TEST STRATEGY FOR WWW.BDVRENTACAR.RO

PROJECT NAME: WWW.BDVRENTACAR.RO

DOCUMENT CONTROL

Version	**1.0**
Date	02.01.2023
Developed by	Matei Dorin

DOCUMENT SIGN-OFF

Version	Status	Date	Approved by	Job Title
1.0	Final	02.01.2023	Prisecaru Alexandru	Product Owner

TABLE OF CONTENTS

1 INTRODUCTION

Through the testing process we will test the rent a car application www.bdvrentacar.ro. We are particularly interested in functional testing based on User Stories for discovering the defects.

2 PURPOSE

The purpose of the testing team is to check the different functionalities of the application www.bdvrentacar.ro before the new release. We will check the functionality of the language change button, displaying the logo and its property to return the user to the main page, as well as displaying phone numbers in the header of application. We will also test the 'Prices' section, the correct display of the cars in the 'Rent a Car' section, as well as the loading of the 'Privacy Policy' page.

3 SYSTEM OVERVIEW

4 SCOPE OF TESTING

4.1 IN SCOPE

The types of testing that are in-scope are:

- Static Testing:
    - Requirements
- Functional Testing
- Regression testing (where required)

4.2 OUT OF SCOPE

The types of testing that are out of scope are:

- Static testing:
    - Architecture
    - Code Review
- Unit Testing
- Non-functional Testing
    - Performance (Load, Stress)
    - Security testing
    - Accessibility
    - Browser compatibility
    - Mobile compatibility
- Operational Acceptance Testing
    - Deployment
- User Acceptance Testing – it will be planned and executed by the client

5 APPROACH TO TESTING

5.1 PRINCIPLES & APPLICATION

5.1.1. Principle

In our testing process we will follow these principles:

1. Testing shows the presence of defects, not their absence
2. Exhaustive testing is impossible
3. Early testing saves time and money
4. Defects cluster together
5. Beware of the pesticide paradox
6. Testing is context dependent
7. Absence-of-errors is a fallacy

5.1.2. Application

In our project we will apply the principles above, in this manner:

1. Testing shows the presence of defects, not their absence
   Our testing team will show that defects are present, but they cannot prove that there are no defects.
2. Exhaustive testing is impossible
   It is impossible to test all combinations of inputs and preconditions.
3. Early testing saves time and money
   Our team will start the testing process as early as possible to avoid additional costs.
4. Defects cluster together
   A small number of modules usually contains most of defects discovered. Our team will try to find those modules to be as efficient as possible in discovering of defects.
5. Beware of the pesticide paradox
   To detect new defects, the team may need to change existing tests and test data and new tests may need to be written.
6. Testing is context dependent
   Testing is done differently in different context. Depending of different types of software, the team needs to perform different types of testing.
7. Absence-of-errors is a fallacy
   We will try to find the defects and to create an usable application. If the software or system is unusable then does not matter how many defects are found and fixed – it is still unusable.

5.2. TEAM – PLANNED ITERATIVE

Our teams will use the Agile working methodology in the application testing.
The Agile methodology is an iterative approach to project management and software development that uses feedback loops and test-driven development to solve the problems.
It is not a single method, but a collection of best practices that involve constant collaboration.
The Agile process has a positive impact on project success. By prioritizing tasks and properly distributing the amount of work required to complete deliverables, the team become more effective.
Among the benefits of using the Agile methodology are also the following:

1. Increased efficiency
2. Satisfied customers
3. Improved product quality
4. Greater flexibility.

5.3. FIRST SPRINT

5.3.1. Objective

The objective of the Sprint1 is to verify that the www.bdvrentacar.ro application works according to the specifications.
The Testing Team will create and execute test cases to ensure that we discover as many defects as possible.

5.3.2. Scope

The scope of testing for Sprint1 is to perform Functional Testing. According to the specifications, we will check the functionality of the language change button, displaying the logo and its property to return the user to the main page, as well as displaying phone numbers in the header of application.

5.3.3. Test Preparation

Test cases will be created in TestCase Lab to meet the customer's requirements.

5.3.3.1. Entry Criteria

- Verify if the Test environment is available and ready for use.
- Verify if test tools installed are ready for use.
- Verify if Testable code is available.
- Verify if Test Data is available and validated for correctness of Data

5.3.3.2. Exit Criteria

- All user stories in the sprint backlog must be completed
- No critical and High severity issues must be open
- All expected and actual results are captured and documented
- All defects are logged

5.3.4. Test Execution

For Test Execution the team will use the TestCase Lab tool.
The team will follow the next steps:

1. Execute each of the test steps in Test Case
2. Mark the status of the Test Case as: Pass, Fail, Blocked
3. Raise defects in JIRA for the failed Test Cases
4. Send the daily status report to Test Lead
5. Participate in Defect Triage Cycle and explain the defects
6. Complete the test execution of all Test Cases

5.4. SECOND SPRINT

5.4.1. Objective

The objective of the Sprint2 is to verify that the www.bdvrentacar.ro application works according to the specifications.
The Testing Team will create and execute test cases to ensure that we discover as many defects as possible.

5.4.2. Scope

The scope of testing for Sprint2 is to perform Functional Testing. According to the specifications, we will test the 'Prices' section, the correct display of the cars in the 'Rent a Car' section, as well as the loading of the 'Privacy Policy' page.

5.4.3. Test Preparation

Test cases will be created in TestCase Lab to meet the customer's requirements.

5.4.3.1. Entry Criteria

- Verify if the Test environment is available and ready for use.
- Verify if test tools installed are ready for use.
- Verify if Testable code is available.
- Verify if Test Data is available and validated for correctness of Data
- Sprint1 is complete

5.4.3.2. Exit Criteria

- All user stories in the sprint backlog must be completed
- No critical and High severity issues must be open
- All expected and actual results are captured and documented
- All defects are logged

5.4.4. Test Execution

For Test Execution the team will use the TestCase Lab tool.
The team will follow the next steps:

1. Execute each of the test steps in Test Case
2. Mark the status of the Test Case as: Pass, Fail, Blocked
3. Raise defects in JIRA for the failed Test Cases
4. Send the daily status report to Test Lead
5. Participate in Defect Triage Cycle and explain the defects
6. Complete the test execution of all Test Cases

6. TEST ENVIRONMENT

The test environment for www.bdvrentacar.ro is: PRODUCTION.

7. TEST DATA REQUIREMENTS

- No data: Check system response when no data is submitted
- Valid data: Check system response when Valid test data is submitted

The application does not have the functionality of creating a user account.
To test the application, the user needs this link: www.bdvrentacar.ro.

8. TESTING TOOLS & TECHNIQUES

The Testing Tools we will use for this application are:

- JIRA Software;
- JIRA Confluence;
- TestCase Lab;
- Planning Poker;
- Microsoft Office Pack;
- Google Chrome Version 108.0.5359.125 (Official Build) (64-bit)
- Developer Tools from Google Chrome;
- Windows 11 - Screenshot;
- Paint from Windows 11
- www.rachy.com for website architecture.

The Testing Techniques we will use to test this application are:

1. Static Testing Techniques:
    a. Reviews
    b. Static analysis
2. Dynamic Testing Techniques:
    a. Error guessing
    b. Exploratory testing

8.1. REQUIREMENTS & USE CASE MANAGEMENT

The Testing Team will manage the requirements following the next steps:

1. Investigation
2. Feasibility
3. Design
4. Construction and Testing
5. Release

Also, the Testing Team should check if the application meets the needs of the customers(Use Case). When the customer uses this application, he/she should be able to find all the information needed to rent a car.

8.2. TEST MANAGEMENT & DEFECT TRACKING

The testing process for this project will be divided in the following stages:

1. Test Planning
2. Test Monitoring and Control
3. Test Analysis
4. Test Design
5. Test Implementation
6. Test Execution
7. Test Completion

For Defect Tracking the Testing Team will use JIRA Software.

9. TESTING ROLES & RESPONSIBILITIES

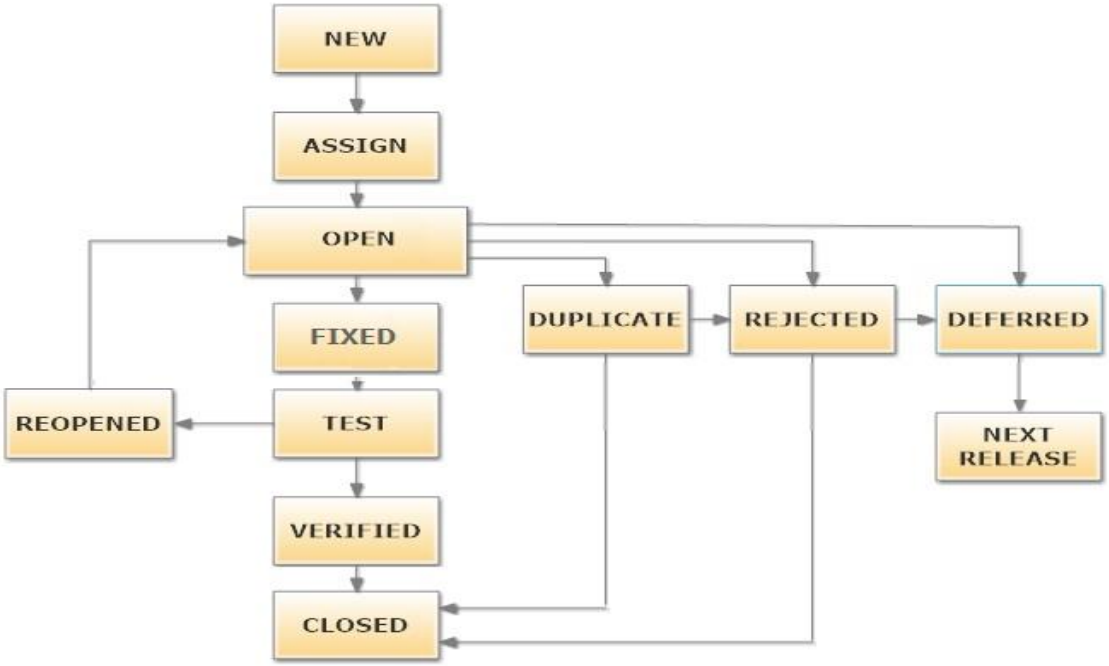
The following table shows the testing roles for the project, together with the individuals involved in the testing effort.

Activity	Responsibility/Ownership	Name
Test Plan Creation	Test Manager	Pocovnicu Alexandru
Test Management	Test Manager	Pocovnicu Alexandru
Test Analysis and Design	Test Manager	Pocovnicu Alexandru
Test Preparation, Execution & Results	QA Analyst	Matei Dorin
Test Defect Submission	QA Analyst	Matei Dorin
Test Summary Reporting	Test Manager	Pocovnicu Alexandru
Test Completion Reporting	Test Manager	Pocovnicu Alexandru

10. TEST MANAGEMENT

Overall responsibility for the Testing Project will be with the Development Company Test Manager Pocovnicu Alexandru.
The role of QA Analyst in the present project is fulfilled by Matei Dorin.
In our team, the Test Manager has the responsibility for Test Plan Creation, Test Management, Test Analysis and Design, Test Summary Report and Test Completion Report.
On the other hand, the QA Analyst has the responsibility for Test Preparation, Test Execution and Results, and Test Defect Submission.

11. DEFECT MANAGEMENT



11.1. DEFECT MANAGEMENT PROCESS

In the process of resolving the defects, the test team will use the following statuses:

1. Duplicate defect – the same bug was reported by another team member.
2. Invalid defect – when QA or Devs Misunderstood the requirements.
3. Won't fix defect – when the defect is valid, but it will not be fixed.
   Reasons:
   - No solution
   - Too expensive
   - No technology support
   - Fix would add too much complexity
4. Can't be reproduced defect
   Reasons:
   - Incomplete navigation steps
   - Server mismatch
   - Platform mismatch
   - Data mismatch
   - Build mismatch
5. Rejected defect – when the developer is not able to understand it
6. Deferred defect – was not fixed because of the time constraints
7. Fixed defect – the defect was fixed

12. TEST SCHEDULE

No.	Task Name	Start Date	End Date
1	Test Strategy	23.12.2022	02.01.2023
2	Test Plan	02.01.2023	09.01.2023
3	Test Case Preparation	02.01.2023	09.01.2023
4	Sprint 1	09.01.2023	16.01.2023
5	Test Completion Report 1	16.01.2023	16.01.2023
6	Sprint 2	16.01.2023	23.01.2023
	Test Completion Report 2	23.01.2023	23.01.2023
6	UAT	30.01.2023	31.01.2023

13. REFERENCED DOCUMENTS

#	Document	Author	Description
1	Project Documentations	Andreea Horhogea	This provides information with regards to the Project as a whole.