Test Plan for an E-commerce Application

1. INTRODUCTION

1.1 Purpose

The purpose of this plan is to develop and present a test process for the "Product List" application. Tests are planned to verify the shopping cart functionality - adding a product or products in different variants to the cart, editing it's content and finalizing the order, meets the requirements. The tests covered by this plan will include only non-logged-in desktop users.

1.2 Project overview

- The "Product List" application is designed to allow the user to purchase products available in the online store
- Project budget:
  -QA team: 3 MT (2 junior, 1 mid), 1 AT (junior), 1 Test Manager
  -Dev team: 2 FE, 2 BE, 1 Product Owner, 1 Scrum Master, 1 Project Manager
- Test devices to be used: 4 computers with different variants of Windows software – 1x Windows 11, 1x Windows 10, 1x Windows 8, 1x Windows 7. The most popular web browsers on the market will be installed on each device
- An international team with branches in EU countries located in Poland: 1 FE and 1 BE from dev team works from Poland, 1 FE and 1 BE from Germany, rest from the Czech Republic. QA split between 3 locations in Poland: Poznań, Katowice, Warszawa
- An external company from Scandinavia is responsible for the BE layer
- Teams work in the Scrum methodology
- Tool stack: Jira & all plugins offered

1.3 Audience

The audience of the app will be developing team, testing team, PO and the management board of the company ordering the project.

2. TEST STRATEGY

2.1 Tests goal

1. Verification of the functionality of:
   - Adding a product to the cart
   - Editing the contents of the cart
   - Finalizing the order

2. Identification of defects
3. Retesting

2.2 Test Assumptions

- Programming of the application has been completed
- The application is ready for testing
- Test environment is available
- The testing team has knowledge of how the application works
- Access to the tools/equipment needed to perform testing

2.3 Levels and types of testing

2.3.1 Unit tests

Purpose: to check the correctness of a module, class, program function

Scope: module, class, function, program method

Testers: development team

Method: automatic tests from the code level

Timing: throughout software development

2.3.2 Integration and system tests

Purpose: minimize the risk of defects getting through to higher levels

Scope: interfaces, infrastructure, service communication → system integration

Testers: QA team

Method: manual, automatic, tool-supported testing

Timing: after integration of modules/systems

2.3.3 Acceptance tests

Purpose: application compliance with requirements

Scope: application as a whole

Testers: client/business/PO

Method: manual testing

Timing: before deployment to production

2.3.4 Exploratory tests

Purpose: getting to know the application

Scope: finished application

Testers: testing team

Method: manual testing

Timing: product/functionality is ready / new person joining the project

2.3.5 Functional tests

Purpose: test the functional aspects of the application (answer the question "What does the application do?")

Scope: specific functions of the application

Testers: testing team

Method: manual tests, automatic tests

Timing: given function is ready

2.3.6 Non-functional tests

Purpose: to test non-functional aspects of the application (answer the question "how does the application work?")

Scope: specific functions of the application

Testers: testing team

Method: manual + support with automated tools

Timing: given function is ready

2.3.7 Automated Regression Testing

Purpose: support for manual regression testing

Scope: specific application functions

Testers: testing team

Method: automated scripts

Timing: the specified function is ready

2.4 Test deliverables

- Test strategy
- Test plan and estimation
- Test scenario
- Test cases and test data
- Test summary report
- Test closure report
- Incident report

2.5 Test effort estimation

QA activity	Test effort (SP)
Test plan development	3SP
Requirements analysis	2SP
Test case development	5SP
Test case execution	8SP
Defect reporting	2SP
Retesting	2SP
Regression	3SP
Final report	3SP
Writing automated test scripts	5SP

3.EXECUTION STRATEGY

3.1 Entry and exit criteria

3.1.1 Entry criteria:

- Business documentation is available
- Test data is available
- Application is ready for testing
- Environment is available

3.1.2 Exit criteria:

- All test cases have been executed
- Retests have been completed
- Defects are closed
- Summary report is written and delivered to business
- No regression in the application
- Acceptance tests have been completed
- Milestones achieved

3.3 Validation and defect management

1. Creating a defect report
2. Forwarding the defect report to the development team
3. Debugging the defect
4. Fixing the defect by the developer
5. Forwarding to retest
6. Retest
7. Error occurs = back to debugging / error fixed = closing the defect report

3.4. Test Metrics

Metric	Formula
Percentage test cases executed	No of test cases executed/ Total no of test cases written X 100
Passed Test Cases Percentage	Number of Passed Tests/Total number of tests executed X 100
Failed Test Cases Percentage	Number of Failed Tests/Total number of tests executed X 100

4.TEST MANAGEMENT PROCESS

4.1 Test management tool

Jira will be used to manage the testing process. Defects will be reported as Bug tickets within Jira.

4.2 Test design process

The **xRay** plugin in Jira and Testrail will be used to manage test cases.

4.3 Test execution process

Test runs will be created based on previously prepared test cases.

4.4 Test risks and mitigation factor

Risk	Probability	Impact	Mitigation plan
Testing schedule is tight. If the start of the testing is delayed due to design tasks, the test cannot be extended beyond the UAT scheduled start date.	**High**	**High**	The testing team can control the preparation tasks (in advance) and the early communication with involved parties.
Not enough resources, resources on boarding too late (process takes around 15 days.	**Medium**	**High**	Holidays and vacation have been estimated and built into the schedule; deviations from the estimation could derive in delays in the testing.
Defects are found at a late stage of the cycle or at a late cycle; defects discovered late are most likely be due to unclear specifications and are time consuming to resolve.	**Medium**	**High**	Defect management plan is in place to ensure prompt communication and fixing of issues.
Non-availability of Independent Test environment and accessibility	**Medium**	**High**	Due to non-availability of the environment, the schedule gets impacted and will lead to delayed start of Test execution.
Lack of experience of testers	**High**	**High**	Support from another tester more diverse configuration of the testing team / Set a longer time for the testing process
Lack of knowledge of requirements	**Medium**	**High**	Read the requirements / read with understanding / ask the PO to discuss the requirements
Documentation errors	**Medium**	**High**	Fix errors in the documentation / clarify the requirements
Team communication errors	**Low / Medium**	**Medium**	Select the team in terms of character / support the Scrum Master in resolving conflicts / integrate the team
Lack of tools	**Medium**	**Medium**	Use free tools / Ask for an increase in the budget / look for alternatives / purchase tools

Underestimation of workload required for testing	**Medium**	**Medium**	Apply for team expansion / outsourcing work / review previous estimates / request an extension of the time for testing
Non-functioning test environment	**High**	**High**	Repair the environment / check the correct configuration of the environment in advance / use an alternative environment
Lack of devices for testing	**Medium**	**High**	Look for alternatives → device farm / device mode in dev tools / rent equipment / buy devices
Application does not work according to requirements	**High**	**High**	Commit to requirements analysis / implement the principle of early testing / put more effort into code review / improve communication with the PO
Staff shortages	**Low / Medium**	**Low / Medium**	Recruitment of employees / employee hire / outsourcing / external contracts

4.5 Test responsibility

4.5.1 QA team

- Developing a test plan
- Executing tests
- Analyzing requirements
- Creating test cases
- Reporting (defects, summary)
- Retesting
- Monitoring the test process

4.5.2 Development team

- Fixing defects
- Writing code
- Writing unit tests
- Debugging
- Code review

5. TEST ENVIRONMENT. COMPATIBILITY CHART

- Testing will be done on pre-production environment
- Windows 11, 10, 8, 7
- Chrome, Mozilla Firefox, Opera, Edge (current version + 1 version back)
- Responsive mode

6. TESTING TOOLS

Area	Tool
Creating documentation - test Plan, test report, defect report	Jira, Confluence
Creating test cases	Testrail
API testing	Postman
Tools for documenting test results	TechSmith Capture
Performance audit, SEO, good practices	Lighthouse
Test execution	Devtools
Automated test scripts	InteliJ
Documentation other than point 1	MS Office Suite
Test execution	Fake mail