Table of contents:

- **Introduction…....2**
- **1. Required tests ……..3**
- 1.1. Manual tests
  - 1.1.1 Exploratory testing
  - 1.1.2 Functional Positive tests
  - 1.1.3 Functional Negative tests
- 1.2 Website automation test (Selenium IDE) **….....4**
- 1.3 website API test
- 1.4. Website Performance Automation test with
- **2. Quality objectives ….....5**
  - 2.1 Primary Objectives
  - 2.2. Secondary Objectives
- **3. Roles and responsibilities 5**
- **4. Entry and exit criteria 6**
- **5. Bug Severity and Priority Definition 7**
- **6. Resource and environment needs 8**
- **7. Test Environment (browsers) 8**
- **8. Bug Life cycle 9**
- **Test schedule.....**

Test plane for Miami Family Club - MFC

Date: 04.04.2023

Document created by: Olesia Siniurina
https://www.linkedin.com/in/olesiasiniurina/
https://github.com/OlesiaSin

Introduction: The customer needs convenient and functional booking of events on the site through an account. The test plan was created to facilitate communication within the team members. The documents will indicate which tests, methods, platforms, timelines will be used.

1. Required tests

Functional to be tested:

- Creating an account
- Event ticket booking
- Log-in; Log-out

1.1. Manual tests

○ 1.1.1 Exploratory testing

You are browsing the site as a normal user, explore site, how it look (UI-test) and how it works (UX-test)

○ 1.1.2 Functional Positive tests

TC01. Create a new account.
TC02. Change account information
TC03. Add new event in account
TC04. Delete event from account.
TC05. Delete account.

Do the right thing, fill in the right information where it should be, and press the right buttons in the right sequence
For this type of test, test cases are already being written, as isolated as possible for each functionality.
If there are no errors, then everything is fine, Positive test - PASS.
If there are errors, then your Positive test- FAIL.

○ 1.1.3 Functional Negative tests

TC06. Create a new account without email.
TC07. Create a new account without a Username.
TC08. Create an account using incorrect data in the Phone number.

We write test cases and run tests on them, understanding how the program should work if the user is doing something wrong. You are doing it WRONG by expecting the program to defend itself and show errors or warnings, but that's what we expect. That is, if you

use incorrect data when registering, the system should not skip your registration and should indicate the fields to correct.
If an error occurs due to incorrect user actions, then everything is in order.
In this case, a negative test- PASS.
If there is no error, then your negative test- FAIL.
If you find bugs, you submit bug reports to Jira.

● 1.2 Website automation test (Selenium IDE)

TC09. Automation of viewing the “Home page” block.
TC10. Automation of viewing and booking an event “Family Snorkeling”**.**
TC11. Automation of viewing page “My Profile” information.

Automate your manual testing using Selenium IDE tool and mark it as PASS or FAIL in "Automation test" sub-task.

● 1.3 website API test (As part of this project, we are testing API site Sololearn)

TC12. Add new address in an existing account
TC13 Change data in an existing address
TC14. Add new product in account
TC15. Delete course from account
TC16. Delete account

The API test should include server response tests, response time, and response size.

● 1.4. Website Performance Automation test with

1.4.1 Lighthouse from USA
1.4.2 GTMetrix from Canada
1.4.3. SpeedLab tools from Australia (Testing on iPhone 12 Pro | Safari 14)

2. Quality objectives

○ 2.1 Primary Objectives

A primary objective of testing is to assure that the system meets the full requirements, including quality requirements (functional and non-functional requirements) and fit metrics for each quality requirement and satisfies the use case scenarios and maintain the quality of the product.
At the end of the project development cycle, the user should find that the project has met or exceeded all of their expectations as detailed in the requirements.
Any changes, additions, or deletions to the requirements document, Functional Specification, or Design Specification will be documented and tested at the highest level of quality allowed within the remaining time of the project and within the ability of the test team.

○ 2.2. Secondary Objectives

The secondary objectives of testing will be to: identify and expose all issues and associated risks, communicate all known issues to the project team, and ensure that all issues are addressed in an appropriate manner before release.
As an objective, this requires careful and methodical testing of the application to first ensure all areas of the system are scrutinized and, consequently, all issues (bugs) found are dealt with appropriately.

3. Roles and responsibilities

Role	**Staff members**	**Responsibilities**
Project manager	Sergei Efremov	1.Acts as a primary contact for development and QA team. 2.Responsible for Project schedule and the overall success of the project
QA	Olesia Siniurina	1.Understand requirements. 2.Writing and executing Test cases 3.Preparing RTM. 4.Reviewing Test cases, RTM. 5.Defect reporting and tracking. 6.Retesting and regression testing 7.Bug Review meeting. 8.Preparation of Test Data. 9.Coordinate with QA Lead for any issues or problems encountered during test preparation/execution/defect handling

4. Entry and exit criteria

● Entry Criteria

• All test hardware platforms successfully installed, configured, and functioning properly.
• All the necessary documentation, design, and requirements information available that will allow testers to operate the system and judge the correct behavior.
• All the standard software tools including the testing tools successfully installed and functioning properly.
• Proper test data is available.
• The test environment such as, lab, hardware, software, and system administration support ready.
• QA resources have completely understood the requirements.
• QA resources have sound knowledge of functionality.
• Reviewed test scenarios, test cases and RTM.

● Exit Criteria

• A certain level of requirements coverage has been achieved.
• No high priority or severe bugs are left outstanding.
• All high-risk areas have been fully tested, with only minor residual risks left outstanding
• Cost - when the budget has been spent.
• The schedule has been achieved.

5. Bug Severity and Priority Definition

Bug Severity and Priority fields are both very important for categorizing bugs and prioritizing if and when the bugs will be fixed. The bug Severity and Priority levels will be defined as outlined in the following tables below. Testing will assign a severity level to all bugs. The Test Lead will be responsible to see that a correct severity level is assigned to each bug. The QA Lead, Development Lead and Project Manager will participate in bug review meetings to assign the priority of all currently active bugs. This meeting will be known as "Bug Triage Meetings". The QA Lead is responsible for setting up these meetings on a routine basis to address the current set of new and existing but unresolved bugs.

Severity List

Severity ID	Severity	Severity Description
1	Highest	The module/product crashes or the bug causes nonrecoverable conditions. System crashes, or database or file corruption, or potential data loss, program hangs requiring reboot are all examples of a Severity 1 bug
2	High	Major system components unusable due to failure or incorrect functionality. Severity 2 bugs cause serious problems such as a lack of functionality, or insufficient or unclear error messages that can have a major impact on the user, prevent other areas of the app from being tested, etc. Severity 2 bugs can have a work around, but the work around is inconvenient or difficult.
3	Medium	Incorrect functionality of component or process. There is a simple work around for the bug if it is Severity 3
4	Low	Documentation errors or signed off Severity 3 bugs

Priority List

Priority	Priority Level	Priority Description
1	Highest	This bug must be fixed immediately; the product cannot ship with this bug.
2	High	These are important problems that should be fixed as soon as possible.
3	Medium	The problem should be fixed within the time available. If the bug does not delay the shipping date, then fix it.
4	Low	It is not important (at this time) that these bugs be addressed. Fix these bugs after all other bugs have been fixed. Enhancements/ Good to have features incorporated-just are out of the current scope.
5	Lowest	Documentation errors or signed off Low 4 bugs.

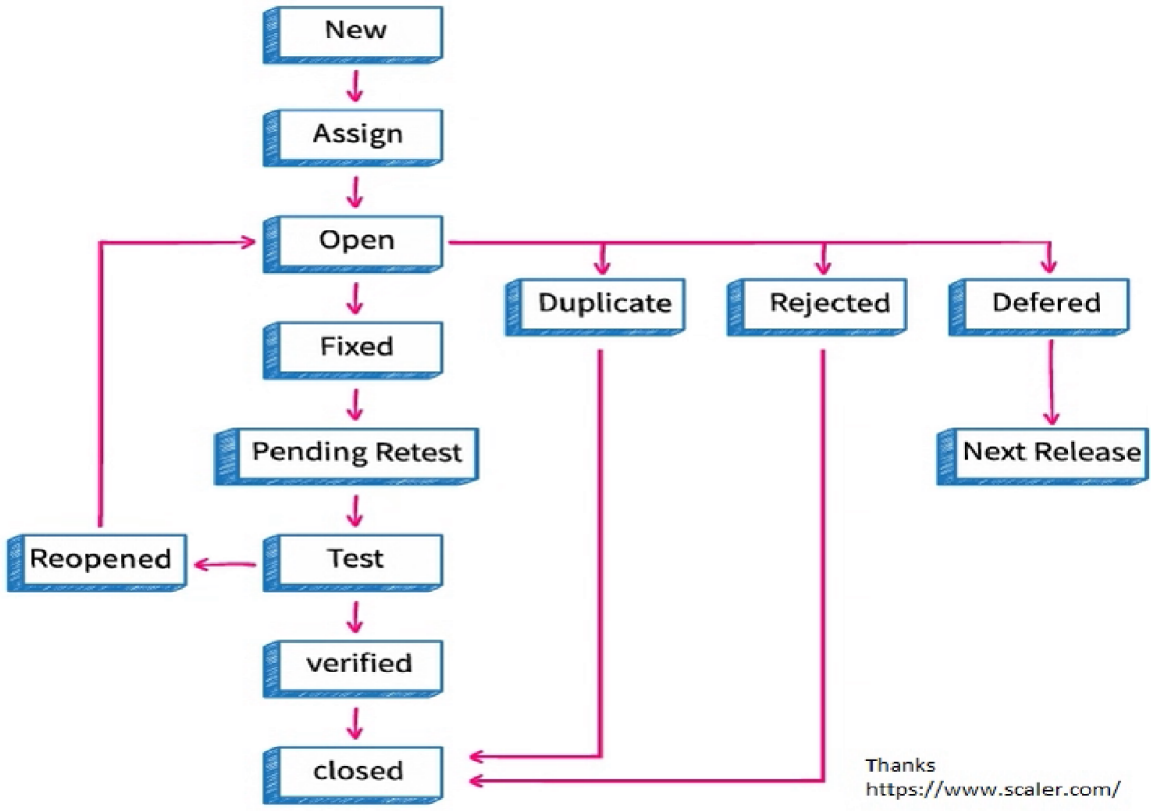
6. Resource and environment needs

Process	**Tool**
Test case creation	PC, Microsoft Excel, Selenium IDE, Postman API, JIRA, GTMetrix, BrowserStuck, Lighthouse
Test case tracking	JIRA, CRM systems
Test case management	Microsoft Excel, JIRA, Confluence, Google drive
Test reporting	JIRA
Check list creating	Microsoft Excel, JIRA

7. Test Environment (browsers)

9.1.1 Windows ; Chrome Version 111.0.5563.148

8. Bug Life cycle



9.Test schedule

Task name	Start	Finish	Actual completion	Comments
Manual tests	04.11.2023	04.25.2023	04.13.2023	
Website automation test (Selenium IDE)	04.11.2023	04.25.2023	04.13.2023	
Website API tests	04.14.2023	04.25.2023	04.15.2023	
Website Performance Automation tests	04.16.2023	04.25.2023	04.17.2023	