

Lab 3 - Refill.Me Prototype Test Plan

Team Iron

Old Dominion University

CS411W

Professor Sarah Hosni

May 4, 2023

Version 2

Table of Contents

1. Objectives.....	2
2. References.....	3
3. Test Procedures.....	5
3.1 User Interface Test.....	5
3.2 Account Management Tests.....	12
3.2.1 Account Creation Test.....	12
3.2.2 Password Security Test.....	15
3.3 Store Search Test.....	17
3.4 Google Maps Integration Test.....	19
3.5 Product Search Test.....	21
3.6 Shopping List Test.....	24
3.7 Container Guide Test.....	27
3.8 Advocacy and Petition Sites Test.....	29
3.9 Customized Container Recommendation Test.....	31
3.10 Reward.Me Test.....	33
3.11 Store Rating Test.....	35
3.12 Lemme.Know Test.....	37
3.13 Product Upload Test.....	40
3.14 Reward Point Conversion Test.....	42
3.15 Profile Tests.....	44
3.15.1 Shopper Profile Test.....	44
3.15.2 Store Owner Profile Test.....	47
3.16 Navigation Menu Tests.....	51
3.16.1 Navigation Menu for the Shopper.....	51
3.16.2 Navigation Menu for the Store Owner.....	56
4. Traceability Matrix.....	62

1. Objectives

The prototype test plan is designed to test the functionality and usability of a web application that is created for users who are interested in conducting package-free shopping and businesses who want to connect with those users. The plan includes identifying any technical issues or bugs in the application, testing the application's usability and user-friendliness, evaluating the application's performance, including its speed and responsiveness, and to ensure that the application is secure and that user data is protected. (*O: Hull, MI: Colombini*)

[This space intentionally left blank]

2. References

Team Iron. (2023, March 8). Lab1 - Refill.Me Product Description. Retrieved April 5, 2023, from <https://cs.odu.edu/~411iron/#labSection>

Team Iron (2023, March 15). Lab 2 - Refill.Me Prototype Product Specification. Retrieved April 5, 2023, from <https://cs.odu.edu/~411iron/#lab2Section>

[This space intentionally left blank]

3. Test Procedures

3.1 User Interface Test (*O: Colombini; M1: Maximova*)

Test Category:		Description:		
System		This test case focuses on testing the User Interface. It includes the login screen, registration screen, user profile dashboard, and user home screen.		
Test Case:		Case Name:	Version:	Written By:
3.1.1		User Interface	1.0	Michael Colombini
Requirements Fulfilled:		Purpose:		
3.1.1.1 - 3.1.1.3		To ensure that the User Interface meets the requirements listed for each screen.		
Setup Conditions:				
1. The Refill.Me application is installed and running on the user’s device.				
2. The user has an active internet connection.				
3. The user has not logged in or registered previously.				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	Launch the Refill.Me application. Verify that the welcome screen appears with a clear and concise message informing the user of the purpose and benefits of the application.			Upon opening the application, the welcome screen should display a clear and concise message to the user, informing them of the purpose and benefits of the application.
2	If welcome screen is not displaying correctly: clear the cache and try to load the page one more time			Cache cleared up, user is able to successful to preview the page without any anomalies

3	<p>Verify that the welcome screen displays the application logo. Verify that the welcome screen displays a button to log in or create an account. Verify that the welcome screen provides navigation options for returning users. Verify that the welcome screen has a responsive design that adjusts to the screen size and orientation of the user's device. Verify that the welcome screen has an appealing design that aligns with the overall branding and style of the application.</p>			<p>The welcome screen should display the application logo, a button to login into the application and the option for the user to access a Forget a Password or to create an account, providing navigation options for returning users, allowing them to quickly access the main functionality of the application. The welcome screen should have a responsive design that adjusts to the screen size and orientation of the user's device, should have an appealing design that aligns with the overall branding and style of the application.</p>
4	<p>Tap on the login button. Verify that the login screen appears and requires users to authenticate themselves before gaining access to the application.</p>			<p>Upon accessing the login screen, users should be required to authenticate themselves before gaining access to the application.</p>
5	<p>Verify that the login screen requires users to enter their username and password to authenticate their identity. Verify that the login screen provides a "forgot password" feature to allow users to reset their password in case they forgot it. Verify that the login screen includes a "remember me" option to allow users to stay logged in on the device they are using.</p>			<p>The login screen should require users to enter their username and password to authenticate their identity, should provide a "forgot password" feature to allow user to reset their password incase they forgot it, and should include a "remember me" option to allow users to stay logged in on the device they are using.</p>

6	<p>Verify that the login screen includes different authentication methods, such as social media, multi-factor authentication, biometric authentication, and other types of authentications. Verify that the login screen provides feedback to users if they enter incorrect login credentials, including error messages that are easy to understand and actionable.</p> <p><i>If the incorrect credentials are used; an error message appears to try again;</i></p> <p><i>In case of fake email' domain/top domain name , an error to use a real email</i></p> <p><i>Please logout from the successful account in order to proceed with the test case</i></p>			<p>The login screen should include different authentication methods, such as social media, multi-factor authentication, biometric, and other types. The login screen should provide feedback to users if they enter incorrect login credentials, including error messages that are easy to understand and actionable.</p> <p>Upon reentering the credentials, everything works;</p> <p>Upon using the correct email domain, Shopper is able to login into the app</p> <p>User logged out</p>
7	<p>Verify that the <u>login</u> screen has a responsive design that adjusts to the screen size and orientation of the user's device. Verify that the login screen has an appealing design that aligns with the overall branding and style of the application.</p>			<p>The login screen should have a responsive design that adjusts to the screen size and orientation of the user's device and should have an appealing design that aligns with the overall branding and style of the application.</p>
8	<p>Tap on the create account button. Verify that the registration screen appears and collects users' information such as name, email, username, password, and any other required information necessary for account creation.</p>			<p>Upon accessing the registration screen, users should be able to enter their information, such as name, email, username, password, and any other required information necessary for account creation.</p>

9	Verify that the registration screen validates user passwords for strength and complexity and provides feedback to users if they enter a weak password. Verify that the registration screen requires email verification to ensure that the email address entered by the user is valid and belongs to the user.			The registration screen should validate that a user's password is at least 10 characters in length and contains at least one number and a symbol. The system should provide feedback to users if they enter a noncompliant password. The registration screen should require email verification to ensure that the email address entered by the user is valid and belongs to the user.
10	Verify that the registration screen offers social media login options.			The registration screen should display social media login options for Google and Facebook.
11	Verify that the registration screen displays password requirements, such as minimum length, special character requirements, and other specifications necessary for account security. Verify that the registration screen provides feedback to users if they enter incorrect or incomplete information, including error messages that are easy to understand and actionable.			The registration screen should display password requirements, such as minimum length, special character requirements, and other specifications necessary for account security. The registration screen should provide feedback to users if they enter incorrect or incomplete information, including error messages that are easy to understand and actionable.

12	Verify that the registration screen has a responsive design that adjusts to the screen size and orientation of the user's device. Verify that the registration screen has an appealing design that aligns with the overall branding and style of the application.			The registration screen should have a responsive design that adjusts to the screen size and orientation of the user's device and should have an appealing design that aligns with the overall branding and style of the application.
13	Tap on the profile dashboard button. Verify that the user profile dashboard screen appears and displays the user account information, including name, profile picture, and other relevant information.			Upon accessing the user profile dashboard screen, users should be able to view their account information, including name, profile, and other relevant information.
14	Verify that the user profile dashboard screen displays the user's saved stores, an option for users to share the application with friends, and their profile settings.			The user profile dashboard screen should display the user's saved store, an option for users to share the application with friends, and their profile settings.
15	Verify that the profile settings screen displays options for the user to edit their account information, change their password, or delete their account. Verify that the profile settings screen provides feedback to users if they enter incorrect or incomplete information, including error messages that are easy to understand and actionable.			The user should be able to edit their account information, change their password, or delete their account. Feedback should be provided to the user if they enter incorrect or incomplete information.
16	Click on the search button. Verify that the search functionality allows users to quickly find relevant information or content with the application. Enter a search term and click			The search functionality works as expected on the user profile dashboard screen.

	search. Verify that the search results are relevant to the search term entered by the user.			
17	Verify that the notifications related to account activity, messages, updates, or alerts are displayed on the user profile dashboard screen.			Notifications related to account activity, messages, updates, or alerts are displayed on the user profile dashboard screen.
18	Verify that the user profile dashboard screen has a responsive design that adjusts to the screen size and orientation of the user's device and has an appealing design that aligns with the overall branding and style of the application.			The user profile dashboard screen has a responsive design that adjusts to the screen size and orientation of the user's device and has an appealing design that aligns with the overall branding and style of the application.
19	Click on the user home screen button on the navigation menu. Verify that the store search product search, container guide, shopping list, Reward.Me, advocacy and petition site, and Contact Us button is displayed on the user home screen.			The user is navigated to the user home screen. The store search, product search, container guide, shopping list, Reward.Me, advocacy and petition site, and contact us button is displayed on the user home screen.
20	Verify that the navigation menu is displayed on the user home screen and includes the following buttons: user home screen button, store search screen button, product search screen button, user shopping list screen button, and profile dashboard screen button.			The screen displays a navigation menu that allows users easy access to different areas of the application and includes the following buttons: user home screen button, product search screen button, user shopping list screen button, and profile dashboard screen button.

21	Verify that the search functionality works as expected on the user home screen.			The screen offers search functionality to help users quickly find relevant information or content within the application.
22	Verify that the user home screen has a responsive design that adjusts to the screen size and orientation of the user's device and has an appealing design that aligns with the overall branding and style of the application.			The screen has a responsive design that adjusts to the screen size and orientation of the user's device and aligns with the overall branding and style of the application.

[This space intentionally left blank]

3.2 Account Management Tests

3.2.1 Account Creation Test (*O: Colombini, M1: Rios*)

Test Category:		Description:		
System		Testing the functionality of creating a new account on the system.		
Test Case: 3.1.2.1.1		Case Name: Account Creation	Version: 1.0	Written By: Michael Colombini
Requirements Fulfilled: 3.1.2.1.1.1 – 3.1.2.1.1.8		Purpose: To verify that a user can create an account.		
Setup Conditions: 1. The user has an internet connection. 2. The Refill.Me application is installed and running on the user’s device. 3. The user is not logged in to an account. 4. The user is viewing the registration screen.				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	Enter valid name, email address, and password in the registration form.			The system shall provide a registration process for users to create a new account, which includes collecting basic information such as name, email, and password.
2	Click on the submit button.			The system shall validate users’ inputs and ensure that all required fields are completed before creating a new account.

				The system shall enforce password requirements to ensure that passwords are strong and secure, including minimum length, complexity, and character type.
3	Verify that the user is redirected to the confirmation page.			The system should handle errors that may occur during the account creation process, such as incorrect inputs or server errors.
4	Verify that a confirmation email is sent to the user's email address.			The system shall authenticate users during the account creation process to ensure that only authorized users can create a new account.
5	Click on the confirmation link in the email.			The system shall provide a confirmation mechanism to verify the user's email address and activate the new account.
6	Verify that the user is redirected to the account activation page.			The system should handle errors that may occur during the account creation process, such as incorrect inputs or server errors.
7	Enter the valid activation code provided in the confirmation email.			The system shall provide a confirmation mechanism to verify the user's email address and activate the new account.

8	Click on the activate button.			The system shall create a user profile for each new account by storing the user's information in a database record. The profile record shall include basic information such as name and email address, and it may also include optional information such as profile picture and location. The profile record shall be associated with the user's account and shall be accessible through the user's profile page.
9	Verify that the user is redirected to the login page.			The system should handle errors that may occur during the account creation process, such as incorrect inputs or server errors.

[This space intentionally left blank]

3.2.2 Password Security Test (O: Hull, M1: Wasikye)

Test Category: Security		Description: Testing the security of user passwords.		
Test Case: 3.4.2		Case Name: Password Security	Version: 1.0	Written By: Katelynn Hull
Requirements Fulfilled: 3.4.2.1-3.4.2.2		Purpose: To allow users to set up secure passwords to prevent unauthorized access to their account information.		
Setup Conditions: 1. User must access Refill.Me with a Wi-Fi enabled device. 2. User must provide their email address to enable verification of account information				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	User clicks register to create an account with Refill.Me			A registration page will open prompting the user to enter their email address and potential password.
2	User enters their email address and a password that is insufficient in length.			The registration page will display a prompt informing the user that the password is too short.
3	User enters a password that meets the length requirements.			A message will be sent to the user’s email address confirming account setup.

4	User confirms account setup and successfully logs into account.			The user will be able to access the rest of Refill.Me's features.
---	---	--	--	---

[This space intentionally left blank]

3.3 Store Search Test (O: Colombini, M1:Wasikye)

Test Category:		Description:		
System		Location based search for package free stores.		
Test Case: 3.1.2.2		Case Name: Store Search	Version: 1.0	Written By: Michael Colombini
Requirements Fulfilled: 3.1.2.2.1 – 3.1.2.2.2		Purpose: The purpose of the test case is to test the store search functionality including location-based searching, store display on a map, and store information display.		
Setup Conditions: 1. The user has a device with GPS capabilities. 2. The user has internet access. 3. The system is operational. 4. The user has an account. 5. The user has logged in to Refill.Me				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	Verify that the user’s current location is displayed on the map. If the user’s location is not displayed, enter a valid zip code. Tap the “Search” button.			The user is able to see their current location on the map
2	Select the option to search for stores			The user is able to search for stores based on their current location or entered zip code
2	Verify that the map displays the 5 nearest stores in the user’s location or entered zip code, marked with pins.			The map displays the 5 nearest stores to the user’s location or entered zip code.

3	Tap on a store's pin to open its information window. Verify that the store's basic information is displayed in the window, including the store name and address. Tap on the "Details" button in the store information window.			The store information window displays the store's name, address, opening hours, website, telephone number, and a "View Products" button of the selected store.
4	Tap the "View Products" button. Verify that the user is taken to a new page displaying the store's products.			The "View Products" button takes the user to a new page displaying the store's products.
5	Tap on another store's pin on the map. Verify that the previous store's information window is closed, and the new store's information window is displayed. Repeat steps 7-14 for at least two other stores.			The user can switch between different store information windows by tapping on different pins on the map.

[This space intentionally left blank]

3.4 Google Maps Integration Test (O: Rios, M1: Hull)

Test Category: User Interfaces		Description: This test is to verify that the Google Maps API can be properly and fully interacted with.		
Test Case: 3.1.1.2.2		Case Name: Google Maps Integration	Version: 1.0	Written By: Gavin Rios
Requirements Fulfilled: 3.1.1.2.2.1 - 3.1.1.2.2.6		Purpose: The system shall use Google’s Maps API to display nearby stores in the Store Search Screen.		
Setup Conditions: 1. The shopper has created an account. 2. The shopper is accessing Refill.Me on an internet-connected device. 3. The shopper has given Refill.Me access to their location. 4. The shopper is logged in to their user account. 5. The user has navigated to the store search screen.				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	The user is viewing the main store search screen			The system shows a map where the user can view nearby stores and limited context information
2	The user gestures with two fingers in a linear ‘drag’ motion			The system pans the map in a direction corresponding to that of the gesture.
3	The user gestures with a two finger pinch			The system zooms out the map.
4	The user gestures by separating two fingers			The system zooms in on the map viewing area.

5	The user uses a 'pinch', 'separating', or 'drag' gesture to change the view area of the map.			New store entries are populated as their location becomes visible on the map. Old store entries are depopulated similarly. Context windows are displayed when there is sufficient space to do so without overlapping.
6	The user taps the 'details' button on a store's context window.			The system navigates the user to that store's store info page.
7	The user is navigated to a store info page.			The system displays the store title, location, hours, and contact information. A "View Products" button is displayed at the bottom of the screen.
8	The user taps on the "View Products" button on a store info page.			The system navigates the user to the store's product listing page.
9	The user is navigated to a store's product listing page.			The system displays a list of the package-free products the store offers. A "Go Back" button is displayed at the bottom of the screen.

[This space intentionally left blank]

3.5 Product Search Test (O: Rios, M1: Colombini, M2: Hull)

Test Category: User Interfaces - Product Search Screen		Description: This test is to verify that information on the Product Search Screen is properly displayed.		
Test Case: 3.1.2.3		Case Name: Search Products	Version: 1.0	Written By: Gavin Rios
Requirements Fulfilled: 3.1.2.3.1 - 3.1.2.3.3		Purpose: To verify that shoppers can search for package-free products by icon or by name, and that the system will display the search results in a prompt and efficient manner.		
Setup Conditions: 1. The shopper has created an account. 2. The shopper is accessing Refill.Me on an internet-connected device. 3. The shopper is logged in to their user account. 4. The user has navigated to the product search screen.				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	The user is viewing the main product search screen.			The system displays options to search by product or icon, with descriptions for each search type
2	The user selects the “Search By Icons” option			The system displays a screen with icons of products representing the corresponding product categories.
3	The user selects the “Search by Name” option			The system displays a screen with a title prompt “I am looking for...” and a search bar.

4	On the “Search by Icon” screen, the user selects the icon for the product they wish to search for			The system displays a product listing screen with results for the product the user selected.
5	On the “Search by Name” screen, the user supplies a search query and presses the search button.			The system displays a product listing screen with results for products substantially similar to the user’s query.
6	The user is viewing the search results screen.			Results are sorted by store, then by the user’s choice. Page navigation is present at the bottom of the screen. 10 results are displayed per page.
7	The user filters the search results by selecting specific product categories.			The search results are filtered by specific product categories.
8	The user filters the search results by selecting specific store names.			The search results are filtered by a specified store name.
9	The user’s query yields no product results			The system displays a screen informing the user that no matching products were found. A button at the bottom of the screen allows the user to “Search Again”.
10	The user’s query yields one or more product results			On the product results screen, the system displays the store, name, and price for each item. Two icon buttons are displayed for viewing the product’s container suggestions and adding the product to the user’s shopping list. A “New Search” button is shown at the end of the results.

11	The user limits the area of the search results to a specific mileage amount.			The system displays the available products within the specified mileage.
12	The user presses the container icon button for a product.			The system displays a dialog box with the recommended container types in a location directly above or below the container icon button.
13	The user presses the add icon button for a product.			The product is added to the user's shopping list.
14	The user presses the "New Search" button			The user is returned to the main product search screen.
15	The user changes the orientation of the device			Content is reorganized to fully utilize the screen area in an organized manner.

[This space intentionally left blank]

3.6 Shopping List Test (O: Chang-Horvath, M1: Colombini)

Test Category: Shopping List		Description: This test is to verify that a user can use the built-in shopping list.		
Test Case: 3.1.2.4		Case Name: Shopping List	Version: 1.0	Written By: Judit Chang-Horvath
Requirements Fulfilled: 3.1.2.4.1 - 3.1.2.4.5		Purpose: To verify that the user can view the shopping list, add/remove items to/from the shopping list, modify the quantity of the items in the shopping list, and get information about the container type recommended for a specific item.		
Setup Conditions: <div>1. Users must run the Refill.Me application on a device with Internet connection.</div> <div>2. Users must have credentials for an account.</div> <div>3. Users must be logged in into the account.</div> <div>4. Users must navigate to the product search page.</div> <div>5. Users must search for a product.</div> <div>6. Users must have a product listing based on his/her product search.</div>				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	User chooses the “+” sign next to the name of a product in the product listing to add an item to the built-in shopping list.			The application will add the item for which the “+” sign was tapped to the built-in shopping list in the user’s account. The “+” sign will change to a check mark to indicate that the product was added to the Shopping List.
2	User navigates to the Shopping List page by choosing the			The application will open the Shopping List page, and display

	corresponding icon at the bottom of the screen.			already added items grouped by store. Under each item's name, a plus sign will be displayed, and a container icon. On the right side of each item the unit price will be shown. Under the unit price, the default amount will show between the up and down arrows that allow the modification of the amount. After each section that displays the product offered by a specific store, a "Container recommendation" button will be displayed.
3	User taps on the container icon under the product's name.			The application will open a pop-up window and display the information regarding the recommended container for the product. When the user taps on screen again, the pop-up window will fade away.
4	User modifies the default amount to his/her preferred amount by using the up or down arrows.			The application will increase/decrease the number representing the amount of the specific product.

5	User taps on the “Container recommendation” button.			The application will open the “Customized Container Recommendation” page.
6	User taps on the “-” sign under the product’s name to remove the product from the Shopping List.			The application will remove the product from the Shopping List by making the entry fade away. Once the product is removed, the remaining products are being shifted to show the product listing in an organized manner.
7	User taps on the “-” sign under the product’s name to remove the last product(s) from the Shopping List.			The application will remove the last item from the Shopping List by making the entry fade away. Once the product is removed, the application will display the message “Shopping List is empty”.

3.7 Container Guide Test (O: Chang-Horvath, M1: Hull)

Test Category: Container Guide		Description: This test is to verify that a user can read information in the Container Guide.		
Test Case: 3.1.1.2.4		Case Name: Container Guide	Version: 1.0	Written By: Judit Chang-Horvath
Requirements Fulfilled: 3.1.1.2.4.1 - 3.1.1.2.4.6		Purpose: To verify that the user can read about different container types, the materials that they are made of, and the types of products that can be stored in them.		
Setup Conditions: 1. Users must run the Refill.Me application on a device with Internet connection. 2. User must navigate to the Container Guide page of the application.				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	User sees/reads the information about the different container types.			The page is populated with information based on the data collected by the algorithm for retrieving information from the database about container types and their characteristics.
2	User scrolls down (and eventually up) the Container Guide page to read all information about the different container types.			The application will allow the user to scroll down and then up on the

				Container Guide page.
--	--	--	--	--------------------------

[This space intentionally left blank]

3.8 Advocacy and Petition Sites Test (O: Chang-Horvath, M1: Hull)

Test Category: Advocacy and Petition sites		Description: This test is to verify that a user can read information on the Advocacy and Petition Sites page		
Test Case: 3.1.1.2.8		Case Name: Advocacy and Petition Sites	Version: 1.0	Written By: Judit Chang-Horvath
Requirements Fulfilled: 3.1.1.2.8.1 - 3.1.1.2.8.8		Purpose: To verify that the user can access the list of advocacy and petition sites, and he/she can access the websites associated with the different advocacy and petition sites.		
Setup Conditions: 1. User must run the Refill.Me application on a device with Internet connection. 2. User must navigate to the Advocacy and Petition Sites page of the application.				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	User sees/reads the list of different advocacy and petition sites showing the name of the website or organization.			The page displays a short message about the purpose of the page, and a list of buttons with the names of different advocacy and petition websites and/or organizations.
2	User scrolls down (and eventually up) the Advocacy and Petition Sites page to see the names of all the different advocacy and petition websites/organizations.			The application will allow the user to scroll down and then up on the Advocacy and Petition Sites page.

3	User taps on a button with the name of a specific advocacy and/or petition website/organization.			The application will ask the user about his/her preferred browser.
4	User chooses his/her preferred browser.			The application will open the website in the browser chosen by the user.

[This space intentionally left blank]

3.9 Customized Container Recommendation Test (O: Rios, M1: Chang-Horvath)

Test Category: User Interfaces - Container Recommendation Screen		Description: This test is to verify that the user gets the information regarding the recommended containers for a particular shopping.		
Test Case: 3.1.2.5		Case Name: Customized Container Recommendation	Version: 1.0	Written By: Gavin Rios
Requirements Fulfilled: 3.1.2.5.1 - 3.1.2.5.5		Purpose: The system shall display a container recommendation screen that allows a shopper to get information about the number and the types of container that can be used for a specific shopping.		
Setup Conditions: 1. The shopper has created an account. 2. The shopper is accessing Refill.Me on an internet-connected device. 3. The shopper is logged in to their user account. 4. The shopper has at least one item in the shopping list. 5. The shopper chose the “Container Recommendation” button on the “Shopping List” page.				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	The user sees a listing showing the number and types of jars needed for specific items, and for the whole shopping in total.			The system displays a listing showing the product name, the amount specified by the user, the number and the type of container that is recommended, the materials that the container

				can be made of, and the volume of the container. On the bottom of the page, the application shows the total number for each container type that is needed for the shopping.
2	The user selects the “Back to My Shopping List” button.			The system navigates the user to the “Shopping List” page.
5	The user adds/removes an item from the shopping list.			The application will update the items in the shopping list.
6	The user chooses the “Container Recommendation” button.			The application will display the “Customized Container Recommendation” page with the updated information regarding the number of types of containers needed for the shopping.

3.10 Reward.Me Test (O: Hull)

Test Category:		Description:		
Algorithms		Testing the functionality of the Reward.Me feature		
Test Case:		Case Name:	Version:	Written By:
3.1.2.6		Reward.Me	1.0	Katelynn Hull
Requirements Fulfilled:		Purpose:		
3.1.2.6.1 – 3.1.2.6.3		The system shall allow shoppers to view and redeem reward points.		
Setup Conditions:				
1. User has created an account.				
2. User has logged into their Refill.Me account.				
3. User has navigated to the Reward.Me feature from the home page.				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	User navigates to the Reward.Me screen.			The user’s total reward points are displayed at the top of the “Reward.Me” page.
2	User clicks the “Point History” button.			The application will display a new page showing a list of the user’s previously awarded points and the date that they were awarded.
3	User clicks the “Back” button.			The application will display the “Reward.Me” screen.

4	User clicks the “Redeem Points” button.			The application will display a new page showing a list of available coupons or discounts that can be redeemed for a certain number of points.
5	User clicks the “Redeem” button next to a coupon/discount that they do not have enough points for.			A message is displayed informing the user that they do not have enough points for that coupon/discount
6	User clicks the “Redeem” button next to a coupon/discount that they have enough points for.			The coupon/discount will be added to the user’s list of coupons and discounts. The value of the coupon/discount will be subtracted from the user’s total reward points.
7	User clicks the “Back” button.			The application will display the “Reward.Me” screen.
8	User clicks the “My Rewards” button.			The application will display a new page showing a list of the user’s current coupons and discounts.
9	User clicks the “Back” button.			The application will display the “Reward.Me” screen.
10	User clicks the “Collect Points” button.			The application will add the points the user received for providing suggested edits to the user’s total points displayed on the “Reward.Me” screen.

3.11 Store Rating Test (O: Colombini, M1:Wasikye)

Test Category:		Description:		
System		Store Rating Functionality		
Test Case: 3.1.2.7		Case Name: Store Rating	Version: 1.0	Written By: Michael Colombini
Requirements Fulfilled: 3.1.2.7.1 – 3.1.2.7.2		Purpose: This test case verifies the functionality of the store rating feature in the system, ensuring that the user can select a store, rate it based on set criteria, submit the rating, and the average rating is calculated and displayed correctly.		
Setup Conditions: 1. The user has a Refill.Me account. 2. The user is logged in to their Refill.Me account. 3. The user has searched for stores to rate using the store search functionality.				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	Select a store from the list of available stores.			The user can select a store from the list of available stores.
2	Rate the store on a numerical scale from 1 to 5, inclusive, based on the provided set of criteria such as quality of service, cleanliness, product variety, and price.			The user can rate the store based on the provided set of criteria.
3	Submit the rating for the selected store.			The submitted rating is saved in the system.
4	Verify that the system displays the average rating for the selected store based on all submitted ratings.			The system displays the average rating for the selected store based on all submitted ratings.

5	Verify that the store owner can view and the feedback provided by users.			The store owner can view the feedback provided by users.
6	Verify that the store owner can respond to the feedback provided by users.			The store owner can respond to feedback given by the users
7	Verify that the rating data is stored securely.			The rating data is stored securely and cannot be accessed or modified by unauthorized users.

[This space intentionally left blank]

3.12 Lemme.Know Test (O: Chang-Horvath)

Test Category: Crowdsourcing: Suggested Edits		Description: This test is to verify that a user can report information about stores, products, and prices.		
Test Case: 3.1.2.8.		Case Name: Lemme.Know	Version: 1.0	Written By: Judit Chang-Horvath
Requirements Fulfilled: 3.1.2.8.1-3.1.2.8.7		Purpose: To verify that the user can submit information about new or closed stores, new or eliminated products, and price changes.		
Setup Conditions: <div>1. User must run the Refill.Me application on a device with Internet connection.</div> <div>2. User must have credentials for an account.</div> <div>3. User must be logged in into the account.</div> <div>4. User must navigate to the Contact Us page of the application.</div>				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	User chooses “New store” from the listing.			The application will open a new page that allows the user to enter information about a new store (store name, store address, opening hours).
2	User enters new store information on the form.			The form is filled with the information entered by the user.
3	User chooses the “Cancel” button.			The application discards the information that was entered and returns to the “Contact Us” page.

4	User chooses the “Submit” button.			The application will store the user added information in the database. The app displays a Thank you message, and a “Back to Contact Us” button.
5	User chooses “Existing store” from the listing.			The application will open a new page that allows the user to choose a store from a drop-down menu.
6	User chooses a store from the drop-down menu.			The application will open a new page that allows the user to enter information about products and price changes related to a specific store (product name, price, whether product is removed from the store’s offerings), or to report that a store temporarily closed.
7	User enters new product and price information on the form.			The form is filled with the information entered by the user.
8	User chooses the “Cancel” button.			The application discards the information that was entered and returns to the “Contact Us” page.

9	User chooses the “Submit” button.			The application will store the user added information in the database. The app displays a Thank you message, and a “Back to Contact Us” button.
10	User chooses “Closed store” from the listing.			The application will open a new page that allows the user to choose a store from a drop-down menu.
11	User chooses a store from the drop-down menu.			The form is filled with the information entered by the user.
12	User chooses the “Cancel” button.			The application discards the store choice that was made by the user and returns to the “Contact Us” page.
13	User chooses the “Submit” button.			The application will store the user’s store choice in the database. The app displays a Thank you message, and a “Back to Contact Us” button.

[This space intentionally left blank]

3.13 Product Upload Test (O: Colombini)

Test Category: System		Description: This test case verifies the functionality of the product upload feature in the system, ensuring that the system allows store owners/employees to update/edit products related to their stores.		
Test Case: 3.1.2.9		Case Name: Product Upload	Version: 1.0	Written By: Michael Colombini
Requirements Fulfilled: 3.1.2.9.1 – 3.1.2.9.3		Purpose: To ensure that the system allows store owners/employees to enter, edit, and delete product information accurately and efficiently.		
Setup Conditions: 1. The user is logged in as a store owner/employee. 2. The user must navigate to the “Products & Prices” page. 3. The user has the necessary product information, including name, product, ID, category, price, and units.				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	Click on the “Product Upload” feature in the system.			The system has enabled the user to navigate to the Product upload page.
2	Enter the name of the new product in the “Name” field. Enter the product ID in the “Product ID” field. Select the appropriate category from the dropdown menu. Enter the price of the product in the “Price” field. Select the appropriate unit form the dropdown menu. Then click on the “Save” button.			The product information is entered accurately and completely.

3	Verify that the new product is successfully added to the store's product list.			The new product was successfully added to the store's product list.
4	Click on the "Edit" button next to the newly added product. Modify the product information (e.g., change the price or unit). Click on the "Save" button.			The product information is modified successfully.
5	Click on the "Delete" button next to the newly added product. Confirm the deletion, verify that the product is removed from the store's product list.			The product is removed from the store's product list.

[This space intentionally left blank]

3.14 Reward Point Conversion Test (O: Colombini, M1: Chang-Horvath)

Test Category: System		Description: Verify that the store owners/employees can add/modify/delete entries in the Reward Point Conversion dashboard.		
Test Case: 3.1.2.11		Case Name: Reward Point Conversion	Version: 1.0	Written By: Michael Colombini
Requirements Fulfilled: 3.1.2.11.1 – 3.1.2.11.3		Purpose: To ensure that the system allows store owners/employees to add/modify/delete entries in the Reward Point Conversion dashboard in order to communicate the rules of reward point redemption to the shoppers.		
Setup Conditions: <div>1. The system is running and accessible to the store owners/employees.</div> <div>2. The store owner/employee has a valid login account and has been granted permission to add/edit/delete reward point conversions.</div> <div>3. The store owner/employee has the necessary information such as product names, reward points needed, and expiry dates for the reward point conversions being modified.</div> <div>4. The reward point system is already set up and operational.</div> <div>5. Store owner/employee navigated to the “Reward Point Conversion” dashboard.</div>				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	The user chooses the “Add Conversion” button.			The application will open a pop-up window with a form where the user can enter the name/identifier of the conversion, the number of reward points, and the expiry date.

2	The user chooses the “Save” button.			The application will add the new reward point conversion entry to the dashboard. The application will return to the “Reward Point Conversion” page.
3	The user chooses the “Edit” button next to an entry in the Reward Point Conversion dashboard.			The application will open a pop-up window where the user can modify the name, number of reward points needed, and/or the expiry date.
4	The user chooses the “Save” button.			The application will add the new reward point conversion entry to the dashboard. The application will return to the “Reward Point Conversion” page.
5	The user chooses the “Delete” button next to an entry in the Reward Point Conversion dashboard.			The application will remove the reward point conversion entry from the dashboard. The application will return to the “Reward Point Conversion” page.

[This space intentionally left blank]

3.15 Profile Tests (*O: Chang-Horvath*)

3.15.1 Shopper Profile Test (*O: Chang-Horvath*)

Test Category: Profile		Description: This test is to verify that a user (a shopper) can access the profile page, and change the settings.		
Test Case: 3.1.1.2.10		Case Name: Profile	Version: 1.0	Written By: Judit Chang-Horvath
Requirements Fulfilled: 3.1.1.2.10.1 - 3.1.1.2.10.11		Purpose: To verify that the user (a shopper) can view and modify information on the profile page, make changes to the settings, log out, or delete the account.		
Setup Conditions: <div>1. User must run the Refill.Me application on a device with Internet connection.</div> <div>2. User must have credentials for an account.</div> <div>3. User must be logged in into the account.</div> <div>4. User must navigate to the Profile page of the application.</div>				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	User chooses the photo camera icon in the main header of the Profile page.			The application will open a pop-up window that allows the user to upload a photo from their photo album or to take a new photo.
2	User chooses a photo from the photo album or takes a new photo.			The application will save the new photo as the profile picture for the user. The application will return to the Profile page, and it will replace the profile icon with the chosen photo.

3	User chooses the “Saved stores” button.			The application will open a new page that lists all the stores that were saved by the user.
4	User chooses the Profile button at the bottom of the page.			The application will return to the Profile page.
5	User chooses the “Share with friends” button.			The application will open a new page that lists the following sharing options: Email, Facebook, Twitter.
6	User chooses Email, Facebook, or Twitter as the sharing method.			The application will open the default email application, Facebook, or Twitter, and it will allow the user to share information about Refill.Me.
7	User navigates back to the Profile page.			The application will open the Profile page.
8	User chooses “Profile settings” from the listing.			The application will open a new page that allows the user to change the following personal information: name, email, password.
9	User changes the name, email, or password.			The application will save the changes indicating it with a check mark, and it will return to the “Profile settings” page.

10	User chooses “Log out” button at the bottom of the page.			The application will log out the user, and it will display the Login page.
11	User chooses the “Delete account” button at the bottom of the page.			The application will delete the user’s account information from the database, and display the Login page.
12	User attempts to log in to the application using the old credentials.			The application will not allow the user to log in using credentials that are not in the database.

[This space intentionally left blank]

3.15.2 Store Owner Profile Test (O: Chang-Horvath, M1: Hull)

Test Category: Profile		Description: This test is to verify that a user (a store owner) can access the profile page, and change the settings.		
Test Case: 3.1.1.3.8		Case Name: Profile	Version: 1.0	Written By: Judit Chang-Horvath
Requirements Fulfilled: 3.1.1.3.8.1 - 3.1.1.3.8.13		Purpose: To verify that the user (a store owner) can view and modify information on the profile page, make changes to the settings, log out, or delete the account.		
Setup Conditions: 1. The user must run the Refill.Me application on a device with Internet connection. 2. The user must have credentials for an account. 3. The user must be logged in into the account. 4. The user must navigate to the Profile page of the application.				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	User chooses the photo camera icon in the main header of the Profile page.			The application will open a pop-up window that allows the user to upload a photo from their photo album or to take a new photo.
2	User chooses a photo from the photo album or takes a new photo.			The application will save the new photo as the profile picture for the user. The application will return to the Profile page, and it will replace the profile icon with the chosen photo.

3	User chooses the “Edit” icon on the “Store information” panel.			The application will open a pop-up window, and it will allow the user to edit the store’s address, opening hours, website url, and phone number.
4	User saves the newly entered information in the pop-up window by tapping on the “Save” button			The application will update the information in the database. The application will return to the “Profile” page, and it will display the updated store information.
5	User chooses “Profile settings” from the listing.			The application will open a new page that allows the user to change the following store information: name, email (used for login), password.
6	User changes the name, email (used for login), or password.			The application will save the changes indicating it with a check mark. The application will return to the “Profile settings” page.
7	User chooses the “Logout” button at the bottom of the page.			The application will log out the user, and it will display the Login page.
8	User chooses the Profile button at the bottom of the page.			The application will return to the Profile page.

9	User chooses the “Notifications settings” button on the Profile page.			The application will open a new page with options describing the different notifications settings.
10	User modifies the settings for one or more notification options.			The application will save the users notification settings, and it will return to the Profile page.
13	User chooses the “Roles & Permissions” button on the Profile page.			The application will open a new window listing the employees of the store who have access to Refill.Me and their permissions with regard to the different functions that can be performed in Refill.Me (for example: product upload, store information update, etc.).
14	User adds or removes employees from the “Roles & Permissions” listing.			The application will update the “Roles & Permissions” listing.
15	User adds or removes permissions to/from specific employees in the “Roles & Permissions” listing.			The application will update the “Roles & Permissions” listing.
16	User chooses the Profile button at the bottom of the page.			The application will return to the Profile page.
17	User chooses “Logout” button at the bottom of the page.			The application will log out the user, and it will display the Login page.

18	User chooses the “Delete account” button at the bottom of the page.			The application will delete the user’s account information from the database, and display the Login page.
19	User attempts to log in to the application using the old credentials.			The application will not allow the user to log in using credentials that are not in the database.

[This space intentionally left blank]

3.16 Navigation Menu Tests (O: Maximova)**3.16.1 Navigation Menu for the Shopper (O: Maximova)**

Test Category: Navigation menu for Shopper		Description: The test case covers how the Shopper would use Refill.Me app on the phone. The case covers: - login into created account, - select Home to see all the navigation paths - select Store Search - select Find Product - select Shopping List - select Profile		
Test Case: 1.0		Case Name: NavMenu for Shopper	Version: 0.1	Written By: Kira Maximova
Requirements Fulfilled: 3.1.1.1.2 – 3.1.1.1.2.8; 3.1.1.2 – 3.1.1.2.1.8.5; 3.1.1.2.2 – 3.1.1.2.2.3.3.; 3.1.1.2.3 – 3.1.1.2.3.4; 3.1.1.2.5 – 3.1.1.2.5.1; 3.1.1.10 – 3.1.1.2.10.4; Additionally briefly mentions: 3.1.1.2.4; 3.1.2.6; 3.1.1.2.8; 3.1.2.1.1.1.3 – 3.1.2.1.1.1.3.2.4;		Purpose: To confirm that the NavMenu works and that the Shopper’ is able to navigate to all the functions within the NavMenu (minimum testing for functions)		
Setup Conditions: 1. The user has a mobile device that has at least half the battery charge. 2. The user successfully installed Refill.Me app on the mobile device. 3. The user has a basic understanding of the app and is able to follow the provided instructions. 4. The user has a stable internet connection to execute the test case.				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	Using the mobile device (Android/iOS), launch Refill.Me app			Welcome screen opens

2	Perform a quick UI validation to make sure : Logo, name of the app, and Login / Register option are available.			Validated, all fields are present. No anomalies found.
3	Enter email & Password: Email: 00validemail@gmail.com Password: Johny-1334! Select 'Submit' Spinner appears during the authentication process			Entered: 00validemail@gmail.com in email field Johny-1334! as password ==== Validated
4	The error appears to check 'Please check login credentials!'			Validated, error listed
5	Correct the password: Johny-1234! Select 'Submit' Spinner appears			Entered: email prefilled: 00validemail@gmail.com ; Johny-1234! ===== Validated
6	Refill.Me app now had a navMenu with five (5) options; and page has seven (7) options. NavMenu white text on deep navy color; current button appears grey Options on page: alters between white and grey, horizontal buttons			All validated, no visual anomalies
7	<u>NavMenu options</u> : Home, Store Search, Product Search, Shopping List, Profile <u>Seven options</u> : Store Search, Product Search, Container Guide, Shopping List, Reward.Me, Advocacy and Petition Sites, Lemme.Know.			Some fields are partially loaded

8	Refresh the current [Home] page – should fix any time out or rendering related issues			All good, no visual issues, everything is loaded and appearing as described
9	Perform spot check (random) two or three out of seven links ex. 1. select 'Container Guide' → wait till its finish loading to see the loaded page → select 'Home' on NavMenu 2. select 'Reward.Me' → wait till its finish loading to see the loaded page → select 'Home' on NavMenu 3. select 'Advocacy & Petition Sites' → wait till its finish loading to see the loaded page → select 'Home' on NavMenu			All paths worked, pages loaded
10	Select on 'Store Search' on NavMenu			Store Search map opens with a modal (assumption it's a first time)
11	Select on 'Home' on NavMenu			Home page with 7 button opens
12	Select on 'Store Search' white button on the screen to validate the path for the button			Store Search map opens with a modal
13	Validate the following fields on the page: 1) Logo 2) Name: Refill.Me 3) Background image 4) a grey modal (popup) with 'Allow "Refill.Me" to use your location?" 5) Sentence: 'Your location will be used to display nearby package-free store			Validated all fields as described and present on the page

14	6) 3 options: Allow Once, Allow While Using App, Don't allow 7) NavMenu: Home (navy inactive), Store Search (grey active), Product Search (navy inactive), Shopping List (navy inactive), Profile (navy inactive)			Validated all fields as described and present on the page
15	Select 'Allow While Using App'			Modal (popup) disappears; Map loads with icons to select for [My] area
16	Selecting on any location icon, popup opens navy color with white letters: Name of the store, Address, Grey button: 'Details' (letter in navy color)			Validated
17	Select on 'Product Search' on NavMenu			Product Search page loads
18	Validate: 1) Name: Product Search, navy letters 2) Two Navigation in White 3) "Search by Icons" 4) Sentence: Save time, and get a quick overview of products that are available package-free 5) "Enter Product Name" 6) Sentence: You know exactly what you are looking for? Please just type it in here!			Validated all fields as described and present on the page
19	Select 'Enter Product Name'			Product Search page opens
20	Enter a random: ex.: !e3bc			An error appears, to type the real product
21	Enter: apple			Apple results in [My] area loads

22	Select on 'Shopping List' on NavMenu			Shopping page opens
23	Validate following fields on the page: 1) Name: My Shopping List, navy text on grey background 2) store name where the item can be found (if applicable)			Validated; #2 wasn't applicable as no items added
24	Validate the message: 'Shopping List empty'			Validated
25	Select on 'Profile' on NavMenu			Profile page opens
26	Validate following fields on the page: on the page: 1) Name on the account: Johny Smith 2) Image of the dog 3) three buttons navy on white background: Saved Stores, Share with Friends, Profile Settings			Validated
27	Validate two buttons: Log Out – navy text on grey background Delete account – white text on navy background			Buttons present - validated
28	Validate the NavMenu: has Profile on grey background as active, others inactive and on navy background			Validated

[This space intentionally left blank]

3.16.2 Navigation Menu for the Store Owner (*O: Maximova*)

Test Category: Navigation menu for Store Owner	Description: The test case covers how the Store Owner would use Refill.Me app on the tablet. The case covers: <ul style="list-style-type: none">- login into created account,- select Home to see all the navigation paths- select Product & Prices- select Reward Point Conversion- Select Store Analytics- Select Profile		
Test Case: 1.0	Case Name: NavMenu for Store Owner	Version: 0.1	Written By: Kira Maximova

Requirements Fulfilled: 3.1.1.1.2 – 3.1.1.1.2.8; 3.1.1.3.1 - 3.1.1.3.1.12; 3.1.1.3.2 – 3.1.1.3.2.5; 3.1.1.3.3 – 3.1.1.3.3.5; 3.1.1.3.4 – 3.1.1.3.4.3.5; 3.1.1.3.8 – 3.1.1.3.8.13; Additionally briefly mentions: 3.1.1.1.2; 3.1.1.3.6.1; 3.1.1.3.7; 3.1.1.3.5;		Purpose: To confirm that the NavMenu works and that the Store’ owner is able to navigate to all the functions within the NavMenu (minimum testing for functions)		
Setup Conditions: 1. The user has a tablet that has at least half the battery charge. 2. The user successfully installed Refill.Me app on the tablet device. 3. The user has a basic understanding of the app and is able to follow the provided instructions. 4. The user has a stable internet connection to execute the test case.				
Test Case Activity		Pass/Fail	Comments	Expected Result
1	Using the tablet (Android/iOS/Windows); launch Refill.Me app			Welcome screen opens
2	Perform a quick UI validation to make sure : Logo, name of the app, and Login / Register option are available.			Validated, all fields are present. No anomalies found.
3	Enter email & Password: Email: 12345@gmail.com Password: Test-1334! Select ‘Submit’ Spinner appears during the authentication process			Entered: 12345@gmail.com in email field Test-1334! as password ===== Validated
4	The error appears to check ‘Please check login credentials!’			Validated, error listed
5	Correct the password: Test-1234! Select ‘Submit’ Spinner appears			Entered: email prefilled: 12345@gmail.com Test-1234!

				=====
				Validated
6	<p>Refill.Me app now had a navMenu with five (5) options; and page has seven (7) options.</p> <p>NavMenu Home, white text on deep navy color; current button appears gray</p> <p>Options on page: gray buttons set in two columns last option is centered</p>			All validated, no visual anomalies
7	<p><u>NavMenu options:</u> Home, Product & Prices, Reward Point Conversion, Store Analytics, Profile</p> <p><u>Seven options:</u> Product & Prices, Store Analytics, Reward Point Conversion, Store Search, Store Rating, Product Search, Profile.</p>			Some fields are partially loaded
8	Refresh the current [Home] page – should fix any time out or rendering related issues			All good, no visual issues, everything is loaded and appearing as described
9	<p>Perform spot check (random) two or three out of seven links</p> <p>ex.</p> <ol style="list-style-type: none"> 1. select 'Store Search' → wait till its finish loading to see the loaded page → select 'Home' on NavMenu 2. select 'Product Search' → wait till its finish loading to see the loaded page → select 'Home' on NavMenu 3. select 'Store Rating' → wait till its finish loading to see the loaded page → select 'Home' on NavMenu 			All paths worked, pages loaded
10	Select on 'Product & Prices' on NavMenu			Product & Prices page opens with a modal

11	Select on 'Home' on NavMenu			Home page with seven (7) buttons open
12	Select on 'Product & Prices' grey button on the screen to validate the path for the button			Product & Prices opens
13	Validate the following fields on the page: 1) Name: Product & Prices 2) Background image 3) a white button 'Add product' navy font on white background 4) greyish Table with 6 cells (font navy)			Validated all fields as described and present on the page
14	5) Validate the name of the cells: - Name (product name) - Product ID (a number with 4 digits) - Category (list the product category: fruit, vegetable, etc.) - Price (US dollars price) - Unit (US unit system: lbs./oz) - Action - Checkbox icon – to edit the record - Trash icon – to delete			Validated all fields as described and present on the page
15	Select on "Reward Point Conversion" on NavMenu			Reward Point Conversion page opens
16	Validate the following fields: 1) Title: Reward Point Conversion (white on background image) 2) Background image 3) Button: white 'Add conversion', navy font on white background 4) Table with 4 cells: 5) Cells name: Name, Reward Points, Expire Date, Action 6) Action to have two icons: checkbox for edit & trashcan icon to delete the record 7) 'Reward Point Conversion' button active (grey) in the navigation menu, other buttons inactive (navy)			Validated all fields as described and present on the page

17	Select on 'Add Conversion' button			A new record loads where the Store Owner can add additional Point Conversion Record
18	Select 'Store Analytics' in the Navigation menu			Store Analytics page opens
19	Validate the following fields on the page: 1) Title (white on background image): Store Analytics 2) Table with the first row – 1 cell 3) Navy font on grey table 4) Sentence in first row: "Recent Searches in Your Area" 5) Table with 5 cells: 6) Cell' name: Product Search, Product Found, Date, Product Available in My Store, Action;			All fields validated; all as described; no anomalies
20	Validate the following fields on the page (continue): 7) 'Product Found' cell lists icons: checkbox or cancel, 'X' 8) Date in the following format: MM/DD/YYYY 9) 'Product Available in My Store' cell lists icons: checkbox or cancel 'X' 10) Action cell: two buttons: 'Save' or 'Saved' 11) 'Save' button white button, navy text 12) 'Saved' button is navy button, white text 13) Two buttons: white buttons, navy text: 'See Full Listing' & 'See Saved Searches' 14) Store Analytics navigation button is grey and active, others are inactive and in navy color			Validated all fields as described and present on the page
21	Select on any 'Save' button listed			White 'Save' button now turned into navy 'Saved' button

22	Select on 'See Full Listing'			A list of all the products listed for the store displays Note: a list, without any pagination
23	Select on 'Profile' in the Navigation Menu			Profile page opens
24	Validate following fields on the page: on the page: 1) Name on the account: Jake Smith 2) Image of Safeway logo 3) Name of the store: Safeway			Validated
25	Validate following fields on the page: Store Information: 4) List the address for the store: 1624 Belle View Blvd, Alexandria, VA 22307 5) Hours of operations (blank) 6) website: safeway.com 7) Phone number: (703) 763-3434			All field validated
26	Validate following fields on the page the right side: 1) presence of three buttons navy text on white 2) Name: Profile Settings, Notification settings, Role & Permissions 3) Smaller button 'Log out' grey button, navy text 4) White texted hyperlink 'Delete account' 5) Profile button in the navigation grey and active, while: Home, Product & Prices, Reward Point Conversion and Store Analytics inactive and navy			All as described; no anomalies found
27	Select on 'Edit' to update the Store information			Store Information is now editable

28	Add the Hours for the store: Attempting to add all days, all 24 hrs Monday – Sunday: from 12:01 am to 12:00 am Select ‘Submit’ button to save changes			An error appears: ‘Please correct the format’
29	Correcting the hours: All Monday – Sunday: 12:01 am to 11:59 pm Select ‘Submit’ button to save changes			Hours submitted
30	Correct the phone for the store: From: (703) 763-3434 To: (703) 765-3434 Select ‘Submit’ button to save changes			Phone updated

4. Traceability Matrix (*O: Hull, M1: Chang-Horvath*)

	Test Cases							
Requirements	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8
3.1.1.1.1								
3.1.1.2.2								
3.1.1.2.1								
3.1.1.2.2								
3.1.1.2.3								
3.1.1.2.4								
3.1.1.2.5								
3.1.1.2.6								
3.1.1.2.7								

3.1.1.2.8								
3.1.1.2.9								
3.1.1.2.10								
3.1.1.3.1								
3.1.1.3.2								
3.1.1.3.3								
3.1.1.3.5								
3.1.1.3.6								
3.1.1.3.7								
3.1.1.3.7								
3.1.1.3.8								
3.1.2.1								
3.1.2.2								
3.1.2.3								
3.1.2.4								
3.1.2.5								
3.1.2.6								
3.1.2.7								
3.1.2.8								
3.1.2.9								
3.1.2.10								
3.1.2.11								

	Test Cases
--	------------

Requirements	3.9	3.10	3.11	3.12	3.13	3.14	3.15	3.16
3.1.1.1.1								
3.1.1.2.2								
3.1.1.2.1								
3.1.1.2.2								
3.1.1.2.3								
3.1.1.2.4								
3.1.1.2.5								
3.1.1.2.6								
3.1.1.2.7								
3.1.1.2.8								
3.1.1.2.9								
3.1.1.2.10								
3.1.1.3.1								
3.1.1.3.2								
3.1.1.3.3								
3.1.1.3.4								
3.1.1.3.5								
3.1.1.3.6								
3.1.1.3.7								
3.1.1.3.7								
3.1.1.3.8								
3.1.2.1								
3.1.2.2								
3.1.2.3								

3.1.2.4								
3.1.2.5								
3.1.2.6								
3.1.2.7								
3.1.2.8								
3.1.2.9								
3.1.2.10								
3.1.2.11								