Prepared by: Bharath Palanismy

Raad Bhuiyan

Cristina Powers

Abigail Noyes

Version 3.0 approved

April 16, 2023

Software Requirements specification

Capstone Course Evaluation System

[ii. Revision History 3](#_Toc132441224)

[1. Introduction 4](#_Toc132441225)

[1.1 Purpose 4](#_Toc132441226)

[1.2 Scope 4](#_Toc132441227)

[1.3 Intended Audience and Reading Suggestions 4](#_Toc132441228)

[1.4 References 5](#_Toc132441229)

[1.5 Overview 5](#_Toc132441230)

[2. General Description 5](#_Toc132441231)

[2.1 Product Perspective 5](#_Toc132441232)

[2.2.2 Admin Features 6](#_Toc132441233)

[2.2.2.1 Add Semester 6](#_Toc132441234)

[2.2.3 Professor Features 6](#_Toc132441235)

[2.2.4 GTA Features 7](#_Toc132441236)

[2.3 User Classes and Characteristics 7](#_Toc132441237)

[2.4 General Constraints 8](#_Toc132441238)

[2.5 Assumptions and Dependencies 8](#_Toc132441239)

[3. Specific Requirements 8](#_Toc132441240)

[3.1 External Interface Requirements 8](#_Toc132441241)

[3.1.1 User Interfaces 8](#_Toc132441242)

[3.1.1.1 Admin Interface 11](#_Toc132441243)

[3.1.1.2 Professor Interface 15](#_Toc132441244)

[3.1.1.3 GTA Interface 25](#_Toc132441245)

[3.1.2 Hardware Interfaces 31](#_Toc132441246)

[3.1.3 Software Interfaces 31](#_Toc132441247)

[3.1.4 Communications Interfaces 31](#_Toc132441248)

[3.2 Functional Requirements 32](#_Toc132441249)

[3.2.1 User Sign In 32](#_Toc132441250)

[3.2.2 User Sign Out 33](#_Toc132441251)

[3.2.3 User Set Password 34](#_Toc132441252)

[3.2.4 User Forgot Password 35](#_Toc132441253)

[3.2.5 Create Semester 36](#_Toc132441254)

[3.2.6 Create Section 36](#_Toc132441255)

[3.2.7 Add Professors 37](#_Toc132441256)

[3.2.8 Edit Professors 37](#_Toc132441257)

[3.2.9 Delete Professors 38](#_Toc132441258)

[3.2.10 Add Section 39](#_Toc132441259)

[3.2.11 Add GTAs 39](#_Toc132441260)

[3.2.12 Edit GTAs 40](#_Toc132441261)

[3.2.13 Delete GTAs 41](#_Toc132441262)

[3.2.14 Add Students 41](#_Toc132441263)

[3.2.15 Delete Student 42](#_Toc132441264)

[3.2.16 Create Groups 43](#_Toc132441265)

[3.2.17 Add Student to a Group 43](#_Toc132441266)

[3.2.18 Edit Groups 44](#_Toc132441267)

[3.2.19 Give Student a Grade 45](#_Toc132441268)

[3.2.20 Pick Groups 46](#_Toc132441269)

[3.2.21 View Groups 46](#_Toc132441270)

[3.2.22 Grade Weekly Reports 47](#_Toc132441271)

[3.2.23 Grade Assignments 47](#_Toc132441272)

[3.3 Non-Functional Requirements 48](#_Toc132441273)

[3.4 Design Constraints 50](#_Toc132441274)

[3.4.1 Database Constraints 50](#_Toc132441275)

[3.5 Logical Database Requirements 50](#_Toc132441276)

[4. Analysis Models 52](#_Toc132441277)

[4.1 Data Flow Diagrams (DFD) 52](#_Toc132441278)

[4.1.1 DFD Level 1 52](#_Toc132441279)

[4.1.2 DFD Level 2 53](#_Toc132441280)

[4.1.2.1 DFD Level 2 – Admin 53](#_Toc132441281)

[4.1.2.2 DFD Level 2 – Professor 54](#_Toc132441282)

[4.1.2.3 DFD Level 2 – GTA 56](#_Toc132441283)

# ii. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 1/24/2023 | Version 1.0 | Bharath Palanisamy  Abigail Noyes  Raad Bhuiyan  Cristina Powers | First Draft |
| 2/18/2023 | Version 2.0 | Bharath Palanisamy  Abigail Noyes  Raad Bhuiyan  Cristina Powers | Second Draft |
| 4/16/2023 | Version 3.0 | Bharath Palanisamy  Abigail Noyes  Raad Bhuiyan  Cristina Powers | Final Draft |

1. Introduction

1.1 Purpose

The purpose of this Software Requirement Specification (SRS) is to give a detailed description of the requirements for the Capstone Course Evaluation System. The document will explain the system constraints, interface, and capabilities. The SRS document will also explain the functional requirements, non-functional requirements, and give an in-depth guide for those who wish to understand the application thoroughly. This document will serve as a reference for the developers and client during the development process.

## 1.2 Scope

This application will simplify the entire Capstone Course Evaluation System (CCES). The CCES application allows three types of users: admins, professors, and Graduate Teaching Assistants (GTAs). The admin will have to create a semester and add a professor before they can sign into the application. The admin can add other admins, remove members from the application, create a section for a professor, and review all GTAs reports.

Once the professor signs in, they will create sections, add GTAs to the application, assign GTAs to specific sections, upload students via an excel file, create and assign the students to groups based on the survey he gives on canvas. The professor can see the grades and evaluations written by a GTA for individual students and groups.

The GTAs must wait until the professor adds them to the system to sign into the application. The GTAs can then pick the groups they would like to work on if there are groups with no assigned GTAs. When the GTA selects a group to be assigned to, no other GTA can select that group. Note, the professor can override the GTA group selection and modify group assignments. The GTA will then be able to access only their groups to grade assignments and give evaluations on the students’ weekly reports. On top of grading each group, the GTA will also grade each person for their individual behavior.

1.3 Intended Audience and Reading Suggestions

|  |  |
| --- | --- |
| **Term** | **Definitions** |
| CCES | The abbreviation for the application name, Capstone Course Evaluation System. |
| User | An individual who uses the Capstone Course Evaluation System |
| Role | The type of user that can sign into the Capstone Course Evaluation System which are admin, professor, and GTA. |
| Admin | A user of Capstone Course Evaluation System who is granted all access. |
| GTA | This abbreviation for Graduate Teaching Assistant who use the Capstone Course Evaluation System |
| FR | The abbreviation for the word functional requirements |
| NFR | The abbreviation for the word non-functional requirements |
| Professor | The teacher assigned to a capstone course who uses the Capstone Course Evaluation System |
| Semester | The term in which the specific class is taking place, ex: Fall 2023, Winter 2023, etc. |
| Section | The specific class that the professor is going to teach for the semester. |

1.4 References  
[1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

[2] “Limitations of MySQL 8.” *Packt Subscription*, https://subscription.packtpub.com/book/data/9781788395199/1/ch01lvl1sec06/limitations-of-mysql-8.

1.5 Overview

Several components are contained in the remaining sections of this SRS document. The second section will discuss the general description of the CCES application. To elaborate a little bit more, this section of the document will explain the software interaction. The third section will give an in-depth explanation of the software requirements of the application. This section includes function and non-functional requirements, external interface requirements, design constraints, and logical database requirements. The final section of the document will show the data flow diagrams.

# General Description

2.1 Product Perspective  
 Capstone Course Evaluation System is a web-based project consisting of three interfaces. One for the admin of the site, the professor of the class, and the GTAs of each section. When a user opens the website, they are greeted with a sign in page asking them to enter their Wayne State University email. Based on who signs in, the application will detect whether the user is an admin, GTA, or professor and present them with their respective dashboards. The admins can view all grades and reports, past, present, and future and has the ability to create semesters and add professors to the application. The professor’s experience focuses on advising the GTAs and viewing their progress with their assigned students. The professor can only see the current semester, not past or future. The GTA’s experience centers around grading their assigned groups and individual students in those groups on performance and progress.

The website will communicate with the database to retrieve and/or send requested information based on the role of the signed in user. The information is either consumed or modified based on the role of the user.

2.2 Product Features

2.2.1 General Features

#### 2.2.1.1 User Sign In

* User will sign into the website if their account has been created.
* The application will detect role of user through the database.
* User can create a password.
* User can reset password if forgotten.

#### 2.2.1.2 Sign Out

* User can sign out of the app.

#### 2.2.1.3 Navigation Menu

* Navigation menu will have an easily accessible screen for all pages on the top left corner of the web application.
* Users can navigate to the “Home” and a “Sign Out” page.
* Users can navigate through an array of different features depending on the role of the user.

### 2.2.2 Admin Features

### 2.2.2.1 Add Semester

* Creates a new semester of Senior Capstone at Wayne State University.
* The semester will expire when it reaches its end date.

#### 2.2.2.2 Add Professor

* Admin can add, update, or disable a professor as a user within the website.

#### 2.2.2.3 Add Section

* Admin can create a section for a current or future semester.
* Admin can assign a professor for each section.

### 2.2.3 Professor Features

#### 2.2.3.1 Add Sections

* Professors can create a section to the active semester if he has been assigned to it.

#### 2.2.3.2 Add GTAs

* Professors have access to add, update, or delete a GTA as a user.
* Professors assign a GTA to a section.

#### 2.2.3.3 Add Students

* Professors can add students to each section by uploading a document such as a CSV file with student identification.

#### 2.2.3.4 Create Groups

* Professors can create one or more project groups in each section.
* Professors will assign the students to a project group.

#### 2.2.3.5 Edit Groups

* Professors have access to edit already existing groups for each section of their assigned semester.

#### 2.2.3.6 Grade Assignments

* Professors can grade the students and groups on the given assignments and presentations given in class.

#### 2.2.3.7 View Student Grades

* Professors have access to view a student’s grades given by both them and the GTA, the Weekly Reports, Prototypes, Midterm and Final grades.

### 2.2.4 GTA Features

#### 2.2.4.1 Pick Groups

* GTAs are allowed to look through created groups of their section not already chosen and choose which ones they want to be assigned to evaluate.

2.2.4.2 View Groups

* Able to view information for each group/member the GTA chose.

2.2.4.3 Weekly Reports

* Will show status of weekly reports.
* Evaluate weekly reports given to them by the students of their assigned groups.

2.2.4.4 Grade Assignments

* Give an evaluation on assigned group’s progress, documentation, and presentations as well as evaluate each individual student in the group.

2.3 User Classes and Characteristics  
 This web application can only be accessed by the professors, GTAs, and an admin, who is a member of the department of computer science at Wayne State University. Thus, the users of this product will be English speakers who are educated in the field of computer science. They will have a great amount of expertise in the technical field, as well as experience in the development of a large programming project with a group in a professional setting. The features added will make sure they can provide their expert evaluations on projects as easily as possible.

2.4 General Constraints

Due to the project needing to be completed during a semester in college, time is something that is a limiting factor that we must take into account when deciding the number of features we want for the site. Decisions on what the team will want to add to the site along with what the client wants must be checked on whether it is something that can realistically be done in a reasonable time. We must also gauge whether the features decided on are something within the team’s ability at this time due no one on the team having utilized the programming language PHP before now causing limiting knowledge of the tech stack.

The database the team has chosen to use is MySQL v.8 which has no limits to the number of tables and databases created. However, limitations like the row size which allows at most 65,535 bytes, table size which has a limit of 4096 total columns allowed, and 61 tables in one join at most are some limitations found with the chosen database.

2.5 Assumptions and Dependencies

The team assumes that users will consist of only faculty members from Wayne State University who are associated with the Senior Capstone course. The users should already have a basic understanding of how the course at Wayne State University is structured and run. This website will not be available to the public or any other business. The users should also have access to a computer with a working browser such as Google Chrome, Microsoft Edge, Opera, Mozilla Firefox, or Safari.

1. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The web application consists of three user interfaces: the admin, professor, and GTAs. All users will be presented with the same sign in page, as shown in Figure 1. A user must be added to the application via an admin or professor before signing in, which is why there is no registration.

Graphical user interface, application

Description automatically generated

**Figure 1**

If it is the users first time signing into their account, they will be sent to the page and see the form in Figure 2 asking them to set their password. The user will be asked to insert their email address that is in our database (their Wayne State University email) and hit “Receive new password by email.” The screen will alert them with the message “Check your email!”

Graphical user interface, text, application

Description automatically generated

**Figure 2**

The user will receive a email that has a personal link that they click to go to open the website link that is shown in Figure 3 below. The user will enter a password and must repeat it. If the passwords do not match, they will be an error messaging telling them to try again. When enters matching inputs and click “Set password” they will be redirected to the Sign In screen in Figure 1 to sign in now with their new password.

Graphical user interface, text, application, email

Description automatically generated

**Figure 3**

When the user enters their email and has already set a password, the screen will refresh with the form shown in Figure 4 below asking for password input. If they enter their password without errors, they will be taken to their respective dashboards based on their roles (admin, professor, GTA). If the user has forgotten their password, they can click the “Forgot password” and go through the progress that was shown in Figures 2 and 3 to reset their password.

Graphical user interface, application

Description automatically generated

**Figure 4**

3.1.1.1 Admin Interface

If the admin is signed in, they will be redirected to their dashboard. The dashboard will display the previous five semesters as buttons. The admin can also search for later semesters underneath as shown in Figure 5.

Graphical user interface, text, application, chat or text message

Description automatically generated

**Figure 5**

When the admin clicks on a given semester button it will take them to that semester page and display the sections and groups in tab format. Within the tabs, the admin will be able to see the reports given by the professor and GTA for those students. The admin is the user of the application that can see everything. They can see the history of past and present semesters, professors and GTA’s, and the reports for each student. A visual is given in Figure 6 below.

Graphical user interface, text, application, email

Description automatically generated

**Figure 6**

In the navigation, the admin will be able go to their “Home” dashboard, “Add Semesters/Sections,” “Add Professors,” and “Sign Out, shown in Figure 7.

Graphical user interface, application

Description automatically generated

**Figure 7**

Figure 8 displays the forms the admin will fill out, entering valid inputs in the fields and selecting “Add” to add the semester/section.

Graphical user interface, application

Description automatically generated

**Figure 8**

Figure 9 below shows how the admin will be able to add professors to our database. The form asks for the first name, last name, and Wayne State University email address. Underneath will show a list of all active and deactivated professors in the system that the admin can search and change their activity status. The admin will also be able to edit the professor’s information (first name, last name, email) if there is an error or change in the name.

Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

**Figure 9**

3.1.1.2 Professor Interface

If a professor is signed in, they will be directed to their dashboard. The dashboard will display the current semester at the top and the sections in tab format, having the groups that are in those sections when the tab is clicked, show in Figure 10.1 below.

Graphical user interface, website

Description automatically generated

**Figure 10.1: Before clicking group.**

Graphical user interface, application, website

Description automatically generated

**Figure 10.2: After clicking on a group.**

When the professor clicks on the groups it will display the students that are in the groups, shown in Figure 10.2 above. The students’ names will be buttons that once clicked will take them to that student’s profile page. The page will display the student's name, group name, and assigned GTA at the top. It will also show the grades from each assignment, and the weekly evaluation done by the GTA. The layout of the page is shown in Figure 11 below.

Graphical user interface, text, application, chat or text message, email

Description automatically generated

**Figure 11**

In the Weekly Reports tab, the professor can click on the flag button to send the GTA of that student an email. The button and the pop-up modal are shown below in Figure 12.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

**Figure 12**

At the top of the navigation bar on every page is a mail icon that the professor can click to “View Messages” or the sent flags. Shown in Figure 13 below the professor can see the date that it was sent, to whom it was sent, about whom, what their message was, and if the GTA has read their message.

Graphical user interface, text, application, email

Description automatically generated

**Figure 13**

Figure 14 below shows the “Add Students” page where the professor can upload a file with the students' names and accessID's to a given section. On this page, they can also view the inactive students and students not assigned to a group and delete a student (mark them as inactive) if they drop out of the course.

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

**Figure 14**

Similarly to the admin feature, the professor can authorize GTA’s to the application. The form asks for the first name, last name, and Wayne State University email address. Underneath will show a list of all active and deactivated GTAs in the system that the professor can search for and change their activity status. The professor will also be able to edit the GTAs information (first name, last name, email) if there is an error or change in the professor's name. The page is shown in Figure 15 below.

Graphical user interface, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

**Figure 15**

The professor can also add a section to the application if the administrator did not add one for them. The page is shown in Figure 16 below.

Graphical user interface

Description automatically generated with medium confidence

**Figure 16**

Figure 17 shows the “Add Groups” page where the professor can choose a section and type in a name to create a group. Also on this page is the feature where a professor can assign a student to a group by selecting a group in the dropdown menu and clicking the checkboxes next to the student they would like to assign. They can also delete a student from the application on this page as well.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

**Figure 17**

The “Edit Group” page is shown in Figure 18 below. On this page, professors can edit all their groups in the current semester. The groups are displayed in a table with their group name, assigned GTA, members of the group, and edit, remove, and delete buttons. They can use the search bar to search and find different information.

If the click the edit button, the pop-up modal in Figure 19 will display. Inside, they can change the group name and the assigned GTA of that group. Figure 20 shows the remove pop-up button that illustrates that to remove a student from a group. The professor has to click the checkbox next to their name. If the professor wants to unassign a GTA from a group and have another GTA pick it up of their own choosing, they can click the delete button and will be presented with the pop-up modal in Figure 21.

Graphical user interface, application

Description automatically generated

**Figure 18**

Graphical user interface, application

Description automatically generated

**Figure 19**

Graphical user interface, text, application, chat or text message

Description automatically generated

**Figure 20**

Graphical user interface, text, application, chat or text message

Description automatically generated

**Figure 21**

The “Grade Assignments” page is displayed in Figure 22 below. The page is laid out in three tired tabs, with the sections being the utmost tabs, then the second layer being the groups within those tabs, and the third being the assignments to grade. Once they click on the assignment tab, the form will display. If they have already filled out some information, it will show in the input spaces for the professor to read and edit.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

**Figure 22**

The four documents written in class are graded on an overall grade, consistency, grammar, topics & correctness, and resubmission after notes. Each student also receives a grade on their presentation of the document. For the prototypes and the final presentation, the group is graded on an overall grade, consistency, functionality, topics & correctness, and group status. There is also a section that asks if the group has a client and if the client is satisfied. Each student gets a grade on their presentation of the prototype and their GitHub activity. For each grade, there is a notes area for the professor to give their reasoning.

Figure 23 shows the navigation for the professor, showing they will have access to their “Home” dashboard, “Add GTAs,” “Add Students,” “Add Section,” “Add Groups,” “Edit Groups,” “Grade Assignments,” and the “Sign Out” button.

Graphical user interface, text, application

Description automatically generated

**Figure 23**

3.1.1.3 GTA Interface

If a GTA is signed in, they will be redirected to their dashboard, where they will see the groups that they have been assigned. When they click on the group names, they will be redirected to that group's profile page where they can grade the assignments, similarly to the professor, illustrated in Figure 24.1 and 24.2 below.

Graphical user interface, website

Description automatically generated

**Figure 24.1**

Graphical user interface, text, application, email

Description automatically generated

**Figure 24.2**

Figure 25 below illustrates the “Weekly Reports” page. When the GTA first comes on the page, they will have to choose the group they would like to grade and click the select button to pull up the table. In the table shows “Incomplete” buttons corresponding to each student with how many weeks are in that semester. When the form is filled out the buttons change color corresponding to the status in the legend and show the value of the submitted and status for that student. Figures 26 and 27 show the group pop-up modal and the individual student pop-up modal respectively. The “Jump to Summary” button takes the GTA to the bottom of the page where it totals up the number of different submitted and status values type, illustrated in Figure 28 below.

Graphical user interface, text

Description automatically generated

**Figure 25**

Graphical user interface, application

Description automatically generated

**Figure 26**

Graphical user interface, application

Description automatically generated

**Figure 27**

A picture containing diagram

Description automatically generated

**Figure 28**

On the “Choose Groups” page, the GTA can assign themselves to groups that do not have a GTA assigned to them. They can also view the groups that they are assigned as shown in Figure 29 below.

Graphical user interface

Description automatically generated

**Figure 29**

The “Grade Midterm” and “Grade Final” pages are very similar. As shown in Figure 30 below, the GTA can see his groups and students in tab format and give them a midterm/final grade. Below the form the GTA can view all the past assignments so that they can determine what grade to give that student.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

**Figure 30**

At the top of the navigation bar is a bell icon that will show the number of unread messages from the professor. When the GTA goes to that page by clicking on the bell they will see the page shown in Figure 31 below. The page is split up into two tables, new messages, and old messages. The message will not display until the GTA clicks the “Reveal Message” button and the notification on the bell will remain until the button is clicked, to ensure the message is addressed. Once the button is clicked the next time the GTA visits the page that message will be moved to the “All Messages” table where they can still see the message.

Graphical user interface, application

Description automatically generated

**Figure 31**

The side navigation, shown in Figure 32, will include links to their “Home” dashboard, “Weekly Reports,” “Choose Groups,” “Grade Midterm,” “Grade Final,” and the “Sign Out” button.

Graphical user interface, application

Description automatically generated

**Figure 32**

3.1.2 Hardware Interfaces

The Capstone Course Evaluation System web application does not require the use of any hardware interfaces.

3.1.3 Software Interfaces

The Capstone Course Evaluation System does not utilize any external software interfaces. Everything is currently installed on the application server.

3.1.4 Communications Interfaces

The Capstone Course Evaluation System web application will use HTTP protocol for communication over the internet. Users will request data through a view. These requests will ‘GET’ and ‘POST’ from the database connection “Dbh.class.php” to the database created in MySQL workbench. After the connection is created, a controller will query the database with another object controller class. The value of the requests will output in the html for users to interact with.

*Diagram

Description automatically generated*

***Figure 33***

## 3.2 Functional Requirements

|  |  |
| --- | --- |
| ID | FR1 |
| Title | 3.2.1 User Sign In |
| Users: | All users |
| Priority | High |
| Description | When opening the application, the user will be able to sign in. They will first be prompted to enter an email and then a password. |
| Inputs | The inputs for signing in are an email and a password. |
| Processing | The application will check if the email is in the database after the user clicks “Next.” Then the application will prompt the user for a password, and after they hit “Submit,” the application will search again in the database to check if the password is valid. |
| Outputs | If the user is valid and enters the correct information, they will be directed to the dashboard of their respective role (admin, professor, GTA). |
| Error Handling | Error: Email is required  If the user tries to press next without entering anything, an error message will display, “Email is required.”  Error: Invalid Email  If the user enters an email that is not in a proper email format, an error message will display, “Invalid email.” The form will not be submitted until a valid email is entered.  Error: Email not in database  If the user enters an email that is not in our database, an error message will display, “Sorry that email is not in our database. Please seek out an admin.”  Error: Password is not valid  If the user enters the incorrect password and hits the submit button, the message “Sorry that password did not work. Try Again.” will display. The form will not submit until a valid password is entered. |
| Dependencies | FR2 |

|  |  |
| --- | --- |
| ID | FR2 |
| Title | 3.2.2 User Sign Out |
| User | All users |
| Priority | High |
| Description | The user should be able to sign out of their account. The user can access the sign out button by going to the top left corner of the screen and opening the navigation. The user will be able to access this button on every screen after signing in. |
| Inputs | The input for signing out is the user clicking the “Sign Out” button. |
| Processing | The system will sign the user out of the application, so they will no longer be able to access their accounts and see and edit their data until they sign in again. |
| Outputs | When the system signs out a user, it will redirect the user to the “Sign In” page |
| Error Handling | N/A |
| Dependencies | FR1 |

|  |  |
| --- | --- |
| ID | FR3 |
| Title | 3.2.3 User Set Password |
| User | All users |
| Priority | High |
| Description | If it is the users first time signing into the application, they will have to create a password. Once the user enters their email and hits next, they will be redirected to a new page where they can “Set [their] Password”. Once the user enters their email and clicks submit, they will receive an email with a link to set their password. Once they relocate to that link, they will enter their new password and confirmation. When done, they can now sign in with their password. |
| Inputs | The inputs for the user creating a password are the email connected with the application, clicking the link sent to their email, a password, and a password confirmation |
| Processing | The system will take the password and set it in the database. |
| Outputs | When the user enters an email and click submit the page will reload with the message, “Check your email!” When the user enters in a password and a confirmation, the page will reload to the sign in page for the user to sign into the application. |
| Error Handling | Error: Email not in Database  When the user is taken to the set email page, they must enter the email that is already in the database (their Wayne State University email), otherwise the error message will display “Sorry, please enter the email connected with your account.”  Error: Confirmation password does not match  If it is the users first time signing into the application, they will be asked to create a password and confirm it. If the user types in two different passwords, the application will display the error message “Sorry those two passwords do not match, please try again.” The application will keep asking for a password and a confirmation until they both match. |
| Dependencies | FR1 |

|  |  |
| --- | --- |
| ID | FR4 |
| Title | 3.2.4 User Forgot Password |
| User | All users |
| Priority | High |
| Description | The user should be able to reset their password if they forgot it. When they get to the screen to enter their password, there is a “Forgot password?” link that will take them to a new page where can enter their email address. Once the user clicks submit, they will receive an email with a link to reset their password. Once they relocate to that link, they will enter their new password and confirmation. When done, they can now sign in with their new password. |
| Inputs | The inputs for the user forgot password feature are click the forgot password link, entering an email, clicking the link sent to their email, a password, and a password confirmation |
| Processing | The system will take the new password and replace the old one in the database. |
| Outputs | When the user enters an email and click submit the page will reload with the message, “Check your email!” When the user enters in a password and a confirmation, the page will reload to the sign in page for the user to sign into the application. |
| Error Handling | Error: Confirmation password does not match  If it is the users first time signing into the application, they will be asked to create a password and confirm it. If the user types in two different passwords, the application will display the error message “Sorry those two passwords do not match, please try again.” The application will keep asking for a password and a confirmation until they both match. |
| Dependencies | FR1 |

|  |  |
| --- | --- |
| ID | FR5 |
| Title | 3.2.5 Create Semester |
| User | Admin |
| Priority | High |
| Description | The admin should be able to create a new semester by going to the ‘Add Semester/Section’ link in the navigation tab. On the ‘Add Semester/Section’ page, the admin will see two forms, one for creating a semester and one for creating a section. The “Semester” form asks for the name of the semester and the start and end date. The admin will click the “Add” button to submit. |
| Inputs | The admin will enter a semester name, a start date, and an end date. |
| Processing | When the user selects submit, the application will add the semester to the application. |
| Outputs | If the admin enters valid information, the page will display the message “Semester added successfully.” |
| Error Handling | Error: Semester already created  The system will not allow an admin to create a semester that’s name is already taken and with the same start dates. |
| Dependencies | FR1 |

|  |  |
| --- | --- |
| ID | FR6 |
| Title | 3.2.6 Create Section |
| User | Admin |
| Priority | High |
| Description | The admin should be able to create sections for a professor within a specific semester by navigating to the “Add Semester/Section” page on the navigation tab. After they fill out the form, they will click the “Add” button to create the section. |
| Inputs | The inputs for creating a section are a section name, a professor, and a semester. |
| Processing | After clicking add, the system will add the section to the database. |
| Outputs | If the admin enters valid information, the page will display the message “Section added successfully.” |
| Error Handling | Error: Section already created  The system will not allow an admin to create a section that’s name is already taken. |
| Dependencies | FR1, FR5 |

|  |  |
| --- | --- |
| ID | FR7 |
| Title | 3.2.7 Add Professors |
| User | Admin |
| Priority | High |
| Description | The admin should be able to add professors to the application by locating the “Add Professors” link on the navigation bar. On the “Add Professors” page, the admin can fill out a form to add professors and view active professors in the application. |
| Inputs | The inputs for creating adding a professor are first name, last name, and email. |
| Processing | After clicking add, the system will add the professor to the database. |
| Outputs | If the admin enters valid information, the page will display the message “Professor added successfully.” |
| Error Handling | Error: Professor already in the system  The system will not allow an admin to add a professor to the application if their email is already in our database.  Error: Email is not a Wayne State University Email  The system will not allow an admin to add a professor to the application if the email entered is not in the Wayne State University accessID format followed by the “@wayne.edu.” domain. |
| Dependencies | FR5 |

|  |  |
| --- | --- |
| ID | FR8 |
| Title | 3.2.8 Edit Professors |
| User | Admin |
| Priority | Medium |
| Description | The admin should be able to edit professors in the application by going to the “Add Professors” link on the navigation bar. This page will list the professors that are active in the application. Next to the professor’s name will be an “Edit” button. When the admin clicks the edit button, a form will pop up for the admin to update the information. After filling out the form, they will hit “Update” to save the edits. |
| Inputs | The admin will click the button “Edit” and change the values they want for the professor. |
| Processing | The system will update the professor information. |
| Outputs | If the professor enters valid information, the page will display the message, “Professor updated successfully.” |
| Error Handling | Error: Blank inputs  When updating a professor, the admin must not leave any input fields blank. If any are not filled out, an error message will display with the message, “All fields are required,” and will not submit. |
| Dependencies | FR7 |

|  |  |
| --- | --- |
| ID | FR9 |
| Title | 3.2.9 Delete Professors |
| User | Admin |
| Priority | Medium |
| Description | The admin should be able to delete professors in the application by going to the “Add Professors” link on the navigation bar. This page will list the professors that are active in the application. Next to the professor’s name will be a delete button. When the admin clicks the delete button, a popup will ask for confirmation of the deletion and delete the professor if the admin clicks “Verify.” |
| Inputs | Clicking the “Delete” button and then clicking “Verify.” |
| Processing | The system will not remove the professor from the database but mark them as inactive so that admins can still view that user’s history. That user can no longer have access to the application. |
| Outputs | After confirming, the page will display the message, “Professor deleted successfully.” |
| Error Handling | N/A |
| Dependencies | FR7 |

|  |  |
| --- | --- |
| ID | FR10 |
| Title | 3.2.10 Add Section |
| User | Professor |
| Priority | High |
| Description | The professor should be able to add a section to the current semester by going to the “Add Section” page from the navigation bar. On this page the form will ask for the section name and the professor to click “Add” for it to work successfully. |
| Inputs | The inputs creating a section by a professor is only a section name. |
| Processing | After clicking add, the system will add the section to the database. |
| Outputs | If the professor enters valid information, the page will display the message “Section added successfully.” |
| Error Handling | Error: Section already in the semester  The system will not allow a professor to create a section with a name that has already been created and added to that semester. |
| Dependencies | FR1, FR5 |

|  |  |
| --- | --- |
| ID | FR11 |
| Title | 3.2.11 Add GTAs |
| User | Professor |
| Priority | High |
| Description | The professor should be able to add GTAs to the current semester by going to the “Add GTAs” page from the navigation bar. On this page, the form will ask for the GTAs first name, last name, email address, and what section they will be assigned. This page will also list the GTAs that are working for that professor. |
| Inputs | The inputs for creating and adding a GTA are first name, last name, and email. |
| Processing | After clicking add, the system will add the GTA to the database. |
| Outputs | If the professor enters valid information, the page will display the message “GTA added successfully.” |
| Error Handling | Error: GTA is already in the system  The system will not allow a professor to add a GTA to the application if their email is already in our database.  Error: Email is not a Wayne State University Email  The system will not allow an admin to add a professor to the application if the email entered is not in the Wayne State University accessID format followed by the “@wayne.edu.” domain. |
| Dependencies | FR10 |

|  |  |
| --- | --- |
| ID | FR12 |
| Title | 3.2.12 Edit GTAs |
| User | Professor |
| Priority | Medium |
| Description | The professor should be able to edit GTAs by going to the “Add GTAs” page from the navigation bar. This page will list the GTAs that are working for that professor. Next to the GTA name will be an edit button. When the professor clicks the edit button, a form will pop up for the professor to update the information. After filling out the form, the professor will hit “Update” to save the edits. |
| Inputs | The professor will click the button “Edit.” |
| Processing | The system will update the GTA information. |
| Outputs | If the professor enters valid information, the page will display the message “GTA updated successfully.” |
| Error Handling | Error: Blank inputs  When updating a GTA, the professor must not leave any input fields blank. If any are not filled out, an error message will display with the message, “All fields are required,” and will not submit. |
| Dependencies | FR11 |

|  |  |
| --- | --- |
| ID | FR13 |
| Title | 3.2.13 Delete GTAs |
| User | Professor |
| Priority | Medium |
| Description | The professor should be able to delete GTAs by going to the “Add GTAs” page from the navigation bar. This page will list the GTAs that are working for that professor. Next to the GTA name will be a delete button. When the professor clicks the delete button, a popup will ask for confirmation of the deletion and will delete the GTA if the professor clicks “Verify.” |
| Inputs | Clicking the “Delete” button and then clicking “Verify.” |
| Processing | The system will not remove the GTA from the database but mark them as inactive, so their history can still be viewed. |
| Outputs | After confirming, the page will display the message, “GTA deleted successfully.” |
| Error Handling | N/A |
| Dependencies | FR11 |

|  |  |
| --- | --- |
| ID | FR14 |
| Title | 3.2.14 Add Students |
| User | Professor |
| Priority | High |
| Description | The professor should be able to add students to the current semester by going to the “Add Students” page from the navigation bar. On this page, the professor will be able to upload a spreadsheet file with 2 columns. The left is for access IDs and the right is for the student’s names. The professor will also have a drop-down menu of sections to choose from to upload the students. |
| Inputs | The inputs for adding students are a file upload and the section. The file type which can be uploaded is (.csv, .ods, .xlsx, .xls). |
| Processing | After clicking upload, the system will add the students to the database. |
| Outputs | If the professor uploads a valid file, the page will display the message “Students added successfully.” |
| Error Handling | The website will check if the file uploaded is of a proper file type (.csv, .ods, .xlsx, .xls). It will also check if the access ID is in the proper format, and if not, the website will ask you to reupload the file. The system will also check if access IDs are already in the database to stop duplicates. |
| Dependencies | FR 6, FR7, FR10 |

|  |  |
| --- | --- |
| ID | FR15 |
| Title | 3.2.15 Delete Student |
| User | Professor |
| Priority | Low |
| Description | The professor should be able to delete a student by going to the “Add Students” page from the navigation bar. This page will list the students in each section. Next to the student’s name will be a delete button. When the professor clicks the delete button, a popup will ask for confirmation of the deletion and will delete the student if the professor clicks “Verify.” |
| Inputs | Clicking the “Delete” button and then clicking “Verify.” |
| Processing | The system will not remove the student from the database but mark them as inactive, so their history can still be viewed. |
| Outputs | After confirming, the page will display the message, “Student deleted successfully.” |
| Error Handling | N/A |
| Dependencies | FR14 |

|  |  |
| --- | --- |
| ID | FR16 |
| Title | 3.2.16 Create Groups |
| User | Professor |
| Priority | High |
| Description | The professor should be able to create groups and add students to those groups by going to the “Add Groups” page from the navigation bar. The page will be in have the sections in tab format. The professor will click on the section tab they would like to manipulate. The professor will click the “Add” button to pop up a form that will take a name and a multi-select dropdown menu of the students in that section that are not already assigned to a group. |
| Inputs | The inputs for creating a group are a name and students’ selection. |
| Processing | After clicking add, the system will add the group to the application. |
| Outputs | If the professor enters valid information, the page will display the message “Group added successfully.” The professor will also see the group added the list of all groups. |
| Error Handling | Error: Group is already in the system  The system will not allow a professor to add a group to the application if their group name is already in the database. |
| Dependencies | FR14 |

|  |  |
| --- | --- |
| ID | FR17 |
| Title | 3.2.17 Add Student to a Group |
| User | Professor |
| Priority | Medium |
| Description | The professor should be able to add a student to a group on the “Add Groups” page, located in the navigation bar. The page will display the list of groups already created, and the professor should be able to click the checkbox next to the student’s name to assign them a group. |
| Inputs | The professor will open the drop-down menu and click on the group they would like. Then the professor can click on the checkbox next to the students name to indicate that they want that student in the group they choose above. After making all the changes they would like, the professor will hit the “Assign Students” button. |
| Processing | The system will now add the group ID to that student so that they belong to that group. |
| Outputs | The page will reload with a success message. |
| Error Handling | N/A |
| Dependencies | FR16 |

|  |  |
| --- | --- |
| ID | FR18 |
| Title | 3.2.18 Edit Groups |
| User | Professor |
| Priority | Medium |
| Description | The professor should be able to edit groups on the “Edit Groups” page, located in the navigation bar. The page will display the list of groups already created, and the professor should be able to edit the groups as they please. |
| Inputs | The professor will click on the edit button located next to the group name. The professor can edit the group name and add/delete students from that group. After making all the changes they would like, the professor will hit the “Update” button. |
| Processing | The system will change the values and update the database. |
| Outputs | The page will reload with the newly updated group. |
| Error Handling | N/A |
| Dependencies | FR16 |

|  |  |
| --- | --- |
| ID | FR19 |
| Title | 3.2.19 Give Student a Grade |
| User | Professor |
| Priority | Medium |
| Description | The professor should be able to grade students. On the “Grade Assignments” page found in the navigation bar, the sections will be in tab format, and the groups in those sections will be within those tabs. The professor will click on the third layer of tabs with the name of the assignment that he would like to grade, and a form will appear. The professor will fill out the form, grading the group and the students in those groups and click the “Submit Form” button at the top to save the grade. |
| Inputs | The inputs for a professor grading a student are a grade, an evaluation, and clicking the “Submit” button. |
| Processing | The system will add the grade and evaluation in the database for the student and mark that it was written by the professor so that it only displays in the professor’s view, not the assigned GTA for that student. |
| Outputs | The page will reload with the message “Grade added.” |
| Error Handling | N/A |
| Dependencies | FR14, FR16 |

|  |  |
| --- | --- |
| ID | FR20 |
| Title | 3.2.20 Pick Groups |
| User | GTA |
| Priority | High |
| Description | The GTA can go to the “Choose Groups” page to assign themselves to a group. There will be a table at the top of the page with the groups that do not have a GTA assigned to them. The GTA can click on the checkbox next to the groups name and click the “Assign” button to assign themselves to that group. |
| Inputs | The GTA will select the group name and click the “Submit” button. |
| Processing | The system will update the application so that GTA is assigned to that group. |
| Outputs | The page will reload, and the group name will appear under “Your groups:” and will also be on the dashboard, weekly reports page, grade midterm page, and grade final page for the GTA to grade |
| Error Handling | N/A |
| Dependencies | FR16 |

|  |  |
| --- | --- |
| ID | FR21 |
| Title | 3.2.21 View Groups |
| User | GTA |
| Priority | High |
| Description | GTAs should have the ability to view the groups by clicking on the group names under the “Your Group” section on their dashboard. The GTA will be reloaded to the group page, where they can click on the students in the group to view their overall stats and grades. |
| Inputs | The GTA will click on a group name. |
| Processing | The system will display the request. |
| Outputs | The GTA will be redirected to the selected groups “Group Page,” which display that groups information and the side navigation for the GTA to grade that group. |
| Error Handling | N/A |
| Dependencies | FR20 |

|  |  |
| --- | --- |
| ID | FR22 |
| Title | 3.2.22 Grade Weekly Reports |
| User | GTA |
| Priority | High |
| Description | GTAs should have the ability to view and edit weekly reports for a specific student. The GTA can click the “Weekly Reports” link on the navigation to be redirected to the Weekly Reports page. After the GTA selects which group they would like to view, the group member names will display horizontally in a table, with the corresponding weeks vertically. Circular buttons will display either green with the word “Completed” across it if the GTA has filled out the weekly form or red with the word “Incomplete” across it if the GTA has not filled out the weekly form. When the GTA clicks on the button, a form will pop up for them to input when it was submitted, the status and the written evaluation that will be saved with a submit button. |
| Inputs | The GTA selects which group to view, then click the circular buttons to fill out the grading form. |
| Processing | The system will add the evaluation to the database |
| Outputs | The page will reload and the circular button under the student’s name for that week will now show green and display the message “Weekly report accessed!” |
| Error Handling | Error: All inputs are required  If the GTA tries to fill out the evaluation by leaving any of the value’s blank, an error message will display, “All fields in the form are required,” and will not submit until all fields have been inputted. |
| Dependencies | FR20, FR21 |

|  |  |
| --- | --- |
| ID | FR23 |
| Title | 3.2.23 Grade Assignments |
| User | GTA |
| Priority | High |
| Description | On the GTA dashboard when the GTA clicks on a given group they will be taken to that Groups Page. All of the assignments will display in tab format and when the GTA clicks on one of the tabs the grading form will appear. The GTA can fill out the grades for that group and for the students in that group and click the "Submit" button to save the grade. |
| Inputs | The inputs for grading filling out the forms for the grades and the notes section for each. |
| Processing | If all parts of the form are filled out, the grade will be added to the database. |
| Outputs | The page will reload with the message “Grade added for student successfully.” |
| Error Handling | N/A |
| Dependencies | FR20 |

3.3 Non-Functional Requirements

|  |  |
| --- | --- |
| ID | 3.3.1 Performance |
| Title | NFR 1: Database Performance |
| Description | The database must be able to display any information from the database on the screen in 5 seconds or less. |

|  |  |
| --- | --- |
| ID | 3.3.2 Reliability |
| Title | NFR 2: Error handling for all (input and data) |
| Description | The application should be able to handle errors based on the functional requirements stated above. |

|  |  |
| --- | --- |
| ID | 3.3.3 Availability |
| Title | NFR 3: Time for maintenance |
| Description | The application should be accessible 365 days a year, 24/7. If bugs were to occur in the application, then the application would be down for a short period to fix the issues. Besides that ever happening, the website will be available all of the time. |

|  |  |
| --- | --- |
| ID | 3.3.4 Security |
| Title | NFR 4: Security of all data |
| Description | The data will be stored in MySQL database which will be properly secured for all users (Admin, Professors and GTA’s). |

|  |  |
| --- | --- |
| ID | 3.3.5 Maintainability |
| Title | NFR 5: Updating Functionalities |
| Description | To maintain our code, we decided to use the framework called MVC. There are three layers to this method, the first one is M which stands for model and this layer manages the data, logic, and rules of the application. The V layer stands for view, this layer represents all visuals such as charts, diagrams, and tables. Finally, the C stands for controller, the controller is controlling the specific process of the application. |

|  |  |
| --- | --- |
| ID | 3.3.6 Portability |
| Title | NFR 6: Web Application Compatibility |
| Description | The application should be able to run on all listed web browsers in section 2.5 (Google Chrome, Microsoft Edge, Opera, Mozilla Firefox, or Safari). The application should also run on devices supported by web browsers, for example, computers, smartphones, and tablets. |

|  |  |
| --- | --- |
| ID | 3.3.7 Usability |
| Title | NFR 7: Responsiveness of Application |
| Description | The application should have a flexible and easy layout that can adapt to different screen sizes, but also be user-friendly. To elaborate more on the responsiveness, the application will be usable on different devices such as tablets, phones and desktops. For example, the GTA decides to open the dashboard on his/her phone and wants to enter in a grade for a student; the grading table will be shrunk to where the GTA can see the table on their phone. To conclude, any devices that has a screen will be usable for this application. |

3.4 Design Constraints  
The following technologies will be used to build the web application: PHP, MySQL, Apache Server, HTML, CSS, and JavaScript.

### 3.4.1 Database Constraints

#### 3.4.1.1 Table Column Count

MySQL 8 has a limitation of 4096 columns for each table.

#### 3.4.1.2 Row Size

MySQL 8 has a limit of 65,535 bytes for a row.

#### 3.4.1.3 Joins

MySQL 8 only allows 61 tables in a single join.

#### 3.4.1.4 Number of Tables/Databases

MySQL 8 has no set limitation on the number of tables or databases however depending on the operating system file limit, there can be a limitation.

3.5 Logical Database Requirements

This application requires a hierarchy starting with admin, then professor, and ending with GTA. To ensure the security of the sensitive grading data contained in the database, the administration is required to add professor data to the database and delineate which data each will be able to access. After being added, a professor can sign in via an administration-provided password and can add their respective GTAs to the database. Authorized GTAs can access the website and all data required for student evaluations. The hierarchy model thereby protects both the security of the data by preventing unauthorized parties from viewing other students’ data, and the integrity of the data by tracking the users who are able to input or edit the database.

Our model-driven architecture will format the CCES data into structured rows and columns. Where the columns represent a set of data values while the rows will populate the data. Each column has a defined field type. These fields include integers, varchar, double, date type, and Boolean values. The data is stored in linked tables to enable modularity, consistency, and security. Allowing for the addition of a new row for each semester prevents the need to delete old data and prevents possible inconsistencies, ensuring modularity and a robust, future-proofed system. The hierarchical flow of data edit authorization ensures consistency by only allowing specific users, whose records are stored in the ‘user’ table, to edit data in other appropriate, authorized tables. Security is likewise ensured, because not only are edit privileges restricted, but view privileges are as well. The linked tables allow the data to be divided and accessed easily without creating unnecessary complexity.

Additionally, the format allows for the manipulation of data throughout the database. Sort, filter, and search functionalities will be present wherever the data exists. Since the database needs to be updated every semester with new data such as new courses, new professors, and new GTAs, data integrity will be an important factor in the design consideration.

Data integrity has been ensured and supported by several protocols. First among them is the use of drop box data selection wherever appropriate. This input style prevents inconsistent or inappropriate data from being input into the database. Anywhere drop boxes do not apply (Names, titles, notes, written evaluations, etc.) data validation protocols are in place to constrain data entry to the correct length, characters, and format. An error message will alert a user to the incongruity of their input if necessary. Access controls are also in place to prevent unauthorized users from editing or viewing data. An audit log will also allow any users with view privileges to see which users have input or edited data. Physical integrity and data backups will depend on the end-users chosen host location. MySQL Server uses a pluggable storage engine architecture that enables storage manipulation, so the only limit to the amount of storage available is the physical limits enacted by the hosting location chosen by the end user.

Retention of the following data elements are necessary for the Capstone Course Evaluation System

* **Users**: First name, last name, email, password, role of the user, and their status in the system.
* **Semester**: Semester name, start date, and end date.
* **Section**: Section name, semester, and professor.
* **Groups**: Group name and section.
* **GTA Assignment**: GTA, section, and group.
* **Students**: Student name, section, and accessID.
* **Weekly Reports:** The Week, date GTA graded, student submission time, status, and evaluation.
* **Grades**: assignment name, date graded, given grade, user who graded it, and evaluation.

For logical requirements, out client has asked for use to deliver our hierarchy through administration within the database. First the administration is required to add all professor data into the database. Once a professor has signed in, they then can add their respective GTAs’ to the database. Only then will a GTA be allowed to access the website. Additionally, the client requires the manipulation of data throughout the database. Sort, filter, and search functionalities will be present wherever the data exists.

# Analysis Models

4.1 Data Flow Diagrams (DFD)

The data flow diagram is categorized from the most basic level, “Level 0”, to the more complex level, “Level 2.” “Level 0” will give a basic understanding of the user data flow. This is the basics of how the application works. “Level 1” is still a basic overview of how the data flows but starts to provide a more in-depth idea of how that information is moving. In the “level 2” data flow diagram, the process has been broken down into sub-processes. Reading and knowing the diagrams will provide a good understanding of the information flow provided in the Capstone Course Evaluation system.

### 4.1.1 DFD Level 1

The data flow diagram “Level 1” gives a greater understanding of how the data flows in its sub-processes and is stored in MySQL. This diagram will start to show the hierarchy of authentication within the web application, with admin at the top, then professors, and lastly GTAs. The admin creates semesters and adds professors. Because they have the highest authority, they can see all grades and view all reports that have ever been written. The professor can only see the current semester that is happening. Within the semester, the professor can add GTAs, sections, and students and create groups. The professor can view all the grades and reports that the GTAs write. The GTA will only be able to see, grade, and write reports about the groups and students that they are assigned.

Diagram

Description automatically generated

***Figure 34***

### 4.1.2 DFD Level 2

The data flow diagram "Level 2" goes another step deeper into parts of "Level 1." This diagram splits into three different sections based on user hierarchy. The first section gives an in-depth take on how the data flows in the admin interface. The second section will dive deeper into the professor interface of the program, exploring just how a professor can use their interface and manipulate data. Finally, the last section of the level 2 diagram is the GTA interface. Reviewing all three parts of the Capstone Course Evaluation Data Flow Diagram Level 2 Diagram will give an excellent understanding of the functionalities of the system.

### 4.1.2.1 DFD Level 2 – Admin

The admin will sign into the Capstone Course Evaluation System using a specific username and password. Once signed in, the admin will view the admin interface, then they will navigate to a menu with three functions: “Add Professors,” “Add Semester,” and “Add Section” page using a menu on the left side of the screen. Admin must add a professor and semester first. Once a professor is added, they can sign into the application. The admin can choose to create a section if they want, however, it requires the input of a semester and a professor.

Diagram

Description automatically generated

***Figure 35***

### 4.1.2.2 DFD Level 2 – Professor

Once the admin adds a professor to the Capstone Course Evaluation System, the professor can now view their own dashboard. The dashboard contains each section the professor creates in a tab format. On the left-hand corner, a navigation menu has four options: “Add Students,” “Add GTA,” “Add Section,” and “Modify Groups.” On the dashboard, the professor can navigate between sections to see its data, such as students, groups, and GTAs. This professor will also see the grades and reports completed by the GTA.

Diagram

Description automatically generated

***Figure 36***

### 4.1.2.3 DFD Level 2 – GTA

Once a professor adds a GTA, the GTA can sign in and view their dashboard. This dashboard contains the groups they are assigned. If there are groups that do not have any assigned GTAs, a drop-down will appear on the dashboard where the GTAs can assign themselves to those groups if they would like. The GTA can select which group they would like to evaluate and review. Once a GTA selects a group from their dashboard, they can evaluate the group and an individual student.

Diagram

Description automatically generated

***Figure 37***