

“NCAT Course Planner” **Requirements Specification**

**Version 1.**2

March 10th**, 202**4

**Table of Contents**

[**1. Executive Summary 2**](#_1fob9te)

[**1.1 Project Overview (Problem Statement) 3**](#)

[**2. Project Description 3**](#)

[**2.1 Product Context 3**](#)

[**2.2 User Characteristics 4**](#)

[**2.3 Assumptions 4**](#)

[**2.4 Constraints 4**](#)

[**2.5 Dependencies 5**](#)

[**3. Requirements 5**](#)

[Functional Requirements 6](#_wl3me2ufxesw)

[3.1 User Interface Requirements 6](#_ktbt4k68g7n4)

[3.2 Usability 7](#_7tyqrujqvbr3)

[3.3 Performance 7](#_uk1cf9am9w72)

[3.4 Manageability/Maintainability 8](#_29vptuz03f9x)

[3.5 System Interface/Integration 9](#_uij8xf7k33vq)

[3.6 Security 10](#_fte35l4k7dsi)

[3.7 Data Management 11](#_4trd4lpcir6b)

[3.8 Standard Compliance 11](#_enjd82lv7xwc)

[3.9 Portability 12](#_9rfunxsprvv2)

[**4. Specifications/Implementation Plan/Flow Chart 12**](#)

[4.1. Implementation Plan 12](#_2xovcfpa4cbf)

[4.2. Flow Chart/ User Scenarios 13](#_jv2sousw4in4)

[4.3. UML Use Case Diagram 29](#_u4q602434628)

[**5. Future works including revised project schedule 29**](#)

[5.1. Bar Chart - Task Balance (CST-498) 30](#_bgrs6f7kod02)

[5.2. Gantt Chart - Project Schedule (CST-498) 30](#_4q1to1v0ikpy)

[5.3. Bar Chart - Task Balance (CST-499) 35](#_9zzije6p8zsx)

[5.4. Gantt Chart - Project Schedule (CST-499) 36](#_7t9jnvmvlsv5)

[**6. Learning Statements 42**](#)

[6.1. Keyera Bruce (Fall 2023) 42](#_4am6yc6gvqo6)

[6.2. Mone’ Bell-Arrington (Fall 2023) 42](#_l2mth55djj46)

[6.3. Ale’A Jones (Fall 2023) 43](#_269huo4fk8xc)

[6.4. Marvin LeGrant (Spring 2024) 43](#_f7q0js2y6tyg)

[6.5. Gary Samuel (Spring 2024) 43](#_4fdao3cabfni)

[6.6. Mone’ Bell-Arrington (Spring 2024) 43](#_4xasg7rch4mu)

[**7. APPENDIX 43**](#_23ckvvd)

[Appendix A. Definitions, Acronyms, and Abbreviations 43](#_gp3hx5ghbx9l)

[Appendix B. References 44](#_7huob8bj3dzl)

[Appendix C. Source Code 45](#_l39jjl1jzmky)

[○ Backend/Frontend 45](#_3615muwhvj97)

# Executive Summary

## ***Project Overview (Problem Statement)***

North Carolina Agricultural and Technical State University students and staff are experiencing an increase in stress and confusion due to the disorganization and instability during the registration process each year. If this problem continues, the university could expect an increase in frequent emails and complaints about student struggles. Based on personal experiences as an NCAT student, it seems like there can be more done to make the registration process simpler for both students and staff. Therefore, there needs to be a proactive approach to identifying and dealing with situations that may result in frustrations during the registration process.

Group 3 of CST 498 has come up with a proactive approach to create an NCAT course planner to lighten the load of both staff and students. Group 1 of CST 499 is taking from the previous course and continuing the project. This course planner would be a simple web application that web scrapes the list of courses that are required to complete a student degree without the manual processes currently established. These manual processes include an out-of-date spreadsheet, frequent communication through email, and constant searches through Aggie Access. This web application tool would help determine leftover courses for the applicable students and what semester or year to take courses based on the student’s status. It will, also, determine the transferable credits required to complete the degree. This could be available to transfer students and future students looking into NCAT being the college of their choice, so they are aware of what courses they need to complete, how many credits they need, and the criteria to achieve the degree they want in the end. This course planner is the solution that will make the registration process simpler for both students and staff. As a group, we hope it will benefit North Carolina Agricultural and Technical State University in the best way possible.

# Project Description

This software application will be an NCAT Course planner and degree auditing program as a replacement process for advisors and students at North Carolina A&T State University. This software will be a web application prioritizing reduced manual processing of degree progress-tracking, college credit transfer eligibility, and semester-based/major-based course availability.

As a portion of the NCAT homepage, this software application will combine the curriculum guide for the desired degree along with the courses that have been taken and are currently in progress into one web page where recommended courses appointed to a set part of the semester will be displayed.

Majority of the working source code will combine HTML/CSS and PHP functionalities that will collect information pdf files for the curriculum and the student’s transcript. Once an overview of possible course attendance has been established, the advisor and the student will review further recommendations and tailor course scheduling that may be beyond the application’s purposes.

## ***Product Context***

It is related to other course planners that were created for other universities and these course planner will be a reference for our prototype. This is something that will lessen the stress of looking for Word Excels for registration. Students will use the NCAT course planner in any instance that has to do with registering or reviewing their courses. If students need a first look at what course they need to take in order to graduate, they will use the NCAT course planner to prepare. We would think that the program is independent and self-contained because the students will depend on it to tell them what are the next steps to take during their time at North Carolina A&T. It should already hold the information of the students if their account is in the system with the correct information.

## ***User Characteristics***

| **Student** | **Student Advisor** | **System Administrator** |
| --- | --- | --- |
| **Background**   * Senior student at North Carolina A&T * In the Information Technology major program * Marital status: Single * No children   **Demographics**   * Gender: Female * Age: 21 * Race: African American * Occupation: Server at Texas Roadhouse   **Identifiers**   * Determined to graduate * Has 5 college courses * Has a 3.5 GPA | **Background**   * Student Advisor at North Carolina A&T * Department of Computer Systems Technology * Marital status: Married * 2 children   **Demographics**   * Gender: Male * Age: 43 * Race: Asian * Occupation: Associate Professor at NC A&T   **Identifiers**   * Determined to help students graduate * Advises over 300 students * Has a Doctorate degree | **Background**   * System Administrator at North Carolina A&T * NC A&T Personnel * Marital status: Married * No children   **Demographics**   * Gender: Female * Age: 40 * Race: African American * Occupation: Computer Technologist at NC A&T   **Identifiers**   * An expert in their profession * A passion for technology * Provide technical support for students and staff |

## ***Assumptions***

* + 1. All users of all roles will require knowledge in website navigation and dexterity in basic computer fundamentals such as mouse-clicking, scrolling, and keyboard inputs.
    2. All users of all roles should have access to the internet with Google Chrome as the recommended browser.
    3. All users of all roles should be registered as an NCAT student or faculty member with access to the website via NCAT UserID and password.
    4. Users with the role as an Advisor or Faulty Member should have access to students’ academic transcripts and currently have authorization to perform substitution and/or exception overrides for course registration.
    5. Users with the role as a Student should understand the processes of registering for courses in each semester and communicating with academic advisors prior to finalizations of course registrations.
    6. All users should have hardware that runs on Windows 11, ChromeOS, and Linux operating systems.

## ***Constraints***

* + 1. **PHP Memory Space**

NCAT Course Planner will require a heavy reliance on PHP data fetching between a student’s academic transcript and the reading of multiple degree curriculum files. Since the NCAT Course Planner will require simultaneous requests, the probability of the tool lagging will be evident if overused. Maximum memory space will be limited to 1500M unless more space is necessary to run the tool without a 500 error code.

* + 1. **Limited User Scope**

This web application will only be used by personnel of NCAT either students or faculty. NCAT members who still have access to their NCAT account are permissible to the tool’s usages. If the tool’s functionalities are deemed stable enough with integrated dynamics and maintainability from rising capstone students or other contributors, there will be a possibility of a wider scope of users across other Universities nationwide.

Current members will create a prototype sufficient enough to continue upon the next semester where the need to expand user scope will become priority. Members will determine host management that dynamically integrates with other institutions.

* + 1. **Mobile Interface Options**

There are possibilities of adding a mobile interface to the NCAT Course Planner however, there is little experience within the team with mobile application programming.Priority to establish known experience into the NCAT Course Planner tool is first before other features are applied to the overall product.

* + 1. **Browser Compatibility**

The NCAT Course Planner will rely on the user to provide rendering of the web interface. Since the time frame is not adequate enough to apply dynamic browser compatibility, the initial release will only support the Google Chrome browser for the desktop. Support for other browsers and mobile hardware will be added upon later releases.

* + 1. **Server Requirements**

Server must be compatible with minimum requirements of PHP 8.2.x, MySQL 8.0 along with Linux 5.13 and Windows 11 operating systems. Memory space will be problematic if limitations are not displaced initially.

## ***Dependencies***

* + 1. Our application is dependent on the structure of our SQL database as well as our HTML/CSS code.
    2. The actions of our users are dependent on the level of access control we give them.
    3. A user's personal information must be inputted before they’re course planner is generated. Student data must be accessible because it is the required information our software needs to generate the intended results.
    4. As we begin to further develop our prototype and source code the dependencies list will expand.

# Requirements

The requirements for the NCAT course planner will lay out all the details to be able to design the system properly. This will ensure that the purpose is fulfilled and students/staff are able to use the system with ease and get the information they need. This section will describe all system requirements in enough detail for designers to design a system satisfying the requirements and testers to verify that the system satisfies requirements. It will also describe every input into the system, every output from the system, and every function performed by the system in response to an input or in support of an output. Each requirement should be numbered and prioritized.

1. Priority 2: Provides personalized term recommendations from students’ declared programs
2. Priority 1: Brings aggregate data for the school’s course demand analysis and classroom scheduling
3. Priority 1: Guides students to graduation quickly and with ease
4. Priority 1: Auto creates plans for students directly from their selected degree audit requirements
5. Priority 3: Provides guidance for students’ basic questions that involve their credits per term or desired graduation date
6. Priority 2: Creates pathways each term with intuitive course recommendations for students based on their selected program requirements
7. Priority 1: Notifies students when there are prerequisite conflicts and availability issues
8. Priority 2: Adjusts plans in real-time to reflect course outcomes
9. Priority 2: Adjust plans in real-time to reflect curriculum changes
10. Priority 3: Allows students to make changes to the auto-generated plans
11. Priority 1: Keeps students on track to complete their degree requirements
12. Priority 3: Validates the planned courses against students’ declared degree requirements to prevent students from getting off track of their degree
13. Priority 1: Shows students’ real-time academic progress
14. Priority 2: Contains reports that are available which include student plan progress, scheduled course demand, and schedule preferences
15. Priority 1: Includes a real-time interface with requirement rules and course catalogs to identify course scheduling details
16. Priority 1: Contains a Current Path Title that will indicate all active academic plans in the student’s record
17. Priority 1: Contains a display of progression that will show in-progress/completed units in the current degree path
18. Priority 2: Contains a display of past courses that the student has completed whether the status is Pass, Failed, Repeated, etc.
19. Priority 3: Contains a Degree Pathway and Semester Edits section to edit and change semesters individually and select, move, and add courses.
20. Priority 2: Courses will automatically check off when students complete them with a passing grade
21. Priority 1: Displays student information including name, ID photo, student number, list of their currently declared majors, minors, and certificates

**Priority Definitions**

The following definitions are intended as a guideline to prioritize requirements.

* Priority 1 – The requirement is a “must have” as outlined by policy/law
* Priority 2 – The requirement is needed for improved processing, and the fulfillment of the requirement will create immediate benefits
* Priority 3 – The requirement is a “nice to have” which may include new functionality

## Functional Requirements

## User Interface Requirements

In addition to functions required, describe the characteristics of each interface between the product and its users (e.g., required screen formats/organization, report layouts, menu structures, error and other messages, or function keys).

The requirements are the aspects of the system that are related to the appearance, layout, navigation, and the interaction of the user interface.

Appearance and layout is the look and feel of the graphical user interface that includes color, shape, layout, and typeface elements. For the NCAT course planner, the appearance and layout will look very minimalistic to keep the clutter to a minimum. This will ensure that all the unnecessary information that may be displayed will not compete with the user’s limited attentional resources because this may affect the user’s memory capacity of relevant information that they use the application system for. One of the features that the users have access to is adjusting the font size from small, medium, and large for users with impaired eyesight.

Navigation is the set of actions that will help guide the users to navigate through the NCAT course planner so they can complete their tasks related to the student courses. The team wants the users to navigate through without having to resort to documentation. The user will be able to click through each of the features with named buttons that will link them to different information they need.

The interaction of the NCAT course planner user interface will aim to define and facilitate interaction between the users by means of this application. It will effectively communicate this system’s interactivity and functionality and is grounded with an understanding of real users’ goals, tasks, experiences, needs, and wants. It will give students a chance to build, compare, review, and register course schedules for available semesters.

## Usability

Performance

* Users will be able to perform a course schedule review for the first time without assistance
* Users will be able to review unit objectives without assistance.
* Users will be able to review the curriculum of their chosen degree program without assistance

Learnability

* The user documentation and help should be complete
* The help should be context sensitive and explain how to achieve common tasks
* The system should be easy to learn

Accessibility

* The web application must be perceivable so users can be aware of the content and information presented
* The web application must be operable without disrupting the user in any way. They must be able to utilize every part of the application’s functionality.
* The application must be robust enough to be easily interpreted and consumable by all users, so HTML will be used to allow assistive technologies that can parse the code without the visual reference.

User Friendliness

* The web application must be highly learnable without any self-absorbed functions.
* The users must complete tasks with ease either by supporting multiple languages, avoid setting up roadblocks for users, etc.
* The web application must behave like the users expect it to.
* The application must allow the users to control what they need/want to control and not control what they are not interested in controlling.

## Performance

* + 1. **Capacity**
       1. The web application must be able to handle usage spikes from up to 10,000 concurrent users during peak enrollment and registration periods along with Student-Advisor course examination meetings. This is based on the average Fall enrollments of 2022 and 2023.
       2. The web application shall be capable of maintaining load efficiency as students, faulty members, and administrators throughout the semester as the expected total member count expands 23% over time.
       3. The software shall ensure degree auditing transactions aim for a maximum processing time of 5 seconds per request.

* + 1. **Availability**

Include specific and measurable requirements for:

Hours of operation

* Must be available to students and staff 99% of the time every month of the semester during business hours.

Level of availability required

* High availability is the level of availability normally expected by users so the system will be available during the specific operating hours without any unplanned downtime or outages.

Coverage for geographic areas

* The services of the web application should be geographically available anywhere in the US especially for distance learning courses, but the main focus should be within the North Carolina area.

Impact of downtime on users and business operations

* The company may lose a specific amount of hours of productivity and customers will be unable to access the application since the necessary tools are not available. This may result in reputational damage.

Impact of scheduled and unscheduled maintenance on uptime and maintenance communications procedures

reliability

* Scheduled maintenance will help prevent future issues. Unscheduled maintenance will help address problems immediately as they occur.  
  + 1. **Latency**
       1. The average time for a service request is 2000 ms.
          1. The average time for a simple request is 1250 ms (1.25 seconds).
          2. The average time for a large request is 5000 ms (5 seconds).
       2. The average time for the initial load of our application is 2500 ms (2.5 seconds).
       3. The average time for a web page load is 1500 ms.
       4. The average start rendering time is 750 ms.

## Manageability/Maintainability

* + 1. **Monitoring**
       1. Group 3 will monitor the application frequently and fix any failures as soon as they are able. We will make use of various web monitoring tools and code to help identify and correct any errors.
       2. Group 3 will make use of the Markup Validation Service and run our HTML code through it periodically or in the case of an error.
       3. Group 3 will regularly check the error reports and logs to identify any errors that occur in our application.
       4. A test case can be conducted if the group has the need or time to set one up. This can be used to determine if our application is meeting our requirements and functions properly.
       5. Users will get error messages if the interface acquires an error that prevents it from operating. The message will reassure them that the application will be back up and running shortly.
       6. A parity check will be used in the case an error occurs. This is the method for correcting errors in our application.
    2. **Maintenance**
       1. Group 3 shall service upgrades to the application’s user interface to create ease of use for all roles applying to the functions of the NCATCourse Planner within the timeframes from August 2023 to December 2024 where new/recurring members of the group will continue to administer inputs to the requirements specification document through the application’s life cycle.
       2. This application must maintain version control using GitHub’s open source repositories with daily and weekly checks for upgrade requests/requirements and bug handling processes.
       3. Group 3 must document any changes or features to the application using all acquired project management tools (Asana), communication tools (Blackboard, Gmail, GroupMe, etc.), and updates to the requirements specification documentation via Google Drive.
       4. This application must have additional support for long-term handling when all current members of Group 3 become Alumni and may not have the same resources to register upkeep with the NCAT Course Planner’s source code.
       5. Group 3 shall reconfigure UI/UX mapping where new/deleted/upgraded features are applied within the desktop web application which is inclusive to mockups shown in the User Interface Requirements section.
       6. This application must be hosted on either NCAT’s server, AWS, Heroku, or other cloud computing platforms.

* + 1. **Operations**
       1. Users will be required to input certain data into the interface. This includes the first and last name, email address and phone number.
       2. The application will time out if a user leaves it unattended with no activity for 15 minutes.
       3. The data that is inputted by users will be validated by using a data-type check to make sure the information they inputted matches the data field.
       4. A consistent backup will be used to backup the system after a user shuts down the application.
       5. Microcopy instructions will be available for the users to see as the input information. This will reassure the user that their information is protected and accurate.
       6. In the event of a disaster, Group 3 will have a plan in place to recover any lost user information or files.

## System Interface/Integration

A database system will be required to store, retrieve, and run queries on data. It will be used to store data within the web application in order to display how it will work. The group will implement their own transcript information and course curriculum within the database as an example to ensure that the program will work how it is intended to. The SQL programming language will be paired with this database system as the group has previously used and currently using it.

* + 1. **Network and Hardware Interfaces**

Specify the logical characteristics of each interface between the product and the hardware or network components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols (e.g., signal handshake protocols).

Network Interfaces

The NCAT course planner will most likely use one of the most common network interfaces, which is the WiFi interface. It allows wireless communication over an IP network and is considered a wireless Ethernet network where the Ethernet packets are exchanged in the same way that wired networks would. It is required to have WiFi to access the database systems that support the course planner to obtain the information that users may need.

Hardware Interfaces

The hardware interfaces are able to help various hardware devices to connect and communicate with one another. The web application should support USB interfaces such as printers, scanners, and keyboards. Other supported hardware interfaces include computer mouses, flash drives, and external storage drives.

Software/Programming Interfaces

The software/programming interfaces are the languages, codes, and messages that the web application will interact with to communicate with one another and to the hardware. The supporting operating systems will include Windows, Mac, and Linux. It can also be available on mobile devices. Other interfaces will be SMTP email and IP network protocols.

* + 1. **Systems Interfaces**

The File Structure

* logical record size
* physical block size
* packing technique

The File Attributes

* the metadata (about the file)
* name, including readable text
* identifier (could be a number)
* type, including a description of the contents within the file
* size
* what device and its location
* protection and security controls including the owner or permissions
* usage information
* time the file was created
* time the file was modified
* time the file was last accessed

The User Information

* the user’s name
* the user identifier (user ID)
* home directory
* access permissions and constraints
* search paths
* list of directories
* etc

## Security

* + 1. **Protection**
* Encryption Keys and algorithms will be used to protect sensitive data including any personal information provided by the user. We will utilize encryption at rest and encryption in transit to ensure user, site and stored data are protected. Users won't be required to login so no password encryption is required.
* A method we use to log what users access our application, when they did it, what actions they used and what pages were visited.

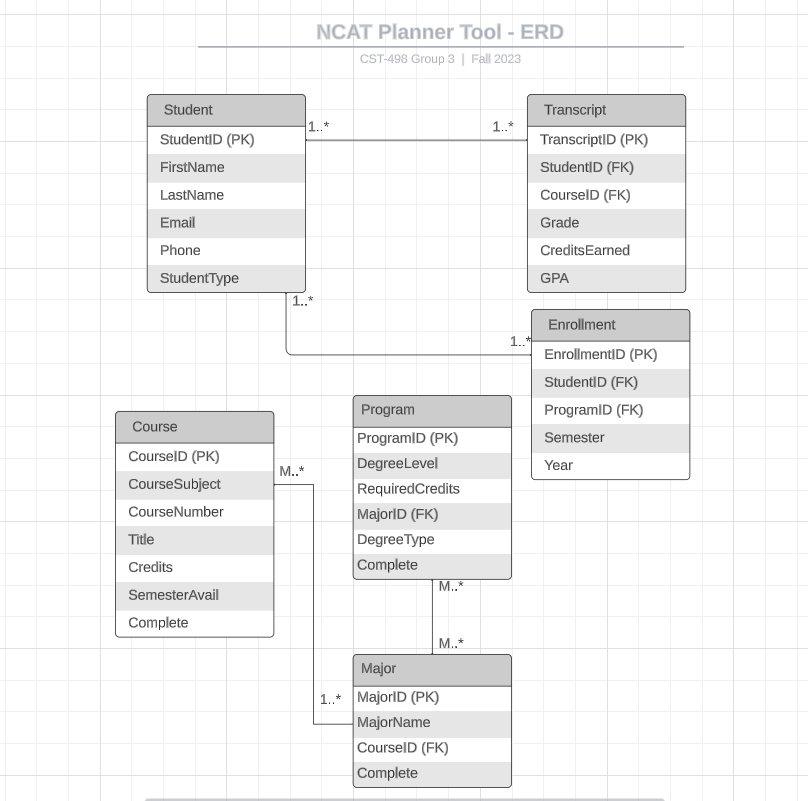
Data integrity Check

* .Group 3 will conduct a data integrity check once a week or as often as needed.
* This will include but is not limited to checking whether data in tables can be modified, if default values are applicable when one isn’t given by the user and whether or not our application is compatible with other software.
* These tests will be conducted in three ways, entity, referential and domain integrity checks.

* + 1. **Authorization and Authentication**
       1. **Authorization**
          1. Enrolled students and current faculty members will access NCAT’s curriculum guide based on the degree and study of their interest.
          2. Users with access to their course transcript will be able upload a copy of their unofficial/official transcript to the program.
          3. Permissions shall be assigned by the System Administrator based on user roles and responsibilities.
          4. Advisors will have access to all Students’ degree progress and privileges to read and update course requirements per student-by-student basis.
       2. **Authentication**
          1. Usernames and passwords are not currently needed in the prototype however, if integrated with the NCAT website, it is required where registered students and faculty members will have access to the tool.
          2. All authentication protocols will be inherited from NCAT’s authentication check if the program integrates with the NCAT website.
          3. Only users enrolled in NCAT are eligible to use the tool, otherwise the results will be inaccurate.

.

## Data Management



## Standard Compliance

* + 1. The software development shall adhere to secure coding practices as outlined by OWASP.
    2. Data at rest and during transmission shall be encrypted using encryption algorithms and protocols such as AES for data encryption and TLS for secure communication.
    3. User authentication shall follow basic standards, including the use of secure hash functions for password storage and MFA.
    4. The user interfaces shall comply with the latest version of HTML/CSS standards to ensure cross-browser compatibility.
    5. Client-side code shall follow best practices and guidelines as defined by the ECMAScript specification, performance, and security.
    6. The software shall adhere to NCAT policies and standards to ensure that enrolled students and current faculty members.
    7. Database interactions and queries shall follow SQL standards to ensure compatibility and portability across various database management systems.

## Portability

* + 1. The web application must be able to run on Linux, Windows 11 and ChromeOS Desktop operating systems with Windows and Android Mobile interfaces applied in future versions.
    2. The web application shall be written in PHP, HTML/CSS, and MySQL. Majority of the source code will consist of PHP functions.
    3. The web application’s database shall be tested on the XAMPP web server prior to publishing.
    4. This software must be tested and compiled in Visual Studio 2022.
    5. The application would be nice if run on NCAT’s network with access to the student's current unofficial transcript and degree curriculum documentation.

# Specifications/Implementation Plan/Flow Chart

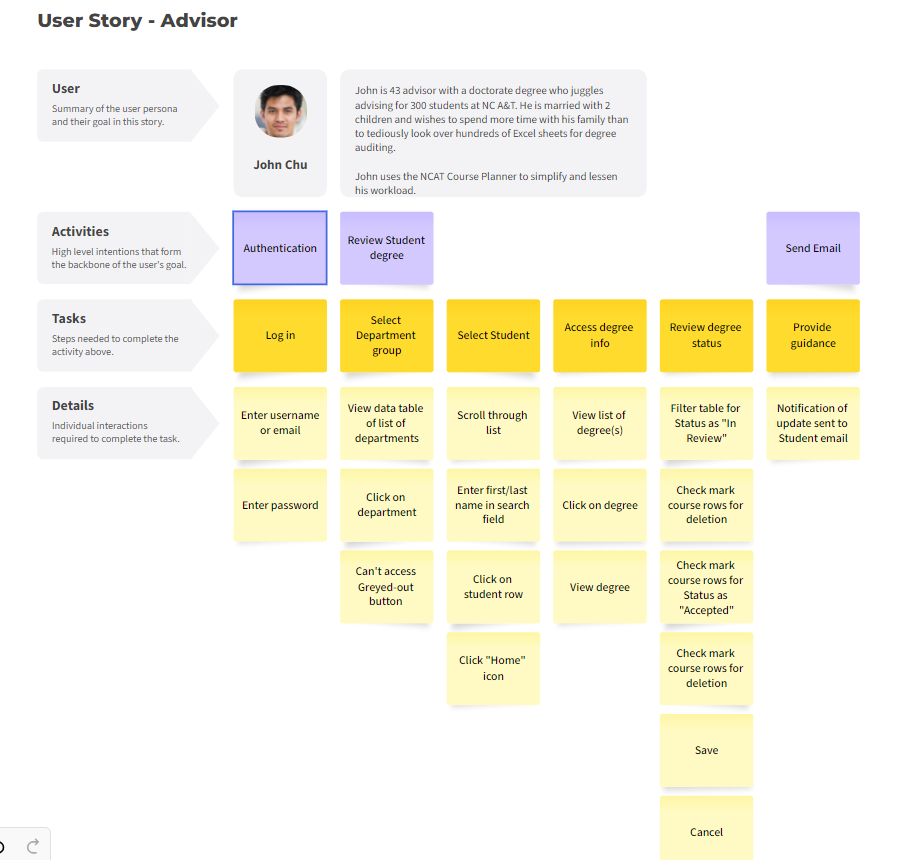
## Implementation Plan

After completion of this Requirements Specification document, Group 3 of CST 498 will review the document as reference to the development and completion of the NCAT Course Planner Tool. By the end of the semester, the tool should have a completed prototype with errorless or close to errorless functionalities. Further implementation will continue onto the CST-499 course.

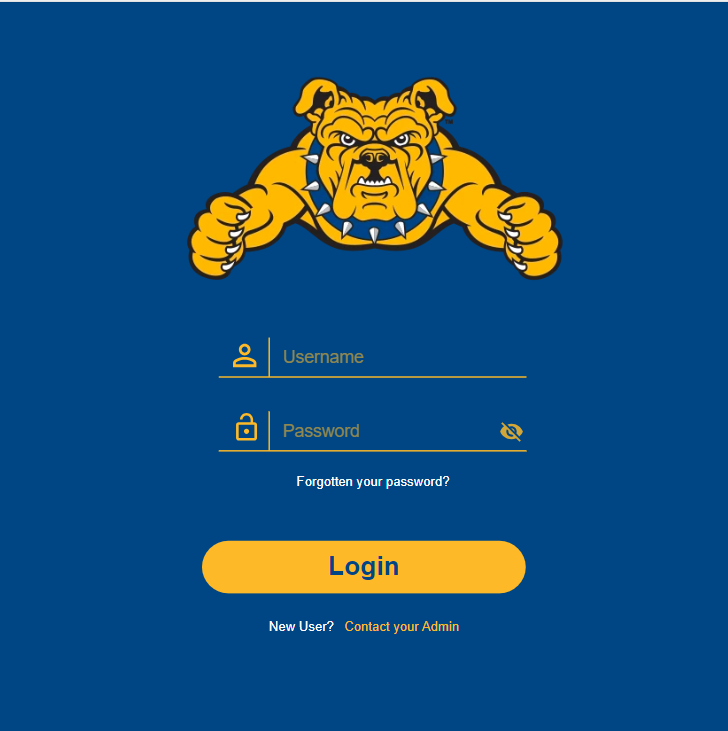
Since this is a web application, no hardware setup is required. Users will be able to harness the benefits of the tool with a few simple clicks and navigations of the web application.

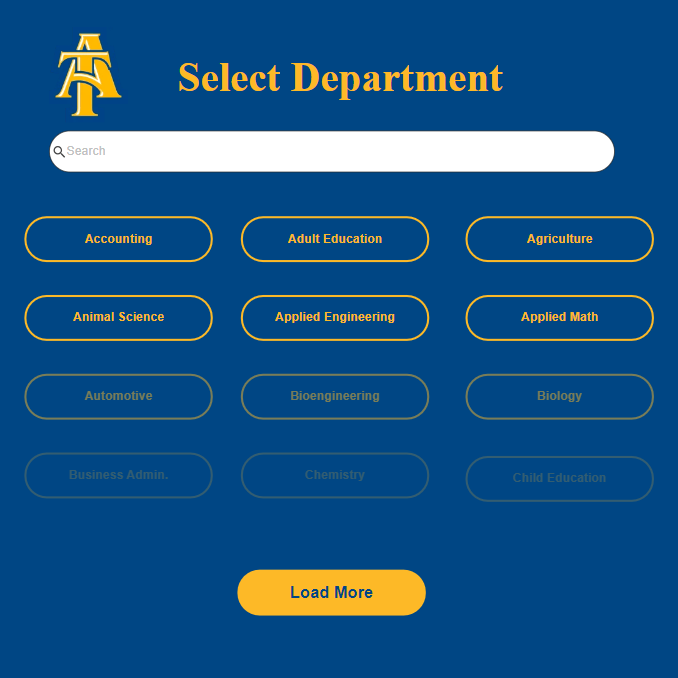
* + 1. Development (Week 1 - 5)
       1. Develop the frontend components based on UI/UX design.
       2. Implement the PHP logic and functionalities.
       3. Implement the user authentication and/or authorization if permissible by faulty members of NCAT that the project can integrate with current NCAT web pages and navigations.
       4. Develop basic features and database interactions for user roles of Student and Advisor.
    2. Testing (Week 6 - 10)
       1. Conduct unit tests for individual components with the use of Gherkin or other relative test case tools.
       2. Conduct programs through various browsers for compatibility check.
       3. Gather feedback from classmates, advisors, and/or other NCAT members for usability testing.
    3. Prototype Deployment (during CST-499)
       1. Finalize SRS documentation based on review and feedback from the professor.
       2. Configure the application for production use during CST-499 stages.
       3. Conduct final functionality checks and error checks.
       4. Integrate data from existing databases.

## Flow Chart/ User Scenarios

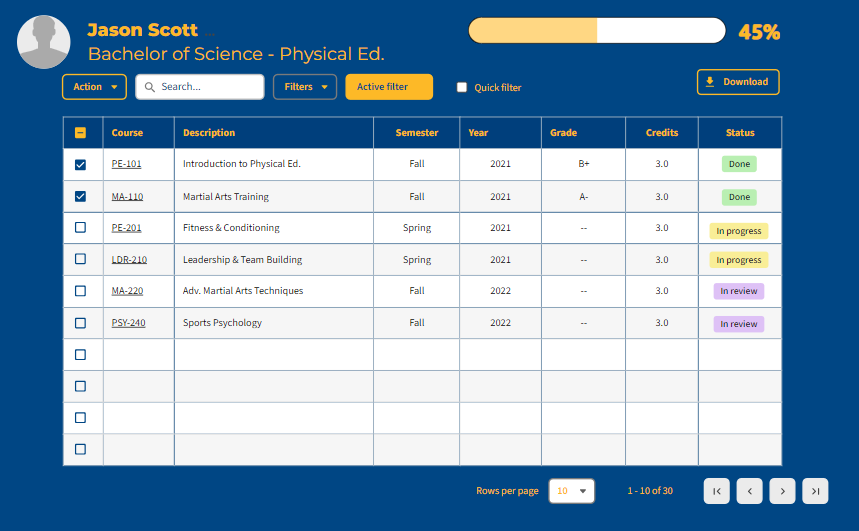


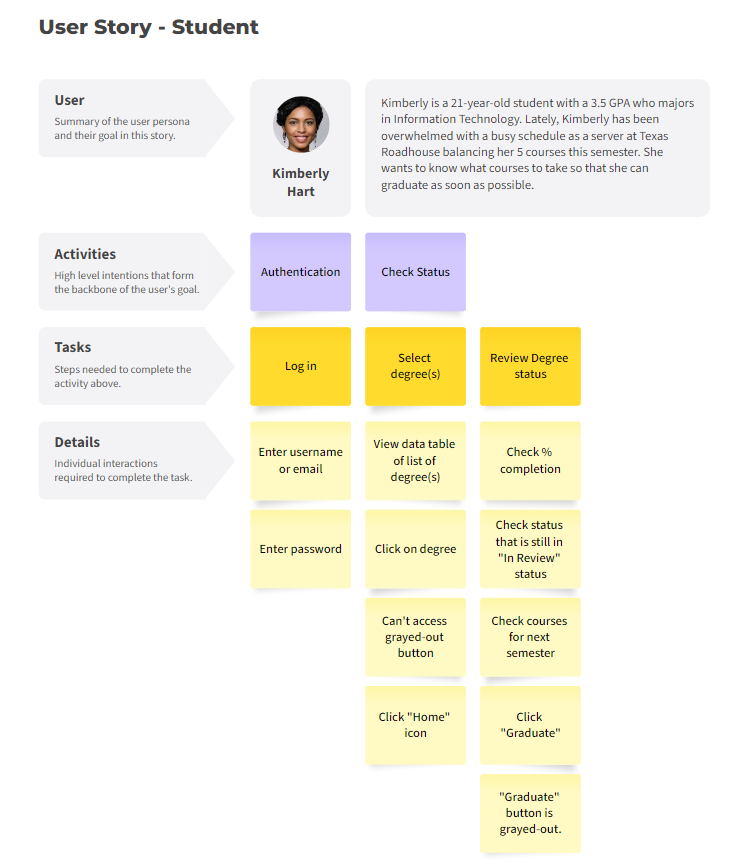




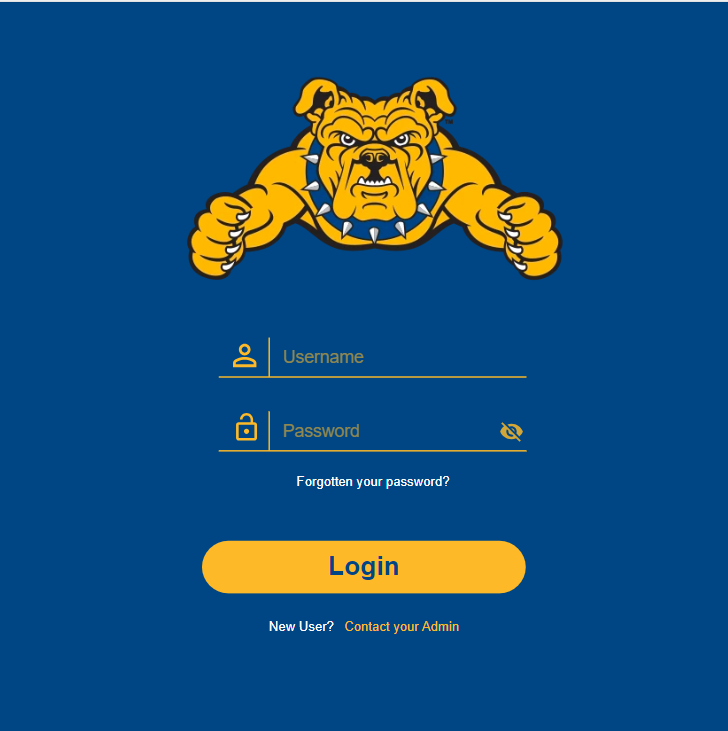


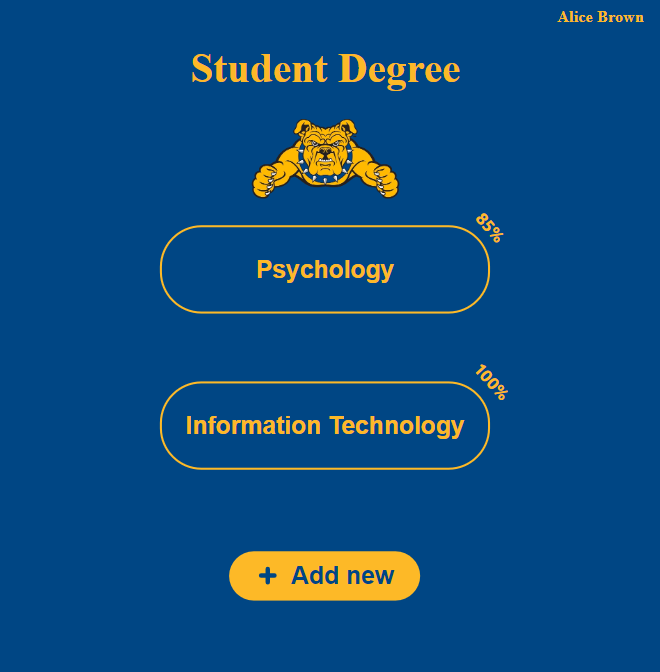


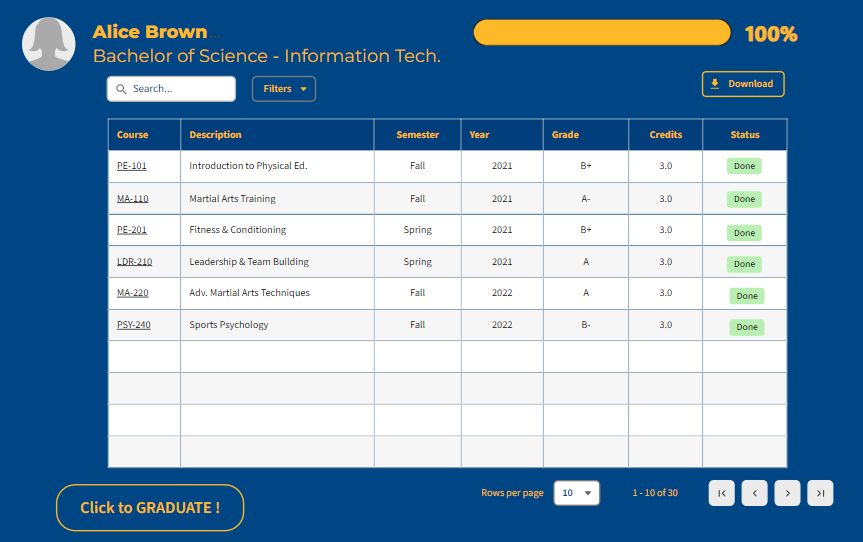
****

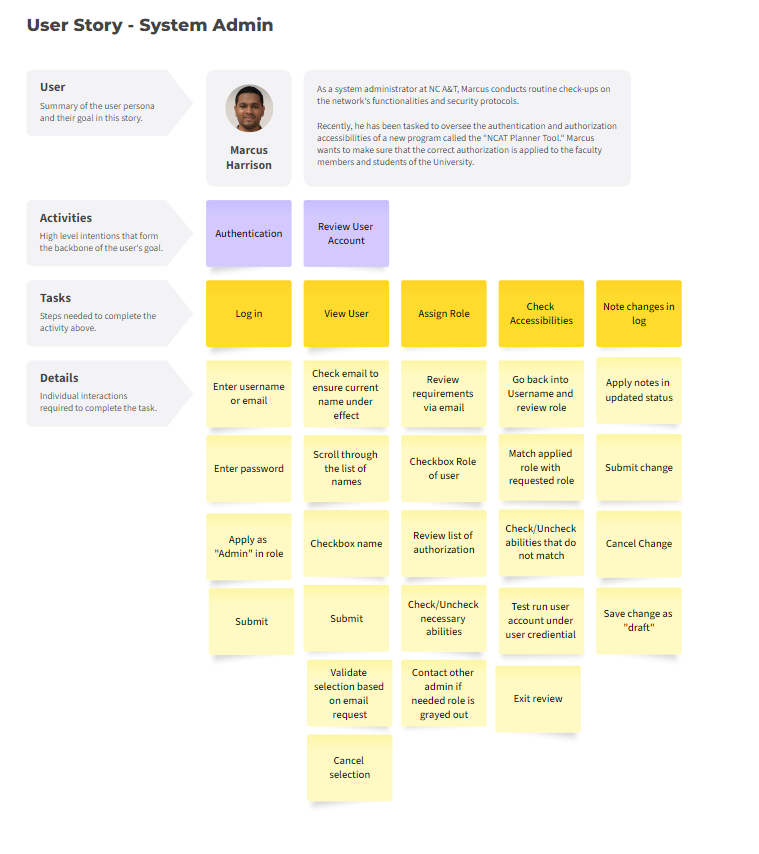
****



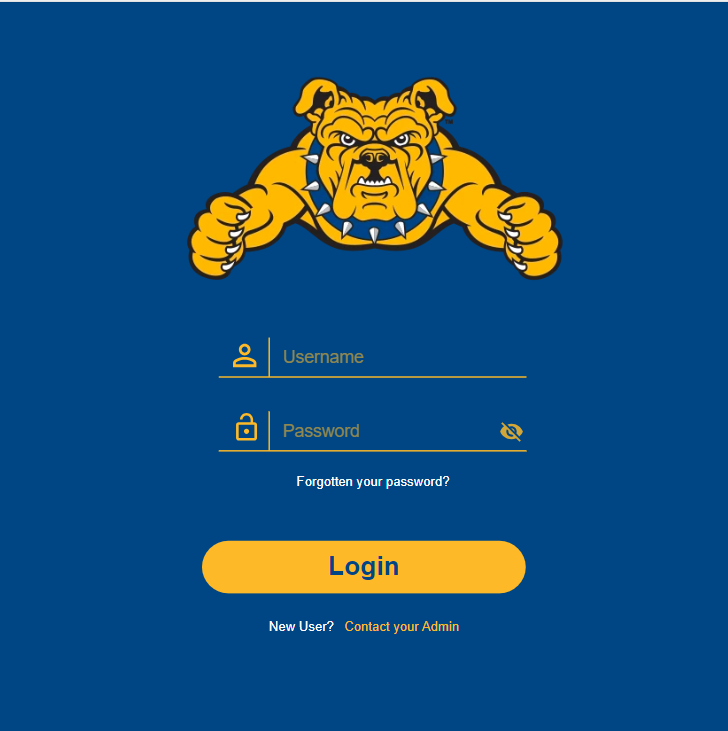


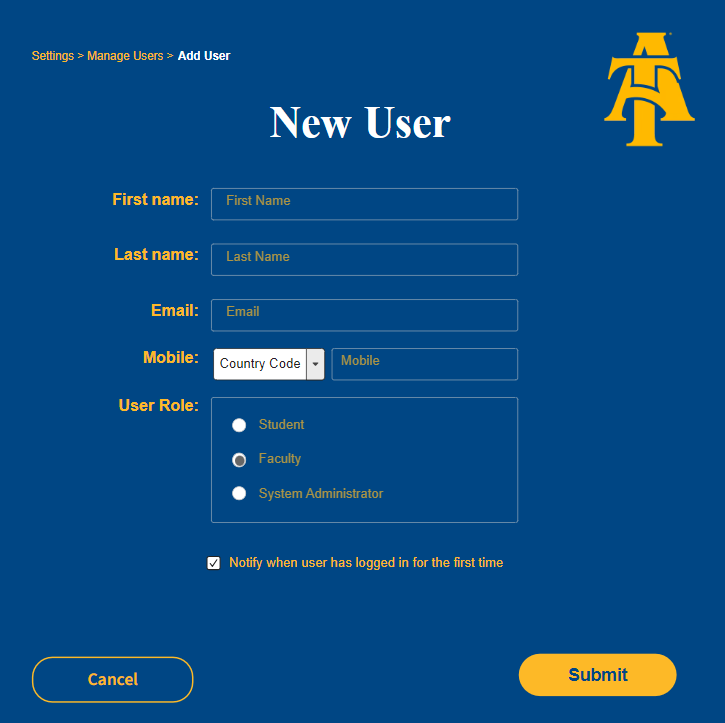
****

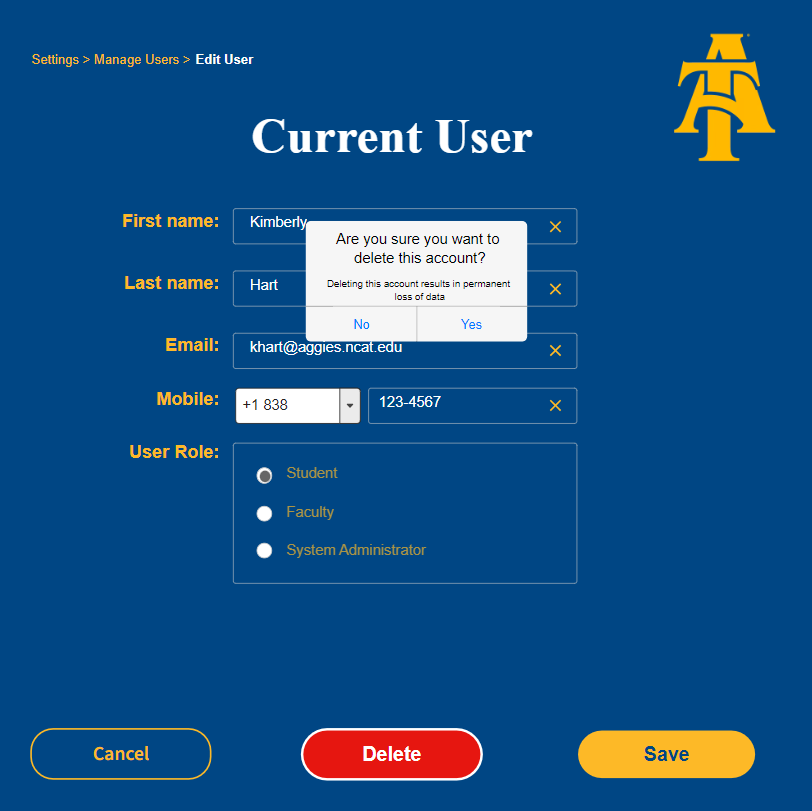
****

****

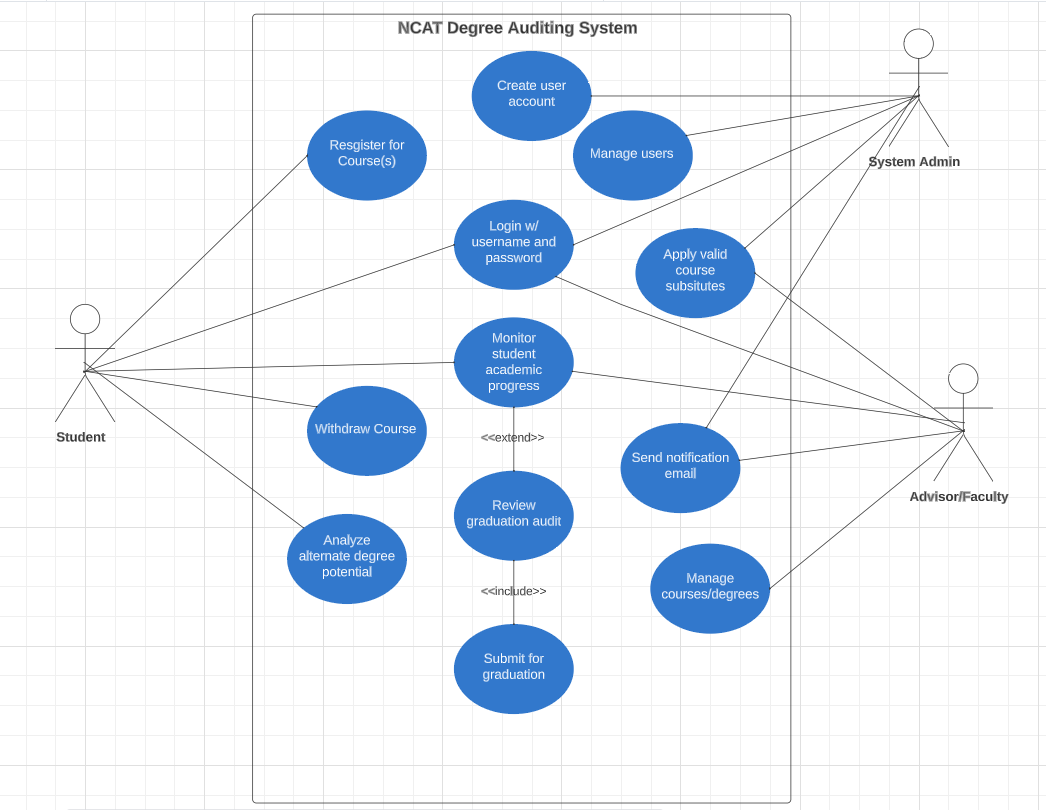








## UML Use Case Diagram



# Future works including revised project schedule

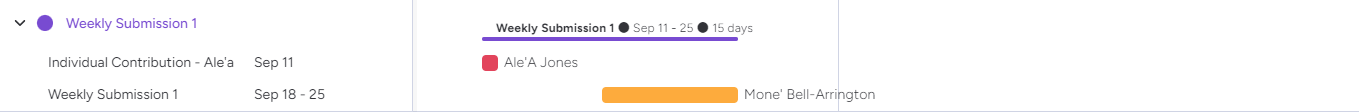
Most tasks are based upon assignments provided on Blackboard for CST-498.01A which is one of the courses in NCAT. There will be 3-4 members in this group with a balanced number of tasks and workload as some tasks require a heavier workload than the others. Each member will do their best in completion of each task within the given deadline if not before it.

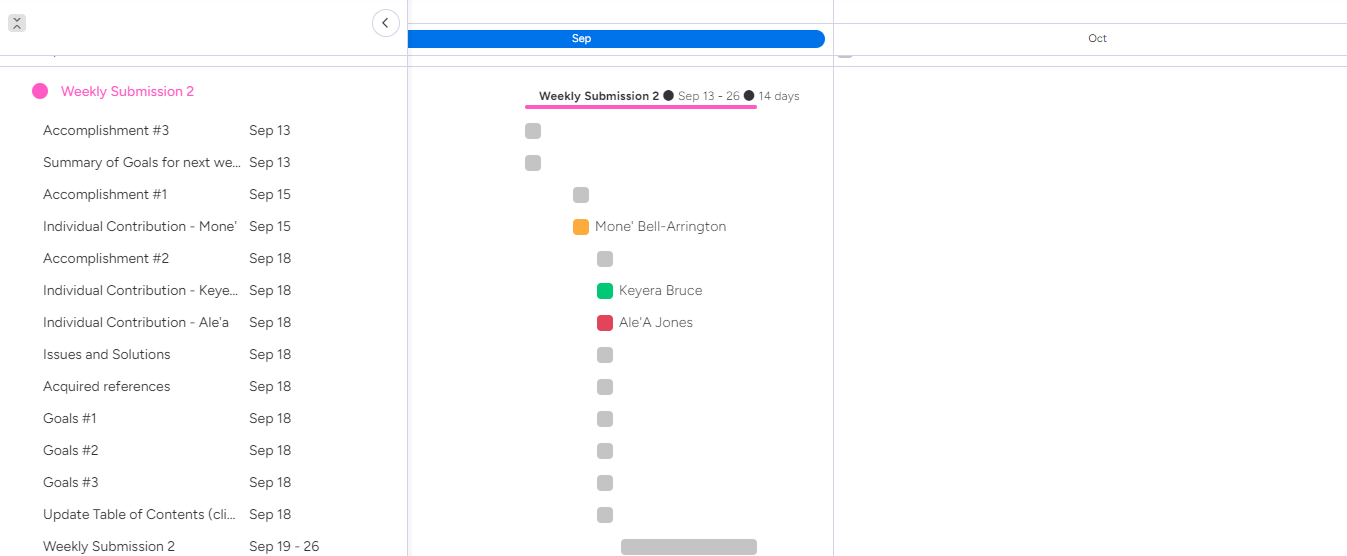
If a workload or quantity of tasks is overwhelming, the member responsible for the task must communicate with other members so as to assist in the completion without deprecating the deadline. The following snapshots below indicate this balance on a week-by-week basis with concurrent tasks, deadlines and assignees.

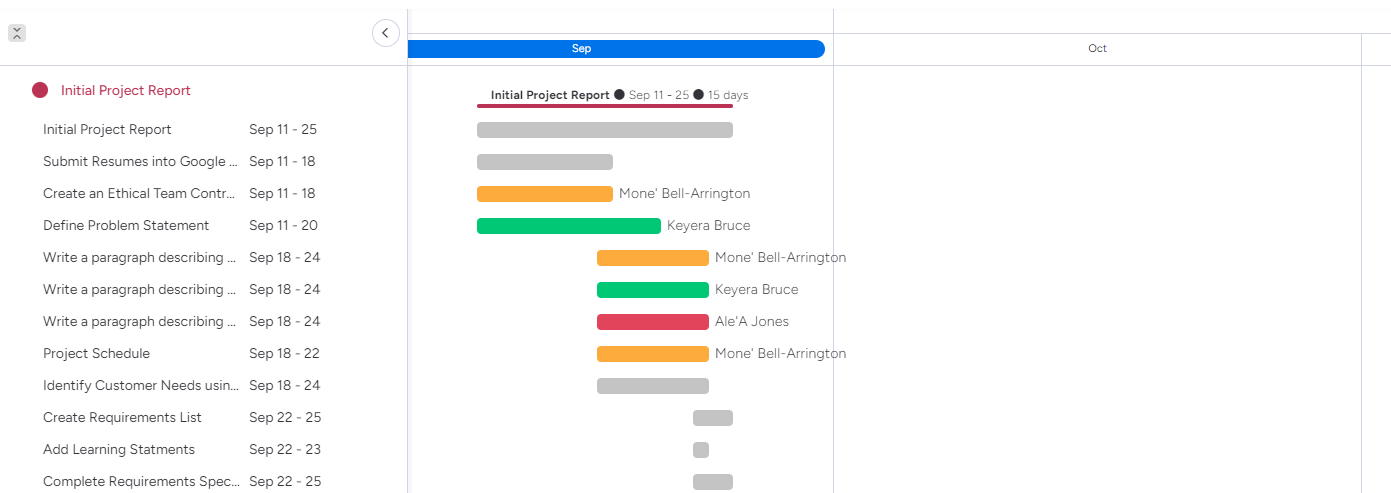
## Bar Chart - Task Balance (CST-498)

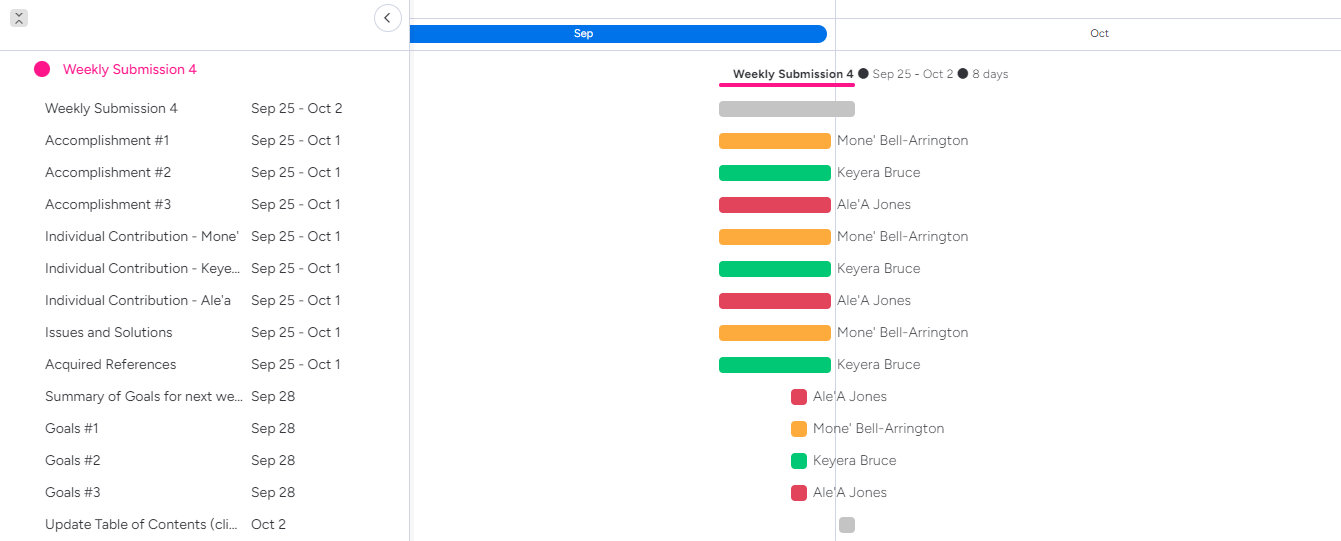


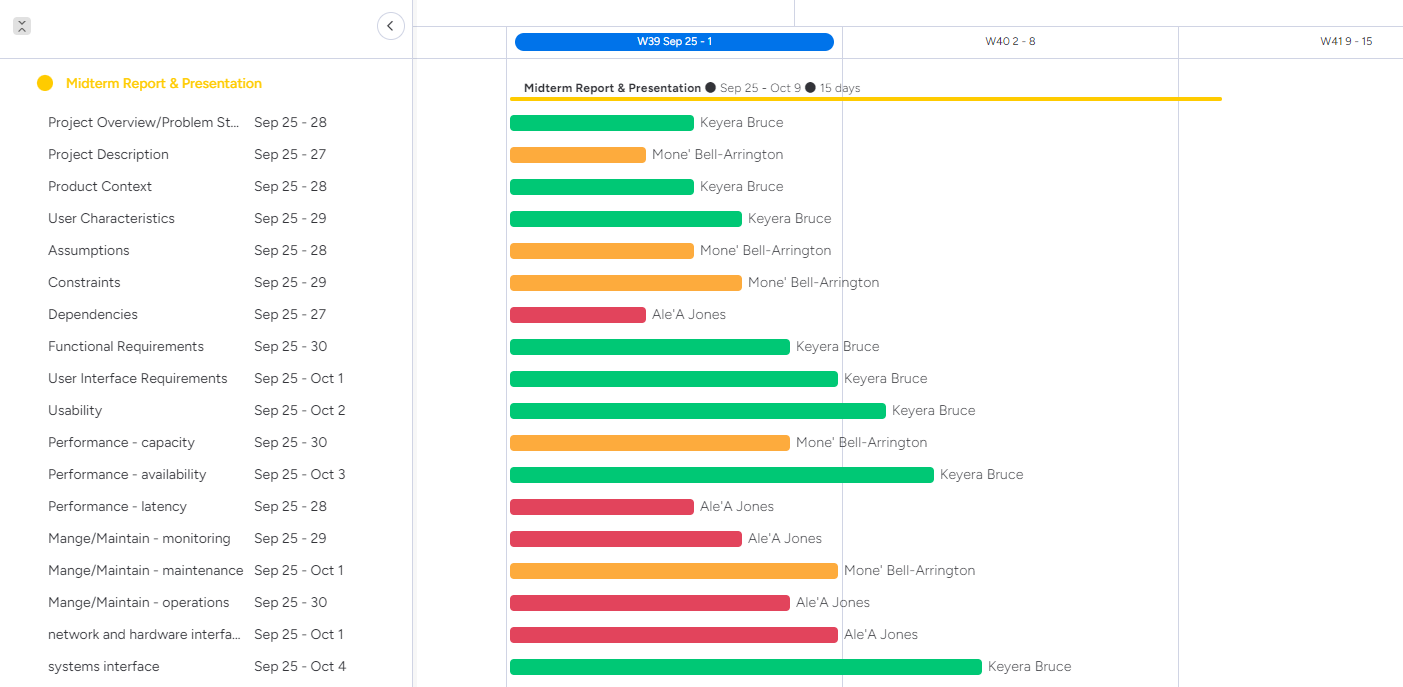
## Gantt Chart - Project Schedule (CST-498)

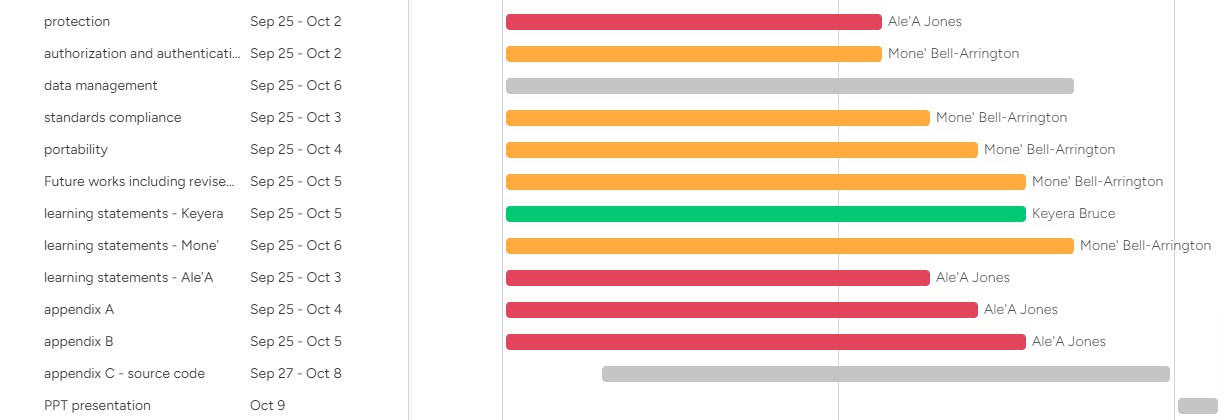


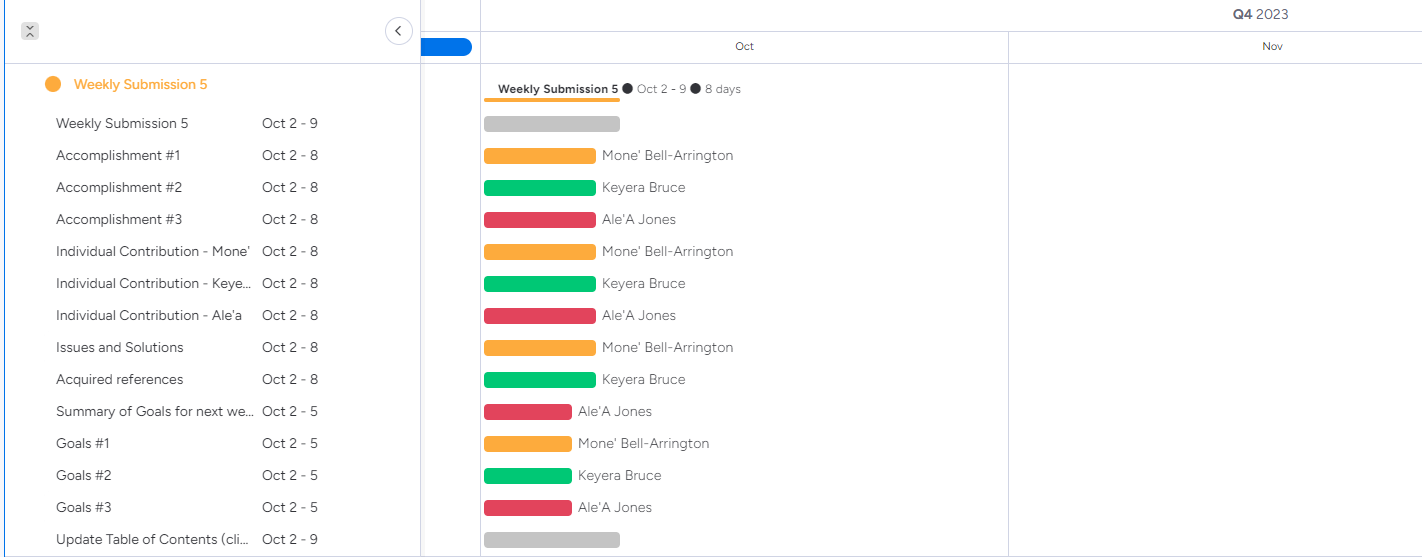


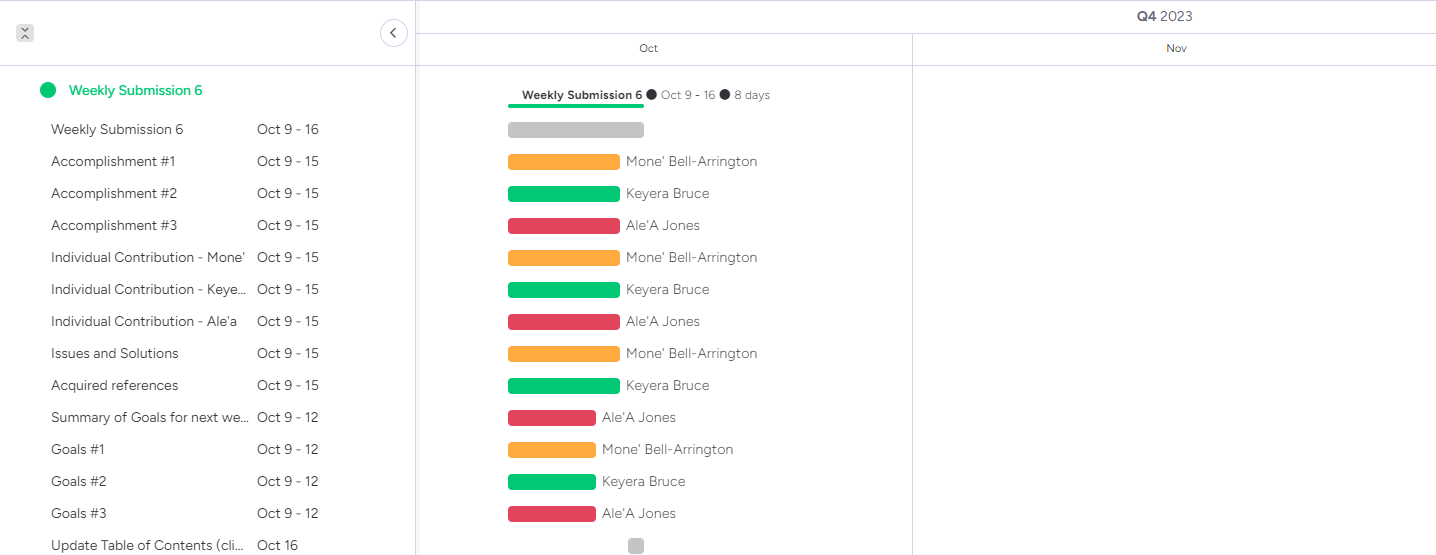




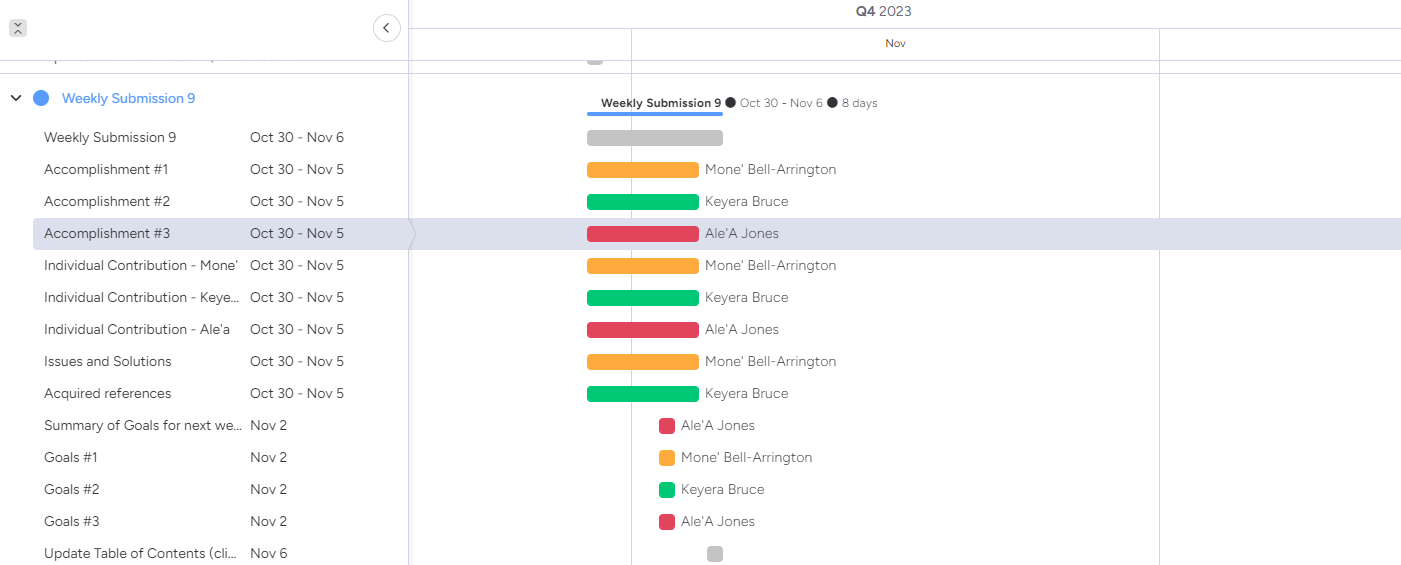


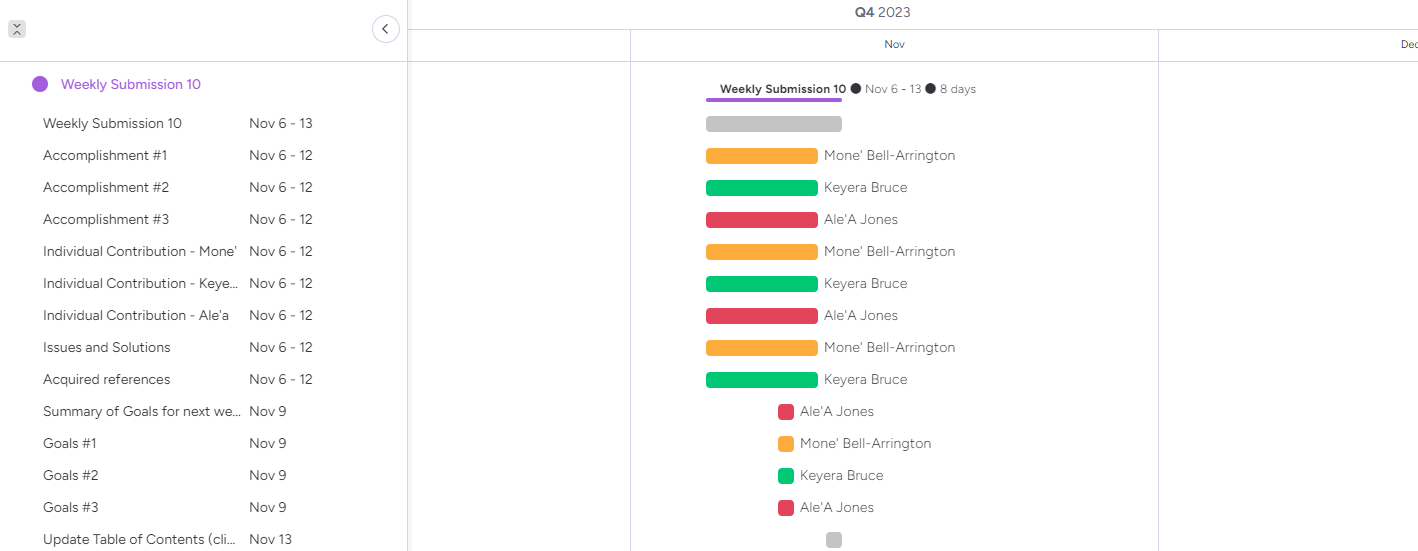




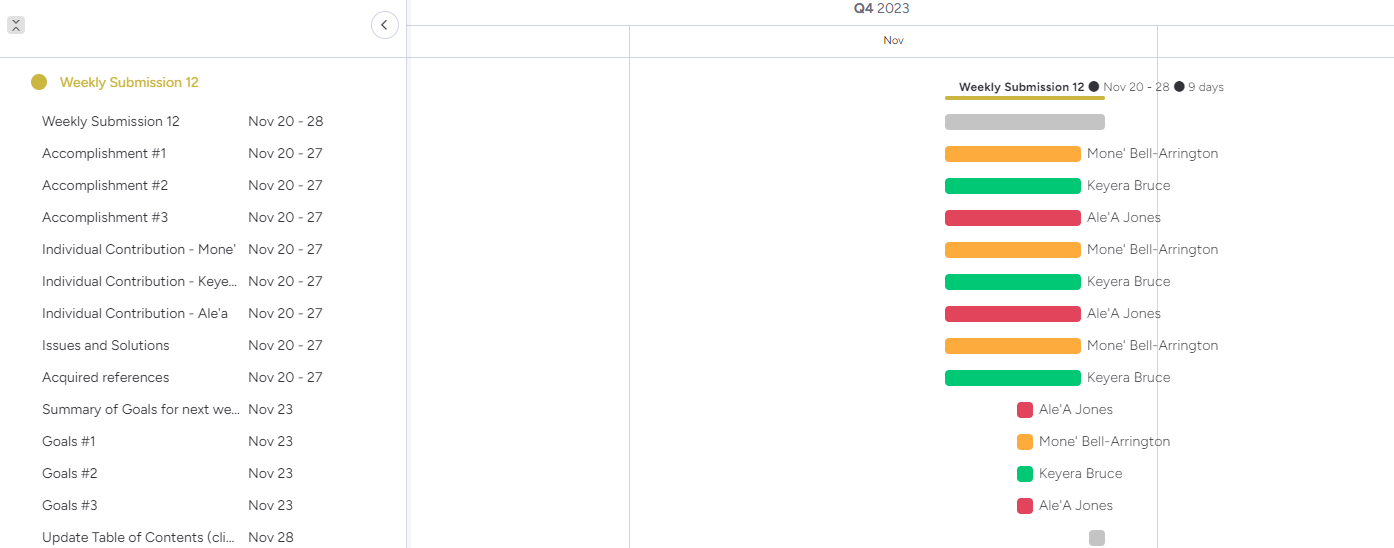


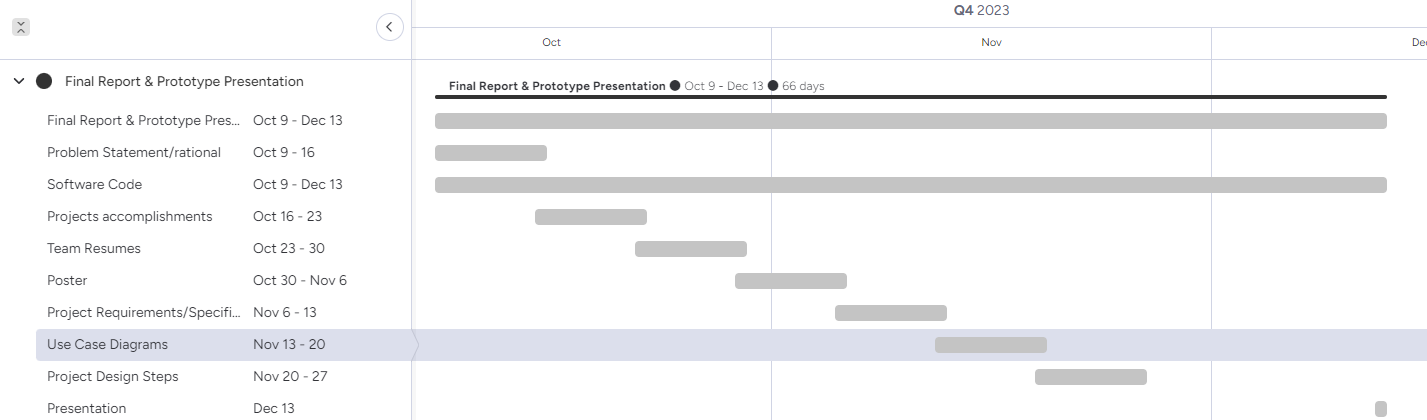




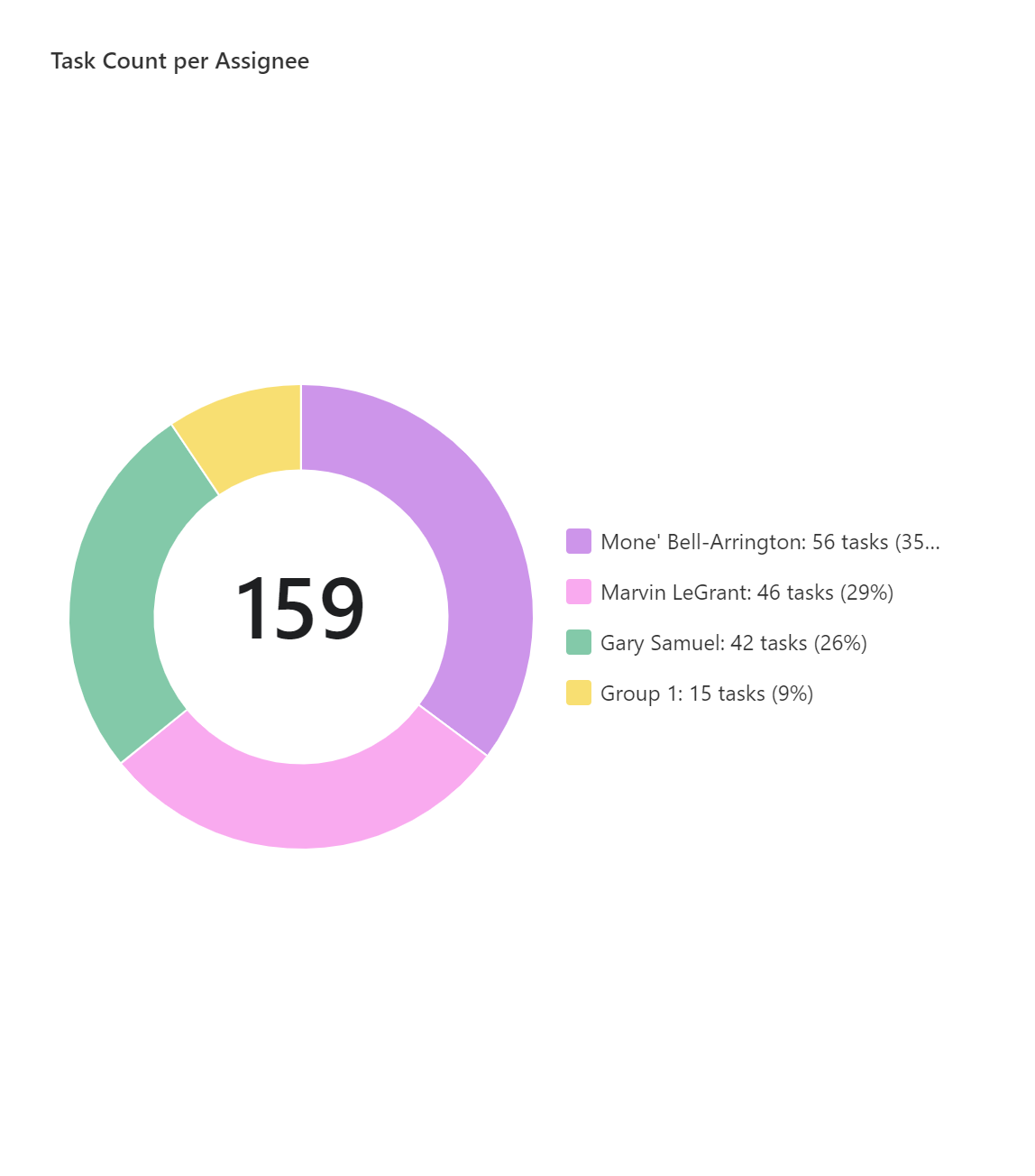




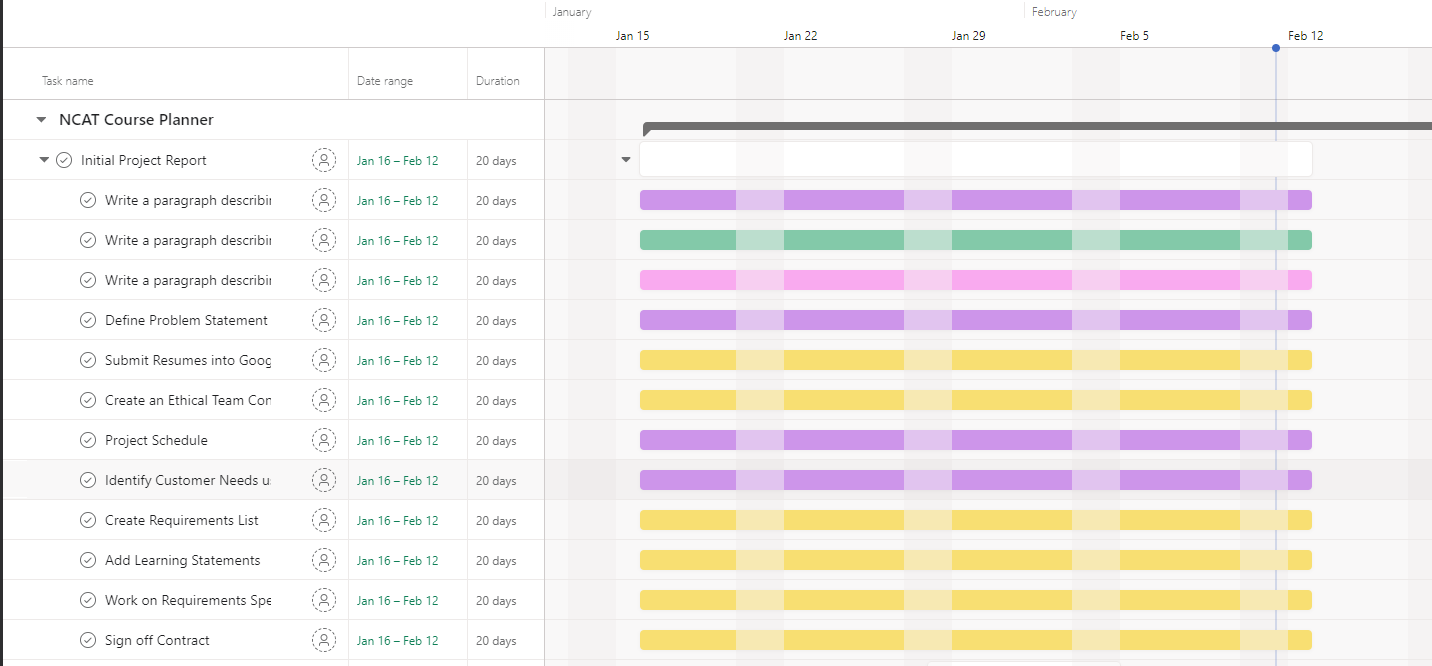


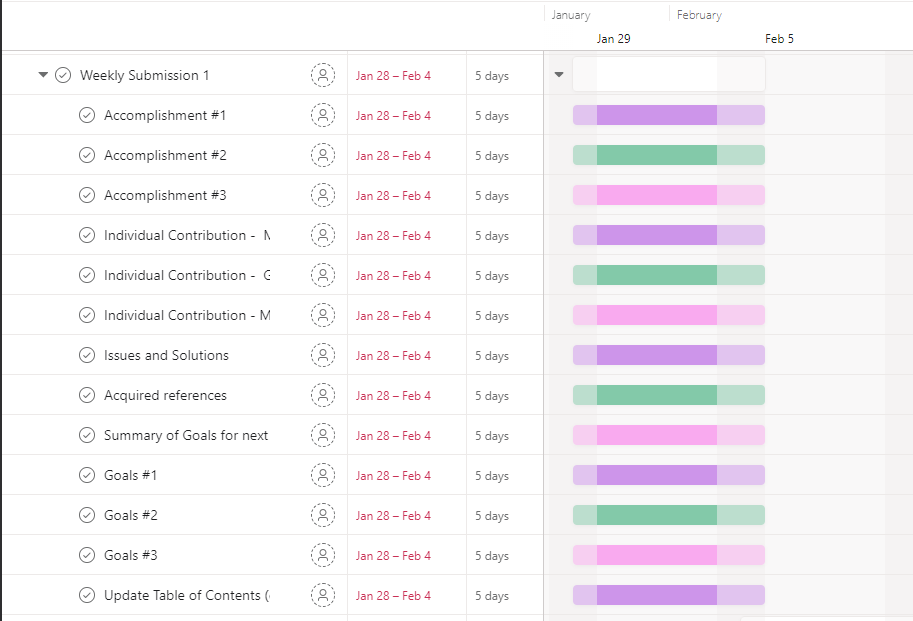


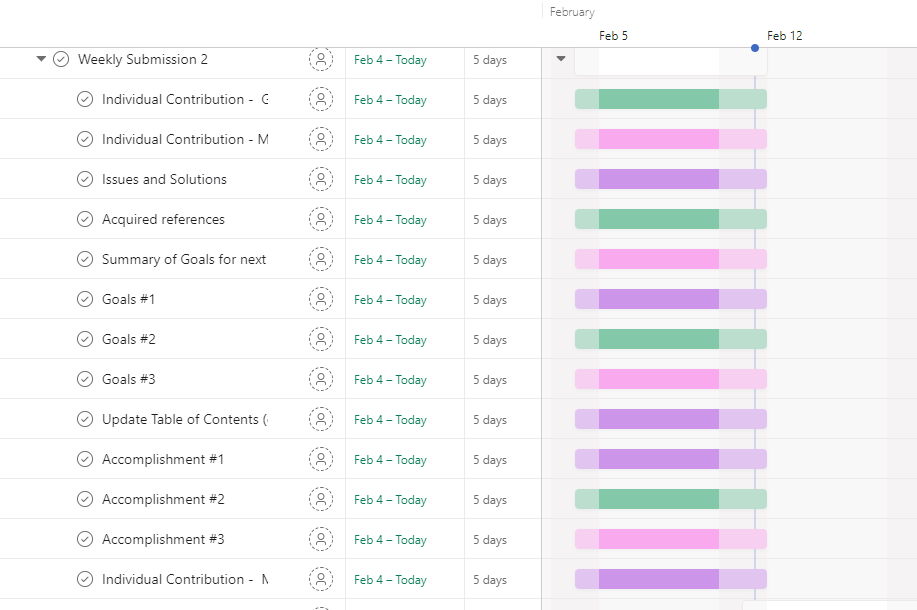
## Bar Chart - Task Balance (CST-499)

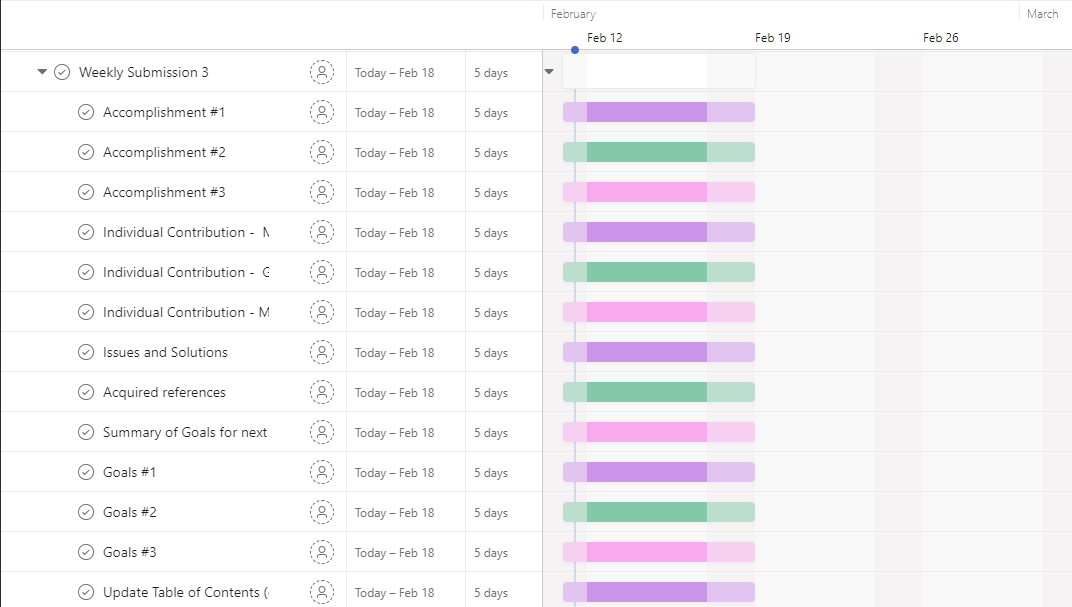


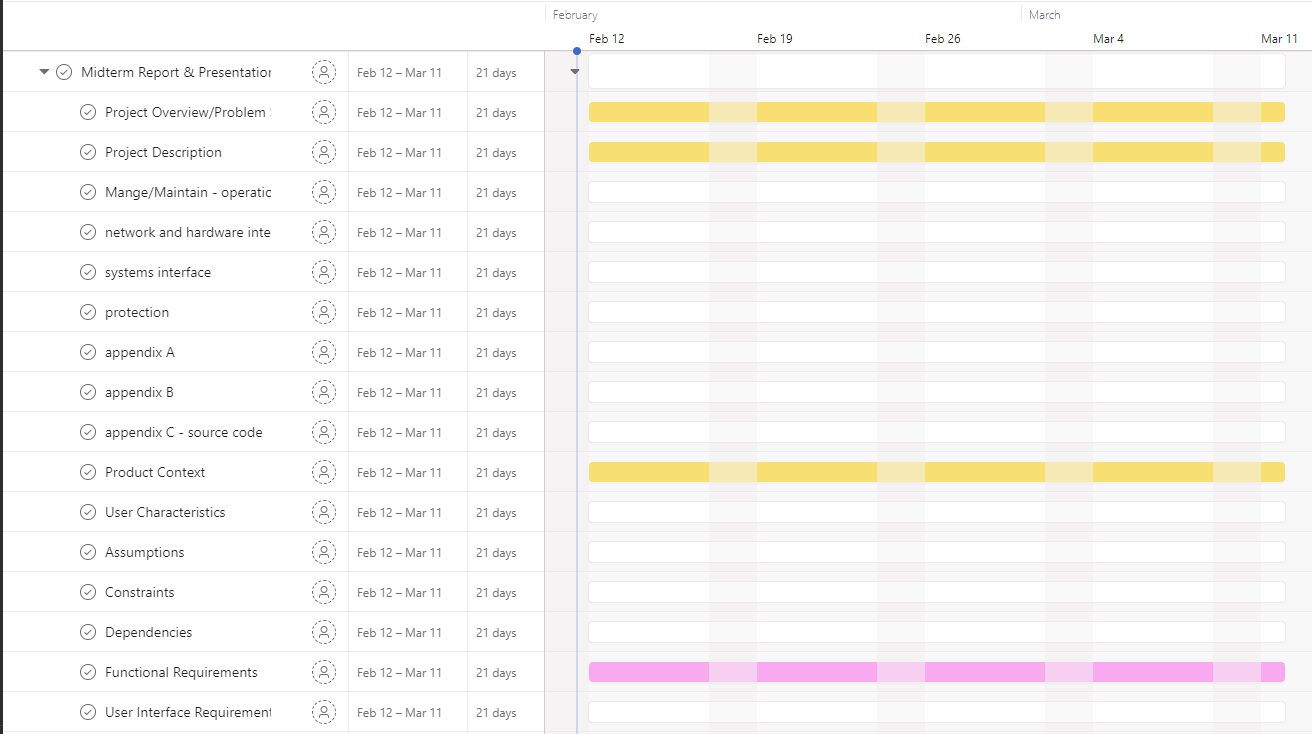
## Gantt Chart - Project Schedule (CST-499)

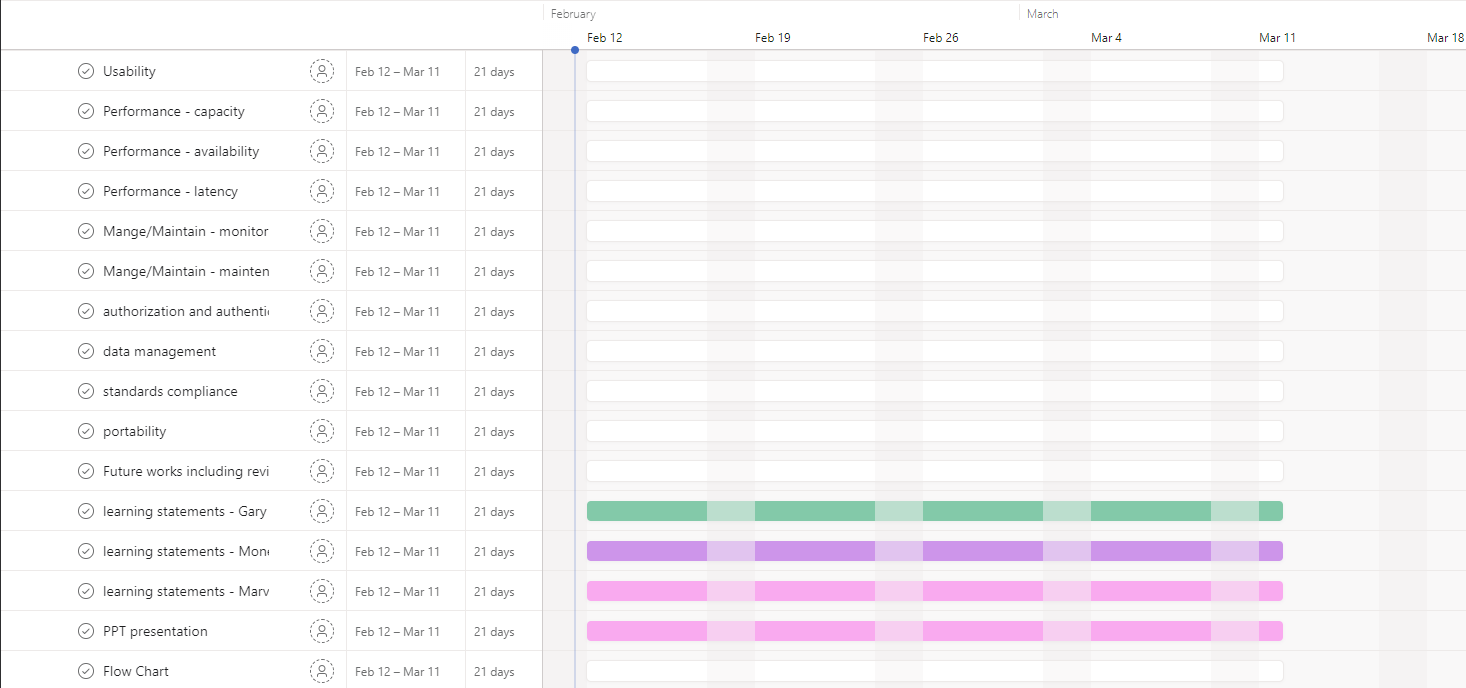


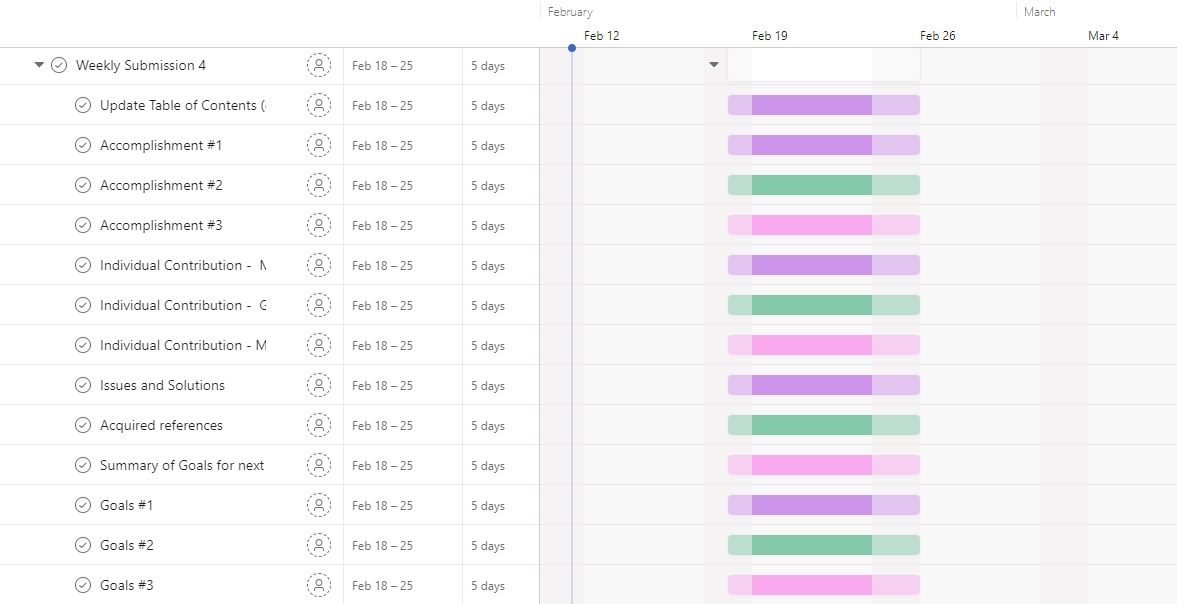


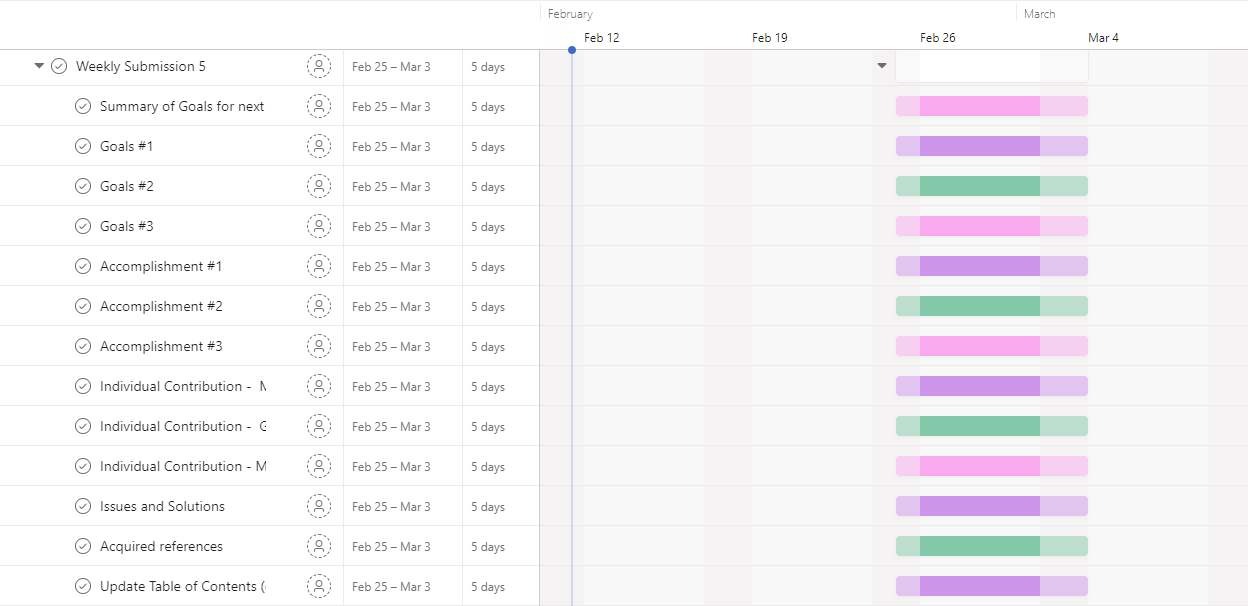


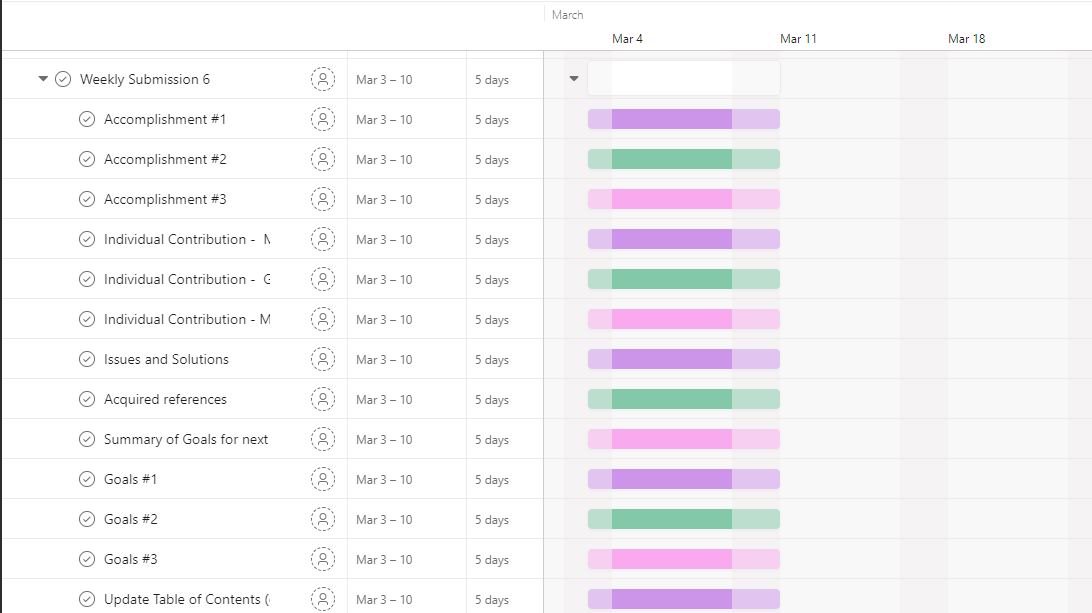


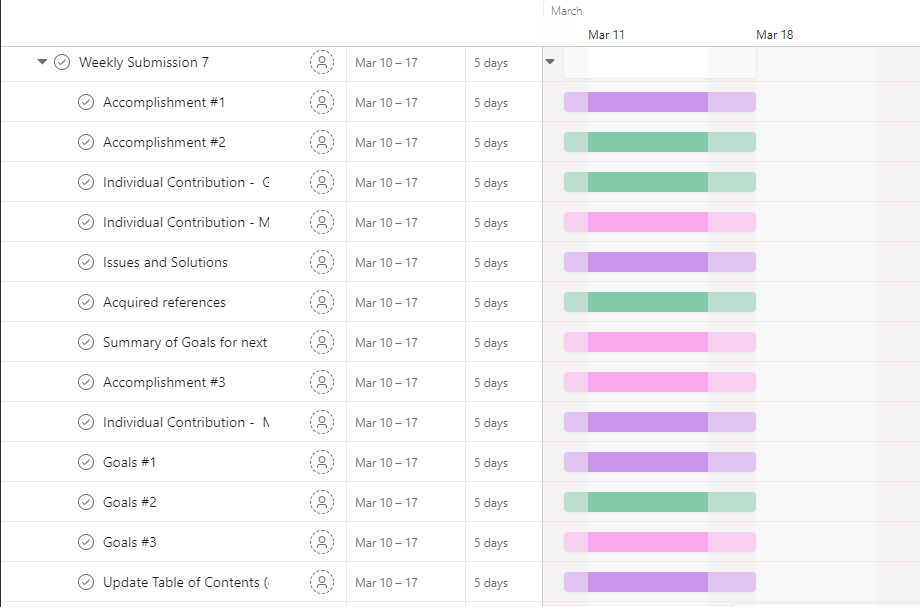


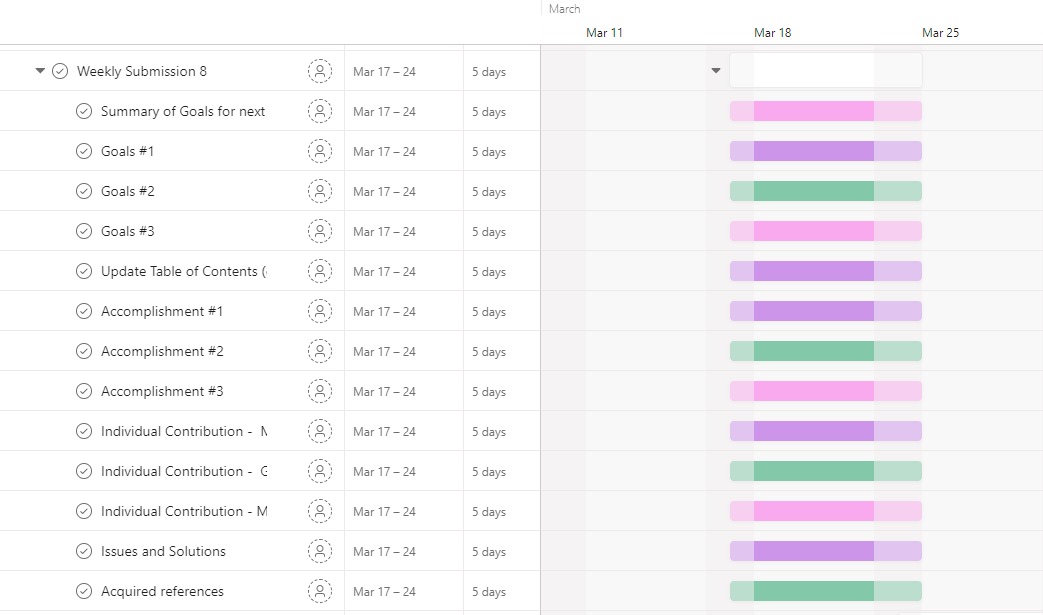


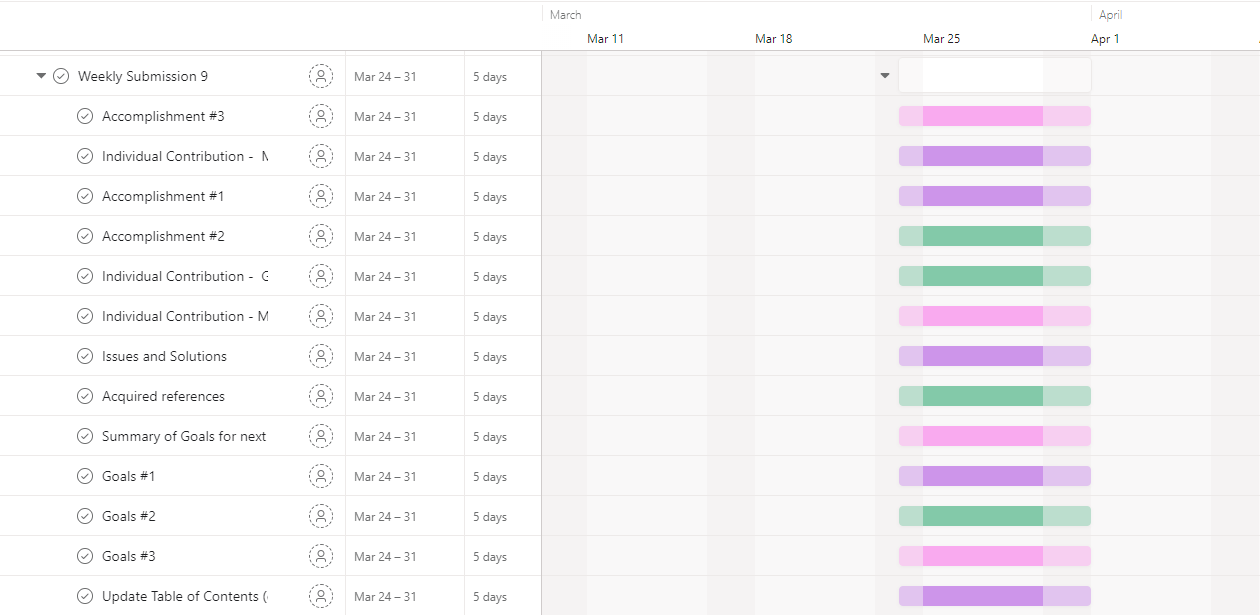


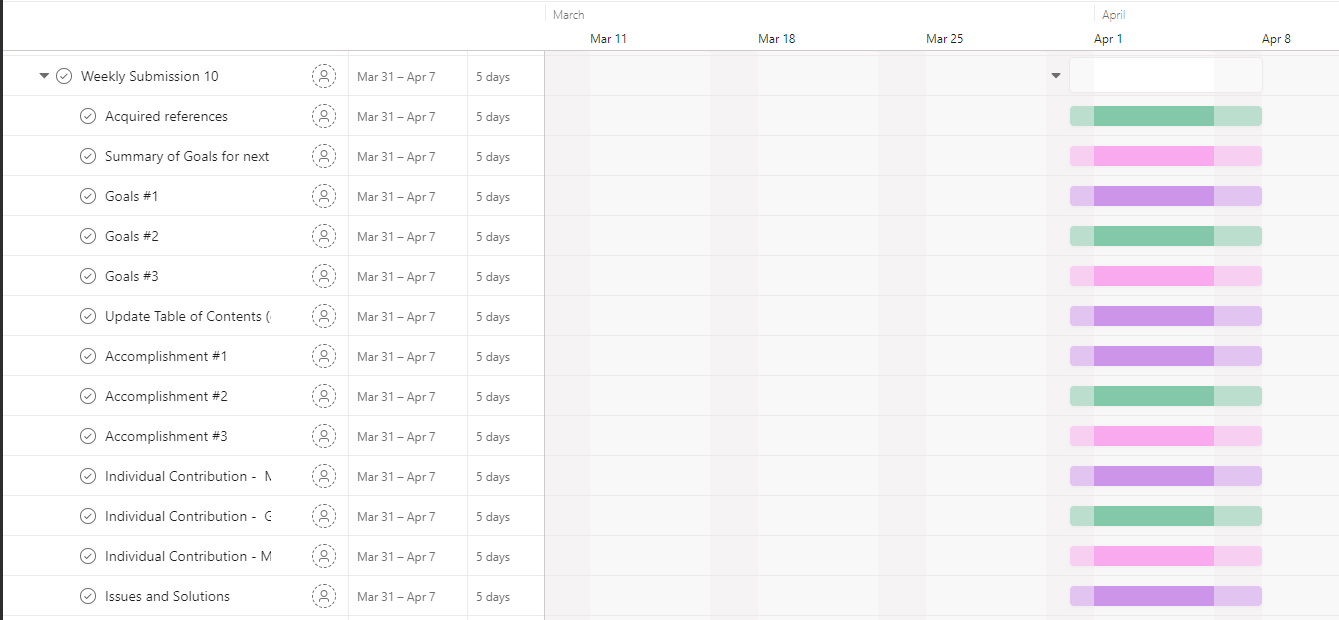


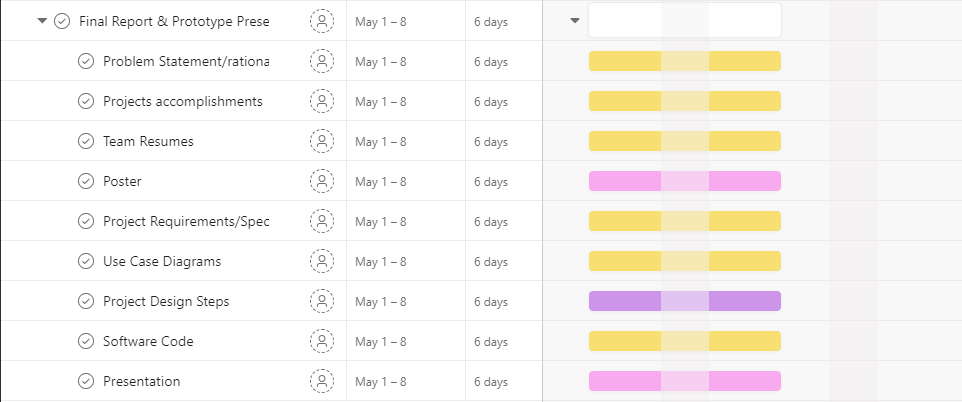












# Learning Statements

## Keyera Bruce (Fall 2023)

* + 1. This is the first time that I have been introduced to such a document like the Requirements Specification. I have learned that the Requirements Specification document is a wonderful tool to organize several sections of a project to ensure that every aspect is completed to the fullest. It has also helped the team split the workload and focus on their own portions in order to ensure that each team member can complete their part with ease.
    2. The time and effort that is required to complete the project is tedious. The weekly submissions are there to keep the group on top of the work towards the final product and that everyone is on the same track from start to finish. I have learned that this could be a great step-by-step process to use with any future projects.

## Mone’ Bell-Arrington (Fall 2023)

* + 1. I understand that one factor of a program’s success stems from creating a solid-structured team dynamic with a balanced skill set among members that will further assist and conclude responsibilities prior to engaging in tasks throughout the project’s creation and competition.
    2. Time management and organizational skills can never be undermined. In a real-world environment, the majority of creating a program isn’t from writing code; rather, the plans set forth to create the program takes the majority of the software development time. This isn’t only applied to software development. This seems like a general concept for anything that requires designing and brainstorming as a procedure to an end result.
    3. Most of the completed sections in the SRS document and the procedures of creating a program from scratch are refreshers from our previous courses applied into one course. It is the application of all that has been learned that may be difficult.

## Ale’A Jones (Fall 2023)

* + 1. This project has helped me be more efficient in my research. I have done deep dives and used multiple resources. This has helped further my ability to learn and teach myself. I have learned that it takes time and can be difficult but the opportunity to work on a project like this makes it worth it.
    2. I have learned the hard lesson that time management is important. It has always been difficult for me to get things done early and not wait until the last minute. I continue to seek the best ways to enhance my time management and this project has helped me discover some of those ways. For instance just getting something done each day is better than nothing.
    3. I have also learned how to work in a team. This is my first big project that involved me working on a team. It has been challenging because of my procrastination but it has taught me now only how to juggle multiple tasks but also collaborate with others who also have busy schedules.

## Marvin LeGrant (Spring 2024)

* + 1. I gained a comprehensive understanding of database management principles and practices. This included designing relational database schemas, writing complex SQL queries with joins, normalizing data for efficiency, and optimizing performance through indexing and query analysis. I learned to ensure data integrity using transactions, automate tasks with stored procedures and triggers, and manage user permissions for security. Additionally, I acquired skills in database backups, using tools like MySQL Workbench, and troubleshooting common issues such as slow queries and deadlocks. Implementing replication and clustering for high availability and scalability, optimizing schema design, and securing sensitive data were also key aspects of my learning. Overall, the project provided me with valuable insights into MySQL database administration, performance optimization, and data security measures.
    2. I have learned how to set up an Azure application involved creating and configuring resources such as virtual machines, databases, and storage accounts in the Azure portal. I gained experience in deploying and managing applications using Azure App Services, containers, and server-less technologies like Azure Functions. Understanding networking concepts such as virtual networks, security groups, and load balancers was essential for ensuring the application's availability and security. Lastly, I learned how to monitor and scale the application using Azure's monitoring tools and auto scaling capabilities to meet changing demands efficiently.

## Gary Samuel (Spring 2024)

* + 1. The project introduced me to the strategic use of a Requirements Specification document, which was instrumental in organizing the project's various components. This approach not only facilitated a clear division of tasks among team members but also ensured a thorough and systematic completion of each project segment.
    2. Beyond reinforcing the importance of time management and organizational skills, this project expanded my technical proficiency. I learned to set up a front-end application service for hosting on Azure, fork projects using GitHub, connect the front-end to GitHub on Azure, and create actions for deployment. Additionally, I gained hands-on experience in database management by creating a MySQL instance, setting up firewall rules for the MySQL server, and connecting to the database using Oracle Workbench.
    3. Working on this team project not only honed my ability to collaborate effectively within a diverse team but also significantly improved my self-learning capabilities. I navigated through complex tasks such as integrating front-end services with GitHub on Azure and managing database connectivity, which taught me the intricacies of deploying a full-stack application and the critical nature of secure database management within a collaborative project framework.

## Mone’ Bell-Arrington (Spring 2024)

* + 1. For this semester’s portion of the project, I would like to further understand the concrete structures of the web application and what other tools it would take to implement the program at a full scale. With different perspectives and members added into our team, it creates a further dynamic insight of skills and what others may bring to the table.
    2. A deeper display of Project Management skills is also something that I would like to instill into my experiences. During this semester, I am taking a course in Project Management with very little hands-on experience in leadership capabilities. Other members have had experience in leadership roles that I can absorb throughout the project’s stages.

# 

# APPENDIX

## Definitions, Acronyms, and Abbreviations

**Azure** Microsoft Cloud Based SAAS suite

**AES** Advanced Encryption Standard

**Asana** Project Management Web Tool designed to help teams manage tasks, organize projects, and leverage workload and maintainability.

**Blackboard** A learning management system that provides customizable architecture through a web-based virtual learning environment.

**CSS** Cascading Styling Sheets

**Gantt** A bar chart that illustrates a project schedule.

**Gmail** A free email service provided by Google.

**Google Drive** A cloud-based storage service that enables users to store and access files online.

**GroupMe** Free group messaging app.

**HTML** Hypertext Markup Language

**MFA** Multi-Factor Authentication

**MySQL** An Oracle-based relational database management system.

**NCAT** North Carolina Agricultural and Technical [State University]

**OS** Operating System(s)

**OWASP** Open Web Application Security Project

**PHP** PHP: Hypertext Preprocessor

**SQL** Structured Query Language

**TeamGantt** A Cloud-based project management software that focuses on creating Gantt charts.

**TLS** Transport Layer Security

**TOTP** Time-based One-Time Password

## References

* + As a group, we figured that web-scraping should be incorporated into our prototype even though we are not the most familiar with it. Python is a well-known programming language and would be a great tool to use as well in this process. Upon further research, there is a great Python Web Scraping Tutorial on YouTube that we should find the most helpful. Source: [Python Web Scraping Tutorial • Step by Step Beginner's Guide](https://www.youtube.com/watch?v=Ew44dS0mw-E)
  + The Capstone Assignment 1 powerpoint was also a great tool to let us know where to start and how we can assist and get to know each other as a team. The entire process is broken down into parts and explained thoroughly so we know what to work on moving forward to get the result we are looking for in the end: [CST 498 Capstone 1 Assignment 1.pptx](https://docs.google.com/presentation/d/1iRycdTpxGBnml--lmxO0NY9jc7edAuSn/edit?usp=sharing&ouid=101493969900022104669&rtpof=true&sd=true)
  + A step-by-step process on how to build a requirements list to ensure the project’s success. This will allow the group to manage the project requirements throughout its timeline to make sure that all features are implemented. Source: [6 Steps to Requirements Gathering for Project Success [2023] • Asana](https://asana.com/resources/requirements-gathering)
  + The group needed to create a project schedule in order to list what assignments need to be done at what date for better time management. The Gantt Chart Maker was a great tool that assists in the planning and scheduling of projects, including ours. It is a graphical representation of activity against time. Source: [TeamGantt](https://www.teamgantt.com/)
  + One of the references that the team has used this week is the conversation that one of the team members had with Professor Hogan about where we are at with the project. The professor gave the team advice about the project and what we are expected to do based on what type of project we wanted to accomplish. The team communicated through email and during student hours.
  + Researched examples on how to start out the project source code for the NCAT course planner. The programming languages that could be used to achieve this project includes HTML/CSS, Python, MySQL, Java, etc. Each team member has been introduced to these languages in previous courses, so this is a great opportunity to create a project that implements all of them.
  + Resume Builder: All resumes have been acquired and reviewed during Thursday’s meeting with great feedback about opportunities and responsibilities that best fits each member. There have been notices of resume building opportunities and other resources that could assist with professionalism. The following sites are listed:

[Flowcv.com/app/resume/content](http://flowcv.com/app/resume/conten) - Resume builder and CV creator with formats and themes that best fits the user's work/school experiences. It is free and provides collaborative insight by creating a sharable link and a section to comment on the resume/CV.

[Discord (CS Career Hub)](https://discord.com/channels/334891772696330241/973598253620789308) - Initially a gaming platform that has expanded into an interest-based collaborative communication platform for anyone who wants to join or make groups. There is a specific group chat that supports rising Computer Science/ Information Technology students with a section that professionally reviews resumes of all levels

## Source Code

### Backend/Frontend

* + - [Github Repo](https://github.com/CrochetVandrella/course_planner)
    - [Azure](http://ncat-cp.azurewebsites.net)
    - [CodePen.io](https://codepen.io/mvbellarrington)