

# Test Plan

Student Multi-Tool

---

10.27.2021

## **Team Marvel**

Albert Toscano

Audrey Brio

Bradley Nickle

Joseph Cutri

Michael Kriesel (Team Leader)

# Version History

## Date

10/18/2021

## Changes

Initial Draft

## Approval Date

# Testing Schedule

Feature	Core/Application	Start Test	End Test	Passed
Network Communication	Core	2/1/2022	2/5/2022	n/a
Data Store Access	Core	2/1/2022	2/5/2022	n/a
Logging/Archiving	Core	2/1/2022	2/5/2022	n/a
Registration	Core	2/15/2022	2/19/2022	n/a
Login	Core	2/15/2022	2/19/2022	n/a
Logout	Core	2/15/2022	2/19/2022	n/a
Error Handling	Core	2/15/2022	2/19/2022	n/a
Regression Testing (everything done so far)	Core	2/16/2022	2/19/2022	n/a
User Management	Core	3/1/2022	3/5/2022	n/a
User Access Control	Core	3/1/2022	3/5/2022	n/a
Usage Analysis Dashboard	Core	3/1/2022	3/5/2022	n/a
Core Testing	Core	3/2/2022	3/5/2022	n/a
Schedule of Classes	Application	3/15/2022	3/19/2022	n/a
Messaging	Application	3/15/2022	3/19/2022	n/a

Student Discounts	Application	3/15/2022	3/19/2022	n/a
Regression Testing (everything done so far)	Both	3/16/2022	3/19/2022	n/a
Schedule Builder	Application	3/29/2022	4/2/2022	n/a
Schedule Comparison	Application	3/29/2022	4/2/2022	n/a
Course Review	Application	3/29/2022	4/2/2022	n/a
Regression Testing (everything done so far)	Both	3/30/2022	4/2/2022	n/a
Matching	Application	4/12/2022	4/16/2022	n/a
Career Opportunities	Application	4/12/2022	4/16/2022	n/a
Regression Testing (everything done so far)	Both	4/13/2022	4/16/2022	n/a
Aid Eligibility	Application	4/23/2022	4/27/2022	n/a
Application Testing	Application	4/24/2022	4/27/2022	n/a
All System Test	Both	2/25/2022	4/27/2022	n/a

TASK TITLE	LEAD	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8
		1/25 - 2/5	2/8 - 2/19	2/22 - 3/5	3/8 - 3/19	3/22 - 4/2	4/5 - 4/16	4/19 - 4/27	4/30 - 5/4
Testing									
Network Communication	Michael								
Data Store Access	Joseph								
Logging/Archiving	Albert								
Registration	Albert								
Login	Audrey								
Logout	Brad								
Error Handling	Joseph								
User Management	Audrey								
User Access Control	Michael								
Usage Analysis Dashboard	Brad								
Schedule of Classes	Michael								
Schedule Builder	Audrey								
Course Reviews	Albert								
Schedule Comparison	Brad								
Messaging	Brad								
Matching	Michael								
Student Discounts	Audrey								
Career Opportunities	Albert								
Aid Eligibility	Joseph								
Deployment									
Set up Web Server	Michael								
Set up DataBase	Audrey								
Security Configuration	Albert								
Deployment	Brad								

# Progression and Regression Testing

This section discusses when progression and regression testing will be done on our system

- Progression Testing
  - Test created as features are implemented
- Regression Testing
  - As features are implemented
  - Dedicated testing sprint near the end of the term whenever that is
  - After deployment, continuously

## Testing Environment

Before deployment, our testing environment will consist of our personal computers. We will try to simulate the network detailed in our network diagram:

- For our datastore, we will use SQL Server 2019 and MariaDB instead of AWS Relational Database Service (AWS RDS).
- We will run our server software on our own devices instead of on AWS Elastic Compute Cloud (AWS EC2).
- We will run our client software locally in Google Chrome.
- We will be excluding the AWS Elastic Load Balancer (AWS ELB).

After deployment, our testing environment will shift to AWS. See our network diagram for more information.

Unit Testing will be performed on individual modules as they are created using Visual Studios. Inputs and Outputs will be validated

# Test Cases

The following sections heavily reference the “Functional and NonFunctional Requirements” section of the Business Requirements Document (BRD).

<https://github.com/mriheel/SeniorProject/blob/main/Milestone%201/CECS491A-BRD.pdf>

The success and failure cases are listed in an exhaustive form, and will all be tested for if the test data and execution steps are followed to completion and verified properly.

## Collaborative Schedule Builder Test Cases

1. The user creates a new schedule
  - a. Requirements Tested
    - i. SB01, SB02, SB03, SB04, SB05, SB06, SB07, SB08
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. Time data in this document will be written in 24-hour format, regardless of what is shown to the user
  - d. Test Data
    - i. Input
      1. One user account
      2. The following schedule items (SI's):
        - a. SI A
          - i. Type: Course
          - ii. Title: “Software Development with Frameworks”
          - iii. Start time: 14:00
          - iv. End time: 15:15
          - v. Days: Monday, Wednesday
          - vi. Category: “CECS”, Course number: “475”
          - vii. Section number: “01”
        - b. SI B
          - i. Type: Commute
          - ii. Title: “Automatic Commute: to CECS475”
          - iii. Start time: 13:00
          - iv. End time: 13:30
          - v. Days: Monday, Wednesday

c. SI C

- i. Type: Commute
- ii. Title: "Automatic Commute: CECS475"
- iii. Start time: 15:30
- iv. End time: 16:00
- v. Days: Monday, Wednesday

d. SI D

- i. Type: Misc
- ii. Title: "Test: Studying"
- iii. Start time: 12:00
- iv. End time: 13:00
- v. Days: Monday, Tuesday, Wednesday, Friday, Saturday, Sunday

ii. Output

- 1. True - all information is on schedule

e. Test Execution Steps

- i. User is on home page
- ii. User selects Build Schedules
- iii. The user clicks the new schedule button
- iv. The user adds SI A, SI B, SI C, and SI D to the schedule
- v. The user names the schedule
- vi. The user exits the edit view/page
- vii. The user reopens the schedule
- viii. The user examines each schedule item for consistency

f. Success Cases

- i. Upon reopening the schedule, all schedule items are the same as they were when first added

g. Failure Cases

- i. A schedule that the user did not select is opened for editing
- ii. Any schedule item cannot be added to the schedule
- iii. Upon reopening the schedule, any schedule item is missing, or has incorrect data

2. The user views a schedule

a. Requirements Tested

- i. SB01, SB02

b. Dependencies

- i. User must be logged in

- c. Assumptions
  - i. User is connected to internet
  - ii. User does not lose connection to internet
- d. Test Data
  - i. Input
    - 1. One user account that owns a copy of both schedule A and schedule B
  - ii. Output
    - 1. Schedule A displayed
    - 2. Schedule B displayed
- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects builds schedules
  - iii. The user opens schedule A for editing
  - iv. The user returns to the selection screen
  - v. The user opens schedule B for editing
- f. Success Cases
  - i. Schedule A is opened first, then schedule B is opened
- g. Failure Cases
  - i. Schedule B is opened when schedule A is supposed to be opened, and vice versa
- 3. The user edits an existing schedule
  - a. Requirements Tested
    - i. SB01, SB02
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. The user should be able to create, update, and delete schedule items (referred to as SI's)
    - iv. The user should be able to create, update, and delete inactive hours
    - v. Schedule items can overlap with inactive hours
    - vi. Time data in this document will be written in 24-hour format, regardless of what is shown to the user
  - d. Test Data
    - i. Input



1. One user account that owns schedule A
2. Test update string: " - updated"
3. The following schedule items (SI's):
  - a. SI A
    - i. Type: Course
    - ii. Title: "Intro to Testing"
    - iii. Start time: 09:00
    - iv. End time: 10:15
    - v. Days: Monday, Wednesday
    - vi. Category: "CECS"
    - vii. Course number: "123"
    - viii. Section number: "01"
  - b. SI B
    - i. Type: Commute
    - ii. Title: "Automatic Commute: to Intro to Testing"
    - iii. Start time: 08:30
    - iv. End time: 08:45
    - v. Days: Monday, Wednesday
  - c. SI C
    - i. Type: Commute
    - ii. Title: "Automatic Commute: from Intro to Testing"
    - iii. Start time: 10:30
    - iv. End time: 10:45
    - v. Days: Monday, Wednesday
  - d. SI D
    - i. Type: Commute
    - ii. Title: "Manual Commute: Test Drive"
    - iii. Start time: 15:00
    - iv. End time: 16:00
    - v. Days: Tuesday
  - e. SI E
    - i. Type: Misc
    - ii. Title: "Test: Studying"
    - iii. Start time: 12:00
    - iv. End time: 13:00
    - v. Days: Monday, Tuesday, Wednesday, Friday,

Saturday, Sunday

ii. Output

1. Schedule A with the following changes

a. SI A

- i. Type: Course
- ii. Title: "Intro to Testing"
- iii. Start time: 08:30
- iv. End time: 09:45
- v. Days: Wednesday, Friday
- vi. Category: "MATH"
- vii. Course number: "123"
- viii. Section number: "02"

b. SI B

- i. Type: Commute
- ii. Title: "Automatic Commute: to Intro to Testing"
- iii. Start time: 08:00
- iv. End time: 08:15
- v. Days: Wednesday, Friday

c. SI C

- i. Type: Commute
- ii. Title: "Automatic Commute: from Intro to Testing"
- iii. Start time: 10:00
- iv. End time: 10:15
- v. Days: Wednesday, Friday

d. SI D

- i. Type: Commute
- ii. Title: "Manual Commute: Test Drive"
- iii. Start time: 14:30
- iv. End time: 15:30
- v. Days: Thursday

e. SI E

- i. Type: Misc
- ii. Title: "Test: Studying"
- iii. Start time: 11:30
- iv. End time: 12:30
- v. Days: Monday, Wednesday, Thursday, Friday,

Saturday, Sunday

- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects builds schedules
  - iii. User opens schedule A for editing
  - iv. Delete all of schedule A's original SI's
  - v. Add SI A, SI B, SI C, SI D and SI E to the schedule
  - vi. For SI A, update the category to "MATH" and update the section number to "02"
  - vii. For all SIs:
    - 1. Append the test update string to the title
    - 2. Shift the start and end time forward by 1 hour
    - 3. Shift the start and end time back by 0.5 hours
  - viii. For SI A, SI B, SI C:
    - 1. Remove Monday from the list of days
    - 2. Add Friday to the list of days
  - ix. For SI D and SI E:
    - 1. Add Thursday to the list of days
    - 2. Remove Tuesday from the list of days
- f. Success Cases
  - i. All SI data match the specified output data
- g. Failure Cases
  - i. An error is raised, preventing any SI from being added to the schedule
  - ii. Any SI is added to the schedule with erroneous data
- 4. Multiple users edit the same schedules at the same time
  - a. Requirements Tested
    - i. SB01, SB02, SB11, SB12, SB13
  - b. Dependencies
    - i. There must be at least two users to perform this test
    - ii. All users must be logged in
    - iii. All users used in this test must have permission to edit the same schedules, regardless of who owns the schedules
  - c. Assumptions
    - i. Users are connected to internet
    - ii. Users do not lose connection to internet
    - iii. Schedules used in this test will be owned by different users, although there can be multiple schedules owned by the same user

- iv. Time data in this document will be written in 24-hour format, regardless of what is shown to the user
- d. Test Data
  - i. Input
    - 1. Two users, who will be referred to as "Alice" and "Bob"
    - 2. A copy of schedule A, which Alice owns and Bob can edit
    - 3. A copy of schedule B, which Bob owns and Alice can edit
    - 4. The following schedule items (SIs):
      - a. SI A
        - i. Type: Misc
        - ii. Title: "Test: Studying"
        - iii. Start time: 12:00
        - iv. End time: 13:00
        - v. Days: Tuesday
      - b. SI B
        - i. Type: Misc
        - ii. Title: "Test: Practicing"
        - iii. Start time: 14:00
        - iv. End time: 15:30
        - v. Days: Thursday
  - ii. Output
    - 1. True - Correct information is changed and visible to both users
- e. Test Execution Steps
  - i. Alice and Bob are on the home page
  - ii. Alice and Bob select builds schedules
  - iii. Alice and Bob each select both schedules to be edited
  - iv. Alice adds SI A to schedule B
  - v. Bob adds SI B to schedule A
  - vi. Alice updates the title of SI B
  - vii. Bob moves the time of SI A forward by 1 hour
  - viii. Alice deletes an SI (not SI A or SI B) from schedule B
  - ix. Bob deletes an SI (not SI A or SI B) from schedule A
  - x. Alice and Bob exit the schedule builder
  - xi. Alice and Bob reopen schedule A to check that their changes have been made
  - xii. Alice and Bob reopen schedule B to check that their changes have been made

- f. Success Cases
  - i. Upon reopening the schedules in the final step, all changes made during the test should still be visible to Alice and Bob
- g. Failure Cases
  - i. Either schedule fails to open or cannot be viewed by Alice and Bob simultaneously
  - ii. Alice cannot see that Bob is also editing the schedules, or vice versa
  - iii. Alice or Bob is prevented from making any change to either schedule
  - iv. Upon reopening the schedules in the final step, Alice cannot see a change that Bob has made to the schedule, or vice versa
- 5. Give Permissions
  - a. Requirements Tested
    - i. SB09
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. Test 1
      - 1. Input
        - a. User Alice
        - b. User Bob
        - c. Permission - edit
        - d. Schedule A
      - 2. Output
        - a. True - Bob can edit schedule A
    - ii. Test 2
      - 1. Input
        - a. User Alice
        - b. User Bob
        - c. Permission - view
        - d. Schedule B
      - 2. Output
        - a. True - Bob can view schedule B
  - e. Test Execution Steps
    - i. Test 1

1. User is on home page
      2. User selects builds schedules
      3. Alice selects schedule A
      4. Alice gives permissions to Bob to edit schedule A
    - ii. Test 2
      1. User is on home page
      2. User selects builds schedules
      3. Alice selects schedule A
      4. Alice gives permissions to Bob to view schedule B
  - f. Success Cases
    - i. Bob can edit and view schedule A
    - ii. Bob can view schedule B
  - g. Failure Cases
    - i. Bob can not edit or view schedule A
    - ii. Bob can not view Schedule B
6. Remove Permissions
- a. Requirements Tested
    - i. SB10
  - b. Dependencies
    - i. User must be logged in
    - ii. User must have granted permissions to another user
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. User Alice
    - ii. User Bob
    - iii. Schedule A
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User selects builds schedules
    - iii. Alice selects schedule A
    - iv. Alice removes permissions for Bob on schedule A
  - f. Success Cases
    - i. Bob can no longer view or edit schedule A
  - g. Failure Cases
    - i. Bob can still view or edit schedule A



## Schedule Comparison Test Cases

1. The user selects two to five schedules to be compared
  - a. Requirements Tested
    - i. SC01, SC03
  - b. Dependencies
    - i. User must be logged in
    - ii. User needs access to at least six schedules to fully perform this test
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. Input
      1. Schedules A, B, C, D, and E
      2. A duplicate of schedule A
    - ii. Output
      1. True or false, depending on whether or not each comparison is executed
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User selects compare schedules
    - iii. For each integer n between 0 and 6(inclusive):
      1. The user selects n schedule(s)
      2. The user clicks the compare button
      3. The comparison is either executed or not
  - f. Success Cases
    - i. The user selects 2, 3, 4, or 5 schedules and the comparison is executed
    - ii. The user selects 0, 1, or 6 schedule(s) and the comparison is not executed
  - g. Failure Cases
    - i. The user only selects 0, 1, or 6 schedule(s) and the comparison is executed
    - ii. The user tries to select 2, 3, 4, or 5 schedules and the comparison is not executed
2. Two to five schedules are compared
  - a. Requirements Tested
    - i. SC01, SC03, SC04, SC05, SC06



- b. Dependencies
  - i. User must be logged in
  - ii. User needs access to at least 5 schedules to fully perform this test
- c. Assumptions
  - i. User is connected to internet
  - ii. User does not lose connection to internet
  - iii. The number of schedules compared will only ever be between 2 and 5 (inclusive)
  - iv. The user owns or has permission to edit schedules A, B, C, D, and E
  - v. The scope of this test is only to test that the comparison logic works correctly
- d. Test Data
  - i. Input
    - 1. The following sets of schedules: {A, B}, {A, B, C}, {A, B, C, D}, and {A, B, C, D, E}
    - 2. The following lists of schedules: (A, A), (B, B, B), (C, C, C, C), and (D, D, D, D, D)
  - ii. Output
    - 1. Compared schedules for {A,B}, {A,B,C}, {A,B,C,D}, {A,B,C,D,E}
    - 2. Compared schedules for (A, A), (B, B, B), (C, C, C, C), and (D, D, D, D, D)
- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects compare schedules
  - iii. Each set or list of schedules is compared
  - iv. Each result is checked for correct free time
- f. Success Cases
  - i. The resultant free time never overlaps with schedule items and inactive hours from any of the compared schedules
- g. Failure Cases
  - i. The resultant free time overlaps with any schedule item or inactive hours

## Course Reviews Test Cases

1. Add a new review
  - a. Requirements Tested
    - i. CR02, CR03
  - b. Dependencies
    - i. User must be logged in
    - ii. The schedule of classes for the school is up to date
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. The school is the one the user listed in their account
    - iv. The classes and professors being entered are accurate
  - d. Test Data
    - i. Review A:
      1. Input:
        - a. Major: Computer Science;
        - b. Course: CECS228;
        - c. Term: Fall 2021,
        - d. Professor: Mehrnia,
        - e. Comments: "This was a fun class";
        - f. Difficulty: 3/5
      2. Output: Review published successfully. Overall difficulty = 3/5
    - ii. Review B:
      1. Input:
        - a. Major: Computer Science;
        - b. Course: CECS228;
        - c. term: Fall 2021,
        - d. professor: Mehrnia,
        - e. comments: "This class was hard";
        - f. Difficulty: 5/5
      2. Output: Review published successfully. Overall difficulty = 4/5
    - iii. Review C:
      1. Input:
        - a. Major: Computer Science;
        - b. Course: CECS228;
        - c. term: Fall 2021,
        - d. professor: Mehrnia,

- e. comments: "This class was easy"
  - 2. Output: Review failed to publish, no difficulty
- e. Test Execution Steps
  - i. The user starts on the page to add a new review
  - ii. The user selects the following information that comes from the corresponding test data above:
    - 1. User selects a major
    - 2. The user searches for and selects the class
    - 3. The user selects the professor
    - 4. The user adds comments about the class & professor
    - 5. The user inputs the desired difficulty rating
    - 6. The user publishes the review
  - iii. This is repeated for all reviews
- f. Success Cases
  - i. The user successfully publishes a new review, and it is able to be viewed from that courses page
  - ii. The user successfully publishes a new review, and it is able to be viewed from that professors page
  - iii. The overall difficulty average is adjusted appropriately according to the output listed in the review
- g. Failure Cases
  - i. The user doesn't input all required data
  - ii. The class entered does not exist
- 2. Add suggested class pairing
  - a. Requirements Tested
    - i. CR04
  - b. Dependencies
    - i. The user must be logged in
    - ii. The schedule of classes for the school is up to date
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. The school is the one listed in the user's account information
  - d. Test Data
    - i. Pairing 1 (2 classes):
      - 1. Input:
        - a. Major: Computer Science

- b. CECS 326 - Xu - Spring 2021
    - c. CECS 327 - Luti - Fall 2021
    - d. Comments: 327 Builds off of 326
  - ii. Pairing 2 (3 classes):
    - 1. Input:
      - a. Major: Computer Science
      - b. CECS 323 - Brown - Fall 2020
      - c. CECS 328 - Sharifian - Fall 2020
      - d. CECS343 - Hoffman - Spring 2021
      - e. Comments: 323 and 328 are useful for 343
  - iii. Pairing 3 (4 classes):
    - 1. Input:
      - a. Major: Computer Science
      - b. CECS 451 - Moon - Fall 2020
      - c. CECS 465 - Whenlu - Spring 2021
      - d. CECS 453 - Nguyen - Spring 2021
      - e. CECS 478 - Giacalone - Fall 2020
      - f. Comments: These electives were all very interesting
  - iv. Pairing 4 (5 classes):
    - 1. Input:
      - a. Major: Computer Science
      - b. CECS 174 - Nachawati - Fall 2018
      - c. CECS 274 - Ghazarian - Spring 2019
      - d. CECS 277 - Cleary - Fall 2019
      - e. CECS 282 - Gold - Fall 2019
      - f. CECS 228 - Mehrnia - Spring 2020
      - g. Comments: These classes are about the fundamentals of programming
  - v. Pairing 5 (Only one class):
    - 1. Input:
      - a. Major: Computer Science
      - b. CECS 328 - Goldstein - Fall 2020
      - c. Comments: This class was very difficult
    - 2. Output:
      - a. False - Review did not post, only one class
  - vi. Pairing 6 (6 classes):
    - 1. Input:

- a. Major: Computer Science
  - b. CECS 323 - Monge - Fall 2020
  - c. CECS 225 - Chelian - Spring 2020
  - d. CECS 282 - Gold - Spring 2020
  - e. CECS 326 - Xu - Fall 2020
  - f. MATH 122 - Johnson - Fall 2021
  - g. PHYS 151 - Klaehn - Fall 2019
  - h. Comments: These classes are not related at all
- 2. Output:
  - a. False - Review did not post, too many classes
- vii. Pairing 7 (same 3 classes):
  - 1. Input:
    - a. Major: Computer Science
    - b. CECS 323 - Brown - Fall 2020
    - c. CECS 328 - Sharifian - Fall 2020
    - d. CECS343 - Hoffman - Spring 2021
    - e. Comments: Taking 343 after 323 was very hard
  - 2. Output:
    - a. Linked to Pairing 2
- viii. Pairing 8 (same 4 classes):
  - 1. Input:
    - a. Major: Computer Science
    - b. CECS 451 - Moon - Fall 2020
    - c. CECS 465 - Whenlu - Spring 2021
    - d. CECS 453 - Nguyen - Spring 2021
    - e. CECS 478 - Giacalone - Fall 2020
    - f. Comments: These classes were fun
  - 2. Output:
    - a. Linked to Pairing 3
- ix. Pairing 9 (3 classes):
  - 1. Input:
    - a. Major: Computer Science
    - b. CECS 323 - Brown - Fall 2020
    - c. CECS 328 - Sharifian - Fall 2020
    - d. CECS343 - Hoffman - Spring 2021
    - e. Comments:
  - 2. Output:

- a. False - Review did not post, missing comment
- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects Course Review
  - iii. The user navigates to the 'add suggested pairing' page
  - iv. The user selects the major (for all tests, Computer Science)
  - v. The user searches for and adds all classes for this test
  - vi. The user adds the comments
  - vii. The user publishes the suggested pairing
  - viii. Steps i-v are repeated for all test data
- f. Success Cases
  - i. The suggested pairing is published
  - ii. A suggested pairing that already exists is linked to the existing one
- g. Failure Case
  - i. The user only selected one class
  - ii. The user selected 6 or more classes
  - iii. The user is missing information
  - iv. One or more classes do not exist
- 3. Search for a review
  - a. Requirements Tested
    - i. CR01, CR06
  - b. Dependencies
    - i. User is logged in
    - ii. The schedule of classes for the school is up to date
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. The school is the one in the user profile
    - iv. All of the test data listed in Course Reviews Test Case 1 is already loaded into the database
  - d. Test Data
    - i. Test 1:
      - 1. Input: CECS 228
      - 2. Output: True - This class has a review
    - ii. Test 2:
      - 1. Input: CECS 328
      - 2. Output: False - This class has no review

- iii. Test 3:
  - 1. Input: CECS 333
  - 2. Output: False - This class does not exist
- iv. Test 4:
  - 1. Input: Mehrnia
  - 2. Output: True - This professor has a review
- v. Test 5:
  - 1. Input: Goldstein
  - 2. Output: False - This professor has no review
- vi. Test 6:
  - 1. Input: Kriesel
  - 2. Output: False - This professor does not exist
- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects Course Review
  - iii. User goes to search bar
  - iv. The user enters major
  - v. The user searches for the class or professor, determined by test case listed above
  - vi. Validate that the desired class could be found
  - vii. Repeat for all testing data
- f. Success Cases
  - i. The desired class is found
  - ii. The desired professor is found
- g. Failure Cases
  - i. The class could not be found
  - ii. No reviews exist for the class
  - iii. The professor could not be found
  - iv. No reviews exist for the professor
- 4. Search for class pairings
  - a. Requirements Tested
    - i. CR04
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet

- iii. The database already contains class pairings from course review test case 2
  - d. Test Data
    - i. Test 1 - Search for major:
      - 1. Input: Computer science
      - 2. Output: All Test 2 Test data
    - ii. Test 2 - Search for a class:
      - 1. Input: CECS323
      - 2. Output: Test 2 Pairing 2
    - iii. Test 3 - Search for class name
      - 1. Input: Database
      - 2. Output: Test 2 Pairing 2
    - iv. Test 3 - Search for major:
      - 1. Input: Physics
      - 2. Output: No pairings found
    - v. Test 4 - Search for a class:
      - 1. Input: CECS 100
      - 2. Output: No pairings found
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User selects Course Review
    - iii. The user searches for the major or class based on test data listed above
    - iv. The user views the pairings
    - v. Verify the pairings are the ones desired
    - vi. Tests are repeated until all are done
  - f. Success Cases
    - i. The user finds desired class pairings
  - g. Failure Cases
    - i. No pairings are available
5. View Rating
- a. Requirements Tested
    - i. CR02, CR06, CR07
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to internet



- ii. User does not lose connection to internet
  - iii. School is same as listed in user settings
  - iv. The test data from Course Reviews test 1 (add a new review) is already loaded into the database
- d. Test Data
  - i. Test 1:
    - 1. Input: CECS 228
    - 2. Output:
      - a. Overall difficulty - 4/5
      - b. Comments: "This class was hard", "This was a fun class"
  - ii. Test 2:
    - 1. Input: Mehrnia
    - 2. Output:
      - a. Overall difficulty - 4/5
      - b. Comments: "This class was hard", "This was a fun class"
- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects Course Review
  - iii. The user navigates to the 'find a review' page
  - iv. The user enters their major
  - v. The user searches for the class
  - vi. The user selects the class
- f. Success Cases
  - i. The desired class is viewed with accurate information
  - ii. The desired professor is viewed with accurate information
- g. Failure Cases
  - i. The class could not be found
  - ii. The professor could not be found
  - iii. The information in the review is inaccurate or not up to date
- 6. View Suggested Class Pairings
  - a. Requirements Tested
    - i. CR04
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet

- iii. School is same as listed in setting
  - iv. The test data from test 2 is already loaded into the database
- d. Test Data
  - i. Test 1 - Search for major:
    - 1. Input: Computer science
    - 2. Output: All Test 2 Test data
  - ii. Test 2 - Search for a class:
    - 1. Input: CECS323
    - 2. Output: Test 2 Pairing 2
- e. Test Execution Steps
  - i. The user navigates to the 'find suggested pairing' page
  - ii. The user searches for and selects the desired pairing
  - iii. Verify the information
- f. Success Cases
  - i. The desired pairing is viewed with accurate information
- g. Failure Cases
  - i. The pairing does not exist
  - ii. The information in the suggested pairing is inaccurate
- 7. Like/Dislike
  - a. Requirements Tested
    - i. CR05
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. The school is the one in the users profile
    - iv. School is same as listed in user settings
    - v. The test data from course review Course Review test case 1 (add a new review) is already loaded into the database
  - d. Test Data
    - i. Test 1:
      - 1. Input: Like review a
      - 2. Output: Review A has 1 like added
    - ii. Tset 2:
      - 1. Input: Dislike review b
      - 2. Output: Review B has 1 dislike added

- e. Test Execution Steps
  - i. The user searches for and selects a review
  - ii. The user likes or dislikes it (depending on the test)
  - iii. Verify the like/dislike counter matches the current test's output
- f. Success Cases
  - i. The class's like/dislike counter is adjusted appropriately
  - ii. A review with a higher ratio of likes to dislikes is more relevant
- g. Failure Cases
  - i. The class's like/dislike counter is not adjusted
  - ii. A review with a lower ratio of likes to dislikes is more relevant

## Matching Test Cases

1. Activity Profile - Correct Information
  - a. Requirements Tested
    - i. MA01, MA02
  - b. Dependencies
    - i. User must be logged in
    - ii. User is required to have at least one schedule created
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. Input
      1. Activities selected: studying, get food (on/off campus), hang out
      2. Schedule A (from test schedules)
    - ii. Output
      1. True - profile successfully created
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User selects matching profile
    - iii. User selects schedule A
    - iv. User selects create/edit activity profile
    - v. User selects studying, get food (on/off campus), hang out as activities
    - vi. User clicks save
  - f. Success Cases
    - i. User successfully created a matching profile for activities and the system can begin matching
    - ii. User receives no matches if a possible match does not exist
  - g. Failure Cases
    - i. User profile was not created
2. Activity Profile - Incorrect Activity Information
  - a. Requirements Tested
    - i. MA01 and MA02
  - b. Dependencies
    - i. User must be logged in
    - ii. User is required to have at least one schedule created
  - c. Assumptions
    - i. User is connected to internet

- ii. User does not lose connection to internet
  - d. Test Data
    - i. Test 1
      - 1. Input
        - a. Activities selected: studying, get food (on/off campus), hang out, exercising, go to event, other
        - b. Schedule A (from test schedules)
      - 2. Output
        - a. False - profile not successfully created
    - ii. Test 2
      - 1. Input
        - a. Activities selected: none selected
        - b. Schedule A (from test schedules)
      - 2. Output
        - a. False - profile not successfully created
  - e. Test Execution Steps
    - i. Test 1
      - 1. User is on home page
      - 2. User selects matching profile
      - 3. User selects schedule A
      - 4. User selects create/edit activity profile
      - 5. User selects studying, get food (on/off campus), hang out, exercising, go to event, other as activities
      - 6. User clicks save
    - ii. Test 2
      - 1. User is on home page
      - 2. User selects matching profile
      - 3. User selects schedule A
      - 4. User selects create/edit activity profile
      - 5. User does not select any activities
      - 6. User clicks save
  - f. Success Cases
    - i. User profile is not created
  - g. Failure Cases
    - i. User selects no activity or user selects more than 5 activities
    - ii. A profile was created
3. Tutoring Profile - Correct Information

- a. Requirements Tested
  - i. MA03 and MA04
- b. Dependencies
  - i. User must be logged in
  - ii. User is required to have at least one schedule created
- c. Assumptions
  - i. User is connected to internet
  - ii. User does not lose connection to internet
- d. Test Data
  - i. Test 1
    - 1. Input
      - a. Schedule B
      - b. Courses: CECS 328, MATH 122
      - c. Group/Individual Setting: Individual
      - d. Offers/Requires Tutoring: Requiring
    - 2. Output
      - a. True - tutoring profile successfully created
  - ii. Test 2
    - 1. Input
      - a. Schedule B
      - b. Courses: CECS 328, MATH 122
      - c. Group/Individual Setting: Group
      - d. Offers/Requires Tutoring: Requiring
    - 2. Output
      - a. True - tutoring profile successfully created
  - iii. Test 3
    - 1. Input
      - a. Schedule B
      - b. Courses: CECS 328, MATH 122
      - c. Group/Individual Setting: Individual
      - d. Offers/Requires Tutoring: Offering
    - 2. Output
      - a. True - tutoring profile successfully created
  - iv. Test 4
    - 1. Input
      - a. Schedule B
      - b. Courses: CECS 328, MATH 122

- c. Group/Individual Setting: Group
    - d. Offers/Requires Tutoring: Offering
  - 2. Output
    - a. True - tutoring profile successfully created
- e. Test Execution Steps
  - i. Test 1
    - 1. User is on home page
    - 2. User selects matching profile
    - 3. User selects schedule B
    - 4. User selects create/edit tutoring profile
    - 5. User creates matching profile for tutoring
      - a. Tutoring in courses CECS 328, MATH 122
      - b. User requires tutoring
      - c. User prefers individual tutoring
    - 6. User clicks save
  - ii. Test 2
    - 1. User is on home page
    - 2. User selects matching profile
    - 3. User selects schedule B
    - 4. User selects create/edit tutoring profile
    - 5. User creates matching profile for tutoring
      - a. Tutoring in courses CECS 328, MATH 122
      - b. User requires tutoring
      - c. User prefers group tutoring
    - 6. User clicks save
  - iii. Test 3
    - 1. User is on home page
    - 2. User selects matching profile
    - 3. User selects schedule B
    - 4. User selects create/edit tutoring profile
    - 5. User creates matching profile for tutoring
      - a. Tutoring in courses CECS 328, MATH 122
      - b. User offers tutoring
      - c. User prefers individual tutoring
    - 6. User clicks save
  - iv. Test 4
    - 1. User is on home page

2. User selects matching profile
    3. User selects schedule B
    4. User selects create/edit tutoring profile
    5. User creates matching profile for tutoring
      - a. Tutoring in courses CECS 328, MATH 122
      - b. User offers tutoring
      - c. User prefers group tutoring
    6. User clicks save
  - f. Success Cases
    - i. User successfully created a matching profile for tutoring and the system can begin matching
    - ii. A user can receive no matches if there does not exist another user who meets a match criteria
  - g. Failure Cases
    - i. Profile not created
4. Tutoring Profile - Incorrect Course Information
  - a. Requirements Tested
    - i. MA03 and MA04
  - b. Dependencies
    - i. User must be logged in
    - ii. User is required to have at least one schedule created
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. Test 1
      1. Input
        - a. Schedule B
        - b. Courses: CECS 328, MATH 122, CECS 323, CECS 277, MATH 323, MATH 123, MUSC 468
        - c. Group/Individual Setting: Individual
        - d. Offers/Requires Tutoring: Requiring
      2. Output
        - a. False - tutoring profile not successfully created
    - ii. Test 2
      1. Input
        - a. Schedule B



- b. Courses: none entered
      - c. Group/Individual Setting: Group
      - d. Offers/Requires Tutoring: Offering
    - 2. Output
      - a. False - tutoring profile not successfully created
  - e. Test Execution Steps
    - i. Test 1
      - 1. User is on home page
      - 2. User selects matching profile
      - 3. User selects schedule B
      - 4. User selects create/edit tutoring profile
      - 5. User creates matching profile for tutoring
        - a. Tutoring in courses CECS 328, MATH 122, CECS 323, CECS 277, MATH 323, MATH 123, MUSC 468
        - b. User requires tutoring
        - c. User prefers individual tutoring
      - 6. User clicks save
    - ii. Test 2
      - 1. User is on home page
      - 2. User selects matching profile
      - 3. User selects schedule B
      - 4. User selects create/edit tutoring profile
      - 5. User creates matching profile for tutoring
        - a. User enters no course
        - b. User offers tutoring
        - c. User prefers group tutoring
      - 6. User clicks save
  - f. Success Cases
    - i. User profile is not created
  - g. Failure Cases
    - i. User does not enter any course or enters more than 6 courses
    - ii. User profile is created
5. Tutoring Profile - Incorrect Requiring/Offering Information
  - a. Requirements Tested
    - i. MA03 and MA04
  - b. Dependencies
    - i. User must be logged in

- ii. User is required to have at least one schedule created
- c. Assumptions
  - i. User is connected to internet
  - ii. User does not lose connection to internet
- d. Test Data
  - i. Input
    - 1. Schedule B
    - 2. Courses: CECS 328, MATH 122
    - 3. Group/Individual Setting: Individual
    - 4. Offers/Requires Tutoring: no selection
  - ii. Output
    - 1. False - tutoring profile not successfully created
- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects matching profile
  - iii. User selects schedule B
  - iv. User selects create/edit tutoring profile
  - v. User creates matching profile for tutoring
    - 1. Tutoring in courses CECS 328, MATH 122
    - 2. User does not select if they require/offer tutoring
    - 3. User prefers individual tutoring
  - vi. User clicks save
- f. Success Cases
  - i. User profile is not created
- g. Failure Cases
  - i. User does not select if they require or offer tutoring
  - ii. User profile is created
- 6. Tutoring Profile - Incorrect Group/Individual Preference Information
  - a. Requirements Tested
    - i. MA03 and MA04
  - b. Dependencies
    - i. User must be logged in
    - ii. User is required to have at least one schedule created
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
  - d. Test Data

- i. Input
    - 1. Schedule B
    - 2. Courses: CECS 328, MATH 122
    - 3. Group/Individual Setting: no selection
    - 4. Offers/Requires Tutoring: Requiring
  - ii. Output
    - 1. False - tutoring profile not successfully created
- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects matching profile
  - iii. User selects schedule B
  - iv. User selects create/edit tutoring profile
  - v. User creates matching profile for tutoring
    - 1. Tutoring in courses CECS 328, MATH 122
    - 2. User does not select their preference for group or individual tutoring
    - 3. User requires tutoring
  - vi. User clicks save
- f. Success Cases
  - i. User profile is not created
- g. Failure Cases
  - i. User does not select if they prefer group or individual tutoring
  - ii. User profile is created
- 7. Profile - No Schedule Information
  - a. Requirements Tested
    - i. MA05
  - b. Dependencies
    - i. User must be logged in
    - ii. User is required to have at least one schedule created
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. Input
      - 1. No schedule selected
    - ii. Output
      - 1. False - cannot create or edit a profile

- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects matching profile
  - iii. User does not select a schedule
  - iv. User selects create/edit activity or tutoring profile
- f. Success Cases
  - i. User is not taken to page to create or edit profile
- g. Failure Cases
  - i. User does not enter a schedule
  - ii. User cannot make or edit a profile
- 8. Select Match from Matching List
  - a. Requirements Tested
    - i. MA07
  - b. Dependencies
    - i. User must be logged in
    - ii. User needs to have matching profile created
    - iii. User needs to be opted in
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. User is matched with at least one other user who has not blocked them
  - d. Test Data
    - i. Input
      - 1. List of matches: Alice, Bob, Claire, Dale, Emma
    - ii. Output
      - 1. Reason for match - Studying, Exercising
      - 2. Schedule showcasing overlapping freetime - Schedule F
      - 3. Direct message option
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User opens matching list
    - iii. User clicks on match - Alice
  - f. Success Cases
    - i. Correct reasons for matching and overlapping freetime schedule shows up after selecting match
  - g. Failure Cases

- i. Incorrect information on why users were matched appears or incorrect information on overlapping freetime appears
- ii. No information on why users were matched appears or no information on overlapping freetime appears
- iii. Message button not displayed

## 9. Message Match

### a. Requirements Tested

- i. MA07

### b. Dependencies

- i. User must be logged in
- ii. User needs to have at least one match
- iii. User needs to be opted in

### c. Assumptions

- i. User is connected to internet
- ii. User does not lose connection to internet
- iii. User is matched with at least one other user who has not blocked them

### d. Test Data

- i. Input
  - 1. Match - Alice
  - 2. True - Message
- ii. Output
  - 1. Direct message with Alice

### e. Test Execution Steps

- i. User is on home page
- ii. User opens matching list
- iii. User clicks on match - Alice
- iv. User clicks on message

### f. Success Cases

- i. User is taken to direct message with Alice

### g. Failure Cases

- i. Clicking message button does not take user to direct message with Alice
- ii. User is taken to direct message with a different person other than Alice

## 10. Opt In

### a. Requirements Tested

- i. MA09
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. Input
      - 1. Yes = true
      - 2. No = false
    - ii. Output
      - 1. True - opted in
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User opens matching profile
    - iii. User clicks opt in
    - iv. User clicks yes or
    - v. User selects no instead
  - f. Success Cases
    - i. If user selected yes, user is successfully included in matching again
    - ii. If user selected no, nothing changes
  - g. Failure Cases
    - i. Yes/no buttons do nothing
    - ii. Information is not saved and the user still does not receive match information
    - iii. Even if click no, information is changed
11. Opt Out
- a. Requirements Tested
    - i. MA09
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. Input
      - 1. Yes = true

- 2. No = false
    - ii. Output
      - 1. True - opted out
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User opens matching profile
    - iii. User clicks opt out
    - iv. User clicks yes or
    - v. User selects no instead
  - f. Success Cases
    - i. If user selected yes, user is successfully not included in matching process
    - ii. If user selected no, nothing changes
  - g. Failure Cases
    - i. Yes/no buttons do nothing
    - ii. Information is not saved and the user still receives match information
    - iii. Even if click no, information is changed
12. Filter Matches
- a. Requirements Tested
    - i. MA08
  - b. Dependencies
    - i. User must be logged in
    - ii. User has received matches
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. User has at least two matches
  - d. Test Data
    - i. Test 1
      - 1. Input
        - a. List of matches - Alice, Bob, Claire, Dale, Emma, Francis, Greg, Hayley
        - b. Filter by date
      - 2. Output
        - a. List of matches - Bob, Claire, Alice, Hayley, Greg, Francis, Dale, Emma
    - ii. Test 2

1. Input
  - a. List of matches - Alice, Bob, Claire, Dale, Emma, Francis, Greg, Hayley
  - b. Filter by activity matched
2. Output
  - a. List of matches - Hayley, Emma, Dale, Greg, Alice, Francis, Claire, Bob
- e. Test Execution Steps
  - i. Test 1
    1. User is on home page
    2. User opens matching list
    3. User selects filter by date
    4. User selects save
  - ii. Test 2
    1. User is on home page
    2. User opens matching list
    3. User selects filter by activity matched
    4. User selects save
- f. Success Cases
  - i. Matches are displayed in new way, order determined by filter chosen
- g. Failure Cases
  - i. Filter buttons do nothing
  - ii. Matches are not displayed in new way
  - iii. Matches are filtered in different way then chosen by users

### 13. Matching Algorithm

- a. Requirements Tested
  - i. MA06
- b. Dependencies
  - i. User must be logged in
  - ii. User has received matches
- c. Assumptions
  - i. User is connected to internet
  - ii. User does not lose connection to internet
  - iii. Users have the same school listed in their settings
- d. Test Data
  - i. User Alice
  - ii. List of users to be matched with - Bob, Claire, Dale, Emma, Francis,



Greg, Hayley

- e. Test Execution Steps
  - i. Run the code
- f. Success Cases
  - i. Users are matched for correct reasons:
    - 1. Activities: Overlapping freetime and at least one activity in common
    - 2. Tutoring: Overlapping freetime and one user needs tutoring in a course that the other user offers tutoring in
- g. Failure Cases
  - i. Users are not matched for correct reasons
    - 1. Do not have anything selected criteria in common
    - 2. Do not have common overlapping free time

#### 14. Generate Matches

- a. Requirements Tested
  - i. MA10
- b. Dependencies
  - i. User must be logged in
- c. Assumptions
  - i. User is connected to internet
  - ii. User does not lose connection to internet
  - iii. All users on list have created a matching profile and are opted in for matching
- d. Test Data
  - i. Input
    - 1. User Alice
    - 2. True - select generate matches
  - ii. Output
    - 1. List of Matches - Bob, Claire, Dale, Emma, Francis
- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects matching list
  - iii. User selects generate matches
- f. Success Cases
  - i. User gets matched to other users, list of matches is generated
- g. Failure Cases
  - i. User does not get matched to other users

## 15. Display List Of Matches

- a. Requirements Tested
  - i. MA11
- b. Dependencies
  - i. User must be logged in
- c. Assumptions
  - i. User is connected to internet
  - ii. User does not lose connection to internet
  - iii. User has at least 2 matches
- d. Test Data
  - i. Input
    - 1. List of Matches: Alice, Bob, Claire, Dale, Emma
  - ii. Output
    - 1. List displayed in no more than 10 seconds
- e. Test Execution Steps
  - i. User is on home page
  - ii. User opens matching list
- f. Success Cases
  - i. List is displayed in less than 10 seconds
- g. Failure Cases
  - i. List took longer than 10 seconds to display

## 16. Display Information of Match when select a Match

- a. Requirements Tested
  - i. MA12
- b. Dependencies
  - i. User must be logged in
  - ii. User has received matches
- c. Assumptions
  - i. User is connected to internet
  - ii. User does not lose connection to internet
- d. Test Data
  - i. Input
    - 1. List of matches: Alice, Bob, Claire, Dale, Emma
  - ii. Output
    - 1. Following information displayed in less than 10 seconds
      - a. Reason for match - Studying, Exercising
      - b. Schedule showcasing overlapping freetime - Schedule F

- c. Direct message option
- e. Test Execution Steps
  - i. User is on home page
  - ii. User opens matching list
  - iii. User clicks on match - Alice
- f. Success Cases
  - i. Match information is displayed in less than 10 seconds
- g. Failure Cases
  - i. Match information takes more than 10 seconds to display
- 17. Match Processing
  - a. Requirements Tested
    - i. MA13
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. All users on list have created a matching profile and are opted in for matching
  - d. Test Data
    - i. Input
      - 1. List of 100 users - Alice, Bob, Claire, Dale, Emma, Francis, Greg, Hayley, Issac, James, Kate, Lucas, Mindy, Nicole, Owen, Paige, Quinn, Riley, Star, Trevor, Ursula, Vanni, Wyatt, Xavier, Yasmin, Zola, Adam, Bri, Chris, Dom, Erica, Frank, George, Henry, Isaiah, Jessica, Kit, Liam, Mikaela, Noelia, Olly, Patty, Rubi, Susie, Tristan, Violet, Zack, Peyton, Brooke, Rachel, Ross, Monica, Chandler, Joey, Phoebe, Arya, Sansa, Jake, Amy, Michael, Audrey, Brad, Joseph, Albert, Dylan, Stiles, Scott, Spencer, Hannah, Elle, Lizzy, Beks, Chester, Kira, Lydia, Melissa, Maddy, Makena, Shaelah, Sydney, Nathan, Paul, Andrew, Cade, Eddie, Elaine, Savannah, Caesar, Arnold, Chloe, Morgan, Blair, Kelly, Rafa, Billy, Rita, Zoey, Austin, David, Perla
    - ii. Output
      - 1. True - system processed more than 100 users in less than 60 seconds
  - e. Test Execution Steps

- i. Run the code
  - f. Success Cases
    - i. System process 100 users for matches in less than 60 seconds
  - g. Failure Cases
    - i. System takes longer than 60 seconds to process 100 users for matches
- 18. Update Profile
  - a. Requirements Tested
    - i. MA14
  - b. Dependencies
    - i. User must be logged in
    - ii. User has already created a profile, they are editing it
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. Test 1
      - 1. Input
        - a. Activities selected: exercising, get food (dining hall), hang out
        - b. Schedule A (from test schedules)
      - 2. Output
        - a. True - profile successfully updated in less than 1 second
    - ii. Test 2
      - 1. Input
        - a. Course: MATH 123, CECS 343
        - b. Prefer group/Individual tutoring: Individual
        - c. Offer/Requires Tutoring: Offers
        - d. Schedule B
      - 2. Output
        - a. True - profile successfully updated in less than 1 second
  - e. Test Execution Steps
    - i. Test 1
      - 1. User is on home page
      - 2. User opens matching profile
      - 3. User selects schedule A
      - 4. User selects create/edit activity profile

- 5. User edits activity profile by selecting new activities: exercising, get food (dining hall), hang out and deselecting previous activities
    - 6. User selects save
  - ii. Test 2
    - 1. User is on home page
    - 2. User selects matching profile
    - 3. User selects schedule B
    - 4. User selects create/edit tutoring profile
    - 5. User edits matching profile for tutoring
      - a. Tutoring in courses CECS 328, MATH 122
      - b. User requires tutoring
      - c. User prefers individual tutoring
    - 6. User clicks save
- f. Success Cases
  - i. Profiles are successfully updated to database in less than one second
- g. Failure Cases
  - i. Profiles take longer than one second to successfully update to the database

## 19. Selecting Generate Matches

- a. Requirements Tested
  - i. MA15
- b. Dependencies
  - i. User must be logged in
- c. Assumptions
  - i. User is connected to internet
  - ii. User does not lose connection to internet
  - iii. All users on list have created a matching profile and are opted in for matching
- d. Test Data
  - i. Input
    - 1. User Alice
    - 2. True - select generate matches
  - ii. Output
    - 1. True - Alice has not selected generate matches within the last hour
- e. Test Execution Steps

- i. User is on home page
- ii. User selects matching list
- iii. User selects generate matches

f. Success Cases

- i. User has not selected generate matches within the last hour

g. Failure Cases

- i. User has selected generate matches within the last hour

## Student Discounts Test Cases

### 1. Post Student Discounts

#### a. Requirements Tested

- i. DI01, DI06

#### b. Dependencies

- i. User must be logged in

#### c. Assumptions

- i. The user will post a valid student discount
- ii. The user has the required student discount info
- iii. User is connected to internet
- iv. User does not lose connection to internet

#### d. Test Data

##### i. Post Establishment Discount A:

##### 1. Input:

- a. Name: Office Depot
- b. Address: 123 Main st. Long Beach, CA
- c. Discount description: 0.20 per copy for students

##### 2. Output:

- a. Display message: Discount posted successfully
- b. If there is already a record of name and address in the database, discount will be appended to that record

##### ii. Post Establishment Discount B:

##### 1. Input:

- a. Name: Office Depot
- b. Address: "" (empty string)
- c. Discount description: 0.20 per copy for students

##### 2. Output:

- a. Display message: Discount failed to be posted. No address provided

##### iii. Post Establishment Discount C:

##### 1. Input:

- a. Name: "" (empty string)
- b. Address: 123 Main st. Long Beach, CA
- c. Discount description: 0.20 per copy for students

##### 2. Output:

- a. Display message: Discount failed to be posted. No name provided

- iv. Post Website Discount A:
  - 1. Input:
    - a. Web address: www.officedepot.com
    - b. Discount description: 0.20 per copy for students
  - 2. Output:
    - a. Display message: Discount posted successfully
- v. Post Website Discount B:
  - 1. Input:
    - a. Web address: "" (empty string)
    - b. Discount description: 0.20 per copy for students
  - 2. Output:
    - a. Display message: Discount failed to be posted. No web address not provided
- e. Test Execution Steps
  - i. From the user home page, the user must click on student discounts button
  - ii. User selects Post student Discounts
  - iii. User selects either post establishment or post website
  - iv. User enters the required info in the input fields
  - v. User clicks Post
- f. Success Cases
  - i. The student discount posted by the user is added to our database
  - ii. Users will be able to see their posted discounts in the search discount option
- g. Failure Cases
  - i. If the user does not enter input fields (name, address, and description for physical establishments or URL and description for websites), a popup error message will be displayed suggesting the user to enter the required field
  - ii. If the user enters wrong information about a establishment or website discount, the record will be added to the database, but it will be incorrect

## 2. Search Student Discounts

- a. Requirements Tested
  - i. Testing DI02 from BRD
  - ii. Testing DI04 from BRD
  - iii. Testing DI05 from BRD



- iv. Testing DI09 from BRD
- b. Dependencies
  - i. User must be logged in
  - ii. The database has records of student discounts
- c. Assumptions
  - i. User is connected to internet
  - ii. User does not lose connection to internet
  - iii. All records of student discounts are correct
- d. Test Data
  - i. Search establishment discount A:
    - 1. Input:
      - a. User selects: Office Depot from the list
      - b. Or User selects: Office Depot from the map
    - 2. Output:
      - a. Display message: 0.20 per copy for students
      - b. Get directions on the map
  - ii. Search establishment discount B:
    - 1. Input:
      - a. User selects: Office Depot from the list
      - b. Or User selects: Office Depot from the map
    - 2. Output:
      - a. Display message: "Error loading map for directions"
  - iii. Search websites discount A:
    - 1. Input:
      - a. User selects: www.officedepot.com from the list
    - 2. Output:
      - a. Display message: 0.20 per copy for students
      - b. Displays and Enables: "Go to website" button
      - c. Displays and Enables: "votes" buttons
  - iv. Search websites discount B:
    - 1. Input:
      - a. User selects: www.officedepot.com from the list
    - 2. Output:
      - a. Display message: "Error loading discount information"
- e. Test Execution Steps
  - i. From the user home page, the user must click on student discounts button

- ii. Select search student discounts
  - iii. User selects either search establishment or search website
  - iv. If the user searches for establishments, there will be a list of establishments and a map marking where they are.
  - v. If the user searches for websites, a list of websites will be displayed.
  - vi. If user clicks on a item in the lists or in the map (for establishments),
- f. Success Cases
  - i. A list of establishments or websites (and a map for establishments) is displayed.
  - ii. The items in the list of discounts will be displayed in order from most votes to least votes
- g. Failure Case
  - i. There are no records of student discounts in the database, therefore, no students discounts will be displayed
- 3. Vote for student discount
  - a. Requirements Tested
    - i. Testing DI03 from BRD
    - ii. Testing DI07 from BRD
    - iii. Testing DI08 from BRD
  - b. Dependencies
    - i. User must be logged in
    - ii. The database must store all discounts' votes info
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. User's vote is correct
  - d. Test Data
    - i. Establishment/website vote A:
      - 1. Input:
        - a. User clicks: Positive vote
      - 2. Output:
        - a. Display message: positive vote selected
        - b. Discount's positive votes increments by 1
    - ii. Establishment/website vote B:
      - 1. Input:
        - a. User clicks: Negative vote
      - 2. Output:

- a. Display message: Negative vote selected
      - b. Discount's negative votes increments by 1
    - iii.
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User selects student discounts
    - iii. User searches for a discount
    - iv. User selects a discount
    - v. User votes for or against the student discount
  - f. Success Cases
    - i. Votes from different users will be cumulative and they will be displayed on their respective student discounts
  - g. Failure Cases
    - i. User tries to vote for the same student discounts before the 30 days period
4. Map of establishments
- a. Requirements Tested
    - i. Testing DI04 from BRD
  - b. Dependencies
    - i. User must be logged in
    - ii. User must search for establishments discounts
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. Establishments addresses are accurate
  - d. Test Data
    - i. Establishments' address A:
      - 1. Input:
        - a. User clicks: Office Depot on map
      - 2. Output:
        - a. Display message: 0.20 per copy for students
        - b. Displays and Enables: "Get directions" button
        - c. Displays and Enables: "votes" buttons
    - ii. Establishments' address B:
      - 1. Input:
        - a. User clicks: Office Depot on map
      - 2. Output:

- a. Display message: "Error loading map"
- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects student discounts
  - iii. User selects search student discounts
  - iv. User selects search establishments
- f. Success Cases
  - i. Map of establishments is displayed on the right side of the web page
- g. Failure Cases
  - i. Map of establishments is not loaded
  - ii. Establishments are not displayed on the map
- 5. List and description of student discounts
  - a. Requirements Tested
    - i. Testing DI02 from BRD
  - b. Dependencies
    - i. User must be logged in
    - ii. User must search for establishments or websites discounts
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. Establishments and websites information is correct
  - d. Test Data
    - i. List of student discounts A:
      - 1. Input:
        - a. User clicks: Office Depot in the list or on the map of student discounts
      - 2. Output:
        - a. Display message: Office Depot  
123 Main st. Long Beach, CA  
0.20 per copy for students
    - ii. List of student discounts B:
      - 1. Input:
        - a. User clicks: Office Depot in the list or on the map of student discounts
      - 2. Output:
        - a. Display message: "Error loading discount information"

- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects student discounts
  - iii. User selects search student discounts
  - iv. User selects search establishments or websites
  - v. User selects a establishment or website
- f. Success Cases
  - i. List of student discounts is displayed and the description of each item in the list is accurate
- g. Failure Cases
  - i. List of student discounts is not displayed
  - ii. The description of student discounts is not accurate or does not match

## Aid Eligibility Test Cases

1. View Aid Eligibility Matches
  - a. Requirements Tested
    - i. Testing AE05 from the BRD
  - b. Dependencies
    - i. User must be logged in
    - ii. User must be connected to the internet
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to the internet
    - iii. User has a valid .edu email address
    - iv. User enters a valid email address and password
  - d. Test Data
    - i. Email of user A:
      1. Input:
        - a. User's email: firstname.lastname@student.csulb.edu
        - b. Password: valid password
      2. Output:
        - a. Displays Aid Eligibility page
    - ii. Email of user B:
      1. Input:
        - a. User's email: firstname.lastname@yahoo.com
        - b. Password: valid password
      2. Output:
        - a. Displays error message: not valid email address
  - e. Test Execution Steps
    - i. User on home page
    - ii. Open Aid Eligibility
    - iii. Enter Username
    - iv. Enter Password
    - v. Click on Authenticate button
  - f. Success Cases
    - i. User enters correct ID and Password information
  - g. Failure Cases
    - i. User enter wrong ID and/or Password information
2. Enter Student's Information for Aid Eligibility Matches
  - a. Requirements Tested

- i. Testing AE01 from the BRD
  - ii. Testing AE02 from the BRD
  - iii. Testing AE03 from the BRD
  - iv. Testing AE04 from the BRD
- b. Dependencies
  - i. User must be logged in
  - ii. User must be connected to the internet
- c. Assumptions
  - i. User is connected to the internet
  - ii. User does not lose connection to the internet
  - iii. User has valid outlook email (to prove they are registered)
- d. Test Data
  - i. User's A information:
    - 1. Input:
      - a. Major: Computer Science
      - b. Current GPA: 3.5
      - c. Current income: 20000
      - d. Veteran Status: Active
      - e. Current semester units enrolled in: 12
      - f. Level of Education/Number of degrees: Undergraduate
      - g. User's age: 23
      - h. Disability Status: No disability
      - i. User is a first generation college student: Yes
      - j. User is associated with other programs/organizations: Yes
      - k. Residency: In-state
    - 2. Output:
      - a. Displays: list of financial aid resources based on user's personal information
  - ii. User's A information:
    - 1. Input:
      - a. Major: Computer Science
      - b. Current GPA: " " (No input)
      - c. Current income: " " (No input)
      - d. Veteran Status: " " (No input)
      - e. Current semester units enrolled in: " " (No input)
      - f. Level of Education/Number of degrees: " " (No input)

- g. User's age: " " (No input)
- h. Disability Status: " " (No input)
- i. User is a first generation college student: " " (No input)
- j. User is associated with other programs/organizations: " " (No input)
- k. Residency: " " (No input)

2. Output:

- a. Displays: list of financial aid resources only where a requirement to qualify is having a major in Computer Science

e. Test execution steps

- i. User is on home page
- ii. User selects aid eligibility
- iii. Open Aid Eligibility
- iv. Enter Username
- v. Enter Password
- vi. Click on Authenticate button

f. Success Cases

- i. User Enters all information in the appropriate fields and nothing is left blank
- ii. After the user enters their required information they are shown their Aid Eligibility Matches

g. Failure Cases

- i. The user entered the wrong username or password
- ii. The user forgot to enter their username or password
- iii. Incorrect User information
- iv. The user isn't shown any aid, when aid items for them exist

3. Apply to Aid Eligibility Matches

a. Requirements Tested

- i. Testing AE06 from the BRD
- ii. Testing AE07 from the BRD
- iii. Testing AE08 from the BRD
- iv. Testing AE09 from the BRD
- v. Testing AE10 from the BRD

b. Dependencies

- i. User must be logged in
- ii. User must be connected to the internet



- c. Assumptions
    - i. User is connected to the internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. User ID/Name and Password for the account is valid
    - ii. Match A:
      - 1. Input:
        - a. User selects: So Cal Scholarship
      - 2. Output:
        - a. Displays So Cal Scholarship details
        - b. Displays and enables the Apply button to be redirected to the website where the user can apply
    - iii. Match B:
      - 1. Input:
        - a. User selects: So Cal Scholarship
      - 2. Output:
        - a. So Cal Scholarship information is not loaded
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User selects aid eligibility
    - iii. User clicks on matches after entering their information
    - iv. User can view their Aid Eligibility matches
    - v. User can sort their Aid Eligibility matches
    - vi. User clicks on a match that he wants to read more about or apply to.
  - f. Success Cases
    - i. Aid Eligibility matches have been populated for the user
    - ii. User can apply to a selected match
  - g. Failure Cases
    - i. No Aid Eligibility matches are visible
4. Aid Eligibility Filtering Results
- a. Requirements Tested
    - i. Testing AE06 from the BRD
    - ii. Testing AE07 from the BRD
    - iii. Testing AE08 from the BRD
    - iv. Testing AE09 from the BRD
    - v. Testing AE10 from the BRD
  - b. Dependencies

- i. User must be logged in
  - ii. User must be connected to internet
- c. Assumptions
  - i. User is connected to the internet
  - ii. User does not lose connection to the internet
- d. Test Data
  - i. Match List A:
    - 1. Input:
      - a. User selects: sort by largest scholarship amount
    - 2. Output:
      - a. List A will be sorted by largest scholarship amount to lowest scholarship amounts
      - b. List A will be displayed
  - ii. Match List B:
    - 1. Input:
      - a. User selects: sort by lowest scholarship amount
    - 2. Output:
      - a. List B will be sorted by lowest scholarship amount to largest scholarship amounts
      - b. List B will be displayed
  - iii. Match List C:
    - 1. Input:
      - a. User selects: sort by nearest scholarship due date
    - 2. Output:
      - a. List C will be sorted by nearest scholarship due date to farthest scholarship due date
      - b. List C will be displayed
  - iv. Match List D:
    - 1. Input:
      - a. User selects: sort by farthest scholarship due date
    - 2. Output:
      - a. List D will be sorted by farthest scholarship due date to nearest scholarship due date
      - b. List D will be displayed
- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects aid eligibility

- iii. User clicks on sorting their Aid matches in the increasing amount given
- iv. User clicks on sorting their Aid matches in the decreasing amount given
- v. User clicks on sorting by the decreasing due date of their Aid matches
- vi. User clicks on sorting by the increasing due date of their Aid matches
- vii. User clicks on sorting by special requirements (additional information needed)

f. Success Cases

- i. User is able to view the sorting of the increasing amount Aid matches
- ii. User is able to view the sorting of the decreasing amount Aid matches
- iii. User is able to view the sorting by decreasing due date of their Aid matches
- iv. User is able to view the sorting by increasing due date of their Aid matches
- v. User is able to view the sorting by special requirements (additional information needed)

g. Failure Cases

- i. User cannot view the proper sorting of the increasing amount of their Aid matches
- ii. User cannot view the proper sorting of the decreasing amount of their Aid matches
- iii. User cannot view the proper sorting of the decreasing due date of their Aid matches
- iv. User cannot view the proper sorting of the increasing due date of their Aid matches
- v. User is able to view the proper sorting by special requirements (additional information needed)

5. Aid Eligibility Removal of Expired Matches

a. Requirements Tested

- i. Testing AE06 from the BRD
- ii. Testing AE07 from the BRD
- iii. Testing AE08 from the BRD
- iv. Testing AE09 from the BRD
- v. Testing AE10 from the BRD

b. Dependencies

- i. User must be logged in

- ii. User must be connected to internet
- c. Assumptions
  - i. User is connected to the internet
  - ii. User does not lose connection to the internet
- d. Test Data
  - i. Once the due date of an Aid match has passed it will be deleted from the user's matches
  - ii. No matches that are past the due date should be visible
- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects aid eligibility
  - iii. User should not be able to see expired Aid matches
- f. Success Cases
  - i. User's expired Aid matches are automatically gone and are no longer viewed within their Aid matches
- g. Failure Cases
  - i. User still has expired Aid matches visible
- 6. Aid Eligibility Updates of Matches
  - a. Requirements Tested
    - i. Testing AE06 from the BRD
    - ii. Testing AE07 from the BRD
    - iii. Testing AE08 from the BRD
    - iv. Testing AE09 from the BRD
    - v. Testing AE10 from the BRD
  - b. Dependencies
    - i. User must be logged in
    - ii. User must be connected to internet
  - c. Assumptions
    - i. User is connected to the internet
    - ii. User does not lose connection to the internet
  - d. Test Data
    - i. There should never be no Aid Eligibility matches
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User selects aid eligibility
    - iii. Ensure that there is at least one Aid Eligibility match
  - f. Success Cases

- i. User should be able to view new matches on a monthly basis at least
  - g. Failure Cases
    - i. User does not see any new matches on a by monthly basis
- 7. Aid Eligibility information opens within 5 secs
  - a. Requirements Tested
    - i. Testing AE09 from the BRD
  - b. Dependencies
    - i. User must be logged in
    - ii. User must be connected to internet
  - c. Assumptions
    - i. User is connected to the internet
    - ii. User does not lose connection to the internet
  - d. Test Data
    - i. Aid Eligibility matches should open within 5 secs
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User selects aid eligibility
    - iii. Ensure that each page is loaded within a minimum of 5 secs
  - f. Success Cases
    - i. Every navigation page within the Aid Eligibility matches is loaded within a 5 sec time frame
  - g. Failure Cases
    - i. One or more pages take longer than 5 secs to load
- 8. Aid Eligibility populates to view within 10 secs
  - a. Requirements Tested
    - i. Testing AE10 from the BRD
  - b. Dependencies
    - i. User must be logged in
    - ii. User must be connected to internet
  - c. Assumptions
    - i. User is connected to the internet
    - ii. User does not lose connection to the internet
  - d. Test Data
    - i. Data on each page of the Aid Eligibility needs to populate within a 10 secs time frame
    - ii. The matches page should be able to populate within a 10 sec time frame

- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects aid eligibility
  - iii. Click on each page to ensure that the 10 sec time frame is met to populate each page
- f. Success Cases
  - i. Every page clicked is loaded and populated with the data in a 10 sec time frame
- g. Failure Cases
  - i. One or more pages takes longer than 10 secs to load and populate the data

## Career Opportunities Test Cases

1. Search for career opportunities
  - a. Requirements Tested
    - i. CO02, CO06, CO07, CO08, CO09, CO10
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. User provides correct personal and professional information
  - d. Test Data
    - i. Career opportunities List A:
      1. Input:
        - a. Student status: Freshman
        - b. Completed units: 23
        - c. Major: Computer Science
        - d. Certifications?: Yes
        - e. Certification title: AWS Certificate
        - f. Past internships?: No
        - g. Past research programs?: No
      2. Output:
        - a. Display list: Internship A  
Internship B  
Job A  
Research program A  
...
    - ii. Career opportunities List B:
      1. Input:
        - a. Student status: Freshman
        - b. Completed units: 23
        - c. Major: Computer Science
        - d. Certifications?: Yes
        - e. Certification title: AWS Certificate
        - f. Past internships?: No
        - g. Past research programs?: No
      2. Output:
        - a. Display message: Error loading career opportunities

- e. Test Execution Steps
    - i. User is on home page
    - ii. User selects career opportunities
    - iii. User enters personal and professional information
    - iv. User clicks on the match button
  - f. Success Cases
    - i. A list of career opportunities is displayed
  - g. Failure Cases
    - i. The list of career opportunities is not loaded
    - ii. Personal information was provided, but the search results were not related to it
2. Filter by
- a. Requirements Tested
    - i. CO01
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. The database contains career opportunities records
  - d. Test Data
    - i. Test A:
      - 1. Input:
        - a. Filter by: Internship
      - 2. Output:
        - a. Display list: Internship A  
Internship B  
Internship C  
Internship D  
Internship E  
Internship F  
...
    - ii. Test B:
      - 1. Input:
        - a. Filter by: Internship
      - 2. Output:
        - a. Display list: Internship A



Job A  
Internship B  
Job B  
Internship C  
Job C  
...

iii. TestC:

1. Input:

- a. Filter by: Internship

2. Output:

- a. Display message: Error, the list could not be loaded

e. Test Execution Steps

- i. User is on home page
- ii. User selects career opportunities
- iii. User enters personal and professional information the first, and information is saved
- iv. User clicks on the match button
- v. User selects the type of filter to apply on the search (by job title, application type, salary, etc)

f. Success Cases

- i. The search results are narrowed and the list of career opportunities is displayed

g. Failure Case

- i. The search results are not affected by the filter and they are displayed without any change

3. Sort by

a. Requirements Tested

- i. CO03, CO08

b. Dependencies

- i. User must be logged in

c. Assumptions

- i. User is connected to internet
- ii. User does not lose connection to internet
- iii. The database contains career opportunities records

d. Test Data

i. Test A:

1. Input:

- a. Sort by: Internship
    - 2. Output:
      - a. Display list: Internship A
        - Internship B
        - Internship C
        - Job A
        - Job B
        - Job C
        - ...
  - ii. Test B:
    - 1. Input:
      - a. Sort by: Internship
    - 2. Output:
      - a. Display list: Internship A
        - Job A
        - Internship B
        - Job B
        - Internship C
        - Job C
        - ...
  - iii. Test C:
    - 1. Input:
      - a. Sort by: Internship
    - 2. Output:
      - a. Display message: Error, the list could not be loaded
- e. Test Execution Steps
- i. User is on home page
  - ii. User selects career opportunities
  - iii. User enters personal and professional information
  - iv. User clicks on the match button
  - v. User clicks on sort by (Date posted, Job title, Type of application, etc.) buttons
- f. Success Cases
- i. Search results will be sorted by selected condition and displayed in the condition's order
- g. Failure Case
- i. The search results are not affected by the sort by functionality and

they are displayed without any change

#### 4. View Career Opportunities

##### a. Requirements Tested

i. CO01

##### b. Dependencies

i. User must be logged in

##### c. Assumptions

i. User is connected to internet

ii. User does not lose connection to internet

iii. Career opportunities' information is correct

##### d. Test Data

i. Test A - Internship A:

1. Input:

a. Select: Space X Internship

2. Output:

a. Display description: Space X Internship

Internship description

ii. Test B - Internship A:

1. Input:

a. Select: Space X Internship

2. Output:

a. Display message: Error, information could not be loaded

##### e. Test Execution Steps

i. User is on home page

ii. User selects career opportunities

iii. User enters personal and professional information

iv. User clicks on the match button

v. User selects a career opportunity

##### f. Success Cases

i. Career opportunity is selected and description of each item in the list is accurate

ii. Display and enable the "Go to website to apply" button

##### g. Failure Cases

i. The description of career opportunities is not accurate or does not match

#### 5. Redirect Users to Site Website

- a. Requirements Tested
  - i. CO04
- b. Dependencies
  - i. User must be logged in
- c. Assumptions
  - i. User is connected to internet
  - ii. User does not lose connection to internet
  - iii. User enters correct information
- d. Test Data
  - i. Input
    - 1. Select apply - True
  - ii. Output
    - 1. User is redirected to new site
- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects career opportunities
  - iii. User enters personal and professional information
  - iv. User clicks on the match button
  - v. User selects on Career opportunity - Boeing
  - vi. User selects on apply
- f. Success Cases
  - i. User is redirected to Boeing page to apply
- g. Failure Cases
  - i. User is not redirected to new page
  - ii. User is redirected to incorrect page
- 6. Hide Career Opportunities
  - a. Requirements Tested
    - i. CO05
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. User enters correct information
  - d. Test Data
    - i. Input
      - 1. Select Hide - true

- ii. Output
    - 1. Career opportunity is hidden from users view
- e. Test Execution Steps
  - i. User is on home page
  - ii. User selects career opportunities
  - iii. User enters personal and professional information
  - iv. User clicks on the match button
  - v. User selects on Career opportunity - Boeing
  - vi. User selects on hide opportunity
- f. Success Cases
  - i. Boeing opportunity is hidden from users view
- g. Failure Cases
  - i. User is still able to see the Boeing job listing
- 7. Career Opportunities No Longer Available
  - a. Requirements Tested
    - i. CO11
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. Career opportunities' information is correct
  - d. Test Data
    - i. Input:
      - 1. Select: Space X Internship
    - ii. Output:
      - 1. Display description: Internship is past due
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User selects career opportunities
    - iii. User enters personal and professional information
    - iv. User clicks on the match button
    - v. User selects a career opportunity
  - f. Success Cases
    - i. Career opportunity is selected and it is noted that that this posting is past due or not available anymore
    - ii. User cannot use the apply button

g. Failure Cases

- i. User can still apply to job posting

## Messaging Test Cases

1. Direct message
  - a. Requirements Tested
    - i. ME01 and ME03
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to the internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. Input
      1. User Alice
      2. User Bob
    - ii. Output
      1. Alice can send and receive messages to Bob
      2. Bob can send and receive messages to Alice
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User opens messages
    - iii. Alice selects messages with Bob or starts a new direct message with Bob
    - iv. Alice sends message to Bob
    - v. Bob sends message to Alice
  - f. Success Cases
    - i. Both users can send and receive messages from each other
    - ii. Messages are only visible to users in the direct message
  - g. Failure Cases
    - i. Past messages do not display
    - ii. Message is not visible to both users
    - iii. Message is visible to another user not included in the original two users
    - iv. New message option won't display
2. Group Message
  - a. Requirements Tested
    - i. ME02
  - b. Dependencies
    - i. User must be logged in

- c. Assumptions
  - i. User is connected to the internet
  - ii. User does not lose connection to internet
- d. Test Data
  - i. Input
    - 1. User Alice
    - 2. User Bob
    - 3. User Claire
  - ii. Output
    - 1. Alice can send and receive messages from Bob and Claire
    - 2. Bob can send and receive messages from Alice and Claire
    - 3. Claire can send and receive messages from Alice and Bob
- e. Test Execution Steps
  - i. User is on home page
  - ii. User opens messages
  - iii. Alice selects group message with Bob and Claire or starts a new direct message with Bob and Claire
  - iv. Alice sends message to group
  - v. Bob sends message to group
  - vi. Claire sends message to group
- f. Success Cases
  - i. All users can send and receive messages from each other
  - ii. Messages are only visible to users in the group message
- g. Failure Cases
  - i. Past messages do not display
  - ii. Message is not visible to all users
  - iii. Message is visible to another user not included in the group
  - iv. New message option won't display
- 3. Group Message - Too many Users
  - a. Requirements Tested
    - i. ME02
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to the internet
    - ii. User does not lose connection to internet
  - d. Test Data



- i. Input
      - 1. User Alice
      - 2. User Bob
      - 3. User Claire
      - 4. User Dale
      - 5. User Emma
    - ii. Output
      - 1. False - Group message could not be created
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User opens messages
    - iii. Alice tries to start a new direct message with Bob, Claire, Dale, and Emma
  - f. Success Cases
    - i. Group message is not created
  - g. Failure Cases
    - i. Number of users exceeds limit to how many users allowed in group chat
- 4. Add Contacts List
  - a. Requirements Tested
    - i. ME05
  - b. Dependencies
    - i. User must be logged in
    - ii. Knowledge of username
  - c. Assumptions
    - i. User is connected to the internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. Input
      - 1. Username - bobbydylan
    - ii. Output
      - 1. True - contact added
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User opens messages
    - iii. User selects add new contact
    - iv. User enters contact information - bobbydylan

- v. User selects save
  - f. Success Cases
    - i. New contact is successfully saved
  - g. Failure Cases
    - i. New page does not pop up
    - ii. Information cannot be entered
    - iii. Information is not saved
- 5. Remove Contacts List
  - a. Requirements Tested
    - i. ME05
  - b. Dependencies
    - i. User must be logged in
    - ii. Knowledge of username
  - c. Assumptions
    - i. User is connected to the internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. Input
      - 1. Username - bobbydylan
    - ii. Output
      - 1. True - contact removed
  - e. Steps
    - i. User is on home page
    - ii. User opens messages
    - iii. User opens contacts list
    - iv. User selects contact bobbydylan
    - v. User selects remove contact
    - vi. User selects save
  - f. Success Cases
    - i. Contact is successfully deleted
  - g. Failure Cases
    - i. Contact page does not pop up
    - ii. User can not click button
    - iii. Information is not saved
- 6. Block Users
  - a. Requirements Tested
    - i. ME06

- b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to the internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. Input
      - 1. Username - bobdylan
    - ii. Output
      - 1. True - contact blocked
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User opens messages
    - iii. User clicks on username - bobdylan
    - iv. User selects block user
    - v. User selects save
  - f. Success Cases
    - i. User is successfully blocked
    - ii. New messages are no longer available to see with newly blocked user
  - g. Failure Cases
    - i. No window allowing user to block shows up
    - ii. User cannot click button
    - iii. Information is not saved
    - iv. Can still see new messages with blocked user after they blocked the user
7. Unblock Users
- a. Requirements Tested
    - i. ME06
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to the internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. Input
      - 1. Username - bobdylan
    - ii. Output

1. True - contact unblocked

e. Test Execution Steps

- i. User is on home page
- ii. User opens settings
- iii. User goes to blocked list
- iv. User selects username - bobjdylan
- v. User selects unblock user
- vi. User selects save

f. Success Cases

- i. User is successfully unblocked

g. Failure Cases

- i. No window allowing user to unblock shows up
- ii. User cannot click button
- iii. Information is not saved

8. Leave Group

a. Requirements Tested

- i. ME02

b. Dependencies

- i. User must be logged in
- ii. User needs to be in at least one group message

c. Assumptions

- i. User is connected to the internet
- ii. User does not lose connection to internet

d. Test Data

- i. Input
  1. Group message with Alice, Bob, Claire, and Dale
- ii. Output
  1. Group message with Alice, Bob, and Claire

e. Test Execution Steps

- i. User is on home page
- ii. User opens messages
- iii. User clicks on group message with Alice, Bob, Claire, and Dale
- iv. Dale selects option to leave group
- v. User selects yes

f. Success Cases

- i. User successfully leaves the group and does not receive messages from that group anymore

- g. Failure Cases
  - i. Group message does not pop up
  - ii. Users cannot select button
  - iii. Information not saved
  - iv. User still receives messages from group
- 9. Search
  - a. Requirements Tested
    - i. ME04
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to the internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. Input
      - 1. Username - bobdylan
    - ii. Output
      - 1. List of search matches - bobdylan
  - e. Test Execution Steps
    - i. User is on home page
    - ii. User opens messages
    - iii. User goes to search option
    - iv. User enters username bobdylan
    - v. User selects search
  - f. Success Cases
    - i. Desired username pops up
  - g. Failure Cases
    - i. Information cannot be entered
    - ii. No username pops up
- 10. Character Length
  - a. Requirements Tested
    - i. ME07
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to the internet
    - ii. User does not lose connection to internet

d. Test Data

i. Test 1

1. Input

- a. Message to send - "According to all known laws of aviation, there is no way a bee should be able to fly. Its wings are too small to get its fat little body off the ground."

2. Output

- a. True - message under 1000 characters, message sends

ii. Test 2

1. Input

- a. Message to send - "Once upon a time there was a lovely princess. But she had an enchantment upon her of a fearful sort which could only be broken by love's first kiss. She was locked away in a castle guarded by a terrible fire-breathing dragon. Many brave knights had attempted to free her from this dreadful prison, but none prevailed. She waited in the dragon's keep in the highest room of the tallest tower for her true love and true love's first kiss. (Laughs, tears out a page of the book) Like that's ever gonna happen. What a load of - (toilet flush). Shrek exits an outhouse and goes about his day in the swamp. In a nearby village, an angry mob gather up to go after Shrek. At night, the villagers head into the swamp and wait outside Shrek's home. Think it's in there? All right. Let's get it! Whoa. Hold on. Do you know what that thing can do to you? Yeah, it'll grind your bones for it's bread. Shrek sneaks up behind them and laughs. Yes, well, actually, that would be a giant. Now, ogres, oh they're much worse. They'll make a suit from your freshly peeled skin..."

2. Output

- a. False - message over 1000 characters, does not send

e. Test Execution Steps

i. Test 1

- 1. User is on home page
- 2. User opens messages

3. User goes to send message to Bob
    4. User types message from test data test 1 input
    5. User presses send
  - ii. Test 2
    1. User is on home page
    2. User opens messages
    3. User goes to send message to Bob
    4. User types message from test data test 2 input
    5. User presses send
- f. Success Cases
  - i. Messages under 1000 characters are sent
- g. Failure Cases
  - i. Messages over 1000 characters do not get sent

## Schedule of Classes Test Cases

1. Obtaining course information from school databases
  - a. Requirements Tested
    - i. CI01, CI02, CI06, CI07
  - b. Dependencies
    - i. IPs or URLs to obtain the course information from
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. Assumes that we have direct access to query the school's database for courses
    - iv. When testing is ready to begin, course sections used as test data will be manually checked for changes and new sections will be added to the test data as needed
    - v. The test data will be updated to include course sections from summer 2022 and/or fall 2022 as they become available
  - d. Test Data
    - i. The following courses from the spring 2022 catalogue:
      1. CECS 328 "Algorithms", Section 01, MW, 09:30-10:45, "Staff"
      2. ENG 100A "Composition I", Section 07, TuTh, 08:00-09:15, "Staff"
      3. MATH 247 "Introduction Linear Algebra", Section 03, MW, 12:30-13:45, "Wu B"
      4. PHYS 151 "Mechanics and Heat", Section 10, Tu, 10:00-12:45, "Luna C"
  - e. Test Execution Steps
    - i. Send the necessary request to obtain a copy of the data and save it to our database
    - ii. Check for the presence of the test courses in
  - f. Success Cases
    - i. All test data course sections are present in the database
  - g. Failure Cases
    - i. Any test data course section is missing from the database
2. Searching for a particular course
  - a. Requirements Tested
    - i. CI03, CI04



- b. Dependencies
  - i. User must be logged in
  - ii. Courses from the user's school must already have been obtained from the school
- c. Assumptions
  - i. User is connected to internet
  - ii. User does not lose connection to internet
  - iii. This test case's steps can be repeated to test the search feature of the schedule builder and the course difficulty ratings features
  - iv. When testing is ready to begin, course sections used as test data will be manually checked for changes and new sections will be added to the test data as needed
  - v. The test data will be updated to include course sections from summer 2022 and/or fall 2022 as they become available
- d. Test Data
  - i. The following courses from the spring 2022 catalogue:
    - 1. CECS 328 "Algorithms", Section 01, MW, 09:30-10:45, "Staff"
    - 2. ENG 100A "Composition I", Section 07, TuTh, 08:00-09:15, "Staff"
    - 3. MATH 247 "Introduction Linear Algebra", Section 03, MW, 12:30-13:45, "Wu B"
    - 4. PHYS 151 "Mechanics and Heat", Section 10, Tu, 10:00-12:45, "Luna C"
  - ii. The following course, which does not exist: CECS 000, "Does Not Exist", Section 00, MW, 00:00-01:15, "Staff"
- e. Test Execution Steps
  - i. The user navigates to the schedule builder and searches for each test course
  - ii. The user searches for CECS 000
  - iii. The user navigates to the course difficulty rating feature and searches for both test courses
  - iv. The user searches for CECS 000
- f. Success Cases
  - i. Searching for each course yields that course as a search, excluding CECS 000
- g. Failure Case
  - i. Searching for either test course yields no results

- ii. CECS 000 is present in search results

## Registration Test Cases

1. Register New User
  - a. Requirements Tested
    - i. RE01, RE02, RE03, RE04, RE05, RE06, RE07
  - b. Dependencies
    - i. User must not already have a registered account
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. An account with the following information does not already exist:
      1. Email - michael.kriesel@student.csulb.edu
      2. Username - mkriesel
    - iv. The account listed above is deleted at the end of every testing cycle
  - d. Test Data
    - i. Test 1:
      1. Input
        - a. Student email - michael.kriesel@gmail.com
        - b. Username - mkriesel
        - c. Password - Pas\$word1
        - d. University - CSULB
      2. Output
        - a. False - This is not a .edu email address
    - ii. Test 2:
      1. Input
        - a. Student email - michael.kriesel@student.csulb.edu
        - b. Username - mkriesel
        - c. Password - password
        - d. University - CSULB
      2. Output
        - a. False - This is not a valid password
    - iii. Test 3:
      1. Input
        - a. Student email - michael.kriesel@student.csulb.edu
        - b. Username - mkriesel
        - c. Password - Pas\$word1
        - d. University - CSULB
      2. Output

- a. False - The user has not entered an email
  - iv. Test 4:
    - 1. Input
      - a. Student email - michael.kriesel@student.csulb.edu
      - b. Username - mkriesel
      - c. Password - Pas\$word1
      - d. University - CSULB
    - 2. Output
      - a. True - user has successfully created an account
  - v. Test 5:
    - 1. Input
      - a. Student email - michael.kriesel@student.csulb.edu
      - b. Username - mkriesel
      - c. Password - Pas\$word1
      - d. University - CSULB
    - 2. Output
      - a. False - An account already exists with this username
  - vi. Test 6:
    - 1. Input
      - a. Student email - michael.kriesel@student.csulb.edu
      - b. Username - mkriesel
      - c. Password - Pas\$word1
      - d. University - CSULB
    - 2. Output
      - a. False - An account already exists with this email
- e. Test Execution Steps
  - i. User navigates to the registration page
  - ii. User selects create new account
  - iii. User enters the information for the current test (from the Test Data above)
  - iv. User selects create account and validates test output
  - v. If creating the test was successful (Test 4), the user opens their email and verifies the creation of the account
  - vi. Steps 1-5 are repeated for all Tests in the Test Data. If the user created an account, the user logs out before repeating the steps
  - vii. The system deletes the account created
- f. Success Cases

- i. User successfully created an account with the system

g. Failure Cases

- i. User does not enter a .edu email address
- ii. User does not enter a valid password
- iii. User is missing required information
- iv. Information is not saved to database
- v. An account already exists with the given email
- vi. An account already exists with the given username

## Login Test Cases

1. Login User
  - a. Requirements Tested
    - i. LI01
  - b. Dependencies
    - i. User must have a registered account
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
    - iii. An account with the following information already exists:
      1. Username - aliceann
      2. Password - Pas\$word1
      3. Email - alice.ann@student.csulb.edu
  - d. Test Data
    - i. Test 1:
      1. Input
        - a. Username - aliceann
        - b. Password - Pas\$word1
      2. Output
        - a. True - user has successfully logged into their account
    - ii. Test 2:
      1. Input
        - a. Email - alice.ann@student.csulb.edu
        - b. Password - Pas\$word1
      2. Output
        - a. True - user has successfully logged into their account
    - iii. Test 2:
      1. Input
        - a. Username - aliceann
        - b. Password - Pas\$word1
      2. Output
        - a. False - The account does not exist
    - iv. Test 3:
      1. Input
        - a. Username - aliceann
        - b. Password - Pas\$word2
      2. Output

a. False - The password is incorrect

e. Test Execution Steps

- i. Navigate to site website <https://studentmulti-tool.com>
- ii. Enter username and password for the current test
- iii. Select login
- iv. Repeat steps 1-3

f. Success Cases

- i. User logs in with username
- ii. User logs in with email

g. Failure Cases

- i. The username is incorrect (does not exist in the database)
- ii. The password is incorrect
- iii. The user is not logged in

## Logout Test Cases

1. Logout User
  - a. Requirements Tested
    - i. LO01, LO02, LO03
  - b. Dependencies
    - i. User must be logged in
  - c. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
  - d. Test Data
    - i. Any valid user account that is logged in
  - e. Test Execution Steps
    - i. User selects logout button
    - ii. User selects yes
  - f. Success Cases
    - i. User is successfully logged out and returned to the login view
  - g. Failure Case
    - i. User is still logged in
2. User Closes the Browser
  - a. Dependencies
    - i. User must be logged in
  - b. Assumptions
    - i. User is connected to internet
    - ii. User does not lose connection to internet
  - c. Test Data
    - i. Valid user account that is logged in
  - d. Test Execution Steps
    - i. User closes the browser
  - e. Success Cases
    - i. User is successfully logged out
  - f. Failure Case
    - i. User is still logged in