CONDUCTOR APP PITCH DOC

Fall 2025

STATEMENT OF PURPOSE

Conductor App

(1) Improves the management of large-scale **Software Engineering courses at UCSD** by...

- Integrating administrative automation
- Hosting structured data capture
- Defining team analytics into a single accessible platform

(2) Addresses coordination challenges faced by Professors, TAs, and students in software engineering courses

- Many students with repetitive tasks (group formation, attendance, grading, communication, and more)

(3) Provides real-time insights & fair evaluation mechanisms

- Instructors can focus more on *teaching and mentoring*
- Students can receive transparent feedback + voice in their team experience

It embodies **modularity and accessibility** to serve as a sustainable teaching tool for future SE courses

USER PERSONAS

Persona 1: Professor

Goals: Have an easy & comprehensive way to keep track of teams + individuals at a higher level, to improve the ability to teach students.

Persona 2: TA

Goals: Be able to assign grades and track interactions with various teams and individuals easily and efficiently at a more granular level.

Persona 3: Tutor

Goals: Have easy access to students who need help & be able to effectively assist them with their assignments

Persona 4: Team Leader

Goals: Have a central space to communicate & run the team, creating more time for leadership and coordination.

Persona 5: Student

Goals: Receive feedback from leaders & staff on coursework + interactions, creating a central place to practice software engineering skills.

FUNCTIONAL REQS

1) User & Access Management

Goal: Ensure secure, role-based participation in large SE classes

2) Collaboration & Tracking Tools

Goal: Record team coordination & participation

3) Evaluation, Feedback & Support

Goal: Ensure fair, consistent grading & responsive student support

4) Infrastructure & Integrations

Goal: Provide reliable, extensible backend for the whole system

NON-FUNCTIONAL REQS

1) Performance, Reliability & Security

Goal: Keep the system fast, stable, & safe for sensitive academic data.

2) Usability, Accessibility & Internationalization

Goal: Make Conductor inclusive, easy to use, and globally accessible

3) Maintainability, Extensibility & Deployment

Goal: Ensure future teaching staff can evolve and maintain the system easily

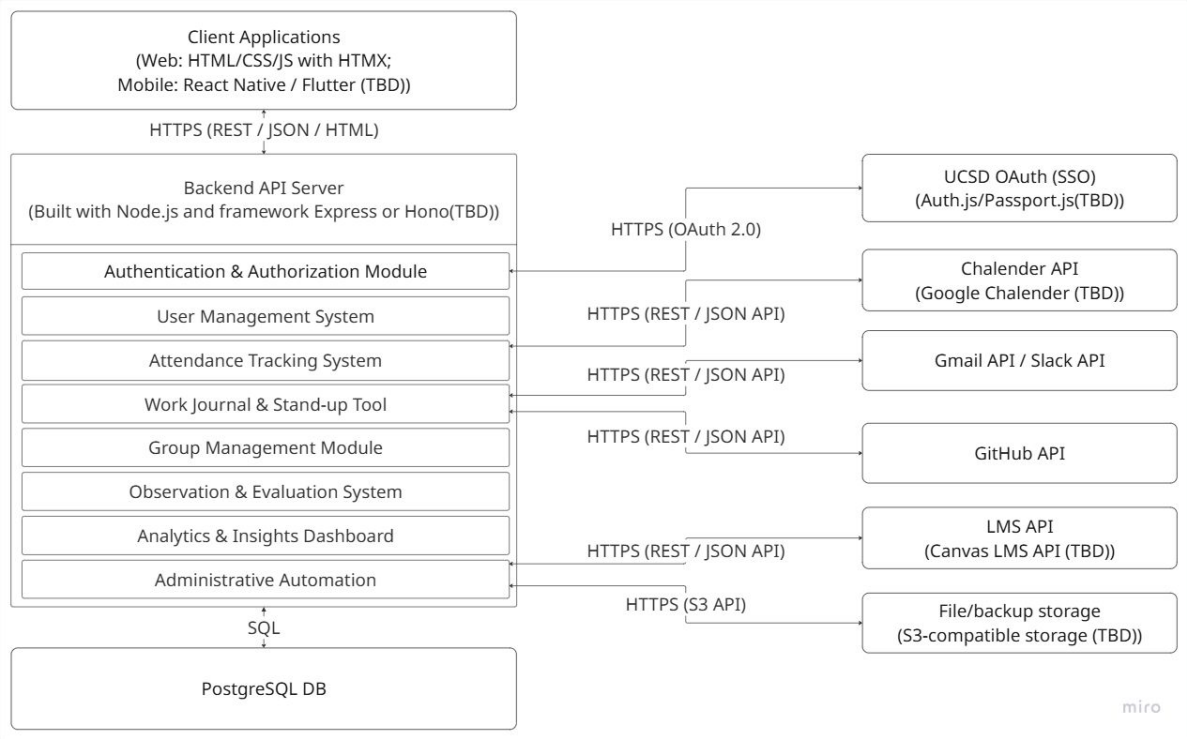
RISKS

- **One thing which seems as a risk** is that student behaviour is unpredictable. Tools that rely on consistent input (daily check-ins, self-reported effort) often see sharp drop-off after Week 2. If the data dries up, the insights become noise.
- **This tool is for one class,** but building it like a generic app to use by all professors (multi-classes, configurable roles, audit logs) will add huge overhead. Also there should be course enrollment step necessary. If student add/drops after few weeks, there must be way to add/remove.
- **Even well-designed tools fail** if students treat them as extra work rather than value-add. There is a huge over-reliance on daily journals and effort logs, students tend to have a slightly painful experience in self-reporting given the course workload and those are as honest as the student's motivation that day.
- **Sentiment or "participation" scores** can unintentionally penalize introverted, non-native, or neurodivergent students.

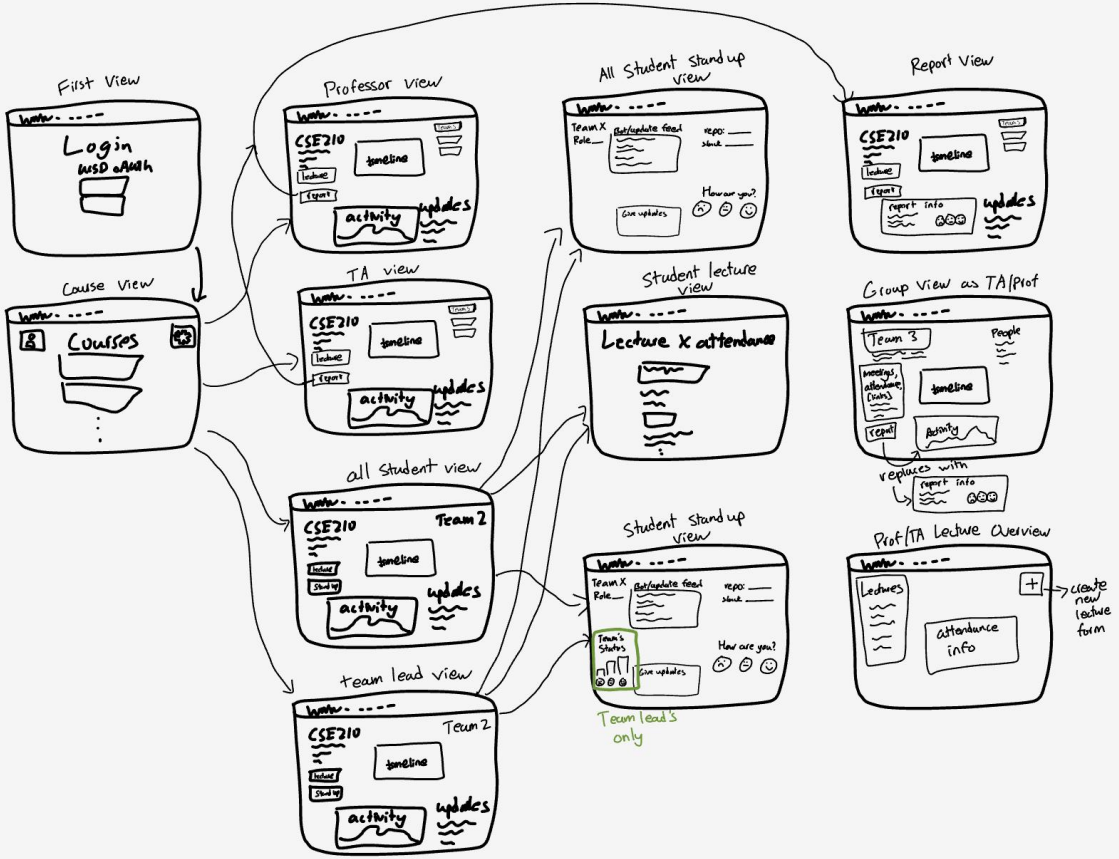
RABBIT HOLES

- Don’t start implementing AI everywhere, starting from scoring journals and exploring patterns among students.
- Avoid integrating with google calendar; a simple shared availability grid is sufficient and it will respect privacy.
- Support translation-ready markup, but don’t build a full i18n pipeline upfront.
- A simple informative figures for different trends is fine; avoid super complex and animated dashboards or trend forecasting.
- Biggest rabbit hole will be to focus on polishing UI instead of bugs. For these apps, UX is much important for UI.
- Prioritize quality compared to quantity for each feature to ensure the application can be run by different developers (such as the TA or Professor).

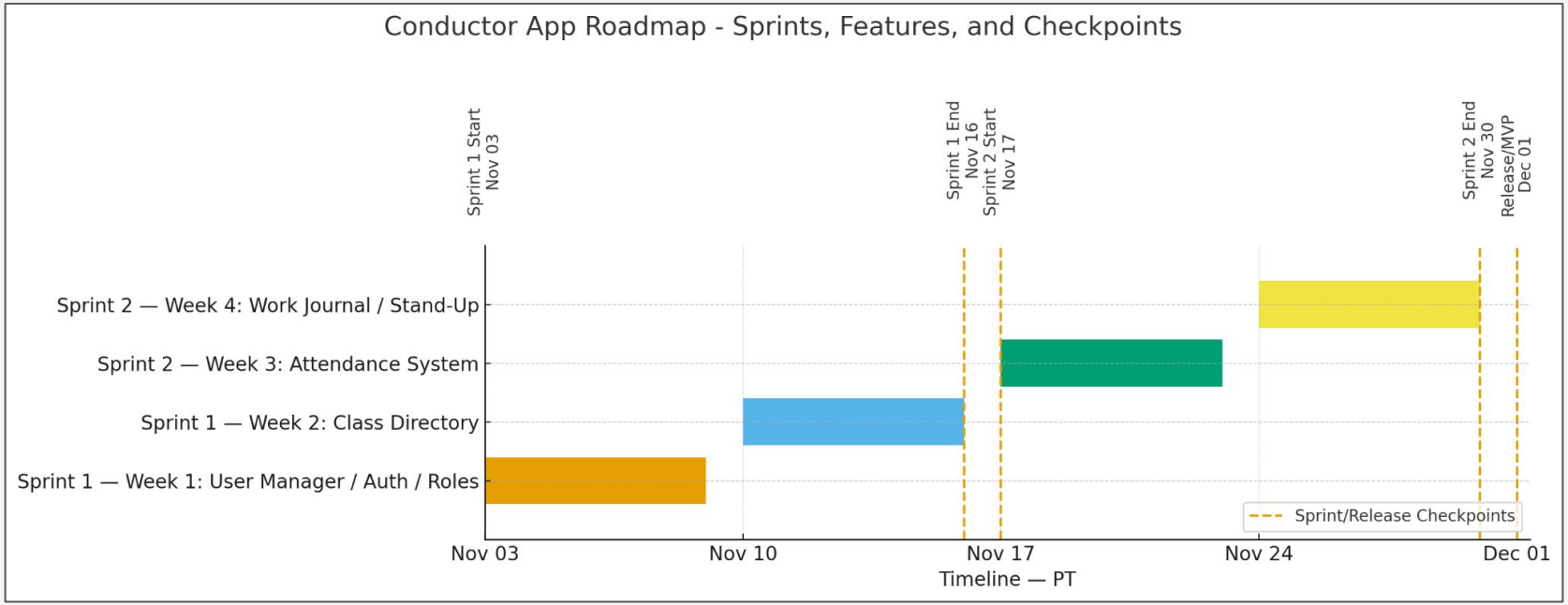
VISUAL


Client Applications
(Web: HTML/CSS/JS with HTMX;
Mobile: React Native / Flutter (TBD))
HTTPS (REST / JSON / HTML)
Backend API Server
(Built with Node.js and framework Express or Hono(TBD))
Authentication & Authorization Module
User Management System
Attendance Tracking System
Work Journal & Stand-up Tool
Group Management Module
Observation & Evaluation System
Analytics & Insights Dashboard
Administrative Automation
SQL
PostgreSQL DB
HTTPS (OAuth 2.0)
UCSD OAuth (SSO)
(Auth.js/Passport.js(TBD))
HTTPS (REST / JSON API)
Chalender API
(Google Chalender (TBD))
HTTPS (REST / JSON API)
Gmail API / Slack API
HTTPS (REST / JSON API)
GitHub API
HTTPS (REST / JSON API)
LMS API
(Canvas LMS API (TBD))
HTTPS (S3 API)
File/backup storage
(S3-compatible storage (TBD))
miro

VISUAL


First View
Login
wsu oAuth
Professor view
CSE210
timeline
lecture
report
activity
updates
All Student Stand up view
Team X
Role
Report View
report info
Course View
Courses
TA view
Student lecture view
Lecture x attendance
Group View as TA/Prof
Team 3
People
meetings, attendance, [team]
replaces with
all Student view
Team 2
study
Student Stand up view
repo:
How are you?
Give updates
Team's Status
Team lead's only
team lead view
Prof/TA Lecture Overview
Lectures
attendance info
create new lecture form

VISUAL


Conductor App Roadmap - Sprints, Features, and Checkpoints
Sprint 1 Start Nov 03
Sprint 1 End Nov 16
Sprint 2 Start Nov 17
Sprint 2 End Nov 30
Release/MVP Dec 01
Sprint 2 — Week 4: Work Journal / Stand-Up
Sprint 2 — Week 3: Attendance System
Sprint 1 — Week 2: Class Directory
Sprint 1 — Week 1: User Manager / Auth / Roles
Sprint/Release Checkpoints
Nov 03
Nov 10
Nov 17
Nov 24
Dec 01
Timeline — PT

THANK YOU

For your attention