

SW Engineering CSC648-848 Fall 2025

## EduGator



## Team 5

Team lead: Grady Walworth - [wwalworth@sfsu.edu](mailto:wwalworth@sfsu.edu)

Github Master: Michael - [mthompson17@sfsu.edu](mailto:mthompson17@sfsu.edu)

Frontend lead: Tejas - [trajan@sfsu.edu](mailto:trajan@sfsu.edu)

Frontend developer: Kameron - [kjacob2@sfsu.edu](mailto:kjacob2@sfsu.edu)

Backend lead: Chris - [cchan39@sfsu.edu](mailto:cchan39@sfsu.edu)

Backend developer: Hardy - [pchang@sfsu.edu](mailto:pchang@sfsu.edu)

## Milestone 1

History Table:

Version 1.0	Due: 10/16/2025
-------------	-----------------

## **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.

## **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:**

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 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: Scheduling and Managing Tutoring Sessions**

**Actors:** Alex (SFSU Student), Dr. Maria Lopez (SFSU Staff)

**Description:**

A Computer Science student struggling with algorithms searches the platform for tutoring sessions in that subject. The system shows available peer tutors and staff-led workshops. The student can book a session that fits their schedule. The staff member overseeing tutoring can update session availability, view attendance, and gather feedback after sessions. This centralized feature eliminates confusion from separate tutoring sign-up sheets or emails, ensuring students can easily access academic support.

**Frequency/ Importance:** Weekly; critical for improving academic performance and retention.

**Environment/Context:** Students and staff use desktop or mobile access within the Student Center or remotely.

#### **Use Case 2: Searching for Tutors by Name and Subject**

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

**Description:**

An SFSU student looking for help in “Data Structures” uses the platform’s search and filter tools. They can type a tutor’s name or select filters such as subject, availability, or tutor rating. The system instantly displays relevant tutors with their bios, subjects, and available time slots. A guest user visiting the site can also search tutors to explore SFSU’s academic support network but cannot access contact or booking features until they register or log in.

**Frequency/ Importance:** Daily or weekly; important for efficient academic help discovery.

**Environment/ Context:** Used primarily on desktop or mobile web browsers in dorms, libraries, or off-campus.

### **Use Case 3: Booking a Tutoring Appointment( Login or Registration Prompt)**

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

**Description:**

After finding a suitable tutor, the student clicks “Book Session.” The system displays the tutor’s available time slots and allows the student to choose one. When a guest user attempts to book, they are prompted with a message: “Only registered SFSU members can book tutoring sessions. Please log in with your SFSU email.” This maintains exclusivity while allowing guests to view tutors and schedules. Once the student books, both tutor and student receive confirmation and reminders.

**Frequency/ Importance:** Weekly; essential for connecting students to academic help.

**Environment/ Context:** Students may book from dorm, labs, or mobile devices between classes.

### **Use Case 4: SFSU Email-Verified Registration (Exclusive Function)**

**Actors:** Alex (SFSU Student), Dr. Maria Lopez (SFSU Staff), Jordan Lim (Guest User)

**Description:**

A unique feature exclusive to this platform is **SFSU Email-Verified Access**. Only individuals with an [@sfsu.edu](mailto:@sfsu.edu) email can register as full users ( student or staff). This ensures that tutoring, academic support, and event tools remain secure and campus-specific. When a new user signs up, the system automatically checks for valid SFSU domain before granting full access. Guest users without SFSU emails are allowed to browse with public content but not interact with internal tutoring or event systems.

**Frequency/ Importance:** Occurs during first-time registration; vital for SFSU exclusivity and trust.

**Environment/ Context:** Occurs through web-based registration, accessible on all devices.

## **Use Case 5: SFSU Smart Match - Automated Tutor Recommendations (Exclusive Function)**

**Actors:** Alex (SFSU Student)

**Description:**

A standout, SFSU-exclusive feature called **Smart Match** uses course registration data to automatically suggest relevant tutors. For example, a student enrolled in “CSC 340: Programming Methodology” receives personalized recommendations for peer tutors who have previously taken and excelled in the same course. The system can also suggest upcoming workshops or study groups related to the student’s enrolled courses. This intelligent recommendation system saves time and strengthens academic connections within the SFSU community.

**Frequency/ Importance:** Regularly suggested throughout the semester; high importance for academic success.

**Environment/Context:** Integrated with the student’s dashboard and course data; accessible from any device.

#### **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. System shall redirect unregistered users to the login/register page when attempting to access certain features.
4. Unregistered users shall be able to view available tutoring subjects and courses.

### **Registered Users (Students and Tutors):**

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

### **System/Admin:**

15. System shall create a record when a tutor creates a tutoring session; this session shall appear on the student's calendar.
16. Admins shall be able to approve or reject tutor applications.
17. Admins shall be able to delete tutoring sessions.
18. Admins shall be able to remove user accounts that violate policies.
19. 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	Our Future Product
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
- Server-Side Language: JavaScript with Node.js v18.19.1
- Additional Technologies:
  - Web Framework: Express v5.1.0
  - IDE: Visual Studio 2022
  - SSL Cert: Let's Encrypt (Cert Bot 4.1.1)
  - Web Analytics: Google Analytics GA4
  - Browsers:
    - Google Chrome v141.0
    - FireFox v144.0

## **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:**

<b>Student Name</b>	<b>School Email</b>	<b>Role</b>
Michael Thompson	<a href="mailto:mthompson17@sfsu.edu">mthompson17@sfsu.edu</a>	Github Master
Tejas Rajan	<a href="mailto:trajan@sfsu.edu">trajan@sfsu.edu</a>	Frontend Lead
Grady Walworth	<a href="mailto:wwalworth@sfsu.edu">wwalworth@sfsu.edu</a>	Team Lead
Christopher Chan	<a href="mailto:cchan39@sfsu.edu">cchan39@sfsu.edu</a>	Backend Lead
Pei Huan Chang	<a href="mailto:pchang@sfsu.edu">pchang@sfsu.edu</a>	Backend Developer
Kameron Jacob	<a href="mailto:kjacob2@sfsu.edu">kjacob2@sfsu.edu</a>	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