Computer Science 3090 Dashboard

Design Document

Team sd25-14

Client: Simanta Mitra

Adviser: Simanta Mitra

Team Members/Roles:

Bradley Gaines - Component Design, Cross-Component Integration, Testing

Breckin Bartels - Component Design, Cross-Component Integration, Testing

Kat Christofferson - Component Design, Cross-Component Integration, Testing

Ria Patel - Component Design, Cross-Component Integration, Project Manager

Thomas Payton - Component Design, Cross-Component Integration, Testing

Varun Jain - Component Design, Cross-Component Integration, Client Interaction

Team email: sdmay25-14@iastate.edu

Team Website: https://sdmay25-14.sd.ece.iastate.edu

Executive Summary

Dr. Simanta Mitra teaches the course Computer Science 3090 at Iowa State University. Throughout this course, Dr. Mitra uses multiple applications including Canvas, Gitlab, Google Forms, and CATme, for assignments, projects, and feedback. With numerous applications being used continuously, organization and efficiency is hard to maintain throughout the semester. Dr. Mitra proposed a project, given his desire to have a dashboard that integrates the data from all applications into one website. That project is our Computer Science 3090 dashboard! This dashboard carries great importance for the entire Computer Science 3090 community, allowing easier access and organization for all class materials, resources, and feedback. Our dashboard is designed to allow use by professors, the head teaching assistant, teaching assistants, and students.

Requirements for our project were originally very minimal. Dr. Mitra requested that the dashboard be available to everyone involved with Computer Science 3090, allowing for different views based on what data pertained to the user accessing the dashboard. He requested that all applications would be represented throughout our dashboard in an organized and meticulous manner. His final requirement requested was to make sure the dashboard could be reused for following semesters without erasing the semester data but rather storing it for future use. Otherwise, Dr. Mitra left design decisions in our hands, requesting that we continuously update him on our choices for feedback on what would be most beneficial for the users.

Our initial approach on the project was to divide our project into components to take advantage of our team's strengths. We separated the project into three components: frontend, backend, and security. While most team members are contributing to multiple components, the breakdown allows us to understand the scope of our project and the best way to proceed to create a dashboard that we were proud of. After we understood the scope of our project, we began to research on how to best implement our dashboard. We decided to use Next.JS for the frontend, Node.js with TypeScript for the backend, Amazon Web Services for hosting, and an SQL database for data management. We began by forming our components separately, then working together to connect frontend to the backend. We planned to implement security measures once we had a functional dashboard.

We are currently at the stage of connecting the frontend and backend. Throughout the semester, we made great progress on the individual components.

Our frontend team have created a functioning dashboard with mock data. We have created features like file upload page, attendance questioner, displaying of each team and their team status. We have also implemented logging in with google authentication. Currently, we have a solid ground work, we just need to work on populate our frontend with real data with POST and GET requests.

From the backend perspective, so far, we have already created the database structure including the data tables, routes, and API requests. We are currently implementing the API calls including GET and POST, and continuing updating the database to fit our needs and increase efficiency. We are also testing GET request for creating Teams from a list that is sent from the front end, and a GET request that to pull information from the Team data table. Forth more we have been implementing API calls for an attendance feature.

Our dashboard meets the first semester requirements as detailed by Dr. Mitra. The requirements given to us for this semester included the creation of a file uploader, an attendance page, and the ability to view teams and their status concisely. We have presented our current design and it has been approved by Dr. Mitra.

We will continue to add more functionality to the Dashboard throughout next semester. This includes the ability to comment on team or profile pages. We will also be working on creating an archival system so that Dr. Mitra can access at prior projects. Towards the end of next semester, we will migrate our application onto an AWS server and AWS database, increasing the security of our application.

Learning Summary

Development Standards & Practices Used

Development Standards:

- IEEE 830-1998: Recommended Practice for Software Requirements Specifications.
- IEEE 1012-2016: Standard for System, Software, and Hardware Verification and Validation.
- ISO/IEC 27001-2022: Information Security, Cyber Security, and Privacy Protection.

Summary of Requirements

Functional Requirements:

- Professors must be able to manage student progress, assignment grades, attendance records, and TA feedback/reflections (constraint).
- Professors need to be able to communicate important class details to students and TAs (constraint).
- Students require access to all course information and resources.
- Students must be able to view attendance records and project progression easily.
- TAs need access to feedback from students and projects.
- TAs must be able to efficiently manage their students and projects.
- Head TAs require an overview of feedback from all TAs.
- Application needs to be able to reprogram for future course sections (constraint).

Resource Requirements:

- Development server (constraint).
- Development database (constraint).

- AWS server (constraint).
- AWS RDS (constraint).

User Interface Requirements:

- Role specific views for professors, students, TAs, and head TAs (constraint).
- Interactive design for both desktop and mobile use.
- Friendly and easy to navigate interface for all users (constraint).

Aesthetic Requirements:

- Clean and simple visual style that includes institutional branding (constraint).

Physical Requirements:

- Application must support both web and mobile platforms (constraint).
- Application must be compatible with various screen sizes and resolutions (constraint).

Quantitative Requirements:

- System must be capable of handling data for over 500 users (constraint).
- Dashboard response time should be less than 5 seconds for any action (constraint).

Applicable Courses from Iowa State University Curriculum

- ComS 309 – Project Development. This course helps us understand how to connect a frontend and backend.
- ComS 319 – Introduction to Javascript and Web development. We are using Typescript as our programming language, which is a variant of Javascript, taking this course helped us understand how to use Javascript effectively to create a web app.
- ComS 363 – Database management. This course helped us understand how to effectively put a database in place. ER diagrams along with some other practice from this course were applied to our project.

New Skills/Knowledge Acquired

Senior design has provided us with the opportunity to explore a diverse range of tools and technologies that extend beyond the scope of the ISU curriculum. For instance, our team has gained proficiency in TypeScript and Node.js for developing our project. Additionally, we have learned to integrate Google Authentication with Node.js and implement React Router for seamless navigation within the frontend. On the backend, we have acquired knowledge of using JWT tokens to enhance the security of our application. These skills have broadened our technical expertise and enabled us to build a robust and secure solution.

TABLE OF CONTENTS

FIGURES, TABLES, SYMBOLS & DEFINITIONS

FIGURES:

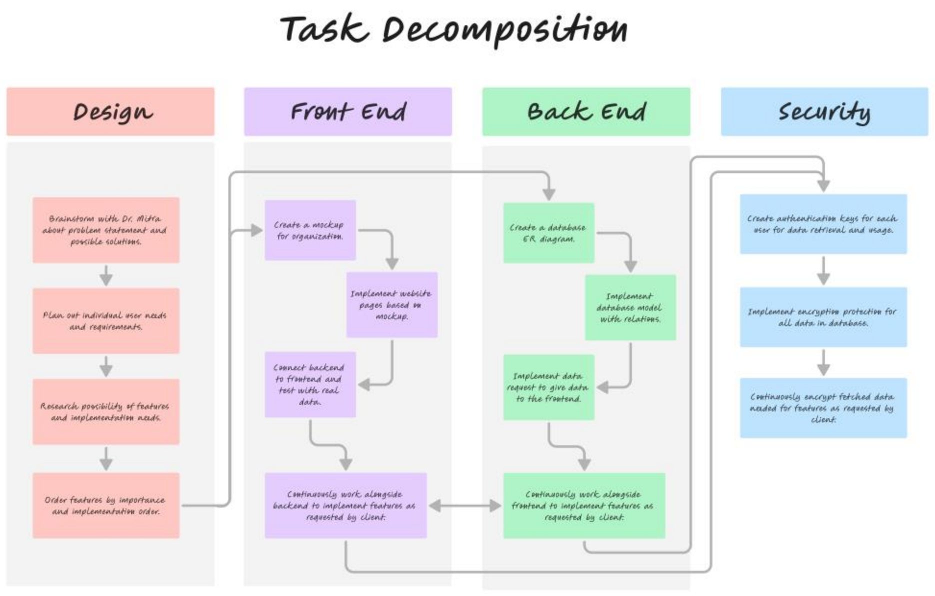
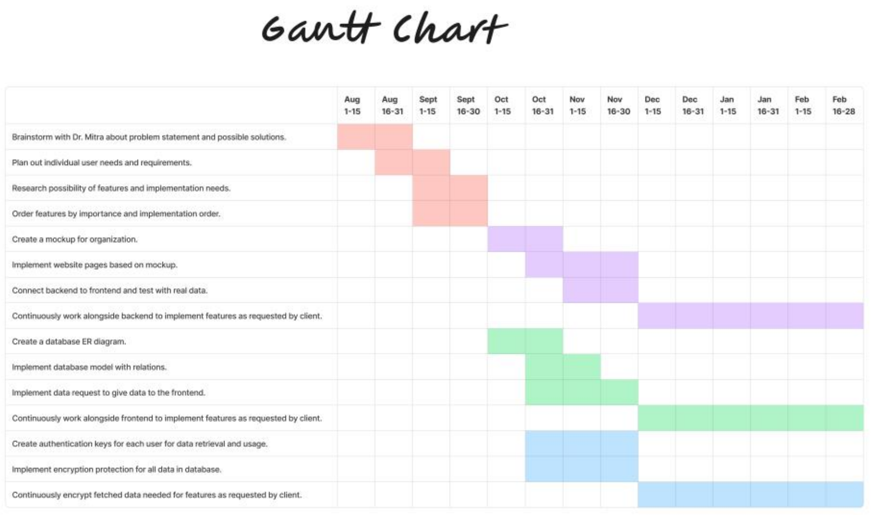

Task Decomposition
Design
Front End
Back End
Security
Brainstorm with Dr. Mitra about problem statement and possible solutions.
Plan out individual user needs and requirements.
Research possibility of features and implementation needs.
Order features by importance and implementation order.
Create a mockup for organization.
Implement website pages based on mockup.
Connect backend to frontend and test with real data.
Continuously work alongside backend to implement features as requested by client.
Create a database ER diagram.
Implement database model with relations.
Implement data request to give data to the frontend.
Continuously work alongside frontend to implement features as requested by client.
Create authentication keys for each user for data retrieval and usage.
Implement encryption protection for all data in database.
Continuously encrypt fetched data needed for features as requested by client.

Figure 1 – Task Decomposition Diagram



	Aug 1-15	Aug 16-31	Sept 1-15	Sept 16-30	Oct 1-15	Oct 16-31	Nov 1-15	Nov 16-30	Dec 1-15	Dec 16-31	Jan 1-15	Jan 16-31	Feb 1-15	Feb 16-28
Brainstorm with Dr. Mitra about problem statement and possible solutions.	🟥	🟥												
Plan out individual user needs and requirements.		🟥	🟥											
Research possibility of features and implementation needs.			🟥	🟥										
Order features by importance and implementation order.			🟥	🟥										
Create a mockup for organization.					🟪	🟪								
Implement website pages based on mockup.						🟪	🟪	🟪						
Connect backend to frontend and test with real data.							🟪	🟪						
Continuously work alongside backend to implement features as requested by client.									🟪	🟪	🟪	🟪	🟪	🟪
Create a database ER diagram.					🟩	🟩								
Implement database model with relations.						🟩	🟩							
Implement data request to give data to the frontend.						🟩	🟩	🟩						
Continuously work alongside frontend to implement features as requested by client.									🟩	🟩	🟩	🟩	🟩	🟩
Create authentication keys for each user for data retrieval and usage.						🟦	🟦	🟦						
Implement encryption protection for all data in database.						🟦	🟦	🟦						
Continuously encrypt fetched data needed for features as requested by client.									🟦	🟦	🟦	🟦	🟦	🟦

Figure 2 – Gantt Chart

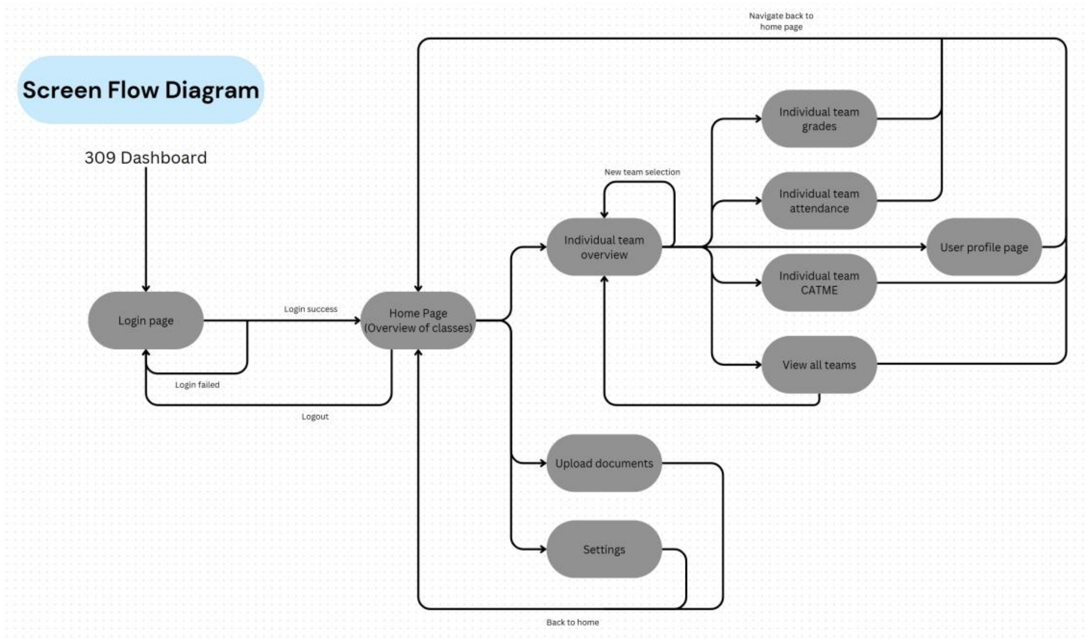


Figure 3 – Screen Flow Diagram

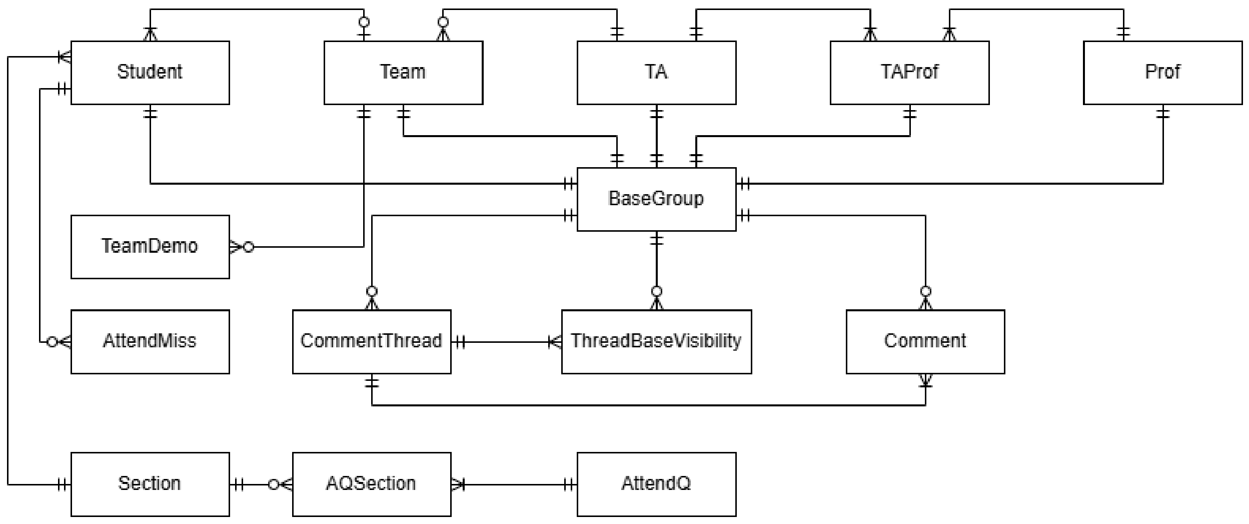


Figure 4 – ER Diagram

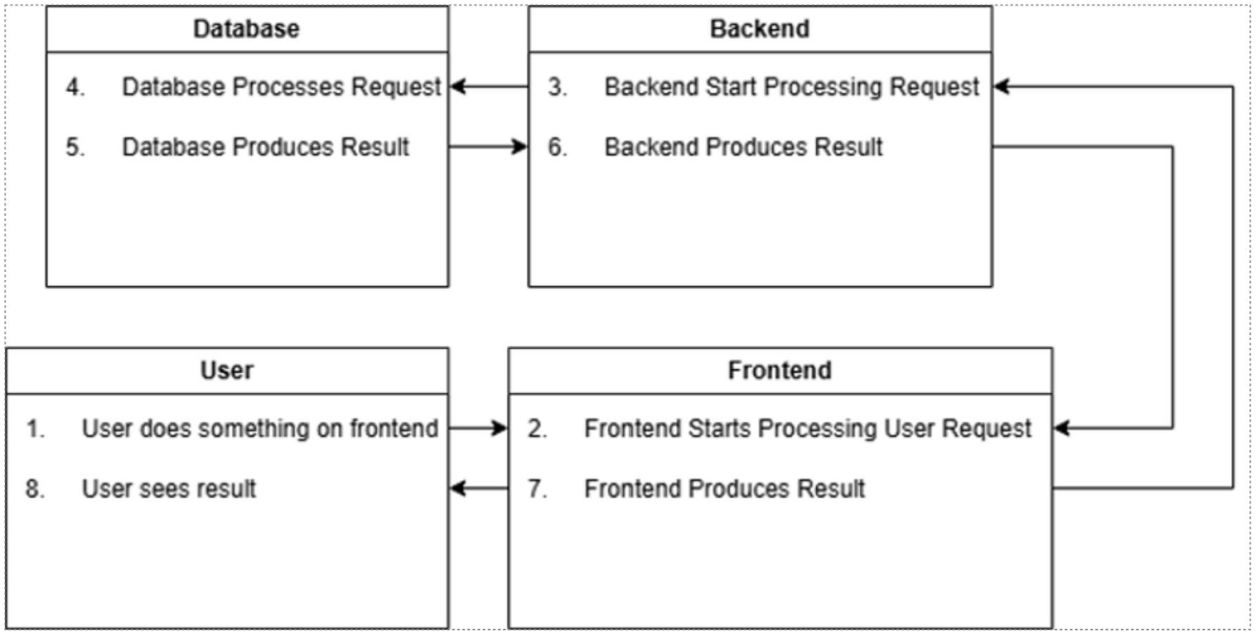

Database
4. Database Processes Request
5. Database Produces Result
Backend
3. Backend Start Processing Request
6. Backend Produces Result
User
1. User does something on frontend
8. User sees result
Frontend
2. Frontend Starts Processing User Request
7. Frontend Produces Result

Figure 5 – Application Connectivity Diagram

Definitions:

ER Diagram: Entity Relationship Diagram

API (Application Programming Interface): A set of rules and protocols that allows different software applications to communicate with each other.

Backend: The part of a system or application that is not directly accessed by the user, normally it is responsible for the manipulation and storing of data.

CATME: A web application used in COM S 309 for managing team formations and peer evaluations.

Frontend: The part of a system or application in which the user directly interacts with.

FERPA: The Family Educational Rights and Privacy Act, a federal law that protects the privacy of student education records.

JWT (JSON Web Token): A secure way of transmitting information between parties as a JSON object.

LMS (Learning Management System): A software application for administering, documenting, tracking, reporting, and delivering educational courses.

REST API: An architectural style for designing networked applications that uses HTTP requests to access and manipulate data.

SQL (Structured Query Language): A standardized programming language used for managing and manipulating databases.

TypeScript: A programming language developed and maintained by Microsoft, it is a strict syntactical superset of JavaScript and adds optional static typing to the language.

UI/UX: User Interface/User Experience - the design and interaction elements of an application that create the user experience.

Waterfall: A linear project management approach where each phase of the project must be completed before moving on to the next.

1 Introduction

1.1 Problem Statement

Dr. Simanta Mitra, professor of Computer Science at Iowa State University, has been proposing a project for numerous capstone groups to master. He desires a web application that would help facilitate the online workings of his Computer Science 3090 course. This course uses multiple platforms, such as CatMe, Canvas, Google Forms, Gitlab, etc., which makes organization and accessibility complicated for Dr. Mitra. To solve this issue, we will be creating a dashboard that will combine all applications used in this course into an easy-to-navigate program with all resources and class information in one place. Since other capstone groups have attempted this application previously, we will be using their code and research to help navigate through our attempt. This application will provide easy access for professors, teaching assistants, and students to review attendance, assignments, reflections, and any other aspects of this course. This dashboard is very important for the course, ensuring that professors, students, and teaching assistants have easier access to all aspects of the course. The dashboard is focused on critical information for the course (attendance, assignments, grades, projects, etc.) but will also be updated with highly desirable features (document scanning, ABET data analysis, etc.) as well.

1.2 Intended Users

Professors, teaching assistants, the head teaching assistant, and students will benefit greatly from this application. Professors need an efficient way to manage multiple aspects of Computer Science 3090, including student progress, assignments, attendance, and teaching assistants' feedback. They also need to communicate important dates and deadlines effectively to students and teaching assistants. Students need a centralized platform to access all course-related information, including assignments, feedback, and progress. They also need to easily track their attendance and project progress. Teaching assistants need a streamlined way to provide and manage feedback for student projects. They also need to efficiently manage the projects assigned to them. The head teaching assistants needs an overview of feedback from all teaching assistants to ensure consistency and quality. This application will satisfy all needs for these groups, with the possibility of more, making all course aspects easily accessible. Given that most of the needs are centralized for all groups, the application will focus on assignments, attendance, and feedback, but we aspire to modify the application in correlation to course changes.

2 Requirements, Constraints & Standards

2.1 Requirements & Constraints

Functional Requirements:

- Professors must be able to manage student progress, assignment grades, attendance records, and TA feedback/reflections (constraint).
- Professors need to be able to communicate important class details to students and TAs (constraint).
- Students require access to all course information and resources.
- Students must be able to view attendance records and project progression easily.
- TAs need access to feedback from students and projects.
- TAs must be able to efficiently manage their students and projects.
- Head TAs require an overview of feedback from all TAs.
- Application needs to be able to reprogram for future course sections (constraint).

Resource Requirements:

- Development server (constraint).
- Development database (constraint).
- AWS server (constraint).
- AWS RDS (constraint).

User Interface Requirements:

- Role specific views for professors, students, TAs, and head TAs (constraint).
- Interactive design for both desktop and mobile use.

- Friendly and easy to navigate interface for all users (constraint).

Aesthetic Requirements:

- Clean and simple visual style that includes institutional branding (constraint).

Physical Requirements:

- Application must support both web and mobile platforms (constraint).
- Application must be compatible with various screen sizes and resolutions (constraint).

Quantitative Requirements:

- System must be capable of handling data for over 500 users (constraint).
- Dashboard response time should be less than 5 seconds for any action (constraint).

2.2 Engineering Standards

Engineering standards are very important in everyday life. They ensure that products, technology, and services are able to work together easily and safely. Standards help maintain quality and reliability, making everyday life safer. Engineers must follow standards to help create solutions that continue to be compatible with existing systems, allowing for a more efficient, safe world.

In relation to our Computer Science 3090 Dashboard, the following standards are most relevant:

- IEEE 830-1998: Recommended Practice for Software Requirements Specifications: This standard provides guidelines for creating software requirements specifications, which would be useful in documenting the project's requirements.
- IEEE 1012-2016: Standard for System, Software, and Hardware Verification and Validation: This standard would be applicable for ensuring the quality and reliability of the dashboard system, which is important for our client relationship throughout the project.
- ISO/IEC 27001-2022: Information Security, Cyber Security, and Privacy Protection: This standard specifies requirements for establishing, implementing, and maintaining information security, which would be necessary for data encryption in our project.

The standards are very relevant to our Computer Science 3090 Dashboard project. IEEE 830-1998 will provide great insight on how we can maintain recommended practices throughout our dashboard. IEEE 1012-2016 will provide guidance on how to make a valuable and strong dashboard, strengthening our reputation with our client. ISO/IEC 27001-2022 will provide great insight on security standards, allowing us to make sure our dashboard respects the privacy of our users. Although there are other standards that are applicable to our project, these three are the most important.

Our team agrees that these standards are most important for our project. IEEE 830-1998 is very relevant for the frontend, backend, and security aspects of our project. IEEE 1012-2016 is extremely important for all members of our team, as we take pride in the work we complete, and want to provide the best possible dashboard that we are capable of for our client Dr. Mitra. ISO/IEC 27001-2022 is particularly important for the security implementation throughout our project, ensuring that we treat all users with the utmost respect for their personal and academic information.

Since our project is still is the design and initial implementation phase, we do not need to make changes to incorporate these standards, but rather keep them in mind while we continue further. Standards are very important to our team as we are entry level engineers, learning how to adapt to the industry. Although standards provide the necessary requirements for such topics, they also inspire us to learn from others experience.

3 Project Plan

3.1 Project Management/Tracking Procedures

The Computer Science 3090 Dashboard project embodies a waterfall and agile combination management style. The waterfall style allows a scheduled order of tasks for the beginning of our project including designing the frontend, designing the backend, implementing the frontend, implementing the backend, and implementing security. This allows us to create a functional, but simple, dashboard by the end of the first semester of our capstone. The agile style enables feedback from our original implementation to be implemented and adjusted in the dashboard during the later parts of our project, primary during the second semester of our capstone. We track progress through diagrams and the dashboard application. Diagrams are created and updated through Figma to represent our up to date feature design. The dashboard shows our up to date feature implementations as well.

3.2 Task Decomposition

Below is our task decomposition chart. This chart elucidates the major milestones for each aspect of our project including design, frontend, backend, and security. As shown below, design milestones focus on interpreting the requests of our client, and brainstorming how we plan to implement features into our dashboard. Frontend and backend follow the same milestone format of creating a high level schematic, implementing the details of the schematic, connecting with the other end, and then continuously updating the end to adjust to new feature implementations. Security has fewer milestones, all of which focus on gaining authorized access to data, encrypting the data, and continuously updating our security measures to adapt to new feature implementations.

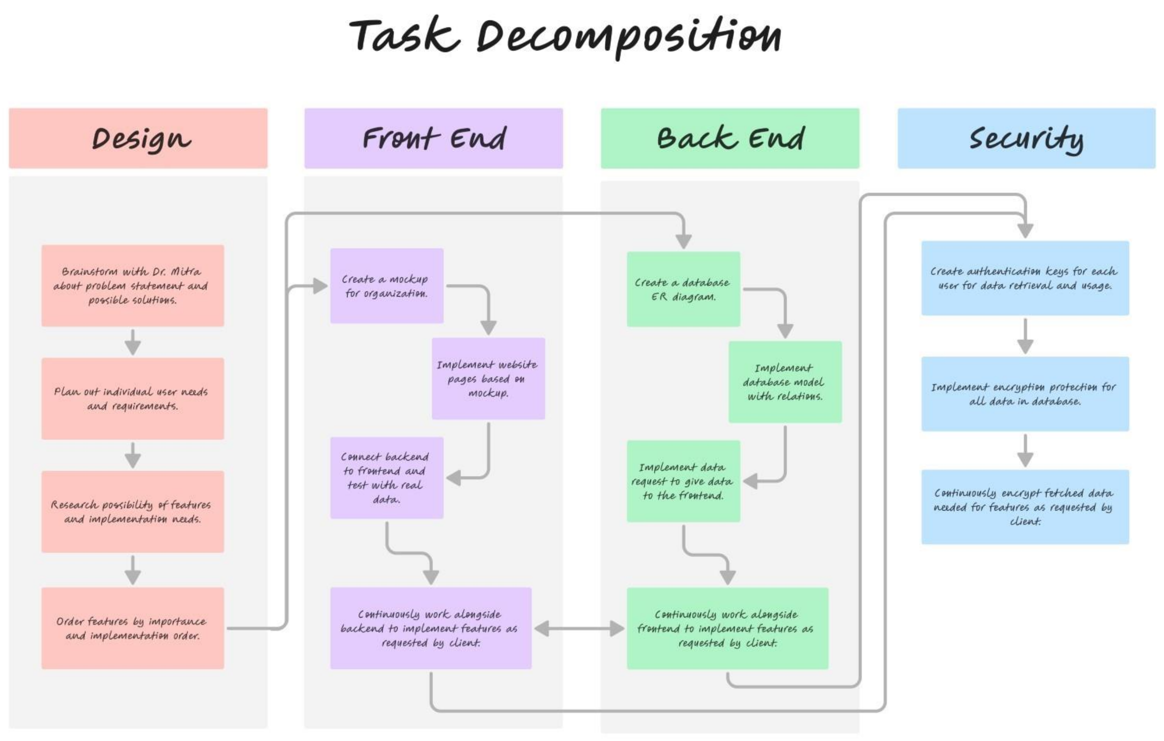

Task Decomposition
Design
Front End
Back End
Security
Brainstorm with Dr. Mitra about problem statement and possible solutions.
Plan out individual user needs and requirements.
Research possibility of features and implementation needs.
Order features by importance and implementation order.
Create a mockup for organization.
Implement website pages based on mockup.
Connect backend to frontend and test with real data.
Continuously work alongside backend to implement features as requested by client.
Create a database ER diagram.
Implement database model with relations.
Implement data request to give data to the frontend.
Continuously work alongside frontend to implement features as requested by client.
Create authentication keys for each user for data retrieval and usage.
Implement encryption protection for all data in database.
Continuously encrypt fetched data needed for features as requested by client.

Figure 1 – Task Decomposition Diagram

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Key milestones are broken down into frontend, backend, security, and overall. Most milestones include design ideas through diagrams, and some are actual implementation milestones. Progress is measured in different ways for all milestones since some are continuously updated to fit the timeline that Dr. Mitra has requested. Due to the layout of our project, milestones are not quantitative but rather qualitative.

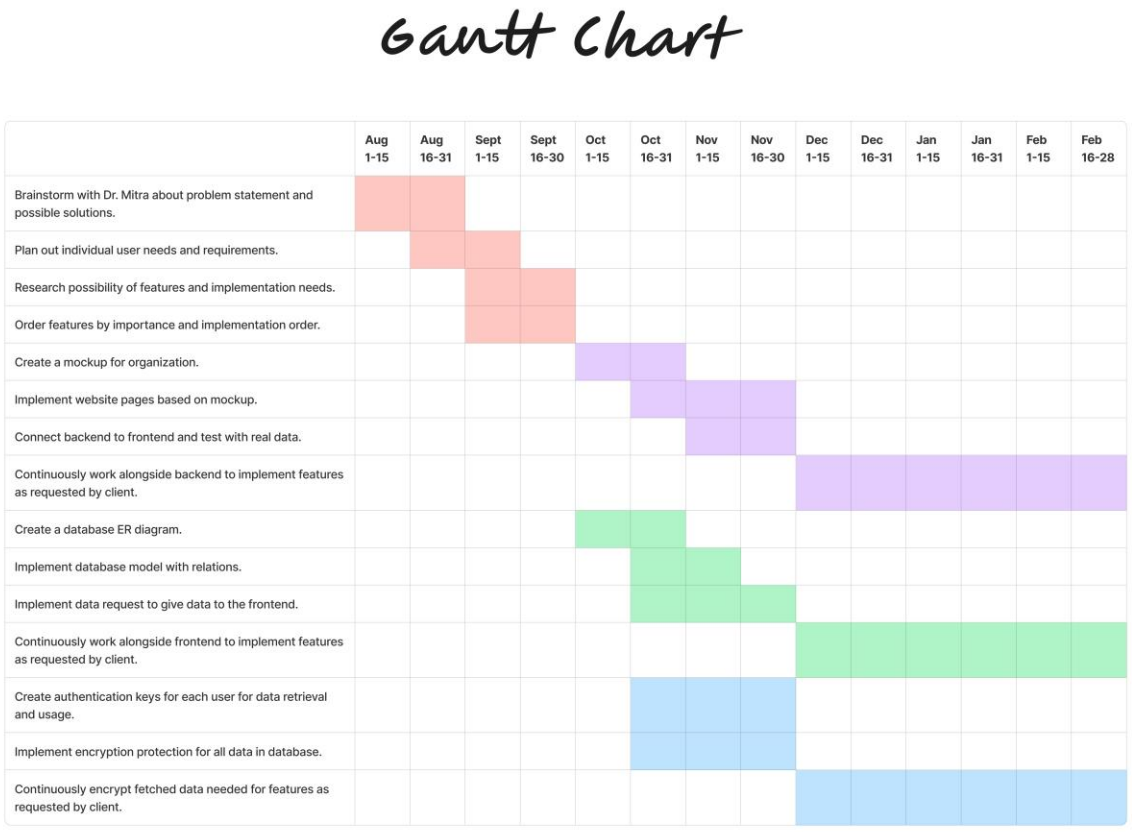
Frontend milestones include a Figma mockup, screen flow diagram, and implementation of pages. The Figma mockup is continuously updated for new features that Dr. Mitra requests. The mockup shows how each feature will be represented in the dashboard. The screen flow diagram follows the same milestone format. The diagram is continuously updated to show the connections between the pages that are requested for representing different features. Final milestones for frontend development are the actual implementation of pages in the dashboard, showing that said features are correctly embedded into the dashboard application.

Backend milestones include an ER diagram, implementation of tables, and the connection of the frontend to the backend, The ER diagram shows all tables necessary to analyze the fetched data from numerous websites. The diagram shows the breakdown of each table grouping, and how they are connected to other tables. Implementation of tables is another milestone for the backend. This ensures that all fetched data is organized and ready to be pulled. Final milestones for the backend development include connecting the frontend to the backend. This creates a full, functioning dashboard that is ready for use.

Security milestones include data encryption, accessibility keys, and login capabilities. Data encryption ensures that we protect the privacy of our users when fetching data from their academic resources. Accessibility keys allow us to grab user data from multiple applications used in the course, including Google Forms, CatMe, Canvas, and Gitlab. Login capabilities make the dashboard a private application, only to be accessed by individuals with an Iowa State email address. This ensures that any user data is safe and kept private. This also allows for individual users to only access the data that they are permitted to review. Professors will have access to all data, teaching assistants will have access to their student groups' data, and students will have access to their own data.

3.4 Project Timeline/Schedule

The Gantt chart below represents our project timeline from the start of capstone semester 1, to midway through capstone semester 2. Each color represents different aspects of the dashboard: pink for design, purple for frontend, green for backend, and blue for security. Design is the primary focus for the first few weeks of our project. Since our project embodies more of an agile management style, we will focus on frontend, backend, and security for the rest of the project timeline. Frontend, backend, and security all start by creating mockups/schematics for ideal design, but most of our time is spent in a trial-and-error phase by updating the dashboard to fit Dr. Mitra's requests as they come. Once we created a functional dashboard, as documented to be completed by November, the rest of our time is flexible to make sure both frontend and backend work together to implement new features. We will continue this simultaneous work style through to the end of our project, while ensuring our security measures are protective as well.



	Aug 1-15	Aug 16-31	Sept 1-15	Sept 16-30	Oct 1-15	Oct 16-31	Nov 1-15	Nov 16-30	Dec 1-15	Dec 16-31	Jan 1-15	Jan 16-31	Feb 1-15	Feb 16-28
Brainstorm with Dr. Mitra about problem statement and possible solutions.	🟥	🟥												
Plan out individual user needs and requirements.		🟥	🟥											
Research possibility of features and implementation needs.			🟥	🟥										
Order features by importance and implementation order.			🟥	🟥										
Create a mockup for organization.					🟪	🟪								
Implement website pages based on mockup.						🟪	🟪	🟪						
Connect backend to frontend and test with real data.							🟪	🟪						
Continuously work alongside backend to implement features as requested by client.									🟪	🟪	🟪	🟪	🟪	🟪
Create a database ER diagram.					🟩	🟩								
Implement database model with relations.						🟩	🟩							
Implement data request to give data to the frontend.						🟩	🟩	🟩						
Continuously work alongside frontend to implement features as requested by client.									🟩	🟩	🟩	🟩	🟩	🟩
Create authentication keys for each user for data retrieval and usage.						🟦	🟦	🟦						
Implement encryption protection for all data in database.						🟦	🟦	🟦						
Continuously encrypt fetched data needed for features as requested by client.									🟦	🟦	🟦	🟦	🟦	🟦

Figure 2 – Gantt Chart

3.5 Risks and Risk Management/Mitigation

Risks for our project are specified for either frontend, backend, security, and a few overall. All of our risks are not quantifiable in terms of a risk factor. None of our risks can be eliminated by purchasing parts, tools, or technology and implementing them. A few of our risks can be eliminated by using a different approach but the features related to these risks are desired as an add on function, not a requirement for our project.

Risks for our frontend include privacy and easy usability. Since our dashboard will display students' academic information, we must be careful how we represent the data. For instance, large text fonts and sensitive academic data should not be visible on the home screen. Easy usability is another risk for our dashboard. Making sure that every user is able to understand the dashboard easily and effectively use its features is one of our primary concerns. The dashboard is essential in making it easier for users to access data and resources for the course.

Risks for our backend include accurate tables and data analysis. Tables help organize the data that we pull from other applications. Making sure that we keep the data organized allows us to determine which users have access to which tables. This goes hand in hand with data analysis. When pulling data from other applications, we need to make sure we have the accessibility keys needed to grab our desired data. Since all applications that we are pulling for have their own security measures, we must analyze the data effectively and keep it organized for better representation in the frontend.

Risks for security include data encryption and academic data fetching. Since we are dealing with sensitive and private data, we need to make sure that all of our dashboard’s data is encrypted. This makes sure that our dashboard is safe from outside users trying to access our data. Since we are fetching academic data, it is a big risk to make sure that we are accessing the right data for each user. We must be very careful in pulling data that only applies to the individual user’s determined access.

Overall risks that we are facing are having different views for different users and ensuring our dashboard is usable for multiple course sections. Our design allows different views based on the login information. For example, when the professor logs in, they will have access to all of the data but when a student logs in, they will only have access to their data. Since different users have access to different data, the number of pages and the representation of the data will look different. Since this application is to be used for multiple semesters of the Computer Science 3090 course, we must make sure the dashboard can be reused fresh. Although the dashboard will be refreshed for new semesters, we must keep all past semester data accessible for the professor.

All risks are important for us team members as we work on this project. We are dealing with very sensitive data, which means that our users privacy comes first. Most of our risks revolve around fetching, organizing, and representing academic data. We plan to resolve these issues with encryption, login features, and private displays of information. There are a few features that Dr. Mitra has mentioned including, PDF scanner, template printouts with student identification, etc., that will possibly be added on after our base dashboard is fully functioning. New risks will arise with these features, but most will be able to be resolved by integrating different applications to implement the features. Our primary goal is to create a functioning dashboard for Dr. Mitra with the requirements he has requested, then we will attempt to add more features.

3.6 Personnel Effort Requirements

Task Requirements	Estimated Person-Hours to Complete
Design: Brainstorm with Dr. Mitra about problem statement and possible solutions.	10
Design: Plan out individual user needs and requirements.	10
Design: Research possibility of features and implementation needs.	20
Design: Order features by importance and implementation order.	8
Frontend: Create a mockup for organization.	8
Frontend: Create a screen flow diagram for page connections.	6
Frontend: Implement website pages based on mockup.	14
Frontend: Connect backend to frontend and test with real data.	18
Frontend: Continuously work alongside backend to implement features as requested by client.	30
Backend: Create a database ER diagram.	10
Backend: Implement database model with relations.	18
Backend: Implement data request to give data to the frontend.	12
Backend: Continuously work alongside frontend to implement features as requested by client.	30
Security: Create authentication keys for each user for data retrieval and usage.	20
Security: Implement encryption protection for all data in database.	20
Security: Continuously encrypt fetched data needed for features as requested by client.	20

All estimated person hours above are a baseline based estimated on our initial design. As the design of our project changes, the hours will as well.

3.7 Other Resource Requirements

Other resources required for the project, outside of team effort, is an AWS database and server. We were able to acquire both from the university. This project does not require any other resources for functional use, but the project does require data from applications that we will be representing throughout our dashboard, including GitLab, Canvas, and CATme.

4 Design

4.1 Design Context

4.1.1 Broader Context

While our primary focus is Computer Science 3090, our dashboard could potentially be adjusted to help the organization and efficiency of other courses that follow similar application use. We are designing our dashboard around the needs and desires of Dr. Mitra, as well as his teaching assistants and students. Our dashboard addresses societal needs for a home page for academic courses that use multiple applications throughout the semester. Our primary goal for the dashboard is it provide a central website that allows everyone to have access to all course materials, resources, and feedback in one place, maximizing efficiency and organization.

	Beneficence	**Nonmaleficence**	**Respect for Autonomy**	**Justice**
Public Health, Safety, and Welfare	Reduces stress of finding class resources.	Does not require users to adjust.	Users can choose their involvement level.	All users have availability to use.
Global, Cultural, and Social	Designed for Computer Science 3090 community.	Users are not required to use.	Users can choose to use.	All users have availability to use.
Environmental	Minimal resources used.	No download required.	Users can access simply with a web address.	Causes no harm to the environment.
Economic	Minimal resource costs.	No purchase required.	Free for all users.	Easily accessible for Computer Science 3090 community.

4.1.2 Prior Work/Solutions

https://seniord.cs.iastate.edu/2022-May-07/files/inline-files/402%20Final%20Presentation.pdf

https://seniord.cs.iastate.edu/2023-Jan-09/files/inline-files/402C%20Demo%201_0.pdf

Before our team started to work on this, there were 2 cs team that attempted this project. However, their attempts were unsuccessful to the requirements. One of the team used node.js, MongoDB to developed this project. They were able to access the multi-page application though navigation bar, and view all the projects and their status. For the backend, they were able to store data in the database, and able to query data from different semester and years. Some of the problem they faced were Integration with GitLab, designing the structure of database. Another problem they faced was an inability to deploy frontend to a server. The second team used React.js and node.js to develop this project. Due to the poor documentation, we are unsure they have completed. They issue they face were no API calls from Catme, higher authority needed for canvas and logging authentication issues.

Literature used throughout:

1. Shadcn/ui, “Introduction,” shadcn/ui, https://ui.shadcn.com/docs (accessed Dec. 5, 2024).
   - 1.1. This resource has been used by the frontend as a method for gathering documentation on features that are being used.
     - 1.1.1. For example, the Frontend is using Shadcn/ui buttons. In order to implement these buttons properly, we utilized the documentation to effectively implement these buttons.
2. “Rest api,” GitLab, https://docs.gitlab.com/ee/api/rest/ (accessed Dec. 5, 2024).
   - 2.1. This resource was used by the frontend to gain knowledge on various API request to Gitlab.
     - 2.1.1. These API request will be used to gather Gitlab commit history information for each team.
3. “typeorm,” GitBook, https://orkhan.gitbook.io/typeorm (accessed Dec. 5, 2024).
   - 3.1. This resource was used by the backend to figure out how to access the database efficiently through the code.

Similar products/solutions in the market:

1. Canvas
   - 1.1. Pros: Comprehensive learning management software, widely adopted
   - 1.2. Cons: Not customizable for specific course needs (like COM S 3090), difficult to connect and explore data from other resources
2. Previous Senior Design Projects

    2.1. Pros: Rudimentary integration with GitLab and Canvas, simple interface
    2.2. Cons: Limited functionality, security concerns, minimal integration with external services

Our solution advantages:

- Custom-built for COM S 3090 specific needs
- Integration with multiple platforms
- Role-based access control
- Simple, intuitive interface
- Secure data handling

Our solution disadvantages:

- Requires maintenance/work for each semester
- Dependent on external APIs which could break in the future
- Limited to single course use
- Requires basic training/explanations for features

4.1.3 Technical Complexity

Our dashboard in its entirety is a professional product, using real world professional tools and services like AWS. We have effectively from the ground up built a more narrow and custom Canvas tool for Dr. Mitra and Com S 3090. This involves all the moving parts of database, backend and frontend like any corporate application and also includes the handling and management of data.

The dashboard's frontend consists of a Next.JS web app utilizing a modern CSS library, shadcn/ui. Next.JS is a modern web framework based on React.JS that has built-in routing based on the directory hierarchy. We've implemented role-based access control directly in the frontend routing system, ensuring users only see content relevant to their permissions level. The user interface is built using reusable components from shadcn/ui, which we've customized to match our specific needs while maintaining a consistent design system throughout the dashboard. The frontend also includes form validation and file uploads all while maintaining responsive design that ensures the dashboard works across different device sizes and browsers.

The dashboard's backend consists of a Node.js server using TypeScript for type safety and code reliability. We've implemented a RESTful API architecture that handles complex data operations. The backend implements JWT (JSON Web Token) authentication to secure API endpoints and manage user sessions. We've created middleware functions to handle role-based access control, request validation, and error handling.

The database consists of a relational SQL structure hosted on a server provided by Iowa State, designed to efficiently store and manage course-related data. Our database schema includes multiple interconnected tables that handle user information, course data, attendance records, and team formations. We've implemented foreign key constraints to maintain data integrity and optimize query performance. The database design allows for semester-based data archival while maintaining historical records for future reference.

4.2 Design Exploration

4.2.1 Design Decisions

Our dashboard involves numerous decisions on design since we were given the opportunity to create an open-ended solution. Our most important decisions related to design included how to distribute the dashboard, how to represent information for different groups, and how to safely store and secure data. All design decisions are made to be in the best interest of our users, primarily the professor, but also the teaching assistants and students.

The first key design decision we made was how we chose to distribute the dashboard for professor, teaching assistant, and student use. We decided to create a website so that everyone has easy access to our solution. This was an important decision because choosing an application could restrict users from use based on their device (iOS, Android, etc.). Our project should be available to everyone involved with Computer Science 3090, so a website was the most universal option.

The second key design decision we made was how to represent information for different groups. Since professors, teaching assistants, and students will have access to the dashboard, we wanted to make sure each individual was only able to see data that was appropriate for them. Professors would be able to see all data, teaching assistants would be able to see data based on their groups, and students would be able to see individual and their group's data. This was a very important decision for us because we must make sure we are not violating class information by allowing individuals to see data that is not relevant or permissible to their situation. We plan to solve this issue by having different pages viewable based on the individual that is signed into the website.

The third key design decision was how to implement security to safely store and secure data. It is very important that we follow FERPA and other academic safety standards throughout our dashboard. In order to prevent outside infiltration of our data, we plan to encode all data coming in and out of our dashboard. To help this, we also plan on fetching academic data as requested, instead of storing it within our dashboard. As for within the dashboard, we are making sure our page designs respect the privacy of our users by not including grades on the home page or in large fonts.

4.2.2 Ideation

One key design decision that we are continuously working on updating is how we are representing data for our main user, the professor. This includes our website pages, as well as how we design those pages to represent data efficiently and legibly. Our goal is to make data more accessible for the professor, so we are continuously updating the design of our pages to provide easy navigation.

In the view of the professor, we originally had all information represented on the home page. We decided to have a home page and multiple drill down pages, to make information easier to find. Over time, we decided that the home page will include the most relevant information in snippets including overall project timeline, students who have continuously not been in attendance for lecture, project groups doing poor, and new comments from students. In order to see more detail the professor can drill down into sections to see overall data for that topic. We are continuously adding more drill down pages to improve the efficiency of finding direct information.

We originally planned to have a link on the home page that would link to Canvas to show class resources including timelines, assignments, etc. After further consideration, we plan to separate resources into multiple links in a drill down page to provide easier navigation for users.

We added a page to represent project demo status, even grouping them by which teaching assistant is the head of the group. We originally decided to show all demo status by color coordination immediately, but after further review we decided to include the most recent demo status then allow drill down to show all project demos.

We implemented another drill down page that will show comments/feedback from students. Our original idea was to show all comments in sequential order. After further review, we plan to have a status system for comments including new, read, and resolved. This would allow the professor to know the status of the comment, and make sure he acted on said comment.

As for the student attendance drill down page, we originally were going to store all attendance records by lecture. In order to see how frequently and when students were missing lecture, we plan to show overall individual attendance records then drill down to every attendance status, in order to find patterns in missing attendance if necessary.

4.2.3 Decision-Making and Trade-Off

Since our project decisions are mostly updating and making things more efficient rather than choosing another design option, we created a pro-con table for the original design versus the updated design. The table shows the pro-con for both design approaches. While our designs are continuously evolving, our pro-con table will as well. The table primarily focuses on our drill down pages stemming from our home page. Most drill down pages have their own drill down pages.

Feature	**Original Design**		**Updated Design**	
	Pros	**Cons**	**Pros**	**Cons**
Home Page	Simple access	Too much information in one place	Important information visible	1 extra step for more information
Resources	Simple access	Resources not available on dashboard	Resources available on dashboard	1 extra step for more information
Demo Status	Simple access	Too much information in one place	Most recent demo status shown immediately	1 extra step for more information
Comments/ Feedback	Simple access	Too much information in one place	Filtering by status to show importance	1 extra step for more information
Attendance	Simple access	Too much information in one place	Overall attendance record shown immediately	1 extra step for more information

4.3 Proposed Design

4.3.1 Overview

Our capstone project, Computer Science 3090 Dashboard, aims to provide easier access for the professor, teaching assistants, and students to all materials and resources used throughout the course. Materials and resources include course resources, assignment information, project information, comments and feedback, and attendance records by analyzing and representing data from Canvas, CATme and GitLab. We are creating a website dashboard that will not only represent all of the data from numerous applications, but organize it for easier navigation.

Our screen flow sequence is designed for the dashboard to greet the professor with a login page. This allows the dashboard to filter what data is accessible for the user, in which the professor has open access. After login is successful, the dashboard home page will show. This will give an overview of important data, with links to secondary pages for more details. Secondary pages include individual team overview, upload documents, and settings. From the individual team overview, the professor can drill down into sub-pages including individual team grades, individual team attendance, individual team CATme, view all teams, and user profile page. All sub-pages include in depth detail analyzed from the data pulled from the respective application.

The professor has access to the entire dashboard, including editing capabilities. Teaching assistants will have access to all pages stemming from individual team overview. Students will have access to all pages stemming from individual team overview, but the data will be restricted to only represent themselves and their team. This ensures that students do not have access to academic data that is not available to them.

4.3.2 Detailed Design and Visual(s)

Our dashboard is primarily broken into three components: frontend, backend, and security. Our team has split in order to address all components based on our degree specialties, while still informing each member about advances made in each component. Frontend focuses on the website visuals of the dashboard. Backend focuses on the connections from analyzing data to their respective location on the frontend. Security focuses on making sure all data and our dashboard itself is secure and follows standard guidelines.

Frontend is primarily organized based on a Screen-Flow diagram. The diagram shown below captures the page-to-page connections throughout the dashboard. The arrows should which page is accessible based on the page that the user is currently on.

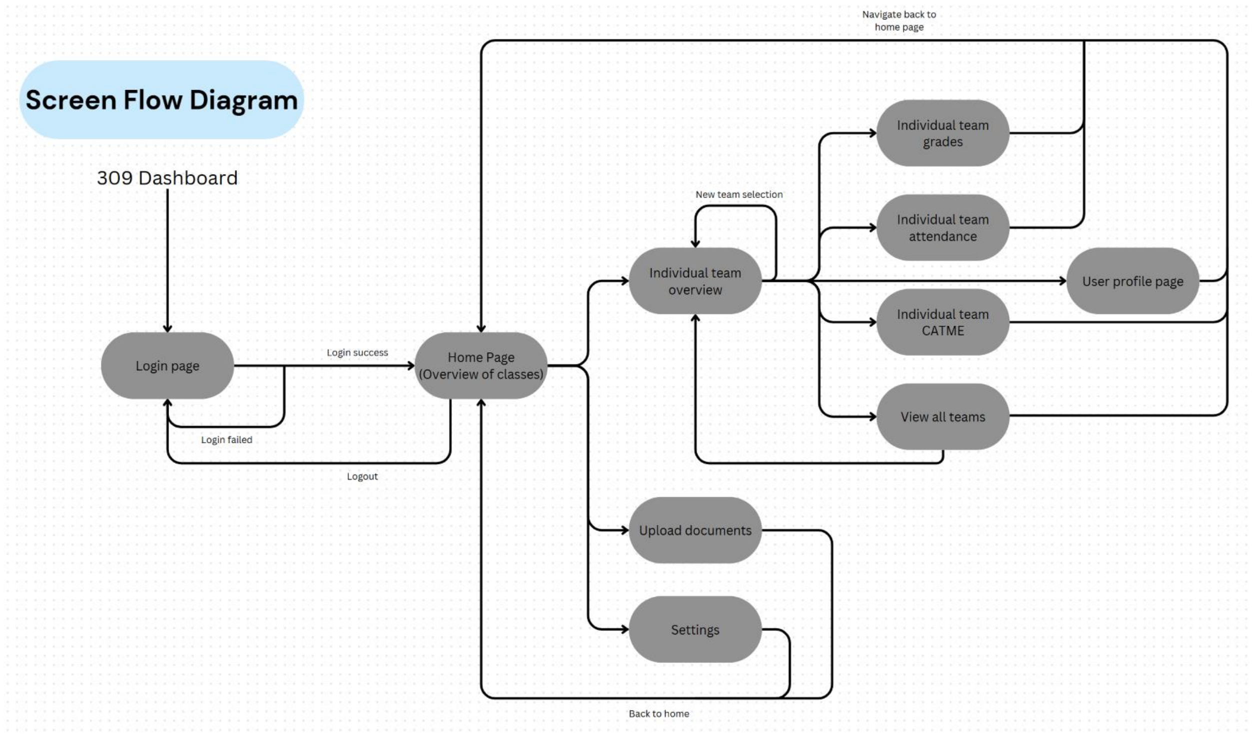


Figure 3 – Screen Flow Diagram

The backend is primarily organized using an ER diagram. The diagram shown below shows the different tables needed to organize the data that we are fetching from Canvas, CATme, and Gitlab. Each table is organized by the topic and followed by the information that is accessible if granted access to that table. Connections are shown through lines, enabling data to work together to make sure the data represented on the dashboard is fully encompassing the available data.

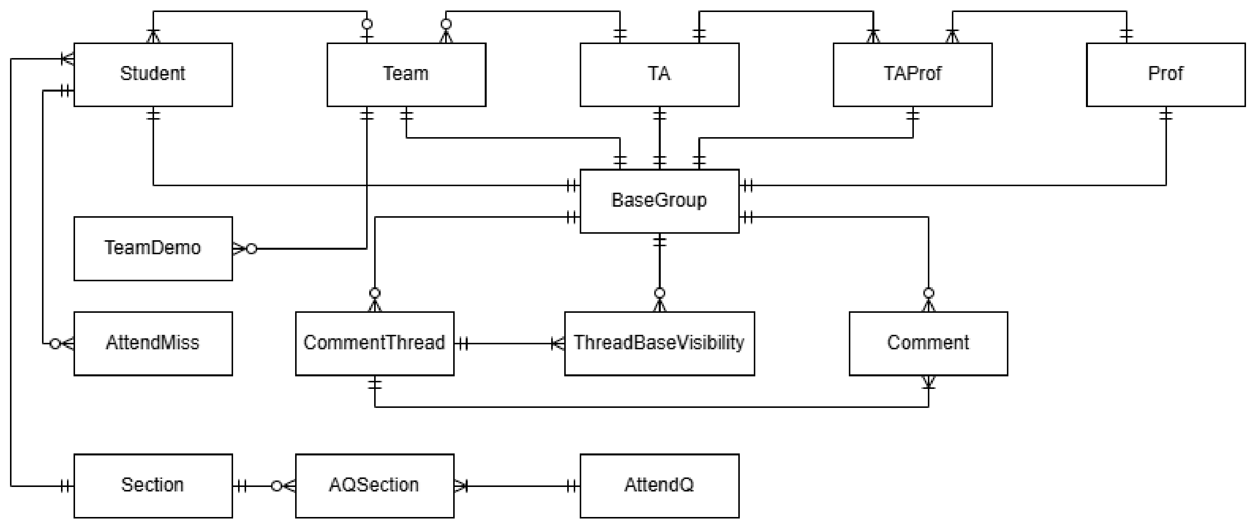
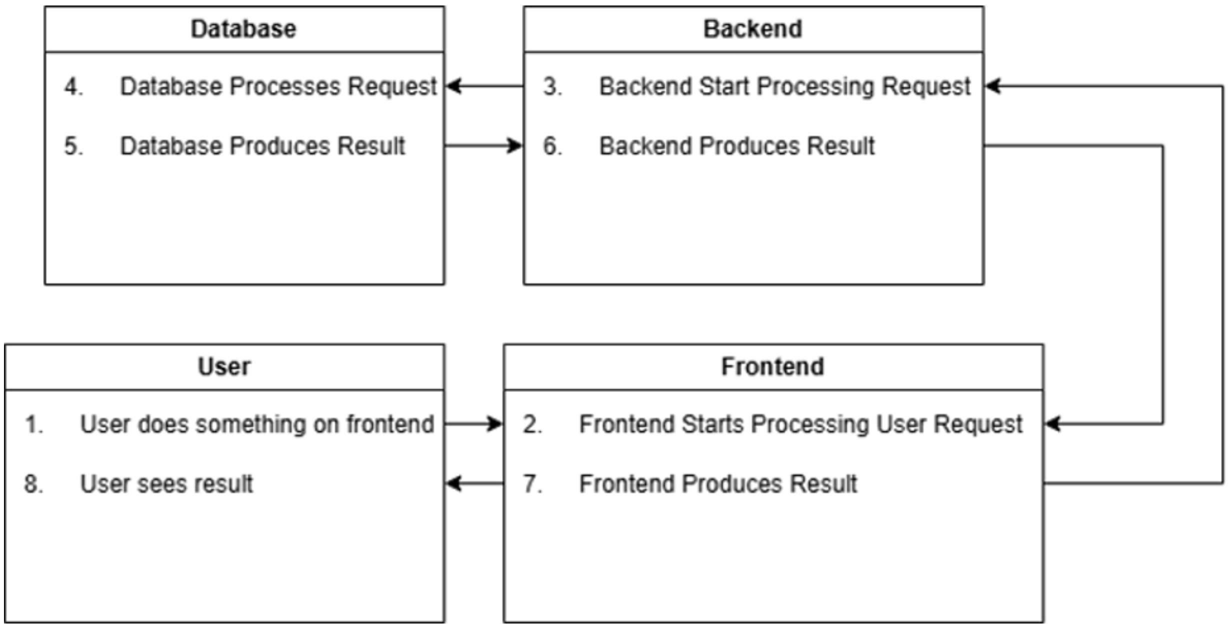


Figure 4 – ER Diagram

The database has been designed to separate unrelated data as much as possible. Thus, we have different tables for each of our user types (Students, Teaching Assistants, and Professors). We use the Team table to group students by team. Connected to it is the TeamDemo table, which tracks team progress throughout the semester. We also have three tables dedicated to Attendance (i.e., AttendMiss, AttendQ, and AQSection) to keep track of Missed Attendance, Attendance questions, and which questions map to which sections. We have also designed functionality to store comments and comment threads, with the remaining tables being used to simplify relationships between the tables.

In order to connect the frontend to the backend, we created an Application Connectivity diagram to elucidate how data will flow. Connections plan to be established once our frontend and backend have initial pages and tables done. This connection will allow continuous updates in the backend to be represented in the frontend.



Database
4. Database Processes Request
5. Database Produces Result
Backend
3. Backend Start Processing Request
6. Backend Produces Result
User
1. User does something on frontend
8. User sees result
Frontend
2. Frontend Starts Processing User Request
7. Frontend Produces Result

Figure 5 – Application Connectivity Diagram

As for security, we plan to encrypt all data flowing in and out of our dashboard. It is very important that we follow FERPA and other academic safety standards throughout our dashboard. In order to prevent outside infiltration of our data, we plan to encode all data coming in and out of our dashboard. To help this, we also plan on fetching academic data as requested, instead of storing it within our dashboard. As for within the dashboard, we are making sure our page designs respect the privacy of our users by not including grades on the home page or in large fonts.

4.3.3 Functionality

Our dashboard is intended to be used by professors, teaching assistants, and students of Computer Science 3090. Everyone will have access to the dashboard. We plan to give the professor access first, so set up the dashboard with assignments and student profiles. From there, teaching assistants will have access to make any changes to their groups if necessary. Once the semester starts, the professor will give the link of our website to students so that they can use the dashboard throughout the semester.

Our dashboard is designed to work for multiple semesters, meaning that it will have a reset function in which the professor will need to input student/group information prior to the start of each semester. All data collected throughout the semester will be stored in the backend, for future use if needed.

We plan to have a function where the professor can input a list of student names and their groups, in which the dashboard will assign each student a profile within their group.

Once the dashboard is completed, the professor will have full control over the dashboard. This will allow the professor to make any changes to the dashboard as desired. The professor must input student information at the beginning of the semester, and wrap up the semester dashboard, in order for the dashboard to work to its best capabilities.

4.3.4 Areas of Concern and Development

The current design and implementation of our project satisfies all users greatly. The professor is also our client, making it easy for us to adjust the dashboard to fit his needs. Most of our group has taken the course prior to this semester, and the feedback from their experience is allowing us to meet all student needs as well. If the design does not fit the needs of the users in the future, the professor has full access to make adjustments as needed.

Our primary concern involves our dependance on Canvas. Most of the data that we are representing throughout our dashboard is being called from Canvas. If Canvas has issues, then our dashboard will have issues. We brainstormed keeping data stored in our dashboard in the case that if Canvas is down, users will still have the most up to date information. After further consideration, we decided it was best to call information when needed in order to have our dashboard be more secure and ensure the privacy of our students.

Our immediate plan for this solution relies on our faith in Canvas, Over our group's time at Iowa State, we have witnessed few Canvas malfunctions. In the case that Canvas is down, it is usually fixed immediately. For the sake of our users, while it is not ideal to have an outage of our dashboard, only Canvas data will be unavailable. All other data will still be represented throughout our Dashboard. In this case that Canvas is back running, users simply need to refresh their website to pull the most recent data from Canvas.

4.4 Technology Considerations

The primary technologies we've chosen to use include Next.JS for the frontend, Node.js with TypeScript for the backend, Amazon Web Services (AWS) for hosting, and an SQL database for data management. For the frontend framework, we selected Next.JS because it is a more modern evolution of

React. Additionally, Next.JS provides several advantages including: simplified routing capabilities, more efficient page loading through server-side rendering, and improved development experience through built-in optimizations. These features allow us to create a more responsive and user-friendly dashboard while maintaining clean and maintainable code.

The backend utilizes Node.js with TypeScript, a choice that creates an ideal development environment since both frontend and backend will be using JavaScript-based technologies. TypeScript adds strong typing to JavaScript, which helps prevent runtime errors and improves code maintainability. This combination allows our team to work more efficiently as we can share code and knowledge across both frontend and backend development.

For hosting, we plan to use AWS in production and a server provided by Iowa State University for development and testing. This approach offers several benefits: the university servers provide a controlled testing environment that closely mirrors the real-world setting where the dashboard will be used, while AWS deployment gives our team valuable experience with industry-standard cloud services. AWS also offers scaling capabilities and reliable uptime, which are crucial for a production level application.

Our database choice of SQL over NoSQL was driven by the relational nature of our data. The dashboard primarily deals with structured data with clear relationships between entities (students, groups, assignments, etc.). SQL databases are much better at handling such relationships and maintaining data integrity through features like foreign keys and transactions. This makes it easier to maintain consistency across different aspects of the dashboard, such as ensuring student data correctly relates to their respective groups and assignments.

These technology choices involve certain trade-offs. While Next.JS offers many modern features, it has a steeper learning curve compared to traditional React. The use of TypeScript adds development overhead but provides better code safety. AWS introduces additional complexity and cost considerations compared to simpler hosting solutions, but the benefits of reliability and scalability outweigh these concerns. The SQL database choice might make some types of data changes more complex compared to NoSQL solutions, but the benefits of data integrity and relationship management are worth it for our application.

4.5 Design Analysis

So far, we have completed an initial mockup of our dashboard. We have pages completed for our home page, and secondary pages. Although we have them completed, they are still continuously being updated to adapt to our client's needs. We have yet to connect to the backend, as we are waiting for permissions to access data from applications such as Canvas, CATme, and Gitlab. We have the frontend, backend, and security planned out in detail, but have yet to fully implement a working dashboard. We are in coordination with the IT department in order to gain access to the applications data.

Our initial design has worked very well for us. Since our project follows both an agile and waterfall management style, we completed initial detailed designs for each component of our project. We are now in the stages of updating each component to work simultaneous with the others, while also adding a few features weekly based on our client's desires. We have not experienced any setbacks as of now. All of our designs have functioned as expected but we are constantly working on making them better.

Our plans for future implementation involve creating the backend with proper calls for data. Once that is completed, we must connect the frontend to the backend. Once connected, we can continue to work on perfecting both ends. We plan to start security implementation once we have a functioning backend as security is not necessary until we have data flowing through our dashboard. Security will also be continuously updated as necessary. Once the frontend is connected to the backend, we will continue with our waterfall management style by working on all ends at the same time, ensuring our dashboard is as efficient as possible.

5 Testing

Testing for our project is uniquely designed. Since our project takes on a waterfall and agile management style, we had to adjust how we categorize tests. Our original prototype testing includes Figma mockups of the frontend design and schematics of the backend organization. The best testing style for our project at this stage includes adding features to our dashboard and then verifying that they work correctly without causing harm to already implemented features. Additionally, after we've verified the new features don't impact others, we present the features to our client to ensure they are worth including or if changes should be made. Therefore, testing is done continuously throughout our development stages.

Our testing plan is unique because we do not follow the traditional prototypes, testing stages, and adjustments. Instead, we have an original prototype that we followed when setting up our initial website. Once the website base was completed, we started testing by adding features on the frontend and adjusting our backend to help ensure data is correctly organized for the frontend. Manual testing is completed after every additional feature is added. This ensures that an additional feature being implemented does not interfere with any previously implemented features. If errors occur in either testing stage, we focus on resolving the issue before continuing development of the feature. Further unit and site reliability testing will be completed once our dashboard is developed to a sufficient state that the client is happy with. This is to avoid spending too much time on tests that could become obsolete depending on what features we choose to include in the final form of the dashboard. Initial security testing was completed when adding login authorization and verification to our dashboard. Further security testing related to data security will be tested once our dashboard if near completion.

Due to the nature of our project being a web application, other testing such as environmental testing, safety testing, and field testing, are not applicable. We are taking all necessary precautions when it comes to functional, performance, reliability, and security testing. If any future additions to our dashboard require different types of testing, we will address the best way to test our dashboard components at that time.

5.1 Unit Testing

Our testing plan does not specifically include unit testing given our current stage. Our dashboard is currently tested through manual testing during and after development of new features. Additionally, we are planning on focusing on testing for user capability and reusability testing later on. Due to the large number of potential users (professors, teaching assistants, and students) we must make sure our dashboard is capable of handling a large number of users consistently. While we haven't spent a significant amount of time on figuring out how to do this quite yet, there are several tools that exist to stress test websites that we can investigate further in the future.

Additionally, Dr. Mitra has requested that our dashboard be able to reset and allow for use in future semesters while still storing past semesters' data. In order to complete this request, we must test the dashboard by inserting synthetic data of a semester's data, compress the data, store the data, and reprogram the dashboard to allow for new inputs (names, groups, identification numbers, etc.). This will likely be accomplished using functional testing to ensure that archiving and resetting the database functions as expected and without any data loss.

6 Implementation

At the current development stage of our project, the design and implementation activities are the same. Thus, we have detailed all our activities within section “4 Design” above.

7 Ethics and Professional Responsibility

Our team approaches engineering ethics and professional responsibility with a stakeholder-centered ideology, with a particular focus on data privacy and community benefit. We define engineering ethics in our project as the principles guiding our development decisions, especially regarding data handling and user privacy. Professional responsibility includes our commitment to delivering a reliable, secure, and practical dashboard for Dr. Mitra, TAs, and future COM S 309 students.

To uphold these principles, we have taken the following steps. First, we maintain strict data handling protocols, including role-based access control, ensuring that sensitive information is accessible only to authorized users. Second, we will actively work with the users of the dashboard during development through regular meetings and iterative development, allowing us to continually improve our dashboard and align it with actual user needs and expectations. Third, we focus on efficient development practices by optimizing our server infrastructure and resources used and implementing practical data storage solutions that balance functionality with security. These steps ensure that our project not only meets technical requirements but also adheres to ethical standards.

7.1 Areas of Professional Responsibility/Code of Ethics

Area of Responsibility	Definition	Relevance to IEEE Code of Ethics
Work Competence	Reliable, timeliness, high quality, and professional work.	“to uphold the highest standards of integrity, responsible behavior, and ethical conduct in professional activities.”
Financial Responsibility	Economically efficient resources and economically reasonable products.	“to avoid unlawful conduct in professional activities, and to reject bribery in all its forms.”
Communication Honesty	Truthful and understandable work.	“to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others.”
Health, Safety, Well-Being	Minimal safety, health, and well-being risks.	“to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment.”
Property Ownership	Respect for property, ideas, and client information.	“to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others.”
Sustainability	Protect the environment and nature.	“to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment.”
Social Responsibility	Beneficial products and services for society.	“to improve the understanding by individuals and society of the capabilities and societal implications of conventional and

		emerging technologies, including intelligent systems.”

Our team is performing very well in social responsibility. Social responsibility is very relevant to our project because our dashboard is centered around helping individuals associated with the course Computer Science 3090. Our approach focuses on providing the best solution for the Computer Science 3090 community, putting their needs first. Our approach upholds ethical and professional responsibilities by being respectful and private with the community’s data.

Our team is performing not so well in sustainability. Sustainability is not very relevant to our project besides ensuring that we are using the least number of resources (servers) as possible. Our approach focuses on using one server while calling data from the other applications instead of storing said data in order to minimize storage needs. While we cannot change this approach for the better, we are committed to continuing to evaluate how to build our database off of one server with minimal storage capacity.

7.2 Four Principles

	Beneficence	**Nonmaleficence**	**Respect for Autonomy**	**Justice**
Public Health, Safety, and Welfare	Reduces stress of finding class resources.	Does not require users to adjust.	Users can choose their involvement level.	All users have availability to use.
Global, Cultural, and Social	Designed for Computer Science 3090 community.	Users are not required to use.	Users can choose to use.	All users have availability to use.
Environmental	Minimal resources used.	No download required.	Users can access simply with a web address.	Causes no harm to the environment.
Economic	Minimal resource costs.	No purchase required.	Free for all users.	Easily accessible for Computer Science 3090 community.

The broader context-principle pair that is most important to our project is beneficence in relation to global, cultural, and social. Our primary goal of our dashboard is to create an application that will benefit all individuals involved in Computer Science 3090 by providing easier access to organized data, materials, and resources. In order to achieve this, we must ensure that our dashboard does not cause any harm to our users, but rather only making access to their information easier. We are creating this dashboard with the primary focus of making their lives easier in relation to the course.

The broader context-principle pair that is lacking in our project is beneficence in relation to economics for the school. We are adding a new tool to the Iowa States systems just for one class and this will add more management, complexity, and room for something to go wrong. Rather than using previously purchased and currently implemented tools we are building something new that is based off some of those tools. Of course, this is the dashboard is the deliverable and primary focus of this project, but it inherently lacks in this context-principle. We make up and justify this deficit with the benefits found in global, cultural, social, and public safety and welfare. This tool will greatly improve the lives of all people in Com S 3090 from professor down to students, by making a free easily accessible and improved tool to manage the class.

7.3 Virtues

Three virtues that are extremely important to our team are honesty, justice, and compassion. We find that these virtues relate most to our dashboard as its main goal is to help individuals involved with Computer Science 309. We have, and will continue to, act in the best interest of our client and his students in order to provide a dashboard that benefits the Computer Science 309 community. Acting in consideration to honesty, courage, and compassion can be seen throughout our dashboard in every decision especially our privacy and security initiatives.

Bradley has demonstrated curiosity throughout this project. He continuously jumped into any role needed throughout this project and helped out on both the front end and back end. Curiosity is important to Bradley, and most of all our group, because he is able to learn about both ends of the project and help his teammates when needed. One virtue that Bradley has not demonstrated greatly throughout this project is creativity. While Bradley is very knowledgeable, he completes his parts to what the group decides without adding a unique perspective. While this is typically a bad characteristic, since we are working collaboratively on this project, it has worked in favor of our group.

Breckin has demonstrated respect throughout this project. He is always considering our team members and the members of the Computer Science 309 community. Respect is important to Breckin as he prides himself in being a reliable and understanding team member. One virtue that Breckin has not demonstrated greatly throughout this project is trust. Breckin is very confident in his own abilities to the point that he does not willingly allow other team members to work parallel with him. Thankfully, he has opened his trust up to a few team members so we can all be collaborative on the project.

Kat has demonstrated commitment throughout this project. She is always willing to work on the project even parts outside of her scope. She has not feared learning new things in order to make sure our project is pristine. Commitment is important to Kat because she takes pride in the work she completes. One virtue that Kat has not demonstrated greatly throughout this project is contentment. Kat is never satisfied with the work she completes and constantly strives to improve it. While this is typically a strong characteristic, it has caused slight timing issues for our group deadlines.

Ria has demonstrated service throughout this project. She is always willing to take on the documentation and presentation assignments. Service is important to Ria because she wants to make sure anyone outside of our team is able to understand the scope of our project, and hopefully learn from what we complete. One virtue that Ria has not demonstrated greatly throughout this project is helpfulness. Ria does not have a vast amount of previous experience with the knowledge needed for this project. While she has been very willing to learn, she does not bring easy answers to the table.

Thomas has demonstrated determination throughout this project. He is always working on the front end and updating the team of what he has completed. Determination is important to Thomas because he always wants to make sure he is completing his work to the best of his ability. One virtue that Thomas has not demonstrated greatly throughout this project is cleanliness. Thomas is very intelligent, so his code is strong, but it is hard for another member to understand his logic because he does not comment throughout his code. Although this could be catastrophic, Thomas has begun to comment throughout his code so that when we hand over the project to the client, updates will be easier to navigate.

Varun has demonstrated gentleness throughout this project. He is always helping other team members on their parts since he possesses prior experience with back end development. Gentleness is very important to Varun because he wants to make sure he is helping his team members succeed whenever possible. One virtue that Varun has not demonstrated greatly throughout this project is patience. While Varun is not angered by his impatience, if group members have not completed their parts, even if prior to the deadlines, Varun will step in and complete it instead. While this is not very helpful for our group since we have never been behind on deadlines, we appreciate Varun's tenacity to get work done.

Curiosity, respect, commitment, service, determination, and gentleness have all being greatly influential in the project that we have completed together. Individual creativity, trust, contentment, helpfulness, cleanliness, and patience are all works in progress throughout our team. Although each member stated a virtue that they feel was lacking throughout our project, our group strongly believes that all of these virtues were not harmful to our group dynamic or project success. We have all adjusted our work styles in order to create a solid team that benefits off of our individual strength.

8 Closing Material

8.1 Conclusions

Our team has made significant progress in developing the COM S 3090 dashboard. We have successfully established the foundational architecture, including the frontend interface with Next.JS, backend infrastructure with Node.js and TypeScript, and initial security precautions. Our primary goals remain focused on creating a secure platform that integrates data from multiple sources while maintaining user privacy and providing role-specific features for professors, TAs, and students.

The best plan of action to achieve our goals includes:

1. Continuing backend and frontend integration
2. Implementing modern security measures for data protection and user privacy
3. Conducting testing across all user roles and features
4. Deploying the dashboard on AWS
5. Creating documentation for future maintenance and updates

Several factors have impact our timeline and implementation:

- Delays in receiving API access and permissions for various platforms (Canvas, GitLab)
- Complexity in managing different user role requirements
- Coordination challenges with multiple team members working on different components

For future iterations, we would recommend:

- Working on gaining API access earlier in the development process to figure out what's feasible

Despite these challenges, our team has kept up steady progress towards delivering a dashboard that meets Dr. Mitra's requirements while prioritizing user privacy and data security. Overall, the experience so far has provided valuable insights into full-stack development, modern web security, and project management.

8.2 References

[1]“TypeORM - Amazing ORM for TypeScript and JavaScript (ES7, ES6, ES5). Supports MySQL, PostgreSQL, MariaDB, SQLite, MS SQL Server, Oracle, WebSQL databases. Works in NodeJS, Browser, Ionic, Cordova and Electron platforms.,” typeorm.io. https://typeorm.io/

[2] Shadcn/ui, “Introduction,” shadcn/ui, https://ui.shadcn.com/docs

[3] jwt, “JWT.IO - JSON Web Tokens Introduction,” jwt.io, 2024. https://jwt.io/introduction

[4] “NextAuth.js,” next-auth.js.org. https://next-auth.js.org/

[5] A. Deshpande, A. Kulkarni, D. Stone, R. Cherian, and N. Eltoum, “Group 9 -309-Dashboard.” Accessed: Dec. 05, 2024. [Online]. Available: https://seniord.cs.iastate.edu/2023-Jan-09/files/inline-files/402C%20Demo2.pdf

[6] “Rest api,” GitLab, https://docs.gitlab.com/ee/api/rest/ (accessed Dec. 5, 2024).

[7] A. Deshpande, A. Kulkarni, D. Stone, R. Cherian, and N. Eltoum, “309 Dashboard -Team 9.” Accessed: Dec. 05, 2024. [Online]. Available: https://seniord.cs.iastate.edu/2023-Jan-09/files/inline-files/402C%20Demo%201_0.pdf

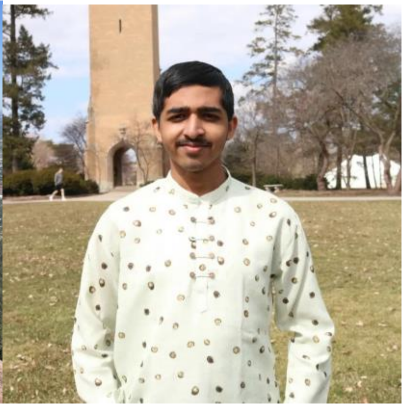
[8] Vercel, “Next.js by Vercel - The React Framework,” nextjs.org, 2024. https://nextjs.org

[9] A. Hanrahan, E. Peterson, N. Hoss, E. Danforth, and L. Thunga, “309 Dashboard.” Accessed: Dec. 05, 2024. [Online]. Available: https://seniord.cs.iastate.edu/2022-May-07/files/inline-files/402%20Final%20Presentation.pdf

9 Team

9.1 Team Members

Bradley Gaines *Computer Engineering*	Breckin Bartels *Software Engineering*	Kat Christofferson *Cyber-Security Engineering*



Ria Patel *Electrical Engineering* *(emphasis in Computer Engineering)*	Thomas Payton *Software Engineering* *(minor in Cyber-Security Engineering)*	Varun Jain *Computer Engineering*

9.2 Required Skill Sets For Your Project

1. Frontend Development
    1.1. React.js for web development
    1.2. React Native for mobile development
    1.3. HTML/CSS and UI/UX knowledge for responsive design
2. Backend Development
    2.1. TypeScript for backend development
    2.2. Database design knowledge
    2.3. AWS service knowledge
3. Database Management
    3.1. SQL/MySQL for data storage and retrieval
4. Project Management
    4.1. Agile knowledge
    4.2. Timeline management
    4.3. Communication skills
5. Testing
    5.1. Unit, integration, performance, and user acceptance testing

9.3 Skill Sets Covered By The Team

- Frontend Development
    - React.js for web development – **Breckin Bartels, Thomas Payton**

    - React Native for mobile development – **Thomas Payton**
    - HTML/CSS and UI/UX knowledge for responsive design – **Breckin Bartels, Thomas Payton**
- Backend Development
    - TypeScript for backend development – **Bradley Gaines, Kat Christofferson, Varun Jain**
    - Database design knowledge – **Varun Jain**
    - AWS service knowledge – **Bradley Gaines, Kat Christofferson**
- Database Management
    - SQL/MySQL for data storage and retrieval – **Bradley Gaines, Kat Christofferson, Varun Jain**
- Project Management
    - Agile knowledge – **Ria Patel**
    - Timeline management – **Ria Patel**
    - Communication skills – **Whole Team**
- Testing
    - Unit, integration, performance, and user acceptance testing

9.4 Project Management Style Adopted By the Team

The Computer Science 3090 Dashboard project embodies a waterfall and agile combination management style. The waterfall style allows a scheduled order of tasks for the beginning of our project including designing the frontend, designing the backend, implementing the frontend, implementing the backend, and implementing security. This allows us to create a functional, but simple, dashboard by the end of the first semester of our capstone. The agile style enables feedback from our original implementation to be implemented and adjusted in the dashboard during the later parts of our project, primary during the second semester of our capstone. We track progress through diagrams and the dashboard application. Diagrams are created and updated through Figma to represent our up-to-date feature design. The dashboard shows our up-to-date feature implementations as well.

9.5 Initial Project Management Roles

Team leadership roles:

- Bradley Gaines: Backend co-lead
- Breckin Bartels: Frontend co-lead
- Kat Christofferson: Security lead
- Ria Patel: Project manager
- Thomas Payton: Frontend co-lead
- Varun Jain: Backend co-lead

9. 6 Team Contract

- Team members: Bradley Gaines, Breckin Bradley, Kat Christofferson, Ria Patel, Thomas Payton & Varun Jain
- Team Procedures:
    - Whole Team Meeting:
        - Type: In person
        - Day: Thursday
        - Time: 2pm
        - Location: Iowa State University Library
        - Frequency: Weekly
    - Frontend Meeting:
        - Type: Virtual
        - Day: No set day

        - Time: No set time
        - Location: No set location
        - Frequency: Whenever necessary
    - Backend Meeting:
        - Type: In person
        - Day: Monday
        - Time: 1pm
        - Location: Iowa State University Library
        - Frequency: Weekly
    - Client/Advisor Meeting:
        - Type: In person
        - Day: Thursday
        - Time: 3pm
        - Location: Dr. Mitra's office
        - Frequency: Biweekly
- Preferred method of communication: Discord.
- Decision making policy: consensus.
- Record keeping procedures: files in CyBox and dashboard project in website.
- Participation Expectations:
    - Everyone is expected to attend the weekly meetings and meetings with our advisor/client unless notified otherwise.
    - Everyone is expected to complete their tasks assigned during sprints. If someone expects a delay, needs additional time, or needs assistance, they should reach out to the rest of the team.

    - Everyone is expected to communicated updates, delays, and/or progress on tasks. Everyone is expected to notify the team if they will not be in attendance for a meeting.
    - Everyone should be made aware of and have a say in team decisions before they are made. Everyone should be present, agree upon, and be informed on tasks before they are assigned.
- Leadership:
    - Roles for each team member:
        - Bradley Gaines: Backend component design, cross component integration, testing.
        - Breckin Bartels: Frontend component design, cross component integration, testing.
        - Kat Christofferson: Backend component design, cross component integration, testing.
        - Ria Patel: Backend component design, cross component integration, project manager.
        - Thomas Payton: Frontend component design, cross component integration, testing.
        - Varun Jain: Backend component design, cross component integration, client interaction.
    - Strategies for supporting and guiding the work of all team members include talking all issues through with team members, making sure that everyone knows what resources to use and what the weekly/long-term expectations are.
    - Strategies for recognizing the contributions of all team members include having separate roles for each team member with different responsibilities, and ensuring that everyone has accomplished what they agreed to complete.
- Collaboration and inclusion:
    - Skills, expertise, and unique perspectives each team member brings to the team:
        - Bradley Gaines: frontend, backend

        - Breckin Bartels: UI/UX, frontend, backend
        - Kat Christofferson: security, backend
        - Ria Patel: frontend, backend
        - Thomas Payton: UI/UX, frontend
        - Varun Jain: frontend, backend, security
    - Strategies for encouraging and supporting contributions and ideas from all team members include all team members helping provide a comfortable group setting while still encouraging others to participate to their best ability. All team members bring different strengths, and we understand that the best ideas come from listening to each other.
    - Procedures for identifying and resolving collaboration or inclusion issues include allowing uncomfortable team members to address issues in individual conversations with the other team member helping to resolve issues.
- Goal setting, planning, and execution:
    - Team goals for the semester focus on having a working web application with a sufficient number of features as requested by Dr. Mitra.
    - Strategies for planning and assigning individual and team work include assigning roles and then establishing sprint goals as well as stories to accomplish within each sprint. We will continually check-in with Dr. Mitra to ensure we're on track and working on the correct tasks.
    - Strategies for keeping on task include having stand up meetings weekly. We will go over the tasks that everyone is working on, any issues that are blocking a task from being complete, and what we will be working on in the future.
- Consequences for not adhering to the team contract:
    - We will handle infractions of any of the obligations of this team contract by discussing the infraction with the team and come up with a plan to resolve the issue depending on the severity. If the issue causes a disruption in the project timeline, come up with a plan to get ourselves back on track. Depending on how frequent issues arise, we will talk with advisors/instructors if needed.

    - If infractions continue, we will address the issue with our advisors and instructor to come up with a solution.

- *I participated in formulating the standards, roles, and procedures as stated in this contract. I understand that I am obligated to abide by these terms and conditions, I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.*
    - Bradley Gaines, 12/5/24
    - Breckin Bartels, 12/5/24
    - Kat Christofferson, 12/5/24
    - Ria Patel, 12/5/24
    - Thomas Payton, 12/5/24
    - Varun Jain, 12/5/24