|  |  |
| --- | --- |
| Document Information | |
| Title | Telerik Academy Learning System |
| Testing Modules | Basic Modules |
|  | Certificates |
|  | Software Academy |
| Due date | Nov 24 2014 |
| Type | Report |
| Status | Version 2.2 |
| Responsible | Redcurrant Team: |
|  | Hristo Tsekov |
|  | Kalin Marinov |
|  | Milena Gyulmezova |
|  | Violeta Roumenova |
|  | Yordan Tsenkov |
| Project URL | [**www.test.telerikacademy.com**](http://www.test.telerikacademy.com/) |

**Revision and Signoff Sheet**

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Updated By** | **Update Comments** |
| 1.0 | 09/20/2014 | Violeta Roumenova | Creating Table of Content |
| 1.1 | 09/20/2014 | Milena Gyulmezova | Fulfill Summary and Introduction |
| 1.2 | 09/21/2014 | Hristo Tsekov | Fulfil Relationship to Other Documents and Test Strategy |
| 1.3 | 09/23/2014 | Yordan Tsenkov | Fulfill Levels of Testing and Test Plan Criteria |
| 1.4 | 09/24/2014 | Kalin Marinov | Fulfill Scope, Risks, Environmental Needs and Test Schedule and Effort Estimate |
| 1.5 | 09/29/2014 | Violeta Roumenova | Edit [Scope](#_Scope), Entry and Exit Criteria, Levels of Testing, Schedules |
| 1.6 | 09/30/2014 | Kalin Marinov | System Testing, Risks |
| 1.7 | 10/01/2014 | Milena Gyulmezova | Edit Audience, Test Objectives, Scope, Exit Criteria, Environment |
| 1.8 | 10/02/2014 | Violeta Roumenova | Test Cases |
| 1.9 | 10/15/2014 | Hristo Tsekov | Added test approach, removed “Glossary” |
| 2.0 | 10/24/2014 | Milena Gyulmezova | Changes further to document “RedCurrantTestPlan\_FormalReviewPapaya\_21.10.14” |
| 2.1 | 11/05/2014 | Violeta Roumenova | Exit criteria, Metrics and Measurements |
| 2.2 | 11/12/2014 | Violeta Roumenova | Schedules, Performance Testing |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Content**

[1. Executive Summary 5](#_Toc403685760)

[2. Introduction 5](#_Toc403685761)

[*2.1.* *Purpose* 5](#_Toc403685762)

[*2.2.* *Project Overview* 5](#_Toc403685763)

[*2.3.* *Audience* 5](#_Toc403685764)

[3. Relationship to Other Documents 6](#_Toc403685765)

[4. Test Strategy 6](#_Toc403685766)

[*4.1.* *Test Objectives* 6](#_Toc403685767)

[*4.2.* *Test Assumption* 6](#_Toc403685768)

[*4.3.* *Test Principles* 7](#_Toc403685769)

[5. Scope 7](#_Toc403685770)

[*5.1.* *Features to be tested* 7](#_Toc403685771)

[5.1.1. Administrator part 7](#_Toc403685772)

[5.1.2. User part 8](#_Toc403685773)

[*5.2.* *Features not to be tested* 8](#_Toc403685774)

[5.2.1. Administrator part 8](#_Toc403685775)

[5.2.2. User part 9](#_Toc403685776)

[6. Test Approach 9](#_Toc403685777)

[6.1. System testing 10](#_Toc403685778)

[6.1.1. Functional Testing 10](#_Toc403685779)

[6.1.2. Non-Functional Testing 10](#_Toc403685780)

[7. Test Plan Criteria 13](#_Toc403685781)

[*7.1.* *Entry Criteria* 13](#_Toc403685782)

[*7.2.* *Pass/Fail Criteria* 13](#_Toc403685783)

[*7.3.* *Exit Criteria* 13](#_Toc403685784)

[*7.4.* *Suspension Criteria and Resumption Requirements* 14](#_Toc403685785)

[7.4.1. Suspension Criteria 14](#_Toc403685786)

[7.4.2. Resumption Requirements 15](#_Toc403685787)

[8. Measurements and Metrics 15](#_Toc403685788)

[9. Risks 15](#_Toc403685789)

[*9.1.* *Description* 15](#_Toc403685790)

[*9.2.* *Risk Mitigation / Contingency* 17](#_Toc403685791)

[10. Environmental Needs 17](#_Toc403685792)

[*10.1.* *Hardware* 17](#_Toc403685793)

[*10.2.* *Software* 17](#_Toc403685794)

[*10.3.* *Tools* 18](#_Toc403685795)

[11. Test Schedule and Effort Estimate 18](#_Toc403685796)

[12. Test Analysis and Design 18](#_Toc403685797)

[*12.1.* *General Test Scenarios* 19](#_Toc403685798)

[*12.2.* *GUI and Usability Test Scenarios* 19](#_Toc403685799)

[*12.3.* *Test Scenarios for Image or Other Files Upload Functionality* 20](#_Toc403685800)

[*12.4.* *Test Scenarios for Excel Export Functionality* 21](#_Toc403685801)

[*12.5.* *Performance Testing Test Scenarios* 21](#_Toc403685802)

[*12.6.* *Security Testing Test Scenarios* 21](#_Toc403685803)

[13. Defect Tracking 22](#_Toc403685804)

[Acronyms 22](#_Toc403685805)

# Executive Summary

This deliverable captures the Telerik Academy Learning System project Test Plan intended to describe the scope, approach, resources, and schedule of the testing activities. The document also identifies the test plan deliverables, the participants responsible for implementing each task, and the risks associated with the plan. This document is intended to serve as a framework document for the consideration of these issues. During the development of the testing activities it shall be used in conjunction with the followings:

* Considered scenarios
* Analysis of scenarios and extraction of functional and non-functional requirements
* System Architecture Specification

# Introduction

## *Purpose*

This test plan describes the testing approach and overall framework that will drive the testing of the Telerik Academy Learning System – [**www.test.telerikacademy.com**](http://www.test.telerikacademy.com/) site. This document introduces:

* Test Strategy: rules the test will be based on, including the givens of the project (e.g.: start / end dates, objectives, assumptions); description of the process to set up a valid test (e.g.: entry / exit criteria, creation of test cases, specific tasks to perform, scheduling, data strategy);
* Execution Strategy: describes how the test will be performed and process to identify and report defects, and to fix and implement fixes;
* Test Management: process to handle the logistics of the test and all the events that come up during execution (e.g.: communications, escalation procedures, risk and mitigation, team roster).

## *Project Overview*

Telerik Academy Learning System is a student system developed for helping the educational initiatives in the Academy of Telerik. It provides detailed information for all on site and online courses and trainings that are organized by the Telerik Training Team. The modules to be tested - The Basic Modules (Settings, Navigation, Information for latest forum posts, videos, blog posts, upcoming courses and social media, Search), Certificates, Software Academy (Applications, Entry exam, Specialties).

## *Audience*

* Project team members perform tasks specified in this document, and provide input and recommendations on this document (Redcurrant Team);
* Project Manager plans the testing activities in the overall project schedule, reviews the document, tracks the performance of the test according to the task herein specified, approves the document and is accountable for the results (Member of Redcurrant Team);
* The stakeholders’ representatives and participants may take part to ensure the business is aligned with the results (Telerik Academy Trainers);

# Relationship to Other Documents

This section explains the relationship of the test plan to the other documents produced during the development effort. It explains how all the tests are related to the functional and nonfunctional requirements, as well as to the system design stated in the respective documents. If necessary, this section introduces a naming scheme to establish the correspondence between requirements and tests.

**Test cases** can be found in Team foundation server – [Team Redcurrant](http://78.83.67.72:8088/tfs/TelerikLearningSystem/TelerikAcademyLearningSystem/_testManagement#planId=1&suiteId=7)

**Results** from test runs, created with “Telerik testing framework” can be found at: [Test run results](https://qatelerikacademy.testrail.com/index.php?/runs/overview/1)

# Test Strategy

## *Test Objectives*

Verify the proper working of the main functionalities of the modules under testing.

Identify any failure modes that may occur when the system is placed under stress of high volumes, transaction rates and hostile environments through comprehensive system testing.

Identify any failure modes that may emerge when the system is integrated on-site through site integration testing.

Identify any faults induced by maintenance activities through comprehensive regression testing of all changes.

## *Test Assumption*

The Test Plan and Test Strategy is based on the following assumptions.

This list of assumptions will be updated as additional testing.

The below mentioned list is not only limited to these points only:

* Resources from the project teams will be allocated to test efforts as outlined in this document.
* Testing tasks will be given a high priority and will supersede other activities during key test phases as defined in the project plan
* All issues and defects identified during the testing process will be addressed in a timely manner during sprints by the assigned team member.
* All working scrum team members will have Laptops, so that everyone can be mobile and can work in scrum team “Skype” chat with all the team members for the respective tracks and releases.

## *Test Principles*

* Testing will focus on security, reliability, usability and efficiency of the application.
* There will be common, consistent procedures for all teams supporting testing activities.
* Testing processes will be well defined, yet flexible, with the ability to change as needed.
* Testing activities will build upon previous stages to avoid redundancy or duplication of effort.
* Testing environment and data will emulate a production environment as much as possible.
* Testing will be a repeatable, quantifiable, and measurable activity.
* Testing will be divided into distinct phases, each with clearly defined objectives and goals.
* There will be entrance and exit criteria.

# Scope

## *Features to be tested*

### Administrator part

#### Basic modules / Основни модули

* Settings / Настройки
* List of Errors / Списък на грешките
* Files Attachments / Качване на файл
* Dynamic pages / Динамични страници
* Feedback / Обратна връзка
* Halls / Зали
* Devices presences / Устройства присъствия
* Search / Търсени думи
* Quests / Анкети

#### Certificates / Сертификати

* Certificates / Сертификати
* Applications for Certificates / Kандидатури за сертификати
* Types of Certificates / Видове сертификати
* Templates for Certificates / Шаблони за сертификати

#### Software Academy / Софтуерна Академия

* Candidates / Кандидати
* Additional documents for applicants / Допълнителни документи на кандидати
* Statistics candidates / Статистика кандидати
* Statistics questions / Статистика въпроси
* Correlation - entrance exams / Корелация - входни изпити
* Entrance exams / Входни изпити
* All candidates / Всички кандидати
* Students in the system / Студенти в академията
* Accepted participants / Приети участници
* Seasons/ Сезони

### User part

#### Basic modules / Основни модули

* Settings / Настройки
* Site navigation / Навигация на сайта
* Notifications / Да получава информация за последните форум постове/ последни видео материали/ най-нови блог постове/ предстоящи курсове/ социални медии
* Site search / Търсене

#### Certificates / Сертификати

#### Software Academy / Софтуерна Академия

* Applications / Кандидатстване за Софтуерната академия
* Entry exam / Входен изпит
* Certificate specialties / Специалности

## *Features not to be tested*

### Administrator part

#### Users/ Потребители

#### Courses and Workshops/ Курсове и семинари

#### Evaluation/ Оценяване

#### AntiCheat

#### Team Work/ Отборна работа

#### Calendars/ Календари

#### Testing System/ Тестова система

### User part

#### Attachment/ Re-attachment of homeworks / Качване/ повторно качване на домашни

#### Registration for exams / Записване за изпити

#### Evaluation of teammates / Оценяване на съотборници

#### Access to comments from other students for homework / assessment of the utility of comment / Достъп до коментари от други студенти за домашното/ оценка на полезност на коментар

#### Result of exam / Резултат от изпита

#### Rate homework / Оцени домашно

#### Teamwork / Отборна работа

#### Navigating between different courses / Навигиране между различните курсове

#### My courses / Моите курсове

#### Archive / Архив

#### Academy Everywhere

#### To accept and receive friend invitations / Да приема и получава покани за приятелство

#### Changing the form of training – on-sight / online / Смяна на формата на обучение - присъствено/онлайн

#### Access to lectures / presentations / homeworks / Достъп до лекциите/презентации/домашни

#### Registration for courses / Записване за курсове

#### Registration / Регистрация

#### Log in / Логване

#### Friends / Приятели

#### Messages / Съобщения

#### Calendars / Kалендари

# Test Approach

Choice of test approach is one of the most powerful factor in the success of the test effort and the accuracy of the test plans and estimate.

The testing process will include only System testing, because the Component, Integration and Acceptance testing are assigned to other teams or third party companies.

## System testing

The application will be tested on multiple browsers like IE, Firefox, Chrome, and Safari with versions approved by client.

System Testing focuses on functional and non-functional testing.

### Functional Testing

* ***Sanity Testing***

Priority: High

Test Attributes: Accuracy and Suitability

The application helps students to organize their studying process and it is important that models work properly. This testing is suitable for Regression testing and will cover the Usability Testing. When requirements are needed the testing team will use Test Oracle technique.

Accuracy is of upmost importance as the system is responsible for calculating grades for exams/homework/teamwork.

* ***Interoperability***

Priority: Medium or Low

It will measure how well different parts from the system are communicating to each other. Are certificates given to correct student, bonus points and grades are assigned to correct student, the student is in the correct course.

### Non-Functional Testing

* ***User Interface Testing***

The purpose of user interface testing is to verify that the system’s GUI using Test Oracle. The verification may include some user interface issues such as aesthetic, validation, navigation, usability and data integrity conditions.

* ***Reliability***

Priority: High

Test Attributes: Robustness

It will be tested how well software reacts to unpredictable circumstances for example pictures right on the size limit, with doc files bigger than normal(50-150 mb). Also with unpredictable inputs, for example inputs without whitespaces in them, with random and strange character groups like Chinese and Hindi.

* ***Security***

Priority: High

Aspects that will be tested:

Authentication: Allow a receiver to have confidence that information it receives originated from a specific known source.

Authorization: Determining that a requester is allowed to receive a service or perform an operation.

Confidentiality: Protect the disclosure of data or information to other parties than the intended.

Integrity: Check that the intended receiver receives the information or data which is not altered in transmission.

Non-repudiation: Interchange of authentication information with some form of provable time stamp.

* ***Performance/Load Testing***

Priority: High

Test Attributes: Efficiency

Telerik Learning System will be tested using Visual Studio performance test process. We will watch and measure the response time of application. We will analyze and present the performance test results at the conclusion of the performance test cycle.

***The areas of testing are:***

* Apply for Academy with login
* Apply for Entry Exam with login
* Search for information without login

***The following tests will be run:***

* Load Test – Apply for Academy and Apply for Entry Exam will be performed as web test with 100 and 200 regular users and the results will be compared. The Search and Navigation to Application page for Academy will be tested with increasing quantities of concurrent users until the count reaches 200 users
* Stress Test – will be perform for Search and Navigation to Application page for Academy as 250 users are constantly using the application
* Endurance Load Test –will drive a continuous load on the application server for an extended period of time (at least 1 hours). The main purpose of this type of test is to ensure the application can sustain acceptable levels of performance over an extended period of time without exhibiting degradation, such as might be caused by a memory leak or server crash.

***Scenarios:***

* Execute web tests for Apply for Academy and Apply for Entry Exam with 100 and 200 users
* Execute load tests for Search and Navigation to Application page for Academy with 200 concurrent users max. The test will star with 10 users, every 10 seconds the count of users will increase with 10 until the max count of users is reached. The duration of test will be 20 minutes
* Execute stress tests for Search and Navigation to Application page for Academy with 250 constant users. The duration of test will be 2 minutes
* Execute endurance test for at least 1 hours with increasing quantities of concurrent users until the users reach count of 200. The test will have the same characteristics as load test.

***Test Measurements, Metrics***

The following metrics will be collected:

Tests(sec), Avg. Test Time(sec), Page(sec), Avg. Page Time(sec) and Avg. Response Time(sec), Failed Tests, Threshold Violations according to the count of users.

***Expected Results and Pass/Fail Criteria***

The following are performance requirements (success criteria) for the performance tests:

1. The average response time is less than 3 seconds for 1000 regular users and less than 1 second for 100 administrators.

2. The maximum number of acceptable server errors, non HTTP-200 status codes on client requests, will be less than 2% of all client requests.

***Load Descriptions***

There is no expected difference between users logging in for the first time or subsequent logins given how the data will be created.

There will be no ramp up time for any of the stress tests. The ramp up time for load tests, should be set to 10 users every 10 seconds.

The wait time between requests is default.

# Test Plan Criteria

## *Entry Criteria*

* Build notes are provided to the testing team.
* Test Closure Report for Integration Testing is signed off by required stakeholders.
* Test plan is created.
* Test cases are created and covers 95% of critical and high functionality
* All test hardware platforms must have been successfully installed, configured, and functioning properly.
* All the standard software tools including the testing tools must have been successfully installed and functioning properly.
* All personnel involved in the testing effort must be trained in the tools to be used in the testing process.
* Proper test data is available.
* The test environment such as, lab, hardware, software, and system administration support should be ready.

## *Pass/Fail Criteria*

* The result for each test match the expected result logged in the Test Case.
* All test suites completed for software modules.
* Tests by priority completed without errors and a percentage with minor defects for hardware modules.

## *Exit Criteria*

Exit criteria can be generalized:

* Main functionality available.

According to tasks given to the team, the application should be able to allow user to apply for the academy, to choose date for entrance exam, to set his/hers profile and have proper navigation, able to execute with 250 users and 100 administrators.

As administrator, the user should be able to make settings, to attached files, checks the attendance, organize entrance exams, monitors students in the system

* Important faults cleared.

Faults which are classified as Blocking, or High, and affect major areas of the system with no reasonable work rounds must be cleared.

* Other faults recorded

Any other faults should ideally be fixed, but if they have not been then they must be recorded. They also need to be judged to see what affect they will have on the next stage of testing.

* Documentation updated.

The documentation that must be updated is any that is necessary for the trace of process of testing as test plan and reports

* Test Report produced.

Test Report gives an overview of the quality of the software at some point.

Lastly we can summarized the above written:

* 100% Test Scripts executed
* 95% of the test cases have passed.
* There are no blocking nor critical bugs.
* All high-risk areas have been fully tested, with only minor residual risks left outstanding.
* 95% of Medium severity defects have been closed
* All remaining defects are either cancelled or documented as Change Requests for a future release
* All expected and actual results are captured and documented with the test script
* All test metrics collected based on reports
* Test Closure Memo completed and signed off
* The schedule has been completed.

## *Suspension Criteria and Resumption Requirements*

### Suspension Criteria

* Unavailability of external dependent systems during execution.
* The build contains many critical defects which seriously prevent or limit testing progress.
* Assigned test resources are not available when needed by the test team.

### Resumption Requirements

* If testing is suspended, resumption will only occur when the problem(s) that caused the suspension have been resolved. When a critical defect is the cause of the suspension, the “FIX” must be verified by the testing team before testing is resumed.

# Measurements and Metrics

|  |  |  |
| --- | --- | --- |
| **Report** | **Description** | **Frequency** |
| Test preparation and Execution Status | To report on % complete, %WIP, % Pass, % Fail Defects severity wise Status – Open, closed, any other Status | Weekly/Daily (optional) |
| Daily execution status | To report on Pass, Fail, Total defects, highlight Show stopper/ Critical defects | Daily |
| Project Weekly Status report | Project driven reporting (As requested by PM) | Weekly |

Metrics to evaluate are:

* Number of bugs found
* Number of test cases produced given by priority.
* Number of automated test produced.
* Percentage of total application covered in functional testing.

# Risks

## *Description*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Description | Type | Impact | Likelihood | Owner |
| 1 | Numerous input fields. A lot of validation is needed in order to verify all inputs. | Security  Robustness | High | Somewhat likely | Developer / QC |
| 2 | Various application tiers, technologies used (languages, tools, platforms) | Maintainability  Testability  Analyzability | Medium | Somewhat  likely | Admin / QC |
| 3 | Lack of specialists in each field | Maintainability  Testability |  | Somewhat  likely | Admin / QC |
| 4 | Lack of requirements | Suitability  Testability | High | Somewhat  likely | QC |
| 5 | Third Party applications | Robustness  Testability | Medium | Not likely | Developer / QC |
| 6 | Lack of cooperation with other QC teams | Maintainability  Testability | Medium | - | QC |
| 7 | Illness of team member | Testability | Medium | Somewhat  Not likely | QC |
| 8 | Lack of information whether component or integration testing is done. | Component | High | Somewhat  Not likely | Developer |
| 9 | Possible issues with concurrency | Robustness  Reliability  Usability | Medium | Occasional | Developer / QC |
| 10 | User/Admin fault tolerance | Maintainability  Recoverability |  |  |  |
| 11 | Performance testing: Limit on the number of virtual users supported by our computers | Robustness | High | Somewhat  likely | QC |
| 12 | Unknown count of users that Application can handle before undesirable behavior occurs when the application is subjected to a particular workload | Robustness | Medium | Somewhat  likely | QC |
| 13 | Unknown capacity of data that database/file server can handle | Robustness | Low | Somewhat  likely | QC |
| 14 | Lack of information what would happen if the production load exceeds the anticipated peak load | Robustness | Low | Somewhat  likely | QC |

## *Risk Mitigation / Contingency*

|  |  |  |
| --- | --- | --- |
| Id | Mitigation | Contingency |
| 1 | Validation for each field. Input escaping (for potentially dangerous input (hacking)) | Record all inputs that resulted in an exception/error on page |
| 2 | Internal training | Help from third party organizations |
| 3 | Heuristic techniques | Help from third party organizations |
| 4 | Reverse engineer the modules/components | - |
| 5 | Support contract with the creator.  If open source attend courses/trainings | - |
| 6 | More communication across teams | - |
| 7 | - | Backup/help from other teams |
| 8 | - | Applying as much as possible system testing |
| 9 | Load/Stress testing all the components. Test the system’s accuracy and precision under stress | - |
| 10 | Test that bad input or data corruption caused by human mistake can be easily reverted | - |
| 11 | Reducing the number of virtual users |  |
| 12 | Testing frequently and updating the application |  |
| 13 | Limit the tests that send data |  |
| 14 | Analyze and test more areas |  |

# Environmental Needs

## *Hardware*

* + - Intel (R) Xeon (R) CPU @ 3.10GHz (Quad Core)
    - Ram 8.00GB
    - HD 100GB

## *Software*

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

## *Tools*

* + - Web testing tools - Telerik Test Studio and Testing Framework
    - Issue tracking system - Team pulse
    - Test Management System – Excel, TFS
    - Visual Studio 2013 Ultimate
    - SQL Server Management Studio
    - SQL Server 2014 Reporting Services Configuration Management

# Test Schedule and Effort Estimate

|  |  |  |
| --- | --- | --- |
| **Test Phase** | **Time in Weeks** | **Owner** |
| Test Plan Creation | Sept 19 - 26, 2014 | Team Members |
| Test Cases Creation / Weekly Review | Sept 26 - Oct 3, 2014 | Team Members/Lead |
| Test Studio Automation / Weekly Review | Oct 3 - 16, 2014 | Team Members/Lead |
| Testing Framework Automation / Weekly Review | Oct 16 - 30, 2014 | Team Members/Lead |
| Testing Framework Automation / Weekly Review | Oct 30 -Nov 7, 2014 | Team Members/Lead |
| Performance/Regression Testing / Weekly Review | Nov 7 - 21, 2014 | Team Members/Lead |
| Review of whole project | Nov 25, 2014 | Team Members/Lead |

# Test Analysis and Design

This section, the core of the test plan, lists the test cases that are used during testing. Each test case is described in detail in a separate Test Case Specification document. If in each execution of these tests a bug is found, it will be documented in a Test Incident Report document. Test cases will be execute with different user roles e.g. admin user, guest user etc., depending of the case.

The Test cases check that application supports following functionality

* + - – Forms with various fields
    - – Application interacts with database
    - – Various search filter criteria and display results
    - – Image upload
    - – Files upload

**Test cases naming scheme**

Test cases are organized in folders. Folders are divided by the type of user (ordinary or administrator), then by the sections that are assigned to the team. Every section folder contains subfolder for every module in the section and then if the module contains pages, there is a folder for each page or element of page.

The names of test cases will indicate from where they have been derived using a system of attributes. The following attributes are used to denote the tests were derived from the following places.

[Settings] – Cases and tests covering the Settings page

[Navigation] - Cases and tests covering Navigation menu

[ApplySofAcademy] - Cases and tests covering Application for Software Academy page

[AttachFile] - Cases and tests covering Attach file page

Etc.

## *General Test Scenarios*

* 1. All mandatory fields should be validated and indicated if not fulfilled
  2. Validation error messages should be displayed properly at correct position
  3. All error messages should be displayed in same CSS style (e.g. using red color)
  4. General confirmation messages should be displayed using CSS style other than error messages style (e.g. using green color)
  5. Dropdown fields should have first entry as blank or text like ‘Select’
  6. Check all input fields for special characters
  7. Field labels should be standard e.g. field accepting user’s first name should be labeled properly as ‘First Name’
  8. Check page sorting functionality after add/edit/delete operations on any record
  9. Check cookies used in an application
  10. Application crash or unavailable pages should be redirected to error page
  11. Check text on all pages for spelling and grammatical error
  12. Check numeric input fields with character input values. Proper validation message should appear
  13. Check for negative numbers if allowed for numeric fields
  14. Check functionality of buttons available on all pages
  15. User should not be able to submit page twice by pressing submit button in quick succession.

## [*GUI*](http://www.softwaretestinghelp.com/gui-testing-on-smart-devices-%e2%80%93-testing-guidelines/) *and Usability Test Scenarios*

1. All fields on page (e.g. text box, radio options, dropdown lists) should be aligned properly

2. Enough space should be provided between field labels, columns, rows, error messages etc.

3. Font size, style and color for headline, description text, labels, infield data, and grid info should be standard as specified in SRS

4. Description text box should be multi-line

5. Disabled fields should be grayed out and user should not be able to set focus on these fields

6. Upon click of any input text field, mouse arrow pointer should get changed to cursor

7. User should not be able to type in drop down select lists

8. Information filled by users should remain intact when there is error message on page submit. User should be able to submit the form again by correcting the errors

9. Check if proper field labels are used in error messages

10. Default radio options should be pre-selected on page load

11. Field specific and page level help messages should be available

12. Check if dropdown list options are readable and not truncated due to field size limit

13. All buttons on page should be accessible by keyboard shortcuts and user should be able to perform all operations using keyboard

14. Check all pages for broken images

15. Check all pages for broken links

16. All pages should have title

17. Confirmation messages should be displayed before performing any update or delete operation

## *Test Scenarios for Image or Other Files Upload Functionality*

1. Check for uploaded image path
2. Check image upload and change functionality
3. Check image upload functionality with image files of different extensions (e.g. JPEG, PNG, BMP etc.)
4. Check image upload functionality with images having space or any other allowed special character in file name
5. Check duplicate name image upload
6. Check image upload with image size greater than the max allowed size. Proper error message should be displayed.
7. Check image upload functionality with file types other than images (e.g. txt, doc, pdf, exe etc.). Proper error message should be displayed
8. Check if images of specified height and width (if defined) are accepted otherwise rejected
9. Image upload progress bar should appear for large size images
10. Check if cancel button functionality is working in between upload process
11. Check if file selection dialog shows only supported files listed
12. Check multiple images upload functionality
13. Check image quality after upload. Image quality should not be changed after upload
14. Check if user is able to use/view the uploaded images

## *Test Scenarios for Excel Export Functionality*

1. File should get exported in proper file extension
2. Check for date format if exported Excel file contains date columns
3. Check number formatting for numeric or currency values. Formatting should be same as shown on page
4. Exported file should have columns with proper column names
5. Default page sorting should be carried in exported file as well
6. Excel file data should be formatted properly with header and footer text, date, page numbers etc. values for all pages
7. Check if data displayed on page and exported Excel file is same
8. Check export functionality when pagination is enabled
9. Check if export button is showing proper icon according to exported file type e.g. Excel file icon for xls/xlsx files
10. Check export functionality for files with very large size
11. Check export functionality for pages containing special characters. Check if these special characters are exported properly in Excel file

## *Performance Testing Test Scenarios*

1. Check if page load time is within acceptable range
2. Check page load on slow connections
3. Check response time for any action under light, normal, moderate and heavy load conditions
4. Check performance of database stored procedures and triggers
5. Check database query execution time
6. Check for load testing of application
7. Check for stress testing of application
8. Check CPU and memory usage under peak load condition

## [*Security Testing*](http://www.softwaretestinghelp.com/security-testing-of-web-applications/)*Test Scenarios*

* 1. Check for SQL injection attacks
  2. Secure pages should use HTTPS protocol
  3. Page crash should not reveal application or server info. Error page should be displayed for this
  4. Escape special characters in input
  5. Error messages should not reveal any sensitive information
  6. All credentials should be transferred over an encrypted channel
  7. Test password security and password policy enforcement
  8. Check application logout functionality
  9. Check for Brute Force Attacks
  10. Cookie information should be stored in encrypted format only
  11. Check session cookie duration and session termination after timeout or logout
  12. Session tokens should be transmitted over secured channel
  13. Password should not be stored in cookies
  14. Test for memory leakage
  15. Test unauthorized application access by manipulating variable values in browser address bar
  16. Test file extension handing so that exe files are not uploaded and executed on server
  17. Sensitive fields like passwords should not have auto complete enabled
  18. File upload functionality should use file type restrictions and also anti-virus for scanning uploaded files
  19. Check if directory listing is prohibited
  20. Password and other sensitive fields should be masked while typing
  21. Check if forgot password functionality is secured with features like temporary password expiry after specified hours and security question is asked before changing or requesting new password
  22. Check if important events are logged in log files
  23. Check if access privileges are implemented correctly

# Defect Tracking

The bugs will be logged in Team pulse. Below is a description of the Severity codes that are used:

|  |  |
| --- | --- |
| **Severity Code name** | **Explanation** |
| 1 - Critical Severity | Tests cannot proceed further within applicable test case (no work around) |
| 2 - High Severity | The test case or procedure can be completed, but produces incorrect output when valid information is input. |
| 3 - Medium Severity | The test case or procedure can be completed and produces correct output when valid information is input, but produces incorrect output when invalid information is input.(e.g. no special characters are allowed as part of specifications but when a special character is a part of the test and the system allows a user to continue, this is a medium severity) |
| 4 - Low Severity | All test cases and procedures passed as written, but there could be minor revisions, cosmetic changes, etc. These defects do not impact functional execution of system |

# Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| GUI | Graphical User Interface |
| QC | Quality Control |
| HTTP | Hyper Text Transfer Protocol Secure |
| IDE | Integrated Development Environment |
| IE | Internet Explorer |
| OS | Operation System |
|  |  |