SW Engineering CSC648/848 Fall 2022

Project Name: New SFSU Student Center

Team Number: 05

Milestone 4

Date: 15 December 2022

Student Name	Roles	Email
Zhenyu Lin	Team Lead, GitHub	zlin4@mail.sfsu.edu
Christopher	Backend Lead	cyee12@mail.sfsu.edu
Michael	Frontend Lead	mchang9@mail.sfsu.edu
Elisa Hsiao-Rou	Team Member	echih@mail.sfsu.edu
Steven Paul	Team Member	sfong10@mail.sfsu.edu
Cameron	Team Member	cyee10@mail.sfsu.edu

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
M4V2	12/15/2022
M5V1	12/15/2022

Contents

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)

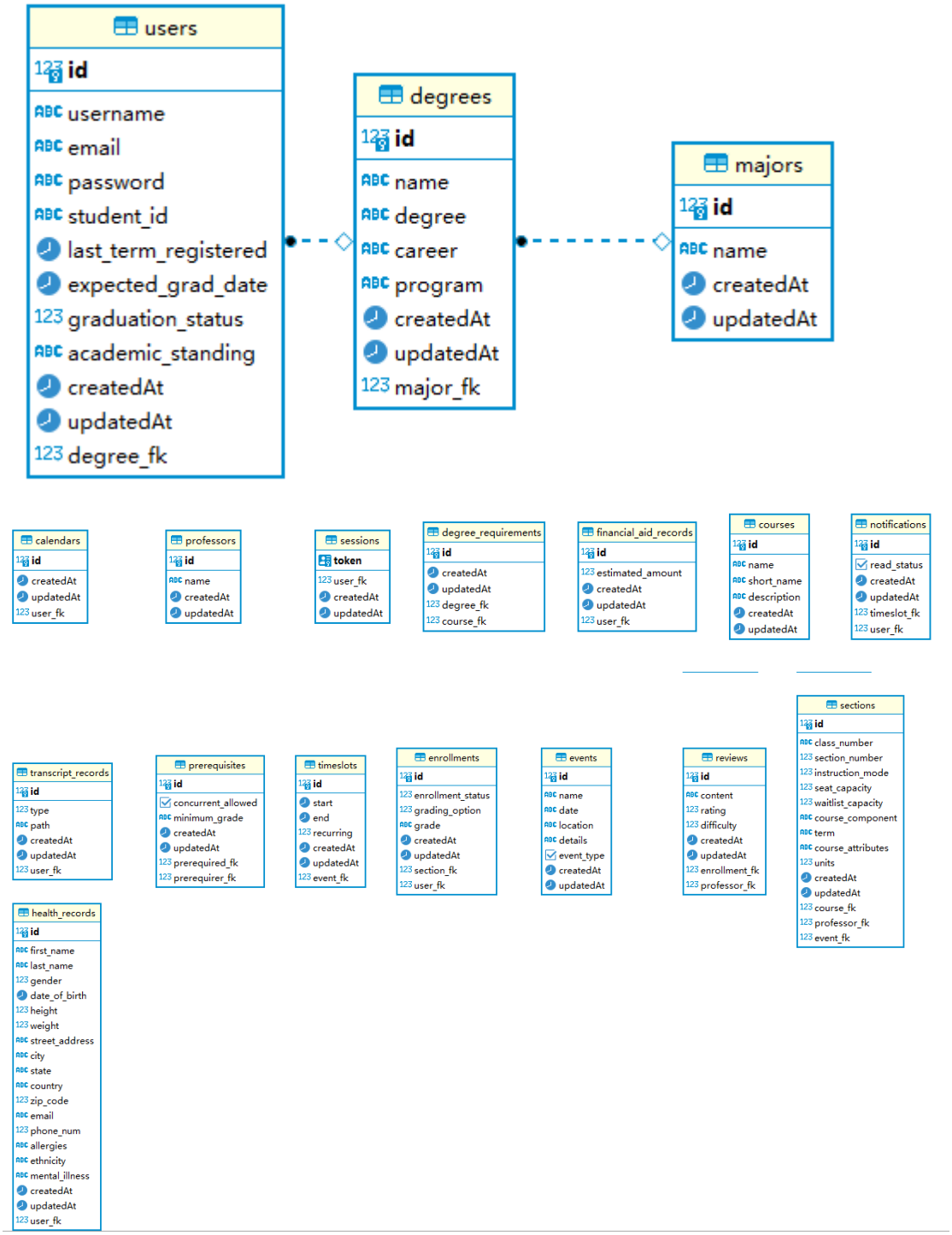
7.4. Searched course sections shall display on mouse-over, less important data in the search listing. (description, past grade averages, professor ranking, etc.)

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

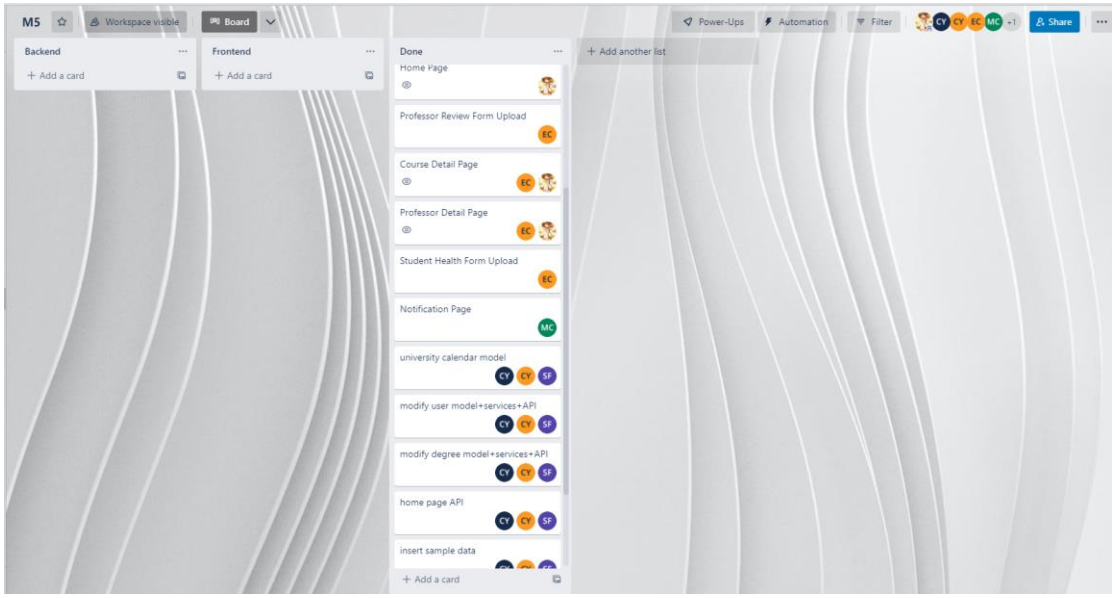
Our version of a student center is easy to use and all the important information a student will need is displayed right on the first page. By displaying the important information on the first page, the student will not have to navigate through multiple pages to get the information they need. This student 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/

Screenshots of key DB tables



Screen shot(s) of your task management system



Team member contributions

Milestone 5 Contribution

Name	Score	Contribution
Elisa Hsiao-Rou Chih	10	Working on Notification Page, Front Page, Professor Review Form, Student Review Form, Course Detail Page, and Professor Detail Page.
Steven Paul Fong	10	Working on sample data for the database
Cameron Michael Yee	10	Working on all enrollment api, login api, register api etc…
Michael Harrison Chang	10	Working on the School calendar page and Notification Page …
Christopher Alan Yee	10	Working on Calendar api and notification api
Zhenyu Lin	10	Fixing frontend bugs and working on all pages.

Overall contribution in fall 2022 semester

Name	Score	Contribution
Elisa Hsiao-Rou Chih	9	Elisa is our frontend UI designer, she designs all of our UI and logo and color.
Steven Paul Fong	9	Steven is our frontend developer, he doesn't talk a lot in the discord channel. However, he always finishes his tasks on time.
Cameron Michael Yee	10	Cameron is our backend developer, he has great contributions to the project. From M1-M3, he mainly worked on the documentation with Chris. The document is high quality and well organized. Starting from M4, he focuses on the backend database development. He built the database and worked on 80% of the backend API.
Michael Harrison Chang	10	Michael is our frontend lead, he also has great contributions to the project. From M1-M3, he focuses on UI design and front-end implementation. In addition, he is good at solving problems and bugs.
Christopher Alan Yee	9	Chris is our backend lead, he did a great job with the documentation. The documents he and Cameron made always get full credit.
Zhenyu Lin	9	I am the leader of this team, I focus on breaking the big task into small assignments and assigning them to the team members. I mainly help my team members solve bugs and questions.

Post Analysis

In this project, the main challenge is team management. Although we have 6 people on a team and 3 weeks to complete the milestone, time is still tight. I think the reason is that we all have different class assignment deadlines, some of the teammates have 3 projects in this semester. This caused us only to be able to spend 10% of the time on this project. The second reason is the frontend team will need to wait for the backend team complete the API first then they can start to connect the frontend and the backend. This causes the team unable to work in parallel.

The biggest mistake in this project is we forget about our unique features. In the process of implementing the priority 1 requirements in M3, we always do the easy requirement first. However, we don’t have enough time to implement all the priority 1. The M3 prototype is missing our unique features. The lesson I learn is when we try to implement requirements, we should implement the requirements that are related to the unique features first instead of the easy ones first.

Milestone Documents

SW Engineering CSC648/848 Fall 2022

Project Name: New SFSU Student Center

Team Number: 05

Milestone 1

Date: 9 September 2022

Student Name	Roles	Email
Zhenyu Lin	Team Lead, GitHub Master	zlin4@mail.sfsu.edu
Christopher Alan Yee	Backend Lead	cyee12@mail.sfsu.edu
Michael Harrison Chang	Frontend Lead	mchang9@mail.sfsu.edu
Elisa Hsiao-Rou Chih	Team Member	echih@mail.sfsu.edu
Steven Paul Fong	Team Member	sfong10@mail.sfsu.edu
Cameron Michael Yee	Team Member	cyee10@mail.sfsu.edu

Milestone/Version	Date
M1V1	09/21/2022
M1V2	10/20/2022

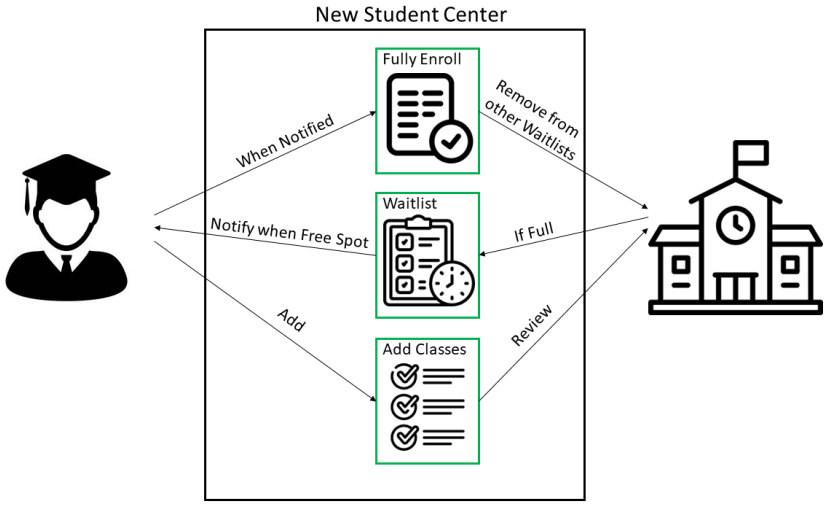
Contents

Project Description

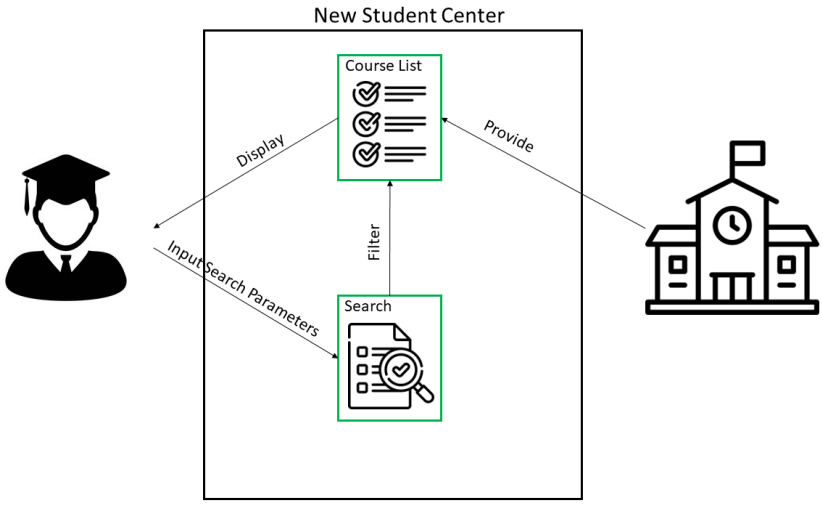
More and more students are enrolling in colleges, and most of them are the first college students in their families. A highly integrated website of college resources is necessary to guide freshmen year students quickly adapt to college life. Many colleges have a website called student center to help freshmen. However, they are not a highly integrated website. The SFSU student center doesn't integrate with the campus clinic. The UCSC student center has 2 types of login systems, one is for basic management of the school such as enrolling in classes, and the other one is for advanced management such as paying for tuition. Many of the student centers had poor user experience due to security reasons and ancient design. This project proposes a modem student center, it integrates a professor ranking system to allow students to choose the best professor for their class. In addition, our student center integrates club resources for non-local students to quickly assimilate into a new community. The survey shows more than 50% of SFSU students want a new highly integrated student center.

Use Cases

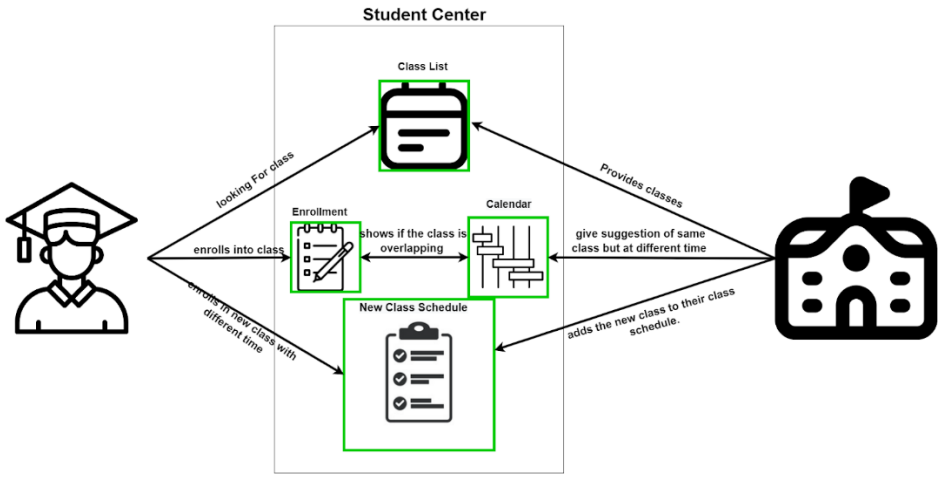
Use Case 1

Title	Adding Classes and WaitLists
Actor	Joe (student), CSM (college)
Description	Joe is a student at the College of San Mateo who wants to sign up for ECON 110-3, but the section is full. Because there are no seats available, the system sends Steve to a waitlist. Realizing that he might not get enrolled in the class, Steve adds sections 1 and 2 of ECON 110 as well, hoping to raise his chances of getting into the class. He is anxious because this is his first time using this kind of enrollment system with waitlisting and add-codes. He is unsure if enrolling in multiple sections is technically allowed, or if a bug in the system is allowing him to do so, and he is afraid of getting into trouble. This confusion is amplified by the fact that section 3 has a time conflict with his MATH 220 class, which he intends to drop if he successfully gets into ECON 110-3. The system reassures Joe that if a spot opens up before the first session of class, he will be notified and will be able to enroll at that time. The system also informs him that even though he is allowed to be on the waitlist of multiple sections of the same class, when he does officially enroll in one of them, he will automatically be dropped from the others, as well as any other classes he is enrolled in which would present a time conflict, such as his MATH 220 class.
Diagram	

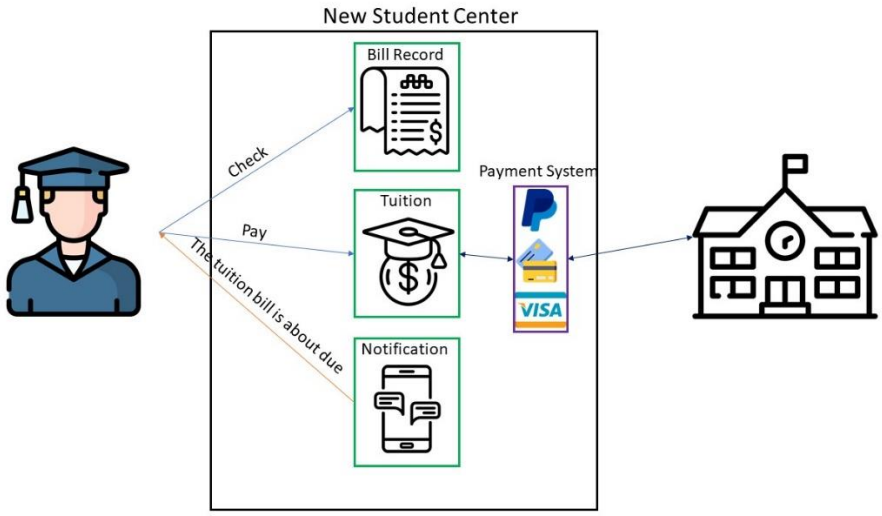
Use Case 2

Title	Searching for Classes
Actor	Erica (student), SFSU (college), Skyline (college)
Description	Erica is a prospective student, browsing SFSU's course catalog online. She is transferring from Skyline, the local community college, and wants to ensure that her transferred courses are valid as prerequisites. She also wants to ensure that all her classes will be online and that the professors will be good as well. Erica is initially intimidated because when she looks at the class bulletin, she is faced with information overload, and making a decision from so many choices is daunting. She frequently searches for the specific class that she needs, and then hand writes down each class's information so she can compare them. Since SFSU has received her transcripts, she can use the system to filter her search to only show classes that she's eligible to take. She can also use the search filters to specify other attributes that she wants, so she can find classes that suit her needs and fulfill her degree's requirements. This way she can compare the classes' information directly on screen, without having to search each one individually and remember the information separately.
Diagram	

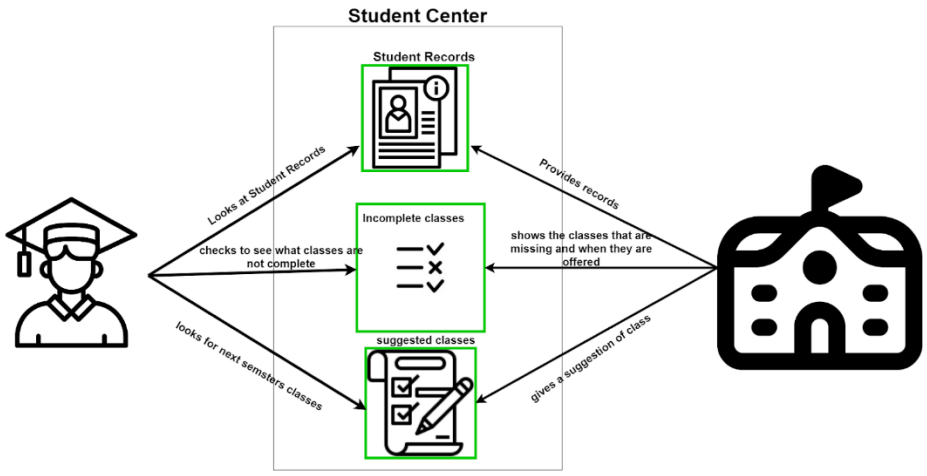
Use Case 3

Title	Scheduling Classes
Actor	Liam (Studnet), San Francisco state university (college)
Description	Liam is an incoming transfer student new to San Francisco State University. He is working on making his new class schedule for the following semester. While he is making his new schedule he is looking to see if any of his classes do not have conflicting times between each of the classes. Our new Student center will help Liam and others to schedule classes. This class schedule will integrate the school's class scheduling website into the student center. So when a student is scheduling their class for the next semester they are able to see what classes are offered and can immediately add those classes without having to go to another window. With this, it would allow the users to see the classes that are added on a calendar, which also shows if the classes are conflicting with time and will give suggestions on the same classes but with different times instead.
Diagram	

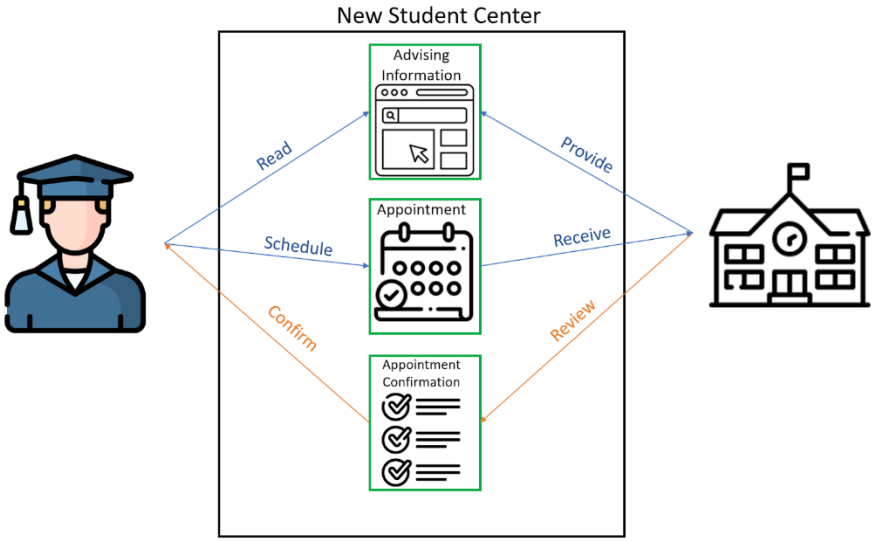
Use Case 4

Title	Paying For Classes
Actor	Andy (freshmen College Student)
Description	Andy is a freshman college student. He complains that the University student center is ancient. He spends a week figuring out which classes he needs to take. He thought the university would send a bill to his house and that he needed to send back a check to pay off the tuition. However, after a week, his student center notifies he needs to pay off the class fee, otherwise, he will be dropped from the classes. The student center doesn't have a payment system for the student to pay off their class. Andy has to go back to school on weekdays to pay his tuition. Our student center provides credit cards, debit cards, and PayPal payment systems to allow the student to pay off their tuition within five (5) minutes. Our student center will notify students when the payment is due by sending a text. In addition, our student center can show the history of payments, so students know how much they spend on the university.
Diagram	

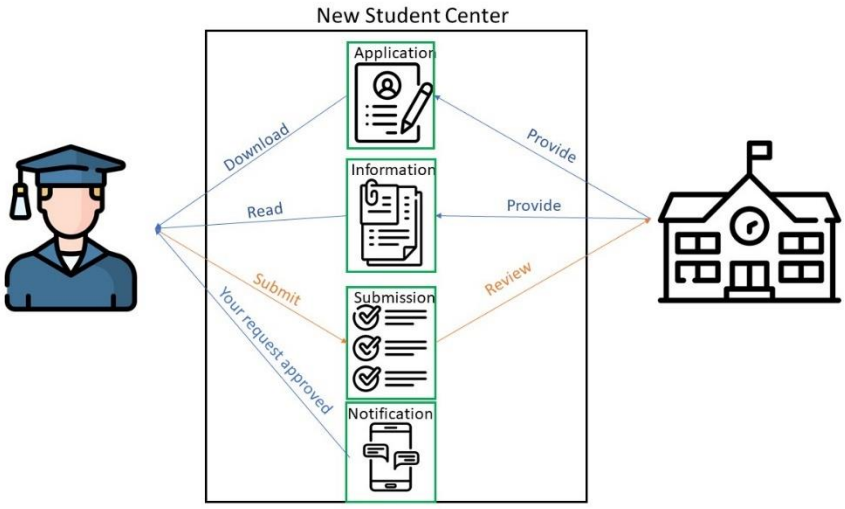
Use Case 5

Title	Accessing Student Records
Actor	Marry (Student), San Francisco state university (college)
Description	Mary is a student that is in her last year at San Francisco State University. She wants to know how many classes she has left. So she looks at her student records to see what classes she has left to take. She also wants to know what classes are offered during both the Spring and Fall semester for the remaining classes she has left to take. Our new student center will make it easier for Mary and others to see how many classes they have left. It will be easier for her to see how many classes she has left by displaying all of her classes at the top of the page starting with the classes required for her major first and then it would also show any of the GED classes that she has left to take. Along with showing the classes she has left, the student records will also show when the classes she needs are offered, i.e during the Spring, Summer, or Fall, right next to them.
Diagram	

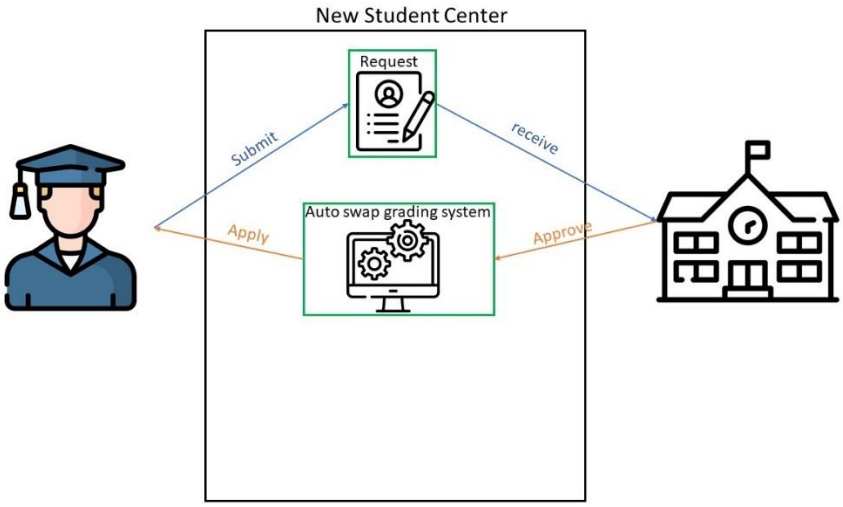
Use Case 6

Title	Scheduling Advising Appointments
Actor	Jack (Student), SFSU
Description	Jack is an incoming freshman at San Francisco State University. He was accepted as a Psychology major, but he is unsure what classes to enroll in for his first semester. He wants to schedule an appointment with an advisor to discuss what classes to take. However, he is unsure how to do this. He checks the student center and can't find any relevant information. He can only find advising plans and notes, but nothing related to actually making an appointment with an advisor. Jack is now left wondering what he has to do to schedule an advising appointment. Our new student center will help Jack and other students schedule advising appointments. Students will no longer have to navigate through the SFSU website to find information about advising hours and how to schedule appointments. All of this information will be provided by the new student center, leading to time saved for students and preventing confusion.
Diagram	

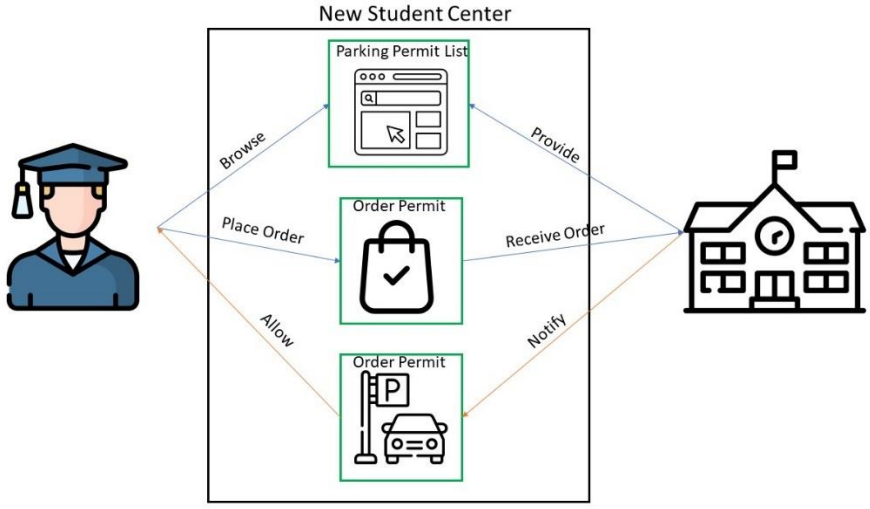
Use Case 7

Title	Changing Major
Actor	Billy (Student)
Description	Billy is a sophomore who has become disillusioned with his major after not enjoying his classes despite passing them. Billy wants to change his major to something he might enjoy more. He deliberates with his parents and he finally decides to switch to Computer Science. He looks around for a bit and has trouble finding where to declare and change his major. He is confused by the wording and the structure of the student center. He is unsure if his request to change his major actually went through, as he did not receive a confirmation via email. Time passes and he checks to see if his major request was accepted. It was accepted, but every time he wanted to check he had to log into his student center. Our new student center will send an email to Billy to tell him of the confirmation that his major request was accepted. We will also give him links to resources for his new major.
Diagram	

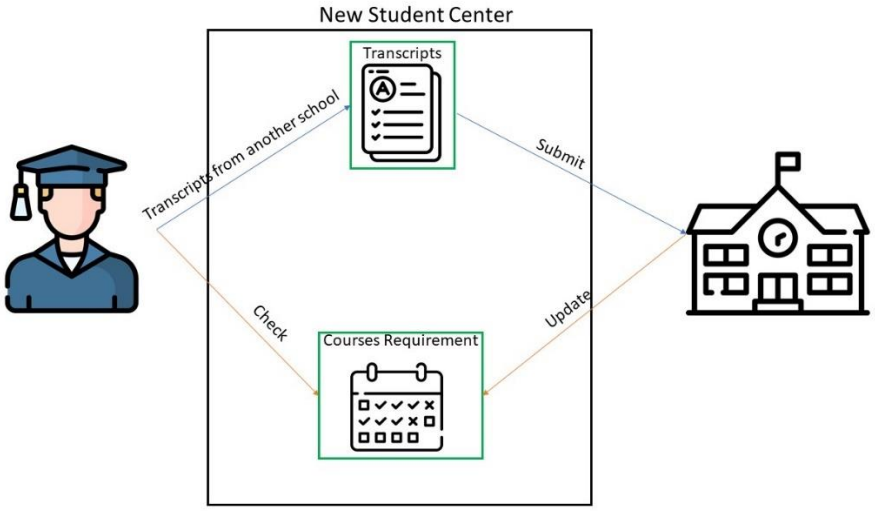
Use Case 8

Title	Appealing For Grade Change
Actor	Jimmy (Student)
Description	Jimmy is a junior who has been going through a rough patch in school. This semester he has not been able to focus on school. Because of this, he changed his grading methods for his classes from letter grade to credit/no credit to prevent a drop in his grade point average (GPA). He plans to retake the classes next semester. However, to Jimmy's surprise, he manages to pass a few of his Major required classes. Because of school policy, he must receive a letter grade for the classes associated with his major. Jimmy must then send an appeal to the dean and the head of his major department for a change from CR/NC to a letter grade. Our new student center will have ways to directly contact the dean and the head of his new major. This will help Jimmy be able to keep all of his concerns within the student center without having to send external emails.
Diagram	

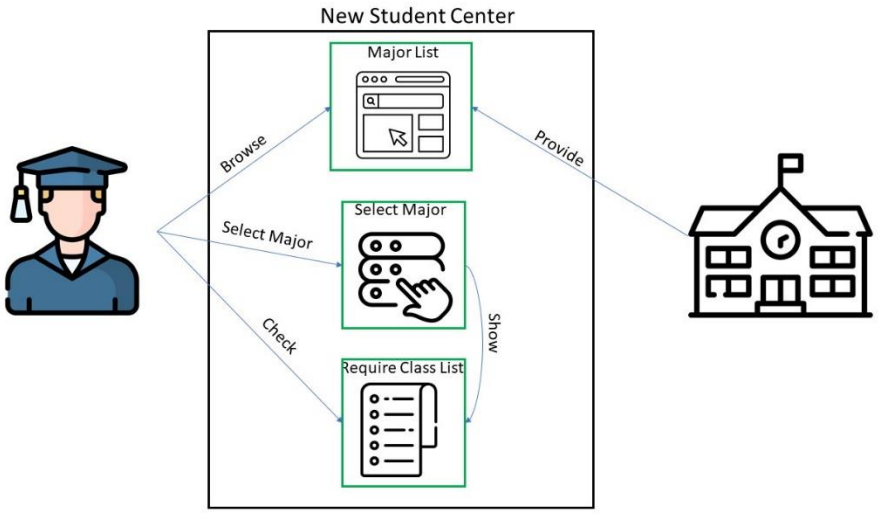
Use Case 9

Title	Paying for Parking Permit
Actor	Annie (Student), SFSU
Description	Annie is a student at San Francisco State University who lives off campus. She drives to class every day and wants to use the school parking lots. However, she does not have much information regarding parking permits. She decides to check the SFSU student center, but cannot navigate to a webpage where she can find information on how to acquire a parking permit. She is now confused about what steps to take next in learning about and acquiring a parking permit Our new student center will help students, such as Annie, find information about parking permits more easily. Students like Annie will be able to learn about the different types of parking permits and their prices through our student center. They will then be able to make a better-informed decision as well as pay for the parking permit. With the new student center, students will no longer have to search through the SFSU website to look for information regarding parking permits.
Diagram	

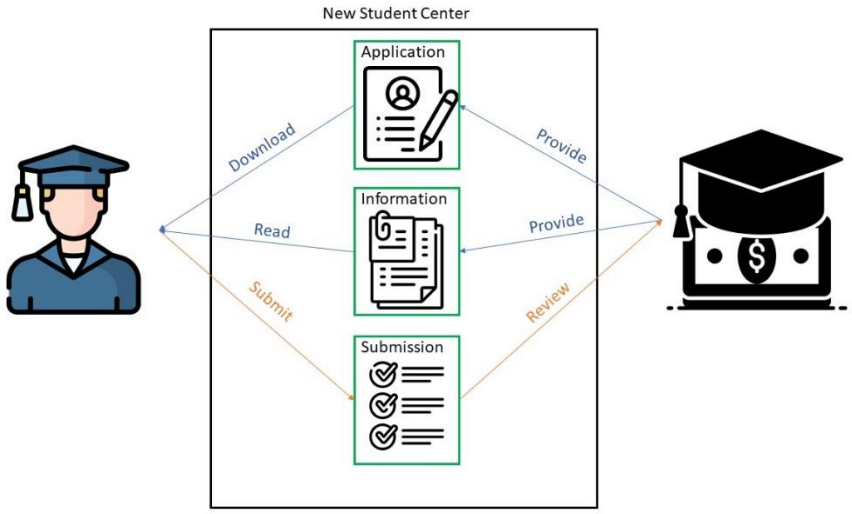
Use Case 10

Title	Transferring credit
Actor	Becky (student)
Description	Becky is transferring to SFSU from a different college. She has already declared her major and has taken the respective courses, but she wants to know which courses will be transferred over. Becky tries to navigate through the SFSU student center, but has to go scrolling through other menus to finally find the transfer credit report. When she does find it, the report doesn't show which major requirements were fulfilled by her transfer classes. It only shows the equivalent SFSU course and the credits she got for it. Becky is left still unsure which courses she has to complete to earn her degree. Our new student center will help Becky and other transfer students quickly and easily figure out which of their courses transferred to SFSU for credit. She can manually select the school and the classes she took at that school and she can then see if the classes are eligible for transfer. Another option is she can enter her old school's credentials and her transcript will be automatically pulled and she can see which credits will be transferred over. The transfer credit report will also show her which of the major requirements were fulfilled by transfer courses.
Diagram	

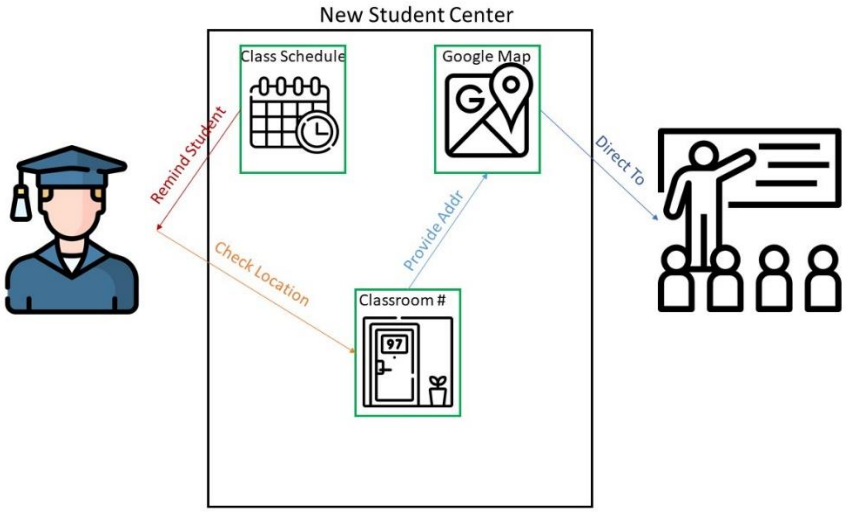
Use Case 11

Title	What-if report
Actor	John (student)
Description	John is a current student at SFSU but wants to know what will happen if he takes certain classes over others or if he were to switch his major. He goes onto the SFSU student center to look for a what-if report. However, he has to scroll through other submenus to find it. When John does find the what-if report feature, he has to go through several steps to generate it. In addition to this, the what-if report is cluttered, leaving John confused. Our new student center will provide students with more organized and clearer what-if reports. If John wants to change majors, he can look through the list of majors offered by SFSU and select one that interests him. A what-if report will be generated, showing him the list of classes he has already completed for the major and the classes he still has to take. He also has the ability to save the list of classes so he does not have to re-enter the information.
Diagram	

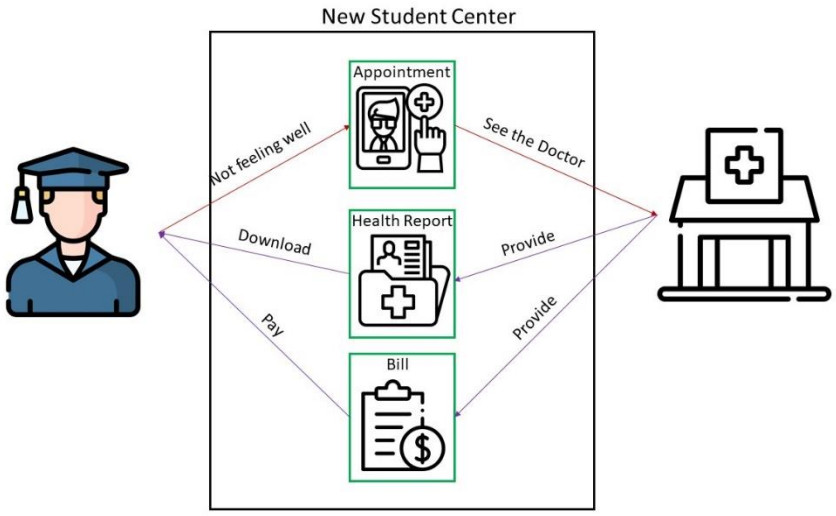
Use Case 12

Title	Unable to find financial aid information
Actor	Student Zhang
Description	Zhang is a graduate high school student and he will be a freshman college student in the following semester. He is anxious because he does not know where to find the information about financial aid. However, the university student center only provides an option that accepts or rejects financial aid. He needs to make an appointment with the university advisor to learn more about financial aid. However, the university advisor is only available next month. Zhang is worried that he will miss the financial aid submission window. Our new student center integrates a financial aid system to help freshmen college students like Zhang know where, how, and when to submit their financial aid applications. Our new student center also provides an appointment scheduling system for students to directly make an appointment with the financial aid advisor.
Diagram	

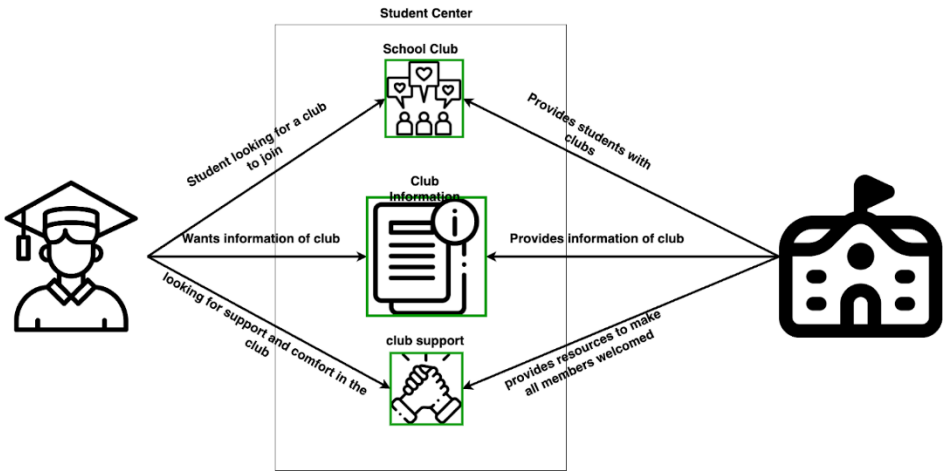
Use Case 13

Title	Difficult to find classroom
Actor	Student Sun
Description	Sun is a freshman college student and tomorrow is the first day of college life. He is exhausted because he has four (4) classes and they are all in different buildings. He spends three (3) hours going around the campus to find all his classrooms. However, he still does not remember the exact locations. He complains that the school should have a digital map. With our new student center integrating Google Maps, freshmen students like Sun can open Google Maps on the student center and the map will guide them to the correct building. Our new student center helps Sun and another student who has a poor sense of direction save lots of time memorizing the location of the classroom.
Diagram	

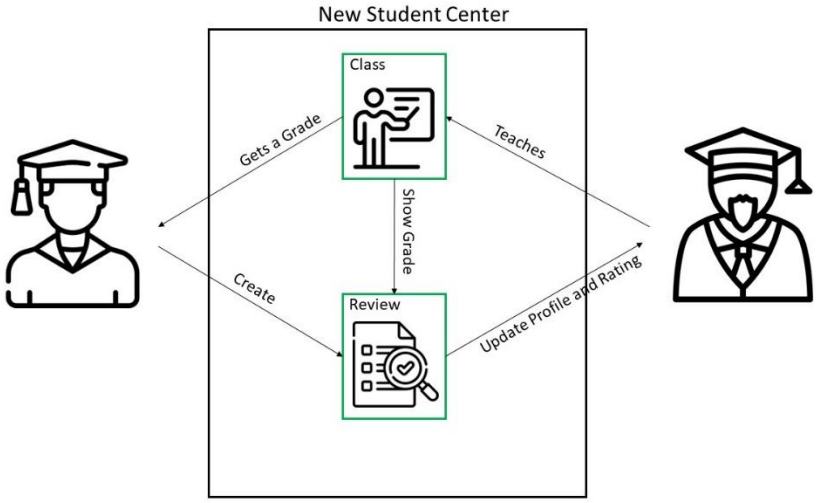
Use Case 14

Title	Lack of clinical support
Actor	Student Han
Description	Han is an international student. Recently, he got sick and he is looking for some clinical support. He goes to the student center to try to make an appointment with the campus clinic. However, the university student center has not integrated the campus clinical service. He needs to google the clinical service for his university to make an appointment. He complains the university service is too scattered and he has to google the corresponding service rather than integral in one place. Our new student center integrates the university health services. Students can make an appointment with a university client and upload their health records to the university client
Diagram	

Use Case 15

Title	Club Resource
Actor	Isla (Student)
Description	Isla is a first-year international student in her second semester. Recently, she has been feeling like she does not know where she should go in life. She has no one she can take with her, and her classes are coming up. She searches for an international club but it is nowhere to be found. Isla feels like there is a lack of resources when trying to find a new community, and the language barrier makes it more difficult to meet new friends. Our new student center is integrated with student activities and student clubs. Students like Isla can easily find clubs or activities they are interested in. She can use our new student center to find information about international student clubs, sports clubs, etc. Our new student center enhances the student's sense of belonging. Our new student center gives as much support as it can to the student.
Diagram	

Use Case 16

Title	Reviewing Professors
Actor	Jody (student), Ralph (professor)
Description	The school semester has ended, and Jody has just received the final grades for each class. She remembers liking professor Ralph's history class and wants to leave a review on the Student Center to express her thanks to the professor, and also to help other students know how useful and interesting the class was. When viewing past semesters' schedules, Jody can view each of the courses taken, and can leave a review for that course if she wants to. The system will warn her that even though the review will be posted anonymously, the review will display the course she took along with the grade she received, and it will be publicly viewable under the professor's profile page. She can then write the review, and select tags that accurately describe her experience taking the course, which will be used both to describe the course as a review, but also as a possible search parameter for any students making searches for professors that have certain teaching styles.
Diagram	

Main Data Item and Entities

1. Student - Our end users. Can enroll in course sections.
2. Transfer Student - A student that came from another college. Has a special student record called ‘past transcripts’ for courses taken at other colleges.
3. Undergraduate Student - A student that has not yet graduated with a degree.
4. Graduate Student - A student that has graduated with a degree.
5. Full-Time Student - A student that is taking enough units that would be equivalent to a full-time job.
6. Part-Time Student - A student that is taking less units than would be equivalent to a full-time job.
7. Course (sometimes referred to as “Class”) - The main product for the student.
8. Credit (sometimes referred to as “Unit”) - A unit of measurement for the time/effort of a course. A student needs a certain amount to graduate.
9. Course Section (sometimes referred to as “Section”) - A specific instance of a course. Associated with one professor and one date & time slot.
10. Online Section - A type of course section that has no location and is held fully online.
11. Synchronous Section - A type of online course section that has synchronous meetings over a video call.
12. Asynchronous Section - A type of online course section that does not meet and has no time slot.
13. Hybrid Section - A type of course section that combines elements of in-person classes and online classes.
14. Subject - An attribute of courses that are often associated with a major.

15. Location - An attribute of course sections that details where it takes place.
16. Time Slot (sometimes referred to as “Date & Time”) - An attribute of course sections that details when it takes place.
17. Course Component - An attribute of a course section describing the kinds of learning activity the student would be doing. (e.g. Lecture, Lab, Self-Study, etc.)
18. Course Materials - An attribute of a course section describing the kinds of materials the student may need to purchase to take the class (e.g. textbooks)
19. Major - An attribute of a student. Can determine what types of classes the student can take.
20. Student Record - A type of form that also acts as an attribute of students. Having certain completed records affects how the student can interact with the system.
21. Transcripts - A type of student record. A list of previously enrolled courses, and the grades received from them.
22. Student Calendar - A list of dates & time slots.
23. Class Schedule - A type of Student Calendar containing only the dates & times of the student’s course sections.
24. Appointments - Has a date & time, and acts as an element in a calendar, just like course sections.
25. Prerequisite - A type, of course, that is also an attribute for another different course. The prerequisite must be completed concurrently with, or before the student takes the associated other courses.
26. Professor - The teacher of a section. Can modify course section information.

27. Advisor - An employee of the university that specializes in helping students with school-related questions.
28. College (sometimes referred to as “University”) - The client organization. Uses the system to offer courses to students.
29. Department - A subsection of a college that facilitates learning for specific subject(s).
30. Campus Clinic - Site that students can visit for health-related issues.
31. Student Club - A school-recognized student organization. Associated with a faculty advisor.
32. Parking Permit - A form that grants students access to parking.
33. Bill- A form that shows students' transactions after purchase.
34. Charge - An element on a bill. A specific amount the student owes for a specific purpose.
35. Aid Form - A form that shows students’ financial aid information.
36. Map - An image showing the geographical layout of a location.
37. Schedule - an itinerary that you follow throughout the day.
38. Notification - an alert that notifies the student of an important message.
39. Holds - A property that prevents students from enrolling until resolved.
40. Search Parameter - An attribute or specific value of an attribute that is used to filter/narrow down elements from a full list of elements.
41. Course Requirements - a course that is required in order to take the next course.
42. Health Records - records that inform the school/doctor about one person's health information.

Functional Requirement

1. 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 able to search for courses.

1.6 Students shall be able to add courses to a shopping cart, prior to enrolling.

1.7 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.

1.8 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 Student calendars shall recommend alternative course sections in order to resolve date & time conflicts on the class schedule.

1.13 Students shall be notified if a course section on their calendar has any of its attributes changed. (students should know if there's a change of professor or change of location)

1.14 Students shall be able to drop course sections.

1.15 Students shall be able to swap one course for another.

1.16 Students shall be able to pay for courses.

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

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

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

1.20 Students shall be notified of payment due dates.

1.21 Students shall be able to access their payment histories.

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

1.23 Students shall be able to schedule advising appointments.

1.24 Students shall be able to change their major.

1.25 Students shall be notified when a change of major is fully processed, regardless of whether it is accepted or rejected.

1.26 Students who’ve recently changed majors shall have access to resources for their new major.

1.27 Students shall receive a grade, upon completing a course.

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

1.29 Students shall be able to switch between grading options before certain date & time slots.

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

1.31 Students shall be able to view their financial aid.

1.32 Students shall be able to purchase parking permits.

1.33 Students shall be notified if their permit purchase was approved.

1.34 Students shall be able to make an appointment with the university clinic.

1.35 Students shall be able to upload their health records.

1.36 Students shall be able to open Google Maps in the student center.

1.37 Students shall be able to search for clubs at the university.

1.38 Students shall be able to generate What-If Reports.

1.39 Students shall be able to save their What-If Reports.

1.40 Students shall be able to receive Financial Aid.

1.41 Students shall be able to schedule financial aid appointments.

1.42 Students shall be notified of upcoming appointments.

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

2. Courses

2.1 Course sections shall have a waitlist.

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

2.3 Courses shall require prerequisites.

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

2.5 Courses shall belong to one (1) subject.

2.6 Course sections shall have time slots.

2.7 Course sections shall have a location.

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

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

2.10 Course sections shall have a number of seats.

3. 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.

4. Searches

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

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

4.1.2 Searches can be filtered by professor.

4.1.3 Searches can be filtered by location.

4.1.4 Searches can be filtered by date & time.

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

4.1.6 Searches can be filtered by course name.

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

4.2 Searches shall display a list of courses.

4.3 Searched course sections shall display all their important data in the listing. (CRN, professor, location, date & time, units, name)

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

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

5. Transcripts

5.1 Transcripts shall list all courses taken in the past.

6. Class schedules

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

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

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

7. Payment

7.1 PayPal shall be supported as a payment method. (non-functional?)

8. Advising

8.1 Advisors shall be able to schedule appointment time slots. (maybe lower priority? I know we wanted to focus on the student side of the website before tackling the faculty side)

9. Transferring

9.1 Transfer students shall be able to search for other colleges' courses to check if they count as being transferred.

10. What-if report

10.1 What-If Reports shall show students the required classes for hypothetical change of major, degree, or other academic career choices.

11. Professors

11.1Professor reviews made by students shall be anonymous.

12. Professor Reviews

12.1 Professor reviews made by students shall be anonymous.

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

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

12.4 The system shall notify reviewers that their received grade will be displayed along with their anonymous review.

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

12.6 Professor reviews shall have tags to help students parameterize searches when filtering for professors with certain teaching styles.

Non-Functional Requirement

1. Security

    1.1 The system shall make sure the data is encrypted.

    1.2 The system shall have security questions.

    1.3 The system shall lock the user out after five failed attempts to log in (functional?)

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

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

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

2. Performance

    2.1 Pages shall load within one (1) second.

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

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

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

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

3. Maintainability

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

    3.2 Maintenance shall be kept as short as possible.

    3.3 Testing shall be performed regularly.

    3.4 The system design have to be easy to maintain.

    3.5 The system design have to be easy to maintain.

4. Usability

    4.1 Pages shall be easy to navigate.

4.2 The system shall run without affecting other applications.

5. Data Integrity

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

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

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.)

7. Support

7.1 The system shall support English.

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

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

7.4 The system shall support PayPal as a payment method.

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

8. TeamWork

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

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

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

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

Competitive Analysis

Stanford Axess

Axess is a student online portal for enrolling in classes, checking grades and degree progress, seeing your financial aid awards and tuition changes, filling out online course evaluations, and many other things. The advantage of Stanford Axess is the high integration. Students can base on the previous average GPA for the course and the review of the professor to choose the best professor for the course. The Stanford Axess has a nice UI to display the class schedule to help students not select the classes with an overlapping schedule. The Stanford Axess is integral with lots of campus resources such as housing, and a health center. The disadvantage of Stanford Axess is that only Stanford students can access it. According to the Stanford admission office, there is only a 5% acceptance rate in the amount of 55,471 applicants. The data shows only a few elites can use the Stanford Axess which means the cost of using Stanford Axess is money and intelligence. The second disadvantage of Stanford Axess is the server will crash on the class enrollment date. This is a serious problem because students are not able to enroll in classes, pay for tuition, or do any other important things. A university student center should be considered to handle high network traffic on the enrollment date.

SFSU Student Center

The SFSU enrollment system is called the Student Center. One of the main advantages is that there is a broad range of important and useful information that is presented to the student upfront. The sidebar on the right is used to notify students of important details, including Holds & Alerts, To Do List, Important Links, etc. Additionally, the two main sections in the middle, both have their most prominent element showing data relating to the current or near future, with the Academics section showing the This Week's Schedule, and the Finances section showing the Outstanding Charges.

The biggest flaws of the SFSU Student Center are its organization and UI. These combined together greatly hinder the user's experience and negatively affect the usability of the site. Outside of the Student Center, the other pages have very different UI patterns, which makes it seem inconsistent and creates a very confusing and difficult navigation experience. There is a very poor use of horizontal space, which leads to excessive data compressed into the left side of the screen, which also leads to unnecessary amounts of scrolling up and down, for the student to find what they're looking for. The deeper you go into the website, often the more horizontal headers will appear at the top of the screen, which is not only a poor use of space but is also confusing when the student is changing scope and these nav bars are constantly appearing and disappearing as the student moves between layers of the website. An example of this kind of confusion is when a student clicks on the "Home" link that is at the top right of the nav bar that is second from the top (these 2 nav bars seem to almost always be present). The student may think they're going back to the Student Center since that's the first page they loaded into, but instead, they are sent to the "Homepage" which seems to be a semi-customizable page that by default

contains a menu that contains the same navigation options as the nav bar that is first from the top. In order to get back to the Student Center, the student would have to navigate this menu (or the redundant nav bar) and find the Student Center inside the Self-Service drop-down section.

Another big flaw is performance. The site loads pretty slowly, and the loading symbol that sometimes shows in the upper right, is often hard to notice, which can lead to students not knowing whether their inputs are being registered. Keep in mind, that it was previously stated that the layout doesn't populate the right side of the screen, so the user isn't conditioned to notice things popping up there, so it is very easy for them to miss it. This loading symbol also doesn't always show up, which is an inconsistency that is also confusing.

There is also often a lack of navigation, even where there are good opportunities for it. An example of this is searching for classes. After performing a search, a student will see a list of course sections that fit the search criteria, with each section also showing some of its details (not all of its information though, e.g. grading options, # of credits). This is a missed opportunity for increased interaction, wherein a student could click on one of these attributes and immediately navigate to a page relating to that attribute (e.g. click on the professor's name and immediately be transported to that professor's bio page, or even a page listing all classes taught by that professor).

The flip side to the above is that the website also has cases of excessive navigation. Often when the student wants to do something, the site will ask the student to confirm their decision but will do so by navigating to a completely separate confirm page, instead of the

traditional popup element. This is not just inefficient, it is also extremely costly to the student, as they have to reorient themselves every time this happens, and also cannot rely as easily upon the browser's built-in go-back and go-forward features, since they often don't lead to the expected page. For whatever reason, the go-back feature also often leads to a page that claims that the page is no longer available, and forces the student to return to the most recent active page. This is confusing and shouldn't happen.

MyOSU

MyOSU is the student center that Oregon State University utilizes. Advantages include the student DashBoard being more organized. The Student Board for Organstate is a lot more organized than the SFSU student dashboard. You are able to access more of the important information without having to use a scroll menu.

The main page gives easy and direct access to your student information without the need to use a scroll menu to go to certain specific pages. They gain easy access to canvas, student records, schedule of classes, the overall overview of their standing, and if the student has any fees that have not been paid yet. Even though their student center doesnt make it so you can hide some information, it is more organized and shows the more prioritized items at the tops of the list and the information that is not as prioritized on the bottom of the list.

One of the main disadvantages to this school's student center would be how most of the stuff for this student center is all there. What I mean is that even though it gives easy access to the student records and class scheduling some of the features could be combined to make it easier to search for what you need. Another thing is that for some who have never used a student center it might be a bit too much to take in because of all the information that is being given to that student.

The main advantage of the MyOSU student center is that access to all the student information is much more convenient and very pleasant to look at because of how organized the whole student center is compared to MySFSU. The SFSU student center is not as organized as the OSU student center. Our for the new student center is to organize

our school information as OSU did by prioritizing our student resources by showing the class schedule for the week on the front of the page and on the sides; we will show you student resources, i.e., student records, scheduling, and everything else that we would usually see when looking at our student resources. It also allows us to prioritize necessary student resources and those that are not as necessary.

One thing that stands out about the OSU student center is how organized they are but separating each section into blocks by showing the student's academic overview on the front of the page. Seeing how you are doing can help students preserve and try to push themselves to do better than they are doing at this point. It is also nice because the overview lets the student know if they have any fees they have to pay and what their academic standing is. This student center integrates the student dashboard and provides easy access to the canvas without having to leave the student center. Another thing to add is how accessible the student center is with a tab for your academics and finance. Accessing these without the need for drop-down menus clears up more space for other sessions and helps with organization. The organization of this student center makes it more user-friendly and straightforward, so you will not be as lost if you never used a student center before.

WebSMART

WebSMART is the student center used for the San Mateo County Community College District, which is composed of Cañada College, College of San Mateo, and Skyline College. The homepage for WebSMART is a lot less cluttered than the SFSU student center. All of the main services offered are clearly displayed. This is contrary to the SFSU student center, where there are dropdown menus to find other services.

The home menu of WebSMART shows the main features. These include student services, financial aid information, and student profile information. Student services include registering for classes, student records, degree progress, and making appointments among other things. The financial aid services offered include viewing your financial aid status and eligibility. Other services related to updating your personal information.

The main advantage of WebSMART is its simplicity. Students can quickly and easily find what they're looking for. As mentioned before, the home page displays the main services offered, allowing students to quickly navigate to what they need. For instance, for anything class-related, students can click on the student services tab, where all the services offered are listed with a short description of what each service does. This is one feature that makes WebSMART stand out from the SFSU student center. On the SFSU student center, certain features such as viewing your degree progress can be hard to navigate to, as it isn't directly listed. You have to go searching for it by opening the dropdown menu under academics, as opposed to WebSMART, where it is directly listed under student services. Furthermore, all of the services are clearly listed without the need of scrolling down the page or clicking any dropdown menus.

One disadvantage of WebSMART is that a lot of the services offered can be overwhelming. For example, under student services, there is a plethora of options ranging from registering for classes to ordering parking permits and updating emergency contact information. WebSMART could be improved by dividing all of the services they offer into more subcategories. Services like updating emergency contact information could be placed in the “My Profile” section. In addition to being overwhelming at times, the front-end design of WebSMART leaves a lot of blank white space, especially on the home page.

A characteristic of WebSMART that makes it stand out from other student centers is that its class search feature is handled through another website called WebSchedule. Students are redirected to WebSchedule when looking up classes to add through WebSMART. WebSchedule allows students to look up classes at Cañada College, College of San Mateo, and Skyline College. Similarly, the degree progress service is handled by a separate tool called DegreeWorks.

My UCSC

MyUCSC is the student center hub utilized by UC Santa Cruz. Advantages include being able to use the back button without being prompted to return to the page you were previously at before pressing the back button when trying to navigate to a new page. The load times are much faster than other student centers such as SF State and seem to be more responsive to user requests.

On the main page of the student center, there is a clear and direct emphasis on what aspects of the student center are important. The important pages that a student would likely go to are placed next to each other. Pages such as Search, Plan, Enroll, Grades, and My Academics are the first things placed on the page. The other lesser important parts are placed in a drop-down part to prevent cluttering and information overload.

The class search has a much nicer UI, with a dedicated page to it that is centered and more aesthetically pleasing. As for the functionality of the search, there is an emphasis on being able to decide which format of classes you wish to participate in. There are checkboxes to choose whether or not you want to take Asynchronous, Online, Hybrid, Synchronous, Online, or In Person classes. These are important aspects that the UCSC class search emphasizes in a post-covid-19 world.

The main advantage that UCSC has over its competitors is the ability to save previous states of the webpage. You are able to use the back button to load a previous page you were on and immediately interact with it. The current SFSU student center does not allow you to do this. Our student center plans for us to be able to also save the state of the

pages we are on and be able to go back to previous pages and have that be the current page state we are on.

One thing that is notable about UCSC is that they have two different forms of login identification. There is a “Gold login” and a “Blue login”. Gold login is similar to a central hub for the user. You can enroll, add classes, manage finances, check grades, plan your degree, view messages, and manage financial aid. This version of their student center has less security and is mainly for basic management of the school environment. “Blue login” is where most of the higher security management takes place. When you want to pay for finances, view medical details, or access financial aid, you must first log in using your “Blue login” credentials. This acts as a sort of a 2FA version of security. This form of security creates a separation between the day-to-day environment of the “Gold login” and the “Blue login” which you will likely only need to do every so often, primarily when making big decisions or paying finances at the beginning of the semester.

The uniqueness of our project

The innovation of our new student center is to bring professor/course ranking to the public universities. In our research, only the private university like Stanford provide professor ranking system to students. The second innovation is our new student center has a modem design UI. In our research, most of the student center in public university like UC and CSU have ancient design. For example, the text is small, and the images doesn't align. Our professor/course ranking system provides a comment section to the professor. That give more reference for the students who are considering taking the course/professor. Our new UI design makes the student center easy to use because it is design for user usability. Student won't waste time look for opening a class selection page. They can finish adding class to their schedule within 10 mins.

Identity Important Features Table

Feature/C	SFSU Student Center	Stanford Axess	MyOSU	WebSMART	MyUCSC
Strengths	Important information is prioritized and noticeable.	High integration, clear and modem UI design, adapt to mobile and desktop devices.	Well organized and prioritizes the student's needs	Simple and easy to use. Important services are clearly displayed on the home page.	Fast loading times and is able to navigate between pages seamlessly.Has a high level of security, with two different passwords for different areasHas easy filtering for online, in-person, synchronous, and asynchronous learning
Weaknesses	Very badly organized. Slow and confusing to navigate. Poor use of space, excessive scrolling, and loading. Lots of redundant pathways and confusing extra features.	The server will crash at the class selection date. The website won't tell the user it's crashed but some of the functions are not able to use.	Some of the resources could be grouped together making a list smaller	The services offered aren't organized well, which can cause confusion. A lot of unused white space.	Does not have the greatest UI. Still has the problem of wasted whitespace. Hard to directly access counselor appointments and jump between pages that utilize different passwords
Pricing	Free to Students and Faculty	Free to Students and Faculty	Free to Students and Faculty	Free to Students and Faculty	Free to Students and Faculty
Social	No sources of Social Media	No sources of Social Media	No sources of Social Media	No sources of Social Media	No sources of Social Media
Onboarding experience	Few steps but long load times sometimes.	Moderate number of Steps	Smooth Instructions	Moderate number of Steps	Fast loading times and seamless transition between pages.

Typical Competitive Features Table

Feature	SFSU Student Center	Stanford Axess	MyOSU	WebSMART	MyUCSC	Our Product
Usability/UI	+	+	+	+	+	++
Class Schedule	+	+	+	+	+	++
Course Shopping Cart	+	+	+	+	+	+
Degree Progress/Transcript	+	+	+	+	+	+
Counselor Appointment	-	+	+	+	-	+
Class Search Feature	+	+	+	+	+	+
Course Management (Canvas/Ilearn)	-	+	+	-	+	+
Financial Aid	+	+	+	+	+	+
Professor Ranking System	-	+	-	-	-	+

High-level system architecture and technologies used

Server Host:	Microsoft Azure Standard B1s
Operating system:	Ubuntu 18.04
Database:	PostgreSQL
Web Server:	Nginx
Server-Side Language:	JavaScript
Front-End Framework:	React
Back-End Framework:	NodeJS

Check List

Team found a time slot to meet outside of the class	DONE
Github master chosen	DONE
Team decided and agreed together on using the listed SW tools and deployment server	DONE
Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing	DONE
Team lead ensured that all team members read the final M1 and agree/understand it before submission	DONE
Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)	DONE

List of Contributions

Name	**Score**	**Contribution**
Elisa Hsiao-Rou Chih	9	Revising the Document with Grammarly Worked on Use Case Worked on non-functional requirements Organized the functional and non-functional sections by groups and categories. Worked on about page
Steven Paul Fong	9	Worked on Use Case 6,9,10,11 Worked on Section: List of Entities Worked on functional requirement Worked on Competitive Analysis, building tables and competitor's research Worked on about page
Cameron Michael Yee	9	Worked on Use cases 1 & 2 Worked on Section: List of Entities Worked on functional requirements Worked on Competitive Analysis and competitor's research Worked on about page
Michael Harrison Chang	10	Worked on Use Case 3&5 Building Diagram for Use Case 15 Worked on Section: List of Entities Worked on Competitive Analysis, building tables and competitor's research Revising the Document with Grammarly Organizes the Document Worked on the non-functional requirement. Worked on functional requirements Worked on about page
Christopher Alan Yee	9	Worked on Use Case 7,8,15 Analysis, building tables, and competitor's research Worked on functional requirements Worked on about page
Zhenyu Lin	9	Worked on Use Case 12,13,14 Draw diagram for half the Use Case Worked Analysis, building tables, and competitor's research Worked on about page

SW Engineering CSC648/848 Fall 2022

Project Name: New SFSU Student Center

Team Number: 05

Milestone 2

Date: 19 October 2022

Student Name	Roles	Email
Zhenyu Lin	Team Lead, GitHub Master	zlin4@mail.sfsu.edu
Christopher Alan	Backend Lead	cyee12@mail.sfsu.edu
Michael Harrison	Frontend Lead	mchang9@mail.sfsu.edu
Elisa Hsiao-Rou	Team Member	echih@mail.sfsu.edu
Steven Paul Fong	Team Member	sfong10@mail.sfsu.edu
Cameron	Team Member	cyee10@mail.sfsu.edu

Milestone/Version	Date
M1V1	09/21/2022
M1V2	10/5/2022
M2V1	10/19/2022

Contents

Data Definitions

1. Student - Our end users. Can enroll in course sections.
    1.1. Date Enrolled
    1.2. Month - Month in which they started enrollment
    1.3. Day - Day in which they started enrollment
    1.4. Year - Year in which they started enrollment
2. Transfer Student
    2.1. Current Year
    2.2. Transfer information - A student that came from another college. Has a special student record called ‘past transcripts’ for courses taken at other colleges.
        2.2.1. College Transferred From
            2.2.1.1. Name of previous college
            2.2.1.2. Dates they were enrolled at previous college
                2.2.1.2.1. Month - Month in which they started enrollment
                2.2.1.2.2. Day - Day in which they started enrollment
                2.2.1.2.3. Year - Year in which they started enrollment
    2.3. Dates Enrolled
        2.3.1. Month - Month in which they started enrollment at current university
        2.3.2. Day - Day in which they started enrollment at current university
        2.3.3. Year - Year in which they started enrollment at current university
3. Undergraduate Student - A student that has not yet graduated with a degree.
    3.1. Dates Enrolled - When the student started enrollment
        3.1.1. Month - Month in which they started enrollment at current university

3.1.2. Day - Day in which they started enrollment at current university

3.1.3. Year - Year in which they started enrollment at current university

4. Graduate Student - A student that has graduated with a degree.

4.1. Previous University

4.1.1. Name - Name of previous university

4.2. Dates Enrolled

4.2.1. Month - Month in which they started enrollment at current university

4.2.2. Day - Day in which they started enrollment at current university

4.2.3. Year - Year in which they started enrollment at current university

4.3. Graduate Program

4.3.1. Name of graduate program

5. Full-Time Student - A student that is taking enough units that would be equivalent to a full-time job.

5.1. Credits - Amount of credits being taken

6. Part-Time Student - A student that is taking less units than would be equivalent to a full-time job.

6.1. Credits - Amount of credits being taken

7. Personal Information

7.1. Name

7.1.1. First Name

7.1.2. Middle name

7.1.3. Last name

7.2. Address

7.2.1. City

7.2.2. State

7.2.3. Country

7.2.4. Zip Code

7.3. Age

7.3.1. DOB

7.3.1.1. Month

7.3.1.2. Year

7.3.1.3. Day

8. Course (sometimes referred to as “Class”) - The main product for the student.

8.1. Course Information

8.1.1. Times - Lists the duration of class length

8.1.1.1. Start time - When class begins

8.1.1.2. End time - When class ends

8.1.2. Start Date - Month and day when class begins

8.1.3. End Date - Month and day when class ends

8.1.4. Instructor - Lists name of instructor

8.1.4.1. Last name

8.1.4.2. First name

9. Credit (sometimes referred to as “Unit”) - A unit of measurement for the time/effort of a course. A student needs a certain amount to graduate.

10. Course Section (sometimes referred to as “Section”)

11. Online Section

11.1. Times - Lists the duration of class length

11.1.1. Start time - When class begins

11.1.2. End time - When class ends

11.2. Start Date - Month and day when class begins

11.3. End Date - Month and day when class ends

11.4. Instructor - Lists name of instructor

11.4.1. Last name

11.4.2. First name

12. Synchronous Section

12.1. Times - Lists the duration of class length

12.1.1. Start time - When class begins

12.1.2. End time - When class ends

12.2. Start Date - Month and day when class begins

12.3. End Date - Month and day when class ends

12.4. Instructor - Lists name of instructor

12.4.1. Last name

12.4.2. First name

13. Asynchronous Section

13.1. Times - Lists the duration of class length

13.1.1. Start time - When class begins

13.1.2. End time - When class ends

13.2. Start Date - Month and day when class begins

13.3. End Date - Month and day when class ends

13.4. Instructor - Lists name of instructor

13.4.1. Last name

13.4.2. First name

14. Hybrid Section

14.1. Times - Lists the duration of class length

14.1.1. Start time - When class begins

14.1.2. End time - When class ends

14.2. Start Date - Month and day when class begins

14.3. End Date - Month and day when class ends

14.4. Instructor - Lists name of instructor

14.4.1. Last name

14.4.2. First name

15. Subject

15.1. Major - The

16. Time Slot (sometimes referred to as “Date & Time”) - An attribute of course sections that details when it takes place.

16.1. Building

16.2. Room Number

17. Time Slot (sometimes referred to as “Date & Time”)

17.1. Start Time

17.2. End Time

17.3. Duration

18. Component - An attribute of a course section describing the kinds of learning activity the student would be doing. (e.g. Lecture, Lab, Self-Study, etc.)
    18.1. Lecture - Class where a professors convey information to the student
    18.2. Lab - Interactive setting where students have hands-on experience with the course material through experiments.
    18.3. Self-Study - Self regulated study, with oversight by professors to determine if a student fulfilled credit requirements.
19. Course Materials - An attribute of a course section describing the kinds of materials the student may need to purchase to take the class (e.g. textbooks)
    19.1. Textbook
20. Major - An attribute of a student. Can determine what types of classes the student can take.
21. Student Record - A type of form that also acts as an attribute of students. Having certain completed records affects how the student can interact with the system.
22. Transcripts - A type of student record. A list of previously enrolled courses, and the grades received from them.
    22.1. Currently enrolled courses- Courses in which the student is currently taking
    22.2. Previously enrolled courses - Courses in which the student has already taken
    22.3. Grades - The course evaluation give to the student by the professor
    22.4. Semester - The time period in which the class was taken by the student
23. Student Calendar - A list of dates & time slots.

23.1. Important dates

23.1.1. Final day to withdraw without a W - Lists the final day a student may drop a class without receiving a W grade.

23.1.2. Final day to drop and receive a refund - Allows a student to drop a class and receive a refund for said credits.

23.1.3. Final day to change to CR/NC - Lists the date a student must change their grading option by

23.1.4. Payment due dates - Lists the day which all payments for tuition are due

23.1.5. Final day to add classes - Lists the final day in which a student can add a class without requiring a permission number

23.1.6. Final day to add a class via permission number - Lists the final day in which a student can add a class from a professor given permission number. After this day the student may no longer enroll in the class

23.1.7. Final day for faculty drops - Lists the final day that faculty may drop students enrolled in their course.

23.1.8. Final day to withdraw from classes or university - Lists the final day in which a student may exist from the university and receive a refund.

23.1.9. Holidays - Lists days in which there will be no courses taught, in which they normally would be

24. Class Schedule - A type of Student Calendar containing only the dates & times of the student's course sections.

24.1. Name of class - The descriptive name of the course

24.2. Course ID - The specific alphanumeric value of the course.

24.3. Section number - Lists the more specific class the student will be in for classes which have multiple time slots.

24.4. Time slot - Lists when the class will run from

24.5. Professor - Lists the instructor teaching the course

25. Appointments - Has a date & time, and acts as an element in a calendar, just like course sections.

26. Prerequisite - A type, of course, that is also an attribute for another different course. The prerequisite must be completed concurrently with, or before the student takes the associated other courses.'

26.1. Courses the prerequisite unlocks

26.1.1. Name of the course the prereq unlocks

26.1.1.1. Name of course

26.1.1.2. Course ID

27. Professor - The teacher of a section. Can modify course section information.

27.1. Courses offered - Lists all courses that the specified teacher will be teaching.

27.2. First name

27.3. Last name

28. Advisor - An employee of the university that specializes in helping students with school-related questions.

28.1. Name of advisor

28.2. Time slots - Times in which the advisor is available

29. College (sometimes referred to as “University”) - The client organization. Uses the system to offer courses to students.
30. Department - A subsection of a college that facilitates learning for specific subject(s).
31. Campus Clinic - Site that students can visit for health-related issues.
    31.1. Hours - Lists the times that the clinic is open
    31.2. Location - Lists where you can find the Campus Clinic
32. Student Club - A school-recognized student organization. Associated with a faculty advisor.
    32.1. Faculty advisor Lists the faculty advisor of the student club
    32.2. Name - Lists the name of the specific club
33. Parking Permit - A form that grants students access to parking.
    33.1. Location - Lists the locations the permit is valid in.
34. Bill - A form that shows students' transactions after purchase.
    34.1. Amount paid
    34.2. Amount remaining - Tells the user how much of their bill they have yet to pay
35. Charge - An element on a bill. A specific amount the student owes for a specific purpose.
    35.1. Name of charge - Tells the user what type of charge is being levied on their account.
    35.2. Value of charge - Tells the user how much they must pay
36. Aid Form - A form that shows students’ financial aid information.

36.1. Loan amount granted - Tells the user the amount of money they may receive from a loan in dollars

36.2. Loan amount accepted - Tells the user how much of the granted loan they have accepted in dollars

36.3. Grant amount - The amount quantity in dollars of a federal pell grant given to a student

36.4. Grant amount accepted - The amount of a federal pell grant the student has accepted in dollars

37. Map - An image showing the geographical layout of a location.

37.1. Format – JPG

38. Schedule - an itinerary that you follow throughout the day.

39. Notification - an alert that notifies the student of an important message.

40. Holds - A property that prevents students from enrolling until resolved.

40.1. Name of hold - Tells the student the name of a hold so they can identify the type of hold they have

40.2. Date

40.2.1. Date created - Tells the student when their hold began

40.2.2. Due date - Tells the student when they must clear the hold by without incurring any penalty

41. Search Parameter - An attribute or specific value of an attribute that is used to filter/narrow down elements from a full list of elements.

41.1. Session

41.1.1. Semester - Lists the semester which class is offered E.g Summer, Fall, Spring

41.1.2. Year - Year of the course which is being offered

41.2. Subject - Specifies and searches for only courses of specified major E.g Statistics

41.3. Course number - The numerical ID attribute of the course

41.3.1. Contains - Returns courses in which the value searched appears in the course number.

41.3.2. Greater than or equal to - Returns courses with course number greater than or equal to the search parameters

41.3.3. Less than or equal to - Returns courses with course number less than or equal to the search parameters

41.3.4. Is exactly - Only searched the courses with the exact string parameter.

41.4. Additional criteria

41.4.1. Start time - Specifies when class begins.

41.4.1.1. Between - Lists courses beginning between the times specified

41.4.1.2. Greater than - Lists courses with start times after the times specified, but not going further than midnight of the next day

41.4.1.3. Greater than or equal to - Inclusively lists courses with start times of the times specified, but not going further than midnight of the next day

41.4.1.4. Is exactly - Lists courses only which begin exactly when specified.

41.4.1.5. Less than - Lists courses beginning before the times specified, but not going further than midnight of the current day

41.4.1.6. Less than or equal to - Inclusively lists courses that begin before the times specified, beginning at midnight of the current day.

41.4.2. End time - Specifies when class ends.

41.4.2.1. Between - Lists courses ending between the times specified.

41.4.2.2. Greater than - Lists courses with end times after the times specified, but not going further than midnight of the next day

41.4.2.3. Greater than or equal to - Inclusively lists courses with end times of the times specified, but not going further than midnight of the next day

41.4.2.4. Is exactly - Lists courses only which end exactly when specified.

41.4.2.5. Less than - Lists courses ending before the times specified, but not going further than midnight of the current day

41.4.2.6. Less than or equal to - Inclusively lists courses that end before the times specified, beginning at midnight of the current day.

41.4.3. Days of week

41.4.3.1. Exclude any of these days - Does not search for courses that only fall on the precisely specified days.

41.4.3.2. Exclude only these days - Does not search for courses that have a session on each of the precisely specified days.

41.4.3.3. Include any of these days - Search for courses that have a session on each of the precisely specified days.

41.4.3.4. Include only these days - Only search classes that have a session on the specified day.

41.4.4. Instructor name

41.4.4.1. Begins with - Searches instructors whose names begin with the string specified

41.4.4.2. Contains - Searches if a string parameter is contained within the instructor's name.

41.4.4.3. Is exactly - Searches for a professor whose last name is exactly the same as the search parameter.

41.4.5. Course attribute

41.4.5.1. GE - General education, needed by the state to graduate.

41.4.5.2. Upper division - Courses in which a student must have a graduate standing to take.

41.4.5.3. Lower division - Courses in which a student must have a undergraduate or higher standing to take

41.4.5.4. Graduation requirement - Courses in which are necessary to graduate, not necessarily a GE class.

41.4.6. Course keyword - Searches for a common keyword, E.g Computer science

41.4.7. Maximum units

41.4.7.1. Greater than - Lists courses with credits greater than the amount of course credits specified.

41.4.7.2. Greater than or equal to - Inclusively lists courses with credits greater than the amount of course credits specified.

41.4.7.3. Is exactly - Lists courses which are exactly the amount of course credits specified.

41.4.7.4. Less than - Lists courses with credits less than the amount of course credits specified.

41.4.7.5. Less than or equal to - Inclusively lists courses with credits less than the amount of course credits specified.

41.4.8. Minimum units

41.4.8.1. Greater than - Lists courses with credits greater than the amount of course credits specified.

41.4.8.2. Greater than or equal to - Inclusively lists courses with credits greater than the amount of course credits specified.

41.4.8.3. Is exactly - Lists courses which are exactly the amount of course credits specified.

41.4.8.4. Less than - Lists courses with credits less than the amount of course credits specified.

41.4.8.5. Less than or equal to - Inclusively lists courses with credits less than the amount of course credits specified.

41.4.9. Location

41.4.9.1. On campus - Courses provided on the main campus

41.4.9.2. Off campus - Courses not provided on the main campus. May be provided at a satellite location or an Annex.

41.4.9.3. Country - Courses provided in a country foregin from the main university’s country.

42. Course Requirements - a course that is required in order to take the next course.
    42.1. Prerequisite - A course that must first be taken before attempting to take another course
43. Health Records - records that inform the school/doctor about one person's health information.
    43.1. Name
        43.1.1. First Name
        43.1.2. Middle name
        43.1.3. Last name
    43.2. Address
        43.2.1. City
        43.2.2. State
        43.2.3. Country
        43.2.4. Zip Code
    43.3. Age
        43.3.1. DOB
        43.3.2. Month
        43.3.3. Year
        43.3.4. Day
    43.4. Allergies - Lists any potential allergies the student has to prevent an allergic reaction.
    43.5. Medical conditions - Lists any medical conditions the student has.

Prioritized Functional Requirements

Priority 1:

1. 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 automatically dropped from waitlists of course sections that are of the same class as one that the student is fully enrolled in.
1.6. Students shall be automatically dropped from waitlists of course sections that cause time conflicts with a course section that the student is fully enrolled in.
1.7. Students shall be notified when they are dropped from a course section.
1.8. Students shall be able to search for courses.
1.9. Students shall be able to add courses to a shopping cart, prior to enrolling.
1.10. Students shall have transcripts.
1.11. Students shall have a class schedule.
1.12. Students shall not fully enroll in multiple sections that overlap on the same date & time slot.
1.13. Students shall have a student calendar, showing the student's class schedule and the college's academic calendar.
1.14. Students shall be able to drop course sections.
1.15. Students shall receive a Hold/Alert if they have overdue charges.
1.16. Students shall be notified whenever new Holds/Alerts are created on their account.
1.17. Students shall be dropped from courses if they have overdue charges after the set deadlines.
1.18. Students shall be able to access their student records (including transcripts and payment receipts).
1.19. Students shall be able to access their payment histories.
1.20. Students shall be able to request to change their major. (changed to a request, cause the school and departments have to accept the request)
1.21. Students shall receive a grade, upon completing a course.
1.22. Students shall enroll in courses with one of two grading options: CR/NC or Letter Grade.
1.23. Students shall be able to switch between grading options within certain date & time slots.
1.24. Students shall be able to view their financial aid.
1.25. Students shall be able to receive Financial Aid.
1.26. Students shall be able to leave feedback reviews for professors of course sections that the student has taken before.
1.27. Students shall be able to contact the department of their major.
1.28. Students shall be able to search for clubs at the university.
1.29. Students shall be able to upload their health records.
1.30. Students shall be notified of payment due dates.
1.31. Students shall be able to schedule advising appointments.

1.32. Students shall be able to schedule financial aid appointments.
1.33. Students shall be able to make an appointment with the university clinic.
1.34. Students shall be notified of upcoming appointments.

2. Course
   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.
3. 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.
4. 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.
5. 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.
6. Transcripts
   6.1. Transcripts shall list all courses taken in the past.
7. Searchs
   7.1. Searches shall have parameters, which filter the displayed courses.
      7.1.1. Searches can be filtered by student's eligibility to enroll in the course.

7.1.2. Searches can be filtered by 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.

Priority 2:

1. Student
   1.1. 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.
   1.2. Student calendars shall recommend alternative course sections in order to resolve date & time conflicts on the class schedule.
   1.3. Students shall be notified if a course section on their calendar has any of its attributes changed. (students should know if there's a change of professor or change of location)
   1.4. Students shall be notified when a change of major is fully processed, regardless of whether it is accepted or rejected.
   1.5. Students who've recently changed majors shall have access to resources for their new major.
   1.6. Students shall be able to swap one course for another.
   1.7. Students shall be able to pay for courses.
   1.8. Students shall be able to generate What-If Reports.
   1.9. Students shall be able to save their What-If Reports.
2. Transferring
   2.1. Transfer students can have a transfer credit report.
   2.2. Transfer students shall be able to search for other colleges' courses to check if they count as being transferred.
3. What-If Report
   3.1. What-If Reports shall show students the required classes for hypothetical change of major, degree, or other academic career choices.
4. Professor Reviews
   4.1. The system shall notify reviewers that their received grade will be displayed along with their anonymous review.
   4.2. Professor reviews shall have tags to help students parameterize searches when filtering for professors with certain teaching styles.

Priority 3:

1. Student

1.1. Students shall be able to purchase parking permits.

1.2. Students shall be notified if their permit purchase was approved.
1.3. Students shall be able to open Google Maps in the student center.

2. Payment
   2.1. PayPal shall be supported as a payment method.
3. Advising

   3.1. Advisors shall be able to schedule appointment time slots.

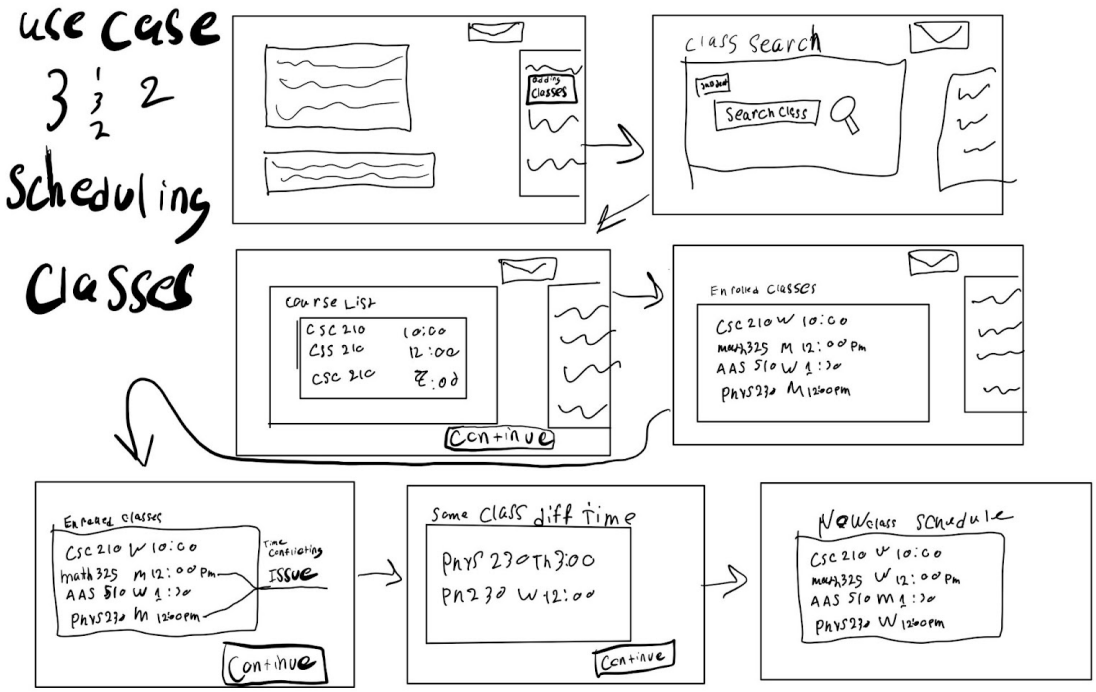
4. Clubs

   4.1. Clubs shall be displayed by the system.

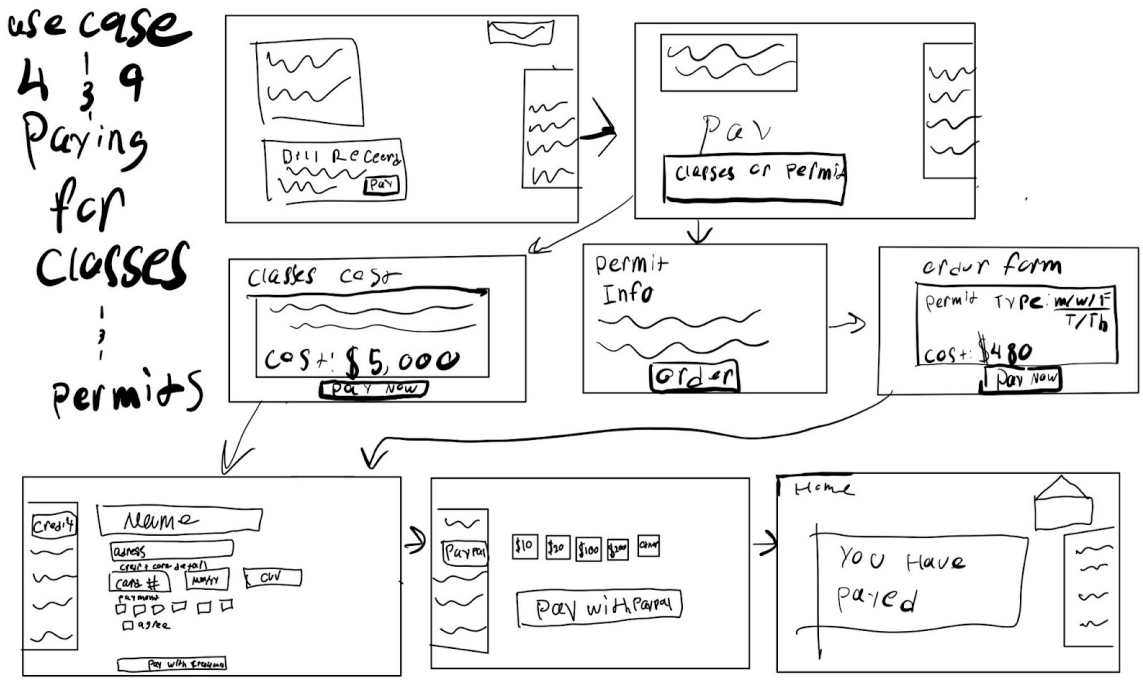
   4.2. Clubs can provide resources to the system for students to see.

UI Mockups and Storyboards (high level only)

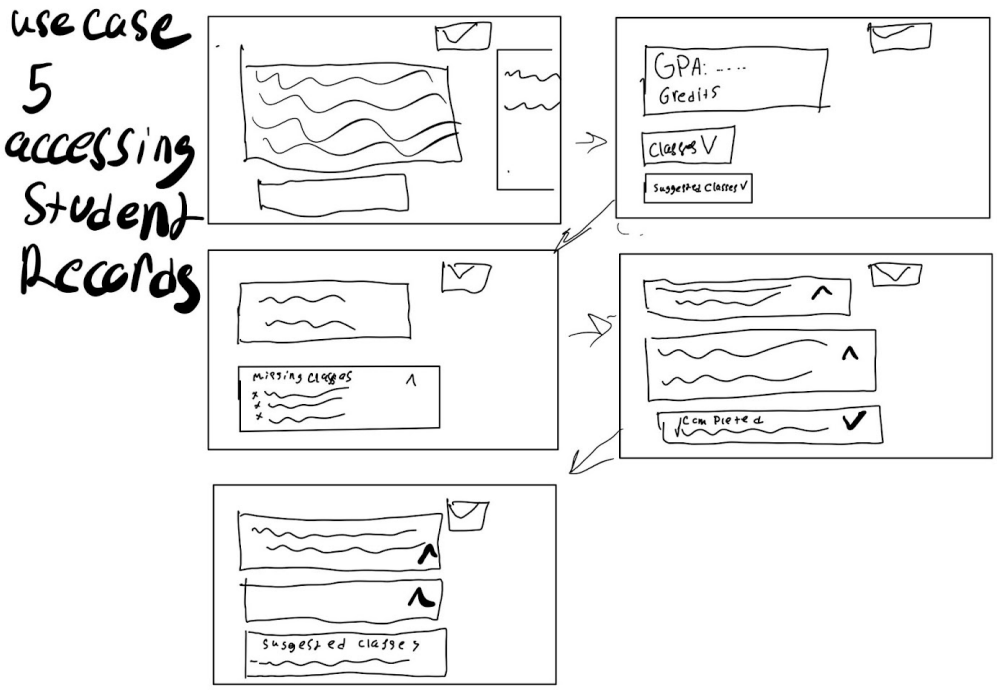
Use Case 2 and 3



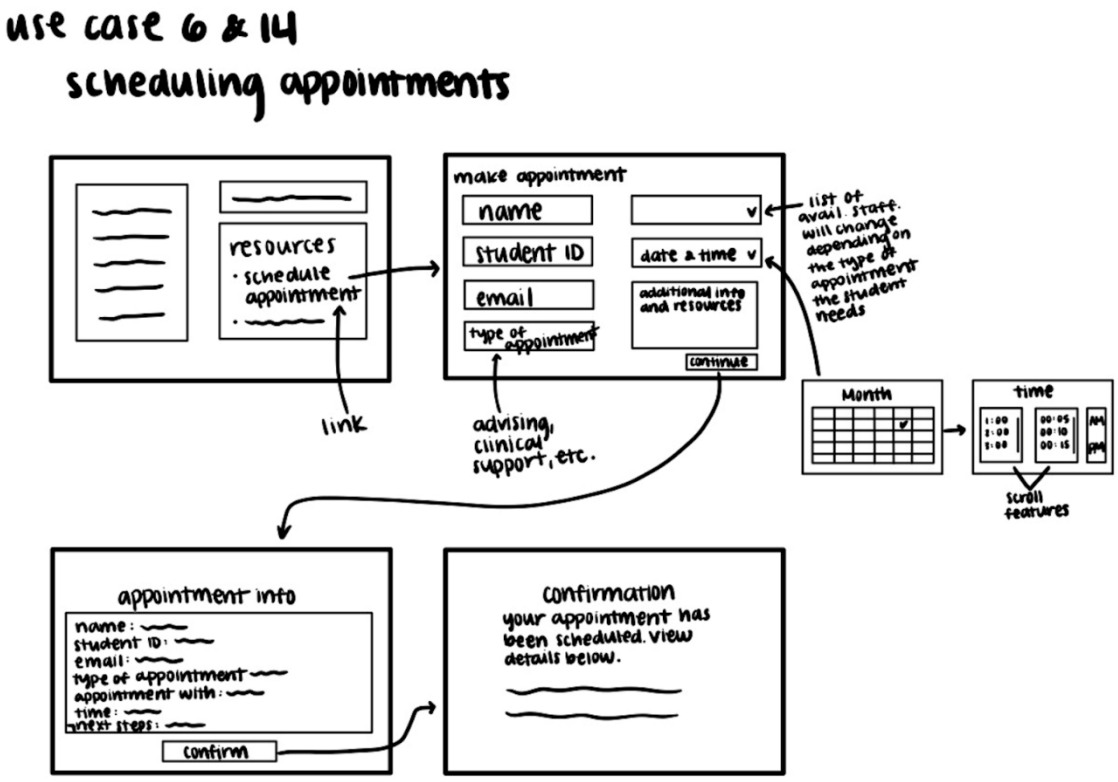
Use Case 4 and 9



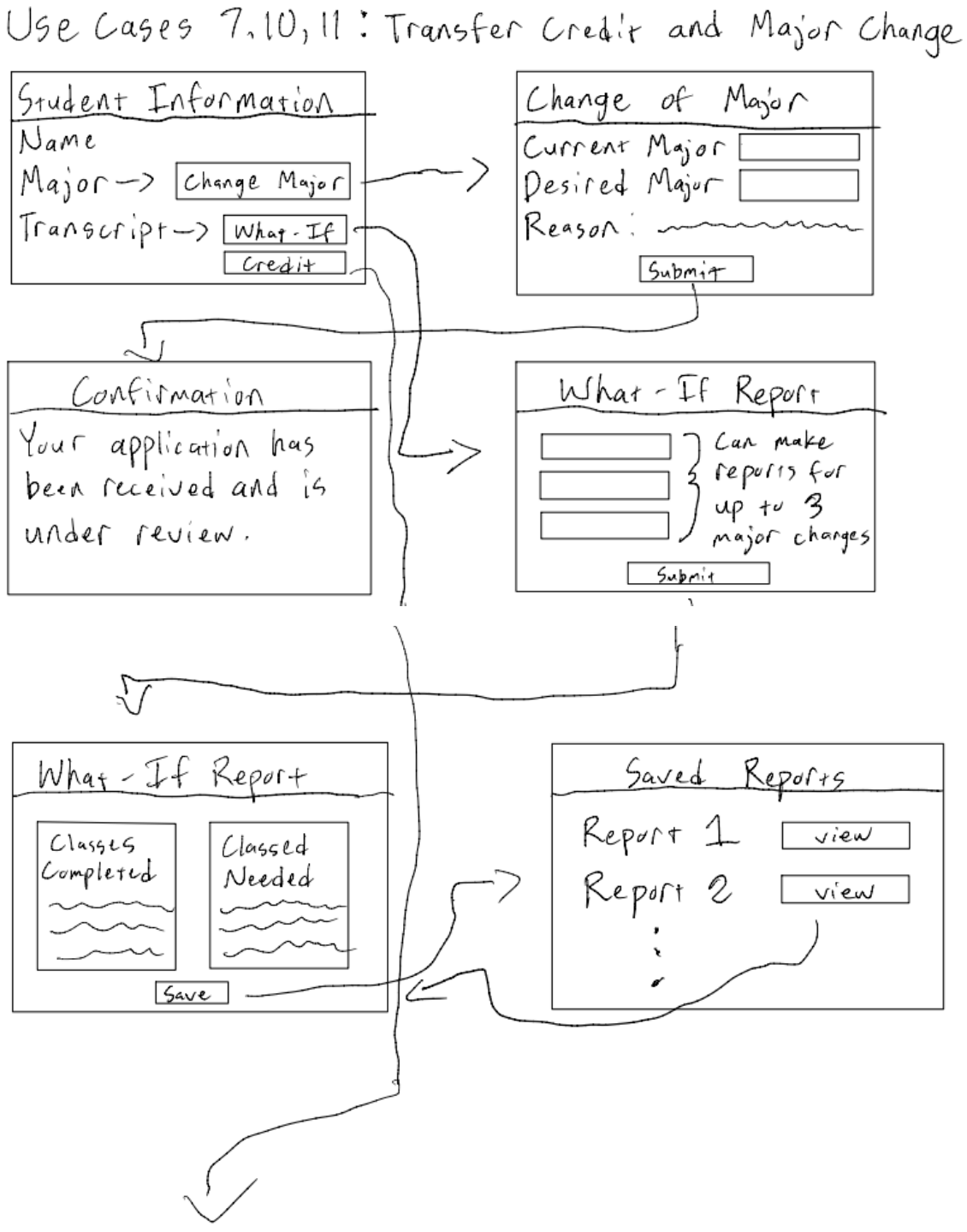
Use Case 5

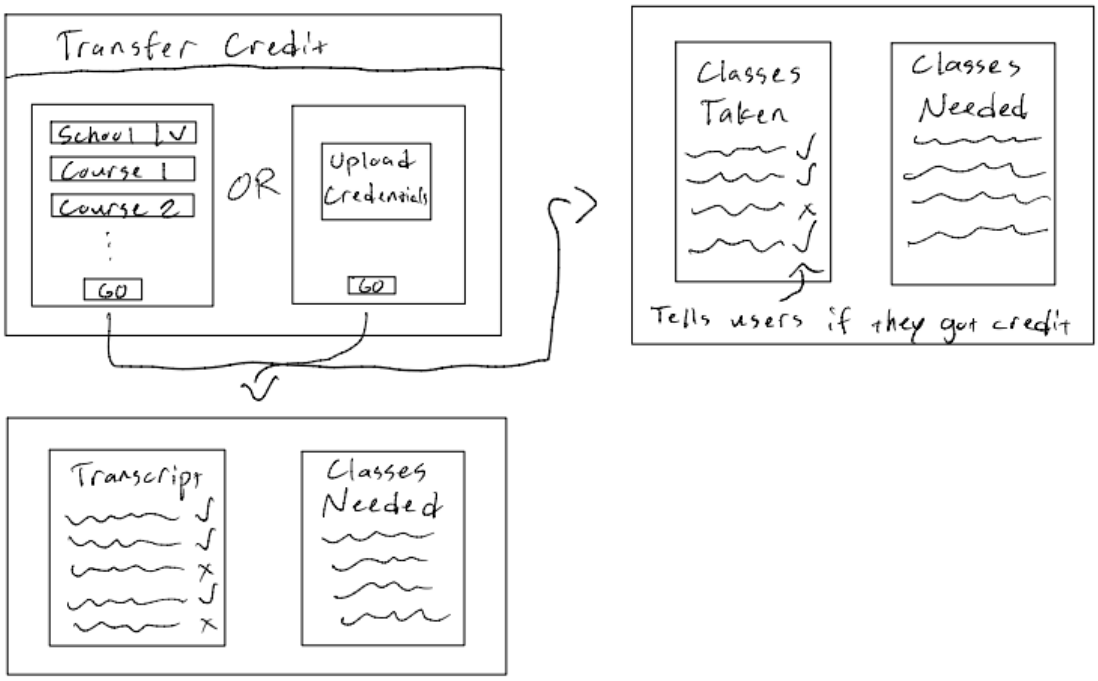

use case
5
accessing
Student
Records
GPA: ....
Credits
Classes V
Suggested Classes V
Missing Classes
Completed
Suggested classes

Use Case 6 and 14

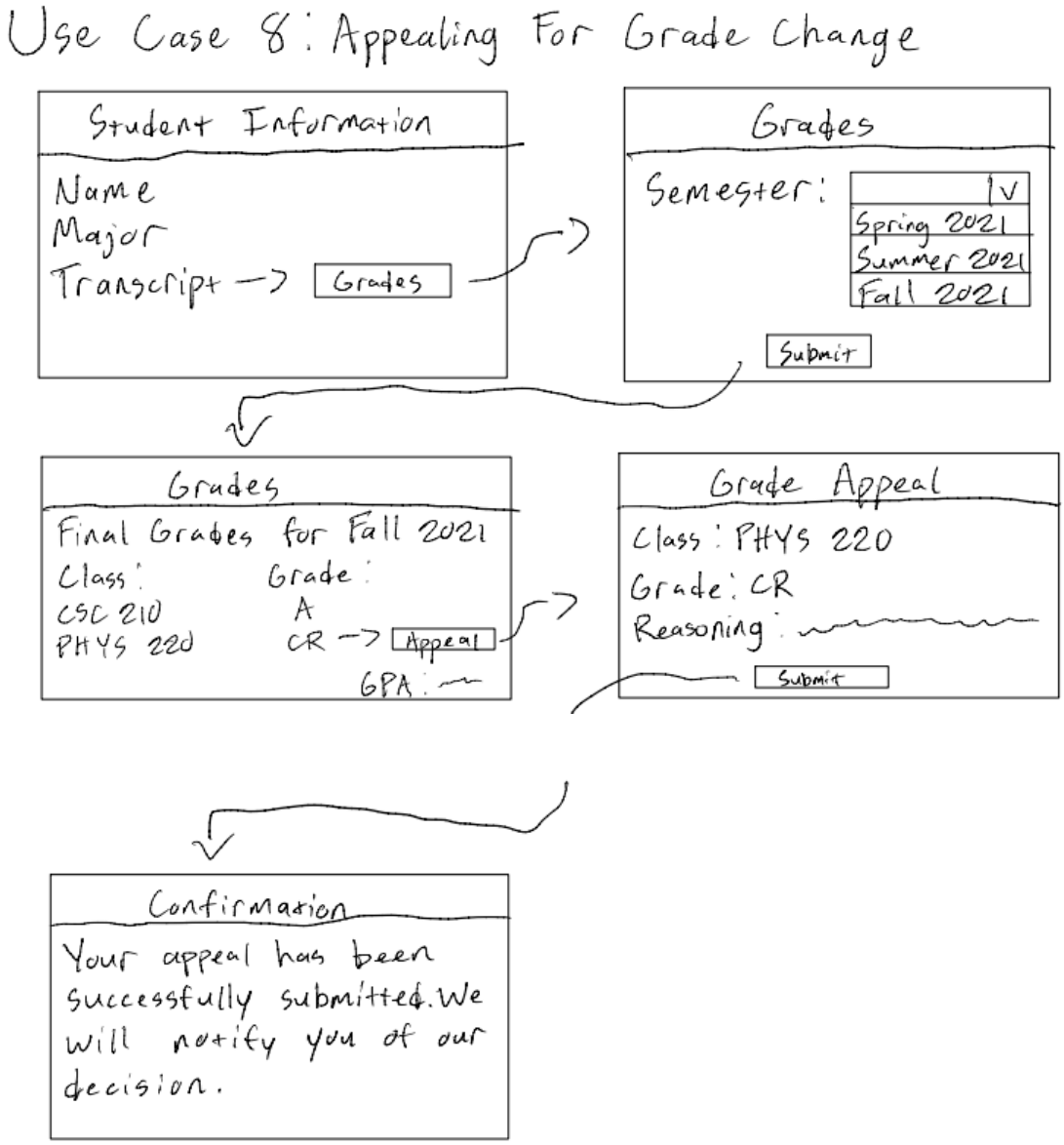


Use Case 7, 10 and 11

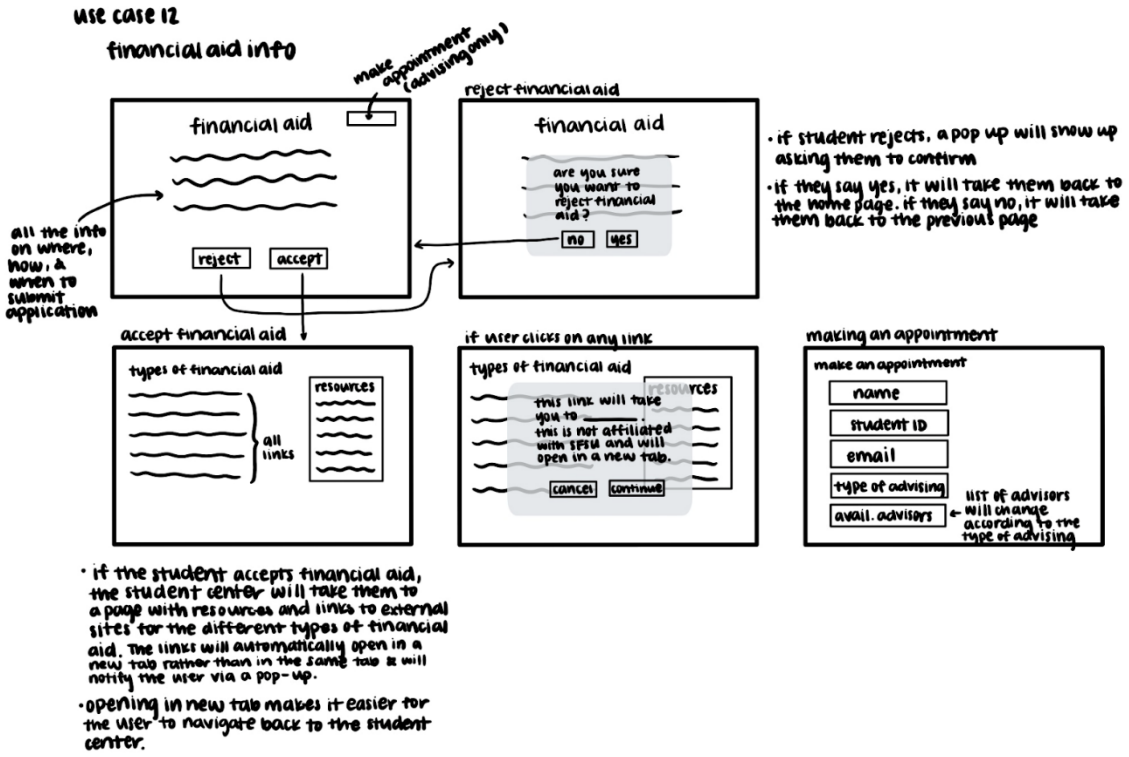



Transfer Credit
School
Course 1
Course 2
GO
OR
Upload Credentials
GO
Classes Taken
Classes Needed
Tells users if they got credit
Transcript
Classes Needed

Use Case 8


Use Case 8: Appealing For Grade Change
Student Information
Name
Major
Transcript -> Grades
Grades
Semester:
Spring 2021
Summer 2021
Fall 2021
Submit
Grades
Final Grades for Fall 2021
Class:
Grade:
CSC 210
A
PHYS 220
CR -> Appeal
GPA:
Grade Appeal
Class: PHYS 220
Grade: CR
Reasoning:
Submit
Confirmation
Your appeal has been successfully submitted. We will notify you of our decision.

Use Case 12

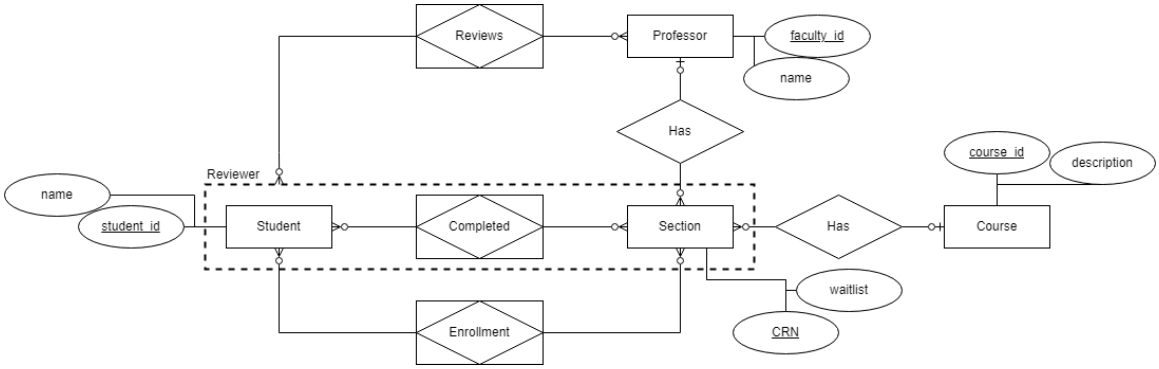

use case 12
financial aid info
make appointment (advising only)
financial aid
all the info on where, how, & when to submit application
reject
accept
reject financial aid
financial aid
are you sure you want to reject financial aid?
no
yes
• if student rejects, a pop up will show up asking them to confirm
• if they say yes, it will take them back to the home page. if they say no, it will take them back to the previous page
accept financial aid
types of financial aid
all links
resources
• if the student accepts financial aid, the student center will take them to a page with resources and links to external sites for the different types of financial aid. the links will automatically open in a new tab rather than in the same tab & will notify the user via a pop-up.
• opening in new tab makes it easier for the user to navigate back to the student center.
if user clicks on any link
types of financial aid
this link will take you to ____ this is not affiliated with SFSU and will open in a new tab.
cancel
continue
making an appointment
make an appointment
name
student ID
email
type of advising
avail. advisors
list of advisors will change according to the type of advising

High level database architecture and organization

Database Requirements:

1. Student
   1.1. A student can enroll in many course sections.
   1.2. A student can complete many course sections.
2. Course
   2.1. A course can have many course sections.
3. Course Section
   3.1. A course section belongs to one course.
   3.2. A course section can enroll many students.
   3.3. A course section is taught by one professor.
   3.4. A course section can have been completed by many reviewers.
4. Reviewer
   4.1. A reviewer is a student.
   4.2. A reviewer has completed many course sections.
   4.3. A reviewer can review many professors.
5. Professor
   5.1. A professor can teach many course sections.
   5.2. A professor can be reviewed by many reviewers.

ERD



The DBMS we will be using is postgreSQL. We are choosing to use postgreSQL because it is reliable, is a relational database and because it is a general purpose OLTP database.

Media storage Strategy:

Our system will store the media file in a file system. The database will store the location of media. One of the unique functionality of our project is professor ranking system. Students can rank their professor and give them feedbacks. Therefore, the most of storage content for our project will be text. Majority of our data will be store on the database. Although our first priority functional requirement has the majority of text data, our second and third functional requirements need to store large files such as student transcript pdf version and school club icons. The database will take huge amount of memory for backend to acquire the data from the database if we choose to encode the large file and store them into database .To effective use the memory and save our database bandwidth, we choose to store the large file like media file or pdf in a file system.

Search/filter architecture and implementation:

In our project, we will have one search bar that across all the pages. Therefore, the search function should be search across all of our database table. We will ask the user about the search category first. Then let them to type in the search content. The backend will base on the category and send the searching query to the database. For example, if the user want to search about the ranking for a specific professor, the user would first choose the searching category - ranking. Then type in the search content. The backend will first process the content by an algorithm. If the user type in the course number, the backend will send an query to the course table to get the professor id and check the professor ranking. Our search strategy is to send multiple search queries to different tables in the database. Then we will combine the search results and return to the front. We will ORM to manipulate data from the database. The benefit of ORM is that it can save lots of time in development. ORM avoids to write poorly-formed SQL and a lot of stuff is done automatically. Therefore, ORM significant reduce the development time.

High Level APIs and Main Algorithms

GET

/home

This API is used for querying the school's information. The front end sends the http request to the backend via this API. The backend will return the relevant school information.

/course

This API is used for querying all of our course information. The front end sends the http request to the backend via this API. The backend will return the relevant course's information.

/university_calendar

This API is used for querying the university calendar. The front end sends the http request to the backend via this API, and the backend returns the university calendar info to the frontend.

/professor

This API is used for querying a professor's profile page. The front end sends the http request to the backend via this API. The backend returns the professor's data, including ranking data, course sections the professor teaches, and any applicable reviews.

POST

/login

This API is used for the login function. The front end sends the user login info to this API by POST method. The backend will compare the given information with the relevant information stored in the database to see if the login is valid, then it will return the login status to the frontend to indicate whether the login was successful or not.

/register

This API is used for the registration function. The front end sends the registration info to this API by POST method. The backend will check if the information is all valid, then it will return the registration status to the frontend to indicate whether the registration was a success or not.

/review

This API is used for the review function. The front end sends the user's review about the course and the professor to this API by POST method. The backend will insert the review into the database along with its tags, the date created, and the grade that the reviewer received for the course, and then return the comment status to the frontend to indicate whether the comment operation was a success or not.

/enrollment

This API is used for enrolling in courses. The front end sends the course id, user id, and login status to the backend via this API. The backend will return the enrollment status to the frontend to indicate whether the enrollment success or not.

/shopping_cart

This API is used for querying user information. The front end sends the user identification and login status to the backend via this API, and the backend returns the corresponding shopping cart info to the frontend

/search

This API is used for the searching function. The front end sends the search category and the keyword to this API by the POST method. The backend then has to use database queries to obtain results of adequate similarity to the keyword. It will first decide which tables to search through depending on the chosen search category, then will use SQL and regex to search those tables for the specific elements.

/transcripts

This API is used for querying user information. The front end sends the user identification and login status to the backend via this API, and the backend returns the corresponding transcript info to the frontend.

/class_schedule

This API is used for querying the users' class schedules. The front end sends the user identification and login status to the backend via this API, and the backend returns the corresponding class schedule info to the frontend.

/notification

This API is used for querying the notification that the school sends to the users. The front end sends the user identification and login status to the backend via this API, and the backend returns the corresponding message to the frontend.

/checkout

This API is used for querying the total amount of tuition for the courses that the user has chosen. The frontend sends the user identification, login status, and course IDs to the backend via this API. The backend will add the student to the course, check if the enrollment caused any conflicts with other enrolled courses, create a tuition charge, and then return the corresponding price to the frontend.

/health

This API is used to keep track of the health status of the user. The frontend sends the user identification and login status. And the health info that's been entered by the user and clinical doctor to the backend via this API. The backend will return the operation status to the frontend to indicate whether the operation succeeded or not

High-Level Algorithm

We design a professor ranking system based on students' feedback and the average grade of the courses. At the end of the semester, the students need to fill out a survey form about their class experience. The survey contains the following categories: Professor's explanation skills, Professor's class management, and Professor's teaching style and class feedback. The score of this will be posted on the professor's profile page and will be used to calculate the overscore of the professor. This ranking strategy can help students choose the most suitable professor for them. In addition, we also post-class feedback to help students succeed in class.

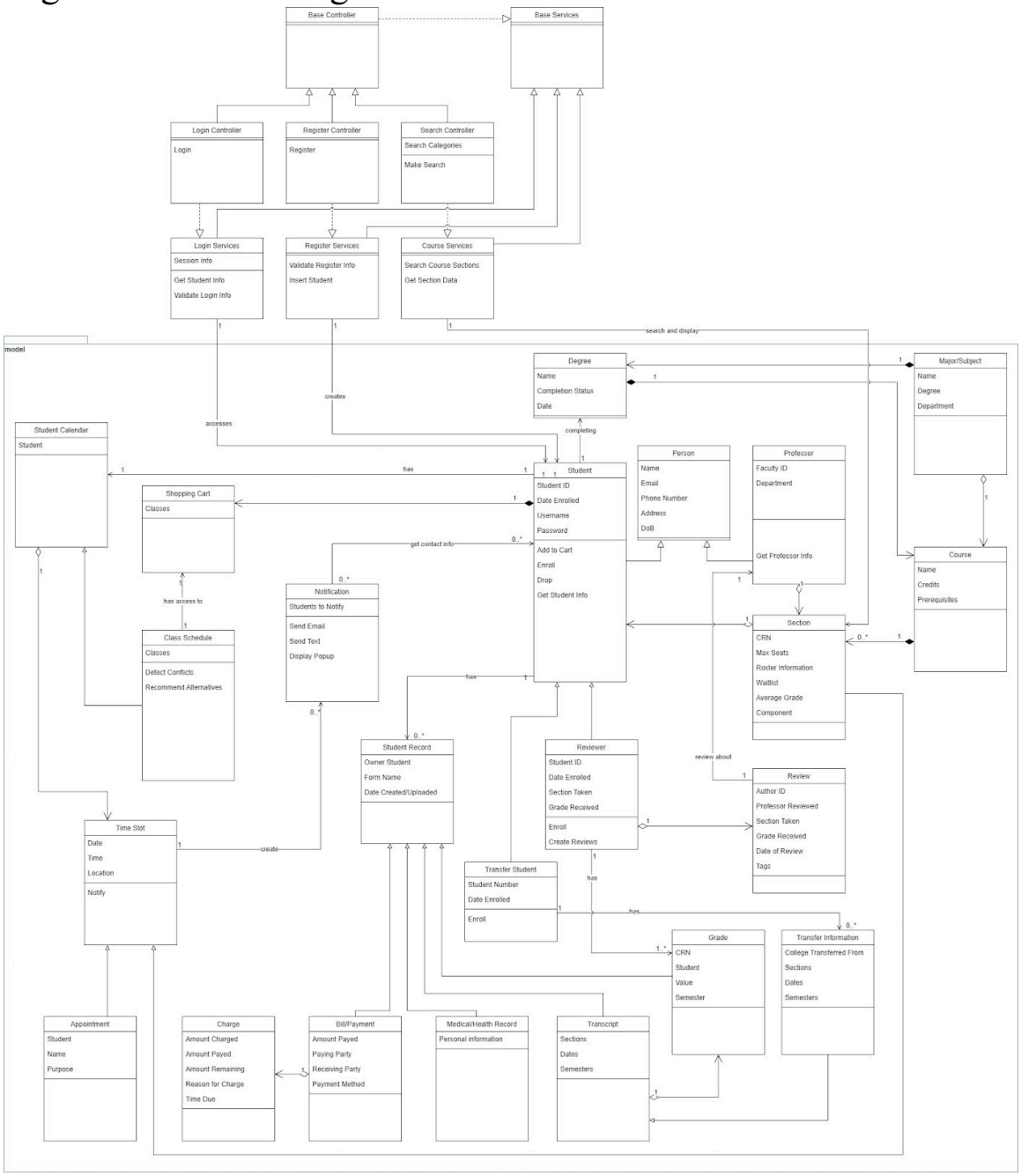
Other SW Tools

We are using Sequelize ORM as an alternative for interacting with the database to avoid using raw SQL.

express-fileupload is used for uploading and transferring files such as transcripts to the backend.

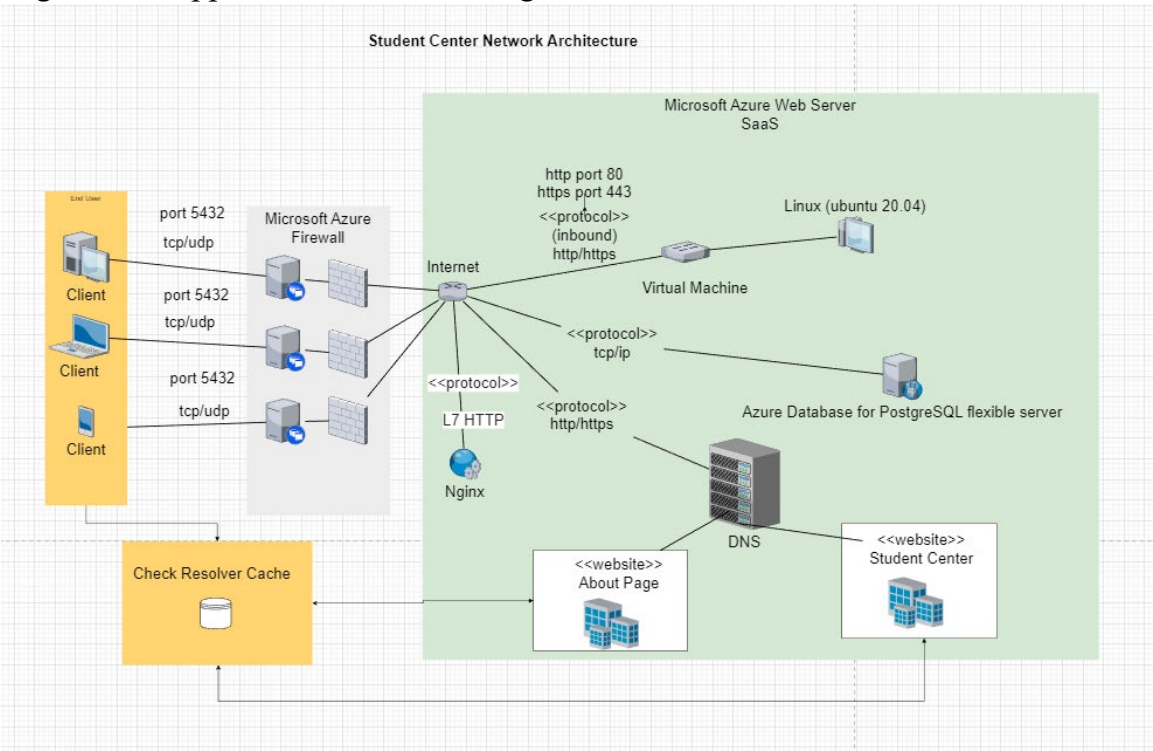
bcrypt is used for encrypting passwords.

High Level UML Diagrams



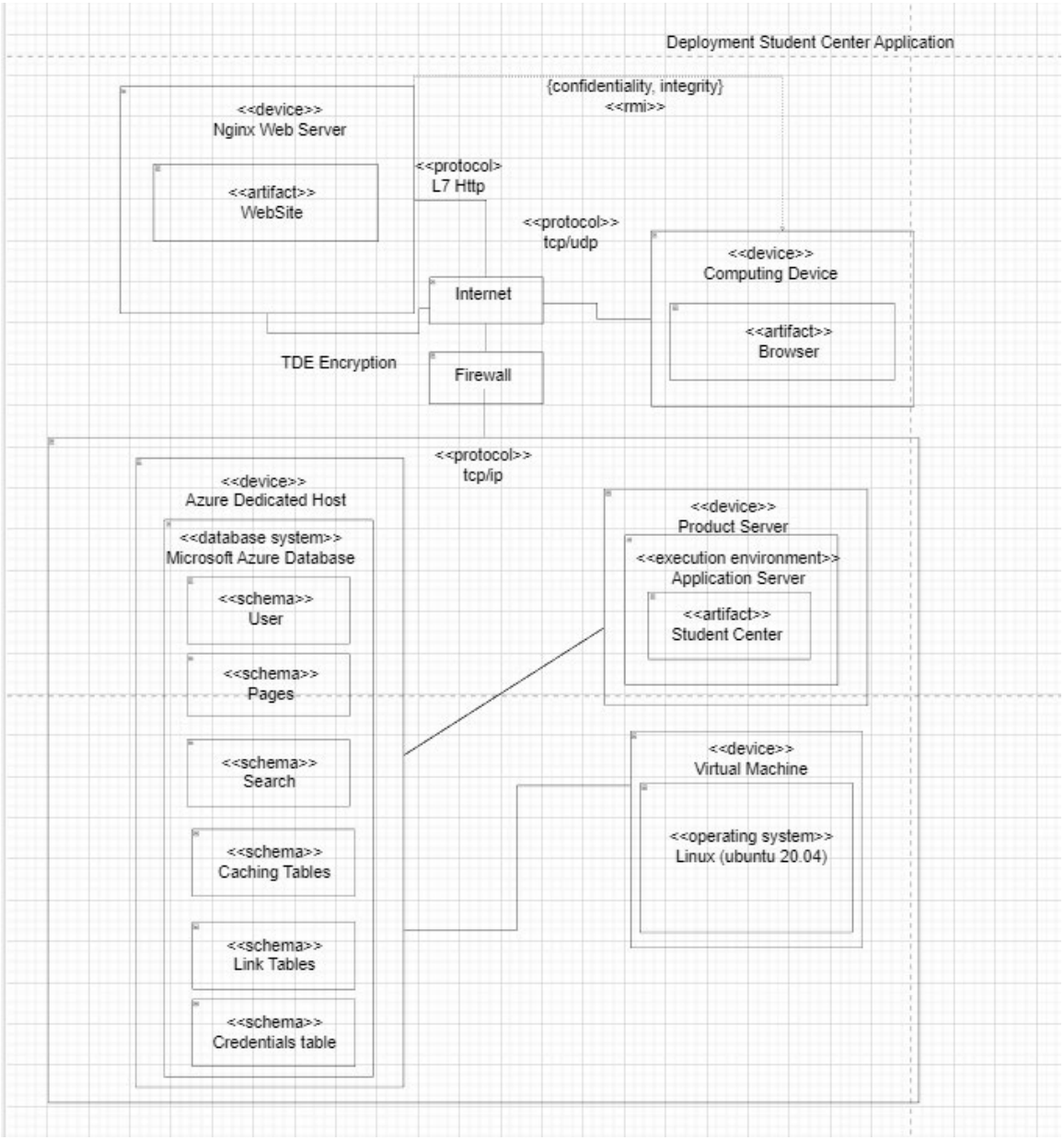
High Level Application Network and Deployment Diagrams

High Level Application Network Diagrams



Our student center network architecture lays out our secure way of connecting the client to our application. The client connects to our application via the internet through a tcp/udp connection. The Microsoft Azure Firewall ensures that only legitimate data is passed through, and ensures a secure connection. The user data requested is sent to our Azure server, and is processed. Depending on what type of information is required, a different protocol is used. To communicate with our VM, http/https is required, for our PostgreSQL server, tcp/ip is required and so on. After any request is made and accepted by the server, the relevant data is returned to the client. In cases where the relevant data is already loaded in our resolve cache, we can skip certain parts and go straight to the necessary webpage.

High Level Application Deployment Diagram



Our deployment diagram displays how our information will be connected in both its physical, and virtual components. One of the main concerns in our non-functional requirements is the guarantee that the client's information will be secure. We do so by using both a firewall, and TDE encryption for any data that is stored within our Azure server. Our Azure server will consist of our physical components, (servers, client computer) and the virtual components, (data tables, pages, search functions, operating systems, schemas and artifacts).

Identify actual key risks for your project at this time

- Time risks
    - Failure to reach deadlines can arise if we are to fall behind on our planned schedules. One deadline can snowball into another, meaning that if we get behind, it is likely we will stay behind for the remainder of the class. To prevent this from happening we will be using Trello and Discord to stay up to date with deadlines. By utilizing these two tools, we can communicate without the need to meet in person and be able to remotely remedy problems before class on Thursday.
    - Failure to arrive at planned meetings can cause a disruption with both the morale of the team and the flow of work. If even one member is behind, they will cause the whole team to fall behind. We will do our best to reach out and communicate to members who have not regularly attended our planned weekly meetings. If a member continually fails to attend meetings, we will inform the professor.
- Technical risks
    - Technical risks may arise with the availability of the services we are using. If a server is down, then our user will be unable to access our product. We will attempt to prevent this by only using established and reliable services, such as Microsoft Azure for our server.
- Teamwork risks
    - Our team may have issues when it comes to levels of commitment from each member. Levels of commitment may be seen as enough by one individual but not enough by another. We will remedy this by attempting to quantify levels of engagement and commitment. We will also inform the instructor of any individual that we do not believe are committed to the team.

- Legal/content risks
    - We are using data from schools in our model, therefore we might have legal problems by using their information without their explicit permission. We will remedy this issue by not using any personal student data that is protected under the Family Educational Rights and Privacy Act (FERPA) or Protection of Pupil Rights Amendment (PPRA).
    - We will be using the internet for much of our research. If we are to use a piece of code that is not ours and without the explicit approval of the writer, then our final product may be in jeopardy.
    - We will remedy this by either asking the original writer of the code for permission, or we will transform the code enough so it is detached from the original owner and is deemed as an original work of our own.
- Knowledge Risks
    - There is a differing level of understanding for fundamentals from each person on our team. If one member does not have the required information, then problems can arise in our implementation. We will attempt to remedy this by encouraging members to properly communicate where they are lacking in certain areas.
    - Further Knowledge Risks that may arise are the knowledge gaps between us and what we wish to implement, and what we are capable of implementing. Many of the frameworks we are utilizing are new to multiple members of the group. Therefore, time will be required to have a base level of understanding for how these frameworks operate. We intend to remedy these problems by learning the baseline fundamentals for our frameworks before we attempt to fully implement them.
- Experience Risks
    - Many of our members have yet to have worked on a large scale group project in their computer science career. Some of us will find it hard to properly communicate and convey our emotions and thoughts. We will remedy this by finding mediums which the individual is comfortable with communicating within. This may come in the form of Discord, Trello, Zoom or in class meetings,
    - Many of our members are still fairly new when it comes to large scale projects. If members attempt to spread themselves too thin, then the team as a whole will suffer. In their attempt to work on many things, they will find it difficult to stick and maintain a single task. To remedy this we will try to maintain a firm stance of separation between the backend and the frontend.
    - Maintaining a proper scope of the project is important to prevent us from getting side tracked. We will only implement what is required, namely the functional requirements. To ensure we stay on track we will frequently need to reference the documents we have laid out beforehand. While

things may change, it is important that we have a baseline fundamental guideline to go off of.

Project management

I divide my team to two small group for this milestone because this assignment has 2 major tasks. The first major task is documentation, and the second major task is the vertical prototype. We used Trello to manage the internal deadline. Trello helps my team to keep track of our milestone work progress. We integral the GitHub notification web hook to discord. The web hook can notify who and when submitted the code to the repository, then we can pull new code from GitHub. This webhook helps up avoid GitHub merge conflict. We use GitHub Action to automatically deploy the latest code to the server and check if the code is able to compile. The GitHub notification web hook and the GitHub Action greatly improved our work efficiency.

Knowing to use the proper tool is not enough to manage a team to finish such a big milestone. As a team leader, I asked every team member if they have any difficulty, or if the code had a bug. I would help or guide them to solve the issues they have. Because of our excellent management strategy, we are able to finish this milestone on time.

Detailed list of contributions

Name	Score	Contribution
Elisa Hsiao-Rou Chih	10	mockups for use cases 6 & 14 and 12, vertical prototype (styling, accessing the server, etc) contribute on every pages on vertical prototype.
Steven Paul Fong	10	mockups for use cases 7, 10, and 11 and use case 8, helped come up with design for home page/nav bar, worked on search results page/styling, worked on search parameter pulldown
Cameron Michael Yee	9	prioritized functional requirements, m1v2 revisions, database architecture ERD + requirements, api's and algorithms editing, UML diagram.
Michael Harrison Chang	10	worked on the login page, resizing the input for login, and register, created the nav, footer and errorpage, took the picture of the background, made the story board for case 2,3,4,5,9 , hard coded these sections classes, student info, important date, student records, finances and recourses, working on the pathing for the home page too
Christopher Alan Yee	9	Worked on UML diagram, Data Definitions, High level application network and deployment diagrams, identifying key risks, updated and created a new database Worked on the document Main Algorithm writing.
Zhenyu Lin	9	Worked on fixing vertical prototype bug. Worked on the document High Level APIs section Finished backend development Worked on part of document High level database architecture and organization

SW Engineering CSC648/848 Fall 2022

Project Name: New SFSU Student Center

Team Number: 05

Milestone 2

Date: 10 November 2022

Student Name	Roles	Email
Zhenyu Lin	Team Lead, GitHub Master	zlin4@mail.sfsu.edu
Christopher Alan	Backend Lead	cyee12@mail.sfsu.edu
Michael Harrison	Frontend Lead	mchang9@mail.sfsu.edu
Elisa Hsiao-Rou	Team Member	echih@mail.sfsu.edu
Steven Paul Fong	Team Member	sfong10@mail.sfsu.edu
Cameron	Team Member	cyee10@mail.sfsu.edu

Milestone/Version	Date
M1V1	09/21/2022
M1V2	10/05/2022
M2V1	10/19/2022
M2V2	11/10/2022
M3V1	11/10/2022

Contents

Data Definitions

I. Data definitions

1. Student - Our end users. Can enroll in course sections.

1.1. Date Enrolled

1.1.1. Month - Month in which they started enrollment

1.1.2. Day - Day in which they started enrollment

1.1.3. Year - Year in which they started enrollment

2. Transfer Student

2.1. Current Year

2.2. Transfer information - A student that came from another college. Has a special student record called ‘past transcripts’ for courses taken at other colleges.

2.2.1. Past Transcript

2.2.1.1. Name of previous college

2.2.1.2. Dates they were enrolled at previous college

2.2.1.2.1. Month - Month in which they started enrollment

2.2.1.2.2. Day - Day in which they started enrollment

2.2.1.2.3. Year - Year in which they started enrollment

2.3. Dates Enrolled

2.3.1. Month - Month in which they started enrollment at current university

2.3.2. Day - Day in which they started enrollment at current university

2.3.3. Year - Year in which they started enrollment at current university

3. Undergraduate Student - A student that has not yet graduated with a degree.

3.1. Dates Enrolled - When the student started enrollment

3.1.1. Month - Month in which they started enrollment at current university

3.1.2. Day - Day in which they started enrollment at current university

3.1.3. Year - Year in which they started enrollment at current university

4. Graduate Student - A student that has graduated with a degree.

4.1. Previous University

4.1.1. Name - Name of previous university

4.2. Dates Enrolled

4.2.1. Month - Month in which they started enrollment at current university

4.2.2. Day - Day in which they started enrollment at current university

4.2.3. Year - Year in which they started enrollment at current university

4.3. Graduate Program

4.3.1. Major

5. Full-Time Student - A student that is taking enough units that would be equivalent to a full-time job.

5.1. Credits - Amount of credits being taken

6. Part-Time Student - A student that is taking less units than would be equivalent to a full-time job.

6.1. Credits - Amount of credits being taken

7. Personal Information

7.1. Name

7.1.1. First Name

7.1.2. Middle name

7.1.3. Last name

7.2. Address

7.2.1. City

7.2.2. State

7.2.3. Country

7.2.4. Zip Code

7.3. Age

7.3.1. Date of Birth

7.3.1.1. Month

7.3.1.2. Year

7.3.1.3. Day

8. Course (sometimes referred to as “Class”) - The main product for the student.

8.1. Course Information

8.1.1. Times - Lists the duration of class length

8.1.1.1. Start time - When class begins

8.1.1.2. End time - When class ends

8.1.2. Start Date - Month and day when class begins

8.1.3. End Date - Month and day when class ends

8.1.4. Instructor - Lists name of instructor

8.1.4.1. Last name

8.1.4.2. First name

9. Credit (sometimes referred to as “Unit”) - A unit of measurement for the time/effort of a course. A student needs a certain amount to graduate.

10. Course Section (sometimes referred to as “Section”)

11. Online Section

11.1. Times - Lists the duration of class length

11.1.1. Start time - When class begins

11.1.2. End time - When class ends

11.2. Start Date - Month and day when class begins

11.3. End Date - Month and day when class ends

11.4. Instructor - Lists name of instructor

11.4.1. Last name

11.4.2. First name

12. Synchronous Section

12.1. Times - Lists the duration of class length

12.1.1. Start time - When class begins

12.1.2. End time - When class ends

12.2. Start Date - Month and day when class begins

12.3. End Date - Month and day when class ends

12.4. Instructor - Lists name of instructor

12.4.1. Last name

12.4.2. First name

13. Asynchronous Section

13.1. Times - Lists the duration of class length

13.1.1. Start time - When class begins

13.1.2. End time - When class ends

13.2. Start Date - Month and day when class begins

13.3. End Date - Month and day when class ends

13.4. Instructor - Lists name of instructor

13.4.1. Last name

13.4.2. First name

14. Hybrid Section

14.1. Times - Lists the duration of class length

14.1.1. Start time - When class begins

14.1.2. End time - When class ends

14.2. Start Date - Month and day when class begins

14.3. End Date - Month and day when class ends

14.4. Instructor - Lists name of instructor

14.4.1. Last name

14.4.2. First name

15. Subject

15.1. Major - The

16. Time Slot (sometimes referred to as “Date & Time”) - An attribute of course sections that details when it takes place.

16.1. Building

16.2. Room Number

17. Time Slot (sometimes referred to as “Date & Time”)

17.1. Start Time - Time in 24 hours format as to when class starts

17.2. End Time - Time in 24 hours format as to when class ends

17.3. Duration - The length in hours minute seconds of the class.

18. Component - An attribute of a course section describing the kinds of learning activity the student would be doing. (e.g. Lecture, Lab, Self-Study, etc.)

18.1. Lecture - Class where a professors convey information to the student

18.2. Lab - Interactive setting where students have hands-on experience with the course material through experiments.

18.3. Self-Study - Self regulated study, with oversight by professors to determine if a student fulfilled credit requirements.

19. Course Materials - An attribute of a course section describing the kinds of materials the student may need to purchase to take the class (e.g. textbooks)

19.1. Textbook

20. Major - An attribute of a student. Can determine what types of classes the student can take.

21. Student Record - A type of form that also acts as an attribute of students. Having certain completed records affects how the student can interact with the system.

22. Transcripts - A type of student record. A list of previously enrolled courses, and the grades received from them.

22.1. Currently enrolled courses- Courses in which the student is currently taking

22.2. Previously enrolled courses - Courses in which the student has already taken

22.3. Grades - The course evaluation give to the student by the professor

22.4. Semester - The time period in which the class was taken by the student

23. Student Calendar - A list of dates & time slots.

23.1. Important dates

23.1.1. Final day to withdraw without a W - Lists the final day a student may drop a class without receiving a W grade.

23.1.2. Final day to drop and receive a refund - Allows a student to drop a class and receive a refund for said credits.

23.1.3. Final day to change to CR/NC - Lists the date a student must change their grading option by

23.1.4. Payment due dates - Lists the day which all payments for tuition are due

23.1.5. Final day to add classes - Lists the final day in which a student can add a class without requiring a permission number

23.1.6. Final day to add a class via permission number - Lists the final day in which a student can add a class from a professor given permission number. After this day the student may no longer enroll in the class

23.1.7. Final day for faculty drops - Lists the final day that faculty may drop students enrolled in their course.

23.1.8. Final day to withdraw from classes or university - Lists the final day in which a student may exist from the university and receive a refund.

23.1.9. Holidays - Lists days in which there will be no courses taught, in which they normally would be

24. Class Schedule - A type of Student Calendar containing only the dates & times of the student's course sections.

24.1. Name of class - The descriptive name of the course

24.2. Course ID - The specific alphanumeric value of the course.

24.3. Section number - Lists the more specific class the student will be in for classes which have multiple time slots.

24.4. Time slot - Lists when the class will run from

24.5. Professor - Lists the instructor teaching the course

25. Appointments - Has a date & time, and acts as an element in a calendar, just like course sections.

26. Prerequisite - A type, of course, that is also an attribute for another different course. The prerequisite must be completed concurrently with, or before the student takes the associated other courses.'

26.1. Courses the prerequisite unlocks

26.1.1. Name of the course the prereq unlocks

26.1.1.1. Name of course

26.1.1.2. Course ID

27. Professor - The teacher of a section. Can modify course section information.

27.1. Courses offered - Lists all courses that the specified teacher will be teaching.

27.2. First name

27.3. Last name

28. Advisor - An employee of the university that specializes in helping students with school-related questions.

28.1. Name of advisor

28.2. Time slots - Times in which the advisor is available

29. College (sometimes referred to as “University”) - The client organization. Uses the system to offer courses to students.

30. Department - A subsection of a college that facilitates learning for specific subject(s).

31. Campus Clinic - Site that students can visit for health-related issues.

31.1. Hours - Lists the times that the clinic is open

31.2. Location - Lists where you can find the Campus Clinic

32. Student Club - A school-recognized student organization. Associated with a faculty advisor.

    32.1. Faculty advisor Lists the faculty advisor of the student club

    32.2. Name - Lists the name of the specific club

33. Parking Permit - A form that grants students access to parking.

    33.1. Location - Lists the locations the permit is valid in.

34. Bill - A form that shows students' transactions after purchase.

    34.1. Amount paid

    34.2. Amount remaining - Tells the user how much of their bill they have yet to pay

35. Charge - An element on a bill. A specific amount the student owes for a specific purpose.

    35.1. Name of charge - Tells the user what type of charge is being levied on their account.

    35.2. Value of charge - Tells the user how much they must pay

36. Aid Form - A form that shows students’ financial aid information.

    36.1. Loan amount granted - Tells the user the amount of money they may receive from a loan in dollars

    36.2. Loan amount accepted - Tells the user how much of the granted loan they have accepted in dollars

36.3. Grant amount - The amount quantity in dollars of a federal pell grant given to a student

36.4. Grant amount accepted - The amount of a federal pell grant the student has accepted in dollars

37. Map - An image showing the geographical layout of a location.

37.1. Format - JPG

38. Schedule - an itinerary that you follow throughout the day.

39. Notification - an alert that notifies the student of an important message.

40. Holds - A property that prevents students from enrolling until resolved.

40.1. Name of hold - Tells the student the name of a hold so they can identify the type of hold they have

40.2. Date

40.2.1. Date created - Tells the student when their hold began

40.2.2. Due date - Tells the student when they must clear the hold by without incurring any penalty

41. Search Parameter - An attribute or specific value of an attribute that is used to filter/narrow down elements from a full list of elements.

41.1. Session

41.1.1. Semester - Lists the semester which class is offered E.g Summer, Fall, Spring

41.1.2. Year - Year of the course which is being offered

41.2. Subject - Specifies and searches for only courses of specified major E.g Statistics

41.3. Course number - The numerical ID attribute of the course

41.3.1. Contains - Returns courses in which the value searched appears in the course number.

41.3.2. Greater than or equal to - Returns courses with course number greater than or equal to the search parameters

41.3.3. Less than or equal to - Returns courses with course number less than or equal to the search parameters

41.3.4. Is exactly - Only searched the courses with the exact string parameter.

41.4. Additional criteria

41.4.1. Start time - Specifies when class begins.

41.4.1.1. Between - Lists courses beginning between the times specified

41.4.1.2. Greater than - Lists courses with start times after the times specified, but not going further than midnight of the next day

41.4.1.3. Greater than or equal to - Inclusively lists courses with start times of the times specified, but not going further than midnight of the next day

41.4.1.4. Is exactly - Lists courses only which begin exactly when specified.

41.4.1.5. Less than - Lists courses beginning before the times specified, but not going further than midnight of the current day

41.4.1.6. Less than or equal to - Inclusively lists courses that begin before the times specified, beginning at midnight of the current day.

41.4.2. End time - Specifies when class ends.

41.4.2.1. Between - Lists courses ending between the times specified.

41.4.2.2. Greater than - Lists courses with end times after the times specified, but not going further than midnight of the next day

41.4.2.3. Greater than or equal to - Inclusively lists courses with end times of the times specified, but not going further than midnight of the next day

41.4.2.4. Is exactly - Lists courses only which end exactly when specified.

41.4.2.5. Less than - Lists courses ending before the times specified, but not going further than midnight of the current day

41.4.2.6. Less than or equal to - Inclusively lists courses that end before the times specified, beginning at midnight of the current day.

41.4.3. Days of week

41.4.3.1. Exclude any of these days - Does not search for courses that only fall on the precisely specified days.

41.4.3.2. Exclude only these days - Does not search for courses that have a session on each of the precisely specified days.

41.4.3.3. Include any of these days - Search for courses that have a session on each of the precisely specified days.

41.4.3.4. Include only these days - Only search classes that have a session on the specified day.

41.4.4. Instructor name

41.4.4.1. Begins with - Searches instructors whose names begin with the string specified

41.4.4.2. Contains - Searches if a string parameter is contained within the instructor's name.

41.4.4.3. Is exactly - Searches for a professor whose last name is exactly the same as the search parameter.

41.4.5. Course attribute

41.4.5.1. GE - General education, needed by the state to graduate.

41.4.5.2. Upper division - Courses in which a student must have a graduate standing to take.

41.4.5.3. Lower division - Courses in which a student must have a undergraduate or higher standing to take

41.4.5.4. Graduation requirement - Courses in which are necessary to graduate, not necessarily a GE class.

41.4.6. Course keyword - Searches for a common keyword, E.g Computer science

41.4.7. Maximum units

41.4.7.1. Greater than - Lists courses with credits greater than the amount of course credits specified.

41.4.7.2. Greater than or equal to - Inclusively lists courses with credits greater than the amount of course credits specified.

41.4.7.3. Is exactly - Lists courses which are exactly the amount of course credits specified.

41.4.7.4. Less than - Lists courses with credits less than the amount of course credits specified.

41.4.7.5. Less than or equal to - Inclusively lists courses with credits less than the amount of course credits specified.

41.4.8. Minimum units

41.4.8.1. Greater than - Lists courses with credits greater than the amount of course credits specified.

41.4.8.2. Greater than or equal to - Inclusively lists courses with credits greater than the amount of course credits specified.

41.4.8.3. Is exactly - Lists courses which are exactly the amount of course credits specified.

41.4.8.4. Less than - Lists courses with credits less than the amount of course credits specified.

41.4.8.5. Less than or equal to - Inclusively lists courses with credits less than the amount of course credits specified.

41.4.9. Location

41.4.9.1. On campus - Courses provided on the main campus

41.4.9.2. Off campus - Courses not provided on the main campus. May be provided at a satellite location or an Annex.

41.4.9.3. Country - Courses provided in a country foregin from the main university’s country.

42. Course Requirements - a course that is required in order to take the next course.

42.1. Prerequisite - A course that must first be taken before attempting to take another course

43. Health Records - records that inform the school/doctor about one person's health information.

43.1. Name

43.1.1. First Name

43.1.2. Middle name

43.1.3. Last name

43.2. Address

43.2.1. City

43.2.2. State

43.2.3. Country

43.2.4. Zip Code

43.3. Age

43.3.1. DOB

43.3.1.1. Month

43.3.1.2. Year

43.3.1.3. Day

43.4. Allergies - Lists any potential allergies the student has to prevent an allergic reaction.

43.5. Medical conditions - Lists any medical conditions the student has.

Functional Requirements

Priority 1:

1. 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.

2. 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.

3. 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.

4. 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.
5. 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.
6. Transcripts
   6.1. Transcripts shall list all courses taken in the past.
7. 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.

Priority 2:

1. Student
   1.1. Student calendars shall recommend alternative course sections in order to resolve date & time conflicts on the class schedule.
   1.2. Students shall be notified if a course section on their calendar has any of its attributes changed. (students should know if there's a change of professor or change of location)
   1.3. Students shall be notified when a change of major is fully processed, regardless of whether it is accepted or rejected.

1.4. Students who've recently changed majors shall have access to resources for their new major.
1.5. Students shall be able to swap one course for another.
1.6. Students shall be able to pay for courses.
1.7. Students shall be able to generate What-If Reports.
1.8. Students shall be able to save their What-If Reports.
1.9. Students shall be automatically dropped from waitlists of course sections that are of the same class as one that the student is fully enrolled in.
1.10. Students shall be automatically dropped from waitlists of course sections that cause time conflicts with a course section that the student is fully enrolled in.
1.11. Students shall receive a grade, upon completing a course.
1.12. Students shall be able to request to change their major.
1.13. Students shall be able to search for clubs at the university.
1.14. Students shall be able to schedule advising appointments.
1.15. Students shall be able to schedule financial aid appointments.
1.16. Students shall be able to make an appointment with the university clinic.
1.17. Students shall be notified of upcoming appointments.

2. Transferring
   2.1. Transfer students can have a transfer credit report.
   2.2. Transfer students shall be able to search for other colleges' courses to check if they count as being transferred.
3. What-if report
   3.1. What-If Reports shall show students the required classes for hypothetical change of major, degree, or other academic career choices.
4. Professor Reviews
   4.1. The system shall notify reviewers that their received grade will be displayed along with their anonymous review.
   4.2. Professor reviews shall have tags to help students parameterize searches when filtering for professors with certain teaching styles.

Priority 3:

1. Student
   1.1. Students shall be able to purchase parking permits.
   1.2. Students shall be notified if their permit purchase was approved.
   1.3. Students shall be able to open Google Maps in the student center.
   1.4. Students shall be able to access their payment histories.
2. Payment

2.1. PayPal shall be supported as a payment method.

3. Advising

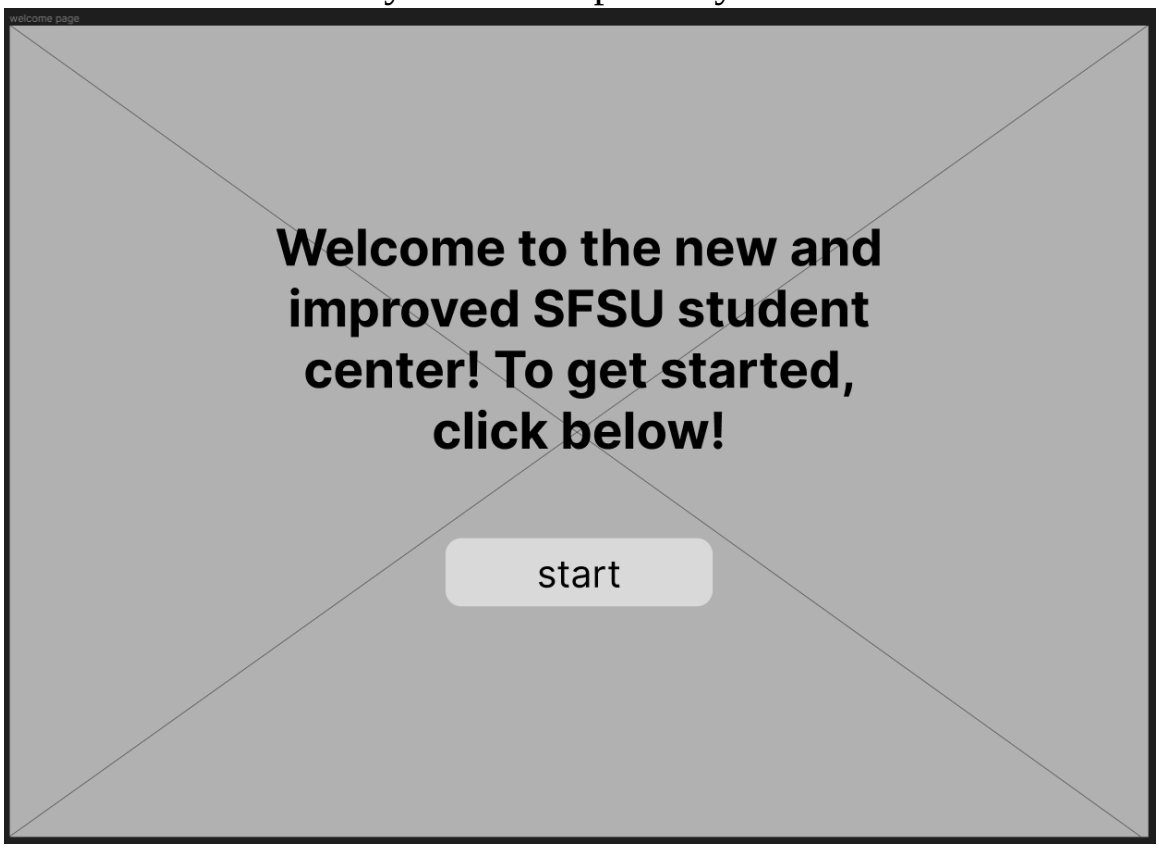
3.1. Advisors shall be able to schedule appointment time slots.

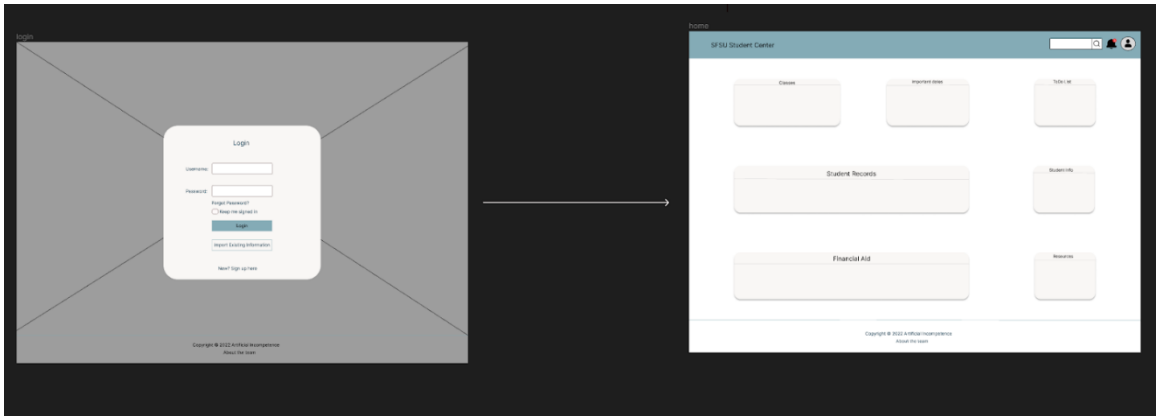
4. Clubs

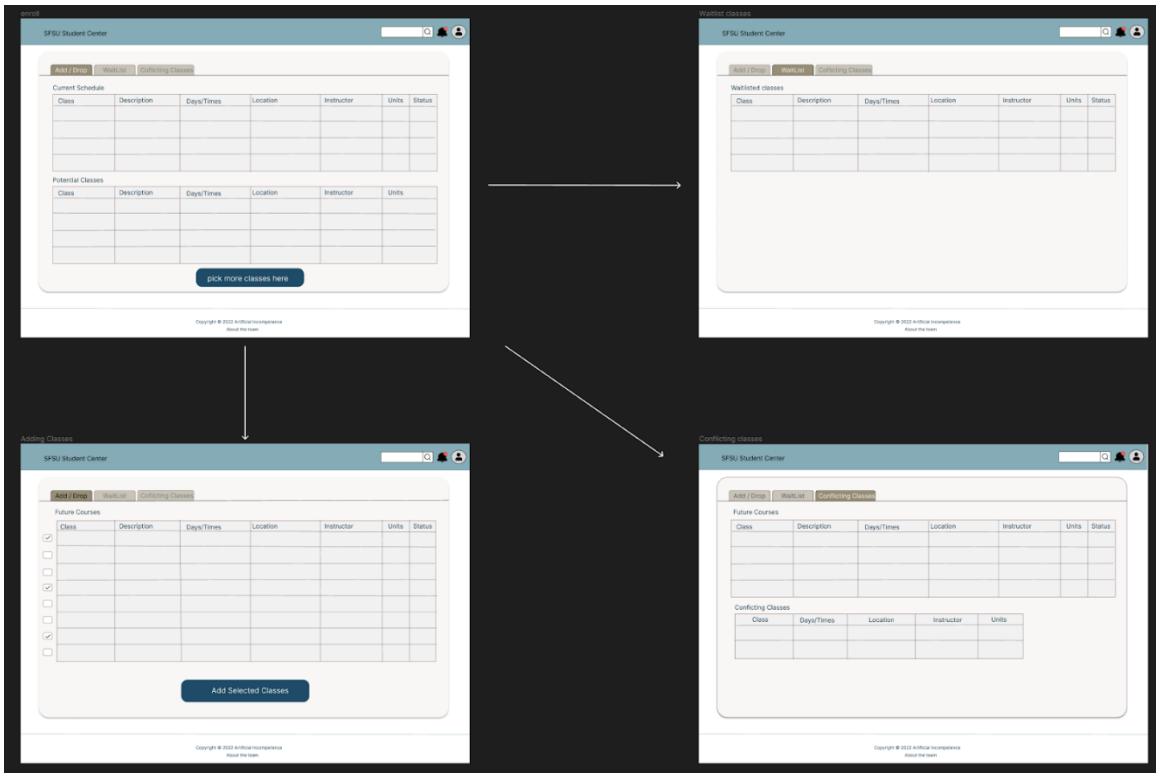
4.1. Clubs shall be displayed by the system.

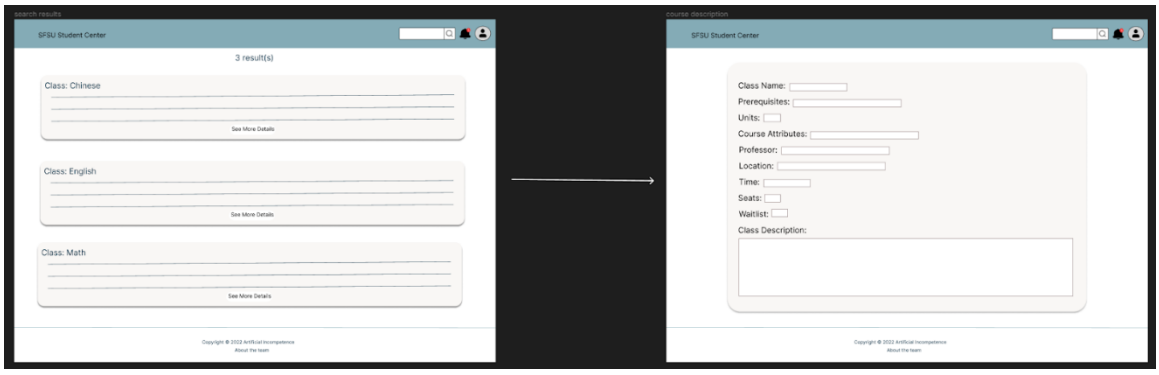
4.2. Clubs can provide resources to the system for students to see.

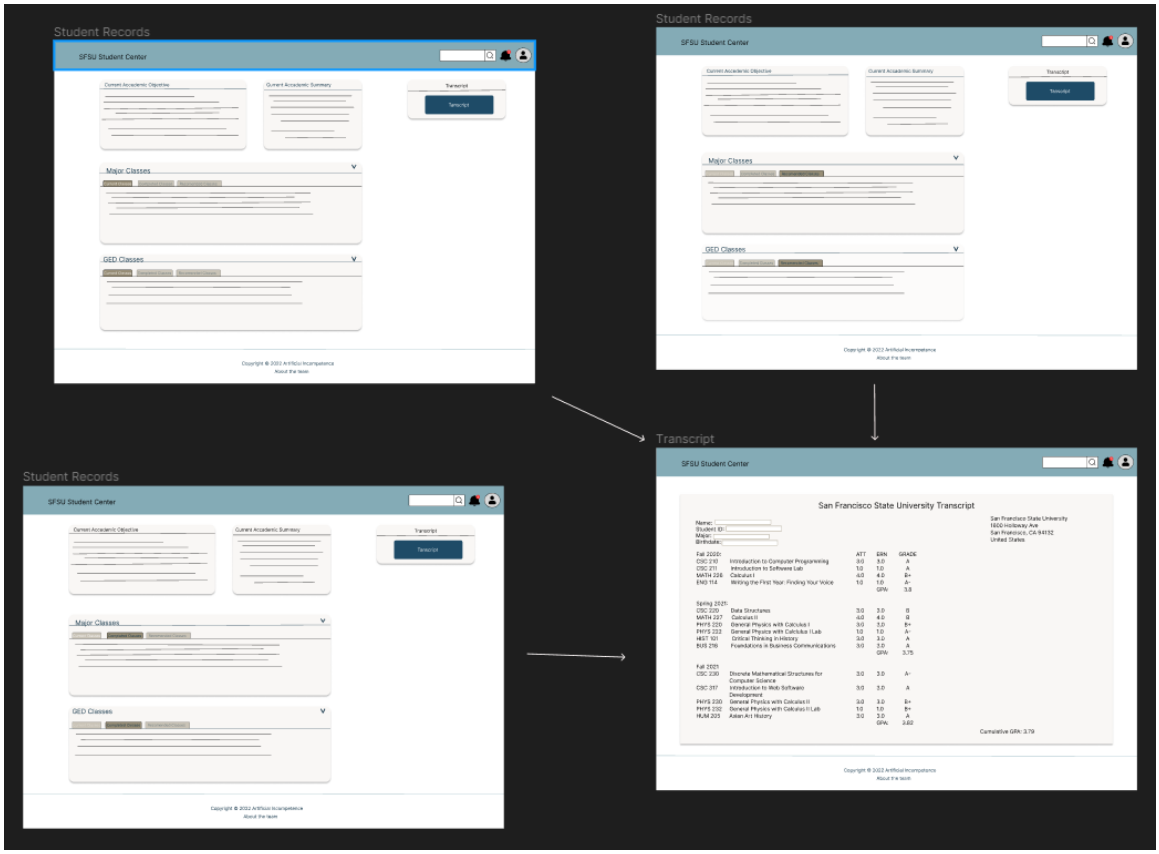
Wireframes Based on your Mockups/Storyboards V2

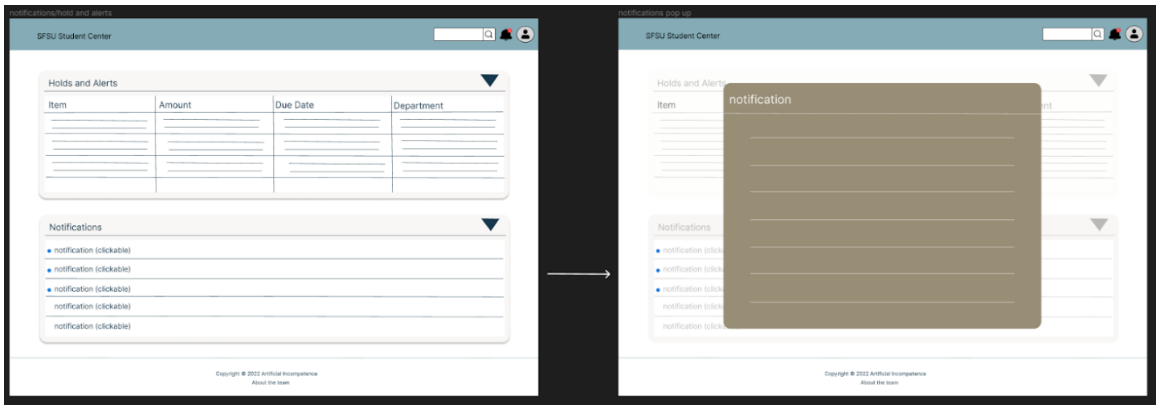


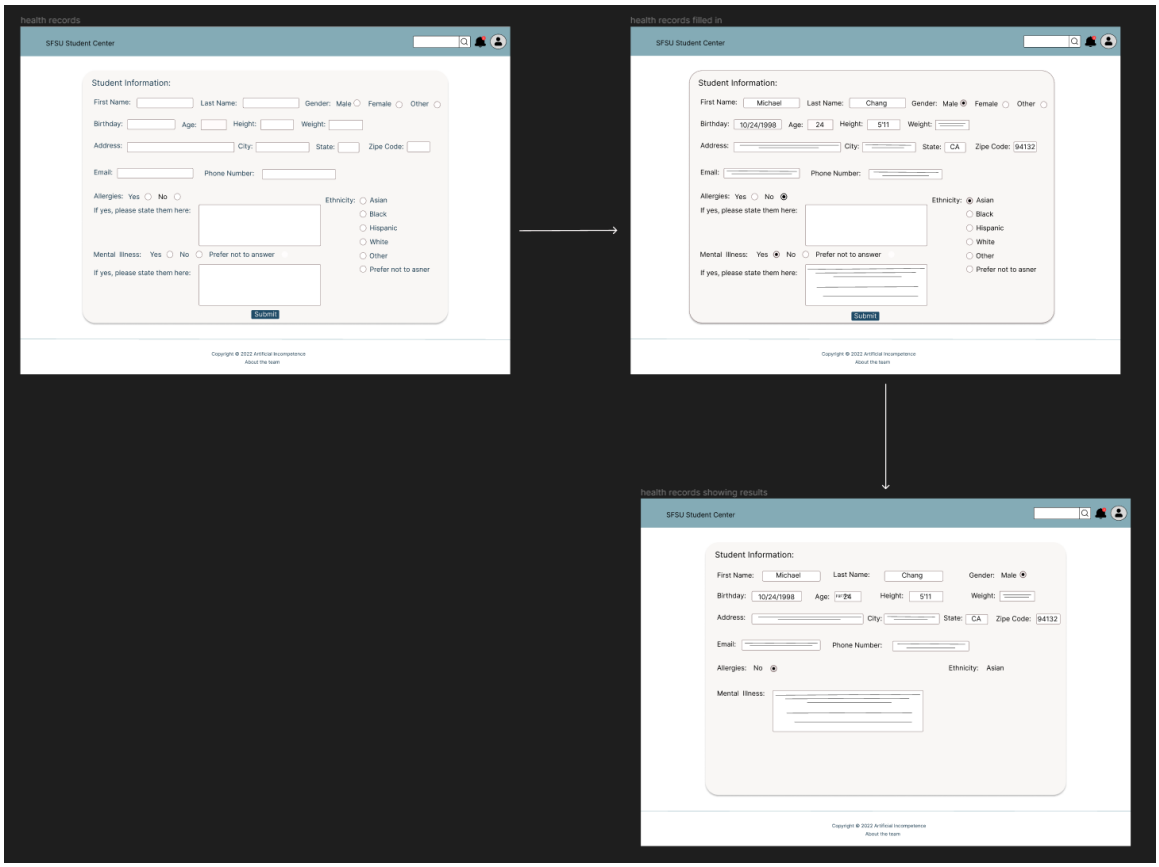


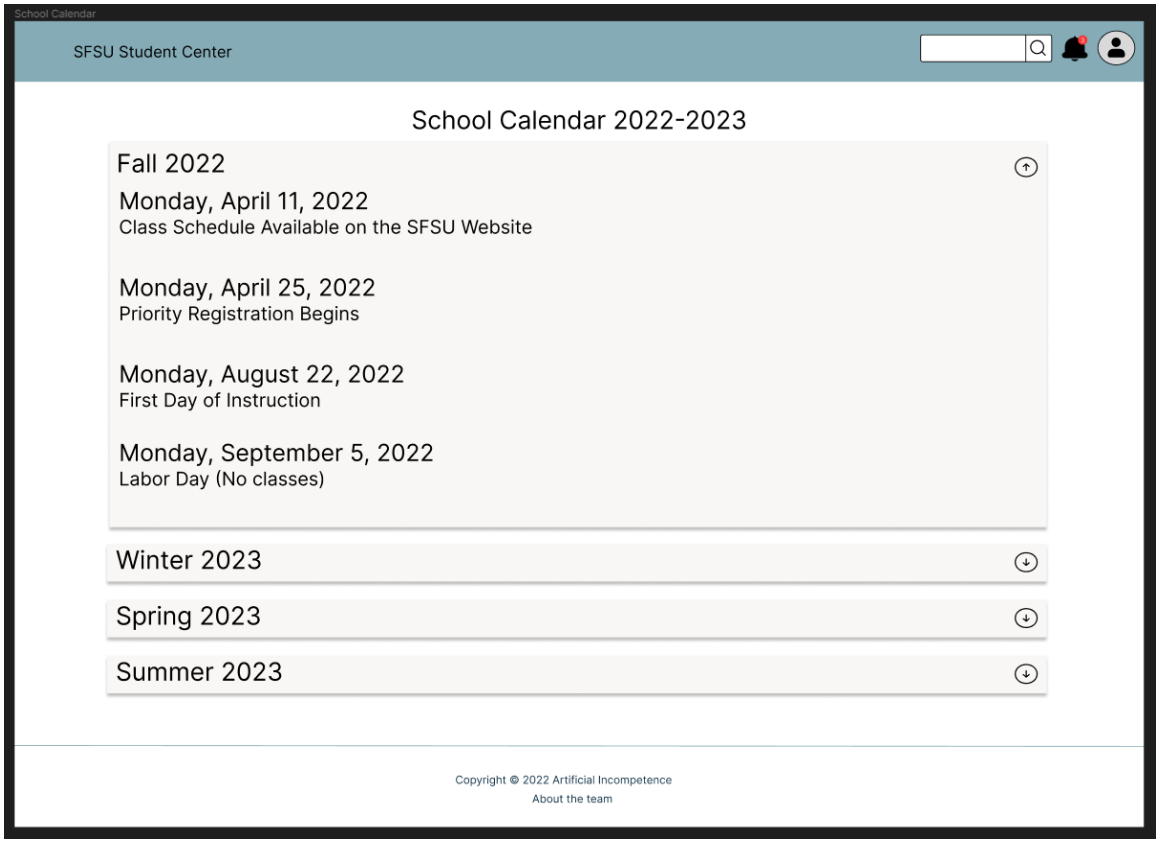

enroll
SFSU Student Center
Add / Drop
Waitlist
Conflicting Classes
Current Schedule
Class
Description
Days/Times
Location
Instructor
Units
Status
Potential Classes
pick more classes here
Waitlist classes
Waitlisted classes
Adding Classes
Future Courses
Add Selected Classes
Conflicting classes
Conflicting Classes

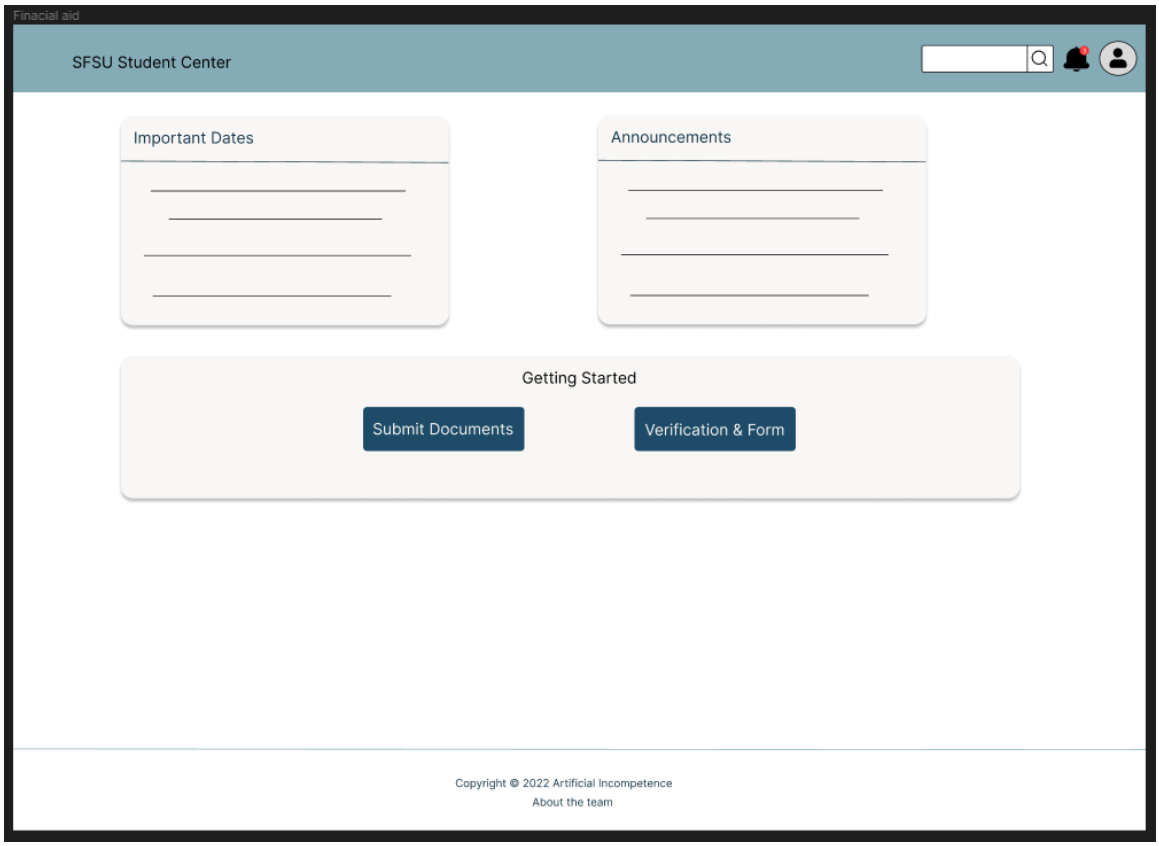

search results
SFSU Student Center
3 result(s)
Class: Chinese
See More Details
Class: English
Class: Math
course description
Class Name:
Prerequisites:
Units:
Course Attributes:
Professor:
Location:
Time:
Seats:
Waitlist:
Class Description:


Student Records
SFSU Student Center
Major Classes
GED Classes
Student Records
SFSU Student Center
Major Classes
GED Classes
Transcript
SFSU Student Center
San Francisco State University Transcript
Student Records
SFSU Student Center
Major Classes
GED Classes


notifications/hold and alerts
SFSU Student Center
Holds and Alerts
Item
Amount
Due Date
Department
Notifications
notification (clickable)
notification (clickable)
notification (clickable)
notification (clickable)
notification (clickable)
notifications pop up
SFSU Student Center
Holds and Alerts
Item
notification
Notifications


health records
SFSU Student Center
Student Information:
First Name:
Last Name:
Gender: Male Female Other
Birthday:
Age:
Height:
Weight:
Address:
City:
State:
Zipe Code:
Email:
Phone Number:
Allergies: Yes No
If yes, please state them here:
Ethnicity: Asian Black Hispanic White Other Prefer not to asner
Mental Illness: Yes No Prefer not to answer
If yes, please state them here:
Submit
Copyright © 2022 Artificial Incompetence
About the team
health records filled in
SFSU Student Center
Student Information:
First Name: Michael
Last Name: Chang
Gender: Male Female Other
Birthday: 10/24/1998
Age: 24
Height: 5'11
Weight:
Address:
City:
State: CA
Zipe Code: 94132
Email:
Phone Number:
Allergies: Yes No
If yes, please state them here:
Ethnicity: Asian Black Hispanic White Other Prefer not to asner
Mental Illness: Yes No Prefer not to answer
If yes, please state them here:
Submit
Copyright © 2022 Artificial Incompetence
About the team
health records showing results
SFSU Student Center
Student Information:
First Name: Michael
Last Name: Chang
Gender: Male
Birthday: 10/24/1998
Age: 24
Height: 5'11
Weight:
Address:
City:
State: CA
Zipe Code: 94132
Email:
Phone Number:
Allergies: No
Ethnicity: Asian
Mental Illness:
Copyright © 2022 Artificial Incompetence
About the team


School Calendar
SFSU Student Center
School Calendar 2022-2023
Fall 2022
Monday, April 11, 2022
Class Schedule Available on the SFSU Website
Monday, April 25, 2022
Priority Registration Begins
Monday, August 22, 2022
First Day of Instruction
Monday, September 5, 2022
Labor Day (No classes)
Winter 2023
Spring 2023
Summer 2023
Copyright © 2022 Artificial Incompetence
About the team


Finacial aid
SFSU Student Center
Important Dates
Announcements
Getting Started
Submit Documents
Verification & Form
Copyright © 2022 Artificial Incompetence
About the team

High level database architecture and organization

Database Requirements:

1. Student
    1.1. A student can enroll in many course sections.
    1.2. A student can complete many course sections.
2. Course
    2.1. A course can have many course sections.
3. Course Section
    3.1. A course section belongs to one course.
    3.2. A course section can enroll many students.
    3.3. A course section is taught by one professor.
    3.4. A course section can have been completed by many reviewers.
4. Reviewer
    4.1. The reviewer is a student.
    4.2. A reviewer has completed many course sections.
    4.3. A reviewer can review many professors.
5. Professor
    5.1. A professor can teach many course sections.
    5.2. A professor can be reviewed by many reviewers.

Database Management System:

PostgreSQL

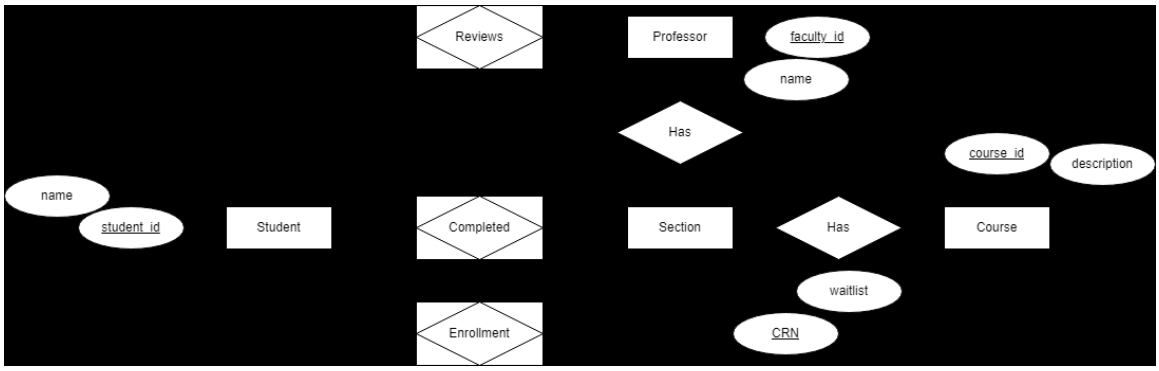
Media storage Strategy:

Our system will store the media file in a file system. The database will store the location of media. One of the unique functions of our project is the professor ranking system. Students can rank their professor and give them feedback. Therefore, the most of the storage content for our project will be text. Majority of our data will be stored on the database. Although our first priority functional requirement has the majority of text data, our second and third functional requirements need to store large files such as student transcript pdf version and school club icons. The database will take huge amount of memory for backend to acquire the data from the database if we choose to encode the large file and store them into database .To effective use the memory and save our database bandwidth, we choose to store the large file like media file or pdf in a file system.

Search/filter architecture and implementation:

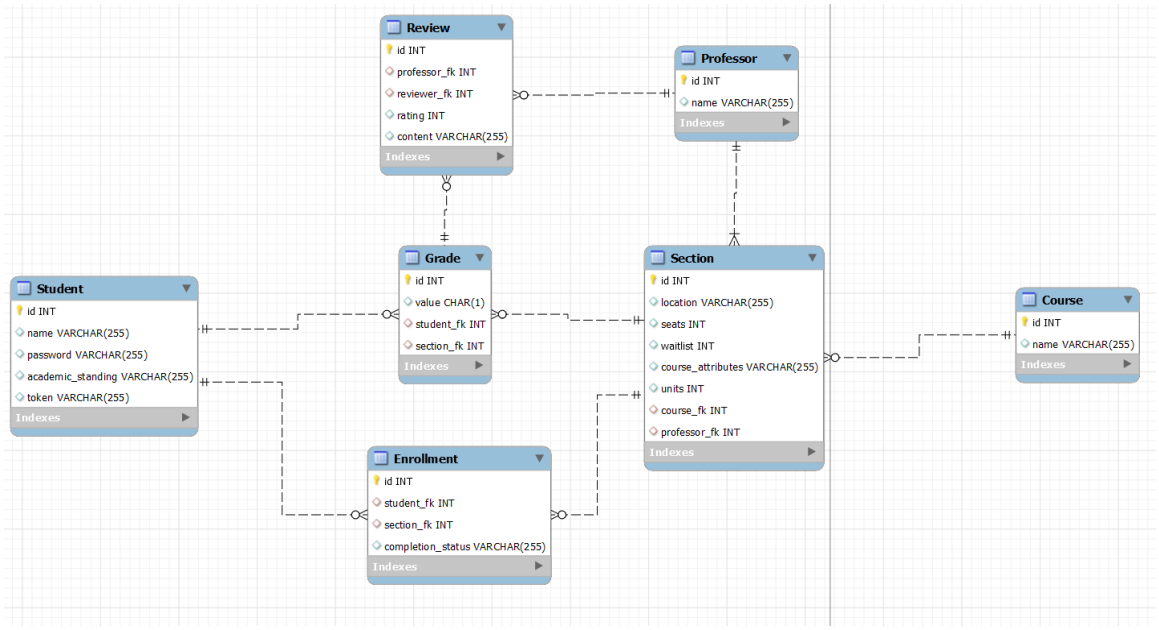
In our project, we will have one search bar across all the pages. Therefore, the search function should be searched across all of our database tables. We will ask the user about the search category first. Then let them type in the search content. The backend will base on the category and send the searching query to the database. For example, if the user wants to search about the ranking for a specific professor, the user would first choose the searching category - ranking. Then type in the search content. The backend will first process the content by an algorithm. If the user types in the course number, the backend will send a query to the course table to get the professor id and check the professor ranking. Our search strategy is to send multiple search queries to different tables in the database. Then we will combine the search results and return to the front. We will ORM to manipulate data from the database. The benefit of ORM is that it can save lots of time in development. ORM avoids writing poorly-formed SQL and a lot of stuff is done automatically. Therefore, ORM significantly reduces the development time.

ERD



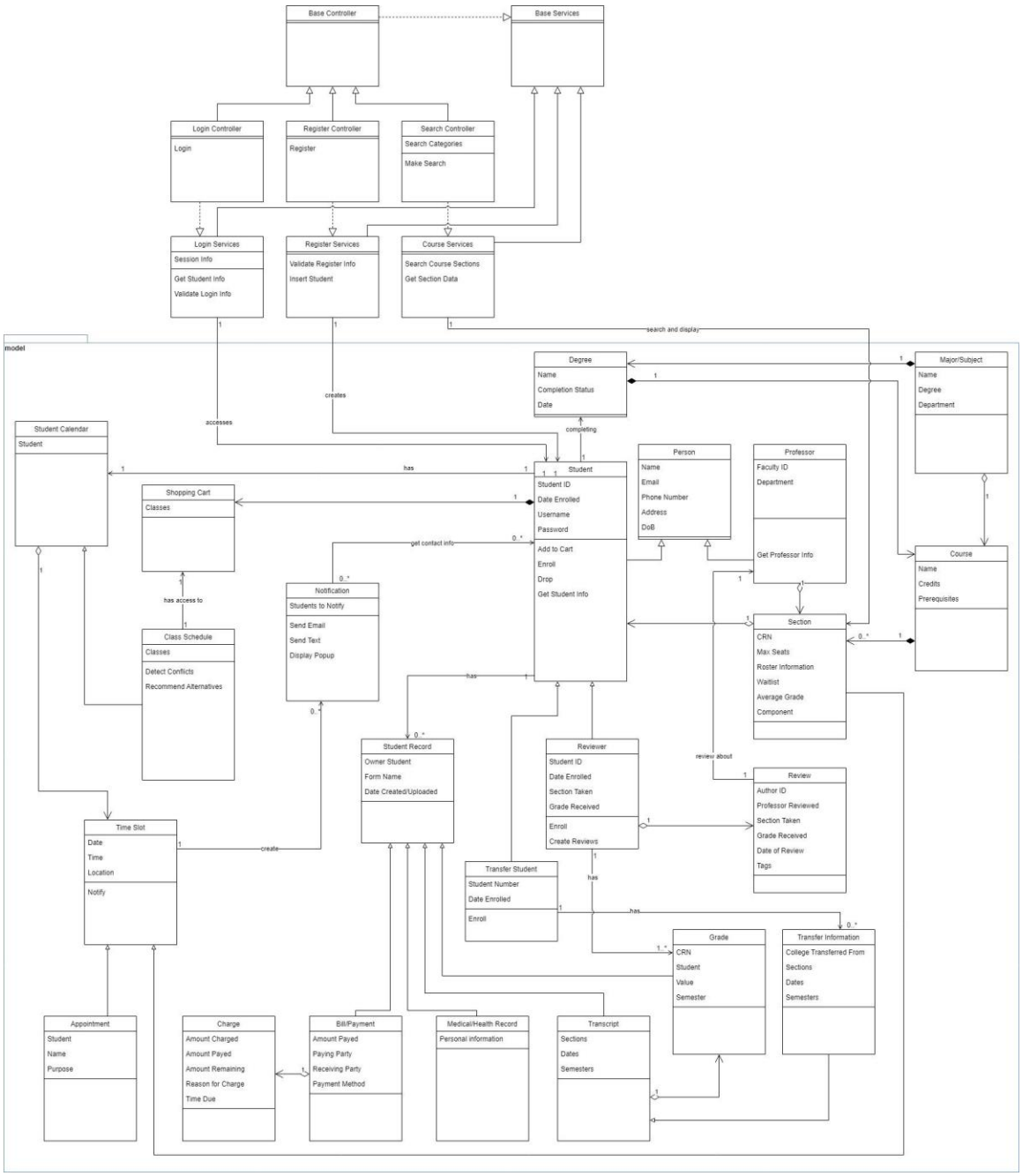
The DBMS we will be using is postgreSQL. We are choosing to use postgreSQL because it is reliable, is a relational database and because it is a general purpose OLTP database.

EER:



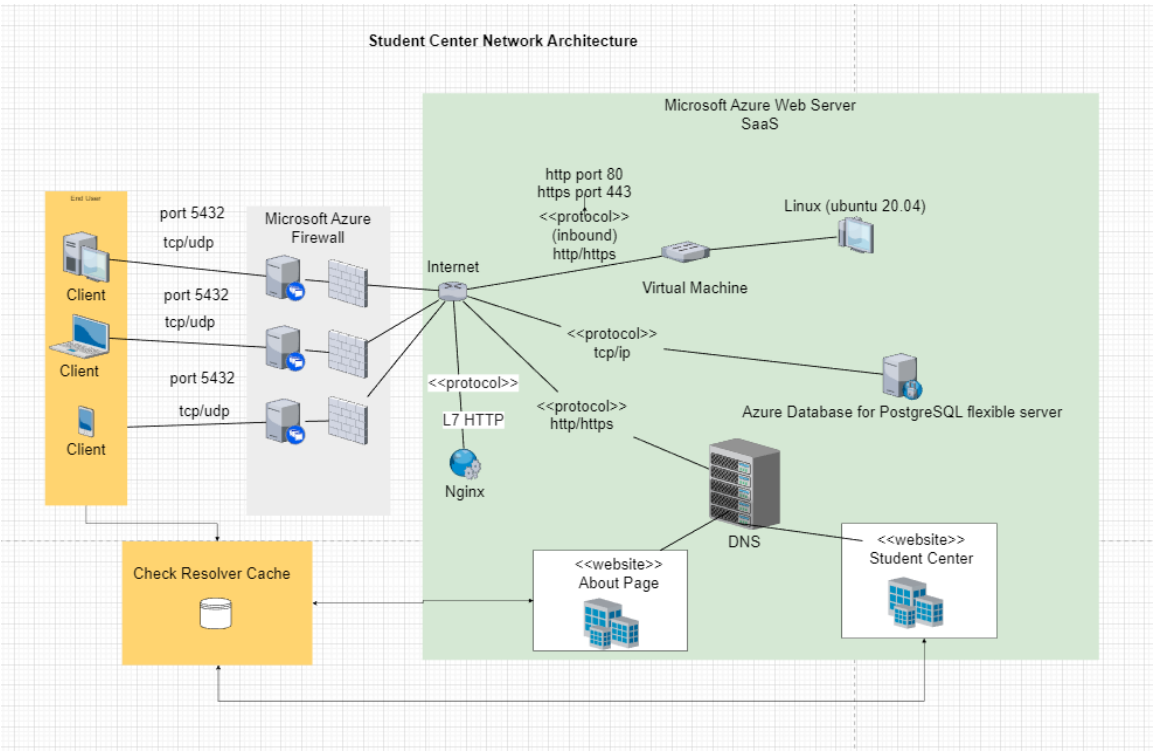
High Level Diagrams

High Level UML Diagrams



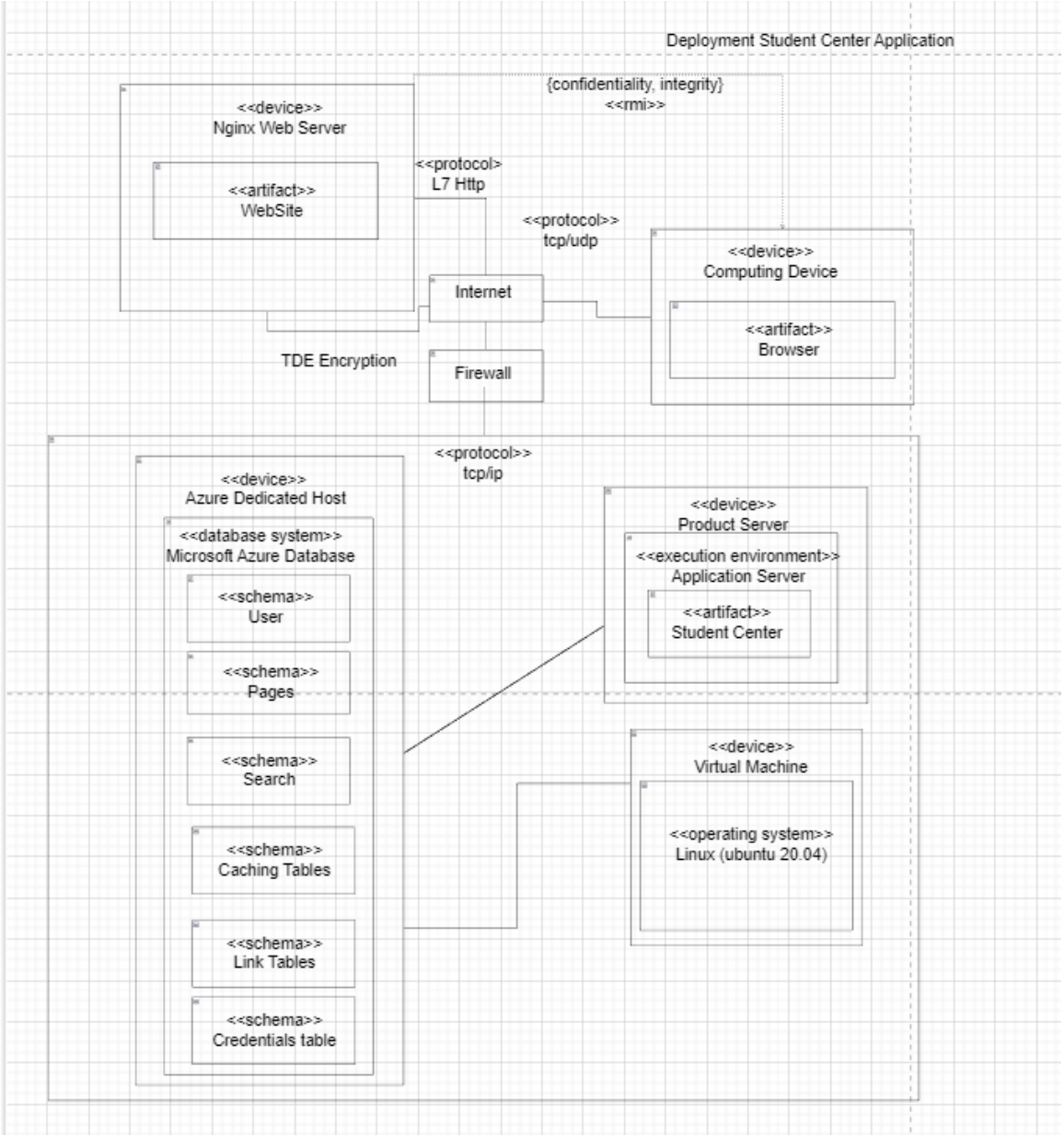
High Level Application Network and Deployment Diagrams

High Level Application Network Diagrams



Our student center network architecture lays out our secure way of connecting the client to our application. The client connects to our application via the internet through a tcp/udp connection. The Microsoft Azure Firewall ensures that only legitimate data is passed through, and ensures a secure connection. The user data requested is sent to our Azure server, and is processed. Depending on what type of information is required, a different protocol is used. To communicate with our VM, http/https is required, for our PostgreSQL server, tcp/ip is required and so on. After any request is made and accepted by the server, the relevant data is returned to the client. In cases where the relevant data is already loaded in our resolve cache, we can skip certain parts and go straight to the necessary webpage.

High Level Application Deployment Diagram



Our deployment diagram displays how our information will be connected in both its physical, and virtual components. One of the main concerns in our non-functional requirements is the guarantee that the client's information will be secure. We do so by using both a firewall, and TDE encryption for any data that is stored within our Azure server. Our Azure server will consist of our physical components, (servers, client computer) and the virtual components, (data tables, pages, search functions, operating systems, schemas and artifacts).

Detailed list of contributions

Name	Score	Contribution
Elisa Hsiao-Rou Chih	8	figma for: welcome, notifications/holds and alerts, notifications popup. updated the colors on all of the wireframes. made the front end pages for notifications/holds and alerts, health records. and did some styling on the front end pages
Steven Paul Fong	7	Figma diagrams for adding classes, school calendar, course description, transcript and implementation of school calendar and course description pages.
Cameron Michael Yee	9	revising fujnctional requirements from m2. revising database section documentation. added EER to database section. added backend API's for the enrollment system
Michael Harrison Chang	7	made the finacial aid page and the student records, set up most of the pages with the nav bar and footer for most of the pages , made the homepage , all healthrecords , all the student records, login page figma
Christopher Alan Yee	9	Updated data definitions Updated which DBMS we are using Reorganized p1, p2 and p3 Network diagram Backend API
Zhenyu Lin	8	Working on Enrollment Page and home page, Working on the Priority 1 function: Add Class to shopping cart, Add Class from shopping cart to current schedule, drop class from the current schedule. Search classes.

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	zlin4@mail.sfsu.edu
Christopher	Backend Lead	cyee12@mail.sfsu.edu
Michael	Frontend Lead	mchang9@mail.sfsu.edu
Elisa Hsiao-Rou	Team Member	echih@mail.sfsu.edu
Steven Paul	Team Member	sfong10@mail.sfsu.edu
Cameron	Team Member	cyee10@mail.sfsu.edu

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

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/

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

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/

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

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

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/ (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 loctaion	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 (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: Test@Test Password: 12346789a	Successful Login	Chrome: Pass Firefox: Pass Edge: Pass
2	Username Login	Attempt to login using a valid username.	Email: TestUser Password: 12346789a	Failed Login	Chrome: Pass Firefox: Pass Edge: Pass
3	ID Login	Attempt to login using a valid student ID.	Email: 123456789 Password: 12346789a	Failed Login	Chrome: Pass Firefox: Pass 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 (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 Firefox: Pass 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 Firefox: Pass 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 Firefox: Pass 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 (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 Firefox: Pass 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 Firefox: Pass 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 Firefox: Pass

			list of finances info".		Edge: 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 (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: Pass Firefox: Pass Edge: 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: Pass Firefox: Pass Edge: 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 Firefox: Pass 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 (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 Firefox: Pass 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 Firefox: Pass

			full list of student record".		Edge: 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: Pass Firefox: Pass Edge: 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

7.3The 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 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 Documentation in m4v1 best security practices 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