SW Engineering CSC648-848 Fall 2023

SF State Tutors.Tech

Team #3

Danial Tahir - Team Lead
dtahir@sfsu.edu

Cleveland Plonsey - Backend Lead

Ava Albert - Frontend Lead

Michael Mathews - Github Master

Bryan Maldonado - Team Member

Griffin Evans - Team Member

Milestone 1

Date Submitted	**Date Revised**
10/1/2023	

1. Executive Summary

The modern college education is more complex than ever with students expected to take on ever-greater challenges in order to earn their degree. Students struggle to make the key connections they need to make in order to succeed. But what if there were a service that helped bridge the gap between students to enable them to help each other? A resource that enables students with the knowledge and experience to help to offer their services to students in need of that help. And what if that service were geared explicitly towards the students of one university, with services unique to their campus? Enter SF State Tutors.Tech.

SF State Tutors.Tech intends to be THE place where tutors and clients connect on the SF State campus. We intend to have an easy onboarding process for tutors whereby they can register and sign up as tutors in one click, or register as students and then onboard as tutors later. We intend searching for tutors to be easy for anyone, with an intuitive display of each tutor's credentials, rates and other information. We plan to enable messaging between tutors and students for registered students. We plan to have SFSU-specific features, such as only allowing students and tutors to register if they have an SFSU email address, and including maps of campus to enable students to meet up with tutors more easily. We also plan to have reviews of tutors by fellow SF State students so students will be able to trust the reputation of their tutor. With these great features we believe we can capture the market for tutors on the SFSU campus.

The SF State Tutors.Tech team is a diverse and dynamic team of software engineers who are eager to deliver this application. We bring a wide range of skills and experience to the table and have the expertise necessary to deliver this project.

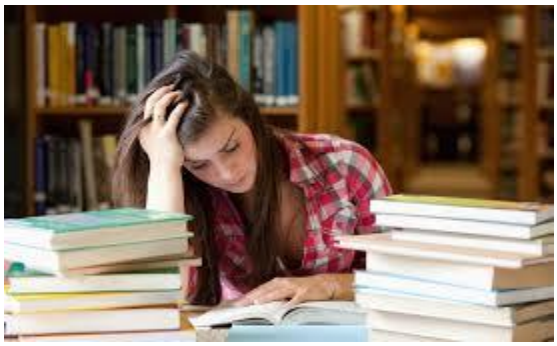
2. Personae and Main Use Cases

Personas: Student, Tutor, Admin
Tutor:

Mark is a 23 year old registered SFSU Tutor. He really enjoys teaching others, and he excels in Math and Chemistry and would like to tutor students who are struggling with those classes. As

he is also looking to receive tutoring with his current classes as he is about to start midterms and is feeling the stress. Mark prefers not to tutor outside of his excels and would like to let students know he only is available for those subjects. Mark prefers to meet in person and he is conscious about the quality of his tutoring.

Use Case:
Mark uses our tutoring platform to offer his expertise in Math and Chemistry to SFSU students seeking academic assistance. Upon registration, he creates a tutor profile, highlighting his qualifications and preferred meeting location (in person). He also sets up his availability calendar to sync with the booking system, ensuring students can easily schedule tutoring sessions with him. Students looking for help in Math or Chemistry can search for Mark and book tutoring sessions based on his availability. The platform notifies both Mark and the students of upcoming sessions, and Mark receives positive reviews and ratings for his tutoring services, attracting more students to seek his assistance.



Alejandra is a 22-year-old student enrolled at SFSU, currently facing challenges in her physics class. She values academic support and is seeking a tutor who can assist her in understanding complex physics concepts. Alejandra has a preference for a female tutor of a similar age due to her comfort level, wanting someone she can easily relate to. Furthermore, she intends to utilize the resources available in the library, such as whiteboards and textbooks, to enhance her learning experience. She additionally works a 9-5 job and needs a way to filter out tutors with less available hours.

Use Case:
Alejandra, the SFSU student, utilizes our tutoring platform to find a suitable physics tutor. She filters her search based on her preferences, specifically seeking a female tutor of a similar age who can accommodate her work schedule. The platform presents Alejandra with a list of potential tutors who meet her criteria and have availability outside of her working hours. After reviewing tutor profiles, Alejandra selects a tutor who aligns with her requirements and offers flexible scheduling options. They agree to meet in the library, where they can access the important whiteboards and reference materials. Our platform ensures Alejandra can receive the academic support she needs without compromising her prior work commitments.

Antonio is a 26 year old Administrator for our tutoring website, he recently graduated from SF State with a Masters degree in Computer Science. With his internship at a software company he learned a lot about backend and enjoys using the mysql workbench to ensure everything is working as planned. He would be maintaining the backend and approving all of the tutors images and videos in order to make sure nothing goes wrong.

Use Case:
Antonio signs on to MySQL workbench and sees that a tutor has submitted a new application. He reviews the media submitted with the application and finds it contains grossly obscene content. He does not approve the video and deletes the user from the database along with their content. He sees another tutor has submitted an application. He finds that they have submitted appropriate content and approves them. He feels proud that he's done a good job for the site.

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

User types:

0. Unregistered Users:
   User who does not have an account or a user that has not logged into their account yet.
   Permissions: can make search requests for tutors NOT students. Can initiate the process of contacting a tutor for help but cannot launch a request until registered as student or tutor.

1. Students:
   Definition:
   A registered user who is not an admin and is not a tutor. These clients are our source of revenue.

   Permissions: can make search requests for tutors NOT students. Can send messages to tutors. Can update profile (change name change password change urlImg change status to isTutor in the profile view.)

2. Tutors:
   Definition: A user that has an account that has become a tutor through the method listed above in student. A tutor inherits all the permissions of a student. Therefore they can also find tutors and pay for tutors and therefore, we also may derive revenue from this type of user.

   Permissions: can make search requests for anyone (this includes tutors as well as students). Can send messages to students only after student has contacted tutor. (can contact a tutor [this is tutor account acting as a student]) Can update profile (change name change password change urlImg change status back to student. Can update their list of SFSU courses that they are qualified to teach. Can)

3. Admin:
   Definition: Super user who inherits all permissions from the above but also can approve images and videos thus activating pending accounts.

   Permissions: Admin can access mysql workbench as a means of screening images and videos uploaded by users. Admin Shall Not modify accounts beyond the ispending mysql column. This is enforced through contract.

Other types:

4. Subject:
   Definition: a database table that exists as an enumeration of all options for a tutor to declare themselves qualified in: (MATH, ART, ENGLISH, CS, PHYSICS, HISTORY, SPANISH, ENGINEERING, ANTHROPOLOGY, BIOLOGY, CHEMISTRY, GEOLOGY, THEATER, and others TBD
   Permissions: none. This is database table that is an enumeration of all the subjects that a student can filter by and that a tutor can claim to be certified in. The tutors will have a foreign key(s) that points to one or more columns in the subject table. Therefore Permissions for this entity(ies) is not applicable.

5. Chat Message:
   Definition: a message that exists in the messages database table. A student/tutor can send a message to a tutor/student respectively. The message object has a date and a recipient and a sender. These messages can be deleted and they can be edited. A priority 3 for this feature is the ability to report a message to the admin which could result in the sender being banned (account deleted and email/ip being placed on a blacklist.)

6. Review:

Definition: a message that exists in the reviews database table. A registered user can write a review of a tutor and have their review listed on that tutor's individual page. Reviews must be approved by an admin before they appear on the page.

SFSU custom functions:

Course numbers:
Database columns: one dedicated column in the user table that is a string that contains the course numbers for example: "CSC210 CSC220 COMM150 ENG114" could be one entry. We can search by course number with the same logic as searching by username via % LIKE MYSQL query. The course number option that we provide the tutor user with is currently our only special function. However, we plan on adding a map of SFSU that is pinnable. See functional requirements on this potential feature.

4. **Initial list of functional requirements**

Unregistered Users

1. User shall be able to search for tutors using a bar at the top of the site
2. When the user searches the search bar shall retain the user's search instead of being blanked
3. The user shall be able to view a search page listing multiple tutors and their basic info (photo, name, classes offered, rates)
4. The user shall be able to go to an individual tutor page that has the tutor's CV, optional video, and reviews of that tutor by registered users
5. User shall be able to view a link "register to contact tutor"
6. User shall be able to click register to sign up with an SFSU email address and a password
7. User shall be able to view a link on the homepage "register as a tutor!"
8. User shall be able to click that link to go to a form that asks for their tutor registration info as well as their SFSU email and a password
9. User shall be able to submit their photo, resume, and other tutoring info, then click register to sign up.
10. Users who submit tutoring info must wait for admin approval before their results become searchable

Registered users

11. Registered users shall inherit all functions from unregistered users except for the tutor signup form
12. Registered user shall be able to click a link on a tutor's individual page to send a message to that tutor
13. Registered user who registers as tutor shall be able to reply to messages they receive
14. Registered user shall be able to register as a tutor without seeing the user registration dialog
15. Registered user who registers as a tutor must still wait for admin approval before their information becomes viewable
16. Registered user shall be able to write reviews of tutors
17. Registered user shall be able to view a map of the sfsu campus
18. Registered user shall be able to "pin" the map for visibility by their contacted students or tutors to indicate meeting places
19. Registered users who sign up as tutors shall be able to offer group tutoring sessions for classes
20. Registered users shall be able to sign up for group tutoring by a tutor

Admins

21. Admins shall have the ability to view the database in MySQL Workbench
22. Admins shall be required to review media submitted by tutors for approval before it becomes visible on the site
23. Admins shall be required to review reviews of tutors before they become visible on the site

5. 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 render well on mobile devices
4. Data shall be stored in the database on the team's deployment server.
5. No more than 50 concurrent users shall be accessing the application at any time
6. Privacy of users shall be protected
7. The language used shall be English (no localization needed)
8. Application shall be very easy to use and intuitive
9. Application shall follow established architecture patterns
10. Application code and its repository shall be easy to inspect and maintain
11. Google analytics shall be used
12. No e-mail 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

13. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
14. Site security: basic best practices shall be applied (as covered in the class) for main data items
15. Media formats shall be standard as used in the market today
16. Modern SE processes and tools shall be used as specified in the class, including collaborative and continuous SW development
17. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Fall 2023. For Demonstration Only" at the top of the WWW page nav bar. (Important so as to not confuse this with a real application).

6. Competitive analysis:

Features	**Service Provider**			
	tutors.tech	**tutoring.sfsu.edu**	**tutor.com/higher-education**	**care.com/college-tutors**
Group tutoring sessions	✓	✓	✓	✓
In-person tutoring (at SFSU)	✓	✓	X	X
Interactive map of SFSU campus	✓	X	X	X
Verified tutors	✓	✓	✓	✓
SFSU-specific classes	✓	✓	X	X
Public tutor reviews	✓	X	X	✓

While our service is similar to others, the main benefit of using ours is better support for in-person tutoring. Our tutors have attended SFSU in the past and are better prepared to tutor SFSU classes. Our interactive map allows groups and one-on-one tutoring sessions to reserve the location in advance. We do not offer academic advising; this will help us focus on being the best tutoring service for SFSU students.

7. **High-level system architecture and technologies used:**

Server Host: AWS micro tier (1 core, 1GB ram)

Operating System: Amazon Linux 2023

Database: Mysql Ver 14.14 Distrib 5.7.43

Web Server: nginx version: nginx/1.24.0

Server Side Language: Javascript / node.js

Additional technologies:

Web application framework: Express

Node process manager: PM2

Frontend: React

SSL Cert: Lets Encrypt (Cert Bot)

Browsers supported: Microsoft Edge (latest version) Chrome (latest version) and Firefox (latest version)

8. **Use of ChatGPT (for those who used it, it is optional)**

We did not use ChatGPT in Milestone 1.

9. **Team and roles**

Dan Tahir - Team Lead

Cleveland Plonsey - Backend Lead

Ava Albert - Frontend Lead

Michael Mathews - Github Master

Griffin Evans - Team Member

Bryan Maldonado - Team Member

10. Checklist

So far all team members are engaged and attending team sessions when required ***OK***
Team found a time slot to meet outside of the class ***DONE***
Back end, Front end leads and Github master chosen ***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 ***ON TRACK***
Team reviewed class slides on requirements and use cases before drafting Milestone 1 ***DONE***
Team lead ensured that all team members read the final M1 and agree/understand it before submission ***DONE***
Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.) ***DONE***