|  |
| --- |
| **Test Plan for** [**www.coursera.org**](http://www.coursera.org)  Student Name: Noel Grace  Student Number: 20009643 |

**DOCUMENT DETAILS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Document Type: | | **Test Plan** | | |
| **Document Version:** | | 1.0 | | |
| **Created by:** | | Noel Grace | Creation Date: | 08/03/16 |
| **Updates** | | | | |
| Version | Date: | Updated by | Comments | |
| 0.1 | 08/03/16 | Noel Grace | New Document | |
| 0.2 | 14/03/16 | Noel Grace | Updated after walkthrough with QA team | |
| 0.3 | DD/MM/YY | XXX | Updated after reviews by Review team | |
| 1.0 | DD/MM/YY | XXX | Approved Document | |

**Walkthroughs/Reviews**

|  |  |  |
| --- | --- | --- |
| **Reviewed By:** | **Role** | **Date** |
| Michael Grace | QA Team | 08/03/16 |
| Pádraic Grace | QA Team | 14/03/16 |
| Stephanie Connolly | Reviewer | DD/MM/YY |
| Sinead Kilcawley | Reviewer | DD/MM/YY |

**Stakeholder Approvals**

|  |  |  |
| --- | --- | --- |
| **Approved By** | **Role** | **Date** |
| Mary Grace | Manager | 08/03/16 |
| Larry Byrne | Project Manager | 08/03/16 |

**Table of Contents**

1 INTRODUCTION 4

1.1 Overview 4

1.1.1 Aim 4

1.2 Purpose of the plan 5

1.3 The test plan 5

1.3.1 Documentation References …………………………………………………………………………………………………7

2 Roles and Responsilities 7

3 TEST SCHEDULE 8

4 FEATURES TO BE TESTED 9

5 FEATURES NOT TO BE TESTED 11

6 TEST APPROACH 11

6.1 Testing Levels 11 6.1.1 Unit Testing………………………………………………………………………………………………………………...11

6.1.2 Test Methods 11

6.1.3 System Testing ……………………………………………………………………………………………………………11

6.2 Test Methods……………………………………………………………………………………………………………….11

6.2.1 Variety of Test, Methods, Levels & Types…...………………………………………………………………………….12

6.3 Test Types 12

6.4 Key Steps for the Test Process 13

7 TESTING DELIVERABLES 13

8 RESOURCE NEEDS 14

8.1 Human 14

8.2 Hardware 14

8.3 Software 14

9 TRAINING NEEDS 14

10 RISKS AND ASSUMPTIONS 15

10.1 Risks 15

10.2 Assumptions 15

11 DEFECTS MANAGEMENT 16

11.1 Defects Process 16

11.2 Defects Categories ……………………………………………………………………………………………………… 17

11.3 Documenting the Defects …………………………………………………………………………………………………17

11.3.1 Defects Tracking Tool…………………………………………………………………………………………………… 18

11.3.2 BugZero Defect Report ……………………………………………………………………………………………………18

11.3.3 BugZero Defect Report Template ……………………………………………………………………………………… 19

11.4 Investigation ……………………………………………………………………………………………………………… 19

11.5 Fixing…………………………………………………………………………………………………………………………19

11.6 Checking…………………………………………………………………………………………………………………….19

11.7 Throughout the process……………………………………………………………………………………………………20

11.8 Fixing Defects

12 EXIT CRITERIA 21

12.1 When can testing be stopped/reduced………………………………………………………………………………… 21

13 TEST CASES, PASS/FAIL CRITERIA AND TEST DATA 22

14 References……………………………………………………………………………………………………………………... 28

15 Abbreviations…………………………………………………………………………………………………………………… 28

# INTRODUCTION

## Overview

|  |
| --- |
| **Provide a short overview of the system, highlighting the main functionalities and benefits (compared to previous systems).** |

Coursera is a [venture-backed](https://en.wikipedia.org/wiki/Venture_capital), for-profit, [educational technology](https://en.wikipedia.org/wiki/Educational_technology) company that offers [massive open online courses](https://en.wikipedia.org/wiki/Massive_open_online_course) ([MOOCs](https://en.wikipedia.org/wiki/MOOCs)).

Coursera works with top universities and organizations to make some of their courses available online, and offers courses in many subjects including:

Physics, engineering, humanities, medicine, biology, social sciences, mathematics, business, computer science, digital marketing, data science, and other subjects.

Users participate in MOOCs by creating an account and selecting courses that they wish to participate in. There is no obligation to complete the course(s). Users have access to fee paying courses provided they pay a course fee and will receive a certification on completion of the course(s). Many MOOCs offer the option for participants to complete the course without paying a fee. In this situation the user will not receive a certificate but will however having the pleasure of completing the course. Users will be required to complete quizzes and submit the answers. Users will be required to achieve a certain mark to successfully complete certain sections of a course. Users can repeat quizzes if they do not meet the pass mark or if they wish to achieve a higher mark. The users can participate in forum sections in order to interact with other users, share experiences, ask questions, and offer advice. Course mentors also participate in the forums to answer questions. Users can follow/like certain posts so that when there is any updates to the topics posted in the forums they will receive an email providing team with a notification. Users have the option to download the course material in the form of videos and pdf if they wish. Once a user completes a course they will still have access to review the material online or to retake a course. Users can participate in as many courses as they like and there is no penalties for not completing a course.

The main benefits compared to previous educational websites are:

1. Participation in course forums
2. Options to participate in courses for free

### 1.1.1 Aim

The aim of this project is to update the Coursera website to a newer version that will allow users who are complete MOOCs to:

* + The options to participate in forums with other users where they can share ideas/opinions, offer advice to fellow users, post questions and submit answers to questions.
  + gain the advice of course mentors

## Purpose of Plan

|  |
| --- |
| **Briefly explain the objective of the test plan, its structure, and content** |

The purpose of the testing activities are to:

1. Ensure that the quality of the Coursera website meets the required quality standards as set out by the company Coursera. This quality may refer to the content, the structure, or the way the page has been advertised. Key metrics such as: User Experience, Popularity, Scalability, Content, Forms, Security, Monitoring and Maintenance will be used to evaluate the quality of the website.
2. Determine that the software is ready for release
3. Discover any defects that will need to be monitored and fixed (see section 13 Defects Management)
4. Demonstrate any improvements since the previous version of the Coursera website.
5. Evaluate the functionalities of the website which refers to the security, usability, accessibility of the website

## 1.3 The test plan

This includes a detailed approach to the testing activities involved. It is not limited to but will include areas such as:

1. Roles & Responsibilities - the stakeholders involved in the testing activities
2. Test Schedule - the different testing activities, with an estimated starting/end date, along with an estimated effort. The key milestones will be identified.
3. Features to be tested – this section identifies the test cases carried out.
4. Features not to be tested – this section briefly lists some of the features that will not be tested as they are considered outside of the scope of this test plan.
5. Test Approach – a description of the test approach, and the different levels, methods and types of test that will be used.
6. Test Process - an overview of the overall testing process
7. Testing Deliverables – A list of the different deliverables
8. Resource Needs – A list of the human, hardware and software resources needed. The resources of the type human are outside the company that is developing the software and relate to beta-testing activities and acceptance testing (e.g. compliance with operational settings, etc.).
9. Training Needs – A description of any necessary training required by the stakeholders involved in the testing process.
10. Risks & Assumptions - This section, specifies potential risks that could impact on the testing process.Italso includes assumptions based on the project stakeholders and environment that can impact on the testing effort.
11. Defects Management – this section covers the processes that will be used to detect and monitor defects throughout the testing process (e.g., procedures, how the defects will be recorded, etc.).
12. Exit Criteria – This section covers a check list of the exit criteria.
13. Test Cases, pass/fail criteria and test data – this is the documented test cases.
14. References – a list of references.
15. Abbreviations – any abbreviations used and their explanations

### 1.3.1 DOCUMENTATION REFERENCES

**Add references to all documents used to compile this test plan. You may refer to the version/release number of the document (SDD, SRS, etc.).**

The following documentation was used in the preparation of this Test Plan:

**Table 1: Documentation Used**

| **Document Name** | **Version** |
| --- | --- |
| Software Requirements Specification | 1.0 |
| BugZero Test Case Reports | 1.0 |
| BugZero Defect Reports | 1.0 |

# ROLES AND RESPONSIBILITIES

List the name, responsibilities, and contact details for the stakeholders involving in the testing activities (QA team, engineers, etc.).

This section/table identifies the roles, responsibilities and contact information of all individuals and departments involved in the testing of the system

Table 2 - Role and Responsibility Matrix

| **Role/Title** | **Name\s** | **Testing Tasks** | **Contact #** |
| --- | --- | --- | --- |
| QA Team | Noel Grace  Michael Grace  Patrick Grace | - Develops the Test Cases - Reviews Test Plan  - Executes The Tests  … | [test@test.com](mailto:test@test.com) |
| Developers | Mary Grace  Roseann Grace  Colm Grace | - Develops Unit tests  - Launch Unit tests  … | [test@test.com](mailto:test@test.con) |

# TEST SCHEDULE

List the different testing activities, with an estimated starting/end date, along with an estimated effort. You may also highlight key milestones associated to deliverables. Activities to be included in this schedule may go beyond testing and account for cognate activities, including:

* Training for QA members not yet used to some testing procedures and introductory courses on software standards six sigma and ISO
* Preparation of the test plan documentation
* Review and approval of the documentation
* Execution of tests
* Documentation of the results
* Defect recording

**Table 3: Test Schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity/Deliverable** | **Estimated**  **Start Date** | **Estimated**  **End Date** | **Estimated Effort (days)** |
| QA team validation training course | 21/03/16 | 22/03/16 | 2 |
| Six Sigma Staff Training | 08/03/16 | 08/03/16 | 0.5 |
| ISO Staff Training | 08/03/16 | 08/03/16 | 0.5 |
| Testing Sign-off | 22/03/16 | 24/03/16 | 2 |
| Lessons Learned workshop | 24/03/16 | 24/03/16 | 1 |
| TOTALS (System Delivery Estimate) | 21/03/16 | 24/03/16 | 5 |

# FEATURES TO BE TESTED

**List the features/requirements to be tested, along with their corresponding test case**

The following features will be tested:

**Table 4: Features to be QA tested**

|  |  |  |
| --- | --- | --- |
| **F REQ #** | **Feature Description** | **Test Case #** |
| REQ\_SEC\_01 | Access to a user’s personal profile on the website will be authorized only if user has logged in with a correct user name and login. | TC\_01 and TC\_02 |
| REQ\_SEC\_01.01 | For a user to register the username field should be separated by the @ symbol and only consist of lowercase letters or numbers with no special characters | TC\_01 and TC\_02 |
| REQ\_SEC\_01.02 | For a user to register the password should consist of alphanumeric characters, upper and lower case letters and should include at least one special character such as (){}~’\_-\*/ @# | TC\_01 and TC\_02 |
| REQ\_UE\_01 | The site should have no broken links | TC\_09 |
| REQ\_UE\_02 | If the user requests a page that does not exist on the server, s/he will be redirected to a default page automatically (404). | TC\_10 |
| REQ\_UE\_03 | If the user requests a page that does exist on the server, s/he will be directed to that page automatically. | TC\_11 |
| REQ\_UE\_04 | The page load time should be fast | TC\_12 |
| REQ\_UE\_04.01 | The page load time should be less than 10 seconds | TC\_12 |
| REQ\_UE\_05 | The website should be highly responsive. | TC\_13 |
| REQ\_UE\_05.01 | The website should be responsive on desktops, laptops, tablets and phones | TC\_13 |
| REQ\_F\_01 | The information entered into the last name field will have a maximum of 30 characters | TC\_14/TC\_15 |
| REQ\_F\_01.01 | The information entered into the last name field will have a minimum of 10 characters | TC\_14 |
| REQ\_F\_01.02 | The information entered into the last name field will be in English | TC\_14 |
| REQ\_LIKE\_01 | The site should be popular | TC\_04, TC\_05 and TC\_06 |
| REQ\_LIKE\_01.01 | The site should have at least 100 hits/day and 10 visitors/day. | TC\_04 |
| REQ\_LIKE\_01.02 | The number of hits and visitors will be monitored by the site recorded at the bottom of the home page inside the footer section. | TC\_04 |
| REQ\_LIKE\_01.03 | The number of likes on the social media sites will be monitored | TC\_05 |
| REQ\_LIKE\_01.04 | The number of likes on the social media site Facebook should be at least 100 | TC\_06 |
| REQ\_LIKE\_01.05 | The number of followers on the social media site twitter should be at least 1000. | TC05 |
| REQ\_C\_01 | The site should have words spelled correctly. | TC\_07 |
| REQ\_C\_01.01 | The site should have words spelled correctly in English (US) settings. | TC\_07 |
| REQ\_NF\_01 | The site should be 100% HTML compliant | TC\_08 |
| REQ\_SCA\_1 | The site should be able to tolerate a lot of users at the same time | TC\_03 |
| REQ\_SCA\_1.01 | The site should be able to tolerate 200 or more users logged in at the same time | TC\_03 |

.

# FEATURES NOT TO BE TESTED

The following features will not be tested:

Anything related to the database management will not be tested.

# TEST APPROACH

## 6.1 Testing Levels

Three levels of testing will be carried out i.e. Unit Testing, Integration Testing and System Testing. Various types of testing come under these levels.

### 6.1.1 Unit Testing

Unit Testing will be used to verify a single program or a section of a single program.

### 6.1.2 Integration Testing

Prerequisite: unit testing completed on all components that compose a system.

Testing will be used to verify interaction between system components. The objective of Integration Testing is to find bugs related to interfaces between modules as they are integrated together.

### 6.1.3 System Testing

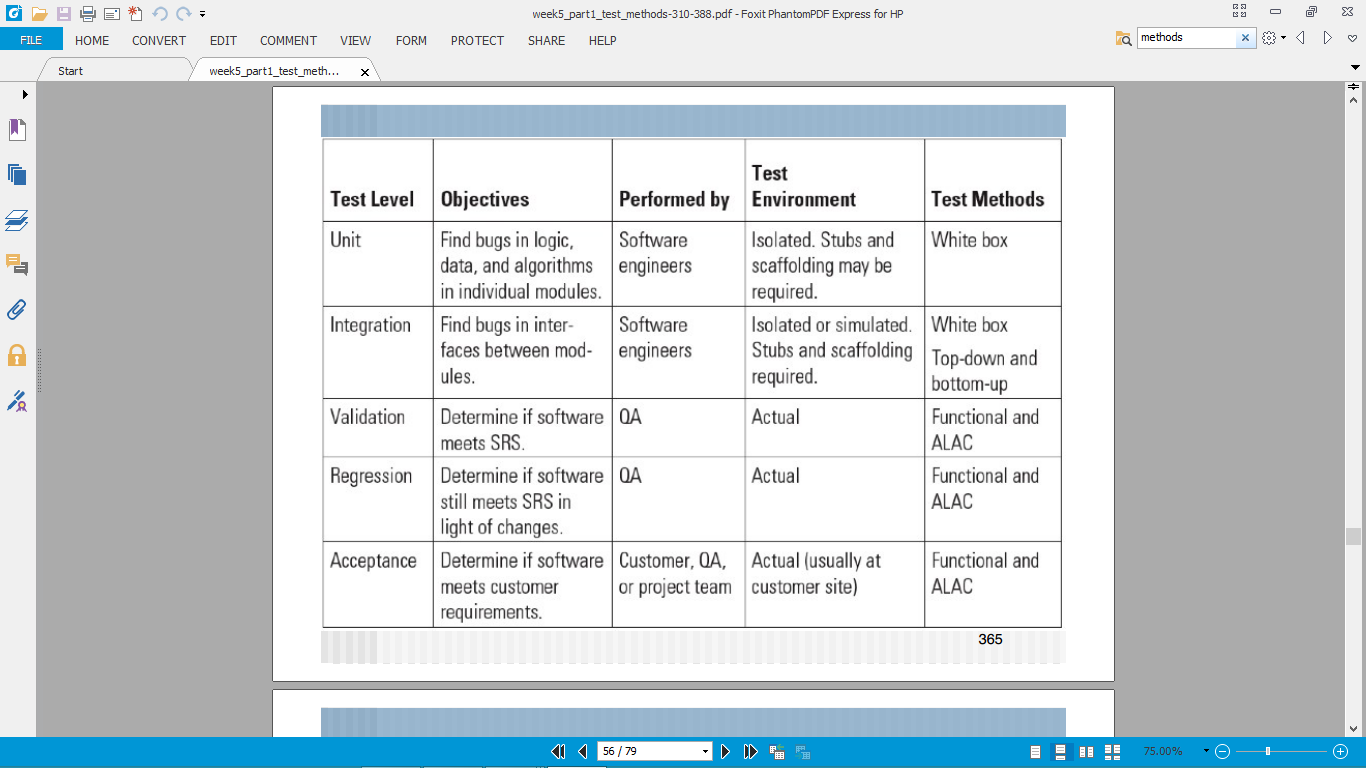
System Testing will be used to verify and validate behaviors of the entire system against the original system objectives.

## 6.2 Test Methods

The following Test Methods will be used:

* White box or Glass Box Testing
* Functional or Black Box Testing
* Top Down and Bottom Up Testing
* Act-Like-A-Customer

### 6.2.1 Variety of Test Methods, Levels & Types



### 6.3 Test Types

A variety of test types will be used including:

§Functional tests: designed to determine if specific functions/features work as specified. These will be carried out on the test cases related to website security (TC\_01 and TC\_02), the User Experience test cases (TC\_09) and also the Functionality test cases (TC\_14 and TC\_15).  
§Positive tests: designed to determine if a feature produces results that are consistent with the stated  
requirements when the software is used properly. These will be carried out on the test cases related to security (TC\_01) and User Experience – broken links (TC\_11).  
§Negative tests: designed to determine if the software behaves reasonably when faced with an invalid input or unexpected operator actions. These will be used for the security test cases (TC\_02) and User Experience (TC\_10).

§Usability tests: to exercise specific user interface features to determine if the software behaves as would be expected by trained/untrained users. These will be used for test cases TC\_09 to TC\_13.  
§Load/stress tests: exercises the product under stated or expected load conditions (e.g., number of users). These will be used to assess the test cases related to scalability/concurrency (TC\_03).6.4 Key Steps for the testing process

* Review Requirements Specification (check that requirements are testable)
* Plan for testing (test plan, risk analysis, mitigation, etc.)
* identify requirements to be tested
* Design Tests (e.g., Unit tests or surveys; specify success conditions, identify tests that can be automatized and or added for regression)
* Setting-up the Environments (e.g., ALAC, hardware, software, physical environments, etc.)
* Execute Tests
* Record Test Results
* Review Test results
* Fix issues
* Check if defects are resolved (if not Fix again)
* Regression Testing
* Final Review
* Testing Sign off

The formal review points are (also called Milestones):

**Table 8: Testing Milestones**

|  |  |  |
| --- | --- | --- |
| **#** | **Milestone** | **Date** |
| 1 | Formal Review of Test Plan | XXX |
| 2 | Review of Test Results | XXX |
| 3 | Final Review – Exit Criteria check | XXX |

# TESTING DELIVERABLES

List the different deliverables

**Table 10: Testing Deliverables**

|  |  |  |
| --- | --- | --- |
| **#** | **Deliverable** | **Date** |
| 1 | Approved Test Plan | DD/MM/YY |
| 2 | Release Notes | DD/MM/YY |

The Release Notes document can be used to describe what defects are known, have been resolved or not resolved.

# RESOURCE NEEDS

## Human

For each resources, include a title, start date, end date, number, and the corresponding names. The resources listed in this section are usually outside the company that is developing the software and relate to beta-testing activities and acceptance testing (e.g. compliance with operational settings, etc.). In the example below, we may need three administrators with different levels of access to test the system.

**Testing Human Resource Requirements**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Resource Title** | **Start Date** | **End Date** | **Number** 🚹 | **Resource Names** |
| Administrators | 08/03/16 | 24/03/16 | 3 | Kevin Grace, Máiread Grace, Lisa Gannon |
| QA | 21/03/16 | 24/03/16 | 2 | Eddie Bowe, William Bowe |
| Developers | 10/03/16 | 17/03/16 | 3 | Barry Kavanagh,  Colin Clarke,  Stephen Clarke |

## Hardware

The website will be designed based on the concept of mobile first, meaning that firstly, the website will be developed with the mobile screen size in mind, and then adapted to larger screen sizes from there. There will also be a responsive design element to the website. The website will adapt to being viewed on different devices such as laptops, desktops, tablets, mobile phones and Apple iPads. Based on these principles the Coursera website will need to be tested on multiple devices as mentioned above to ensure responsiveness.

## Software

It is intended to develop a website that is cross platform independent. The website will be tested on the most common browsers including Chrome, Internet Explorer, and Mozilla Firefox.

The website will also be tested on different operating systems such as Linux, Windows, OS x (Apple-Macintosh), Android and iOS (iPhone).

.

# TRAINING NEEDS

Describe any necessary training required by the stakeholders involved in the testing process. The team/stakeholder involved should be specified along with the name of the training, its starting date and duration (a link to the training description may be useful).

# RISKS AND ASSUMPTIONS

**In this section, you need to specify potential risks that could impact on the testing process. Each risk needs to be associated to an id, a possible cause, an impact (or analysis), an impact level (e.g., low, medium, high), a probability of occurring (e.g., low, medium, high), and mitigation actions.**

**You also need to include assumptions on the project stakeholders and environment that can impact on the testing effort.**

## Risks

**Table: Testing Risks**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Risk** | **Analysis** | **Mitigation** | **Probability** | **Impact** |
| 1 | No time for testing, rushed and may not cover all test case scenarios | Danger of defects not being eliminated before production. | Test high-priority requirements first. | Medium | High |
| 2 | Not all test cases will be discovered before the website has been upgraded to the newest version | Danger of defects not being eliminated before production. | Any additional test cases discovered at a later date will be covered in the next upgrade of the website. | High | Medium |

## Assumptions

.

**Table 13: Testing Assumptions**

|  |  |
| --- | --- |
| # | **Assumption** |
| 1 | The users will be computer literate |
| 2 | A Trainer will be provided to train the QA team up in validation skills |

# DEFECTS MANAGEMENT

## Defects Process

This section details the Defect Management process that will be used to detect and monitor defects throughout the testing process (e.g., procedures, how the defects will be recorded, etc.).

## 11.2 Defect categories

With the knowledge gained from testing, the defects found will be categorised. Defects can be categorized into different types based on the core issues they address. Some defects address security or database issues while others may refer to functionality or UI issues.

**Security Defects:** Application security defects generally involve improper handling of data sentfrom the user to the application. These defects are the most severe and given highest priority for a fix.

*Examples:*

* Authentication: Accepting an invalid username/password
* Authorization: Accessibility to pages though permission not given

**Data Quality/Database Defects:** Deals with improper handling of data in the

Database (the database side of the Coursera website/app will be considered outside of the scope of this document).

*Examples:*

* Values not deleted/inserted into the database properly
* Improper/wrong/null values inserted in place of the actual values

**Critical Functionality Defects:** The occurrence of these bugs hampers the crucial functionalityof the application.

*Examples:*

* Exceptions

**Functionality Defects:** These defects affect the functionality of the application.

*Examples:*

* All JavaScript errors
* Buttons like Save, Delete, Cancel not performing their intended functions
* A missing functionality (or) a feature not functioning the way it is intended to
* Continuous execution of loops

**User Interface Defects:** As the name suggests, the bugs deal with problems related to UI areusually considered less severe.

*Examples:*

* Improper error/warning/UI messages
* Spelling mistakes
* Alignment problems

## 11.3 Documenting the Defect

Once the test cases are developed using the appropriate techniques, they are executed which is when the bugs occur. It is very important that these bugs be reported as soon as possible because, the earlier you report a bug, the more time remains in the schedule to get it fixed.

The defects will be reported as soon as they are detected. This is so that, the chances that it will be fixed are very high. If any defects are reported within a few hours of the release, the odds are that it won’t be fixed until the next version of the website is released.

Apart from finding the defects; these will be reported/communicated clearly and efficiently, so that the people who will be reading the defect can easily understand it.

The following is the procedure for documenting the defects:

* Identify the issue
* Identify the current configuration/version of the software
* Identify the severity (to what extend it affects the software)
* Identify when the problem occurs
* Identify the platform
* Identify the corresponding test cases/Requirements
* Record all relevant information related to the defect and submit it so that it can be investigated
* Steps to reproduced this issue
* Expected results
* Actual result

### 11.3.1 Defect Tracking Tool

The BugZero tool will be used to aid the testers in reporting and tracking the bugs/defects found in the software application. This will provide a means of consolidating a key element of project information in one place. The Project manager can then see which bugs have been fixed, which are outstanding and how long it is taking to fix defects. Senior management will be able to use reports to understand the state of the development process.

### 11.3.2 BugZero Defect Report

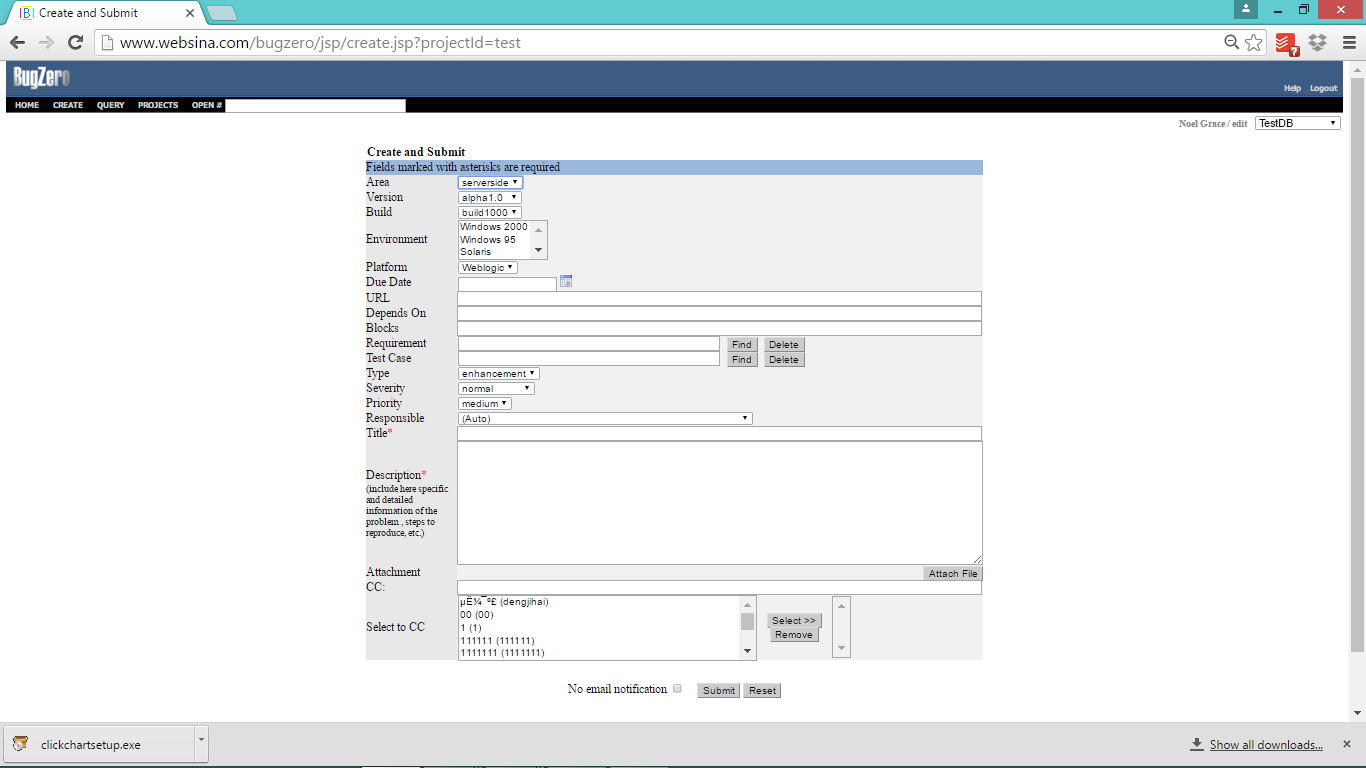
A defect report will be saved on BugZero. The report will provide enough detail while reporting the bug so that the people who will use it will be able to fully understand it – i.e. test lead, developer, project manager, other testers, new testers assigned etc. Therefore, the documented report will be concise, straight and clear. The following are the critical details that the BugZero Defect report will contain for the Coursera website:

* Defect Title
* Defect identifier (number, ID, etc.)
* Area
* The application name or identifier and version
* The function, module, feature, object, screen, etc. where the bug occurred
* Environment (Operating System, Browser and its version)
* Defect Type or Category/Severity/Priority
  + Defect Category: Security, Database, Functionality (Critical/General), UI
  + Defect Severity: Severity with which the bug/defect affects the application – normal, critical, serious, non-critical, show-stopper or minor.

High, Medium, Low, Very Low

* Defect Priority: Recommended priority to be given for a fix of this bug – high, low, medium or urgent.
* Defect status (Open, Pending, Fixed, Closed, Re-Open)
* Test case name/number/identifier
* Requirement name/number/identifier
* Defect description
* Steps to Reproduce
* Actual Result
* Tester Comments

### 11.3.3 BugZero Defect Report Template



11.4 Investigation  
The Engineers will try to reproduce the issue to assess whether it is a bug or the software is not used the way it should be (or just not a feature).  
  
11.5 Fixing 

* If the issue is a bug/defect, then the software is modified
* Relevant tests (if not already created are implemented
* The test documentation is updated (to include this test case)
* The status of the bug is updated (solved)
* The code change may be added to the next release note

## 11.6 Checking

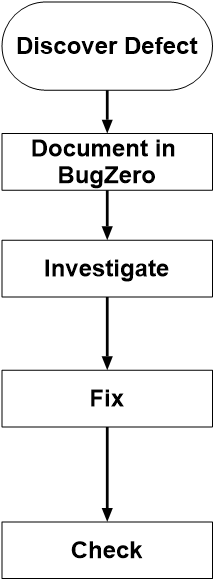
QA team will check that a solved issue can be closed  
  
11.7 Throughout the process  
Status of the issue is updated (new, investigated, solved, and closed)

## 

## 11.8 Fixing defects

Once the reported defect is fixed, the tester will re-test to confirm the fix. This is usually done by executing the possible scenarios where the bug can occur. Once retesting is completed, the fix can be confirmed and the bug can be closed. This marks the end of the bug life cycle.

**Defect Flow Chart**



# EXIT CRITERIA

The following table details the exit criteria checklist. This will be done to ensure the product is ready for release. Firstly, the test plan will have been completed in full and the document reviewed by a designated reviewer who is a member of the QA team. Any necessary amendments will have been rectified and/or updated before the Test Plan is signed off. Once the Test Plan has been signed off by the Reviewer (QA representative) then the Test Plan will be sent to the Manager and Project Manager pending final sign off/approval. After approval of the document has been achieved with this final sign off the status of the Test Plan will be moved to the completed stage.

After the Test Plan has been completed then and only then will the process move to the next stage i.e. testing. The Testers will perform/execute the test cases, and will progress through with the defect management procedure. If it is not possible within the timeframe to resolve all of the defects or to complete all of the test cases at this stage then those defects and test cases with the high priority will be resolved/implemented first. Any defects which have not been fixed or any test cases which may not have been implemented will be covered in the next version of the website.

## When can testing be stopped/reduced?

If testing is to be reduced then certain Test Cases/ Defects will be prioritized over other ones. The decision to stop testing or to complete priority Test Cases/ fix priority Defects will be based on the following factors:

* Deadlines (release deadlines, testing deadlines, etc.)
* Test cases completed with certain percentage passed
* Test budget depleted
* Coverage of code/functionality/requirements reaches a specified point
* Bug rate falls below a certain level
* Beta or alpha testing period ends

Once the Testing and Defects process have been completed then the product will be released.

**Table: Exit Criteria Checklist**

|  |
| --- |
| **Exit Criteria** |
| Test Plan completed |
| QA testing 100% Complete (All test cases have been executed at least once) |
| … |

# 

# 13 TEST CASES, PASS/FAIL CRITERIA AND TEST DATA

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Level** | XXX | **Test Date** | DD/MM/YY – DD/MM/YY |
| **Test Component(s)** | N/A | **Version** | 1.0 |
| **Tested By** | XXX | **Overall Result** | TBD |

|  |
| --- |
| **Test Preparation Details** |
| Describe any preparations that need to be performed prior to these tests |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Req ID #** | **Test Case (TC) No.** | **Test Description** | **Expected Result** | **Test Data** | **Result – (Pass or Fail)** | **Comments** |
| REQ\_SEC\_01 | **TC\_01** | 1. Open the URL 2. Enter the correct username/password details in the correct fields 3. Click sign in | The user will gain access to their personal profile on the site if the correct details have been entered. | URL = https://www.coursera.org/ |  | Manual positive test |
| REQ\_SEC\_01 | t  **TC\_02** | 1. Open the URL 2. Enter the incorrect username and/or password in the required fields 3. Click sign in | The user will not gain access to the site and the message incorrect details have been entered will be displayed. | URL = https://www.coursera.org/ |  | Manual negative test |
| REQ\_SCA\_01 | r  **TC\_03** | 1. Open the Load Impact URL 2. Enter your login credentials 3. Open the load impact web tool 4. Enter the URL to be tested 5. Enter a value of 100 in the max virtual users field 6. Enter a value of 5 into the duration (minutes) field 7. Click run test 8. When the test is complete click on the metrics tab 9. Check the results | The site will tolerate 100 or more users being logged in concurrently | URL 1 = https://app.loadimpact.com/  URL 2 = <https://www.coursera.org/>  Email = [noelgrace@gmail.com](mailto:noelgrace@gmail.com)  Password = secret  Max Virtual Users = 100  Duration (minutes) = 5 |  | Manual test |
| REQ\_LIKE\_01  REQ\_LIKE\_01.01  REQ\_LIKE\_01.02 | t  **TC\_04** | 1. Open the URL 2. Scroll to the footer section of the home page. 3. View the number of hits and visitors | The number of hits should be at least 100/day and the number of visitors should be at least 10/day. | URL = https://www.coursera.org/ |  | Manual |
| REQ\_LIKE\_01  REQ\_LIKE\_01.03  REQ\_LIKE\_01.05 | **TC\_05** | 1. Open the URL 2. Check the number of followers at the top of the page | The number of followers should be at least 1000 | URL = https://twitter.com/coursera |  | Manual |
| REQ\_LIKE\_01  REQ\_LIKE\_01.04 | **TC\_06** | 1. Open the URL 2. Check the number of members at the top of the 3. third row division | The number of members should be at least 500 | URL = https://www.facebook.com/groups/CourseraLearningHowtoLearn/ |  | Manual |
| REQ\_C\_01  REQ\_C\_01.01 | **TC\_07** | 1. Open the webpage validator site http://app.validator.pro/ 2. Enter the URL to test 3. Select spelling 4. Select US English 5. Click start 6. The validator will check the spelling of all the web pages for this site | The spelling should be 100% correct | URL 1 = <https://www.coursera.org/>  URL 2 = http://app.validator.pro/ |  | Manual |
| REQ\_NF\_01 | **TC\_08** | 1. Open the webpage validator site <http://app.validator.pro/> 2. Enter the URL to test 3. Select validate 4. Click start 5. The validator will check the HTML   content of all the pages for this site | The HTML should be 100% compliant | URL 1 = <https://www.coursera.org/>  URL 2 = http://app.validator.pro/ |  | Manual |
| REQ\_UE\_01,  REQ\_UE\_01.01, REQ\_UE\_01.02 | **TC\_09** | 1. Open the url 2. Enter the url of the website in the text field called “Free check for broken links” (url2) 3. Click on “Find Broken Links” 4. Enter the verification code. 5. Select the option “Report distinct broken links only” 6. Click on “Find Broken Links Now”. | After the test is complete, no broken links should be reported | URL 1 = <http://www.brokenlinkcheck.com/>  URL 2 = <https://www.coursera.org/> |  | Manual |
| REQ\_UE\_02 | **TC\_10** | 1. Type a URL that doesn’t exist into the browser address bar 2. Press enter | If the user requests a page that doesn’t exist on the server, s/he will be directed to the error 404 page automatically | URL = https://www.coursera.org/hfhty |  | Manual negative test |
| REQ\_UE\_03 | **TC\_11** | 1. Type a URL that does exist into the browser address bar 2. Press enter | If the user requests a page that does exist on the server, s/he will be directed to that page automatically | URL = https://www.coursera.org |  | Manual positive test |
| REQ\_UE\_04  REQ\_UE\_04.01 | **TC\_12** | 1. Open the web page test site 2. Enter the URL into the URL field 3. Select your test location from the drop down menu 4. Select your browser 5. Click start test | The page load speed should be less than 10 seconds | URL 1 = <http://www.webpagetest.org/>  URL 2 = https://www.coursera.org |  | Manual |
| REQ\_UE\_05  REQ\_UE\_05.01 | **TC\_13** | 1. Open the responsive design checker website 2. Enter your URL 3. Click go | The website should be responsive on desktops, laptops, tablets and phones | URL 1 = http://responsivedesignchecker.com/  URL 2 = https://www.coursera.org |  | Manual |
| REQ\_F\_01  REQ\_F\_01.01  REQ\_F\_01.02 | **TC\_14** | 1. Type information into the last name field in the correct format 2. Fill in all other fields correctly 3. Click submit | If the user enters information into the address bar in the correct format and has correctly filled in all other fields then a message “You have successfully registered. You will receive an email to the email address you provided and will need to click on the registration link to activate your account.” | URL = <https://www.coursera.org/>  Name= Noel  Last Name = Grace  Email = [noelgrace@gmail.com](mailto:noelgrace@gmail.com)  Phone Number = 051-422329 |  | Manual positive test |
| REQ\_F\_01 | **TC\_15** | 1. Type information into the last name field with only 9 characters 2. Click submit | A message “You must enter a minimum of 10 characters into the last name filed will appear in a pop up box”. | URL = https://www.coursera.org/ |  | Manual negative test |

## References

1. <https://en.wikipedia.org/wiki/Coursera>
2. Beginners Guide To Software Testing by Padmini C

## Abbreviations

1. MOOC - [Massive Open Online Courses](https://en.wikipedia.org/wiki/Massive_open_online_course)
2. URL -  Uniform Resource Locator (a reference an address to a resource on the Internet).