

# Front End Test Report

By: Aaron Huggins and Bryar Frank

## Overview

The frontend testing suite provides large-scale testing of the frontend application deployed at

<http://ahuggins-warehousemanager-frontend.s3-website.us-east-2.amazonaws.com/>

And goes through several processes such as navigation, data manipulation, and usability. It does this by utilizing Selenium and Cucumber to orchestrate a POM architecture matching the site's, allowing for the extensible >100 tests to be run.

## Page Object Model (POM)

The POM of the frontend testing suite provides a unique concrete page implementation for:

- Landing page: Page,
- Signup page: FormPage,
- Login page: FormPage,
- Home page: Page,
- Account page: ObjectPage,
- Warehouses page: ObjectPage,
- Items page: ObjectPage.

Which each derive from a hierarchy of interfaces and abstract classes that assist in the implementation of the Facade pattern used in the Step Definitions. These interfaces are as follows:

- IPage: The base interface for all pages, declaring methods that allow for page navigation, login and verification, and button clicks.

- IFormPage: This interface declares methods that allow for the interaction with a form that is on a given page, and the verification that the form gave a proper response.
- IObjectPage: This interface declares methods that allow for interaction with a data model, and the verification that such interactions are providing a proper response.

In addition to these interfaces are the following abstract classes, which provide basic functionality for the concrete pages:

- Page
- FormPage
- ObjectPage

By utilizing the concrete page objects, the test cases created using Cucumber feature files are able to seamlessly integrate with Selenium by instantiating pages, using their navigational capabilities, and then using their interactional capabilities.

## Tests

### Acceptance Criteria

The test suite ran 116 tests with a 79.7% pass rate. Individual acceptance criteria test pass rates below.

Acceptance Criteria Identifier	Pass/Fail	Success Rate (Outlines only)
A1	Pass	-
A2	Pass	-
A3	Pass	-
A4	Pass	-
A5	Fail	-
A6	Pass	-

A7	Pass	-
A8	Pass	-
A9	Pass	-
W1	Pass	-
W2	Pass	-
W3	Pass	-
W4	Pass	-
W5	Pass	-
W6	Pass	-
I1	Pass	-
I2	Pass	-
I3	Pass	-
I4	Pass	-
I5	Pass	-
I6	Pass	-
I7	Pass	-
I9	Pass	-
I10	Pass	-
U2 [Outline]	Pass	100% - 24/24
U3 [Outline]	Pass	100% - 19/19
U4 [Outline]	Pass	100% - 4/4
U5 [Outline]	Pass	100% - 5/5
N1 [Outline]	Fail	45.2% - 19/42

## Gherkin Feature Files

In order to compile all frontend-designated acceptance criteria, the following 5 feature files were created, separating the acceptance criteria based on user story:

- AdministratorBehaviors
- DirectNavigation
- InventoryBehaviors
- Usability
- WarehouseBehaviors

These feature files used common step language, resulting in the creation of the following steps.

### Given Steps

- Given I Am On “<page>”
- Given I Am Logged “<out/in>”
- Given I Am Performing “<action>”
- Given I Enter “<correct/incorrect>” Information
- Given I “<correct/incorrect>” Update “<model>”

### When Steps

- When I Attempt To Directly Navigate To “<page>”
- When I Attempt To Navigate To “<page>”
- When I Click “<button>”
- When I Submit The Form

### Then Steps

- Then I Will Be Performing “<action>”
- Then I Am Taken To “<page>”
- Then A New “<model>” Is Created
- Then I See An Error Message
- Then I “<can/can't>” See “<model>” Information

- Then “<model>” Fields Have “<been/not been>” Changed
- Then “<model>” No Longer Exists
- Then I “<am/am not>” Logged In

## Step Definitions

Due to the simplification and standardization of the feature steps, only a single Step Definitions file was required to act as a test running engine. Within, it contains a current page object which keeps track of the state and performs necessary functions as described in the POM section.

## Notes on Failed Tests

Acceptance Criteria Identifier	Failure Note
A5	The application fails to prevent invalid user credential updates, allowing empty passwords to become the stored passwords for users. Once this happens, any future attempt to log in is impossible due to the frontend login page’s correct protection against empty fields. Only a manual reset using the API directly can allow the user to regain access to their account.
N1	The application does not maintain user state upon refresh or direct navigation. Due to this, all test cases for N1 that directly navigate to logged-in-only pages have failed.

## Other Important Notes

Note Identifier	Description
N-A5	<p>The application fails to prevent invalid user credential updates, allowing empty passwords to become the stored passwords for users. Once this happens, any future attempt to log in is impossible due to the frontend login page’s correct protection against empty fields. Only a manual reset using the API directly can allow the user to regain access to their account.</p> <p>Tests including the account update will be omitted until this is fixed on both the frontend and backend implementations.</p>
N-A8	Upon manual testing, this was found to be working as intended, indicating a

	defect in the testing suite. Further investigation required.
N-W1	Upon manual testing, this was found to be working as intended, indicating a defect in the testing suite. Further investigation required.
N-W4	This test failed due to a failure to log in, indicating a defect in the testing suite either in the login process tests or in this test. Further investigation required.
N-W6	Upon manual testing, this was found to be working as intended, indicating a defect in the testing suite. Further investigation required.
N-I5	Upon manual testing, this was found to be working as intended, indicating a defect in the testing suite. Further investigation required.
N-I7	Upon manual testing, this was found to be working as intended, indicating a defect in the testing suite. Further investigation required.
N-U4	<p>Multiple tests within this outline failed, due to the following reasons:</p> <ul style="list-style-type: none"> <li>• Discovered defect in the testing suite where “Account Delete” action is instead named “Deleting Account” inside of the verification method.</li> <li>• Defects in the expected XPath paths within the testing suite.</li> </ul> <p>Further investigation required.</p>
N-U5	<p>Multiple tests within this outline failed, due to the following reasons:</p> <ul style="list-style-type: none"> <li>• Discovered defect in the testing suite where “Account Delete” action is instead named “Deleting Account” inside of the verification method.</li> <li>• A defect in the testing suite where the add warehouse form is not recognized.</li> <li>• Defects in the expected XPath paths within the testing suite.</li> </ul> <p>Further investigation required.</p>
N1	Due to the overall lack of descriptors in the frontend code, the maintainability of the project is severely low. While not a code failure, it is highly recommended to increase the number of descriptors such as tag IDs, classes, and names.

# Future Testing

Due to the large scale of the current testing suite, modifications to the overall design will not be necessary until the frontend design is changed. However, it can be improved upon for scalability in the following ways:

- Further abstraction of the page components into individual classes, such as with forms that open and close.
- Fixes and improvements on existing tests so that failures in the testing suite are more limited.