

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 33 Alexander Malinov Blvd. Sofia 1729, Bulgaria |  | Tel.: + 359 2 809 98 50 |  | Team Griffin |

Test Plan

Testing Base Modules and front-end functionalities from Telerik ACADEMY LEARNING SYSTEM (TALS)

Table of Contents

1. [Revision History 3](#_Toc437532056)
2. [Introduction 4](#_Toc437532057)

[Purpose 4](#_Toc437532058)

[Scope 4](#_Toc437532059)

1. [Testing summary 4](#_Toc437532060)

[Scope of testing 4](#_Toc437532061)

[In Scope 4](#_Toc437532062)

[Out of scope 5](#_Toc437532063)

1. [Software Risk Issues 5](#_Toc437532064)
2. [Approach 5](#_Toc437532065)

[Testing Levels 5](#_Toc437532066)

[Component/Unit testing 5](#_Toc437532067)

[Integration testing 5](#_Toc437532068)

[System testing 6](#_Toc437532069)

[Acceptance 6](#_Toc437532070)

[Test Tools 6](#_Toc437532071)

[Meetings 6](#_Toc437532072)

[Measures and Metrics 7](#_Toc437532073)

1. [Item Pass/Fail Criteria 7](#_Toc437532074)

[System testing 7](#_Toc437532075)

[Entry Criteria 7](#_Toc437532076)

[Exit Criteria 7](#_Toc437532077)

1. [Suspension Criteria and Resumption Requirements 7](#_Toc437532078)

[Suspension Criteria 8](#_Toc437532079)

[Resumption Requirements 8](#_Toc437532080)

1. [Test Deliverables 8](#_Toc437532081)

[Before testing phase 8](#_Toc437532082)

[During testing Phase 8](#_Toc437532083)

[After testing Phase 8](#_Toc437532084)

1. [Remaining Test Tasks 8](#_Toc437532085)
2. [Environmental Needs 8](#_Toc437532086)

[Central Test environment 8](#_Toc437532087)

[Client Test environments 9](#_Toc437532088)

1. [Staffing and Training Needs 9](#_Toc437532089)

[Training needs 9](#_Toc437532090)

[Staffing needs 9](#_Toc437532091)

1. [Responsibilities 9](#_Toc437532092)
2. [Schedule 11](#_Toc437532093)
3. [Planning Risks and Contingencies 11](#_Toc437532094)
4. [Approvals 12](#_Toc437532095)
5. [Glossary 12](#_Toc437532096)
6. [References 12](#_Toc437532097)
7. [Index 13](#_Toc437532098)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Description** |
| 24.11.2015 | 1.0 | L. Nikodimov | Test Plan First Draft |
| 25.11.2015 | 1.1 | L. Nikodimov | Introduction, Test Summary, Glossary, References |
| 26.11.2015 | 1.2 | L. Nikodimov | Updated Test Summary, Schedule, Staffing and Training Needs |
| 26.11.2015 | 1.3 | N. Bogomirov | Updated Software Risk Issues and Test Deliverables |
| 26.11.2015 | 1.4 | E. Georgieva | Responsibilities and Planning Risks and Contingencies |
| 26.11.2015 | 1.5 | P. Paunov | Approach, Item Pass/Fail Criteria |
| 27.11.2015 | 1.6 | N. Nenov | Environmental needs and Suspension Resumption Criteria |
| 27.11.2015 | 1.7 | N. Nenov | Approach Testing Levels User Interface Testing and Approach Testing Levels Function/System Testing |
| 27.11.2015 | 1.8 | E. Georgieva | Approach Testing Levels User Interface Testing |
| 03.12.2015 | 1.9 | N. Nenov | Updated Approach Testing Levels, Test Tools, Measures and Metrics, Remaining Test Tasks, Staffing and Training Needs, Item Pass/Fail Criteria |
| 04.12.2015 | 1.10 | E. Georgieva | Updated Introduction and Responsibilities |
| 10.12.2015 | 1.11 | L. Nikodimov | Updated Scope of Testing, Test Tools, Approvals, Schedule, Navigation, Index, fixed small typos |
| 11.12.2015 | 1.12 | E. Georgieva | Updated Approach, Measures and Metrics, Schedule and small fixes |

# Introduction

## Purpose

This document details the testing that will be performed by Team Griffin for the Teamwork Project “Design and implement test plan template for testing main functionalities of TALS (Telerik Academy Learning System)” as a part of the SQA (Software Quality Assurance) Track @ Telerik Academy. It defines the overall testing requirements and provides an integrated view of the project test activities.

Link for test environment provided for TALS [here](http://stage.telerikacademy.com/).

## Scope

The current project is limited to testing the functionality of the Main and Software Academy modules in the admin part of TALS and the functionalities listed under Front-End Functionalities in the section [In Scope](#_In_Scope). Additional testing will be done by other teams to ensure that all system modules work according to requirements.

# Testing summary

## Scope of testing

### In Scope

The document details what is in scope from a testing perspective for the project team:

* **Administration Functionalities**:
  + **Main Modules**
    - Settings
    - Files
    - Pages
    - Feedback Reports
    - Cleanup
    - Calendar
    - Training Rooms
    - Devices
    - Custom Events
    - Moved Lectures
    - Surveys
    - Search Terms
  + **Software Academy Modules**
    - Candidates
    - Additional Documents
    - Candidate Statistics
    - Candidate Form Q&A Statistics
    - Candidate Correlations
    - Candidate Entry Exams
    - Students
    - Seasons
* **Front-End Functionalities:**
  + User Settings
  + Calendar
  + Site Navigation
  + Panels for Last Forum Posts, Last Video Materials, Newest Blog Posts, Incoming Courses, Social Media Buttons
  + Search
  + Application for Software Academy
  + Entry Exam

### Out of scope

The document details what is out of scope from a testing perspective for the project team

* Any modules in the Administration part excluding Main modules and Software Academy modules
* Any front end functionality excluding: User Settings, Calendar, Site Navigation, Panels for Last Forum Posts, Panels for Last Video Materials, Newest Blog Posts, Incoming Courses, Social Media Buttons, Search, Application for Software Academy and Entry exam
* Component/Unit testing
* Integration testing

# Software Risk Issues

Main risk issues associated with the test system are:

* Discrepancy between the expected and the actual system functionality caused by misunderstanding of the original requirements.
* Failures caused by hardware problems of the test environment
* Failures caused by wrong settings of the test environment
* Database crash

# Approach

## Testing Levels

The following levels of testing are recognized as required for the system under test:

### Component/Unit Testing

This step is already performed by the development team and is considered done.

### Integration Testing

This step is already performed by the development team, when new modules were developed and added to the system. QA Team may also perform integration testing to ensure that modules integrate without defect issues from users’ perspective. Such testing will be performed using black-box testing method with manual and automated tests.

### System Testing

This test level will be covered through manual and automated tests

* **Functional testing**
* **User interface testing** – will ensure that users are provided with user role access and navigation in order to use the functionalities of the modules that are in the scope.
* **Validation testing** – will ensure that the modules in scope of testing are capable of performing the tasks they are created for. Validation testing will be performed with use cases converted to test cases.
* **Non-functional testing**
* **Performance testing** – will be done automatically by performing Load and Stress tests. This type of testing will ensure that the response time of all modules in scope of testing is no more than 30 seconds. The actual time depends on the functionality of each of the modules. More specific criteria will be defined in a separate document containing details for the performance testing.
* **Security Testing** – will be done manually/automatically for each module where security risks are recognized. It will include:
* **Functional privileges testing** – only users with admin role privileges may access the admin part
* **Testing the ability to insert malicious code/programs** – automatic testing will ensure that no script/SQL injections are possible on all input fields present in the modules.
* **Regression testing** – after modification of a component automatic tests will be performed on components related directly or indirectly to the modified one.

### Acceptance Testing

The stakeholders and the product owner have decided that actual acceptance testing would be skipped. Instead the test team should deliver a presentation of all testing activities on a final teamwork defense.

## Test Tools

* **Telerik Team Pulse (TTP)** – for bug tracking.
* **Telerik Test Studio (TTS)** – for test automation, performance testing and load testing
* **Visual Studio Team Services (VSTS)** – for test case management
* **Microsoft Visual Studio (VS)** – for test creation
* **Team Foundation Server (TFS)** – for source control ([server link](https://lnikodimov.visualstudio.com/))

## Meetings

The test team will meet every day to evaluate progress to date and to identify error trends and problems as early as possible. The meetings can be from distance using Skype, if it is necessary. The test team leader will meet with product owner once every two weeks.

## Measures and Metrics

The following **bug priority** metrics were outlined by the bug tracking system:

* **1 (Immediate)** – Defects must be resolved as soon as possible, because the defects are affecting the system severely. System/Feature is unusable until the problem is fixed.
* **2 (Next release)** – Defects should be resolved in the normal course of development activities. The problems are not as severe and can wait until the new build/version is created.
* **3 (On occasion)** – Defects that have minor impact on the user or have many workarounds and should be resolved after more severe defects have been fixed.
* **4 (Open)** – Defects that have very low visibility and don’t impact the functionality in any way. These should be resolved if there is time and resources and there are no other higher priority defects present.

The following **bug severity** metrics were recognized by the bug tracking system:

* **Blocking** - Stops the user from using the feature as it is meant to be used. No reasonable workaround
* **1 (Critical)** – Defects that render the system/feature unusable with the possibility of data corruption and no acceptable workaround.
* **2 (High)** – Defects do not result in system/feature termination, but cause the system to produce incorrect, incomplete or inconsistent results with possible workarounds.
* **3 (Medium)** – Defects do not result in termination and do not damage the system/feature usability. The results can be easily obtained with minor workarounds.
* **4 (Low)** - Defects that are related to cosmetic issues and have many workarounds and low visibility to users.

The following **test case priority** metrics are outlined with an objective to reduce the overall number of test cases in the total testing feat.

* **1 (Urgent)** – Test cases that have urgent priority and target very basic tests. The tests have to be executed in any case and must pass, otherwise the delivery date will be affected. They can be launch of components, loading or saving some test documents etc. In another word it is a kind of smoketest.
* **2 (High)** – Test cases that have high priority and target very common functionality that is used by most users. The tests must be executed before the final delivery. They can be information adding, editing or deleting, pictures inserting, creating table or using functions.
* **3 (Medium)** – Test cases that have medium priority and target common functionality that is used by typical experienced users. The tetss which can be executed, only when time permits. They can be page navigating in grid view, sorting, filtering, modifying text style etc.
* **4 (Low)** – Test cases that have low priority and target additional functionality.The tests which can wait and can be executed even after the delivery date.

# Item Pass/Fail Criteria

## System testing

### Entry Criteria

* Unit testing performed for each module
* Integration testing performed for all modules under test
* The testing environment described under section [Environmental needs](#_Environmental_Needs) must be set up and functional
* Design documentation and requirements should be available so the testing team can verify their test
* All team members must have completed the necessary training

### Exit Criteria

* No bugs with critical and high severity left outstanding
* At least 85% of all medium priority test cases should pass successfully
* All high risk areas are tested thoroughly
* System continues to function without blocking defects under anticipated and realistic loads

# Suspension Criteria and Resumption Requirements

### Suspension Criteria

* Critical failure that prevents the application from functioning and no acceptable workarounds were found
* The number of open incidents produces a situation, where further testing has no value
* System or environment downtime

### Resumption Requirements

* The testing will resume when adequate fix for the issue (set of issues) has been provided and deployed in the test environment
* In case of system or environment downtime smoke testing has to be performed and confirmed by the testing team before resuming the testing

# Test Deliverables

During entire project the following test deliverables will be provided:

## Before Testing Phase

* Test plan document
* Test cases documents
* Test design specifications

## During Testing Phase

* Test tools
* Test data
* Error logs

## After Testing Phase

* Test results/reports
* Defect report
* Release notes

# Remaining Test Tasks

Below is a link to all test cases, which are pending to be covered by tests:

[Visual Studio Team Services - Telerik Academy Learning System](https://lnikodimov.visualstudio.com/DefaultCollection/Telerik%20Academy%20Learning%20System/_dashboards)

# Environmental Needs

## Central Test Environment

***OS***: Windows Server 2012 R2

***Web Server***: MS IIS 8.5

***SQL Server***: MS SQL Server 2012

***CPU:*** Intel Xeon CPU @ 3.10 GHz (Quad Core)

***RAM***: 8GB

***Storage***: 100GB

## Client Test Environments

* Windows environment with latest stable versions of following browsers
* Google Chrome
* Mozilla Firefox
* Safari
* Microsoft Internet Explorer

# Staffing and Training Needs

This section describes any specific training and staffing needs that are required to deliver the test acceptance plan. The Responsibilities section and Approach section of this document will affect the staffing and training needs.

The main benefit of this section is that it communicates the resource requirements and seeks the approval for these resources to support delivery of the test plan.

## Training Needs

The following training needs are recognized for all team members:

* Training for Telerik Team Pulse (TTP)– already completed basic functionality training
* Training for Telerik Test Studio (TTS)– already completed basic functionality testing
* Training for Microsoft Visual Studio Team Services (VSTS) – all members can already use the system
* Training for design and execution of Performance tests
* Training for design and execution of Load tests

## Staffing Needs

Staffing for the duration of the project is fixed and the team will be with size of 5 members acquainted with the Testing Tools that the team will use (described under section Test Tools). Furthermore all team members should have knowledge of the system under test, of the modules they are assigned to test (described under section [Responsibilities](#_Responsibilities)) and how the different modules interact with each other.and the team leader will be different for every sprint.During each sprint a different team member will have team leader responsibilities

# Responsibilities

This section describes team members’ responsibilities regarding functionalities described in this document. This issue includes all areas of the plan.

* **Nikola Nenov – Team Member**
  + **Main Modules – Administration Functionalities**
    - Calendar
    - Training Rooms
    - Devices
    - Custom Events
    - Moved Lectures
  + **Front-End Functionalities**
    - Calendar
* **Nikola Bogomirov – Team Member**
  + **Main Modules – Administration Functionalities**
    - Cleanup
    - Surveys
    - Search Terms
  + **Front-End Functionalities**
    - Search
* **Lyudmil Nikodimov – Team Member**
  + **Main Modules – Administration Functionalities**
    - Settings
    - Files
    - Pages
    - Feedback Reports
  + **Front-End Functionalities**
    - User Settings
* **Plamen Paunov – Team Member**
  + **Software Academy – Administration Functionalities**
    - Candidate Statistics
    - Candidate Form Q&A Statistics
    - Students
    - Seasons
  + **Front-End Functionalities**
    - Site Navigation
    - Panels for Last Forum Posts, Last Video Materials, Newest Blog Posts, Incoming Courses, Social Media Buttons
* **Emiliya Georgieva – Team Member**
  + **Software Academy – Administration Functionalities**
    - Candidates
    - Additional Documents
    - Candidate Correlations
    - Candidate Entry Exams
  + **Front-End Functionalities**
    - Application for Software Academy
    - Entry Exam

# Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Estimation** | **Comments** |
| Test plan preparation | 25.11.2015 | 27.11.2015 | 3d |  |
| Test plan presentation to stakeholders | 27.11.2015 | 27.11.2015 | 1d |  |
| Fulfill training and environmental needs | 30.11.2015 | 02.12.2015 | 3d |  |
| Test cases preparation | 30.11.2015 | 04.12.2015 | 5d |  |
| Test scripts preparation | 08.12.2015 | 11.12.2015 | 4d |  |
| Smoke testing |  |  |  |  |
| System testing |  |  |  |  |
| Load testing |  |  |  |  |
| Performance testing |  |  |  |  |
| Security testing |  |  |  |  |
| Test closure activities |  |  |  |  |
| Final teamwork defense | 05.02.2016 | 05.02.2016 | 1d |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Teamwork Project Task** | **Start Date** | **Finish Date** | **Team Lead** |
| Week 3 - Test Case Management System + Test Plan | 25.11.2015 | 27.11.2015 | Nikola Nenov |
| Week 4 - Academy's System - Test Cases | 30.11.2015 | 04.12.2015 | Emiliya Georgieva |
| Week 5 - Automation tests with Test Studio | 08.12.2015 | 11.12.2015 | Lyudmil Nikodimov |
| Week 6 – Automation tests with Telerik Testing Framework | 14.12.2015 | 18.12.2015 | Plamen Paunov |
| Week 7 - Automation tests with Telerik Testing Framework | 21.12.2015 | 08.01.2016 | Nikola Bogomirov |

# Planning Risks and Contingencies

What are the overall risks to the project with an emphasis on the testing process?

* Lack of personnel resources when testing is to begin.
* Lack of availability of required hardware, software, data or tools.
* Delays in training on the application and/or tools.
* Changes to the original requirements or designs.
* Complexities involved in testing the applications – for example not working module of the TALS system.

Possible solutions of expected problems:

* If personnel resources are missing for any team member, other team members should prepare meeting and decide how to be useful.
* If any required hardware, software, data or tools are missing, the member may use another member resources until the team think up some new solutions.
* If there is delay in training, the team members may find on their own resources and documentation and explore the tool or application on a meeting.
* If the original requirements change - the test schedule and development schedule will move out an appropriate number of days (this is not recommended since the team have fixed delivery date).
* If any module is unavailable or not working, on the automated tests can be set priority and run manually.

# Approvals

The final approval of all performed testing activities will be done by the stakeholders and the product owner as a part of final teamwork presentation.

# Glossary

The following acronyms and terms have been used throughout this document:

|  |  |
| --- | --- |
| **Term/Acronym** | **Definition** |
| TALS | Telerik Academy Learning System 5.7 (initial build 20151119.44a964b) |
| TTP | Telerik TeamPulse Version 2014.2.815.0 |
| TTS | Telerik TestStudio Ultimate Version 2015.3.1015.0 |
| VSTS | Microsoft Visual Studio Team Services |
| TFS | Team Foundation Server |
|  |  |
|  |  |