



Course Assessment Management System

PROJECT DOCUMENTATION

May 2024

ACS 56000 - SOFTWARE ENGINEERING

2024

Prepared By
Group 1

TABLE OF CONTENTS

1. Introduction
2. Requirement Glossary
3. Use Case Models
4. Design & Implementation Modules
5. Project Risks
6. Testing Report
7. Tools and README
8. Project Demo
9. Functional & Non Functional Requirements
10. Good Programming Practices
11. Project Execution Results
12. Project Management

May 2024

1. INTRODUCTION

1.1 About the Project:

The Course Assessment Management System (CAMS) represents a transformative step forward in how educational institutions handle the critical process of course assessments. CAMS is a sophisticated web-based platform meticulously designed to revolutionize the management of assessments within educational settings.

1.2 Problem Statement:

The Course Assessment Management System (CAMS) aims to address several key challenges faced by educational institutions, including:

Outreach and Brand Presence:

Many educational institutions struggle with effectively showcasing their offerings and maintaining a strong brand presence in a competitive landscape. CAMS provides a solution by offering a streamlined platform that enhances the efficiency and quality of course assessments, thereby contributing to the institution's reputation for excellence in education.

Audience Engagement:

Engaging instructors, administrators, and students in the assessment process is essential for gathering accurate data and fostering a collaborative learning environment. CAMS facilitates audience engagement by providing user-friendly interfaces, communication channels, and actionable insights that empower stakeholders to actively participate in the assessment process.

Event Showcasing:

Educational institutions often host events to showcase their achievements, research, and initiatives to the wider community. CAMS contributes to event showcasing by enabling administrators to generate reports, analytics, and visualizations that highlight the institution's commitment to data-driven decision-making and continuous improvement in teaching and learning processes.

Integration with Other Applications:

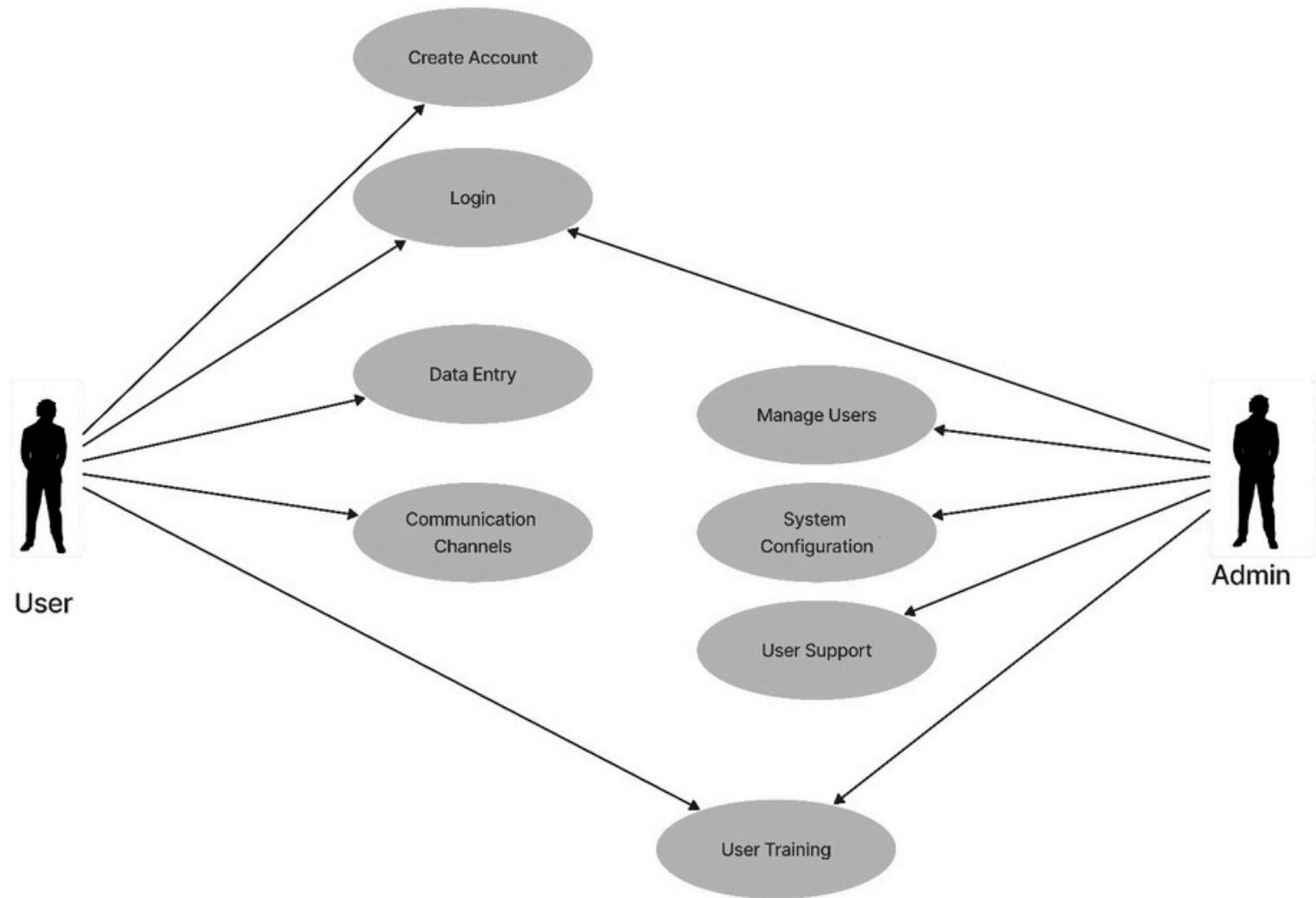
In today's digital ecosystem, educational institutions rely on a variety of applications and systems to support their operations, including Learning Management Systems (LMS), student information systems, and communication platforms. CAMS offers seamless integration capabilities.

2. REQUIREMENT GLOSSARY

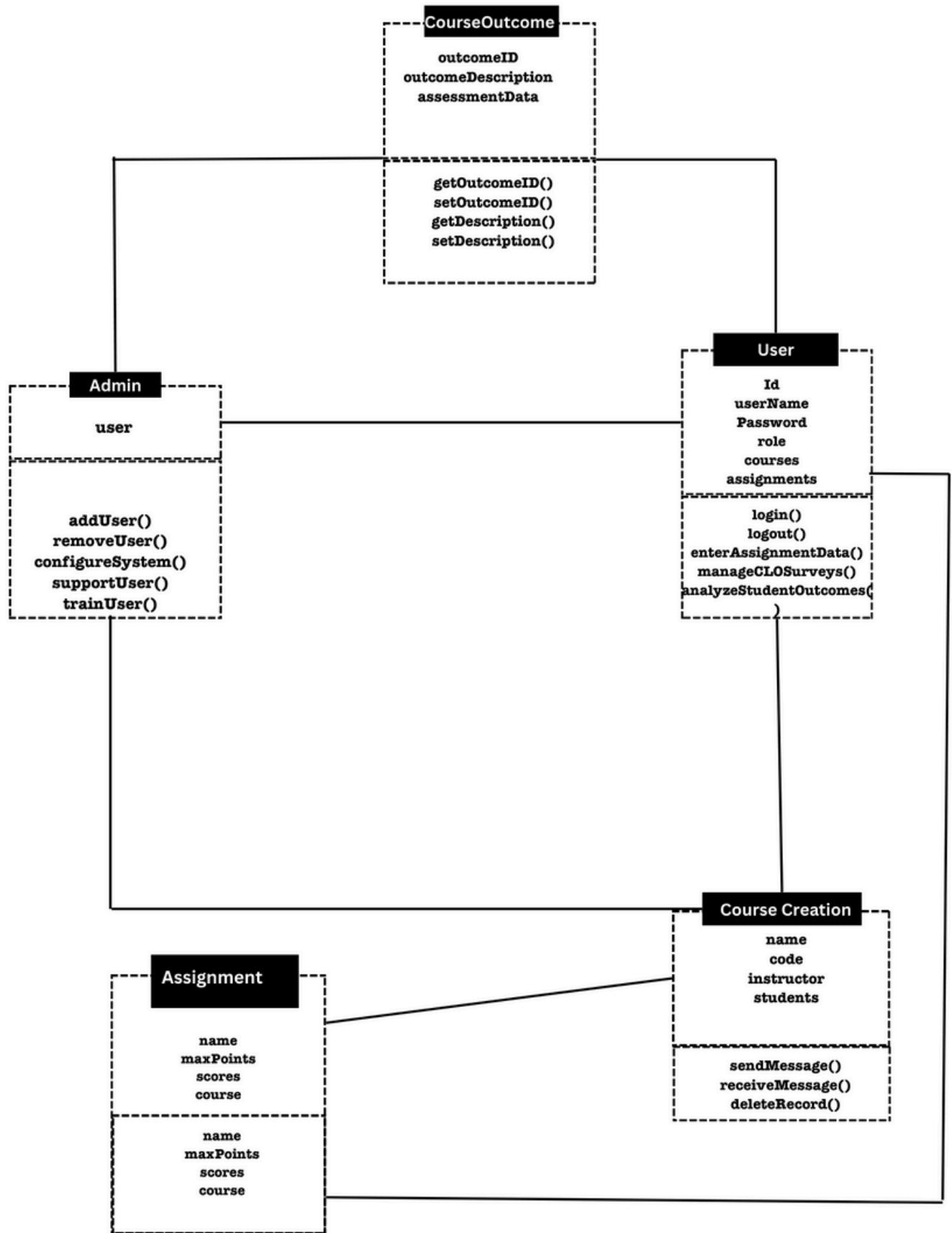
1. **User (Instructor):** In the context of the Course Assessment Management System (CAMS), an instructor refers to an individual responsible for various tasks related to managing course assessments. This includes entering assignment data, conducting surveys related to Course Learning Outcomes (CLO), analyzing student outcomes, and utilizing the functionalities provided by CAMS to enhance the teaching and learning process.
2. **Admin (IT):** The Admin (IT) role within CAMS pertains to personnel with administrative privileges responsible for overseeing the technical aspects of the system. This includes tasks such as managing user accounts, configuring system settings, providing support to users (instructors), and ensuring the smooth operation of CAMS from a technological standpoint.
3. **Other roles:** CAMS should be designed to accommodate additional user roles beyond instructors and IT admins, depending on the specific needs and structure of the educational institution.
4. **Efficiency:** Efficiency in the context of CAMS refers to the system's ability to streamline and optimize processes related to course assessment management. This involves reducing the time and effort required for tasks such as data entry, analysis, and reporting, thereby improving overall productivity and workflow for instructors and administrators.
5. **Accuracy:** Accuracy entails ensuring the reliability and correctness of assessment data stored and processed within CAMS. By minimizing errors associated with manual data entry and calculation, the system aims to maintain the integrity of assessment results and provide users with reliable information for decision-making purposes.
6. **Insight:** CAMS should provide instructors with actionable insights into student performance and learning outcomes. This involves analyzing assessment data to identify trends, strengths, and areas for improvement, thereby enabling instructors to make informed decisions and interventions to support student success.
7. **Collaboration:** Collaboration features within CAMS facilitate communication and interaction between instructors, administrators, and other stakeholders involved in the course assessment process. This includes tools for sharing feedback on assessments, discussing strategies for improvement, and coordinating activities to ensure effective collaboration and teamwork.
8. **Continuous Improvement:** CAMS should support a culture of continuous improvement in teaching and learning processes by providing mechanisms for ongoing assessment, feedback, and refinement.

3. USER CASE MODELS

3.1 USER CASE MODELS

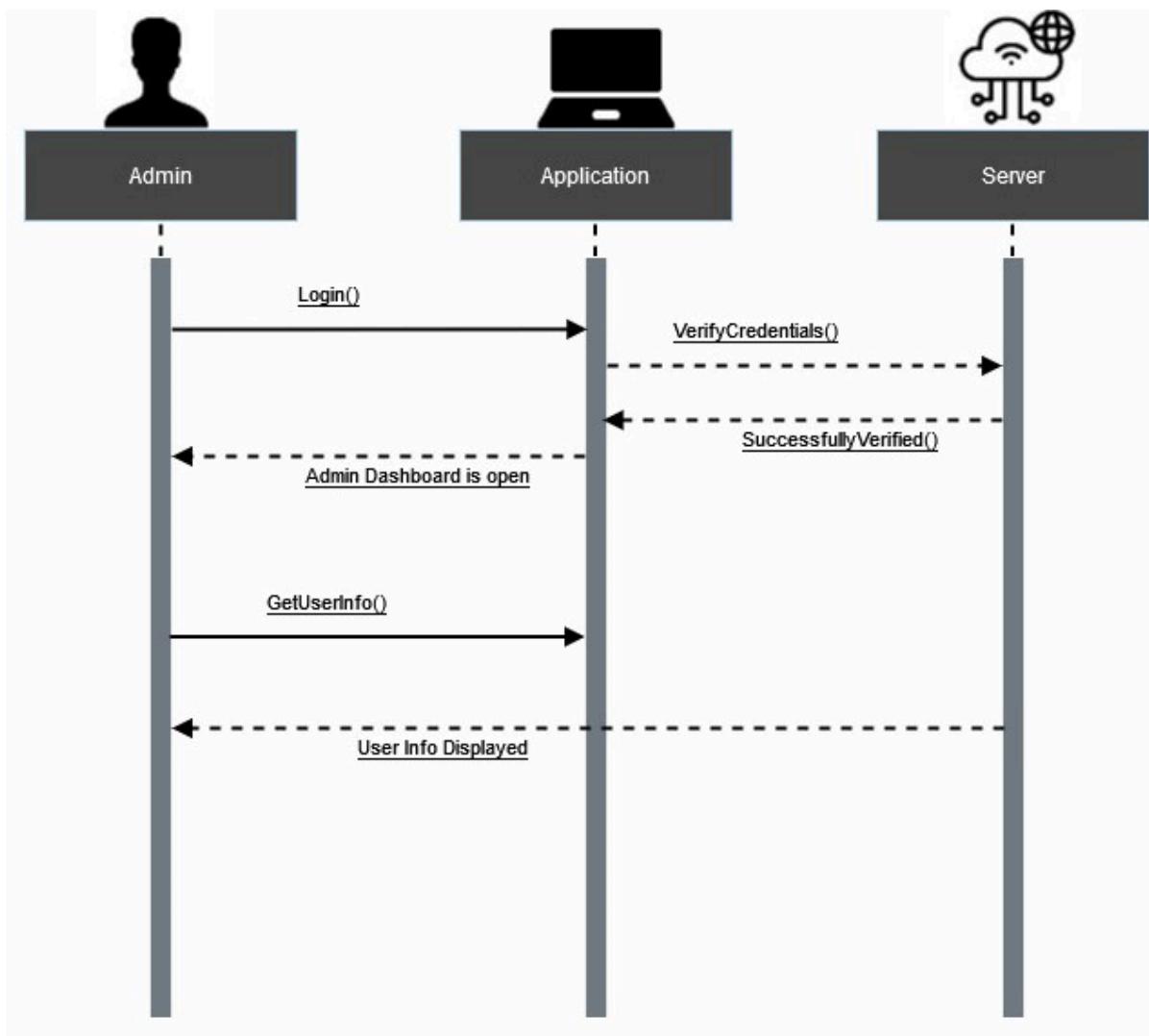


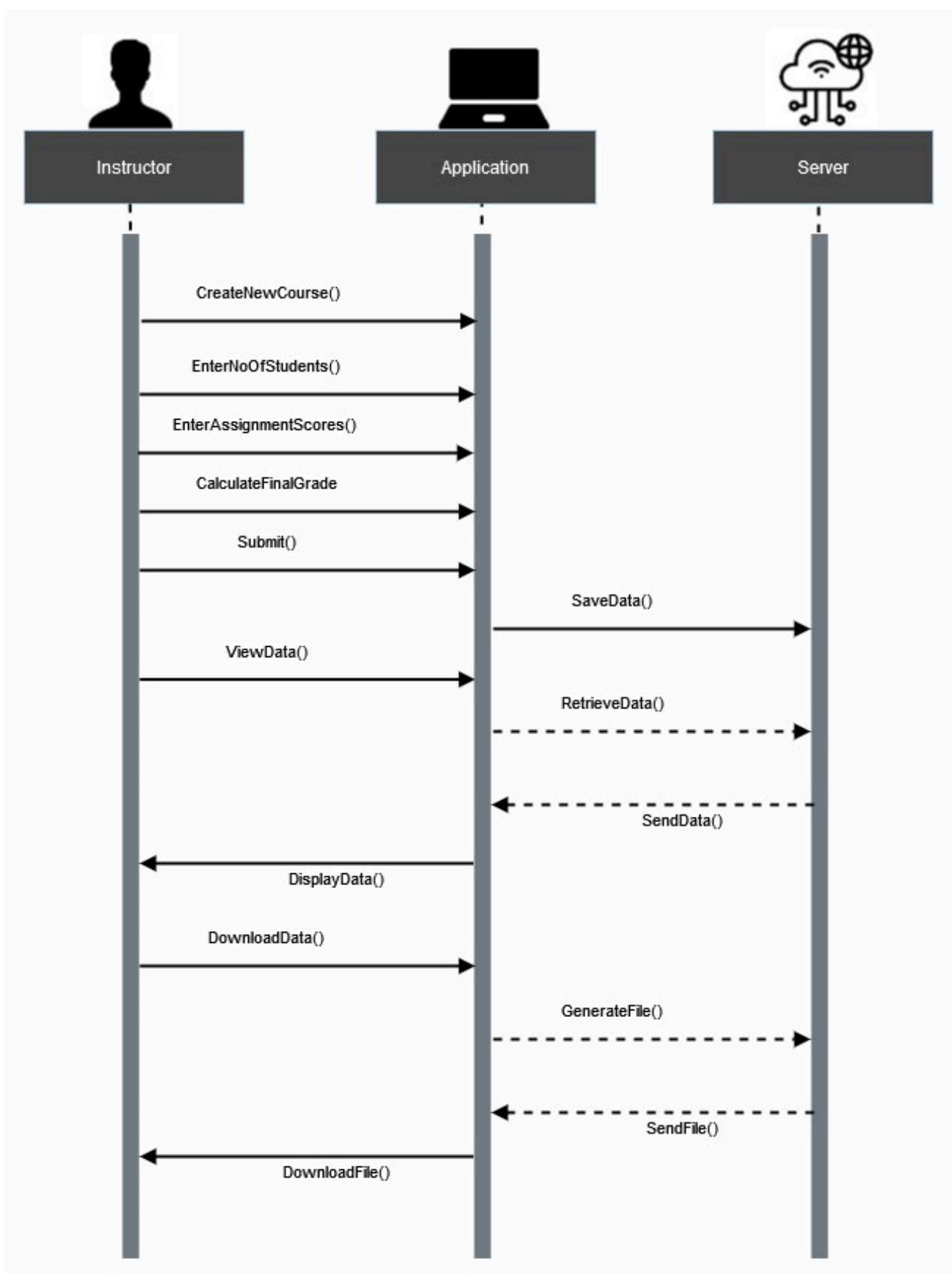
3.2 CLASS DIAGRAM

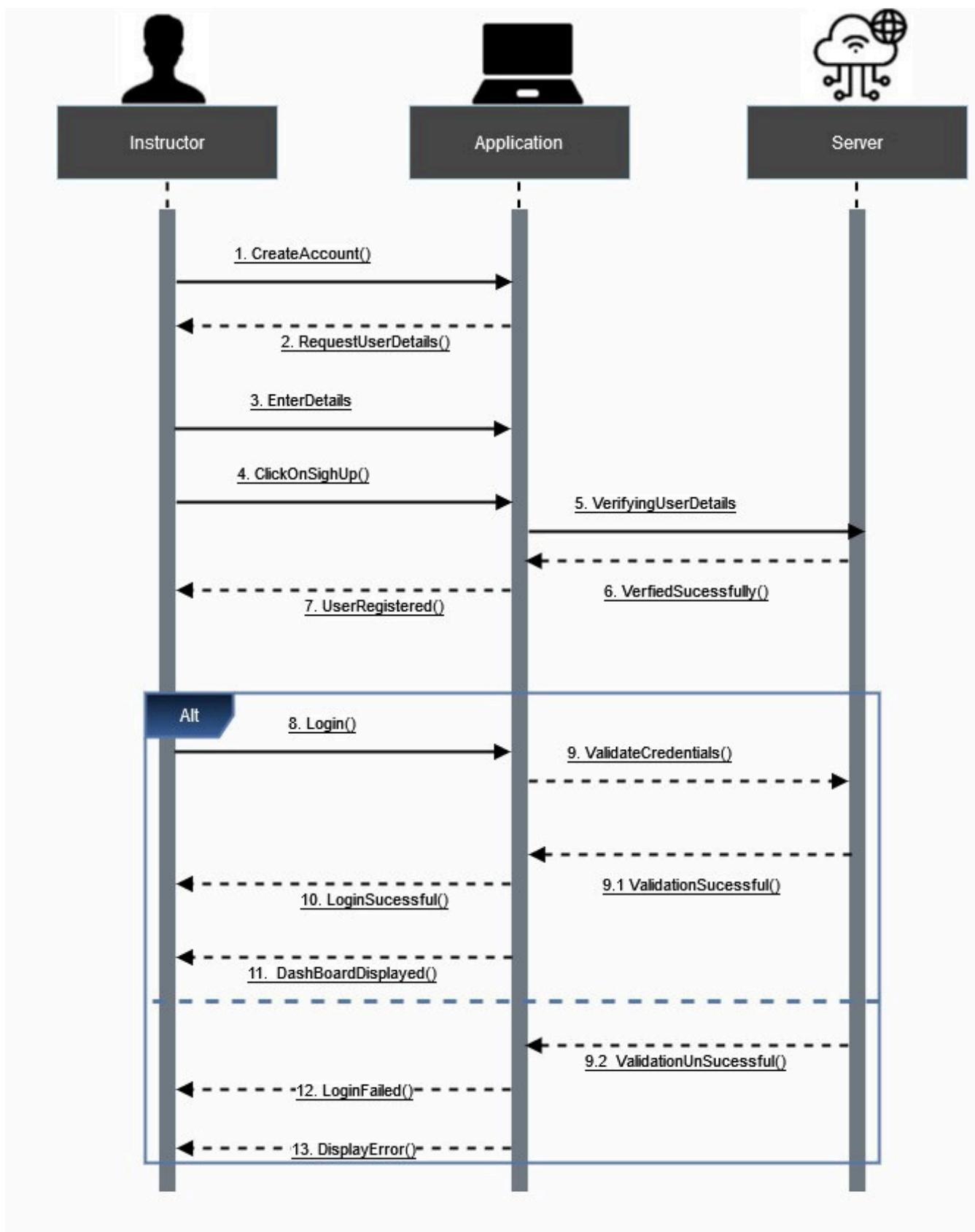


4. DESIGN & IMPLEMENTATION MODULES

4.1 USER SEQUENCE DIAGRAMS







Login:

- This use case is crucial for providing secure access to the system. Users (both instructors and admins) need to authenticate themselves by entering their unique credentials (username and password).
- The system verifies the entered credentials against a secure user database to ensure only authorized users gain access.
- If the credentials are valid, the system grants access and directs the user to their personalized dashboard based on their role (instructor or admin).
- In case of invalid credentials, appropriate error messages are displayed, guiding users on corrective actions such as retrying with correct credentials or resetting their password.

Personalized Dashboard:

- After successful login, users are presented with a personalized dashboard tailored to their role within the system.
- Instructors see functionalities related to managing courses, assessments, and student performance. This may include options like viewing course lists, creating new assessments, and accessing student grades.
- Admins are provided with functionalities for managing users, system configuration, and user support. This may include options like managing user accounts, configuring system settings, and accessing support resources.

Data Entry (Instructor):

- Instructors need to input assessment data into the system, including assignment details such as names, maximum points, and student scores.
- The system should provide user-friendly forms or interfaces to facilitate easy data entry by instructors.
- Additionally, the system should support the ability to upload bulk assessment data from external files (e.g., CSV, Excel) to enhance efficiency and reduce manual data entry efforts.

Manage Users (Admin):

- Admins have the responsibility to manage user accounts within the system, including instructors and students.
- This use case involves actions such as adding new users, editing existing user profiles, and deactivating or deleting user accounts as needed.

- Admins can assign roles and permissions to users based on their responsibilities within the system, ensuring proper access control and security.

System Configuration (Admin):

- Admins have the authority to configure various system settings and parameters to customize the system according to institutional requirements.
- This may involve setting password policies, managing assessment time limits, configuring backup schedules, and other system configurations.
- Flexibility and user-friendliness are essential aspects of system configuration to accommodate changes in institutional policies or requirements efficiently.

5. PROJECT DEMO

5.1 PROTOTYPING

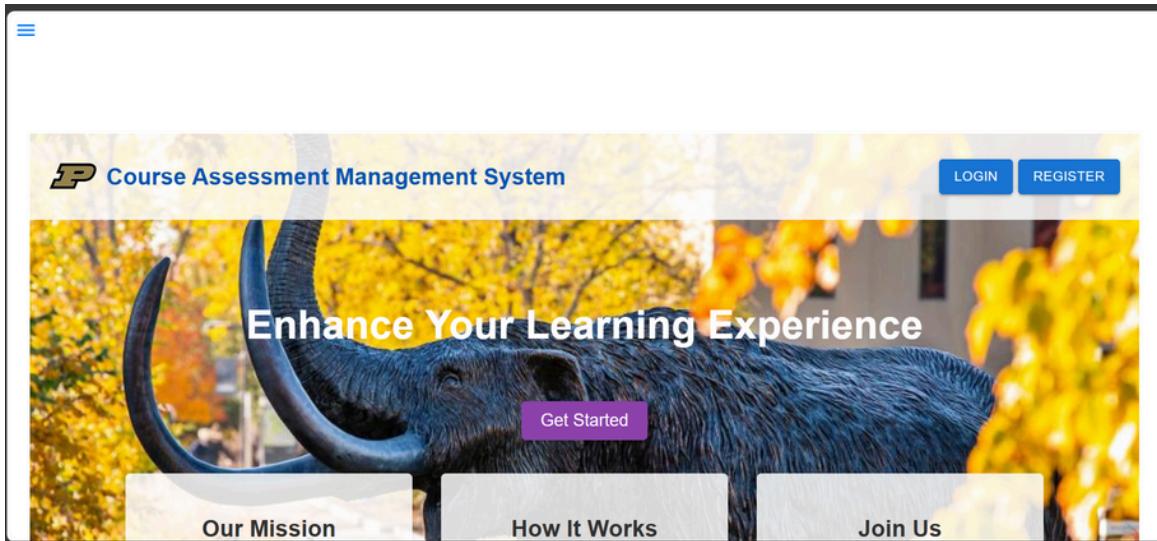


Fig 1: Get Started page

Fig 1: The CAMS Get Started page presents clear options for logging in or registering, ensuring accessibility for users. It succinctly outlines CAMS mission, functionality, and how users can get involved, providing a streamlined introduction to the platform.



Purdue University
Fort Wayne
CAMS

LOGIN REGISTER

Register

Name *

Role

Instructor

Email Address *

Fig 2: Login and Register page

Fig 2: The Login and Register Page, users input their email and password for login or account creation. They must also select their role as instructor or admin. This ensures proper access control within the system.

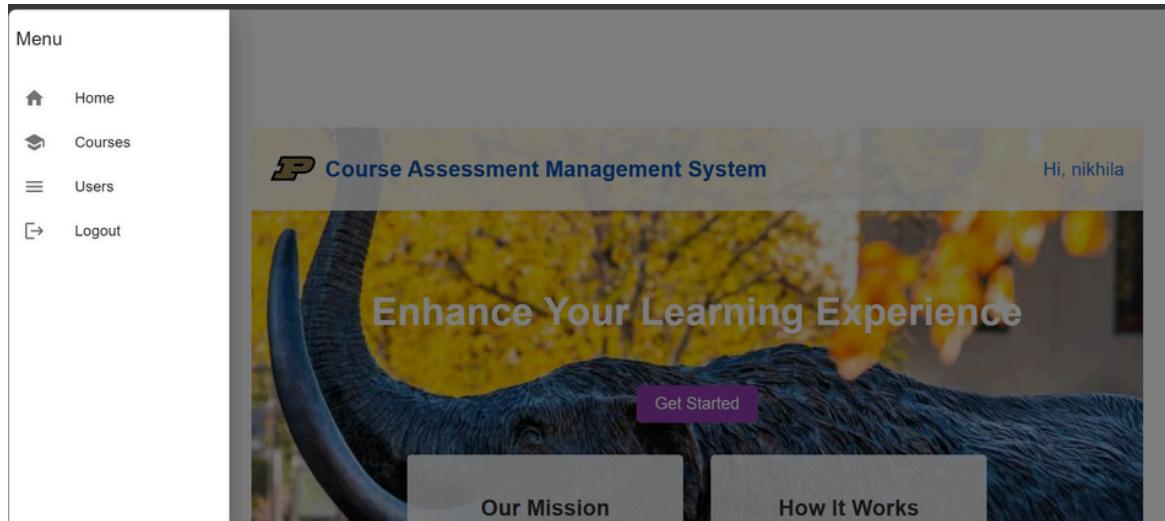


Fig 3: Admin Login

Fig 3: The Admin login provides access to essential functionalities for managing courses and users within the CAMS website. Admins can create new courses and assign them to instructors, facilitating course management processes. Additionally, they have permission to add students to the courses, ensuring proper enrollment and organization. Furthermore, admins can create user accounts for the CAMS website, enabling smooth user management and system operation. These features empower admins to efficiently oversee course administration and user management tasks within the Course Assessment Management System. User can logout from using the option present in Hamburger menu.

Grade	Grade	Grade	Grade
11	12	13	12
<input type="button" value="SUBMIT GRADE"/>			

Fig 4: Course page

Fig 4: The Course Page in the Admin profile presents course details, student lists, and grades. It displays student-entered grades and calculates the average grade for the class, aiding instructors in monitoring student progress efficiently.

User Management			
ADD NEW USER			
Name	Email	Role	Actions
v	v	instructor	EDIT DELETE
a	a@gmail.com	instructor	EDIT DELETE
swetha	swethabommireddy9@gmail.com	instructor	EDIT DELETE
rishi	rishi@email.com	admin	EDIT DELETE
Bomms	bommsr01@pfw.edu	admin	EDIT DELETE

Fig 5: User Management

Fig 5: The User Management System page is accessible only through admin login. Admins have the authority to add users to the CAMS website from this page. Instructors and other admins can log in only if their credentials are added to the user management system by the admin. This ensures proper control and authorization over user access to the CAMS website, maintaining security and management integrity.

Course Management

ADD NEW COURSE			
software engineering	cs578	Test Automation	software development
cs578	CS567	cs542	cs542
EDIT	DELETE	EDIT	DELETE

Fig 6: Course management page from Instructor profile

Fig 6: The Course Page within the Instructor profile showcases course details, student lists, and grades. It enables instructors to view student-entered grades and calculates the class's average grade, facilitating efficient student progress monitoring. Moreover, instructors possess the capability to edit course details, add new courses, and modify grades associated with students. They can also override course details to rename courses or include new students, ensuring flexibility and control over course management within the Course Assessment Management System. The Course Page in the Instructor profile presents course details, student lists, and grades. It displays student-entered grades and calculates the average grade for the class, aiding instructors in monitoring student progress efficiently.

6. TESTING REPORT

The testing report aims to outline the comprehensive testing procedures, methodologies, and results conducted on the website. The testing focused on ensuring functionality, usability, security, and performance across various modules and features. The testing covered all critical aspects and functionalities of the Course Assessment Management System (CAMS). This encompassed user roles and permissions, registration processes, assessment management, dashboard functionalities, communication channels, system configuration, and user support features.

For this project, we utilized the Cypress tool for automated testing, ensuring thorough and efficient testing of the web application. Cypress provides robust capabilities for end-to-end testing, allowing for the simulation of user interactions and validations across different scenarios.

Objectives:

1. Ensure all user roles (Instructors & Admin) function as intended with their respective permissions and access levels.
2. Validate the seamless login and registration process for users and administrators, including account creation and validation procedures.
3. Test the efficiency and security of the dashboard functionalities for instructors and administrators.
4. Assess the usability and completeness of data entry features for instructors, including assessment creation and management.
5. Validate the functionality and security of the admin panel for managing users, system configuration, and user support operations.
6. Verify the implementation and effectiveness of security measures to protect user data and system integrity.
7. Test data backup and recovery protocols to ensure data integrity and system reliability.

The Cypress tool facilitated automated testing of various functionalities, including user authentication, dashboard navigation, data entry, user management, and system configuration. By leveraging Cypress, we ensured thorough testing coverage and efficient identification of any issues or discrepancies within the system.

Overall, the utilization of Cypress for automated testing significantly contributed to the comprehensive evaluation of the Course Assessment Management System, ensuring its functionality, security, and reliability.

6.1 TESTING SNIPS

Code

```
describe('Home Page', () => {
  beforeEach(() => {
    cy.visit('https://cams-purdue.netlify.app/'); // Assuming your Home component is served at the root URL
  });

  it('renders university name and logo correctly', () => {
    cy.get('img[alt="University Logo"]').should('be.visible'); // Ensure the university logo is rendered
    cy.contains('Course Assessment Management System').should('be.visible'); // Ensure the university name is rendered
  });

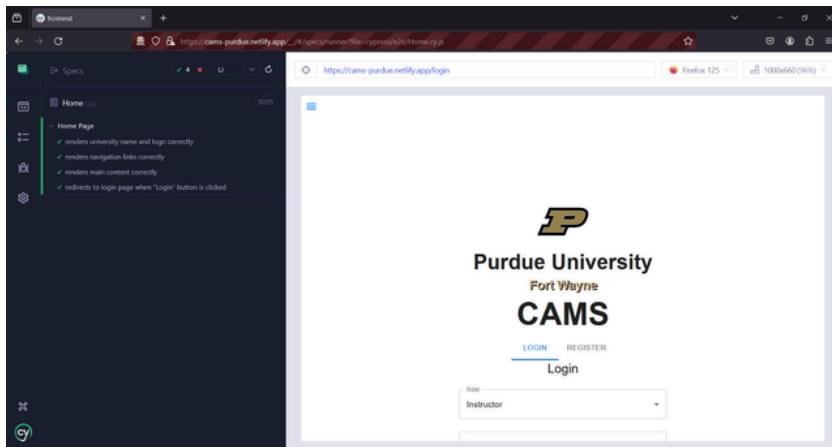
  it('renders navigation links correctly', () => {
    cy.contains('Login').should('be.visible'); // Ensure the login link is rendered
    cy.contains('Register').should('be.visible'); // Ensure the register link is rendered
  });

  it('renders main content correctly', () => {
    cy.contains('Enhance Your Learning Experience').should('be.visible'); // Ensure the main title is rendered
    cy.contains('Our Mission').should('be.visible'); // Ensure the mission section title is rendered
    cy.contains('How It Works').should('be.visible'); // Ensure the how it works section title is rendered
    cy.contains('Join Us').should('be.visible'); // Ensure the join us section title is rendered
  });

  it('redirects to login page when "Login" button is clicked', () => {
    cy.contains('Login').click(); // Click the "Login" button
    cy.url().should('include', '/login'); // Ensure redirection to the login page
  });

  // Add more tests as needed for form validation, user interactions, etc.
});
```

Results



Please refer to the attached git link for testcases: [Git-link-for-CAMS](#)

7. TOOLS AND README

Front-End: React JS

Back-End: Node JS

Database: MongoDB

Testing: Cypress

Deployment: Netlify, Render

Version Control: GIT

Prototyping & Illustrations: Figma

7.1 README INSTRUCTIONS

This web application utilizes a React.js frontend and a Node.js backend. The UI is fully functional and interacts with the backend API.

Requirements

- Node.js (recommended version: 16+)
- NPM package manager

Getting Started

Frontend:

1. Clone the repository.
2. Open a terminal in the project directory.
3. Install dependencies: `npm install`.
4. Start the development server: `npm run dev`.
5. Navigate to <http://localhost:3000> in your browser.

Backend:

1. Navigate to the backend directory.
2. Run the main application file: `npm run dev`.

8. PROJECT RISKS

1. Technical Complexity: Integrating multiple technologies and frameworks such as React.js, Node.js, MongoDB, and API integration with Learning Management Systems (LMS) can pose challenges in terms of technical complexity. There may be difficulties in understanding the interactions between different components, leading to development delays or unexpected issues.

2. Resource Constraints: Limited availability of skilled developers, insufficient budget, or inadequate infrastructure may pose risks to the project timeline and deliverables. If the team lacks necessary expertise or resources, it could lead to delays in development or compromises in the quality of the system.

3. Compatibility Issues: Ensuring compatibility with existing systems or software dependencies, such as Learning Management Systems (LMS) and student information systems, may present challenges. Incompatibilities could require additional development efforts or compromise the functionality of the Course Assessment Management System (CAMS).

4. Scope Creep: Unclear or evolving requirements can lead to scope creep, where additional features or functionalities are requested beyond the initial project scope. This could result in delays, budget overruns, and dissatisfaction among stakeholders if not managed effectively.

5. Performance Bottlenecks: Inadequate scalability or optimization of the system may lead to performance issues, such as slow response times or system downtime during peak usage periods. Failure to address performance concerns could impact user experience and hinder the success of the Course Assessment Management System (CAMS).

These risks should be carefully monitored and mitigated throughout the development process to ensure the successful delivery of the project. Strategies such as regular communication, risk assessment, contingency planning, and proactive problem-solving can help mitigate these risks and ensure the project stays on track.

9. FUNCTIONAL & NON FUNCTIONAL REQUIREMENTS

Functional Requirements:

1. User Authentication:

- Users (instructors and admins) can securely log in to the system using unique credentials.
- Password reset functionality is available for users who forget their login credentials.

2. Dashboard:

- Instructors have access to a personalized dashboard displaying key metrics and summary statistics related to course assessments.
- The dashboard provides an overview of ongoing assessments, upcoming deadlines, and overall performance trends.

3. Data Entry:

- Instructors can easily input assessment data, including assignment names, maximum points, and student scores, through user-friendly forms.
- The system supports the ability to upload bulk assessment data from external files (e.g., CSV, Excel) for efficiency.

4. Collaboration Features:

- Communication channels are available for instructors to collaborate with each other and with administrators.
- Instructors can share feedback on assessments, discuss strategies for improvement, and schedule appointments for further discussions.

5. User Management (Admin):

- Administrators can manage user accounts, including adding new users, editing existing user profiles, and deactivating or deleting user accounts as needed.
- User management functionalities include assigning roles and permissions to users based on their responsibilities within the system.

6. System Configuration (Admin):

- Administrators have the authority to configure system settings and parameters.
- Configuration options are flexible and user-friendly to accommodate changes in institutional policies or requirements.

7. User Support and Training (Admin):

- Administrators provide user support and training resources to help instructors and users effectively utilize CAMS.
- Support services include documentation, tutorials, and helpdesk assistance to address user inquiries, troubleshoot issues, and optimize system usage.

8.Regulatory Compliance (Admin):

- Administrators are responsible for ensuring regulatory compliance with relevant data privacy regulations (e.g., GDPR, FERPA) to protect the confidentiality and security of student assessment data.
- Compliance measures are regularly reviewed and updated to align with evolving regulatory requirements and best practices in data protection.

Nonfunctional Requirements:

1. Performance:

- The system must respond quickly and efficiently to user requests, especially during peak usage periods, to ensure user productivity and satisfaction.

2. Usability:

- The system interface should be user-friendly and intuitive to maximize user adoption and minimize the need for training and support.

3. Reliability:

- The system should perform consistently over time, minimizing errors, crashes, or data loss to maintain the integrity and availability of assessment data.

4. Security:

- Robust security measures should be implemented to protect sensitive assessment data from unauthorized access, breaches, and other security threats to comply with data privacy regulations.

5. Scalability:

- The system must be able to handle increased workload and user load as the institution grows, ensuring performance and functionality are not compromised. Scalability is crucial for the long-term viability of the system.

10. GOOD PROGRAMMING PRACTICES

1. We modularized complex logic into smaller, more manageable functions or modules, enhancing code comprehensibility and ease of maintenance.
2. We Implemented error messages that guide users when something goes wrong. It's like having a helpful friend who explains what's happened and how to fix it.
3. We incorporated well-established design patterns to address common programming challenges, fostering code reusability and scalability. Leveraged design patterns to enhance modularity, comprehensibility, and maintainability of the codebase.
4. We made sure user data is safe and secure, following the highest standards of data protection.
5. We conducted comprehensive testing covering unit, integration, and acceptance tests to ensure code functionality and identify potential bugs prior to release.
6. We utilized automated tools including IDEs, version control systems, and performance testing tools to maintain code consistency and reliability.

11. PROJECT EXECUTION RESULTS

Proposal: The project "Development of a Course Assessment Management System" aims to create a web-based platform to streamline the management of course assessments within educational institutions. The system will cater to the needs of both instructors and administrators, providing efficient data entry, analysis, and reporting functionalities.

Deliverables:

1. **System Requirements Specification (SRS) Document:** This document outlines the detailed requirements, functionalities, and constraints of the Course Assessment Management System.
2. **Prototype Design:** A visual representation of the system's interface and user interactions, showcasing the layout and features of the CAMS platform.
3. **Implementation of Login/Registration:** Creation of secure authentication mechanisms for users (instructors and admins) to access the system.
4. **User Screens with Add/Edit Entries:** Development of user interfaces for data entry, allowing instructors to input assessment data easily.
5. **Adding, Removing, Deleting Users for Admin:** Implementation of user management functionalities for administrators to manage user accounts within the system.
6. **Connection between Backend and Frontend:** Integration of backend (Node JS, MongoDB) with frontend (React JS) to ensure seamless data flow and functionality.
7. **UI Test Automation and Cypress Testing:** Automation of UI testing using Cypress to ensure the reliability and usability of the system.

12. PROJECT MANAGEMENT

Timeline



Sprints

<input checked="" type="checkbox"/> UN-68 Designing and Implementing the Contact Page	USER DASHBOARD AN...	IN PROGRESS	-
<input checked="" type="checkbox"/> UN-69 Designing and Implementing the About Page	USER DASHBOARD AN...	IN PROGRESS	-
<input checked="" type="checkbox"/> UN-64 Optimizing the website to search engine standards	BEAUTIFICATION	IN PROGRESS	-
<input checked="" type="checkbox"/> UN-84 Application ReadMe File	BEAUTIFICATION	DONE	-
<input checked="" type="checkbox"/> UN-31 React Routing	LANDING PAGE AND R...	IN PROGRESS	-
<input checked="" type="checkbox"/> UN-5 Bhajan Society Content Data Collection	STRUCTURE, DATA COL...	IN PROGRESS	-
<input checked="" type="checkbox"/> UN-6 Media Files Collection and Designing Elements	STRUCTURE, DATA COL...	IN PROGRESS	-
<input checked="" type="checkbox"/> UN-54 Research and Design an User Dashboard	USER DASHBOARD AN...	IN PROGRESS	-
+ Create issue			

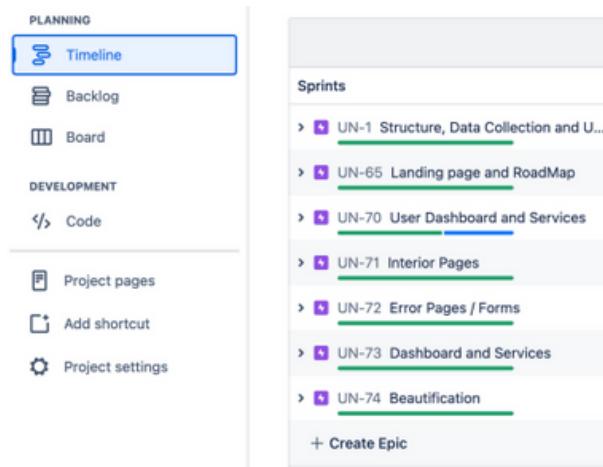
<input checked="" type="checkbox"/> UN-47 Design and Implement Sign Up / Login Page	LANDING PAGE AND R...	IN PROGRESS	-
<input checked="" type="checkbox"/> UN-76 Designing an Admin Dashboard and Assigning Privileges	USER DASHBOARD AN...	IN PROGRESS	-
<input checked="" type="checkbox"/> UN-79 Creating DB tables with parameters		TO DO	-
<input checked="" type="checkbox"/> UN-80 Mongo3T Testing, Password Hashing and Cookies Storage		IN PROGRESS	-
<input checked="" type="checkbox"/> UN-83 Event Registration and User Routes	INTERIOR PAGES	TO DO	-
<input checked="" type="checkbox"/> UN-4 Updating Misc Pages	INTERIOR PAGES	TO DO	-
+ Create issue			

<input checked="" type="checkbox"/> UN-78 Registration Page Testing & Continuation	USER DASHBOARD AN...	TO DO	-	
<input checked="" type="checkbox"/> UN-87 Finding out Page Vulnerabilities	DASHBOARD AND SER...	TO DO	-	

Backlog Tasks

Backlog (6 issues)		0 0 0	Create sprint
<input checked="" type="checkbox"/> UN-30 Discussion / Form Page (exempt)		TO DO	-
<input checked="" type="checkbox"/> UN-61 Blog Style Layout to add useful user stories / suggestions		TO DO	-
<input checked="" type="checkbox"/> UN-66 Web Testing		TO DO	-
<input checked="" type="checkbox"/> UN-67 Bugs Audit		TO DO	-
<input checked="" type="checkbox"/> UN-75 Re-Designing / Re-Structuring specific sections of the website		TO DO	-
<input checked="" type="checkbox"/> UN-77 Newsletter Design		IN PROGRESS	-

Final week timelines of the JIRA Board



WORKFLOW PLAN



FREQUENCY OF MEETINGS

Twice/Thrice a week in-person meetings

COLLABORATION TOOLS

Discord, Zoom: Video Conferencing

JIRA, Notion: Task Assignment

Miro: Design Board

May 2024

**PURDUE UNIVERSITY.
FORT WAYNE
CAMS**

A Document Prepared By

Group 1

Bommireddy Swetha Reddy

Velamuri Soumya

Akula Nikhila

Guda Ridhi