f f

Test Case specification

Capstone Course Evaluation System

Version 3

By: Abigail Noyes, Bharath Palanisamy,

Cristina Powers, Raad Bhuiyan

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Authors** | **Comments** |
| 04/02/2023 | Version 1 | Abigail Noyes,  Bharath Palanisamy,  Crsitina Powers,  Raad Bhuiyan | Initial draft of the test case specification |
| 04/07/2023 | Version 2 | Abigail Noyes,  Bharath Palanisamy,  Crsitina Powers,  Raad Bhuiyan | Second draft of the test case specification |
| 04/16/2023 | Version 3 | Abigail Noyes,  Bharath Palanisamy,  Crsitina Powers,  Raad Bhuiyan | Final draft of the test case specification |

Table of Contents

[1.Introduction 8](#_Toc132871391)

[2. Functional Test Cases 8](#_Toc132871392)

[2.1 Approach 8](#_Toc132871393)

[2.4 Suspension / Resumption Criteria 8](#_Toc132871394)

[2.2 Pass/Fail Criteria 8](#_Toc132871395)

[2.3 Entry / Exit Criteria 8](#_Toc132871396)

[2.5 Risks / Issues 8](#_Toc132871397)

[2.6 User Sign In 9](#_Toc132871398)

[2.6.1 Sign in – Admin Successful 9](#_Toc132871399)

[2.6.2 Sign in – Professor Successful 9](#_Toc132871400)

[2.6.3 Sign in – GTA Successful 9](#_Toc132871401)

[2.6.5 Sign in– Incorrect Email 10](#_Toc132871402)

[2.6.6 Sign in – No Password 11](#_Toc132871403)

[2.6.7 Sign in – Incorrect Password 11](#_Toc132871404)

[2.6.8 Sign in – Inactive User 11](#_Toc132871405)

[2.7 User Sign Out 12](#_Toc132871406)

[2.7.1 Sign Out - Button 12](#_Toc132871407)

[2.7.2 Sign Out – Close Browser 12](#_Toc132871408)

[2.8 User Set Password 13](#_Toc132871409)

[2.8.1 User Set Password – Successful 13](#_Toc132871410)

[2.8.2 User Set Password – No Email 13](#_Toc132871411)

[2.8.3 User Set Password – Invalid Email 14](#_Toc132871412)

[2.8.4 User Set Password – No Password & No Password Confirmation 15](#_Toc132871413)

[2.8.5 User Set Password – No Password & Password Confirmation 15](#_Toc132871414)

[2.8.6 User Set Password – Password & No Password Confirmation 15](#_Toc132871415)

[2.8.7 User Set Password – Password with different Password Confirmation Entry 16](#_Toc132871416)

[2.9 User Forgot Password 16](#_Toc132871417)

[2.9.1 User Forgot Password – Successful 16](#_Toc132871418)

[2.9.2 User Forgot Password – No Email 17](#_Toc132871419)

[2.9.3 User Forgot Password – Invalid Email 18](#_Toc132871420)

[2.9.4 User Forgot Password – No Password & No Password Confirmation 18](#_Toc132871421)

[2.9.5 User Forgot Password – No Password & Password Confirmation 19](#_Toc132871422)

[2.9.6 User Forgot Password – Password & No Password Confirmation 19](#_Toc132871423)

[2.9.7 User Forgot Password – Password with different Password Confirmation Entry 20](#_Toc132871424)

[2.10 Create a Semester 20](#_Toc132871425)

[2.10.1 Create new semester – Successful 20](#_Toc132871426)

[2.10.2 Create new semester – Missing Semester Name 21](#_Toc132871427)

[2.10.3 Create new semester – Missing Semester Start Date 21](#_Toc132871428)

[2.10.4 Create new semester – Missing Semester End Date 22](#_Toc132871429)

[2.10.5 Create new semester – Existing Semester 23](#_Toc132871430)

[2.11 Create a Section 24](#_Toc132871431)

[2.11.1 Admins creates a new section - Successful 24](#_Toc132871432)

[2.11.2 Admin creates a new section – Missing section name 24](#_Toc132871433)

[2.11.3 Admins creates a new section – Missing semester criteria 25](#_Toc132871434)

[2.11.4 Admin creates a new section – Missing professor 26](#_Toc132871435)

[2.11.5 Admin creates a new section – Existing section 26](#_Toc132871436)

[2.12 Add Professor 27](#_Toc132871437)

[2.12.1 Add New Professor - Successful 27](#_Toc132871438)

[2.12.2 Add New Professor – Missing Form Requirements 28](#_Toc132871439)

[2.12.3 Add New Professor – With Existing Email 28](#_Toc132871440)

[2.13 Edit Professor 29](#_Toc132871441)

[2.13.1 Edit Professor – Successful 29](#_Toc132871442)

[2.13.2 Edit Professor – Input Blank Name 30](#_Toc132871443)

[2.13.3 Edit Professor – Input Any Email 30](#_Toc132871444)

[2.14 Deactivate Professor 31](#_Toc132871445)

[2.14.1 Deactivate Professor – Successful 31](#_Toc132871446)

[2.14.2 Deactivate Professor – Close Modal 32](#_Toc132871447)

[2.15 Add Section 32](#_Toc132871448)

[2.15.1 Add Section (Professor)- Successful 32](#_Toc132871449)

[2.15.2 Add Section (Professor)- Missing section name 33](#_Toc132871450)

[2.16 Add GTA 33](#_Toc132871451)

[2.16.1 Add GTA – Successful 33](#_Toc132871452)

[2.16.2 Add GTA – Missing Form Requirements 33](#_Toc132871453)

[2.16.3 Add GTA - With Existing Email 34](#_Toc132871454)

[2.17 Edit GTA 35](#_Toc132871455)

[2.17.1 Edit GTA – Successful 35](#_Toc132871456)

[2.17.2 Edit GTA – Input Incorrect Data 35](#_Toc132871457)

[2.18 Deactivate GTA 36](#_Toc132871458)

[2.18.1 Deactivate GTA – Successful 36](#_Toc132871459)

[2.18.2 Deactivate GTA – Close Modal 37](#_Toc132871460)

[2.19 Add Students 37](#_Toc132871461)

[2.19.1 Add Students- Successful 37](#_Toc132871462)

[2.19.2 Add Students- Existing Student in Section 38](#_Toc132871463)

[2.19.3 Add Students- Invalid Access Id 39](#_Toc132871464)

[2.20 Delete Student 39](#_Toc132871465)

[2.20.1 Delete Student - Successful 39](#_Toc132871466)

[2.21 Create Groups 40](#_Toc132871467)

[2.21.1 Create Groups- Successful 40](#_Toc132871468)

[2.21.2 Create Groups- Existing Group 40](#_Toc132871469)

[2.22 Assign Students to a Group 41](#_Toc132871470)

[2.22.1 Assign Students to a Group - Successful 41](#_Toc132871471)

[2.23 Edit Groups 41](#_Toc132871472)

[2.23.1 Edit Groups - Successful 41](#_Toc132871473)

[2.23.2 Edit Groups – Edit Group Name 42](#_Toc132871474)

[2.23.3 Edit Groups – Edit Assigned GTA 43](#_Toc132871475)

[2.23.4 Edit Groups – Remove Group Members 44](#_Toc132871476)

[2.24 View Student 44](#_Toc132871477)

[2.24.1 View Student – Successful 44](#_Toc132871478)

[2.25 View Grades 45](#_Toc132871479)

[2.25.1- View Grades (professor) – successful 45](#_Toc132871480)

[2.25.1- View Grades (professor) – Unsuccessful 45](#_Toc132871481)

[2.26 Pick Group 46](#_Toc132871482)

[2.26.1- Pick Group (GTA)- successful 46](#_Toc132871483)

[2.27 View Group 46](#_Toc132871484)

[2.27.1- View Group (GTA)- successful 46](#_Toc132871485)

[2.28 Grade weekly reports 47](#_Toc132871486)

[2.28.1 Grade weekly reports- Successful student report entry 47](#_Toc132871487)

[2.28.2 Grade weekly reports- Missing submission criteria 48](#_Toc132871488)

[2.28.3 Grade weekly reports - Missing status criteria 48](#_Toc132871489)

[2.28.4 Grade weekly reports - Missing evaluation criteria 49](#_Toc132871490)

[2.28.5 Grade weekly reports- Successful group report entry 50](#_Toc132871491)

[2.28.6 Grade weekly reports - unsuccessful group report entry 51](#_Toc132871492)

[2.28.7 Grade weekly reports- Successful jump to summary work 51](#_Toc132871493)

[2.29 Grade Assignments (Professor Interface) 52](#_Toc132871494)

[2.29.1 Grade Assignments (Professor Interface)- successful entry (Group and student entry) 52](#_Toc132871495)

[2.29.2 Grade Assignments (Professor Interface)- unsuccessful entry (Group and student entry) 53](#_Toc132871496)

[2.29.3 Grade Assignments (GTA Interface)- Successful entry (Group and student entry) 54](#_Toc132871497)

[2.29.4 Grade Assignments (GTA Interface)- Unsuccessful entry (Group and student entry) 55](#_Toc132871498)

[3. Non-Functional Test Cases 56](#_Toc132871499)

[3.1 Approach 56](#_Toc132871500)

[3.2 Pass / Fail Criteria 56](#_Toc132871501)

[3.3 Entry / Exit Criteria 56](#_Toc132871502)

[3.4 Suspension / Resumption Criteria 56](#_Toc132871503)

[3.5 Risks / Issues 56](#_Toc132871504)

[3.6 Non-functional Test Cases 57](#_Toc132871505)

[3.6.1 Database Performance 57](#_Toc132871506)

[3.6.2 Reliability 57](#_Toc132871507)

[3.6.3 Availability 58](#_Toc132871508)

[3.6.4 Security 58](#_Toc132871509)

[3.6.5 Maintainability 59](#_Toc132871510)

[3.6.6 Portability 59](#_Toc132871511)

[3.6.6 Usability 60](#_Toc132871512)

[4. Integration Testing 61](#_Toc132871513)

[4.1 Approach 61](#_Toc132871514)

[4.2 Pass / Fail Criteria 61](#_Toc132871515)

[4.3 Entry / Exit Criteria 61](#_Toc132871516)

[4.4 Suspension / Resumption Criteria 61](#_Toc132871517)

[4.5 Risks / Issues 61](#_Toc132871518)

[4.6 Integration Test Cases 62](#_Toc132871519)

[4.6.1 Integration between User Authentication and Role-Based Access Control 62](#_Toc132871520)

[4.6.2 Integration between Course Information and Evaluation Data 62](#_Toc132871521)

[4.6.3 Integration between Evaluation Data and Database 63](#_Toc132871522)

[4.6.4 Integration between Professor and GTA Views 64](#_Toc132871523)

[5. System Testing 65](#_Toc132871524)

[5.1 Approach 65](#_Toc132871525)

[5.2 Pass / Fail Criteria 65](#_Toc132871526)

[5.3 Entry / Exit Criteria 65](#_Toc132871527)

[5.4 Suspension / Resumption Criteria 65](#_Toc132871528)

[5.5 Risks / Issues 65](#_Toc132871529)

[5.6 System Testing – Admin User 66](#_Toc132871530)

[5.6.1 Sign in - Admin 66](#_Toc132871531)

[5.6.2 Add Professor - Admin 66](#_Toc132871532)

[5.6.3 Create a Semester - Admin 66](#_Toc132871533)

[5.6.4 Create a Section - Admin 67](#_Toc132871534)

[5.7 System Testing – Professor User 68](#_Toc132871535)

[5.7.1 Sign in - Professor 68](#_Toc132871536)

[5.7.2 Add Section - Professor 68](#_Toc132871537)

[5.7.3 Add GTA - Professor 68](#_Toc132871538)

[5.7.4 Add Students - Professor 69](#_Toc132871539)

[5.7.5 Add Group - Professor 70](#_Toc132871540)

[5.7.6 Assign Students to Group - Professor 70](#_Toc132871541)

[5.7.7 Grade Assignments - Professor 71](#_Toc132871542)

[5.8 System Testing – GTA User 71](#_Toc132871543)

[5.8.1 Sign in - GTA 71](#_Toc132871544)

[5.8.2 Choose Group - GTA 72](#_Toc132871545)

[5.8.3 Grade Assignments - GTA 72](#_Toc132871546)

[5.8.3 Grade Weekly Report - GTA 73](#_Toc132871547)

[Schedule 74](#_Toc132871548)

[Traceability Matrix 77](#_Toc132871549)

[Appendix 79](#_Toc132871550)

# 1.Introduction

1.1 Purpose

This document outlines the execution of test cases and verifies that the Capstone Course Evaluation System adheres to the requirements outlined in the Telehealth Requirements Specification. Its purpose is to ensure that the system is thoroughly tested and meets all the necessary specifications.

1.2 References

Each test case presented below is associated with a use case outlined in the Design Specification, making it easier to trace and analyze the system’s behavior during testing. The tracing is also illustrated in the usability matrix that is provided in the appendix at the bottom of the document.

# 2. Functional Test Cases

## 2.1 Approach

To perform functional testing, SQL scripts are utilized to reset the database and populate it with specific data for each test case. The tester then follows step-by-step instructions outlined in the test case and examines the results. By adopting this approach, every test is carried out using a controlled and predictable dataset, which is ideal for functional testing purposes.

## 2.4 Suspension / Resumption Criteria

If a test fails, all testing must be halted until a solution is found, preventing further testing from continuing. The testing process will resume once the underlying cause of the test failure has been identified and addressed, which may involve updating either the test or the system software.

## 2.2 Pass/Fail Criteria

To pass the assessment, the function must accurately process all input and output types, execute all necessary operations with precision, adhere to stakeholder requirements, and deliver the intended outcomes. The criteria for failure include instances where the function produces unexpected results, fails to handle errors appropriately, is unable to manage specific input or output, and results in crashes or delayed response times.

## 2.3 Entry / Exit Criteria

The entry criteria for functional testing running the beginning script to create all the tables. The exit criteria for functional testing is met when all the outlined test cases have been successfully completed.

## 2.5 Risks / Issues

Most functional tests require the execution of a SQL script to reset the database and insert the necessary data for the test scenario. However, this can lead to errors if the SQL script populates the database with unforeseen data, causing erroneous test failures.

## 2.6 User Sign In

### 2.6.1 Sign in – Admin Successful

|  |  |
| --- | --- |
| Test Case ID | TC-1 |
| Test Case Name | Sign in – Admin Successful |
| Description | Successfully signed in with the provided username and password and redirected to the admin dashboard. |
| Preconditions | 1. DBS-00.sql script has been run by copy and pasting it into MySQL workbench executing each line. 2. On the sign in page |
| Test Steps | 1. Enter “[admin@wayne.edu](mailto:admin@wayne.edu)” in the username input field. 2. Click the “Next” button. 3. Enter “pass” in the password input field. 4. Click the “Submit” button. |
| Expected Results | 1. A new session is created for the current user. 2. The user is redirected to the admin dashboard. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.6.2 Sign in – Professor Successful

|  |  |
| --- | --- |
| Test Case ID | TC-2 |
| Test Case Name | Sign in – Professor Successful |
| Description | Successfully signed in with the provided username and password and redirected to the professor dashboard. |
| Preconditions | 1. DBS-00.sql script has been run by copy and pasting it into MySQL workbench executing each line. 2. On the sign in page |
| Test Steps | 1. Enter “[kj1234@wayne.edu](mailto:kj1234@wayne.edu)” in the username input field. 2. Click the “Next” button. 3. Enter “pass123” in the password input field. 4. Click the “Submit” button. |
| Expected Results | 1. A new session is created for the current user. 2. The user is redirected to the professor dashboard. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.6.3 Sign in – GTA Successful

|  |  |
| --- | --- |
| Test Case ID | TC-3 |
| Test Case Name | Sign in – GTA Successful |
| Description | Successfully signed in with the provided username and password and redirected to the GTA dashboard. |
| Preconditions | 1. DBS-00.sql script has been run by copy and pasting it into MySQL workbench executing each line. 2. On the sign in page |
| Test Steps | 1. Enter “[js1234@wayne.edu](mailto:js1234@wayne.edu)” in the username input field. 2. Click the “Next” button. 3. Enter “test12” in the password input field. 4. Click the “Submit” button. |
| Expected Results | 1. A new session is created for the current user. 2. The user is redirected to the GTA dashboard. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

2.6.4 Sign in– No Email

|  |  |
| --- | --- |
| Test Case ID | TC-4 |
| Test Case Name | Sign in– No Email |
| Description | Attempt to sign in without entering an email. |
| Preconditions | 1. DBS-00.sql script has been run by copy and pasting it into MySQL workbench executing each line. 2. On the sign in page |
| Test Steps | 1. Do not type anything into the username input field. 2. Click the “Next” button. |
| Expected Results | 1. An error message displays under the username input box that says “\* Email is required.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail |

### 2.6.5 Sign in– Incorrect Email

|  |  |
| --- | --- |
| Test Case ID | TC-5 |
| Test Case Name | Sign in– Incorrect Email |
| Description | Attempt to sign in with an email that is not in the database. |
| Preconditions | 1. DBS-00.sql script has been run by copy and pasting it into MySQL workbench executing each line. 2. On the sign in page |
| Test Steps | 1. Enter “[ab1234@wayne.edu](mailto:ab1234@wayne.edu)” in the username input field. 2. Click the “Next” button. |
| Expected Results | 1. An error message displays under the username input box that says “\* Sorry that email is not in our database. Please seek out an admin for assistance.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.6.6 Sign in ­– No Password

|  |  |
| --- | --- |
| Test Case ID | TC-6 |
| Test Case Name | Sign in ­– No Password |
| Description | Attempt to sign in without entering a password. |
| Preconditions | 1. DBS-00.sql script has been run by copy and pasting it into MySQL workbench executing each line. 2. On the sign in page |
| Test Steps | 1. Enter “[admin@wayne.edu](mailto:admin@wayne.edu)” in the username input field. 2. Click the “Next” button. 3. Do not type anything in the password input field. 4. Click the “Submit” button. |
| Expected Results | 1. An error message displays under the password input box that says “\* Password is required.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.6.7 Sign in – Incorrect Password

|  |  |
| --- | --- |
| Test Case ID | TC-7 |
| Test Case Name | Sign in – Incorrect Password |
| Description | Attempt to sign in with an incorrect password. |
| Preconditions | 1. DBS-00.sql script has been run by copy and pasting it into MySQL workbench executing each line. 2. On the sign in page |
| Test Steps | 1. Enter “[admin@wayne.edu](mailto:admin@wayne.edu)” in the username input field. 2. Click the “Next” button. 3. Enter “Pass1” in the password input field. 4. Click the “Submit” button. |
| Expected Results | 1. An error message displays under the password input box that says “\* Sorry that password didn’t work try again.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.6.8 Sign in – Inactive User

|  |  |
| --- | --- |
| Test Case ID | TC-8 |
| Test Case Name | Sign in – Inactive User |
| Description | Attempt to sign in as a user that has a deactivated account |
| Preconditions | 1. DBS-01.sql script has been run by copy and pasting it into MySQL workbench executing each line. 2. On the sign in page |
| Test Steps | 1. Enter [jb1234@wayne.edu](mailto:jb1234@wayne.edu) in the username input field. 2. Click the “Next” button. |
| Expected Results | 1. An error message will display under the username input box that says, “\* Sorry that email is not in our database. Please seek out an admin for assistance.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.7 User Sign Out

### 2.7.1 Sign Out - Button

|  |  |
| --- | --- |
| Test Case ID | TC-9 |
| Test Case Name | Sign Out - Button |
| Description | Successfully signed out of the application |
| Preconditions | 1. Sign in as any user into the application. |
| Test Steps | 1. Click the hamburger icon in the top left corner of the header to open the navigation bar. 2. Click the “Sign Out” button. |
| Expected Results | 1. The session is destroyed. 2. The user is redirected to the sign in page |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.7.2 Sign Out – Close Browser

|  |  |
| --- | --- |
| Test Case ID | TC-10 |
| Test Case Name | Sign Out – Close Browser |
| Description | Successfully signed out of the application by closing the browser |
| Preconditions | 1. Sign in as any user into the application. |
| Test Steps | 1. Close the entire browser by clicking on the “X” in the top right corner of the window. 2. Open a new browser. 3. Type in your localhost directory that you are using to test followed by app/adminDash.php to go to that page. |
| Expected Results | 1. The session is destroyed. 2. The user will be relocated back to the sign in page |
| Priority | Low |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.8 User Set Password

### 2.8.1 User Set Password – Successful

|  |  |
| --- | --- |
| Test Case ID | TC-11 |
| Test Case Name | User Set Password – Successful |
| Description | When the user is first time signing into the application, they will be prompted to set their password. |
| Preconditions | 1. For test purposes, you will want to add your email, or an email account that you can sign into to check the mail inbox. An example is given in DBS-02.sql to run by copy and pasting it into MySQL workbench executing each line. 2. Locate to the sign in page. |
| Test Steps | 1. Enter “your email” into the username input field. 2. Click the “Next” button. 3. The screen will reload to the “Request Password” page. Enter “your email” again in the email input field. 4. Click the “Receive new password by email” button. 5. Sign into your email and click on the message with the subject “Set password for CCES” from [capstonecoursewsu@gmail.com](mailto:capstonecoursewsu@gmail.com) 6. Click on the link. 7. Enter “Password” into the password field. 8. Enter “Password” into the repeat password field. 9. Click the “Set Password!” button |
| Expected Results | 1. The password is now saved into the database with that email. 2. The user is redirected to the sign in page |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.8.2 User Set Password – No Email

|  |  |
| --- | --- |
| Test Case ID | TC-12 |
| Test Case Name | User Set Password – No Email |
| Description | When a user is first time signing into the application, they will be prompted to set their password, but will not enter an email. |
| Preconditions | 1. For test purposes, you will want to add your email, or an email account that you can sign into to check the mail inbox. An example is given in DBS-02.sql to run by copy and pasting it into MySQL workbench executing each line. 2. Locate to the sign in page. |
| Test Steps | 1. Enter “your email” into the username input field. 2. Click the “Next” button. 3. The screen will reload to the “Request Password” page. 4. Do not enter anything in the email input field. 5. Click the “Receive new password by email” button. |
| Expected Results | 1. An error message will appear under the “Receive new password by email” button that says “\* Please enter a valid email.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.8.3 User Set Password – Invalid Email

|  |  |
| --- | --- |
| Test Case ID | TC-13 |
| Test Case Name | User Set Password – Invalid Email |
| Description | When the user is first time signing into the application, they will be prompted to set their password. |
| Preconditions | 1. For test purposes, you will want to add your email, or an email account that you can sign into to check the mail inbox. An example is given in DBS-02.sql to run by copy and pasting it into MySQL workbench executing each line. 2. Locate to the sign in page. |
| Test Steps | 1. Enter “your email” into the username input field. 2. Click the “Next” button. 3. The screen will reload to the “Request Password” page. 4. Enter “josmith@gmail.com” into the email input field. 5. Click the “Receive new password by email” button. |
| Expected Results | 1. An error message will appear under the “Receive new password by email” button that says “\* Please enter a valid WSU email.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.8.4 User Set Password – No Password & No Password Confirmation

|  |  |
| --- | --- |
| Test Case ID | TC-14 |
| Test Case Name | User Set Password – No Password & No Password Confirmation |
| Description | Attempt to reset password with no entry in the password and password confirmation section. |
| Preconditions | 1. Run DBS-02.sql script by copy and pasting it into MySQL workbench executing each line. 2. Run through TC-11 to get an email with a link to the “Set Password” page. |
| Test Steps | 1. Do not enter anything in the password input field. 2. Do not enter anything in the password confirmation field. 3. Click the “Set Password” button. |
| Expected Results | 1. An error message will appear under the password input field that says “\*Password is required.” Similarly, an error message will appear under the password confirmation input field that says “\*Password Confirmation is required.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.8.5 User Set Password – No Password & Password Confirmation

|  |  |
| --- | --- |
| Test Case ID | TC-15 |
| Test Case Name | User Set Password – No Password & Password Confirmation |
| Description | Attempt to reset password with no entry of password but entry of password confirmation section. |
| Preconditions | 1. Run DBS-02.sql script by copy and pasting it into MySQL workbench executing each line. 2. Run through TC-11 to get an email with a link to the “Set Password” page. |
| Test Steps | 1. Do not enter anything in the password input field. 2. Enter “Password” in the password confirmation field. 3. Click the “Set Password” button. |
| Expected Results | 1. An error message will appear under the password input field that says “\*Password is required.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.8.6 User Set Password – Password & No Password Confirmation

|  |  |
| --- | --- |
| Test Case ID | TC-16 |
| Test Case Name | User Set Password –Password & No Password Confirmation |
| Description | Attempt to reset password with no entry of pass but not password confirmation section. |
| Preconditions | 1. Run DBS-02.sql script by copy and pasting it into MySQL workbench executing each line. 2. Run through TC-11 to get an email with a link to the “Set Password” page. |
| Test Steps | 1. Enter “Password” in the password input field. 2. Do not enter anything in the password confirmation field. 3. Click the “Set Password” button. |
| Expected Results | 1. An error message will appear under the password input field that says “\*Password Confirmation is required.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.8.7 User Set Password – Password with different Password Confirmation Entry

|  |  |
| --- | --- |
| Test Case ID | TC-17 |
| Test Case Name | User Set Password – Password with different Password Confirmation Entry |
| Description | Attempt to reset password with entry of password but different entry in the password confirmation. |
| Preconditions | 1. Run DBS-02.sql script by copy and pasting it into MySQL workbench executing each line. 2. Run through TC-11 to get an email with a link to the “Set Password” page. |
| Test Steps | 1. Enter “Password” in the password input field. 2. Enter “Password1” in the password confirmation input field. 3. Click the “Set Password” button. |
| Expected Results | 1. An error message will appear under the password input field that says “\*Passwords do not match.” Similarly, an error message will appear under the password confirmation input field that says “\*Passwords do not match.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.9 User Forgot Password

### 2.9.1 User Forgot Password – Successful

|  |  |
| --- | --- |
| Test Case ID | TC-18 |
| Test Case Name | User Forgot Password – Successful |
| Description | When the user enters their email but cannot remember their password, they can click the “Forgot Password |
| Preconditions | 1. For test purposes, you will want to add your email, or an email account that you can sign into to check the mail inbox. 2. Locate to the sign in page. |
| Test Steps | 1. Enter “your email” into the username input field. 2. Click the “Next” button. 3. On the sign in form underneath the “Submit” button, click the “Forgot password?” link. 4. The page reloads to the “Request Password” page. 5. Enter “your email” again in the email input field. 6. Click the “Receive new password by email” button. 7. Sign into your email and click on the message with the subject “Set password for CCES” from [capstonecoursewsu@gmail.com](mailto:capstonecoursewsu@gmail.com) 8. Click on the link. 9. Enter “Password” into the password field. 10. Enter “Password” into the repeat password field. 11. Click the “Set Password!” button |
| Expected Results | 1. The password is now saved into the database with that email. 2. The user is redirected to the sign in page. 3. If you sign in with your email and “Password” you will be let into the system. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.9.2 User Forgot Password – No Email

|  |  |
| --- | --- |
| Test Case ID | TC-19 |
| Test Case Name | User Forgot Password – No Email |
| Description | Attempt to receive email with reset link without entering an email |
| Preconditions | 1. For test purposes, you will want to add your email, or an email account that you can sign into to check the mail inbox. 2. Locate to the sign in page. |
| Test Steps | 1. Enter “your email” into the username input field. 2. Click the “Next” button. 3. The screen will reload to the “Request Password” page. 4. Do not enter anything in the email input field. 5. Click the “Receive new password by email” button. |
| Expected Results | 1. An error message will appear under the “Receive new password by email” button that says “\* Please enter a valid email.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.9.3 User Forgot Password – Invalid Email

|  |  |
| --- | --- |
| Test Case ID | TC-20 |
| Test Case Name | User Forgot Password – Invalid Email |
| Description | Attempt to receive an email with reset link by entering an email that is not in the database. |
| Preconditions | 1. For test purposes, you will want to add your email, or an email account that you can sign into to check the mail inbox. 2. Locate to the sign in page. |
| Test Steps | 1. Enter “your email” into the username input field. 2. Click the “Next” button. 3. The screen will reload to the “Request Password” page. 4. Enter “josmith@gmail.com” into the email input field. 5. Click the “Receive new password by email” button. |
| Expected Results | 1. An error message will appear under the “Receive new password by email” button that says “\* Please enter a valid email.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.9.4 User Forgot Password – No Password & No Password Confirmation

|  |  |
| --- | --- |
| Test Case ID | TC-21 |
| Test Case Name | User Forgot Password – No Password & No Password Confirmation |
| Description | Attempt to reset password with no entry in the password and password confirmation section. |
| Preconditions | 1. Run DBS-02.sql script by copy and pasting it into MySQL workbench executing each line. 2. Run through TC-11 to get an email with a link to the “Set Password” page. |
| Test Steps | 1. Do not enter anything in the password input field. 2. Do not enter anything in the password confirmation field. 3. Click the “Set Password” button. |
| Expected Results | 1. An error message will appear under the password input field that says “\*Password is required.” Similarly, an error message will appear under the password confirmation input field that says “\*Password Confirmation is required.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.9.5 User Forgot Password – No Password & Password Confirmation

|  |  |
| --- | --- |
| Test Case ID | TC-22 |
| Test Case Name | User Forgot Password – No Password & Password Confirmation |
| Description | Attempt to reset password with no entry of password but entry of password confirmation section. |
| Preconditions | 1. Run DBS-02.sql script by copy and pasting it into MySQL workbench executing each line. 2. Run through TC-11 to get an email with a link to the “Set Password” page. |
| Test Steps | 1. Do not enter anything in the password input field. 2. Enter “Password” in the password confirmation field. 3. Click the “Set Password” button. |
| Expected Results | 1. An error message will appear under the password input field that says “\*Password is required.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.9.6 User Forgot Password – Password & No Password Confirmation

|  |  |
| --- | --- |
| Test Case ID | TC-23 |
| Test Case Name | User Forgot Password – Password & No Password Confirmation |
| Description | Attempt to reset password with no entry of pass but not password confirmation section. |
| Preconditions | 1. Run DBS-02.sql script by copy and pasting it into MySQL workbench executing each line. 2. Run through TC-11 to get an email with a link to the “Set Password” page. |
| Test Steps | 1. Enter “Password” in the password input field. 2. Do not enter anything in the password confirmation field. 3. Click the “Set Password” button. |
| Expected Results | 1. An error message will appear under the password input field that says “\*Password Confirmation is required.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.9.7 User Forgot Password – Password with different Password Confirmation Entry

|  |  |
| --- | --- |
| Test Case ID | TC-24 |
| Test Case Name | User Forgot Password – Password with different Password Confirmation Entry |
| Description | Attempt to reset password with entry of password but different entry in the password confirmation. |
| Preconditions | 1. Run DBS-02.sql script by copy and pasting it into MySQL workbench executing each line. 2. Run through TC-11 to get an email with a link to the “Set Password” page. |
| Test Steps | 1. Enter “Password” in the password input field. 2. Enter “Password1” in the password confirmation input field. 3. Click the “Set Password” button. |
| Expected Results | 1. An error message will appear under the password input field that says “\*Passwords do not match.” Similarly, an error message will appear under the password confirmation input field that says “\*Passwords do not match.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.10 Create a Semester

### 2.10.1 Create new semester – Successful

|  |  |
| --- | --- |
| Test Case ID | TC-25 |
| Test Case Name | Add a new semester– Successful |
| Description | This feature allows admins to create new semesters. With this feature, the admin can create semesters by entering in the specific dates for the start and end of the semester. |
| Preconditions | * + - 1. Admin is logged in.  1. Admin must have access to the “Add semester/section” page. |
| Test Steps | 1. Navigate to the "Add semester/section" page in the hamburger tab. 2. Enters “Winter 23” in the “Semester Name” by typing. 3. Enter “01/10/2023” in the “Start Date” input field by either typing or using the date picker. 4. Enter “05/02/2023” in the “End Date” input field by either typing or using the date picker. 5. Clicks the “Add” button at the bottom of the form. |
| Expected Results | 1. A new semester with the dates entered, is added to the database with the information provided. 2. The new added semester's information is outputted in the admin dashboard. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.10.2 Create new semester – Missing Semester Name

|  |  |
| --- | --- |
| Test Case ID | TC-26 |
| Test Case Name | Create a new semester – Missing Semester Name |
| Description | The admin wants to create a new semester, but they create the semester with the start date and end date. But the admin forgets to enter the semester name. |
| Preconditions | * + - 1. Admin is logged in.  1. Admin must have access to the “Add semester/section” page. |
| Test Steps | 1. Navigate to the "Add semester/section" page in the hamburger tab. 2. Misses “Semester Name”. 3. Enter “01/10/2023” in the “Start Date” input field by either typing or using the date picker. 4. Enter “05/02/2023” in the “End Date” input field by either typing or using the date picker. 5. Clicks the “Add” button at the bottom of the form. |
| Expected Results | 1. The application displays an error handling message which says, “Semester Name is required.” 2. Since there is an error, the expected result should be that the semester should not update in the admin dashboard. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.10.3 Create new semester – Missing Semester Start Date

|  |  |
| --- | --- |
| Test Case ID | TC-27 |
| Test Case Name | Add a new semester– – Missing Semester Start Date |
| Description | The admin wants to create a new semester, but they create the semester with the semester name and end date. But the admin forgets to enter the start date. |
| Preconditions | * + - 1. Admin is logged in.  1. Admin must have access to the “Add semester/section” page. |
| Test Steps | 1. Navigate to the "Add semester/section" page in the hamburger tab. 2. Enters “Winter 23” in the “Semester Name” by typing. 3. Misses “Start Date”. 4. Enter “05/02/2023” in the “End Date” input field by either typing or using the date picker. 5. Clicks the “Add” button at the bottom of the form. |
| Expected Results | * + - 1. The application displays an error handling message which says, “Start date is required.”  1. Since there is an error, the expected result should be that the semester should not update in the admin dashboard. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.10.4 Create new semester – Missing Semester End Date

|  |  |
| --- | --- |
| Test Case ID | TC-28 |
| Test Case Name | Add a new semester– Missing Semester End Date |
| Description | The admin wants to create a new semester, but they create the semester with the semester name and start date. But the admin forgets to enter the end date. |
| Preconditions | * + - 1. Admin is logged in.       2. Admin must have access to the “Add semester/section” page. |
| Test Steps | * + - 1. Navigate to the “Add semester/section” page in the hamburger tab.  1. Enters “Winter 23” in the “Semester Name” by typing. 2. Enter “01/10/2023” in the “Start Date” input field by either typing or using the date picker. 3. Misses “End Date”. 4. Clicks the “Add” button at the bottom of the form. |
| Expected Results | * + - 1. The application displays an error handling message which says, “End date is required.”       2. Since there is an error, the expected result should be that the semester should not update in the admin dashboard. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.10.5 Create new semester – Existing Semester

|  |  |
| --- | --- |
| Test Case ID | TC-29 |
| Test Case Name | Create a new semester– Existing Semester |
| Description | The admin is filling out the information for “Add Semester” page. But the user(admin) but creates the two of the same semesters, with the same dates and name. |
| Preconditions | * + - 1. Admin is logged in.       2. Admin must have access to the “Add semester/section” page.       3. Run the script DB-14.sql by copy and pasting it into MySQL workbench executing each line. |
| Test Steps | 1. Navigate to the “Add semester/section” page in the hamburger tab. 2. Enters “Winter 23” which exists in the database. 3. Enter “01/10/2023” in the “Start Date” input field by either typing or using the date picker. 4. Enter “05/02/2023” in the “End Date” input field by either typing or using the date picker. 5. The admin or professor clicks the “Add” button. |
| Expected Results | 1. The web application displays the error message “The semester name has already been added, please enter a semester that hasn’t been added yet.” This error message shows that the admin has already inputted a semester name exactly as the one that they try to enter. 2. If the admin enters the same date that has been entered previously, the application will display its error handling message which states “The dates entered have already been entered previously, please enter new dates.” The message displayed is to show that the dates are already in the database and can’t be entered again. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.11 Create a Section

### 2.11.1 Admins creates a new section - Successful

|  |  |
| --- | --- |
| Test Case ID | TC-30 |
| Test Case Name | Admins creates a new section - Successful |
| Description | This feature allows admins of the specific program to add sections based on the semester they create. The admin also has the power to assign the section to a specific semester. They can also assign a specific professor to that section. |
| Preconditions | * + - 1. Admin is logged in.  1. Admin has to have access to the “Add semester/section” page. |
| Test Steps | 1. Navigates to the “Add semester/section” page in the hamburger tab. 2. Enter “Section 001” in the “Section Name” by typing it in the input field. 3. Assign the section to a “Semester” by using the drop-down menu, assign it to “Winter 23” Semester. 4. Assign a “Professor” using the drop-down menu. Assign the professor to “Section 001”. 5. Clicks the “Add” button at the bottom of the form. |
| Expected Results | 1. Once the add button is clicked, the admin will see on the admin dashboard which section is in what semester and which professor is assigned to that section. 2. The database is updated for the correct input results entered by the user. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.11.2 Admin creates a new section – Missing section name

|  |  |
| --- | --- |
| Test Case ID | TC-31 |
| Test Case Name | Admin creates a new section – Missing section name |
| Description | The admin wants to create a new section, but they create the section with the semester which the section is assigned to and professor. But the admin forgets to enter the section name. |
| Preconditions | 1. The admin is logged in. 2. The admin has access to the “Add section/semester” page. |
| Test Steps | 1. Navigates to the “Add semester/section” page in the hamburger tab. 2. Misses “Section Name”. 3. Assign the section to a “Semester” by using the drop-down menu, assign it to “Winter 23” Semester. 4. Assign a “Professor” using the drop-down menu. Assign the professor to “Section 001”. 5. Clicks the “Add” button at the bottom of the form. |
| Expected Results | The web application displays an error “Section Name is required”.   1. The database and the “Section” on the admin dashboard will not be updated because of the missing criteria needed by the user. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.11.3 Admins creates a new section – Missing semester criteria

|  |  |
| --- | --- |
| Test Case ID | TC-32 |
| Test Case Name | Admin creates a new section – Missing semester criteria |
| Description | The admin wants to create a new section, but they create the section with the section name and assign a professor. But the admin misses the semester criteria. |
| Preconditions | The admin is logged in.  The admin has access to the “Add section/semester” page. |
| Test Steps | Navigates to the “Add semester/section” page in the hamburger tab.  Enter “Section 001” in the “Section Name” by typing it in the input field.  Misses “Semester”.  Assign a “Professor” using the drop-down menu. Assign the professor to “Section 001”.  Clicks the “Add” button at the bottom of the form. |
| Expected Results | 1. The web application displays an error “Semester is required”. 2. The database and the “semester” on the admin dashboard will not be updated because of the missing criteria needed by the user. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.11.4 Admin creates a new section – Missing professor

|  |  |
| --- | --- |
| Test Case ID | TC-33 |
| Test Case Name | Admin creates a new section – Missing professor |
| Description | The admin wants to create a new section. They fill out the section name and decide which semester this section will be placed in. But the admin forgets to pick the professor for the group. |
| Preconditions | 1. The admin is logged in. 2. The admin has access to the “Add section/semester” page. |
| Test Steps | 1. Navigates to the “Add semester/section” page in the hamburger tab. 2. Enter “Section 001” in the “Section Name” by typing it in the input field. 3. Assign the section to a “Semester” by using the drop-down menu, assign it to “Winter 23” Semester. 4. Misses “Assign a professor”. 5. Clicks the “Add” button at the bottom of the form. |
| Expected Results | 1. The web application displays an error “Professor is required”. 2. The database and the “Section” on the admin dashboard will not be updated because of the missing criteria needed by the user. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.11.5 Admin creates a new section – Existing section

|  |  |
| --- | --- |
| Test Case ID | TC-34 |
| Test Case Name | Admin creates a new section – Existing section |
| Description | The admin is filling out the information for the “Add semester/section” page. But the user(admin) but creates the two of the same sections with the same name and tries to assign multiple professors to the same section. |
| Preconditions | 1. Login capabilities must be required for the admin to login and access the specific page. 2. The admin must have access to the “Add section/semester” page. 3. DBS-14.sql has been run by copy and pasting it into MySQL workbench executing each line.. |
| Test Steps | 1. Navigates to the “Add semester/section” page in the hamburger tab. 2. Enter “Section 001” in the “Section Name” by typing it in the input field. 3. The entered “Section” already exists in the database. 4. Assign the section to a “Semester” by using the drop-down menu, assign it to “Winter 23” Semester. 5. Assign a “Professor” using the drop-down menu. Assign the professor to “Section 001”. 6. Clicks the “Add” button at the bottom of the form. |
| Expected Results | 1. The web application displays the error message “The section name has already been added, please enter a section that hasn’t been added yet.” This error message shows that the admin has already inputted a section name exactly as the one that they try to enter. 2. If the user adds multiple professors to the section, the web application displays the message “A professor has already been inputted to that section. Please add another professor that hasn’t been added to that section.” This error handling message tells the admin that you must add different professors to different sections to be a valid section. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.12 Add Professor

### 2.12.1 Add New Professor - Successful

|  |  |
| --- | --- |
| Test Case ID | TC-35 |
| Test Case Name | Add new Professor – Successful |
| Description | The admin fills out the “Add Professor!” form successfully. |
| Preconditions | 1. The admin is logged in. |
| Test Steps | 1. Navigate to the “Add Professor” page in the hamburger tab. 2. Enter all data in the “Add Professor!” form. 3. Click the “Add” button at the button of the form. |
| Expected Results | 1. A success button appears for three seconds. 2. The new professor is added to the database with the provided information. 3. The new professor’s information is shown in the “Activated Professors” list. 4. The new professor can now log in and set up a password. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.12.2 Add New Professor – Missing Form Requirements

|  |  |
| --- | --- |
| Test Case ID | TC-36 |
| Test Case Name | Add new Professor – Missing Form Requirements |
| Description | The admin fills out the “Add Professor!” form but some required information is missing. |
| Preconditions | 1. The admin is logged in. 2. On the “Add Professor” page. |
| Test Steps | 1. Enter some, but not all, information required in the “Add Professor!” form. 2. Click the “Add” button. 3. Verify that an error message appears with a prompt at the bottom of the field in the form that is missing the requirements. |
| Expected Results | 1.The web application displays one or more error message within the form.  A) “\* Last name is required”  B) “\* First name is required”  C) “\*Email is required”  2.The “Edit Professors” table is not updated. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.12.3 Add New Professor – With Existing Email

|  |  |
| --- | --- |
| Test Case ID | TC-37 |
| Test Case Name | Add New Professor – With Existing Email |
| Description | The admin fills out the “Add Professor!” form but the email is matching to an email in the database. |
| Preconditions | 1. The admin has logged in. 2. On the “Add Professor” page. |
| Test Steps | 1. Look at the deactivated and activated professor tables. 2. Find a user that already exists. 3. Enter their email into the “Add Professor!” form. 4. Click the “Add” button. 5. Verify that an error message appears with a prompt at the bottom of the form. 6. Verify a new professor was not added into the “Activated Professor Accounts” table below. |
| Expected Results | 1. The web application displays the error message “this email already belongs to a user” within the form. 2. The “Edit Professors” table is not updated |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.13 Edit Professor

### 2.13.1 Edit Professor – Successful

|  |  |
| --- | --- |
| Test Case ID | TC-38 |
| Test Case Name | Edit Professor - Successful |
| Description | The admin clicks the “edit” button for a specific professor and a modal appears on the screen. |
| Preconditions | The admin is logged in.  On the “Add Professor” page.  The professor already exists in the database.  The professor’s data is displayed in the input form.  The edit form can store and retrieve data. |
| Test Steps | 1. Navigate to the “Activated Professor Accounts” table and click the “Edit” button. 2. Verify that the “Edit Professor” pop up modal works. 3. Edit any of the fields in the edit pop up modal. 4. Click “Update Data”. 5. Verify that the table displays the new data by redoing step 2 to view the professor’s information. |
| Expected Results | 1. The admin can open the edit professor form by clicking the “edit” button.  2. The database is updated after the form has been filled out.  3. The webpage displays the updated data.  4. The “Edit” button and “Edit Professor” modal is responsive. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.13.2 Edit Professor – Input Blank Name

|  |  |
| --- | --- |
| Test Case ID | TC-39 |
| Test Case Name | Edit Professor – Input Blank Name |
| Description | The admin clicks the “edit” button for a specific professor and a modal appears on the screen. |
| Preconditions | 1. 1.The admin is logged in. 2. Navigate the “add professor” page. 3. The professor already exists in the database. 4. The professor’s data is displayed in the input form. 5. The edit form can store and retrieve data. |
| Test Steps | 1. Navigate to the “Activated Professor Accounts” table and click the “Edit” button.  2. Verify that the “Edit Professor” pop up modal works.  3. Edit the first name, last name, or email field with a blank entry.  4. Verify that the modal stays open.  5. Verify that the modal displays an error message at the bottom of the edit modal:  A) “First name is required.”  B) “Last name is required.”  C) “Email is required.” |
| Expected Results | * + - 1. The pop-up modal does not close.       2. The database is not updated after the form has been filled out.       3. The pop-up modal displays the correct error messages. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.13.3 Edit Professor – Input Any Email

|  |  |
| --- | --- |
| Test Case ID | TC-40 |
| Test Case Name | Edit Professor – Input Any Email |
| Description | The admin clicks the “edit” button for a specific professor and a modal appears on the screen. |
| Preconditions | 1.The admin is logged in.  2. Navigate the “add professor” page.  3. The professor already exists in the database.  4. The professor’s data is displayed in the input form.  5. The edit form can store and retrieve data. |
| Test Steps | 1. Navigate to the “Activated Professor Accounts” table and click the “Edit” button.  2. Verify that the “Edit Professor” pop up modal is responsive.  3. Edit the email field with any email other than a Wayne State email.  4. Verify that the modal stays open.  5. Verify that the modal displays an error message: “Email must be a valid @wayne.edu email address.” at the bottom of the edit modal |
| Expected Results | 1. The pop-up modal does not close.  2. The database is not updated after the form has been filled out with the wrong email.  3. The pop-up modal displays the correct error messages. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.14 Deactivate Professor

### 2.14.1 Deactivate Professor – Successful

|  |  |
| --- | --- |
| Test Case ID | TC-41 |
| Test Case Name | Deactivate Professor – Successful |
| Description | The admin clicks the “deactivate” button in the “Activated Professor Accounts” list. |
| Preconditions | 1. The professor has logged in. 2. On the “Add Professor” page. 3. The professor already exists in the database. |
| Test Steps | 1. Click the “Deactivate” button in the “Activated Professor Accounts” form. 2. A modal appears on the screen with a “Deactivate” button and a “Close” button. 3. Click “Deactivate”. 4. Confirm the professor was deactivated by looking at the “Deactivated Professor Accounts” table. |
| Expected Results | 1. The professor’s account is successfully deactivated. 2. The professor’s data remains in the database. 3. The professor can no longer log into the web application. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.14.2 Deactivate Professor – Close Modal

|  |  |
| --- | --- |
| Test Case ID | TC-42 |
| Test Case Name | Deactivate Professor – Close Modal |
| Description | The admin clicks the “deactivate” button in the “Activated Professor Accounts” list but does not deactivate a professor’s account. |
| Preconditions | 1. Logged in as admin. 2. On the “Add Professor” page. 3. The professor already exists in the database. |
| Test Steps | 1. Click the “Deactivate” button in the “Activated Professor Accounts” form. 2. A modal appears on the screen with a “Deactivate” button and a “Close” button. 3. Click “Close”. 4. Confirm the professor was not deactivated by looking in the “Activated Professor Accounts” table. |
| Expected Results | * + - 1. The professors account stays activated.       2. The professor’s data remains in the database.       3. The professor can log into the web application. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.15 Add Section

### 2.15.1 Add Section (Professor)- Successful

|  |  |
| --- | --- |
| Test Case ID | TC-43 |
| Test Case Name | Add section (Professor)– Successful entry |
| Description | Allows professors to add sections and assign themselves to that section. |
| Preconditions | 1. The professor is signed in and is on the “Add Section” page. |
| Test Steps | 1. Enter the section name along with the three section numbers. 2. Click add. |
| Expected Results | 1. The added section will show up on the dashboard of the professors. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.15.2 Add Section (Professor)- Missing section name

|  |  |
| --- | --- |
| Test Case ID | TC-44 |
| Test Case Name | Add section (Professor)– section name |
| Description | Trying to create a section without a section name. |
| Preconditions | * + - 1. The professor is signed in and is on the “Add Section” page. |
| Test Steps | 1. Enter the section name along with the three section numbers. 2. Click add. |
| Expected Results | 1. An error message will be displayed saying that the “Entered information is incorrect.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.16 Add GTA

### 2.16.1 Add GTA – Successful

|  |  |
| --- | --- |
| Test Case ID | TC-45 |
| Test Case Name | Add GTA – Successful |
| Description | The professor clicks the “edit” button for a specific professor and a modal appears on the screen. |
| Preconditions | 1. The professor is logged in and has access to GTA information. 2. On the “Add Professor” page. |
| Test Steps | 1. Enters all data in the “Add GTA!” form. 2. Click the “Add” button at the bottom of the form. |
| Expected Results | 1. The new GTA is added to the database with the provided information. 2. The GTA information is show in the “Activated GTA Accounts” table. 3. The GTA can now login and set up a password. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.16.2 Add GTA – Missing Form Requirements

|  |  |
| --- | --- |
| Test Case ID | TC-46 |
| Test Case Name | Add GTA – Missing Form Requirements |
| Description | The professor fills out the “Add GTA!” form but some required information is missing. |
| Preconditions | The professor is logged in.  On the “Add GTA” page. |
| Test Steps | 1. Enter some, but not all, information required in the “Add GTA!” form. 2. Click the “Add” button. 3. Verify that an error message appears with a prompt below each field of the form that has missing information. |
| Expected Results | 1. The web application displays one or more error message within the form.    1. “\* Last name is required”    2. “\* First name is required”    3. “\*Email is required” 2. The database and the “Activated GTA Accounts” table is not updated. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.16.3 Add GTA - With Existing Email

|  |  |
| --- | --- |
| Test Case ID | TC-47 |
| Test Case Name | Add GTA - With Existing Email |
| Description | The professor fills out the “Add GTA!” form but the email is matching to an email in the database. |
| Preconditions | 1. Log in as a professor and navigate to the “Add GTA” page. |
| Test Steps | 1. Enter an email in the “Add GTA!” form that has already been added in the “Activated GTA Accounts” table below the form. 2. Click the “Add” button. 3. Verify that an error message appears below the email field of the form. 4. Verify a new professor was not added into the “Activated GTA Accounts” table below. |
| Expected Results | 1. The web application displays the error message “this email already belongs to a user” within the form. 2. The database and the Activated GTA Accounts table is not updated |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.17 Edit GTA

### 2.17.1 Edit GTA – Successful

|  |  |
| --- | --- |
| Test Case ID | TC-48 |
| Test Case Name | Edit GTA - Successful |
| Description | The professor clicks the “edit” button for a specific professor and a modal appears on the screen. |
| Preconditions | 1. The professor is logged in. 2. On the “Add GTA” page. 3. The GTA already exists in the database. 4. The GTA’s data is editable in the input form. 5. The edit form can store and retrieve data. |
| Test Steps | 1. Press the “Edit” button. 2. Verify that the “Edit GTA” pop up modal works. 3. Edit any of the fields in the edit pop up modal. 4. Verify that the webpage displays the new data that was filled out by repeating step 1 and viewing the GTAs information. |
| Expected Results | The professor can see the Edit GTA form by clicking the “edit” button for the added GTA.  The database is updated after the form has been filled out.  The webpage displays the updated data.  The “Edit” button and “Edit GTA” modal is responsive. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.17.2 Edit GTA – Input Incorrect Data

|  |  |
| --- | --- |
| Test Case ID | TC-49 |
| Test Case Name | Edit Professor – Input Blank Name |
| Description | The professor clicks the “edit” button for a specific GTA in the activated or deactivated GTA tables, a modal appears on the screen. |
| Preconditions | The professor is logged in.  On the “Add GTA page”.  The GTA already exists in the database.  The professor’s data is editable in the input form.  The edit form can store and retrieve data. |
| Test Steps | Navigate to the “Activated GTA” table and click the “Edit” button.  Verify that the “Edit GTA” pop up modal works.  Edit the first name, last name, and email field with a blank entry.  Verify that the modal stays open.  Verify that the modal displays an error message: A) “First name is required.” B) “Last name is required.” C) “Email is required.” D) “Email must be a valid @wayne.edu email address.” at the bottom of each field in the modal.  Edit the last name and email field with a blank entry.  Verify that the modal stays open.  Verify that the modal displays an error message. A) “Last name is required.” B) “Email is required.” C) “Email must be a valid @wayne.edu email address.” at the bottom of each field in the modal.  Edit the email field with a blank entry.  Verify that the modal stays open.  Verify that the modal displays an error message: A) “Email is required.” B) “Last name is required.” at the bottom of each field in the modal.  Edit the email field any email other than the Wayne State email.  Verify the modal stays open.  Verify that the modal Verify that the modal displays an error message. A) Email must be a valid @wayne.edu email address. at the bottom of each field in the modal. |
| Expected Results | * + - 1. The pop-up modal does not close.       2. The database is not updated after the form has been filled out.       3. The pop-up modal displays the correct error messages. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.18 Deactivate GTA

### 2.18.1 Deactivate GTA – Successful

|  |  |
| --- | --- |
| Test Case ID | TC-50 |
| Test Case Name | Deactivate GTA – Successful |
| Description | The professor clicks the “Deactivate” button in the “Activated GTA Accounts” list. |
| Preconditions | 1. The professor is logged in. 2. On the “Add GTA” page. 3. The GTA already exists in the database. |
| Test Steps | 1. Click a “Deactivate” button for a specific GTA. 2. A modal appears on the screen with a “Deactivate” button and a “Close” button. 3. Click “Deactivate”. 4. Confirm the GTA was deactivated by looking at the “Deactivated GTA Accounts” table, located under the “Activated GTA Accounts” table. |
| Expected Results | * + - 1. The GTA’s account is successfully deactivated.       2. The GTA data remains in the database.       3. The GTA can no longer log into the web application. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.18.2 Deactivate GTA – Close Modal

|  |  |
| --- | --- |
| Test Case ID | TC-51 |
| Test Case Name | Deactivate GTA – Close Modal |
| Description | The professor clicks the “Deactivate” button in the “Activated GTA Accounts” list but does not deactivate a GTA’s account. |
| Preconditions | 1. The professor is logged in. 2. On the “Add GTA” page. 3. The GTA already exists in the database. |
| Test Steps | 1. Click the “Deactivate” button. 2. A modal appears on the screen with a “Deactivate” button and a “Close” button. 3. Click “Close”. 4. Confirm the GTA was not deactivated by looking in the “Activated Professor Accounts” table. |
| Expected Results | 1. The GTA’s account will remain activated. 2. The GTA data remains in the database. 3. The GTA can log into the web application. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.19 Add Students

### 2.19.1 Add Students- Successful

|  |  |
| --- | --- |
| Test Case ID | TC-52 |
| Test Case Name | Add Students- successful |
| Description | Professor adds students to a section by uploading a spreadsheet file that contains the access id and the student names. |
| Preconditions | 1. Professor is signed into the application. 2. On the “Add Students” page. |
| Test Steps | 1. Click the choose file button and select the spreadsheet file of type .csv, .ods, .xls, or .xlsx with the students access id and name. 2. Select a section. 3. Click the “Upload” button. |
| Expected Results | 1. The new students are added to the database with the provided information. 2. The new students’ names will be displayed in the “Unassigned Active Students” form below. 3. The new students’ names will be displayed in the “Add Group” page and the “Unassigned Active Student” form below. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.19.2 Add Students- Existing Student in Section

|  |  |
| --- | --- |
| Test Case ID | TC-53 |
| Test Case Name | Add Students- Existing Student in section |
| Description | The professor uploads a spreadsheet of students, but the file contains students who are already in the section. |
| Preconditions | * + - 1. Professor is signed into the application.       2. On the “Add Students” page.       3. DBS-10.sql has been run by copy and pasting it into MySQL workbench executing each line. |
| Test Steps | 1. Create a spreadsheet of type .csv, .ods, .xls, or .xlsx with the access id column containing “jk3333” and the name column containing “Smith John”. 2. Navigate to the “Add Students” page in the hamburger tab. 3. Click the choose file button and selects the spread sheet file with the student’s information. 4. Select the section. 5. Click the “Upload” button. 6. Verify the error message appears at the bottom of the form. 7. Verify the student was not added to the “Unassigned Active Students” form below. |
| Expected Results | * + - 1. The web application displays the error message “Cannot add existing students in section.”       2. The web application prints out the access ids of the students who are duplicates on the top of the web page.       3. The information is not inserted in the “Unassigned Active Student” form below. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.19.3 Add Students- Invalid Access Id

|  |  |
| --- | --- |
| Test Case ID | TC-54 |
| Test Case Name | Add Students- Invalid Access Id |
| Description | The professor is uploading a spreadsheet of a student to a section, but the Access Id is not in the proper format |
| Preconditions | Professor is signed into the application.  On the “Add Students” page. |
| Test Steps | 1. Create a spreadsheet of type .csv, .ods, .xls, or .xlsx with the access id column containing “jk33335” and the name column containing “Test Student”. 2. Click the choose file button and selects the spreadsheet file with the student’s information. 3. The professor continues to choose the section that they want the students to be added to. 4. The professor then clicks the “Upload” button. 5. Verify the error message is printed on top of the web page. |
| Expected Results | 1. The web application displays the error message “\* Invalid Access ID contained in file. Please reupload file” in the bottom of the form. 2. The information is not inserted in the “Unassigned Active Student” form below. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## Delete Student

### 2.20.1 Delete Student - Successful

|  |  |
| --- | --- |
| Test Case ID | TC-55 |
| Test Case Name | Delete Student - Successful |
| Description | The professor can remove students from a semester. |
| Preconditions | * + - 1. Professor is signed into the application.       2. On the “Add Students” page.       3. The student that will be deleted is not in a active group. |
| Test Steps | 1. The professor navigates to the “Add Student” page in the hamburger tab and scrolls to the bottom of the page. 2. The professor chooses the section tabs that they want to view the students for. 3. The professor checks the boxes for every student they want to remove in the window. 4. The professor then clicks the “Delete Student(s)” button at the bottom of the form |
| Expected Results | 1. The students selected are no longer active members of the semester. 2. The students selected will appear in the “inactive students” window. |
| Priority | Medium |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.21 Create Groups

### 2.21.1 Create Groups- Successful

|  |  |
| --- | --- |
| Test Case ID | TC-56 |
| Test Case Name | Create Groups - successful |
| Description | The professor creates a project group. |
| Preconditions | * + - * 1. Professor is signed into the application.         2. On the “Add Groups” page. |
| Test Steps | * + - 1. Enter the data for the name field in “Add a Group” form.       2. Click the “Save Group(s)” button at the bottom of the form |
| Expected Results | 1. The group is entered into the database with the provided information.  2. The group can be selected from the “Assign your students to a group” form. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.21.2 Create Groups- Existing Group

|  |  |
| --- | --- |
| Test Case ID | TC-57 |
| Test Case Name | Create Groups - Existing Group |
| Description | The professor is fills out the “Add a Group” form but the group name was already entered in the section. |
| Preconditions | 1. Professor is signed into the application. 2. Professor is on the “Add Groups” page. 3. Script DBS-11.sql has been run by copy and pasting it into MySQL workbench executing each line. |
| Test Steps | 1. Enter “Test Team” in the name field in the “Add a Group” form. 2. Click the “Save Group(s)” button. 3. Verify the error message appears. 4. Verify the group given was not added to the dropdown menu in the “Assign your students to a group” form below |
| Expected Results | * + - 1. The web application displays the error message “\* Group already inserted.” within the form.       2. The repeated group name will not be inserted into the database. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.22 Assign Students to a Group

### 2.22.1 Assign Students to a Group - Successful

|  |  |
| --- | --- |
| Test Case ID | TC-58 |
| Test Case Name | Assign Students to a Group - Successful |
| Description | The professor assigns the students to a group. |
| Preconditions | * + - 1. Log in as a professor.       2. Professor is on the “Add Groups” page.       3. Script DBS-12.sql has been run by copy and pasting it into MySQL workbench executing each line. |
| Test Steps | 1. Select group “Testing Team” using the drop-down menu. 2. Select “Lou Will” and “Jenny Lee” among the list of available students using checkboxes. 3. Click the “Assign Students” button. |
| Expected Results | The web application displays the message “Students Added”  Table in “Edit Groups” page is updated |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.23 Edit Groups

### 2.23.1 Edit Groups - Successful

|  |  |
| --- | --- |
| Test Case ID | TC-59 |
| Test Case Name | Edit Groups - Successful |
| Description | The professor modifies the group name, the GTA assigned to group and the members of the group all at once. |
| Preconditions | 1. The professor is logged in. 2. On the “Edit Group” page. 3. Script DBS-13.sql has been run by copy and pasting it into MySQL workbench executing each line. |
| Test Steps | 1. Click the “Edit” button on the “Testing Team” row. 2. Enter another group name like “Team 1” in the “Group Name” field. 3. Select another GTA like “Nick Smith” from the dropdown menu. 4. Select a student checkbox like “Sara Steve” to remove. 5. Click the “Save” button. 6. Verify that the table in the “Edit Groups” page is updated with the new information. |
| Expected Results | * + - 1. The edit group form successfully opens when clicking the edit button.       2. The database is updated after the form has been filled out.       3. The webpage displays the success message “Group updated successfully!”       4. The webpage displays the updated data. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.23.2 Edit Groups – Edit Group Name

|  |  |
| --- | --- |
| Test Case ID | TC-60 |
| Test Case Name | Edit Groups – Edit Group Name |
| Description | The professor modifies the group name. |
| Preconditions | 1. The professor is logged in. 2. On the “Edit Group” page. 3. Script DBS-13.sql is run by copy and pasting it into MySQL workbench executing each line. |
| Test Steps | 1. Click the “Edit” button for the “Testing Team” row. 2. Enter another group name like “Team 2” in the “Group Name” field. 3. Click the “Save” button. 4. Verify that the table in the “Edit Groups” page is updated with the new group name. |
| Expected Results | 1. The edit group form successfully opens when clicking the edit button. 2. The database is updated after the changes. 3. The webpage displays the success message “Group updated successfully!” 4. The webpage displays the updated data. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.23.3 Edit Groups – Edit Assigned GTA

|  |  |
| --- | --- |
| Test Case ID | TC-61 |
| Test Case Name | Edit Groups – Edit assigned GTA |
| Description | The professor changes the GTA assigned to a group. |
| Preconditions | 1. The professor is logged in. 2. On the “Edit Group” page. 3. Script DBS-13.sql is run by copy and pasting it into MySQL workbench executing each line. |
| Test Steps | 1. Click the “Edit” button for the “Testing Team” row. 2. Select another GTA from the dropdown menu like “Nick Smith”. 3. Click the “Save” button. 4. Verify that the table in the “Edit Groups” page is updated with the updated GTA. |
| Expected Results | 1. The edit group form successfully opens when clicking the edit button. 2. The database is updated after the changes. 3. The webpage displays the updated data. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.23.4 Edit Groups – Remove Group Members

|  |  |
| --- | --- |
| Test Case ID | TC-62 |
| Test Case Name | Edit Groups – Remove Group Members |
| Description | The professor unassigns a student from a group. |
| Preconditions | 1. The professor is logged in. 2. On the “Edit Group” page. 3. Script DBS-13.sql is run by copy and pasting it into MySQL workbench executing each line. |
| Test Steps | 1. Click the “Edit” button for the “Testing Team” row. 2. Select a student checkbox to remove such as “Mary Jane” 3. Click the “Save” button. 4. Verify that the table in the “Edit Groups” page is updated with the removed student. |
| Expected Results | 1. The edit group form successfully opens when clicking the edit button. 2. The database is updated after the changes. 3. The webpage displays the updated data. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.24 View Student

### 2.24.1 View Student – Successful

|  |  |
| --- | --- |
| Test Case ID | TC-63 |
| Test Case Name | View Student – Successful |
| Description | The professor is viewing the evaluation information for the selected student. |
| Preconditions | 1. Log in as a professor. 2. On the professor dashboard 3. Groups and students have been added for that specific professor account. |
| Test Steps | 1. Click on a group. 2. Click on a student in the selected group. 3. View the weekly report. 4. Make sure the Weekly report displays the date submitted, week, submitted, status and evaluation column. |
| Expected Results | 1. The web application displays all information related the weekly reports.  2. The information is accurate and reflects the data the GTA inputs. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.25 View Grades

### 2.25.1- View Grades (professor) – successful

|  |  |
| --- | --- |
| Test Case ID | TC-64 |
| Test Case Name | View grades- successful |
| Description | The feature allows the professor to view the grades of students which inputted by the GTA’s. |
| Preconditions | 1. The professor is signed in and is on their dashboard or “Home” page. |
| Test Steps | 1. Clicks on the “section” tab. 2. Select “group name” in the nested tab. 3. Clicks on the “student name” to view students. 4. Grades are outputted. |
| Expected Results | 1. The page reloads to the student view page. 2. The professor will be able to access all the grades for the students based on the inputs by themselves and GTA’s. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.25.1- View Grades (professor) – Unsuccessful

|  |  |
| --- | --- |
| Test Case ID | TC-64 |
| Test Case Name | View grades - Unsuccessful |
| Description | The feature does not allow the professor to view the grades of students which inputted by the GTA’s, |
| Preconditions | 1. The professor is signed in and is on their dashboard or “Home” page. |
| Test Steps | 1. Clicks on the “section” tab. 2. Select “group name” in the nested tab. 3. Clicks on the “student name” to view students. 4. Grades aren’t outputted. |
| Expected Results | The pages reloads to the student view page.  The professor will not be able to access the grades of the students based on the inputs from the GTA’s. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.26 Pick Group

### 2.26.1- Pick Group (GTA)- successful

|  |  |
| --- | --- |
| Test Case ID | TC-65 |
| Test Case Name | Pick Group (GTA)- successful |
| Description | The GTA chooses which groups to assign themselves. |
| Preconditions | 1. GTA is logged in to the application. 2. On the “Choose Group” page. 3. Script DBS-11.sql has been run by copy and pasting it into MySQL workbench executing each line. |
| Test Steps | 1. Select a group from the “Choose Your Groups form like “Test Team”. 2. Click "Assign" button on the bottom of the form. |
| Expected Results | 1. The web application displays the message “Groups added” 2. Window below called “Your Groups” is updated |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.27 View Group

### 2.27.1- View Group (GTA)- successful

|  |  |
| --- | --- |
| Test Case ID | TC-66 |
| Test Case Name | View group- successful |
| Description | The GTA can successfully view groups in their Dashboard. Once the GTA clicks on a certain group, the GTA will be taken to a different page where the GTA can view and grade each student. |
| Preconditions | User is signed into the application.  On the GTA dashboard. |
| Test Steps | 1. GTA decides on the group they want to help. 2. The GTA then clicks add group. |
| Expected Results | 1. The groups selected by the GTA will be displayed on the dashboard of the GTA. |
| Priority | High |
| Pass/Fail Criteria | Pass |

## 2.28 Grade weekly reports

### 2.28.1 Grade weekly reports- Successful student report entry

|  |  |
| --- | --- |
| Test Case ID | TC-67 |
| Test Case Name | Grade weekly reports- Successful student report entry |
| Description | This feature allows the GTA’s to grade students for their weekly report. The GTA’s will input the students weekly report information correctly based on how well the students did. |
| Preconditions | 1. User is signed into the application. 2. User is on GTA dashboard. 3. The user must be able to enter to the weekly report page. |
| Test Steps | * + - 1. GTA selects “Wayne State Evaluation System”.       2. Click “Select”.       3. The application will display the students in a table.       4. GTA clicks on the incomplete link to access the student’s weekly report status.       5. Clicks advanced for “Submission”.       6. Clicks advanced for “Status”.       7. Fills out “Evaluation” by typing “Student is doing amazing”.       8. Clicks “update data”. |
| Expected Results | 1. The student’s submission and status must be shown on the weekly report page. 2. The information must display on the student page on the professor side. 3. Information should also be stored in the tables below the weekly report, this table is the status and submission of weekly report for each student. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.28.2 Grade weekly reports- Missing submission criteria

|  |  |
| --- | --- |
| Test Case ID | TC-68 |
| Test Case Name | Grade weekly reports- missing submission criteria |
| Description | This feature allows the GTA’s to grade students for their weekly report. The GTA’s will input the students weekly report information correctly based on how well the students did. But the GTA forgets to fill out the submission portion of the grading criteria and fills out the other required criteria’s. |
| Preconditions | 1. User is signed into the application. 2. User is on GTA dashboard. 3. The user must be able to enter to the weekly report page. |
| Test Steps | * + - 1. GTA selects “Wayne State Evaluation System”.       2. Click “Select”.       3. The application will display the students in a table.       4. GTA clicks on the incomplete link to access the student’s weekly report status.       5. Misses “Submission” input field.       6. Clicks advanced for “Status”.       7. Fills out “Evaluation” by typing “Student is doing amazing”.       8. Clicks “update data”. |
| Expected Results | The student’s status will be shown on the weekly report page without the submission criteria because it wasn’t filled out.   1. Since the GTA missed the submission, the professor will only see the evaluation of the GTA and status of the weekly report. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.28.3 Grade weekly reports - Missing status criteria

|  |  |
| --- | --- |
| Test Case ID | TC-69 |
| Test Case Name | Grade weekly reports - missing status criteria |
| Description | This feature allows the GTA’s to grade students for their weekly report. The GTA’s will input the students weekly report information correctly based on how well the students did. But the GTA forgets to fill out the status portion of the grading criteria and fills out the other required criteria. |
| Preconditions | 1. User is signed into the application. 2. User is on GTA dashboard. 3. The user must be able to enter to the weekly report page. |
| Test Steps | * + - 1. GTA selects “Wayne State Evaluation System”.       2. Click “Select”.       3. The application will display the students in a table.       4. GTA clicks on the incomplete link to access the student’s weekly report status.       5. Clicks advanced for “Submission”.       6. Misses “Status”.       7. Fills out “Evaluation” by typing “Student is doing amazing”.       8. Clicks “update data”. |
| Expected Results | 1. The student’s submission will be shown on the weekly report page without the status criteria because it wasn’t filled out. 2. Since the GTA missed the submission, the professor will only see the evaluation of the GTA and submission of the weekly report. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.28.4 Grade weekly reports - Missing evaluation criteria

|  |  |
| --- | --- |
| Test Case ID | TC-70 |
| Test Case Name | Grade weekly reports - missing evaluation criteria |
| Description | This feature allows the GTA’s to grade students for their weekly report. The GTA’s will input the students weekly report information correctly based on how well the students did. But the GTA forgets to fill out the evaluation portion of the grading criteria and fills out the other required criteria. |
| Preconditions | 1. User is signed into the application. 2. User is on GTA dashboard. 3. The user must be able to enter to the weekly report page. |
| Test Steps | 1. GTA must select which group. 2. Once group selected, the GTA must click “Select”. 3. GTA clicks on the incomplete link to access the student’s weekly report status. 4. Fills out pop-up box with the submission and status. 5. Misses evaluation criteria. 6. Clicks “update data”. |
| Expected Results | 1. The student’s submission and status will be shown on the weekly report data table and will be shown on the summary page.  2. Since the GTA missed the evaluation, the professor will only see the status and submission of the weekly report. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.28.5 Grade weekly reports- Successful group report entry

|  |  |
| --- | --- |
| Test Case ID | TC-71 |
| Test Case Name | Grade weekly reports- Successful group report entry |
| Description | This feature allows the GTA’s to rate students weekly report. The weekly report has a feature which allows GTA’s to write an evaluation for the group. This feature is called group report feature, this can be accessed by click the X before the first student. |
| Preconditions | 1. User is signed into the application. 2. User is on GTA dashboard. 3. The user must be able to enter to the weekly report page. |
| Test Steps | 1. GTA must select which group.  2. Click “Select” button.  3. GTA clicks on the X.  5. GTA to enter in the group report evaluation by typing.  6. GTA clicks “update data” |
| Expected Results | The student’s submission and status must be shown on the weekly report page.  The information must display on the student page on the professor side. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.28.6 Grade weekly reports - unsuccessful group report entry

|  |  |
| --- | --- |
| Test Case ID | TC-72 |
| Test Case Name | Grade weekly reports- unsuccessful group report entry |
| Description | This feature allows the GTA’s to rate students weekly report. The weekly report has a feature which allows GTA’s to write an evaluation for the group. This feature is called group report feature, this can be accessed by click the X before the first student. |
| Preconditions | 1. User is signed into the application. 2. User is on GTA dashboard. 3. The user must be able to enter to the weekly report page. |
| Test Steps | 1. GTA must select which group. 2. Click “Select” button. 3. GTA clicks on the X. 4. GTA misses group report evaluation. 5. GTA clicks “update data” |
| Expected Results | 1. The information for the group would not display on the student page on the professor side because the GTA forgot to fill out the status of the weekly report for the group. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.28.7 Grade weekly reports- Successful jump to summary work

|  |  |
| --- | --- |
| Test Case ID | TC-73 |
| Test Case Name | Grade weekly reports- Successful jump to summary |
| Description | This feature allows the GTA’s to rate students weekly report. The weekly report has a feature which allows GTA’s to write an evaluation for the group. Once the GTA is done grading each student and the group, the GTA can see the summary of each students’ weekly report status and submission. This can be done by scrolling to the bottom of the page or clicking the jump to summary link which is located next to the weekly report table on the top left of the screen. |
| Preconditions | 1. User is signed into the application. 2. User is on GTA dashboard. 3. The user must be able to enter to the weekly report page. |
| Test Steps | * + - 1. GTA selects “Wayne State Evaluation System”.       2. Click “Select”.       3. The application will display the students in a table.       4. GTA clicks on the incomplete link to access the student’s weekly report status.       5. Clicks advanced for “Submission”.       6. Clicks advanced for “Status”.       7. Fills out “Evaluation” by typing “Student is doing amazing”.       8. Clicks “Update data”.       9. Once finishing the student weekly report. Next, finish the group report.       10. Click the “X” icon.       11. Enter “Group is doing well” by typing in the evaluation box of the group report.       12. Clicks “Update data”. |
| Expected Results | 1. After clicking the jump to summary, the GTA will be taken to the summary page which is located on the bottom of the weekly report page. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 2.29 Grade Assignments (Professor Interface)

### 2.29.1 Grade Assignments (Professor Interface)- successful entry (Group and student entry)

|  |  |
| --- | --- |
| Test Case ID | TC-74 |
| Test Case Name | Grade weekly reports- Successful grade entry |
| Description | This feature allows the professor to enter grades for students based on individual assignments. Also allows professors to enter grades for groups. |
| Preconditions | 1. User is signed into the application. 2. The user is on the “Grade assignment” page. |
| Test Steps | 1. Selects “Wayne State Evaluation System”. 2. Chooses “Development Plan”. 3. Selects “A+ - F” in the drop-down menu, for each criteria. 4. Writes down evaluation for students. 5. Clicks on button “submit grade” |
| Expected Results | 1. The assignment data table will be updated with the current grades. Meaning, once the professor clicks submit grade the table will be updated with most recent time the professor clicked “Submit grade”. 2. The grades will be updated in the view grades table. Which will allow the professors to view the grades of students in a table form. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.29.2 Grade Assignments (Professor Interface)- unsuccessful entry (Group and student entry)

|  |  |
| --- | --- |
| Test Case ID | TC-75 |
| Test Case Name | Grade weekly reports- unsuccessful grade entry |
| Description | This feature allows the professor to enter grades for students based on individual assignments. But the professor misses one of the grading criteria’s for the group and students. |
| Preconditions | 1. The professor is signed into the application. 2. The user is on the “Grade assignment” page. |
| Test Steps | 1. Selects “Wayne State Evaluation System”. 2. Chooses “Development Plan”. 3. Selects “A+ - F” in the drop-down menu, for each criteria. 4. Misses certain grades in the data table. 5. Enters “Student is doing well” and “Group is on track”. 6. Clicks on button “submit grade” |
| Expected Results | 1. The assignment data table will be updated with the current grades. Since the professor did not fill out all the grading criteria, the professor will only be able to see the filled-out criteria in the view grade page. 2. In the view grades page, since the assignment table was not filled out completely, the professor will be able to view the grades that was filled out by the GTA and the professors. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### Grade Assignments (GTA Interface)- Successful entry (Group and student entry)

|  |  |
| --- | --- |
| Test Case ID | TC-76 |
| Test Case Name | Grade weekly reports- Successful grade entry |
| Description | Allows the GTA to grade students and groups based their performance in class for each assignment. |
| Preconditions | 1. Logged in as GTA. 2. Accesses the GTA dashboard. 3. Selects the “group” for grading. |
| Test Steps | 1. Selects “Wayne State Evaluation System”. 2. Jumps to “GTA Assignment Table”. 3. Selects “A – F” in the drop-down menu for each criteria for each student and group. 4. Fills out “Evaluation” for each criteria, by typing that “Student is doing well”. 5. Fills out “Evaluation” for each criteria, by typing that “Group is doing well”. 6. Clicks “submit grade”. |
| Expected Results | 1. Once the GTA is done grading and clicks the “submit grade”. The filled-out grade will be shown in the data table (the table which the GTA filled the grades) after the “submit grade”. 2. The summary table which the professor can view will be updated with only the information the GTA fills out. In this case, all the required information is filled out. So that means the professor will be able to view all the grades which is filled out. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 2.29.4 Grade Assignments (GTA Interface)- Unsuccessful entry (Group and student entry)

|  |  |
| --- | --- |
| Test Case ID | TC-77 |
| Test Case Name | Grade weekly reports- unsuccessful grade entry |
| Description | Allows the GTA to grade students and groups based their performance in class for each assignment. However, the GTA forgets to fill out of the grading criteria for the student and group. |
| Preconditions | 1. Logged in as GTA. 2. The GTA is on the GTA dashboard. 3. The GTA has the ability select the group that needs to be graded. |
| Test Steps | 1. Selects “Wayne State Evaluation System”. 2. Jumps to “GTA Assignment Table”. 3. Misses “A – F” in the drop-down menu for each criteria for each student and group. 4. Misses out “Evaluation” for each criteria, by typing that “Student is doing well”. 5. Misses out “Evaluation” for each criteria, by typing that “Group is doing well”. 6. Clicks “submit grade”. |
| Expected Results | * + - 1. Since GTA did not fill out of all the required criteria, the GTA will only see the selected grades shown in the data table. The written evaluation will also be shown in the table.       2. The database will be updated with the grades for the students. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

# 3. Non-Functional Test Cases

## Approach

To perform testing, SQL scripts are utilized to reset the database and populate it with specific data for each test case. The tester then follows step-by-step instructions outlined in the test case and examines the results. By adopting this approach, every test is carried out using a controlled and predictable dataset, which is ideal for testing purposes.

## Pass / Fail Criteria

To pass the assessment, the function must accurately process all input and output types, execute all necessary operations with precision, adhere to stakeholder requirements, and deliver the intended outcomes. The criteria for failure include instances where the function produces unexpected results, fails to handle errors appropriately, is unable to manage specific input or output, and results in crashes or delayed response times.

## Entry / Exit Criteria

The entry criteria for non-functional testing follow the conducting of functional testing and all of those tests have passed. The exit criteria for non-functional testing is met when all the outlined test cases have been successfully completed.

## Suspension / Resumption Criteria

If a test fails, all testing must be halted until a solution is found, preventing further testing from continuing. The testing process will resume once the underlying cause of the test failure has been identified and addressed, which may involve updating either the test or the system software.

## Risks / Issues

Most functional tests require the execution of a SQL script to reset the database and insert the necessary data for the test scenario. However, this can lead to errors if the SQL script populates the database with unforeseen data, causing erroneous test failures.

## Non-functional Test Cases

### 3.6.1 Database Performance

|  |  |
| --- | --- |
| Test Case ID | TC-78 |
| Test Case Name | Database Performance |
| Description | The database must be able to display any information from the database on the screen in 5 seconds or less. |
| Preconditions | 1. Script DBS-13.sql has been run by copy and pasting it into MySQL workbench executing each line. 2. On the “Edit Groups” page. |
| Test Steps | 1. Use the search bar above the table such as the one in the “Groups” form and type an existing group name like “Testing Team”. 2. Record the time taken to display the updated data on the table. |
| Expected Results | 1. The data should be displayed on the screen within 5 seconds of clicking the "Enter" button when inputting a value in the search bar. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 3.6.2 Reliability

|  |  |
| --- | --- |
| Test Case ID | TC-79 |
| Test Case Name | Reliability |
| Description | The application should be able to handle errors based on the functional requirements stated above. |
| Preconditions | 1. Script DBS-01.sql has been run by copy and pasting it into MySQL workbench executing each line. 2. On the “Log in” page; |
| Test Steps | 1. Attempt to fill form incorrectly such as logging in with an invalid email like “ddd1231@gmail.com” or password like “testpass”. 2. Verify that the website catches the improper data, and no functionality continues its operation. |
| Expected Results | 1. The website should print out an error message informing the user that the information they provided is not accepted by the website. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 3.6.3 Availability

|  |  |
| --- | --- |
| Test Case ID | TC-80 |
| Test Case Name | Availability |
| Description | The application should only be scheduled for maintenance with the permission of the GTA’s and professor. |
| Preconditions | 1. The system is in normal operation. |
| Test Steps | 1. A time for maintenance should be scheduled by the GTA and professor. 2. When maintenance time, an application wide message will display maintenance time inbound. |
| Expected Results | 1. No user will be allowed to use it until maintenance is over. 2. Only users to have access is admins who are rolling out the maintenance. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 3.6.4 Security

|  |  |
| --- | --- |
| Test Case ID | TC-81 |
| Test Case Name | Security |
| Description | The data will be stored in MySQL database which will be properly secured for all users (Admin, Professors and GTA’s). |
| Preconditions | 1. Script DBS-20.sql has been run by copy and pasting it into MySQL workbench executing each line. 2. Attempt to sign in with email “td1234@wayne.edu” |
| Test Steps | 1. Correctly follow the instructions of the page and fill out the information as directed for set password. 2. Verify that the data is properly inserted into the database correctly by displaying user’s hashed password for added security and not just how the password was inserted by user. |
| Expected Results | 1. The information collected by users is stored in our database correctly and securely. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 3.6.5 Maintainability

|  |  |
| --- | --- |
| Test Case ID | TC-82 |
| Test Case Name | Maintainability |
| Description | With the addition of other functionalities, the application will be maintainable to admins, professors, and GTA’s. |
| Preconditions | 1. The system is in normal operation. |
| Test Steps | 1. The application is working with a new feature being added. 2. Checking old functionalities with the incorporation of new functionalities. |
| Expected Results | 1. The new features that are added must work in the application. 2. The old and new features must work together and update in the database. |
| Priority | Medium |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 3.6.6 Portability

|  |  |
| --- | --- |
| Test Case ID | TC-83 |
| Test Case Name | Portability |
| Description | The admin, professor, and GTA can use this application on any device and web browser they deem fit. |
| Preconditions | The system is in normal operation. |
| Test Steps | 1. Use phone to open application. Should notice that application is responsive. 2. Usage of different web browsers to open the application. 3. Shrinking and maximizing the application to notice the change in the application. |
| Expected Results | 1. The page should be responsive. No matter the web browser or technology, the application should be responsive and should work on all platforms. |
| Priority | Low |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 3.6.6 Usability

|  |  |
| --- | --- |
| Test Case ID | TC-84 |
| Test Case Name | Usability |
| Description | The application is usable on any device no matter the width of the screen. |
| Preconditions | 1. DBS-00.sql has been run by copy and pasting it into MySQL workbench executing each line. 2. Sign into the application as all three users |
| Test Steps | 1. Once signed into the application, right click the mouse, and select “Inspect.” 2. Click the “Toggle device toolbar” in the top left corner. 3. Change the size of the screen to see the styling change. |
| Expected Results | 1. The page should be responsive. All buttons and tables on the screen should be visible and not overlap with one another. |
| Priority | Low |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

# 4. Integration Testing

## 4.1 Approach

To perform functional testing, SQL scripts are utilized to reset the database and populate it with specific data for each test case. The tester then follows step-by-step instructions outlined in the test case and examines the results. By adopting this approach, every test is carried out using a controlled and predictable dataset, which is ideal for functional testing purposes.

## 4.2 Pass / Fail Criteria

To pass the assessment, the function must accurately process all input and output types, execute all necessary operations with precision, adhere to stakeholder requirements, and deliver the intended outcomes. The criteria for failure include instances where the function produces unexpected results, fails to handle errors appropriately, is unable to manage specific input or output, and results in crashes or delayed response times.

## 4.3 Entry / Exit Criteria

The entry criteria for integration testing follow the conducting of integration testing and all of those tests have passed. The exit criteria for integration testing is met when all the outlined test cases have been successfully completed.

## 4.4 Suspension / Resumption Criteria

If a test fails, all testing must be halted until a solution is found, preventing further testing from continuing. The testing process will resume once the underlying cause of the test failure has been identified and addressed, which may involve updating either the test or the system software.

## 4.5 Risks / Issues

Most functional tests require the execution of a SQL script to reset the database and insert the necessary data for the test scenario. However, this can lead to errors if the SQL script populates the database with unforeseen data, causing erroneous test failures.

## 4.6 Integration Test Cases

### 4.6.1 Integration between User Authentication and Role-Based Access Control

|  |  |
| --- | --- |
| Test Case ID | TC-85 |
| Test Case Name | Integration between User Authentication and Role-Based Access Control |
| Priority | High |
| Preconditions | N/A |
| Postconditions | The application is open and only allows authenticated users with the correct roles to access the application. |
| Test Steps | Open Application   1. Log in as a Admin. 2. Verify the Admin can access the Admin Home Page, Add Semester/Section and Add Professor page. 3. Log Out. 4. Log In as a professor. 5. Verify the professor can access the Home, Add GTAs, Add Students, Add Section, Add Groups, Edit Groups and Grade Assignments. 6. Log out. 7. Log in as a GTA. 8. Verify the GTA can access the Home, Weekly Reports and Choose Groups. |
| Expected Results | 1. When a user attempts to access a page that requires authentication they will be logged in and assigned to a role or Admin, Professor or GTA. 2. After a user logs in with valid credentials, the ser is redirected to the appropriate Home page based on their assigned role. 3. If a user attempts to log in to access a page for a role they are not assigned to, they will be directed only to the pages they have authentication to use. 4. If a user logs out, they will be redirected to the log in page and should not be able to access any authenticated pages until they log back in. |

### 4.6.2 Integration between Course Information and Evaluation Data

|  |  |
| --- | --- |
| Test Case ID | TC-86 |
| Test Case Name | Integration between Course Information and Evaluation Data |
| Priority | High |
| Preconditions | N/A |
| Postconditions | The course information and evaluation data are integrated correctly and displayed in the professor homepage. |
| Test Steps | 1. Log In 2. Verify that the relevant course information in the professor Homepage, including the Section, Group Name, and Assignment data is present and accurate. 3. Verify that the relevant evaluation data has been added to the system. This includes choose grade, notes, consistency, grammar, topics & correctness, and resubmission data. 4. Fill out an evaluation form and submit it for a Document or Prototype. 5. Navigate to homepage. 6. Navigate to the group that was just evaluated. 7. Click on any student. 8. Verify the document or prototype grades are displayed and accurate. |
| Expected Results | 1. The course information is correct in the system. 2. The evaluation data is expected to be stored accurately within the system, reflecting the appropriate group and professor it corresponds to. 3. The evaluation data inputted by the professor is expected to be displayed correctly for each individual student. This includes ensuring that the data matches up with the appropriate course, instructor, and student information within the system. |

### 4.6.3 Integration between Evaluation Data and Database

|  |  |
| --- | --- |
| Test Case ID | TC-87 |
| Test Case Name | Integration between Evaluation Data and Database |
| Priority | High |
| Preconditions | N/A |
| Postconditions | The evaluation data is accurately stored within the database, allowing for accurate retrieval when necessary. |
| Test Steps | 1. Navigate to the professor home page. 2. Click grade assignments. 3. Click on a section. 4. Click on a group. 5. Click on an assignment. 6. Submit a grade for the selected assignment. 7. Verify the submitted evaluation data accurately reflects the evaluation data in the database. 8. Verify the applications performance to ensure the integration between evaluation data and the database have no effect on the systems functionality. |
| Expected Results | 1. The evaluation data is accurately reflecting in the generated reports. 2. The layout and formatting of the generated reports are consistent. 3. The integration between the evaluation data and the database has no negative effects on the systems performance. |

### 4.6.4 Integration between Professor and GTA Views

|  |  |
| --- | --- |
| Test Case ID | TC-88 |
| Test Case Name | Integration between Professor and GTA Views |
| Priority | High |
| Preconditions | N/A |
| Postconditions | All evaluation data is correctly displayed in the Professor and GTA views. |
| Test Steps | 1. Log in as a GTA. 2. Navigate to the weekly reports page. 3. Select a group. 4. Click on an “incomplete” weekly report for a student and fill out the form. 5. Submit the form. 6. Sign out. 7. Log in as a professor. 8. Click on the same group and student that was just evaluated by the GTA. 9. Verify the data is displayed accurately in the Weekly Reports view. 10. Verify the date submitted and week number is accurate. |
| Expected Results | 1. The evaluation data updated by the GTA is displayed correctly in the Professor and GTA views. 2. The professor and GTA can update student evaluations accurately. 3. The integration does not affect the system’s ability to perform its necessary functionalities. 4. The professor and GTA each have access to the appropriate information according to their system role. |

# 5. System Testing

## 5.1 Approach

During system testing, the entire live system will be subjected to parameterized test cases. To create the initial admin, professor, and GTA users, only one SQL script will be employed throughout the system testing level. All other data will be live and generated by utilizing the Capstone Course Evaluation System. This approach aims to evaluate the Capstone Course Evaluation System's performance within a more realistic environment. These tests are similar to the ones tested above however it goes in order in which information is created in order to use that data.

## 5.2 Pass / Fail Criteria

To pass the assessment, the function must accurately process all input and output types, execute all necessary operations with precision, adhere to stakeholder requirements, and deliver the intended outcomes. The criteria for failure include instances where the function produces unexpected results, fails to handle errors appropriately, is unable to manage specific input or output, and results in crashes or delayed response times.

## 5.3 Entry / Exit Criteria

The entry criteria for system testing follow the conducting of functional testing and all of those tests have passed. The exit criteria for integration testing is met when all the outlined test cases have been successfully completed.

## 5.4 Suspension / Resumption Criteria

If a test fails, all testing must be halted until a solution is found, preventing further testing from continuing. The testing process will resume once the underlying cause of the test failure has been identified and addressed, which may involve updating either the test or the system software.

## 5.5 Risks / Issues

Because there is only one SQL script to run, this introduces the possibility of errors in handling unforeseen data. It is crucial to execute the test cases in a specific order since executing certain test cases creates data required for other tests. Failure to execute the test cases in the designated order may result in false test failures.

## 5.6 System Testing – Admin User

### 5.6.1 Sign in - Admin

|  |  |
| --- | --- |
| Test Case ID | TC-89 |
| Test Case Name | Sign in – Admin |
| Preconditions | 1. The user must already have been created. |
| Test Steps | 1. Enter “[admin@wayne.edu](mailto:admin@wayne.edu)” in the username input field. 2. Click the “Next” button. 3. Enter “pass” in the password input field. 4. Click the “Submit” button. |
| Expected Results | A new session is created for the current user.  The user is redirected to the admin dashboard. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 5.6.2 Add Professor - Admin

|  |  |
| --- | --- |
| Test Case ID | TC-90 |
| Test Case Name | Add Professor – Admin |
| Preconditions | 1. The admin is signed into the application. 2. The admin is on the “Add Professors” page, located in the side navigation bar. |
| Test Steps | 1. Enter a first name like “Lily” into the first name input field. 2. Enter a last name like “Appleton” into the last name input field. 3. Enter an email like “[la9485@wayne.edu](mailto:la9485@wayne.edu)”in the email input field. 4. Click the “Add” button. |
| Expected Results | 1. The page will reload for the user with the message “Professor added successfully.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 5.6.3 Create a Semester - Admin

|  |  |
| --- | --- |
| Test Case ID | TC-91 |
| Test Case Name | Create a Semester – Admin |
| Preconditions | 1. The admin is signed into the application. |
| Test Steps | 1. Enter a semester name in the form “Fall2020” in the semester name input field. 2. Choose a start date or type in a date (in the form MM/DD/YYYY) in the start date input field. 3. Choose an end date or type in a date (in the form MM/DD/YYYY) in the end date input field. 4. Click the “Submit” button. |
| Expected Results | 1. The page will reload with the message “Semester added successfully” and will now be in the application. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 5.6.4 Create a Section - Admin

|  |  |
| --- | --- |
| Test Case ID | TC-92 |
| Test Case Name | Create a Section – Admin |
| Preconditions | 1. The admin is signed into the application. 2. A professor has been added to the application. 3. There should be an active semester in the system. |
| Test Steps | 1. Enter a section name in the form “Section010” in the section name input field. 2. Choose a semester from the semester drop-down menu. 3. Choose a professor from professor the drop-down menu. 4. Click the “Add” button. |
| Expected Results | * + - 1. The page will reload with the message “Section added successfully.”       2. The professor will now have access to that section. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 5.7 System Testing – Professor User

### 5.7.1 Sign in - Professor

|  |  |
| --- | --- |
| Test Case ID | TC-93 |
| Test Case Name | Sign in – Professor |
| Preconditions | 1. The user must already have been created. |
| Test Steps | Enter “[professor@wayne.edu](mailto:professor@wayne.edu)” in the username input field.   1. Click the “Next” button. 2. Enter “pass123” in the password input field. 3. Click the “Submit” button. |
| Expected Results | 1. A new session is created for the current user. 2. The user is redirected to the professor dashboard. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 5.7.2 Add Section - Professor

|  |  |
| --- | --- |
| Test Case ID | TC-94 |
| Test Case Name | Add Section – Professor |
| Preconditions | 1. The admin is signed into the application is on the “Add Section” page, located in the side navigation bar. |
| Test Steps | 1. Enter a section name in the form “Section010” in the section name input field. 2. Click the “Add” button. |
| Expected Results | * + - 1. The page will reload for the user with the message “Section added successfully.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 5.7.3 Add GTA - Professor

|  |  |
| --- | --- |
| Test Case ID | TC-95 |
| Test Case Name | Add GTA – Professor |
| Preconditions | 1. The professor is signed into the application is on the “Add GTA” page, located in the side navigation bar. 2. A section has already been added for the professor. |
| Test Steps | 1. Enter a first name like “Lily” into the first name input field. 2. Enter a last name like “Appleton” into the last name input field. 3. Enter an email like “[la9485@wayne.edu](mailto:la9485@wayne.edu)”in the email input field. 4. From the section drop-down menu, select the section you would like to add the GTA to. 5. Click the “Add” button. |
| Expected Results | 1. The page will reload for the user with the message “GTA added successfully.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 5.7.4 Add Students - Professor

|  |  |
| --- | --- |
| Test Case ID | TC-96 |
| Test Case Name | Add Students – Professor |
| Preconditions | 1. The professor is signed in and has navigated to the “Add Students” page. 2. A section has been created. |
| Test Steps | 1. Upload a file in CSV, .ODS, .XLSX, .XLS) form, with the student’s access ID in one column and the student’s name in the other. 2. From the section drop-down menu, select the section you would like to add the students to. 3. Click “Upload.” |
| Expected Results | 1. The page will reload for the user with the message “Students added successfully.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 5.7.5 Add Group - Professor

|  |  |
| --- | --- |
| Test Case ID | TC-97 |
| Test Case Name | Add Group – Professor |
| Preconditions | 1. The professor is signed in and has navigated to the “Add Groups” page. 2. A section has been created with students added to that section. |
| Test Steps | 1. Click the tab with the section that you uploaded the students to. 2. In the “Assign your students to a group”, select the group you created from the group-down menu. 3. Click the radio-button next to the students’ names that you would like to add to that group. 4. Click “Assign Students” |
| Expected Results | * + - 1. The page will reload for the user with the message “Groups added successfully.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 5.7.6 Assign Students to Group - Professor

|  |  |
| --- | --- |
| Test Case ID | TC-98 |
| Test Case Name | Assign Students to Group – Professor |
| Preconditions | 1. The professor is signed in and has navigated to the “Add Groups” page. 2. A group has been created. |
| Test Steps | 1. Click the tab with the section that you uploaded the students to. 2. In the “Add a Group” form, enter a project group name. 3. Click the “Save Group(s)” button. |
| Expected Results | 1. The page will reload for the user with the message “Students assigned successfully.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 5.7.7 Grade Assignments - Professor

|  |  |
| --- | --- |
| Test Case ID | TC-99 |
| Test Case Name | Grade Assignments - Professor |
| Preconditions | 1. The professor is signed in and has navigated to the “Grade Assignment” page. 2. A group has been created with students added into it. |
| Test Steps | 1. Click the tab with the section that you uploaded the students to. 2. Click the tab with the group you would like to grade. 3. Click the tab with the name of the assignment you would like to grade. 4. Fill out the form. 5. Click “Submit form” button. |
| Expected Results | 1. The page will reload for the user with the message “Update Success.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

## 5.8 System Testing – GTA User

### 5.8.1 Sign in - GTA

|  |  |
| --- | --- |
| Test Case ID | TC-100 |
| Test Case Name | Sign in – GTA |
| Preconditions | 1. The user must already have been created. |
| Test Steps | Enter “[gta@wayne.edu](mailto:gta@wayne.edu)” in the username input field.  Click the “Next” button.  Enter “test12” in the password input field.  Click the “Submit” button. |
| Expected Results | 1. A new session is created for the current user. 2. The user is redirected to the GTA dashboard. |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 5.8.2 Choose Group - GTA

|  |  |
| --- | --- |
| Test Case ID | TC-101 |
| Test Case Name | Choose Group – GTA |
| Preconditions | 1. The GTA is signed in and has located to “Choose Groups” page. |
| Test Steps | 1. In the “Choose your Groups” form, click the radio-button next to the groups names that you would like to add to that group. 2. Click the “Assign” button. |
| Expected Results | * + - 1. The page will reload for the user with the message “Group Assigned.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 5.8.3 Grade Assignments - GTA

|  |  |
| --- | --- |
| Test Case ID | TC-102 |
| Test Case Name | Grade Assignments - GTA |
| Preconditions | 1. The GTA is signed in and is on their dashboard or their “Home” page. |
| Test Steps | 1. Click the group button with whom you would like to grade. 2. Click the tab with the name of the assignment you would like to grade. 3. Fill out the form. 4. Click “Submit form” button. |
| Expected Results | 1. The page will reload for the user with the message “Update Success.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

### 5.8.3 Grade Weekly Report - GTA

|  |  |
| --- | --- |
| Test Case ID | TC-103 |
| Test Case Name | Grade Assignments - GTA |
| Preconditions | 1. The GTA is signed in and is navigated to the “Weekly Reports” page. 2. The GTA has a group assigned to them |
| Test Steps | 1. Select the group in the drop-down menu you would like to grade. 2. Click the “Select” button. 3. A table appears with the students in the group along the x-axis with the weeks along the y-axis with bubbles that say “Incomplete.” 4. Click the “Incomplete” button with the corresponding week and student name that you would like to grade. 5. Fill out the form. 6. Click the “Update Data” button. |
| Expected Results | 1. The page will reload for the user with the message “Update Success.” |
| Priority | High |
| Pass/Fail Criteria | The test will pass if all expected results are achieved; otherwise, it will fail. |

# Schedule

All the test cases were implemented early and throughout the development process to ensure that the system would be able to function to the best of its ability. Below is a list of the test cases next to the team member name who tested and confirmed that the functionality works.

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Name** | **Deadline Date** |
| TC-1 | Cristina Powers | 03/15/2023 |
| TC-2 | Cristina Powers | 03/15/2023 |
| TC-3 | Cristina Powers | 03/15/2023 |
| TC-4 | Cristina Powers | 03/15/2023 |
| TC-5 | Cristina Powers | 03/15/2023 |
| TC-6 | Cristina Powers | 03/15/2023 |
| TC-7 | Cristina Powers | 03/15/2023 |
| TC-8 | Cristina Powers | 03/15/2023 |
| TC-9 | Cristina Powers | 03/15/2023 |
| TC-10 | Cristina Powers | 03/15/2023 |
| TC-11 | Cristina Powers | 03/15/2023 |
| TC-12 | Cristina Powers | 03/15/2023 |
| TC-13 | Cristina Powers | 03/15/2023 |
| TC-14 | Cristina Powers | 03/15/2023 |
| TC-15 | Cristina Powers | 03/15/2023 |
| TC-16 | Cristina Powers | 03/15/2023 |
| TC-17 | Cristina Powers | 03/15/2023 |
| TC-18 | Cristina Powers | 03/15/2023 |
| TC-19 | Cristina Powers | 03/15/2023 |
| TC-20 | Cristina Powers | 03/15/2023 |
| TC-21 | Cristina Powers | 03/15/2023 |
| TC-22 | Cristina Powers | 03/15/2023 |
| TC-23 | Cristina Powers | 03/15/2023 |
| TC-24 | Cristina Powers | 03/15/2023 |
| TC-25 | Bharath Palanisamy | 03/18/2023 |
| TC-26 | Bharath Palanisamy | 03/18/2023 |
| TC-27 | Bharath Palanisamy | 03/18/2023 |
| TC-28 | Bharath Palanisamy | 03/18/2023 |
| TC-29 | Bharath Palanisamy | 03/18/2023 |
| TC-30 | Bharath Palanisamy | 03/18/2023 |
| TC-31 | Bharath Palanisamy | 03/18/2023 |
| TC-32 | Bharath Palanisamy | 03/18/2023 |
| TC-33 | Bharath Palanisamy | 03/18/2023 |
| TC-34 | Bharath Palanisamy | 03/18/2023 |
| TC-35 | Abigail Noyes | 03/21/2023 |
| TC-36 | Abigail Noyes | 03/21/2023 |
| TC-37 | Abigail Noyes | 03/21/2023 |
| TC-38 | Abigail Noyes | 03/21/2023 |
| TC-39 | Abigail Noyes | 03/21/2023 |
| TC-40 | Abigail Noyes | 03/21/2023 |
| TC-41 | Abigail Noyes | 03/21/2023 |
| TC-42 | Abigail Noyes | 03/21/2023 |
| TC-43 | Bharath Palanisamy | 03/21/2023 |
| TC-44 | Bharath Palanisamy | 03/23/2023 |
| TC-45 | Abigail Noyes | 03/23/2023 |
| TC-46 | Abigail Noyes | 03/23/2023 |
| TC-47 | Abigail Noyes | 03/23/2023 |
| TC-48 | Abigail Noyes | 03/23/2023 |
| TC-49 | Abigail Noyes | 03/23/2023 |
| TC-50 | Abigail Noyes | 03/23/2023 |
| TC-51 | Abigail Noyes | 03/23/2023 |
| TC-52 | Raad Bhuiyan | 03/23/2023 |
| TC-53 | Raad Bhuiyan | 03/23/2023 |
| TC-54 | Raad Bhuiyan | 03/23/2023 |
| TC-55 | Raad Bhuiyan | 03/23/2023 |
| TC-56 | Raad Bhuiyan | 03/25/2023 |
| TC-57 | Raad Bhuiyan | 03/25/2023 |
| TC-58 | Raad Bhuiyan | 03/25/2023 |
| TC-59 | Raad Bhuiyan | 03/25/2023 |
| TC-60 | Raad Bhuiyan | 03/25/2023 |
| TC-61 | Raad Bhuiyan | 03/25/2023 |
| TC-62 | Raad Bhuiyan | 03/25/2023 |
| TC-63 | Abigail Noyes | 03/25/2023 |
| TC-64 | Abigail Noyes | 03/25/2023 |
| TC-65 | Raad Bhuiyan | 03/25/2023 |
| TC-66 | Bharath Palanisamy | 03/25/2023 |
| TC-67 | Bharath Palanisamy | 03/25/2023 |
| TC-68 | Bharath Palanisamy | 03/28/2023 |
| TC-69 | Bharath Palanisamy | 03/28/2023 |
| TC-70 | Bharath Palanisamy | 03/28/2023 |
| TC-71 | Bharath Palanisamy | 03/28/2023 |
| TC-72 | Bharath Palanisamy | 03/28/2023 |
| TC-73 | Bharath Palanisamy | 03/28/2023 |
| TC-74 | Bharath Palanisamy | 03/28/2023 |
| TC-75 | Bharath Palanisamy | 03/28/2023 |
| TC-76 | Bharath Palanisamy | 03/28/2023 |
| TC-77 | Bharath Palanisamy | 03/28/2023 |
| TC-78 | Raad Bhuiyan | 03/28/2023 |
| TC-79 | Raad Bhuiyan | 03/30/2023 |
| TC-80 | Bharath Palanisamy | 03/30/2023 |
| TC-81 | Raad Bhuiyan | 03/30/2023 |
| TC-82 | Bharath Palanisamy | 03/30/2023 |
| TC-83 | Bharath Palanisamy | 03/30/2023 |
| TC-84 | Cristina Powers | 03/30/2023 |
| TC-85 | Abigail Noyes | 03/30/2023 |
| TC-86 | Abigail Noyes | 03/30/2023 |
| TC-87 | Abigail Noyes | 03/30/2023 |
| TC-88 | Abigail Noyes | 03/30/2023 |
| TC-89 | Cristina Powers | 04/02/2023 |
| TC-90 | Cristina Powers | 04/02/2023 |
| TC-91 | Cristina Powers | 04/02/2023 |
| TC-92 | Cristina Powers | 04/02/2023 |
| TC-93 | Cristina Powers | 04/02/2023 |
| TC-94 | Cristina Powers | 04/02/2023 |
| TC-95 | Cristina Powers | 04/02/2023 |
| TC-96 | Cristina Powers | 04/02/2023 |
| TC-97 | Cristina Powers | 04/02/2023 |
| TC-98 | Cristina Powers | 04/02/2023 |
| TC-99 | Cristina Powers | 04/02/2023 |
| TC-100 | Cristina Powers | 04/02/2023 |
| TC-101 | Cristina Powers | 04/02/2023 |
| TC-102 | Cristina Powers | 04/02/2023 |
| TC-103 | Cristina Powers | 04/02/2023 |

# Traceability Matrix

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

# Appendix

|  |  |
| --- | --- |
| Script ID | Data Description |
| DBS-00 | Creates the database with all the necessary tables inserting 1 Admin user (email = [admin@wayne.edu](mailto:admin@wayne.edu), password = Hash of “pass”, roleOfUser = “admin”, activeUser = 1)  1 Professor user (email = [professor@wayne.edu](mailto:professor@wayne.edu), password = Hash of “pass123”, roleOfUser = “professor”, activeUser =1)  1 GTA user (email = [gta@wayne.edu](mailto:gta@wayne.edu), password = Hash of “test12”, roleOfUser = “GTA”, activeUser = 1) |
| DBS-01 | Insert users(firstname, lastname, email, pass, roleOfUser, activeUser) VALUES (“John”, “Smith”, “[jb1234@wayne.edu](mailto:jb1234@wayne.edu)”, Hash of “pass”, “professor”, 0) |
| DBS-02 | Insert users(firstname, lastname, email, roleOfUser, activeUser) VALUES ([Your first name], [Your last Name], [your email], admin, 1) |
| DBS-10 | Insert a student  students (accessID = “jk3333”, activeStudent = 1, sectioned = 1, name = “Smith John”) |
| DBS-11 | Insert a group (sectionid = 1, groupName= “Test Team”) |
| DBS-12 | Insert students and create empty group.  students (accessID = “sd1233”, activeStudent = 1, sectioned = 1, name = “Lou Will”)  students (accessID = “as8122”, activeStudent = 1, sectioned = 1, name = “Jenny Lee”)  Insert a group (groupid = 999, sectionid = 1, groupName= “Testing Team”) |
| DBS-13 | Insert a group (groupid = 1000, sectionid = 1, groupName= “Testing Team”)  students (accessID = “je8790”, activeStudent = 1, sectioned = 1, name = “Mary Jane”, groupid = 1000)  students (accessID = “oc9100”, activeStudent = 1, sectioned = 1, name = “Sara Steve”, groupid = 1000)  GTA user (userid = 99, firstname = “Curtis”, lastname = “Don”, email = cd1234[@wayne.edu](mailto:gta@wayne.edu), password = Hash of “test12”, roleOfUser = “GTA”, activeUser = 1)  GTA user (userid = 100, firstname = “Nick”, lastname = “Smith”, email = ns1234[@wayne.edu](mailto:gta@wayne.edu), password = Hash of “test12”, roleOfUser = “GTA”, activeUser = 1)  Insert GTA into Group (sectionid = 1, gta = 99, groupid = 1000) |
| DBS-14 | insert into semester (semester\_name, startDate, endDate) VALUES ('Winter2023', STR\_TO\_DATE('2023-01-09', '%Y-%m-%d'), STR\_TO\_DATE('2023-04-22', '%Y-%m-%d'));  insert into section (section\_name, semesterid, professor) VALUES ('Section111', 1, 2); |
| DBS-20 | Insert GTA user (firstname = “Todd”, lastname = “Johnson”, email = [td1234@wayne.edu](mailto:td1234@wayne.edu), roleofuser= “GTA” activeuser = 1) |