

-TEAM 1

Team Lead: Shum Usami | susami@sfsu.edu,

Frontend Lead: Adharsh Thiagarajan,

Backend Lead: Devon Huang,

GitHub Master: Kim Nguyen,

Software Developer: Thiha Aung

BRAIN BUFFS

SW Engineering CSC648-848 Fall 2024

**Milestone 1: Use Cases, High Level Requirements, Architecture, and
Competitive Study**

Date Submitted:	Oct. 10, 2024
Date Revised:	

1. Executive Summary

Brain Buffs is a tutoring website by students for students. We understand what students want, a simple to use website to quickly search for someone who has experienced and excelled at the course that they are currently having trouble with.

Our functions shall be to allow SFSU students to browse for tutors and make it easy to register as a student and/or a tutor on our website. Users shall be able to create booking requests using their name and school email to contact tutors. Tutors shall be able to make tutor posts about what courses they are available to tutor for, which shall require approval from website admins. Tutors shall be able to check their dashboards for booking requests sent by users.

Brain Buffs is made up of SFSU students, Shun Usami, Adharsh Thiagarajan, Devon Huang, Thiha Aung, and Kim Nguyen, all of whom understand the importance of community-led academic success. Our goal for this website is to foster academic growth and collaboration among peers.

2. Persona

Emily Rodriguez (Student)

Age: 20, Undergraduate Computer Science student at SFSU



Pain Points:

- Overwhelmed by complex topics like file systems in CSC415 Operating System
- Unable to find tutor who completed the CSC415 course with the same professor

Goals:

- Find a tutor who is familiar with CSC 415 to improve their understanding and boost midterm grades

Character:

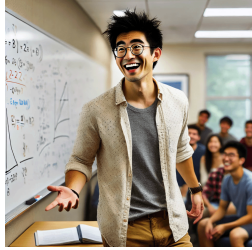
- Balances coursework, part-time job, and social life
- Part time working to pay her tuition, doesn't want to spend extra money

Skills:

- Has basic web browsing, online meeting, email skills on both PC/mobile

Michael Lee (Tutor)

Age: 20, Mathematics graduate student and tutor at SFSU



Pain Points:

- Overloaded with coordinating schedules
- Finished course, cannot find students to tutor

Goals:

- Efficiently manage multiple tutoring sessions
- Provide personalized support to students and earn money

Character:

- Passionate about teaching but overwhelmed with commitments
- Balances tutoring with research and personal projects

Skills:

- Proficient with online communication and scheduling tools on both PC/mobile

Persona: Dr. Sarah Thompson

Age: 45, Biology professor at SFSU



Pain Points:

- Organizing tutoring resources takes too much time;
- Limited insight into the effectiveness of tutoring services.

Goals:

- Enhance student engagement and performance
- Provide reliable tutoring support outside class hours

Character:

- Innovative educator committed to student success
- Embraces new teaching tools

Skills:

- Proficient with educational technologies and data analysis tools

3. High-level Use Cases

Case 1: Student Schedules a Tutoring Session (could not find appropriate tutor on another website)

Emily Rodriguez is a 20-year-old Computer Science student at SFSU, struggling with her CSC 415 Operating Systems class. She has been unable to find an appropriate tutor on other platforms. Frustrated, she discusses this with her classmate about this and they recommend Brain Buffs as it helped them through the previous semesters. Emily logs into **Brain Buffs**, a tutoring platform tailored to SFSU students. She uses the **search** feature to find a tutor who completed the same course. After filtering through some **tutor posts**, she discovers Michael Lee, a highly rated tutor with expertise in CSC 415. Emily quickly reviews his profile, checks his availability, and completes a **booking request**.

Case 2: Tutor Updates Availability and Manages Sessions

Michael Lee, a 20-year-old Mathematics graduate student, recently finished tutoring several students in CSC 415 and is eager to find new students to tutor. He logs into **Brain Buffs** and updates his weekly availability, specifying his tutoring subjects. Shortly after, Michael receives a notification from Emily Rodriguez requesting a tutoring session. Using his **dashboard**, Michael confirms the session, which automatically updates his schedule. With all his sessions in one place and no need to juggle multiple apps, Michael can efficiently manage his tutoring workload and focus on helping students like Emily succeed.

Case 3: Faculty Monitors Tutoring Support for Courses

Dr. Sarah Thompson, a Biology professor, notices an increase in her students seeking extra help outside of class hours. She accesses the Brain Buffs platform to monitor the tutoring activities related to her courses. Dr. Thompson reviews the profiles of available tutors, ensures they align with her course requirements, and provides feedback on their performance. This oversight helps maintain the quality of tutoring services and ensures that her students receive the support they need to excel.

Case 4: Student Provides Feedback and Rates Tutors

After a successful tutoring session, Emily wants to provide feedback on her experience. She logs into Brain Buffs, navigates to the session history, and leaves a **rating** and review for Michael Lee. This feedback not only helps Michael improve his tutoring methods but also assists other students in selecting the best tutors for their needs. Positive reviews boost Michael's profile, increasing his visibility on the platform.

Case 5: Tutor Shares Educational Resources with Students

During a tutoring session, Michael wants to share additional resources to help Emily further her understanding of algorithms. He uses the resource sharing feature on Brain Buffs to upload relevant study materials, such as lecture notes and practice problems. Emily accesses these resources directly through the platform, enhancing her learning experience and providing her with the tools needed to succeed in her coursework.

4. List of Main Data Items and Entities – Data Glossary/Description

Defined Terms:

Guest

- Can browse and **search** for tutors.
- Doesn't need to **login**.
- Cannot send **booking requests**, or make **tutor posting**.

User

- Represents anyone who has **registered** on the platform.
- Can be a student, a tutor, or both.
- Can create **booking requests**, browse and **search**.
- Can make a **Tutor posting** that would need to be approved by **Admin**.

Admin

- Manages and oversees the platform.
- Responsible for approving **tutor posting**.
- Ensure that the platform's content is appropriate and compliant with the platform's policies.
- Can delete inappropriate **users** and **tutor postings**.
- Doesn't need **login**, manages through workbench.

Tutor Posting

- Application process offered by a **user** who wants to be a tutor.
- applicants can advertise their subjects, services, rates, a detailed description of their offerings, a profile picture for the **tutor posting** (the picture will be a .jpeg image stored in a file system), and their CV(must be uploaded as a .pdf file).
- Are visible to **users** and **guests** searching for tutors and are subject to approval by the **admin**.

Booking Request

- Allows **users** to contact a tutor on the platform.
- Allow students to coordinate meetings.
- Is only seen privately by the tutor being requested.

Dashboard

- Interface for registered **users** (both tutors and students).
- Tutors can view their scheduled sessions, while students can track their **booking requests** with tutors.

Search

- This allows both unregistered and registered **users** to **search** for tutors based on tutor name, class subject, and rates.

Registration

- Enables **guests** to sign up for the platform, providing name and student email to create an account (this information is stored in the database).

Login

- A guest enters valid credentials (usually a student email and password) to access the platform as a **user**.
- After logging in, **users** gain full platform access and can interact with the site's features such as sending **booking requests** to tutors, viewing their **dashboard**, **searching** and browsing.
- Used by a pre-existing registered **user**.

Rating

- **User** can leave a 1-5 star rating for tutor

Data Items:

Registered User

- user id
- username
- password
- email
- name

Post

- post id
- user id
- profile picture
- title
- subject name
- class number
- content
- rate
- approved - status indicating if it was approved by **admin**

Booking -

- booking id
- sender id
- receiver id
- content
- approved - status indicating if it was approved by **admin**

Subject -

- subject id
- subject name

5. List of High-Level Functional Requirements

User Management

1. User Registration and Authentication

- Allow users to create accounts and securely log in using their school email.

2. Profile Management

- Enable users to create and update their profiles with relevant information, including subject expertise for tutors and academic interests for students.

Tutoring Management

3. Tutor Matching

- Provide functionality for students to **search** for and match with tutors based on subjects, availability, and **ratings**.

4. Availability Management for Tutors

- Provide tutors with tools to update and manage their availability, ensuring they can efficiently handle multiple **booking requests**.

5. Admin Approval for Tutor Posts

- Require website **administrators** to approve **tutor posts** related to course availability, maintaining quality and relevance of tutoring services.

6. Resource Sharing

- Allow tutors to share educational resources, such as lecture notes, practice problems, and study guides, with their students.

Session Management

7. Scheduling Sessions

- Allow users to schedule, reschedule, and cancel tutoring sessions through an intuitive calendar interface.

8. Session Notifications

- Send automated notifications to users about upcoming sessions, changes in schedule, and new **booking requests** to ensure timely communication.

Communication

9. In-app Messaging

- Facilitate real-time communication between students and tutors through an integrated messaging system.

10. Notification Settings

- Allow users to customize their notification preferences, choosing how and when they receive updates and alerts.

Feedback and Ratings

11. Review and Rating System

- Enable students to **rate** and review tutors after sessions, contributing to tutor profiles and helping future users make informed decisions.

Search and Filtering

12. Search and Filter Functionality

- Allow students to **search** for tutors using various filters such as subject, availability, **ratings**, and location.

Design and Integration

13. Responsive Design

- Ensure the application is fully responsive and accessible on various devices, including desktops, tablets, and smartphones.

14. Integration with SFSU Systems

- Seamlessly integrate with existing SFSU systems for user authentication, course listings, and other relevant data.

Security and Support

15. Security and Data Protection

- Implement data encryption and secure authentication mechanisms to protect user information and ensure privacy.

16. User Support and Help Center

- Provide users with access to a help center and support resources to assist them with any issues or questions related to the platform.

Analytics

17. Analytics and Reporting

- Enable administrators to access analytics and reports on tutoring activities, user engagement, and platform performance.

18. Backup and Recovery

- Implement robust backup and recovery procedures to ensure data integrity and availability in case of system failures.

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 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 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
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 (Functions/Features Only)

Feature/Function	Brain Buffs	Chegg Tutors	Wyzant	TutorMe
Custom SFSU Integration	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tutor Availability Updates	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
In-app Messaging	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Rating System	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Resource Sharing	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Summary:

Brain Buffs is designed to cater specifically to the SFSU community, setting it apart from broader tutoring platforms like Chegg Tutors, Wyzant, and TutorMe. Its custom integration with SFSU ensures that students can seamlessly access features tailored to their university experience, such as resource sharing unique to SFSU. This allows SFSU students who have an SFSU email or ID to sign up for a tutoring session for a certain subject or class being taught at SFSU or even become a tutor within SFSU to help a student with a subject they need help with. While competitors offer general tutor availability and messaging features, Brain Buffs enhances this with real-time updates specific to SFSU tutors and an in-app messaging system that fosters communication tailored to the academic environment. The built-in rating system helps ensure quality and accountability among tutors, further increasing user trust. By addressing specific needs of SFSU students, Brain Buffs provides a targeted solution that is not only convenient but also community-focused, ultimately increasing the likelihood of successful academic connections.

8. High-level System Architecture and Technologies Used

Database: MySQL v 8.0.39-0ubuntu

Web Server: NGINX 1.26.2

Operating System: Ubuntu 24.04

Server Server-Side Language: Python

Web Framework: Flask

Deployment cloud servicer: AWS, t2.micro, 1vCPU / 1GB Mem, x86 , Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz, 30GiB gp3 storage

Front-end frameworks: Bootstrap (newly added)

Python code formatting: Black (newly added)

HTML/CSS/JavaScript code formatting: Prettier (newly added)

Local development environment: NixOS/devenv

Workbench: MySQL workbench

Browsers:

1. Google Chrome:

Latest Version: 129.0.6668.89/.90 (October 1, 2024)

Previous Version: 128.0.6613.137/.138 (September 10, 2024)

Google Chrome is the most widely used web browser globally. The latest and previous versions ensure to cover the vast majority of Chrome users. These versions include latest security updates, new performance and feature improvements for web application faster and more reliable for users.

2. Mozilla Firefox:

Latest Version: 131.0 (October 1, 2024)

Previous Version: 130.0 (September 3, 2024)

Mozilla Firefox is one of the most popular browsers among users globally. 131.0 and 130.0 are stable releases designed to offer various performance improvements, security updates, and new features to improve user experience. Using the latest versions helps protect against vulnerabilities and ensures compatibility with modern web standards.

Major additional external open source APIs:

1. Google Analytics API: For tracking user interactions and gaining insights into website traffic.
2. Google Maps API: To integrate location features, such as finding tutors and departments' location.

9. Use of GenAI Tools like ChatGPT and Copilot for Milestone 1

Tool Used:

ChatGPT (Version 4, o1 preview)

Tasks and Usefulness:

1. Drafting Documentation
 - Usefulness: HIGH
 - ChatGPT assisted in creating initial drafts for the Executive Summary and Use Cases, saving time and ensuring clarity.
2. Generating Code Snippets
 - Usefulness: MEDIUM
 - Used to generate boilerplate code for tasks
3. Brainstorming Features
 - Usefulness: HIGH
 - Facilitated brainstorming sessions to identify unique SFSU-specific functionalities that differentiate Brain Buffs from competitors.

Examples and Prompts:

- "ChatGPT, help me draft an executive summary for a tutoring website tailored to SFSU students."
- Generate a list of high-level use cases for a student-tutor matching application.
- The usage of ChatGPT in Section 4 helped provide rough ideas of what each term meant. Also helped refine the rough ideas to make them more readable and precise.

- ChatGPT helped organize the competitive analysis by creating a clear comparison table and highlighting Brain Buffs' unique features. It also summarized the platform's advantages concisely, aligning with assignment goals.
- Utilization of ChatGPT, seeking an appropriate browser extension or configuration that is compatible with both Chrome and Firefox to efficiently run the project across different platforms.
- The usage of ChatGPT, primarily employed for conducting comprehensive grammar checks and improving the overall clarity and coherence of written content to ensure it meets professional standards.









Additional Comments:

- GenAI tools significantly enhanced our efficiency in documentation and initial code development. They provided valuable suggestions that the team refined and implemented.
- ChatGPT offered well-organized ideas and suggestions. Copilot ensured consistency across the codebase, making collaboration smoother.
- These tools also served as learning aids, allowing team members to refine AI-generated code and grasp new coding techniques.

10. Team and Roles

Name	Role
-----	-----
Shun Usami	TeamLead
Adharsh Thiagarajan	Frontend Lead
Devon Huang	Backend Lead
Kim Nguyen	Github Master
Thiha Aung	Software Developer

11. Team Lead Checklist

So far all team members are fully engaged and attending team sessions when required	
Team found a time slot to meet outside of the class	 (Thu. 2pm)
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	
Team reviewed class slides on requirements and use cases before drafting Milestone 1	
Team reviewed non-functional requirements from “How to start...” document and developed Milestone 1 consistently	
Team lead checked Milestone 1 document for quality, completeness, formatting and compliance with instructions before the submission	
Team lead ensured that all team members read the final M1 and agree/understand it before submission	
Team shared and discussed experience with genAI tools among themselves	
Github is organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)	