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

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.

List of main data items and entities

Data items	Description	Mandatory / Optional
Users	A person who logins through the SFSU tutoring portal. Sub items: ● User ID ● SFSU Email ● Password ● First and Last Name	Mandatory
Student	A person enrolled at SFSU, using the website. Sub items: ● Course Request	Mandatory
Tutor	An SFSU enrolled person (or student) who offers tutoring services. Sub items: ● Biography ● Hourly rate ● Language ● Profile picture ● Schedule management	All optional except hourly rate and schedule management
Student & Tutor	A user who serves both as tutor and as a student. Sub items: ● Course Request ● Schedule management	Mandatory

Admin	SFSU tutoring staff who manages the tutoring website. Sub items: ● Management tools (Approve and ban, course management, reports / maintenance)	Mandatory
Class Options	The list of available SFSU courses for the semester as configured by the Admin. Sub items: ● Course title ● Tutor name ● Course number	Mandatory
Course Listings	Courses offered by tutors. Sub items: ● Course title ● Tutor name ● Course number	Mandatory
Tutor profile	Information about tutor name, subjects, schedule, and credentials. Sub items: ● Biography ● Hourly rate ● Schedule management ● Profile picture	All optional except hourly rate and availability schedule
Student profile	Contains information about enrolled classes, requests.	Mandatory

Tutoring Session	A scheduled meeting between student and tutor including course, time, location or meeting link. Sub items: ● Location / meeting link	Mandatory
Meeting Link	The meeting link automatically sent to both tutor and student when a session is confirmed.	Mandatory
Course Coverage Request	A request submitted by a student when a desired course is not currently offered for tutoring.	Mandatory
Session Feedback	Feedback after a completed session. Sub items: ● Review ● Rating ● Comment	Comment optional, rest mandatory,
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.	Optional
Session	Booked tutoring meeting student and tutor.	Mandatory
SFSU Portal	Authentication gateway for all users (students, tutors, admins) to log in securely. Sub items: ● User ID ● SFSU Email ● Password ● First and last name	Mandatory
Session Reports	Records that show student attendance, tutor activity, and overall platform usage, used by Admin for monitoring.	Optional
Search Filter	Users can filter by course number, subject, or language using a search bar and category menu.	Mandatory

Functional Requirements - prioritized

Priority 1

Unregistered Users

Browse Tutor Profiles – Users can browse visible tutor listings without logging in.
Search and Filter Tutors – Users can filter by course number, subject, or language using a responsive search bar and category menu

Registered Students

Book Tutoring Session – Students can schedule sessions with verified tutors; confirmations automatically include Zoom link + calendar invite.
Request Unavailable Courses (SFSU specific) – If no tutor exists for a course, students can submit a Course Coverage Request containing course number, topic and availability; system notifies qualified tutors.
View and Manage Bookings – Students see all confirmed, pending, and past sessions in their dashboard.

Registered Tutors

Manage Tutor Profile – Tutors create/update profiles including subjects, rates, languages, and availability
Approve/Reject Cycle – Edited profiles enter "Pending" state for admin review before visibility.
Receive Booking Notifications – Tutors are notified in-app and by email when a student books a session.

Administrators

Approve Tutor Profiles – Admins review, approve, or reject pending profiles.
Manage Course Listings – Admins add, edit, or deactivate courses each semester using official SFSU catalog data.
Handle Reports and Maintenance – Admins resolve reported items, remove duplicate or outdated tutors, and maintain system integrity.

Priority 2

Registered Students

In-site Messaging – One-to-one messaging between students and tutors within portal (privacy-protected).
Rate Completed Sessions – After each session, students submit feedback (rating + comments) for quality control.

Registered Tutors

Schedule Management – Tutors add or remove time slots and sync with Google Calendar or some 3rd party calendar app

Access Session Analytics – Tutors view a simple summary of completed sessions and ratings.

Administrators

Dashboard Analytics – Admins see usage metrics

Archive Semester Data – Admins archive past semester sessions for audit without affecting live records.

Priority 3

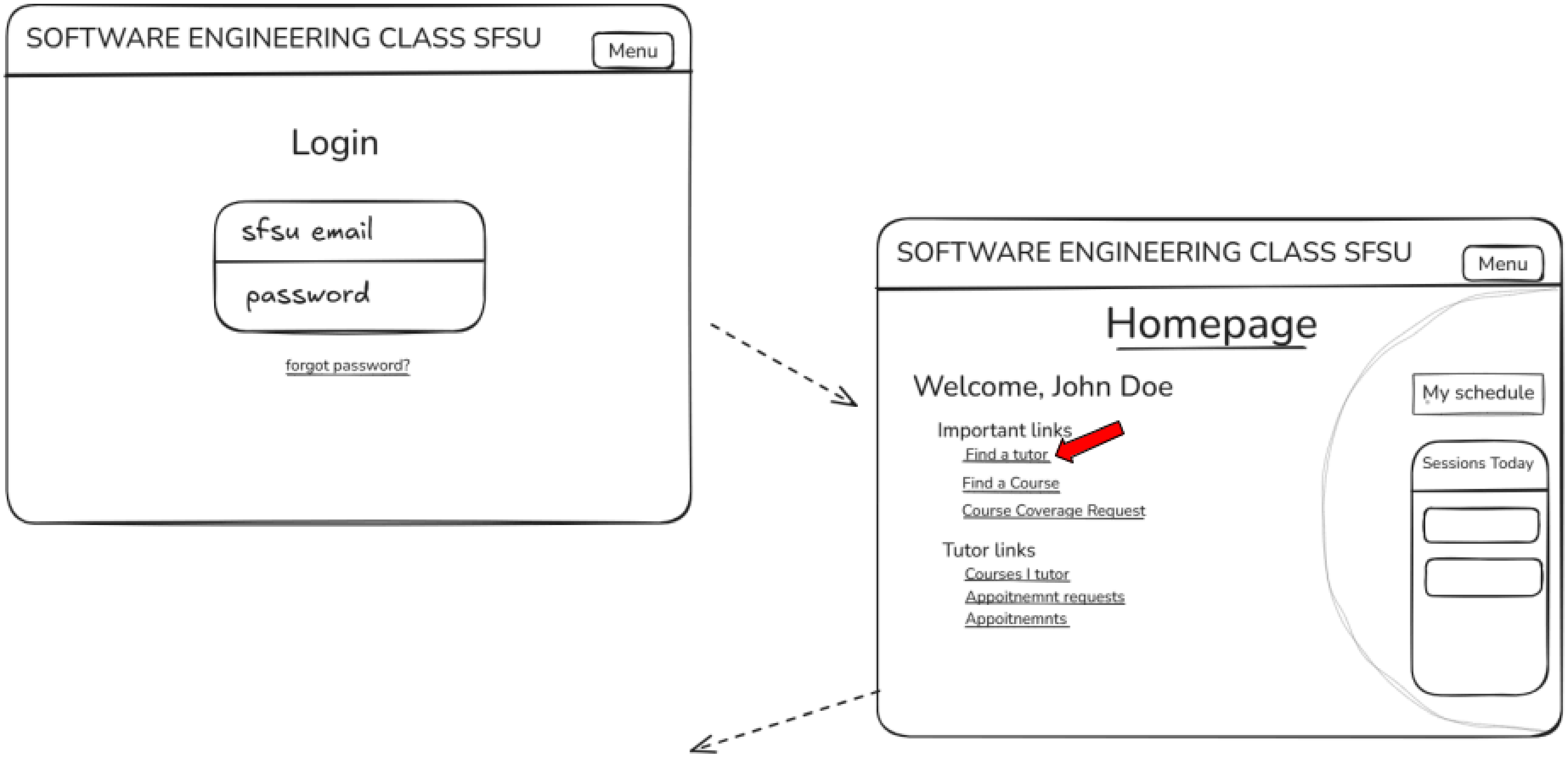
Registered Students

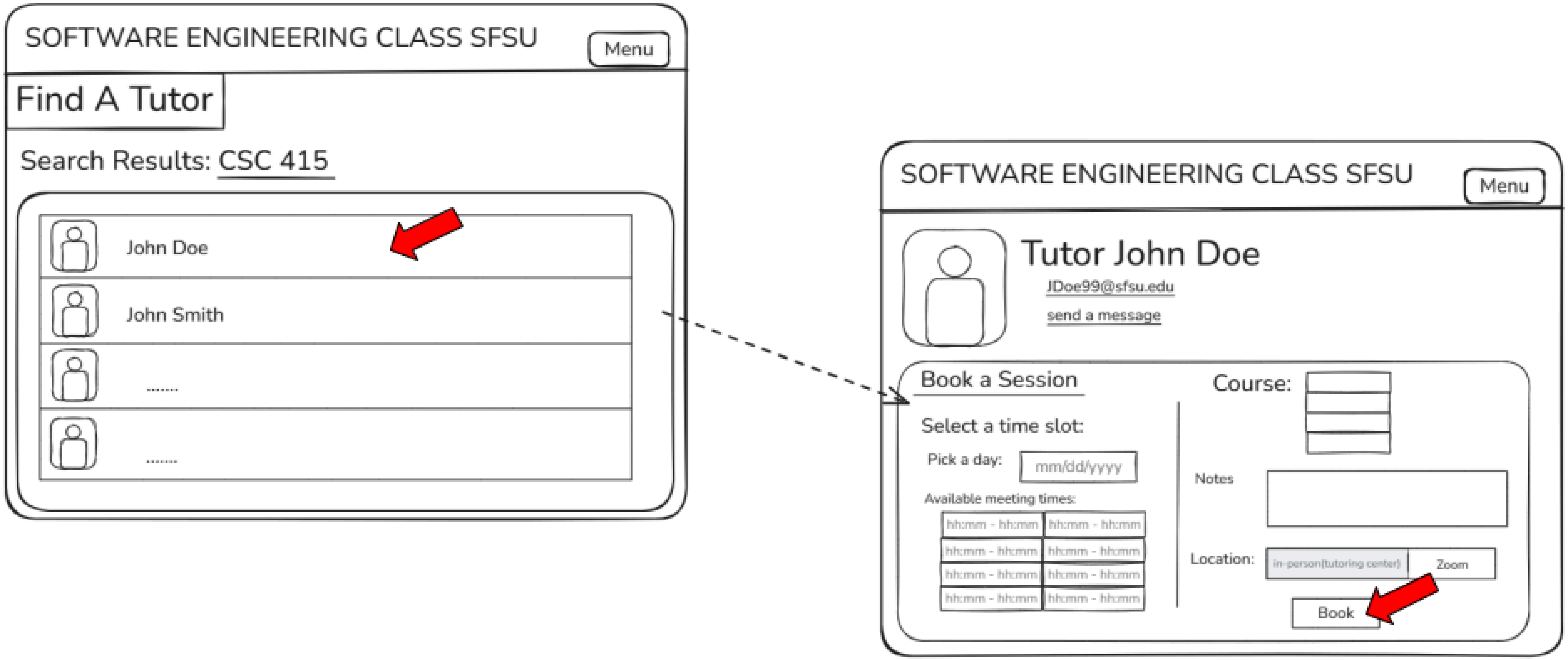
Registered Tutors

Administrators

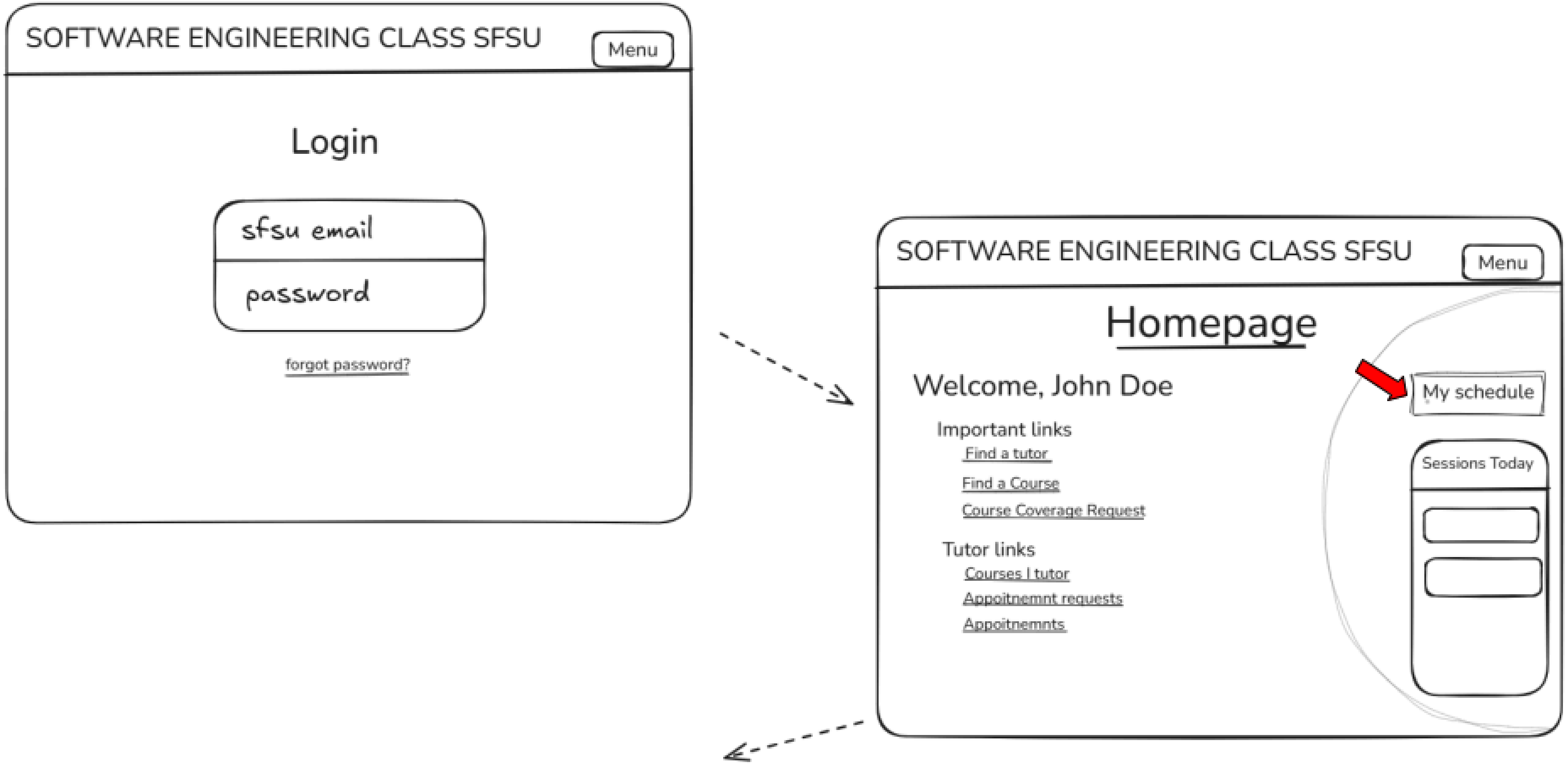
UI Storyboards for each main use case

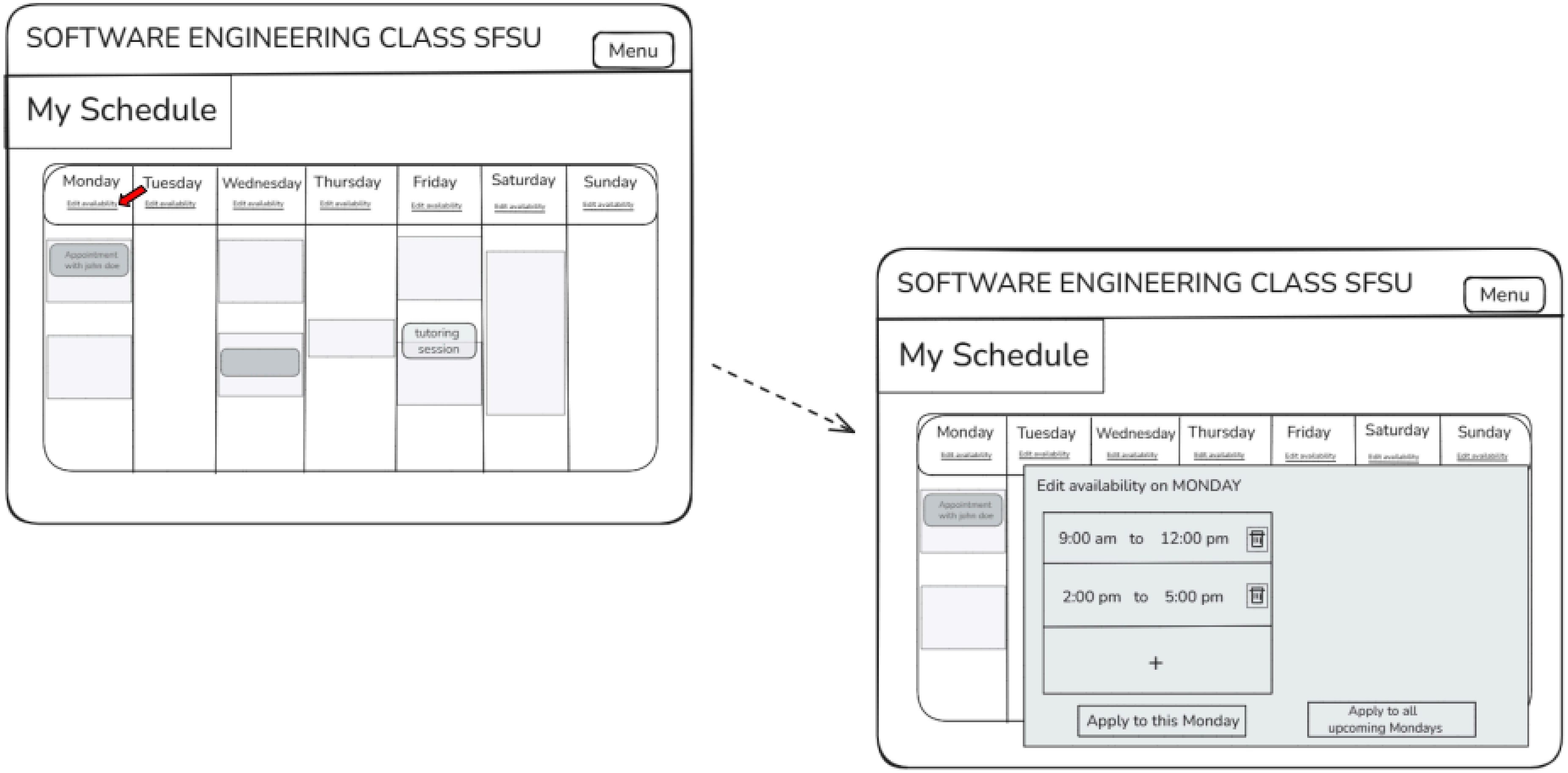
Storyboard 1: Student schedules a tutoring meeting



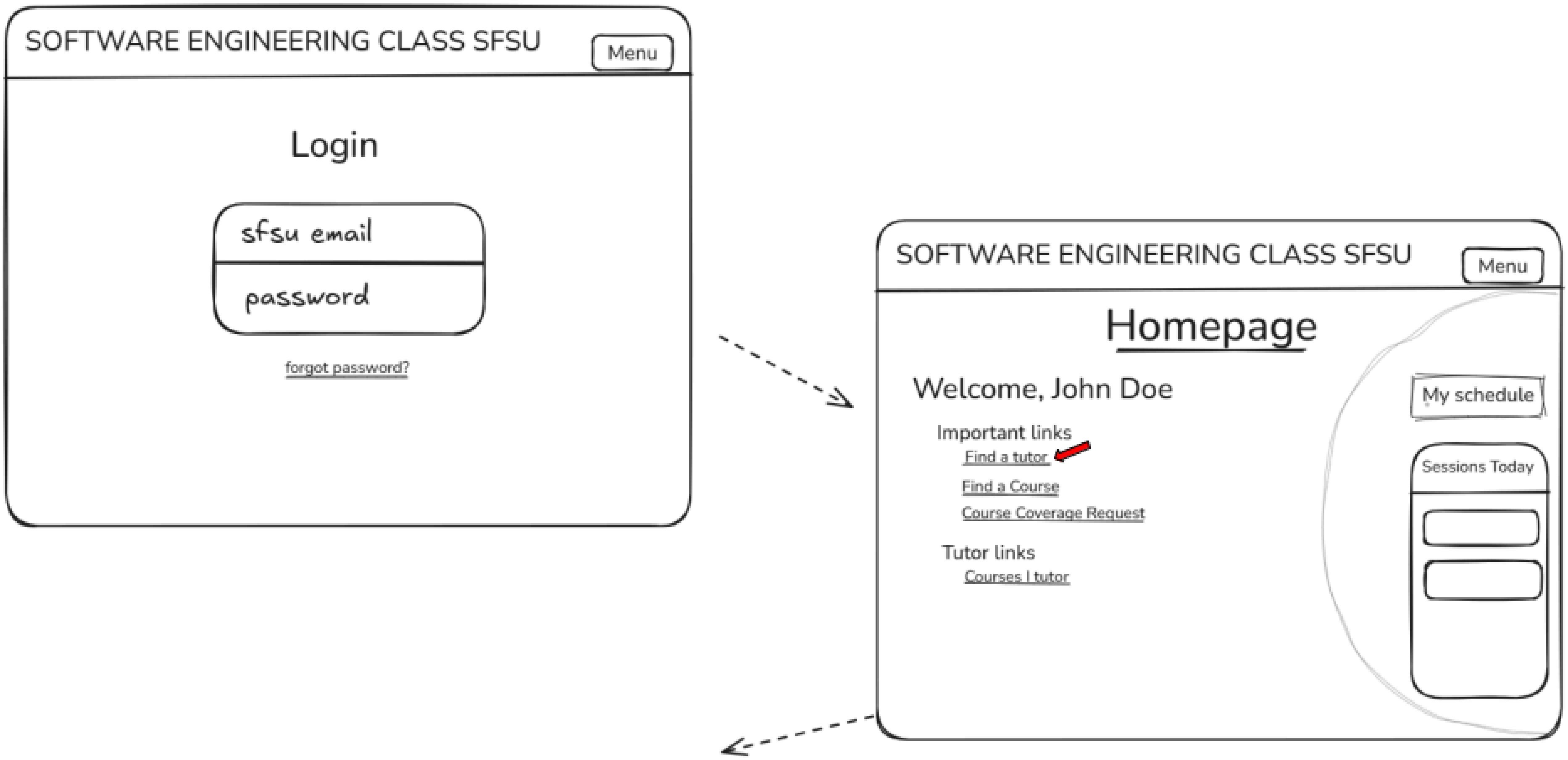

SOFTWARE ENGINEERING CLASS SFSU
Menu
Find A Tutor
Search Results: CSC 415
John Doe
John Smith
....
....
SOFTWARE ENGINEERING CLASS SFSU
Menu
Tutor John Doe
JDoe99@sfsu.edu
send a message
Book a Session
Course:
Select a time slot:
Pick a day:
mm/dd/yyyy
Available meeting times:
hh:mm - hh:mm
hh:mm - hh:mm
hh:mm - hh:mm
hh:mm - hh:mm
hh:mm - hh:mm
hh:mm - hh:mm
hh:mm - hh:mm
hh:mm - hh:mm
Notes
Location:
in-person(tutoring center)
Zoom
Book

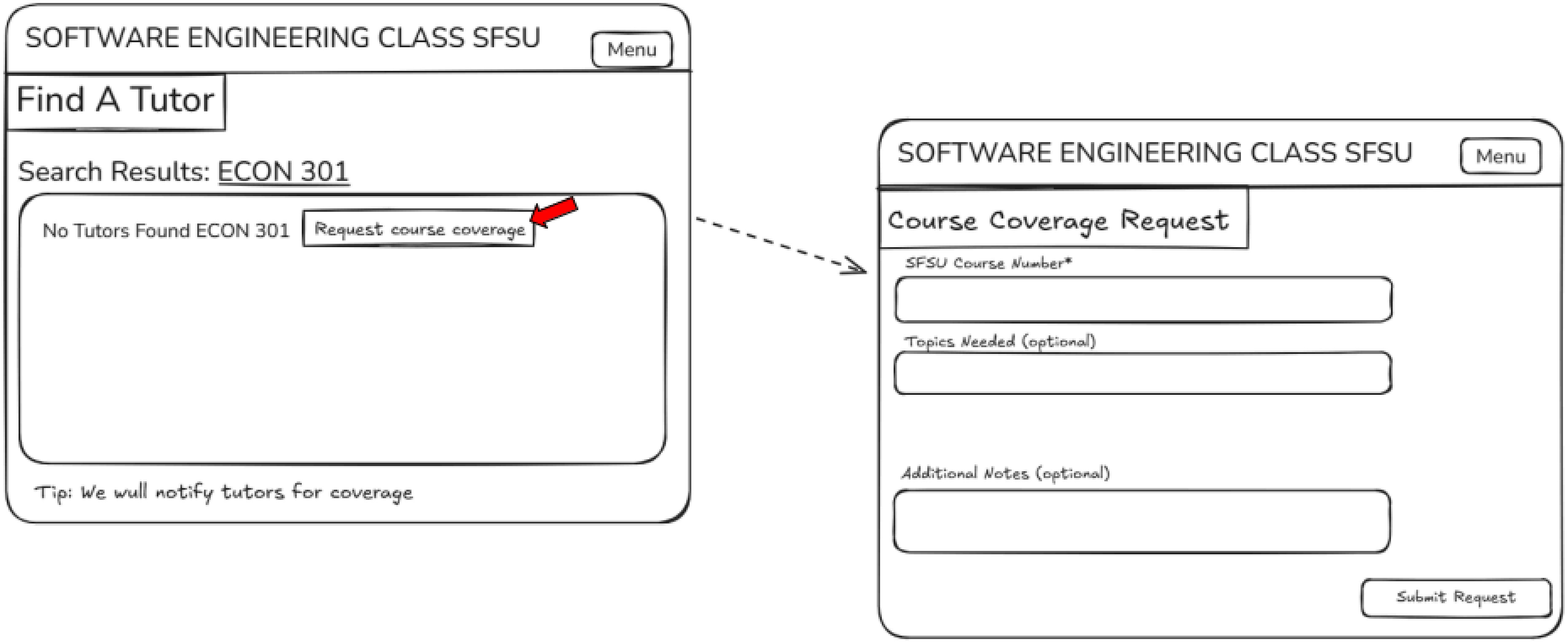
Storyboard 2: Tutor inserts their time availability




SOFTWARE ENGINEERING CLASS SFSU
Menu
My Schedule
Monday
Edit availability
Tuesday
Edit availability
Wednesday
Edit availability
Thursday
Edit availability
Friday
Edit availability
Saturday
Edit availability
Sunday
Edit availability
Appointment with john doe
tutoring session
SOFTWARE ENGINEERING CLASS SFSU
Menu
My Schedule
Monday
Edit availability
Tuesday
Edit availability
Wednesday
Edit availability
Thursday
Edit availability
Friday
Edit availability
Saturday
Edit availability
Sunday
Edit availability
Appointment with john doe
Edit availability on MONDAY
9:00 am to 12:00 pm
2:00 pm to 5:00 pm
+
Apply to this Monday
Apply to all upcoming Mondays

Storyboard 3: Student request coverage for a not offered course





SOFTWARE ENGINEERING CLASS SFSU

Find A Tutor

Search Results: ECON 301

No Tutors Found ECON 301

Tip: We wull notify tutors for coverage

SOFTWARE ENGINEERING CLASS SFSU

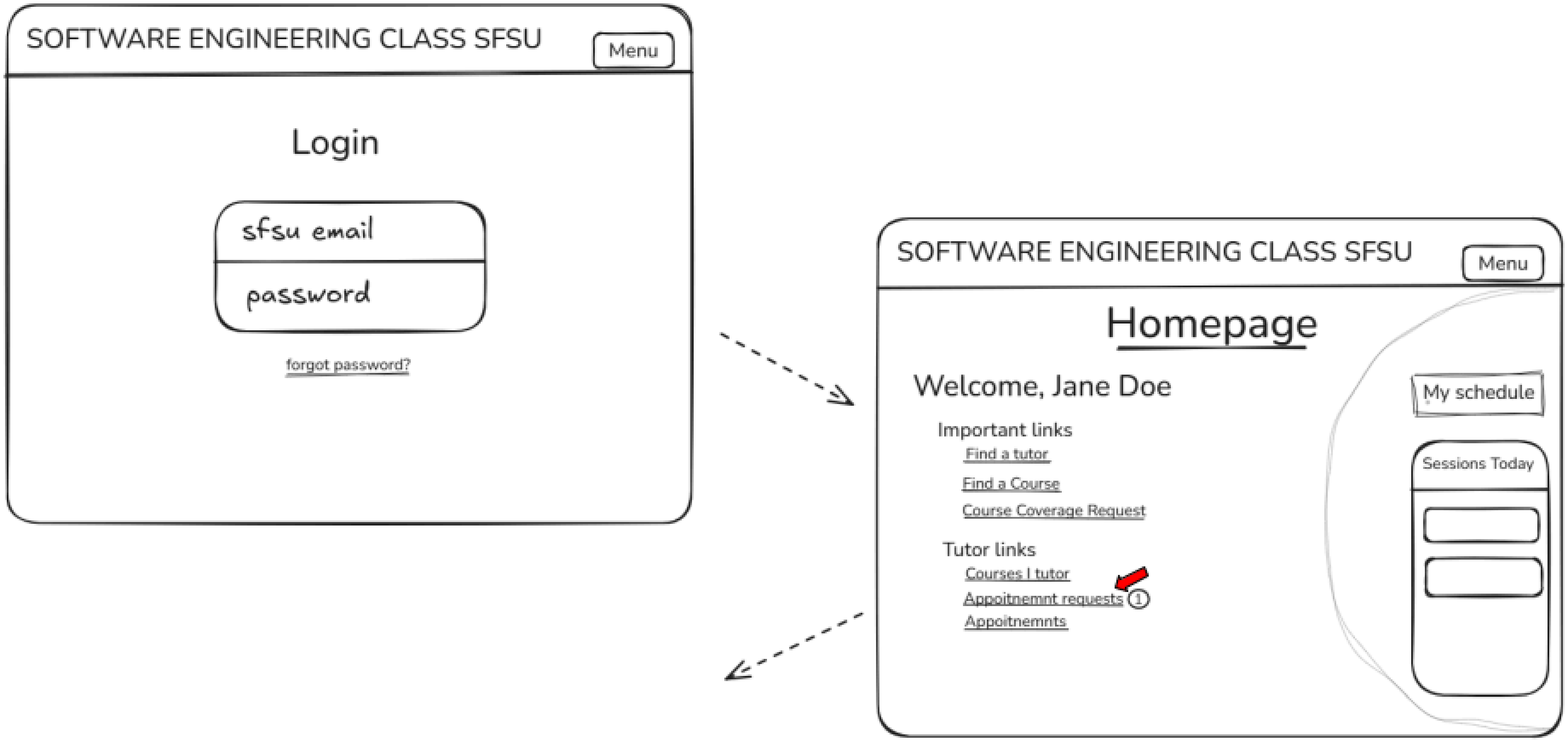
Course Coverage Request

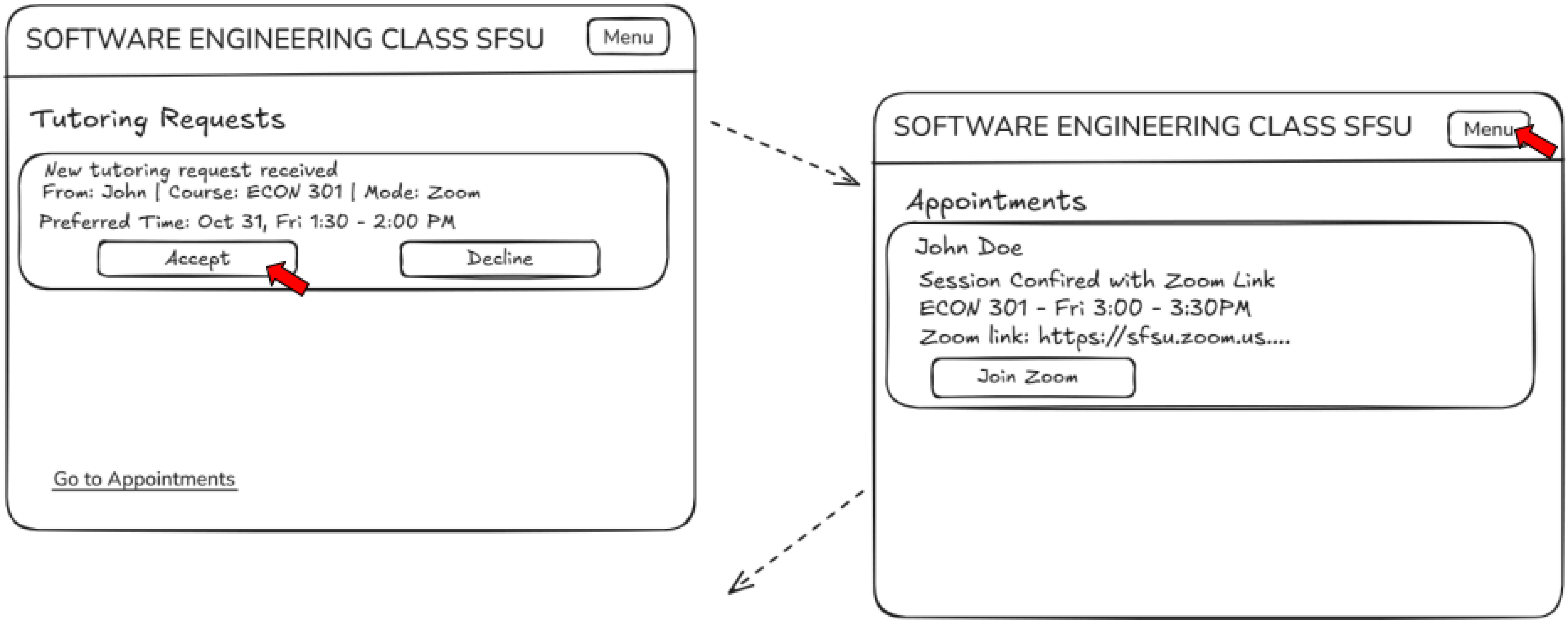
SFSU Course Number*

Topics Needed (optional)

Additional Notes (optional)

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

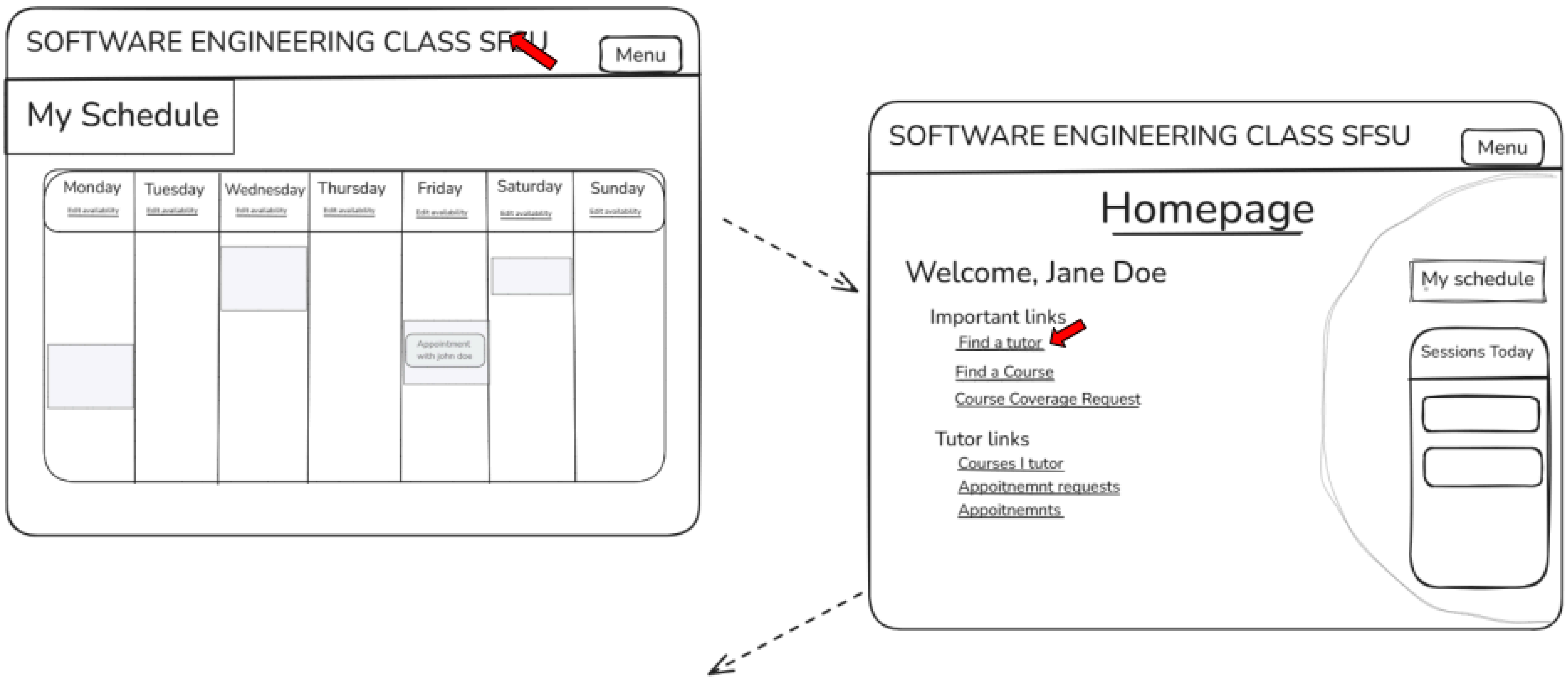
New tutoring request received
From: John | Course: ECON 301 | Mode: Zoom
Preferred Time: Oct 31, Fri 1:30 - 2:00 PM

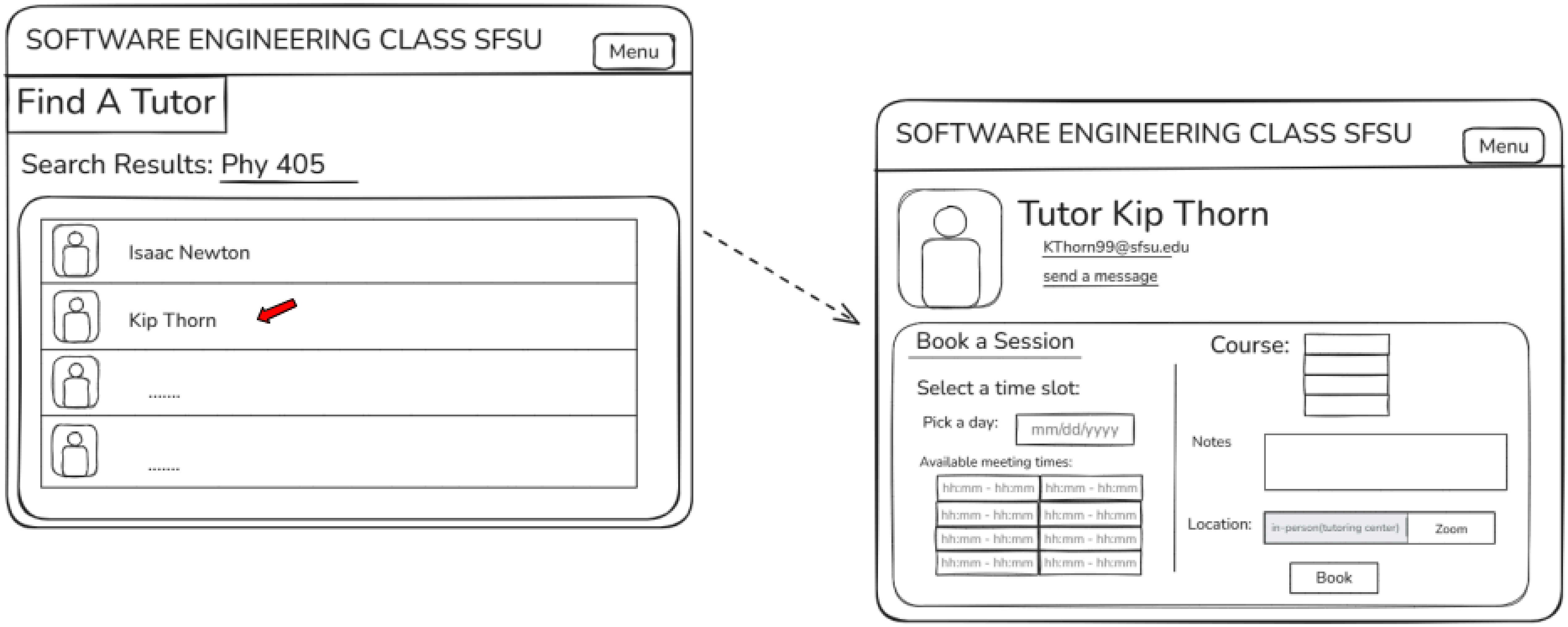
Go to Appointments

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....


SOFTWARE ENGINEERING CLASS SFSU
Menu
My Schedule
Monday
Edit availability
Tuesday
Edit availability
Wednesday
Edit availability
Thursday
Edit availability
Friday
Edit availability
Saturday
Edit availability
Sunday
Edit availability
Appointment with john doe
SOFTWARE ENGINEERING CLASS SFSU
Menu
Homepage
Welcome, Jane Doe
Important links
Find a tutor
Find a Course
Course Coverage Request
Tutor links
Courses I tutor
Appoitnemnt requests
Appoitnemnts
My schedule
Sessions Today


SOFTWARE ENGINEERING CLASS SFSU
Menu
Find A Tutor
Search Results: Phy 405
Isaac Newton
Kip Thorn
....
....
SOFTWARE ENGINEERING CLASS SFSU
Menu
Tutor Kip Thorn
KThorn99@sfsu.edu
send a message
Book a Session
Course:
Select a time slot:
Pick a day:
mm/dd/yyyy
Available meeting times:
hh:mm - hh:mm
hh:mm - hh:mm
hh:mm - hh:mm
hh:mm - hh:mm
hh:mm - hh:mm
hh:mm - hh:mm
hh:mm - hh:mm
hh:mm - hh:mm
Notes
Location:
in-person(tutoring center)
Zoom
Book

High level Architecture, Database Organization summary only

Database Schema:

users :
Columns:
user_id Private 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 Private 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 Private 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,
Private Key(tutor_id, course_id)
Description: Many-to-many join linking tutors to courses they teach.

availability_slots :
Columns:
slot_id Private 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 Private 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.

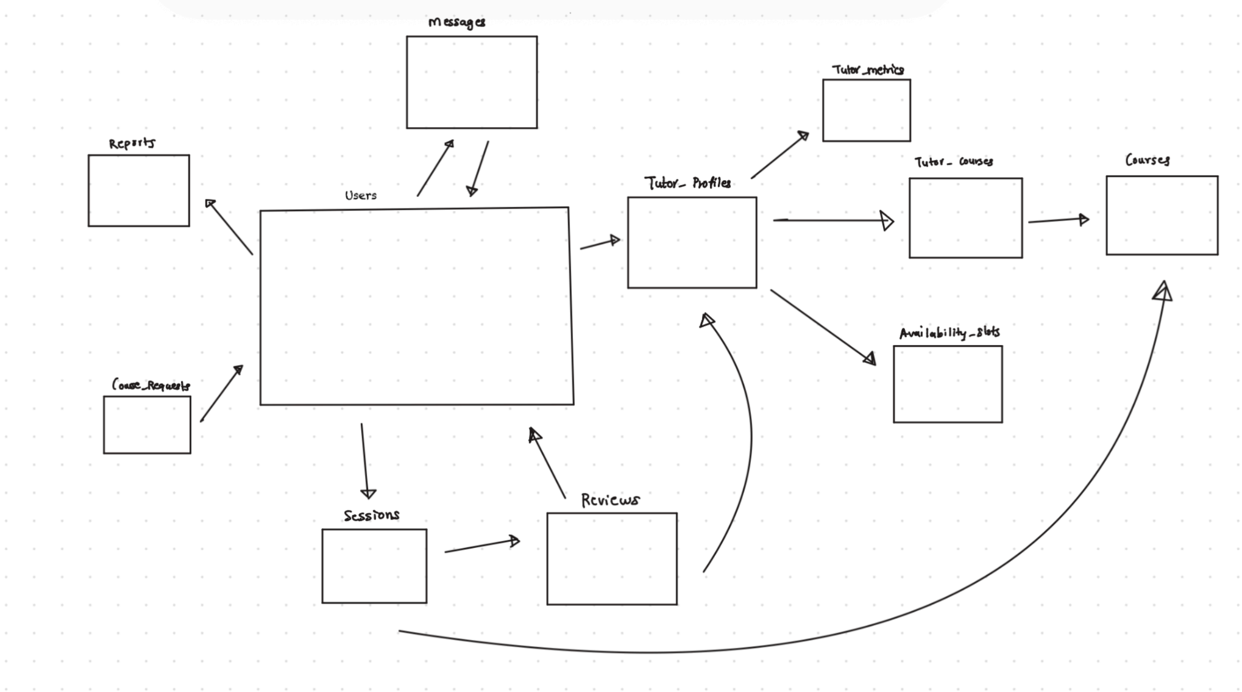
messages :
Columns:
message_id Private Key,
from_user_id Foreign Key, to_user_id Foreign Key, tutor_id Foreign Key,
body, created_at
Description: Private messages between student and tutor for coordination.

reports :
Columns:
report_id Private 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 Private 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 Private 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 Private 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.

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