

## **CSC 648 848 Fall 2024 Milestone 1 - Team 6**

### **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:** []
- **Motivation:** [Raise money for tuition]

## John - Struggling Student



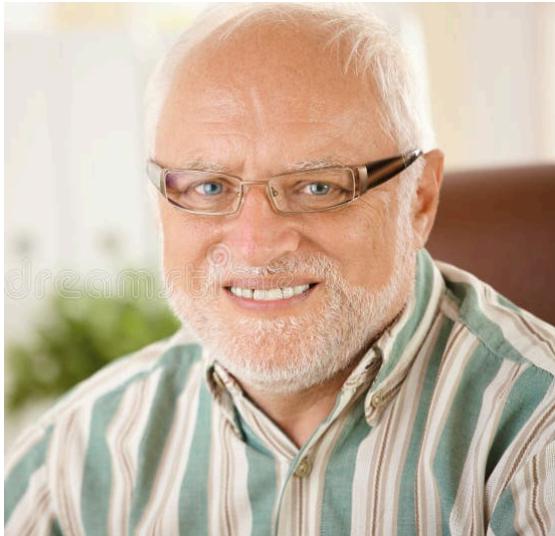
- Is Studying nursing
- Struggles in Biology class
- Does not have a lot of time
- Wants to schedule a tutor easily

### 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:** [None]
- **Pain Points:** [Cannot find his class listed on the site]
- **Motivation:** [Wants to pass BIO and needs a tutor to do so]

## Herald - Old 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 old 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]



## Dave - Student Comes tutor

- A student who used gator tutor to learn more about Math
  - Learned a lot and passed their class
  - Now wants to become a tutor
  - Wants to switch from looking for a tutor to becoming one

### Goal And scenario -

Dave got tutoring in math from Gator Tutor and now wants to bring help to other people just like him. He now wants to switch from looking for tutors to creating his own profile so that students can now contact him and set up appointments for his own tutoring.

- **Name:** Dave
- **Demographics:** [21, Male, Student]
- **Goals:** [Wants to tutor]
- **Skills:** [Math Skills and expert in using WWW]
- **Pain Points:** [Is unable to figure out how to switch from being a student user to a tutor user]
- **Motivation:** [Wants to help students like the way he was helped so needs a application that is specific for SFSU]

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

Main use cases:

Use Case Title: Student Explores Tutoring Options

Case Number 1.0

Actors: Student, Tutor

Preconditions: Student is logged into the app with a verified SFSU email. They don't know what they are looking for exactly. They know the subject but not the tutor. They are here to explore their options.

Trigger: Student wants to find a tutor for a specific subject.

Main Flow:

1. Student navigates to the "Browse Tutors" page.
2. Student enters search criteria such as subject and availability.
3. System displays a list of tutors matching the criteria.
4. Student views detailed profiles of selected tutors.
5. Student selects a tutor and contacts them.
6. Student and tutor agree on a meeting time and location.
7. Student and tutor meet for tutoring session.
8. Student rates tutor.

Postconditions: Student has found a tutor and has completed a tutoring session.

Exceptions/Alternatives: If no tutors match the criteria, the system suggests similar options.

Narrative style: As a student, I want to find a tutor for a specific subject so that I can get help with my studies. I go to the site and click on the "Browse Tutors" page. I enter my search criteria such as subject and availability. The system displays a list of tutors matching the criteria. I view detailed profiles of selected tutors. I select a tutor and contact them. My tutor and I agree on a meeting time and location. My tutor and I met for a tutoring session. I rate my tutor.

Use Case Title: Student Hears About A Specific Tutor From Friend

Case Number 2.0

Actors: Student, Tutor

Preconditions: Student is logged into the app with a verified SFSU email. They know what they are looking for. They are here to find a specific tutor.

Trigger: Student hears about a specific tutor from a friend.

Main Flow:

1. Student clicks in the search bar and types the name of the tutor they want to find.
2. The site shows a list of tutors with that name, or similar names.
3. Student clicks on the tutor's name.
4. The site shows the tutor's profile.
5. Student contacts the tutor with a message and a way to contact them.
6. Student and tutor agree on a meeting time and location.
7. Student and tutor meet for tutoring session.
8. Student rates tutor.

Postconditions: Student has found a tutor and has completed a tutoring session.

Exceptions/Alternatives: If the tutor is not available, the system suggests similar tutors.

Narrative style: As a student, I want to find a specific tutor that I have heard about from a friend. I go to the site and click on the search bar. I type the name of the tutor I want to find. The site shows a list of tutors with that name, or similar names. I click on the tutor's name. The site shows the tutor's profile. I contact the tutor with a message and a way to contact them. My tutor and I agree on a meeting time and location. My tutor and I meet for a tutoring session. I rate my tutor.

Use Case Title: A new tutor signs up for the site

## Case Number 3.0

Actors: Tutor

Preconditions: Tutor is not logged into the app. They are here to sign up for the site.

Trigger: Tutor wants to sign up for the site.

Main Flow:

1. Tutor navigates to the "Sign Up" page.
2. Tutor enters their name, SFSU email, and password.
3. Tutor clicks on the "Sign Up" button.
4. Tutor is redirected to the "Become a Tutor" page.
5. Tutor fills out the form with their subject, price, bio, weekly availability, photo, and optional phone number, video.
6. Tutor clicks on the "Submit" button.
7. Tutor is redirected to their profile page, where they can see their new tutor post.
8. Tutor is let known that they will be notified when their tutor post is approved.
9. Tutor receives an email notifying them that their tutor post is live.
8. They are let known that they will be notified when a student sends them a message.

Postconditions: Tutor has signed up for the site and their tutor post is live.

Exceptions/Alternatives: A tutor might not be able to submit their tutor post because they are not a student at SFSU.

Narrative style: As a tutor, I want to sign up for the site so that I can offer my tutoring services to students. I go to the site and click on the "Sign Up" page. I enter my name, SFSU email password and whether I am a tutor or a student. The site redirects me to the "Become a Tutor" page. I fill out the form with my subject, price, bio, weekly availability, photo, and optional phone number, video. I click on the "Submit" button. I am redirected to my profile page, where I can see my new tutor post. I am let known that I will be notified when my tutor post is approved. I receive a message notifying me that my tutor post is live. I am let known that I will be notified when a student sends me a message.

Use Case Title: User wants to become a tutor

## Case Number 4.0

Actors: User (Becomes a Tutor)

Preconditions: User is logged into the app with a verified SFSU email. They are not a tutor. They wish to become a tutor.

Trigger: User wants to become a tutor.

Main Flow:

1. Because all accounts are users even if they have a tutor post, all the user must do is create a tutor post.
2. The user clicks the "Become a Tutor" button on the nav bar.
3. The user fills out the form with their subject, price, bio, weekly availability, photo, and optional phone number, video.
4. The user clicks the "Submit" button.
7. Tutor is redirected to their profile page, where they can see their new tutor post.
8. Tutor is let known that they will be notified when their tutor post is approved.
9. Tutor receives an email notifying them that their tutor post is live.
8. They are let known that they will be notified when a student sends them a message.

Postconditions: User has signed up for the site and their tutor post is live.

Narrative style: As a user, and have used this site to find tutors but now I want to become a tutor. I go to the site and click on the "Become a Tutor" button on the nav bar. I fill out the form with my subject, price, bio, weekly availability, photo, and optional phone number, video. I click the "Submit" button. I am redirected to my profile page, where I can see my new tutor post. I am let known that I will be notified when my tutor post is approved. I receive a message notifying me that my tutor post is live. I am let known that I will be notified when a student sends me a message.

Use Case Title: User/Tutor receives a message from another user/tutor

Case Number 5.0

Actors: User, Tutor

Preconditions: User/Tutor is logged into the app with a verified SFSU email. One of them is a tutor with a tutor post. Another user/tutor sends them a message. After seeing the others Tutor's posts, they are interested in talking to them, and have sent them a message.

Trigger: User/Tutor receives a message from another user/tutor.

### Main Flow:

1. The user/tutor sees a message notification pop up on the nav bar.
2. The user/tutor clicks on the notification.
3. The user/tutor is redirected to a chat page.
4. The user/tutor can now chat with the other user/tutor.
5. The user/tutor can see the history of the chat. Less like SMS and more like email.
6. The site's job is done at this point. The user/tutor can now discuss the details of the tutoring session.

Postconditions: User/Tutor has sent and received a message from another user/tutor.

Exceptions/Alternatives: If the other user/tutor is not online, the message is added to their inbox.

### Narrative style:

User: I have created my account and have found a tutor and sent them an initial message. Now I want to discuss the details of the tutoring session. I go to the site and click on the chat page. I see the history of the messages and I can send and receive messages.

Tutor: I have created my tutor post and have received a message from a user, I see the notification pop up on the nav bar. I click on the notification and am redirected to the chat page. I see the history of the messages and I can send and receive messages.

### Special Use Cases:

Use Case Title: User wants to report a tutor

Case Number 5.0

Actors: User

Preconditions: User is logged into the app with a verified SFSU email. They had an issue with a tutor.

Trigger: User wants to report a tutor.

### Main Flow:

1. User navigates to their profile page. Where can see a history of tutors they have sent messages to.
2. The user goes down to the history section and clicks on the tutor they want to report.
3. The user is redirected to the tutor's profile page.

4. The user clicks on the "Report" button.
5. The user is redirected to the report page.
6. The user fills out the report form with the tutor's name, email, and a description of the issue.
7. The user clicks on the "Submit" button.
8. The user is redirected to the home page.
9. The user is let known that the report has been submitted.

Postconditions: User has reported the tutor.

Notes: Anyone can report a tutor. For any reason. For example if a profile is fake, or has inappropriate content that somehow passed the verification process.

Narrative style: As a user, I want to report a tutor I had a bad experience with so that they are removed from the site. I use the message history to navigate to their profile page. I click on the "Report" button on their profile page. I am redirected to the report page. I fill out the report with the reason I had a bad experience. I click on the "Submit" button. I am redirected to the home page. I am let known that the report has been submitted.

Use Case Title: An admin wants to approve a tutor post

Case Number 6.0

Actors: Admin

Preconditions: Admin doesn't have a special account. There is no admin page. An admin exists. An admin has access to the database.

Trigger: Admin wants to approve a tutor post.

Main Flow:

1. An admin is notified that there is a new tutor post.
2. The admin goes to the database.
3. The admin knows what tutor posts need to be approved by seeing the ones that are pending.
4. The admin approves the tutor post by changing the status to "approved".
5. The system lets the tutor know that their tutor post is approved.
6. The tutor can now see their tutor post on the site.

Postconditions: The tutor post is approved. The tutor is notified that they are now a tutor.

Exceptions/Alternatives: If the tutor post is denied, the tutor post is deleted. The tutor is notified that their tutor post is denied. No reason is given.

Narrative style:

I received a notification that there is a new tutor post. I go to the database and see the ones that are pending. I check the tutor post to see if it meets the requirements. I make sure to view the uploaded image, and video. I approve the tutor post by changing the status to "approved".

Use Case Title: An admin wants review pending reports

Case Number 7.0

Actors: Admin

Preconditions: A report has been made that needs to be reviewed by an admin. There exists guidelines for what is considered valid for the admin to follow.

Trigger: Admin wants to review pending reports.

Main Flow:

1. An admin is notified that there is a new report.
2. The admin goes to the database.
3. The admin knows what reports need to be reviewed by seeing the ones that are pending.
4. The admin reviews the report and cross references it with the tutor post.
5. The admin reviews the tutor post in question.
6. The admin deletes the tutor post if it meets the requirements and the report is valid.
7. The admin sets the report status to "resolved".

Postconditions: The report is reviewed and the tutor post is deleted if it meets the requirements and the report is valid. The report status is set to "resolved".

Narrative style: I receive a notification that there is a new report. I go to the database and see the ones that are pending. I review the report and cross reference it with the tutor post. I review the tutor post in question. I delete the tutor post if it meets the requirements and the report is valid. I set the report status to "resolved".

## 4. Main Data and Entities:

### Main Data And Entities:

- **Entity Name:** [User]
- **Meaning:** [Anyone who accesses the site]
- **Usage:** [Has access to the site]
- **User Types:** [This is the main user]

- **Tutor:**

- **Entity Name:** [Tutor]
- **Meaning:** [Able to create tutoring posts on the website ]
- **Usage:** [This role will allow the user to create a tutoring post that other users can look at, then find the information of the tutor]
- **User Types:** [Users who are looking for a tutor use what this role posts]

- **Admin:**

- **Entity Name:** [Admin]
- **Meaning:** [Has full access to the site and data workbench]
- **Usage:** [Approves messages and can delete posts]

**User Types:** [Who interacts with it]

### Unapproved Tutor:

- **Entity Name:** [Unapproved Tutor]
- **Meaning:** [Tutors that have made posts that need to be vetted by a admin]
- **Usage:** [This role is assigned during the waiting process of creating a post and then until it is approved by a admin role user]

**User Types:** [Admins interact with this role]

- **Banned User:**

- **Entity Name:** [Banned User]
- **Meaning:** [Users who have been banned from the site]

- **Usage:** [Once banned it prevents this type of user from doing or accessing anything within the site]

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

- **Browsing** - Users shall be able to browse and easily navigate the website and all it has to offer
- **Searching** - A static search bar at the top of the page that shall be used by students to identify tutors that fit their criteria
- **Creating Tutor Posts** - Tutors shall be able to create a post in order to advertise their services on the website
- **Dashboard for the tutors and students to show postings and messages received** -  
Students and Tutors shall have a functioning dashboard with messages received as well as postings
- **Functions for site administrator who will be required to approve each uploaded tutoring item info before it can go live, as well as to delete inappropriate items or users** - Site Admin shall be able to approve or deny of any tutor post through the mySQL Workbench
- **SFSU Specific Tutoring** - Students shall be able to search for SFSU specific classes and Professors in order to find hyper-relevant material

## **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 grow 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:**

<b>Role</b>	Github Master	Front-end Lead	Back-End Lead	Programming Lead	Team Leader
<b>Name</b>	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**