# Test Plan

**Test Plan:**

Telerik Academy Forum

**Prepared by:**

Simona Diamandieva-Ilieva

**Date:** 15.07.2023

**Table of contents:**

1. Overview
2. Purpose of this document
3. Objective of testing
4. Test base
5. Pre-conditions before starting system test
6. Planning
7. Tasks
8. Entry Criteria
9. Scope
10. Test cases
11. Functionalities to be tested
12. Functionalities not to be tested
13. Testing strategy/Test types
14. Unit/Component testing
15. Integration testing
16. Regression functional UI testing
17. Smoke/Sanity testing
18. Browser compatibility and responsive design testing
19. Graphical design test
20. User Acceptance Testing
21. Resources - Hardware requirements
22. Environment requirements
23. Test Schedule
24. Control Procedures
25. Problem Reporting
26. Change requests
27. Roles and Responsibilities
28. Deliverables
29. Significantly Impacted Departments (SIDs)
30. Risks
31. Approval
32. Tools
33. Exit Criteria

**1. Overview**

a. Purpose of this document

This document describes the test plan for Telerik Academy Forum project which can be found at the url: https://forum.telerikacademy.com/. This website is mainly used by the people involved in learning process: students, teachers, administration etc.

The main idea of the forum is to ask questions, answer them and sharing important news related to academy.

The main purpose of the given test plan is the testing description of the project. This document provides guidance on planned works to be conducted and terms.

Another purpose of this document is to define the strategy, approach, roles and responsibilities of the parties involved in the test process during the construction of Telerik Academy Forum.

It describes what will be tested, by whom and their responsibilities.

The goal of the testing is the thorough verification of the most important features with the typical scripts of its usage. During testing, the different types and approaches will be applied testing. The testing will be based on technical requiremets.

b. Objective of testing

The main purpose of the test process is to verify if the product meets defined requirements, make sure reported bugs are properly solved and to report deviations to stakeholders.

c. Test base

Testing will be executed based on written specifications. These are stored and shared in electronic format.

During the sprint’s specifications will be added and modified.

Other documents or ad-hoc communication such as emails that have been used as input for producing specifications, but these will not be used as input for testing.

d. Pre-conditions before starting system test

The software will be delivered in multiple sprints (development iterations) each lasting 2 weeks.

Each sprint will have several Epics/Stories.

Before testing on an Epic/Story can start, the following conditions should be met.

\*Specifications should be finalized and approved

\*Story should have written acceptance criteria and in scope details

\*Development based on the approved specifications has been finished

\*The Epics/Stories are delivered on a working development test environment with sample test data

Each Story has a testing task added. The tasks may be split into test design (prepare test cases) and test execution, or in some cases other test types such as security test etc.

Part of the “definition of done” of a Story is that the tests have been executed and a limited number of defects are open.

e. Planning

For all planning related see the general project planning.

f. Tasks

1. Testing;

2. Re-testing;

3. Problem reporting and etc.

**2. Entry Criteria**

\* Requirements are defined and approved.

\* Availability of sufficient and desired test data.

\* Testing environment is established.

\* Test cases should be prepared.

\* Resources should be available.

**3. Scope**

a. Test cases

Each test will be executed as part of the applicable sprint if this is possible if not - it will be executed during the next sprint. If defects are found, the bugs will be logged in Jira to be fixed.

After being fixed by developers the defects will be re-tested and depending on their impact on functionality some partial regression tests will also be executed.

Selected sprint test cases will be automated.

These test cases will be used later for automated regression suite.

Detailed test cases will be written for at least high risk and critical functionality, such as registration, login, creation of topic, etc.

Detailed test cases will not be written for low risk and non-critical functionality, such as bold title of topic. Tests executed will be logged and reported upon.

Detailed test cases might not be written for non-functional tests, such as compatibility and security tests. In such cases the test approach will be described, and tests executed are logged and reported upon.

Xray for Jira will be used to store, organize and run manual tests / test cases. The test cases will cover not only the “Happy paths” but also alternative paths to validate that the delivered functionality is built according to the requirements.

b. Functionalities to be tested

\* Login form

\* Active button “New topic”

\* Forum topic creation (as a user)

\* Create а topic with and without registration.

\* Create а topic without title and with long title (more than 256 characters), with special character etc.

\* Create a topic adding different media (picture, music, video, presentation etc.)

\* Create topics with same names.

\* Create topic with empty title, empty description, with long description (more than 32000 characters) and etc.

\* Edit the topic

\* Delete the created topic

\* Create a comment with valid description, with invalid description, with empty description and ect.

\* Edit a comment

\* Delete a comment

\* State transitions of comment and topic

\* Frond-end website

\* etc.

c. Functionalities not to be tested

\* \*\*Security\*\* and \*\*performance\*\* testing are not subject to this agreement. It will be performed by another QA team.

\* Cross-browser testing will not be done on Microsoft Edge(Internet Explorer on older devices). (low risk)

\* Different user roles and managing settings (Will be released but not tested or documented as a functional part of the release of this version of the software.)

\* Register new user

\* Forum topic creation as an admin

\* Bold/italic titles/descriptions in topic or comment

**4. Testing strategy/Test types**

a. Unit/Component testing

\*\*Definition:\*\* The unit or component testing is type of testing in which small part of applications is tested. The main reason of that is to make sure

that this piece of code is working as it's designed to work.

\*\*Participants:\*\* developer team - Martin Petrov, Ivan Markov, Katya Ivanova

\*\*Methodology:\*\* The Unit tests should cover more than 95 % of the newly developed code. Unit test scenario can be reviewed to ensure coverage all relevant cases.

b. Integration testing

\*\*Definition:\*\* Level of software testing where individual units / components are combined and tested as a group.

\*\*Participants:\*\* Developer team and QA team - Martin Petrov, Ivan Markov, Katya Ivanova, Maria Simeonova, Chavdar Iliev

\*\*Methodology:\*\* The Integration tests should cover more than 95 % of all relevant cases.

c. Regression functional UI testing

\*\*Definition:\*\* Regression testing is the selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still works as specified in the requirements.

During each sprint it will be decided which test cases will be automated having in mind possibilities and efficiency. Automated tests will be used for executing automated regression tests and smoke tests.

Test automation will be done on test cases that successfully passed the manual tests and have no blocking or critical issues open.

The goal for the Regression suite is to verify that no regression issues were introduced in the most important functionalities.

The execution of the tests should be under 1 hour so that we can receive rapid feedback on the forum status after each check-in.

Execution should be triggered through Jenkins jobs:

~~~

\* Run On Demand Job

\* CI Test automation Run with Environment locking

\* Nightly Execution

~~~

\*\*Participants:\*\* QA team - Maria Simeonova, Chavdar Iliev

Test results:

The results of the tests executed should be tracked through Jenkins. Automatic emails with Test Results reports should be sent to the whole Team.

\* \*\*Front-end testing\*\*

\* The front-end should follow the design patterns and guidelines supplied by the Designers.

\* The Prototype or wireframe should match the functional front-end, or if not, all discrepancies should be discussed

The front-end should follow the design patterns and guidelines supplied by the Designers

The Prototype or wireframe (if available) should match the functional front-end, or if not, all discrepancies should be discussed

Tests Setup

Automatic Image verification tests should be implemented for the most important pages on the Website.

• Tests should be run Nightly using Jenkins

• Tests should be able to be run on demand

• Test results should be collected in one place and reviewed regularly

Crawler to be run periodically to compare screenshots of all pages + broken links

d. Smoke/Sanity testing

\*\*Definition:\*\* Smoke testing, also called build verification testing or confidence testing, is a software testing method that is used to determine if a new software build is ready for the next testing phase.

This testing method determines if the most crucial functions of a program work but does not delve into finer details.

The goal of smoke testing is to discover simple but severe failures using test cases that cover the most important functionalities of a software.

\*\*Participants:\*\* - QA team - Maria Simeonova, Chavdar Iliev

\*\*Methodology\*\* A limited set of steps will be maintained as “smoke test” to verify a successful deployment of any update to different environments.Smoke tests could be automated to increase accuracy and frequency of execution

e. Browser compatibility and responsive design testing

Browser compatibility and responsive design

The forum will be tested for compatibility with devices and browsers listed in this document.

Since repeating all tests on each of the supported devices/browsers will take to much test effort, a matrix made to specify which tests will be executed on with device/browser combination.

- Conduct all functional testing with the selected browsers

- Do smoke test for each backlog item in each sprint for the selected browsers/platforms.

- Tests with mobile devices will include tests for the different responsive design sizes

\*\*Cross-browser compatibility\*\*

The forum will be tested on the 5 most common browsers.

https://gs.statcounter.com/

~~~

\* Chrome

\* Safari 7 under Mac

\* Firefox

\* Edge

\* Opera

~~~

\*\* The responsiveness of the design will be checked under the following mobile devices\*\*:

~~~

\* Samsung Galaxy A51

\* Apple iPhone 12 Pro Max

\* iPhone 6

\* Apple iPad Pro 12.9 (2020)

\* Tablet Samsung - Galaxy Tab A T290

~~~

f. Graphical design test

The customer [application] layout will be checked on compliance with the approved graphical design. The graphical design will be checked by the graphical designer and the tested during the sprint.

The responsive design breakpoints will be tested to ensure that the layout of the page is supporting desktop, tabled and smartphone devices.

Responsive design will first be tested via the desktop with a browser emulator that simulate different screen sizes. After that selected devices (smartphones and tablets) will be used for verification.

JavaScript and (functional) cookies must be enabled to use all functionality (it is not allowed to use cookie less sessions).

g. User Acceptance Testing

\*\*Definition:\*\* The purpose of acceptance test is to confirm that the system is ready for operational use.

During acceptance test, end-users (customers) of the system compare the system to it's initial requirements and expectations.

\*\*Participants:\*\* One of our customers.

\*\*Methodology\*\*:We will provide the customer script what will contains steps to reproduce the functionality. After that will have a conversation on things he/she likes and which doesn't and

make changes according to new requirements.

**5. Resources**

Hardware requirements:

\* Computers for every member of the team

\* Mobile divices (pointed in this document)

\* Tablets

**6. Environment requirements:**

\*Backend databases.

\*Data required to prep the test bed.

\*The testing server.

\*Relevant network configurations.

\*Test automation frameworks.

\*Robust test documentation tools.

\* Workstation - 10 free spaces in the office for working teams.

**7. Test Schedule**

\* Requerment documents review - 1 day (8 hours)

\* Test planning - 2 days (16 hours)

\* Test design - 3 days (24 hours)

\* Test types - 2 days (16 hours)

\* Test cases - 4 days (32 hours)

\* Staffing and training new test resources - 2 days (16 hours)

\* All Functional testing - 5 days (40 hours)

\* Cross-browser compatibility 4 days (32 hours)

\* Responsive Design - 1 day (8 hours)

\* Regression functional UI testing - 1 hour after each check-in.

\* Front-end testing - 1 day (8 hours)

\* Non-functional testing – 3 days (24 hours)

The product should be delivered in 2 months.

**8. Control Procedures**

a. Problem Reporting

All defects must be submitted in or imported into Jira. Defects are classified as bugs only if:

\*They are clearly described with steps to reproduce, expected result and actual result

\*Have an impact/severity assigned

\*Issues related to graphical design should be provided with a screenshot

\*They can be reproduced

\*They are deviations from agreed specifications or concern overly clear errors

\*Reported in the English language

The following information should be shared at the end of each sprint with the development team and the Product Owner.

This can be in the form of a report, or during a regular meeting via sharing screens of the project management and test management tool.

\*Total number of open bugs, per severity

\*Total number of solved issues, per severity

\*Test progress and coverage, general impression of software quality

b. Change requests

In some cases, submitted bugs are evaluated as changes or additions to existing requirements. In such cases a change of the bug into a Change Issue type will be done, described and implemented by request of the client.

If there is a need for modifications to the software, the must requirements are the follow: write a change request id, person who can requested is only development team leader, dare ot request,

reason for the request, current capabilities of the software, expected capabilities, impact of change, current status of software.

**9. Roles and Responsibilities**

\* Project Manager - Zhana Kioseva (zhana.koiseva@gmail.com)- Ensure project is on-track, prepare status reports, project plan, track project development status, manage changes/issues and resource issues.

\* Technical Lead - Ivan Georgiev (ivan.georgiev@gmail.com) - Provide technical expertise and ensure deliverables are proceeding in the correct technical direction.

\* Services Manager - Sonya Kostadinova (sonya.kostadinova@gmail.com) - Deliver the necessary Business Services

\* Test manager - Kristina Kostova (kristina.kostova@gmail.com) - Coordinates activities, plans, monitors the process, reports, defines methodologies

\* Test analyst - Kristina Kostova (kristina.kostova@gmail.com) - Create a test design, preparate only specifications about test period, preparate test data

\* Manual/Automation tester 1 - Maria Simeonova (maria.simeonova@gmail.com) - implement and execute tests, write test cases, write reports with result

\* Manual/Automation tester 1 - Chavdar Iliev (chavdar.iliev@gmail.com) - implement and execute tests, write test cases, write reports with result

**10. Deliverables**

Major deliverables are:

\* Test plan (this document)

\* Test reports – the frequency depends on project needs

\* Test cases accessible through X-Ray. On request this can be exported to word/excel and shared

\* Test summary reports

\* Test Incident reports

\* Test item transmittal reports

\* Automated test scripts

**11. Significantly Impacted Departments (SIDs)**

\* Development Department

\* QA Department

**12. Risks**

\*Lack of personnel resources when testing is to begin.

\*Lack of availability of required hardware, software, data or tools.

\*Late delivery of the software, hardware or tools.

\*Delays in training on the application and/or tools.

\*Changes to the original requirements or designs.

If some of these events happen the following solutions are required:

\*The test schedule and development schedule will move out an appropriate number of days.

\*The number of test performed will be reduced.

\*The number of acceptable defects will be increased.

\*Resources will be added to the test team.

\*The test team will work overtime.

\*The scope of the plan may be changed.

**13. Approval**

Project manager will be the person who can approve the process as complete.

**14. Tools**

\*Jira - Project and defect tracking tool

\* X-ray for Jira – Test case management

\* Google Chrome

\* Microsoft Edge

\* Apple Safari emulators to test the responsive designs

\* Selenium WebDriver – development of automated test scripts

Additional test tooling could be using this project. The development team will be informed in such cases.

**15. Exit Criteria**

\* All test cases with priority 1 and 2 are passing the given criteria with no failures

\* There are no critical and blocking bugs.

\* All unit and integration tests are passing.

\* Time runs out.