

SW Engineering CSC648/848 Fall 2022

Project Name: New SFSU Student Center

Team Number: 05

Milestone 4

Date: 1 December 2022

| Student Name    | Roles             | Email  |
|-----------------|-------------------|--|
| Zhenyu Lin      | Team Lead, GitHub | <a href="mailto:zlin4@mail.sfsu.edu">zlin4@mail.sfsu.edu</a>     |
| Christopher     | Backend Lead      | <a href="mailto:cyee12@mail.sfsu.edu">cyee12@mail.sfsu.edu</a>   |
| Michael         | Frontend Lead     | <a href="mailto:mchang9@mail.sfsu.edu">mchang9@mail.sfsu.edu</a> |
| Elisa Hsiao-Rou | Team Member       | <a href="mailto:echih@mail.sfsu.edu">echih@mail.sfsu.edu</a>     |
| Steven Paul     | Team Member       | <a href="mailto:sfong10@mail.sfsu.edu">sfong10@mail.sfsu.edu</a> |
| Cameron         | Team Member       | <a href="mailto:cyee10@mail.sfsu.edu">cyee10@mail.sfsu.edu</a>   |

| Milestone/Version | Date       |
|-------------------|------------|
| M1V1              | 09/21/2022 |
| M1V2              | 10/05/2022 |
| M2V1              | 10/19/2022 |
| M2V2              | 11/10/2022 |
| M3V1              | 11/10/2022 |
| M3V2              | 12/1/2022  |
| M4V1              | 12/1/2022  |

## Contents

|  |    |
|--|----|
| Product Summary .....  | 3  |
| Usability Test Plan .....  | 9  |
| Test objectives .....  | 9  |
| Test Effectiveness.....  | 13 |
| User Satisfaction.....   | 13 |
| QA Test Plan .....   | 15 |
| Code Review .....  | 21 |
| Internal Peer Review .....   | 21 |
| External Peer Review (Other team member review our frontend code): ..... | 21 |
| Self-check on best practices for security .....                          | 23 |
| Self-check: Adherence to original Non-functional specs .....             | 24 |
| list of contributions.....   | 28 |

## Product Summary

The New SFSU Student Center is a new and improved version of the current student center at San Francisco State University. A student center is an essential tool for a student to ensure that they succeed. It includes many features, in which some may not be as helpful, but with the new and improved version of the SFSU's student center, it includes only the important features needed to guarantee success. For our product, we organized our features by priorities, with priority one (1) being the most important and priority three (3) being the least important. The features that fell under priority one (1) include:

### Student

- 1.1. Students shall log in before accessing the system.
- 1.2. Students shall be able to enroll in course sections.
- 1.3. Students shall not be able to enroll in a class that would cause the student to exceed the set unit limit.
- 1.4. Students shall not fully enroll in more than one section of the same class.
- 1.5. Students shall be notified when they are dropped from a course section.
- 1.6. Students shall be able to search for courses,
- 1.7. Students shall be able to add courses to a shopping cart, prior to enrolling.
- 1.8. Students shall have transcripts.

1.9. Students shall have a class schedule.

1.10. Students shall not fully enroll in multiple sections that overlap on the same date & time slot.

1.11. Students shall have a student calendar, showing the student's class schedule and the college's academic calendar.

1.12. Students shall be able to drop course sections.

1.13. Students shall receive a Hold/Alert if they have overdue charges.

1.14. Students shall be notified whenever new Holds/Alerts are created on their account.

1.15. Students shall be dropped from a course if they cannot prove they have first taken the course's prerequisites, or are currently taking the course's prerequisites.

1.16. Students shall be dropped from courses if they have overdue charges after the set deadlines.

1.17. Students shall be able to access their student records (including transcripts and payment receipts).

1.18. Students shall enroll in courses with one of two grading options: CR/NC or Letter Grade.

1.19. Students shall be able to switch between grading options within certain date & time slots.

1.20. Students shall be able to view their financial aid.

1.21. Students shall be able to receive Financial Aid.

1.22. Students shall be able to leave feedback reviews for professors of course sections that the student has taken before.

1.23. Students shall be able to contact the department of their major.

1.24. Students shall be able to upload their health records.

1.25. Students shall be notified of payment due dates

## Courses

2.1. Course sections shall have a number of seats.

2.2. Course sections shall have a waitlist.

2.3. Course sections that are full shall place enrolling students on the waitlist.

2.4. Courses shall tell the students which classes are required as prerequisites.

2.5. Courses shall belong to one (1) subject.

2.6. Courses shall require prerequisites.

2.7. Course sections shall have time slots.

2.8. Course sections shall have a location. (can be online)

2.9. Course sections shall have a list of the average grade received by students in past semesters.

2.10. Courses shall tell the student if the class is online, in person, hybrid, synchronous or asynchronous

#### Waitlist

3.1. Waitlisted students shall be notified when they are able to fully enroll in the section.

3.2. Waitlisted students shall be automatically enrolled if space is available.

3.3. Waitlisted students shall be notified if they are dropped from the waitlist

#### Class Schedules

4.1. Class schedules shall show a student's enrolled courses.

4.2. Class schedules shall show a student's waitlisted courses.

4.3. Class schedules shall show courses currently in the student's shopping cart.

#### Professor Reviews

5.1. Professor reviews made by students shall be anonymous.

5.2. Professor reviews made by students shall show the grade of the student publishing the grade.

5.3. Professor reviews shall only be made by students who have completed a course section that the professor has taught.

5.4. Professor reviews shall be displayed under a professor's profile, as well as within the attributes of any course section taught by that professor

## Transcripts

6.1. Transcripts shall list all courses taken in the past.

## Searches

7.1. Searches shall have parameters, which filter the displayed courses.

7.1.1. Searches can be filtered by a student's eligibility to enroll in the course.

7.1.2. Searches can be filtered by the professor.

7.1.3. Searches can be filtered by location.

7.1.4. Searches can be filtered by date & time.

7.1.5. Searches can be filtered by attribute. (online, asynchronous, lab, lecture)

7.1.6. Searches can be filtered by course name.

7.1.7. Searches can be filtered by course number. (not CRN)

7.2. Searches shall display a list of courses.

7.3. Searched course sections shall display all their important data in the listing.

(CRN, professor, location, date & time, units, name)

7.4. Searched course sections shall display on mouse-over, less important data in the search listing. (description, past grade averages, professor ranking, etc.)

7.5. Searched courses shall be add-able to the student's shopping cart

Our version of a student center is easy to use and all the important information a student will need is displayed right on the first page. By displaying the important information on the first page, the student will not have to navigate through multiple pages to get the information they need. This student center also allows students to leave reviews anonymously about the professors they have taken courses with. This is typically done on an external platform, but we have implemented it into this student center which allows the students to see the reviews right then and there. Our student center also has a shopping cart that will display the time conflicts between the courses a student chooses, if there are any.

Here is the link to view our product: [http://52.146.22.198/beta\\_prototype/](http://52.146.22.198/beta_prototype/)



## Usability Test Plan

### Test objectives

#### Courses

The course function testing allows users to add courses to their shopping cart. This allows the user to fill their cart with the classes they intend on taking. The reason we want to test this is to make sure that the usability of this function is straightforward for the user when they are looking up courses and applying for the classes that they want.

#### Class Schedule

We are using the class schedule to display the student's classes that they are taking during the current semester or quarter for the school year. Being able to see their schedule when they log in gives them an easy way to see the classes they are taking. The reason we want to add this feature is to allow students an easier way to display the class schedule for the current semester.

#### Professor Reviews

We are testing to make sure that the user is able to post a review on the professor just like rate my professor but it will show for other students to see when they are looking at new classes when applying for it. The reason we want them to test this function is to make sure it works when it comes to the actual site because it is very helpful when you can see information on a professor before taking them.

#### Transcript

With the transcript it allows the user to see their academic transcript by going to the student records. The reason we want to test this is to make sure all the data on the student is the proper information that is being shown.

## Searches

The search is gonna test the website's ability to search for a certain topic when you input a certain word or topic. The reason to test this is to see if our search function will output the correct information for the user when in use.

## Test description

## Courses

The way we have the website set up is by when you are logged in you see the search bar and with the search bar you are able to access the courses through the search bar. The way you start is by going to the search bar and searching up the course that you are interested in and from there you get your result. The course also gives a description of the course. When you first login The intended user using this is going to be the student because it allows them to access their courses and see what each of the courses is about and their information.

## Link:

[http://52.146.22.198/beta\\_prototype/SearchResults?category=course&field=Chinese](http://52.146.22.198/beta_prototype/SearchResults?category=course&field=Chinese)

## Class Schedule

We have it so that when you log in to your account you are shown the class schedule in the top left corner of the screen. The starting point is by being once you log into the website and are shown the home page that includes the class schedule after you click on the button for enroll you are brought to the enroll page and which also shows the class schedule. This is intended for the student because it shows the student their current schedule for the current semester.

Link:[http://52.146.22.198/beta\\_prototype/](http://52.146.22.198/beta_prototype/)

### Professor Reviews

We will have the professor reviews shown inside the student records in their own column showing the review for the course in the completed courses. The way you are able to see these by going to the student records from the home page and will be shown in the completed courses. The intended user for this is a student that wants to know if the professor they are going to take is a good professor or a bad one.

Link: [http://52.146.22.198/beta\\_prototype/studentrecords](http://52.146.22.198/beta_prototype/studentrecords)

### Transcript

The system setup for the transcript is shown by opening a new tab on your web browser with your current transcript for the semester and past semesters. You start off by going to the student records from the home page and then selecting the transcript button which will open the transcript right up. The intended user is a student that is interested to see what their current transcript has to say and sending their transcript to an internship for jobs.

Link: [http://52.146.22.198/beta\\_prototype/studentrecords](http://52.146.22.198/beta_prototype/studentrecords)

## Searches

The system setup for the search is within the nav bar you are able to go to the search bar and search between courses or a specific professor. The way you are able to access it by logging in to the home page and looking to the top right of the nav bar. Once you get there you are given the a list of suggested things to search for. This in intended to help the user find a certain course or professor a lot easier.

Link: [http://52.146.22.198/beta\\_prototype/](http://52.146.22.198/beta_prototype/) (Search through the navbar.)

## Test Effectiveness

| Test/Use          | % completed | Errors  | Comments  | Time  |
|-------------------|-------------|---|---|-------|
| Course            | 60%         | No Errors   | The courses are shown with a description about them. Didn't recommend similar topic of course | 1.74s |
| Class Schedule    | 60%         | No Errors   | You are able to see the class schedule in both the enroll page and the home page              | 745ms |
| Professor Reviews | 60%         | No Errors   | Not able to see the professor review that been submitted                                      | 1.22s |
| Transcript        | 100%        | No Errors   | Displays transcript properly  | 259ms |
| Searches          | 70%         | Courses not able to be searched by the date or time | Does not search by the time, date or location   | 2.13s |

## User Satisfaction

| Question  | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
|---|-------------------|----------|---------|-------|----------------|
| 1. Courses were easy to add.  |                   |          |         | ✓     |                |
| 2. Courses were easy to find.   |                   |          |         | ✓     |                |
| 3. Course descriptions provided plenty of details.                                    |                   | ✓        |         |       |                |
| 4. The class schedule provides enough courses within the topic you are interested in. |                   | ✓        |         |       |                |
| 5. The class schedule displays properly on the home page.                             |                   |          |         | ✓     |                |
| 6. The class schedule provides the student easy access to enroll in classes.          |                   |          |         | ✓     |                |

|  |  |   |  |   |  |
|--|--|---|--|---|--|
| 7. Professor reviews are easy to find.   |  | ✓ |  |   |  |
| 8. Professor reviews provide adequate information pertaining to the professor for that specific topic. |  | ✓ |  |   |  |
| 9. Professor reviews can only be left by students that have taken a course with that professor.        |  | ✓ |  |   |  |
| 10. Transcript gives the user easy access to the transcript  |  |   |  | ✓ |  |
| 11. The transcript displays all of the correct courses and grades.                                     |  |   |  | ✓ |  |
| 12. Transcript is displays the correct personal information.   |  |   |  | ✓ |  |
| 13. Searches are filtered properly.  |  | ✓ |  |   |  |
| 14. I can easily access the courses I searched for from the results.                                   |  |   |  | ✓ |  |
| 15. Search results give the proper information on what you searched for                                |  |   |  | ✓ |  |

## QA Test Plan

### 1. Security

- Test Objectives

We will be testing to see if login is supported using only email. Users should be able to register for an account using an email and this email is then used to login to the student center.

- HW and SW Setup

Hardware Setup:

- Computer, Laptop, or Mobile Device
- Modem
- Wi-Fi Router
- Mouse
- Keyboard

Software Setup:

- Operating System (Windows, macOS, Linux)
- Internet Browser (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, etc.)
- [http://52.146.22.198/beta\\_prototype/front](http://52.146.22.198/beta_prototype/front) (URL of student center)

- Feature to be Tested

Login feature

- QA Test Plan

| Test Number | Test Title     | Test Description                           | Test Input                                    | Expected Output  | Pass/Fail   |
|-------------|----------------|--|---|------------------|---|
| 1           | Email Login    | Attempt to login using a valid email.      | Email:<br>Test@Test<br>Password:<br>12346789a | Successful Login | Chrome:<br>Pass<br>Firefox:<br>Pass<br>Edge: Pass |
| 2           | Username Login | Attempt to login using a valid username.   | Email:<br>TestUser<br>Password:<br>12346789a  | Failed Login     | Chrome:<br>Pass<br>Firefox:<br>Pass<br>Edge: Pass |
| 3           | ID Login       | Attempt to login using a valid student ID. | Email:<br>123456789<br>Password:<br>12346789a | Failed Login     | Chrome:<br>Pass<br>Firefox:<br>Pass<br>Edge: Pass |

## 2. Capacity

- Test Objectives
 

We'll be testing if the search feature caps the number of courses displayed. The current SFSU student center doesn't let students search for items that would result in 300+ results. The new student center should have no cap on the number of courses displayed.
- HW and SW Setup
 

Hardware Setup:

  - Computer, Laptop, or Mobile Device
  - Modem
  - Wi-Fi Router
  - Mouse
  - Keyboard

Software Setup:

  - Operating System (Windows, macOS, Linux)
  - Internet Browser (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, etc.)
  - [http://52.146.22.198/beta\\_prototype/front](http://52.146.22.198/beta_prototype/front) (URL of student center)
- Feature to be Tested
 

Search function
- QA Test Plan

| Test Number | Test Title      | Test Description                              | Test Input  | Expected Output                                    | Pass/Fail                                   |
|-------------|-----------------|---|---|--|---|
| 1           | Class Search #1 | Search for a class using the search function. | Enter "Chinese" into the search bar and click search. | Search results page with Chinese course displayed. | Chrome: Pass<br>Firefox: Pass<br>Edge: Pass |
| 2           | Class Search #2 | Search for a class using the search function. | Enter "English" into the search bar and click search. | Search results page with English course displayed. | Chrome: Pass<br>Firefox: Pass<br>Edge: Pass |
| 3           | Class Search #3 | Search for a class using the search function. | Enter "Math" into the search bar and click search.    | Search results page with Math course displayed.    | Chrome: Pass<br>Firefox: Pass<br>Edge: Pass |



### 3. Page Performance

- Test Objectives
 

We will be testing to see if the pages of the student center load within a second. Whenever the user navigates to a new page, the page should load within one second of the user changing pages.
- HW and SW Setup
 

Hardware Setup:

  - Computer, Laptop, or Mobile Device
  - Modem
  - Wi-Fi Router
  - Mouse
  - Keyboard

Software Setup:

  - Operating System (Windows, macOS, Linux)
  - Internet Browser (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, etc.)
  - [http://52.146.22.198/beta\\_prototype/front](http://52.146.22.198/beta_prototype/front) (URL of student center)
- Feature to be Tested
 

Page loading times
- QA Test Plan

| Test Number | Test Title                 | Test Description  | Test Input  | Expected Output                                 | Pass/Fail                                   |
|-------------|----------------------------|---|---|---|---|
| 1           | Home Page Speed Test       | Navigate to the home page of the student center.            | Click the “SFSU Student Center Logo” on the top left portion of the page.               | Home page displayed within a second.            | Chrome: Pass<br>Firefox: Pass<br>Edge: Pass |
| 2           | School Calendar Speed Test | Navigate to the school calendar page of the student center. | On the home page, under “Important Dates”, click “See the full list of important date”. | School calendar page displayed within a second. | Chrome: Pass<br>Firefox: Pass<br>Edge: Pass |
| 3           | Finance Page Speed Test    | Navigate to the finance page of the student center.         | On the home page, under “Finances”, click “See the full                                 | Finance page displayed within a second.         | Chrome: Pass<br>Firefox: Pass               |

|  |  |  |                         |  |               |
|--|--|--|-------------------------|--|---------------|
|  |  |  | list of finances info”. |  | Edge:<br>Pass |
|--|--|--|-------------------------|--|---------------|

#### 4. Search Performance

- Test Objectives

We will be testing to see if search results are loaded within two seconds.

When the user searches for something, results should be displayed within two seconds.

- HW and SW Setup

Hardware Setup:

- Computer, Laptop, or Mobile Device
- Modem
- Wi-Fi Router
- Mouse
- Keyboard

Software Setup:

- Operating System (Windows, macOS, Linux)
- Internet Browser (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, etc.)
- [http://52.146.22.198/beta\\_prototype/front](http://52.146.22.198/beta_prototype/front) (URL of student center)

- Feature to be Tested

Search function

- QA Test Plan

| Test Number | Test Title           | Test Description                         | Test Input  | Expected Output  | Pass/Fail  |
|-------------|----------------------|--|---|--|--|
| 1           | Search Speed Test #1 | Search for a class using the search bar. | In the search bar, enter “Chinese” and hit enter. | Search results page with the “Chinese” course displayed within two seconds of hitting enter. | Chrome:<br>Pass<br>Firefox:<br>Pass<br>Edge:<br>Fail |
| 2           | Search Speed Test #2 | Search for a class using the search bar. | In the search bar, enter “English” and hit enter. | Search results page with the “English” course displayed within two seconds of hitting enter. | Chrome:<br>Pass<br>Firefox:<br>Pass<br>Edge:<br>Pass |

|   |                      |  |  |   |   |
|---|----------------------|--|--|---|---|
| 3 | Search Speed Test #3 | Search for a class using the search bar. | In the search bar, enter “Math” and hit enter. | Search results page with the “Math” course displayed within two seconds of hitting enter. | Chrome: Pass<br>Firefox: Pass<br>Edge: Pass |
|---|----------------------|--|--|---|---|

## 5. Support

- Test Objectives
  - We will be checking to see that the student center supports English. All of the pages of the student center should have English support.
- HW and SW Setup
  - Hardware Setup:
    - Computer, Laptop, or Mobile Device
    - Modem
    - Wi-Fi Router
    - Mouse
    - Keyboard
  - Software Setup:
    - Operating System (Windows, macOS, Linux)
    - Internet Browser (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, etc.)
    - [http://52.146.22.198/beta\\_prototype/front](http://52.146.22.198/beta_prototype/front) (URL of student center)
- Feature to be Tested
  - Language support
- QA Test Plan

| Test Number | Test Title      | Test Description  | Test Input  | Expected Output  | Pass/Fail                                   |
|-------------|-----------------|---|---|--|---|
| 1           | Home Page       | Navigate to the home page and check if English support is provided.                   | Click the “SFSU Student Center” logo on the top left portion of the page. | The home page of the student center in English.            | Chrome: Pass<br>Firefox: Pass<br>Edge: Pass |
| 2           | Student Records | Navigate to the student records page and check to see if English support is provided. | On the home page, scroll down to “Student Records” and click “See the     | The student records page of the student center in English. | Chrome: Pass<br>Firefox: Pass               |

|   |                |  |  |  |  |
|---|----------------|--|--|--|--|
|   |                |  | full list of student record".                                    |  | Edge:<br>Pass  |
| 3 | Search Results | Use the search function and enter "Chinese". Then check if the results are displayed in English. | Enter "Chinese" into the search bar and click the search button. | The search results page with results displayed in English. | Chrome:<br>Pass<br>Firefox:<br>Pass<br>Edge:<br>Pass |

## Code Review

Internal Peer Review (Backend developer review the frontend code):

I'm not super familiar with frontends in JS, but it seems like most of the styling is done within the js files themselves, such as in HoldsAndAlerts.js and NotificationComponenets.js. This could just be a minor problem, depending on how much styling you want to put in. I'm not sure if you have other files for CSS in this folder (I noticed some in the pages folder), but if you dont you could consider placing them into CSS files for organizational purposes.

External Peer Review (Other team member review our frontend code):

What I like :

Everything is separated in react components, that's make the code cleaner and file shorter.

Code is self documented. I mean I don't need comment because I know react/html.

File and directory architecture

ClassName for css have good name

What I don't like :

Personally for web dev I prefer use tab of 2 instead of 4 because there are lot of tab.

Maybe lot of files for only one page. I know it's because of react components but maybe there are too many components. I know that's I said I like the way that's you use react components but try to use less because in a big project you will have maybe too many file. I think react components are useful only if you will reuse them later. I you use them only once, maybe is not useful to create component.

## Self-check on best practices for security

- List major assets you are protecting:
  - The major assets we are protecting are user information such as their personal information, usernames, passwords, transcript information, addresses, financial information.
  
- Confirm that you encrypt PW in the DB:
  - We have yet to encrypt our password in our database. We have plans to do so using Bcrypt, but we have yet to implement them.
  
- Confirm Input data validation (list what is being validated and what code you used):
  - Input data validation is done in a series of checks within our controllers. The data being validated are tokens that determine if a user is logged in. StudentID which checks who the student is in our database. Emails and passwords for when the user is logging in to check if they have the proper credentials. A general check is done to see if our data is valid, e.g the data is not null or undefined. Likewise, we check to see if a student I.D only contains numbers, as a student ID should not have any letters or special characters.

## Self-check: Adherence to original Non-functional specs

### 1. Security

1.1 The system shall make sure the data is encrypted.

-DONE

1.2 The system shall have security questions.

-Issue, because we think we don't have enough time to complete this nonfunctional requirement on time since it require extra table in the database.

1.3 The system shall lock the user out after five failed attempts to log in

-Issue, because we will need to make a reset password function for the user that has been locked. However, the reset password function isn't our Priority

1.

1.4 The system shall automatically log users out after a certain period of time for security.

- ON TRACK

1.5 The system shall only support login via email, username, or student ID.

- DONE

1.6 The system shall not allow authenticated persons to log in to the admin panel.

- Issue, we don't have any admin panel in our application.

### 2. Performance

2.1 Pages shall load within one (1) second.

- Issue, some of the pages require sending a request to the backend, the network condition can be various base on each user's network environment, and we are not able to guarantee page shall load within 1 second for all users.



2.2 The current SFSU student center can take ten (10) or more seconds to load. - DONE

2.2 Pages shall adjust accordingly to the user's device. -DONE

2.3 The system shall be able to handle multiple visits at once. -DONE

2.4 Searches shall execute in under one (1) second.

-Issue, the network condition can be various base on each user's network environment, and we are not able to guarantee page shall load within 1 second for all users

### 3. Maintainability

3.1 Maintenance shall be done at night in the college's timezone. -DONE

3.2 Maintenance shall be kept as short as possible. -DONE

3.3 Testing shall be performed regularly. -DONE

3.4 The system design have to be easy to maintain. -DONE

3.5 The system design have to be easy to maintain. -DONE

### 4. Usability

4.1 Pages shall be easy to navigate. -DONE

4.2 The system shall run without affecting other applications. -DONE

### 5. Data Integrity

5.1 Data in the system shall be backed up every day. -DONE

5.2 Professor reviews shall be approved before they're published. -DONE

### 6. Capacity

6.1 The search feature shall have no cap on the amount of courses displayed. (currently, SFSU won't let a student make a search that would result in 300+ classes. If necessary, split it into pages rather than prevent a student from searching.) – ON TRACK

## 7. Support

7.1 The system shall support English. - DONE

7.2 The System shall use Google Translate to support languages other than English. – ON TRACK

7.3 The system shall use Google Maps as the integrated map system. – ON TRACK

7.4 The system shall support PayPal as a payment method.

- Issue: payment system isn't our Priority 1

7.3 The system shall reflect any updated payment/financial information within 24 hours of the transaction's initiation.

- Issue: payment system isn't our Priority 1

## 8. TeamWork

8.1 The team shall form a consensus agreement before any push to the main git branch.-DONE

8.2 The front-end team and back-end team shall all agree before pushing edits to the master branch. -DONE

8.3 Both front-end and back-end leads shall get approval before pushing a major edit to the master branch. -DONE

8.4 Edits to the database shall be approved by the database master. -DONE

## list of contributions

| Name                   | Score | Contribution   |
|------------------------|-------|--|
| Elisa Hsiao-Rou Chih   | 9     | Product summary, headers on the login, register, school calendar, and search results pages   |
| Steven Paul Fong       | 9     | QA Testing section of the documentation  |
| Cameron Michael Yee    | 10    | back-end coding and testing<br>coding models, controllers, and services.   |
| Michael Harrison Chang | 9     | worked on writing the utility test section, the comment headers for the pages, choosing the pages for the code review  |
| Christopher Alan Yee   | 9     | Controllers for API: enrollment, financial aid, notification, review, search, shopping cart, transcript<br>Documentation in m4v1 best security practices<br>Tests created in the index |
| Zhenyu Lin             | 9     | Working on organizing the code structures and the front end of the beta prototype. Working on connecting the front and the backend by the API  |