Virtual Classroom (Vlass)

Software Requirements Specification

COP 4331C, Fall, 2015

Modification History

Version	Date	Who	Comment
V 0.0	10/01/2015	J. Casserino	Initial Draft
V 0.1	10/06/2015	J. Bender	Rough Draft - Added Assumptions
V 0.2	10/07/2015	M. Friedman	Updated Rough Draft - Added Resource Requirements - Added Quality Assurance Requirements
V 0.3	10/07/2015	J. Casserino	Updated Rough Draft - Added Introduction - Added Stakeholders - Added Event Table - Added Physical Environment Requirements - Added Users and Human Factors Requirements - Added Documentation Requirements
V 0.4	10/08/2015	C. Armstrong	Updated Rough Draft - Added Data Requirements - Added Security Requirements
V 1.0	10/08/2015	J. Casserino	Proof read the Rough Draft and created the Final Draft

Team Name: Group 26

Team Members:

Joseph Bender	jbender94@knights.ucf.edu
Joshua Casserino	Joshua.casserino@knights.ucf.edu
Chad Armstrong	chad.armstro@knights.ucf.edu
Miles Friedman	milesfriedmanfl@gmail.com

Contents of this Documents

Introduction

Software to be Produced

The Virtual Classroom App is an iPhone app that will improve the current learning management system, Canvas. The system's main goal is to increase the availability of information the Students have directly from the Teachers, to improve the data integrity; and to increase the information the Teachers have about the current level of understanding of the students. The secondary goal is to improve the current system's ease of use for all users. With a direct line of communication between the Teacher and Students in real time, the Teacher will be able to cover sections more thoroughly. The real time lecture feed will also lessen the damages of out of school emergencies, which will plague everyone at least once in their academic career. With these improvements to an already successful system, the Virtual Classroom App will only improve the education of students and increase the proficiency of the teachers.

Reference Documents

Concept of Operations;

github.com/bender-joe/VirtualClassroom/tree/master/Documetation/Deliverable%201

Project Management Plan;

github.com/bender-joe/VirtualClassroom/tree/master/Documetation/Deliverable%201

Test Plan;

github.com/bender-joe/VirtualClassroom/tree/master/Documetation/Deliverable%201

Applicable Standards

The standards are the same as stated in the Project Management Plan.

Definition, Acronyms, and Abbreviations

iOS - an operating system used for mobile devices manufactured by Apple Inc.

PHP - Personal Home Page

MySQL - a full-featured relational database management system (RDBMS) that competes with the likes of Oracle DB and Microsoft's SQL Server.

LED - light-emitting diode

TA - Teacher's Assistance

GB - GigaByte

ID - Identification

App - Application

UML - Unified Modeling Language

UMlet - is an open-source Java-based UML tool designed for teaching the Unified Modeling Language and for quickly creating UML diagrams.

Product Overview

Assumptions

The mobile iOS application will interface with a PHP web-service that will query the MySQL database setup to house all of the applications data entities and relationship data. The iOS app will be deployed to an iPhone 6 running iOS 9.0.2 for purposes of demoing the application. The iPhone 6 features a 4.7" LED multi-touch display, an A8 chip with 64-bit architecture, and a capacity of 64 GB. The system user interface does inspire from user interface design concepts and process flows within Instructure's Canvas iOS application.

Stakeholders

The stakeholders for this project are primarily the group members; Joseph Bender, Joshua Casserino, Chad Armstrong and Miles Friedman. These members act as the developers and users of the project. Secondary stakeholders will be the TA's; Gurkan Solmaz and Anvesh Thota. These members will act as the regulatory agency. Finally, the professor of the course Dr. Damla Turgut, will be considered the customer.

Event Table

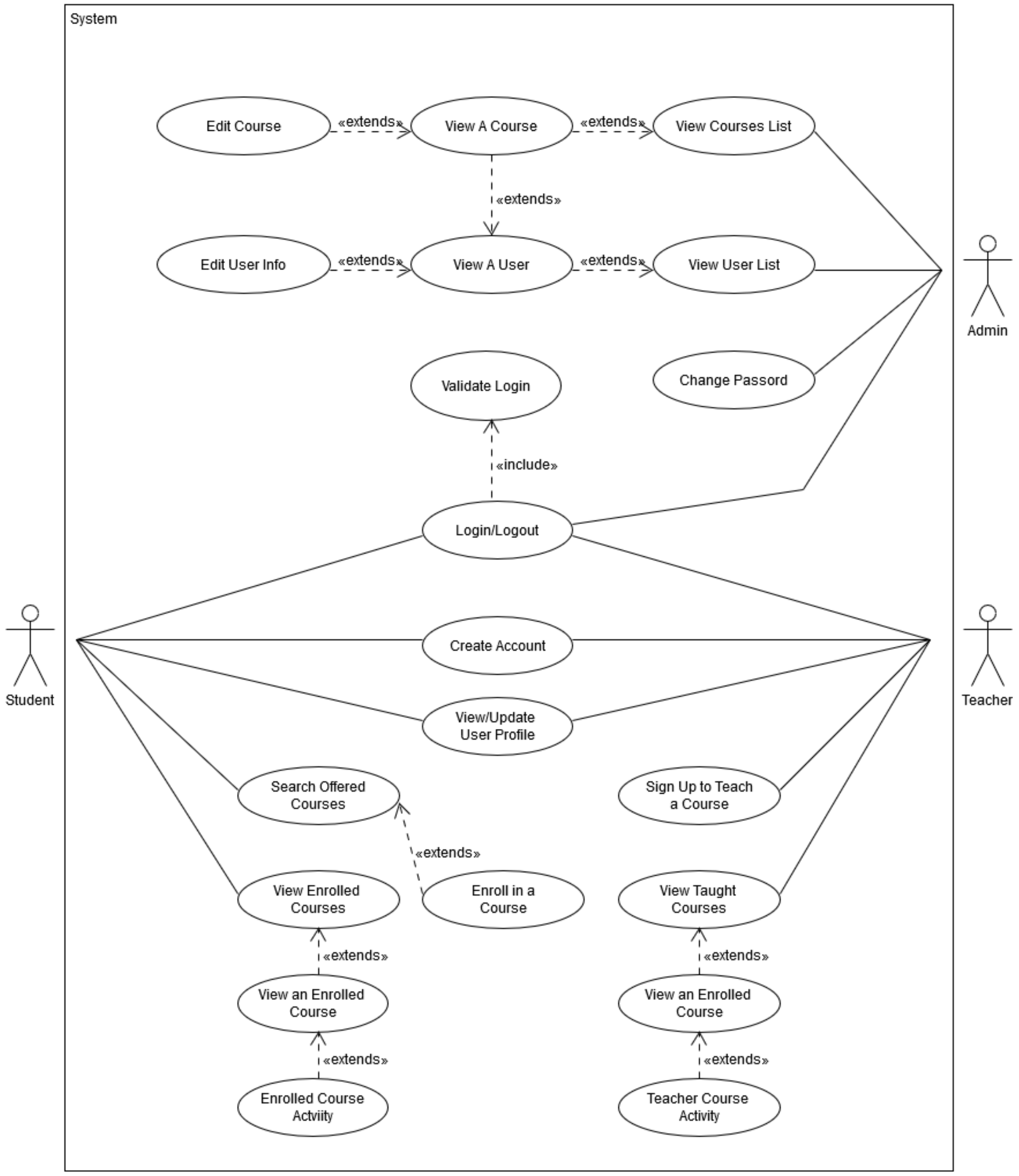
Event Name	External Stimuli	External Responses	Internal data and state
Creating user accounts and profiles	The Admin user will log into the system and create user accounts and profiles.	The new accounts and profiles will be created.	The program will update the database to add the accounts and profiles.
Creating a course	The Admin user will log into the system and create an account.	The new course will be created.	The program will update the database to add the new course.
Enrolling users in course	The Admin will update the Student user to add them to the course.	The account will be updated.	The program will update the database to include the course.
Users being able to log in (Successful)	A user will input their email address	The user will gain access to the system.	The system will check the credentials

	and password to gain access the system.		of the user and let them gain access.
Users being able to log in (Failure)	A user will input an email address and password to try and gain access the system.	The user will refuse access to the system.	The system will check the credentials of the user and deny access.
Users being able to upload/access course files (within profile limits)	The user will try and upload/access course data.	The user will be able to upload/access course data.	The system will check the credentials of the user and grant them access.
Users being able to upload/access course files (outside profile limits)	The user will try and upload/access course data.	The user won’t be able to upload/access course data.	The system will check the credentials of the user and deny them access.
Privacy of personal data (within profile limits)	The user will try and access other user data.	The user will be able to access user data.	The system will check the credentials of the user and grant them access
Privacy of personal data (outside profile limits)	The user will try and access other user data.	The user will be denied access to user data.	The system will check the credentials of the user and deny them access

Teacher being able to make/grade assignments (Successful)	The Teacher user will try and assign a grade/assignment to the Student user.	The Teacher user will be able to assign a grade/assignment to the Student user.	The system will check the credentials of the user and grant them access.
Teacher being able to make/grade assignments (Failure)	The Teacher user will try and assign a grade/assignment to the Student user.	The Teacher user won’t be able to assign a grade/assignment to the Student user.	The system will check the credentials of the user and deny them access.
Course discussion section (Successful)	The user will try and access the discussion section.	The user will be able to access the discussion section.	The system will check the credentials of the user and grant them access.
Course discussion section (Failure)	The user will try and access the discussion section.	The user won’t be able to access the discussion section.	The system will check the credentials of the user and deny them access.
Teacher being able to create an interactive poll (Successful)	The Teacher user will try and create an interactive poll.	The Teacher user will create an interactive poll.	The system will check the credentials of the user and grant them access.

Teacher being able to create an interactive poll (Failure)	The Teacher user will try and create an interactive poll.	The Teacher user won’t be able to create an interactive poll.	The system will check the credentials of the user and deny them access.

Use Case Diagram



Use Case Descriptions

Shared

Students, Teachers and Admins should have the ability to login and logout of the system. Teachers and Students will share the ability to create accounts and update their user profile. Because Admin simply plays the role of a super user, the Admin user will not have a user profile and there is no need for the Admin user to have to create an account, a default one will be provided.

Teacher

Teachers will have the ability to sign up to teach a class. Once they do this, the class they create and sign up to teach will be listed when they view their courses being taught this semester. From here they will have the ability to select one of their courses being taught and dive into multiple options. These options are all encompassed by teacher course activity in the use case diagram and include grading students assignments, posting assignments, making announcements to the class, uploading files such as the syllabus and lectures for Students to download, closing the course, and possibly more depending on where we decide to go with the project and how our time constraints affect production.

Students

Students will have the ability to search courses offered by the school they are in by either viewing all courses offered, or searching based on course ID. From the viewed selection they may then enroll in a course of their choosing. This course will be listed along with the rest of a Student's enrolled courses when they choose to view their enrolled courses. Once viewing a list of all their currently enrolled courses, they may select to view a specific enrolled course which will open up a plethora of additional options pertaining to the selected class, labeled as enrolled course activity. The options will include: viewing their grades, submitting assignments, messaging the teacher, posting to a class discussion board, downloading files that the teacher uploaded, dropping the class and possibly more depending on where we decide to go with the project and how our time constraints affect production.

Admin

The Admin will have the ability to search through the database and view all users of the system. This will be a list including all Teachers and Students from which they may select/view a specific user. The will then have the option to either view a specific course associated with that Student, Teacher or edit that specific users profile information. (This could entail changing the users' password, associated school, dropping a class, changing a grade, and possibly more) The Admin will also have the ability to search and view the list of courses offered by the school, from which they may also view a specific course. Once the Admin has accessed a course, they will have many options which are encompassed in the diagram by Edit Course. These options will include all options that both Teachers and Students have, and possibly more.

Specific Requirements

Functional Requirements

No: 001
Statement: The system shall allow users to login to the system using their email address and created password.
Source: Login
Dependency: 002

Conflicts: None
Supporting Materials: *insert here*
Evaluation Method: A test case is fully executed where an account is created with an email address and password. Then the password and email address are input at the Login screen, and the application presents the user with their profile home page.
Revision History: J. Bender, 10/6/2015, Rough Draft

No: 002
Statement: The system shall allow a student to create an account by providing their first name, last name, school, email address (as login ID), and a valid password against the security constraints of must contain 8 characters, must contain at least one numeric character, and must contain at least one uppercase letter. This information will be saved and the user will receive validation that the account creation was either successful or not.
Source: Student Account Creation
Dependency: None
Conflicts: None
Supporting Materials:
Evaluation Method: After the input information is received, either a new account is created (if one does not already exist under the provided email address) and the information is stored in the database, and the user receives a message on screen verifying account creation. If an account is already in place in the database under the email address provided, then the user shall receive an onscreen message that the account could not be created because the email is already in use.
Revision History: J. Bender, 10/6/2015, Rough Draft

No: 003
Statement: The system shall allow an Teacher to create an account by providing their first name, last name, school, email address (as login ID), and a valid password against the security

constraints of must contain 8 characters, must contain at least one numeric character, and must contain at least one uppercase letter. This information will be saved and the user will receive validation that the account creation was either successful or not.
Source: Teacher Account Creation
Dependency: None
Conflicts: None
Supporting Materials:
Evaluation Method: After the input information is received, either a new account is created (if one does not already exist under the provided email address) and the information is stored in the database, and the user receives a message on screen verifying account creation. If an account is already in place in the database under the email address provided, then the user shall receive an onscreen message that the account could not be created because the email is already in use.
Revision History: J. Bender, 10/6/2015, Rough Draft

No: 004
Statement: The system shall allow students to browse courses available at their school and add select ones to their profile by course name and ID.
Source: Student Course Enrollment
Dependency: 001, 002
Conflicts: None
Supporting Materials:
Evaluation Method: The system displays the available courses in a scrollable window on screen and also allows the users to enter in the course name and ID into a text field to search for the course. A course will be able to be selected and added to the user’s profile. These changes should be reflected in the user’s home page, and within the database.

Revision History: J. Bender, 10/6/2015, Rough Draft

No: 005
Statement: The system shall allow Teachers to create a course they are instructing at their school of employment. The input should include the course name and ID, the ability to upload files, and the ability to create assignments.
Source: Teacher Course Creation
Dependency: 001, 003
Conflicts: None
Supporting Materials:
Evaluation Method: The system requires a course name and ID to be input, and all aspects described above are able to be input for the course. All of these updates and changes for the course should reflect on the Teacher’s home page and in the database.
Revision History: J. Bender, 10/6/2015, Rough Draft

No: 006
Statement: The system shall allow student users to view and download files for each course.
Source: Students Viewing/Downloading Course Files
Dependency: 001, 003, 004
Conflicts: None
Supporting Materials:
Evaluation Method: This requirement does not require any input. The system will show a list of files for a specific course, and a user should be able to download a file (docx, pdf, pptx) and view the content on the device.
Revision History: J. Bender, 10/6/2015, Rough Draft

No: 007
Statement: The system shall allow student users to view and submit assignments for each course.
Source: Student Assignment View/Submit
Dependency: 001, 003, 004
Conflicts: None
Supporting Materials:
Evaluation Method: This requirement will show the user a screen displaying all of the current assignments for the course. A file of the specified type will be able to be submitted electronically.
Revision History: J. Bender, 10/6/2015, Rough Draft

No: 008
Statement: The system shall allow student users to view their current grades for a course.
Source: Student View Grades
Dependency: 001, 003, 004
Conflicts: None
Supporting Materials:
Evaluation Method: This requirement does not require any input. The system will show a list tabular view displaying all of the current grades the student has received from the Teacher for each assigned item in the course.
Revision History: J. Bender, 10/6/2015, Rough Draft

No: 009
Statement: The system shall allow student users to create, view, and post on discussion boards
Source: Student Discussion Post
Dependency: 001, 003, 004
Conflicts: None
Supporting Materials:

Evaluation Method: This requirement will provide a screen to view all existing discussions in the course, and provide a text field for users to append a comment on the discussion topic. The system will also provide a functionality to create an entirely new discussion thread with a title and other pertinent details.
Revision History: J. Bender, 10/6/2015, Rough Draft

No: 010
Statement: The system shall allow Teacher users to view assignment submissions and provide a grade for them.
Source: Teacher Grade Assignments
Dependency: 001, 003, 005, 007
Conflicts: None
Supporting Materials:
Evaluation Method: The system will allow for an Teacher account to view all of the submissions for an assignment, and provide a grade for the assignment.
Revision History: J. Bender, 10/6/2015, Rough Draft

Interface Requirements

The system will store and retrieve data using a local SQL database.

Input

- Username - Unique string must be identical to username in database. Created by the user when request an account.
- Password - Unique string must be identical to password in database. Created by the user when request an account.
- Text - Any text. Student and Teacher users input profile data. Admin users will input information about courses upon course creation.
- Files - Any text, document or picture file. Available whenever a user access the relevant course there attached too.

- Discussion post - Any text. Created by the Student or Teacher users.
- Grades - A number value. Uploaded by the Teacher user.
- Poll - Button click. Student users will select one of a number of preselected button to click.

Output

- Files - Any text, document or picture file. Available whenever a user accesses the relevant course there attached too.
- Discussion post - Any text. Available whenever a user accesses the relevant course there attached too.
- Grades - A number value. Available only to the Student profile they are attached too. Also available to the Teacher user of that course.
- Poll - Bar graph. Available to Student and Teacher users upon the completion of an interactive poll.

Physical Environment Requirements

The Virtual Classroom App will run on an iPhone 6 in the beginning but once all tests are complete the app will be available to other iPhone models. The MySQL database and the PHP web-service will require a server, which will be in the custody of a third party agency that has yet to be determined. Because of this the physical location will is unknown and the environmental requirements will be decided by the agency the server is in the custody of.

Users and Human Factors Requirements

The different types of users and their required skill level are;

Student - This user will only need a minimal understanding of how to use a computer app. The most complex task that will be asked of the user is the ability to upload documents to the system and to remember their credentials.

Teacher - This user will need a moderate understanding of how to this computer app. The Teacher type users will need to be able set up polls, assignments, assign grades, and upload documents.

Admin - This user will need an advanced skill level with this computer application. The Admin user type will be required to access the database, create profiles/users, create courses, update profiles, upload data to the database, provide database maintenance and must be able to complete all the other tasks other users need to complete.

Documentation Requirements

Once the Virtual Classroom App is complete, a short user guide will be created to assist the users in operating the system. The app itself will be easy to use without instruction as the interface will be easily navigable. The only user required to have advance knowledge of any computer systems will be the Admin user, which the user guide will cover what is required for that user to know.

Data Requirements

The application itself will not contain any data calculations, for it just takes in and displays the data. The data that gets retained by the system would be the grades that are entered by the professor, of which no calculations or digit precision is not necessary to implement, for the grades are just for viewing and are as precise as the professor dictates. This data will be stored in the database, so it won’t need to be rendered or calculated, just taken from the server that contains the database.

Resource Requirements

The skilled personnel required to build our app include: Joe Bender – Our project manager and lead developer. Miles Friedman – Who will be developing the IPhone app using Swift and Xcode, Chad Armstrong – Who will be in charge of setting up a database and linking it to our app, and Josh Casserino – Who will be lead on project organization and documentation. All members will help where they are needed but these are where our focuses will lie.

As far as workstations, we plan on using each of our homes along with the UCF library. Anywhere there is a power outlet may also be used as a workstation. We have arranged to meet every Tuesday, along with possible meetings on Thursdays and/or Fridays if needed. We plan to follow the timeline described in our Deliverables 1 and will schedule meetings every week in order to meet our milestones.

Our hardware resources are the Macbooks and IPhones belonging to Joe and Miles. We will develop our app using the XCode compiler/development environment and will write the code using Apple’s language: Swift. For storing information to be used in conjunction with the app, we will implement a database using the MySQL relational database management software and will write our database queries using the SQL database language. Our documentation will be created using Microsoft Word for text documents, and UMlet for UML diagrams. For issue tracking and quality assurance we will use LeanTesting, a website designed to track bugs or document issues with your code that need attention. Finally, we will use a GitHub repository for both version control and a place where we store our code for access by all developers.

Security Requirements

The security of this application must adhere to the FERPA requirements, restricting access of content between each student and a professor's ability to access information. All student data should be able to be accessed by that student or by their professor, so the data for each student must be isolated, and access to the system must be controlled. This problem can mainly be solved through a secure log in feature, which will limit access to the users properly. This system should be backed up and securely store its back up once a month in a different location then the server, and should encrypt its data in the process, this will ensure the FERPA requirements are met. As far as physical security from fire, water or theft, if the server catches fire, all the data will be backed up separately, the application will be run on mobile devices, so the security of the system will remain intact if the mobile device catches fire. A theft of the mobile device would be pointless due to the need to log in to use the application, and a theft of the back-up would be encrypted to retain the security of the system.

Quality Assurance Requirements

Our mobile app should be accessible through anyone's IPhone device. Users should have access to the system and database from anywhere so long as they have they have service or a Wi-Fi connection. This will easily be demonstrated anytime a user tries to access the application. The system should also secure a user's information by only allowing access through a login system which will access a secure database to pull user information. This will be demonstrated by preventing entrance to a user's profile until correct user information is inputted at the login screen and by testing the login feature against SQL injections.

The system will not have the capability to detect or react to faults in the system. If the application fails however, it should restart once the user exits and reopens the app. The application should allow users to switch between pages instantly with small loading times and have continued access to the database server as needed.

Supporting Material

None

Virtual Classroom

Test Plan

COP 4331C, Fall, 2015

Modification History

Version	Date	Who	Comment
V 0.0	10/01/2015	J. Casserino	Created Initial Draft
V 0.1	10/06/2015	J. Casserino	Updated Rough Draft - Added the Introduction - Added Description of Test Environment
V 0.2	10/08/2015	J. Casserino J. Bender	Updated Rough Draft - Added Overall Stopping Criteria - Description of Individual Test Cases
V 1.0	10/08/2015	J. Casserino	Check for errors and updated Rough Draft to Final Draft

Team Name: Group 26

Team Members:

Joseph Bender	jbender94@knights.ucf.edu
Joshua Casserino	Joshua.casserino@knights.ucf.edu
Chad Armstrong	chad.armstro@knights.ucf.edu
Miles Friedman	milesfriedmanfl@gmail.com

Contents of this Document

Introduction

Overall Objective for Software Test Activity

The initial software tests are to make sure that we have a working and bug-free application before issuing the software to customers. We hope to identify any flaws or possible improvements in the program during these tests in an effort to make the product release ready.

Reference Documents

Concept of Operations;

github.com/bender-joe/VirtualClassroom/tree/master/Documetation/Deliverable%201

Project Plan;

github.com/bender-joe/VirtualClassroom/tree/master/Documetation/Deliverable%201

SRS;

github.com/bender-joe/VirtualClassroom/tree/master/Documetation/Deliverable%202

Description of Test Environment

An iPhone with a current operating system and internet access will be used for testing. The testers will be team and non-team members, who are non-developers, to use the software objectively without knowing all the ins-and-outs of the implementation. The developers will perform tests in developed functionalities and performance, to test individual features, before beta testing. The environment for development will be the same as the live environment.

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

Test Case 1 – Login

- Test Objective: This will test the functionality of logging in, and reject or grant access to the system based on input credentials.
- Test Description: The login function will have inputs that contain 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: Testing the functionality of allowing the user to input information and create an account.
- Test Description: Create account function 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 an email addresses to test the applications data verification functionalities. Once the Teacher or Student user request a new account the Admin user will either approve or deny the request.
- Test Condition: The application will be in a neutral mode as this functionality is available to both Student, Teacher and Admin 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 – Privacy of personal data

- Test Objective: This test will check the application’s security.
- Test Description: A Student user will attempt to access other Student users personal info (Email address, Grades. Etc.).
- Test Condition: The application will be in a Student mode as this functionality only requires Student users.
- Expected Results: The system will not allow the personal data to be accessed.

Test Case 4 – Create Course and Content

- Test Objective: Testing the functionality for Teacher and Admin users 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 and 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 Admin and Teacher users.
- Expected Results: The application should create a course based on the input from the ‘create course’ function 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: Testing the functionality for Students to select an available course and add it to their inventory of courses.
- Test Description: The test case will simulate a Student user searching and adding a course to their available course list. The system will then inform the Teacher

user of that Student is trying to access the course. Teachers can accept or deny access to that course.

- Test Condition: This test case will require the mode of Students and Teacher users.
- Expected Results: If the Teacher user allows the application should add the course to the Students available/current courses list otherwise the application will reject the request.

Test Case 6 – Students Submit Assignments

- Test Objective: Testing the functionality for Students to view and upload assignments available for that course.
- Test Description: This test case will simulate a Student user selecting to view available assignment. It will also test if the Student will be able to upload completed assignments to the course.
- Test Condition: This test case will require the mode of Students users.
- Expected Results: The application should allow the user access to view and upload assignments.

Test Case 7 – Student have access to course data

- Test Objective: Testing the functionality for Students to have access to an available course.
- Test Description: The test case will simulate a Student user selecting to access the course and the course's material of a previously approved course.
- Test Condition: This test case will require the mode of Students users.

- Expected Results: The application should allow access to the available course and its available content.

Test Case 8 – Teacher Create/Grade Assignments

- Test Objective: Testing the functionality for the Teacher user to create an assignment for a course.
- Test Description: The test will simulate a Teacher user uploading an assignment into a course and inputting requirements (i.e. Due Date, Date Available, File type, etc.).
- Test Condition: This test case will require the mode of Teacher users.
- Expected Results: The application should upload the assignment and give Students currently in the course access to the relevant content.

Test Case 9 – Creating an Interactive Poll

- Test Objective: Testing the functionality for Teacher users to create an interactive poll for that course.
- Test Description: The test case will simulate a Teacher user selecting to view create an interactive poll. Once the poll is created a Student user will interact with poll.
- Test Condition: This test case will require the mode of Students and Teacher users.
- Expected Results: The application should create a poll and allow the Student user access to view content.

Test Case 10 – Student Post/Create Discussion

- Test Objective: Testing the functionality for Student users to create a discussion and/or post to a current discussion for the courses.
- Test Description: The test will simulate Student users creating a new discussion and/or post to a current discussion attached to that course.
- Test Condition: This test case will require the mode of Student users.
- Expected Results: The application should upload the discussion and/or discussion post and give Students currently in the course access to the relevant content.