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

*Very abstract – put in detail. Use your imagination to guess how the system will work. Write it down in plain English.*

# System Requirements

*Summarize system requirements*.

## Functional Requirements

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

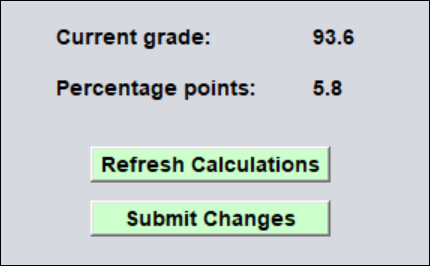
* 1. The user shall select the semester and class from the drop-down menus.



* 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”.



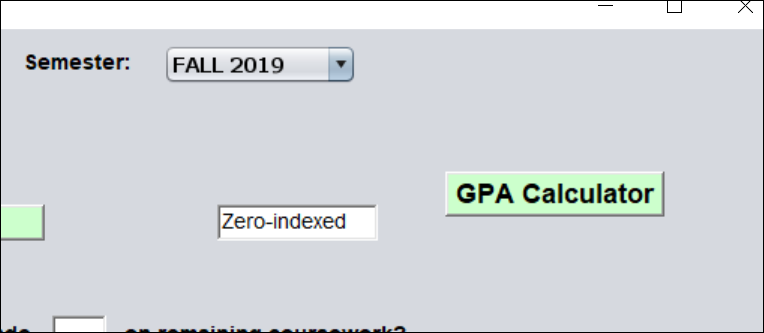
3.4. The user shall select the grade from the combo box that he/she wishes to delete.

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.



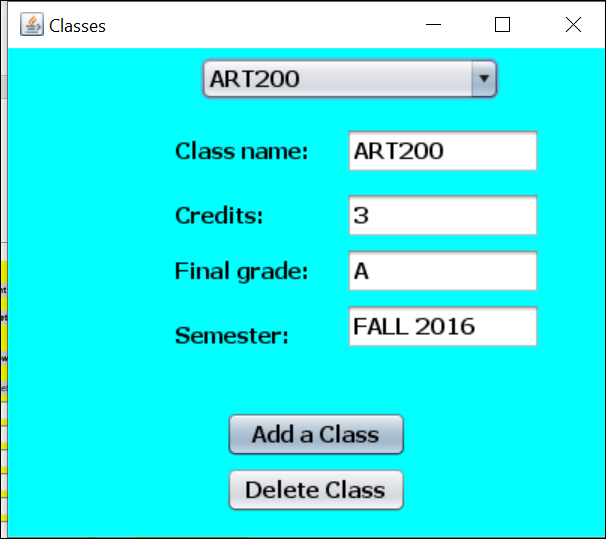
4.2. The system shall calculate the user’s GPA by fetching the user’s final grades.

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.

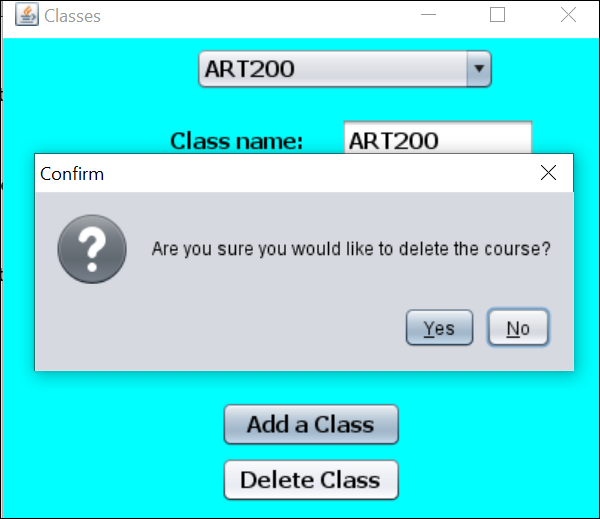
R8. The system shall allow a user to delete a completed class.

8.1. The user shall go to the “Add/View Classes” screen.

8.2. The system shall place all the completed classes in a drop-down menu and display the first one.

8.3. The user shall select the class he/she wishes to delete and click the “Delete Class” button.

8.4. The system shall ask the user to confirm the delete request.

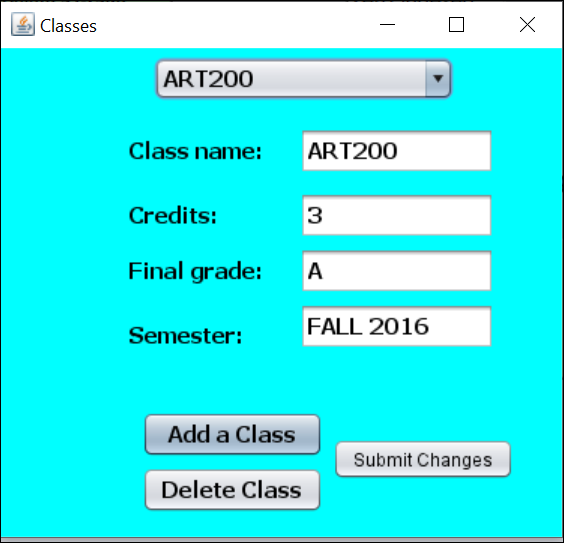


8.5. The user shall click the “Yes” option.

8.6. The system shall remove the class from the database, take it off the screen, and display a success message.

R9. The system shall allow a user to modify a completed class.

9.1. The user shall go to the completed classes screen.



9.2. The user shall enter the change(s) to the class you wish to modify (for example, perhaps the user wants to change the class’s name from CSC320 to CSC311).

9.3. The user shall click the “Submit Changes” button.

9.4. The system shall click “Yes”.

9.5. The system shall submit the changes to the database.

## Nonfunctional Requirements

## Domain Requirements (optional!!!)

# Use Case Diagram

Put some description of the diagram here. Describe all use cases using a few sentences. COMPLETE SENTENCES, PLEASE. This phase is the “what” phase. Design phase is the “how” phase.

# Data Flow Diagrams

## Context DFD (level 0)

## Level-1 DFD

## Level-2 DFD *etc.*

# Conclusion

# Data Dictionary