

Software Testing: An Introduction

Team 1: Edmon Adams, Joshua Jones, Trevor Fellbaum

The purpose of this document is to provide testing documentation of Team 1's Class Enrollment System. The testing process begins with defining the requirements for the software system. These requirements were obtained from Team 1's SRS and reproduced for the reader's convenience. The reason for this duplication is because the requirements of our software span across functionality, efficiency, and the interface; this was a more logical way to present the software's requirements rather than the having the reader search through the entire SRS for this information. This is due to the sheer amount of requirements this software system has. Then, a list of testing requirements was generated. Each testing requirement can be traced to individual requirements, listed in the previous section. Every requirement can be traced to a testing requirement and vice-versa. The next section covers the test cases for the testing requirements. Every testing requirement has been covered by the test cases. Finally, the documentation concludes with a testing report. This report highlights whether the test case passed or failed. If the test case failed, an explanation of how it failed is included.

Requirements Specification

REQ 1: The software must add classes into a shopping cart or actually enroll the student in less than five seconds after the student has hit add to shopping cart or enroll.

REQ 2: The web application requires an internet connection and will communicate via HTTP on various web browsers. It also requires communications to a MySQL database where the data will be secured.

REQ 3: Users must be able to search for a specific class after provided with recommended classes.

REQ 4: Students should not be able to change their grades, see other students grades, or alter classes, majors, or minors in any way, shape, or form. Only admins have that privilege. So students must not be able to access the admin portal.

REQ 5: Admin or student logins must be encrypted as well so that credentials are not sent as plain text over the connection.

REQ 6: There will be two displays for classes enrolled during the Selected term on this screen. The first display will be a list view while the other is a calendar view. There will be buttons to toggle academic terms, go to the browse classes screen, and logout.

REQ 7: Classes will appear as pill-shaped elements and will immediately start populating the search display when the user selects their category. There will be buttons to search keywords, choose a new category, and logout. By clicking on a class, the pill element will expand revealing more class information as well as the option to preference time or professor when selecting a section to enroll in. The interface will cycle through time select, professor select, and section selects screens where the user chooses from a list of times, professors, and section based off their preferences and actions on prior screens. The user will then proceed to the enrollment screen.

REQ 8: There will be a button for cancelling and confirming the enrollment in a course.

REQ 9: The user will provide a username and password via text fields to access their account including all of their stored data so they know which classes they have taken and which classes they need to take.

REQ 10: When a password is correctly entered, the UI will display information pertaining to the username entered.

REQ 11: When a username is correctly entered, the text in the password field leads the UI to display information pertaining to the username entered when the password matches the

password set for that username.

REQ 12: The password field, when a username is correctly entered, leads the UI to not display information pertaining to the username entered when the password does not match the password set for the username.

REQ 13: The user will input the classes they select via buttons for enrollment and will be enrolled in those classes in their selected term if they meet the requirements.

REQ 14: Software will display enrollment of selected term in both calendar and list form.

REQ 15: The user can input a class code to search for classes via text field.

REQ 16: The user can input a keyword that into a text field to find classes.

REQ 17: The user will select filters or groups to find classes organized by a given characteristic.

REQ 18: Classes will be displayed based on search criteria in list form.

REQ 19: Software will provide a degree audit in the form of displaying which criteria the students have not met in the filter dropdown.

REQ 20: The software will write to the database the classes the user enrolls in.

REQ 21: Students will be able to view their term history via a dropdown menu. This will be complete with course descriptions and grades.

REQ 22: Software must be able to pull class details stored in the database.

REQ 23: Software must be able to recognize what terms the student has been with the institute.

REQ 24: Software must be able to display all database information.

REQ 25: Classes will be recommended based on previous course completion, major(s), and minor(s).

REQ 26: The user can filter their classes via a dropdown box.

REQ 27: Software must be able to communicate with class details stored in database.

REQ 28: Software must know what classes the student has taken and recommend classes based off that information with maximum accuracy.

REQ 29: Users must be able to add courses to their shopping cart, but not be able to enroll past the maximum amount of credit hours that the University allows.

REQ 30: Enrolled classes should be added to the students information in the database.

REQ 31: Only unfulfilled categories should be displayed in the filters dropdown box.

REQ 32: All class information should be accurate and include class abbreviations, professor, place of instruction, time, and description.

Testing Requirements

Testing Requirement	Traces to Specification Requirement
1. Add a class to cart	1, 2, 7, 13, 24, 27, 29, 32
2. Enroll in selected classes	1, 8, 13, 20, 24, 27, 30, 32
3. Delete class from cart	2, 8
4. Connect to software with internet connection	2
5. Connect to software without internet connection	2
6. Search for class by entering CSCE	2, 3, 15, 16, 18, 22, 24, 27, 28, 32
7. Enter correct username and password to login successfully to the student portal	2, 4, 5, 9, 10, 11, 27
8. Enter valid username and invalid password	2, 12, 27
9. Visit "Home" screen	2, 6, 14, 22, 24, 27, 32
10. Select class and filter by professor	2, 7, 13, 22, 24, 27, 32
11. Select class and filter by time	2, 7, 13, 22, 24, 27, 32
12. Select a filter	2, 17, 19, 22, 24, 25, 26, 27, 28, 31, 32
13. Change term using the term Drop down	2, 6, 21, 22, 23, 27

Test Cases

Test 1

PURPOSE

Test the addition of a class to a student's enrollment cart.

SETUP

Populate the database with a student with an empty cart.

INPUTS and EXPECTED OUTPUTS

Inputs	Outputs
Select required class "CSCE 361" in the Browse UI. Click on a professor "Hasan", to add class to cart.	CSCE 361 Class is added to cart in less than five seconds.

TESTING REQUIREMENTS COVERED

1

Test 2

PURPOSE

Test the enrollment of selected class(es).

SETUP

Populate the database with the desired classes and student data.

INPUTS and EXPECTED OUTPUTS

Inputs	Outputs
Click "Enroll" to add the classes in the cart to the student's enrollment.	The enrolled classes will update the database with the new information tied to the student account in less than five seconds.

TESTING REQUIREMENTS COVERED

2

Test 3

PURPOSE

Test the deletion of a class with a cart.

SETUP

Populate the database with a student that has at least one class in their cart.

INPUTS and EXPECTED OUTPUTS

Inputs	Outputs
Select the red dash remove button on the enrollment screen.	The selected class is removed from the cart.

TESTING REQUIREMENTS COVERED

3

Test 4

PURPOSE

Test the database connection with an active Internet connection.

SETUP

None Required.

INPUTS and EXPECTED OUTPUTS

Inputs	Outputs
Attempt to login to the software system without an internet connection.	No response.
Attempt to login to the software system with an active internet connection.	The user (with the correct credentials) is taken to the Home Screen with the proper student information.

TESTING REQUIREMENTS COVERED

4, 5

Test 5

PURPOSE

Test the keyword and code search functionality.

SETUP

Populate the database with test classes.

INPUTS and EXPECTED OUTPUTS

Inputs	Outputs
Input "Software" into the keyword search text field.	Software displays the "CSCE 361 Software Engineering" Course.
Input "CSCE 156" into the code search text field.	Software displays the "CSCE 156 Computer Science II" Course.

TESTING REQUIREMENTS COVERED

Test 6

PURPOSE

Test the functionality of logging in.

SETUP

Populate the database with a student.

INPUTS and EXPECTED OUTPUTS

Inputs	Outputs
Input "cchow" in the Username text field and "chowsRu13" for the Password text field.	The desired user is taken to the Home Screen with the proper student information linked to the account.
Input "cchow" in the Username text field and "fail" for the Password text field.	No response.

TESTING REQUIREMENTS COVERED

7, 8

Test 7

PURPOSE

Test the correctness of “Home” interface.

SETUP

Populate the database with a test student.

INPUTS and EXPECTED OUTPUTS

Inputs	Outputs
Click “Home” option in navigation bar.	“Home” screen is loaded and class list and term history information is loaded.
Stay on current page.	Screen and information are not changed.
Select a different term from the dropdown menu	Class list and term history is updated to reflect the relevant information for the selected term from the dropdown menu.

TESTING REQUIREMENTS COVERED

Test 8

PURPOSE

Test the ability to sort a class by professor or time of instruction after a course is selected.

SETUP

Populate the database with test classes and students.

INPUTS and EXPECTED OUTPUTS

Inputs	Outputs
Select class "CSCE 361". Select filter by time option. Select "1:00-2:00 M/W/F".	Sort the CSCE 361 course by professors who teach at "1:00-2:00 M/W/F".
Select class "CSCE 361". Select filter by professor "Hasan".	List of "CSCE 361" classes sorted by professor "Hasan" appears.

TESTING REQUIREMENTS COVERED

10, 11

Test 9

PURPOSE

Test the ability to filter classes by group criteria.

SETUP

Populate the database with test classes and students.

INPUTS and EXPECTED OUTPUTS

Inputs	Outputs
Select filters dropdown box. Select the “ACE 1” filter.	Browse screen populates with classes that are under the chosen “ACE 1” filter.
Filter not selected.	Classes do not change.

TESTING REQUIREMENTS COVERED

Test 10

PURPOSE

Test the ability to change term and update information accordingly.

SETUP

Populate the database with a test student.

INPUTS and EXPECTED OUTPUTS

Inputs	Outputs
Select the change term dropdown box. Select "Fa17".	Interface is populated with information relevant to the selected "Fa17" term.
Do not select new term.	Information is not updated.

TESTING REQUIREMENTS COVERED

Testing Report

Test Cases	Pass/Fail + Description
Test Case 1	FAIL: Users enrollment for the term is not updated.
Test Case 2	FAIL: The database does not update and the displayed information does not change.
Test Case 3	FAIL: The cart does not update.
Test Case 4	PASS
Test Case 5	FAIL: Keyword search warrants no response.
Test Case 6	PASS
Test Case 7	PASS
Test Case 8	FAIL: Filter by professor is not responsive when CSCE 361 is selected.
Test Case 9	FAIL: No response when "ACE 1" selected.
Test Case 10	PASS