

TEST PLAN FOR

Company: WESTWINGNOW

Feature: SHOP THE LOOKS

1. INTRODUCTION

This document describes the plan for testing the shop the looks feature on Westwingnow website (<https://www.westwingnow.de/looks/>)

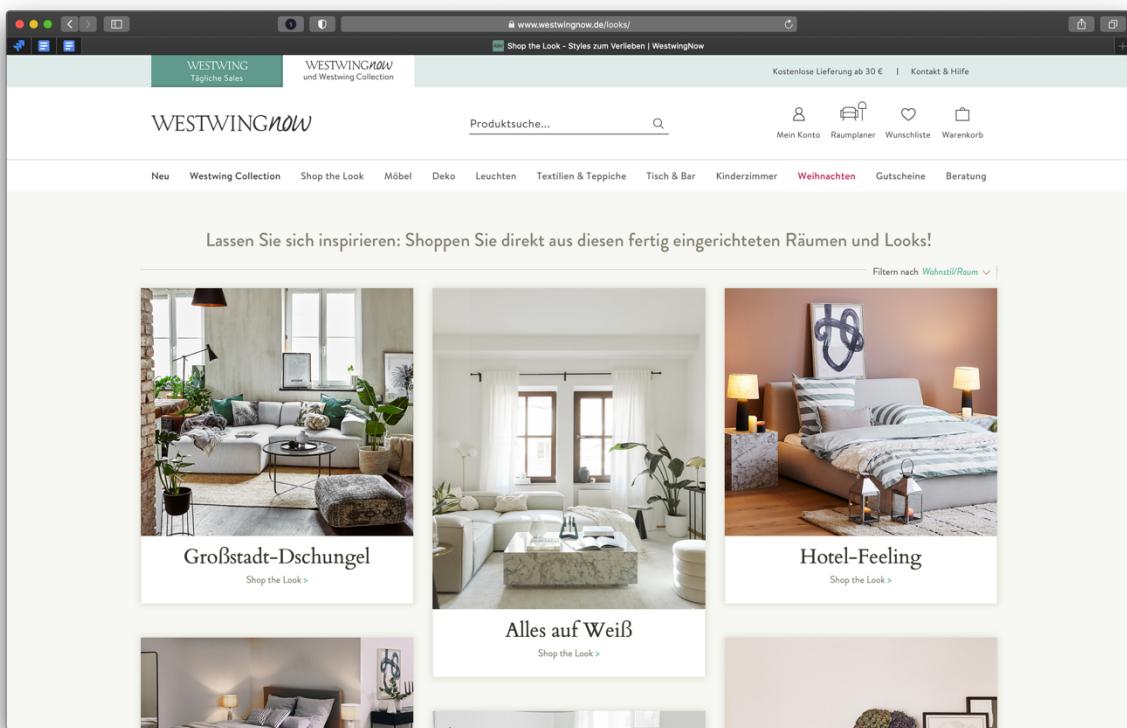
This document supports the following objectives:

- Identify the applications that should be tested.
- List the recommended test requirements (high level).
- Recommend and describe the testing strategies to be employed.
- Identify the required resources and provide an estimate of the test efforts.
- List the deliverable elements of the test activities.

2. TEST ANALYSIS

2. 1 Application under test (AUT):

- **AUT – 01:** Looks listing default page



User activities and state transition analysis:

- Navigate from another page to LOOK listing page
- Navigate from LOOK listing page to another page
- Scroll down/scroll up to view all looks in long list
- Mouse hover on a look/ Mouse away from a hook
- Select to view detail of a look/ Go back to listing page
- Change size of browser window (full screen – not full screen – smaller screen)
-

Functional Test Data Proposal:

a.Input:

- Images with the different sizes, following requirements of the allowed image sizes; image dimensions
- Title of look: short text & long text, following the standard design for the allowed length of title

(These should be mentioned in Acceptance Criteria of User Story)

b.Output:

- Nicely view of look listing page: good responsive design- images rescale and title showing fully

Non-Functional Test Data Proposal (for Basic Performance Test):

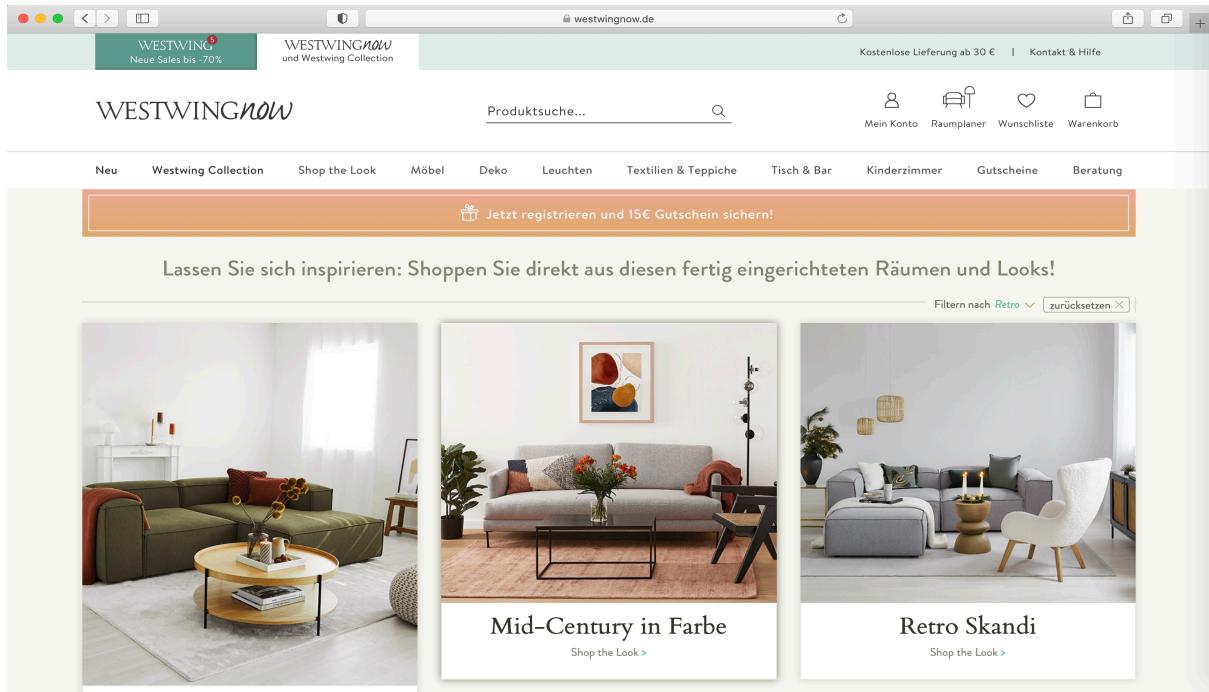
a.Input:

- Use lots of the different image file capacities in the range, following the requirement of the allowed image capacity

b.Output:

- Good performance when loading images on look listing page

▪ **AUT-02: Look listing page filtered by room and style**



User activities and state transition analysis :

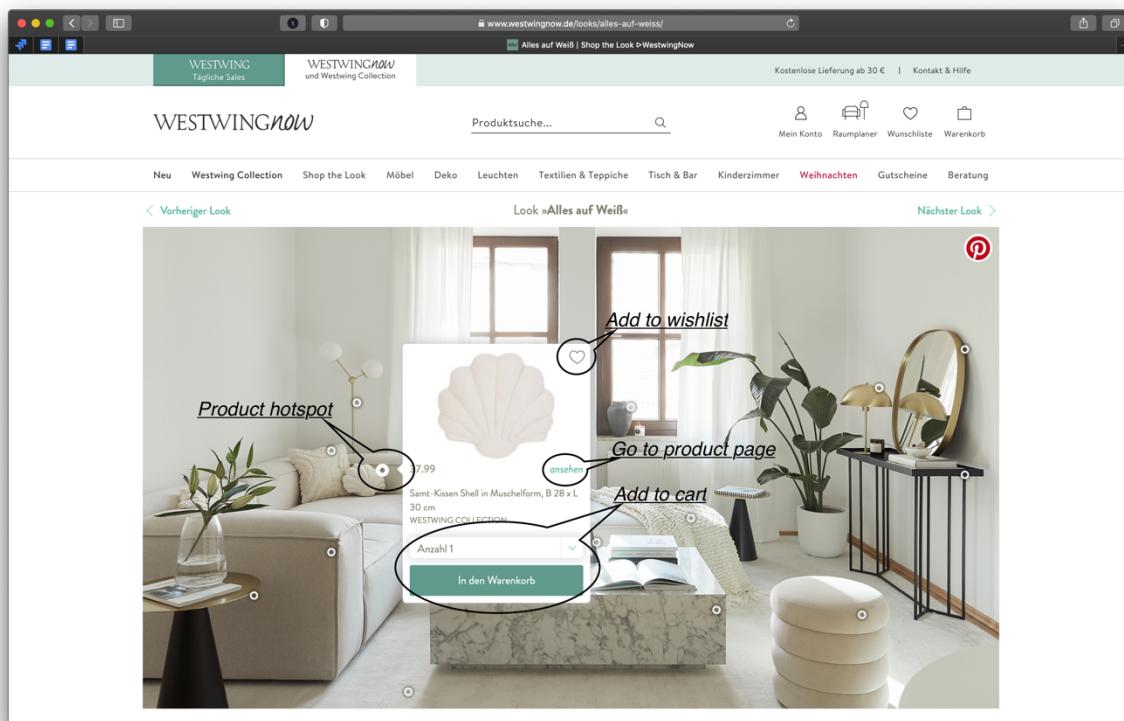
- Mouse hover on the filter by conditions -> a dropdown list of conditions showing
- Filter by Living Style

Living Style	Filterable	Reset to default
Morden	1	1
Skandi	1	1
Romantisch	1	1
Glamour	1	1
Industrial	1	1
Retro	1	1
Ethno	1	1
Klassisch	1	1
Country	1	1
Total	9	9
Number of test cases	18	

- Filter by Room

Living Style	Filterable	Reset to default
Wohnzimmer	1	1
Flur	1	1
Special	1	1
schlafzimmer	1	1
esszimmer	1	1
küche	1	1
Outdoor	1	1
arbeitszimmer	1	1
gästezimmer	1	1
Total	9	9
Number of test cases	18	

- Filter by Living Style and Room: using pairwise analysis to decrease number of test cases if Reset to default = (NO/YES)
→ number of test cases = $9 \times 9 = 162$
- Filter result : no data; $3 \geq$ number of looks ≥ 1 ; number of looks > 3
 - **AUT-03:** Look detail page, e.g.



User activities and state transition analysis:

- View detail of a look: look image & short caption

Navigation to	Previous look	Next look
- First look in the list		X
- Looks in the middle of list	X	X
- Last look in the list	X	

- List of the associated products
- Hover mouse on hotspots on the product image to get preview of a product in image.
- From preview of a product, add it to wishlist (logged in user, not login user)
 - User case (1). Continue preview other products, add it into wishlist, then go to wishlist page
 - User case(2). Go to wishlist page
- From preview, select size (if any) and quantity, then add product into shopping cart → Preview dialog is auto closed →(1) Continue preview another product, and add it into shopping cart (2) Go to shopping cart to checkout.
- Open detail page of preview product on hotspot.

2.2 Related pages:

- Shopping cart preview: showing the 2 newest added products with name, quantities, unit price and the total price of all products in basket
- Shopping cart : showing the correct added products with quantities
- Product detail page

3. TESTING SCOPE

3.1 In Scope:

The following table describes the core functions with the detail requirements and number of test scenarios of each requirement as well as number of test cases that should be executed. This information will be used to calculate test coverage.

Core Functions	Requirement ID	Requirements	No.of Test Scenarios	No.of Test Cases

Look listing page	RQ-LLP-01	Has a list of all existing looks, with an image, title and link		
Look listing page	RQ-LLP-02	Has option to filter looks by Living Style AND Room		
Look detail page	RQ-LDP-01	Has an image, look name and a short caption		
Look detail page	RQ-LDP-02	Has a list of products associated with this look		
Look detail page	RQ-LDP-03	Has hotspots on the product image, that show product preview and allow to add the product to cart/wishlist, lead to the detail page of the product		
Look detail page	RQ-LDP-04	Has navigation to the next and previous look		
Total number of requirements:	6			

3.2 Out Scope

To guarantee the good quality of this feature, **API testing, Performance testing and Security testing** must be considered in the master test plan and should be tested before release. Those kinds of test are not in the scope of this test plan.

4.TEST LEVEL requires:

- Unit test
- API test
- Integration test: checking how backend & frontend work together
- Compatibilities test: checking on multiple browser types & responsive ability
- Acceptance testing

5.TEST TYPES:

- Manual UI testing - UI automation testing
To define
- Explore testing
- API testing - API automation testing
- Performance testing

6.TEST ENVIRONMENT:

Depends on Master Test Plan of whole e-commerce site, but in general the following test environments should be covered:

- Window OS, Mac OS: Test on the different Screen Resolutions
- Android devices, IOs devices: Android/ IOs software version and screen sizes
- Browsers: Chrome/ Firefox/ Safari
- Browsers on Android devices, Ios devices

Test Environment will be defined more details base on the real business, and will be decided by all stakeholders

7.HOW AND WHO RUN TEST

- Developer implements unit test and will run unit test whenever they have new changes on code
- QA prepare API test cases and implements API automation test using Mock-API and then will execute automated API test when API is ready on DEV env
- QA or Developer will execute automated API test on CI when there is new change from backend site vs each environment: DEV/ STAGING/ PRODUCTION
- QA prepare UI test cases which includes integration test. Developer will follow this test suite for checking his code before merging it.
- QA will execute UI test and explore testing on DEV/ STAGING env
- QA will run regression test on each environment when there is new change

Team responsibilities in the testing flow:

- QA has responsibilities for bug report, bug tracking on JIRA
- Developer has responsibilities for fixing bugs; deploy the fix into each environment.

- QA has responsibilities for verify and update status of bug in each environment

8. TEST COVERAGE

Test coverage will be calculated from the following factors:

- Unit test code coverage
- Feature testing metric
 - Requirements coverage

$$\text{Requirements Coverage} = \frac{\text{Number of requirements covered}}{\text{Total number of requirements}} \times 100\%$$

- Test cases by the requirement(request)

Request	Test Case	Test Result
Request 1	Test Case 1	Pass
Request 2	Test Case 2	Failed
Request 3	Test Case 3	Incomplete

- Requirements without Test Coverage

9. EXIT CRITERIA

- Ensuring all critical Test Cases are passed
- Achieving complete Functional Coverage
- Identifying and fixing all the high-priority defects
- Fixing all the 'Show Stopper defects' or 'Blockers' and ensuring that none of the identified Critical/Severity 1 defects are in Open Status
- Re-testing and closing all the high-priority defects to execute corresponding Regression scenarios successfully

How To Select Correct Test Cases For Automation Testing

Step 1:

Identify the parameters on which you will base your test case as a candidate for automation.

- Test case is in the critical part of application
- Test case executed with different sets of data.
- Test case executed with different browsers.
- Test case executed with different environments.
- Test case executed with complex business logic
- Test case executed with a different set of users
- Test case involves a large amount of data
- Test case takes long time to execute test
- Test case has any dependency
- Test case requires Special data

Step 2:

Analyze and try to identify the test cases which should be automated based on the parameters. This steps to define **automation candidate test cases**

For example: Based on the above parameter, we should automate test 24 / total 44 test cases of LOOK feature

Step 3:

Consolidate and group the number of test cases for each feature and **giving an estimate** to finish the testing manually.

Present data so it'll be easier to do test automation ROI calculation

Features	Number of test cases to be automated	Time required to do manually execute	Time to run test using automation script
Look	24		

Test Automation ROI Calculation