tEAM pROJECT Grade System: RINO SYSTEMS

Ivan Olivas & Jeronimo Reyes

december 1, 2021

csc 440: Applied software engineering

Table of Tables

1. Table of Contents 3
2. Table of Figures 4

Table of Contents

[Table of Tables 1](#_Toc89343007)

[Table of Figures 3](#_Toc89343008)

[I. Introduction 4](#_Toc89343009)

[A. Problem Statement 4](#_Toc89343010)

[B. System Proposal 4](#_Toc89343011)

[II. System Description 4](#_Toc89343012)

[III. System Requirements 4](#_Toc89343013)

[A. Functional Requirements 4](#_Toc89343014)

[B. Non-functional Requirements: 18](#_Toc89343015)

[IV. Use Case Diagram 19](#_Toc89343016)

[IV. Class Diagram 20](#_Toc89343017)

[V. Sequence Diagrams 21](#_Toc89343020)

[VI. Activity Diagrams 28](#_Toc89343021)

[VII. State Diagrams 34](#_Toc89343022)

[VIII. Database Design 38](#_Toc89343023)

[A. ER Diagram 38](#_Toc89343024)

[B. Table Schema 39](#_Toc89343025)

[IX. Conclusion 39](#_Toc89343026)

[X. Data Dictionary 40](#_Toc89343027)

Table of Figures

[Figure 1. Select file button 5](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343496)

[Figure 2. Select File to upload 5](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343497)

[Figure 3. Display file path selected 6](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343498)

[Figure 4. Successful message for uploading data to database 6](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343499)

[Figure 5. Home Menu 7](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343500)

[Figure 6. StudentID Textbox 7](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343501)

[Figure 7. Student Not Found 8](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343502)

[Figure 8. Display CRN textbox 8](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343503)

[Figure 9. Enter a CRN number message 9](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343504)

[Figure 10. Form enabled to insert course and grade 9](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343505)

[Figure 11 - GUI: Only grade textbox enabled 10](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343506)

[Figure 12. All Fields must be filled out form label 10](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343507)

[Figure 13. Process Successfully done message 11](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343508)

[Figure 14. Display list of courses student has taken 11](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343509)

[Figure 15. Display Edit/Delete Buttons 12](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343510)

[Figure 16 . Dialog box for updated grade 12](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343511)

[Figure 17. Listview update after edit grade use case 13](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343512)

[Figure 18. Course deleted successfully message and updated list 14](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343513)

[Figure 19. Enter studentID to print report 14](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343514)

[Figure 20. Must enter a studentID number for printing a report 15](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343515)

[Figure 21. Student not in database when printing report 15](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343516)

[Figure 22. Transcript successfully created message 16](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343517)

[Figure 23. Log-In Menu 16](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343518)

[Figure 24. Fields missing when logging in 17](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343519)

[Figure 25. Invalid login combination 17](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343520)

[Figure 26. Main Menu Panel 18](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343521)

[Figure 27 – Use case diagram for Grade system 19](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343522)

[Figure 28 - Class Diagram 20](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343523)

[Figure 29 - Sequence Diagram for signing into the grade system 21](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343524)

[Figure 30 - Sequence Diagram for signing out of the system 22](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343525)

[Figure 31 - Sequence Diagram for adding a new entry to the grade system 23](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343526)

[Figure 32 - Sequence Diagram for editing a grade 24](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343527)

[Figure 33 - Sequence Diagram for deleting a grade 25](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343528)

[Figure 34 - Sequence Diagram for importing data 26](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343529)

[Figure 35 - Sequence diagram for print report 27](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343530)

[Figure 36 - Activity Diagram for signing in 28](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343531)

[Figure 37 - Activity Diagram for signing out 29](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343532)

[Figure 38 – Activity diagram for adding a new entry 30](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343533)

[Figure 39 - Activity diagram for editing a grade and deleting a course 31](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343534)

[Figure 40 - Activity Diagram for importing data 32](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343535)

[Figure 41 - Activity Diagram for printing report 33](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343536)

[Figure 42 - State diagram for signing in 34](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343537)

[Figure 43 - State diagram for signing out 34](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343538)

[Figure 44 - State diagram for adding a new entry 35](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343539)

[Figure 45 - State diagram for editing a grade or deleting a course 36](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343540)

[Figure 46 - State diagram for importing data to the system 37](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343541)

[Figure 47 - State diagram for printing a student report 37](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343542)

[Figure 48 - ER Diagram for structuring the database 38](file:////Users/ivanolivas/Desktop/CSC%20440/OCT_TeamReport_Jeronimo_Ivan.docx#_Toc89343543)

# Introduction

## Problem Statement

EKU’s Registrar Office needs to update student course records after receiving student grades from all of the faculty at the end of each semester. The office is asking for us to develop a system to manage the data for them. They want to be able to perform different operations with the data and use it to generate reports of student’s grades.

## System Proposal

We propose a software system to help the Registrar office with the desired functionality of the system. A software system would be the best approach to provide all the needs of the client.

# System Description

Each faculty member provides the office an Excel file with letter grades (A, B, C, D, or F) of the students who took course(s) with them. These excel files will be used to import the grades to the database. The software system will be able to do the following tasks:

1. Add new grades of courses for each student to a database
2. Editing, changing, or adding a grade to the database
3. Print a report card (or transcript) for a student
4. Logging In and Logging Out of system for security purposes

# System Requirements

## Functional Requirements

1. The system shall allow the staff to import new grades of a course for each student to the database.
   1. The staff member shall click the “Import Data” button from the home menu.
   2. The system shall open a new window.
   3. The system shall display a button, saying, “Select File”, for the user to browse the computer for an excel file. (Figure 1)
   4. The system shall display button, saying, “Upload Data”, for the user to insert the file data selected to the database (Figure 1)



Figure . Select file button

* 1. The staff member shall click the “Select File” button and select an excel file to upload. (Figure 2)

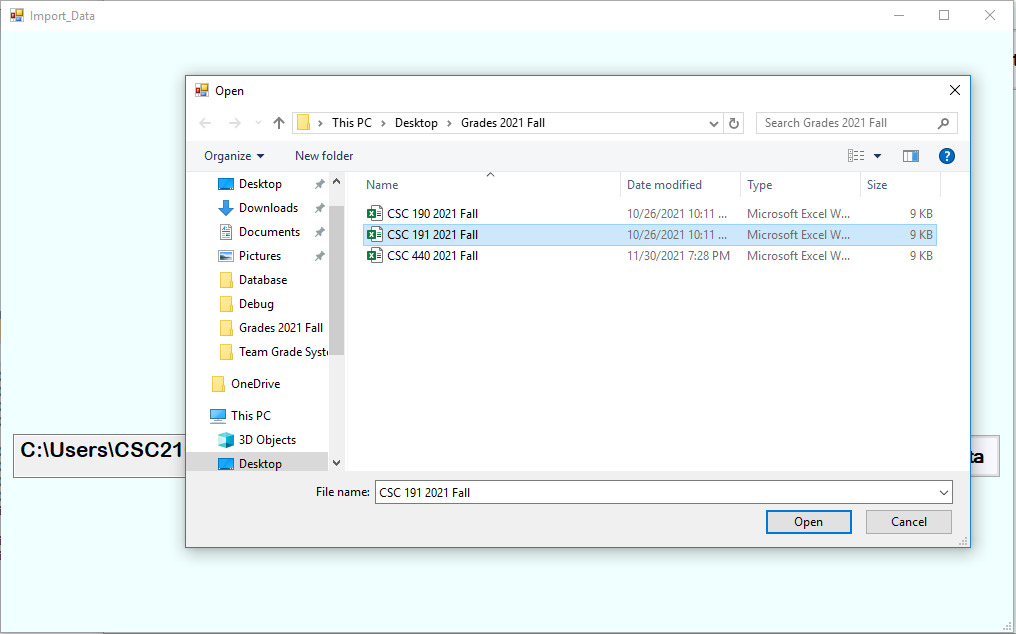


Figure . Select File to upload

* 1. They system shall read the excel file selected and extract the information to store.
  2. The system shall display the path of the file selected. (Figure 3)

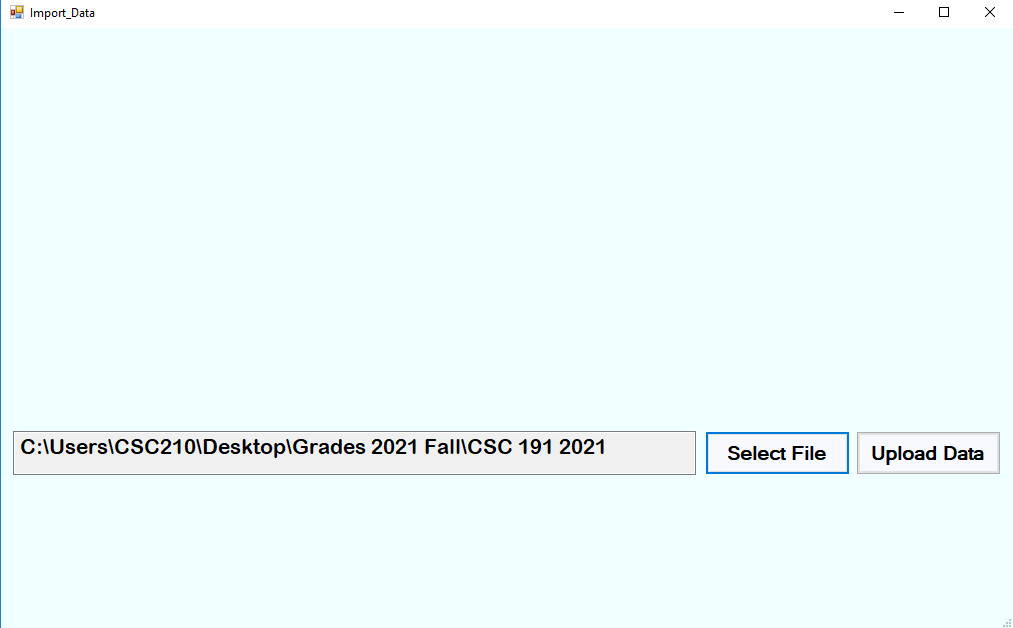


Figure . Display file path selected

* 1. The staff member shall click on the “Upload Data” button in order to insert the data in the file to the database.
  2. The system shall connect to the database.
  3. The system shall parse the information and insert it into the database.
  4. The system shall close the connection to the database.
  5. The system shall display a message saying, “Student’s data was inserted to database successfully!”, with an “OK” button. (Figure 4)

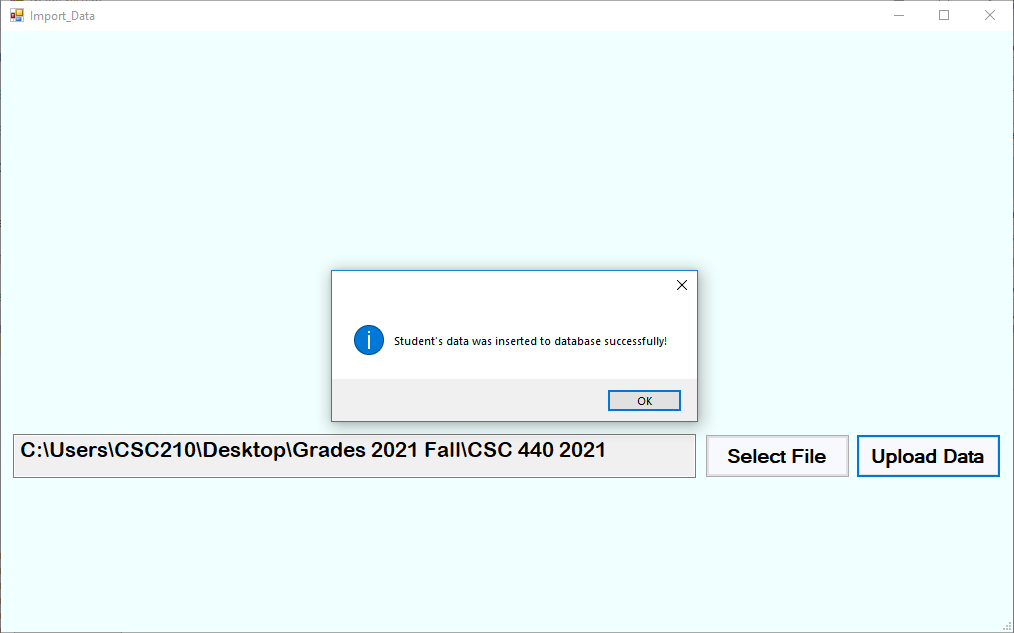


Figure . Successful message for uploading data to database

* + 1. The staff member shall press the “OK” button to return to the home menu. (Figure 5)



Figure . Home Menu

1. The staff member shall select “Add New Entry” button
   1. The system shall display a textbox to input the studentID number, along with buttons to search for the student or to go back to the main menu. (Figure 6)



Figure 6. StudentID Textbox

* 1. The staff member shall input the studentID of an existing student.
     1. If studentID entered does not exist in the database, there will be a message displayed saying, “Student not found”. (Figure 7)
     2. If studentID is found, continue to step 2.3.



Figure 7. Student Not Found

* 1. The system shall display an empty form to the staff member to fill out. (Figure 8)
     1. The form displayed will only have the CRN textbox enabled.

Graphical user interface

Description automatically generated

Figure 8. Display CRN textbox

* 1. The staff member will input the CRN number.
     1. If no CRN is entered, then an error message will display. (Figure 9)



Figure . Enter a CRN number message

* 1. The staff member clicks on the “Check” button.
  2. The system will check to see if that CRN number corresponds to a course in the Catalog table.
     1. If the course does not exist, the all the textboxes in the form will be enabled. (Figure 10)
        1. The system will make the check button become a submit button.



Figure . Form enabled to insert course and grade

* + 1. If the course already exists in the database, only the grade textbox will be enabled. (Figure 11)
       1. The system will make the check button become a submit button.



Figure 11 . Only grade textbox enabled

* 1. The system will check if the required textboxes are filled with data.
     1. If there are missing fields in the form, the system will convey a message to the staff member, “ERROR, All Textboxes must be filled out!”

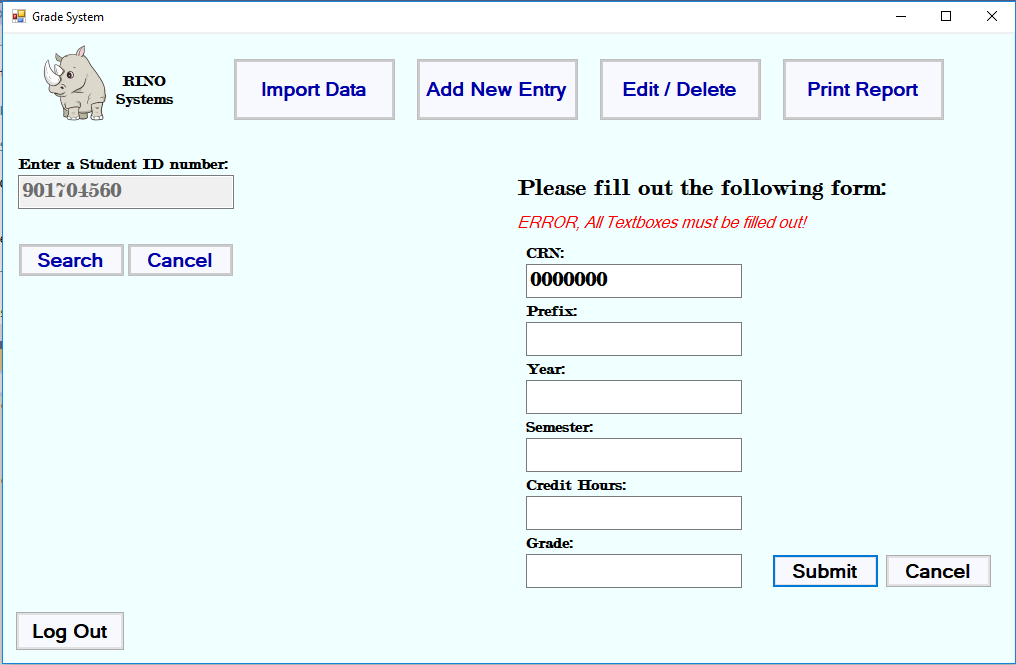


Figure . All Fields must be filled out form label

* 1. The staff member shall click the submit button.
  2. The system shall connect to the central database.
  3. The system shall store the information to the database.
  4. The system shall display a successful message to the staff member. (Figure 13)



Figure . Process Successfully done message

* 1. The staff member shall confirm the message.
  2. The system shall close the connection to the database.

1. The system shall allow the staff to edit grades of each student for a course in the database
   1. The staff member shall click the “Edit/Delete Entry” button.
   2. The system shall prompt the staff member to enter a student ID number. (Figure 6)
   3. The staff member shall enter a student ID number.
   4. The staff member shall click “Search” button.
      1. The system shall validate the student ID.
         1. The system shall display a message “Student not found” if the student ID is invalid. (Figure 7)
   5. The system shall retrieve the courses of the specified studentID.
   6. The system shall allow the staff member to select a course from the list view. (Figure 14)
   7. The system shall display a message saying, “Please select a CRN from the list.”

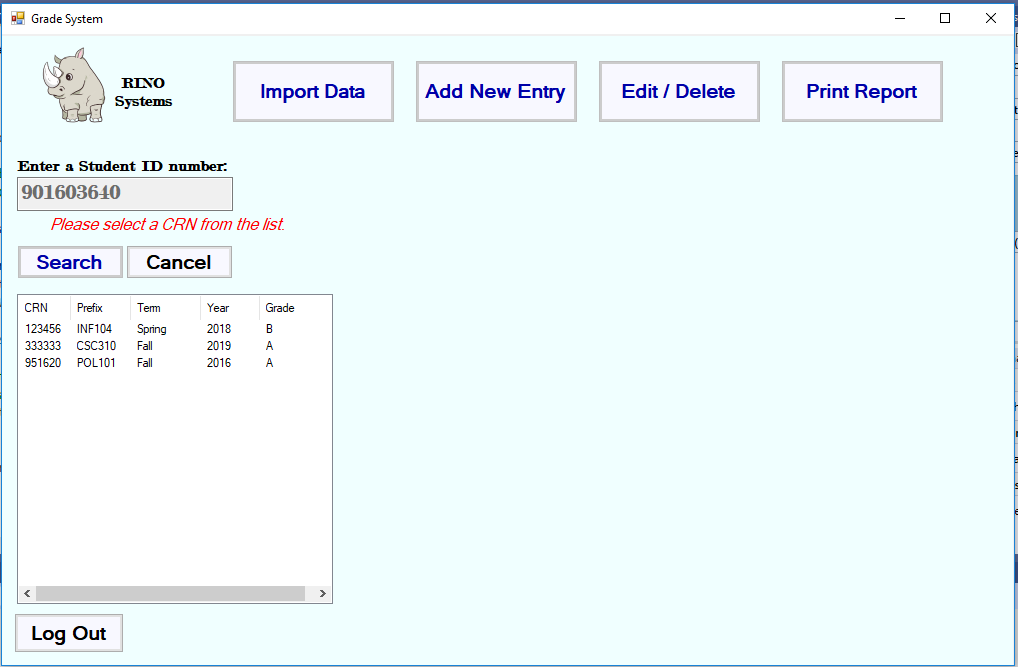


Figure . Display list of courses student has taken

* 1. The staff member shall select a course to view options.
  2. Once the staff member selects a course, the system will display an “Edit”, and “Delete” button. (Figure 15)

Graphical user interface, application

Description automatically generated

Figure 15. Display Edit/Delete Buttons

* 1. The staff member shall select “Edit”, or the “Delete” button.
  2. The staff member shall select “Edit” Button.
     1. The system shall display a dialog box to enter the updated grade. (Figure 16)
     2. The staff member shall enter a new grade for the selected course.
     3. The staff member shall click “Save” button.
        1. The system will check if a new grade was entered.
           1. If a grade was not entered, a message will be displayed, saying “Must Enter a Grade!”

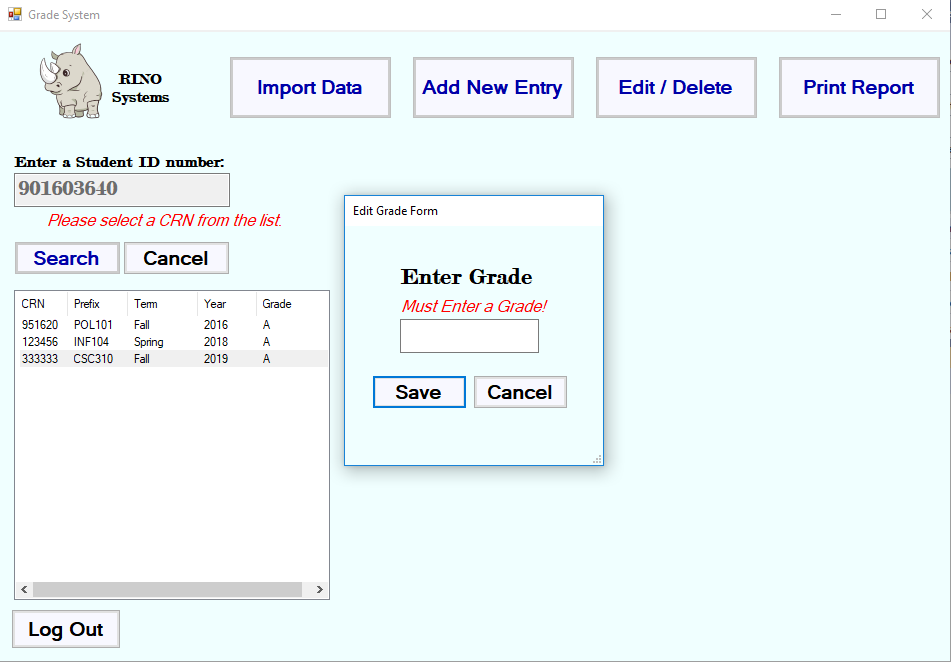


Figure 16 . Dialog box for updated grade

* + - * 1. If a grade was entered, continue to step 3.11.4
    1. The system shall connect to the central database.
    2. The system shall update and save the database.
    3. The system shall close the database connection.
       1. The system shall update the list panel with the new grade. (Figure 17)

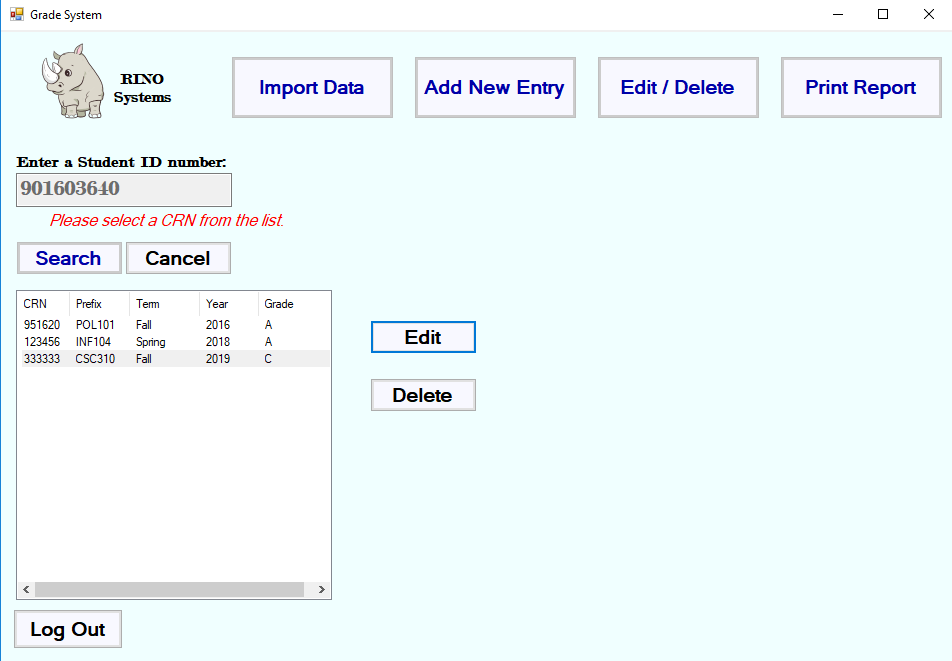


Figure . Listview update after edit grade use case

1. The system shall allow the staff to delete grades of each student in the database
   1. The staff member shall click the “Edit/Delete Entry” button.
   2. The system shall prompt the staff member to enter a student ID number. (Figure 6)
   3. The staff member shall enter a student ID number.
   4. The staff member shall click “Search” button.
      1. The system shall validate the student ID.
         1. The system shall display a message “Student not found” if the student ID is invalid. (Figure 7)
   5. The system shall retrieve the information from the central database.
   6. The system shall allow the staff member to select a course.
   7. The staff member shall select a course to view options. (Figure 14)
   8. The system will display a “Edit”, and “Delete” button.
   9. The staff member shall select the “Delete” Button.
      1. The system shall open a connection to the central database.
      2. The system shall delete the selected row from the database.
         1. The system shall update the list panel with the deleted course.
      3. The system shall display a message saying, “Course deleted successfully!” (Figure 18)
         1. The system shall update the list of courses.
   10. The system shall close the database connection.

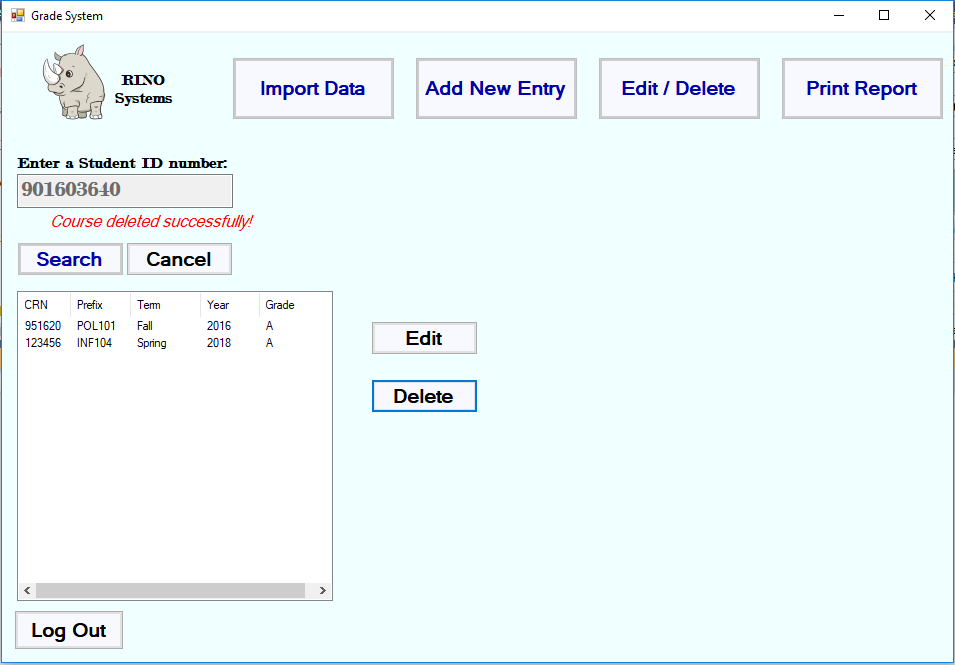


Figure . Course deleted successfully message and updated list

1. The system shall allow a staff member to print a report card (transcript) for a selected student.
   1. The staff member shall click the “Print Report” button.
   2. The system shall display a pop-up box with a button saying “Save Report” for the staff member to generate and save a report based on the student ID number.

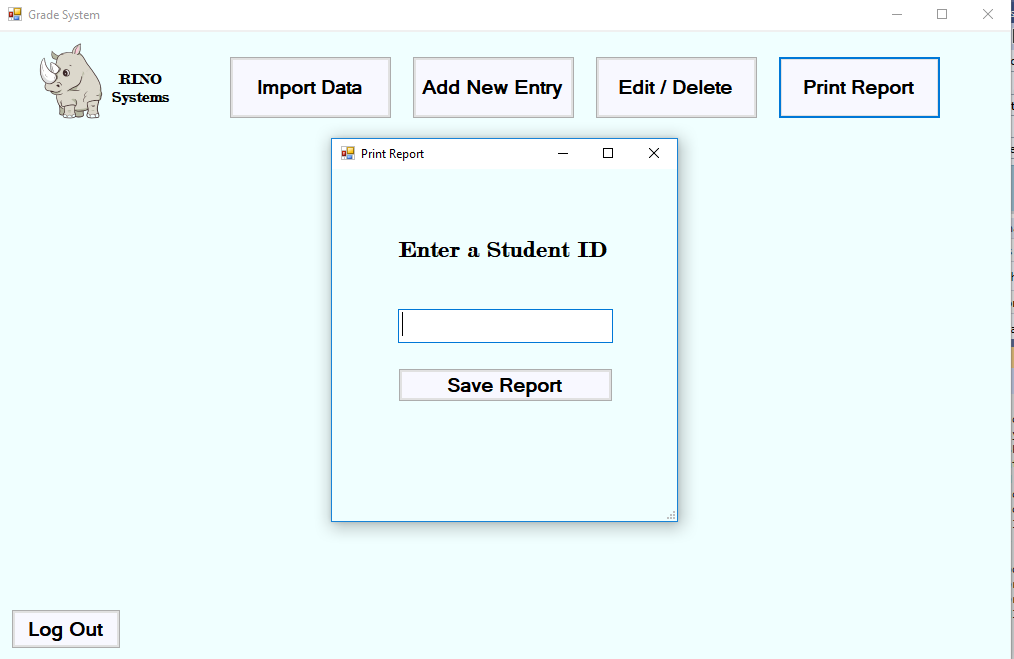


Figure . Enter studentID to print report

* + 1. The staff member shall enter a student ID number.
    2. The staff member shall click the “Save Report” button.
       1. If no studentID has been entered, a message will appear. (Figure 20)

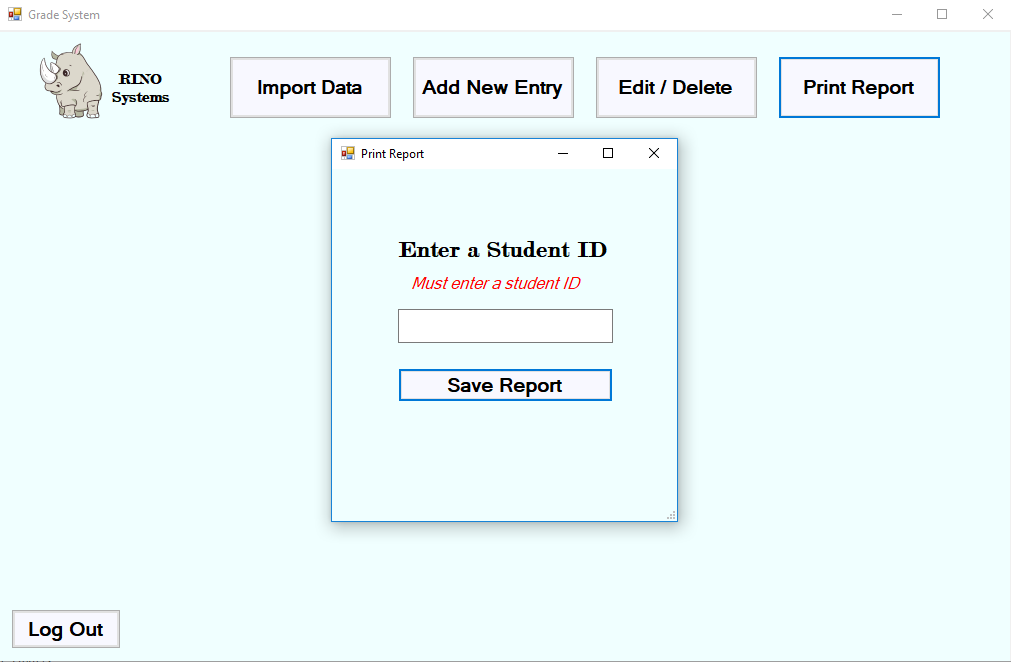


Figure . Must enter a studentID number for printing a report

* + - 1. If the studentID entered is not found in the database, a message will appear. (Figure 21)

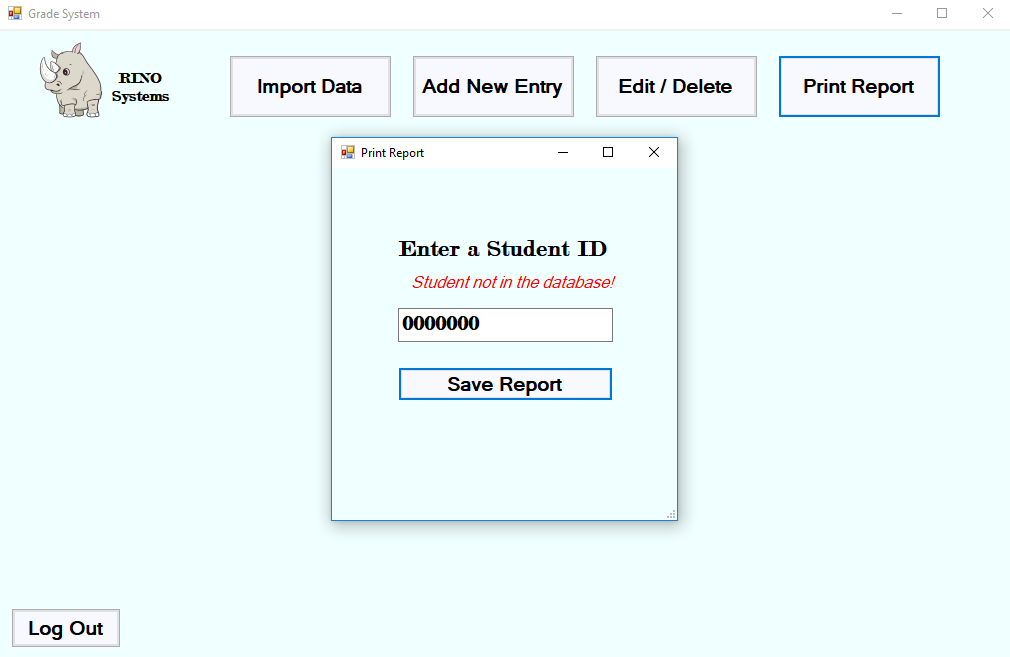


Figure . Student not in database when printing report

* + - 1. The system shall retrieve the student information from the central database system with the student’s ID.
      2. The system shall calculate the student’s GPA.
      3. The system will create a transcript which lists the classes and grades received for courses previously enrolled in.
    1. The system shall format all the information and save it to a PDF file.
    2. The system shall save the PDF file to the Desktop.
       1. The name of the PDF file will be “Transcript”
       2. The system shall display a message box, saying “Your document has been created successfully” (Figure 22)
       3. The system shall display an “OK” button.
       4. The staff member shall click “OK” and return to main menu.

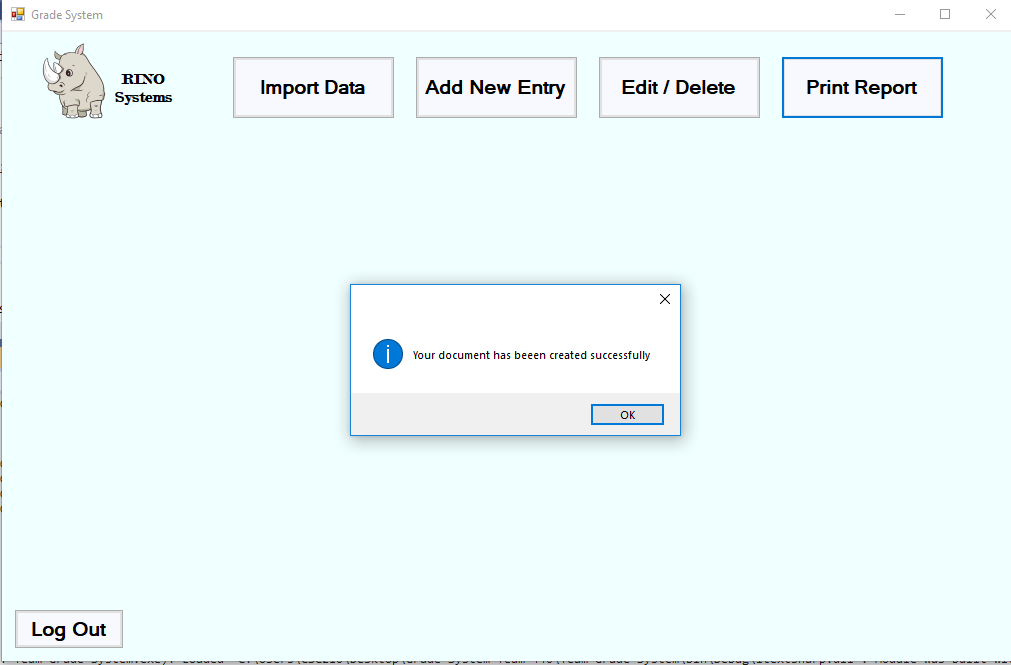


Figure . Transcript successfully created message

1. The system shall allow a user to sign in the system to access the main menu.
   1. The user will be prompted to login into the grade system using username and password. (Figure 23)



Figure . Log-In Menu

* 1. The staff member shall enter their username and password.
  2. The system shall check if both a username and a password were entered.
     1. If there is a missing field, a message will display saying, “Fields must be filled out!” (Figure 24)



Figure . Fields missing when logging in

* + 1. If there are no missing fields, continue to step 6.4.
  1. The system shall validate the login combination.
     1. If login combination is not found in database, a message will display, saying “Username or password are incorrect!” (Figure 25)

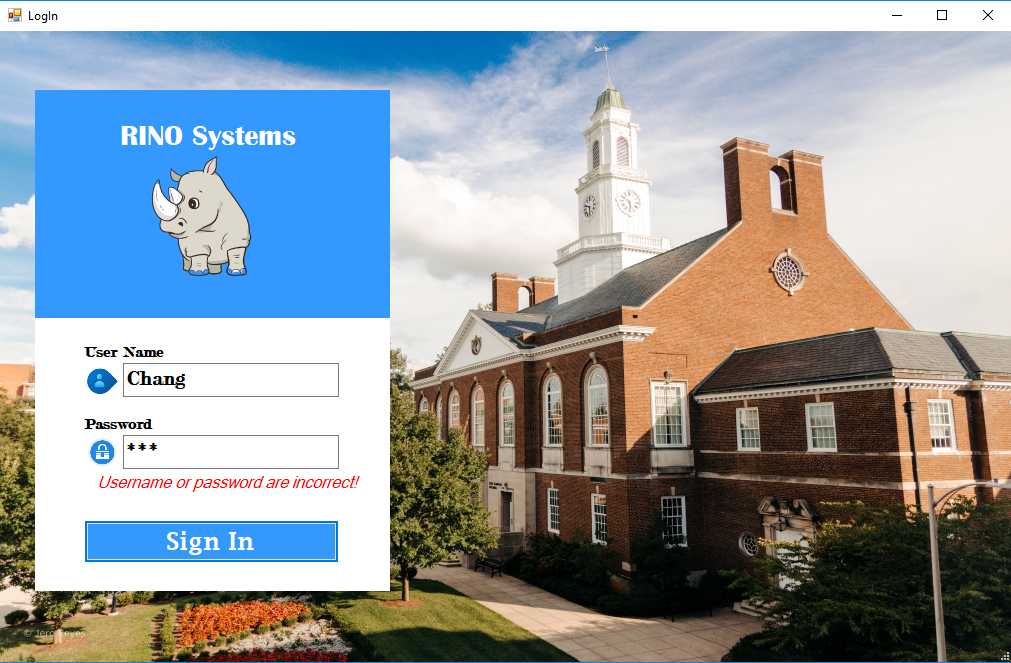


Figure . Invalid login combination

* + 1. Graphical user interface, application, Word

       Description automatically generatedIf log-in combination is found in the database, the system shall display a menu to the staff member. (Figure 26)

Figure 26. Main Menu Panel

1. The system shall allow the staff member to log out from grade system
   1. The system shall display a user menu to the staff member.
   2. The system shall display a “Log Out” Button on the main menu to the customer. (Figure 26)
   3. The customer shall select “Log Out” in the menu.
   4. The system shall display the sign in screen. (Figure 23)

## Non-functional Requirements:

1. Each username must be unique.
2. Staff members will be inserted manually into the system.
3. The grade system can only serve one staff member at a time.
4. A student cannot have more than one grade for the same course in the same term.
5. Grades are only limited to A, B, C, D, and F, or P(for pending).
6. The excel file to read and parse the grades will be in a folder named, “Grades [Year] [Semester]”, with file names following this order, “[Course Prefix] [Number] [Year] [Semester]”.
7. The grade point average is out of 4.0.
8. All possible courses will be already in the catalog section of the database.
9. The system will save the report in the Desktop folder of the computer.

# Use Case Diagram

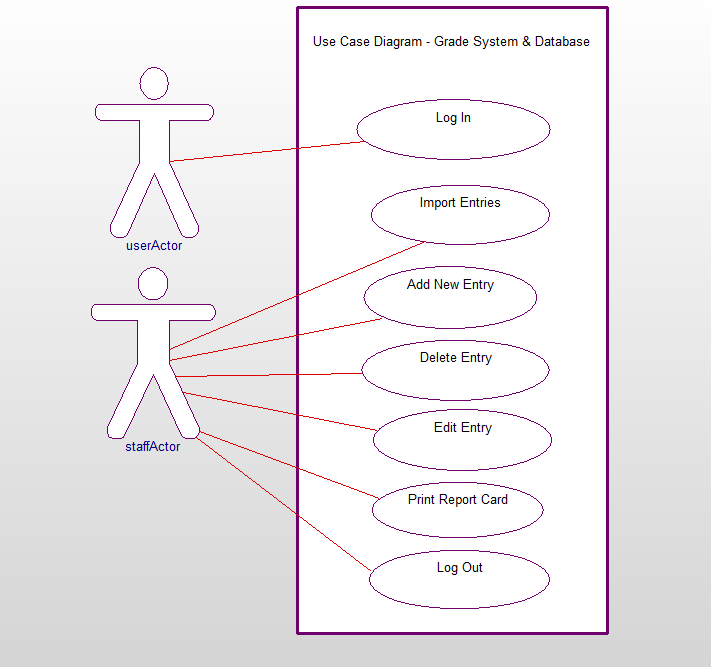


Figure 27 – Use case diagram for Grade system

The userActor, who has not been authenticated, will only have access to login. The staff member will be able to import entries, add a new entry, edit or delete an entry, print a report card/transcript, and log out.

# Class Diagram

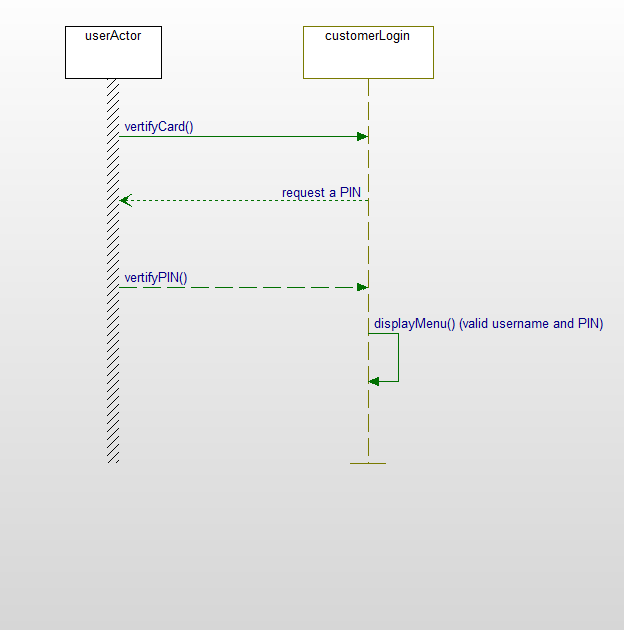
# A picture containing diagram Description automatically generated

# 

Figure 28 - Class Diagram

This is a diagram to show the related attributes and operations within each class that the system will use to perform the functionalities needed. The coursework class will have most of the functions in the system.

# Sequence Diagrams



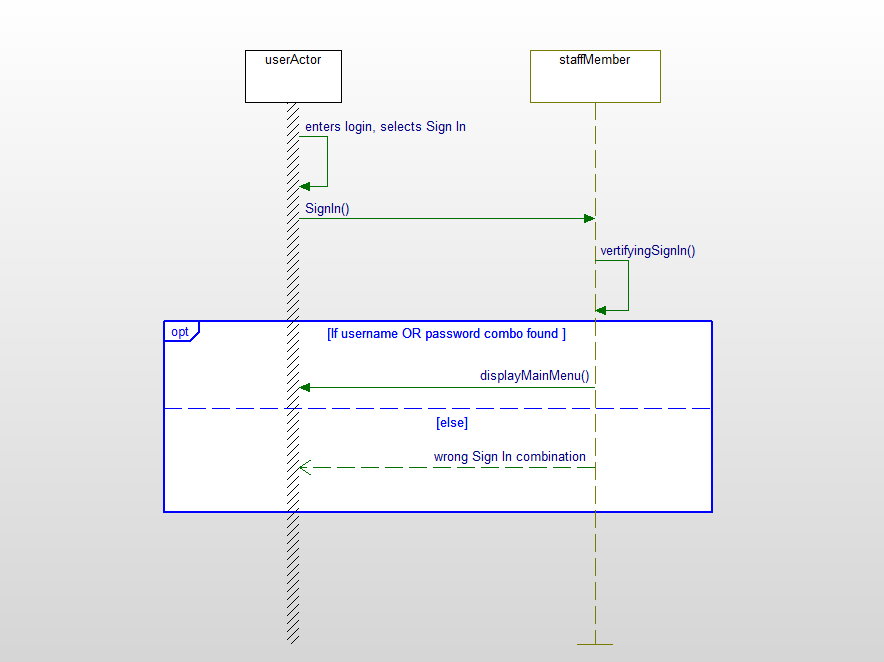


Figure 29 - Sequence Diagram for signing into the grade system

This is the diagram that shows the steps of the “Signing In” functionality. The system will verify the username and password of the user to display the menu if the combination is valid.

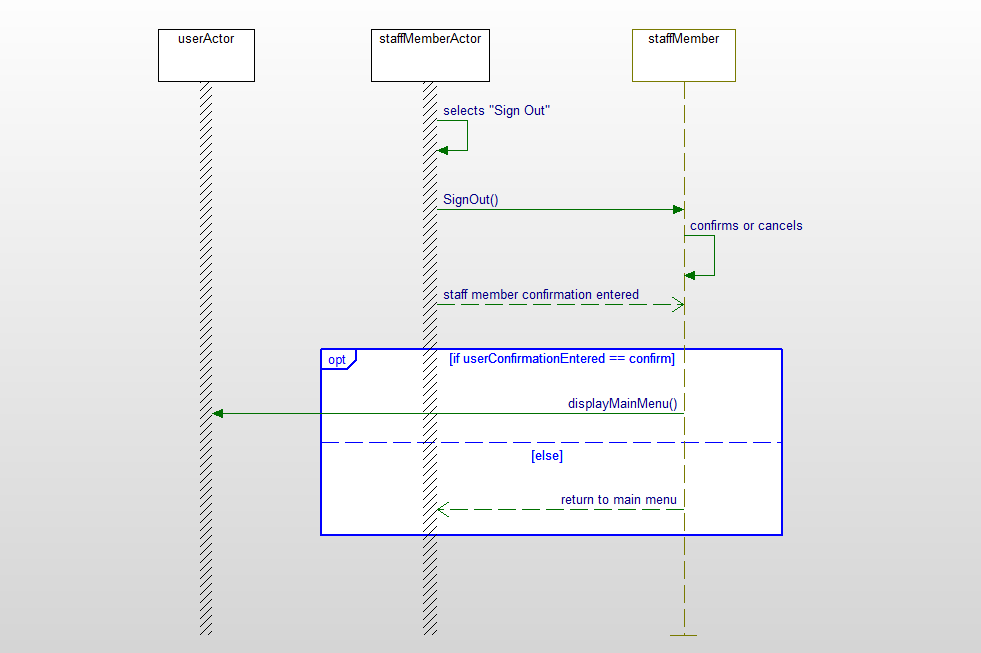


Figure 30 - Sequence Diagram for signing out of the system

This diagram shows the steps the system will take when signing out of the system. After signing out, the system will display the login menu.

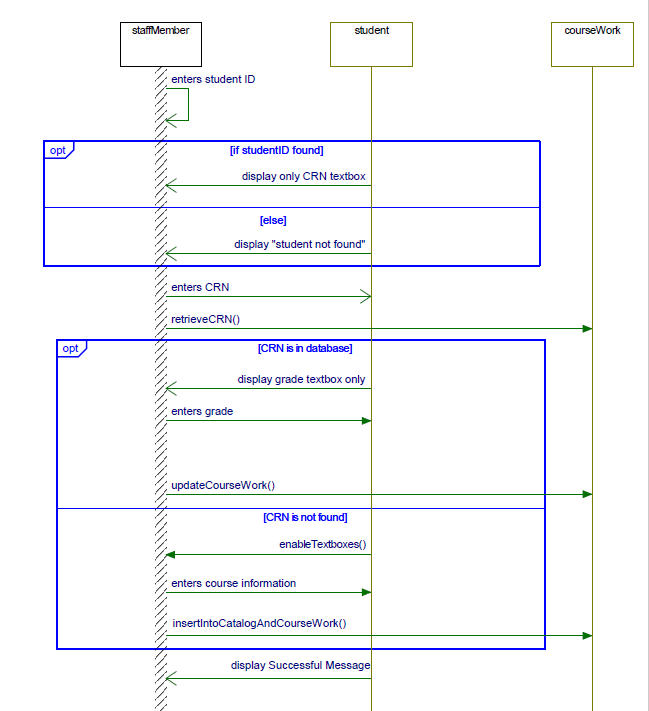


Figure 31 - Sequence Diagram for adding a new entry to the grade system

This diagram shows the steps the system will take when adding a new entry. The system will require the input of the student ID and the CRN in order to look for the student and see if the course entered already exists in the database.

Diagram

Description automatically generated with medium confidence

Figure 32 - Sequence Diagram for editing a grade

This diagram shows the steps the system will take when editing a grade. The system will look for the student and, once found, it will display all the courses the student has taken so the staff member can select a course. Once a course is selected a dialog box will pop up that will allow the updated grade to be inserted and saved.

A picture containing diagram

Description automatically generated

Figure 33 - Sequence Diagram for deleting a grade

This diagram shows the steps the system will take to delete a course taken by a student. The system will ask for the student’s ID and display a list of courses taken. Once a course is selected the staff member will have the option to delete that course with a button.

Diagram, timeline

Description automatically generated

Figure 34 - Sequence Diagram for importing data

This is the steps the system will take when importing data into the database. The system will give the user the ability to select a file to upload. The system will insert the data from the selected excel file.

Diagram

Description automatically generated

Figure 35 - Sequence diagram for print report

This is the sequence diagram for printing a report. The system will ask for the studentID and it will retrieve all that student’s information. This is also when the student’s GPA is calculated. The system will then write that information to a PDF file.

# Activity Diagrams

Figure 36 - Activity Diagram for signing in

This diagram shows the various functions the system will call when the user signs in. The system will check to see if the login combination is associated with an authorized account.

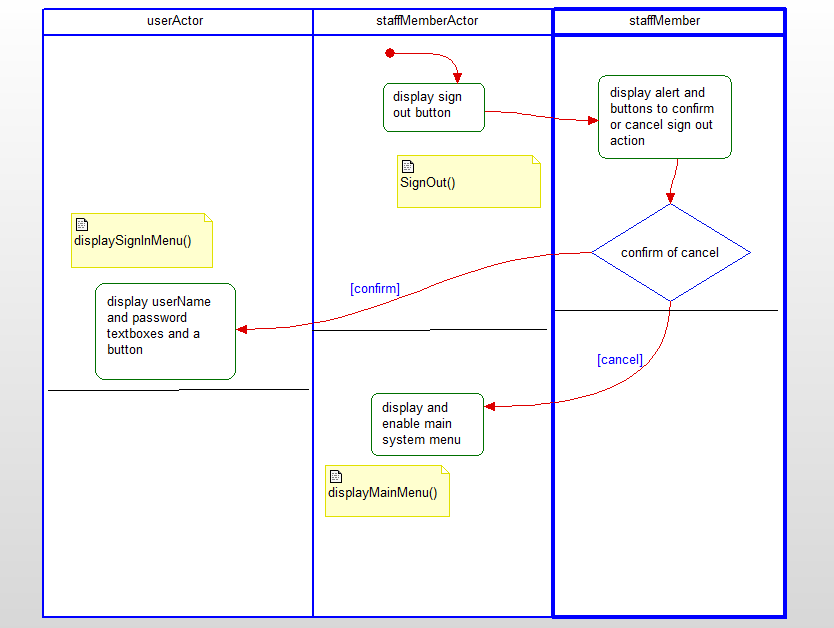


Figure 37 - Activity Diagram for signing out

The diagram shows the methods used to sign out of the system. After signing out, the system will display the user login menu.

Diagram

Description automatically generated

Figure 38 – Activity diagram for adding a new entry

This diagram shows the functions that will be used for adding a new entry into the database. The system will check to see if the entered CRN number is in the database and if it is not then it will ask for all the information of that course so it can be inserted into the catalog. It will also ask for the grade that the student received so it can be added to that student’s coursework history.

Diagram

Description automatically generated

Figure 39 - Activity diagram for editing a grade and deleting a course

This diagram shows the functions that will be used for editing a grade and/or deleting a course from a student’s coursework history. When the staff member selects a course from the screen it has the option to either edit it or delete it given the buttons displayed.

Diagram

Description automatically generated

Figure 40 - Activity Diagram for importing data

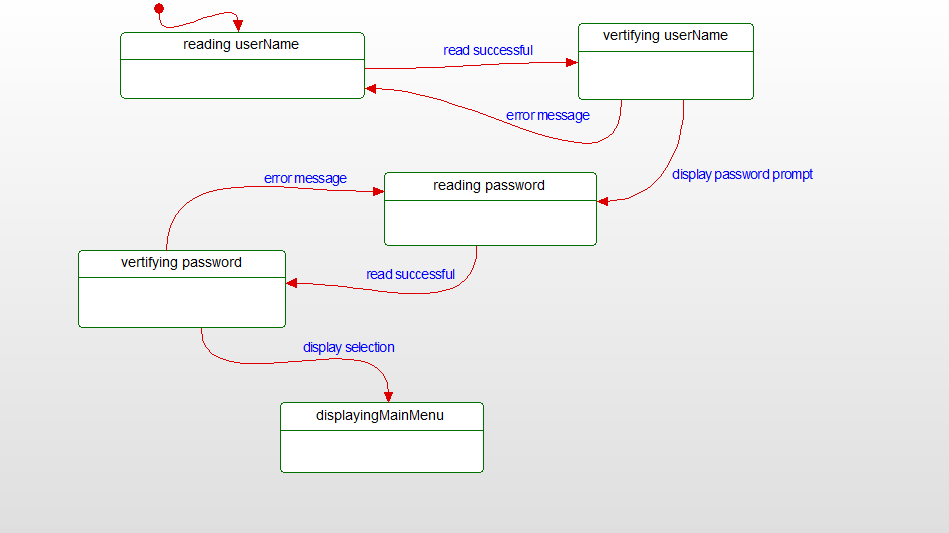
This diagram shows the functions associated with importing data. The system will let the staff member pick the file they want to upload to get its data.

Diagram

Description automatically generated

Figure 41 - Activity Diagram for printing report

This diagram shows the function involved when the staff member wants to print a report of a student. The system will first calculate the GPA of the student and then retrieve the student’s information from the database and save that information to a file.



# State Diagrams

Figure 42 - State diagram for signing in

This diagram shows the various states the system is in when a user signs in. If the user does not enter a correct login, they will have the option to re-enter the credentials.

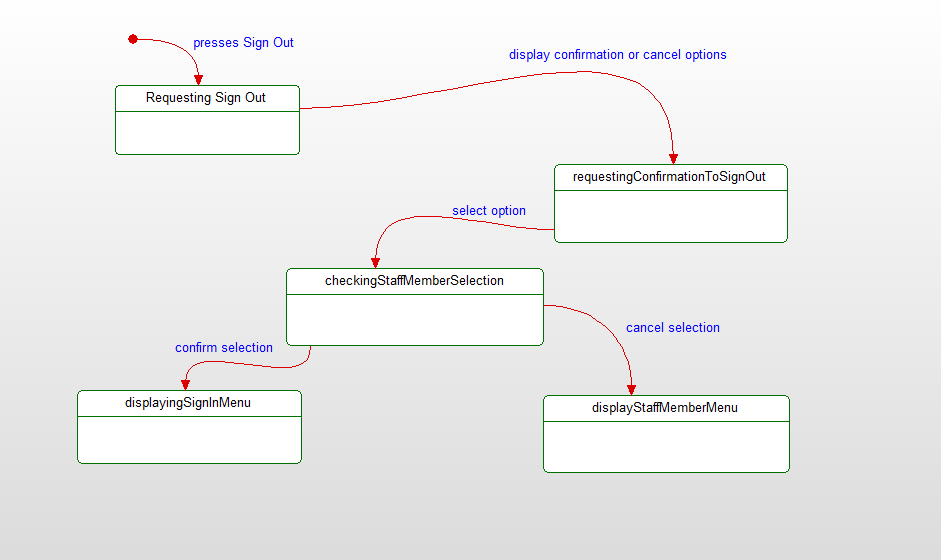


Figure 43 - State diagram for signing out

This diagram shows the states the machine is in when signing out of the system.

This diagram shows the states of the system when adding a new entry. The system Diagram

Description automatically generatedwill check if the studentID entered is in the system and if the CRN entered already exists or not.

Figure 44 - State diagram for adding a new entry

Diagram

Description automatically generatedThis diagram shows the states of the system for editing a grade or deleting a course. Both use cases overlap at the beginning given that the method used to retrieve the studentID and select the course is the same.

Figure 45 - State diagram for editing a grade or deleting a course

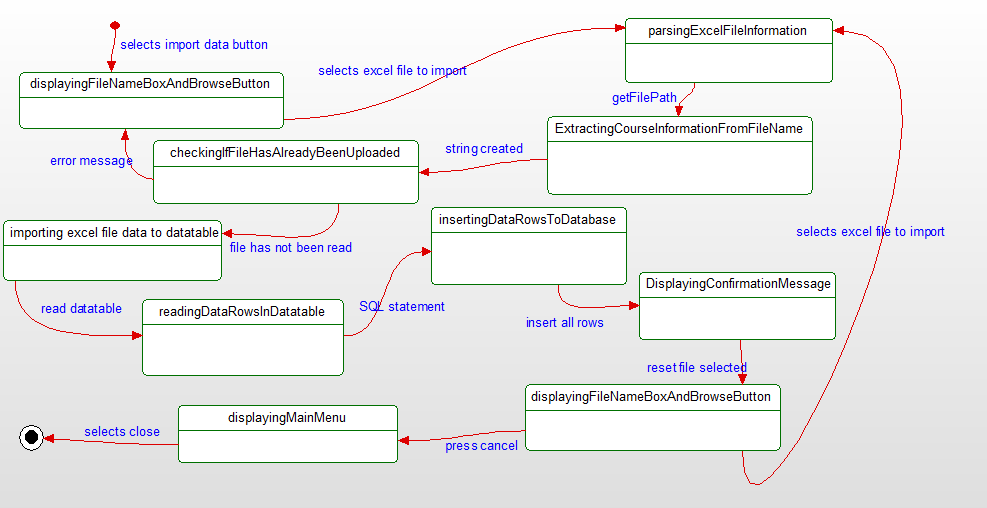


Figure 46 - State diagram for importing data to the system

This is the diagram that shows the different states of the system when the staff member wants to import data to the system. There are 10 different system states associated with this use case.

Figure 47 - State diagram for printing a student report

Graphical user interface, diagram, application

Description automatically generated

This diagram shows the different states the system is in when the staff member requests the print report button. The system will proceed to find the desired student and calculate the GPA. Then it will generate the report to a file with all the student’s previous coursework.

# Database Design

## ER Diagram

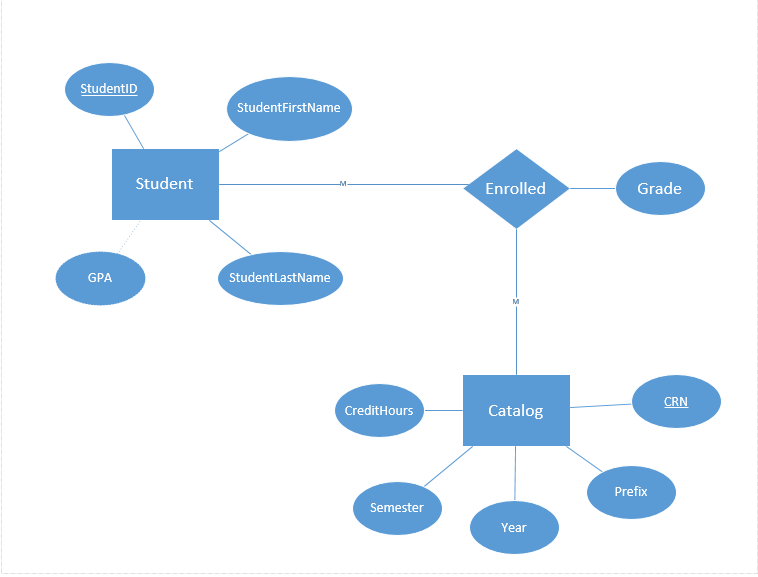


Figure 48 - ER Diagram for structuring the database

This is the ER diagram that is used to design the database. The relationship between the student and the catalog will be a many to many and the Enrolled relationship will create a bridge table between the two. The bridge table will serve as a table to store the grades of each course that each student takes. The bridge table will utilize the primary keys of both StudentID and CRN.

## Table Schema

Student

|  |  |  |  |
| --- | --- | --- | --- |
| StudentID | StudentFirstName | StudentLastName | GPA |

StudentCourseWorkFromCatalog

|  |  |  |
| --- | --- | --- |
| CRN | StudentID | Grade |

Catalog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CRN | Prefix | Year | Semester | CreditHours |

Users

|  |  |
| --- | --- |
| username | password |

This is a visualization of the table schema within the database. The Catalog table serves as a list of all the possible courses a student can take. The student table will store information of each particular student, this is also where the GPA will be stored upon calculation. The users table has logging in combination for verifying a user who wants access.

# Conclusion

This project consisted of being very through with all the needs of the registrar’s office. The functionality of the system needed to be very detailed in order to set up and implement the logic and the code. The sequence, state, and activity diagrams contributed to being able to map out how the system would function and work in different ways. The database implementation was done using heidiSQL, and the front-end, along with the communication to the database was all done in c#. The edit and delete functions were integrated within the same button in order to save space, time and money. We decided to add a login and logout feature to the system in order to provide security to the system given that student’s grades are an important piece of data.

# Data Dictionary

**Activity Diagram** – describes parallel and conditional activities, use cases, and system functions at a detailed level.  
**Attribute** – a specification that defines a property of an object, element, or file.  
**Class** – blueprint or a set of instructions to build a specific type of object

**Entity** (in database) – is a thing, person, place, unit, object or any item about which the data should be captured and stored in the form of properties, workflows and tables.  
**ER Diagram** – An entity relationship diagram shows the relationships of entity sets stored in a database

**Function** – part of a program that has its own name. This name can be used in the program as a command. When a function is called, the commands of which it consists of are executed.

**Grade System** – The system whose operations consist of being able to import data, edit a grade, delete a course, add new entries, and print a report card of a student.

**Operation** – an action that is carried out to accomplish a given task  
**Sequence Diagram** – type of interaction diagram because it describes how and in what order a group of objects works together

**Use Case** – a description of how users will perform tasks on a system or software  
**State diagram** – describes the behavior or status that the system may be in. It consists of states, transitions, events, and activities the system may perform