



SPARK

COURSE MANAGEMENT SYSTEM

TEST REPORT

Version 1.0 Alpha
May 3rd, 2020

VERSION HISTORY

The development and distribution of the Spark - Course Management System Test Report was controlled and tracked through various machines with different hardware specifications and operating systems from each team member.

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Luis Guzman	05/03/2020	Luis Guzman	05/03/2020	Initial testing report
1.0	Forrest Smith	05/03/2020	Forrest Smith	05/03/2020	Initial testing report
1.0	Anita Kotowska	05/03/2020	Anita Kotowska	05/03/2020	Initial testing report

Table of Contents

1.0	INTRODUCTION	4
2.0	TESTING ENVIRONMENT	4
3.0	UNIT TESTING	5
3.1	Testing Objective	5
3.2	Testing Summary	5
3.3	Test Cases	10
4.0	INTEGRATION TEST	23
4.1	Testing Objective	23
4.2	Testing Summary	23
4.3	Test Cases	23
5.0	SYSTEM TEST	26
5.1	Testing Objective	PG #
5.2	Testing Summary	PG #
5.3	Test Cases	PG #

1.0 INTRODUCTION

This Spark - Course Management System Test Report provides a summary of the results of tests performed as outlined within this document.

2.0 TESTING ENVIRONMENT

For our testing environment, we had 3 different machines, one from each group member:

System #1 - Luis Guzman:

Windows 10 Operating System

Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz

12 GB of Installed RAM

System #2 - Forrest Smith:

Windows 10 Operating System

Intel i9-9900K @ 4.1ghz

16GB DDR4 RAM

System #3 - Anita Kotowska:

Windows 10 Operating System

Intel(R) Core(TM) i7-4700HQ CPU @ 2.40GHz

8 GB of Installed RAM

3.0 UNIT TESTING

3.1 TESTING OBJECTIVE

The objective of unit testing is to ensure all corners of the Spark - Course Management System platform are properly functioning. All use cases for each user, Student, Professor and Administrators are individually tested through thorough test procedural steps.

3.2 TEST SUMMARY

Function	Description	%TCs Executed	%TCs Passed	TCs Pending	Priority	Remarks
EnterCourse	Enter course view for selected course	100	100	0	H	
initHomePage	Initializes Home Tab with appropriate information for each residing element	100	100	0	H	
ViewAssignmentDetails	Action taken for when "View Assignment Details" button is clicked on the Home Tab	100	100	0	H	
initAssignments (StudentCourseView)	Initializes Assignments Tab with appropriate information for each residing element	100	100	0	H	
displayAssignmentDetails	Displays selected Assignment details on the side bar	100	100	0	H	
PublishAssignment	Retrieves Student Submission and writes to Course Assignment	100	100	0	H	
UploadAttachment	Prompts user to select file and initializes global uploadFile variable	100	100	0	H	
initGrades	Initializes Grades	100	100	0	H	

Spark - Course Management System

	Tab with appropriate information for each residing element					
initAssignments (ProfessorCourseView)	Initialize the Assignments tab by populating the list of assignments	100	100	0	H	
AddNewAssignment	Clear input fields to prepare for assignment creation	100	100	0	H	
PublishAssignment	Publish and save new assignment with input	100	100	0	H	
EditPublishAssignment	Edit a selected assignment and re-publish it	100	100	0	H	
DeleteAssignment	Remove an assignment from a course's set of assignments	100	100	0	H	
initSubmissions	Initialize the Submissions tab with the list of assignments and the student submissions they hold	100	100	0	H	
SubmitFeedback	Submit written feedback and grade for a submission	100	100	0	H	
DeleteFeedback	Remove feedback (write-up and grade) for a submission	100	100	0	H	
DownloadSubmission Attachment	Download an attachment, if a student uploaded one, to downloads folder	100	100	0	H	
initAnnouncements	Initialize	100	100	0	H	

Spark - Course Management System

	Announcements tab with list of announcements					
AddNewAnnouncement	Clear input fields to prepare for new announcement creation	100	100	0	H	
PublishAnnouncement	Create and add a new announcement to course based on input fields	100	100	0	H	
EditPublishAnnouncement	Edit and re-publish a selected announcement	100	100	0	H	
DeleteAnnouncement	Remove a selected announcement from the course's set of announcements	100	100	0	H	
initStudents	Initialize Students tab with list of students, their submissions, and course analytics	100	100	0	H	
SubmitFinalGrade	Enter a final grade for a student into the GradeBook. Either the calculated grade or custom adjusted grade	100	100	0	H	
DeleteFinalGrade	Remove a student's final grade from the GradeBook	100	100	0	H	
RefreshClassAnalytics	Recalculate and re-display class analytics	100	100	0	H	
AddUser	Adds a new User on button push	100	100	0	H	
AddCourse	Adds new Course	100	100	0	H	

Spark - Course Management System

RegisterStudent	Button clicked to register a student to class that is currently being viewed	100	100	0	H	
registerAndUpdate	Helper method to register a student to a course and updates the list view	100	100	0	H	
updateStudentListView	Helper method to update the list view with newly registered students	100	100	0	H	
handleMouseClickedStudents (AdminStudentView)	Mouse click handler for list of students. Changes labels and tableView list of courses to match currently selected student	100	100	0	H	
setCourseList	Helper method updates tableView list of courses for a selected student	100	100	0	H	
handleMouseClickedStudents (AdminSearchResult)	Mouse click handler for list of students. Changes labels and tableView list of courses to match currently selected student	100	100	0	H	
setTableView	Set courses and corresponding course grade of selected student from mouse click handler to table view	100	100	0	H	
CreateCourse	Creating a new course handled by NewCourseDialogC	100	100	0	H	

Spark - Course Management System

	ontroller					
CreateUser	Creating a new user handled by NewUserDialogController	100	100	0	H	
SearchStudent	Search for student containing input	100	100	0	H	
ViewProfessor	Go into course view for selected professor	100	100	0	H	
handleMouseClicked (AdminDashboardController)	Handler for admin clicking on a professor from list of professors. Updates courses taught list, students taught list, and overall average student grade	100	100	0	H	
ViewCourse	Goes to the student view of selected course	100	100	0	H	
handleMouseClickedCourse	Populate TableView with courses selected student is taking and the students grade in that course	100	100	0	H	

3.3 TEST CASES

3.3.1 STUDENT TEST CASES

Test Case Identifier	TC-1 Luis
Use Case Tested	UC-1 - ViewClasses
Test Case Description	Check to see if the appropriate details for the classes a Student is registered for are displayed.
Pass / Fail Criteria	The test passes if the Student user account can see all the classes they're registered for and can see the right details upon entering a course page. The test fails if the Student user account sees incorrect classes they're not registered for.
Input	Student Account, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Log into a Student account 2. Compare if Student Dashboard page displays all classes a Student is registered in and matches the data from the Courses.JSON file (<i>course.registeredStudents</i>) 3. Log out of Student account and log in as Admin account 4. Register Student account into new course 5. Log out of Admin account and log into Student account 6. View if newly registered class appears on Student Dashboard
Output & Result	All classes a student is registered for are displayed on the Student Dashboard page. The test case passed as the pass criteria from above was successfully met and all Test Procedural Steps passed.

Test Case Identifier	TC-2 Luis
Use Case Tested	UC-2 - ViewClassAssignments
Test Case Description	Check to see if a Student can view all assignments for a specific class the Student is registered for.
Pass / Fail Criteria	The test passes if the Student user account can see all the class assignments along with each assignment details. The test fails if the Student user account cannot see all the class assignments.
Input	Student Account, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Log into a Student account

Spark - Course Management System

	<ol style="list-style-type: none"> 2. Select a Course from the Dashboard Page 3. Compare if all class assignments are displayed and are matched to the Courses.JSON file (<i>course.assignments</i>) 4. Log out of Student account and log into Professor account 5. Add a new Assignment to a specific course 6. Log out of the Professor account and log into Student account 7. View if new assignment appears for the registered students
Output & Result	<p>Class Assignments are properly displayed on the course assignments tab page.</p> <p>The test case passed as the pass criteria from above was successfully met and all Test Procedural Steps passed.</p>

Test Case Identifier	TC-3 Luis
Use Case Tested	UC-3 - SubmitAssignment
Test Case Description	Check to see if a Student can add a submission to an assignment from a class that they're registered for.
Pass / Fail Criteria	<p>The test passes if the Student user account can make a submission to a class assignment and that submission is stored.</p> <p>The test fails if the Student user account cannot make a submission to a class assignment.</p>
Input	Student Account, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Log into a Student account 2. Select an Assignment from a specific Course 3. Add submission text and click "Submit" button 4. Log out of Student account and log back into same Student account 5. Compare if submission text remains the same
Output & Result	<p>Pop up dialog appears saying "Submission Accepted!"</p> <p>The test case passed as the pass criteria from above was successfully met and all Test Procedural Steps passed.</p>

Test Case Identifier	TC-4 Luis
Use Case Tested	UC-4 - ViewAnnouncements
Test Case Description	Check to see if a Student can view all announcements a Professor published for a Course they're registered in.
Pass / Fail Criteria	The test passes if the Student user account can view all

Spark - Course Management System

	<p>announcements a Professor has published.</p> <p>The test fails if the Student user account cannot view all course announcements.</p>
Input	Student Account, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Log into a Student account 2. Select a Course from the Student Dashboard page 3. Compare if all class announcements are displayed and are matched to the Courses.JSON file (<i>course.announcements</i>) 4. Log out of Student account and log back into Professor account 5. Publish new announcement to a course 6. Log into Student account and determine if recently published announcement can be viewed
Output & Result	<p>Class announcements are properly displayed on the course assignments tab page.</p> <p>The test case passed as the pass criteria from above was successfully met and all Test Procedural Steps passed.</p>

Test Case Identifier	TC-5 Luis
Use Case Tested	UC-5 - ViewClassmates
Test Case Description	Check to see if a Student can view all classmates for a specific course they're registered for.
Pass / Fail Criteria	<p>The test passes if the Student user account can view all classmates for the course they're registered in.</p> <p>The test fails if the Student user account cannot view all classmates for the course they're registered in.</p>
Input	Student Account, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Log into a Student account 2. Select a Course from the Student Dashboard page 3. Compare if all classmate usernames from the home tab page are displayed and are matched to the Courses.JSON file (<i>course.registeredStudents</i>) 4. Log out of Student account and log into Admin account 5. Register new Student to a course 6. Log into Student account and determine if recently added Student username is displayed under classmates list
Output & Result	<p>Classmates are properly displayed on the course home tab page.</p> <p>The test case passed as the pass criteria from above was successfully met and all Test Procedural Steps passed.</p>

Test Case Identifier	TC-6 Luis
Use Case Tested	UC-6 - ViewClassGrades
Test Case Description	Check to see if a Student can view their assignment grades and overall class grade
Pass / Fail Criteria	<p>The test passes if the Student user account can view their assignment grades and overall class grade. If the assigned Professor of the class has not submitted a grade yet, "Pending Grade" is to appear.</p> <p>The test fails if none of the above actions from above happen.</p>
Input	Student Account, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Log into a Student account 2. Select a Course from the Student Dashboard page 3. Compare if Student assignment grades are displayed and are matched to the Courses.JSON file (<i>course.studentSubmissions.grade</i>) 4. Compare if Student overall course grade is displayed and matches to the Courses.JSON file (<i>course.gradeBookFinal</i>) 5. Log out of Student account and log into Professor account 6. Adjust Student assignment grades and course grade 7. Log into Student account and determine if recently adjusted grades are displayed
Output & Result	<p>Course grade and assignment grades are properly displayed on the course "My Grades" tab page.</p> <p>The test case passed as the pass criteria from above was successfully met and all Test Procedural Steps passed.</p>

3.3.2 PROFESSOR TEST CASES

Test Case Identifier	TC-7 Forrest
Use Case Tested	(UC-7 - Publish Assignment)
Test Case Description	To publish specific class assignments for students to view and submit responses to
Pass / Fail Criteria	Pass: If the assignments is preserved between sessions and can be viewed and completed by student Fail: If the assignment is not stored in any way and cannot be observed by the student
Input	Users.JSON, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Log into Professor account 2. Select a course from the professor dashboard 3. Create a new assignment with “Publish” button 4. Repeat steps 1 - 2 and view the assignment 5. Log in as a student enrolled in the course 6. Check in the Assignments tab that the new assignment exists 7. Submit a response 8. Repeat steps 1 - 2 9. Check for student submissions
Output & Result	Courses.JSON is updated to reflect the new assignment

Test Case Identifier	TC-8 Forrest
Use Case Tested	(UC-8 - EditAssignment)
Test Case Description	To edit an existing assignment and have the same results as TC-7
Pass / Fail Criteria	Pass: An existing assignment is updated with new field inputs Fail: The targeted assignment is not changed to show new field input, or is deleted
Input	Users.JSON, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Log into Professor account 2. Select a course from the professor dashboard 3. Create a new assignment with “Publish” button 4. Repeat steps 1 - 2 and view the assignment 5. Log in as a student enrolled in the course

Spark - Course Management System

	<ol style="list-style-type: none">6. Check in the Assignments tab that the new assignment exists7. Submit a response8. Repeat steps 1 - 29. Check for student submissions
Output & Result	Courses.JSON is updated to reflect the edited assignment

Test Case Identifier	TC-9 Forrest
Use Case Tested	(UC-9 - GradeAssignment)
Test Case Description	Grade a student's submission for an assignment with feedback
Pass / Fail Criteria	Pass: Student can view grade and feedback. Students' calculated grade is adjusted Fail: Student cannot see grade and feedback. Student's calculated grade is not adjusted
Input	Users.JSON, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none">1. While logged in as a professor, navigate to the Submissions pane of a course2. Submit a feedback description and grade for any student submission.3. View that their calculated grade has changed in the Students tab.4. While logged in as that student, navigate to the assignments page to view the assignment's feedback and grade
Output & Result	Courses.JSON is updated to reflect the graded submission, and student's automatic grade.

Test Case Identifier	TC-10 Forrest
Use Case Tested	UC-10 - PublishAnnouncement
Test Case Description	Publish an announcement for a class. All students registered for the class can see the announcement and its details (title, date, body)
Pass / Fail Criteria	Pass: Students can see announcements Fail: Students cannot see announcements
Input	Users.JSON, Courses.JSON

Spark - Course Management System

Test Procedural Steps	<ol style="list-style-type: none"> 1. While logged in as a professor, navigate to the Announcements pane of a course 2. Create and publish a new announcement 3. While logged into a student account registered for the course, view on the homepage the new announcement
Output & Result	Courses.JSON is updated to reflect the new announcement

Test Case Identifier	TC-11 Forrest
Use Case Tested	UC-11 - GradeStudent
Test Case Description	Assign a final grade for the class for a student
Pass / Fail Criteria	Pass: Student will see their final course grade Fail: Student does not see their final course grade
Input	Users.JSON, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. While logged in as a professor, navigate to the Students tab of a course 2. Select a student and assign a final grade (either calculated or adjusted) 3. Log in to that student's account to check their final grade on the My Grades tab of their course view.
Output & Result	Courses.JSON is updated with an assignment of the Student's final grade in the final grade book.

Test Case Identifier	TC-12 Forrest
Use Case Tested	(UC-12 - ViewStudentDetails)
Test Case Description	View details of a registered student
Pass / Fail Criteria	Pass: Professor can view the full name and ID of a student registered in one of their courses Fail: A professor cannot view a student's details
Input	Courses.JSON, Users.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. While logged in as a professor, navigate to the Students tab of a course 2. Select a student and view their details
Output & Result	Professor can see on the right side of the Students tab, the details of a selected student

3.3.3 ADMINISTRATOR TEST CASES

Test Case Identifier	TC-14 Anita
Use Case Tested	UC-14 - ViewProfessorDetails
Test Case Description	View a report of a Professor's class(es).
Pass / Fail Criteria	Pass: Administrator can view the course names and IDs taught by a professor, the full names and IDs of students registered in all of their courses, and the average grade of all the students they have taught Fail: An administrator cannot view the professor's details
Input	Courses.JSON, Users.JSON
Test Procedural Steps	1. While logged in as an Administrator, select a professor from the list view on the left side
Output & Result	Administrator can view the professor details on the right side of the dashboard

Test Case Identifier	TC-15 Anita
Use Case Tested	UC-15 - SearchStudents
Test Case Description	Search for a Student in the University by name or ID.
Pass / Fail Criteria	Pass: Administrator enters text into the search bar that matches at least one Student in the system Fail: Administrator presses search without entering input or there are no Students matching the input
Input	Users.JSON
Test Procedural Steps	1. While logged in as an Administrator, enter text into the search bar 2. Click search button
Output & Result	Administrators can view the results that match the inputted text in the "search results" window. If there are no results, an alert is displayed in the newly opened window.

Spark - Course Management System

Test Case Identifier	TC-16 Anita
Use Case Tested	UC-16 - ViewAnyStudentDetails
Test Case Description	View the details of any individual Student.
Pass / Fail Criteria	Pass: Administrator can view any students name, ID, registered courses, and grades Fail: Administrator cannot view any of the details of any student
Input	Courses.JSON, Users.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. While logged in as an Administrator, enter text into input and select the search button 2. Select a student from the list view on the left 3. Select back button 4. Select a Professor from the list view on the left and click the View Professor button 5. Select a course from the list view on the left and click the View Course button 6. Select a student from the list view on the left to view their details
Output & Result	A student's name, ID, grade, and courses are displayed on the right side of the page.

Test Case Identifier	TC-17 Anita
Use Case Tested	UC-17 - CreateUserAccount
Test Case Description	Create login accounts for Students and Professors.
Pass / Fail Criteria	Pass: A user with the inputted username does not already exist. Fail: A user with the inputted username already exists
Input	Users.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Select the Add New User button from the login page 2. Enter new users first name, last name, username, password, and select the user type from the drop down menu 3. Click add new user 4. Log in as an Administrator 5. Select the New User button 6. Repeat step 2

Spark - Course Management System

Output & Result	A new user is added to the users database. If an administrator has added a new Professor user from their dashboard, the list view is refreshed with the new professor
----------------------------	--

Test Case Identifier	TC-19 Anita
Use Case Tested	UC-19 - RegisterStudentIntoClass
Test Case Description	Register a specific student into a specific class.
Pass / Fail Criteria	Pass: The selected student has been registered into the class and is saved in that course to the database Fail: The selected student has not been registered into the class and is not visible when viewing the list of students in the course
Input	Courses.JSON, Users.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. While logged in as an Administrator, select a Professor from the list view on the left and click the View Professor button 2. Select a course from the list view on the left and click the View Course button 3. Select the Register a Student button 4. Select a student from the drop down menu and click OK
Output & Result	The student is added to the list view of students in the course that they have been newly registered to. The student is also added to the courses database.

Test Case Identifier	TC-20 Anita
Use Case Tested	UC-20 - CreateCourse
Test Case Description	Create a new course and assign course details.
Pass / Fail Criteria	Pass: A course with the inputted ID does not already exist Fail: A course with the inputted ID already exists
Input	Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. While logged in as an Administrator, select the New Course button 2. Enter the new course title, ID, and select a Professor from the drop down menu 3. Select the Add New Course button

Spark - Course Management System

Output & Result	
-----------------	--

	A new course is added to the courses database.
--	--

3.3.4 GENERAL TEST CASES

Test Case Identifier	TC-21 Forrest
Use Case Tested	UC-21 - Login
Test Case Description	Can any user successfully log into system and load their data
Pass / Fail Criteria	Pass: Each account can only be logged into by their unique username and password Fail: A user account can be logged into
Input	Users.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Log in as some user with valid credentials and view the dashboard is correct. 2. Log out, and try to log-in to the same account with a different password
Output & Result	Appropriate user data is loaded upon logging in

Test Case Identifier	TC-22 Forrest
Use Case Tested	UC-22 - Logout
Test Case Description	Can any user successfully log out of system and log into another account without issue
Pass / Fail Criteria	Pass: A user can successfully log out so that another user can log in and load their specific data Fail: Logging out does not clear data from previous session
Input	Users.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Log in as some user with valid credentials and view the dashboard is correct. 2. Log out, and try to log-in to a different account with a different password
Output & Result	A new user can log in and load their data

Test Case Identifier	TC-23 Forrest
Use Case Tested	UC-23 - LoadData
Test Case Description	Can load relevant data after logging in as a user and selecting a course, or as administrator

Spark - Course Management System

Pass / Fail Criteria	Pass: A user can successfully log out so that another user can log in and load their specific data Fail: Logging out does not clear data from previous session
Input	Users.JSON, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Log in as a student or professor 2. Enter a course, and verify its contents are correct 3. Log out, then log in as an admin 4. Verify all students, professors, and courses are visible
Output & Result	All correct Spark data is loaded.

Test Case Identifier	TC-24 Forrest
Use Case Tested	UC-24 - SaveData
Test Case Description	Can save relevant data after any action taken by a user, so any quit session has all data saved up until the latest action.
Pass / Fail Criteria	Pass: After any action that saves data, reloading after quitting the application the new data from the last session is preserved. The JSON files are updated accordingly. Fail: Data is not preserved between sessions. The JSON files are not updated accordingly.
Input	Users.JSON, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Log in as a Student or Professor and enter a course 2. Take any action that saves (creating a new assignment, announcement, submission; editing or grading an assignment; submitting a final grade) 3. Check the JSON files, or quit and reload Spark to check persistence of the data
Output & Result	All Spark data is saved, and can be re-loaded in new sessions

4.0 INTEGRATION TESTING

4.1 TESTING OBJECTIVE

The objective of integration testing is to ensure all individually created units are combined and tested together to ensure they're all properly interacting with each other accordingly. This testing focuses more on checking the data communication between shared object classes. The type of integration we used was Top-Down Integration as we initially designed all shared interfaces between each user first. Thereafter, each team member focused on designing the user experience for a specific user type.

4.2 TESTING SUMMARY

Test cases were made to test data transfer between scenes, and ensure that proper pages are connected. These scenes involved the dashboards for Student and Professor accounts, the course views for Student and Professor accounts, and the Admin views. Each test case was successful.

4.3 TESTING CASES

Test Case Identifier	TC-25 Login Page Data Transfer
Test Case Objective	Verify the interface link between the login page and the home page based upon the user type.
Pass / Fail Criteria	The test passes if the correct home page for a specific type of user is displayed. The test fails if the incorrect user home page is displayed.
Input	Login Credentials, Users.JSON
Test Procedural Steps	<ol style="list-style-type: none">1. Log into a Student account2. Verify the Student Dashboard page is displayed3. Logout and log into Professor account4. Verify the Professor Dashboard page is displayed5. Logout and log into Administrator account6. Verify the Administrator Dashboard page is displayed
Output & Result	All proper interface links between the login page and the home page are correctly set up. The test case passed as the pass criteria from above was successfully met and all Test Procedural Steps passed.

Test Case Identifier	TC-26 Student Dashboard to Course Page Data Transfer
Test Case Objective	Verify the interface link between the Student Dashboard page and the Student Course View page.
Pass / Fail Criteria	The test passes if the correct information for a selected class is

Spark - Course Management System

	displayed on the Student Course View page. The test fails if the incorrect course information is passed from the Student Dashboard page to the Student Course View page.
Input	Student Account, Course Selection, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none">1. Log into a Student account2. Select a course from the Student Dashboard Page3. Verify the Course information displayed such as announcements, classmates, assignments, student submissions and assignment/course grades.
Output & Result	All proper information between the Student Dashboard page and the Student Course View page are correctly set up. The test case passed as the pass criteria from above was successfully met and all Test Procedural Steps passed.

Test Case Identifier	TC-27 Professor Dashboard to Course Page Data Transfer
Test Case Objective	Verify the interface link between the Professor Dashboard page and the Professor Course View page.
Pass / Fail Criteria	The test passes if the correct information for a selected class is displayed on the Professor Course View page. The test fails if the incorrect course information is passed from the Professor Dashboard page to the Professor Course View page.
Input	Professor Account, Course Selection, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none">1. Log into a Professor account2. Select a course from the Professor Dashboard Page3. Verify the Course information displayed such as announcements, classmates, assignments, student submissions and assignment/course grades.
Output & Result	All proper information between the Professor Dashboard page and the Professor Course View page are correctly set up. The test case passed as the pass criteria from above was successfully met and all Test Procedural Steps passed.

Test Case Identifier	TC-28 Administrator Dashboard to Administrator Courses Page Data Transfer
Test Case Objective	Verify the interface link between the Administrator Dashboard and the Courses View page
Pass / Fail Criteria	The test passes if the correct information for a selected Professor is displayed on the Courses View page. The test fails if the incorrect professor information is passed from the Administrator Dashboard page to the Courses View page.
Input	Administrator Account, Professor Selection, Users.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Log into an Administrator account 2. Select a professor from the Administrator Dashboard Page 3. Verify the Professor information displayed (courses are all taught by selected professor)
Output & Result	All proper information between the Administrator Dashboard page and the Administrator Courses page are correctly set up. The test case passed as the pass criteria from above was successfully met and all Test Procedural Steps passed.

Test Case Identifier	TC-29 Administrator Courses Page to Administrator Students Page Data Transfer
Test Case Objective	Verify the interface link between the Administrator Courses page and the Administrator Students page
Pass / Fail Criteria	The test passes if the correct information for a selected Course is displayed on the Students View page. The test fails if the incorrect Course information is passed from the Administrator Courses page to the Administrator Students page.
Input	Administrator Account, Course Selection, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Log into an Administrator account 2. Select a professor from the Administrator Dashboard Page 3. Select a course from the Courses page 4. Verify the Course information displayed (students are all registered to selected course)
Output & Result	All proper information between the Administrator Courses page and the Administrator Students page are correctly set up. The test case passed as the pass criteria from above was

Spark - Course Management System

	successfully met and all Test Procedural Steps passed.
--	--

Test Case Identifier	TC-29 Administrator Courses Page to Administrator Student Results Page Data Transfer
Test Case Objective	Verify the interface link between the Administrator Dashboard page and the Administrator Student Results page
Pass / Fail Criteria	The test passes if the correct result information for the inputted text is displayed on the Student Results page. The test fails if the incorrect student information is passed from the Administrator Courses page to the Administrator Student Results page.
Input	Administrator Account, Text Input, Users.JSON
Test Procedural Steps	<ol style="list-style-type: none">1. Log into an Administrator account2. Type text into the search bar and select the Search button3. Verify the Student information displayed (students matching inputted text)
Output & Result	All proper information between the Administrator Dashboard page and the Administrator Student Results page are correctly set up. The test case passed as the pass criteria from above was successfully met and all Test Procedural Steps passed.

5.0 SYSTEM TESTING

5.1 TESTING OBJECTIVE

The objective of system testing is to ensure the system as a whole is properly working from a user point of view. The requirements and expectations for each type of user and the overall system are laid out in the Spark - Course Management System Software Requirements Specification.

5.2 TESTING SUMMARY

Various tests were executed to ensure the system is working from end to end from a user's perspective using test data. Test cases performed include Functionality testing, Performance testing, and Security testing all of which successfully passed and fulfilled their objective.

5.3 TESTING CASES

Test Case Identifier	TC-28 Functionality Testing
Test Case Objective	Verify the functionality of Spark is working per the requirements defined in the Software Requirements Specification.
Pass / Fail Criteria	The test passes if the correct functionality for each user is allowed to be completed and each user requirement is fulfilled. The test fails if any user has incorrect functionality within the program.
Input	Users.JSON, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none">1. Log into each type of user one at a time2. Perform all functional requirements laid out in the Software Requirements Specification (pg. 11) for the logged in user type3. Log out and repeat until all user types have been tested
Output & Result	All functional requirements were successfully able to be performed. The test case passed as the pass criteria from above was successfully met and all Test Procedural Steps passed.

Spark - Course Management System

Test Case Identifier	TC-29 Performance Testing
Test Case Objective	Verify the system's performance under various conditions using different machine types with large amounts of data to read in.
Pass / Fail Criteria	The test passes if the program can successfully read in large amounts of user and course data while keeping its stability. The test fails if the program unsuccessfully reads in large amounts of user and course data.
Input	Users.JSON, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Log into each type of user one at a time 2. Perform all functional requirements laid out in the Software Requirements Specification (pg. 11) for the logged in user type 3. Analyze load speed and response time 4. Log out and repeat until all user types have been tested
Output & Result	All functional requirements were successfully able to be performed in a timely manner without affecting the overall user experience. The test case passed as the pass criteria from above was successfully met and all Test Procedural Steps passed.

Test Case Identifier	TC-30 Security Testing
Test Case Objective	Verify the system does not allow unauthorized access to data and resources unless permitted to.
Pass / Fail Criteria	The test passes if the correct authorization to read and write data for each user type matches the Functional Requirements from the SRS (pg. 11). The test fails if any user has incorrect authorization to data within the program.
Input	Users.JSON, Courses.JSON
Test Procedural Steps	<ol style="list-style-type: none"> 1. Log into each type of user one at a time 2. Analyze the data accessible to the logged in user and perform all functional requirements based on the SRS for the logged in user type 3. Log out and repeat until all user types have been tested
Output & Result	All users had the right permissions to data they've been granted to. The test case passed as the pass criteria from above was successfully met and all Test Procedural Steps passed.