**Capstone Course Evaluation System**

**Team Members:** Cristina Powers, Raad Bhuiyan,Abigail Noyes, Bharath Palanisamy

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Description | Author | Comments |
| 1/25/2023 | Version 1.0 | Bharath Palanisamy  Abigail Noyes  Raad Bhuiyan  Cristina Powers | First Draft |
| 1/27/2023 | Version 2.0 | Bharath Palanisamy  Abigail Noyes  Raad Bhuiyan  Cristina Powers | First Revision after GTA comments |

Contents

[1 Project Overview 3](#_Toc125642618)

[2 Project Purpose, Scope, and Objective 3](#_Toc125642619)

[3 Team Organization 4](#_Toc125642620)

[3.1 Roles 4](#_Toc125642621)

[3.2 Responsibilities 4](#_Toc125642622)

[4 Problem Resolution Policies 5](#_Toc125642623)

[5 Project Plan 6](#_Toc125642624)

[5.1 Iterations 6](#_Toc125642625)

[5.2 Project Schedule 6](#_Toc125642626)

[6 Configuration Management 7](#_Toc125642627)

[7 Technologies 8](#_Toc125642628)

[7.1 Backend 8](#_Toc125642629)

[7.2 Database 8](#_Toc125642630)

[7.3 Frontend 8](#_Toc125642631)

[7.4 Server 8](#_Toc125642632)

# 1 Project Overview

The Capstone Course Evaluation System is a web application that serves as a secure centralized documentation system to organize grades of the students in the Capstone Course for the GTAs and the professor. The goal is to provide a platform that is easily accessible and will store feedback between users in one place. The application will automate many tasks and organize all documentation using a systematic file management system. The users of the Capstone Course Evaluation can read and write to the grading material in the system for ease of access.

# 2 Project Purpose, Scope, and Objective

The aim of this project is to simplify the workload for GTAs by providing them with a more efficient means of organizing their evaluations of students' performance. Instead of relying on manual methods or Excel, the project will deliver a collaborative web application that allows for organized querying and viewing of data.

The scope of this project is to develop a web application that includes a secure and well-organized documentation storage system. The administration, represented by the computer science department of Wayne State University, will be required to create the semester for the professors and GTAs to use. They must also authenticate professors in the system so that they can sign in. Professors will have the ability to authenticate the GTAs' credentials before they can sign in. User interface controllers will be established to enable the GTA to assign themselves to specific groups and students. Once signed in, professors and GTAs can read and write data from their local computer to the Course Evaluation System. All data will be stored in a MySQL database and displayed in an organized manner on a PHP-based web application. This grading system is solely for editing and adding information about student evaluations and does not allow users to upload files. The web application will provide a single platform for the admin, professors, and GTAs to view, organize, and update their students' evaluations, with no provision for private notes. The notes will be visible only to the author and the professor, and other GTAs cannot view them. Access to the application will be limited to registered users only.

The Course Evaluation System aims to offer reusable software that can support GTAs and professors in their grading requirements. The system will utilize PHP for the frontend, while a newly designed MySQL database will serve as the backend technology. Communication between the MySQL database and the user interface will be established using the local host server Apache. The user interface will comprise a login screen, student evaluation files, messages from the professor, and a note-taking feature. As these features are subject to evolution, it is crucial to adhere to different design standards without significant modifications. To address the problem, the structure of the MySQL database will be designed accordingly, with clear classes and objects. This will be achieved through interface design, hierarchical structure establishment, and appropriate variable naming conventions.

# 3 Team Organization

## 3.1 Roles

Cristina Powers: Team Leader, Frontend team member, Presentation lead

Raad Bhuiyan: Backend team member, Presentation QA, Documentation QA

Abigail Noyes: Backend team member, Frontend QA, Documentation QA

Bharath Palanisamy: Documentation Lead, Frontend team member, Backend QA

## 3.2 Responsibilities

Team Leader

The team lead will organize all meetings with their team, client, GTA, and instructor. The team leader will delegate work or re-delegate when necessary. They will enforce the Problem Resolution Policies when group members have conflicting ideas on how to move on with the project. When the team is fraught with indecision among the group, the team leader will make the final decision. As a last resort, the team leader may also request advice from the GTA for help on a decision.

Presentation Lead

The presentation lead will oversee all presentations and demonstrations of the product. They will lead group presentations by delegating parts of each presentation to each team member while deciding which topics are most important to cover.

Documentation Lead

The documentation lead will face the role of overseeing all documentation required for this project. Some responsibilities include creating the skeleton, assigning sections, and deciding what information is necessary to add to the documentation. It is necessary for the lead to ensure all grammar is correct while ensuring the style is appropriate.

Quality Assurance (QA)

The quality assurance team members will take part in reviewing their respective fields. They will verify that the program given to them is of the highest quality possible with no errors. They will also provide the information or functionality requested to improve the program.

Frontend/Backend

The frontend and backend team members will oversee working on their respective domains of the website. It is their responsibility to learn the necessary information and skills needed to fulfill all the requests of the client. They will also be the first to be asked technical questions about their respective roles within the project. It is important that each team member has a wide understanding of their code and can give meaningful answers to all questions that are asked of them during presentations or meetings. If team members are having trouble implementing any features, they will be expected to do an adequate amount of research and reach out to the entire team with questions, before asking the GTA assigned to the group.

# 4 Problem Resolution Policies

When conflicting decisions come up between group members about the project specifications, the team leader has the final say in design specifications after the client. Due to a similar project being worked on in the past, the most reliable technology is the one recommended to us by the client.

For matters not related to the technology side, one warning will be given out to a team member for not completing a deadline or for missing a meeting with no prior notice. For medical/family emergencies, team members will be given a three to seven day leniency on their tasks. For these types of emergencies, a warning may still be given out to a team member if a deadline is quickly approaching and said member has not completed an adequate amount of their assigned work. After a second warning is given out, team members will turn to the GTA for further consulting. Our team understands that since a GTA’s time is valuable, requesting problem resolutions from a GTA could result in extreme consequences for the team member receiving a second warning.

# 5 Project Plan

## 5.1 Iterations

The team will have five reoccurring meetings throughout the week with their assigned GTA to discuss the application progress and ask questions. On Friday mornings the team will host “Q and A’s” with the client and update them on the production status. The team will meet on Sunday, Monday, and Wednesday to discuss their plans and update each other on their daily progress. Posted below is the precise meeting schedule:

* Team Zoom Meeting
  + Sundays from 5:00 PM - 6:00 PM EST
* GTA Zoom Meeting
  + Mondays from 12:00 PM – 1:00 PM EST
* Team Undergraduate Library Meeting
  + Mondays & Wednesdays from 2:00 PM to 5:00 PM EST in room 2323 of the UGL
* Client Zoom Meeting
  + Fridays from 11:00 AM to 12:00 PM EST

## 5.2 Project Schedule

Our team has a strict project schedule shown below:

* First Prototype: **February 4th**
  + The database will be designed and fully functional.
* Software Requirements Specification (SRS): **February 11th**
  + The document will clearly contain the functional and nonfunctional elements of the application and discuss interface requirements and design constraints.
* Design Specification: **February 25th**
  + The document will hold the necessary information needed to build the application, including an in-depth view of the architectural and system design.
* Second Prototype: **March 4th**
  + A completed dashboard for the GTAs and professor with basic UI.
* Third Prototype: **March 25th**
  + The notification aspect will be fully developed, and the forms for the GTAs to fill out will have a basic UI.
* Test Plan: **April 1st**
  + This document will be an extensive report for testing the application, the functional and nonfunction requirements individually, and the system.
* Final Application: **April 15th**
  + The complete application will be finished and ready for presentation**.**

# 6 Configuration Management

The team will use Discord as their form of communication. From there, the team will have different channels to give SCRUM updates, ask questions, and share documentation work. All documentation will be shared in a OneDrive folder provided by the documentation lead, ensuring that every member can participate in development. The team will also use Trello’s Kanban board to aid in tracking progress while helping the team to understand the state of the project and visualize where to move on from there.

The team will use Git as our source control method while using GitHub to host our code. Git will keep a history of all changes made, which includes reverting, creating, and deleting files when needed. Each member will have their own branch to work on their designated tasks. When a member finishes their designated job, they will create a pull request, explicitly assigning the other member of their respective team (frontend/backend) and their assigned QA. The two will then perform a code review. They will analyze the modified code and leave questions and comments that need addressing before the merge. A branch will not be committed to the main if it is not fully functioning.

# 7 Technologies

## 7.1 Backend

For the backend language, our team will use PHP, a powerful language for creating dynamic webpages that can interact with databases. The version of PHP the team decided to use is 8.2. Many people use PHP because of its simplicity, flexibility, and speed.

## 7.2 Database

The application will utilize Version 8.0 of MySQL for its database language. Our team decided to use MySQL for many reasons, the biggest one being MySQL is a relational database management system. Since MySQL is a relational database management system, the team can utilize multiple tables together to get the best output. MySQL databases help automate data retrieving and provide great data support in PHP MySQL web application development.

## 7.3 Frontend

The bootstrap UI is a framework used in web development that our team will adapt as part of our frontend development. Bootstrap has the basic requirements for responsive web development, which allows users to insert code into a pre-defined grid system. The team will use the most recent version, which is Bootstrap 5. Bootstrap 5 is an improvement because it does not require the dependency of jQuery but allows it, if necessary, by offering more freedom with styling web applications. Our team will also use Figma to help design a blueprint before development.

When it comes to unit testing, our team will conduct multiple tests to help us find bugs in the application and fix them before the final product. The team will use PHP Unit version 10, a unit-testing framework designed to help locate mistakes or errors. By doing this consistently, our team will identify issues early to carry out proper corrections in the software.

## 7.4 Server

Since our application is on the web, our team will need a server to host and deploy the web application. Our host server is Apache version 2.4.55. This version has the benefit of having fewer bugs than the previous versions. It has many security features with an HTTP server that is integrated with Apache. While using Apache, we do not need to find a cloud-based server to host and deploy our web application. If our team were to use a cloud-based server, an annual fee would be needed to proceed. Since we are using a local server, there is no need to worry about cloud services now.