The Problem: Teaching Staff Are Drowning in Administrative Chaos

It's Wednesday afternoon. Dr. X is staring at her screen, cycling through five different tools just to understand how her class is doing. **It will take her 90 minutes.**

> **90 min**
> EVERY WEEK, JUST TO CHECK CLASS HEALTH

Canvas Roster | GitHub Activity | Google Sheets | Slack Messages | Email Threads

She's looking for Sarah Martinez. Attendance is dropping. Commits are slowing. Journal entries stopped. But by the time Dr. X pieces these signals together from five disconnected systems, **it's week 8. Too late.**

The Scale Challenge

Managing at Scale

500+ students across 50 project teams. Limited TA hours. Every student deserves to be seen, but the tools make it impossible.

- TAs spend 60% of time on admin, not mentoring
- Students slip through cracks until crisis point
- Fair grading impossible without data
- No early warning system for at-risk students

The Hidden Costs

Current systems fail everyone involved:

- Professors can't see individual students
- TAs grade based on incomplete information
- Team Leaders drown in coordination
- Quiet contributors become invisible

> *"Every student deserves to be seen, even in a class of 500. Technology should help with that, not make it harder."*
>
> — Dr. X, Software Engineering Professor

Our Appetite

5 weeks. Four core features. A system that transforms teaching at scale from impossible to manageable. This is a small-batch, high-impact project scoped to deliver a working MVP by December 5, 2024.

The Pattern: Good People Failing Because of Bad Tools

By Week 8:

- Alex has graded based on incomplete information, feeling guilty about unfairness
- Dr. X has discovered three at-risk students, too late to help effectively
- Marcus is burned out, considering stepping down from team leadership
- Aisha's contributions are invisible, her grade doesn't reflect her work

The Human Cost: Who This Affects

Four people. Four different pain points. One broken system.

Alex Chen

The TA (Group Mentor)

Pain: "I have 10 teams. Notes scattered everywhere. Can't track who's falling behind until it's too late."

Needs: Quick meeting prep. Fair grading data. Early warning signals. Structured note-taking that doesn't take forever.

Dr. X

The Professor

Pain: 90 minutes every Wednesday cycling through tools. Almost missed Sarah's crisis. Spent 10 hours manually forming teams.

Needs: Holistic student view. Early intervention. Sustainable tools that survive vendor changes. One place to see everything.

Marcus Johnson

The Team Leader

Pain: "I spend more time coordinating than leading. 11pm and I still haven't written code. Can't document team dynamics anywhere."

Needs: Team dashboard. Availability tracking. Private channel to flag concerns. Time to actually lead, not chase people down.

Aisha Patel

The Student

Pain: "40 commits. Most of the backend. But I'm quiet in meetings. Will anyone know I contributed? No safe place for feedback."

Needs: Recognition for actual work. Clear evaluation criteria. Private feedback channel. Transparent participation tracking.

The Solution: One Unified Platform

Replace the chaos with clarity. Automate the administrative burden so teaching staff can focus on what matters: teaching and mentoring.

> **90 min → 5 min**
> TIME TO UNDERSTAND CLASS HEALTH

Four Integrated Modules

Authentication & User Management

Secure UCSD-based login with role-based permissions. Instructors manage users, assign roles, and configure access for 500+ students.

Class Directory

Centralized hub for everyone in the course. Searchable roster, detailed profiles, group pages. The "phone book" for the entire class.

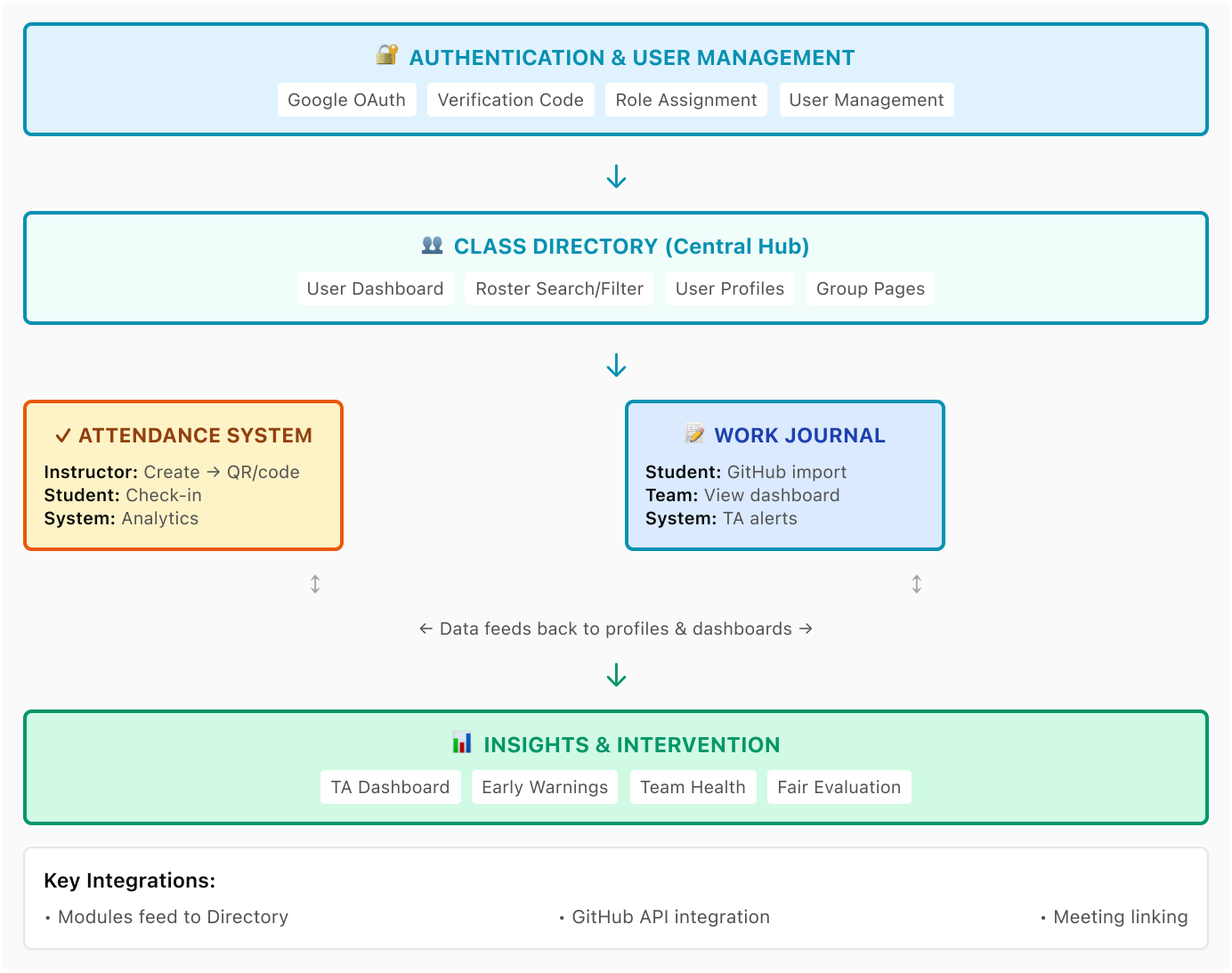
Attendance System

Quick-entry tracking for lectures, meetings, and office hours. QR code check-ins. Visual analytics. Takes <60 seconds for 200+ students.

Work Journal / Standup

Daily 2-minute check-ins with GitHub auto-population. Team dashboards for transparency. Automatic blocker escalation and early warnings.

How The System Flows



Core Principle

Automate administrative tasks. Amplify teaching impact.

By handling the busywork automatically, Conductor Tool gives teaching staff time to focus on mentoring students and helping teams succeed.

The Transformation

Before: Scattered Chaos

- 5+ tools to check class status
- Manual attendance spreadsheets
- Notes lost in text files
- Problems discovered week 8
- Grading based on gut feeling

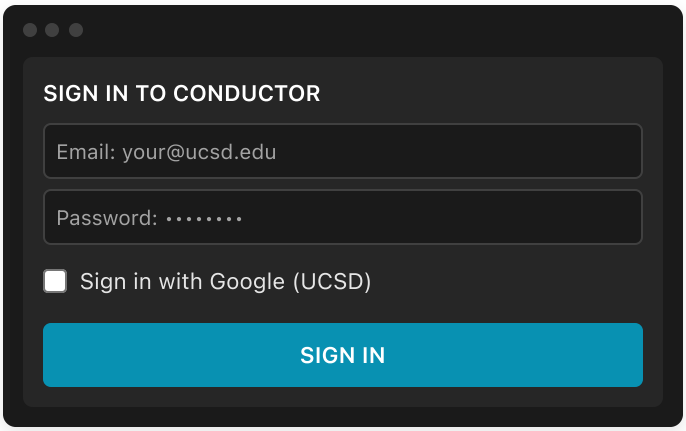
After: Unified Clarity

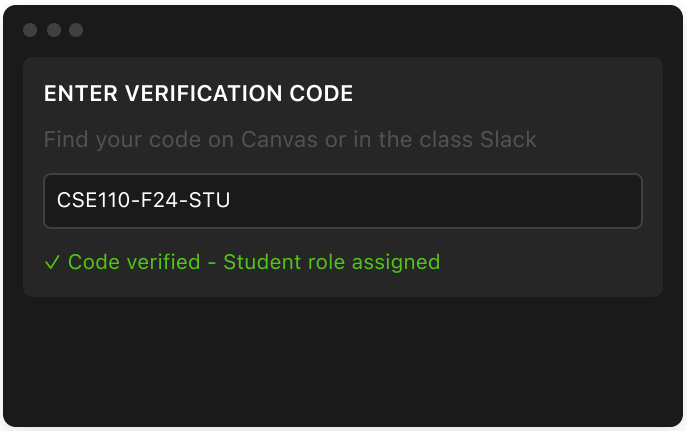
- One platform, complete picture
- Attendance in <1 minute
- Structured, searchable notes
- Early warnings week 3
- Fair, data-driven evaluation

Feature 1: Authentication & User Management

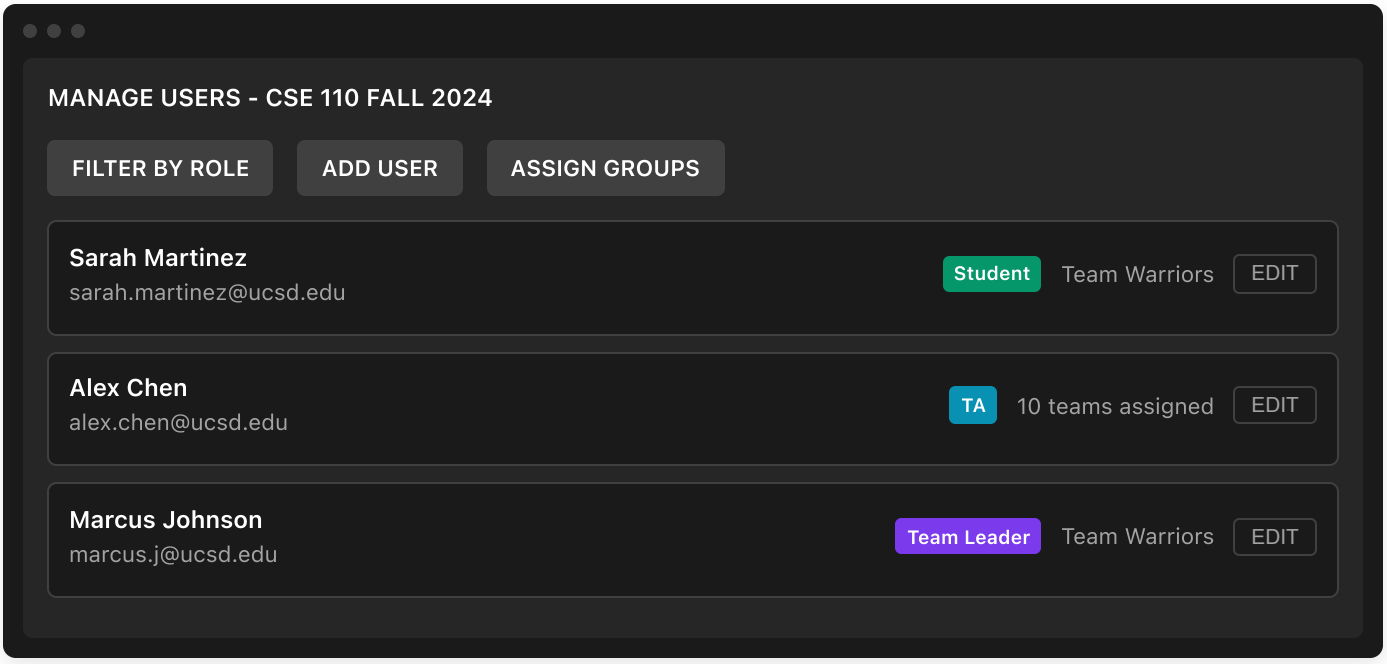
Secure, role-based access that handles 500+ students including UCSD extension students. Built for administrators to easily manage users and permissions.

Sign-In Flow





User Role Management



Key Features

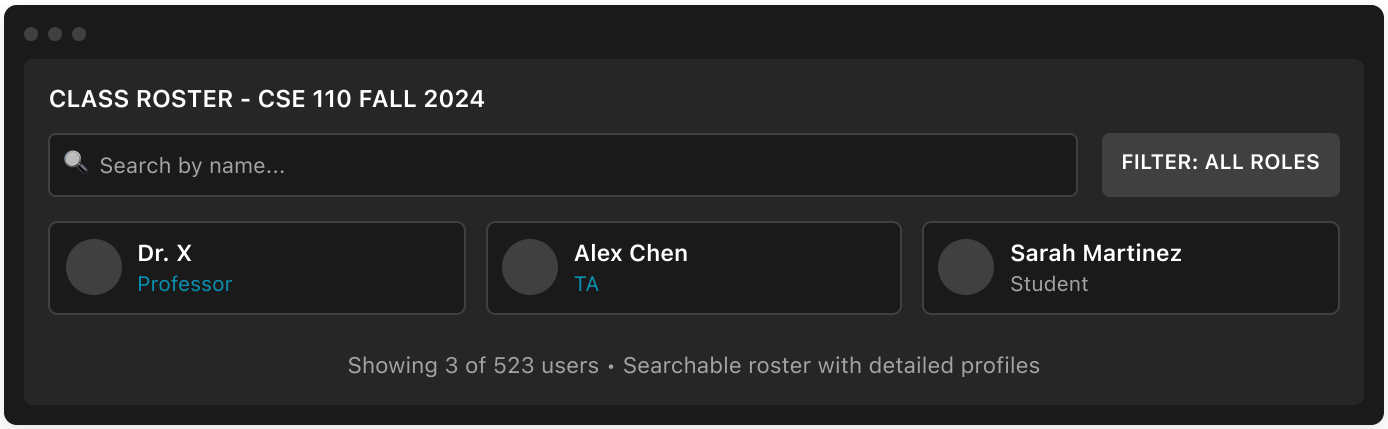
- UCSD Google OAuth integration
- Verification code system for role assignment
- Extension student request workflow
- 6-tier role hierarchy (Admin → Student)
- Bulk user management for 500+ students

Security & Compliance

- Role-based access control (RBAC)
- FERPA-compliant data handling
- Audit logging for user actions
- Encryption at rest in database
- Admin-only password resets

Feature 2: Class Directory

The centralized "phone book" for the entire course. Searchable roster, detailed profiles, and group pages with all relevant information in one place.



Directory Features

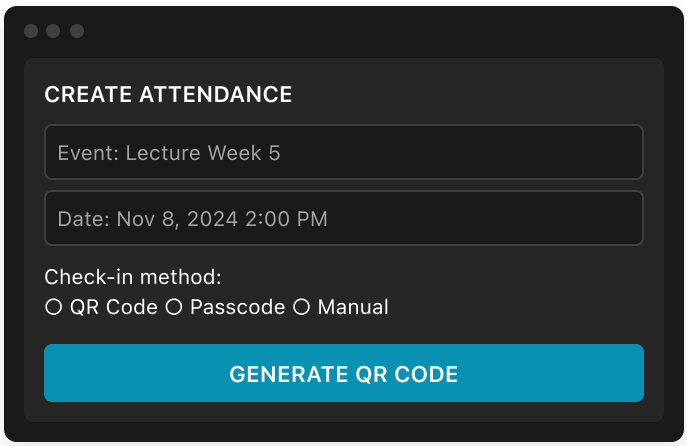
- Searchable roster with advanced filtering
- Detailed user profiles with pronouns, availability
- Group pages with team info and links
- Activity tracking integration

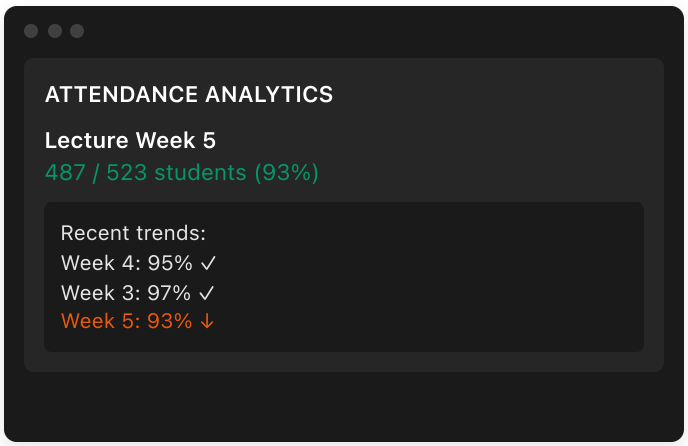
Benefits

- One place to find anyone in the class
- Transparent team structures
- Easy coordination across 500+ students
- Replaces scattered contact lists

Feature 3: Attendance System

Quick-entry tracking for lectures, meetings, and office hours. QR code check-ins for students. Visual analytics and trends.





Key Features

- QR code generation for quick check-ins
- Manual entry for TAs and professors
- Tracks lectures, meetings, office hours
- Visual analytics and trend tracking

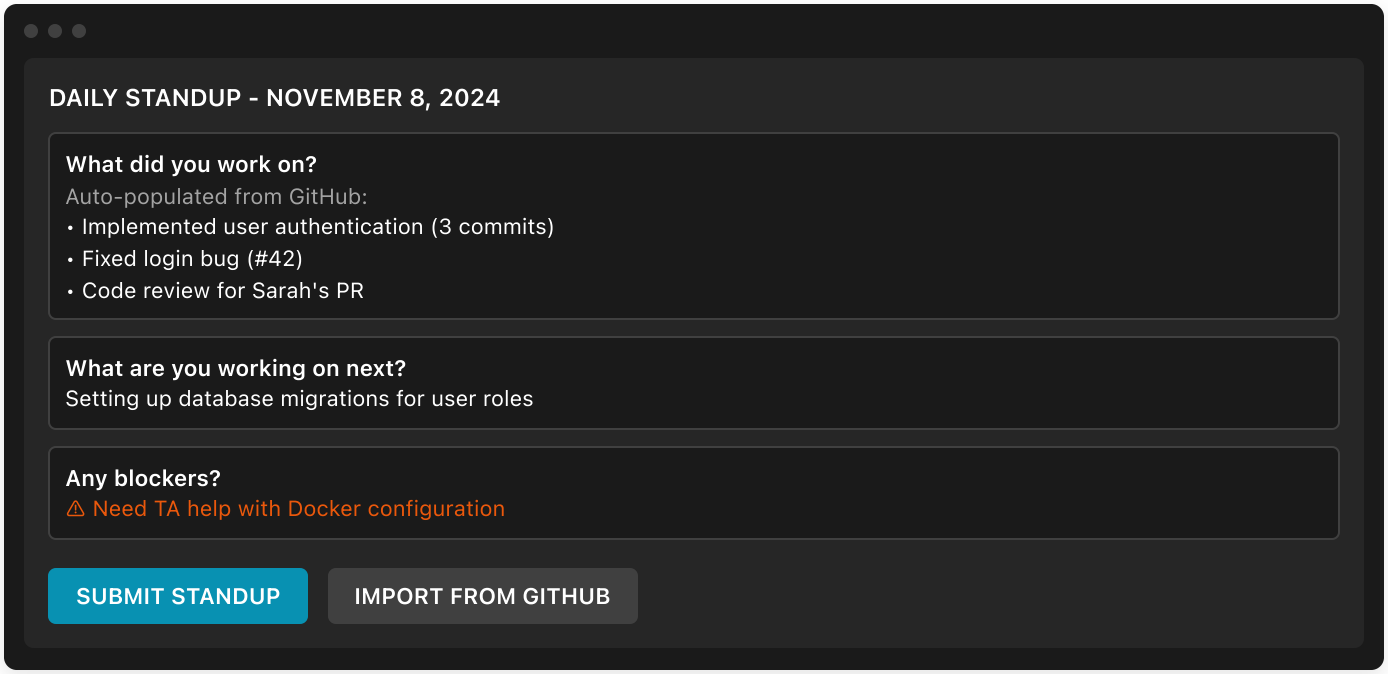
Time Savings

- <60 seconds to record 200+ students
- Replaces manual spreadsheets
- Automatic alerts for patterns
- Integration with student profiles

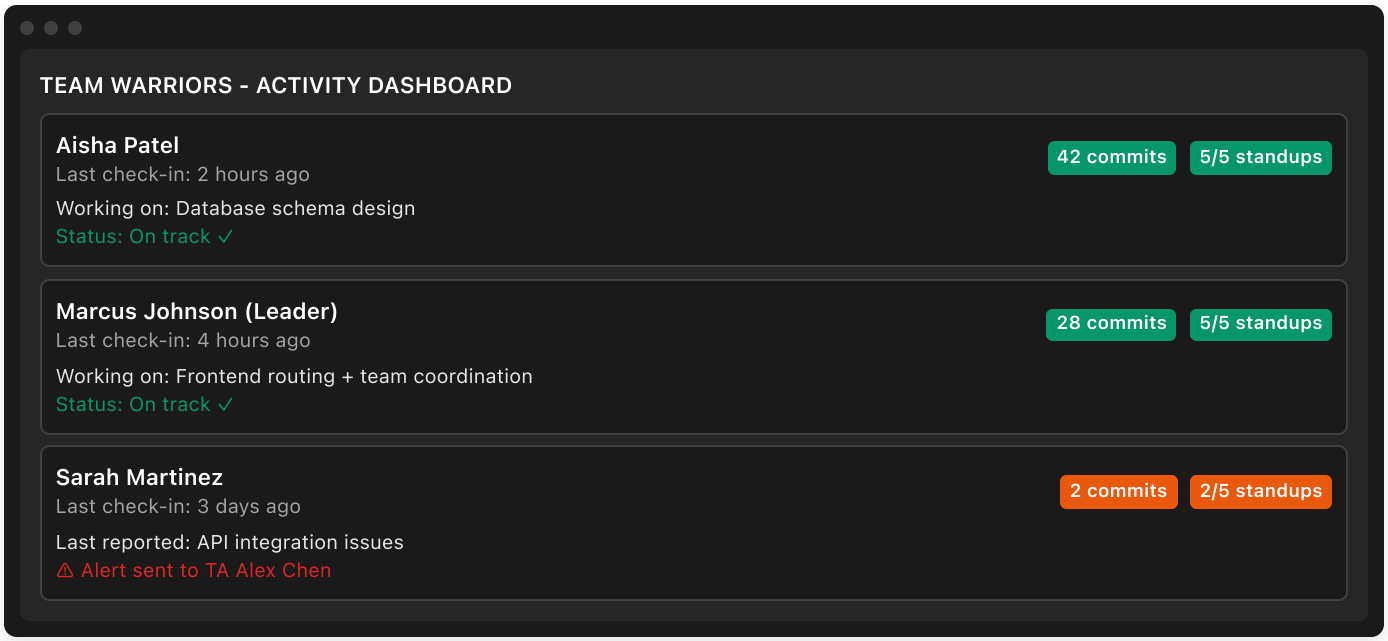
Feature 4: Work Journal / Standup

Daily 2-minute check-ins with GitHub auto-population. Team dashboards for transparency. Automatic blocker escalation. Early warning system for at-risk students.

Student Journal Entry



Team Dashboard View



Key Features

- 2-minute daily check-ins
- GitHub activity auto-population
- Team transparency dashboards
- Blocker escalation to TAs
- Early warning system

Impact

- Catch at-risk students by week 3
- Fair evaluation of contributions
- Team coordination simplified
- TAs get structured meeting prep
- Quiet contributors become visible

Technical Approach

Technology Stack

React	Node.js	PostgreSQL	GitHub API
Frontend	Backend	Database	Integration

Architecture Principles

- Modular design for incremental delivery
- API-first approach for flexibility
- Role-based access control throughout
- Performance optimized for 500+ users

Data Strategy

- Encrypted at rest and in transit
- FERPA-compliant data handling
- Audit logging for accountability
- Privacy controls for students

Rabbit Holes to Avoid

⚠ Scope Creep Traps

- ⚠ **Advanced analytics:** Start with simple counts and trends. No ML or predictive models in v1.
- ⚠ **Mobile apps:** Responsive web only. No native iOS/Android apps.
- ⚠ **Real-time notifications:** Email digests are sufficient. No push notifications or websockets.
- ⚠ **Custom grading rubrics:** Fixed rubric for v1. No customizable grading systems.
- ⚠ **Team formation algorithms:** Manual assignment only. No auto-matching based on skills/preferences.

No-Gos: What We're NOT Building

⛔ Out of Scope

- ⛔ **LMS replacement:** Not competing with Canvas. Integration only.
- ⛔ **Code review tools:** Link to GitHub, don't rebuild their features.
- ⛔ **Video conferencing:** Zoom/Teams links only. No built-in video.
- ⛔ **Assignment submission:** Use Canvas. We track participation, not deliverables.
- ⛔ **Peer evaluation system:** Too complex for v1. Manual feedback only.

Risks & Mitigation

Technical Risks

- **GitHub API limits:** Cache aggressively, batch requests
- **Scale to 500+ users:** Load testing from week 2
- **OAuth complexity:** Use proven library, test early

Execution Risks

- **Feature creep:** Strict scope enforcement, say no often
- **Integration delays:** Mock external APIs for parallel work
- **User adoption:** Pilot with one class first

Timeline: 5 Weeks to MVP

Small-batch, high-impact delivery. Four core features. Incremental deployment. Working MVP by December 5, 2024.

1. **Week 1: Foundation (Nov 11-15)**
   Database schema, authentication framework, basic UI shell. Get Google OAuth working. Deploy infrastructure.
2. **Week 2: Core Features (Nov 18-22)**
   User management complete. Directory with search/filter. Attendance tracking with QR codes. First usability tests.
3. **Week 3: Work Journal (Nov 25-29)**
   Daily standup system. GitHub API integration. Team dashboards. Blocker escalation. Pilot with small group.
4. **Week 4: Integration (Dec 2-6)**
   Connect all modules. TA dashboard with early warnings. Analytics and reporting. Performance optimization. Bug fixes.
5. **Week 5: Polish & Launch (Dec 9-13)**
   User testing with real class. Documentation. Training materials. Final deployment. Monitoring and support plan.

Expected Impact

18x	60%	5 weeks
Faster class health checks	Less time on admin	Earlier intervention

Success Metrics

Quantitative

- Dr. X spends <10 min/week checking class health
- TAs save 8+ hours/week on admin tasks
- Attendance recorded in <60 seconds for 200+ students
- 90% of students submit 4+ standups per week

Qualitative

- TAs report feeling prepared for meetings
- Students feel their work is recognized
- At-risk students identified by week 3
- Team leaders report coordination is easier

The Ask

We're seeking approval to build this 5-week MVP with the following resources:

- Development team of 5 engineers (already allocated)
- Access to UCSD infrastructure (Google OAuth, hosting)
- Pilot class for weeks 4-5 (50-100 students)
- Weekly check-ins with Dr. X for feedback

Why This Matters

Every quarter, **thousands of students** take project-based software engineering courses. Some thrive. Others slip through the cracks—not because they're incapable, but because **the tools fail them**. Conductor Tool changes that.

This isn't just about efficiency. It's about **equity at scale**. It's about making sure that in a class of 500, the quiet contributor in the back row gets the same attention as the vocal student in the front. It's about giving teaching staff the tools they need to **actually teach**, not just administrate.

Let's build a system where every student can be seen.

Even in a class of 500.