Business Requirements Document

Student Multi-Tool

2.27.2022

Team Marvel
Albert Toscano
Audrey Brio
Bradley Nickle
Devarsh Patel
Jacob Delgado
Joseph Cutri
Michael Kriesel (Team Leader)
SzeMan Tang

Version History

Date	Changes	Approval Date
10/06/2021	Initial Draft	
10/18/2021	Added requirement codes based on feature name abbreviations	
11/05/2021	Added requirements for user management and logging	
12/15/2021	Added requirements for rest of core features, updated based on feedback received.	
2/06/2022	Adding new application features to accommodate new team members.	
2/25/2022	Readjustments for final product	

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. We will be using the 24 hour time format for this project.
- 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
- Event Planning
    - PST time zone
- Recipe Sharing
    - Will be taking the average of prices from stores located within 5 miles from CSULB
- 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

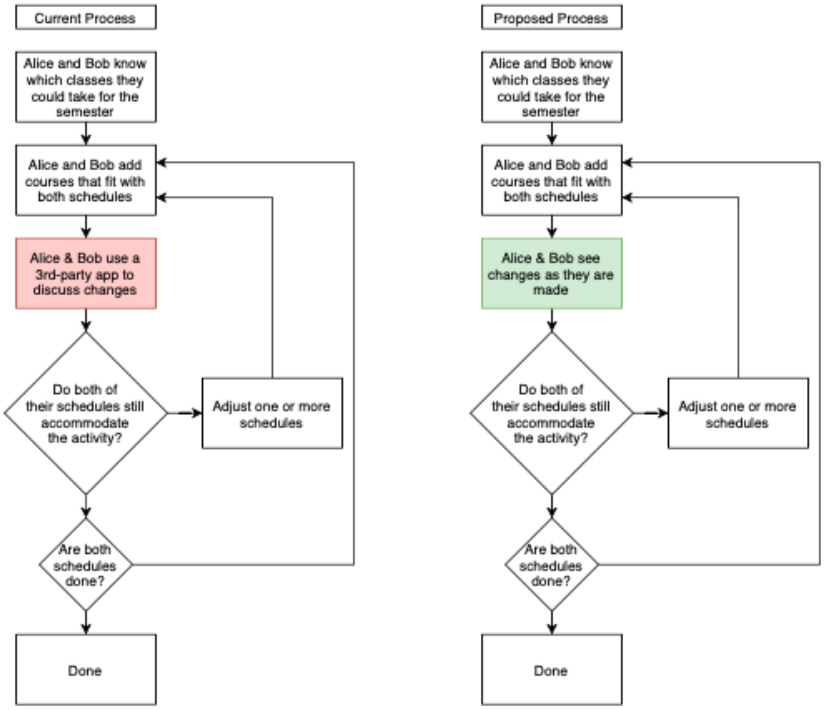
Out of Scope

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.

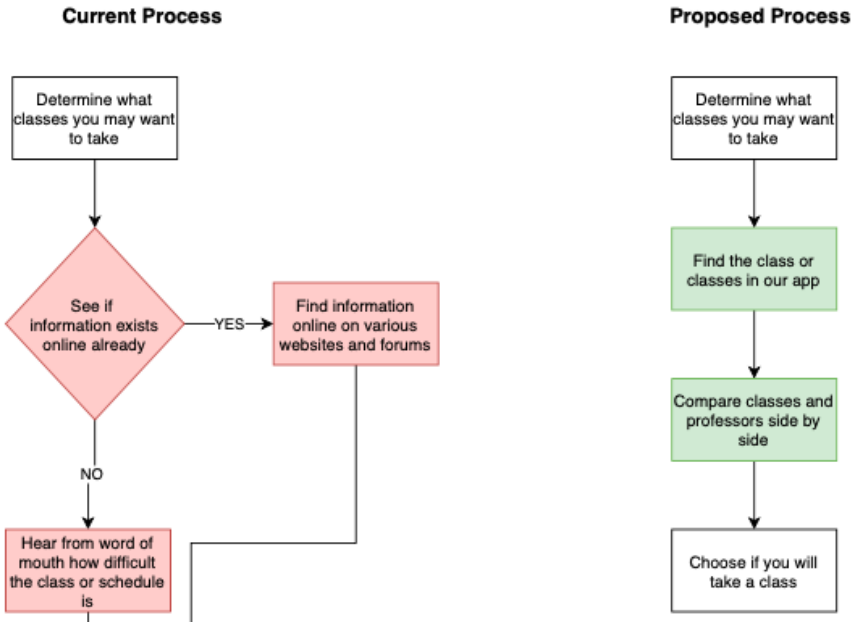


    c.
2. Schedule Comparisons
    a. Current process - If wanting to decide when you and your friends have free time, then all parties must send their schedule and everyone would need to manually check when everyone else is free to determine when free time overlaps between everybody

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

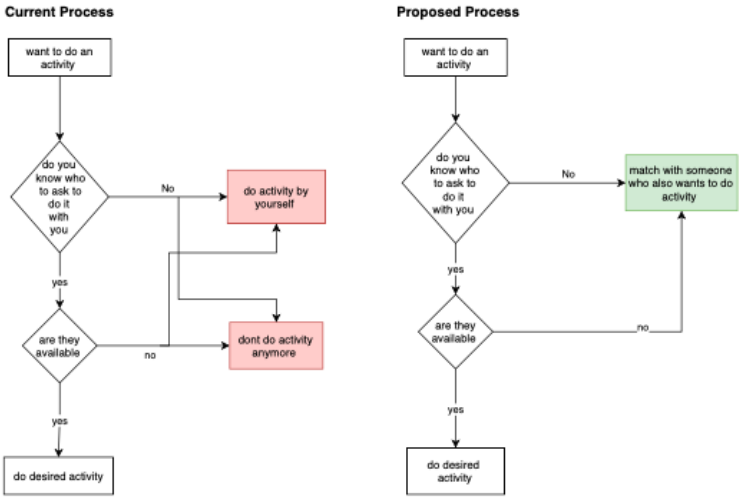
c.

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.

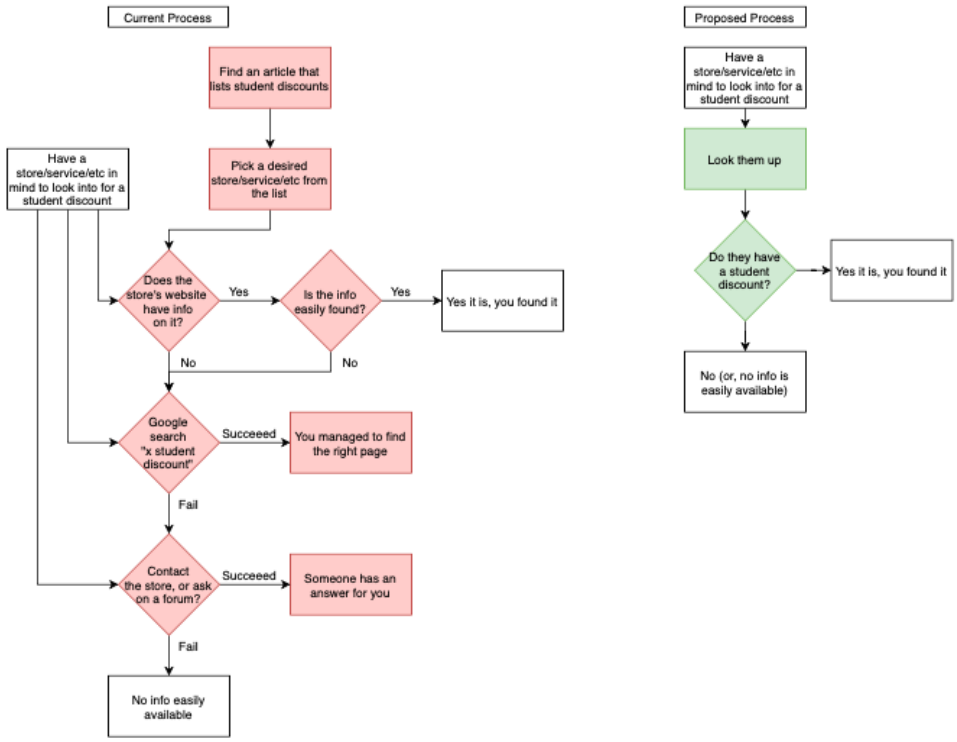
c. Tutoring

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.

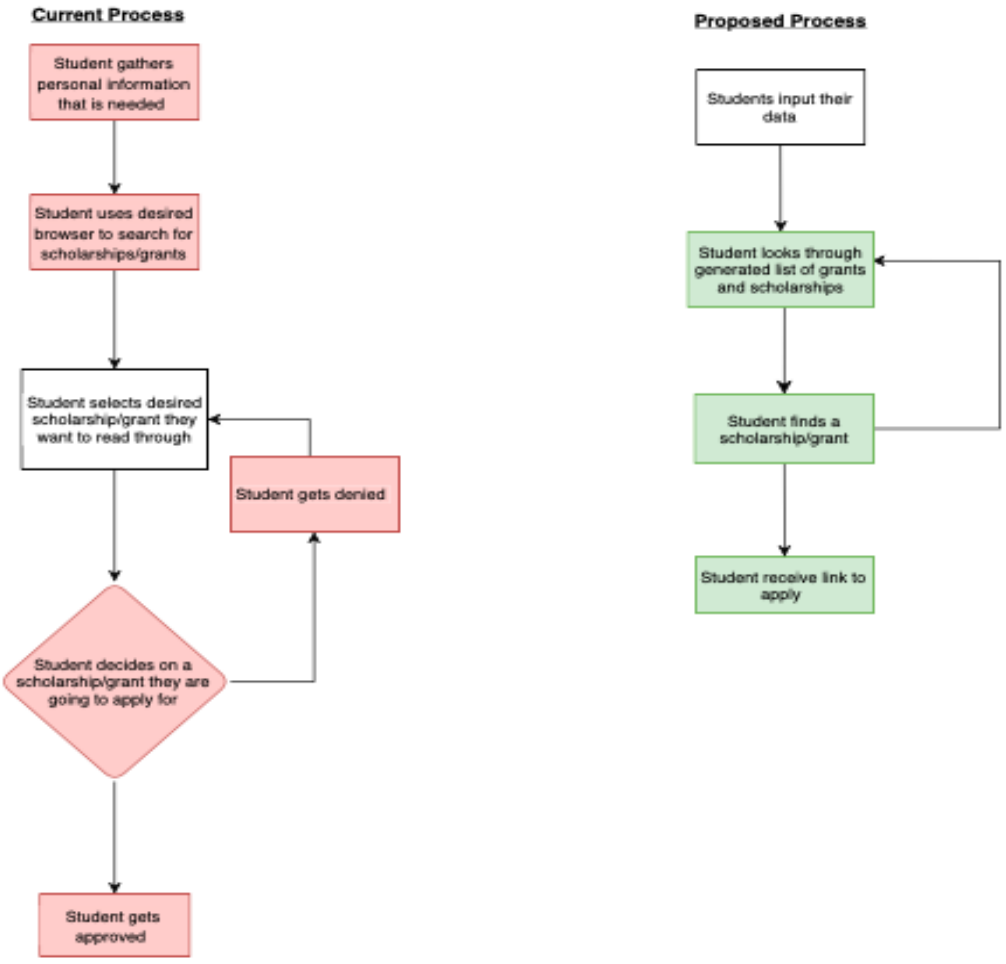
ii. Proposed process - if a user needs tutoring, they simply enter in the information of what they need tutoring in and get matched with a person who tutors in that course and has matching free time.

d.

5. Student Discounts
    a. 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.
    b. 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
    a. 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 there were interested in to see if they are eligible for it
    b. 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.

Functional and NonFunctional Requirements

Application Features

1. Collaborative Schedule Builder (SB)
    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:
        i. SB01: The system shall allow up to five users to view or edit the same schedule simultaneously

ii. SB02: The system shall allow up to five schedules to be viewed or edited on the same view
iii. SB03: All schedule items should have the following data visible in the schedule builder's default view:
    1. Title ("Intro to Computer Science", "Work", "Drive home", etc.)
    2. Start time
    3. End time
    4. Day(s) of week ("Mon.", "Tue., Thu.", etc.)
iv. SB04: Course sections shall be able to contain the following data:
    1. Category ("CECS", etc.)
    2. Course number ("100", etc.)
    3. Section number ("01", etc.)
    4. Professor ("Smith", "John Smith", etc.)
v. SB05: The user shall be able to add a course section, commute, or miscellaneous schedule item to their schedule
vi. SB06: The user shall be able to manually enter data about each schedule item
vii. SB07: The user shall be able to automatically add two commutes to any day on their schedule, which have the following default values:
    1. The first commute ends 15 minutes before their first schedule item for that day begins
    2. The second commute begins 15 minutes after their last schedule item for that day ends
    3. Both commutes have a 15 minute duration
viii. SB08: Commutes on the same day can be edited independently of each other
ix. SB09: The user shall be able to grant other users permission to view or edit their schedule
x. SB10: The user shall be able to revoke permissions of other users to view or edit their own schedule
xi. SB11: The user shall be able to view their own schedules, and schedules that they have permission to view or edit
xii. SB12: Each user shall be able to see who has made changes to schedules they own

d. Nonfunctional Requirements:
    i. SB13: Changes made to a schedule should be visible to all users in under 10 seconds

2. Schedule Comparison (SC)
    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:
        i. SC01: The user shall be able to create and view schedule comparisons
        ii. SC02: The user shall be able to send schedule comparison results in a message
        iii. SC03: The user shall be able to compare up to five schedules
        iv. SC04: 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
        v. SC05: Schedule comparisons shall contain the following data:
            1. Day(s) of week to be compared
            2. The schedules to be compared
            3. Common free time on campus
            4. Common free time off campus
    d. Nonfunctional Requirements:
        i. SC06: Schedule comparisons should be calculated within 5 seconds
3. Course Reviews (CR)
    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:
        i. CR01: Users shall be able to search for reviews of classes and professors
            1. Searching requires specifying school and major
            2. Specify if searching for a class or a professor
            3. Display results of the search
            4. Select a result and go to the page for that class or professor
        ii. CR02: Users shall be able to add a new review with the following information:
            1. School
            2. Major
            3. Course
            4. Term Taken

5. Professor
6. Comments
7. A “difficulty” rating on a scale of 1-5

iii. CR03: Users shall be able to rate schedules for overall difficulty on a 1-5 scale

iv. CR04: Users shall suggest pairings of classes that can be taken together
    1. Users can select up to 5 classes to pair together
        a. All classes require the information: School, course, major, term taken, and professor
    2. Users must provide a comment to explain why the classes work well together
    3. Users can search for popular class pairings by major
    4. Users can search for pairings for a specific class

v. CR05: Users can agree or disagree with any type of review
    1. This like/dislike system will be used to determine which reviews and descriptions are most relevant.

d. Nonfunctional Requirements:
    i. CR06: The same reviews will be used for both the professor’s review page, and for the course review page, as they are linked together.
    ii. CR07: Reviews on the class page can link to the professor page, and vice versa.

4. Matching (MA)
    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
        i. MA01: The system shall allow a user to select up to five activities when creating or editing their activity profile
        ii. MA02: All profiles should have the following data visible and selectable in the activity profiles default view
            1. Activities (the user can select up to five):
                a. Studying
                b. Exercising
                c. Get food (on/off campus)

d. Get food (dining hall)
e. Go off campus
f. Go to specific location on campus
g. Hang out
h. Go to event
i. Other

iii. MA03: The system shall allow a user to select up to six courses when creating or editing their tutoring profile

iv. MA04: All profiles should have the following data visible in the tutoring profiles default view
    1. Whether the user requires or offers tutoring
    2. Class(es) they need tutoring in or the class(es) they offer tutoring in
    3. Group or individual tutoring session

v. MA05: The system shall allow a user to choose one schedule from a list of schedules they built to be used for their profiles.
    1. A profile cannot be matched with another profile if there is not a schedule selected.

vi. MA06: Reasons why the system can match two users
    1. If users must have the same university specified in their settings
    2. If two users have overlapping free time (using schedule comparison)
    3. Either
        a. If one users tutoring needs is same as another's tutor offerings
            i. 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
        b. If two users select the same activity
            i. If two users share more than one activity they will only be matched once, not multiple times for each activity

vii. MA07: The system shall allow a user to:
    1. See why they were matched
    2. Message the person matched

viii. MA08: The system shall allow a user to filter through their matches

1. Reasoning for a match
2. By default matches are ordered by most recent to least recent matches

ix. MA09: The system shall allow users to opt out of matching if they wish to
1. By default users will be opted in to matching
2. Users won't be considered in the matching logic until they opt back in

x. MA10: To begin matching, users must select generate matches button

d. Nonfunctional requirements
i. MA11: The system shall display matches in a list that takes no more than 10 seconds to show up
ii. MA12: When a user selects a match from their list of matches, the system shall display to the user in no more than 10 seconds
1. The reasoning for the match
2. The overlapping free time with selected match
3. A message option will be available to use if they please
iii. MA13: The server should take less than 60 seconds to process 100 users for matching and return results to the client
iv. MA14: If users change their matching profile information, the updated information should be saved to the database within 1 second

5. Student Discounts (DI)
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:
i. DI01: Users shall be able to post information on discounts
1. Required information (brick and mortar):
a. Establishment name
b. Physical address
c. Description of discount (up to 250 characters)
2. Required information (website):
a. Web address
b. Description of discount (up to 250 characters)
ii. DI02: Users shall be able to view information about discounts
iii. DI03: Vote to confirm discounts

1. Users can vote for an establishment/website once every 30 days
    a. Users can vote again for the same establishment/website the first day of every 30 days.
2. Users can vote to confirm that a discount exists (a positive vote)
3. Users can vote to deny that a discount exists (a negative vote)

iv. DI04: Users shall be able to get directions to establishments
v. DI05: Users shall be able to get access to the website's link
vi. DI06: All post about the same establishment/website will be appended
vii. DI07: Net positive votes will indicate that an establishment/website has a discount
viii. DI08: Net negative votes will indicate that an establishment/website has no discount
ix. DI09: The most recent votes will carry the most weight
1. Each vote will initially have a value of 1.0 (representing "yes"), or a value of -1.0 (representing "no")
2. Every 30 days, votes will lose 10 percent of its original value

d. NonFunctional:
    i. DI10: Websites will be sorted by number of positive votes by default

6. Aid Eligibility Estimation (AE)
    a. Purpose: To allow users to input personal information and receive a list of aid items (grants or scholarships) that they might qualify for.
    b. Scope: To provide information about current aid eligibility to users, not to allow users to directly apply.
    c. Functional Requirements:
        i. AE01: Users will be matched to aid items based on the following criteria:
            1. Units enrolled in
            2. Age.
            3. Disability status.
            4. Expected graduation date.
            5. First generation college student.
            6. GPA.
            7. Income.
            8. Major.
            9. Number of degrees.

10. Tax filing status (whether or not the student is a dependent).
11. Veteran status.
12. Years spent in school to date.
13. If the student is a participant in any school or community program(s).
14. If the student belongs to a minority group.

ii. AE02: The user shall input their information to determine which aid items they are eligible for
iii. AE03: The system shall parse aid information to determine eligibility of users
iv. AE04: Aid information shall be obtained from an external source
v. AE05: The system shall re-authenticate the user before accessing the user's personal info
vi. AE06: Expired aid items will be automatically removed
vii. AE07: Updates daily newly formed aid items to be populated when they get imported into the database
viii. AE08: 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:
   i. AE09: An aid item's information should be opened within 5 seconds
   ii. AE10: The application to view aid items should load within 10 seconds

7. Schedule of Classes Information (CI)
   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
      i. CI01: Obtain class information, including course name, number, description, prerequisites/corequisites.
      ii. CI02: Obtain course information, including day of week, time of day, start date, end date, term, and professor.
      iii. CI03: Search for classes based on any of the information.
         1. Display classes with relevant information, click for more information
      iv. CI04: Show connections to the Schedule Building and Comparison feature, and the Course Difficulty Rating feature.

v. CI05: Information shall be updated once a month, or when user notifies that the information is outdated

d. Nonfunctional requirements
    i. CI06: Obtain information through crowdsourcing when students enter information on their own class
    ii. CI07: Searching can be completed in under 5 seconds

8. Messaging (ME)
    a. Purpose: To allow users to directly communicate with each other via text.
    b. Scope:
    c. Functional Requirements
        i. ME01: Users shall be able to send messages to directly to other users from a list of contacts
        ii. ME02: Users shall be able to create groups with up to four other users to message
            1. All users in a group shall receive all messages sent to that group
            2. Users shall be able to leave a group to stop receiving messages sent to it
            3. 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
        iii. ME03: Messages shall only be able to be seen by the sender and recipient(s)
        iv. ME04: Users shall be able to search for other users by username to message or add to a group message
        v. ME05: Users shall be able to add and remove other users to and from a contacts list
        vi. ME06: Users shall be able to block users that they do not wish to receive messages from
            1. Users shall be able to unblock users that they have blocked
    d. Nonfunctional Requirements
        i. ME07: Messages shall be up to 1000 characters in length
        ii. ME08: Messages will be secured using an approved encryption method
        iii. ME09: When messaging a contact, the application will automatically try to reconnect if the connection is lost.
    e. Expansions Points:
        i. Notifications of unread messages

ii. Ability to send non-text and file attachments.

9. Book Selling (BKS)
    a. Purpose: To allow users to directly purchase books from verified and unverified students that have taken the course.
    b. Scope: CSULB student books from the most recent fall 2021 semester or older.
    c. Functional Requirements
        i. BKS01: Users shall be able to view the contact information of the seller.
        ii. BKS02: Users shall be able to update their listing.
        iii. BKS04: Users shall be able to search for books by book title and edition number or ISBN- 10 number.
        iv. BKS05: Users shall be able to have their posts promoted if they have been confirmed to have taken the course based on the schedule builder.
    d. Nonfunctional Requirements
        i. BKS06: Images of books will be limited to a minimum of 2 and a maximum of 8.
        ii. BKS07: Each image must be less than 5MB in size and must be in JPEG format.
        iii. BKS08: Each post is required to disclose the:
            1. Condition of the book.
            2. Edition of the book, if applicable.
            3. The seller's asking price for the book.
            4. The title of the book.
            5. The book's ISBN-10 number.
            6. Whether or not the seller has been confirmed to have taken the course based on the schedule builder.
10. Event Planning (EP)
    a. Purpose: To allow users to efficiently post future events based on their allotted free time from the schedule builder as recommendation.
    b. Scope: CSULB students who are currently taking courses.
    c. Functional Requirements
        i. EP01: Users shall be able to create and view event posts
            1. Required information
                a. Name of event
                b. Time of event

c. Date of event
d. Location of event
e. Optional description of event (Up to 1000 characters)

ii. EP02: Users shall be able to update the event
1. Update options will notify new updates or changes of events to students.

iii. EP03: On campus events will be prioritized over off campus due to easier accessibility.

iv. EP04: Users shall be able to add an event to their calendar via an outlook link.

d. Nonfunctional Requirements
i. EP05: Posts will be sorted by closest distance as default.
ii. EP06: Event post shall be shown within 5 seconds after created.
iii. EP07: All dates and times will be using the PST zone.

11. Recipe Sharing (RE)
a. Purpose: Allow students who are on a budget to find healthy and tasty recipes.
b. Scope: Using one or more stores within a 5 mile radius from CSULB to find the average of price per item
c. Functional Requirements
i. RE01: Students shall be able to post recipes containing the following information:
1. Name
2. Category (vegetarian, vegan, breakfast, dinner, etc...)
3. Caloric value
4. Overall price
5. Date posted
6. Meals it makes (for one)
7. Step-by-step instructions (up to 5,000 characters)

ii. RE02: Students will be able to search recipe posts by
1. Name
2. Category
3. Overall price
4. Caloric Value

iii. RE03: Students will be able to add in their daily budget and how many meals per day they are looking at recipes for

d. Nonfunctional Requirements

i. RE04: By default posts will be sorted by a calculation of most calories per dollar
ii. RE05: Dates posted will be used to estimate the current market price of the meal and show it to users.
    1. Year over year and month over month inflation data will be taken from the Bureau of Labor Statistics. https://www.bls.gov/cpi/
    2. Current market prices will be taken as an average from stores located within a 5 mile radius to CSULB

12. Automated Moderating (AM)
    a. Purpose: Prevents users from posting offensive content.
    b. Scope: Looks for certain words and phrases in any body of text up to 2000 characters in length.
    c. Functional Requirements
        i. AM01: The system shall maintain a blacklist of words and phrases (referred to in this document as "terms") that are not allowed to be posted.
        ii. AM02: The system shall stop users from creating publicly visible posts that contain blacklisted terms.
        iii. AM03: The system shall notify users with a message box or alert when their post contains blacklisted terms.
        iv. AM04: The system shall censor blacklisted terms in private or group messages.
        v. AM05: To minimize user evasion of this feature by replacing letters with numbers, the system shall be capable of detecting blacklisted terms that contain one or more of the following replacements:
            1. "O" or "o" being replaced with "0"
            2. "I" or "i" being replaced with "1" or "!"
            3. "L" or "l" being replaced with "1" or "!"
            4. "E" or "e" being replaced with "3"
            5. "A" or "a" being replaced with "4"
            6. "S" or "s" being replaced with "5" or "$"
        vi. AM06: The detection of blacklisted terms shall not be case-sensitive.
    d. Nonfunctional Requirements
        i. AM07: Any blacklisted term in a message will be censored with a quantity of asterisks equal to the length of the terms.

ii. AM08: Checking a text for blacklisted terms shall not take longer than 5 seconds.
iii. AM09: The blacklist will contain at least common profanity, insults, and slurs.
iv. AM10: Admin users shall be able to update the blacklist by downloading the current blacklist and then uploading a revised edition in .txt format.
v. AM11: When applicable, users shall be informed of which blacklisted terms their post contains.

Core Features

1. Registration (RE)
    a. Purpose: To produce an account to associate the user with and enable them to log in
    b. Scope: Anyone with a valid .edu email address can register. We are not checking that the email address belongs to a college student.
    c. Preconditions:
        i. User must not have an active authenticated session
        ii. User must be on account creation view
    d. Functional Requirements
        i. RE01: The user shall provide the following information to register:
            1. Desired username
            2. Email address
            3. Desired pass phrase
            4. School
        ii. RE02: The system shall verify that both the user's desired username and email address are not already in use by an existing account
        iii. RE03: The system shall verify that the user's email address is from a .edu domain
        iv. RE04: The system shall verify that the user's desired pass phrase meets the following complexity requirements:
            1. At least eight total characters
            2. Blank space
            3. A-Z
            4. a-z
            5. 0-9
            6. ,.@!

v. RE05: The system shall verify that valid username consists only of the following characters
    1. a-z
    2. 0-9
    3. ,.!@
vi. RE06: The user must verify ownership of their email address to complete registration
vii. RE07: The system shall notify the user of the success or failure of registration
viii. RE08: The system shall not allow system administrators to be created using the Create an Account feature
ix. RE09: The system shall store all user accounts in a persistent data store.

e. Nonfunctional Requirements
    i. RE10: Email ownership verification occurs after completing the registration form
    ii. RE11: Failures from registration shall not cause the system to go offline

2. Login (LI)
    a. Purpose: To identify a valid user and authenticate them to be able to access application features and their personal information
    b. Scope: To authenticate any user on their current device attempting to use the system
    c. Preconditions
        i. User must not already have an authenticated session within the system on the current device
        ii. User must be on login view or attempting to access protected resources
    d. Functional Requirements:
        i. LI01: The system shall verify that a valid student email and its associated pass phrase were correctly entered
            1. A one time password will be sent to associated student email that the user will use to login
        ii. LI02: The system shall only allow users with valid username and one time password to enter into the system
            1. Valid usernames consist of following
                a. a-z
                b. 0-9
                c. .,@!

2. One time passwords as defined in NIST SP 800-63b section 5.1.4.1
    a. OTP is changed upon every successful use
    b. OTP expires every 2 minutes
    c. OTP must be at minimum 8 characters
    d. Valid characters will consist of the following:
        i. a-z
        ii. A-Z
        iii. 0-9

iii. LI03: The system shall keep an account disabled until the user or administrator performs a successful account recovery
    1. Upon a successful recovery, the fail count will be reset to 0

e. Nonfunctional Requirements:
    i. LI04: The system shall allow a maximum of 5 attempts within a 24 hour time period to authenticate an account
        1. After the 5th failed attempt, the account will be disabled
    ii. LI05: A 24 timer will begin after the first failed authentication attempt
        1. If the account is not disabled, the fail count will reset to 0
    iii. LI06: For each failed attempt, the account undergoing authentication and the IP address that initiated the authentication request will be recorded
    iv. LI07: A link or button to the registration view must be present
    v. LI08: System failures from authentication must not result in the system going offline

3. Logout (LO)
    a. Purpose: To end the user's authenticated session on their current device
    b. Scope: Any authenticated user
    c. Preconditions
        i. User must have an active authenticated session on the device, otherwise the user is unable to perform the operation
        ii. User must be on the view with logout option
    d. Functional Requirements:
        i. LO01: The session is ended and the user is logged out of their account
            1. The system shall understand the closing of the browser as the user's intent to logout
        ii. LO02: Upon being logged out, the user is returned to the login view
    e. Nonfunctional Requirements:
        i. LO03: Logging a user out should be completed within 5 seconds

        ii. LO04: Failures caused by logging out shall not cause the system to go offline
4. Error handling (EH)
    a. 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.
    b. Scope: This will handle errors across the entire system
    c. Functional Requirements:
        i. EH01: Any user input error should not prevent the user from attempting to enter the correct information
        ii. EH02: All errors are logged and documented in a consistent format
        iii. EH03: Any failure from this feature will not cause the system to go offline
    d. Nonfunctional Requirements:
        i. EH04: All errors shall be reported to the user within 5 seconds
        ii. EH05: The following system errors are allowed:
            1. Web Server loses internet access
            2. AWS has an outage
        iii. EH06: System failures from error handling must not result with system going offline
5. User Management (UM)
    a. Purpose: A mechanism for administration of any user account
    b. Scope: Any system administrator user will be able to create, delete, update, enable, and/or disable user accounts.
    c. Preconditions:
        i. User must have an active authenticated session
        ii. User must be on management view
        iii. User must be a system administrator
    d. Functional Requirements:
        i. UM01: All UM operations are applied to the persistent data store
        ii. UM02: The system shall require at least one administrator at all times
            1. Only administrators will have access to UM view
            2. Only administrators will be able to create or add other users as system administrators
        iii. UM03: The system shall allow an administrator to
            1. Create a single user
            2. Delete a single user
            3. Update a single user
            4. Enable a single user
            5. Disable a single user

iv. UM04: The system shall allow for no more than 10 thousand bulk operations within the same request
    1. Requests can be made through an uploaded file extract no larger than 2 GB in size
v. UM05: The system shall allow an administrator to view and modify all user profiles within the system

e. Nonfunctional Requirements:
    i. UM06: A user will be logged out of their session if there is no activity after 15 mins.
    ii. UM07: All users should be able to have a page load within 5 secs.
    iii. UM08: Single UM operation should be performed in 5 seconds
    iv. UM09: Admin users should be able to perform 10 thousand tasks within 60 secs.
    v. UM10: Failures from User Management shall not cause the system to go offline

6. Logging (LG)
    a. Purpose: To generate information about events that occur while the system is running, to aid in analysis, security, performance, and other statistical and diagnostic purposes.
    b. Scope: This feature is a solution for creating logs and storing them. It logs events and information about them.
    c. Preconditions:
        i. Persistent data store must be active
        ii. Persistent data store must be accessible by the system
        iii. Persistent data store must have storage capacity for log entries
    d. Functional Requirements:
        i. LG01: Log entries must be immutable.
        ii. LG03: All logs shall contain the following information:
            1. The time and date (UTC) that the log was generated
            2. The username of the user
            3. A description or message
            4. One of the following classifications:
                a. Debug: information about system status
                b. Error: information about errors
                c. Info: information about normal user/system events
                d. Warning: information that may cause system failures
            5. One of the following abstraction layers:
                a. UI
                b. Security
                c. Business

            d. Error
            e. Data Access
        e. Nonfunctional Requirements:
            i. LG02: Logging should be implemented in a generic way that can be applied, extended, or used systemwide.
            ii. LG04: Logging shall not block users from performing any interaction with the system.
            iii. LG05: Logs should be created in a way that allows for searching and/or querying once stored.
            iv. LG06: System failures from logging shall not cause the system to go offline.
            v. LG07: The logging process shall take no more than 5 seconds to complete, starting from invocation.
7. Archiving
    a. Purpose: A way to store log entries long-term without taking up system resources
    b. Scope: All log entries within the system
    c. Preconditions:
        i. Persistent data store must be active
        ii. Persistent data store must be accessible by the system
        iii. Archival destination must have storage capacity
    d. Functional Requirements:
        i. AR01: Archiving must execute at 00:00:00 AM on the 1st of the month
        ii. AR02: Archiving must offload all entries older than 30 days
        iii. AR03: Archiving must offload entries from the main data store to another location, a file store
        iv. AR04: Archiving must consolidate and compress entries being archived
        v. AR05: Archiving must remove entries from the system data store after verifying successful archiving
    e. Nonfunctional Requirements:
        i. AR06: System failure of this feature must not result in the system going offline.
        ii. AR07: Archival process must complete within 60 seconds upon invocation
        iii. AR08: Failures from archiving shall not cause the system to go offline
8. Usage Analysis Dashboard (AS)
    a. Purpose: A way to visualize user behavior within the system
    b. Scope: Any system administrator account of the system
    c. Preconditions:

i. Persistent data store must be active
ii. Persistent data store must be accessible by the system
iii. User must have authenticated session
iv. User must be on Usage Analysis Dashboard view
v. User must be a system administrator

d. Functional Requirements:
i. AS01: The system shall fetch the data from a relational database
ii. AS02: The system shall keep track of key performance indicator data and display the following in a bar chart
1. Top five most visited view of all time
2. The top 5 average duration per view of all time
3. Top 5 Universities with the most logins
iii. AS03: The system shall keep track of key performance indicator data and display the following in a trend chart
1. The number of logins per day within the span of 3 months
2. The number of registrations per day within the span of 3 month
3. The number of matches per day within the span of 3 months

e. Nonfunctional Requirements:
i. AS04: The system shall display the graphs in a dashboard
ii. AS05: Key performance indicators shall be automatically refreshed in intervals of 60 seconds
iii. AS06: The system must load the view within 15 seconds upon completion of navigation
iv. AS07: Failures from usage analysis dashboard shall not cause the system to go offline

9. Authorization (AU)
a. Purpose: To restrict access to a to protected resources and functionalities to users that are valid
b. Scope: Any user attempting to use system
c. Preconditions
i. User must be authenticated to enforce user specific restrictions
ii. User account must be active

d. Functional Requirements:
i. AU01: The system shall require the user to have an admin role to access user management data
ii. AU02: The system shall requiem the user to log in again before accessing aid eligibility

iii. AU03: The system shall restrict access to privileged resources to any user not authorized to use the system
    1. Users will not be able to view, modify, delete, execute, or interact with data they are not privileged to
iv. AU04: The system shall by default only give access to resources and functionalities that do not require select access
v. AU05: The system shall record the operation and timestamp of each unauthorized access attempt
vi. AU06: The system shall allow any user access modification to be active upon the next successful authentication

e. Nonfunctional Requirements:
    i. AU07: Failures from authorization shall not cause the system to go offline

10. Account Deletion (AD)
    a. Purpose: Mechanism for deleting a user account.
    b. Scope: Any registered user within the system.
    c. Preconditions:
        i. User must have an active authenticated session
        ii. User must be on account deletion view
        iii. User has permission to delete account
    d. Functional Requirements:
        i. AD01: The system shall only allow system administrators to delete other system administrator accounts
        ii. AD02: The system shall permanently delete all the users personal identifiable information from the database
        iii. AD03: The system shall not allow for account deletions to be reversible.
    e. Nonfunctional Requirements:
        i. AD04: System failures from account deletion shall not cause the system to go offline.
11. User Privacy (UP)
    a. Purpose: A mechanism to inform and protect user data from being used without their consent
    b. Scope: All user related data within the system.
    c. Functional Requirements:
        i. UP01: The system shall follow the rules and guidelines as set by EULA per GDPR or California Consumer Privacy Act (CCPA) / California Privacy Rights Act (CRPA)
        ii. UP02: The system shall allow a user to opt out of data collection and the selling of their data

iii. UP03: The system shall provide an explanation to the user about the use of their data
iv. UP04: The system shall allow for the deletion of all the user's data and their account if they wish

12. UI/UX (UI)
    a. Purpose: To provide a intuitive interface for users to interact with the system
    b. Scope: All features that require user interaction within the system
    c. Functional Requirements:
        i. UI01: All text must be presented in U.S. English
        ii. UI02: All formats must align with the selected unit of measurement
    d. Nonfunctional Requirements:
        i. UI03: All views must be able to be understood by a single user in order to understand how to interact with it.
        ii. UI04: All system messages to the user must be displayed in U.S. English for authenticated and unauthenticated users
        iii. UI05: All system message must appear within 5 seconds of the resolution of an operation
        iv. UI06: System failures of the UI must not result in the system going offline.
13. Documentation (DO)
    a. Purpose: To provide artifacts that describe our system in detail
    b. Scope: All features within the system
    c. Nonfunctional Requirements:
        i. DO01: Low level design documents
        ii. DO02: User Manuals
        iii. DO03: FAQs

Expansion Features

1. Advice (AD)
    a. Purpose: Provides recommendations to users who are interested in applying to honor programs and checks availability for priority registration based on provided information.
    b. Scope: CSULB students who are currently taking courses.
    c. Functional Requirements
        i. AD01: Users shall enter course information that they already took or are currently taking with their gpa.
        ii. AD02: Users shall be able to view a list of recommended honor programs based on their information

iii. AD03: Users shall be able to view the requirements for priority registration
    1. Requirements
        a. GPA requirement
        b. Course requirement
iv. AD04: Course requirements shall be able to contain the following data:
    1. Category (“CECS”, etc.)
    2. Course number (“100”, etc.)

d. Nonfunctional Requirements
    i. AD05: The added item shall be shown within 10 seconds in schedule.

2. Internship Resume Builder (IRB)
    a. Purpose: To help students update their resume with relevant academic experience for specific internships.
    b. Scope: Uses internship postings to produce resume suggestions for students.
    c. Functional Requirements
        i. IRB01: The system shall search for keywords in each internship posting and match them to classes that the student has taken.
        ii. IRB02: For each of a student’s classes that is matched to a keyword, the class/keyword pair shall be added to that student’s list of relevant classes.
        iii. IRB03: All classes that a student has taken that are relevant to an internship posting shall be compiled into a list of relevant experience.
        iv. IRB04: The system shall warn the student if an internship posting might be inappropriate for them, if any of the following checks are met:
            1. The internship is offered for higher-year college students, but the student is not a higher-year student (e.g. the internship is for juniors, but the student is a freshman).
            2. The internship’s hours (if available) conflict with the student’s current schedule.
        v. IRB05: Students shall be able to view their list of relevant academic experience, or download it as a .txt file.
3. Student Career Opportunities (CO)
    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:
    i. CO01: Users shall be able to view jobs, internships, and research programs and filter search results using the following information:
        1. Job title
        2. Submission deadline (if applicable)
        3. Application type
        4. Salary
        5. Zip code
    ii. CO02: Users shall be able to input the following information about themselves:
        1. Their student status (freshman, sophomore, junior, senior)
        2. Their number of units completed
        3. Their major
        4. Their certifications (if applicable, the user shall enter the name of the certification)
        5. Their past internships (if applicable, the user shall enter the name of internship)
        6. Their past research programs (if applicable, the user shall enter the name of research program)
    iii. CO03: Users shall be able to sort search results by:
        1. Date posted
        2. Salary (USD)
        3. Distance (miles)
        4. Deadline
    iv. CO04: The system shall redirect users to official website to apply
    v. CO05: users shall be able to hide career opportunities individually
d. NonFunctional:
    i. CO06: Users status shall be considered for career opportunities requirements (freshman, sophomore, junior, or senior)
    ii. CO07: Completed units shall be considered for career opportunities requirements (0-999)
    iii. CO08: By default, search results will be sorted by date posted
    iv. CO09: The search result will contain a maximum of 250 available career opportunities

v. CO10: The search result page will display a maximum of 10 career opportunities per page
vi. CO11: Display message when career opportunities are available/ no longer available

4. GPA/Grade Calculator (GPA)
    a. Purpose: To allow users to quickly and efficiently calculate their GPA throughout the semester and see where they stack up against other students.
    b. Scope: CSULB students who are currently taking courses.
    c. Functional Requirements
        i. GPA01: Users shall enter in their grades for classes that correspond with their schedule builder with numerical values.
            1. The grades will then be Instantly calculated to an average GPA.
        ii. GPA02: Users shall be able to create and save previous semester GPAs.
        iii. GPA03: Users shall be able to view their current rank within the class leaderboard anonymously.
    d. Nonfunctional Requirements
        i. GPA04: Grades may be entered as numerical values for classes.
        ii. GPA05: Grades will be displayed as “A,B,C,D,F” for semester values.
        iii. GPA06: Leaderboards will be class based on and data from students who use the GPA calculator and schedule builder.
            1. The leaderboards will be anonymous so that no privacy laws are infringed upon.
            2. Leaderboards will be viewable only for classes, not semesters.

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 the 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 the 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
2. US Bureau of Labor and Statistics https://www.bls.gov/cpi/