

# MID-SEMESTER PROJECT REPORT

*COMPUTER SCIENCE LEARNING CENTER WEBSITE PROJECT*

*CRINGE CODERS TEAM*

**AUSTIN BAILEY  
MYA BELL  
LINDSEY LANGDON  
NOLAN "OG" GREGORY**

MID-SEMESTER PROJECT REPORT FOR THE  
COMPUTER SCIENCE LEARNING CENTER WEBSITE PROJECT

DR. HARVEY SIY; COLLEGE IS&T  
UNIVERSITY OF NEBRASKA - OMAHA  
CAPSTONE COURSE  
OCTOBER 12TH, 2023

*Report prepared in partial fulfillment of CSCI 4970: Computer Science Capstone Project*

# Contents

---

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Motivation . . . . .	3
1.2	Similar Applications . . . . .	3
1.3	Theoretical Foundations of the Project . . . . .	3
1.4	Software Engineering Challenges . . . . .	4
1.5	System Context . . . . .	4
1.5.1	Subject Facet . . . . .	4
1.5.2	Usage Facet . . . . .	4
1.5.3	IT System Facet . . . . .	4
1.5.4	Development Facet . . . . .	5
1.5.5	Legal & Ethical Facet . . . . .	5
<b>2</b>	<b>Requirements</b>	<b>6</b>
2.1	Goals . . . . .	6
2.1.1	Goal Model . . . . .	6
2.1.2	Goal Descriptions . . . . .	6
2.2	Functional Requirements . . . . .	6
2.3	Wireframes . . . . .	7
2.4	Object Model . . . . .	7
2.5	Nonfunctional Requirements . . . . .	8
2.6	Analysis Requirements . . . . .	8
2.7	External System Interfaces . . . . .	8
2.8	Requirements Not Implemented . . . . .	8
<b>3</b>	<b>Architecture and Design</b>	<b>9</b>
3.1	Overview . . . . .	9
3.2	Logical Decomposition View . . . . .	9
3.3	Technology Stack . . . . .	10
3.3.1	Languages . . . . .	10
3.3.2	Libraries & Services . . . . .	10
3.4	Deployment View . . . . .	10
3.5	Development View . . . . .	10
3.6	Data View . . . . .	11
3.7	Concurrency View . . . . .	11
3.8	Execution Flow View . . . . .	11
3.9	Screenshots . . . . .	11
3.10	Summary of Design Changes . . . . .	12
3.11	Design Modularity and Extensibility . . . . .	12
<b>4</b>	<b>Implementation</b>	<b>13</b>
4.1	Directory Structure . . . . .	13
4.2	Technical Issues Encountered . . . . .	13
4.3	User-Reported Bugs . . . . .	13
<b>5</b>	<b>Testing</b>	<b>14</b>
5.1	Test Plan . . . . .	14
5.2	Obtaining Realistic Test Data . . . . .	14
5.3	Tests Conducted . . . . .	14
5.4	Test Results . . . . .	14
5.5	Automated Test Outputs . . . . .	14
5.6	Tests Not Conducted . . . . .	14
<b>6</b>	<b>Summary</b>	<b>15</b>

6.1	Summary of Project Organization	15
6.2	Outcome of Risks	15
6.3	Milestone Summary	15
6.4	Lessons Learned	15
6.5	Future Extensions	15
<b>7</b>	<b>Local and Global Impacts</b>	<b>16</b>
<b>8</b>	<b>Appendix</b>	<b>17</b>
8.1	Project Proposal	18
8.2	Project Plan	35
8.3	Initial Requirements	57
8.4	Context Document	59
8.5	Setup/build/installation/deployment Instructions	62
8.6	Team Meeting Notes	62
8.7	Individual Journals	63
8.7.1	Austin Bailey	63
8.7.2	Mya Bell	63
8.7.3	Nolan Gregory	64
8.7.4	Lindsey Langdon	64
8.8	Transcripts of Client Meetings	64
8.9	Transcript of Initial Client Meeting	64
<b>9</b>	<b>References</b>	<b>76</b>

# CHAPTER 1

---

## Introduction

---

### 1.1 Motivation

The Computer Science Learning Center (CSLC), is a student service hosted through the College of Information Science & Technology (IS&T) for academic assistance and tutoring. The CSLC accepts tutoring by appointment or walk-ins Monday through Saturdays, and distance tutoring over Zoom. Student-tutors are paid for their services, and the University has administrators to oversee and manage CSLC staffing levels, work shifts, and payroll.

The existing CSLC portal has incomplete or missing features, functionality, and services. The current ticket submission system functions through a Google Form tied to a spreadsheet with the resulting student information.

To better meet the needs of the CSLC, the Cringe Coders Team proposes a full stack refresh of the existing technology stack. The ticketing system at present will be rebuilt to provide tutors and students with a more efficient way for exchanging essential information. This encompasses sharing specifics about the class assistance needed, articulating the present comprehension of the problem, identifying the respective course instructor, and the level of priority. The new portal will be able to facilitate this seamless sharing of this information between tutors and students.

### 1.2 Similar Applications

Last semester's capstone course had a group lay the foundations of the project work, including a basic website with some more advanced functionality like authentication. While a CSLC tutoring portal exists, the current solution does not have the full suite of functionality that the client desires. This includes a ticketing for students to request tutoring, dynamic web page elements to show new tickets without the need to refresh the page, UI/UX considerations, and report generation for tracking ticket trends.

Core elements of infrastructure were established with the prior group, but some supplemental features were given work-around solutions. Case and point is the ticketing and reporting functionality. The current implementation is to use Google Forms to submit tickets, which allows for a limited (albeit incorrectly and partially formatted) report.

In light of the limitations of current solutions, the Cringe Coders Team takes this as a motivation to overhaul the UI/UX and add further back-end functionality, as is highlighted in our Motivation section via a proposed solution outlined in Chapter 3: Architecture and Design.

### 1.3 Theoretical Foundations of the Project

The back-end database structure established by the prior group will likely need modification and updates, and thus necessitates additional modeling and design of structured tables and potentially databases (Connolly & Begg, 2014). Additionally, the data needs to be accessible via REST API, such that Django REST can make API calls to the database, for both read and write operations (Fielding, 2000).

Though the CSLC is labeled 'Computer Science', the CSLC services all IS&T majors and students. Consequently, security practices should be implemented in the development and deployment of a CSLC portal. This includes protecting against common security flaws, such as Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF) (OWASP, 2021). Much of this security will derive from proper implementation of front-end developments through proper project management of Django (Django Project, 2021).

## 1.4 Software Engineering Challenges

We anticipate only trivial computer science challenges will arise in the scope of work. The only challenge revolves around the synchronization of the front-end and back-end for real-time updates. Successfully establishing seamless communication between these elements, alongside the portal's relational sqlite database, is mentioned because the implementation is novel to the Cringe Coders Team.

This is not to suggest we anticipate no challenges. The success of this project hinges on everyone's familiarity with the technology stack in use. For some team members, and potentially the entire team, this means dedicating time to become acquainted with new frameworks and programming languages. The capacity to acquire new skills is essential in the world of software engineering. Also, given that not all team members have worked together before, getting an understanding on each other's work processes presents an additional challenge. In essence, the success of this project not only hinges on the team's technical expertise but also equally relies on the team's collaboration and communication.

## 1.5 System Context

The CLSC portal will occupy the intersection of faculty and student academic interests. The current and proposed solution are designed to facilitate tutoring by, and for, IS&T students. Below Cringe Coders Team outlines the various facets of the system's context.

### 1.5.1 Subject Facet

Kyle runs the CLSC, which tutors students via a ticketing system. Students request help in a certain subject or class, and the CLSC pairs them with an appropriate tutor. The CLSC is the official tutoring center for IS&T. Kyle uses ticket history to run a periodic report, detailing the classes that students struggle with most. Tutors at the center are paid employees and they use the ticket queue to find students they're qualified to help.

### 1.5.2 Usage Facet

The primary stakeholder is Kyle Reestman, who is the director of the Computer Science Learning Center and therefore oversees all of the operations within. The users of the system will include the students, student-tutors, teachers, and other staff members associated with the computer science department of the University of Nebraska at Omaha.

### 1.5.3 IT System Facet

Submitting a ticket on the CSLC Portal will be one of the first interactions a student will have with the CSLC. Therefore, it is crucial that this application will seamlessly fit into the already established technological environment. For context, the CSLC is tied to the University of Nebraska-Omaha. As a result, the reconstructed portal will have to conform with UNO's current authentication system and present a polished and professional user interface.

#### 1.5.4 Development Facet

The main development concern with this portal is the familiarity with technology stack. To create a web application up to professional standards, everyone must be well-versed with the languages at hand. The team's overall proficiency with the technology stack will not only enhance the development process but also contribute to the long-term success and sustainability of the project.

#### 1.5.5 Legal & Ethical Facet

We foresee no legal requirements, as the ingestion of data is self-reported by students, and only the tutors and Kyle can see it. Since this project will not integrate with other systems, basic security and architecture protections will suffice. The only possible issue we foresee is FERPA regulations limiting the projects integrating or sharing data with other systems.

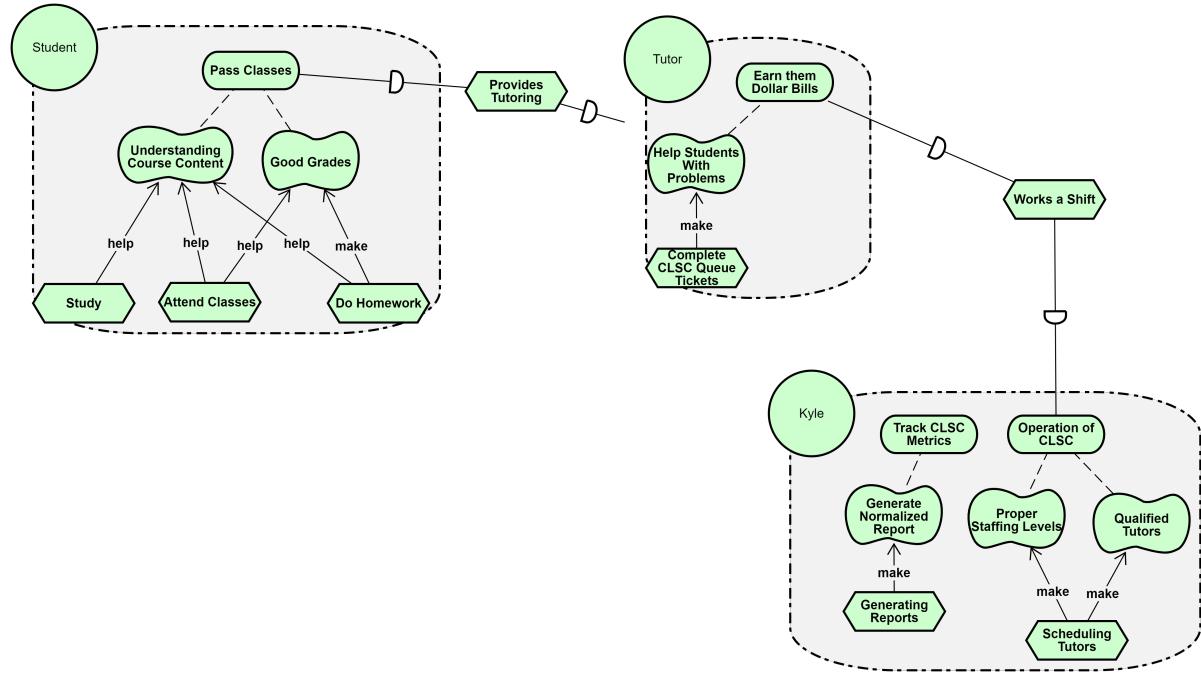
Ethically, we believe any value-added aspects, such as 'About Us' or mission and vision statements have been addressed by the last capstone group. Our focus ethically will be around presenting a clean and professional interface for students. We believe that a haphazard or poorly designed interface will reflect poorly on the CLSC and serve to discredit the tutors by showing poor computer science principles on the CLSC website.

# CHAPTER 2

# Requirements

## 2.1 Goals

### 2.1.1 Goal Model



### 2.1.2 Goal Descriptions

The Student is one of the users of the system. Each student has an overarching goal of passing their classes, which the CSLC portal is designed to aid in. Students will use the portal as a tool that will connect them with people who will help them to understand the content they need help with.

The Tutor is employed by the CSLC to tutor students. One of the main functionalities of the CSLC portal is the ticketing system. This function makes it easy for tutors to plan and organize what they need to do.

Kyle oversees the CSLC, so he has a goal of ensuring the smooth operation of the whole thing. The CSLC portal is designed to aid him in this goal by making it easy to schedule tutors and track what is actually going on.

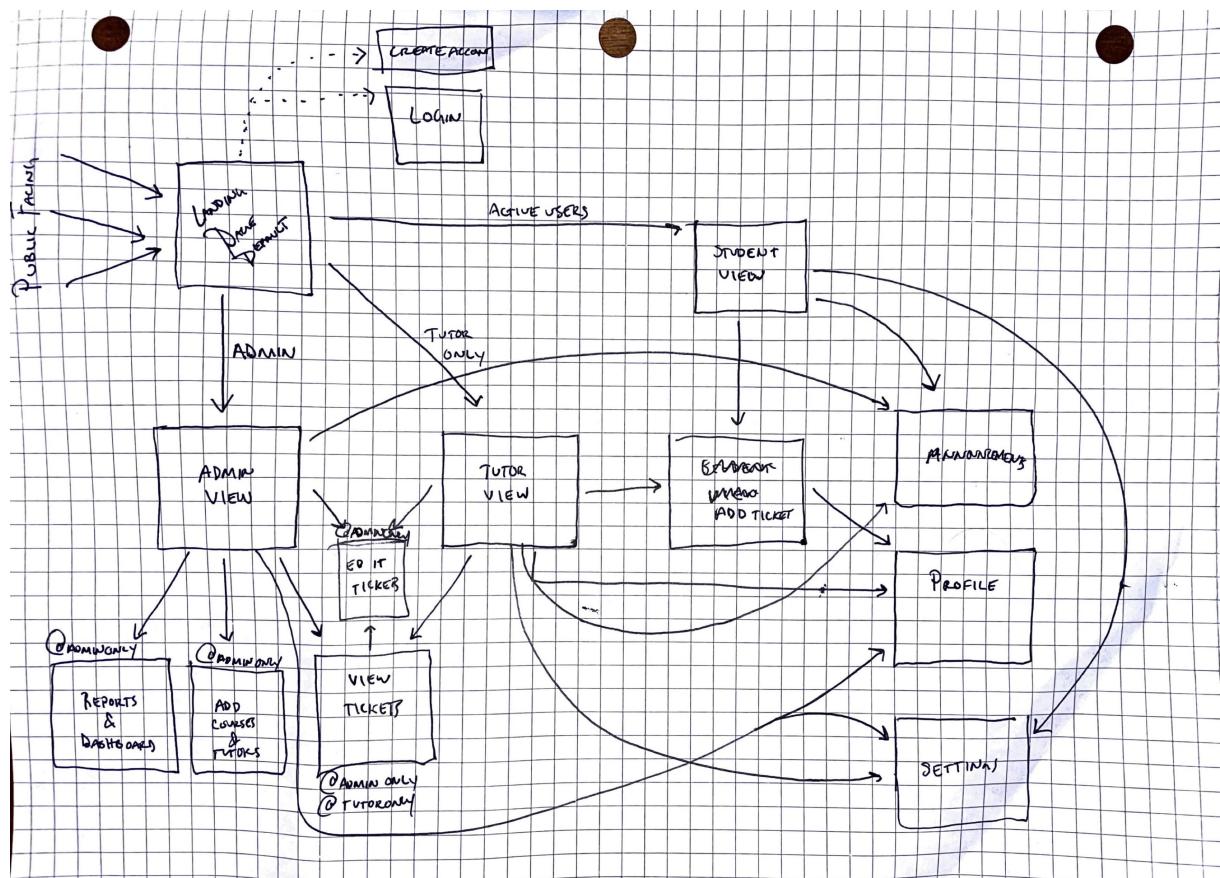
## 2.2 Functional Requirements

1. Must have. As the director of the Computer Science Learning Center, I want the CSLC portal to have efficient ticket management, so that tutoring requests can be managed.
2. Must have. As a tutor at the Computer Science Learning Center, I want to be able to interact

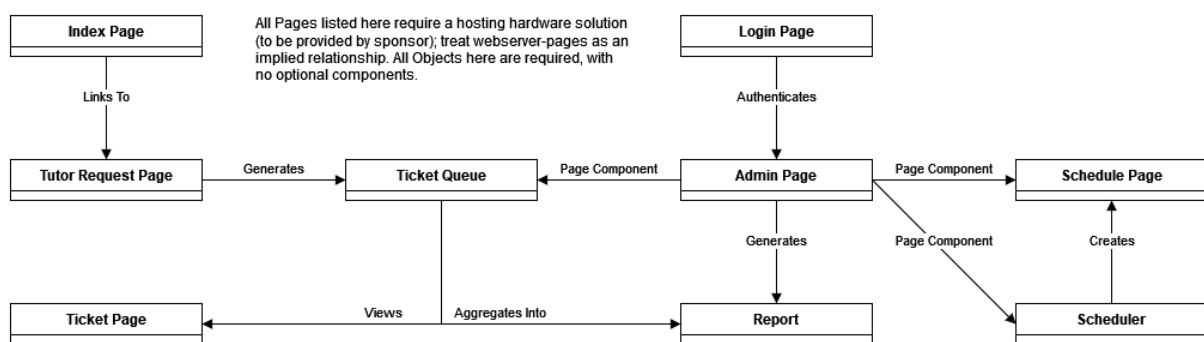
with students and administrators regarding tutoring requests and progress as well as close and edit tickets, so that I can effectively provide help to students.

3. Must have. As a student I want to be able to make tutoring requests and see available tutors so that I can receive the help I need.
4. Should have. As an administrator, I want to be able to view and manage student requests, so that I can monitor progress and efficiency within the CSLC.
5. Could have. As a student, I want to be able to see how tutors are rated among other students, so that I can make an informed decision about choosing a tutor.
6. Won't have. As a user of the CSLC portal, I do not want to have to log in and authenticate redundantly, so that ticket submission is time efficient.

## 2.3 Wireframes



## 2.4 Object Model



## 2.5 Nonfunctional Requirements

1. Must have: Evaluating the CSLC Tutoring Portal application on ease of use from the point of view of the students in the context of submitting a ticket for help.
  - Q1: How easy is it to navigate the user interface?
    - M1. Task time
    - M2. Stroke count
    - M3. Problem count
  - Q2: How quickly can tasks be accomplished?  
Reuse M1, M2, M3
  - Q3: Do users feel satisfied after using the portal?  
M4. Conduct surveys to find the percentage of students reporting high satisfaction after using the portal
2. Should have: Assess the CSLC Tutoring Portal application on maintainability from the point of view of the developers in the context of ensuring that the application is up to date.
  - Q1: How maintainable are the languages in the stack?  
M1. Check how often the languages are updated
  - Q2: How many dependencies are there?  
M2. Count the dependencies

## 2.6 Analysis Requirements

No Analysis requirements have been determined as of yet. (TBD in Final Draft)

## 2.7 External System Interfaces

- API specifications of existing systems that interact with your product, e.g., the name of the application being accessed, services called, parameters passed, outputs returned, set up instructions (e.g., API keys)
- file formats your product is required to use, e.g., if you had to input from, or output to a file with a certain csv format  
(add content as needed)

## 2.8 Requirements Not Implemented

(TBD in Final Draft)

# CHAPTER 3

## Architecture and Design

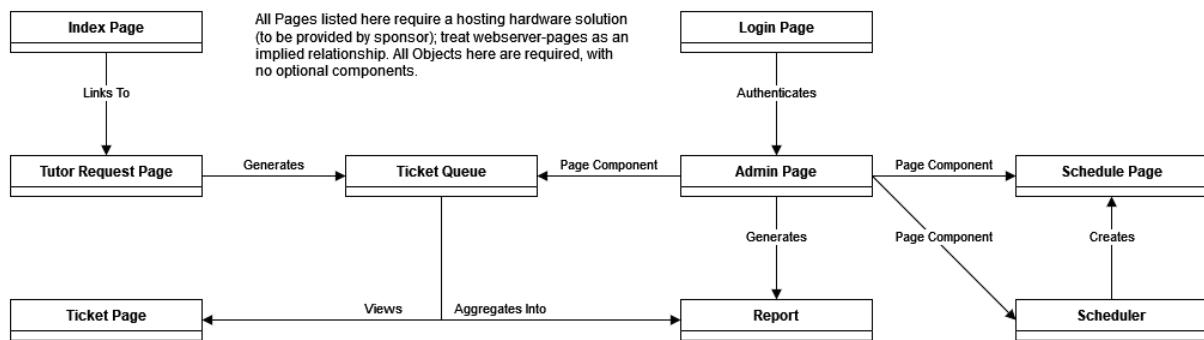
### 3.1 Overview

The CSLC portal will comprise a handful of web pages hosted on the <https://cslc.unomaha.edu/> subdomain, with integration of a back-end database. at a high-level technology-agnostic view, we have the following design requirements

1. A guest, index, or public page that links to the portals resources.
2. A ticket or tutoring request submission resources.
3. A tutor or employee view that allows access to the ticket queue and shift schedules of tutors.
4. An admin view that allows the client to export reports based on ticket history, and to configure work schedules.
5. A back-end database to facilitate the ticketing processes.

Subsequent sections will cover these designs through the various lens, culminating in a more concrete description of architecture.

### 3.2 Logical Decomposition View



Cringe Coders Team provides a simplified UML diagram of both the public and administrative functions of the CLSC portal. Note that the decomposition has merged the employee and admin pages, as the Cringe Coders Team's current intent is to implement RBAC roles associated with authenticated users that reveal components of the admin page. This will ideally minimize redundant pages and code.

For the sake of simplicity and ease of conveying design, the database subsystem has been omitted, as it will integrate with almost every other subsystem.

to see the Cringe Coders Team's accompanying functional requirements, see the Functional Requirements section.

### 3.3 Technology Stack

The Cringe Coders Team intends to replace the current portal with a full refresh of the technology stack. Additionally, will add database tie-ins to the portal to replace the current Google Form solution.

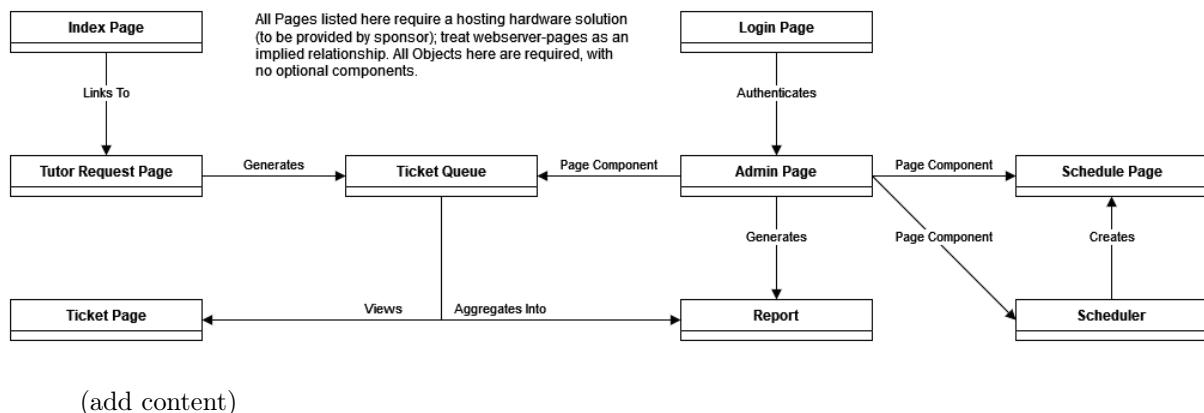
#### 3.3.1 Languages

For this project, we anticipate using the following languages: Python, JavaScript, HTML, and CSS. We will also use the following frameworks and libraries: Django, React, and Tailwind.

#### 3.3.2 Libraries & Services

The front-end will use the React library and Tailwind CSS to enable better UI/UX. While our back-end will be written using the Django ORM. We will treat it as a REST back-end by using the Django REST framework. This will allow us to populate the webpage with real time database information via API calls. We will use Postman to test our API, and we will use sqlite as our database. Additionally, we will deploy our application using docker for containerization, Kubernetes for automating deployment/management of containerized applications, and we will host our application on Azure, pending any client objections

### 3.4 Deployment View

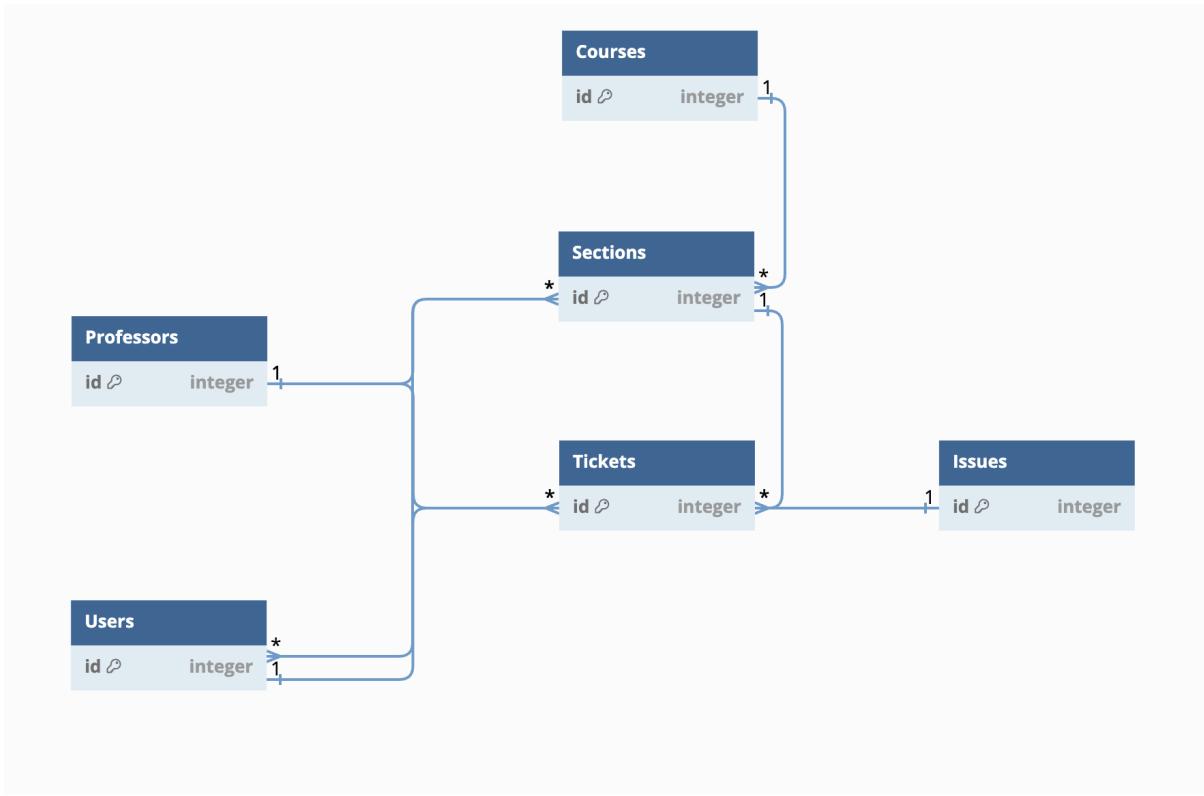


### 3.5 Development View

- discuss how the code is organized and how this organization is related to the previously identified subsystems
- discuss the role of the framework(s) used and how your code interacts with them
- identify which code modules or components were reused (e.g., open source libraries, existing code from another project, etc.)
- describe API calls to external services and third-party libraries
- corresponds to C4
- Links to an external site. Levels 3 and 4 diagrams
- 1 or more UML class diagrams are required
  - if there are 15 or less modules/classes, show interconnections between classes
  - if there are more than 15, use a UML package diagram to show the hierarchy

(add content)

## 3.6 Data View



Our current back-end relational database uses six tables. A More detailed version, and additional details will be made available in the final report.

## 3.7 Concurrency View

Cringe Coders Team does not anticipate work that would involve this section. we include this section as a placeholder and an acknowledgement that future work may re-scope us into this area.

## 3.8 Execution Flow View

- discuss illustrative scenarios that show data and control flow through the application modules to deliver a particular service
- 1 or more UML sequence diagrams are required

(add content)

## 3.9 Screenshots

(TBD in Final Draft)

## 3.10 Summary of Design Changes

- summarize as applicable how the design has evolved from the original concept to the final version and explain the reason for the changes  
(TBD in Final Draft)

## 3.11 Design Modularity and Extensibility

A stretch goal currently slated for work (but unlikely to be accomplished within the scope of our work) would be an integration with the registrar's office to automatically populate current IS&T courses. Both the client and the Cringe Coders Team is unaware of an existing structured data source.

Future work would involve scoping and understanding the data's formatting requirements. A likely implementation would be an ingestion of a csv file through the admin portal. Via REST API, the csv data would populate a table in the back-end that would populate elements such as drop down menus for selecting a course on tutoring tickets.

# CHAPTER 4

---

## Implementation

---

### 4.1 Directory Structure

(TBD in Final Draft)

### 4.2 Technical Issues Encountered

(TBD in Final Draft)

### 4.3 User-Reported Bugs

(TBD in Final Draft)

# CHAPTER 5

---

## Testing

---

### 5.1 Test Plan

This test plan outlines the high level testing strategy for the CSLC Tutoring Portal. The primary objective of this testing plan is to ensure the robustness, functionality, and performance of the web application. This plan includes both integration testing and end-to-end testing, enabling our team to deliver a high performing web application that will meet the demands of the end users.

In the integration testing scenario, we need to test the API integration and the component integration. With testing the API integration, we need to verify that the front-end properly communicates with the REST API back-end by testing API calls and responses for different endpoints. For component integration testing, Jest can be employed to ensure that the different web components of the application work together as expected. As Jest is primarily known for its unit testing, we plan to develop unit tests for our components to help identify potential bugs during the testing phase.

The end-to-end testing approach in this plan is designed to comprehensively evaluate the users' journey from start to finish. We plan on using Selenium to stimulate users actions such as clicks and inputs to asses the application's performance. Additionally, Selenium can be used for cross-browser compatibility testing to ensure that our application works across a variety of browsers. Similar to the integration testing, the end-to-end testing process is focused on identifying any potential bugs during the testing phase, ensuring the delivery of a dependable and user-friendly application.

### 5.2 Obtaining Realistic Test Data

(TBD in Final Draft)

### 5.3 Tests Conducted

(TBD in Final Draft)

### 5.4 Test Results

(TBD in Final Draft)

### 5.5 Automated Test Outputs

(TBD in Final Draft)

### 5.6 Tests Not Conducted

(TBD in Final Draft)

# CHAPTER 6

---

## Summary

---

### 6.1 Summary of Project Organization

(TBD in Final Draft)

### 6.2 Outcome of Risks

(TBD in Final Draft)

### 6.3 Milestone Summary

(TBD in Final Draft)

### 6.4 Lessons Learned

(TBD in Final Draft)

### 6.5 Future Extensions

(TBD in Final Draft)

# CHAPTER 7

---

## Local and Global Impacts

---

(TBD in Final Draft)

# CHAPTER 8

---

# Appendix

---

*This Page is intentionally left blank*

Project Proposal

## PROJECT PLAN DOCUMENTATION

*Cringe Coders*

AUSTIN BAILEY  
MYA BELL  
NOLAN GREGORY  
LINDSEY LANGDON

COMPREHENSIVE PROJECT OVERVIEW  
FOR THE CSLC PORTAL

DR. HARVEY SIY; COLLEGE IS&T  
UNIVERSITY OF NEBRASKA - OMAHA  
CAPSTONE COURSE  
AUGUST 31ST, 2023

## Project Proposal

---

# Contents

---

<b>1 Client Information</b>	<b>2</b>
1.1 Organization . . . . .	3
1.2 Point of Contact . . . . .	3
<b>2 Motivation</b>	<b>4</b>
2.1 Context . . . . .	5
2.2 System Needs . . . . .	5
2.3 Proposed Solution . . . . .	5
2.4 Client Role . . . . .	5
<b>3 Project Description</b>	<b>6</b>
3.1 Project Overview . . . . .	7
3.2 Features and Services . . . . .	7
3.3 Languages . . . . .	7
3.4 Technology Stack . . . . .	7
3.5 Documentation . . . . .	7
3.6 Potential Features and Services . . . . .	7
<b>4 Implementation Strategy</b>	<b>8</b>
4.1 Website Refresh . . . . .	9
4.2 Back-End Database & New Features . . . . .	9
4.3 Security Controls . . . . .	9
<b>5 Computer Science Challenges</b>	<b>10</b>
5.1 Challenges . . . . .	11
<b>6 Computer Engineering Challenges</b>	<b>12</b>
6.1 New Frameworks & Technologies . . . . .	13
<b>7 Team Members</b>	<b>14</b>
7.1 Background on the Cringe Coders . . . . .	15
7.2 Austin Bailey . . . . .	15
7.3 Mya Bell . . . . .	15
7.4 Nolan Gregory . . . . .	15
7.5 Lindsey Langdon . . . . .	16

Project Proposal

CHAPTER 1

---

## Client Information

---

## Project Proposal

---

CHAPTER 1. CLIENT INFORMATION**1.1 Organization**

The Computer Science Learning Center (CSLC), is a student service hosted through the College of Information Science & Technology (IS&T) for academic assistance and tutoring. The CSLC accepts tutoring by appointment or walk-ins Monday through Saturdays, and distance tutoring over Zoom. Student-tutors are paid for their services, and the University has administrators to oversee and manage CSLC staffing levels, work shifts, and payroll.

**1.2 Point of Contact**

Kyle Reestman is the director of the Computer Science Learning Center. Apart from his directorial position, he also serves as an instructor for CIST1400, an introductory computer science course. This course is designed to provide students with a comprehensive overview of programming basics utilizing the Python programming language. Furthermore, as the director of the CSLC, Kyle oversees the orchestration and management of the tutoring center's operations. He coordinates the tutors schedules, supervises the tutors, and ensures that the center's tutors are well-equipped with the necessary teaching materials. Through his capacities as both an educator and a director, Kyle Reestman guarantees the smooth operation of the CSLC while fostering students' development in computer science.

For the duration of this capstone, Kyle Reestman will be the primary point of contact. Secondaries, subject matter expert, and other consultation contacts will be at the delegation of Kyle Reestman, to be determined as necessary throughout the course of the project.

Project Proposal

CHAPTER 2

---

**Motivation**

---

## Project Proposal

---

CHAPTER 2. MOTIVATION

## 2.1 Context

The Computer Science Tutoring Portal is a comprehensive web application designed to streamline and enhance the tutoring experience within the Computer Science Learning Center. This portal serves as a centralized platform for students, tutors, and administrators to manage tutoring requests, track progress, and facilitate efficient communication. The application's key features include ticket management, tutor and course administration, and seamless interaction between students, tutors, and administrators.

## 2.2 System Needs

The existing CSLC portal has incomplete or missing features, functionality, and services. The current ticket submission system functions through a Google Form tied to a spreadsheet with the resulting student information.

## 2.3 Proposed Solution

To better meet the needs of the CSLC, the Cringe Coders proposes a full stack refresh of the existing technology stack. The ticketing system at present will be rebuilt to provide tutors and students with a more efficient way for exchanging essential information. This encompasses sharing specifics about the class assistance needed, articulating the present comprehension of the problem, identifying the respective course instructor, and the level of priority. The new portal will be able to facilitate this seamless sharing of this information between tutors and students.

## 2.4 Client Role

The client holds the important role of qualifying goals, and acceptance criteria. Throughout this project the client will be the source of truth and final authority for product quality and completeness. This power requires that the client provides periodic input into goals and the scope of work. To prevent redundant or extraneous work, a scheduled periodic meeting and an ad hoc communication channel are required during the continuous delivery process.

Project Proposal

CHAPTER 3

---

## Project Description

---

## Project Proposal

---

CHAPTER 3. PROJECT DESCRIPTION

### 3.1 Project Overview

The initial conception of the scope of work involves a complete overhaul of front-end and back-end frameworks. Feature augmentations will be added as part of the refresh.

### 3.2 Features and Services

Concurrent with the replacement the current portal with a full refresh of the CSLC portal, Cringe Coders will add database tie-ins to the portal to replace the current Google Form solution.

### 3.3 Languages

For this project, we anticipate using the following languages: Python, JavaScript, HTML, and CSS. We will also use the following frameworks and libraries: Django, React, and Tailwind.

### 3.4 Technology Stack

The front-end will use the React library and Tailwind CSS to enable better UI/UX. While our back-end will be written using the Django ORM. We will treat it as a REST back-end by using the Django REST framework. This will allow us to populate the webpage with real time database information via API calls. We will use Postman to test our API, and we will use sqllite as our database. Additionally, we will deploy our application using docker for containerization, Kubernetes for automating deployment/management of containerized applications, and we will host our application on Azure, pending any client objections.

### 3.5 Documentation

Lastly, we will use a combination of Sphinx and Autodoc for our technical documentation. We will also use LATEX to write our documentation (such as this document you are reading).

### 3.6 Potential Features and Services

At present no further work is scoped. This will be revisited and addressed according to client needs.

Project Proposal

CHAPTER 4

---

## Implementation Strategy

---

## Project Proposal

## CHAPTER 4. IMPLEMENTATION STRATEGY

**4.1 Website Refresh**

This transformation starts with a redesign of the website's user interface. This shift requires moving away from the CSLC's current reliance on Google Forms. It involves constructing a new front-end using react, a widely recognized Javascript framework, and Tailwind CSS. This front-end will be intricately linked to a back-end using the Django REST framework and a database server.

**4.2 Back-End Database & New Features**

Later continuous delivery cycles will focus on creating a sqllite database and integrating it with the refreshed portal website.

**4.3 Security Controls**

At all stages of delivery, security controls will be implemented at an architectural level, and periodically audited for validity and completeness.

Project Proposal

CHAPTER 5

---

## Computer Science Challenges

---

## Project Proposal

---

CHAPTER 5. COMPUTER SCIENCE CHALLENGES

### 5.1 Challenges

We anticipate only trivial computer science challenges will arise in the scope of work. The only challenge revolves around the synchronization of the front-end and back-end for real-time updates. Successfully establishing seamless communication between these elements, alongside the portal's relational sqlite database, is mentioned because the implementation is novel to the Cringe Coders team.

Project Proposal

CHAPTER 6

---

## Computer Engineering Challenges

---

## Project Proposal

---

CHAPTER 6. COMPUTER ENGINEERING CHALLENGES**6.1 New Frameworks & Technologies**

The success of this project hinges on everyone's familiarity with the technology stack in use. For some team members, and potentially the entire team, this means dedicating time to become acquainted with new frameworks and programming languages. The capacity to acquire new skills is essential in the world of software engineering. Also, given that not all team members have worked together before, getting an understanding on each other's work processes presents an additional challenge. In essence, the success of this project not only hinges on the team's technical expertise but also equally relies on the team's collaboration and communication.

Project Proposal

CHAPTER 7

---

**Team Members**

---

## Project Proposal

CHAPTER 7. TEAM MEMBERS**7.1 Background on the Cringe Coders**

The Cringe Coders consists of four members with a range of hard skills; soft skills; professional computer science experience; and, experiences in adjacent fields of information assurance and mathematics. Based on internal deliberation among team members, we have aligned ourselves to the roles most apt for individual skill sets.

**7.2 Austin Bailey**

*Client Liaison  
Information Security Analyst  
Quality Assurance Tester*

Austin Bailey is responsible for thoroughly testing the application to identify vulnerabilities, usability issues, and performance bottlenecks. Austin will create and execute test cases, provide feedback to the development team, and ensure a high-quality final product. Likewise, Austin will simulate real-world cyberattacks to identify vulnerabilities and weaknesses. Austin will help our team improve our security posture and protect sensitive information by uncovering any potential security flaws.

**7.3 Mya Bell**

*Requirements Analyst  
Quality Assurance Tester  
Client Liaison*

Mya Bell will oversee the development process, and ensure that the project stays on track, objectives are met, and communication flows smoothly between team members. Likewise, Mya will facilitate communication and collaboration between leadership and team players to ensure a successful outcome. Lastly, Mya will create and execute test cases to ensure a high-quality final product.

**7.4 Nolan Gregory**

*Architect  
Technical Lead  
Back-End Developer  
Dev-Ops and Database*

Nolan Gregory is responsible for making technical decisions and guiding the development process. Nolan will provide technical expertise, review code, and ensure that the architecture and technologies chosen align with the project's goals. Nolan will handle all server-side logic, database management, and API calls. Nolan will build the core functionalities of the application, including ticket management, user authentication, and communication features. Nolan will design and maintain the database structure, ensuring efficient data storage, retrieval, and integrity. Lastly, Nolan will set up the deployment infrastructure, manage continuous integration/continuous deployment (CI/CD) pipelines, and ensure smooth deployment and scaling of the ticketing portal.

## Project Proposal

---

CHAPTER 7. TEAM MEMBERS**7.5 Lindsey Langdon***UI/UX Lead**Front-End Developer**Mobile Design Lead*

Lindsey Langdon is responsible for creating an intuitive and visually appealing user interface. Lindsey will design wireframes, mockups, and prototypes, ensuring a user-friendly experience and consistent branding. Lindsey will be responsible for implementing the user interface and ensuring that the application is responsive and visually engaging. They work closely with the UI/UX Designer to translate design concepts into functional front-end components. Lindsey will ensure that the front-end can run on mobile or tablet devices, as well as on desktop systems, with a focus on mobile-first design.

Project Plan

## PROJECT PLAN DOCUMENTATION

*CRINGE CODERS*

AUSTIN BAILEY  
MYA BELL  
NOLAN GREGORY  
LINDSEY LANGDON

COMPREHENSIVE PROJECT OVERVIEW  
FOR THE CSLC PORTAL

DR. HARVEY SIY; COLLEGE IS&T  
UNIVERSITY OF NEBRASKA - OMAHA  
CAPSTONE COURSE  
AUGUST 25TH, 2023

## Project Plan

---

# Contents

---

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Project Organization</b>	<b>4</b>
2.1	Austin Bailey . . . . .	5
2.2	Mya Bell . . . . .	5
2.3	Nolan Gregory . . . . .	5
2.4	Lindsey Langdon . . . . .	6
<b>3</b>	<b>Risk Analysis</b>	<b>7</b>
3.1	Data Breach or Unauthorized Access . . . . .	8
3.2	Application Vulnerabilities . . . . .	8
3.3	Denial of Service (DoS) Attacks . . . . .	8
3.4	Inadequate Authentication and Authorization . . . . .	8
3.5	Poorly Managed Sessions and Cookies . . . . .	8
3.6	Data Loss or Corruption . . . . .	9
3.7	Lack of Input Validation . . . . .	9
3.8	Third-Party Dependencies . . . . .	9
<b>4</b>	<b>Requirements</b>	<b>10</b>
4.1	Software Requirements . . . . .	11
4.1.1	Integrated Development Environments . . . . .	11
4.1.2	Version Control System . . . . .	11
4.1.3	Languages . . . . .	11
4.1.4	Front-end and UI/UX . . . . .	11
4.1.5	Back-end and Database . . . . .	11
4.1.6	Deployment . . . . .	11
4.1.7	Documentation . . . . .	11
4.1.8	Management . . . . .	11
<b>5</b>	<b>Work Breakdown Structure</b>	<b>12</b>
5.1	Role Assignment . . . . .	13

## Project Plan

<b>CONTENTS</b>	<b>2</b>
5.2 Project Initiation . . . . .	13
5.3 Requirements Gathering and Analysis . . . . .	13
5.4 System Design . . . . .	13
5.5 Front-End Development . . . . .	13
5.6 Back-End Development . . . . .	14
5.7 Database Development . . . . .	14
5.8 Testing and Quality Assurance . . . . .	14
5.9 Security and Performance Optimization . . . . .	14
5.10 Documentation . . . . .	14
5.11 Deployment and Launch . . . . .	14
5.12 Project Closure . . . . .	15
<b>6 Project Schedule</b>	<b>16</b>
6.1 Milestone One . . . . .	17
6.2 Milestone Two . . . . .	17
6.3 Milestone Three . . . . .	18
6.4 Milestone Four . . . . .	18
6.5 Wrapping Up . . . . .	19
<b>7 Monitoring and Reporting Mechanisms</b>	<b>20</b>
7.1 Project Management Tools . . . . .	21
7.2 Version Control and Collaboration . . . . .	21
7.3 Meetings and Client Communication . . . . .	21
7.4 Document Sharing . . . . .	21
7.5 Code Reviews, Testing, and QA . . . . .	21

Project Plan

## CHAPTER 1

### Introduction

The Computer Science Tutoring Portal is a comprehensive web application designed to streamline and enhance the tutoring experience within the Computer Science Learning Center. This portal will serve as a centralized platform for students, tutors, and administrators to manage tutoring requests, track progress, and facilitate efficient communication. The application's key features include ticket management, tutor and course administration, and seamless interaction between students, tutors, and administrators.

Project Plan

CHAPTER 2

---

## Project Organization

---

## Project Plan

## CHAPTER 2. PROJECT ORGANIZATION

5

## 2.1 Austin Bailey

*Client Liaison  
Information Security Analyst  
Quality Assurance Tester*

Austin Bailey is responsible for thoroughly testing the application to identify vulnerabilities, usability issues, and performance bottlenecks. Austin will create and execute test cases, provide feedback to the development team, and ensure a high-quality final product. Likewise, Austin will simulate real-world cyberattacks to identify vulnerabilities and weaknesses. Austin will help our team improve our security posture and protect sensitive information by uncovering any potential security flaws.

## 2.2 Mya Bell

*Requirements Analyst  
Quality Assurance Tester  
Client Liaison*

Mya Bell will oversee the development process, and ensure that the project stays on track, objectives are met, and communication flows smoothly between team members. Likewise, Mya will facilitate communication and collaboration between leadership and team players to ensure a successful outcome. Lastly, Mya will create and execute test cases to ensure a high-quality final product.

## 2.3 Nolan Gregory

*Architect  
Technical Lead  
Back-End Developer  
Dev-Ops and Database*

Nolan Gregory is responsible for making technical decisions and guiding the development process. Nolan will provide technical expertise, review code, and ensure that the architecture and technologies chosen align with the project's goals. Nolan will handle all server-side logic, database management, and API calls. Nolan will build the core functionalities of the application, including ticket management, user authentication, and communication features. Nolan will design and maintain the database structure, ensuring efficient data storage, retrieval, and integrity. Lastly, Nolan will set up the deployment infrastructure, manage continuous integration/continuous deployment (CI/CD) pipelines, and ensure smooth deployment and scaling of the ticketing portal.

## Project Plan

### *CHAPTER 2. PROJECT ORGANIZATION*

6

#### **2.4 Lindsey Langdon**

*UI/UX Lead  
Front-End Developer  
Mobile Design Lead*

Lindsey Langdon is responsible for creating an intuitive and visually appealing user interface. Lindsey will design wireframes, mockups, and prototypes, ensuring a user-friendly experience and consistent branding. Lindsey will be responsible for implementing the user interface and ensuring that the application is responsive and visually engaging. They work closely with the UI/UX Designer to translate design concepts into functional front-end components. Lindsey will ensure that the front-end can run on mobile or tablet devices, as well as on desktop systems, with a focus on mobile-first design.

Project Plan

CHAPTER 3

---

**Risk Analysis**

---

## Project Plan

## CHAPTER 3. RISK ANALYSIS

8

**3.1 Data Breach or Unauthorized Access**Threat Level: **Moderate** to **High**

Our team will implement encryption algorithms for sensitive data in transit. We will use a multi-factor authentication for user accounts (*OAuth*). Our team will conduct security reports and vulnerability assessments. Lastly, we will employ access control mechanisms to limit access based on user roles and user permissions.

**3.2 Application Vulnerabilities**Threat Level: **High**

Our team will follow secure coding practices such as: conducting regular security code reviews, implementing input validation and output encoding to prevent common vulnerabilities like SQL injection and cross-site scripting (XSS), and keep libraries and frameworks up to date to address known vulnerabilities.

**3.3 Denial of Service (DoS) Attacks**Threat Level: **Moderate**

Our team will implement rate limiting and CAPTCHA mechanisms to mitigate automated attacks and monitor server and network performance for unusual spikes in traffic.

**3.4 Inadequate Authentication and Authorization**Threat Level: **High**

Our team will implement a strong authentication mechanism, namely OAuth, and enforce role-based access control.

**3.5 Poorly Managed Sessions and Cookies**Threat Level: **Moderate**

Our team will use secure session management techniques, such as session timeouts and unique session identifiers.

## Project Plan

## CHAPTER 3. RISK ANALYSIS

9

**3.6 Data Loss or Corruption**Threat Level: **MODERATE**

Our team will implement regular data backups and ensure data integrity. Likewise, we will use redundancy for pivotal components of the system.

**3.7 Lack of Input Validation**Threat Level: **HIGH**

Our team will implement strict input validation for all user inputs. We will validate and sanitize user inputs to prevent malicious input from causing security vulnerabilities.

**3.8 Third-Party Dependencies**Threat Level: **HIGH**

Our team will carefully review third-party libraries and components for security vulnerabilities, as these libraries can sometimes be the target for attacks.

Project Plan

CHAPTER 4

---

**Requirements**

---

## Project Plan

## CHAPTER 4. REQUIREMENTS

11

**4.1 Software Requirements**

Listed below are the software requirements for the CSLC Tutoring. These requirements are necessary in order to correctly and reliably implement the application. Likewise, these requirements are needed to appropriately develop the codebase.

**4.1.1 Integrated Development Environments**

For this application, we will use several Integrated Development Environments (IDE's). These include Visual Studio Code for full stack development, JetBrains WebStorm for front-end development, and JetBrains PyCharm for back-end development.

**4.1.2 Version Control System**

For this application, we will use Git as our version control system. Similarly, we will use GitHub to track changes and branches across members. We have chosen to use this Version Control System (VCS) as it is simple to use and easy to keep tabs on.

**4.1.3 Languages**

For this application, we will use the following languages: Python, JavaScript, HTML, and CSS. We will also use the following frameworks and libraries: Django, React, and Tailwind.

**4.1.4 Front-end and UI/UX**

Our front-end will be written using the React library. Likewise, we will use Tailwind CSS to develop elegant user interfaces and components.

**4.1.5 Back-end and Database**

Our back-end will be written using the Django ORM. We will treat it as a REST back-end by using the Django REST framework. This will allow us to populate the webpage with real time database information via API calls. We will use Postman to test our API, and we will use SQLite as our database.

**4.1.6 Deployment**

We will deploy our application using Docker for containerization, Kubernetes for automating deployment/management of containerized applications, and we will host our application on Azure.

**4.1.7 Documentation**

We will use a combination of Sphinx and Autodoc for our technical documentation. We will also use L<sup>A</sup>T<sub>E</sub>X to write our documentation (such as this document you are reading).

**4.1.8 Management**

We will use Jira and Trello for managing tasks and tracking project progress. We will communicate through a designated discord server, and use Google as our file distribution network.

Project Plan

CHAPTER 5

---

## Work Breakdown Structure

---

## Project Plan

## CHAPTER 5. WORK BREAKDOWN STRUCTURE

13

**5.1 Role Assignment**

ROLE ASSIGNMENT	
Ticket Management System	Nolan Gregory
User Roles and Authentication	Nolan Gregory & Lindsey Langdon
Tutor and Course Management	Nolan Gregory
Communication and Feedback	Lindsey Langdon & Mya Bell
Analytics and Database Architecture	Nolan Gregory
UI/UX Design	Lindsey Langdon
Front-End Implementation	Lindsey Langdon
Back-End Implementation	Nolan Gregory
Testing and Quality Assurance	Mya Bell & Austin Bailey
AGILE Management	Mya Bell
Cybersecurity Analyst	Austin Bailey
Documentation	Austin Bailey
Deployment and Infrastructure	Nolan Gregory

**5.2 Project Initiation**

- Define project scope and objectives
- Identify stakeholders and establish communication plan
- Create project plan and timeline

**5.3 Requirements Gathering and Analysis**

- Define user stories and use cases
- Gather functional and non-functional requirements
- Conduct interviews with stakeholders to gather detailed requirements

**5.4 System Design**

- High-level architecture design
- Database schema design
- UI/UX wireframing and design
- Design API endpoints and data flow

**5.5 Front-End Development**

- Set up project structure and version control
- Develop user registration and login components
- Create user dashboard for students, tutors, and admins

## Project Plan

### CHAPTER 5. WORK BREAKDOWN STRUCTURE

14

- Implement ticket creation and management UI
- Develop UI for communication features (messaging, feedback)

#### **5.6 Back-End Development**

- Set up server environment and framework
- Develop authentication and authorization logic
- Create API endpoints for user management
- Implement ticket management and linking with tutors, courses, and students
- Develop tutor and course management functionalities
- Integrate messaging and communication features

#### **5.7 Database Development**

- Set up database server
- Create tables for users, tickets, courses, messages, etc.
- Implement relationships between database entities
- Establish data integrity constraints and indexes

#### **5.8 Testing and Quality Assurance**

- Develop unit tests for individual components
- Conduct integration testing of front-end and back-end
- Perform user acceptance testing (UAT)
- Identify and fix bugs and issues

#### **5.9 Security and Performance Optimization**

- Implement encryption for sensitive data
- Conduct security testing and vulnerability assessment
- Optimize database queries for performance
- Implement caching mechanisms for improved responsiveness

#### **5.10 Documentation**

- Create user documentation for students, tutors, and admins
- Document API endpoints and usage
- Prepare deployment and setup guides
- Compile project documentation for future reference

#### **5.11 Deployment and Launch**

- Prepare production environment
- Deploy application to hosting server or cloud platform
- Configure domain and SSL certificates
- Perform final testing in the production environment

## Project Plan

### CHAPTER 5. WORK BREAKDOWN STRUCTURE

15

#### 5.12 Project Closure

- Perform final project review
- Document lessons learned and best practices
- Hand over project to maintenance and support team

Project Plan

CHAPTER 6

---

## Project Schedule

---

## Project Plan

## CHAPTER 6. PROJECT SCHEDULE

17

**6.1 Milestone One**Time to complete: *Four Weeks***1. Project Initiation**

- Define project scope and objectives
- Identify stakeholders and establish communication plan
- Design and develop project plan and timeline

**2. Project Planning**

- Define user stories and use cases
- Gather detailed functional and non-functional requirements
- Conduct product owner meetings for clarification

**3. Project Setup**

- Setup VCS
- Establish environments

**6.2 Milestone Two**Time to complete: *Three Weeks***1. Architecture Design**

- Develop high-level architecture design
- Create database schema design
- Define API endpoints and data flow
- Create API endpoints for user management

**2. Develop Database Schema**

- Develop database model diagram
- Write Database models
- Develop authentication and authorization logic

**3. User-Interface Design and Setup**

- Design UI/UX wireframes and layouts
- Develop UI for communication features
- Implement ticket creation and management UI
- Develop user registration and login components
- Create user dashboard for students, tutors, and admins

**4. Application Logic Design**

- Develop authentication and authorization logic
- Implement ticket management and linking with tutors, courses, and students
- Develop tutor and course management functionalities
- Integrate messaging and communication features

## Project Plan

## CHAPTER 6. PROJECT SCHEDULE

18

**6.3 Milestone Three**Time to Complete: *Three Weeks***1. Testing**

- Develop unit tests for components
- Conduct integration testing of front-end and back-end
- Perform user acceptance testing
- Create API endpoints for user management

**2. Rudimentary Optimization**

- Identify and fix bugs and issues
- Optimize database queries for performance [potential]
- Implement caching mechanisms for responsiveness

**3. Security Perspectives**

- Conduct security testing and vulnerability assessment
- Implement encryption for sensitive data

**6.4 Milestone Four**Time to complete: *Two Weeks***1. Wrap-up Documentation**

- Finalize user documentation for students, tutors, and admins
- Finalize documentation of API endpoints and usage
- Design deployment and setup guides
- Perform final testing in the production environment

**2. Continuous Delivery and Integration**

- Finalize the CI/CD pipeline
- Prepare production environment
- Configure domain and SSL certificates
- Deploy application to hosting server or cloud platform

**3. Security Perspectives**

- Conduct security testing and vulnerability assessment
- Implement encryption for sensitive data

## Project Plan

### CHAPTER 6. PROJECT SCHEDULE

19

## 6.5 Wrapping Up

Time to complete: *< 1 Week*

### 1. KT on Prod

- Conduct training sessions for tutors
- Conduct training sessions for product owner
- Conduct training sessions for application maintainers

### 2. Final Loose Ends

- Document lessons learned and best practices
- Hand over project to owner

Project Plan

CHAPTER 7

---

## Monitoring and Reporting Mechanisms

---

## Project Plan

### CHAPTER 7. MONITORING AND REPORTING MECHANISMS

21

#### 7.1 Project Management Tools

We will use Jira to create tasks, assign responsibilities, set due dates, and track progress. By using Jira, we will be able to manage tasks, prioritize work, and monitor the project status.

#### 7.2 Version Control and Collaboration

We will use a version control system (Git) with GitHub to manage code changes, collaborate on code, and track contributions.

#### 7.3 Meetings and Client Communication

We will schedule regular team meetings to discuss progress, challenges, and upcoming tasks. Likewise, we will conduct daily stand-up meetings for brief updates and address any "blockers".

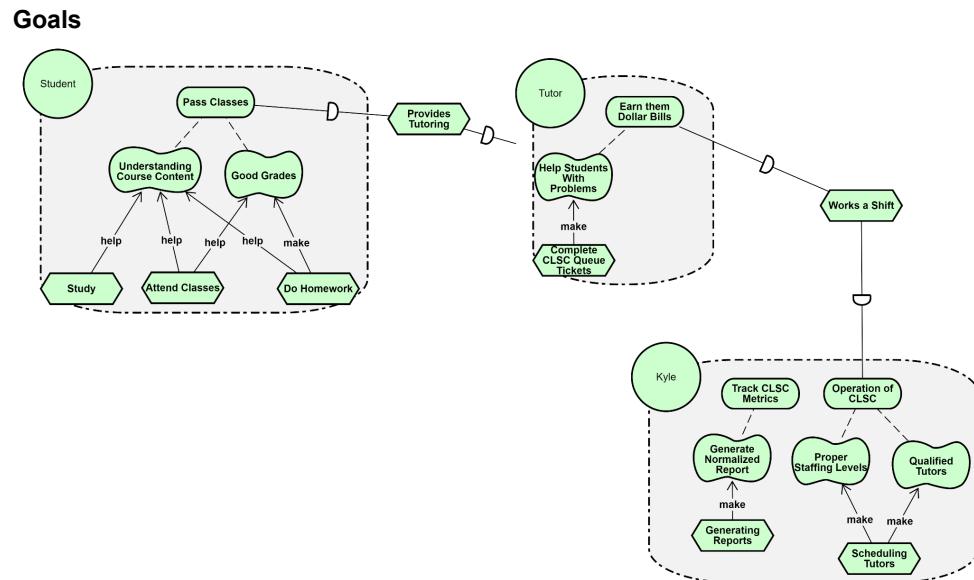
#### 7.4 Document Sharing

We will use Google Drive and Confluence to share project-related documents, reports, and documentation. This will ensure that everyone has access to the latest information.

#### 7.5 Code Reviews, Testing, and QA

We will implement a code review process using GitHub's pull request feature. We will also set up a testing environment and conduct regular testing and quality assurance activities. By setting up our GitHub integration testing, we can ensure each push is valid and will not break the production environment.

## Initial Requirements

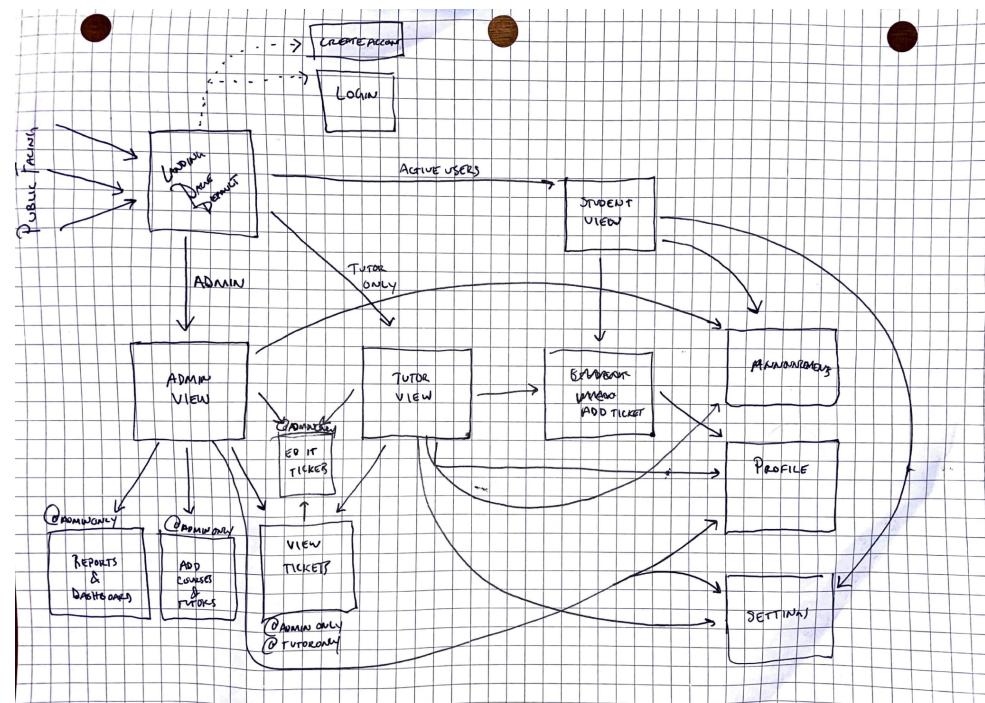


## Functional Requirements

1. Must have. As the director of the Computer Science Learning Center, I want the CSLC portal to have efficient ticket management, so that tutoring requests can be managed.
2. Must have. As a tutor at the Computer Science Learning Center, I want to be able to interact with students and administrators regarding tutoring requests and progress as well as close and edit tickets, so that I can effectively provide help to students.
3. Must have. As a student I want to be able to make tutoring requests and see available tutors so that I can receive the help I need.
4. Should have. As an administrator, I want to be able to view and manage student requests, so that I can monitor progress and efficiency within the CSLC.
5. Could have. As a student, I want to be able to see how tutors are rated among other students, so that I can make an informed decision about choosing a tutor.
6. Won't have. As a user of the CSLC portal, I do not want to have to log in and authenticate redundantly, so that ticket submission is time efficient.

## Initial Requirements

## Wireframes



Context Document

## CONTEXT DOCUMENT

*CRINGE CODERS*

**AUSTIN BAILEY  
MYA BELL  
LINDSEY LANGDON  
NOLAN "OG" GREGORY**

CONTEXT DOCUMENT FOR THE  
COMPUTER SCIENCE LEARNING CENTER WEBSITE PROJECT

DR. HARVEY SIY; COLLEGE IS&T  
UNIVERSITY OF NEBRASKA - OMAHA  
CAPSTONE COURSE  
SEPTEMBER 7TH, 2023

Context Document

---

## Contents

---

Subject Facet . . . . .	2
Usage Facet . . . . .	2
IT System Facet . . . . .	2
Development Facet . . . . .	2
Legal & Ethical Facet . . . . .	2

## Context Document

CONTENTSCONTENTS**Subject Facet**

Kyle runs the CLSC, which tutors students via a ticketing system. Students request help in a certain subject or class, and the CLSC pairs them with an appropriate tutor. The CLSC is the official tutoring center for IS&T. Kyle uses ticket history to run a periodic report, detailing the classes that students struggle with most. Tutors at the center are paid employees and they use the ticket queue to find students they're qualified to help.

**Usage Facet**

The primary stakeholder is Kyle Reestman, who is the director of the Computer Science Learning Center and therefore oversees all of the operations within. The users of the system will include the students, student-tutors, teachers, and other staff members associated with the computer science department of the University of Nebraska at Omaha.

**IT System Facet**

Submitting a ticket on the CSLC Portal will be one of the first interactions a student will have with the CSLC. Therefore, it is crucial that this application will seamlessly fit into the already established technological environment. For context, the CSLC is tied to the University of Nebraska-Omaha. As a result, the reconstructed portal will have to conform with UNO's current authentication system and present a polished and professional user interface.

**Development Facet**

The main development concern with this portal is the familiarity with technology stack. To create a web application up to professional standards, everyone must be well-versed with the languages at hand. The team's overall proficiency with the technology stack will not only enhance the development process but also contribute to the long-term success and sustainability of the project.

**Legal & Ethical Facet**

We foresee no legal requirements, as the ingestion of data is self-reported by students, and only the tutors and Kyle can see it. Since this project will not integrate with other systems, basic security and architecture protections will suffice. The only possible issue we foresee is FERPA regulations limiting the projects integrating or sharing data with other systems.

Ethically, we believe any value-added aspects, such as 'About Us' or mission and vision statements have been addressed by the last capstone group. Our focus ethically will be around presenting a clean and professional interface for students. We believe that a haphazard or poorly designed interface will reflect poorly on the CLSC and serve to discredit the tutors by showing poor computer science principles on the CLSC website.

## 8.5 Setup/build/installation/deployment Instructions

### 8.6 Team Meeting Notes

Sans the initial client meeting, the Cringe Coders Team has felt no sense of urgency or disconnect between us and our client that would necessitate extra-ordinary meetings outside of class time. Consequently, we have held no meetings outside of class times, and our meetings during class time typically span about five minutes. Therefore, we have neglected to keep a record of internal team meeting notes.

We do have a record and transcription of our initial client meeting attached in the appendix separate from our team meetings, and we do have the below interaction with our client Kyle. Much, if not all of the intended content, such as assignment of work and completion of work are readily implied by the Cringe Coders Team's individual journals.

#### *October 3rd Mid-Semester Meeting Scheduling*

---

**Nolan** Hey [Kyle]! Hope everything has been going well these past few weeks. We have made some pretty solid progress on the portal so far. During our first meeting we talked about meeting up sometime around mid-October for another, so I was just reaching out to get a date figured out. I think fall break is in two weeks, so would you have availability to meet up sometime next week? I think the best time for our group is Tuesday/Thursday from 1:00-1:30 or sometime around 2:00ish - it's hard to tell how long Harvey will lecture for. Just let us know if any of those times work for you, and we can figure something else out if you're booked during those times. thanks!

**Kyle** Well, I don't know if any of you are going out of town for Fall Break, but I could do the Tuesday during it (the 17th) in the afternoon and then you wouldn't have to worry about when the lecture gets out. If not, we could do that Thursday (the 19th).

Also I forgot to give the order I was wanting the reports to be in. Here is that order: URL, Student Email, Student First Name, Student Last Name, Assignment, Question, Problem Type, Status, Time Created, Time Closed, Was Successful, Primary Tutor, Assistant Tutor, Semester, Course Number, Section Number, Professor, ANYTHING NEW

And the url it should use: <https://cslc.unomaha.edu/>

**Nolan** Hey Kyle, I have spoken with the group and I believe Thursday will work best as some team members have plans for the midterm break. I'd be happy to meet you in your office or at the CSLC or something and pick out a time that works!

Also, we have some exciting news to share: Harvey looks to have gotten some VMs for us to use, so hopefully we can get that figured out so you can have an easy test environment to mess around with instead of pulling down and running docker containers :) I'll keep you posted with that, still need to figure out exactly how he has them set up!

Looking forward to seeing you soon, and let us know if you have any questions in the meantime :) have a good weekend!

---

## 8.7 Individual Journals

### 8.7.1 Austin Bailey

*Client Liaison  
Information Security Analyst  
Quality Assurance Tester*

Austin Bailey is responsible for thoroughly testing the application to identify vulnerabilities, usability issues, and performance bottlenecks. Austin will create and execute test cases, provide feedback to the development team, and ensure a high-quality final product. Likewise, Austin will simulate real-world cyberattacks to identify vulnerabilities and weaknesses. Austin will help our team improve our security posture and protect sensitive information by uncovering any potential security flaws.

#### Journal

8/31 - built project context with lindsey  
 9/5 - met with client; record and will transcribe  
 9/7 - met helped build context document  
 9/12 - record context document with Mya  
 9/14 - Attended class; Built a goal model; provided pen + paper for wire framing  
 9/19 - Attended class; consolidated documents for submitting to canvas with team; scoped work of UML diagram to be done by EOD thursday (9/21)  
 9/21 - attended class  
 9/26 - Weekly Meeting with Siy; attended class  
 9/28 - Attended class  
 10/3 - Coordinated with Nolan for follow-up meeting with client; attended class  
 10/5 - no changes to project; team agreed no changes were necessary. Attended class  
 10/10 - context document formalization; project report work; attended class  
 10/12 - Mid-semester Project report finalization

### 8.7.2 Mya Bell

*Client Liaison  
Requirements Analyst  
Quality Assurance Tester*

Mya Bell will oversee the development process, and ensure that the project stays on track, objectives are met, and communication flows smoothly between team members. Likewise, Mya will facilitate communication and collaboration between leadership and team players to ensure a successful outcome. Lastly, Mya will create and execute test cases to ensure a high-quality final product.

#### Journal

August 31 - Discussed and finalized individual roles, project proposal, and project plan  
 September 5 - First meeting with client  
 September 7 - Wrote and recorded context document  
 September 12 - Wrote initial requirements document  
 September 14 - Finished requirements document  
 September 19 - Started nonfunctional requirements  
 September 21 - Discussed object model and finished nonfunctional requirements and updated requirements  
 September 26 - Went over progress on code milestone 1  
 September 28 - Check in with Mr. Siy  
 October 3 - Updated client on progress and scheduled next meeting with client  
 October 5 - Check in with Mr. Siy and reviewed current project plan  
 October 10 - Automated testing tools demo  
 October 12 - Check in with Mr. Siy and worked on midsemester project report

### 8.7.3 Nolan Gregory

*Architect*

*Technical Lead*

*Back-End Developer*

*Dev-Ops and Database*

Nolan Gregory is responsible for making technical decisions and guiding the development process. Nolan will provide technical expertise, review code, and ensure that the architecture and technologies chosen align with the project's goals. Nolan will handle all server-side logic, database management, and API calls. Nolan will build the core functionalities of the application, including ticket management, user authentication, and communication features. Nolan will design and maintain the database structure, ensuring efficient data storage, retrieval, and integrity. Lastly, Nolan will set up the deployment infrastructure, manage continuous integration/continuous deployment (CI/CD) pipelines, and ensure smooth deployment and scaling of the ticketing portal.

#### **Journal**

(add content)

### 8.7.4 Lindsey Langdon

*UI/UX Lead*

*Front-End Developer*

*Mobile Design Lead*

Lindsey Langdon is responsible for creating an intuitive and visually appealing user interface. Lindsey will design wireframes, mockups, and prototypes, ensuring a user-friendly experience and consistent branding. Lindsey will be responsible for implementing the user interface and ensuring that the application is responsive and visually engaging. They work closely with the UI/UX Designer to translate design concepts into functional front-end components. Lindsey will ensure that the front-end can run on mobile or tablet devices, as well as on desktop systems, with a focus on mobile-first design.

#### **Journal**

8/31 - built project context with Austin and worked on Project Proposal

9/5 - went over client meeting notes

9/7 - worked on wireframing exercises

9/12 - assisted with writing context documentation

9/14 - recorded Wireframes and Goal Model Requirements presentations

9/19 - worked on GQM Modeling

9/21 - attended class

9/26 - weekly Meeting with Siy

9/28 - attended class

10/3 - attended class

10/5 - team agreement that there were no changes to project plan

10/10 - recorded Testing Tool Demo presentation

10/12 - worked on Mid-semester Project report

## 8.8 Transcripts of Client Meetings

## Transcript of Initial Client Meeting

**TRANSCRIPT OF INITIAL CLIENT MEETING***COMPUTER SCIENCE LEARNING CENTER WEBSITE PROJECT**CRINGE CODERS TEAM*

AUSTIN BAILEY  
MYA BELL  
LINDSEY LANGDON  
NOLAN "OG" GREGORY

TRANSCRIPT OF INITIAL CLIENT MEETING FOR THE  
COMPUTER SCIENCE LEARNING CENTER WEBSITE PROJECT

DR. HARVEY SIY; COLLEGE IS&T  
UNIVERSITY OF NEBRASKA - OMAHA  
CAPSTONE COURSE  
SEPTEMBER 7TH, 2023

## Transcript of Initial Client Meeting

Transcript of Initial Client Meeting

**Nolan** So this is Austin.

**Austin** Hi.

**Kyle** Nice to meet you

**Nolan** And Maya.

**Kyle** Okay, Nice meet you.

**Nolan** I think they both you took cs1 in Java, right? Okay.

**Kyle** Okay.

**Austin** Yeah. Also I'm old guard.

**Kyle** Did you take cs2 and grab with them as well?

**Austin** Yeah, I started in 2018.

**Kyle** Okay.

**Austin** So i i've been here a minute

**Kyle** may or may not have had me for CS2 then depending on

**Austin** Um, i don't, no, i took a cs2 in the summer so i had So the guy teaches theory of Comp.

**Austin, Kyle** Solheim.

**Austin** Yeah, I had him because I decided to do calc 3 and java 2 in the summer com-pressed.

**Kyle** Yeah. But, Yeah, because i yeah, i was doing cs2 the online section of cs2 for A while when i was in java.

**Austin** Was that during COVID or?

**Kyle** Yes. During COVID as well.

**Austin** Okay, Because I I was here 2018 to 19 and then 19 to 20 activity gap year, a lot and then I came back to COVID but I had already finished up pretty much all the The low level cores.

## Transcript of Initial Client Meeting

Transcript of Initial Client Meeting

**Kyle** Yeah, I don't know. Do you want it closed or not? I don't know if you're recording, if it's going to screw with sound or something

**Austin** it shouldn't And it doesn't have to be perfect

**Nolan** Leave it cracked because I don't know how people, you know, everyone else reacts to sound traveling. Don't want to step on any of your professor's toes.

**Kyle** So yeah, awesome. Um I know that i think they got in contact with you the previous group, but yeah, what the code base looks like

**Nolan** Yep. So they used flask and pretty much just flask for everything. Um, just fine. So i looked at a probably going to reuse all their model, like their database diagrams that they have But we're going to use a different ORM.

We're gonna use Django instead Just so that can plug in with any I guess the database structure that you choose. Uh, whether that be like mysql or postgres or anything like that, And then, We're going to use react also, so that like whenever we have a component change or something like that, like if a student submit a ticket we don't have to refresh the page. So it's all like Loaded dynamically.

**Kyle** Okay

**Nolan** Just because that's kind of annoying. Whatever. I was working with the tutoring center. You always had. Like, I remember I downloaded a chrome extension. That's like an auto refresher like every five seconds or something. Just get the students.

**Kyle** Oh Okay

**Nolan** They're like submitted a ticket but it didn't pop up or something like that. Um, and then because that was something that Clim and Clayton, Clayton? Yeah, yeah. That was what they wanted to do that but they ran out of time because they said that, they Something happened where they weren't able to start development until the end of march.

**Kyle** Yeah, so there was an issue with originally we had we didn't have a hosting solution

**Nolan** okay?

**Kyle** So our first thought was trying to host it on the UNO community page. Which doesn't allow python for some reason. You can do it. The thing like, is set up to do it. But like UNO, specifically doesn't allow python for it.

**Nolan** What do they prefer

**Kyle** PHP.

**Nolan** Seriously? In 2023?

## Transcript of Initial Client Meeting

Transcript of Initial Client Meeting

**Kyle** So so they were they were like looking into that as a potential option to do like a lamp stack and pHP. Um, but then decided that that was just called That was just a bit too much. So we pivoted at the last minute and tried to find other hosting solutions, I do think O have a hosting solution now.

**Nolan** Cool.

**Kyle** Which is the good news, I haven't super logged on to it much. But I was just looking at that information.

**Nolan** Uhm

**Kyle** So, there is A new domain here. I'm just compiling kind of a list of notes of things. Um, That'll be cslc.umaha.edu, so like there should be a domain created and everything

**Nolan** okay

**Kyle** I will have access to it. So you had mentioned like late november. Part of this will be to help deploy it. So I have access to that so we can like meet and set up a time to actually deploy.

**Austin** It's just a working session for an hour or two?

**Kyle** Essentially. Yeah, just to get it up and running and then from there, you can only do your testing on it. If things need to change, then it should be easier for me to do redeploys the second time after i, like go through the first time. So

**Nolan** cool. Yeah we can help set up like a integration or something just so that it's easier to deploy if something happens or something needs to change the code base or something. Its just a quick little thing push it and it just automatically.

**Kyle** Yeah. I like the script essentially again. Cool, that sounds great. Um, so yeah, i think that that will be part of the the goals i guess is because i know i know you had mentioned like late, november?

**Nolan** that's, Yeah. it's like ideal

**Kyle** if that's the if that works. Um, And then they have some documentation. I don't know how much more documentation we'll need to be added, of course, with this, which to django, there will be somethings Um, Because i, i, All of the set up documentation as well. Should be updated then. To be django.

**Nolan** Yeah. Like to build stuff.

**Austin** It'll mostly just be translating it from

**Kyle** basically. Yeah

## Transcript of Initial Client Meeting

Transcript of Initial Client Meeting

**Nolan** that's a big part of this is basically just making it jump over from flask to django to Django. So that's easier.

**Kyle** Um yeah there is one thing i don't i didn't get to look too much at their reporting piece because that was one thing that i had talked to them about and asked about and they they had showed it off a little bit but didn't fully look into it. There is a format that i would prefer because i have a program on my end, that does an analysis. Of the report file and like, spits out things for me.

**Nolan** You want it as a CSV or a JSON or something?

**Kyle** Yeah, i take it in as a CSV,

**Nolan** okay?

**Kyle** And I have like, the specific. Fields. That i'm kind of looking for like we, we can add more fields, that's perfectly fine. Um –

**Nolan** but it's like a base

**Kyle** but that'd be a base of like what i'm looking for in that report. Um, and then if we add anything else, probably best at the end, just i'm thinking backwards compatibility-wise for my analysis program. So don't have to change too much myself.

**Nolan** Um, so like an advent panel sort of thing.

**Kyle** Yeah, yeah. So there should be an admin panel that set up in there and then there should be a way to generate a report. Um, i don't remember if they set up like Over certain times that you can generate it or for certain classes, i'm not certain.

**Nolan** It could be pretty. I mean doing it for a certain things. Like you could like filter it but things that would be pretty easy to do. That was one of the things that I worked on previously, actually. So,

**Kyle** So so yeah, so that's kind of the That's I guess the first goal and I don't know if we wanted to to meet up. Maybe sometime around fall break. Just to kind of see where you were at, as well.

**Nolan** Yeah, when it's fall break is that

**Austin** it's like October 18th or something?

**Kyle** yeah, like 16

**Austin** october.

**Nolan** Yeah, i'm trying to remember when the First milestone is okay. So i think it's around that time now, october something,

## Transcript of Initial Client Meeting

Transcript of Initial Client Meeting

**Austin** I mean, in all honesty, I'm not too concerned with the capstone's milestones. I'm more so concerned with Incremental progression the end state which i don't think will have an issue with scopes.

**Kyle** Yeah. And that would be six weeks as well. So that's that's a decent amount of time. Just to at least, you know, Start work on it but also like have an idea of it so that we can talk more about it on like what exact expectations will be

**Nolan** okay.

**Kyle** But um, Yeah, definitely. Like if you want to switch over to Django, get react up. That'd be great.

**Nolan** Yeah. Is there like, i mean, if that seems too com – not complex, but anything that's like out of the, ordinary.

**Kyle** I I think that should work because there is already a code base, it's just shifting the framework, and then adding a little bit on top. I think that'll be fine. Get the reports working and, like, to to the set standard because I don't know if they quite match the standard with their report that I would prefer, okay? And then, Uh getting the deploy set up.

**Nolan** Okay?

**Nolan** So that that really are the main takeaways that i'd like from this.

**Nolan** Yeah, that's um, easy, right? Easy. But that'll be, yeah. Fun to do.

**Kyle** And then so like if we want to meet around like that week after fall break or something, Like so that'd be six weeks. Um, so we could chat.

**Nolan** Okay, let me throw it out little thing here. So don't forget so that week after fall break

**Kyle** or the week of fall but would whichever whichever Sometime around that ever. Yeah, sometime mid october. Um, And then maybe. Is it, is it due? The eighth is that when Like due date is for the capstone.

**Nolan** Oh, the final report? Yeah, it's sometime around the eighth of July, something like that.

**Austin** Yeah, the cyber capstone is the eighth or at least that's when they do the presentation,

**Nolan** i think. Mya i think, oh, yeah, the eighth, Mya i put it on the check at the Jira board here. So yeah

**Austin** I know for the cyber one. They they actually start winding down about the 24th of november because then you go into Um,

## Transcript of Initial Client Meeting

Transcript of Initial Client Meeting

**Kyle** thanksgiving.

**Austin** Yeah, thanksgiving and then you have dead week.

**Nolan** Yeah. That i would want it to be live like the last week of november at the latest just so that we could test it and see if there's any bugs.

**Austin** If we can get it live before thanksgiving, then that just gives us kind of three weeks. Of –

**Kyle** yeah

**Austin** bug fixing UI/UX

**Kyle** that was kind of my thoughts. So either either that week before, thanksgiving, or the week of like the first few days of that, um, i think would work. And then, yeah, because always, after thanksgiving, there's a week of class dead week, finals week, and that dead week is when the The final milestones are due, so, If we can get it done before thanksgiving, then you can have that last week of like class after Thanksgiving to Actually, do testing and, and milestone checking and all that correct.

**Nolan** And then, if i set up like a little test, like a production environment, like, do you know what type of database this we'll use or does it?

**Kyle** I'm guessing it's my sequel. I don't quite remember, i'm pretty sure they have it in the documentation which one they used as well.

**Nolan** Okay, oh yeah.

**Kyle** But i would, i would guess

**Nolan** I think that they used mySQL. Also.

**Austin** I mean, that's what they teach. So

**Kyle** It's it's what should be on all the things? Um, and then, i don't know if you want to set up like a discord group for, if you have another way that you're communicating,

**Nolan** yeah, we have one Clim was saying that they had one with you last semester. Um, if you want, we could just do that one. Also, if you already have that or

**Kyle** oh, if you want to join that.

**Nolan** Yeah. Because he was saying, like, it's just so that you don't have a million different chats all the time. Other things. Doesn't matter to us and we have a discord chat. We could add you to it. Or

## Transcript of Initial Client Meeting

Transcript of Initial Client Meeting

**Kyle** whichever one, you're more comfortable with because if you still want to have one without me, that's that's fine. Okay.

**Nolan** Like we could

**Kyle** I encourage you to not always have to bring me into the conversation.

**Nolan** Yeah, we could do it. Like the way that tutoring one's set up where it's just like you have the one channel, like an advisor channel or something like,

**Kyle** oh, you have like a, you have like a server?

**Nolan** Oh yeah, we have a server.

**Kyle** Oh, yeah. Yeah. If you want to do it that way. That's easy. Okay. So sorry they had a group chat.

**Nolan** Yeah, i actually a group, would be better because i want to clog out your left side screen with another server because that happens about this – every class.

**Austin** Don't you group yours at all?

**Nolan** No, i just

**Kyle** oh, i do some grouping. I have like a

**Nolan** i don't even know that

**Kyle** you know, everything folder

**Austin** like i have. Like my UNO ones. I like my friends, social ones and gaming and like one tab and yeah, i'll just like minimize them and then i have like three things over there.

**Nolan** I didn't know that i'll have to do that because

**Austin** you're looking all cluttered over there bud.

**Nolan** Yeah, this isn't even my this is just my like one of my personal discords

**Kyle** I just have the one that i use for everything.

**Nolan** I got scared, i didn't want to like have my personal one, be linked to school stuff and then accidentally send like a meme and like a school server or something. I don't know.

**Austin** Are you saying your memes are too spicy?

## Transcript of Initial Client Meeting

Transcript of Initial Client Meeting

**Nolan** Maybe, i don't know.

**Austin** I don't believe you.

**Nolan** Uh, okay, so cool. That's a That's good, like starting.

**Kyle** Well and then and then i can send you so so once once you, you end up setting that up,

**Nolan** okay? Okays

**Kyle** Um, like the the url to use, because i know that that's something that Clim was asking for toward the end of the last semester was that he needed an actually, a URL. So, i can do that and, as well as the report Uh, Fields.

**Nolan** Oh yeah, that would be perfect. Just database like relationships and stuff. But yeah, voice for voice pulling stuff down. It should be pretty easy with like the relationality of the database just to like filter things to grab like Oh i want all of these professors and stuff like that, that would be pretty easy. But there was like some things i was thinking. And i don't know, like how Not ethical, but like Uh, rating system for like tutors, for example, like a student could leave, like a thing, that's kind of like In concept. It's good. But i feel like i'm practice really bad idea.

**Kyle** Yeah, what we've used and i don't know. I haven't checked it in a little while. But we had like a A feedback form.

**Nolan** Okay

**Kyle** not many students would go and do it. It's only if they had like a particularly bad experience is when they would do.

**Nolan** Yeah that seems

**Kyle** but um we did have a link to a feedback form. So, i mean, i could I could, we could look to that and include that. Otherwise,

**Nolan** it's up to you. Yeah. I was just kind of thinking of things that could be fit up at me.

**Kyle** Yeah, there's a few other things. Um, One that i would like that is not currently a field. But i would like to be a field is whether it was an online ticket or not. And some way of marking it as an online ticket. I think i think they did do that, but i definitely want that to be included. Um, And then i was also thinking of

**Nolan** We could make it like a required one too, so, it's not always like, uh, you forget them?

**Kyle** Yeah.

## Transcript of Initial Client Meeting

Transcript of Initial Client Meeting

**Nolan** Or something like that. That would be really good.

**Kyle** Because then the other one that i was thinking of and this would be A lot more work. So this is very much not a requirement. Um, but if there was an easy way, a script that could run That could. Look through the corse search or maybe i know that there was a group, a capstone group that did like the course scheduler for the university.

**Nolan** Oh, and like auto populate

**Kyle** auto populate for the semester because currently what happens is i go into and did it by hand. Yeah i would like look through the course search populated by hand. Okay. So if there was some way of Automating that if not like i can look into that myself in the future but

**Nolan** yeah we definitely could because i've written scripts before that will go through and create the like database fields and stuff but I don't know what that looks like.

**Kyle** Exactly.

**Austin** So does UNO publish it. Outside of mavlink.

**Nolan** Probably not. I don't think that it's going to be public data. I think it probably would have to get it from the university, maybe.

**Kyle** Yeah, that's, that's the question. I don't quite know.

**Nolan** Okay.

**Kyle** So that's why it's, it's not a requirement it's something to maybe think about. Would it be possible? But it's definitely not a request.

**Nolan** Yeah, sometimes i don't know, once we start getting the main stuff done, if we have like an extra bit of time, i can go to the registrar and ask or if you want to. You can just

**Kyle** yeah, i could talk to and i could also talk to Dr. C there there was a group i know. Last semester. That was doing like a computer science registrar system.

**Nolan** Yeah, yeah, i've seen that. I think that's what ethan did. Also, he did the schedule. Yeah, he did it with Con but they scrapped theirs. I'm guessing because they were rendering all their back end like aligning and padding and i like no, no other front end stuff was being calculated in the back end. It was very, it was very interesting setup. It was all c, sharp. So, That's okay to do, but I guess i didn't follow up with it. They had another takeover but That's a tough. That's a tough one, because they were having to write, like, genetic algorithms to, like, determine if they were, like, if this room was taken by the next best fit for the room, and i remember at the end of the semester, Ethan was almost like, i'm just, we just can't do this. Like, this is just too much for, you know, four months of work.

## Transcript of Initial Client Meeting

Transcript of Initial Client Meeting

**Nolan** Yeah. So, i I don't know how possible it is, but Yeah, thatd be something id like to add making a little bit easier for you so you don't have to go through manually. Add every single professor update things, if things need to be updated.

**Kyle** But again, if if you don't do it, no worries. I can always try and figure something out of myself.

**Nolan** Okay. Um, we have to go because the class start.

**Kyle** Yeah, i'm sorry. Taking you right after the time.

**Nolan** Yeah. I'll make a chat and then, yeah.

**Kyle** And then i can Get back to you on in there.

**Nolan** Cool.

**Kyle** And we'll keep in touch.

**Nolan** Yeah

**Austin** perfect.

**Nolan** And let's schedule some time in mid october.

**Kyle** Awesome

**Nolan** Cool

**Kyle** Have a good rest of the day

**Nolan** Yeah, see you later. Yeah.

# CHAPTER 9

---

## References

---

- Connolly, T. M., & Begg, C. E. (2014). Database Systems: A Practical Approach to Design, Implementation, and Management. Pearson.
- Django Project. (2021). Django Documentation. <https://docs.djangoproject.com/en/3.2/>
- Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine.
- Open Web Application Security Project (OWASP). (2021). OWASP Top Ten. <https://owasp.org/www-project-top-ten/>