WeSki

Search Flow Regression

STD

# Revision History:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Change** | **Version** | **Remarks** | **Author** |
| 8/4/24 | Initial draft | 1.0.0 |  | Tomer Granit |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Environments**:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Release** | **Version** | **OS** | **Mobile** | **Browser** | **Localization/Lang** | **Remarks** |
| 2.5.24 |  | Win |  | Chrome | US/EN |  |
| 2.5.24 |  | Win |  | FireFox | GB/EN |  |
| 2.5.24 | 1.24.12 |  | Iphone |  | IL/HE | App Installation |
| 2.5.24 |  |  | Android | Chrome |  |  |

## Defect reporting:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Release** | **JIRA#** | **Remarks** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of content:

[Revision History: 1](#_Toc163500150)

[Environments 1](#_Toc163500151)

[Defect reporting: 1](#_Toc163500152)

[1. Test Objectives: 2](#_Toc163500153)

[To verify the functionality and reliability of WeSki's ski trip search flow by conducting a comprehensive regression test. This test will cover the complete flow from searching for a trip to selecting a desired package, ensuring that all features work as intended, including handling of edge cases and different scenarios. 2](#_Toc163500154)

[2. Test Scope: 2](#_Toc163500155)

[3. Pre-Requites: 2](#_Toc163500156)

[4. Search UI 3](#_Toc163500157)

[4.1 UI Texts 3](#_Toc163500158)

[4.2 UI DropDown Icons and Place holder / defaults 3](#_Toc163500159)

[4.3 UI Search button 4](#_Toc163500160)

[5. Search Dropdown 4](#_Toc163500161)

[Stopping to full test and I want to list concerns and scenarios: 5](#_Toc163500162)

[6. Appendix A: 6](#_Toc163500163)

## Test Objectives:

## To verify the functionality and reliability of WeSki's ski trip search flow by conducting a comprehensive regression test. This test will cover the complete flow from searching for a trip to selecting a desired package, ensuring that all features work as intended, including handling of edge cases and different scenarios.

## Test Scope:

* Search functionality
* Filtering options
* Sorting options
* Display of search results
* Package details
* Booking process

## Pre-Requites:

The regression tests will be conducted in a dev environment,

Base URL: <https://dev.weski.co.uk/?set_origin=uk>

Version deployed: XXX

Reverify previous regression defects: #1924, # 1952, #1833

## Search UI

## 4.1 UI Texts

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **T** | **Expected Results** | **Pass/Fail** |
|  | Base URL | <https://dev.weski.co.uk/?set_origin=uk>  available |  |
|  | Title:  Unlimited combinations. Unbeatable prices. | Title is available:   1. 2 lines 2. Each line ends with a “.” 3. Test in bold |  |
|  | Leaving title:  LEAVING FROM | Title is available:  Capital letters |  |
|  | Going title:  GOINT TO | Title is available:  Capital letters |  |
|  | Date title:  DATES | Title is available:  Capital letters |  |
|  | Guests title:  GUESTS | Title is available:  Capital letters |  |
|  | Searching title:  SEARCHING FOR | Title is available:  Capital letters |  |

## 4.2 UI DropDown Icons and Place holder / defaults

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Test** | **Expected Results** | **Pass/Fail** |
|  | Leaving:  Icon: Plain – up  Holder – Any London Airport |  |  |
|  | Going to:  Icon: Double Mountains  Holder - Tignes |  |  |
|  | Dates:  Icon: Calendar  Holder: Dec 2 – Dec 9 |  |  |
|  | Dates:  Icon: 2 people  Holder: 2 adults |  |  |
|  | Searching For:  Icon 1: Bed  Icon 2: Plain up  Icon 3: Transfers |  |  |

## 4.3 UI Search button

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Test** | **Expected Results** | **Pass/Fail** |
|  | Button:  Test: Find Your Trip  Color: Blue  Icon: Magnifier |  |  |

## Search Dropdown

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Test** | **Expected Results** | **Pass/Fail** |
|  | Leaving Dropdown 1. Click on ‘Any London Airport’  2. Other than default value the list should contain pairs of airport and abbreviations  3. Make a selection | 1. Dropdown opened 2. Correct format available 3. Field updated according to selection |  |
|  | Leaving Dropdown - data:  According with test localization compare with Appendix A (Leaving list) that all airports are available | 1. All airports are available 2. No spelling mistakes 3. Abbreviations are correct |  |
|  | Going to Dropdown   1. Click on ‘Tignes’ 2. According with test localization compare with Appendix A (Going list) that all countries and sites are available 3. Make a selection | 1. All Countries are available 2. The correct flag is assigned to country 3. Correct sites are assigned to each country 4. No spelling mistakes 5. Field updated according to selection |  |
|  | Guests Dropdown adults   1. Click on ‘2 adults; 2. Adults have a default of 2 3. Children has a default of 0 4. Use arrow to remove 2 adults 5. Use arrow to add 1 adult 6. Remove children 7. Add children 8. Click on Add age 9. Options between <1 and 1 - 17 10. Remove children 11. Click on Done | 1. Drop page opened      1. 2 adults as default 2. Children has 0 as default 3. Can not have 0 adults, 1 is the minimum 4. Can add one adult, numbers change accordingly 5. Nothing happened 6. Number of children change   Title: CHILD’S AGE UPON RETURN   1. Dropdown list opened 2. Options are <1 and 1 – 17 3. The menu closed and the Guests number changed according to selection |  |

## Stopping to full test and I want to list concerns and scenarios:

**Search Module:**

* Verify that users can search for ski trips by entering different origin and destination airports.
* Ensure the search results display relevant trip options based on the user's search criteria.
* Test the filtering and sorting functionality within the search results.
* Validate the handling of edge cases, such as no results, too many results, or invalid search inputs.
* Ensure the search module is responsive and accessible across different devices and screen sizes.

**Dates Module:**

* The Dates module has a lot of functionality and should have a separate test suite to cover it.
* Verify the user's ability to select different travel dates and durations.
* Test the handling of date-related edge cases, such as unavailable dates, minimum/maximum stay requirements, and date conflicts.
* Ensure the date selection functionality is intuitive and user-friendly.

**Guests Module:**

* Verify the user's ability to select the number of guests (adults, children, and infants).
* Ensure the guests selection is synchronized with the search results and pricing.
* Test the handling of guest-related edge cases, such as minimum/maximum guest requirements and invalid guest combinations.

**API Endpoint Testing:**

To understand how to fully cover the search, I should create a matrix that covers all API endpoints.

For example, when we submit the form, a different endpoint is used if we choose a different origin airport or a different destination. To make sure we test all available options in the submit form, I first want to ensure that all endpoints are covered.

* Create a matrix that covers all API endpoints used in the search flow.
* Ensure that different combinations of origin airport, destination, and other search parameters trigger the correct API endpoints.
* Validate the responses from each API endpoint, including successful responses, error responses, and edge cases.

**Assertion and Scenario Testing:**

Having a set of scenarios that will be the base line for all regression tests giving us the confidence that those scenarios are not broken.

* Create built-in scenarios to cover situations such as no results, too many results (what is kept, what is cut), and a result from each type (e.g., one with only stay and ski pass, one with stay, flights, and ski pass).
* Validate the search results and ensure they match the user's expectations based on the selected criteria.

**Login and Session Management:**

As this is a B2C product, localization, zones, and language are highly important and should be thoroughly tested.

Verify the search flow functionality for both logged-in and non-logged-in users.

Verify search flow while navigating to different pages forward and back.

Test the handling of login, logout, session expiry, and cookie management.

**Non-Functional testing:**

**Performance:**

A part of regression testing is making sure the new version is still within agreed spec, in terms of Performance. Conduct load testing to ensure the search flow can handle high traffic and concurrent user sessions without performance degradation.

Measure the response times for various actions within the search flow, such as submitting the search form, loading the search results, and applying filters.

**Usability:**

Usually when doing a big UI change, then we can do A/B testing, evaluate the overall user experience and ease of use of the search flow.

Many tools are available to gather more information on user behavior, I just wanted to quickly mention usability. (That is not part of a regression).

## Appendix A: