

Business Requirements Document

Student Multi-Tool

10.06.2021

Team Marvel

Albert Toscano

Audrey Brio

Bradley Nickle

Joseph Cutri

Michael Kriesel (Team Leader)

Version History

Date	Changes	Approval Date
10/06/2021	Initial Draft	

Executive Summary

This Business Requirements Document (BRD) outlines the requirements for the Student Multi-Tool project. It contains both functional and non-functional requirements, an overview of the current process, as well as the proposed process once the solution is implemented. It is used to determine what needs to be done, and is a starting point for a solution design.

Student Multi-Tool

Student Multi-Tool is created for college users to help them navigate key issues they may face throughout their time in college. This application aims to provide users the ability to create collaborative schedules, meet new people, see past ratings on the difficulty of courses and professors, find potential jobs and internships and learn which grants and scholarships they are eligible for. It is the Student Multi-Tool's goal to be the one stop shop to help solve problems that college users face.

Project Scope

In Scope

The following areas are in scope for this project:

- Chrome Browser using the latest version available in January 2022. We plan on supporting each version until the latest version of Chrome available in April 2022.
- An internet connection is required for all interactions with the system
- The supported language will be U.S. English
- The supported currency will be U.S Currency
- Our target audience is college students
- US Calendar (Sunday-Monday)
- Since our focus is on campuses in California, we will be using the Pacific Time Zone for times and date purposes. If there are any users using the web application that are not located in the Pacific Time Zone, dates and times will still show up in the Pacific Time Zone.
- Collaborative schedule builder feature
 - Generic weekly schedules for each term
- Collaborative schedule comparison feature
 - Comparison of up to five schedules to find common free time
- Course difficulty feature
 - Provides information on difficulty of course and professors
- Matching feature
 - Provides way to connect and get in contact with other users on campus
- Student Discounts
 - Provides users with information on establishments that offer discounts to users
- Aid Eligibility
 - California grants and scholarships
- During development we intend to focus on California State University Long Beach, and after our initial deployment, expand to other California State University and University of California campuses.
 - All 90 majors offered by CSULB
 - [Colleges and Majors, CSULB](#)^[1]
 - All majors offered by other schools in California
 - Courses offered by the school within the past year

Out of Scope

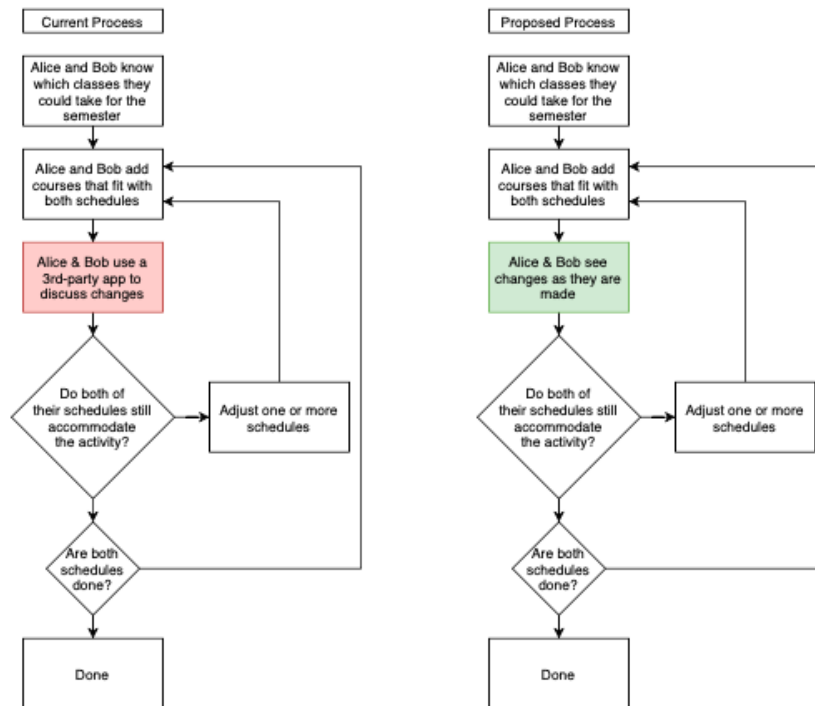
The following areas are out of scope for this project:

- Only checking if user has valid .edu email address, not whether they are legitimately a student or not
- The schedule features are not a replacement for College Advisers, and does not provide information on what classes are required for graduation.
- Grant scholarship and eligibility
 - Not submitting application for the user
- Student Discounts
 - Not providing student discounts for the user
- Course Difficulty
 - Not for providing recommendations, just other users experience in a given class
- Matching
 - Not forcing a communication between matches
- When comparing schedules, we are only checking when users have common free time in general, as opposed to on specific dates
- There is no offline access available for this product, all interactions require an internet connection

Current Processes vs. Proposed Processes

1. Collaborative Schedule Builder

- a. Current process - if two users wanted to make their schedules together so that they would have a particular item in common (class, carpool, other activity) then they would need to individually make their schedules while communicating with each other.
Assume Alice and Bob are college users, that they want to collaborate with each other to fit some common desired activity (take a class together, carpool, or do something else on a regular basis) into both of their schedules. How would they do that using MYCSULB? Note that these flowcharts assume that Alice and Bob each have some combination of classes that accommodates their desired activity, which isn't guaranteed in the real world
- b. Proposed process - if two or more users wanted to make their schedules together so that they can have a particular item in common, they would be able to see live changes to a schedule and can edit a schedule together.

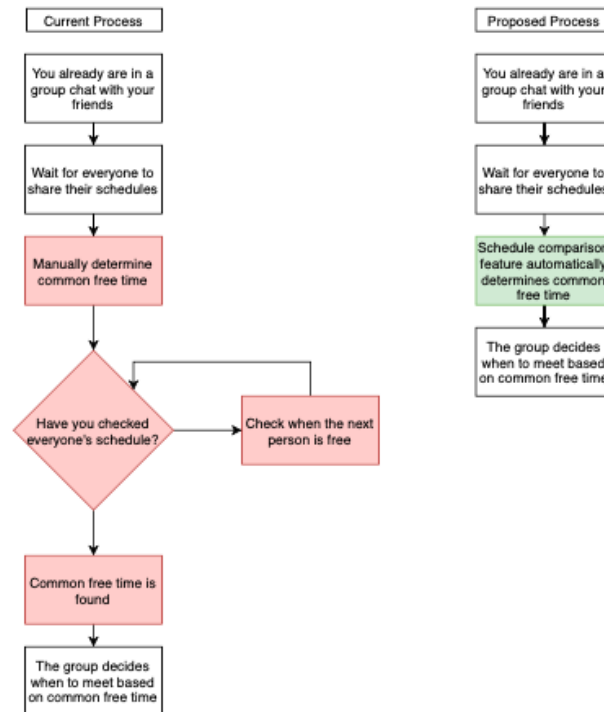


c.

2. Schedule Comparisons

- a. Current process - If wanting to decide when you and your friends have free time, then all parties must send their schedule and everyone would need to manually check when everyone else is free to determine when free time overlaps between everybody

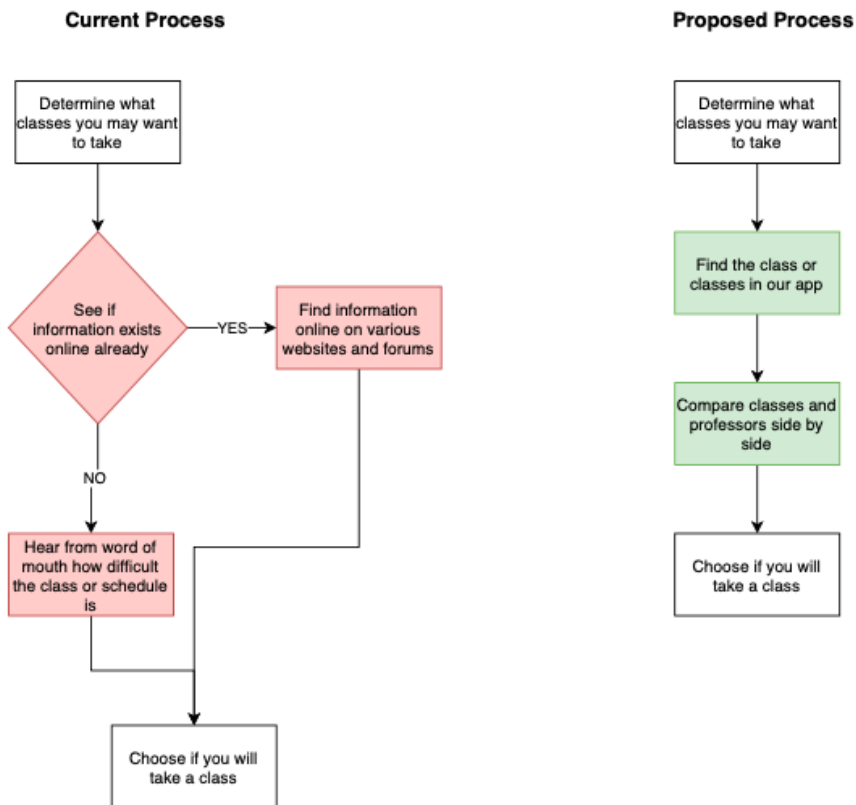
- b. Proposed process - If wanting to decide when you and your friends have free time, all schedules need to be shared and the website will do the work of finding overlapping free time for you



c.

3. Course Difficulty Ratings

- a. Current process - If wanting to find how hard a particular class or teacher is, one needs to either try and find something about it online or hear about it from word of mouth
- b. Proposed process - users can find how hard a particular course or professor is by looking at past reviews left by other users

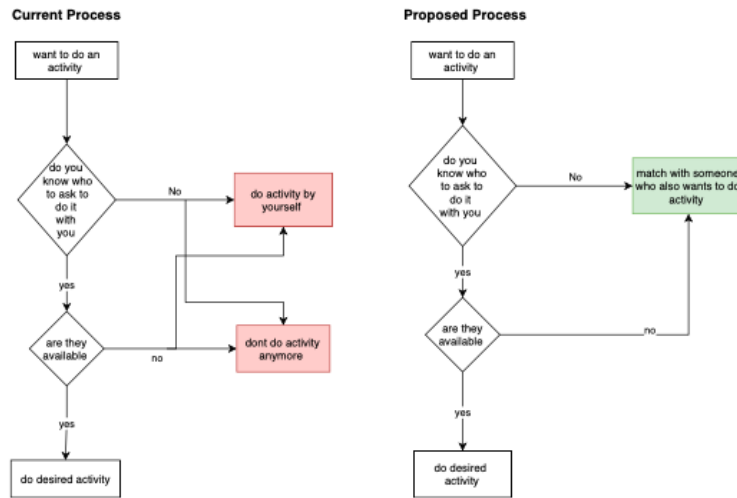


c.

4. Matching

a. Activity

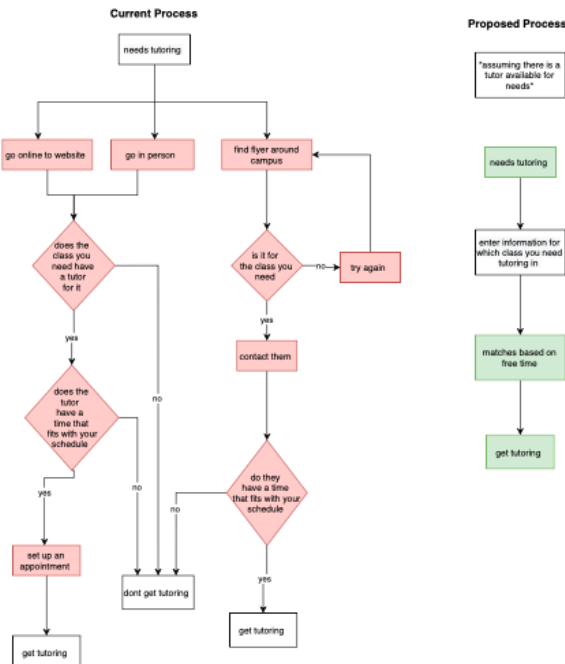
- i. Current process - if someone wants to do an activity with another person then they must know or have a person in mind to do that activity with. If that person is not available, the person has a choice to either not do that activity anymore or do it by themselves.
- ii. Proposed process - now if a user wants to do an activity, they don't even have to have someone in mind to do it with them, they simply just enter the information of which activity they would like to do and get matched with another person who wants to also do that activity



b.

c. Tutoring

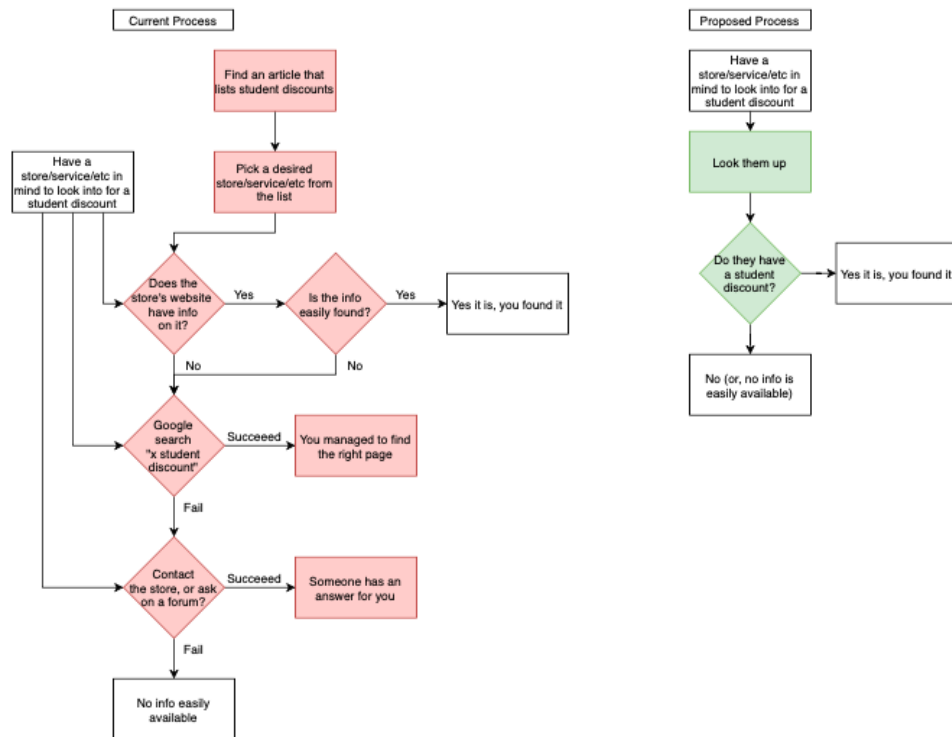
- i. Current process - if a person requires tutoring, they either need to go make an appointment with a tutor or find a flyer on campus that offers tutoring that matches their particular needs and then make sure that they have a time that can work with the tutor.
- ii. Proposed process - if a user needs tutoring, they simply enter in the information of what they need tutoring in and get matched with a person who tutors in that course and has matching free time.



d.

5. Student Discounts

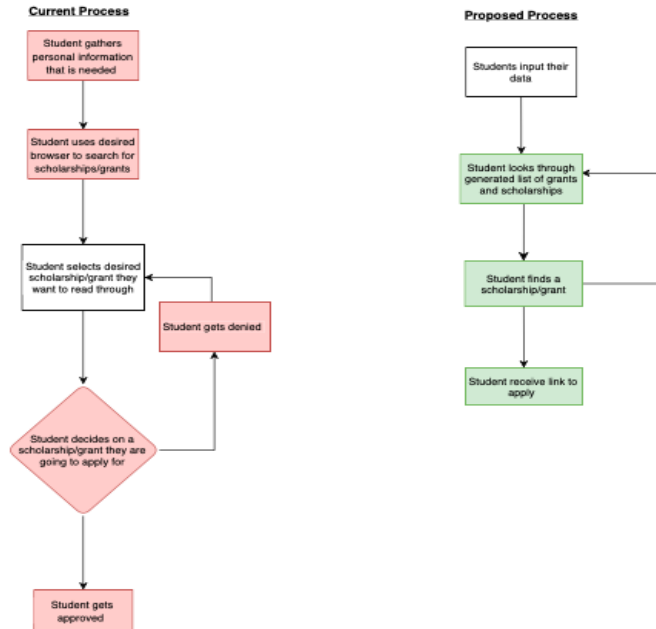
- Current process - if a user wanted to know if a particular establishment offered student discounts they would have to find an article or ad about it online or else call the establishment themselves.
- Proposed process - if a user wanted to know if a particular establishment offered student discounts they would need to look that establishment up on our website and will be able to see if they offer one.



c.

6. Aid Eligibility Estimation

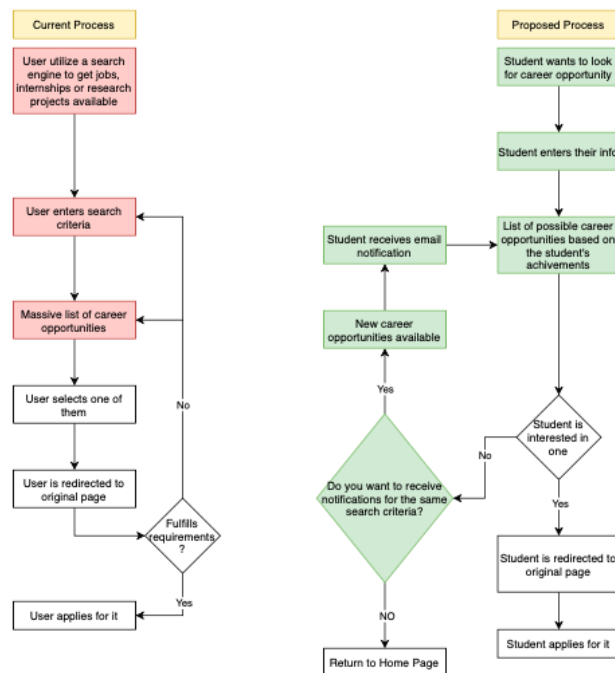
- Current process - if a user wishes to find information on a grant or scholarship, they would need to look up that information themselves and go through each one they were interested in to see if they are eligible for it
- Proposed process - if a user wishes to find information on a grant or scholarship they can enter their information and our website would generate a list of grants and scholarships they are eligible for.



c.

7. Student Career Opportunities

- Current process - if users want to look for an internship, research program or a job related to their field, they would use a search engine and get probably thousands of results
- Proposed process - After the users enter their information, they can opt to be notified about new career opportunities' requirements meet student's qualifications



c.

Functional and NonFunctional Requirements

Application Features

1. Collaborative Schedule Building
 - a. Purpose: To enable users to create class schedules together to attempt to take classes together, carpool, or participate in other activities together.
 - b. Scope: This feature will focus on generic, weekly schedules for each school term, not a monthly or yearly calendar.
 - c. Functional Requirements:
 - The system shall allow up to five users to view or edit the same schedule simultaneously
 - The system shall allow up to five schedules to be viewed or edited on the same view
 - All schedule items should have the following data visible in the schedule builder's default view:
 - Title ("Intro to Computer Science", "Work", "Drive home", etc.)
 - Start time
 - End time
 - Day(s) of week ("Mon.", "Tue., Thu.", etc.)
 - Course sections shall be able to contain the following data:
 - Category ("CECS", etc.)
 - Course number ("100", etc.)
 - Section number ("01", etc.)
 - Professor ("Smith", "John Smith", etc.)
 - The user shall be able to add a course section, commute, or miscellaneous schedule item to their schedule
 - The user shall be able to manually enter data about each schedule item
 - The user shall be able to automatically add two commutes to any day on their schedule, which have the following default values:
 - The first commute ends 15 minutes before their first schedule item for that day begins
 - The second commute begins 15 minutes after their last schedule item for that day ends
 - Both commutes have a 15 minute duration

- Commutes on the same day can be edited independently of each other
- The user shall be able to grant other users permission to view or edit their schedule
- The user shall be able to revoke permissions of other users to view or edit their own schedule
- The user shall be able to view their own schedules, and schedules that they have permission to view or edit
- Each user shall be able to see who has made changes to schedules they own

d. Nonfunctional Requirements:

- Changes made to a schedule should be visible to all users in under 10 seconds

2. Schedule Comparison

- a. Purpose: To enable users to automatically determine when they have common free time on or off campus.
- b. Scope: Compare schedules to reveal common free time throughout the day.

c. Functional Requirements:

- The user shall be able to create and view schedule comparisons
- The user shall be able to send schedule comparison results in a message
- The user shall be able to compare up to five schedules
- The user shall be able to specify some range of inactive hours (to prevent being matched based on “free time” in the middle of the night), including a default value
- Schedule comparisons shall contain the following data:
 - Day(s) of week to be compared
 - The schedules to be compared
 - Common free time on campus
 - Common free time off campus

d. Nonfunctional Requirements:

- Schedule comparisons should be calculated within 5 seconds

3. Course Difficulty Rating

- a. Purpose: users can rate the difficulty of the classes that they took, and share this with other users.
- b. Scope: This will tell users how difficult a course is for a particular student and how their experience with a particular teacher was. It will not give a definite census on a particular class and/or teacher.

c. Functional Requirements:

- Users shall be able to search for reviews of classes and professors
 - Searching requires specifying school and major
 - Specify if searching for a class or a professor
 - Display results of the search
 - Select a result and go to the page for that class or professor
- Users shall be able to select a class to view:
 - All professors who have recently taught the class, or who are teaching it in an upcoming term
 - Reviews for the class, and for professors teaching the class
 - Average difficulty rating, and average rating for individual professors
 - Recommended class pairings when applicable
- Users shall be able to select a professor to view:
 - All classes they have recently taught
 - All classes they are currently teaching, and are scheduled to teach for an upcoming term
 - Average difficulty rating, and average ratings for individual classes they teach
- Users shall be able to add a new review with the following information:
 - School
 - Course
 - Term Taken
 - Professor
 - Comments
 - A “difficulty” rating on a scale of 1-5
- Users shall be able to rate schedules for overall difficulty on a 1-5 scale
- View schedules similar to the one you are currently taking/building
- Users shall suggest pairings of classes that can be taken together
 - Users can select up to 5 classes to pair together
 - All classes require the information: School, course, term taken, and professor
 - Users must provide a comment to explain why the classes work well together
 - Users can search for popular class pairings by major

- Users can agree or disagree with any type of review
 - This like/dislike system will be used to determine which reviews and descriptions are most relevant.
- d. Nonfunctional Requirements:
 - The same reviews will be used for both the professor's review page, and for the course review page, as they are linked together.
 - Reviews on the class page can link to the professor page, and vice versa.

4. Matching

- a. Purpose: Makes learning and socializing during college simpler by helping users meet new people that have similar academic needs, social needs, and availability as them.
- b. Scope: This connects users to each other and will provide them with the ability to start talking, but not communicate for them.
- c. Functional requirements
 - The system shall allow a user to select up to five activities when creating or editing their activity profile
 - All profiles should have the following data visible in the activity profiles default view
 - Activities (the user can select up to five):
 - Studying
 - Exercising
 - Get food (on/off campus)
 - Get food (dining hall)
 - Go off campus
 - Go to specific location on campus
 - Hang out
 - Go to event
 - Other
 - The system shall allow a user to select up to six courses when creating or editing their tutoring profile
 - All profiles should have the following data visible in the tutoring profiles default view
 - Whether the user requires or offers tutoring
 - Class(es) they need tutoring in or the class(es) they offer tutoring in
 - Group or individual tutoring session

- The system shall allow a user to choose one schedule from a list of schedules they built to be used for their profiles.
 - A profile cannot be matched with another profile if there is not a schedule selected.
- Reasons why the system can match two users
 - If two users have overlapping free time (using schedule comparison)
 - Either
 - If one users tutoring needs is same as another's tutor offerings
 - If tutor can help in more than one of the tutoree's classes, then they will only be matched once, not multiple times for each course
 - If two users select the same activity
 - If two users share more than one activity they will only be matched once, not multiple times for each activity
- The system shall allow a user to:
 - See why they were matched
 - Message the person matched
- The system shall allow a user to filter through their matches
 - Reasoning for a match
 - By default matches are ordered by most recent to least recent matches
- The system shall allow users to opt out of matching if they wish to
 - By default users will be opted in to matching
 - Users won't be considered in the matching logic until they opt back in
- The system shall not match a user with a previous match again unless it is for different reasons

d. Nonfunctional requirements

- The system shall display matches in a list that takes no more than 10 seconds to show up
- When a user selects a match from their list of matches, the system shall display to the user in no more than 10 seconds
 - The reasoning for the match
 - The overlapping free time with selected match

- A message option will be available to use if they please
- The server should take less than 60 seconds to process 100 users for matching and return results to the client
- If users change their matching profile information, the updated information should be saved to the database within 1 second

5. Student Discounts

- a. Purpose: to provide users information about student discounts.
- b. Scope: Student Discounts provides the information to the users, it does not give the actual student discounts. Discounts at brick-and-mortar establishments and websites are included.
- c. Functional:
 - Users shall be able to post information on discounts
 - Required information (brick and mortar):
 - Establishment name
 - Physical address
 - Description of discount (up to 250 characters)
 - Required information (website):
 - Web address
 - Description of discount (up to 250 characters)
 - Users shall be able to view information about discounts
 - Vote to confirm discounts
 - Users can vote for an establishment/website once every 30 days
 - Users can vote again for the same establishment/website the first day of every 30 days.
 - Users can vote to confirm that a discount exists (a positive vote)
 - Users can vote to deny that a discount exists (a negative vote)
 - Users shall be able to get directions to establishments
 - Users shall be able to get access to the website's link
 - All post about the same establishment/website will be appended
 - Net positive votes will indicate that an establishment/website has a discount
 - Net negative votes will indicate that an establishment/website has no discount
 - The most recent votes will carry the most weight

- Each vote will initially have a value of 1.0 (representing “yes”), or a value of -1.0 (representing “no”)
- Every 30 days, votes will lose 10 percent of its original value

d. NonFunctional:

- Websites will be sorted by number of positive votes by default

6. Aid Eligibility Estimation

- Purpose: To allow users to input personal information and receive a list of aid items (grants or scholarships) that they might qualify for.
- Scope: To provide information about current aid eligibility to users, not to allow users to directly apply.

c. Functional Requirements:

- Users will be matched to aid items based on the following criteria:
 - Units enrolled in
 - Age.
 - Disability status.
 - Expected graduation date.
 - First generation college student.
 - GPA.
 - Income.
 - Major.
 - Number of degrees.
 - Tax filing status (whether or not the student is a dependent).
 - Veteran status.
 - Years spent in school to date.
 - If the student is a participant in any school or community program(s).
 - If the student belongs to a minority group.
- The user shall input their information to determine which aid items they are eligible for
- The system shall parse aid information to determine eligibility of users
- Aid information shall be obtained from an external source
- The system shall re-authenticate the user before accessing the user’s personal info
- Expired aid items will be automatically removed
- Updates daily newly formed aid items to be populated when they get imported into the database

- Users will be able to sort through aid items by amount, deadline date, and whether or not the aid item has special requirements
- d. Nonfunctional Requirements:
 - An aid item's information should be opened within 5 seconds
 - The application to view aid items should load within 10 seconds

7. Student Career Opportunities

- a. Purpose: users will be able to see opportunities for jobs, internships, research programs, and other activities they may qualify for based on their current student status (e.g. freshman or senior).
- b. Scope: The scope is to provide information, rather than to enable users to directly apply.
- c. Functional:
 - Users shall be able to view jobs, internships, and research programs and filter search results using the following information:
 - Job title
 - Submission deadline (if applicable)
 - Application type
 - Salary
 - Zip code
 - Users shall be able to input the following information about themselves:
 - Their student status (freshman, sophomore, junior, senior)
 - Their number of units completed
 - Their major
 - Their certifications (if applicable, the user shall enter the name of the certification)
 - Their past internships (if applicable, the user shall enter the name of internship)
 - Their past research programs (if applicable, the user shall enter the name of research program)
 - The user shall receive email alerts when there are more available career opportunities
 - Users shall be able to sort search results by:
 - Date posted
 - Salary (USD)
 - Distance (miles)
 - Deadline

- The system shall redirect users to official website to apply
- users shall be able to hide career opportunities individually
- d. NonFunctional:
 - users status shall be considered for career opportunities requirements (freshman, sophomore, junior, or senior)
 - Completed units shall be considered for career opportunities requirements (0-999)
 - By default, search results will be sorted by date posted
 - The search result will contain a maximum of 250 available career opportunities
 - The search result page will display a maximum of 10 career opportunities per page
 - Display message when career opportunities are available/ no longer available

8. Schedule of Classes Information

- a. Purpose: Necessary for the implementation of other features, this will create a database of all classes offered.
- b. Scope: This is for planning current and future terms, and finding information on relevant courses. This is not for actual enrollment.
- c. Functional requirements
 - Obtain class information, including course name, number, description, prerequisites/corequisites.
 - Obtain course information, including day of week, time of day, start date, end date, term, and professor.
 - Search for classes based on any of the information.
 - Display classes with relevant information, click for more information
 - Show connections to the Schedule Building and Comparison feature, and the Course Difficulty Rating feature.
 - Information shall be updated once a month, or when user notifies that the information is outdated
- d. Nonfunctional requirements
 - Obtain information by obtaining database access from the schools
 - Create web crawling when direct database access is not available
 - Searching can be completed in under 5 seconds

9. Messaging

- a. Purpose: To allow users to directly communicate with each other via text.
- b. Scope:
- c. Functional Requirements
 - Users shall be able to send messages to directly to other users
 - Users shall be able to create groups with up to four other users to message
 - All users in a group shall receive all messages sent to that group
 - Users shall be able to leave a group to stop receiving messages sent to it
 - If a user has blocked another user in a group that they are both in, neither user will be able to see each other's messages
 - Messages shall only be able to be seen by the sender and recipient(s)
 - Users shall be able to search for other users by username to message or add to a group message
 - Users shall be able to add and remove other users to and from a contacts list
 - Users shall be able to block users that they do not wish to receive messages from
 - Users shall be able to unblock users that they have blocked
- d. Nonfunctional Requirements
 - Messages shall be up to 1000 characters in length

Core Features

- Registration
 - Purpose: To produce an account to associate the user with and enable them to log in
 - Scope: Anyone with a valid .edu email address can register. We are not checking that the email address belongs to a college student.
 - Functional Requirements
 - The user shall provide the following information to register:
 - Desired username
 - Email address
 - Desired password
 - School
 - The system shall verify that both the user's desired username and email address are not associated with an existing account

- The system shall verify that the user's email address is from a .edu domain
 - The system shall verify that the user's desired password meets the following complexity requirements:
 - At least one uppercase letter
 - At least one lowercase letter
 - At least one number
 - At least one symbol
 - Cannot contain the user's desired username
 - Cannot contain the user's email address (the entire address, the substring preceding the @ symbol, or the substring following the @ symbol)
 - The user must verify ownership of their email address to complete registration
 - The system shall notify the user of the success or failure of registration
- Nonfunctional Requirements
 - Email ownership verification occurs after completing the registration form
- Login
 - Purpose: To identify the user and enable them to access application features and their personal information
 - Scope: To authenticate the user on their current device.
 - Functional Requirements:
 - The system shall verify that a valid email address and its associated password were correctly entered before logging the user in
 - Nonfunctional Requirements:
 - A link or button to the registration view must be present
- Logout
 - Purpose: To log the user out and revoke access to their account from the current device
 - Scope: To deauthenticate the user on their current device.
 - Functional Requirements:
 - The session is ended and the user is logged out of their account
 - The system shall understand the closing of the browser as wanting to logout
 - Upon being logged out, the user is returned to the login view
 - Nonfunctional Requirements:
 - Logging a user out should be completed within 10 seconds

- Error handling
 - Purpose: When errors occur, including front-end and back-end, the website will be able to continue working for that user and others without shutting down.
 - Scope: This will handle errors in user input, site navigation, and database access, among others. It can also handle some general errors, but it cannot handle errors not previously accounted for during development
 - Functional Requirements:
 - Any user input error should not prevent the user from attempting to enter the correct information
 - All errors are logged and documented in a consistent format
 - Any error will allow the site to continue running without crashing
 - Nonfunctional Requirements:
 - All errors shall be reported to the user within 1 second

Glossary

- Aid - umbrella term for grants and scholarships
- Aid item(s) - umbrella term for an individual grant or scholarship
- Application type - whether the application is an internship, part time or full time job offering
- Average difficulty rating - all ratings for a particular class or professor averaged together
- Career opportunities - potential job and internship listings to help users get a job
- Class pairings - grouping similar classes together
- Commute - travel distance a user takes between home and school or work and school
- Completed units - total amount of units that a user has completed up to the current moment in time, not counting current units enrolled in
- Corequisite - a course that a user can take before or at the same time as another course
- Establishments - business organization that exists in a physical location or online
- Inactive hours - hours that a user will be sleeping during/wish to not be bothered and therefore will not be available
- Negative vote - a vote that confirms an establishment does not have a student discount
- Net negative vote - more negative than positive votes. Indicates that there is no student discount
- Net positive vote - more positive than negative votes. Indicates that there is a student discount
- Positive vote - a vote that confirms an establishment has a student discount
- Prerequisite - a course needed to take before a user takes another course
- Registered Users - the users that created an account with the system
- Schedule items - courses, commutes, work, misc items a user can add to their schedule
- Schedule of classes - overall list of classes offered by a university in a particular term
- Student Discount - reduced priced item or service given to people enrolled in school
- Student status - whether user is a freshman, sophomore, junior, or senior
- System - refers to the Student Multi-Tool application as a whole
- Term - a fixed period of time in which a student takes classes, typically lasting weeks

References

1. SOAR Colleges and Majors

https://web.csulb.edu/divisions/students/soar/workshop/colleges_majors.htm