individual Project: grade and gpa calculator

Vaughan, Ethan G.

CSC 440 – Applied software engineering

october 14, 2019

Design report

Table of Contents

[Table of Figures 2](#_Toc25778979)

[I. Introduction 4](#_Toc25778980)

[A. Problem Statement 4](#_Toc25778981)

[B. Proposal 4](#_Toc25778982)

[II. System Description 4](#_Toc25778983)

[III. System Requirements 5](#_Toc25778984)

[A. Functional Requirements 5](#_Toc25778985)

[B. Nonfunctional Requirements 13](#_Toc25778986)

[IV. Use Case Diagram 14](#_Toc25778987)

[V. Data Flow Diagrams 15](#_Toc25778988)

[A. Context Level DFD 15](#_Toc25778989)

[B. Level-1 DFD 16](#_Toc25778990)

[C. Level-2 DFDs 17](#_Toc25778991)

[D. Level-3 DFDs 23](#_Toc25778992)

[VI. Structured Chart 25](#_Toc25778993)

[A. Top Level Structure Charts 25](#_Toc25778994)

[B. Lower Level Structure Charts 26](#_Toc25778995)

[VII. Data Store Design 29](#_Toc25778996)

[VIII. Algorithm Design 31](#_Toc25778997)

[IX. Conclusion 35](#_Toc25778998)

[X. Data Dictionary 35](#_Toc25778999)

# Table of Figures

[Figure 1 Adding a Grade 5](#_Toc25779000)

[Figure 2 Submitting a Grade to the Database 5](#_Toc25779001)

[Figure 3 Performing What-If Grade Calculation 6](#_Toc25779002)

[Figure 4 Viewing What-If Results 6](#_Toc25779003)

[Figure 5 Delete Grade 7](#_Toc25779004)

[Figure 6 Choose a Grade to Delete 7](#_Toc25779005)

[Figure 7 GPA Calculator Button 7](#_Toc25779006)

[Figure 8 Student GPA Screen 8](#_Toc25779007)

[Figure 9 Completed Classes Overview 8](#_Toc25779008)

[Figure 10 Adding a Completed Class 9](#_Toc25779009)

[Figure 11 Success Message 9](#_Toc25779010)

[Figure 12 Modifying a Grade 9](#_Toc25779011)

[Figure 13 Deleting a Current Class 10](#_Toc25779012)

[Figure 14 Adding a Class 10](#_Toc25779013)

[Figure 15 Info for a new Current Class 11](#_Toc25779014)

[Figure 16 Student Progress Overview 11](#_Toc25779015)

[Figure 17 What-If GPA Results 12](#_Toc25779016)

[Figure 18 Viewing Current Courses 13](#_Toc25779017)

[Figure 19 Use Case Diagram 14](#_Toc25779018)

[Figure 20 Context Level DFD 15](#_Toc25779019)

[Figure 21 Level-1 DFD 16](#_Toc25779020)

[Figure 22 Level-2 DFD for Adding an Assignment/Test Grade 17](#_Toc25779021)

[Figure 23 Level-2 DFD for Modifying an Assignment/Test Grade 17](#_Toc25779022)

[Figure 24 Level-2 DFD for Deleting an Assignment/Test Grade 18](#_Toc25779023)

[Figure 25 Level-2 DFD for Deleting an Assignment Grades Course 18](#_Toc25779024)

[Figure 26 Level-2 DFD for Adding an Assignment Grades Course 19](#_Toc25779025)

[Figure 27 Level-2 DFD for Performing What-If Grade Calculations 19](#_Toc25779026)

[Figure 28 Level-2 DFD for Adding a Final Grade 20](#_Toc25779027)

[Figure 29 Level-2 DFD for Modifying Course Info 20](#_Toc25779028)

[Figure 30 Level-2 DFD for Deleting a Finished Course 21](#_Toc25779029)

[Figure 31 Level-2 DFD for Tracking EKU CS Concentration Progress 21](#_Toc25779030)

[Figure 32 Level-2 DFD for Performing What-If GPA Calculation 22](#_Toc25779031)

[Figure 33 Level-2 DFD for Fetch Courses for Viewing Current Courses 23](#_Toc25779032)

[Figure 34 Level-3 DFD for Fetching Different (Assignment) Course Grades 23](#_Toc25779033)

[Figure 35 Level-3 DFD for Calculating Degree Progress 24](#_Toc25779034)

[Figure 36 Level-3 DFD for Calculating Overall Grade and Earned Percentage Points in a Course 24](#_Toc25779035)

[Figure 37 Top Level Structure Chart 25](#_Toc25779036)

[Figure 38 Structure Chart for Viewing Assignment Grades 26](#_Toc25779037)

[Figure 39 Structure Chart for Adding an Assignment Grade 26](#_Toc25779038)

[Figure 40 Structure Chart for Modifying an Assignment Grade 27](#_Toc25779039)

[Figure 41 Structure Chart for Deleting an Assignment Grade 27](#_Toc25779040)

[Figure 42 Structure Chart for Performing What-If GPA Audit 28](#_Toc25779041)

[Figure 43 Structure Chart for Modifying a Final Grade 28](#_Toc25779042)

[Figure 44 Structure Chart for Deleting a Final Grade 29](#_Toc25779043)

[Figure 45 Structure Chart for Adding a Final Grade 29](#_Toc25779044)

[Figure 46 Algorithm Design for Performing What-If GPA Audit 31](#_Toc25779045)

[Figure 47 Algorithm Design for Performing Tracking Degree Progress 32](#_Toc25779046)

[Figure 48 Algorithm Design for Adding an Assignment Grade 33](#_Toc25779047)

[Figure 49 Algorithm Design for Viewing Assignment Grades 34](#_Toc25779048)

[Figure 50 Algorithm Design for Fetching Assignment Courses 34](#_Toc25779049)

# 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.

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.

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

The system is required to give information to EKU computer science students regarding their degree progress, GPA, and grades in current classes. The system tracks the student’s GPA as he/she submits final grades, and allows a student to see what GPA he/she would have if he/she received certain final grades in future classes (that is, this system can perform “what-if” calculations). The system uses the information about completed classes to show a user his/her progress towards the different C.S. concentrations that EKU offers (users may view their progress under the different concentrations, so a user is not “stuck” viewing progress for merely one concentration). Users may submit their grades for assignments, quizzes, tests, etc. in current classes and view their overall grade in the courses. Then, they may ask the calculator what grade they would need on remaining coursework in order to finish with a certain grade overall.

## Functional Requirements

R1. The system shall allow a user to submit grades for completed assignments/tests.

* 1. After the user has chosen a course to view (see R12), he/she shall click the “Add Grade” button.
  2. The system shall create a new row that the user can place the grade information into.



Figure Adding a Grade

* 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.



Figure Submitting a Grade to the Database

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

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

2.1.2. 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.

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



Figure Performing What-If Grade Calculation

2.2.1. 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.

2.3. 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?”

2.3.1. 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.

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



Figure Viewing What-If Results

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

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



Figure Delete Grade

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

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



Figure Choose a Grade to Delete

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

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

3.5. 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.



Figure GPA Calculator Button

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.



Figure Student GPA Screen

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.



Figure Completed Classes Overview

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”).



Figure Adding a Completed Class

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.



Figure Success Message

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



Figure Modifying a Grade

5.2. The user shall click on the “Submit Changes” button (see Figure 2).

5.2.1. 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).

5.3. The system shall refresh the user’s current grade in the course. The grade modification(s) shall be sent to the database.

5.4. The system shall display a success message to the user.

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

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



Figure Deleting a Current Class

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

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

6.3. 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.



Figure Adding a Class

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”).



Figure Info for a new Current Class

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 2) 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.



Figure Student Progress Overview

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.



Figure What-If GPA Results

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 (see Figure 9).

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 (see Figure 9).

11.4. The system shall ask the user to confirm the changes.

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

11.6. The system shall submit the changes to the database and display a success message.

R12. The system shall allow a user to view current classes.

12.1. The user shall select the semester for the class that he/she wishes to view classes.



Figure Viewing Current Courses

12.2. The system shall fetch all the classes for the chosen semester and place them in a dropdown box.

12.3. The user shall select the course from the dropdown that he/she wishes to view grades for.

12.4. The system shall fetch all available grades for the selected course and put them in the jTable. The system shall also calculate the user’s current grade in the course.

## Nonfunctional Requirements

NR1. All buttons must have a purpose

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

NR2. 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.

NR3. Any information in the database should be secure.

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

NR4. The functions of this system should be consistent and reliable.

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

# Use Case Diagram

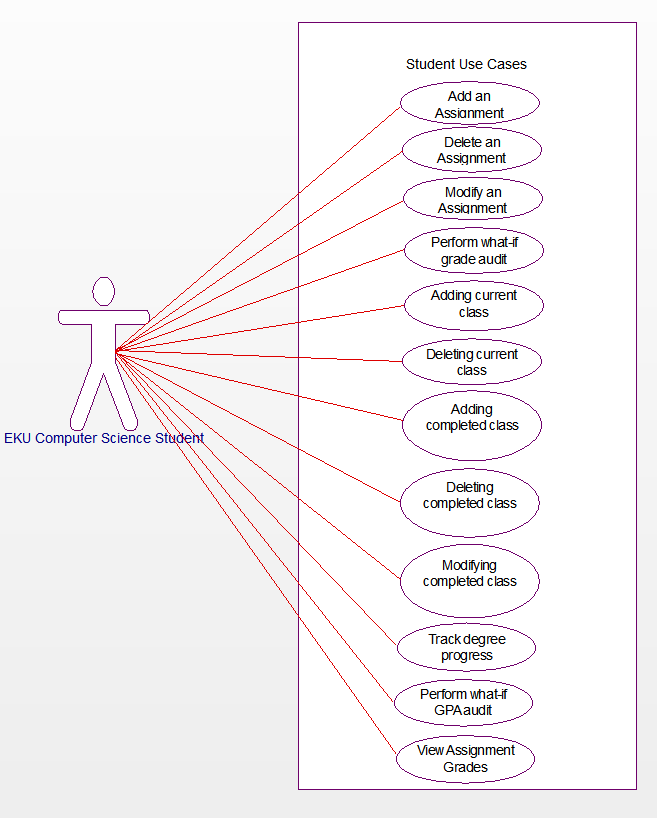


Figure Use Case Diagram

Adding an assignment – Use Case to add a grade for a class to the system.

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

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

Perform what-if grade audit – Use Case for a student to see what his/her grade will be given certain grades on future assignments.

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

Adding a current class – Use Case to add a current class. Once a current class is added, user can submit grades for that class.

Adding a completed class – Use Case to add information about a completed class (such as class name, final grade, etc.).

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

Modifying a completed class – Use Case to modify information about a completed class (e.g., class name).

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

Perform what-if GPA audit – Use Case to see what your GPA will be given certain final grades.

View Current Classes – Use Case for the student to see what their current classes are.

# Data Flow Diagrams

## Context Level DFD



Figure Context Level DFD

Figure 20 shows the data flow from the user (an EKU computer science student) and the system. Users can submit final grades, assignment grades, desired grades, a class to delete, a grade to delete, a projected final grade, a grade modification, and a final grade modification.

## Level-1 DFD



Figure Level-1 DFD

Figure 21 gives an overview of all the interactions between user and the major subprocesses.

## Level-2 DFDs



Figure Level-2 DFD for Adding an Assignment/Test Grade

Figure 22 shows how a user can add an assignment/test grade. First, he/she clicks the “Add a Grade” button and the empty boxes are displayed. Then, input is entered and the user clicks “Add”. The information is validated and then the student’s new grade and percentage points are calculated.



Figure Level-2 DFD for Modifying an Assignment/Test Grade

Figure 23 shows how a user can edit a grade for a current course. See Figure 36 for a detailed level-3 DFD describing the process for calculating a user’s overall grade in a course and the percentage points earned.



Figure Level-2 DFD for Deleting an Assignment/Test Grade

To delete a grade, a student must first confirm that he/she wishes to delete it. Next, it is removed from the database and the screen. Then, the overall grade and percentage points are recalculated.



Figure Level-2 DFD for Deleting an Assignment Grades Course

Figure 25 shows the sequence of operations for deleting a current course. After the user has selected a course from the list of current courses (see Figure 33 and R12), the user shall click the “Delete Course” button. The button changes to “Confirm”, prompting the user to click it again. After clicking the button a second time, the system passes the course’s name to be removed from the database and screen. Another course is fetched from the list of courses. It’s information is displayed and grade calculated.



Figure Level-2 DFD for Adding an Assignment Grades Course

Figure 26 shows how a student adds a new assignment grade. First, the data is validated. Then, after passing validation, it is sent to the database. Finally, the student’s overall grade and percentage points are recalculated.



Figure Level-2 DFD for Performing What-If Grade Calculations

Figure 27 shows how a what-if grade analysis is performed. First, the numbers a student enters (desired final grade and grade a student might make on remaining coursework, as percentages) are validated. Then, the analysis is performed and the student is shown what grade he/she must have on remaining coursework to get a specified final grade, and what grade he/she will have overall, assuming a certain grade is made on remaining coursework.



Figure Level-2 DFD for Adding a Final Grade

Figure 28 shows how a final grade is deleted.



Figure Level-2 DFD for Modifying Course Info

Figure 29 shows how a course’s information is modified.



Figure Level-2 DFD for Deleting a Finished Course

Figure 30 shows how a course is deleted.



Figure Level-2 DFD for Tracking EKU CS Concentration Progress

Figure 31 shows how degree progress is tracked.



Figure Level-2 DFD for Performing What-If GPA Calculation

Figure 32 shows how a user performs a what-if GPA audit. The final grades for some future classes are entered (these are presumed final grades – the user may or may not have actually received such grades and they will not be saved to the database). Then, the information is validated. The user’s future GPA is calculated and the results are displayed in a popup window.



Figure Level-2 DFD for Fetch Courses for Viewing Current Courses

Figure 33 shows how a the system fetches courses to display on the main window.

## Level-3 DFDs



Figure Level-3 DFD for Fetching Different (Assignment) Course Grades

Figure 34 shows how the assignment grades are fetched for a specific class.



Figure Level-3 DFD for Calculating Degree Progress

Figure 35 shows how a student’s degree progress is calculated for a given computer science concentration.



Figure Level-3 DFD for Calculating Overall Grade and Earned Percentage Points in a Course

Figure 36 shows how the overall grade and earned percentage points are calculated. First, the grades for a class are fetched. From there, is just a matter of summing all the weights multiplied by the grades to see the student’s earned percentage points. The overall grade is calculated by dividing the earned percentage points by the sum of weights completed.

# Structured Chart

## Top Level Structure Charts



Figure Top Level Structure Chart

Figure 37 shows an overview of all the main processes performed by this system.

## Lower Level Structure Charts



Figure Structure Chart for Viewing Assignment Grades

Figure 38 shows that, to view assignment grades, the system first fetches them. Then, if some grades are returned, it will sum the weights multiplied by the grades received.



Figure Structure Chart for Adding an Assignment Grade

Figure 39 shows that, to add an assignment grade, the grade’s information is first validated. When that is done, it will attempt to insert the grade into the database. Finally, it will update the student’s overall grade and percentage points earned.



Figure Structure Chart for Modifying an Assignment Grade

Figure 40 shows that, to modify a grade, the information is first validated. Then, the database is updated. Finally, the student’s overall course grade is updated.



Figure Structure Chart for Deleting an Assignment Grade

Figure 41 shows how a grade is deleted. First, it is removed from the database. Then, the student’s grade and percentage points are recalculated.



Figure Structure Chart for Performing What-If GPA Audit

Figure 42 shows how a what-if grade audit is performed. First, the information supplied is validated (the desired grade, assumed grade). Then, the report is done and finally the student is shown what his/her grade will be.



Figure Structure Chart for Modifying a Final Grade

Figure 43 shows how a final grade is modified. First, the system asks the user to confirm the change request. Then, after the user has clicked “Yes”, the input is validated. Finally, a confirmation is displayed.



Figure Structure Chart for Deleting a Final Grade

Figure 44 shows how a to delete a final grade.



Figure Structure Chart for Adding a Final Grade

# Data Store Design

The Assignment table will hold information about specific grades on assignments, quizzes, tests, etc. Each grade will occur in a course. Id is the primary key in the assignment table. The assignment title, combined with the coursID of the course the assignment was taken in, must be unique. The coursID is foreign key to coursID in the Course table. It is in 1NF, as every row contains only atomic values. It is in 3NF as the weight and grade received are not interrelated.

Every course in the course table has a title. It has the number of credits the course was worth (double). The final\_grade is the grade the student received in the course. The requirement\_satisfaction is the specific degree requirement that the course satisfies (such as Core, Gen Ed E1, writing intensive requirement, etc.). Year\_taken is the year the student took the course. Semester\_taken is the semester the student took the course.

Assignment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Title | Weight | Grade | courseID |

Course

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | title | Credits | Final\_grade | Requirement\_satisfaction | Year\_taken | Semester\_taken |

# Algorithm Design



Figure Algorithm Design for Performing What-If GPA Audit

The above figure shows how a what-if GPA audit is performed. This function essentially finds out what a student’s final grade will be, assuming he/she earns a certain grade on all remaining assignments. It will also show you what grade you will need on remaining coursework in order to receive a certain final grade overall in a course.



Figure Algorithm Design for Performing Tracking Degree Progress

The above figure shows how the degree progress is tracked. Each category of courses (gen eds, core courses, supporting courses, and concentration courses) are stored in different lists. The courses a student is currently taking or has completed are taken off of the list (“check-marked” as complete, you might say).



Figure Algorithm Design for Adding an Assignment Grade

The above figure shows how an assignment grade is added for a class.



Figure Algorithm Design for Viewing Assignment Grades

The above figure shows how a student views his/her assignment grades that were submitted. The overall grade and percentage points for the class are calculated based on these assignment grades.



Figure Algorithm Design for Fetching Assignment Courses

The above figure shows how the assignment grades are fetched from the database.

# Conclusion

The goal of any computer science student is to receive good grades and graduate in a timely manner. It is often stressful to keep up with grades in several different classes, and many times students find themselves worrying about what final exam grades they will need to pass a class or receive 79% in a course. At other times, students may wonder what impact final grades in certain classes will have on their overall GPA. This system offers an efficient, robust, and easy-to-use answer to all these issues faced computer science students at Eastern Kentucky University. It allows students to quickly check their grade, GPA, and degree progress, as well as to see what impact certain grades (final and current) will have on their overall grade/GPA.

In this report, I have outlined the capabilities of the system, its context, and constraints. Suggestions on how to improve the system are welcome.

# Data Dictionary

**assignmentGrades** – a data structure holding the student’s grades and weights for specific assignments in a course.

**Assumed grade on remaining assignments** – this is a grade a user assumes she will make on the rest of the assignments/tests in a course. For example, suppose that 25% of the coursework has not yet been graded, and the user would like to see what overall grade she will have if she makes 87% on this remaining 25% of coursework. The grade calculator will show the user what her final grade in the course would be, given this information.

**Class info** – information regarding a course (grade(s), weights, courseName, etc.).

**Completed assignments/tests** – assignments/tests that count towards a student’s final grade in a class. Every test/assignment/quiz score must consist of the name of the task, its weight, and the grade the student received on it.

**Completed/finished classes/courses** – classes that a student has finished and has received a final grade for.

**Concentration** – the computer science specific degree track that a student is enrolled in. As of September 2019, there are five computer science concentrations offered by Eastern Kentucky University. They are general C.S., computer technology, interactive multimedia, artificial intelligence in data science, and digital forensics and cybersecurity.

**courseGrade (double)** – the overall grade a student has in a course.

**courseName (String)** – the course name, e.g. “ENG101”.

**Current class/Current course** – a class that a student is currently enrolled in. These are classes that a student would ordinarily enter information about current grades (grades for assignments, quizzes, homework, etc.).

**EKU C.S. Student/EKU CS Student** – a computer science student at Eastern Kentucky University.

**Final grade** – the grade awarded to a student at the end of a class. This system is setup to handle “A” (for excellent, worth 4 quality points), “B” (for good, worth 3 quality points), “C” (for average, worth 2 quality points), “D” (for poor, worth 1 quality point), “F” (for failed, worth 0 quality points), “S” (for satisfactory, does not count toward student’s GPA), and “W” (for withdrawn, does not count toward student’s GPA) as final grades.

**finalGrade (char)** – the final grade a student has in a course. Example: “C”.

**finalGrades** – a data structure that holds the final grades of finished courses. Each data element will hold the course name and the grade earned in the course.

**Future class** – a class that a student has not yet taken but plans on enrolling in. This term is usually used in the context of the GPA calculator, when a student would like to know what GPA he/she would have in the future, if certain final grades are made in future, unfinished courses.

**GPA** – Grade Point Average. GPA is a student’s weighted average of final grades. Calculated by dividing a student’s total number of grade points received by the number of credit hours attempted.

**gradeInfo** – a data structure containing information about a *single* current grade (it’s weight, student’s score, assignment/test/quiz name).

**initialGrade** – a data structure used when a user creates a new current course. This data structure only appears when a user initially creates a current course. It contains the assignment/quiz/test name, weight, and grade earned.

**overallCourseGrade (double)** – a student’s overall grade in a course.

**Percentage point** – a percentage point is that part of an assignment/quiz/test grade that has been earned by a student. For example: if a student earns 85% on Homework 1, which is weighted at 10%, then the student is awarded 8.5 percentage points.

**percentagePoints (double)** – the data element that represents the “Percentage point” term.

**percentagePoints (double) –** the earned percentage points of a student in a course.

**What-if grade audit/calculation** – this calculation is useful if a user has submitted some, but not all, of the coursework for a class. The user may ask the system what grade he would need on remaining coursework in order to receive a certain, desired grade overall. Additionally, a user may ask the system what grade she would receive in the course if he were to get a certain grade on remaining coursework.