QA Team “Apple”

Telerik Academy 2014

**Evaluation & Teamwork Modules**

**Test Plan**

Version 2.0

October 1, 2014

# Table of Contents

[Table of Contents 2](#_Toc399951653)

[1. INTRODUCTION 4](#_Toc399951654)

[1.1. Project Background 4](#_Toc399951655)

[1.2. Objectives 4](#_Toc399951656)

[1.3. Testing Strategy 4](#_Toc399951657)

[1.4. Scope 4](#_Toc399951658)

[1.5. Reference Material 5](#_Toc399951659)

[2. FEATURES TO BE TESTED 5](#_Toc399951660)

[3. FEATURES NOT TO BE TESTED 5](#_Toc399951661)

[4. APPROACH 5](#_Toc399951662)

[4.1. Functional Testing 6](#_Toc399951663)

[4.2. User Interface Testing 6](#_Toc399951664)

[4.3. Integration Testing 6](#_Toc399951665)

[4.4. Acceptance Testing 6](#_Toc399951666)

[4.5. System Testing 7](#_Toc399951667)

[4.6. Performance Testing 7](#_Toc399951668)

[4.7. Load Testing 7](#_Toc399951669)

[4.8. Stress Testing 7](#_Toc399951670)

[4.9. Regression Testing 7](#_Toc399951671)

[5. MEASURES AND METRICS 8](#_Toc399951672)

[6. SYSTEM TEST ENTRANCE / EXIT CRITERIA 8](#_Toc399951673)

[6.1. Entrance criteria 8](#_Toc399951674)

[6.2. Exit criteria 9](#_Toc399951675)

[7. PASS / FAIL CRITERIA 9](#_Toc399951676)

[7.1. Suspension Criteria 9](#_Toc399951677)

[7.2. Resumption Criteria 9](#_Toc399951678)

[7.3. Approval Criteria 9](#_Toc399951679)

[8. TESTING PROCESS 10](#_Toc399951680)

[8.1. Test Deliverables 10](#_Toc399951681)

[8.2. Testing Tasks 10](#_Toc399951682)

[8.3. Responsibilities 10](#_Toc399951683)

[8.4. Schedule 10](#_Toc399951684)

[9. ENVIRONMENTAL REQUIREMENTS 11](#_Toc399951685)

[9.1. Hardware 11](#_Toc399951686)

[9.2. Software 11](#_Toc399951687)

[9.3. Security 11](#_Toc399951688)

[9.4. Tools 12](#_Toc399951689)

[9.5. Risks and Assumptions 12](#_Toc399951690)

[10. CHANGE MANAGEMENT PROCEDURES 13](#_Toc399951691)

[11. PLAN APPROVALS 13](#_Toc399951692)

[12. VERSION HISTORY 14](#_Toc399951693)

# INTRODUCTION

This document describes the procedure and expectations for testing module “Evaluation” in the student system being developed by Telerik Academy institution.

## Project Background

The existing system allows the students and administrators to evaluate homework, exams, teammates, etc. in courses they participate. This project is intended to provide testing for the modules “Evaluation” and “Teamwork” in the [Telerik Student System](http://test.telerikacademy.com/).

## Objectives

The objective of this test plan is to ensure a high level of confidence in the correctness and usefulness of the project deliverables.

## Testing Strategy

The strategy for testing the “Evaluation” and “Teamwork” modules of the Telerik Academy Student System is a combination of automated and manual tests. The components of the system that don't involve user interaction will be tested automatically. The components involving user interaction will be tested manually by the developers/ testers.

We will use Microsoft Test Manager to design the test cases.

## Scope

The scope of the current project is limited to the “Evaluation” and “Teamwork” modules of the student system and to the provided requirements. Testing is a very “dependent” activity and this is why test plans must be developed for each module of system to ensure its overall usefulness and correctness.

## Reference Material

* [Evaluation and Teamwork modules requirements (ETWMR)](https://docs.google.com/document/d/19bZ8dV3cdyiHQioqacxgeQHqeTTlHC-5o_NBYD7Cxj4/edit?usp=sharing)

# FEATURES TO BE TESTED

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

* Homework Upload / Re-upload
* Homework Comments and Grading
* Homework Evaluation
* Exam Sign-up
* Exam Upload / Re-upload
* Exam Results / Exam Peer Review
* Exam Evaluation
* Teamwork Sign-up
* Teamwork Upload / Re-upload
* Teammate Grading

# FEATURES NOT TO BE TESTED

* Modules that the tested modules are dependent on and are assigned to other teams.

# APPROACH

The testing will be done manually until the site is sufficiently stable to begin developing automatic tests. The testing will cover the requirements and efficiently reduce the risk.

## Functional Testing

The objective of this test is to ensure that each element of the application meets the functional requirements of the business as outlined in the Requirements Documentation (ETWMR).

This stage will also include **Validation Testing** - which is intensive testing of the designed front-end fields - valid, invalid and limit data input.

## User Interface Testing

The goal of UI testing is to ensure that the UI provides the user with the appropriate access and navigation through the functions of the target-of-test. In addition, UI testing ensures that the objects within the UI function work as expected and conform to corporate or industry standards.

## Integration Testing

Integration tests will exercise the interaction of the “Evaluation” module with other modules that it depends on. The goal is to ensure that they work nicely together.

## Acceptance Testing

This test, which is planned and executed by testers, ensures that the system operates in the manner expected. Testers will provide feedback regarding changes, which must be implemented to the functionality and the UI. Feedback will be provided in the form of verbal communication at meetings and via email reports as necessary. All major changes will be documented in revised versions of the test plan.

## System Testing

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it.

## Performance Testing

Performance testing will be conducted manually. It will be done using black-box testing method. Testers will interact with the user interface to the system and determine whether or not the system responds in a reasonable time. Reasonable is defined as the amount of time a data entry person would expect the system to respond in. The tests will be performed on the most common used browsers (IE, Chrome, Firefox, Safari, Opera).

## Load Testing

Load testing will test the behavior of the system when an increasing number of users work with it at the same time.

## Stress Testing

Stress testing will test system behavior when overloaded. This test is particularly important for exam uploading.

## Regression Testing

When a change is made to the system, all test cases for all components relating (directly or indirectly) to the modified component will be re-executed. The design is to execute all tests necessary to ensure no regression occurs but not to needlessly expend resources on unrelated tests.

# MEASURES AND METRICS

* Test coverage
* UI – buttons, menus, dropdowns, forms, etc.
* Coverage percent of the required features
* Defects
  + Ratio between total found and total fixed bugs
* Priority metrics
  + 1 – Immediate
  + 2 – Next release
  + 3 – On occasion
  + 4 – Open
* Severity metrics
  + 1 – Blocking
  + 2 – Critical
  + 3 – High
  + 4 – Medium
  + 5 – Low

# SYSTEM TEST ENTRANCE / EXIT CRITERIA

## Entrance criteria

* The testing environment described in the environment requirements must be set up and functioning properly.
* All the necessary documentation, design and requirements information should be available that will allow the testers assure the system’s correct behavior.
* All the personnel involved in the testing must have the needed training to use the tools for the testing.
* Prepare appropriate test data or database.

## Exit criteria

* Certain level of requirements is achieved.
* No high-priority or severe bugs are left outstanding.
* All high-risk areas have been fully tested.
* The schedule has been achieved.
* The budget has been spent.

# PASS / FAIL CRITERIA

The entrance criteria's for each phase of testing must be met before the next phase can commence. Now the criteria’s for pass and fail are given below.

## Suspension Criteria

Test case execution will be suspended if a critical failure that impedes the ability or value in performing the associated test(s) is discovered.

## Resumption Criteria

Test case execution will be resumed when the problem causing suspension has been fixed. All test cases that deal with the modified part of the project will be re-executed.

## Approval Criteria

The results of each test case will be considered “approved” if the results meet the expected results description in the test case.

# TESTING PROCESS

## Test Deliverables

A test report will be included in the project deliverables. This report will contain the set of test cases, a history of all formal test executions and a summary of the final state of the test suite.

## Testing Tasks

* Develop Test Cases
* Develop scripts for the automated tasks
* Execute tests
* Report defects
* Complete test report
* Manage change

## Responsibilities

All developers are responsible for the completion of all testing tasks.

The members of the team are responsible for approving the Test Plan and the Test Cases. They are also responsible for critiquing the demonstrations and final acceptance of all work products.

## Schedule

|  |  |  |
| --- | --- | --- |
| Task | Deliverable | Week Performed |
| Develop test cases | Test Cases document | 26.09.14 - 03.10.14 |
| Develop scripts for automated testing | Test scripts | 03.10.14 - 10.10.14 |
| Prepare testing database |  | 10.10.14 - 17.10.14 |
| Execute tests | Daily Test Reports | 17.10.14 - 24.10.14 |
| Report defects | Weekly Bug Reports | 24.10.14 - 31.10.14 |
| Complete test report | Test Case Report | * + 1. - 07.11.14 |

# ENVIRONMENTAL REQUIREMENTS

## Hardware

* 1 Microsoft Windows PC computer with a broadband connection
* 1 Microsoft Windows PC with a broadband connection to the PC above – to ensure server reliability

## Software

* 1 installation of Windows XP or above
* 1 installation of Windows Server
* 1 browser support – Chrome, Firefox, Safari, Opera, IE
* 1 Microsoft Office (Word, Excel) for reports and defect tracking

## Security

The test environment has a specific security requirement – administrator access has to be provided.

## Tools

* Selenium Web Driver – for testing
* Microsoft Test Manager 2013 – for test case management system



## Risks and Assumptions

The main risk would be the dependency on the other modules in the system. The module that we are testing is dependent on the functions and data from other modules, which are assigned to other teams. Communication between teams’ members will be important to keep the workflow going.

Another risk would be that the system is internet-based and if the server is down we will not be able to perform the testing of the module which is assigned to Team “Apple”.

# CHANGE MANAGEMENT PROCEDURES

The procedure for changing the test plan is as follows:

* Propose the change to the software development team. This can be done at a team meeting or via email. The team will discuss the proposal and either reject it, accept it, or accept it with modifications.
* If the team accepts the proposal, then any agreed upon modifications will be implemented and the test plan is updated.

# PLAN APPROVALS

|  |  |  |
| --- | --- | --- |
| Name | Signature | Date |
| Kiril Todorov |  |  |
| Vanina Nenova |  |  |
| Svetlin Nyagolov |  |  |
| Valeria Dimitrova |  |  |
| Stanislav Iliev |  |  |
| Mladen Mladenov |  |  |

# VERSION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Reason for Change | Revised By | Date |
| 1.0 | Initial Test Plan | Team Apple | 26.09.14 |
| 2.0 | Major rework after the first review | Team Apple | 03.10.14 |
|  |  |  |  |
|  |  |  |  |