Vaughan, Ethan G.

CSC 440 – Applied software engineering

individual Project: grade and gpa calculator

September 14, 2019

Requirements specifications report

*Table of contents goes here*

*Table of figures goes here*

# Introduction

## Problem Statement

Computer science students at Eastern Kentucky University need to find out their grades in certain classes. They will have their grades on several completed assignments, with more assignments to come. They need to know what grade(s) they will need on future assignment(s) in order to receive, say, 89.5% overall in a course. Perhaps they would like to know what grade they will get overall in a course, assuming they make, say, 75% on remaining assignments and/or tests. This grade calculator will allow students to perform these “what-if” scenarios. It will show them which classes they need to focus their attention on and hard they should study in each class.

Additionally, computer science students needto keep track of their GPA. They may want to know their GPA in different areas, such as their major GPA, overall GPA, and GPA in supporting courses. And, before a semester is over, a computer science student would like to know what effects different final grades will have on the student’s overall GPA. This calculator will have a functionality for that as well. It lets them enter the classes they have taken and the final grade in each class (A, B, etc.). The GPA calculator then lets students perform “what-if” scenarios on future classes, similar to the grade calculator portion of this project. They can ask the calculator what GPA they will have overall if they make a C on one class, a B in another, and two A’s in the other classes.

One final piece of information that computer science students would like to keep track of is their progress towards their concentration specific C.S. degree. C.S. students would like to track which classes they need to take (general education courses, supporting courses, core courses, etc.).

## Proposal

My solution to the needs of computer science students at Eastern Kentucky University is a grade and GPA calculator, with added functionality to keep track of a student’s progress towards degree completion. The grade calculator would allow students to record grades on assignments/tests and perform “what-if” scenarios, showing them what grades they would need on remaining coursework in order to receive a certain final grade overall (such as 88%) and what grade they would end up with in a course if they received a specified grade on remaining coursework.

For the GPA portion of my application, the application would calculate a student’s overall GPA as they enter different final grades. The app would show a student what GPA he/she would have if certain final grades were achieved in current or future courses.

Lastly, this app would track a student’s progress towards a degree concentration. It is tailored specifically for computer science students at Eastern Kentucky University, taking one of EKU’s C.S. concentrations.

# System Description

The project is to build a system that allows students to input grades in for individual classes to help calculate the grade in that class, also allowing to calculate GPA, and know how close they are to fulfilling their majors’ requirements. The system shall give the student options to add or remove grades as needed while doing the same for classes current or completed. The system shall allow the student to modify the grades or classes. The system shall allow the student to perform what-if scenarios on grades and overall GPA.

# System Requirements

*Summarize system requirements*.

## Functional Requirements

R1. The system shall allow a user to store grades for completed assignments.

* 1. The user shall select the semester and class from the drop-down menus.



* 1. The user shall click the “Add Grade” button.
  2. The system shall create a new row that the user can place the grade information into.



* 1. The user shall enter the assignment/test description (e.g. “Assignment 1”), the grade received on the assignment (as a percentage), and the assignment’s weight (as a percentage).
  2. The user shall press the “Submit Changes” button.



* 1. The system shall recalculate the user’s overall grade in the course and send the new grade information to the database.

R2. The system shall allow a user to figure out what grades will be needed on remaining assignments in order to receive a desired overall grade in a course. Additionally, the system shall tell the user what grade he/she will have in the course, given a certain grade on remaining coursework.

2.1. The user shall select the class for which he/she wishes to perform a what-if grade scenario.

2.2. The system shall display the user’s grades for that class.

2.2.1. The system shall show the assignment names, grades received for the assignments, and weight of the assignments.

2.2.2. The system shall calculate the user’s current grade in the course (calculated using grades the user has already submitted) as well as the “percentage points” earned for that class.

2.3. The user shall enter the desired grade, as a percentage, in the box “What if I made \_\_\_ on remaining coursework?”.



2.3.1. This part of the what-if calculation will figure out what grade the student would end up with in the course if he/she made the specified grade on all remaining coursework.

2.4. The user shall enter the desired grade, as a percentage, in the text box “What do I have to make on remaining coursework to get \_\_\_ overall?”

2.4.1. This calculation will show the user what grade he/she must make on remaining coursework in order to receive specified grade overall in the course.

2.5. The system shall perform the calculations and display the results.



R3. The system shall allow a user to delete a grade.

3.1. The user shall select the class for which he/she wishes to delete a grade from.

3.2. The user shall click the “Delete a Grade” button.



3.3. The system shall unhide a combo box that contains all the grades for the class.

3.3.1. The system shall also change the text in the button from “Delete a Grade” to “Delete”.



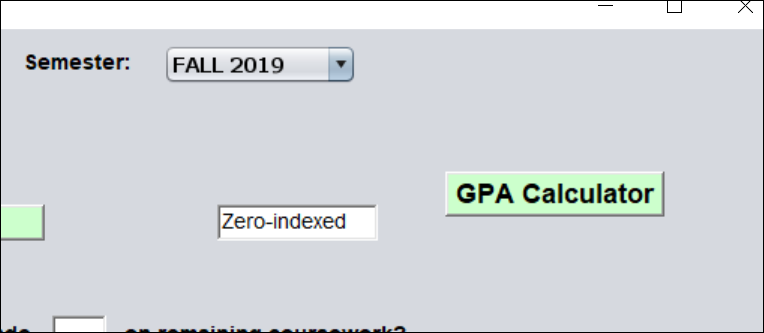
3.4. The user shall select the grade from the combo box that he/she wishes to delete.

3.5. The user shall click the “Delete” button.

3.6. The system shall hide the combo box, reset the text in the button to “Delete a Grade”, remove the deleted grade from the list of grades, recalculate the user’s current grade and percentage points in the course, and remove the grade from the database.

R4. The system shall allow a user to add a final grade for a completed class.

4.1. The user shall click on the “GPA Calculator” button from the start screen.



4.2. The system shall calculate the user’s GPA by fetching the user’s final grades.

4.3. The user shall click on the “Add/View Classes” button.



4.4. The system shall display a combo box with the user’s completed classes.

4.4.1. When the user selects a new class from the combo box, that class’s credits, the user’s final grade, and the semester taken shall be displayed.



4.5. The user shall click the “Add a Class” button.

4.6. The system shall prompt the user to enter the information about the class (class name, such as “ENG101”, credits, final grade (A, B, etc.), category (such as “Gen ed element 6”), and semester, such as “fall 2013”).



4.7. The user shall click the “Submit” button.

4.8. The system shall add the class to the combo box of classes.

4.8.1. The screen for adding a class will disappear and a success message will be displayed.



4.8.2. The class and its information will be sent to the database.

4.8.3. The user’s GPA will reflect the new final grade submitted.

R5. The system shall allow a user to modify a grade.

5.1. The user shall select a class from the list of classes.

5.2. The user shall click on the part of the grade (assignment/test name, grade received, weight) and change the text as needed.



5.3. The user shall click on the “Submit Changes” button (figure xxx).

5.3.1. If the user only wanted to see what effect the modified grade would have without submitting it to the database, he/she shall click on the “Refresh Calculations” button and the system shall display what grade the user would have overall given the modified grade(s).

5.4. The system shall refresh the user’s current grade in the course. The modified grade shall be sent to the database.

R6. The system shall allow a user to delete a class.

6.1. The user shall select the class he/she wishes to delete.

6.2. The user shall click the “Delete Class” button.



6.2.1. The “Delete Class” button shall change to “Confirm”.

6.3. The user shall click on the “Confirm” button, to confirm that the class should be deleted.

6.4. The system shall remove the class from the screen, display a new class, and submit the changes to the database.

R7. The system shall allow a user to add a current class.

7.1. The user shall click the “Add A Class” button.



7.2. The system shall unhide the following text boxes: class name (where the user puts the class’s 6-character name, such as “ENG101”), initial assignment description (for the first graded assignment/test, example: “Exam 1”), grade received (grade received on the initial assignment/test, e.g., 88.5%), weight (weight of the initial assignment/test, e.g., 10%), and semester (such as “fall 2015”).



7.3. The user shall enter all the above information and click “Submit”.

7.4. The system shall send the data to the database and add the class to the drop-down menu.

R8. The system shall track a user’s degree progress.

8.1. The user shall click the “View Degree Progress” button (see figure xxx) from the initial screen.

8.2. The system shall display the user’s progress, based on the information already submitted about the user’s completed classes (hours, class type, etc.).

8.3. The user shall click the degree concentration he/she is seeking, and the system shall adjust the student’s progress, based on the newly selected computer science concentration.



R9. The system shall allow a user to calculate future GPA, given certain final grades in classes.

9.1. The user shall go to the GPA calculator by clicking the “GPA Calculator” button on the initial screen.

9.2. The system shall display the user’s current GPA, based on final grades that have been submitted previously for completed classes.

9.3. The user shall enter the final grade and credit hours for some classes.

9.4. The user shall click on the “Calculate” button.

9.5. The system shall display what GPA the user will have based on the submitted final grades and credit hours and the information just submitted.



R10. The system shall allow a user to delete a completed class.

10.1. From the initial screen, the user clicks on the “View Completed Classes” button.

10.2. The system displays a screen that shows all the user’s completed classes in a drop-down box.

10.3. The user selects the class he/she wishes to delete.

10.4. The user clicks the “Delete Class” button.

10.5. The system asks the user to confirm the delete.

10.6. The user clicks the “Yes” to confirm.

10.7. The system deletes the class from the database and the drop-down box, and displays a success message.

R11. The system shall allow a user to modify a completed class (e.g., change the class’s category, credit hours, final grade, etc.).

11.1. The user shall go to the final grades overview for completed classes.

11.1.1. From the initial screen, user clicks on “View Completed Classes”.

11.2. The user makes desired changes.

11.3. The user clicks the “Submit Modifications” button.

11.4. The system asks user to confirm changes.

11.5. The user clicks the “Yes” button to confirm changes.

11.6. The system submits the changes to the database and displays a success message.

## Nonfunctional Requirements

1. All buttons must have a purpose

1.1. When a button is pressed it will have an outcome of some sort. Whether be visible or not.

2. System shall not crash under invalid data input.

2.1. When the user enters data into the system that is wrong data type or just invalid the system will error check and inform the user.

3. Any information in the database should be secure.

3.1. Only those who have access to that information should receive that information.

4. The functions of this system should be consistent and reliable.

4.1. The system should always output the same content given the same input.

## Domain Requirements (optional!!!)

# Use Case Diagram



Adding a grade – Use Case to add a grade to the system.

Deleting a grade – Use Case to delete grade from the system.

Modifying a grade – Use Case to modify a grade in the system.

Perform what-if grade audit – Use Case to do a what-if grade in the system.

Delete a current class – Use Case to delete a current class.

Adding a current class – Use Case to add a current class.

Adding a completed class – Use Case to add a completed class.

Deleting a completed class – Use Case to delete a completed class.

Modifying a completed class – Use Case to modify a completed class.

Track degree progress – Use Case to see how close you are to graduating.

Perform what-if GPA audit – Use Case to do a what-if GPA in the system.

# Data Flow Diagrams

## Context DFD (level 0)

## Level-1 DFD

## Level-2 DFD *etc.*

# Conclusion

# Data Dictionary