

SW Engineering CSC648-848 Fall 2025

GatorGuides

Team 09

Jason Javandel (jjavandel@sfsu.edu) – Team/Server Lead

Andy Yip – (ayip3@sfsu.edu) – Frontend Lead

Connor Moore – (cmoore14@sfsu.edu) - Database Lead

Shirish Maharjan - (smaharjan@sfsu.edu) - GitHub Lead

Manea Fadl Manea – (mmanea@sfsu.edu) - Backend Lead

Milestone 2 Part 1

Submission Date	Revised Date
11/2/2025	

Table of Contents

SW Engineering CSC648-848 Fall 2025	1
GatorGuides	1
Team 09	1
Milestone 2 Part 1.....	1
Table of Contents.....	2
Executive Summary.....	3
List of main data items and entities	4
Functional Requirements.....	5
Priority 1	5
Visitor (Unregistered).....	5
User (Registered Student)	6
Tutor	6
Admin	7
Priority 2 (Desired).....	7
User (Registered Student)	7
Tutor	9
Priority 3	10

Tutor	10
Admin	10
Justification Notes:.....	10
UI Mockups	11
High level Architecture	15
Identify key risks.....	16
Project management	16

Executive Summary

GatorGuides is a tutoring platform built by and for student of San Francisco State. We help simplify connecting tutors and students by eliminating the need to schedule in person, as well as providing students with a rating system to understand the tutor's teaching styles and overall feedback.

Tutors will have their own page that they can customize with a biography, the subjects they tutor, as well as a background/text color(tentative). Tutors can select days and time ranges in which they are open for appointments. These pages will be visible without any authentication, but scheduling will be restricted to users that are signed in.

Students will be able to filter subjects to be tutored in, which tutor they want, as well as a time frame, and be shown relevant results in a clean way. Students can look at reviews and their experiences with the tutor from previous students and decide who they want to select. If they do not care about which tutor or timeframe they want, they will be presented with a list of the soonest appointments to select from. After a tutoring session has concluded, the user will be given a notification asking them to leave a review on the tutor for themselves.

Our team of students is committed to providing an easy to use and hassle-free tutoring service for the students of SFSU. We aim to provide this ease-of-use without compromising looks, something that seems to be ever more common inside the world of educational software. Being students ourselves, we have unique insights and views on what other students would want out of a service like ours, giving us an advantage over existing solutions.

List of main data items and entities

1. Admin – User that holds control over the registered users in the system; allows for the additions, deletion, and alterations of the website
2. Visitor – Person that has not made an account or not logged in; allows for browsing of tutors, but not allowed to schedule an appointment
3. User – Person that has made an account and is logged into; allows for browsing of tutors, scheduling, and calendar functions
4. Tutor – User approved for mentorship; allows for creation of the tutor profile, appointments, and hourly fees
5. Tutor authentication – Verification for tutors; Confirmation of SFSU attendance—email and ID—and peer review of teaching ability
6. Session – Allotted time determined by tutor for study
7. Glossary – List of all tutors, subjects, and topics available
8. Calendar – Showcases all available tutoring sessions on a weekly or monthly format
9. Tags – List of subjects or topics taught by the tutor
10. Browsing – Allows for searching of glossary
11. Scheduling – Registered user's reserving a tutoring session
12. Boolean search – Allows for AND, OR, and NOT in filtered searches
13. Fees – Priced determined by tutors for teaching services
14. Coursing tracking – Provides the user the ability input courses under their profile, allowing for better filtering when searching for tutors

Functional Requirements

Priority 1

Visitor (Unregistered)

1. Should enable visitor to get list of tutors available.

1.1 System should have a list of all active tutors that a visitor can see without any form of verification.

1.2 Glossary have to indicate tutor name, subjects and limited availability.

2. Visitor should be capable of searching glossary in case of a certain subject or tutor.

2.1 Glossary page should feature search bar.

2.2 Search should allow narrowing the search to a specific subject area (e.g., Mathematics, Biology, Computer Science).

2.3 Search will have to enable the tutor name (partial and full name match) filtering.

2.4 Results of a search should be dynamically updated without the need to refresh the page.

3. Should have the ability to create an account.

3.1 Registration form should demand SFU email address (@sfsu.edu or mail.sfu.edu).

3.2 Account should demand: first name, last name, SFSU email and secure password.

3.3. Account Creation System should authenticate SFSU email domain.

User (Registered Student)

4. The user can only be permitted to create a single account.

4.1 System should verify other accounts that have the same SFSU email.

4.2 Repeated registrations should show the error message: Account already exists with this email.

4.3 System should have an option of forgotten password to existing accounts.

6. A user needs to be logged in to book a tutoring session.

6.1 Scheduling button should take one to the login page in case of non-authentication.

6.2 User should be redirected to scheduling page after being logged in.

6.3 Session booking has to demand active and validated account.

11. To become Admin (SFSU-Specific Requirement), users have to verify their status by manual verification to be SFSU.

11.1 User is required to present legitimate SFSU documentation(student ID or staff ID).

11.2 Admin verification request should have the purpose of accessing the administration.

11.3 Certified SFSU status should be placed on user profile.

Tutor

12. The tutor accounts should be in a position to make lists of their services.

12.1 Tutor should develop at least one service listing in order to make it visible in glossary.

12.2 Listing should contain: subjects on offer, and timing.

12.3 Listing should not achieve publicity without approval of the administration.

12.4 Tutor should be in a position to make various listings regarding various subjects.

15. Tutor should be in a position to view the sessions that are scheduled with them.

15.1 The tutor dashboard should show all the subsequent meetings in chronological order.

15.2 Every session should indicate: name of the student, subject, time of session and place.

15.3 System shall provide email notification per new booking.

15.4 Tutor should be in a position to see session details by clicking on session.

Admin

18. Admin should be in a position to cancel user and tutor accounts.

18.1 Admin It should have access to user management dashboard.

18.2 Account termination should have the option of reason selection by a pre-defined list.

18.3 System should provide termination notification email to user concerned.

18.4 Terminated accounts do not have to be deleted out of database, they must be listed as inactive.

Priority 2 (Desired)

User (Registered Student)

5. When out of the system, user may not access calendar.

5.1 Calendar page should verify status of authentication and then load.

5.2 Unauthenticated access attempts shall be redirected to the page of login and message.

5.3 System shall not change intended destination URL on post-login redirect.

7. A single user should have the ability of creating several sessions on behalf of other users.

7.1 Booking form should have the possibility to choose to book on behalf of another SFU student.

7.2 System should check whether SFSU email of other students are present in the database.

7.3 Email confirmation should be advised to both the booking user and recipient of the session.

7.4 User should be able to access and maintain all the sessions that they have booked.

8. Any sessions that are scheduled before the date should be correctly reflected in calendar.

8.1 Calendar should show all the confirmed sessions in real-time.

8.2 Past sessions should be automatically described as completed.

8.3 Calendar should indicate session status: Upcoming, In Progress, Completed, Cancelled.

8.4 System should maintain a history of sessions not less than one year of academic sessions.

9. User should be in a position to cancel session and reflect the same to tutor.

9.1 User should be in a position to cancel sessions at least 24 hours beforehand.

9.2 Cancellation This has to cause instant email notification to tutor.

9.3 Cancelled time slot should be made available to the other users on demand.

9.4 System should be able to monitor historical cancellation and cancellation reasons.

10. User should be able to create an account of a tutor.

10.1 User should have an opportunity to request tutors status on their profile.

10.2 Tutor application should entail: subjects to be taught, qualifications, availability.

10.3 User needs to submit evidence of competency (transcript, certification, etc.)

10.4 The activation of tutor accounts should not be allowed without the approval of the admins.

Tutor

13. Tutor should have the option of editing his page whenever he wants.

13.1 Tutor should go to edit mode on his or her profile dashboard.

13.2 Profile fields cannot be changed other than verification status.

13.3 amendments should be stored and updated on open profile.

13.4 System should have a record of audit of all changes to the profile.

14. Tutor should have the ability to insert tags and bio on their page.

14.1 Tutor will be presented with a predefined list of subject tags, which he/she will need to choose at least 3.

14.2 Tutor should be capable of summing up to 10 subject/skill tags.

14.3 Tags should be searchable on the glossary search feature.

16. Tutor should also be capable of canceling the session and showing it to the user.

16.1 Tutor is required to give a 12-hour minimum notice of cancellation.

16.2 Cancellation should be conditioned upon a choice of reason (emergency, illness, etc.).

16.3 User should be provided with instant email and in-app notification.

16.4 System shall provide the option to the user to reschedule with the same or the different tutor.

Admin

20. Admin should have the potential of communicating with users and tutors.

20.1 Admin should have incorporated message system to all platform users.

20.2 Admin should be in a position to send individual and broadcast messages.

20.3 message history should be kept not less than 90 days.

20.4 Admin should be capable of seeing discussions with any user.

Priority 3

Tutor

17. The tutors should not be permitted to send messages to the users prior to a session having been scheduled.

17.1: Default feature of messaging between tutor and user must be disabled.

17.2: Messaging to enable 48 hours prior to scheduled session.

17.3 System should not allow any effort to thwart messaging restrictions.

17.4 The users should be capable of reporting any inappropriate messaging attempts.

Admin

19. Admin should have the capability of deleting posts.

19.1 Admin should have the capability of deleting improper listings of tutor.

19.2 Admin has to have the ability to remove the comments or reviews that are harsh.

19.3 Deletion shall need selection of reasons and be recorded in the system.

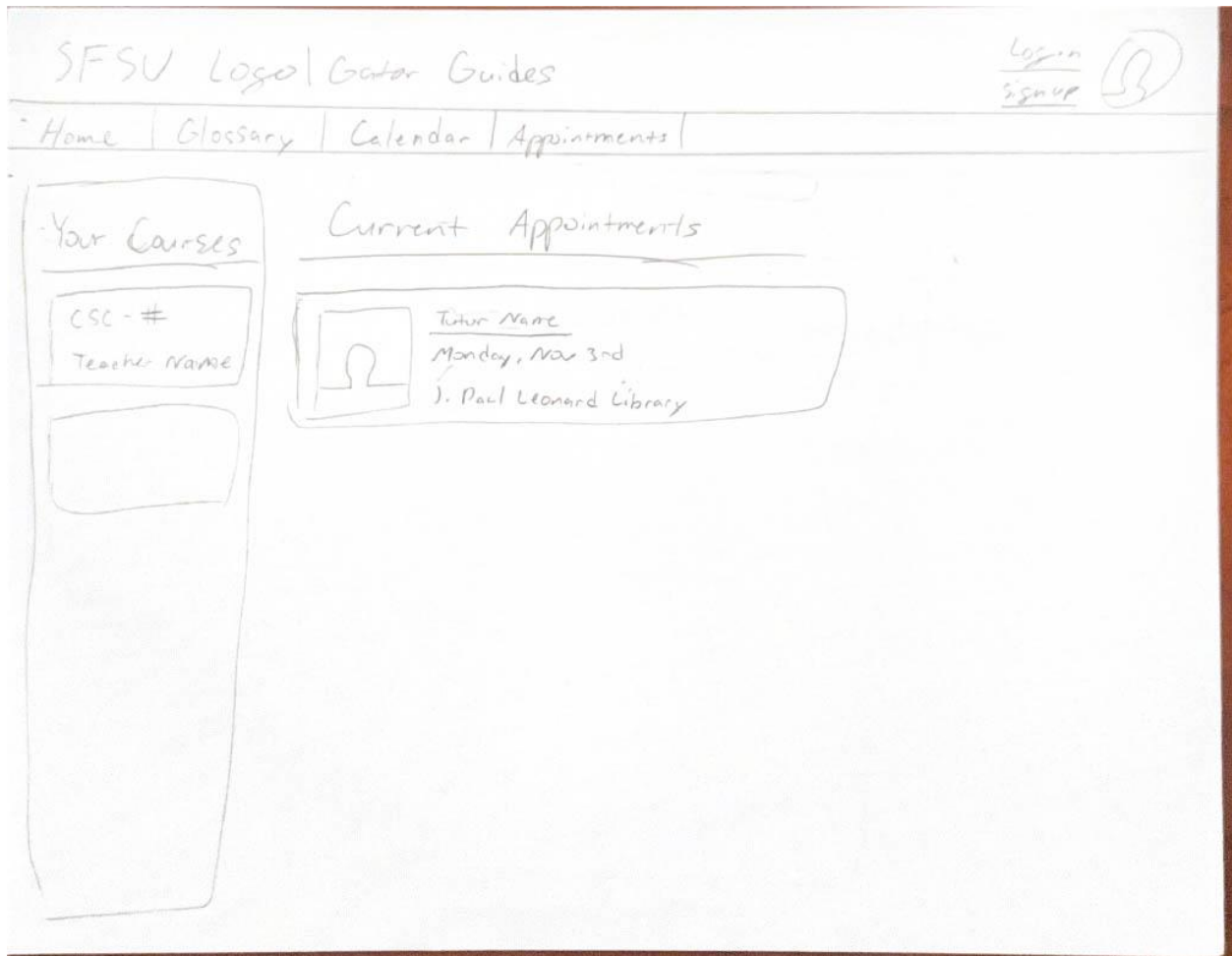
19.4 Original poster shall have reasoned content removal.

Justification Notes:

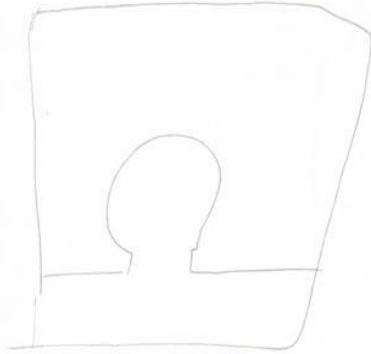
1. Priority 1 is concerned with the basic functionality required to operate the platform.
2. Priority 1 to verify platform integrity is SFSU verification (Requirement 11).
3. Important features that can be used to increase user experience are found in Priority 2.
4. Priority 3 has nice-to-have features that can be introduced in case of time.
5. Each requirement is traceable to initial high-level requirements.

UI Mockups

Home page



Individual tutor page



Tutor Name

Available Appointments

Monday, Nov 3rd
J. Paul Leonard Library Schedule

Wednesday, Nov 5th
J. Paul Leonard Library Schedule

Contacts

student@sfsu.edu
(123) 456-7890

Pop up

Log In

X

Email

Password

Submit

A hand-drawn sketch of a 'Tutor Sign Up' form. The form is enclosed in a rounded rectangle. At the top, the title 'Tutor Sign Up' is written and underlined. Below the title are four input fields: 'Full Name', 'Email', 'Student ID', and 'Code'. Each field is represented by a horizontal rectangle. The 'Code' field is smaller and has a dashed line inside. At the bottom of the form is a 'Submit' button, represented by a rounded rectangle with the word 'Submit' inside.

High level Architecture

(No normalization or attention was put into optimization atm)

Users:

Full name

Email

Hashed-password

Type

Rating

Messages

User_id

Content

Datetime

Posts

User_id

Content

Tags

Datetime

Profile

Status

Tags

Bio

-Media will be stored in the filesystem

-Ratings will be implemented and automated.

Database Organization

Identify key risks

Project management

For divvying up the work, I decided to overhaul the Trello board we use to make it easier to assign and reassign different tasks. We still use slack as our main form of communication, although sometimes an email will get through quicker/better. The only problem is that most of the assignments have due dates in other classes, but this class does not, so it becomes much harder to remember the assignment if you forget about it. I want to look further into Trello to see if it can send reminders to the members of the board about due dates and check-ins, as I have to do it manually right now.

Use of GenAI

GenAI was not used in this milestone

Team Lead Checklist

- So far all team members are fully engaged and attending team sessions when required
 - OK.
- 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
 - OK
- Team reviewed suggested resources before drafting Milestone 2
 - OK
- Team lead checked Milestone 2 document for quality, completeness, formatting and compliance with instructions before the submission
 - DONE
- Team lead ensured that all team members read the final Milestone 2 document and agree/understand it before submission
 - DONE
- Team shared and discussed experience with GenAI tools among themselves
 - OK