Telerik Academy Learning System

Test plan

Project URL:

***http://stage.telerikacademy.com/***

*Document Revision History*

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Comment |
| 23.11.2015 | 1.0 | I. Havalyova | Initial version. |
| 24.11.2015 | 1.1 | D. Dechev |  |
| 25.11.2015 | 1.2 | D. Panayotov | Added performance Testing part. |
| 25.11.2015 | 1.3 | Y. Yosifov | Formatting changes. Added Functional Testing part. |
|  |  |  |  |
|  |  |  |  |

**Table of contents**

1. Team members
2. Introduction
3. Objective
4. Scope
5. Testing process and methodology
6. Test Strategy
   1. Testing types
      1. Black box testing
      2. UI Testing
      3. Functional testing
      4. System testing
      5. Performance testing
      6. Stress testing
      7. Security and Access control testing
      8. User acceptance testing
   2. Tools
7. Test environment
8. Entry criteria
9. Exit criteria
10. **Team members**

|  |  |  |
| --- | --- | --- |
| Name | User | Role |
| Decho Dechev | decho.D | QA |
| Dimitar Panayotov | The.Bager | QA |
| Ilvie Havalyova | havaliova | QA |
| Yane Yosifov | Curiosity | QA |
| Ivan Zhelev | Ivan.jelev.16 | QA |

1. **Introduction**

Telerik Academy Learning System is student system developed for the purpose of training at the Telerik Academy. It is built and maintained by the Academy of Telerik centralized information system, which stored information about training courses, participants in the Telerik Academy, their results in the courses (homework, projects, assessments and achievements), schedules classrooms, courses and workshops, and other information related to teaching and activities in the Academy.

1. **Objective**

Objective of this test plan is to ensure that the functionalities of the Telerik Academy Student System listed in article 4a of this document are tested. The top priority functionalities will be “Login” and “Registration”. Medium priority will be the “Courses enrolment”, “Access to lectures/presentations/home works” and “Changing the form of training attendance/online”. Low priority are “Friends” and “Messages”. Users must be able to register and login with a user and not to be able to login or change another user account. Users must have the opportunity to enroll to a course and access all the courses resources and eventually switch from onsite to online education. Users must be able to invite other users to be friends and if the friendship is accepted they must be able to exchange messages.

1. **Scope**

|  |  |
| --- | --- |
| Functions to be tested by the team | |
| Users | Admin |
| Changing the form of training - attendance / online | Front |
| Access to lectures / presentations / home works | Front |
| Courses enrollment | Front |
| Registration | Front |
| Log in | Front |
| Friends | Front |
| Messages | Front |

1. **Assumptions**

The web application to be tested is located at <http://stage.telerikacademy.com/> and is in complete code freeze, ready for testing. The testing is going to be performed by five independent teams of quality assurance trainees. Each of the teams will have responsibility to test a set of functionalities. The cost of testing will be the cost of the trainer staff and the infrastructure to perform the training. The tools used will be either open source or Telerik products.

1. **Risk Analysis**

Major risk for the team test results is lack of organization and lack of competence from the trainees. The teams will be self-organized and weekly reviews of each team’s progress will be performed in order to minimize these risks and provide the team with guidance.

1. **Test strategy** 
   1. **Functional testing**

***Links***

* *Test all internal links.*
* *Test links used to send message to admin or other users.*
* *Check for orphan pages.*
* *Check for broken links.*
* *Check links for indication of a link action that can be performed or was performed (i.e. color different from normal text).*

***Forms***

* *Check validations on each field.*
* *Check for the default values of fields.*
* *Confirm inputs and boundary values for all fields.*
* *Confirm options to create forms if any, delete, view or modify the forms.*
* *Check different user types and roles.*
* *Data*
* *Check data accuracy and integrity*
* *Check data input validation*

***Cookies and sessions***

* *Test enabling or disabling of cookies in different browsers’ options.*
* *Test for cookie encryption before writing to user machine.*
* *Check for login sessions and user statistics after session end.*
* *Confirm effect on application security by deleting the cookies.*
* *Check if the session is recorded during correct time.*
* *Test for application robustness for rejecting cookies.*
* *Check if the session cookies or persistent cookies are used appropriately.*

***Passwords, Permissions and Roles***

* *Password should be displayed as asterisk or other masking.*
* *Password should be encrypted in source code.*
* *Change password should be take effect after login in the same computer.*
* *Password should not be able to be copied/pasted.*
* *Password security rule check (password strength): length, letter & number combination etc.*
* *Each user should have correct privileges as per their role to execute tasks and access information.*

***User Interface***

* *All components should be displayed in the correct location even changes in page sizes.*
* *Check for overlap or truncation for each component.*
* *Fonts and colors should be clear and easy to discern.*
* *Check for garbage characters.*
* *Check different resolutions to ensure that all application components display correctly.*
* *Check changing window sizes and ensure all application components display correctly.*
* *Check supported browser/OS to ensure that all application components display correctly.*
  1. **Performance testing**

The top priority functionalities to be tester for performance issues and bottlenecks are “Access to lectures/presentations/home works” and “Messages”. The functionalities that invoke the database for access to its content as the messaging system and the database of all the lectures, presentations and home works must be optimized to not slow down the system, especially in cases where multiple user are accessing the at the same time. The “Friends” management system is not as performance dependent as the top priority ones but in a future state it may also generate a large amount of data to be transferred if the user list manages to grow large so it should also be specifically tested to ensure adequate future behavior.

* 1. **Load testing**
  2. **Stress testing**
  3. **Regression testing**

After all other types of testing is done regression test should be performed. As we all know frequently, a fix for a problem in one area causes a bug in another area. So the purpose of doing regression test is to ensure that all currently implemented functionalities are still fully functional and no new bug are introduced, weather a new functionality is added or an old issue is fixed. This test will be done again after another bug have been found and fixed. This will be done by reusing the automated test suite written in order to ensure the quality of the currently implemented functionalities.

1. **Test Tools**

|  |  |  |
| --- | --- | --- |
| Tool Name | Vendor | Version |
| Telerik TeamPulse | Telerik |  |
| Selenium | Open Source |  |
| Telerik Test Studio | Telerik |  |