SW Engineering CSC648-848
Fall 2025
Team 08

Kojiro Miura Kmiura@sfsu.edu	**Team Lead**
Atharva Walawalkar	**Backend Lead**
Addhyan Kohli	**(Frontend Lead)**
Krinjal Basnet	**(Frontend Dev)**
Aketzali Zeledon	**Backend Dev**
Sonam Tobgyal	**Github Master**

Initial Submission	11/2/2025
Revised submission	

Table of Contents

1. Executive summary

The SFSU Tutoring Portal is an exclusive academic platform developed to connect San Francisco State University students with trusted and verified tutors in a secure and professional environment. Designed to make academic support simple, reliable, and accessible, the platform allows students to easily search for tutors based on course, subject, or language, and schedule sessions that fit their needs. Each user registers with an official SFSU email address, ensuring that the system remains exclusive to the university community and maintains academic credibility. Tutors can manage their profiles, share learning resources, and set availability, while administrators review and approve all listings to uphold quality and integrity.

Students using the platform can search for tutors by course, subject, or language, making it easier than ever to find help that directly matches their academic goals. The system's user-friendly interface allows for smooth navigation, quick scheduling, and instant access to support that fits any learning style or schedule. Tutors can manage their profiles, showcase their expertise, and upload learning materials to enhance student understanding. The platform ensures that all listings are verified and approved by administrators, maintaining a high standard of quality and compliance with university policies. Communication between tutors and students is streamlined through an in-site messaging system that promotes professionalism and protects privacy, ensuring that every interaction stays productive and secure.

What makes the SFSU Tutoring Portal truly stand out is its commitment to inclusivity, convenience, and academic growth. Its modern and mobile-friendly design enables students to access tutoring anytime, anywhere, turning academic help into a seamless part of campus life. Whether students need support for challenging coursework, test preparation, or ongoing mentoring, the portal offers an environment that encourages continuous learning and personal improvement. More than just a website, it functions as a trusted academic ecosystem that empowers every SFSU student to take ownership of their education, build confidence, and achieve excellence through guided collaboration.

Behind this impactful initiative stands a passionate and highly skilled team dedicated to transforming the way academic support is delivered. The project is led by **Kojiro Miura**, who serves as the Team Lead and guides the team with vision and direction. **Atharva Walawalkar** oversees the backend development as Backend Lead, while **Addhyan Kohli** leads the frontend design and user experience. **Krinjal Basnet** contributes as Frontend Developer, ensuring functionality and aesthetic balance. **Sonam Tobgyal** manages version control and coordination as GitHub Master, and **Aketzali Zeledon** strengthens backend performance and reliability as Backend Developer. Together, this talented group has created more than just a platform as they have built a digital foundation that strengthens academic connections, supports every student's learning journey, and redefines what it means to learn and grow within the SFSU community.

2. List of main data items and entities

- **Users -** A person who logins through the SFSU tutoring portal.
  - User ID - Mandatory
  - SFSU Email - Mandatory
  - Password - Mandatory
  - First and Last Name - Mandatory
- **Student** - A person who logins through the SFSU tutoring portal.
  - Course request - Mandatory
- **Tutor** - An SFSU enrolled person (or student) who offers tutoring services.
  - Biography - Optional
  - Hourly rate - Mandatory
  - Language - Optional
  - Profile picture - Mandatory
  - Schedule management - Optional
- **Student & Tutor** - A user who serves both as tutor and as a student.
  - Course management - Mandatory
  - Schedule management - Mandatory
- **Admin** - SFSU tutoring staff who manages the tutoring website.
  - Management tools - Mandatory
- **Class Options** - The list of available SFSU courses for the semester as configured by the Admin.
  - Course title - Mandatory
  - Tutor name - Mandatory
  - Course number - Mandatory
- **Course Listings** - Courses offered by tutors.
  - Course title - Mandatory
  - Tutor name - Mandatory
  - Course number - Mandatory
- **Tutor profile** - Information about tutor name, subjects, schedule, and credentials.
  - Biography - Optional
  - Hourly rate - Mandatory
  - Language - Optional
  - Schedule management - Mandatory
- **Student profile** - Contains information about enrolled classes, requests.
- **Tutoring Session** - A scheduled meeting between student and tutor including course, time, location or meeting link.
  - Meeting link - Mandatory

- **Meeting link** - The meeting link automatically sent to both tutor and student when a session is confirmed.
- **Course Coverage Request** - A request submitted by a student when a desired course is not currently offered for tutoring.
- **Session Feedback** - Feedback after a completed session.
- **Tutor Management Panel** - Admin interface used to update tutor information, assign courses, and remove accounts.
    - Schedule management - Mandatory
- **Message** - Messaging system between students and tutors.
- **Session** - Booked tutoring meeting student and tutor.
- **SFSU Portal** - Authentication gateway for all users (students, tutors, admins) to log in securely.
    - User ID - Mandatory
    - SFSU Email - Mandatory
    - Password - Mandatory
    - First and last name - Mandatory
- **Session Reports** - Records that show student attendance, tutor activity, and overall platform usage, used by Admin for monitoring.
- **Search Filter** - Users can filter by course number, subject, or language using a search bar and category menu.
- **Schedule Management** - Tutors are able to manage their time slots by adding or removing time slots.
- **Management tools -** Tutors are able to remove duplicated or outdated entries.

3. Functional Requirements - prioritized

Priority 1

Browse Tutor Profiles: Unregistered users shall be able to view publicly visible tutor listings without the need to log in.

Search and Filter Tutors: Unregistered users shall have access to a responsive search interface that allows filtering of tutors by course number, subject area, or language.

Register/Account Creation - Users shall be able to create an account by registering with a valid SFSU email address. The registration process shall include input validation, email verification, and secure password handling.

Login
Registered users shall be able to securely log in using their SFSU email address and password. Login sessions shall utilize secure authentication protocols.

Logout
Registered users shall be able to log out at any time, ensuring that all session data is cleared from the active session cache.

Registered Students

In-Site Messaging: Registered students shall be able to engage in secure, one-to-one messaging with tutors within the platform. All communications shall be privacy-protected.

Book Tutoring Sessions: Registered students shall be able to schedule tutoring sessions with verified tutors. Upon confirmation, the system shall automatically generate a Zoom meeting link and a calendar invitation.

Request Unavailable Courses (SFSU-Specific): If no tutor is available for a particular course, students shall be able to submit a *Course Coverage Request* containing the course number, topic, and availability. The system shall notify qualified tutors of the request.

View and Manage Bookings: Students shall be able to view all confirmed, pending, and completed tutoring sessions through a centralized dashboard.

Student Dashboard and Navigation- The student home page shall provide quick-access links to key features including: Find a Tutor, Find a Course, Course Coverage Request, My Schedule, Today's Sessions

Registered Tutors

In-Site Messaging: Registered tutors shall be able to communicate privately and securely with students through the platform's internal messaging system.

Manage Tutor Profile: Tutors shall be able to create and update their profiles, including information such as subjects offered, hourly rates, languages spoken, and availability.

Receive Booking Notifications: Tutors shall receive both in-app and email notifications whenever a student books a session.

Tutor Dashboard and Navigation- The tutor home page shall include links to: Find a Tutor, Find a Course, Course Coverage Request, My Schedule, Courses Taught ,Appointment Requests, Upcoming Sessions

Administrators

Manage Course Listings: Administrators shall be able to add, edit, or deactivate course listings each semester, in alignment with official SFSU catalog data.

Handle Reports and Maintenance: Administrators shall resolve reported issues, remove duplicate or outdated tutor entries, and ensure overall system integrity and data accuracy.

Administrative Dashboard and Tools- The administrator home page shall include tools and navigation links for: Viewing all registered students and tutors, Reviewing tutor course requests, Managing course coverage requests, Reviewing student and tutor reports

Priority 2

Registered Students

Rate Completed Sessions: Upon completion of a tutoring session, students shall be prompted to provide feedback consisting of a rating and optional written comments for quality assurance purposes.

Registered Tutors

Schedule Management: Tutors shall be able to manage their availability by adding or removing time slots. The system shall provide an option to synchronize schedules with Google Calendar or a third-party calendar application.

Access Session Analytics: Tutors shall have access to a summary dashboard displaying session statistics and student ratings.

Administrators

Dashboard Analytics: Administrators shall have access to analytical dashboards summarizing system usage metrics.

Archive Semester Data: Administrators shall be able to archive session data from previous semesters for auditing and reporting purposes without affecting current operational records.

Priority 3

Registered Students

(To be determined in future iterations.)

Registered Tutors

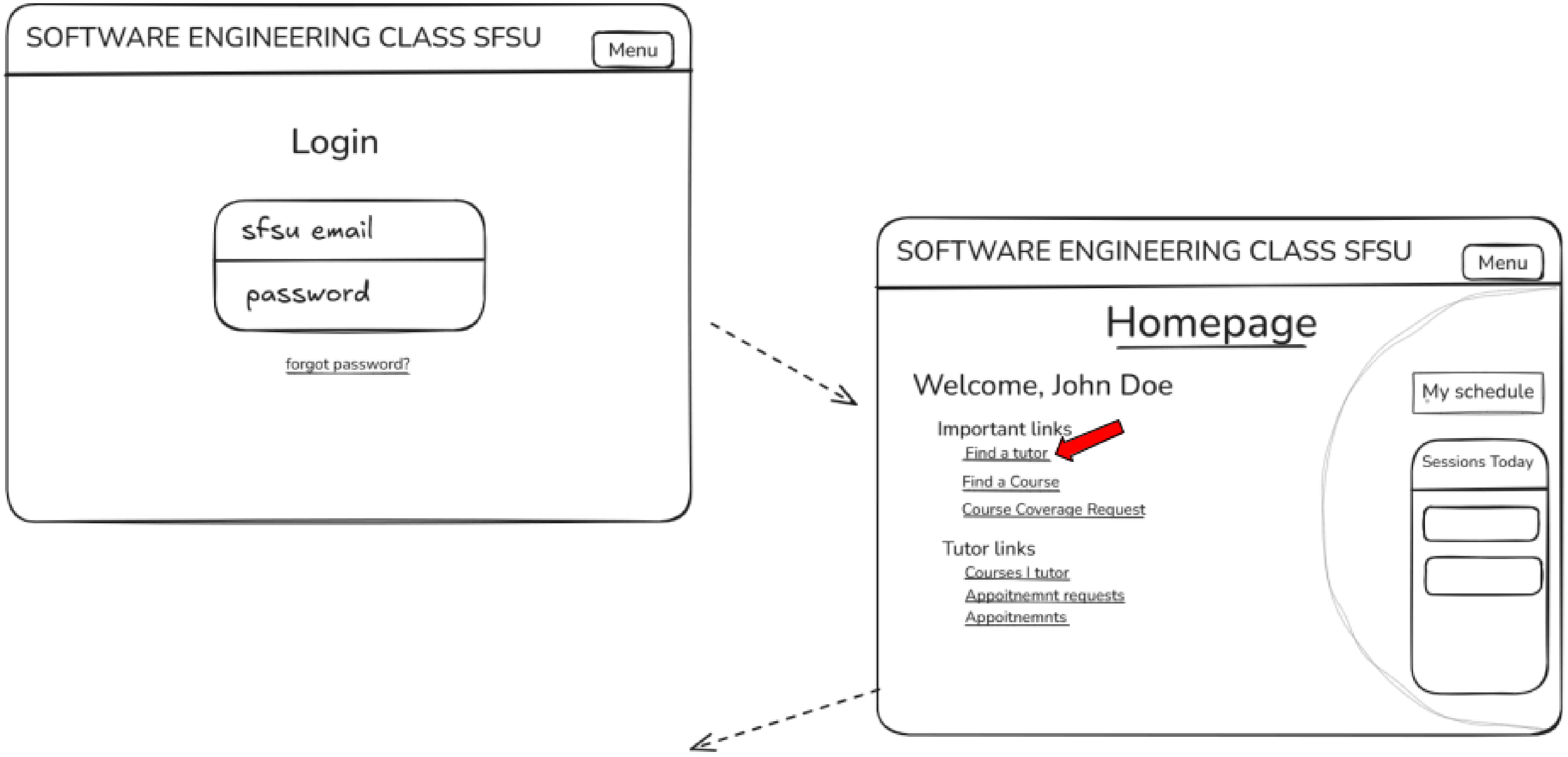
(To be determined in future iterations.)

Administrators

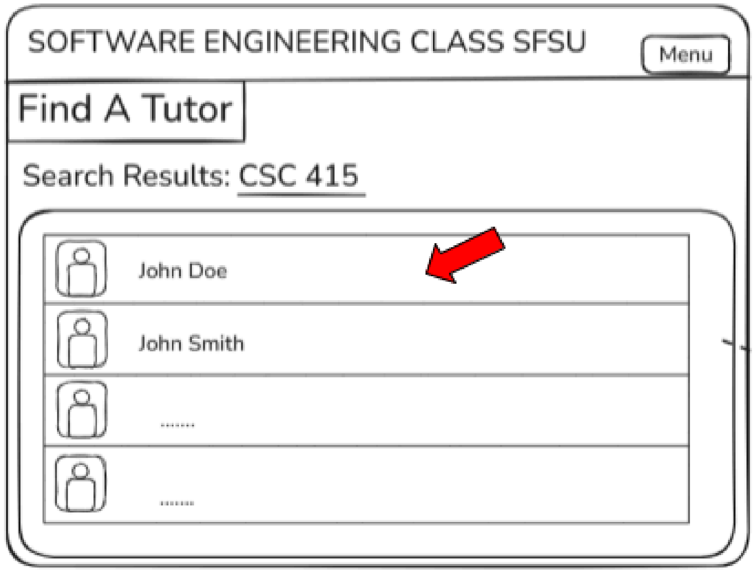
(To be determined in future iterations.)

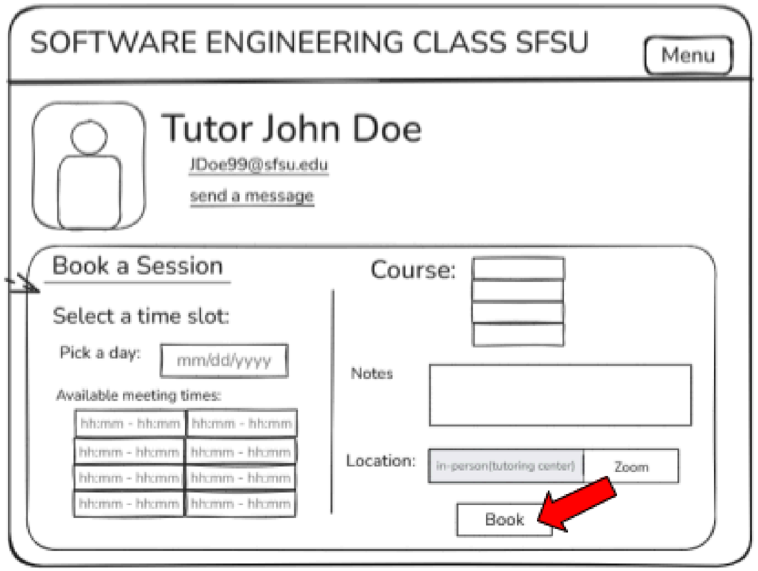
UI Storyboards for each main use case

Storyboard 1: Student schedules a tutoring meeting



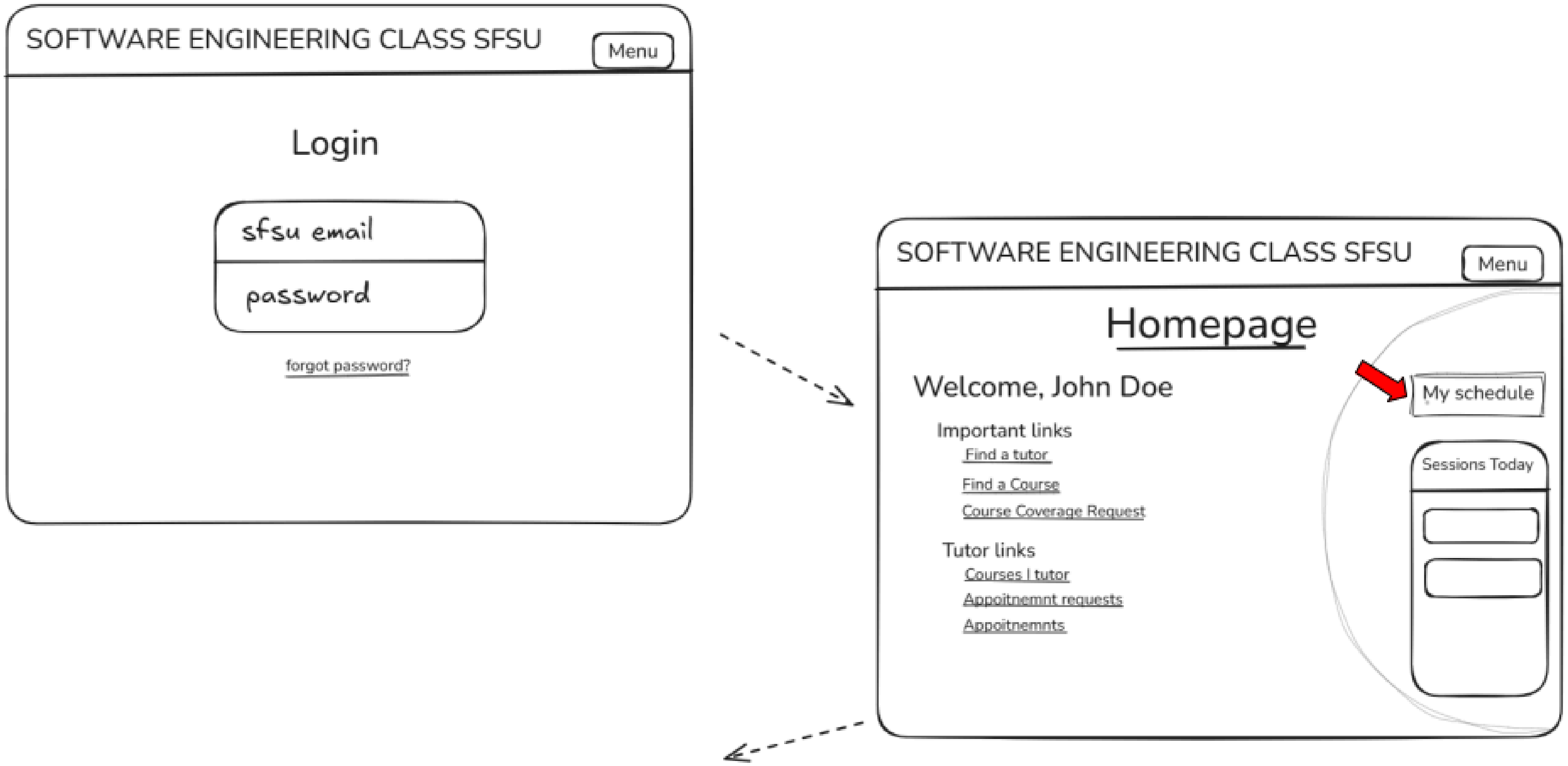
Search results for query based on course or tutor name

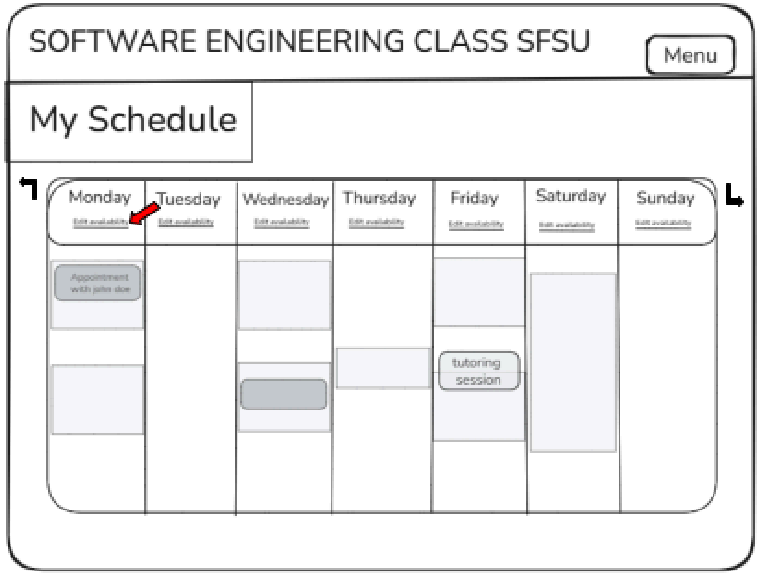




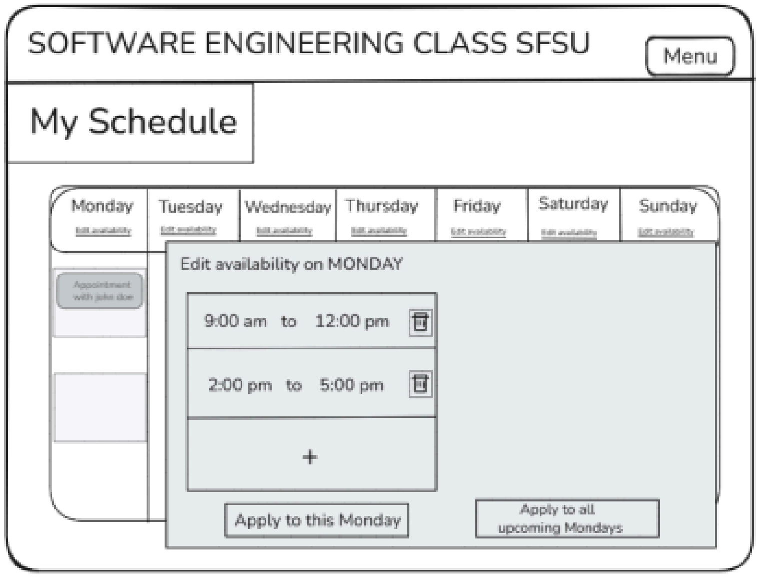
Book for a course on an available time slot

Storyboard 2: Tutor inserts their time availability



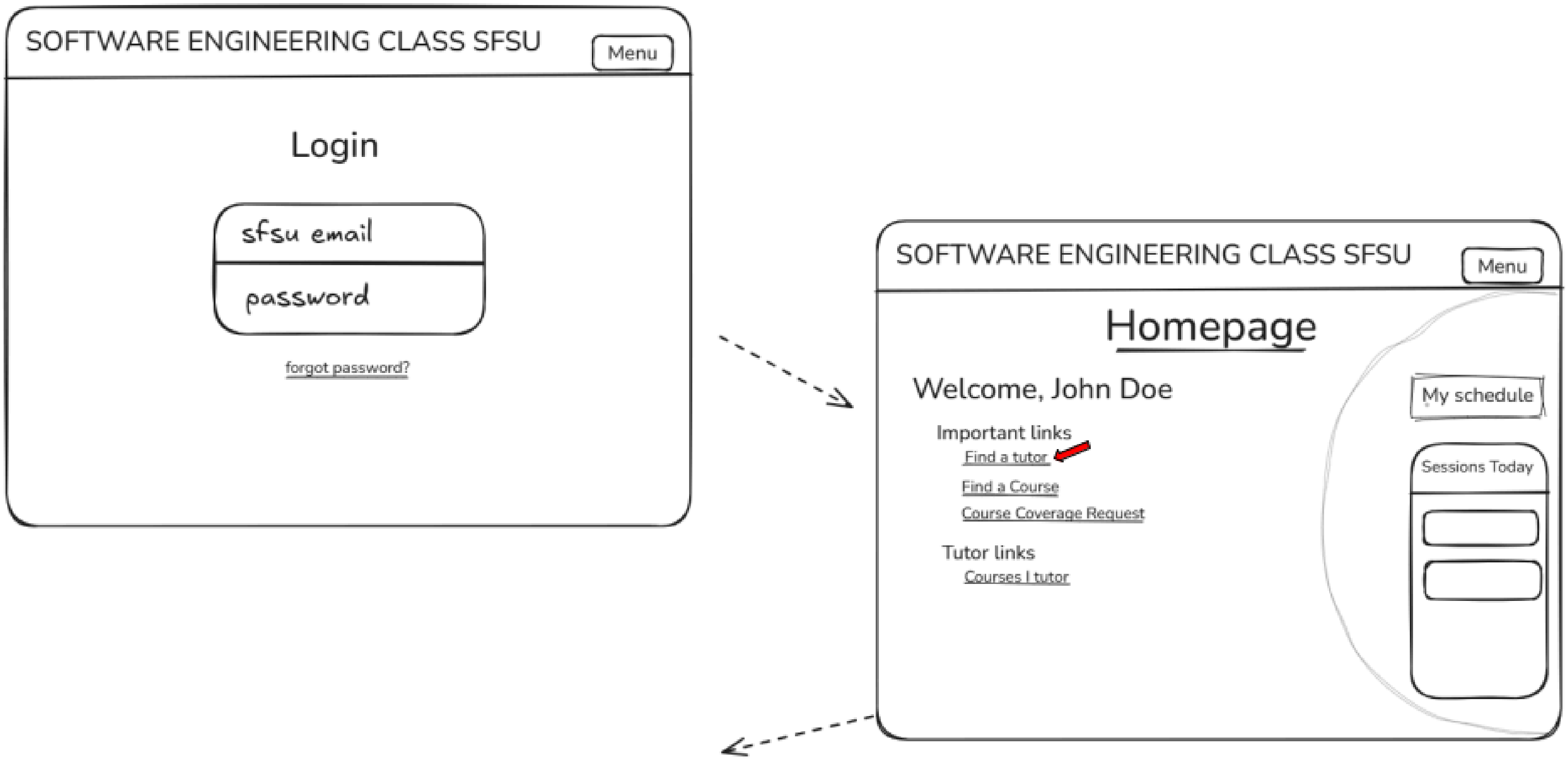


Calender for sessions and classes for the week

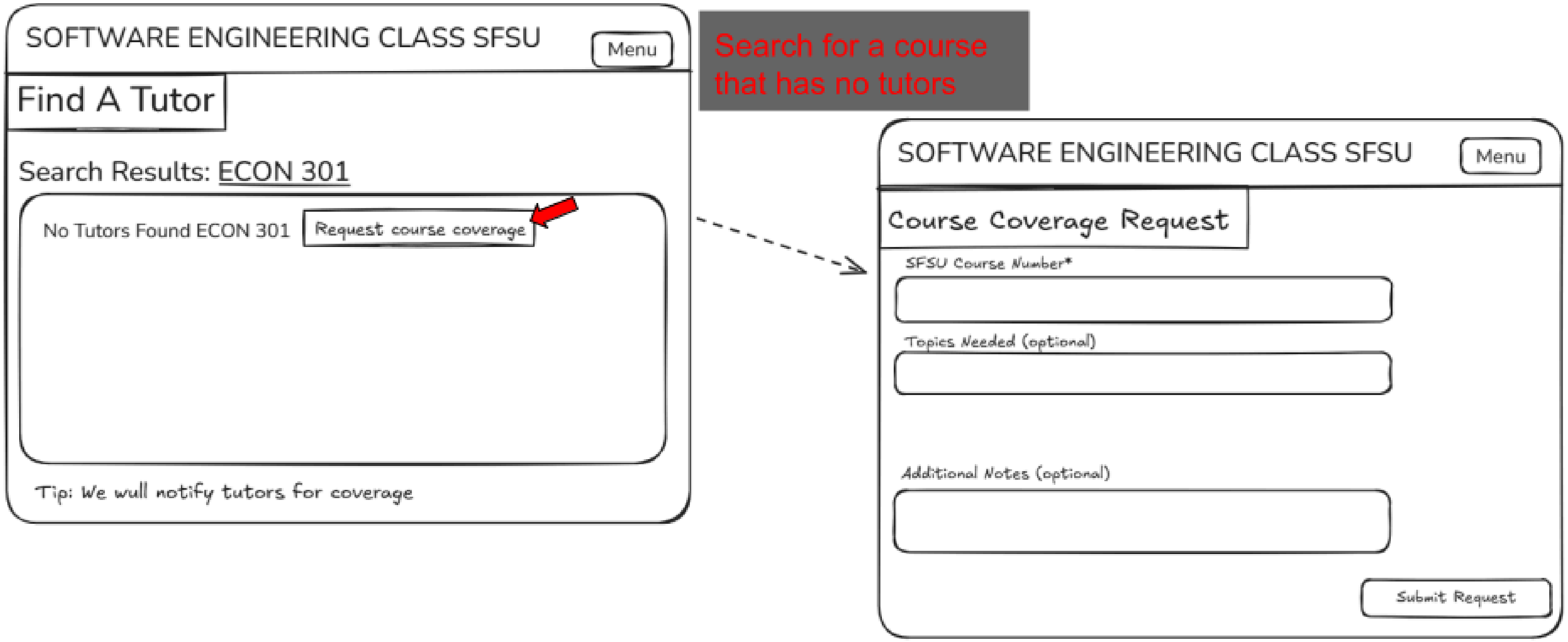


Apply edits to this week or all weeks for the semester

Storyboard 3: Student request coverage for a not offered course



]



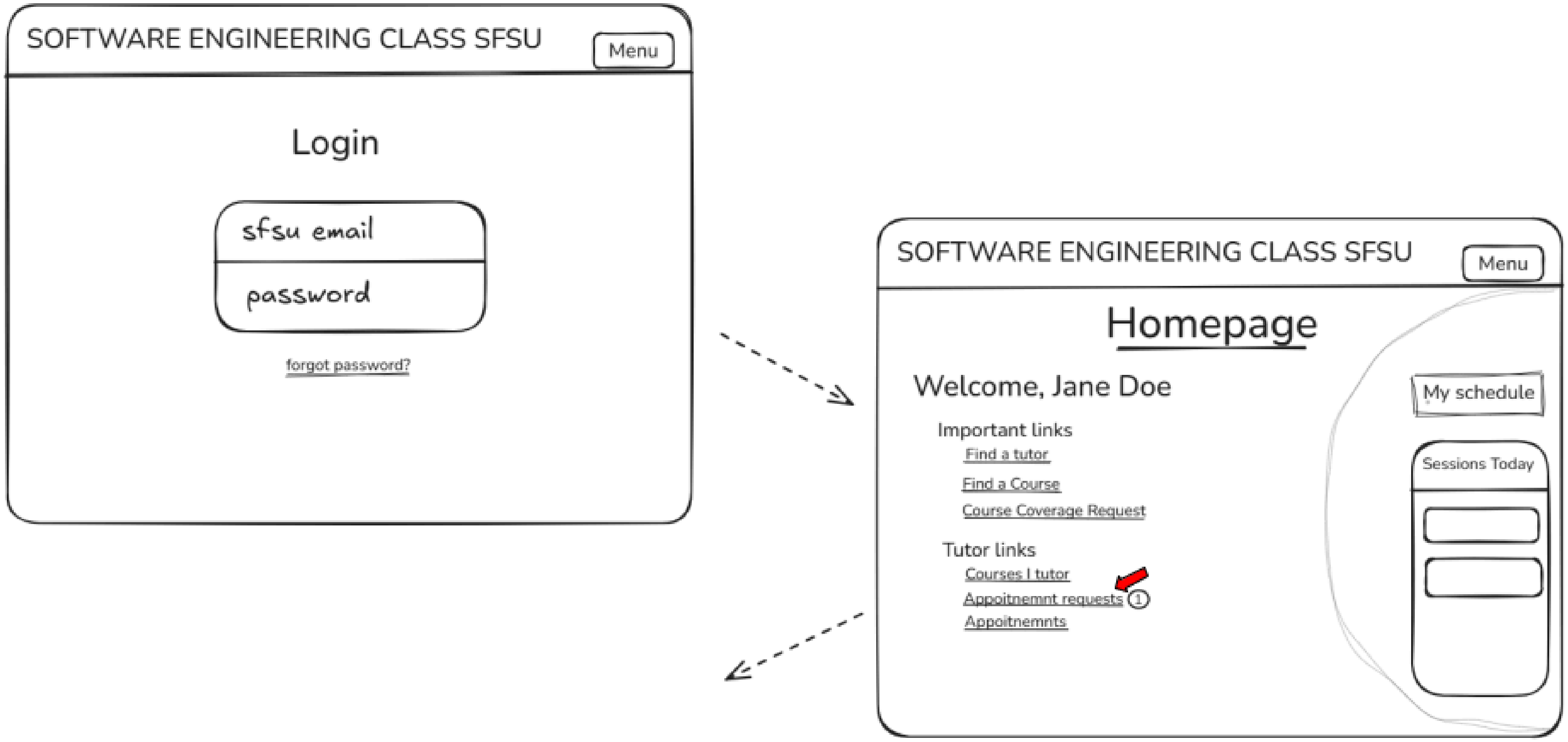
Search Results: ECON 301

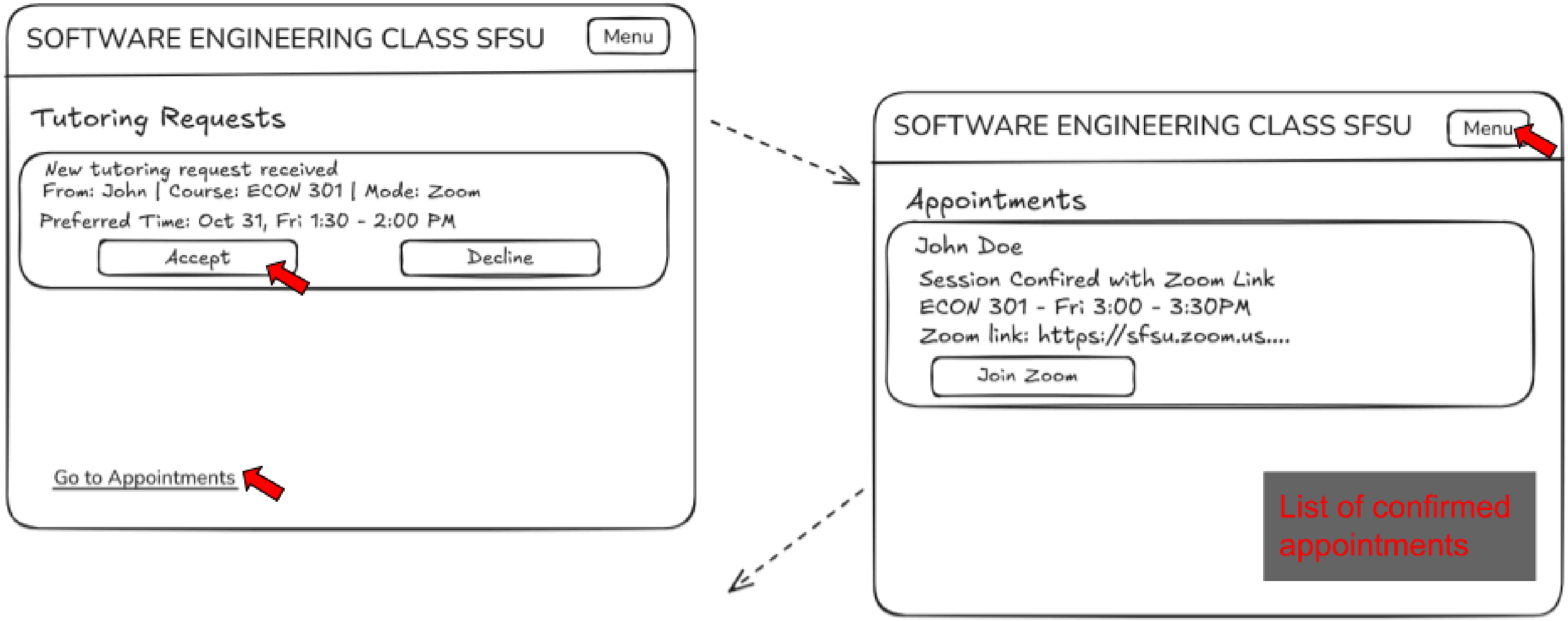
No Tutors Found ECON 301

Tip: We wull notify tutors for coverage

Search for a course that has no tutors

Storyboard 4: A Student and tutor, receives appt req along with a zoom link and books a session with another tutor





SOFTWARE ENGINEERING CLASS SFSU

Tutoring Requests

New tutoring request received
From: John | Course: ECON 301 | Mode: Zoom
Preferred Time: Oct 31, Fri 1:30 - 2:00 PM

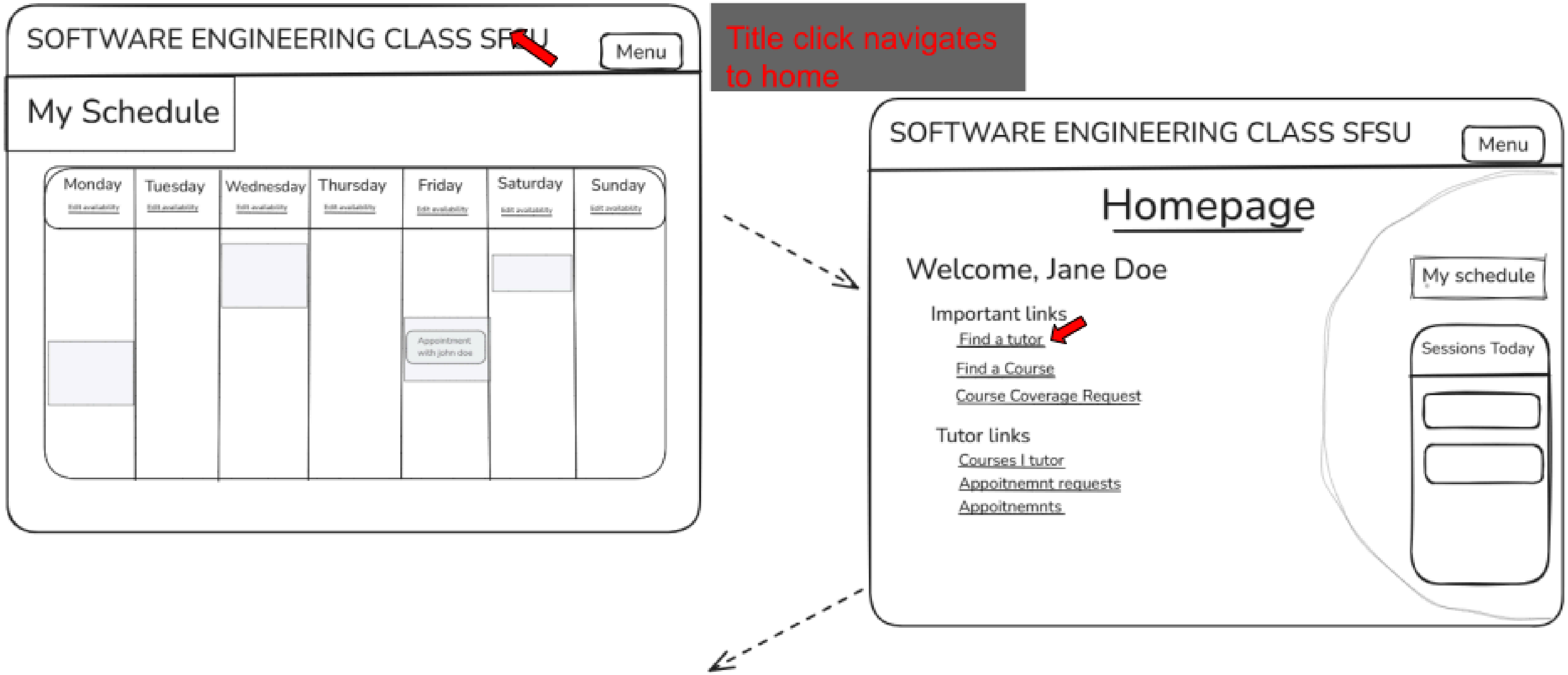
SOFTWARE ENGINEERING CLASS SFSU

Appointments

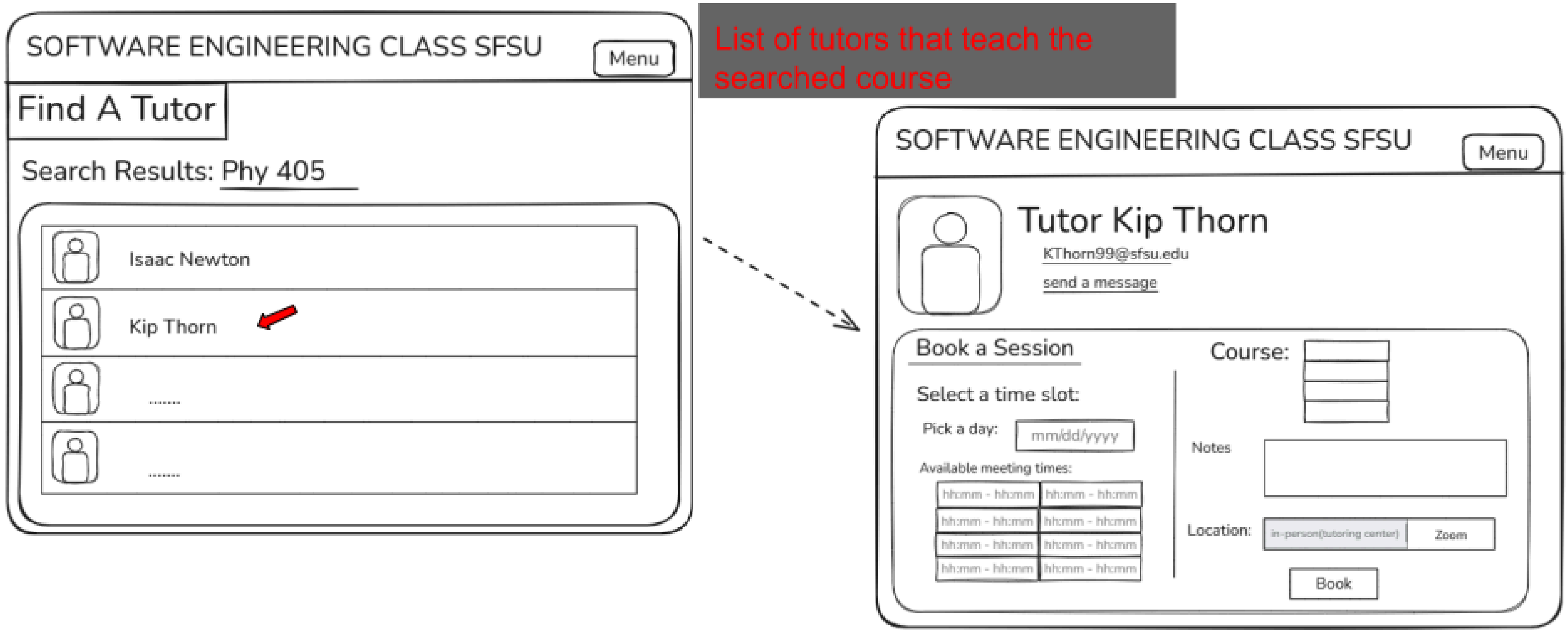
John Doe
Session Confired with Zoom Link
ECON 301 - Fri 3:00 - 3:30PM
Zoom link: https://sfsu.zoom.us....

List of confirmed appointments

Navigate to My schedule through Menu options



Title click navigates to home



List of tutors that teach the searched course

High level Architecture, Database Organization summary only

Database Schema:

users :

Columns:

user_id primary Key,
sfsu_email UNIQUE,
first_name,
last_name,
role ENUM('student','tutor','both','admin'),
password_hash,
created_at,
updated_at

Description: All registered accounts. Core identity table every other one links to.

Tutor_profiles :

Columns:

tutor_id primary Key/Foreign Key→users.user_id,
bio,
hourly_rate_cents,
languages,
status ENUM('pending','approved','rejected'),
profile_image_path_full,
profile_image_path_thumb,
last_active_at

Description: Extra details for tutors; only approved ones are visible in search.

courses :

Columns:

course_id primary Key,
department_code,
course_number,
title,
is_active

Description: Canonical list of courses; drives category dropdowns.

tutor_courses :

Columns:

tutor_id Foreign Key→tutor_profiles.tutor_id,
course_id Foreign Key→courses.course_id,
primary Key(tutor_id, course_id)

Description: Many-to-many join linking tutors to courses they teach.

availability_slots :
Columns:
slot_id primary Key,
tutor_id Foreign Key,
weekday TINYINT,
start_time TIME, end_time TIME,
location_mode ENUM('online','campus'), location_note
Description: Time windows tutors are available; used for scheduling filters.

sessions :
Columns:
session_id primary Key,
student_id Foreign Key→users.user_id,
tutor_id Foreign Key→users.user_id,
course_id Foreign Key→courses.course_id,
start_datetime, end_datetime, meeting_link,
status ENUM('pending','confirmed','completed','cancelled'),
created_at
Description: Booked tutoring meetings between students and tutors.

messages :
Columns:
message_id primary Key,
from_user_id Foreign Key,
to_user_id Foreign Key,
tutor_id Foreign Key,
body, created_at
Description: Private messages between student and tutor for coordination.

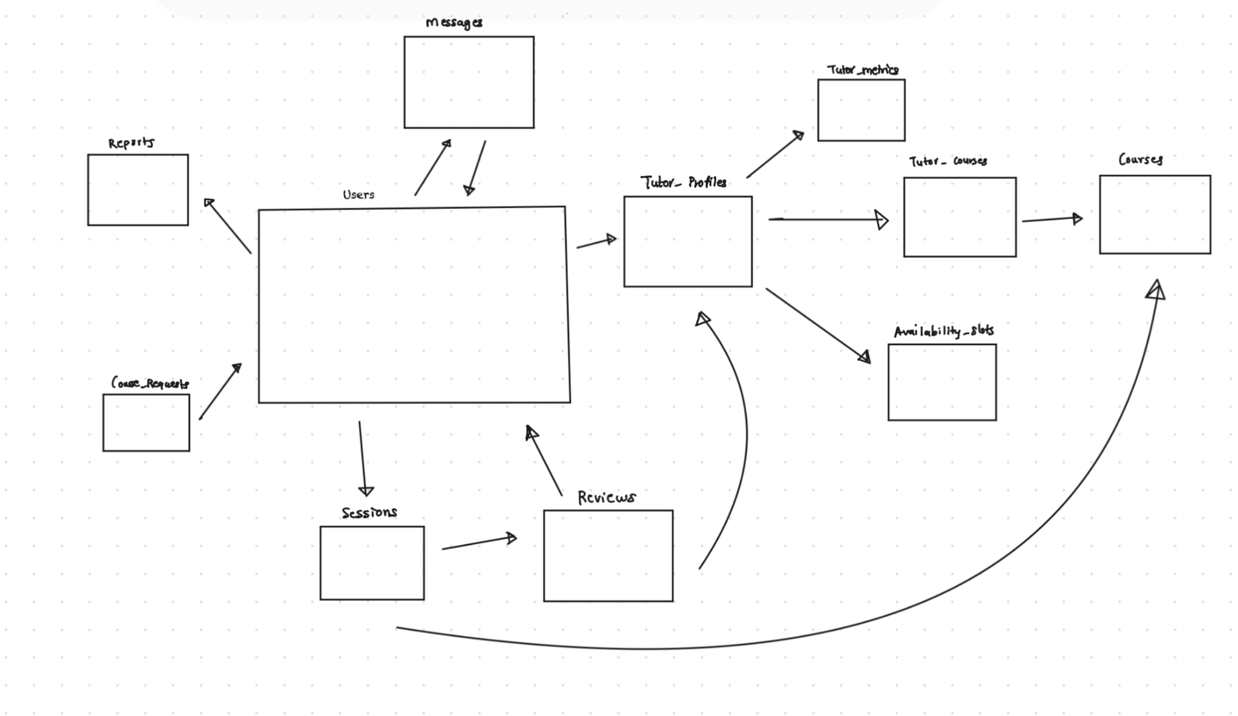
reports :
Columns:
report_id primary Key,
reporter_user_id Foreign Key,
reported_tutor_id Foreign Key,
reason, created_at
Description: User-submitted complaints or issue logs.

course_requests :
Columns:
request_id primary Key,
student_id Foreign Key,
department_code, course_number,
Topic_notes,
preferred_rate_cents_min,

preferred_rate_cents_max,
availability_note,
status ENUM('open','matched','closed'),
created_at
Description: Students requesting help for unlisted or new courses.

reviews :
Columns:
review_id primary Key,
session_id Foreign Key→sessions.session_id,
student_id Foreign Key,
tutor_id Foreign Key,
rating TINYINT(1-5),
comment, created_at
Description: Feedback after a completed session; feeds ranking metrics.

tutor_metrics :
Columns:
tutor_id primary Key/Foreign Key→tutor_profiles.tutor_id,
sessions_completed INT,
avg_rating DECIMAL(3,2),
response_time_seconds_avg INT, views_count INT,
last_session_at DATETIME
Description: Aggregated performance data for ranking and analytics.



Multimedia Storage:

Media storage including images and video/audio will be kept in a file system structure.
FastAPI will handle uploads using endpoints and will be saved in a */multimedia* file of the file system structure. This is followed by storing the relative path in the table of ‘tutor_profiles’.As shown in the database organization section, we will store the image and image thumbnails in file relative paths under names such as “profile_image_path_full” and “profile_image_path_thumb”. Then Ngnx will serve the static files and react will display them via the stored paths.

Searching and Filtering:
Search will be a combination of categories with fuzzy text search. Categories will be associated with specific SQL database tables and %LIKE will be used to implement the fuzzy text search for the specified database table.

Search Categories Displayed to User	**Associated DB Table fields**
Courses	courses.department_code, courses.course_number, courses.title
Tutor	tutor_profiles, user.first_name, user.lastname
Availability	availability_slots.weekday
Languages	tutor_profiles.languages
Price	tutor_profiles.hourly_rate_cents

Non-Trivial Algos:
Individual tutor “ratings” are contained in the ‘reviews’ table and as a new review is created, the “avg_rating” of a tutor will need to be updated in the ‘tutor_metrics’ table.

Identify actual key risks for your project at this time

Skills Risk:

Team members have varying levels of experience with React, FastAPI, and AWS deployment.
Solution: Both backend and frontend devs will be working alongside their Dev Leads to better learn their respective systems.

Schedule Risks:

Limited time between M2 part 1 and M2 part 2 and the upcoming M3. M2 part 2 is the start of the vertical prototype build, so design and coding work may compete for time with the completion of M2 part 1.

Solution: Working on both parts of M2 simultaneously helps with our current schedule as well as prepare as best we can for the future Milestones.

Technical Risks:

Teamwork Risks:

Project management

So far, I've been simply organizing what tasks I've given everyone into an excel sheet. I don't really give them any hard deadlines until our due dates are given. I just try to have things at the very least started. Once due dates are given I will give them a date of when I would like their tasks to be completed in order for us as a team to look over it before I hand it off to the class CTO and CEO (professor).

Use of GenAI tools

For Functional Requirements, I used ChatGPT to take our M1 functional requirements and had it help me brainstorm our priority list while also adjusting it according to our plans for implementation. I also used it to make the language used more formal.

For the backend I used Claude as a technical reviewer to critique my backend database schema and search algorithm design. Its feedback helped me identify potential issues and edge cases.

For the list of main data types and entities, I tried using ChatGPT, however it didn't come up with any useful ideas to expand on.

Team Lead Checklist

- So far all team members are fully engaged and attending team sessions when required
  - On Track

- 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
  - Completed

- Team reviewed suggested resources before drafting Milestone 2
  - Completed

- Team lead checked Milestone 2 document for quality, completeness, formatting and compliance with instructions before the submission
  - Completed

- Team lead ensured that all team members read the final Milestone 2 document and agree/understand it before submission
    - Completed

- Team shared and discussed experience with GenAI tools among themselves
    - Completed