System Test Specification

# Team 14

Luke Beisser

Carissa Bostian

Tyrone Bundy

Alex Johnson

April 19, 2023

Used SDS by Team #14 (CSCN 471 Only)

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Section/Sub-section Title** | **Author**  **[person(s) with primary accountability**  **for this section]** | **Page #** |
| 1. Test Plan   1.1 Testing Team Organization and Resource Assignments 1.2 Testing Schedule 1.3 Test Cases | Carissa Bostian | 3 |
| 1. Test Specification 2.1 Test Case … 2.n Test Case | Luke Beisser | 5 |
| 1. Test Results 3.1 URL, Login Information, and Tester Instructions for the Working Application 3.2 Test Results Table | Alex Johnson | 9 |
| Appendix A - Updated Requirements Traceability Matrix | Tyrone Bundy | 12 |

# 1. Test Plan **[Carissa Bostian]**

## 1.1. Testing Team Organization and Resource Assignments

|  |  |
| --- | --- |
| **Team Member** | **Roles and Responsibilities** |
| Luke Beisser | Test Analyst. Define and describe steps for each test requirement. |
| Carissa Bostian | Test Manager. Organize and create a testing schedule, assign team roles, define test cases. |
| Tyrone Bundy | Team Organizer. Update and define requirements and traceability matrix. |
| Alex Johnson | Software Tester. Describe the test case results with input and expected output. |

***1.2. Testing Schedule***  
*N/A*

***1.3. Test Cases***

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Requirement Number** | **Test Goal** | **Test Method/Technique** |
| 1 | 1.1.1 | Pages Required:  · Login  · Forgot Password  · Index  · Assignments  · CRID Students / Faculty  · Profile  · Classes | Black box - Equivalence Partitioning |
| 2 | 1.2.2 | Login Page:  · Sends credentials to backend database for validation  · Authenticates user roles  · Form submission capabilities  · Smartphone capabilities | Black box - Equivalence Partitioning |
| 3 | 1.4.3 | Private Data will be encrypted | White Box –  Condition Testing |
| 4 | 1.2.3 | Index page must have a F4 function to search as well as query the database for the resulting information | Black box - Equivalence Partitioning |
| 5 | 1.2.4 | Assignments page must have F3 function to search for the file to be submitted | Black box - Equivalence Partitioning |
| 6 | 1.3.3 | Services:  · MySQL for the database  · ReactJS for frontend webpages  · NodeJS for backend programming  · Azure for cloud computing | White box – Condition Testing |
| 7 | 2.1.3 | The system must be able to support at least 550 user accounts | Black box - Equivalence Partitioning |

# 2. Test Specification **[Luke Beisser]**

***2.1 Test Case ID: 1***

|  |  |
| --- | --- |
| **Test Case ID: 1** | **Goal: Find all pages required** |
| **Steps** | **Sub-Steps** |
| 1.Verify Login Page | |  | | --- | | 1.1 Go to site URL | | 1.2 Check for login before entry | |  | |
| 2. Verify Forgot Password | |  | | --- | | 2.1 Click on forgot password | | 2.2 Check for redirection to password resetter | |  | |
| 3.Verify Index system | |  | | --- | | 3.1 Locate index system | | |  | | --- | |  | | |
| 4.Verify Assignments page | |  | | --- | | 4.1 Locate Assignments page | | |  | | --- | | 4.2 Check for user assignment Info | |  | | |
| 5.Verify CRID students/Faculty | |  | | --- | | 5.1 Check for student/ faculty verification | |  | |
| 6.Verify Profile | |  | | --- | | 6.1 Check for profile page | | |  | | --- | | 6.2 Correct profile shown? | |  | | |
| 7.Verify Classes page | |  | | --- | | 7.1 Locate Classes page | | |  | | --- | | 7.2 Check for user Class Info | |  | | |

**Test Completion Indicator:** All pages have been verified.

**Evaluation Process:** Ensure user checked correct URL and pages.

***2.2 Test Case ID: 2***

|  |  |
| --- | --- |
| **Test Case ID: 2** | **Goal: Check for proper login** |
| **Steps** | **Sub-Steps** |
| 1.Verify Login Page | |  | | --- | | * 1. Go to site URL.   2. Check for login before entry. | |
| 2. Load all Incorrect info | |  |  | | --- | --- | | 2.1 Enter variations of incorrect Usernames.  2.2 Enter variations of incorrect Passwords.   |  | | --- | |  | | |
| 3. Load correct username | 3.1 Enter correct username.  3.2 Enter incorrect password. |
| 4. Load correct password | 4.1 Enter incorrect username.   * 1. Enter correct password. |
| 5. Load similar username and passwords | 5.1 Enter incorrect variations of the correct username.  5.2 Enter incorrect variations of the correct password. |
| 6. Load all correct info | 6.1 Enter correct username.  6.2 Enter correct password. |

**Test Completion Indicator:** User has logged in and gained system access.

**Evaluation Process:** Ensure tester has tested all possibilities to prevent erroneous user system access.

***2.3 Test Case ID: 3***

|  |  |
| --- | --- |
| **Test Case ID: 3** | **Goal: Check encryption** |
| **Steps** | **Sub-Steps** |
| 1. Check private data | |  | | --- | | 1.1 Go into the system  1.2 Try and access private data  1.3 Check data is encrypted | |

**Test Completion Indicator:** Private data is encrypted

**Evaluation Process:** Try and access data

***2.4 Test Case ID: 4***

|  |  |
| --- | --- |
| **Test Case ID: 4** | **Goal: Check for search and query ability in index** |
| **Steps** | **Sub-Steps** |
| 1. Locate Index | |  | | --- | | 1.1 Navigate to index | |
| 2. Check ability of F4 | 2.1 Hit F4 key  2.2 Find the resulting pop up  2.3 Search for a word you see  2.4 Type a SQL query to pull a table from data. |

**Test Completion Indicator:** User can search and query from the F4 key.

**Evaluation Process:** Follow the steps listed in the test case.

***2.5 Test Case ID: 5***

|  |  |
| --- | --- |
| **Test Case ID: 5** | **Goal: Check for searchability in Assignments** |
| **Steps** | **Sub-Steps** |
| 1. Locate assignments | |  | | --- | | 1.1 Navigate to assignments | |
| 2. Check ability of F3 | 2.1 Hit F3 key  2.2 Find search bar  2.3 Enter an existing file |

**Test Completion Indicator:** User can search from the F3 key.

**Evaluation Process:** Use the F3 key and search

***2.6 Test Case ID: 6***

|  |  |
| --- | --- |
| **Test Case ID: 6** | **Goal: Check for correct services** |
| **Steps** | **Sub-Steps** |
| 1. Check for mySQL | |  | | --- | | 1.1 Inspect the Source code  1.2 Check for mySQL to be used as DB | |
| 2. Check for ReactJS | 2.1 Inspect the source code  2.2 Check that ReactJS is use |
| 3. Check for NodeJS | 3.1 Inspect the source code  3.2 Check that NodeJS is used |
| 4. Check for Azure | 4.1 Inspect the source code  4.2 Check that Azure is used for cloud storage |

**Test Completion Indicator:** The system uses the correct services

**Evaluation Process:** Check for all the required services

***2.7 Test Case ID: 7***

|  |  |
| --- | --- |
| **Test Case ID: 7** | **Goal: System can handle traffic** |
| **Steps** | **Sub-Steps** |
| 1. Simulate users | |  | | --- | | 1.1 Make user accounts  1.2 Create script to login users  1.3 Run script | |
| 2. Test system | 2.1 Check system operates with 550 users |

**Test Completion Indicator:** System can handle 550 users

**Evaluation Process:** Login 550 users

# 3. Test Results **[Alex Johnson]**

***3.1. URL, Login Information, and Tester Instructions for the Working Application***

## N/A

***3.2. Test Results Table***

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Input & Program State** | **Expected Output** | **Tester** |
| 1 | 1.1 User accessed website  1.2 User gets directed to Login page | 1.1 Login page is verified | Johnson, Alex |
| 1 | 2.1 User enters valid User ID  2.2 User clicks on “forgot password” prompt | 2.1 User gets directed to the forgot password page  2.2 Forgot password page is verified | Johnson, Alex |
| 1 | 3.1 User gains system access  3.2 User clicks to Index system page link | 3.1 Index system page is verified | Johnson, Alex |
| 1 | 4.1 User gains system access  4.2 User directs to Assignment page  4.3 User clicks on assignment info link | 4.1 Assignments page is verified | Johnson, Alex |
| 1 | 5.1 User gains system access  5.2 User clicks on system/faculty verification page link | 5.1 CRID students/faculty page is verified | Johnson, Alex |
| 1 | 6.1 User gains system access  6.2 User clicks on profile page link | 6.1 Profile page is verified | Johnson, Alex |
| 1 | 7.1 User gains system access  7.2 User clicks on classes page link | 7.1 Classes page is verified | Johnson, Alex |
| 2 | * 1. User enters invalid username   2. User enters invalid password | 1.1 user does not gain system access | Johnson, Alex |
| 2 | 2.1 User enters valid username  2.2 User enters invalid password | 2.1 User does not gain system access | Johnson, Alex |
| 2 | 3.1 User enters invalid username  3.2 User enters valid password | 3.1 user does not gain system access | Johnson, Alex |
| 2 | 4.1 User enters valid username  4.2 user enters a valid, but incorrect, password | 4.1 user does not gain system access | Johnson, Alex |
| 2 | 5.1 User enters valid and correct username  5.2 User enters valid and correct password | 5.1 User gains system access | Johnson, Alex |
| 3 | 1.1 Developer checks the private data  1.2 ensure that the private data is encrypted | 1.1 private data is encrypted | Johnson, Alex |
| 4 | 1.1 User navigates to Index page  1.2 User hits F4 key | 1.1 Search table pops up for user input | Johnson, Alex |
| 5 | 1.1 User navigates to assignments page  1.2 User hits F3 key | 1.1 Search key pops up for user input | Johnson, Alex |
| 6 | 1.1 Developer inspects source code  1.2 Developer ensures that MySQL is being used in relation to the database | 1.1 MySQL is used in the code for the database communication | Johnson, Alex |
| 6 | 2.1 Developer inspects the source code  2.2 Developer ensures that ReactJS is being used | 2.1 ReactJS is being used in the source code | Johnson, Alex |
| 6 | 3.1 Developer inspects the source code  3.2 Developer ensures that NodeJS is being used | 3.1 NodeJS is being used in the source code | Johnson, Alex |
| 6 | 4.1 Developer tests Azure application | 4.1 Azure is being used for the application | Johnson, Alex |
| 7 | 1.1 create user script for maximum user count  1.2 run user simulation script  1.3 test on system to ensure capability | 1.1 maximum user count is feasible after running script | Johnson, Alex |

**Appendix A: Updated Requirements Traceability Matrix [Tyrone Bundy]**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Priority** | **Requirement # by Category** | **Description** | **MVC Scenario Section** | **SRS Section** | **SDS** **Section** | **STS** **Section** |
| N/A | 1 | **Interface Requirements** | Technical Overview: Functional Components | 4 | 4 | 2.1 |
| N/A | 1.1 | ***User Interfaces*** | Technical Overview: Functional Components | 4.1 | 5 | 2.1 |
| 2 | 1.1.1 | Pages Required:   * Login * Forgot Password * Index * Assignments * CRUD Students / Faculty * Profile * Classes | Technical Overview: Functional Components | 4.1 | 5 | N/A |
| N/A | 1.2 | ***Hardware Interfaces*** | Technical Overview: Technology Environment | 4.2 | 2.2, 2.3 | 2.2 |
| 1 | 1.2.1 | MVC Web App requires internet, TCP/IP protocols, HTTPS, and prevents access to insecure networks. | Technical Overview: Security | 4.2 | 2.2, 1.3 | N/A |
| 1 | 1.2.2 | Login Page:   * Sends credentials to backend database for validation * Authenticates user roles * Form submission capabilities * Smartphone capabilities | Technical Overview: Functional Components | 4.2 | 5.3.1 | N/A |
| 2 | 1.2.3 | Index page must have a F4 function to search as well as query the database for the resulting information | Technical Overview: Functional Components | 4.2 | 2.2,2.3 | N/A |
| 2 | 1.2.4 | Assignments page must have F3 function to search for the file to be submitted | Technical Overview: Functional Components | 4.2 | 5.3.7.3 | N/A |
| 2 | 1.2.5 | Assignments page must have F4 function for redirection to grading view | Technical Overview: Functional Components | 4.2 | 5.3.7.3 | N/A |
| 2 | 1.2.6 | Assignments page must have F6 function to quick edit assignments | Technical Overview: Functional Components | 4.2 | 5.3.7.3 | N/A |
| N/A | 1.3 | ***Software Interfaces*** | Technical Overview: Technology Environment | 4.3 | N/A | 2.6 |
| 1 | 1.3.1 | Connections:   * MySQL 8.0.32 * Windows 7/10/11, MacOS 10 * ReactJS, NodeJS, Azure | Technical Overview: Technology Environment | 4.3 | 2.1 | N/A |
| 1 | 1.3.2 | Data Items/Messages:   * Error Alerts * Success Alerts * Upcoming Events | Technical Overview: Data | 4.3 | 2.4 | N/A |
| 1 | 1.3.3 | Services:   * MySQL for the database * ReactJS for frontend webpages * NodeJS for backend programming * Azure for cloud computing solution | Technical Overview: Technology Environment | 4.3 | 2.1, 1.3 | N/A |
| 1 | 1.3.4 | Shared Data will have submission travel through the backend as well as each time a page is opened the data will be grabbed from the backend | Technical Overview: Technology Environment | 4.3 | 3.2, 2.4.1 | N/A |
| N/A | 1.4 | ***Communications Interfaces*** | Technical Overview: Functional Components | 4.4 | 2.2 | 2.3 |
| 2 | 1.4.1 | Requirements:   * College email to every student * Chrome or Firefox for web browser * HTTPS for secure communications | Technical Overview: Functional Components | 4.4 | 2.1, 3.2 | N/A |
| 2 | 1.4.2 | Message Formatting will use HTTPS | Technical Overview: Security | 4.4 | 1.3 | N/A |
| 2 | 1.4.3 | Private data will be encrypted | Technical Overview: Security | 4.4 | 3.1 | N/A |
| N/A | 2 | **Nonfunctional Requirements** | Technical Overview: Functional Components | 5 | 5 | N/A |
| N/A | 2.1 | ***Performance Requirements*** | Technical Overview: Functional Components | 5.1 | 2 | 2.7 |
| 2 | 2.1.1 | The system must be able to process 500 requests a second | Technical Overview: Functional Components | 5.1 | 2.1 | N/A |
| 2 | 2.1.2 | The system must possess a response time of 1.0 seconds or less | Technical Overview: Functional Components | 5.1 | 2.1 | N/A |
| 2 | 2.1.3 | The system must be able to support at least 550 user accounts | Technical Overview: Functional Components | 5.1 | 2.1 | N/A |
| 2 | 2.1.4 | The system requires the most current network infrastructure, updated servers and most current browser | Technical Overview: Functional Components | 5.1 | 2.1 | N/A |
| N/A | 2.2 | ***Safety Requirements*** | MVC Scenario | 5.2 | 2 | 2.3 |
| 1 | 2.2.1 | Adhere to Privacy Rights Act and FTC Act by securing personal data | MVC Scenario | 5.2 | 2.2 | N/A |
| 1 | 2.2.2 | MVC must adhere to Property Protection and Claims Management Policy | MVC Scenario | 5.2 | 2.1 | N/A |
| 1 | 2.2.3 | MVC SIS must have an agreement for Account Usage Terms and Conditions | MVC Scenario | 5.2 | 5 | N/A |
| N/A | 2.3 | ***Security Requirements*** | Technical Overview: Security | 5.3 | 2.1 | 2.3 |
| 2 | 2.3.1 | Data should be encrypted using AES | Technical Overview: Security | 5.3 | 2.1 | N/A |
| 2 | 2.3.2 | Data backups must happen once a week | Technical Overview: Security | 5.3 | 2.1 | N/A |
| 1 | 2.3.3 | Compliance with FTC | Technical Overview: Security | 5.3 | 2.1 | N/A |
| 2 | 2.3.4 | Human Error Policy:   * Strong passwords * No device left unattended * Student data restricted to SIS * Software downloading restrictions | Technical Overview: Security | 5.3 | 2.4.2 | N/A |
| 2 | 2.3.5 | Background checks, IDS, and zero trust architecture to prevent insider threats | Technical Overview: Security | 5.3 | 2.1 | N/A |
| 2 | 2.3.6 | Input validation and parametrized queries to mitigate SQL injection risk | Technical Overview: Security | 5.3 | 2.1 | N/A |
| N/A | 2.4 | ***Software Quality Attributes*** | Technical Overview: Technology Environment | 5.4 | 4 | N/A |
| 1 | 2.4.1 | Attributes:   * Student data can be modified * Proper replacement of hardware * Cyber resiliency for system robustness * SUS for testing system usability | Technical Overview: Technology Environment | 5.4 | 4.6.3 | N/A |
| N/A | 3 | **Other Requirements** | MVC Scenario | 6 | 2 | N/A |
| N/A | 3.1 | ***Database Requirements*** | Technical Overview: Technology Environment | 6 | 2.1 | N/A |
| 1 | 3.1.1 | Proper database management support must be present | Technical Overview: Technology Environment | 6 | 2.1 | N/A |
| 2 | 3.1.2 | Access Control Lists to determine database access privileges | Technical Overview: Security | 6 | 2.1, 3.2 | N/A |
| N/A | 3.2 | ***Legal Requirements*** | MVC Scenario | 6 | 2.1 | N/A |
| 1 | 3.2.1 | MVC SIS must display a Privacy Policy | MVC Scenario | 6 | 2.1 | N/A |
| 1 | 3.2.2 | MVC SIS must display a Cookie Policy | MVC Scenario | 6 | 2.1 | N/A |
| 1 | 3.2.3 | MVC SIS must display and require agreement on Account Usage Terms and Conditions | MVC Scenario | 6 | 2.1 | N/A |
| N/A | 3.3 | ***Reuse Requirements*** | MVC Scenario | 6 | 2.1 | N/A |
| 2 | 3.3.1 | MVC student data is to be reused, as well as the database and infrastructure | MVC Scenario | 6 | 2.1 | N/A |