

# **SW Engineering CSC 648-848 Fall 2024**

## **GatorTutor**

### **Team 6**

Team Lead, Alex Hoff - ([ahoff2@mail.sfsu.edu](mailto:ahoff2@mail.sfsu.edu))

Github Master, Dylan Faulder - (dfaulder@mail.sfsu.edu)

Backend Lead, Austin Ng - (Ang@mail.sfsu.edu)

FrontEnd Lead, Dalan Choy - (dchoy3@mail.sfsu.edu)

Programming Lead, Jack Richards - ([jrichards7@mail.sfsu.edu](mailto:jrichards7@mail.sfsu.edu))

### **Milestone 1**

#### **History**

<b>Date Submitted:</b>	<b>October 12th, 2024</b>
<b>Date Revised:</b>	<b>October 15th, 2024</b>

## **1. Executive Summary:**

GatorTutor is a tutoring service created by San Francisco State University students for SFSU students. We aim to provide a place that facilitates the matching between all SFSU students with their best fit SFSU tutor. We feel that the dividing factor between a successful student and a struggling student is gaps in their knowledge and our goal is to bridge that gap. Our service helps match tutors with experience in those subjects and the students seeking assistance. Since our application is institution specific, our tutors are able to provide a level of service unique to SFSU that foreign services can not compete with.

Since our team is composed of SFSU students, we are able to target institution specific problem areas that's stopping students from achieving their potential through tutoring. Our application allows students to search for tutors by university specific classes. Professors don't always teach the same curriculum as others, so tutors are able to specify which ones they have experience with, allowing students to select them based on their issues. Tutors are able to create profiles unique between individuals through multimedia to customize their impressions, introduce themselves through text descriptions, show interested students their rates along with their availability, and appear on searches by course identifiers. Our platform also facilitates communication between interested students and tutors with our built-in messaging service.

## **2. CSC 648 Personae:**

### **Amy - Student**

#### **Amy is a student at SFSU**



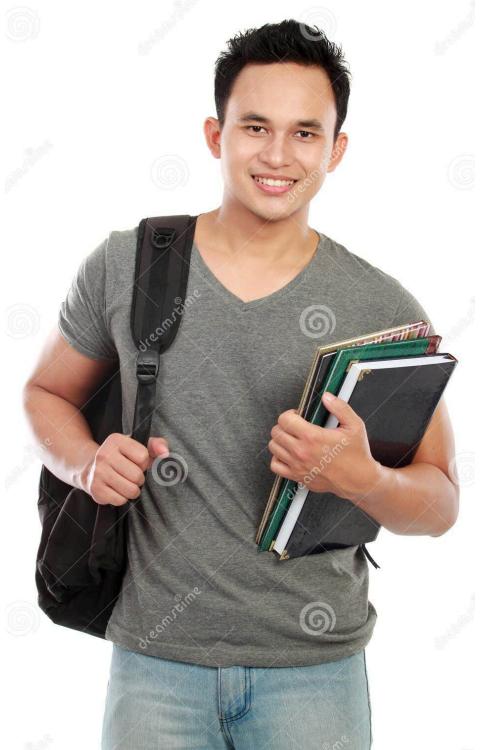
- Has basic website navigation skills
- Has strong math Knowledge
- Needs \$\$
- Wants to become a tutor

#### **Goal And scenario -**

Amy wants to be able to easily sign up for GatorTutor and place all her information so that any students who need math help can easily find and contact her.

- **Name:** Amy
- **Demographics:** [20, Female, Student]
- **Goals:** [Wants to raise money for tuition by tutoring on gator tutor]
- **Skills:** [Strong Math skills, Basic WWW experience]
- **Pain Points:** [Lacks skills to function on using the mobile version]
- **Motivation:** [Raise money for tuition]

### **John - Struggling Student**



- Is Studying nursing
- Struggles in Biology class
- Does not have a lot of time
- Wants to schedule a tutor easily

#### **Goal And scenario -**

John Wants to schedule an appointment with a tutor for his BIO class. He wants to look to see if any of his classes have tutors. If he is able to find one he wants to be able to contact and set up an appointment.

- **Name:** John
- **Demographics:** [18, Male, Student]
- **Goals:** [Wants to pass BIO class]
- **Skills:** [Knows how to use mobile]
- **Pain Points:** [Cannot find his class listed on the site]
- **Motivation:** [Wants to pass BIO and needs a tutor to do so]



### **Herald - Adult Continuing student looking for help**

- Came back to SFSU to study writing
- Has very bad writing skills and wants to get a tutor to ask how they write better essay for their class

- Is a Adult student who does not use technology at all
- Needs to be able to simply sign up and look for a tutor

### **Goal And scenario -**

An older student who has bad technology skills wants to find a tutor in as little steps as possible. Wants to first browse the tutors to see if there is anyone he would like.

- **Name:** Herald
- **Demographics:** [65, Male, Student]
- **Goals:** [Wants to improve his writing skills and needs a tutor to do so]
- **Skills:** [Very little experience in using websites]
- **Pain Points:** [Is hard to navigate the website]
- **Motivation:** [Wants to improve their writing skills for their class and wants a tutor who is good at essays to help them outline a better paper]

### **Sarah- Admin**

- A site admin

- Has to approve submissions and posts on Gator Tutor
- Is skilled and knowledge about the site
- Monitors the site to make sure there are no problems

### **Goal And scenario -**

Sarah is a site admin who has control over the site and is in charge of taking care of monitoring the website and approving new tutors and messages.



- **Name:** Sarah
- **Demographics:** [26, Female, Admin]
- **Goals:** [Able to maintain the site and allow/take down posts]
- **Skills:** [Strong skills in understanding the site]
- **Pain Points:** [Has trouble overlooking the site]
- **Motivation:** [Wants to make sure the website is well maintained and there are no bad faith tutors or posts getting made without proper overview so that users can have a good safe area to find tutors]

### **3. SFSU Tutoring Web Application High Level Use Cases:**

#### **Use Case 1: Amy Explores Tutoring Options**

Case Number: 1

Actors: Amy

Narrative: Amy, a student at SFSU, visits the tutoring app to explore help for her calculus class. Without requiring her to log in or create an account, the system allows her to browse available tutors by subject and filter by availability. After reviewing a few profiles, Amy decides to contact a tutor who seems like a good fit. The system then prompts her to complete a registration (lazy registration) before she can send a message or arrange a meeting time through the built-in messaging system. Once registered, she proceeds to arrange the session and later rates the tutor based on her experience.

Postconditions: Amy successfully connects with a tutor, attends the session, and leaves a rating.

Exceptions: If no tutors match her search criteria, the system suggests alternative options or prompts Amy to broaden her search

---

#### **Use Case 2: John Finds a Specific Tutor**

Case Number: 2

Actors: John

Narrative: John, another student at SFSU, hears about a specific tutor from a friend and wants to find them on the platform. John logs in and enters the tutor's name in the search bar. The system returns a list of tutors with matching or similar names. John clicks on the tutor's profile to view more information and contacts them via the messaging feature to arrange a session. They finalize the session time through the chat.

Postconditions: John successfully finds the tutor he was looking for and schedules a session.

Exceptions: If the tutor is unavailable, the system suggests similar tutors or conveys to John that the specific tutor he is looking for is not available at any time.

---

#### **Use Case 3: Amy Becomes a Tutor**

Case Number: 3

Actors: Amy

Narrative: Amy, who has previously used the platform to find tutoring help, now wants to become a tutor herself. Amy fills out the required information, including subjects she can tutor, her rates, availability, and a brief bio. She reviews her post and submits it for approval. After submitting, Amy is notified that her tutor post is pending approval. Once approved, her post goes live, and she receives notifications when students contact her.

Postconditions: Amy successfully submits her tutor post, and it goes live once approved.

Exceptions: If her post is rejected, Amy receives a notification. She can then try to make a new post.

---

#### **Use Case 4: Amy Receives a Message from a Student**

Case Number: 4

Actors: Amy, John

Narrative: After Amy's tutor post goes live, John, a student, sends her a message asking about her availability for a tutoring session. Amy receives a notification in the app, indicating that she has a new message. She clicks the notification and is redirected to the chat interface, where she can view John's message. The conversation appears as a thread, similar to an email exchange. Amy responds, and they arrange a time for the session.

Postconditions: Amy successfully communicates with John and schedules a tutoring session.

Exceptions: If either party is offline, messages remain in the inbox until the user returns.

---

#### **Use Case 5: John Reports a Tutor**

Case Number: 5

Actors: John

Narrative: John, after a negative experience with a tutor, decides to report the issue. He logs into the app, navigates to his message history, and selects the tutor's profile. John clicks the 'Report' button (may change) and is redirected to a form where he can describe the issue. He fills out the details, explaining the problem, and submits the report. The system informs John that the report has been submitted and will be reviewed within 24 hours.

Postconditions: John submits a report, and the system confirms it will be reviewed by an admin.

Exceptions: If the report does not meet the platform's criteria for action, John is informed and provided with relevant information about the reporting process.

---

### **Admin Use Case: Sarah Approves Tutor Posts**

Case Number: 6

Actors: Sarah

Narrative: Sarah, the admin responsible for approving tutor posts, logs into the admin interface. She is notified of new pending tutor posts. Sarah reviews each post to ensure it meets the platform's guidelines, including valid subject matter, appropriate pricing, and a clear bio. If a post is acceptable, Sarah approves it, and the tutor is notified. If it doesn't meet the standards, Sarah rejects it. Once approved, the post is live on the platform, and tutors can start receiving messages from students.

Postconditions: Tutor posts are reviewed, approved, or rejected. Approved posts go live, and tutors are notified.

Exceptions: If Sarah rejects a post, the tutor receives feedback? (possibly just deleted with no context) and is asked to revise and resubmit.

---

### **Admin Use Case: Sarah Reviews Pending Reports**

Case Number: 7

Actors: Sarah

Narrative: Sarah, the admin, receives a notification about a new report submitted by a user. She logs into the workbench system and checks the list of pending reports. Sarah reviews the details of the report and cross-references it with the tutor post in question. If the tutor's post violates the platform's guidelines, Sarah deletes the post and marks the report as resolved. If the report is not valid, Sarah dismisses it.

Postconditions: Sarah resolves the report, either by removing the offending tutor post or dismissing the report if it is invalid.

Exceptions: If additional information is required, Sarah contacts the reporting user for clarification before taking further action (TODO, may change).

---

## **4. Main Data and Entities:**

Guest User

**Entity Name:** Guest User

**Description:** A general term for anyone who interacts with the system.

**Usage:** All users must register with SFSU credentials to use the platform. They can create accounts, search for tutors, and communicate with others. Some users may create tutor posts.

### **Registered User**

**Entity Name:** Registered User

**Description:** A user who has successfully registered on the platform.

**Usage:** Registered users can create posts, search for tutoring services, send messages, and interact with other users. They can also associate themselves with tutor posts.

### **Tutor Posts**

**Entity Name:** Tutor Posts

**Description:** Posts created by registered users who offer tutoring services.

**Usage:** These posts contain details such as subjects offered, availability, and hourly rate. Registered users can create these posts and students can search for and contact tutors through them. Tutor posts are visible to other users, and interactions are initiated through them.

### **Admin**

**Entity Name:** Admin

**Description:** An individual designated to manage the platform and handle database operations.

**Usage:** Admins do not interact with the platform as regular users do. They only use tools such as an MQL workbench to manage the system's data. They handle tasks such as profile approvals, database queries, and maintenance. Admins do not have standard user profiles.

### **Message**

**Entity Name:** Message

**Description:** A communication between registered users, typically between students and tutors.

**Usage:** Messages are exchanged through the platform for inquiries about tutoring services and general communication.

### **Subject**

**Entity Name:** Subject

**Description:** A specific academic subject at SFSU.

**Usage:** Subjects are associated with tutor posts. Students search for tutors based on these subjects.

## **5. High Level Functional Requirements:**

- 1. Tutor - Creating Tutor Posts:** Tutors shall be able to create a post to advertise their services on the website.
- 2. Tutor - Dashboard for Tutors:** Tutors shall have a functioning dashboard that displays their postings and messages received.
- 3. Student - Browsing:** Users shall be able to browse and easily navigate the website and all it has to offer.
- 4. Student - Searching:** A static search bar at the top of the page that shall be used by students to identify tutors that fit their criteria.
- 5. Student - Dashboard for Students:** Students shall have a functioning dashboard that displays their messages and any relevant postings.
- 6. Student - SFSU Specific Tutoring:** Students shall be able to search for SFSU-specific classes and professors to find hyper-relevant material.
- 7. Admin - Approval of Tutor Posts:** Site Admin shall be able to approve or deny any tutor post through the MySQL Workbench.
- 8. Admin - Deleting Inappropriate Content:** Site Admin shall be able to delete inappropriate items or users.

## **6. Nonfunctional 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 render 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 email clients shall be allowed. Interested users can only message to sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application
14. Pay functionality, if any (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 2024. 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:**

GatorTutor aims to provide tutoring services tailored towards the needs of students enrolled at San Francisco State University. Unlike its competitors, this platform connects students to tutors who are familiar with the university's curriculum and can help those who are struggling with a particular course. However, this objective does not come at the expense of the simple and intuitive interface that users are accustomed to working with on other platforms. GatorTutor makes connecting with a tutor an easy process through a search filter. Additionally, the platform

also provides support for both online and in-person tutoring which ensures anyone will find a tutor that meets their needs. Tutors are also given full control over their profiles. They can set their own hourly rates, availability, and subjects. These features will collectively make GatorTutor the primary destination for all SF State students looking to further their learning and academic growth.

	GatorTutor	Varsity Tutor	Preply	Wyzant
SFSU Specialized Tutoring	vv			
Course-specific tutoring	vv			
Filtered Searching	vv		vv	vv
User-Friendly Interface	vv	v	v	vv
Online and In-Person Tutoring	vv	v		vv
Review	vv	vv	vv	v

## 8. High-Level System Architecture and Technologies Used:

**Main Software Components:** Node.js and MySQL

**Deployment Cloud Service:** AWS

**Front-End Frameworks:** ExpressJS

**Supported Browsers:** Chrome and Firefox

**External APIs:** Google Analytics

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

**Our team used GenAI in order to:**

- Create a structure for the Executive Summary
- Give us ideas for Personas about what a student might be looking for in a website
- Help us detail the non-functional requirements and what their purposes might be
- Aide us in the competitive analysis, thinking of what features a competitive tutoring website may have

**Was it useful?**

GenAI was quite useful in spitballing ideas to us that we, as a group, could then further explore and refine. Although we only used it for a few items in M1, we found it to be very helpful where we needed it and used it more as a tool of inspiration than anything.

## **10. Team and Roles:**

Role	Github Master	Front-end Lead	Back-End Lead	Programming Lead	Team Leader
Name	Dylan	Dalan	Jack	Austin	Alex

## **11. Team Lead 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 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**