

Test Strategy Document

Feature : eBay Best Seller Related Products

Name : Pramoda Piyumali Gunasekara

Position : Software Quality Engineer

Date : 27.02.2026

1.Introduction

1.1 Purpose

The purpose of this feature is to enhance the user experience by displaying related products as best sellers on the Main product page. These products should always be within a specific price range and related to the same category.

2.Scope

2.1 In Scope:

The following areas will be covered within this testing scope

Functional Testing

- Related products display validation
- Price range validation (50% - 500%)
- Product Categories validation
- Maximum Products count validation (Max 6)
- Duplicate products validation

Boundary Value Testing

Based on the assumptions:

Main Product Price = LKR 1000

Valid range:

- Minimum = LKR 500 (50%)
- Maximum = LKR 5000 (500%)

Boundary tests:

- 499 → invalid
- 500 → valid
- 5000 → valid
- 5001 → invalid

UI/UX Testing

- Related product images
- Related product names
- Related product prices
- Responsiveness
- Layout alignments

Compatibility Testing

- Testing across Chrome, Edge, Firefox, web browsers
- Testing across mobile OS (iOS and Android)

Error Handling

- Out of stock scenarios
- No matching products scenario
- API failures

2.2 Out of Scope:

- Checkout (Cart) functionalities
- Payment Gateway
- Database testing

3 Testing Objectives

1. Validate the price range calculation (50% to 500% of the main product price) works correctly
2. Validate "Same Category" matching logic (not considering color or material variations)
3. Verify shipping costs are not included in the price range comparison
4. Verify maximum related products count behavior (1-6 products)
5. Verify out of stock products exclusion from the related products section
6. Verify consistent behavior for logged in users and guest users
7. Verify system stability during failure scenarios
8. Verify UI/UX

4 Assumptions

ID.	Question	Assumption
CL_1	How is the price range defined? Is it a percentage or an amount?	Percentage based: 50% to 500% of the main product price
CL_2	Does it need to include the shipping cost for price range comparison?	Shipping costs are not included in the price comparison
CL_3	How is the "Same category" defined? (with colors, materials, size, or design)	Does not consider color, material or size variations
CL_4	If the searched product belongs to multiple categories, which category should be prioritized?	The primary category is used
CL_5	Is there a minimum count? If only 1 product matches, how do we handle it?	If 1-5 products match, all are displayed. No minimum threshold
CL_6	What happens if no related products match the category or price range?	The related products section is completely hidden if 0 products match
CL_7	What happens if there are more than 6 related products that match? How to prioritize the display of 6 similar products?	Display the top 6 based on sales rank
CL_8	If a product meets the category and price range but is out of stock, should it still appear in the similar product list?	Exclude from the related products section
CL_9	Should the main product be excluded from the related products list ?	Main product is never shown in its related products list
CL_10	Does it need to consider the region when displaying best seller products list ?	Best sellers are not region specific
CL_11	Does it need to display the same related products list for logged-in users and guest users?	shows the same generic related products for both users

CL_12	What happens if the similar products related API fails? Are there any failure handling methods?	Show a proper error message to inform the system failure
-------	---	--

5 Testing Approach

5.1 Testing Levels

Level	Description
Functional Testing	<ul style="list-style-type: none"> • Related products validation • Category validation • Price range validation • Related products count validation
Boundary Value Testing	<ul style="list-style-type: none"> • Minimum boundary (50%) • Maximum boundary (500%) • Below minimum (49%) • Above maximum (501%)
UI Testing	<ul style="list-style-type: none"> • Proper layout • Responsiveness
Compatibility Testing	<p>Verify feature works on:</p> <ul style="list-style-type: none"> • Chrome • Firefox • Edge • iOS • Android
Integration Testing	<p>Verify integration between:</p> <ul style="list-style-type: none"> • Frontend UI • Backend API

6 Test Environment

- **Devices:** Desktop (Windows/Mac), Mobile (iPhone, Samsung).
- **Browsers:** Chrome, Edge, Firefox
- **Automation Testing:** Playwright and GitHub
- **Tools: Test Management:** Jira & **API Testing:** Postman

7 Test Data Requirement

Test data will include:

Valid data:

- Products within the price range
- Same category products
- Best seller products

Invalid data:

- Products outside the price range
- Different category products
- Duplicate products

Boundary data:

- 50%
- 49%
- 500%
- 501%

8 Test Deliverables

- Requirement Clarification document
- Test Strategy document
- Test Plan document
- Test Cases document
- Test Data

- Test Execution Reports
- Bugs Reports
- Test Summary Report

9 Risks and Mitigations

Risk	Mitigation
Environment instability	Coordinate with the development and DevOps team
Insufficient or Incorrect Data	Use verified test data
API performance issues	Include performance tests in strategy

10 Entry and Exit Criteria

10.1 Entry Criteria

- Feature requirements are finalized and clarified
- Test environment is ready
- Test data is prepared
- Development build is deployed to staging

10.2 Exit Criteria

- All test cases are executed
- No critical or high severity bugs are open
- Cross browser testing is completed
- UI matches the approved Figma

11Automation Strategy

The following test scenarios will be automated using Playwright:

1. Test case 01 : Maximum 6 related products
2. Test case 02 : Verify related products are within acceptable price range (50% to 200%)
3. Test case 03 : Verify less than or equal to 6 best sellers available
4. Test case 04 : Verify product title is displayed
5. Test case 05 : Verify product price is displayed
6. Test case 06 : Verify no duplicate products are displayed

Framework follows:

- Page Object Model (POM)
- Reusable functions
- Separation of test logic and page elements

End