Abstract

This document contains the test plan for modules “Calendars”, “Kids Academy” and “Teacher”

Test Plan

Telerik Academy Learning System

Document History and Approvals

1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision # | Revision Date | **Description of Change** | **Author** |
| 1.0 | 23 Sep 2014 | Initial version | Team Fig |
| 1.1 | 01 Oct 2014 | Structural improvements;  Testing tasks added | Vassil Kolarov |
| 2.0 | 09 Oct 2014 | Scope, strategy, schedule, environment sections improvements | Rossitza Nikolova |
| 2.1 | 24 Oct 2014 | Test case priority table added; corrected styling inconsistencies;  Removed redundancies; | Rossitza Nikolova |
| 3 | 29 Oct 2014 | Schedule table extended with time estimation column;  Metrics and reports added | Rossitza Nikolova |
| 3.01 | 29 Oct 2014 | Formatting corrections | Vassil Kolarov |

1. Plan Approvals

|  |  |
| --- | --- |
| **Signature:** |  |
| **Name:** |  |
| **Role:** |  |
| **Date:** |  |

|  |  |
| --- | --- |
| **Signature:** |  |
| **Name:** |  |
| **Role:** |  |
| **Date:** |  |

Contents

[1. Introduction 3](#_Toc402351155)

[1.1 Project Overview 3](#_Toc402351156)

[1.2 Test Scope 3](#_Toc402351157)

[1.2.1 Testing Goals 3](#_Toc402351158)

[1.2.2 Test Objectives 4](#_Toc402351159)

[2. Test Items 4](#_Toc402351160)

[3. Features To Be Tested 4](#_Toc402351161)

[4. Features Not To Be Tested 5](#_Toc402351162)

[5. Test Strategy 5](#_Toc402351163)

[6. Test Process 6](#_Toc402351164)

[6.1 Test Deliverables 6](#_Toc402351165)

[6.2 Test Process Workflow 7](#_Toc402351166)

[6.3 Pass / Fail Criteria 7](#_Toc402351167)

[6.3.1 Suspension Criteria 7](#_Toc402351168)

[6.3.2 Resumption Criteria 7](#_Toc402351169)

[6.4 Test Case Priority Classification 7](#_Toc402351170)

[6.5 Defect Severity Classification 8](#_Toc402351171)

[6.6 Metrics and Reports 8](#_Toc402351172)

[6.7 Schedule 8](#_Toc402351173)

[6.8 Risks and Assumptions 9](#_Toc402351174)

[6.9 Tools 10](#_Toc402351175)

[7. Test Environment Requirements 10](#_Toc402351176)

[7.1 Hardware 10](#_Toc402351177)

[7.2 Software 11](#_Toc402351178)

[8. Testing Team 11](#_Toc402351179)

# Introduction

The following Software Test Plan is designed to prescribe the overall test scope, strategy, resources and schedule of all activities involved in the process of testing the *Telerik Academy Learning System*. The plan describes the modules and features to be tested, the types of testing to be conducted, the personnel responsible for testing, and the risks associated with the test process. This document is subject to continuous updating and modification during all test stages.

## **1.1 Project Overview**

Telerik Academy Learning System is a web application used to manage the whole teaching process at Telerik Academy. It provides access to students and courses information, learning materials, video lectures related to a certain course. The system allows homework submission and evaluation, subscription to courses and exams, access to the forums, etc.

Overview of the application modules to be tested:

* + The Calendars module is designed to provide information and management features for official holidays, future events (e.g. lectures and exams)
* The Kids Academy module is designed to provide global management features for the teaching process in the Kids Academy (Kids Academy is a part of Telerik Academy)
* The Kids Academy registration form allows users to register in the application with specific restrictions

## **1.2 Test Scope**

### **1.2.1** **Testing Goals**

The overall goals in testing the application include validating the functional suitability, usability, reliability and security:

* Suitability / Functional correctness
  + The degree to which the application meets the functional requirements
  + The degree to which the application meets the user needs and expectations
* Usability
  + The degree to which the application features are operable and effective
* Reliability
  + The degree to which the application avoids unexpected and unacceptable behavior under normal conditions
* Security / Access Control
  + Verification that actions and data are available only to those users with correct authorization

### **1.2.2** **Test Objectives**

The quality objectives of testing the application are to ensure comprehensive validation of the business and software requirements, and to gain confidence in the overall level of quality of the application by performing the following activities:

* Finding anomalies and / or defects caused by development and / or design errors
* Managing the defect tracking process
* Preventing future defects by extensive communication with the development team and the project manager
* Designing, documenting and organizing test cases to validate and verify both the functional and non-functional quality attributes listed in testing goals section
* Making the tests repeatable for use in regression testing during the project lifecycle
* Providing test metrics and test summary reports
* Conducting continuous test control and adjusting the test plan to newly introduced conditions

# Test Items

* Calendars module
* Kids Academy module
* Kids Academy registration form
* Kids Teacher module

# Features To Be Tested

The following is a list of the areas to be focused on during testing of the application:

* Calendars module
  + Calendar
  + Official holidays
  + Events
  + Rescheduled lectures
* Kids Academy module
  + Schools
  + School types
  + Kids in schools
  + Contests
  + Entry tests
  + Candidates
  + Documents
* Kids Academy registration form
* Kids Teacher module
  + Kids in school

# 4. Features Not To Be Tested

All other modules and features of Telerik Academy Learning System, different than the above mentioned are in the other testing teams’ domains.

# 5. Test Strategy

The test strategy is based on the assumption that comprehensive unit testing has already been done by the developers team. The testing activities are planned to be performed from a black-box perspective, not based on any knowledge of internal design or code.

Tests cases will be prioritized, and those with priority 1 and 2 will be automated.

Regression testing will be performed after each application build. Re-testing will be performed to ensure that a previously reported defect has been successfully fixed.

In addition to the considerations provided for each test type below, testing should only be executed using known, controlled database, in a secured test environment.

The following test types are planned to be implemented:

|  |  |  |
| --- | --- | --- |
| Test Type | Details | Exit criteria |
| Functional System Testing | * + Functional system tests will be created to cover at least 90% of the functions of each module   + Testing each function with valid and invalid data   + Validation that the expected results occur when valid data is used   + Validation that the appropriate error / warning messages are displayed when invalid data is used   + Tests with priority 1 and 2 are to be automated in such a repeatable manner as to allow for regression testing | * + All planned tests have been executed   + No open critical defects   + No blocking defects   + At least 85% test pass rate |
| Usability Testing | * + Manual usability tests will be designed, executed for each module | * + The average new user is able to find out how to complete their tasks within 5 minutes   + The average user is able to understand newly introduced functionality within 5 minutes |
| Security Testing | * + Access control testing will be performed to verify that users who log in with the following roles are only granted access to the correct data and functionalities:   + **Administrator** is unable to access the Kids Academy module   + **Administrator** is unable to access the Kids Teacher module   + **KidsAdministrator** is able to access the Kids Academy module   + **KidsAdministrator** is unable to access the Teacher module   + **KidsTeacher** is unable to access the administration module   + **KidsTeacher** is able to access the Kids Teacher module | * + All planned tests have been executed   + All identified defects with severity other than *Low* have been closed |
| Performance Testing | * + Verification of the application response time under a the following two conditions:   + Normal anticipated load   + Anticipated medium to worst case load   + Reusing the functional tests   + Modifying data files (to increase the number of transactions) or modify tests to increase the number of iterations each transaction occurs   + Tests should be run on one machine (best case to benchmark single user, single transaction) and be repeated with multiple clients (virtual or actual) | * + Single user: Successful completion of the test scripts without any failures and within the expected / required time allocation   + Multiple users: Successful completion of the test scripts without any failures and within acceptable time allocation |

# 6. Test Process

## 6.1 Test Deliverables

* Test plan
* Test cases
* Automated test suites
* Defect reports
* Weekly test execution reports
* Test closure report

## 6.2 Test Process Workflow

1. Test planning and control
2. Test analysis and design
   1. Develop test suites
   2. Identify and describe test cases
   3. Identify and structure test scripts
   4. Review the test coverage
3. Test implementation
   1. Record or program test scripts
4. Test execution
   1. Execute test scripts or manual test steps
   2. Evaluate the execution of the test
   3. Recover from halted test
   4. Verify the results
   5. Investigate unexpected results
   6. Log defects
5. Test results evaluation and reporting
   1. Evaluate test case coverage
   2. Analyze defects
   3. Determine if test completion criteria and success criteria have been achieved
   4. Create test evaluation reports

## 6.3 Pass / Fail Criteria

### 6.3.1 Suspension Criteria

1. Not all interface function necessary to begin testing are available**.**
2. Missing prerequisites required to initiate testing process, e.g. missing system functionality to create test items necessaryfor further testing.
3. Blocking bugs.

### 6.3.2 Resumption Criteria

1. Missing interface functions necessary to begin testing are available.
2. Missing prerequisites required to initiate testing process are available.
3. Blocking bugs have been fixed.

## 6.4 Test Case Priority Classification

|  |  |
| --- | --- |
| Priority | Impact |
| 1 (Top) | * This test case is classified as top priority * Covers some major functionality with no workaround * Most probably, it is supposed to be executed most often * Subject of automation |
| 2 (High) | * This test case is classified as high priority * Covers some major functionality with few workarounds * Subject of automation |
| 3 (Medium) | * This test case is classified as medium priority * Covers some non-critical functionality with several workarounds * Subject of manual execution |
| 4 (Low) | * This test case is classified as low priority * Covers some functionality with little impact on the users * Subject of manual execution |

## 6.5 Defect Severity Classification

|  |  |
| --- | --- |
| Severity | Impact |
| 1 (Blocking) | * This defect prevents functionality from being used, there is no work-around, and blocking progress on multiple fronts |
| 2 (Critical) | * This defect is critical enough to crash the system, cause file corruption, or cause potential data loss * It causes an abnormal return to the operating system (crash or a system failure message appears) * It causes the application to hang and requires re-booting the system |
| 3 (High) | * It causes a lack of vital program functionality with workaround |
| 4 (Medium) | * This defect will degrade the quality of the application. However there is an intelligent workaround for achieving the desired functionality - for example through another screen * This defect prevents other areas of the product from being tested. However other areas can be independently tested |
| 5 (Low) | * There is an insufficient or unclear error message, which has minimum or no impact on product use |

## 6.6 Metrics and Reports

* Defects by module and severity
* Number of test cases per module
* Number of automated test cases and their current status

## 6.7 Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint | Test Manager | Task | Estimated Time  (Hours) |
| 26 Sept 2014 – 02 Oct 2014 | N/A | Initialize test planning | 10 |
| Analyze testing conditions | 0.5 |
| Design test cases in MS Excel | 1.5 |
| Implement test cases | 1 |
| 03 Oct 2014 – 09 Oct 2014 | Natalia Dragomirova | Design test cases continued | 8 |
| Implement test Cases in MS Test Manager | 16 |
| Initialize test execution | 2 |
| Detect and report defects | 0.5 |
| Automation using Telerik Test Studio | 16 |
| Test control | 3 |
| Prepare test execution reports | 1 |
| 10 Oct 2014 – 16 Oct 2014 | Rossitza Nikolova | Test automation using Telerik Test Studio | 24 |
| Test automation with Telerik Testing Framework | 5 |
| Test control | 3 |
| Prepare test execution reports | 1 |
| 17 Oct 2014 – 23 Oct 2014 | Vassil Kolarov | Test automation with Telerik Testing Framework | 30 |
| Test control | 3 |
| Prepare test execution reports | 1 |
| 24 Oct 2014 – 30 Oct 2014 | Vassil Kolarov | Test automation with Telerik Testing Framework | 30 |
| Test control | 3 |
| Prepare test execution reports | 1 |
| 31 Oct 2014 – 06 Nov 2014 | Mihail Vakov | Test automation with Telerik Testing Framework | 20 |
| Execute security testing | 10 |
| Test control | 3 |
| Prepare test execution reports | 1 |
| 07 Nov 2014 –13 Nov 2014 | Vladimir Vassilev | Test automation with Telerik Testing Framework | 8 |
| Execute usability tests | 4 |
| Test control | 3 |
| Prepare test execution reports | 1 |
| Evaluation of exit criteria | 5 |
| 14 Nov 2014 –20 Nov 2014 | N/A | Evaluation of exit criteria | 8 |
| Prepare test closure report | 8 |

## 6.8 Risks and Assumptions

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Prob. | Impact | Mitigation Plan |
| SCHEDULE  Testing schedule is tight. If the start of the testing is delayed due to design tasks, the test cannot be extended beyond the usability testing scheduled start date. | Low | High | The testing team can control the preparation tasks (in advance) and the early communication with involved parties.  Some buffer has been added to the schedule for contingencies, although not as much as best practices advise. |
| DEFECTS  Defects are found at a late stage of the cycle or at a late cycle; defects discovered late are most likely be due to unclear specifications and are time consuming to resolve. | High | High | Defect management plan is in place to ensure prompt communication and fixing of issues. |
| SPECIFICATIONS  Absence of specification may lead to inconsistencies with the requirements and related defects | High | High | The project manager has been asked to provide the specification.  There are defects that can be raised during testing because of unclear document specification. These defects can yield to an issue that will need time to be resolved. |
| ENVIRONMENT  Non-availability of Independent test environment and accessibility | Low | High | Due to non-availability of the environment, the schedule gets impacted and will lead to delayed start of Test execution. |
| NEW ISSUES  Delayed testing due to new issues | Medium | High | During testing, there is a chance that some “new” defects may be identified and may become an issue that will take time to resolve.  There are defects that can be raised during testing because of unclear document specification. These defects can yield to an issue that will need time to be resolved.  If these issues become showstoppers, it will greatly impact on the overall project schedule.  If new defects are discovered, the defect management and issue management procedures are in place to immediately provide a resolution. |

## 6.9 Tools

|  |  |
| --- | --- |
| Activity | Tool |
| Project Management | * Telerik TeamPulse * Microsoft Office |
| Test Management | * Microsoft Test Manager * Microsoft Visual Studio Online |
| Test Design | * Microsoft Office |
| Incident Management | * Telerik TeamPulse |
| Test Automation | * Telerik Test Studio * Telerik Testing Framework * Microsoft Visual Studio |

# 7. Test Environment Requirements

The test environment is accessible on the following address: <http://test.telerikacademy.com/>

## 7.1 Hardware

* Intel (R) Xeon (R) CPU @ 3.10GHz Quad Core
* RAM 8GB
* HDD 100GB

## 7.2 Software

* Windows Server 2008 R2
* IIS7
* MS SQL Server 2008 R2

# 8. Testing Team

*Team Fig* consists of 5 students from Telerik QA Academy 2014. The test manager’s role is planned to be shifted over to every team member throughout the test process as specified in the schedule. All team members are listed in alphabetical order:

* Mihail Vakov
* Natalia Dragomirova
* Rossitza Nikolova
* Vassil Kolarov
* Vladimir Vassilev