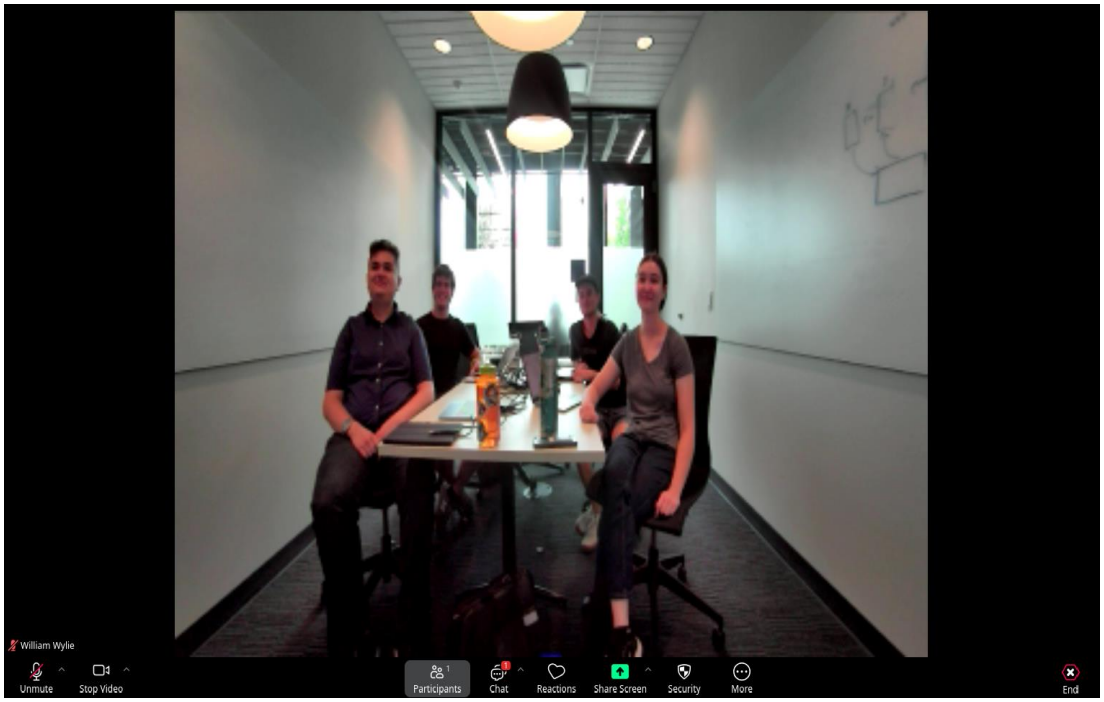
CSCI 370 FINAL REPORT

Mines Green Labs/ CSM Poole 1

Brian Luzar
Mathias Mares
Morgan Steele
William Wylie

Revised December 1, 2025



CSCI 370 Fall 2025

Mr. Tree Michael

Table of Contents

Table 1: Revision history

Revision	Date	Comments
New	9/1/2025	Gluten is everywhere. This is our initial draft.
Rev – 2	9/15/2025	Sprint 2 Additions
Rev – 3	10/8/2025	Sprint 3 Additions: Ethical Considerations (of Gluten of course)
Rev – 4	10/22/2025	Sprint 4 Additions: Results Document
Rev – 5	12/1/2025	Sprint 5: Final Paper Review Day
Rev – 6	12/3/2025	Polishing to turn in final draft

Introduction

Our team developed a comprehensive platform that streamlines laboratory certification for energy efficiency and environmental sustainability. This centralized system allows clients to track certification progress, ensure regulatory compliance, and significantly reduce administrative workload. This new system offloads user management and manual tracking of laboratory certification.

At the core of the platform is a dynamic scoring engine that benchmarks laboratory sustainability practices against industry standards. By generating quantifiable performance metrics, the system delivers clear, actionable feedback, showing facilities exactly where they stand and what improvements will advance their certification status.

The certification journey begins with an intuitive onboarding survey that captures essential operational data as defined by the sustainability coordinator. This initial assessment creates a baseline profile for each laboratory and provides a straightforward entry point for organizations committed to sustainable practices.

This platform does more than simplify paperwork, it creates a framework for measurable environmental progress, transparent evaluation, and ongoing operational excellence in laboratory management.

Requirements

Functional Requirements

- Survey Administrators
    - Sign in with admin privileges
    - Create and manage labs
    - View completed surveys for each lab
    - Access basic analytics
    - Dashboards for results
- Sustainability Survey
    - Survey form with questions on lab practices
    - Different questions based on the lab type
    - Auto save feature
    - Submit surveys and upload data to database
- Data
    - Calculate results based on survey answers
- Web Pages
    - Main landing page to select the lab
    - Sign in or create new account page
    - Survey page for recording sustainability practices
    - Web applications are accessible from mobile devices and computers

Non – Functional Requirements

- Authentication
    - Email and password login
    - Role-based access
    - Roles
        - Create account and log in
        - Create lab profile with specified lab type
        - Fill out sustainability survey for assigned lab
        - Add other users to lab group to edit survey
- Email Notifications
    - Reminders to finish started survey
    - Regular reminder to retake survey after 1 year
    - Results of survey and ideas for improvement
- Lab coordinator survey item bookmarking

Risks

- Mines hosting
    - Relying on Mines-provided hosting infrastructure may introduce limitations in uptime guarantees, resource allocation, and technical support responsiveness. Institutional hosting environments may have restricted access to certain technologies, slower deployment cycles due to approval processes, and potential conflicts with other applications sharing the same infrastructure.
- Auto-save Performance
    - Implementing auto-save functionality for survey responses and certification data may create performance bottlenecks, especially with multiple concurrent users. Frequent database writes could slow down the application, cause data conflicts, or result in incomplete saves if network connectivity is unstable.
- Scalability
    - The application architecture may not adequately support growth in the number of laboratories, users, certification requests, or data volume. Without proper design for horizontal scaling, the system could experience degraded performance as adoption increases.
- Reliability
    - System failures, unhandled errors, or inadequate backup procedures could result in service downtime or data loss. Dependencies on third-party services or databases introduce additional failure points.
- Sensitive Data Exposure
    - The application will handle confidential laboratory information, operational data, and potentially proprietary sustainability practices. Inadequate encryption, insecure APIs, or improper access controls could expose this sensitive information to unauthorized parties.

- Weak Authentication
    - Insufficient authentication mechanisms (such as basic username/password without multi-factor authentication, weak password policies, or lack of session management) could allow unauthorized access to the platform.

Definition of Done

A clearly documented pathway for the Colorado School of Mines Lab Sustainability Coordinator to complete the certification application process, including:

- A minimum viable product (MVP) application that is production-ready and deployable.
- Successful deployment to a production environment.
- Testing completed with at least one pilot laboratory using real-world data.
- Various documentation for the end users and the administrators.
- Acceptance criteria validated by the sustainability coordinator.
- Any critical bugs or blockers resolved.

System Architecture

Database Diagram

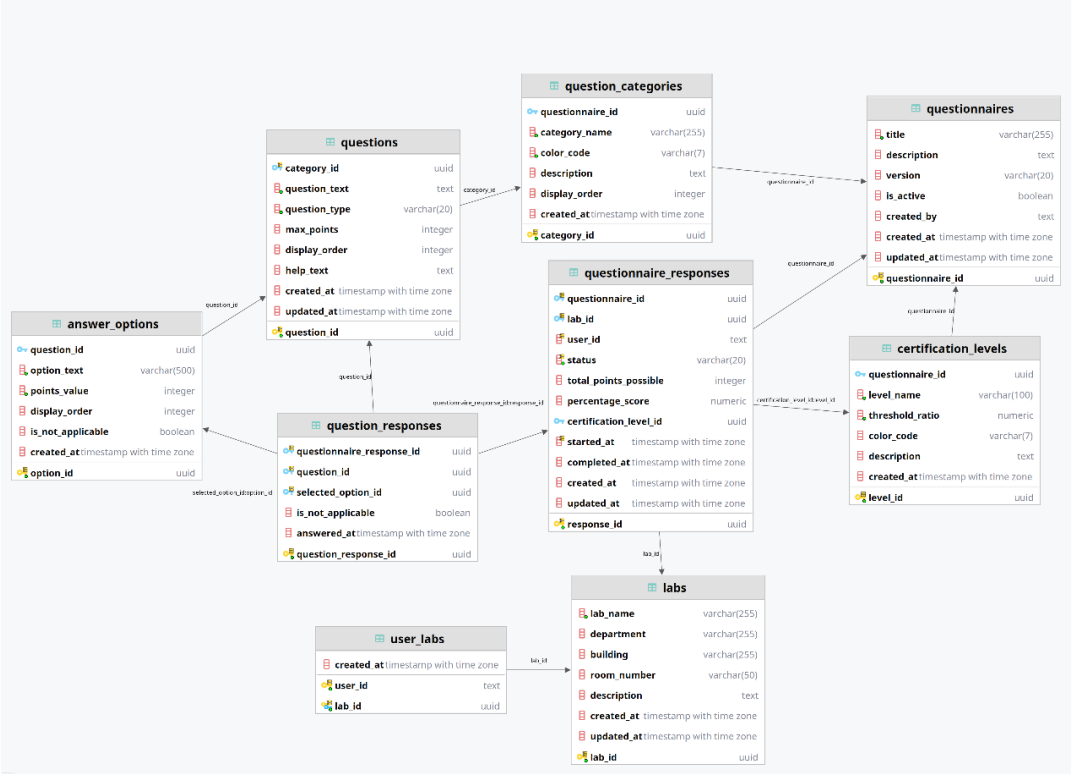


Figure 1: A database diagram showing the relationships between users, laboratories, and surveys to assign admin and normal users to the appropriate permissions.

User Actions

Figure 2: This diagram shows the different actions which Site Administrators can take.

User Actions

Figure 3: This diagram shows the different actions which users can take.

Workflows

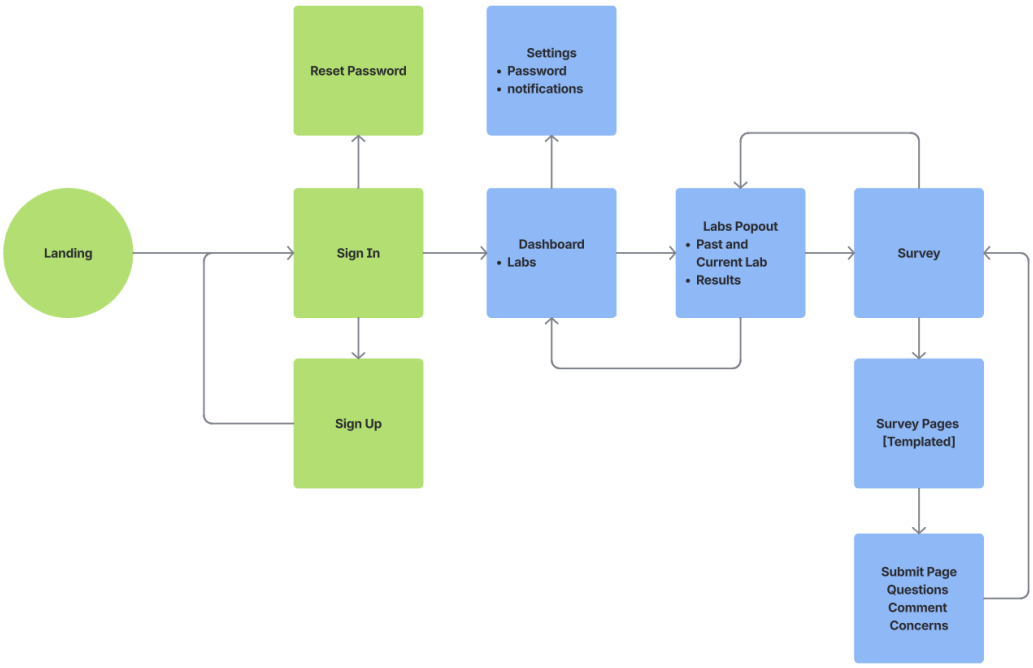


Figure 4: This diagram shows how users interact with the program from login to the completion of a survey.

Quality Assurance

- Unit testing
    - Create tests to ensure that our database is up and running.
        - Database is editable.
        - New entries are populated correctly.
        - Requests get the appropriate data returned correctly.
    - Check that webpages populate correctly.
        - This includes the Labs page, Overview page, and Question pages.
- Manual User Interface Testing
    - Ensure the buttons show correctly
        - The submit button only works if all questionnaires are completed.
        - The certify button only works if the user is an admin and the questionnaire is in review.
        - All other buttons like “New Survey”, and “Next” buttons take the user to the correct page.
    - Ensure all links work correctly

- Integration testing
    - Create tests to ensure that code integrates including CI/CD pipelines.
- Code reviews
    - Review new code as a group before merging to the repo.
    - Merge requests require two approvals .
- Code metrics
    - Response time.
    - Initial page load.
- ESLint static program analysis
    - Ensure consistent code writing and catches type errors, unused variables, and more.

Project Ethical Considerations

ACM Principles Relevant to Development

Principle 1.1 - Public Interest:

This system directly serves the public interest by promoting environmental sustainability in academic laboratories. Labs consume significant energy and resources, and this certification system encourages practices that reduce environmental impact, benefit society, and future generations.

Principle 1.2 - Public Health and Safety:

The sustainability questionnaire addresses chemical management, waste disposal, and safety practices. Proper implementation ensures labs follow safe environmental practices that protect both lab workers and the broader community from hazardous materials.

Principle 2.1 - Professional Competence:

The system must accurately assess sustainability practices. We are working with our client, who is an expert in environmental science, to ensure the scoring algorithms and certification thresholds are scientifically accurate.

Principle 2.5 - Comprehensive and Thorough Evaluation:

The percentage-based scoring system and "not applicable" options ensure it is a fair assessment. Each lab's unique context is considered rather than applying a one-size-fits-all approach.

ACM Principle Dangers if Our Product Fails

Principle 1.3 - Honesty:

Risk: Users might take advantage of the system by marking questions as "Not Applicable" when they apply for certifications, or by misrepresenting their practices to achieve higher certifications.

Negative Impact: False certifications undermine the entire program's credibility. Institutions might claim environmental achievements they haven't made, leading to "greenwashing" rather than genuine environmental improvement. This damages public trust and wastes resources on ineffective programs.

Principle 1.4 - Fairness and Non-Discrimination:

Risk: The scoring system might inadvertently favor well-funded research labs over teaching labs or smaller institutions. Questions about expensive equipment (VAV fume hoods, recirculating cooling systems) could penalize labs that can't afford upgrades despite good practices.

Negative Impact: Creates equity issues where resource-poor labs are labeled as "unsustainable" despite doing their best with available resources. Could discourage participation from institutions that need the most help improving.

Principle 1.6 - Privacy:

Risk: Lab performance data can be sensitive. Poor certifications might affect funding, reputation, or competitive positioning. If data isn't properly secured, it could be accessed by unauthorized parties.

Negative Impact: Labs might refuse to participate if they fear negative consequences from honest reporting. Data breaches could expose institutional weaknesses to competitors or funding agencies prematurely.

Michael Davis Tests

Test 1: Publicity Test

Applied to the certification survey: If the public were to find out the workings of the scoring mechanisms would they find it fairly based?

Pass: If the thresholds are evidence-based and validated by sustainability experts. The transparency of showing "63 out of 75 possible points = 84%" is defensible because it clearly shows what was assessed and what wasn't applicable.

Fail: If the thresholds were arbitrary (e.g., setting platinum at 86% simply because it sounds exclusive) without scientific basis, this would not withstand public scrutiny.

Action Item: Document the rationale for all certification thresholds with input from environmental science experts and sustainability professionals.

Test 2: Reversibility Test

Applied to the "not applicable" feature: As a lab coordinator, would I find it fair if my lab's score excluded questions that genuinely don't apply to my facility?

Pass: Yes, this is ethical. A physics teaching lab shouldn't be penalized for not having chemical fume hoods. The “N/A” option recognizes that different labs have different operational contexts.

Potential Issue: Need safeguards against abuse. As a coordinator, I might be tempted to mark things "N/A" that actually do apply but where I'm performing poorly.

Action Item: Implement audit mechanisms, such as having an administrator review submissions for suspicious patterns.

Ethical Consideration: Software Quality

If Testing is Inadequate:

Calculation Errors: The percentage calculation trigger might miscalculate scores, awarding certifications to labs that don't deserve them or vice versa. This creates false data that institutions might use for funding decisions or public relations.

"N/A" Logic Bugs: If the system incorrectly handles "N/A" responses, results become meaningless.

If Data Validation is Missing:

Garbage In, Garbage Out: Without input validation, users could enter anything. Empty descriptions, invalid room numbers, or contradictory responses would trash the database and make reporting unreliable.

If Security Testing is Insufficient:

Data Manipulation: Coordinators might exploit vulnerabilities to change their responses after submission, alter point values, or manipulate certification thresholds.

Trust Erosion: If one institution discovers they can "hack" better certifications, word spreads and the program will have a bad reputation.

If User Acceptance Testing is Skipped:

Unusable Interface: Complex forms might lead to completion errors where users accidentally select wrong answers, misunderstand questions, or abandon the process entirely. This creates data that doesn't reflect actual practices.

Security Considerations

Authentication & Authorization:

Risk: Unauthorized access to modify lab data, questionnaires, or certification levels.

Mitigation: Implement role-based access control (RBAC). Coordinators can only modify their assigned labs; administrators control questionnaire creation; regular auditing of permission changes.

Data Integrity:

Risk: Tampering with completed questionnaires to improve scores, or changing certification thresholds retroactively.

Mitigation: Implement audit logs tracking all data modifications (who changed what, when). Make completed questionnaires immutable or require administrator approval for changes. Version control for questionnaires (prevent changing questions/points after responses exist)

SQL Injection & Input Validation:

Risk: Malicious input could compromise the database or inject false data.

Mitigation: Use parameterized queries, input sanitization, and validation on both client and server side. UUID-based keys help prevent sequential ID attacks.

Data Privacy:

Risk: Lab performance data could be competitive or politically sensitive.

Mitigation: Encrypt data at rest and in transit (HTTPS/TLS) Implement data access controls (coordinators only see their labs) Clear data retention policies. Privacy law compliance for user personal information

Session Management:

Risk: Session hijacking could allow attackers to impersonate users and submit false data.

Mitigation: Secure session tokens, proper timeout policies, CSRF protection.

Availability:

Risk: DDoS attacks or system failures during certification deadlines could prevent submissions.

Mitigation: Rate limiting, CDN protection, database backups, disaster recovery plan.

API Security:

Risk: If APIs aren't properly secured, external parties could scrape data or automate fraudulent submissions.

Mitigation: API authentication (JWT tokens), rate limiting, API gateway with monitoring.

Audit Trail Requirements:

Risk: If lab coordinators abuse the “N/A” feature they can get a passing score when they don’t deserve it.

Mitigation: Keep a log of all the following items:

- All “N/A” selections with timestamps
- Any modifications to submitted responses
- Changes to questionnaire structure or point values
- Failed authentication attempts
- Administrator actions

Use the logs gathered in a visual dashboard to allow for Survey Administrators to look for statistical outliers and allow for manual inspection of surveys.

Results

Project Overview

This project aims to develop a web-based platform that enables laboratory researchers to assess their environmental practices and obtain green lab certification. The application facilitates a collaborative certification process where multiple team members can contribute to a single laboratory's evaluation while providing administrative oversight for the approval workflow.

Current Progress

The frontend development is substantially complete, featuring a polished user interface that includes:

- A welcoming homepage that introduces the certification program
- A personalized user dashboard displaying lab detail cards for quick status overview
- Comprehensive survey results visualization
- Secure sign-in and authentication interfaces

The platform supports a multi-user collaboration model where individual researchers maintain separate accounts while being affiliated with a shared laboratory workspace. This architecture allows team members to collectively complete the certification survey, with concurrent saves to the database.

On the administrative side, we've built interfaces that allow program administrators to monitor participating laboratories, review submitted surveys, and approve certifications based on criteria which is preset but can be adjusted to the certifying administrators preference.

Technical Challenges

We are currently facing two significant obstacles:

1. **Backend Integration**: While backend services have been developed, we have encountered complications connecting these services to the frontend application. Specifically, we have had issues with understanding how to link our service layer to the API layer along with making sure that the right information made it to the frontend.

2. **Hosting Infrastructure**: Mines IT has presented barriers to deploying our required infrastructure. As an institution the school has moved to a cloud preferred model while only allowing apps which are specifically maintained under an end user license agreement and maintenance contract to be deployed.

Future Work

Backend Completion and Optimization

The backend infrastructure requires finalization and full integration with the frontend. While the core structure is established, several critical components need completion:

- **Service-API Layer Connection**: The architectural gaps between the service layer and API endpoints need resolution to ensure proper data flow and business logic execution.
- **Data Pipeline Verification**: Comprehensive validation should be implemented to confirm that information correctly propagates from the backend through the API to the frontend components.
- **Database Operations**: Transaction management, query optimization, and data consistency mechanisms across concurrent user operations require completion.

Performance optimization through caching strategies, query refinement, and efficient data processing would improve user experience during high-traffic periods.

Frontend Refinement

Once backend integration is stable, the frontend would benefit from quality and user experience improvements:

- **Visual Consistency**: Design elements including color schemes, typography, spacing, and component styling could be standardized across all pages to create a more cohesive brand identity.
- **User Experience Enhancements**: Navigation flows, contextual help and tooltips, form validation feedback, and survey progression could be refined for improved intuitiveness.
- **Responsive Design**: Full compatibility across desktop, tablet, and mobile devices with appropriate layout adjustments for different screen sizes should be verified and enhanced.
- **Loading States and Feedback**: Clear visual indicators for auto-save operations, data fetching, form submissions, and background processes would keep users better informed of system status.

System Robustness and Reliability

Several measures would strengthen the application's long-term maintainability and operational dependability:

- **Error Handling:** Robust error catching throughout the application with user-friendly error messages and graceful degradation when services become temporarily unavailable needs development.
- **Testing Infrastructure:** The testing suite should be expanded on, including unit tests for backend logic, integration tests for API endpoints, and end-to-end tests for critical user workflows such as survey completion and certification approval.
- **Performance Monitoring:** Logging systems and performance tracking would help identify bottlenecks and monitor system health in production.
- **Scalability Planning:** The database schema should be evaluated for future growth, infrastructure resource requirements assessed, and the architecture prepared to handle increased numbers of laboratories and concurrent users.
- **Deployment Resolution:** A viable hosting solution needs to be established. This may include continued dialogue with Mines IT to explore approved cloud platforms that meet institutional requirements, or investigation of alternative deployment strategies that comply with the cloud-preferred model and licensing constraints. Establishing a clear path to production deployment is essential for delivering value to the laboratory community.

Potential Feature Expansion

Depending on stakeholder priorities and available resources, several enhancements could add value to the platform:

- **Reporting Dashboards**: Analytics views that allow administrators to track certification trends, identify common improvement areas, and generate institutional sustainability reports.
- **Certification Renewal Workflows**: Periodic recertification processes to ensure laboratories maintain their green standards over time.
- **Comparative Analytics**: Features enabling laboratories to benchmark their performance against anonymized peer data to identify best practices and improvement opportunities.
- **Export Functionality**: Capabilities allowing users to generate PDF reports of their certification status, survey results, and improvement recommendations for internal use and external recognition.

User feedback from pilot testing with participating laboratories would help guide prioritization of these features and inform iterative improvements to the platform.

Lessons Learned

Backend-First Development Approach

A critical lesson from this project is the importance of prioritizing backend architecture and implementation early in the development cycle. Our experience revealed that retrofitting backend services to accommodate an already-developed frontend is significantly more challenging than the inverse approach. When the frontend is built first, it often makes assumptions about data structures, API response formats, and business logic that may not align with optimal backend design. This misalignment creates substantial technical debt and requires extensive refactoring to achieve proper integration.

In future projects, establishing the backend infrastructure first—including database schemas, API endpoints, service layer logic, and data models—would provide a stable foundation upon which the frontend can be built with confidence. This approach ensures that the user interface accurately reflects the capabilities and constraints of the underlying system, rather than forcing the backend to contort itself to meet frontend expectations that may not be technically feasible or efficient.

Comprehensive Planning and System Design

Another key takeaway is the necessity of thorough upfront planning, particularly regarding how different system components will interact. Before writing any code, the team should invest significant time in:

- **Architecture Design**: Creating detailed diagrams that map out how the frontend, backend, database, and external services communicate with each other.
- **API Specification**: Defining all API endpoints, request/response formats, authentication mechanisms, and error handling protocols before implementation begins.
- **Data Flow Mapping**: Documenting how data moves through the system from user input through processing to storage and retrieval.
- **Integration Strategy**: Establishing clear contracts between frontend and backend teams about data structures, naming conventions, and communication protocols.
- **Technology Stack Alignment**: Ensuring all chosen technologies are compatible and that the team has adequate expertise with each component.

While comprehensive planning may feel like it slows initial progress, our experience demonstrated that time invested in design prevents far more time lost to rework, debugging integration issues, and reconciling architectural mismatches later in development. The cost of changing a design document is negligible; the cost of refactoring implemented code across multiple interconnected systems is substantial.

Infrastructure and Deployment Considerations

We also learned the importance of addressing hosting and deployment requirements at the project's outset rather than treating them as afterthoughts. Engaging with IT infrastructure teams early to understand institutional constraints, approval processes, licensing requirements, and available platforms would have allowed us to design our application within those parameters from the beginning.

Discovering late in the project that our planned deployment approach conflicts with institutional policies created significant delays and uncertainty.

Future projects should include infrastructure stakeholders in initial planning conversations and secure deployment commitments before extensive development begins.

Communication and Coordination

Effective communication between team members working on different system components proved essential but sometimes challenging. Regular synchronization meetings to discuss interface contracts, data requirements, and integration points helped prevent divergent implementations. However, even more structured communication protocols—such as shared API documentation that both frontend and backend developers commit to following—would have further reduced integration friction.

Iterative Integration Testing

Rather than developing frontend and backend components in isolation and attempting integration only near project completion, continuous integration testing throughout the development process would have identified compatibility issues much earlier when they were easier to address. Establishing automated integration tests and requiring that new features pass end-to-end testing before being considered complete would have maintained system cohesion throughout development.

Team Profile

Mathias Mares is a Senior who enjoys building computer systems and distributed programs. While his focus isn't necessarily on Software Engineering, he does enjoy doing it from time to time but enjoys system engineering a whole lot more. He serves as the team's documentation master and chief of review. His favorite cheese category is Blue Cheese.

Morgan Steele is a Junior in Computer Science with a focus in cyber security. She enjoys coding useless mini games and threat vector hunting. One of her top favorite kinds of cheese is Feta, but how can one truly choose from so many great options?

Brian Luzar is a Junior in Computer Science who enjoys working with computer graphics. Although not related to computer graphics, he has experience working with full stack web development. He serves as the team's system design lead. His cheese preference is cotija.

William Wylie is a Junior Computer Science major who enjoys Cybersecurity and Artificial Intelligence. He also enjoys learning new skills within the scope of Computer Science, which is where this project came into scope. His favorite cheese is Sharp Chedder.