

Test plan for [www.sololearn.com](http://www.sololearn.com).

Version 1.0 03.27.2023

Document created by: [Aleksei Siniurin](https://github.com/QAleksei)

**Introduction:**

The customer needs a perfectly working personal account on the

website <https://www.sololearn.com/profile/> “Catalog”. The Test Plan has been created to facilitate communication within the team members. Documents will indicate which tests, methods, platforms, deadlines will be used.

**1.Required tests.**

**1.1. Manual tests:**

**1.1.1 Exploratory testing:**

You act like a normal person, not a tester, just explore the site, how it looks (UI-test) and how it works (UX-test).

**1.1.2 Functional Positive test:**

Create new account.

Delete account.

Change account information.

Add new courses in account.

Delete courses from 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 is Fail.

If you find bugs, create bug reports in Jira (if you are already working in Jira).

**1.1.3 Functional Negative test:**

Create new account with incorrect email.

Create new account without name.

Try add profile photo more than 800kb.

// Write test cases and run tests on them, understanding how the program should work if the user does something wrong.

But You're 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 enter the wrong password, you cannot get into your account. And if the program lets you into your account with the wrong password or username, or without them at all, then this is an obvious BUG! And so on.

If there is an error due to incorrect user actions, then everything is fine, we are just waiting for this Program Error. In this case the Negative test is PASS. If there is no error, then your Negative test is Fail.

If you find bugs, as usual, you make bug reports in Jira.

**1.2 Website automation test (Selenium IDE):**

1.2.1

1.2.2

1.2.3

1.2.4

1.2.5

// 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:**

1.3.1 Create new Account

1.3.2 Change or add Account information

1.3.3 Delete account

1.3.4 Add new course in account

1.3.5. Delete corset from 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

1.4.2 TMetrix

1.4.3. SpeedLab tools

**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 | Sergey 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 | Aleksei  Siniurin | 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**

**6.1 Entry Criteria**

● All test hardware platforms must have been successfully installed, configured, and functioning properly.

● All the necessary documentation, design, and requirements information should be available that will allow testers to operate the system and judge the correct behavior.

● All the standard software tools including the testing tools must have been successfully installed and functioning properly.

● Proper test data is available.

● The test environment such as, lab, hardware, software, and system administration support should be ready.

● QA resources have completely understood the requirements.

● QA resources have sound knowledge of functionality.

● Reviewed test scenarios, test cases and RTM.

**6.2 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.

**7. 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.

**Serenity 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 component 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 to the user, prevents 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. It would be an embarrassment to the company if this bug shipped. |
| 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. |

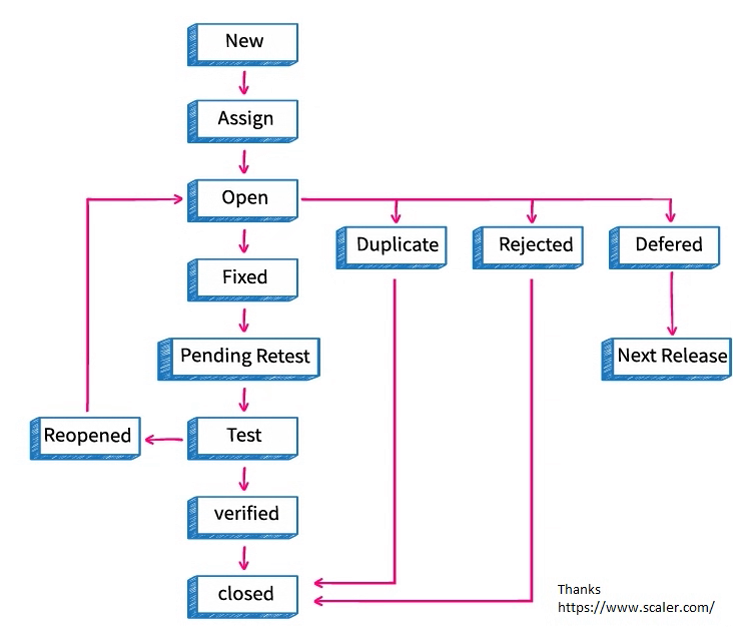
**8. Resource and environment needs.**

| Process | Tool |
| --- | --- |
| Test case creation | PC, Microsoft Excel, Selenium IDE, Postman API, JIRA, GTMetrix, Browserstuck, Lighthouse. |
| Test case tracking | JIRA, CRM sistems. |
| Test case management. | Microsoft Excel, JIRA, Confluence, Google drive. |
| Test reporting | JIRA. |
| heck list creating | Microsoft Excel, JIRA. |

**9.1Test Environment (browsers):**

9.1.1 Windows : Edge, Chrome (latest), Firefox (latest), Safari (latest)

9.1.2. Mac OS : Chrome (latest), Firefox (latest), Safari (latest).

**10. Bug Life cycle:**

**10. Test schedule up to 04.02.2023**