SW Engineering CSC648-848 Fall 2025 - GatorGuides

Milestone	Submission Date	Revised Date
Milestone 1	10/17	10/

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.

Upon signing up, the user will be asked for their name, student id, and password. All newly created accounts will be set to a student by default, at which point; users will be asked if they want to become a tutor account or not.

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.

Personae

1. Non-tech inclined student (Jake)
   a. Probably uses a MacBook or Windows with Safari/Chrome
   b. Likes using one password for everything
   c. Spends time reading through webpage to learn more
   d. Wants to have an easy sign-up process to get started
2. Tech inclined student (Diego)
   a. Probably uses Mac or Linux with any type of browser
   b. Speeds through website instead of reading anything
   c. Has long/complex/multi character type passwords
   d. Wants to have an easy sign-up process to get started
3. Tutor (Julia)
   a. Online scheduling should be easy and send reminders
   b. Should be able to see all appointments
   c. Doesn't have a lot of time to fiddle with setup/organization
   d. Wants to be able to customize their profile to make it their own
   e. Wants to have an easy sign-up process to get started

High-level Use cases

Case 1: Student Browsing and Searching for Teachers/Tutors:

Alia wants to learn more about the professors that lecture in her department, which is computer science, so she logs onto the web program. Using the website, she can browse or search by name or subject to view a list of teachers. She primarily wants to know who teaches which classes and what other students think of their teaching. Alia wants to find out as soon as possible which instructors are available and deserving of her consideration for her next classes.

Case 2: Submitting A Rate and Review

Alia finished her software development class in Fall 2024 and wants to share her experience with the class and the professor's teaching style. To do this, she logs into the website and searches for the professor's page/profile so she can rate and review the professor based on her experience. Alia decided to contribute to the community by sharing her insights and helping other students gain a better understanding of what to expect just as she was able to find professors who matched her considerations based on previous ratings and reviews from students.

Case3: Teacher/Tutor Views Ratings and Reviews

Professor Jose logs onto the website to review how students have evaluated their teaching. He reads through the ratings and comments, paying attention to both the positive feedback and the areas where students suggest improvement. This reflection helps him identify strengths in his teaching style as well as opportunities for growth. The feedback is not only valuable for Professor Jose personally, but also beneficial for teachers and tutors in general, as it provides insight into how they can continually improve the learning experience for their students.

List of main data items and entities

1. Admin – Individual that holds control over the registered users in the system; allows for the additions, deletion, and alterations of the website
2. Unregistered User – User that has not made an account or not logged in; allows for browsing of tutors, but not allowed to schedule an appointment
3. Registered User – User that has made an account and is logged into; allows for browsing of tutors, scheduling, and calendar functions
4. Tutor – Registered user approved for mentorship; allows for creation of the tutor profile, appointments, and hourly fees
5. Tutor authentication – Verification for tutors
6. Calendar – Showcases all available tutoring sessions

List high level functional requirements

1. Must allow the user to access glossary of available tutors

2. Must allow users to create accounts

3. Accounts must be logged in to schedule a tutoring session

4. Calendar must accurately reflect any sessions scheduled until past the date

5. User must be able to search glossary for specific subjects or tutors

6. User must have the ability to create a tutor account

7. Tutor accounts must be able to make listings for their services

8. Tutor must be able to see sessions scheduled with them

9. User must be able to cancel session and have it reflected to tutor

10. Tutor must be able to cancel session and have it reflected to user

11. User must be forced to login to schedule a session

12. User cannot access calendar while signed out

13. Tutor must be able to populate their page with tags and a bio

14. Tutor must be able to edit their page at any time (add or remove info)

15. User must only be allowed to make one account

16. One account must be able to make multiple sessions for different users

17. Tutors must not be able to message users before a session is scheduled

18. Admin must be able to terminate user and tutor accounts

19. Admin must be able to delete posts

20. Admin must be able to communicate with both users and tutors

21. Users must verify their SFSU status through manual review to become Admin

List of non-functional requirements

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers
3. All or selected application functions shall be rendered well on mobile devices (no native app to be developed)
4. Posting of tutor information and messaging to tutors shall be limited only to SFSU students
5. Critical data shall be stored in the database on the team's deployment server.
6. No more than 50 concurrent users shall be accessing the application at any time
7. Privacy of users shall be protected
8. The language used shall be English (no localization needed)
9. Application shall be very easy to use and intuitive
10. Application shall follow established architecture patterns
11. Application code and its repository shall be easy to inspect and maintain

12. Google analytics shall be used
13. No e-mail clients shall be allowed. Interested users (clients) can only message service providers via in-site messaging. One round of messaging (from client to service provider) is enough for this application. No chat functions shall be developed or integrated
14. Pay functionality (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
15. Site security: basic best practices shall be applied (as covered in the class) for main data items
16. Media formats shall be standard as used in the market today
17. Modern SE processes and tools shall be used as specified in the class, including collaborative and continuous SW development and GenAI tools
18. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Fall 2025. For Demonstration Only" at the top of the WWW page Nav bar. (Important so as to not confuse this with a real application).

Competitive analysis

Features	Tutor.com	wyzant.com	SFSU TASC	GatorGuides
Tutor authentication	+	+	+	++
Browsing	+	+	-	++
Scheduling	+	+	+	++
Boolean Search	-	-	-	+
Course Tracking	-	-	-	+

Keys: Exists: + Exceeds: ++ Void: -

Summary:

One of the advantages that we hope to bring in our application is the tutor authentication, ensuring that aspiring tutors are peer-reviewed and respectful of their mentee. Another notable feature is the ability to have a personalized schedule for courses that you are currently enrolled in to match the needs of registered users, streamlining the process to find a tutor. Browsing, scheduling, and searching are made to be easy to navigate, but also specific enough for registered users to find what they want.

High-level system architecture and technologies

SW components and versions

Database

- MySQL 9.4.0

Deployment cloud servicer planned to use

- AWS 1 CPU 1 GB RAM

Front end frameworks planned to use

Frontend

- SvelteKit 2.22.0 (routing)
- Svelte 5.0.0
- Tailwind CSS 4.0.0

Backend

- Python 3.13.7
- FastAPI 0.119.0

Browsers planned to support
Google chrome and Mozilla Firefox (latest 2 versions)

Major additional external open-source APIs planned to use

Use of GenAI Tools

GenAI was used to come up with a name for our application, as well as giving an outline for defining the personaes.

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

Team Checklist

- So far all team members are fully engaged and attending team sessions when required
    - DONE/OK
- Team found a time slot to meet outside of the class
    - DONE/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
    - DONE/OK
- Team reviewed class slides on requirements and use cases before drafting Milestone 1
    - DONE/OK
- Team reviewed non-functional requirements from “How to start…” document and developed Milestone 1 consistently
    - DONE/OK
- Team lead checked Milestone 1 document for quality, completeness, formatting and compliance with instructions before the submission
    - DONE/OK
- Team lead ensured that all team members read the final M1 and agree/understand it before submission
    - DONE/OK
- Team shared and discussed experience with GenAI tools among themselves
    - DONE/OK
- Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)
    - DONE/OK