  
**Then:** the modal is closed, the item is removed from the cart and the chart icon in the navigation bar is updated to show how many items are now in the cart

*Test 6:*

**Scenario:** A customer/tester tries to get an order delivered to their doorstop

Acceptance criteria:

**Given:** the user is not logged, is in the “Checkout” page, has items in the cart, has entered valid delivery details (Name, Email, Mobile, Delivery Address, Delivery Method, Preferred Delivery Time) and has entered valid payment details  
**When:** the user clicks the “Confirm” button

**Then:** the page displays the message “Completed! Thank you for shopping at Super Price!” and the order is added to the system

**Given:** the user is logged in, is in the “Checkout” page, has items in the cart, has changed the delivery details to be valid (auto filled in from user profile details) and has entered valid payment details

**When:** the user clicks the “Confirm” button

**Then:** the page displays the message “Completed! Thank you for shopping at Super Price!” and the order is added to the system

**Given:** the user is in the “Checkout” page. Has items in the cart and the delivery details or the payment details are invalid

**When:** the user clicks the “Next” button (for delivery details) or “Confirm” button (for payment details)

**Then:** an error message for every incorrect field is displayed under the respective text field and the the text field is highlighted read

*Test 7:*

**Scenario:** A customer/tester compares the price of a product across supermarkets

**User story:** As a customer, I want to be able to compare the price of a product between supermarkets so that I know where I can get the product the cheapest

Acceptance criteria

Criterion 1:

**Given** the user has entered the url "/api/products/" followed by a valid product id in their client

**When** the user submits this url

**Then** the user's client will receive a JSON object containing their requested product (and http ok response)

**And** with this data, the user can view the base price and deals of the product from each supermarket and calculate which supermarket has the product the cheapest

Criterion 2:

**Given:** the user is in the “Product” page for their target product

**When:** theuser looks at the product cards (with price after the deal is applied shown) underneath the title “Available At”

**Then:** the user can see that the product across supermarkets is arranged from cheapest on the left to most expensive on the right

*Test 8:*

**Scenario:** A customer/tester will be notified when a product is on special.

Acceptance criteria:

**Given:** the user is on the SuperPrice application, and their browser is constantly asking the server for deals (sends a request once every 4 seconds)

**When:** a deal for a product at a supermarket has been added to the system

**Then:** then a popup for that product will appear on the website.

*Test 9:*

**Scenario:** The customer/tester can navigate to at least 3 pages

Acceptance criteria:

**Given:** the user is on the SuperPrice application

**When:** the user clicks the SuperPrice logo

**Then:** the user is redirected to the “Home” page

**Given:** the user is not logged in

**When:** the user clicks the Guest profile image icon in the navigation bar

**Then:** the user is redirected to the “Login”

**Given:** the user is logged in

**When:** the user clicks their profile image in the navigation bar

**Then:** the user is redirected to their “Profile” page

**Given:** the user is in the “Login” page

**When:** the user clicks the “Register” link under “New to SuperPrice?”

**Then:** the user is redirected to the “Register” page

*Test 10:*

**Scenario:** A customer/tester tries to register for an account

Acceptance criteria:

**Given:** the user is in the “Register” page, has entered a name, a valid email address and two matching passwords

**When:** the user presses “Register” button

**Then:** their credentials will be added to the system (password hashed), and they will be redirected to their “Profile” page

**Given:** the user is in the “Register” page and has entered incorrect registration details

**When:** the user presses the “Register” button  
**Then:** error messages will be displayed, one under each incorrect field which will be highlighted red

*Test 11:*

**Scenario:** A customer/tester tries to log into their account

Acceptance criteria:

**Given:** the user is in the “Login” page and has entered valid credentials (existing email, password pair)

**When:** the user presses the “Login” button

**Then:** they will be logged into their account and be redirected to their “Profile” page.

**Given:** the user is in the “Login” page and has entered invalid credentials (not an existing email, password pair),

**When:** the user presses the “Login” button

**Then:** the error message “Invalid Email or Password” will be displayed under both text fields, and they will be highlighted red

*Test 12:*

**Scenario:** A customer/tester tries to log out from their account

Acceptance criteria:

**Given:** the user is logged is either in their “Profile” page

**When:** the user presses the “Logout” button in the sub navigation bar

**Then:** the system will terminate the session and the user will be redirected to the “Home” page.

*Test 13:*

**Scenario:** A customer/tester tries to write a review for an item

Acceptance criteria:

**Given:** the user is logged in, is in the “Product” page for the target item, has entered their rating (has clicked the Nth star from the left for an N out of 5 rating) and their comment (in a text area) under the “Write a review?” title in the “Review” section

**When:** the user presses the “Submit” button

**Then:** the “Write a review?” box’s title is changed to “Edit your review?”, the creation date is displayed underneath the title and the “Submit” button is changed to “Edit” button.

**Given:** the user tried the same conditions above but was not logged in  
**When:** the user presses the “Submit” button

**Then:** the user will be redirected to the “Login” page.

*Test 14:*

**Scenario:** A customer/tester tries to edit a review for an item

Acceptance criteria:

**Given:** the user is logged in, is in the “Product” page for the target item, has changed their rating (has clicked the Nth star from the left for an N out of 5 rating) or comment (in a text area) under the “Edit your review?” title in the “Review” section

**When:** the user presses the “Edit” button

**Then:** the edit date underneath the box’s title should be changed to the new edited date.

**Given:** the user is logged in, and is in the “Review History” page, has scrolled down to the target product’s review box and has changed its rating or comment

**When:** the user presses the “Edit” button

**Then:** the edit date underneath the box’s title should be changed to the new edited date.

**Given:** the user is logged in and did not change the reviews rating or the comment (in “Review History” or “Product” page)

**When:** the user presses the “Edit” button

**Then:** the user is shown the warning message “There was no change to the review!” under the comment text area

*Test 15:*

**Scenario:** A customer/tester tries to change their password

Acceptance criteria:

**Given:** the user is logged in, is in their “Profile” page and has entered a strong password which matches with the confirmation password

**When:** the user presses the “Save” button

**Then:** the system will change the account’s password and both text fields will be highlighted green.

**Given:** the user is logged in, is in their “Profile” page but did not enter a strong password

**When:** the user presses the “Save” button

**Then:** error messages will be displayed, one under each incorrect field which will be highlighted red

*Test 16:*

**Scenario:** A customer/tester tries to find the summary of a previous purchase

Acceptance criteria:

**Given:** the user is logged in, is in their “Purchase History” page

**When:** the user clicks on their target purchase

**Then:** a box containing the full summary of the purchase will appear under the clicked button

*Test 17:*

**Scenario:** A customer/tester tries to find a review but cannot remember the product the review was written for

Acceptance criteria:

**Given:** the user is logged in and is in their “Review History” page

**When:** they locate the target product’s review box

**Then:** they can see their review rating and comments for the product

*Test 18:*

**Scenario:** A customer/tester tries to search for a specific product through their client

**User story:** As a customer, I want to see access information about a particular product so that I don’t have look through the entire product list to find information about my required product

Acceptance criteria:

Criterion 1:

**Given** that the user has entered the url "/api/product/" followed by an valid existing product id (type long) into their client

**When** the customer submits the url

**Then** the customer's client will display a JSON object showing the attributes of the product (id, name, description, brand, category, but not including the image) of the specific id and receive a http ok status response

Criterion 2:

**Given** that the user has entered the url "/api/product/" followed by an valid non-existing product id (type long) into their client

**When** the customer submits the url

**Then** the customer's client will receive no JSON object and will get a http ok status response

Criterion 3:

**Given** that the user has entered the url "/api/product/" follow by an invalid product id (not of type long, for example "a") into their client

**When** the customer submits the url

**Then** the customer's client will receive no JSON object and will get a http bad request status response

Criterion 4:

**Given** that the user is in the "Product list" page on their browser and their mouse is hovering on a product card's item name link and the product exists in the system

**When** the user clicks on the product card's item name link

**Then** the customer is redirected to the product's page which has information about product's name, description, brand and supermarkets which sells the product (including the base price and any deals)

Criterion 5:

**Given** that the user is in the "Home" page on their browser and their mouse is hovering on a product deal card's item name link and the product exists in the system

**When** the user clicks on the product card's item name link

**Then** the user is redirected to the product's page which has information about product's name, description, brand and supermarkets which sells the product (including the base price and any deals)

Criterion 6:

**Given** that a product doesn't exists in the system and the user's mouse is hovering on a product's link

**When** the user clicks on a product's link (from "Product List" or "Home" page)

**Then** the user is redirected to an error product page which displays the error message "Product does not exist"

*Test 20:*

**Scenario:** A customer/tester wants to view the logo for a specific supermarket

**User story:** As a customer, I want to view the logos for specific supermarkets so I can better recognise which products belong to which supermarket

Acceptance criteria:

Criterion 1:

**Given** that the user has entered the url "/api/images/supermarket" followed by a valid existing supermarket id (type long) into their client

**When** the customer submits the url

**Then** the customer's client will display a PNG image for the supermarket of the requested id and receive a http ok status response

Criterion 2:

**Given** that the user has entered the url "/api/images/supermarket" followed by a valid non-existing supermarket id (type long) into their client

**When** the customer submits the url

**Then** the customer's client will receive no PNG image and will get a http ok status response

Criterion 3:

**Given** that the user has entered the url "/api/images/supermarket" follow by an invalid supermarket id (not of type long, for example "a") into their client

**When** the customer submits the url

**Then** the customer's client will receive no PNG image and will get a http ok status response

Criterion 4:

**Given** that the user is in the "Product", "Product List", "Home”, "Cart" page and all the relevant supermarket logos exists in the system

**When** the page contents fully load

**Then** the user's browser should display the correct supermarket logos for each product

Criterion 5:

**Given** that the user is in the "Product", "Product List", "Home”, "Cart" page and not all the relevant supermarket logos exists in the system

**When** the page contents fully

Then the user's browser should display error images for supermarket logos that don't exist in the system

*Test 21:*

**Scenario:** A customer/tester wants to view image for a specific product

**User story:** As a customer, I want to view the images for specific product so I can better recognise what product each product represents

Acceptance criteria:

Criterion 1:

**Given** that the user has entered the url "/api/images/product" followed by a valid existing product id (type long) into their client

**When** the customer submits the url

**Then** the customer's client will display a PNG image for the product of the requested id and receive a http ok status response

Criterion 2:

**Given** that the user has entered the url "/api/images/product" followed by a valid non-existing product id (type long) into their client

**When** the customer submits the url

**Then** the customer's client will receive no PNG image and will get a http ok status response

Criterion 3:

**Given** that the user has entered the url "/api/images/supermarket" follow by an invalid product id (not of type long, for example "a") into their client

**When** the customer submits the URL

**Then** the customer's client will receive no PNG image and will get a http ok status response

Criterion 4:

**Given** that the user is in the "Product", "Product List", "Home”, "Cart" page and all the relevant product images exists in the system

**When** the page contents fully load

**Then** the user's browser should display the correct product images for each product card

Criterion 5:

**Given** that the user is in the "Product", "Product List", "Home”, "Cart" page and not all the relevant product images exists in the system

**When** the page contents fully

Then the user's browser should display error images for product images that don't exist in the system

*Test 22:*

**Scenario:** A customer/tester tries to filter for products

**User story:** As a customer, I want to filter and sort products by certain fields so that I can locate the specific product that I am looking for

Acceptance criteria:

Criterion 1:

**Given** that the user has entered the url "/api/items/browse?" followed by valid filter query parameters (term (required), brand (list or null), category (list or null), priveAbove (decimal), priceBelow (decimal), sortBy (string default product name), orderBy (default ascending order), page (integer default 0)).

**When** the user submits the url

**Then** the user client will display products that match the filters (and get a HTTP ok response)

Criterion 2:

**Given** that the user has entered the url "/api/images/product" followed by a valid non-existing filter into their client

**When** the user submits the url

**Then** the user's client will display products that match the filters (it ignores the filters that doesn’t exist) and will receive an ok HTTP response.

Criterion 3:

**Given** that the user has entered the url "/api/images/supermarket" follow by invalid filter query parameters into their client

**When** the user submits the url

**Then** the user's client will receive no results and will get a http bad request status response

Criterion 4:

**Given** that the user is in the "Product List page" and has a filter modal open (brand, category, supermarket, price) OR their mouse is hovering over a "sort by" selection box element, and if at least one product exists in the system if the filters were to be applied

**When** the user either clicks the "Apply" button for the filter modal or clicks on a "sort by" selection box element

**Then** the user's browser will display all the products (as cards) which matches the filters that they have applied

Criterion 5:

**Given** that the user is in the "Product List page" and has a filter modal open (brand, category, supermarket, price) OR their mouse is hovering over a "sort by" selection box element, and if no product exists in the system if the filters were to be applied

**When** the user either clicks the "Apply" button for the filter modal or clicks on a "sort by" selection box element

**Then** the user's browser will display an error message saying that "Sorry! We could not find results with those filters!"

*Test 23:*

**Scenario:** A customer/tester tries to view products which has deals

**User story:** As a customer I want to be able to view the latest deals so I can purchase products for a lower price

Acceptance criteria:

Criterion 1:

**Given** that the user has entered the url "/api/items/deals" into their client

**When** the user enters the url

**Then** the user client will receive all the products which have deals and will get an ok HTTP response

Criterion 2:

**Given** that the user is in the "Home" page

**When** page fully loads

**Then** the user will see all the products which have a deal

Criterion 3:

**Given** that the user is in the "Home" page and a deal has expired

**When** the user reloads the page

**Then** the browser will no longer display the expired deal