David Goodell
Dustin Michels
Jack Kennelly
Kilian Roberts

Grade Tracker

Project Goal

The goal of our project is to build a grade tracking program that allows students to log assignments and helps them understand how their grades would change based on performance on future assignments or exams. Our intention is for students to be able to use the program to track their performance in their classes and to determine which courses they should prioritize and which they can afford to spend less time on.

The user should be able to use the program to see a visual summary of the assignments they have completed for their classes, as well as any known future assignments. The user should also be able to use the program to determine how well they need to do on future assignments in order to get a particular grade in the course. If we have time, we may also implement a extra functions such as one that allows students to use the program to keep track of how long they studied throughout the term, or better data visualization features, or a way for students to use the program to store digital versions of their homework assignments. The goal of the extra features would be to let the program display more and easier to understand information for the users.

Implementation

Features

- Core Features
    - User can add their current courses, and set up the grading structure for each course
        - A setup screen allows user to customize category types (tests, quizzes, participation, etc) and their weight, as well as map final percentages to the proper letter grades.
    - User can easily record actual and predicted grades in the proper category for a given class
    - Program can calculate current grade based on actual and predicted scores, allowing user to gauge overall performance, strategize, and predict final results

        - Include a slider when inputting predicted grades, allowing the user to change the potential grade and see what happens to their overall grade
    - User can easily adjust predicted values to see different outcomes
- Extended features (Likely will pick and choose as time permits, but will not include all)
    - Create graphs / other visual aides to help people understand their grades
    - A “what grade do I need” button for each uncompleted entry (ie Final Exam) that displays what grade would yield an A, B, C, D, and F in the class.
    - Ability to record courses and grades across multiple terms
    - Ability to export grade info in a visual format (text file, PDF, spreadsheet)
    - Additional data fields for assignments that get entered, such as notes, or document uploads (ie, you can upload a word document of your HW to the entry for that HW.)
        - We could include some optional, numerical fields (like how much time did you spend on this assignment, or how much did you enjoy this assignment on a scale of 1-10, and then offer some charts or statistics correlating this entries to performance.)
    - A “tips” feature where the program will attempt to make useful statements in natural language, such as “You can get in A in the course if you get 85% or higher on the final exam.”
    - Basic security for the data, such as including password protection.

Our program will allow students to enter information on each of their classes, including the grading scale, and the weighting applied to tests and assignments. As students complete assignments and tests and enter the grades into the program, the program will track the history of the student’s grade in the class and display the history for the student. The program should also allow the students to enter theoretical grades for future assignments and see how it affects their projected grade for the course, and it should allow the students to enter a grade and see how well they would have to do on their uncompleted tests or assignments in order to get the grade they want.

If we have extra of time, we can extend the program by adding extra features, like a way for the users to to use the program to record their study habits and display a visual summary that shows both the user’s study habits over time and their assignment grades over time.

Other extra features could include a more intuitive, intelligent UI that streamlines our most common use cases for students. An example of this would be a “What Grade Would I Need” functionality that gives students predicted grades based on a grade-goal for a class. Another approach to improving the user interface would be to add more palatable presentations of user data, such as using a pie chart to show the various weights of their grades for a class. A third functionality would be to build the application out into a tool to help students see patterns in their work by uploading their homeworks or notes.

Design Patterns

We plan to use the model-view-controller design pattern to organize our program. The model will store data on the student's classes and grades, as well as creating the projections of future grades. It should also be able to save a student's data as and retrieve it from a text file. The view should be able to present students with an overview of their classes, assignments, grades, and grade projections. It should also be able to allow students to make changes to the program's record of their grades, add new classes and assignments, and new assignment grades and test grades. Finally, the view will provide students with an easy to use way to generate and alter projections of the student's final grade contingent on performance in future assignments.

We will likely end up using the composite pattern to create classes for assignments and assignment types. The challenging thing about designing assignment or assignment type classes is that some assignment types,such as homework problem-sets, include many small assignments while others, particularly final exams, may include/be a single assignment, while others, such as projects in CS 257 can be thought of either as one large assignment or a series of smaller assignments. Using the composite pattern to design our assignment objects should make it easier for the rest of the program to manage assignments without having to worry about the specifics of what the assignment is, and it should give the user more flexibility when they enter assignments into the program, since they won't have to rely on a small collection of preprogrammed assignment types.

Sketches of GUI

Image 1: Courses. See all current courses and the predicted grade. Easy to add a new courses using (+) button.

Image 2: Assignment categories view. Upon clicking on a course, we move to this screen. Easily see assignment categories and relevant information fields. Colors are used to indicate which values are set up in a categories setting (dark grey), which are computed (yellow) and which the user can edit directly (green). The (+) button still allows user to add a class. The gear icon allows user to change the settings for a course (ie, weight). The purple button with an arrow indicates that a particular category has sub-items, and can be clicked to reveal subitems.

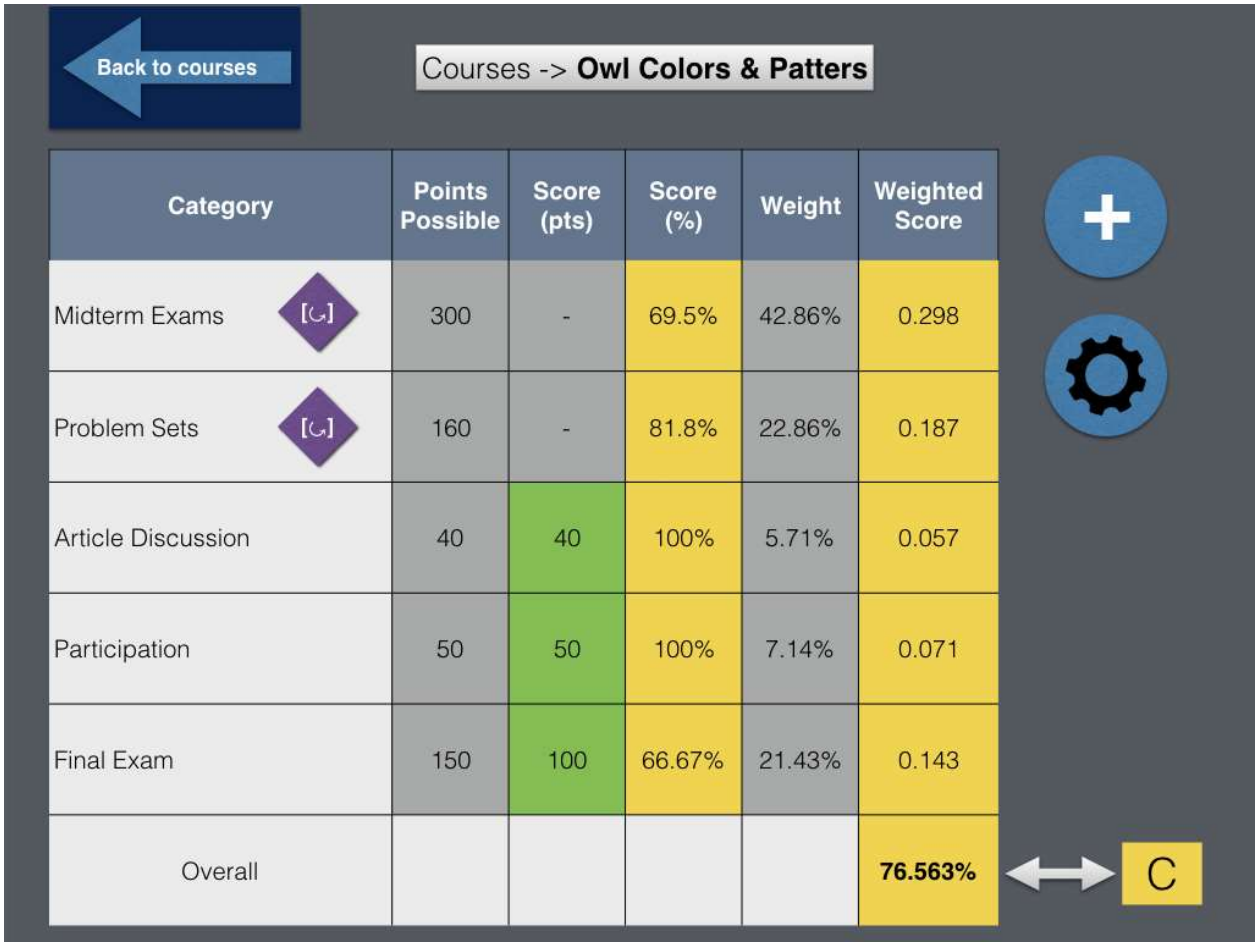
Image 3: Another assignment category view. Shows that this screen can be configured in different ways depending on nature of class to show only relevant fields.

Image 4: Assignment view. Upon clicking the "quizzes" category from image 3, we see the individual quizzes. Again, relevant fields are displayed (and this can be configured), new items can be added, and colors differentiate the types of input fields (preset, calculated, editable.)

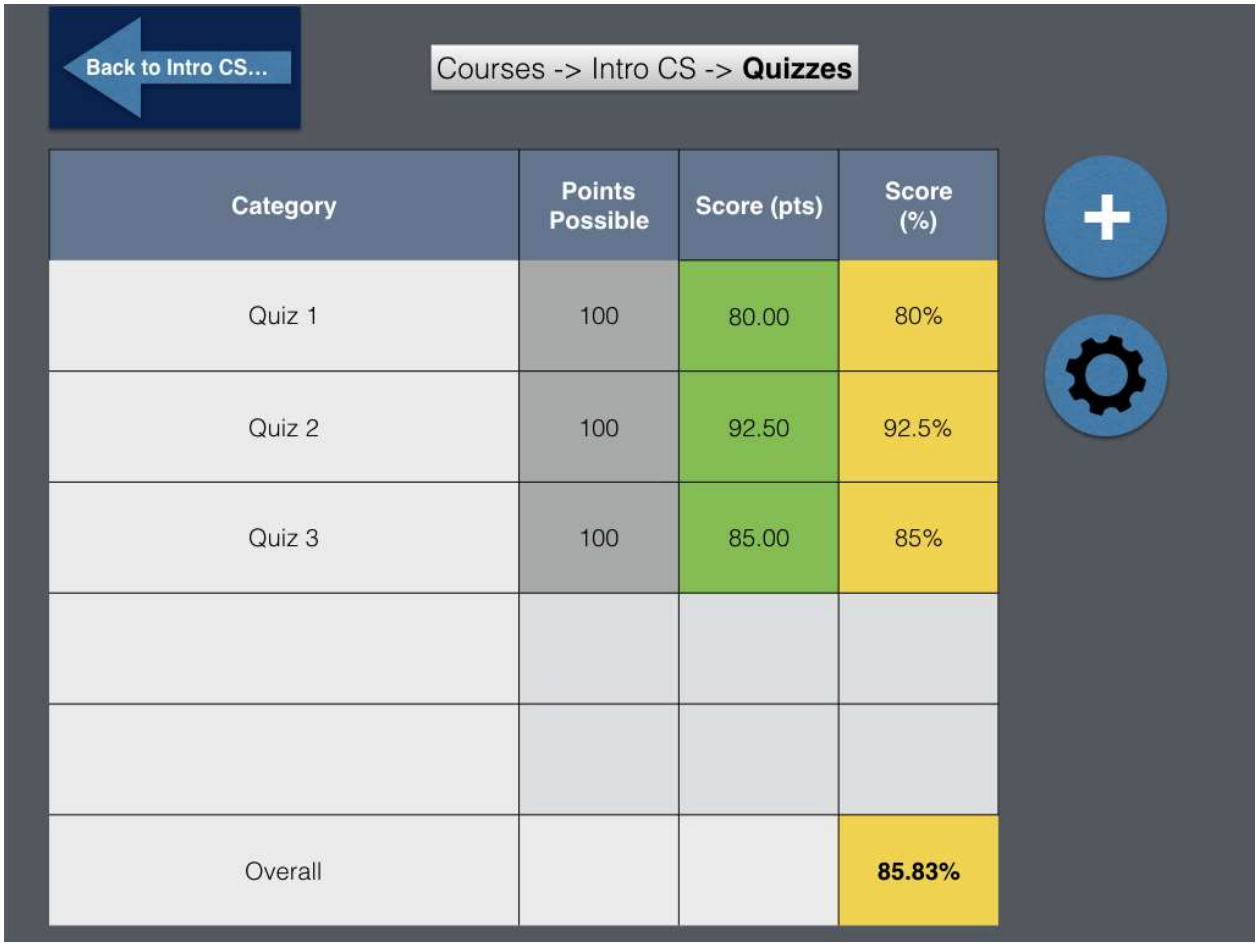
Images 5-7: Setup screens. This images illustrate what it might look like to add a new class or assignment category. A setup screen should pop up, allowing user to enter core info about class. Default values should be pre-filled, where possible. Hitting the gear icon will allow the user to edit this settings once again.

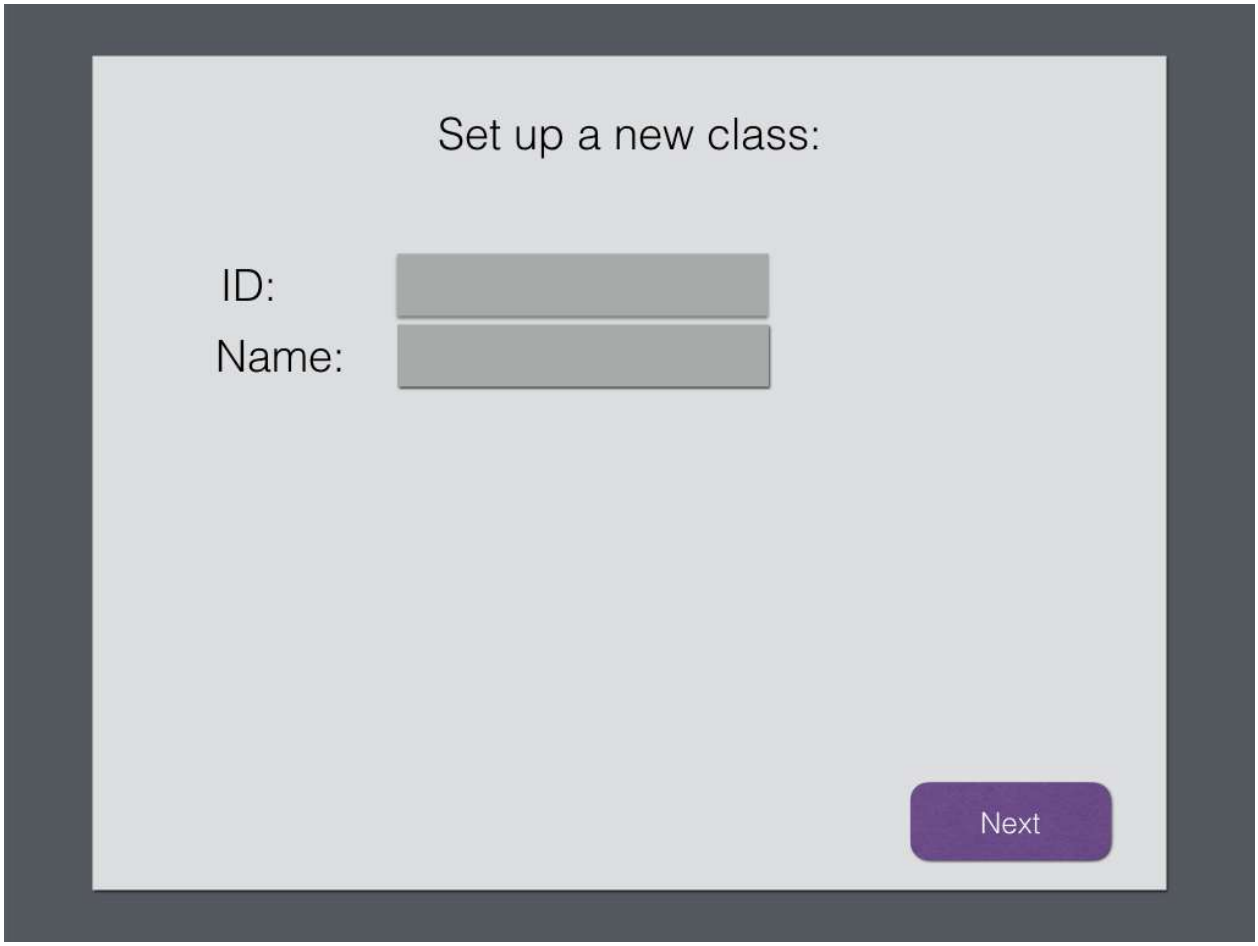
Courses

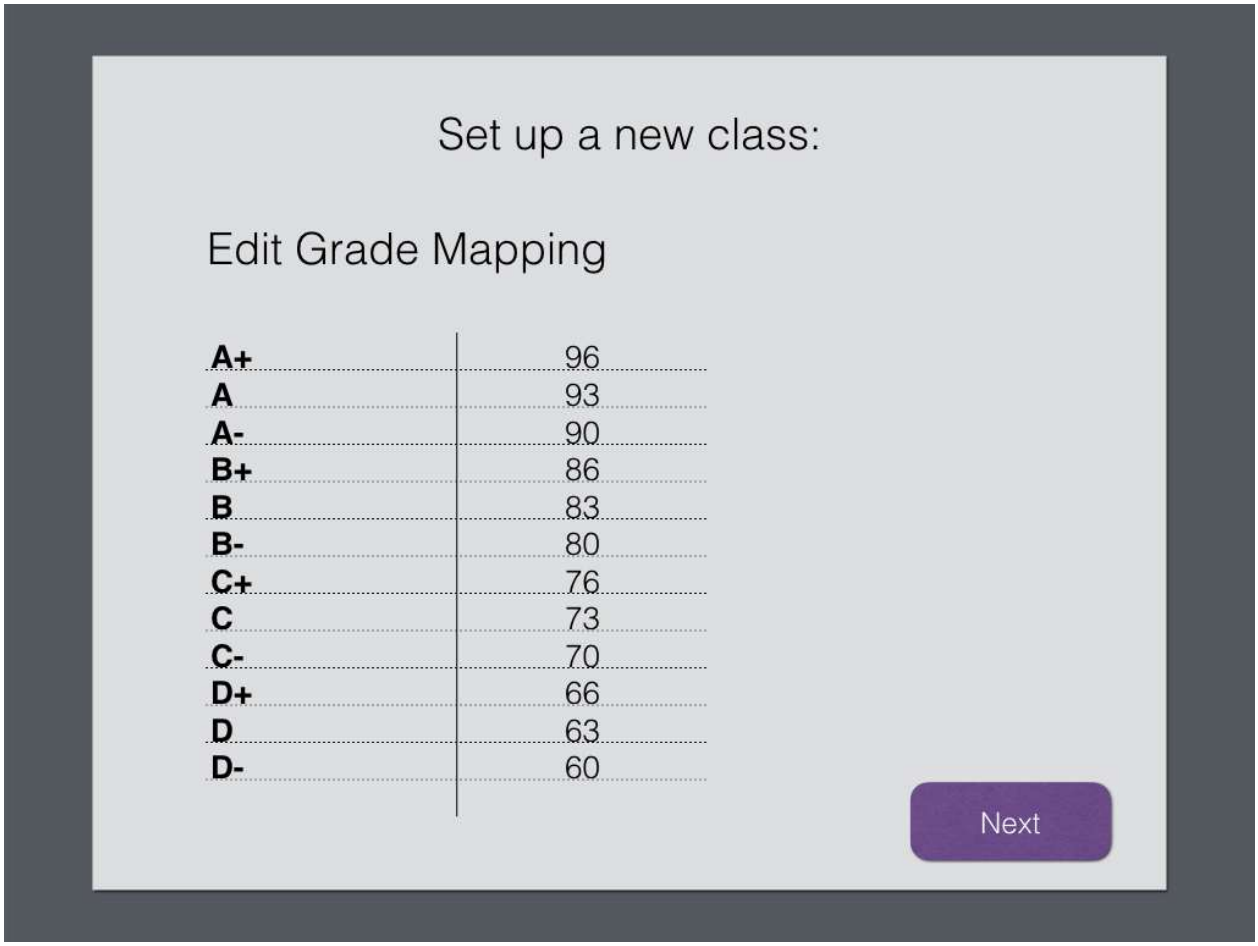
Course ID	Name	Predicted Grade
BIOL.362	Owl Patterns and Colors	C
CS.111	Intro CS	A-
RELG.355	Popes, People, & Pears	A+


Back to courses
Courses -> Owl Colors & Patters
Category
Points Possible
Score (pts)
Score (%)
Weight
Weighted Score
Midterm Exams
300
-
69.5%
42.86%
0.298
Problem Sets
160
-
81.8%
22.86%
0.187
Article Discussion
40
40
100%
5.71%
0.057
Participation
50
50
100%
7.14%
0.071
Final Exam
150
100
66.67%
21.43%
0.143
Overall
76.563%
C


Back to courses
Courses -> Intro CS
Category
Score (%)
Weight
Weighted Score
Assignments
91.67%
60.00%
0.298
Quizes
86.25%
25.00%
0.187
Final
95%
15.00%
0.057
Overall
90.813%
A-


Back to Intro CS...
Courses -> Intro CS -> Quizzes
Category
Points Possible
Score (pts)
Score (%)
Quiz 1
100
80.00
80%
Quiz 2
100
92.50
92.5%
Quiz 3
100
85.00
85%
Overall
85.83%


Set up a new class:
ID:
Name:
Next


Set up a new class:
Edit Grade Mapping
A+ 96
A 93
A- 90
B+ 86
B 83
B- 80
C+ 76
C 73
C- 70
D+ 66
D 63
D- 60
Next