

BuyMore Project

Revision History

Date	Description	Author	Comments
01.10.2024	Test Plan for version 0.55	Marcu Cornel	versiunea 1.0

Table of Content:

1. Introduction
 - 1.1 Project objective
 - 1.2 Functionalities in scope
 - 1.3 Functionalities and tests out of scope
2. Test process
 - 2.1 Test planning
 - 2.2 Test analysis
 - 2.3 Test design
 - 2.4 Test implementation
 - 2.5 Test execution
 - 2.6 Test closure
 - 2.7 Test monitoring and control
3. Test deliverables
 - 3.1 Test plan
 - 3.2 Test conditions
 - 3.3 Test cases
 - 3.4 Daily test summary reports
 - 3.5 Traceability matrix
 - 3.6 Test case results
 - 3.7 Bugs report
 - 3.8 Test completion report

1. Introduction

Buy More. Ro is an online store for Romanians, with delivery exclusively on the territory of Romania. Since its establishment – 2017, our team aims to come in front of customers with the best quality-price ratio for the products sold. We carefully choose the products listed on Acasa and maintain the quality of the products, constantly diversifying the range of products.

So far, we have managed to cover an important segment of the market: basic clothing. With the help of our customers, we have identified the biggest needs in the niche and adapted so that we can also offer quality experiences, not just quality products at fair prices. We identified the biggest fear of customers in this niche: the fear of not choosing the right size. Thus, on Acasa you benefit from free, simple and fast returns. The free return is also a guarantee for the quality of the products.

Also, by actively listening to the needs of customers, we have discovered that more and more

1.1 Project Objective

Before releasing the project to customers, it's crucial to elevate trust in its quality to the highest level achievable.

Application being tested: <https://buymore.ro/>

Documentation: <https://buymore.ro/despre-noi/>

1.2 Functionalities in scope

All features, functionalities, and capabilities outlined in the BuyMore business requirements for the customer's point of view will be subject testing across various types, including functional testing, system testing, acceptance testing, component testing, and compatibility testing. The BuyMore application will be tested on latest versions of Google Chrome.

1.3 Functionalities and tests out of scope

- Non-functional testing like stress, performance, security and maintenance is beyond scope of this project.
- No QA support for mobile applications developed. Only web applications will be tested.
- Automation testing is beyond scope.
- All features that are not under Admin Module

2. Test process

2.1 Test planning

Roles and responsibilities

Matei -QA junior	will test: User Management, Job, Qualifications
Adelina - QA Senior	will test: Organization, Nationalities, Configuration

Entry criteria:

- smoke test passed (being the most basic type of test, this is a very important entry criteria in the process of testing)
- testing environment is up and running
- roles and responsibilities for the project are allocated
- functional business specifications are defined

Exit criteria:

- 100% of tests are executed
- 90% of tests are passed
- no Critical defects have Open status
- User Add functionality tests are 100% passed
- Exploratory testing was performed user module

Risks:

- stability risks (crashes, disconnects, etc)
- IE browser might have performance issues
- the web page pagination could be impacted when opened on mobile devices
- stress conditions might impact the web application
- new browser might not be supported
-

2.2 Test analysis

- Analyze the business requirements to make sure that we have all the details for creating the test conditions
- Write test conditions (What ?)
- We plan on running a full regression test on the current version

2.3 Test design

- All the test cases (How?) are written and reviewed
- All test cases are created in Jira as test management tool

2.4 Test implementation

- all the test data is available and reviewed (test data= email examples, password examples, employee, user with admin role)
- this test run includes only regression testing in which we will run tests that have the highest priority, this will be main priority
- Create the test suites (Cycle Summary) (grouped by release version or testing type)

2.5 Test execution

- The tests will be executed on the latest versions of browsers: Chrome, Mozilla Firefox, IE. If time will be available we will extend tests on other browsers.
- Bugs (defects) will be created based on the failed test cases
- The full regression testing will be done after new application changes
- Retesting will be done after a bug fix

2.6 Test closure

- at least 90% of tests are passed
- no Critical issues have Open status

2.7 Test monitoring and control

- Status reports will be generated to reflect the current status of testing process
- In case of major problems, control measures will be taken

3. Test deliverables

3.1 Test plan - link to test plan

3.2 Test conditions

The test conditions will be created based on the business requirements validated in the test analysis phase and will represent the features to be tested and transformed into test cases.

- [test conditions will be exported from Jira and added here](#)

3.3 Test cases

- link to test cases

- [test cases will be exported from Jira and added here](#)

3.4 Daily/Weekly/Monthly test summary report

- link to daily test summary report (number of tests ran today, % of them failed, passed, re-test, etc)

- [reports will be exported from Jira and added here](#)

3.5 Traceability matrix

- link to traceability matrix

- [traceability matrix will be exported from Jira and added here](#)

3.6 Test case results

- link to test case results

- [test case result will be exported from Jira and added here](#)

3.7 Bugs report

- link to bugs report

- [bug reports will be exported from Jira and added here](#)

3.8 Test completion report

- link to test completion report

- [test completion report will be exported from Jira and added here](#)

3.9 Schedule

- we have 10 days of testing

- we have 30 test cases
- in order to finish the test run we would need to run an \sim of 3 tests/day