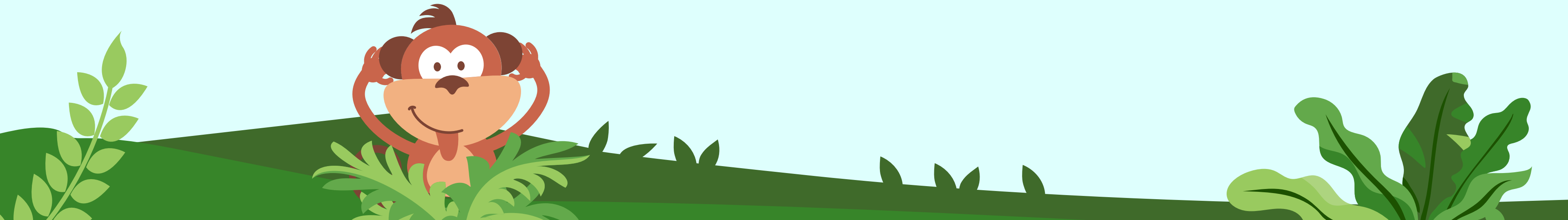
Project Pitch Conductor Tool

Team 2



Statement of Purpose

Goal: Develop course management tool for Software Engineering classes to run smoothly and let students focus on learning rather than logistics

Problem: Professors, TAs, tutors and students use too many disconnected tools. There is no single platform for teamwork, scheduling, and communication

Solution:

- Course management tool for SWE course
- Single-stop tool for professors, TAs, tutors, student leaders and all students
- Integrate group management, scheduling, tutoring support and participation tracking

Platform Technologies

Backend:
- Node.js
- Express.js

Database:
- PostgreSQL (Prisma)

Frontend:
- HTML/CSS
- JS

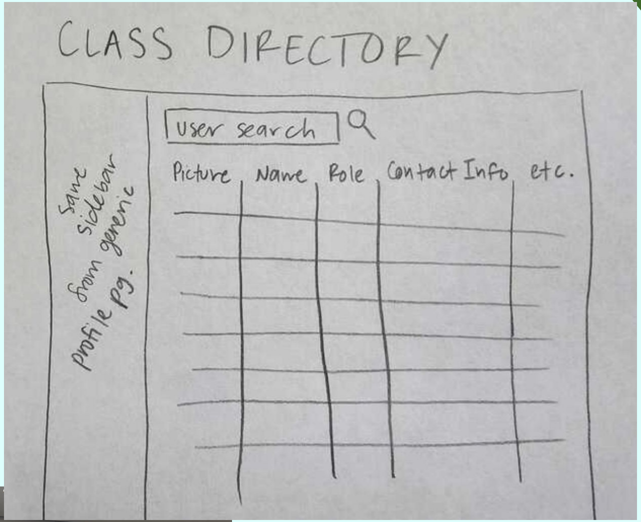
DevOps and Deployment:
- Github
- Oracle VM
- Apache Web Server

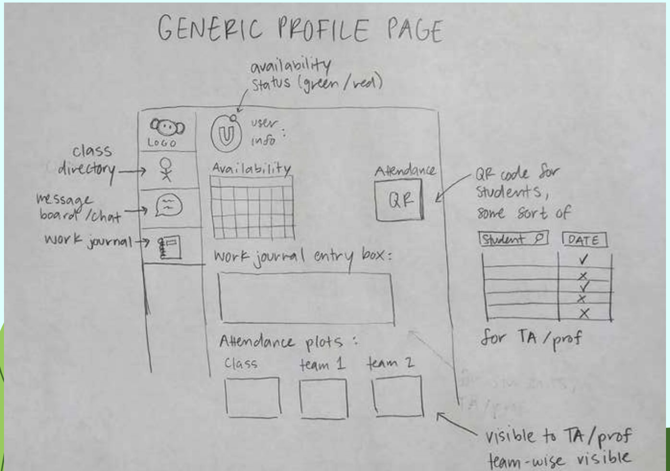
Workflow:

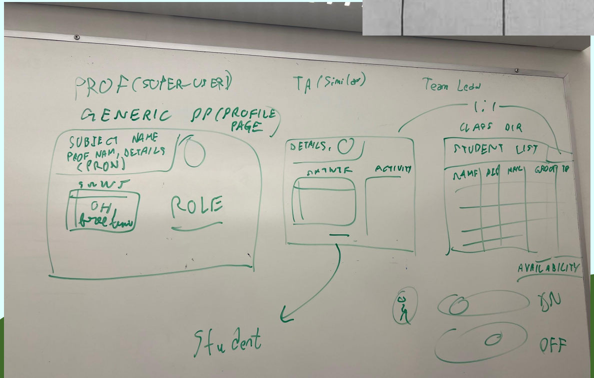
- Frontend and backend teams work in parallel
- Integration after major milestones
- Standardized Practices
  - Commit message conventions
  - Branching strategy
  - Pull request reviews
- Business-Driven Development (BDD) testing approach

Fat-Marker Sketches

- Each user will have a generic profile page that includes:
  - Role (Prof, TA, tutor, lead, student)
  - Calendar/availability
  - Course and Professor details
  - Contact information
- Class Directory
  - Will varies by role
  - Lists student name, id, name, group assignment
- Role-based accessibility







User Personas Summary

Role	Goals	Frustrations	Needs	User Story
Professor	Wants to run large software engineering class smoothly while ensuring fair evaluation and meaningful feedback	Juggles multiple tools (Canvas, Gradescope) to monitor hundreds of students, often losing track of group progress. Manual tracking and fragmented data	Unified directory to manage courses and people, analytics, and evaluation tools	"I spend more time organizing than teaching. I want a simple system where I can focus on teaching rather than logistics"
TA	Keeps teams on track, answer student questions and track progress	Switching between different platforms and heavy administrative duties. Hard to keep track of many students	One central hub for recording attendance, viewing group progress and way to provide evaluations. Clear rubric and grading tools.	"It is hard to stay consistent and help students when progress is hard to track"
Tutor	Answer student questions about course material	No quick way to see student question context or to prioritize students on queue. Too much friction in tutoring logistics.	A lab queue with student details, profile to check and logging of each session. A FAQ section	"I want to be able to focus on helping my students without repeating myself and doing work unrelated to tutoring"
Team Leader	Coordinate team, plans meetings, ensure deadlines are met and ensures team is making progress	Communication lost across tools, inefficient workflow, and hard to track participation + blockers	Shared calendar, meeting notes, team dashboard, and DoD checklist to track team progress	"As team lead, half my effort goes into team planning instead of helping build the project"
Student	Contribute to the project, get recognized for effort in project, and be able to bring up questions and concerns	Feels evaluation is unclear and participation is not always visible to instructors	Dashboard to access course material, journal, feedback tools and grade transparency.	"I want my work to be seen and graded fairly. If I have blockers, I want to be able to communicate to course staff"

Risks and Rabbit Holes

Risks

Limited Time

- Availability and weekly workload vary
- Need clear milestones to stay on track

Unclear Ownership

- Risk of overlapping responsibilities
- Must assign tasks for backend, frontend, and integration

Ambiguous Requirements

- Multiple user roles increase complexity
- Need early agreement on use cases and scope

Rabbit Holes

Too Many Features…

- Temptation to overbuild
- Focus on MVP essentials/core features first

Overinvesting Time

- Risk of polishing “nice-to-have” features before core systems are complete
- Need to be aware of short timeline and to keep making progress

Technical Overreach

- Advanced integration (auth, live queue, analytics) may extend beyond timeline

Core Features

1
Role Based COntrol

- Manages account creation and secure-role based access
- Invite code for users without UCSD email
- Role actions for each level; Professor is super-user

2
Class Directory / Group Form

- List showing all users in course with name, ID, group and role
- Search or filter by group/role
- Option to automatically form groups of selected size or invite to created group

3
Attendance

- QR code for students to scan; generated for unique session and prompt login
- Professor/TAs can search a student and a date and toggle attendance
- % graphs for class and for each team, plotted over time

4
Work Journal / Standup Tool

- Students to log daily progress (current/future) and blockers for team projects; stored in postgres
- Each student has dropdown for past journal entries; can edit
- Team leads can see entries grouped by day from members to monitor progress

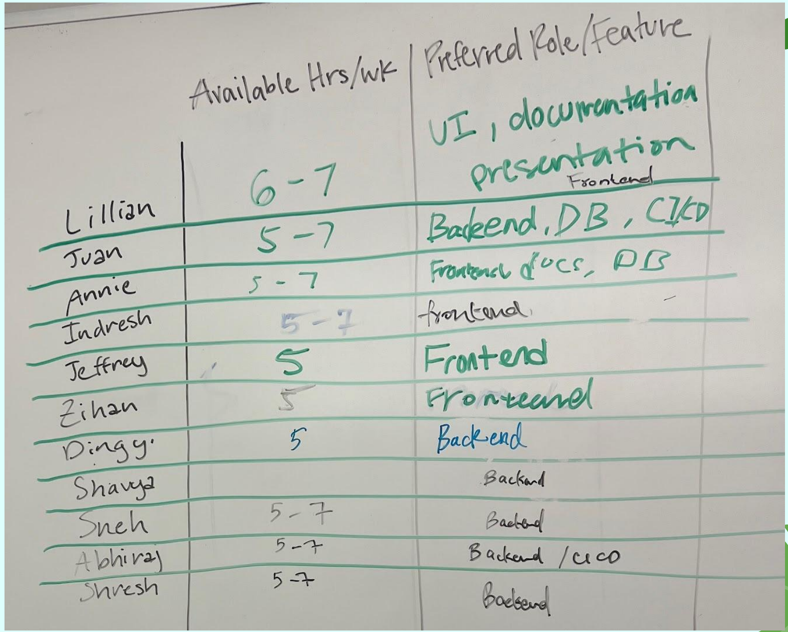
Implementation Plan

Implementation Plan:

- Split into frontend and backend teams based on member preferences
- Weekly integration checkpoints
- Core MVP features completed after 3.5 weeks
- Extended features below if time allows

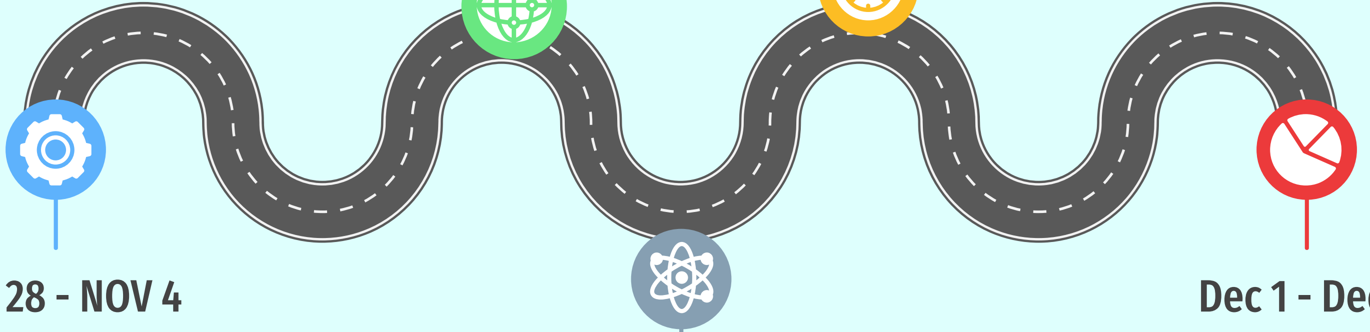
Extended Features:

- Evaluation journal
- Evaluation rubric tool
- Group formation tool
- Participation system to track contributions



	Available Hrs/wk	Preferred Role/Feature
Lillian	6-7	UI, documentation presentation Frontend
Juan	5-7	Backend, DB, CI/CD
Annie	5-7	Frontend docs, DB
Indresh	5-7	frontend
Jeffrey	5	Frontend
Zihan	5	Fronteend
Dingyi	5	Backend
Shavya		Backend
Sneh	5-7	Backend
Abhiraj	5-7	Backend /CICD
Shresh	5-7	Backend

Check our plan: https://github.com/orgs/CSE210-FA25-Team02/projects/1/views/1



OCT 28 - NOV 4

Init repo frontend and backend with CI, prisma linters

Nov 5 to Nov 11

Authentication and authorization features, few features in class directive

Nov 12 - Nov 18

Finishing of features in class directive and starting with journaling module

Nov 19 - Nov 26

Finishing of journaling module and attendance module

Dec 1 - Dec 8

Carry overs, hot-fixes, deployment

For detailed Roadmap: https://github.com/orgs/CSE210-FA25-Team02/projects/1/views/4