Virtual Classroom

Test Plan

COP 4331C, Fall, 2015

## **Modification History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Who | Comment |
| V0.0 | 10/01/2015 | J. Casserino | Initial Draft |

Team Name: Group 26

Team Members:

Joseph Bender [jbender94@knights.ucf.edu](mailto:jbender94@knights.ucf.edu)

Joshua Casserino [Joshua.casserino@knights.ucf.edu](mailto:Joshua.casserino@knights.ucf.edu)

Chad Armstrong [chad.armstro@knights.ucf.edu](mailto:chad.armstro@knights.ucf.edu)

Miles Friedman [milesfriedmanfl@gmail.com](mailto:milesfriedmanfl@gmail.com)

Contents of this Document

Introduction............................................................................................................................

Overall Objective for Software Test Activity............................................................

Reference Documents................................................................................................

Description of Test Environment...........................................................................................

Overall Stopping Criteria.......................................................................................................

Description of Individual Test Cases.....................................................................................

**Introduction - JOSH**

**Overall Objective for Software Test Activity**

A few sentences stating what you expect the software test effort to accomplish

**Reference Documents**

Concept of Operations <include link here>

Project Plan <include link here>

SRS <include link here>

<any other relevant documents; include full reference information or link>

**Description of Test Environment - JOSH**

< What is the hardware and software in the environment in which you will run the test? Who will be the testers (actual users? developers? ...) Will this test environment be the same environment in which the software will operate? If not, how does it differ?>

**Stopping Criteria**

During development of the application, the developers will be running their own unit tests as they develop to increase the quality of the code produced. During the testing phase, the team will test the software logging all defects and giving them a rank of severity in relation to the usability of the application. As issues and bugs are reported, the developer(s) will immediately begin addressing the highest severity issues and form a resolution plan to be approved with the remainder of the team before implementation. This will ensure that a comprehensive solution to the issue is to be implemented.

Multiple test cases will be run, feeding in 90% of all possible permutations of variable input for each functionality and component of the system. Once these test cases have all passed without error and the system is observed to behave correctly in all scenarios, then the software may be deemed ready for delivery. This will require that there are no known errors within the system. The test cases will involve completing all tasks for each functionality in the application and passing all permutations of possible input to the applications.

**Description of Individual Test Cases – ALL(10 CASES)**

Describe EACH individual test to be run: (so if you plan to run 20 test cases, you would answer the following questions for each of them. Either a tabular format or a bulleted format is acceptable.)

* Test Objective: <exactly what are does this specific test demonstrate?>
* Test Description: <exactly what will you test? What test data will be used (specifically -- what data values, what data files? This data must be determined in advance. So if you plan to use a test file, include a link to the file here. If you plan to key in specific data, include the data here. Make sure the data you choose will allow you to achieve your stated objective for this test.>
* Test Conditions: <Under what conditions will you run this test? This is relevant for software for which there are multiple "modes". For some systems, the test conditions are totally described in the above test environment section -- in this case, the response to this would be "See Test Environment".
* Expected Results: <If the test executes correctly, what will be the result -- i.e., exactly what will the output look like; what will be the resulting data in the database, ...>

Test Case 1 – Login

* Test Objective: Test the functionality of logging in to reject or grant access to the system based on input credentials.
* Test Description: The login functionality will be input with both false and valid username and password combinations. A test user name of TestUser@email.com and a password of Passw0rd in order to test login acceptance. Invalid combinations involving all permutations will be used to test login rejection.
* Test Condition: This is not applicable as this functionality is under a general mode for the application.
* Expected Results: For valid test username and password combinations that are in the database the system will log the user in and present the home page to them. Invalid usernames and passwords will be rejected further access to the system.

Test Case 2 – Create User Account

* Test Objective: to test the functionality of allowing the user to input information and create an account.
* Test Description: the create account will receive input with all ranges of data. The only character string of information that will require validation are the password and email fields. The test will provide both valid and invalid passwords and email addresses to test the applications data verification functionalities
* Test Condition: the application will be in a neutral mode as this functionality is available to both student and instructor users
* Expected Results: when the system receives valid information an account will be created and the user will receive verification that the account has been created. If the information provided is invalid against the constraints the user will receive notification that the account could not be created.

Test Case 3 – XXXXXXXX

* Test Objective:
* Test Description:
* Test Condition:
* Expected Results:

Test Case 4 – Create Course and Content

* Test Objective: testing the functionality for instructors/administrators to create a course module and populate it with relevant content
* Test Description: The test case will provide input into the content fields when creating a course. The data validation functionalities will be tested with invalid and valid data strings in fields such as Course Name, Course ID. This test case will also provide testing for publishing content into the files and assignments sections of the course.
* Test Condition: This test case will require the mode of administrators and instructor users.
* Expected Results: The application should create a course based on the input from the ‘create course’ functionality and if valid character strings are received for the course code, the application will create a new course with all provided data.

Test Case 5 – Student Select and Add Courses

* Test Objective:
* Test Description:
* Test Condition:
* Expected Results:

Test Case 6 – Student View and Submit Assignments

* Test Objective:
* Test Description:
* Test Condition:
* Expected Results:

Test Case 7 – Student have access to course data

* Test Objective:
* Test Description:
* Test Condition:
* Expected Results:

Test Case 8 – Instructor Create/Grade Assignments

* Test Objective:
* Test Description:
* Test Condition:
* Expected Results:

Test Case 9 – Student View Course Files

* Test Objective:
* Test Description:
* Test Condition:
* Expected Results:

Test Case 10 – Student Post/Create Discussion

* Test Objective:
* Test Description:
* Test Condition:
* Expected Results: