SW Engineering CSC648-848 Fall 2025

EduGator

Team 5

Team lead: Grady Walworth - wwalworth@sfsu.edu
GitHub Master: Michael
Frontend lead: Tejas
Frontend developer: Kameron
Backend lead: Chris
Backend developer: Hardy

Milestone 1

History Table:

Version 1.0	Due: 10/16/2025
Version 2.0	Submitted: 10/28/2025

1. Executive Summary

Our project, EduGator, is a peer-to-peer tutoring platform designed exclusively for San Francisco State University students. It provides a trusted, intuitive space where members of the SFSU community can connect for academic support, share knowledge, and build ongoing learning relationships. The goal is to make finding or offering tutoring as simple and secure as connecting with a classmate.

Unlike generic tutoring apps or social media groups, EduGator verifies every user through their SFSU credentials, ensuring a safe and authentic academic community. The novelty of our approach lies in its focus on campus exclusivity, mutual trust, and recurring peer connections rather than one-time transactions. All tutoring is community-driven and non-commercial, emphasizing collaboration, mentorship, and academic growth over profit.

When students register, the platform automatically tailors their experience by pre-filtering content based on the subjects they are studying. Tutoring opportunities are organized by general subject areas (not specific course numbers) and distinguished by Lower Division and Upper Division levels to keep the system streamlined yet meaningful.A color-coded calendar makes scheduling simple. Students can view tutor availability at a glance, filter by subject, and choose between Drop-in sessions (open group tutoring) or Appointments (1-on-1 meetings). The integrated search bar allows users to find sessions by subject, tutor name, or day

By funding this project, the university will empower students to support one another academically, reduce learning barriers, and foster a stronger, more connected campus culture. EduGator not only enhances student success but also reinforces the values of accessibility, inclusivity, and community service that define SFSU.

Our team is made up of passionate San Francisco State University students who want to help our peers succeed. EduGator is built by students, for students. Our goal is to create something that truly reflects the spirit of SFSU. We see EduGator as more than an app, it’s a step toward building a stronger culture of mentorship and mutual growth across our university.

2. Personae

Persona 1: SFSU Student User - Alex (SFSU Student)

General Characteristics:

The undergraduate student is majoring in Computer Science at San Francisco State University. He frequently attends campus events, workshops, and tech meetups to build his professional skills and meet other students.

Goals:

The guest user hopes to use the platform to discover public events, browse photos and descriptions, and get a clear sense of the university's culture. By accessing this information easily, they can better evaluate whether SFSU aligns with their academic goals and lifestyle preferences.

Pain Points:

Within SFSU's Student Center, students often find it difficult to locate the specific functions or services they need. Important information is scattered across multiple pages and systems. For example, course registration, event announcements, and student services are managed separately, which makes the navigation process confusing and time-consuming. Students hope for a unified and well-organized platform where all campus-related information is centralized and easy to access. This would allow them to complete tasks more smoothly and efficiently.

Persona 2: SFSU Staff - Dr. Maria Lopez

General Characteristics:

A faculty or staff member at San Francisco State University, such as a professor or event coordinator. She frequently organizes campus events, workshops, and extracurricular activities to increase student engagement and foster a stronger sense of campus community.

Goals:

The staff member aims to simplify and centralize event management. They want to post new activities efficiently, edit event details when plans change, and track participation through an easy-to-use dashboard. They also wish to communicate updates and collect feedback from students within a single platform to improve event organization.

Pain Points:

Many instructors at SFSU struggle because there is no single, consistent place to post important information such as class announcements, assignments, or upcoming exams. Each course or event often uses a different system, which forces both faculty and students to switch between multiple platforms like Canvas, email, and department websites. This fragmented setup makes it difficult for teachers to ensure that all students receive timely updates, and it often results in confusion or missed information.

Persona 3:Guest User - Jordan Lim

General Characteristics:

This persona represents a high-school senior interested in applying to San Francisco State University. The guest user wants to explore what campus life is like and understand what kinds of student activities are available before making a college decision.

Goals:

The guest user hopes to use the platform to discover public events, browse photos and descriptions, and get a clear sense of the university's culture. By accessing this information easily, they can better evaluate whether SFSU aligns with their academic goals and lifestyle preferences.

Pain Points:

Many official university websites are not intuitive, and some event details require login credentials that prospective students do not have. It is difficult for them to find accurate, up-to-date information about campus life. They seek a simple, welcoming, and easy-to-navigate interface that allows non-students to explore the community freely.

3. **High-level Use cases**

Use Case 1: Searching for Tutors by Name

Actors: Alex (SFSU Student)
Description:
An **SFSU Student** logs into the tutoring platform and searches for a specific **Tutor** by name using the search bar. The system displays matching **Tutor** profiles and their availability from the **Calendar**. The student can view each **Profile** for more details such as ratings and **Review / Feedback**, then proceed to filtering or booking.
Frequency/Importance: Weekly; important for quickly locating preferred tutors.
Environment/Context: Accessed via desktop or mobile browser through the website.

Use Case 2: Filtering Tutors

Actors: Alex (SFSU Student)
Description:
After performing a search, the **Student** refines results using filters such as **Subject**, availability, and **Review / Feedback** rating. The system dynamically updates the list to show **Tutor**s matching the selected filters and displays their earliest available sessions from the **Calendar**. This enables students to efficiently find suitable academic support without browsing irrelevant results.
Frequency/Importance: Weekly; enhances the efficiency of tutor discovery and booking.
Environment/Context: Used from dorms, libraries, or off-campus via web browser or mobile device.

Use Case 3: Guest Browsing

Actors: Jordan Lim (Guest User)

Description:

A **Guest User** visits the tutoring website to explore available **Tutor**s and **Subject**s. The system allows the guest to search for **Tutor**s by name and apply filters such as **Subject**, availability, or rating to refine results. Guests can view public content, including **Tutor** summaries, general **Post**s, and open time slots on the **Calendar**. When a guest attempts to view a full **Profile** or book a **Session**, the system prompts: “Only registered SFSU members can continue. Please log in with your SFSU email.” Upon logging in, the user is returned to their original page to continue seamlessly. This encourages exploration while maintaining SFSU exclusivity.

Frequency/Importance: Daily; supports user onboarding and smooth transition to full access.

Environment/Context: Publicly accessible on desktop or mobile browsers.

Use Case 4: Booking a Tutoring Session

Actors: Alex (SFSU Student), Jordan Lim (Guest User), Tutor

Description:

An **SFSU Student** selects a preferred **Tutor** from the search results or **Profile** and clicks “Book Session.” The system displays the **Tutor**’s available slots from the **Calendar**, allowing the student to choose a convenient time. A confirmation panel shows **Session** details including **Subject**, session type (from **Post**), and duration. Once confirmed, the **Session** appears on both the **Tutor** and **Student Calendar**s. If a **Guest User** attempts to book, a prompt requests SFSU login before proceeding.

Frequency/Importance: Weekly; essential for connecting students to personalized academic support.

Environment/Context: Used via website on any browser.

Use Case 5: Staff Management of Tutoring Sessions

Actors: Dr. Maria Lopez (SFSU Staff), Tutor

Description:

A **Tutor** logs into the tutoring system to manage tutoring operations. They can create and edit **Post**s for upcoming sessions, adjust **Calendar** availability, view **Session** attendance, and review **Feedback** submitted by students. This centralized management ensures consistent quality and effective coordination of tutoring services across the platform.

Frequency/Importance: Weekly; critical for maintaining accurate schedules and supporting academic success.

Environment/Context: Accessed from office desktops or remotely through the website.

4. **List of main data items and entities – data glossary/description**

- **Admin**:
    - can access all data and content and modify the database. Needs to login/register
    - Allows for managing users, subjects/courses, sessions, disputes, payments, site content, and analytics.
    - Can impersonate users for support. Requires login/registration with elevated privileges.
- **Tutor**
    - Has their own profile, can set availability, offer session types/prices, accept/decline bookings, can deliver sessions, and record outcomes/notes.
    - Requires login/registration and verification.
- **Student**
    - Has their own profile, can search tutors, view profiles, book sessions or packages, attend sessions, message tutors, and manage cancellations.
    - Requires login/registration for booking.
- **Unregistered User**
    - Does not have an account
    - Can browse public content (tutor summaries, subjects, available), initiates signup.
    - Can see available times
    - No booking/messaging until registered.
- **Subject**
    - Represents a general academic area (e.g., “Mathematics,” “Computer Science”).
    - Used to categorize tutors and sessions for easier searching and filtering.
    - Ensures consistent organization across all tutoring Posts.
- **Post**
    - Created by tutors to advertise availability for tutoring sessions.
    - Includes details such as subject, date/time, description, and session type (Drop-in or Appointment).
    - Acts as the primary way tutors make themselves discoverable to students.

- **Session**
    - A confirmed tutoring meeting between a tutor and a student.
    - Contains information such as participants, subject, location, time, and status.
    - Can originate from a Post or direct booking request.
    - Generates confirmation and reminder notifications for both parties.
- **Calendar**
    - The central scheduling feature shows all available and booked sessions.
    - Allows users to filter by subject, tutor, or color-coded session type.
    - Tutors and students have personalized views to manage their schedules.
- **Review / Feedback**
    - Submitted by students after a tutoring session.
    - Contains a rating, written comments, and a timestamp.
    - Supports quality assurance and helps other students choose tutors.
- **Profile**
    - Displays personal and academic details of a user.
    - Tutor profiles include expertise, bio, and session history.
    - Student profiles may show booked sessions or preferred subjects.
    - Accessible only to authenticated users.
- **Analytics Record**
    - Aggregated data used by admins to analyze system usage.
    - Tracks metrics such as active users, popular subjects, and session frequency.
    - Does not contain personally identifiable information.

5. **List high level functional requirements**

Unregistered Users:

1. Unregistered users shall be able to browse tutors and view public tutor profiles.
2. Unregistered users shall be able to register as a student or tutor.
3. Unregistered users shall be able to register using their SFSU email.
4. System shall redirect unregistered users to the login/register page when attempting to access certain features.
5. Unregistered users shall be able to view available tutoring subjects and courses.

Registered Users (Students and Tutors):

6. Users shall be able to sign in using their SFSU email.
7. Users shall be able to create and edit their personal profile.
8. Users shall be able to edit their personal profile.
9. Tutors shall be able to post their available tutoring times.
10. Tutors shall be able to receive notifications for requests, and confirmations.
11. Tutors shall be able to state the courses and subjects they offer.
12. Students shall be able to view a tutor's expertise.
13. Students shall be able to filter tutors by course, subject, keyword, or availability.
14. Students shall be able to cancel their tutoring requests.
15. Students shall be able to reschedule their tutoring requests.
16. Students shall be able to rate and provide feedback about the tutors.
17. Tutors shall be able to update previously posted tutoring sessions.
18. Tutors shall be able to remove previously posted tutoring sessions.

System/Admin:

19. System shall create a record when a tutor creates a tutoring session; this session shall appear on the student’s calendar.
20. Admins shall be able to approve or reject tutor applications.
21. Admins shall be able to delete tutoring sessions.
22. Admins shall be able to remove user accounts that violate policies.
23. System shall notify tutors when a tutoring request is made.

6. 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 the 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).

7. **Competitive analysis (functions/features only, not business or marketing analysis)**

Our advantages:

Comparison Table:

Feature	Tutoring SFSU	Superprof.com	Tutor.com	Bayareatutor .org	EduGator
Text Search	+	++	+	+	+
Browse	+	+	+	+	++
Calendar for scheduling	+	-	-	-	++
User verification via email	++	+	+	+	+

\+ Feature exists ; ++ superior feature; - does not exist

Summary:

Our planned tutoring platform builds on the strengths of existing services like Superprof, Tutor.com, and BayAreaTutor.org while focusing on unique SFSU-specific advantages. Unlike general tutoring sites, our system is designed exclusively for SFSU students and tutors, providing verified access through university email authentication to ensure trust and safety. It also offers a superior integrated scheduling calendar for easy coordination of sessions, a powerful browsing and search system to quickly match students with suitable tutors, and the convenience of meeting on campus in familiar spaces like the library or CS lab. These tailored features make our platform both more secure and more relevant to the SFSU community, giving it a competitive edge over broader, less personalized tutoring solutions.

8. **High-level system architecture and technologies used**

    - Server Host: Amazon AWS EC2, Instance: t3.micro
    - Operating System: Ubuntu 24.04 LTS
    - Database: MySQL 8.0.43
    - Web Server: NGINX 1.24.0
    - Frontend Tools: HTML5, CSS3, JavaScript ECMAScript 2025
    - Server-Side Language: JavaScript with Node.js v18.19.1
    - Additional Technologies:
        - Web Framework: Express v5.1.0
        - IDE: Visual Studio 2022
        - Web Analytics: Google Analytics GA4
        - Browsers:
            - Google Chrome v141.0 - v144.0
            - FireFox v144.0 - v147

9. **Use of GenAI tools like ChatGPT and copilot for Milestone 1**

- Executive Summary: ChatGPT was helpful in combining our goals, rough outline, and list of features into a professional introduction for the platform that is easy to read and represents our ideas accurately and concisely.
- High-level Use cases: ChatGPT was used to organize, and make it more professional and structured. The content, ideas were originally developed by author.
- High-level Functional Requirements: ChatGPT helped word the requirements in a more professional way. At first they were somewhat informal, but it allowed us to make them sound more professional.
- Competitive Analysis: ChatGPT was used to update the summary paragraph for the Comparison Table for a more professional and thoughtful summarization of our competitive analysis.
- High Level System Architecture - Useful in conjunction with internet searches to verify latest LTS versions of software that we are using.

10. Team and roles

Team 5:

Student Name	School Email	Role
Michael Thompson	mthompson17@sfsu.edu	Github Master
Tejas Rajan	trajan@sfsu.edu	Frontend Lead
Grady Walworth	wwalworth@sfsu.edu	Team Lead
Christopher Chan	cchan39@sfsu.edu	Backend Lead
Pei Huan Chang	pchang@sfsu.edu	Backend Developer
Kameron Jacob	kjacob2@sfsu.edu	Frontend Developer

11. Team Lead Checklist to be completed by team lead

• So far all team members are fully engaged and attending team sessions when required
DONE

• Team found a time slot to meet outside of the class
DONE

• Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing
DONE

• Team reviewed class slides on requirements and use cases before drafting Milestone 1
DONE

• Team reviewed non-functional requirements from “How to start...” document and developed Milestone 1 consistently
DONE

• Team lead checked Milestone 1 document for quality, completeness, formatting and compliance with instructions before the submission
DONE

• Team lead ensured that all team members read the final M1 and agree/understand it before submission
DONE

• Team shared and discussed experience with GenAI tools among themselves
DONE

• Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)
DONE