
Ticket 3 SRS

CS5213 Group B

Feb 10, 2024

CONTENTS

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Scope | 1 |
| 1.2 | Purpose | 2 |
| 1.3 | Definition | 2 |
| 2 | Description | 3 |
| 2.1 | Overview | 3 |
| 2.2 | Architecture | 4 |
| 2.3 | Features | 4 |
| 2.4 | User Characteristics | 6 |
| 2.5 | Constraints | 7 |
| 2.6 | Operating Interfaces | 8 |
| 3 | Requirements | 9 |
| 3.1 | Functional Requirements | 9 |
| 3.2 | User Registration | 9 |
| 3.3 | Detail Data Required/User Inputs | 13 |
| 3.4 | For Project Proposal and Approval | 15 |
| 3.5 | For Project Repository | 17 |
| 3.6 | For Team Formation | 19 |
| 3.7 | For Communication and Collaboration Platform | 21 |
| 3.8 | For Milestone Tracking | 21 |
| 3.9 | For Feedback and Evaluation | 22 |
| 3.10 | For Calendar and Notifications | 23 |
| 3.11 | For Reporting and Analytics | 24 |
| 3.12 | External Interface Requirements | 24 |
| 3.13 | Non Functional Requirements | 26 |
| 4 | References | 30 |

INTRODUCTION

1.1 Scope

When defining the scope for our capstone project management system, we should focus on what we are aiming to achieve and the advantages it offers. We are looking at making project management smoother for everyone involved, ensuring that communication is easy and effective, and that projects are completed efficiently and to a high standard. This approach not only meets our main objectives but also provides real value to students, faculty, and administrators alike.

1.1.1 Benefits

Our system is designed to transform the way capstone projects are managed and experienced. Here's how:

- **Enhanced Collaboration:** We're making it easier than ever for students, faculty, and administrators to work together. Our platform ensures that everyone can communicate and collaborate smoothly, breaking down traditional barriers and fostering a community of shared knowledge and support.
- **Efficiency in Management:** Say goodbye to the complexities of managing capstone projects. From the initial proposal to the final submission, our system streamlines every step. This not only saves precious time but also reduces the resources required to manage projects effectively.
- **Improved Quality of Work:** We believe in empowering students to produce their best work. Our system provides the tools and frameworks necessary for students to excel, including structured feedback and clear milestones. This guidance helps elevate the standard of projects, ensuring students can achieve their full potential. By focusing on these key benefits, our capstone project management system is set to revolutionize the educational experience, making project management a breeze and helping students to not just succeed, but to thrive.

1.1.2 Objectives

Our system is built with two core objectives in mind, aimed at enhancing the educational landscape:

- **Creating a Unified Platform:** We're bringing every aspect of capstone project management together into one easy-to-use platform. This means less hassle and more focus on what truly matters—learning and innovating.
- **Supporting Educational Goals:** Our platform does more than just manage projects; it's designed to boost student learning and help push the boundaries of research and innovation. By aligning our system with the broader objectives of educational institutions, we are helping to cultivate a richer, more productive learning environment. By focusing on these objectives, we're committed to providing a system that not only simplifies project management but also actively contributes to the educational journey.

1.1.3 Goals

Our system is designed with clear goals to enhance the educational experience: Boost Student Engagement and Success: We are equipped with a full range of project management tools aimed at engaging students more deeply in their work, thereby increasing the success rates of capstone projects.

- **Empower Faculty and Administration:** We provide a comprehensive set of tools for monitoring, feedback, and evaluations, making it easier for faculty and administrators to support student projects effectively.
- **Facilitate Data-Driven Decisions:** Our advanced reporting and analytics capabilities enable informed decision-making, helping to refine project guidelines and enhance support structures for future cohorts. This document outlines our commitment to improving the educational process for all parties involved, through a focused approach on engagement, support, and informed improvements.

1.2 Purpose

A Software Requirement Specification (SRS) forms the basis of an organization's entire project. It sets out the framework that all the development teams will follow. It provides critical information to all the teams, including development, operations, quality assurance (QA) and maintenance, ensuring the teams are in agreement. Using the SRS helps an enterprise confirm that the requirements are fulfilled and helps business leaders make decisions about the lifecycle of their product, such as when to retire a feature. In addition, writing an SRS can help developers reduce the time and effort necessary to meet their goals as well as save money on the cost of development. In a Capstone Management System, the SRS is a bridge between the stakeholders (like students, teachers, and school officials) and the people who make the system (the developers). It's a very detailed plan that tells exactly what the system should do and what limits it has. This is super important because it makes sure that everyone agrees on what the final system will look like and what it can do. The SRS makes talking about the system easier because everyone can look at it and understand the project better. When you have a big project with lots of parts, this is crucial to keep everything on track.

The SRS also helps a lot with building the system. It's the map that developers follow when they are designing, making, and checking the system to make sure it works right. It tells them what the system needs to do, how fast it should do it, and how it will fit in with other systems. When planning the project, the SRS is very useful because it gives details that help figure out how much it will cost, what people will need to work on it, and how long it will take. It even helps find possible problems before they happen. Later on, when the system is running, the SRS is still helpful. It can be used to fix any issues or to make the system better over time. Plus, it's a great tool to help write guides and instructions for users so they can understand how to use the system for their capstone projects, from starting a proposal to finishing the project, working with their team, and keeping track of everything.

1.3 Definition

The Capstone Management System is characterized as an advanced web-based platform meticulously engineered to simplify and enhance the orchestration of capstone projects within the educational landscape. It acts as a unifying force, offering a seamless solution for students, faculty, and administrators alike. By seamlessly integrating features such as user authentication, project repositories, communication tools, and analytics, the system aspires to redefine the educational experience by cultivating engagement, operational efficiency, and data-driven decision-making.

DESCRIPTION

2.1 Overview

2.1.1 Background

The **Capstone Management System (CMS)** is a groundbreaking solution designed to transform capstone project management within academic institutions. Driven by a commitment to collaboration, efficiency, and transparency, the CMS redefines project coordination, communication, and assessment in the academic sphere.

2.1.2 Core Purpose

The CMS is strategically engineered to elevate the capstone project experience for various stakeholders. By offering a centralized, feature-rich environment, the system transcends conventional project management limitations, fostering a collaborative and dynamic ecosystem to propel academic and professional growth.

2.1.3 Strategic Alignment

The CMS aligns with the institution's commitment to academic excellence and innovative educational practices. As a technological enabler, it supports the vision of providing a cutting-edge learning environment where technology seamlessly enhances educational outcomes.

2.1.4 Key Objectives

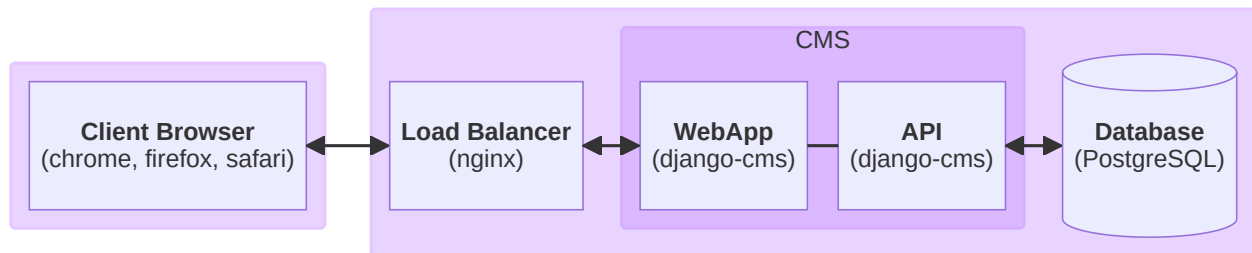
1. **Enhanced Collaboration:** The CMS facilitates interdisciplinary collaboration, breaking down communication barriers and encouraging diverse skill and perspective convergence.
2. **Efficient Project Management:** Streamlining the capstone project lifecycle, the CMS provides tools for tracking, milestone setting, and progress evaluation, reducing administrative overhead.
3. **Data-Driven Decision-Making:** Equipping stakeholders with reporting and analytics tools enables evidence-based decision-making, fostering continuous improvement at both individual and institutional levels.

2.1.5 Stakeholders

1. **Users:** Encompassing students, faculty, mentors, and administrators, the CMS assures reliability, functionality, and security for each stakeholder group.
2. **Investors:** The department relies on the CMS for strategic planning, resource allocation, and decision-making.
3. **Maintainers:** Developers find value in ongoing development and optimization to meet user needs and technological advancements.

2.2 Architecture

The basic architecture of the system upon which the CMS will be built is depicted in the diagram below.



The stakeholders will use a web browser to access the hosted CMS application. The basic components of the server and CMS application are given as the load balancer, web app, API, and database. These components form layers which handle different responsibilities within the CMS system. This is one of the simplest yet still secure architectures that this type of system can be implemented on. While it is possible to deploy an application such as this to a single server, it is equally feasible to split the components out in the cloud and scale, if necessary.

2.3 Features

The CMS boasts a robust set of features designed to streamline and enhance capstone project management. Each feature addresses specific needs, contributing to a comprehensive and user-friendly platform.

2.3.1 1. User Authentication and Authorization

- **User Registration:** Seamless registration for students, faculty, mentors, and administrators.
- **Login Functionality:** Secure login mechanisms for all users.
- **Role-Based Access Control:** Customized access levels for stakeholders.

2.3.2 2. Dashboard

- **Centralized Overview:** Provides a comprehensive overview of capstone project status, deadlines, and relevant information.

2.3.3 3. Project Proposal and Approval

- **Submission and Review:** Components for submitting and reviewing project proposals.
- **Faculty Feedback:** Features for faculty to provide feedback on proposals.
- **Approval Workflow:** Streamlined project approval process.

2.3.4 4. Project Repository

- **Secure Storage:** A repository for storing project documents, code, and related files.

2.3.5 5. Team Formation

- **Team Creation:** Tools for students to form project teams.
- **Teammate Search:** Capability to search for teammates and join existing teams.

2.3.6 6. Communication and Collaboration

- **Integrated Messaging:** Messaging system for team communication.
- **Collaboration Tools:** Features for collaborative work on project documents and code.

2.3.7 7. Milestone Tracking

- **Define Milestones:** Services to define project milestones.
- **Progress Tracking:** Track progress and set deadlines for project deliverables.

2.3.8 8. Feedback and Evaluation

- **Peer Evaluations:** Components for peer evaluations.
- **Faculty Assessments:** Tools for faculty to assess project progress.
- **Student-Mentor Relations:** Features to facilitate and track student-mentor relationships.

2.3.9 9. Calendar/Notifications and Reminders

- **Key Milestones Scheduling:** Calendar for scheduling milestones, sprints, and deadlines.
- **Automated Notifications:** Reminders for upcoming events and deadlines.

2.3.10 10. Reporting and Analytics

- **Generate Reports:** Services to generate reports on project outcomes and student performance.
- **Analytics:** Tools to analyze relevant metrics for continuous improvement.

2.4 User Characteristics

Understanding user profiles is crucial for tailoring the CMS to provide an optimal experience. Here's an overview of key user characteristics for each stakeholder group:

2.4.1 1. Students

- **Technological Proficiency:** Ranging from basic to advanced.
- **Collaborative Spirit:** Engages in team-based projects.
- **Time Sensitivity:** Manages project timelines and deadlines.

2.4.2 2. Faculty

- **Administrative Roles:** Utilizes the CMS for administrative tasks.
- **Mentorship:** Acts as mentors, requiring tools for tracking and guiding project progress.
- **Mixed Technological Familiarity:** Varies in technological proficiency.

2.4.3 3. Mentors

- **Industry Expertise:** Contributes industry-specific knowledge to projects.
- **Collaborative Engagement:** Interacts with students and faculty to ensure project alignment.
- **Limited Availability:** Requires efficient communication and project management tools.

2.4.4 4. Administrators

- **Strategic Oversight:** Leverages the CMS for planning, resource allocation, and decision-making.
- **Data-Driven Decision-Making:** Relies on CMS analytics for insights.
- **System Administration:** Manages user roles, access permissions, and system configurations.

2.5 Constraints

Effective deployment and utilization of the CMS are contingent upon an awareness of inherent constraints:

2.5.1 1. Technical Constraints

- **Compatibility:** Seamless compatibility across devices and browsers is imperative.
- **Internet Dependency:** Optimal functionality relies on consistent internet access.
- **Security Measures:** Robust security measures may impact system performance.

2.5.2 2. Resource Constraints

- **Server Capacity:** Performance is subject to server capacity.
- **Data Storage Limits:** Adherence to storage limits necessitates careful content management.

2.5.3 3. User Adoption Constraints

- **Training Needs:** Varying levels of technological proficiency require thoughtful training considerations.
- **Resistance to Change:** Potential resistance to adopting new tools can influence user adoption speed.

2.5.4 4. Regulatory and Compliance Constraints

- **Data Privacy Regulations:** Adhering to data protection laws impacts data handling and storage.
- **Accessibility Standards:** Compliance with standards may dictate design choices.

2.5.5 5. Budgetary Constraints

- **Development Costs:** Budget limitations require careful CMS development consideration.
- **Maintenance Costs:** Ongoing maintenance costs must align with the allocated budget.

2.5.6 6. Timeline Constraints

- **Development Timeline:** Influenced by academic calendars and project deadlines.
- **Implementation Period:** Rapid implementation is needed to align with academic schedules.

2.5.7 7. Scalability Constraints

- **User Growth:** Designing for scalability is imperative to accommodate potential user growth.
- **Feature Expansion:** Future scalability considerations may impact new feature integration.

Meticulous consideration and proactive management of these constraints are integral to ensuring effective CMS deployment and enduring viability.

2.6 Operating Interfaces

The CMS interacts with various components to ensure seamless functionality across hardware, software, and network environments:

2.6.1 1. Hardware Interfaces

- **Computing Devices:** Desktop computers, laptops, tablets.
- **Peripheral Devices:** Printers, scanners.

2.6.2 2. Software Interfaces

- **Operating Systems:** Windows, macOS, Linux.
- **Web Browsers:** Google Chrome, Mozilla Firefox, Microsoft Edge.
- **Database Management Systems (DBMS):** MySQL, PostgreSQL, MongoDB.
- **Collaboration Tools:** Slack, Microsoft Teams, Google Workspace.

2.6.3 3. Network Interfaces

- **Internet Connectivity:** Reliable internet access.
- **Security Protocols:** HTTPS.
- **Firewall Compatibility:** Compatibility for uninterrupted access.

Optimizing these interfaces ensures the efficient operation and accessibility of the CMS.

REQUIREMENTS

3.1 Functional Requirements

Functional requirements for the Capstone Management System define the specific behaviors and capabilities that the system must possess to support the needs of students, faculty, and company representatives engaged in capstone projects. These requirements encompass user registration and authentication, with secure role-based access to ensure that users can perform actions pertinent to their roles—such as submitting project proposals, collaborating on documents, tracking project milestones, and evaluating progress. We also include a centralized dashboard for an overview of project statuses, a project repository for document and code storage, team formation tools, integrated communication facilities, a robust calendar with notifications, and comprehensive reporting and analytics features to monitor and assess project outcomes and student performance. Collectively, these functional requirements facilitate a seamless, efficient, and secure management process for all participants in the capstone project lifecycle.

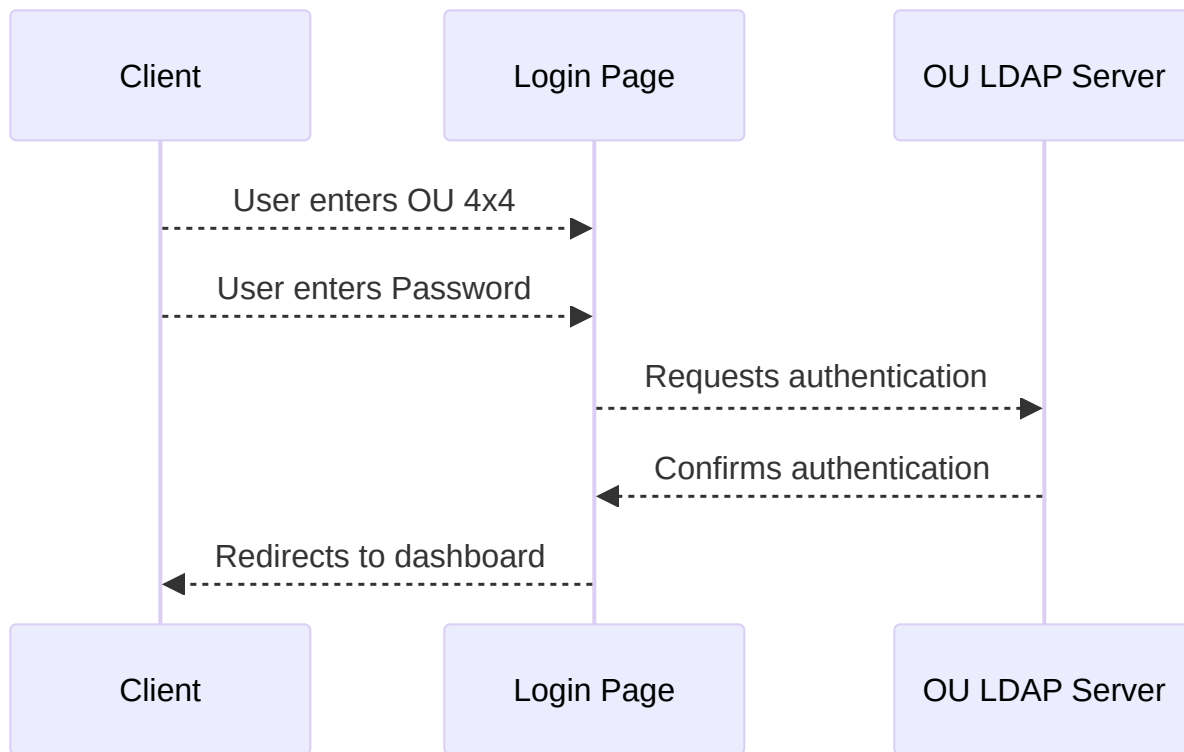
3.2 User Registration

3.2.1 Details Functionality

Account Creation Interface

- Our system will feature a user-centric interface for account creation, prioritizing clarity and ease of use.
- A form with mandatory fields will be presented for data entry, ensuring all necessary information is captured.
- We plan to apply strict form validation to maintain the integrity of the data. This includes checking the format of emails and requiring strong passwords.
- For students and faculty, the active directory servers located on-campus may be used to simplify the login process (see the diagram below).

Student/Faculty OU 4x4 Login Workflow



Essential Information Collection

- **Full Name:** The registration form will accommodate entries for first, middle, and last names.
- **Contact Details:** We will collect email addresses and phone numbers, supporting the international format for the latter.
- **Affiliation:** Users will identify their role (Student, Instructor, or Company Representative) via dropdown menus or radio buttons.
- **Password Creation:** Password fields will incorporate real-time strength evaluation and display criteria to encourage secure account creation.

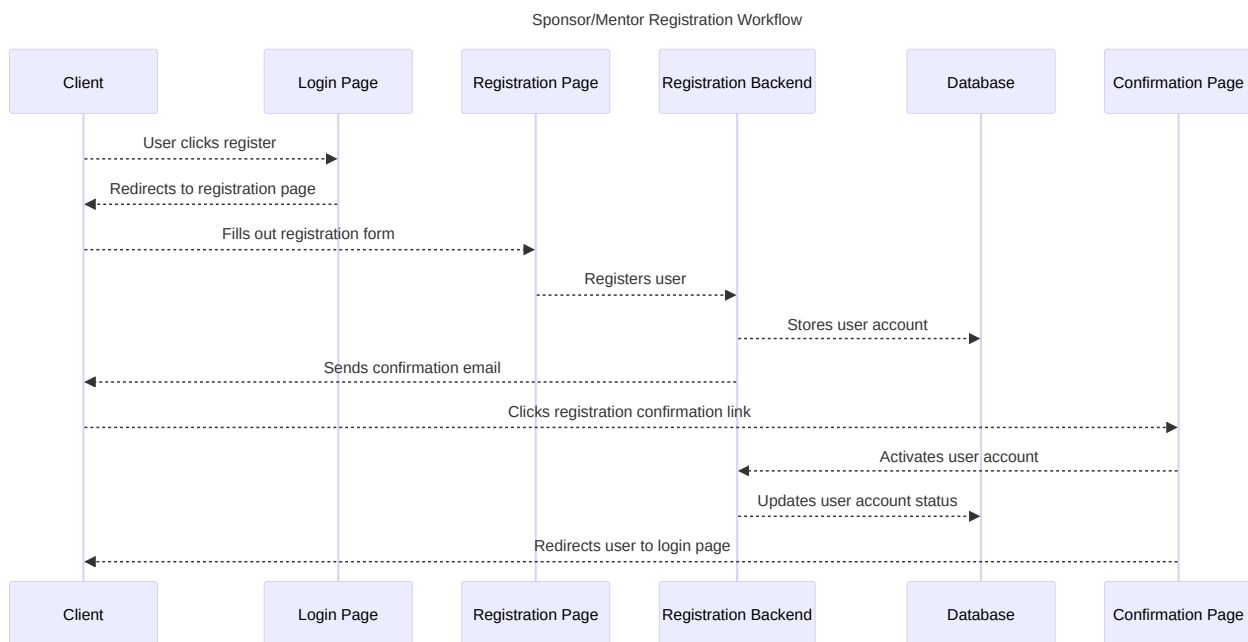
Identity Verification

- Submission of the registration form will initiate a verification process through an emailed link or SMS code.
- Users must complete the verification within a set timeframe, adding a layer of security to the registration process.

Role Selection and Assignment

- During registration, users will select their intended role within the system.
- The system will notify administrators and faculty, allowing them to approve the assignment of role-based permissions, dictating access levels post-login.
- Roles will be integrated into user profiles for subsequent Role-Based Access Control.

An example of sponsor and mentor registration workflow is given below:



3.2.2 Role-Based Access Control (RBAC)

Role Definitions

In the system, every user role will have specific permissions that clearly outline which features and actions they can access. We will store these permissions in the database to ensure they are consistently applied across the system.

Permissions for Instructors

Instructor will have the ability to oversee project proposals, participate in giving feedback, and supervise assessments. The system will enable monitoring of student advancement and allow for direct interactions with both students and colleagues.

Permissions for Company Representatives

Company representatives will have access to customized interfaces designed according to their interaction needs, emphasizing project management and feedback provision. Access for these representatives will be strictly regulated to protect confidential information. Access Control Enforcement:

Permissions for Students

- **Access to Projects and Management Capabilities:** Students are granted the ability to access, manage, and interact with their project proposals, documents, and relevant materials within the project repository. This includes submitting proposals, updating documentation, and reporting progress.
- **Collaboration within Teams:** Students are allowed to form teams, extend invitations to peers for project participation, and work collaboratively on shared documents and resources. The platform supports communication among team members through integrated messaging systems or discussion forums.
- **Milestone Submission and Progress Tracking:** Students are permitted to submit work pertaining to project milestones, monitor their advancement towards these milestones, and view upcoming deadlines to effectively organize their time.
- **Receiving Feedback:** Students can receive constructive feedback from instructors and company representatives, with the facility to view evaluations and comments on their work, fostering continuous improvement.
- **Management of Personal Profiles:** Students have the facility to oversee their personal profiles, which includes the ability to update contact details and modify passwords.
- **Access to Calendar and Notifications:** Students can access a calendar to track project timelines, deadlines, and scheduled meetings. They will also receive notifications and reminders concerning significant deadlines and events related to their capstone projects.
- **Access to Resources:** Students are provided access to educational resources, guidelines, and templates to support their capstone project development.
- **Generation of Limited Reports:** Students can produce reports on their project status and their performance metrics, where applicable.
- **Restricted Access:** Students do not have access to administrative functions such as the approval of project proposals, grade assignments, or the viewing of other groups' confidential project details. Access to sensitive information regarding faculty assessments and internal company notes is also restricted.

3.3 Detail Data Required/User Inputs

3.3.1 For User Registration

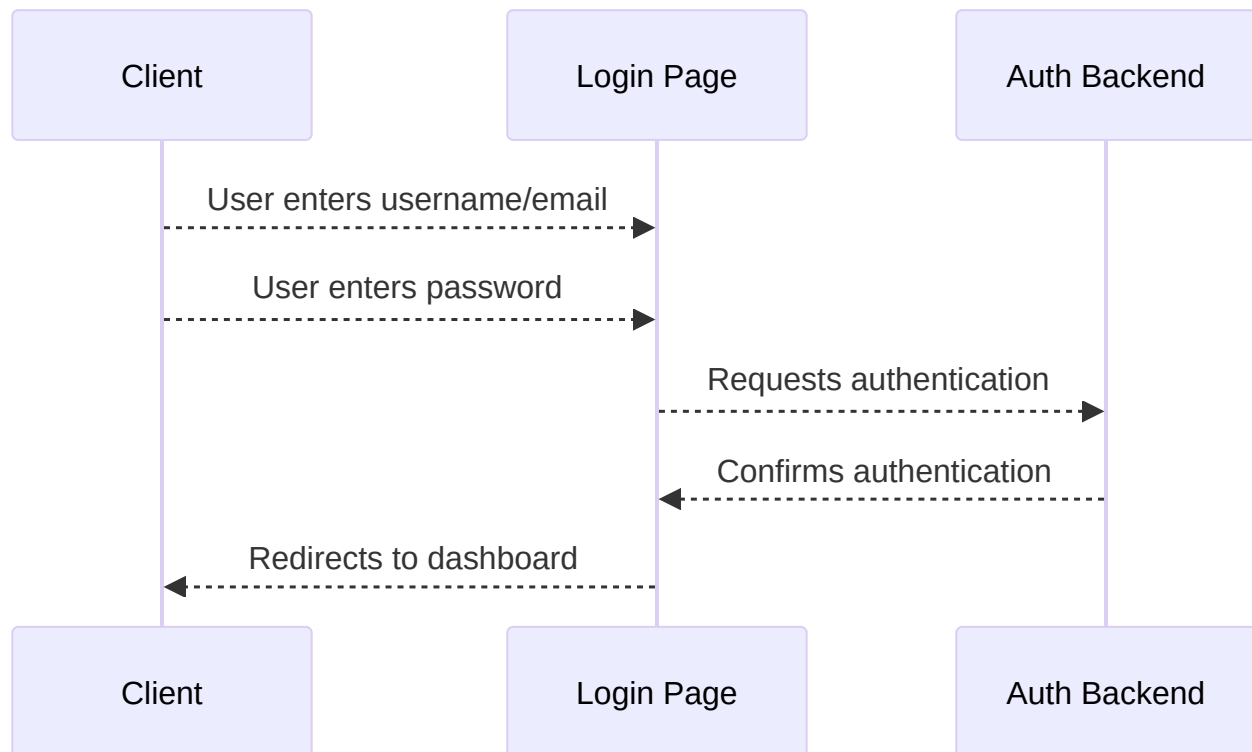
- **Full Name:** The registration process should prompt users to enter their full legal name, ensuring accurate identification in the system. This should include separate fields for the first name, middle name(s), and last name to accommodate various naming conventions.
- **Email Address:** Users must provide a valid email address that will serve as a primary means of communication and as part of their login credentials. The system should verify that the email address is in a valid format and is not already registered.
- **Affiliation:** Users are required to select their affiliation with the institution from predefined options such as 'Student', 'Instructor', or 'Company Representative'. This selection will dictate the level of access and permissions granted to the user within the system.
- **Password:** A password must be created by the user, adhering to defined strength criteria, such as a minimum number of characters, and the inclusion of uppercase letters, lowercase letters, numbers, and symbols. The system should provide real-time feedback on the password's strength and compliance with the criteria.
- **Contact Number:** As an optional field for additional verification, users may enter a contact number. This number should support international formats and may be used for account recovery or multi-factor authentication purposes.

3.3.2 For Login

- **Username/Email Address:** To access their account, users will be required to enter their unique username or the email address associated with their account.
- **Password:** The corresponding password for the account must be entered. This password should be transmitted securely and validated against the stored credentials.
- **Multi-factor Authentication Code:** If MFA is enabled, after entering the username and password, the user will be prompted to provide a code that has been sent to their pre-configured device or email as an additional layer of security.

Sponsor and mentor login will behave similar to 4x4 login, but instead of using LDAP, accounts shall be stored in the database and the authentication backend of the CMS will approve authentication requests. See the diagram below.

Sponsor/Mentor Login Workflow



3.3.3 For Role-Based Access Control (RBAC)

- **User Role Data:** Upon registration or by administrative assignment, users will have their role within the system recorded—whether they are an Instructor or a Company Representative. This role data is crucial for determining the features and data the user can access.
- **Permission Levels:** Each role will have associated permission levels which are sets of access rights within the system. These define what actions users can take and what data they can view or modify. Permissions should be granular to allow precise control over user actions and access within the system.

3.3.4 Pathways from Specific Requirement

- **From Registration to System Access:** When you sign up, we make sure your details are correct by sending you an email to check. You need to click on a link in this email to confirm your account. After you confirm through the email, you'll need to finish setting up your account by adding any more information we need. We'll give you a starting role in the system based on if you're a student, teacher, or company person. This can be changed later if needed. Then, we'll send you over to the login page. Here, you can get into the system with the username and password you just made.
- **From Login to Role-Specific Dashboard:** When you log in, we check your username and password to make sure it's really you. If you have extra security set up (we call this Multi-factor Authentication or MFA), we'll ask for a special code that's sent to your phone or email. Once we know it's you, we'll take you to your main page in the system. This page will look different depending on if you're a student, teacher, or from a company. It'll have everything you need for your projects.
- **Role-Based Access Implementation:** Right after you log in, the system looks at what role you have. This decides what you can see and do in the system. As you move around in the system, what you can access depends

on your role. Some things are only for teachers, some only for students, and some only for company people.

3.4 For Project Proposal and Approval

3.4.1 Detailed Functionality

Proposal Submission Interface

- A dedicated section where students can fill out and submit their project proposals.
- It should include form fields for project title, abstract, detailed description, objectives, methodology, expected outcomes, and any special requirements or resources needed.
- The system should allow for the attachment of relevant documents or images that support the proposal.

Faculty Review Interface

- A portal for faculty members where they can view submitted proposals.
- Faculty should be able to filter proposals by various criteria such as submission date, student name, or project category.
- An area for faculty to write comments or request additional information from the student.

3.4.2 Approval Workflow

- Once a proposal is submitted, an automated notification should be sent to the designated faculty for review.
- Faculty can approve, reject, or request revisions to the proposal.
- A tracking feature for students to follow the status of their proposal through the approval process.

Feedback Loop

- A system for faculty to provide structured feedback, including comments, suggestions, and necessary improvements.
- Students should receive notifications when feedback is provided and be able to view and respond to the comments.

Revision Submissions

- If revisions are requested, students should be able to submit an updated proposal through the system.
- The revised proposal should maintain a link to the original submission for reference.

Final Approval and Project Initiation

- Once a proposal is approved, a project workspace should be created automatically in the system.
- Students and faculty should have access to this workspace based on their roles.

3.4.3 Detailed Data Required / User Inputs

For Proposal Submission

- **Project Title:** A concise title for the project.
- **Abstract:** A brief summary of the project's purpose and goals.
- **Description:** A detailed explanation of the project including background information and the problem statement.
- **Objectives:** Specific objectives the project aims to achieve.
- **Methodology:** The approach or methods the students plan to use in their project.
- **Expected Outcomes:** The anticipated results or products of the project.
- **Resources:** Any special resources or support required for the project.

For Faculty Review

- **Filters:** Options for sorting and searching through proposals.
- **Comments Section:** A text area for faculty to write their observations and suggestions.

For Approval Workflow

- **Notifications:** Automatic alerts for new submissions and status updates.

For Feedback Loop

- **Feedback Form:** A structured format for providing feedback with specific sections for different types of comments.

For Revision Submissions

- **Revision Tracking:** A system to track changes and link revisions to the original proposal.

Pathway from Specific Requirement

- When a student submits a proposal, the system checks to make sure all required fields are filled and then sends it to the faculty for review.
- Faculty members receive a notification and review the proposal at their interface, where they can approve it, ask for changes, or reject it.
- The student gets a notification about the faculty decision. If changes are needed, they can submit a revised proposal.

- Once the faculty approves the revised proposal, the system will create a new project workspace and notify the student that their project can start.

3.5 For Project Repository

3.5.1 Detailed Functionality

Secure File Storage

- The system will provide a secure area online where all project-related files can be stored. This will include documents, images, spreadsheets, presentation files, and source code.
- It will use encryption to protect files from unauthorized access both during transfer (as they are uploaded or downloaded) and while at rest (stored on the server).

File Management System

- Users will have the ability to create folders, rename them, and organize files as needed within the repository.
- The system should support version control for documents and code, allowing users to track changes over time and revert to previous versions if necessary.

Upload and Download Capabilities

- Students and faculty will be able to upload files to the repository, with the system checking for any viruses or malware as part of the upload process.
- Users should be able to download files from the repository, with the system logging each download for security and auditing purposes.

Access Control

- The repository will have a permissions system that controls who can view, upload, or download each file or folder based on their role in the project.
- Permissions can be set for individuals or groups, such as a project team or class section.

Collaboration Tools

- The system will include features for commenting on documents, tagging other users, and suggesting edits.
- There should be a mechanism for managing and resolving conflicts when multiple users edit the same document simultaneously.

Search Functionality

- Users will be able to search the repository for files based on name, content, date uploaded, and other metadata.

Integration with Development Tools

- The repository will be able to integrate with software development environments (IDEs) and other tools commonly used in computer science projects.

Backup and Recovery

- The system will perform regular backups of all files in the repository.
- There should be a process for recovering files in case of accidental deletion or loss.

3.5.2 Detailed Data Required / User Inputs

For File Storage

- **Files:** The actual files to be stored, which can be of various types and sizes.
- **Metadata:** Information about the files, such as the author, date created, and a description of the contents.

For File Management

- **Folder Names:** Custom names for folders created by users.
- **File Versions:** Records of each file version when updates are made.

For Upload and Download

- **Upload Data:** Files selected by users for upload.
- **Download Requests:** User requests for file downloads, specifying the file or files they wish to retrieve.

3.5.3 Pathway from Specific Requirement

- When a user wants to store a file, they will log into the system, navigate to the project repository, and upload the file. The system will check the file for security and then store it, encrypted, in the correct location.
- To organize files, users can create new folders and move files into them. Each change will be recorded in the system's version history.
- If a user needs to find a file, they can use the search tool by entering keywords, file names, or other details that describe the file.
- When working on files, users can collaborate with teammates by leaving comments or making edits, with the system managing changes to prevent conflicts.
- Regular backups will be made of all files so that nothing gets lost. If a user deletes something by mistake, they can ask for it to be restored from the backup.

3.6 For Team Formation

3.6.1 Detailed Functionality

Team Creation Interface

- A clear and simple-to-use interface where students can create a new team for their capstone project.
- The interface should allow the team creator to set a team name, description, and specify the number of members required.
- It should enable the team leader (creator) to define roles or skills needed for the project, such as ‘programmer’, ‘designer’, or ‘researcher’.

Teammate Search and Invitation

- A search functionality where students can look for potential teammates by various criteria such as skills, past project experience, academic focus, or availability.
- The ability to send invitations to join the team directly through the system.
- An option for students to post a “team member request” on a public board where other students can apply to join.

Team Joining Mechanism

A feature for students to browse existing teams looking for members.

- A request to join option that allows students to apply to be part of a team and an approval process for the team leader to accept or reject applications.
- The system should manage membership requests, showing pending approvals and current team status to the team leader.

Team Management Tools

- Tools for the team leader to manage the team such as adding or removing members, assigning roles within the team, and setting permissions for document access.
- A function to dissolve the team or leave the team for students who wish to exit their current group.

Communication Tools

- Integrated messaging within the team formation interface to facilitate communication between team members and the team leader.
- Discussion boards or forums for team interaction, planning, and collaboration.

3.6.2 Detailed Data Required / User Inputs

For Team Creation

- **Team Name:** The chosen name for the capstone project team.
- **Team Description:** A brief overview of the project and the team's goals.
- **Member Roles:** Specific roles or skills needed to complete the team, which the team creator can list.

For Teammate Search and Invitation

- **Search Criteria:** Keywords or filters that students use to find potential teammates, such as 'software development' or 'graphic design'.
- **Invitation Message:** A personalized message that accompanies the invitation to join the team.

For Team Joining Mechanism

- **Join Requests:** Inputs from students who wish to join a team, including a brief introduction or cover letter.
- **Approval Status:** Notifications for students who have applied to join a team, indicating whether they have been accepted or not.

For Team Management Tools

- **Member Management:** Options for the team leader to add new members, assign roles, or remove members from the team.

For Communication Tools

- **Messages:** Text input for team communication, including initiating conversations or discussions.

3.6.3 Pathway from Specific Requirement

- A student starts by creating a team, entering all the required information about the project and what kind of team members they are looking for.
- Other students can search for teams that match their skills and interests and send a request to join, or respond to a public 'team member request'.
- The team leader receives these requests and can review the profiles of the applicants before deciding to accept or reject them.
- Once the team is formed, the leader has tools to manage the team members and their roles, and all team members can communicate with each other through built-in communication tools.
- If needed, the team leader can make changes to the team, like adding more members or removing someone if they leave the project.

3.7 For Communication and Collaboration Platform

The communication and collaboration platform component in our product includes integrated messaging or discussion forums for team communication and tools for collaboration on project documents and code. We have some functionality and data/user input involves detailing the specific capabilities and interactions users can expect from the system. Here's a more detailed breakdown:

3.7.1 Functionality

- **Messaging System:** Real-time text messaging allows for instant communication between team members and faculty, including group chats and private messages. Supports text formatting and embedding images or links.
- **File Sharing:** Secure uploading and downloading of files with support for various file types (documents, images, presentations). Includes version tracking and permission settings to control access.
- **Discussion Forums:** Spaces for broader discussions, allowing for threaded conversations organized by topic to facilitate class-wide or team-specific discussions.
- **Task Management:** Enables teams to create, assign, and track tasks and deadlines within the project scope, enhancing collaboration and accountability.

3.7.2 Data/User Input

- **Text Messages:** Includes the actual text input by users, formatting choices (bold, italics, etc.), and any embedded media or links.
- **Files for Sharing:** Information on the files being shared, such as file name, type, size, version number, and metadata (e.g., author, upload date).
- **Discussion Posts:** Text content of posts, associated topics or tags, attachments, and user engagement metrics (likes, replies).
- **Task Details:** Task descriptions, assigned members, due dates, status updates, and any associated files or links.
- **Meeting Schedules:** Dates, times and participants, along with any necessary access links or codes.

This details provides a more comprehensive view of the communication and collaboration platform, specifying how users interact with the system and the types of data it needs to process. This detailed description will help ensure the development team fully understands the requirements and expectations for this component of the capstone management system.

3.8 For Milestone Tracking

Here is a description of functionality and data/user input for the milestone tracking component in our product:

3.8.1 Functionality

- **Milestone Creation:** Allows users to define and set key project milestones with specific objectives, including start and end dates.
- **Progress Monitoring:** Enables real-time tracking of milestone completion status, with visual indicators for on-track, at risk, and delayed milestones.
- **Deadline Alerts:** Automated notifications and reminders for upcoming or missed deadlines to ensure timely completion of tasks.
- **Update and Revision:** Facilitates the updating of milestone details and deadlines, allowing for project flexibility and adjustments.
- **Integration with Calendar:** Milestones automatically populate in the system's calendar, providing a visual timeline of the project's key dates and events.

3.8.2 Data/User Input

- **Milestone Descriptions:** Textual information outlining the purpose and objectives of each milestone.
- **Deadlines:** Specific dates for the completion of milestones, including any changes or extensions.
- **Completion Status:** User-updated information on the progress toward milestone achievement, which can be marked as not started, in progress, completed, or delayed.

This detailed breakdown will guide the development of a robust milestone tracking system, ensuring users have the tools necessary to effectively plan, monitor, and adjust their project milestones.

3.9 For Feedback and Evaluation

Here is a description of functionality and data/user input for the feedback and evaluation component in our product:

3.9.1 Functionality

- **Continuous Feedback:** A platform for faculty to provide ongoing, incremental feedback on project progress, with the ability to attach comments to specific project parts or deliverables.
- **Final Evaluation:** Allows for comprehensive final assessments of projects, including qualitative feedback and quantitative scores based on predefined criteria.
- **Peer Review:** Facilitates structured peer review processes among students, enabling them to provide and receive feedback on project work.
- **Rubric-Based Evaluation:** Incorporates customizable evaluation rubrics, allowing faculty and peers to rate projects against specific criteria and standards.
- **Feedback Compilation and Access:** Automatically compiles feedback for easy access by students, enabling them to review and respond to comments and evaluations.

3.9.2 Data/User Input

- **Feedback Comments:** Textual feedback provided by faculty and peers, including suggestions, commendations, and areas for improvement.
- **Evaluation Criteria:** Predefined metrics or criteria used for assessing projects, which can include innovation, technical quality, adherence to objectives, presentation, and collaboration.
- **Scores:** Numerical ratings assigned based on the evaluation criteria, which may be aggregated to form an overall project score.

This detailed approach ensures that the feedback and evaluation system supports a comprehensive and constructive assessment process, enhancing learning and project outcomes.

3.10 For Calendar and Notifications

Here is a description of functionality and data/user input for the calendar and notifications component in our product:

3.10.1 Functionality

- **Event Scheduling:** Enable users to schedule project milestones, deadlines, meetings, and other key events within a personalized calendar.
- **Automatic Reminders:** System-generated reminders for upcoming events, customizable to user preferences for lead time and notification method (e.g., email, SMS, in-app notifications).
- **Calendar Integration:** Ability to integrate with external calendar services (such as Google Calendar or Microsoft Outlook) for seamless synchronization of events.
- **Event Modification and Cancellation:** Facilities for users to modify or cancel scheduled events, with automatic notification to affected parties.
- **Sharing and Permissions:** Options for users to share their calendar with team members or faculty, including configurable view/edit permissions.

3.10.2 Data/User Input

- **Event Details:** Information about each event, including title, description, date, time, and location (virtual or physical).
- **Reminder Preferences:** User-selected preferences for how and when to receive notifications about upcoming events.
- **Calendar Sharing Settings:** User choices regarding who can view or edit their calendar, including specific permissions for different users or groups.

This provides the development team with clear guidelines on the requirements for the calendar and notifications component, ensuring it supports effective time management and communication within the capstone management system.

3.11 For Reporting and Analytics

Here is a description of functionality and data/user input for the reporting and analytics component in our product:

3.11.1 Functionality

- **Custom Report Generation:** Enables users to create customized reports on project progress, including specific data points such as milestone completion, team participation, and overall project status.
- **Analytics Dashboard:** Provides an interactive dashboard that displays analytics on system usage, including user engagement, project submission rates, and feedback effectiveness.
- **Performance Metrics:** Tracks and reports on key performance indicators (KPIs) for projects, such as adherence to deadlines, quality of submissions, and peer evaluation scores.
- **Data Filtering and Segmentation:** Offers advanced filtering options to analyze data by time period, project type, department, or any other relevant criteria.
- **Export and Sharing Options:** Allows users to export reports and analytics in various formats (e.g., PDF, Excel) and share them with stakeholders through email or integrated platforms.

3.11.2 Data/User Input

- **Report Parameters:** Specific criteria selected by users for customizing reports, such as date range, project category, or specific KPIs.
- **Analytics Criteria:** User-defined metrics and dimensions for analyzing system performance and user engagement.
- **Export Options:** Preferences for report format, resolution, and delivery method, ensuring reports are accessible and useful to all stakeholders.

This detailed description will guide the development of a comprehensive reporting and analytics system, ensuring stakeholders have access to meaningful insights into project and system performance.

3.12 External Interface Requirements

This section aims to discuss the performance needs, applications utilized, and security protocols utilized to exchange data across the different endpoints throughout the application. By utilizing the proper programs and protocol, both high performance metrics and mitigate security risks. An important note is that these systems are custom to change as the tech stack, design features and requirements, and scale requirements are finalized by the customer and development team.

3.12.1 User Interface

The User Interface (UI) is an essential component of the application, since this is where the users will directly access and interact with the application. The CMS that will be developed will be a GUI-driven interface, since the landing page once the user logs in will be the user-dashboard. While this application will be tailored for use on a computer, the frontend frameworks like React.js and Angular both have mobile app/web development features enables for use by the development team. The team may also decide to directly use HTML5 pages for the application, which are web pages that are specifically designed for use on both computer and mobile devices, which would allow for more flexibility for the users.

3.12.2 External System Interfaces

A variety of system interfaces may be used due to the complexity and variety of the features that will be implemented for the application. For the general web pages, a RESTful API would be a strong choice for organization because the flow of the application through the web pages would be easy to follow for both the development team and the users. Additionally, this structure helps with the flow of data throughout the website, with the dashboard having much of the data displayed, and web pages for the smaller scale features only exhibiting only the data that is central to the functionality of the feature.

3.12.3 Data IO Interfaces

The safe and secure exchange of data throughout the application's infrastructure is pivotal to the application's success. Data can be sent through the different channels of the program through a number of means, whether that be through reading CSVs, pulling JSON files, or exchanging XML data. While these tools are used in a variety of circumstances, Django and its derivatives have QuerySets, which are functions that query the data straight from the database, and Django plugs into React.js and Angular. Because of this, utilizing files to transfer data would be inefficient, require additional hardware, and expose the program data to additional vulnerabilities. When using these plug-ins from the application's endpoints to the backend, security protocols for the data transfers will be necessary, which will be discussed in the security interfaces section.

3.12.4 Hardware and Software Interfaces

The development of this program will not require the team to manage hardware services, eliminating the need for maintainers to handle hardware malfunctions. Since this project will most likely be hosted on a server, the maintenance team will deal with connectivity and other software interface issues. This will deal with keeping the system up-to-date, clearing any potential dependencies within the different frameworks that will be used to develop and deploy the application, and maintaining the database. The frameworks that have been selected for the application at this time are compatible with each other and have a history of being so, however this is something that could change in the future.

3.12.5 Communication and Security Interfaces

Maintaining security on this project is vital, and the communication the application performs between the various platforms that are utilized presents the highest risk associated with the project. The program will be password protected, which reduces the risk of non-users accessing data in the program. Additionally, being password protected will help make sure that users do not get access to information that their access level should not be able to see. In Django and its derivatives, Secure Sockets Layer (SSL) encryption and Hypertext Transfer Protocol Secure (HTTPS) communication protocol is available to be utilized. For authentication, Django also has a built-in username-password authentication system, which eliminates the need for the development team to develop such service. This mitigates the potential exposure of unauthorized users accessing information that otherwise they would not get access to.

3.12.6 Performance Interfaces

Monitoring the application throughout its lifecycle is important for scalability, performance and maintenance adjustments, and implementing features in the future. While optimized coding practices and designing optimal processes from the onset, adapting to scaling requests, and performance metrics on the website post launch will help the maintenance team optimize the application to the user preferences throughout the service life of the application. There are a number of performance interfaces that allow for tracking the performance of the application that is compatible with Django. Here is a list and a description of what each program provides:

- 1. Google Analytics:** This tracks the interactions of the website from the frontend/UI. This is often free to access for your specified application. This tool may not help optimize the application, however it gives free insight on the traffic

experienced by the application **2. Solarwinds Appoptics:** This application provides sophisticated visualizations and tracks performance across the platform. **3. Atatus:** Provides real-time metrics on the application. Utilizing realtime data will allow the maintenance team to address concerns in a quicker and more targeted manner. This will improve the customer's experience with the application

There are many applications and performance tracking programs that can be used to track this application. The best program to use will depend on the customer needs, scale of the application, and risks identified throughout the development cycle of the application.

3.13 Non Functional Requirements

Nonfunctional requirements (NFRs) are essential for guiding the design and development of a software system. They specify system qualities and attributes that are not directly related to functionality but significantly impact the system's success. For the Capstone Management System, the non-functional requirements (NFRs) play a crucial role in ensuring the system's usability, reliability, performance, and security. Here are the NFRs categorized by concern:

3.13.1 Performance Requirements

- **Response Time:** The system should respond to user interactions within 2 seconds under normal load conditions. Page load times for the dashboard and other critical features should be less than 2 seconds.
- **Concurrent Users:** The system must support up to 500 concurrent users during peak usage without significant degradation of performance.
- **Data Processing:** Project proposal approvals, feedback submissions, and report generation should not exceed 5 seconds under normal conditions.
- **Load Testing:** System should be tested under simulated peak loads to ensure performance meets requirements.
- **Monitoring and Alerting:** System performance should be monitored continuously with alerts for potential issues. Performance metrics should be logged for analysis and optimization.
- **Geographic distribution of users:** If users are spread across continents, factor in latency and network performance variations.
- **Peak usage periods:** Identify predictable times of high activity (e.g., submission deadlines) and ensure optimal performance during those times.
- **Performance benchmarks:** Research similar systems and compare their performance metrics to establish reasonable targets.

3.13.2 Security Requirements

- **Authentication:** The system must implement secure authentication mechanisms for all users, including two-factor authentication for administrators. Protection against unauthorized access and data breaches is critical.
- **Authorization:** Role-based access control (RBAC) should be strictly enforced, ensuring users can only access information and functionalities pertinent to their roles.
- **Data Encryption:** All sensitive data, including user information and project documents, must be encrypted in transit and at rest.
- **Audit Trails:** The system should maintain detailed audit logs of all user actions, accessible only to authorized administrators.
- **Vulnerability Management:** System should be regularly scanned for vulnerabilities and patched promptly. Secure coding practices should be followed to minimize vulnerabilities.

- **Penetration Testing:** System should be subjected to regular penetration testing to identify and address security weaknesses.
- **Data Backup and Recovery:** Regular backups of system data should be performed and stored securely offsite. Disaster recovery plan should be in place to ensure quick recovery in case of system failure.
- **Access Logging and Auditing:** Logs should be retained for a defined period and be tamper-proof.
- **Privacy Compliance:** System should comply with all relevant data privacy regulations (e.g., GDPR, CCPA). User data should be collected and used responsibly with clear consent mechanisms.

3.13.3 Software Quality Attributes

1. Availability & Reliability Requirements:

- **System Availability:** The target availability should be at least 99.5% outside of scheduled maintenance windows.
- **Data Backup:** Regular backups of all system data, with a recovery point objective (RPO) of 24 hours and a recovery time objective (RTO) of 4 hours.
- **Fault Tolerance:** The system should be designed to gracefully handle failures, with mechanisms to automatically recover from common errors without user intervention.

2. Scalability Requirements:

- **Horizontal Scaling:** The system architecture must support horizontal scaling to accommodate growth in users and data volume.
- **Load Balancing:** Implement load balancing to distribute traffic evenly across servers, ensuring no single point of failure.

3. Usability Requirements:

- **User Interface:** The interface should be intuitive and accessible, following best practices for UI/UX design. It should be accessible to users with disabilities, adhering to WCAG 2.1 guidelines. Consistent design elements and clear labeling are essential.
- **Documentation:** Comprehensive user and technical documentation should be provided, easily accessible through the system.

4. Compatibility Requirements:

- **Browser Compatibility:** The web interface must be compatible with the latest versions of major browsers (Chrome, Firefox, Safari, and Edge).
- **Mobile Responsiveness:** The system should offer a responsive design, ensuring usability on smartphones and tablets across different operating systems.

5. Maintainability and Extensibility Requirements:

- **Code Maintainability:** The system should be developed with clean, modular, and well-documented code to facilitate easy maintenance and future enhancements. Updates and modifications should not disrupt existing functionality.
- **API Extensibility:** Where applicable, the system should provide well-documented APIs to allow for integration with external systems and future extensions.

6. Legal and Regulatory Requirements:

- **Data Privacy:** The system must comply with relevant data protection regulations (e.g., GDPR, CCPA) in handling personal information of users.
- **Intellectual Property:** It should respect and enforce intellectual property rights, ensuring that users can only share and access content for which they have rights.

7. Environmental Requirements:

- **Energy Efficiency:** The system should be designed for energy efficiency, minimizing the carbon footprint of its operation.

3.13.4 Casptone Management System Requirements**1. User Authentication and Authorization:**

- Login process should be secure and use strong password encryption.
- Role-based access control should be granular and ensure appropriate permissions for each user role.
- System should support two-factor authentication for additional security.

2. Dashboard:

- Dashboard should be customizable and allow users to personalize their view.
- Information displayed on the dashboard should be accurate and up-to-date.
- Users should be able to filter and search for specific information.

3. Project Proposal and Approval:

- Proposal submission process should be streamlined and efficient.
- Review process should be transparent and provide clear feedback to students.
- System should track the status of each proposal and notify relevant stakeholders.

4. Project Repository:

- Repository should support multiple file types and versions.
- Access to files should be controlled based on user permissions.
- Repository should allow for easy collaboration and file sharing within teams.

5. Team Formation:

- System should make it easy for students to find potential teammates based on skills and interests.
- Teams should be able to set their own rules and guidelines.
- System should facilitate communication and collaboration among team members.

6. Milestone Tracking:

- Milestones should be easily defined and tracked within the system.
- System should send automated notifications and reminders for upcoming deadlines.
- Progress towards milestones should be visible to all stakeholders.

7. Feedback and Evaluation:

- System should provide a structured way for faculty to provide feedback to students.
- System should facilitate peer evaluation and feedback sharing among team members.
- Mentorship relation should be traceable and transparent for both parties.

8. Notifications and Reminders:

- Automated notifications should remind users of upcoming deadlines, meetings, and milestones.
- Reminders can be sent via email, SMS, or in-app notifications.

9. Collaboration Tools:

- Integrated messaging or discussion forums should facilitate team communication.
- Version control for project documents and code is essential.

10. Reporting and Analytics:

- Generate reports on project outcomes, student performance, and faculty assessments.
- Analytics can provide insights into usage patterns and system effectiveness.

11. Calendar Integration:

- Integrate with existing calendar systems (e.g., Google Calendar, Outlook) for scheduling.
- Display project milestones, sprints, and deadlines on users' calendars.

By addressing these non-functional requirements, the Capstone Management System will be well-equipped to provide a secure, reliable, and user-friendly experience for its stakeholders, while also being prepared for future growth and changes in technology and regulations.

REFERENCES

1. <https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>
2. <https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document>
3. <https://apps.dtic.mil/sti/tr/pdf/ADA301537.pdf>
4. [https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#:~:text=A software requirements specification \(SRS\) is a document that describes, stakeholders \(business%2C users\).](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#:~:text=A software requirements specification (SRS) is a document that describes, stakeholders (business%2C users).)
5. <https://www.geeksforgeeks.org/software-requirement-specification-srs-format/>
6. <https://www.javatpoint.com/software-requirement-specifications/>
7. <https://www.verasseti.com/insights/how-to-write-a-system-specification#:~:text=The SRS is a technical, its hardware and software requirements./>
8. [https://www.techtarget.com/searchsoftwarequality/definition/software-requirements-specification#:~:text=A software requirements specification \(SRS, will be expected to perform.](https://www.techtarget.com/searchsoftwarequality/definition/software-requirements-specification#:~:text=A software requirements specification (SRS, will be expected to perform.)
9. <https://forum.djangoproject.com/t/https-during-development/10509>
10. <https://forum.djangoproject.com/t/setting-up-ssl-in-settings-py/7147>
11. <https://docs.djangoproject.com/en/3.1/topics/security/#ssl-https>
12. <https://www.tek-tools.com/cloud/best-tools-to-monitor-django-performance>
13. Virtual Exam. 2023, December 4. CS619 Final Project | SRS | Software Requirement Specification | Functional Requirements |. Youtube. <https://www.youtube.com/watch?v=rAvlSHhodBw/>