Brian Connerton
Dennis Law
Chris Selagea
Ethan Stewart

Group 8 Milestone 3

Part 1:

Since there are two types of users in this system, the student and the administrator, we must have each perform different tasks. First, we will look at the student's tasks.

A student is to log in using his or her MyID username and password and take the course. After reading the course material, the student then takes the quiz. At this point we can gather a plethora of quantitative and qualitative information. During the course, we will keep a timer to track the amount of time the student takes over the length of the course and quiz sections. The student might want to refer back to previous course material slides, so we implemented a counter to count each time the student hits the "back" arrow. Some qualitative information we gather is by observing the student's body language or facial expressions. We can learn if the student is tired or bored while reading the course material when he or she yawns, or if the student is frustrated when he or she scrunches his or her face or presses buttons forcefully.

An administrator needs to be able to login to the system, import a list of students, add a student, remove a student, search for students, email flagged students, and unflag students. We will time how long it takes to complete each action and how many mistakes the user makes in completing each task. Before giving the administrator these tasks, the he or she will be allowed to explore the system for a short period of time in order to familiarize him or herself with the system.

We will gather data on how the administrators view the system by conducting interviews about the feel of the system after testing is completed. Along with these reviews of the system, we will interview the students and ask them questions regarding the information from the course. We will ask them questions about their knowledge of the resources UGA offers to help find internships and job opportunities.

Part 2:

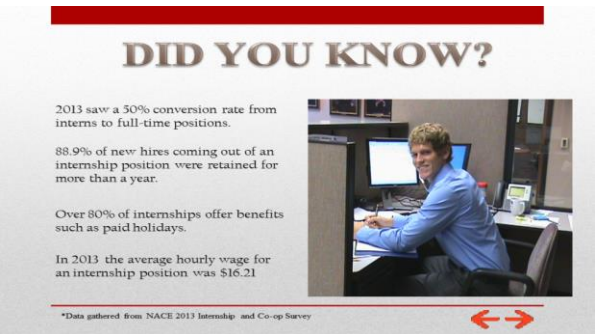
System Description:

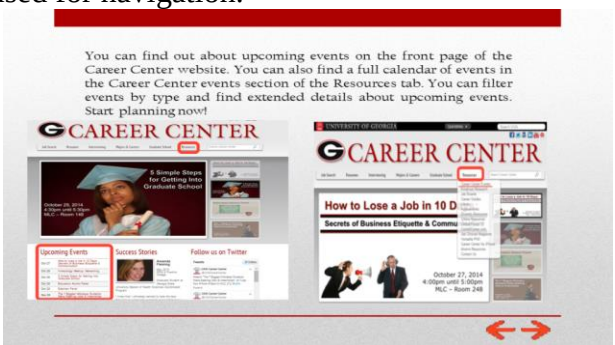
Our system is divided into two primary components. The first is a course that students will take to get educated on the services and resources UGA provides for finding internships/employment. This is a course that can be mandated on a departmental level, and is optional across departments. If a department does choose to make use of the system, then it will become a requirement for students who have completed more than 30 hours. Requirement here means that once a student has reached 30 hours, their student account becomes "flagged," and a hold will be put on their registration for the next semester. So, for example, if Bob finishes the

Fall 2014 semester and his completed hours rises above 30, then the system will flag his account for Spring 2015. The flag here will put a hold on his registration for Fall 2015. This is designed to give the student ample time to get the course done, while still being required to complete it early on in their college career. Since the system is department based, if a student who has more than 30 hours joins a department, they will be treated by the system as if they are a student who has received 30 in the current semester, and the same process as before would occur.

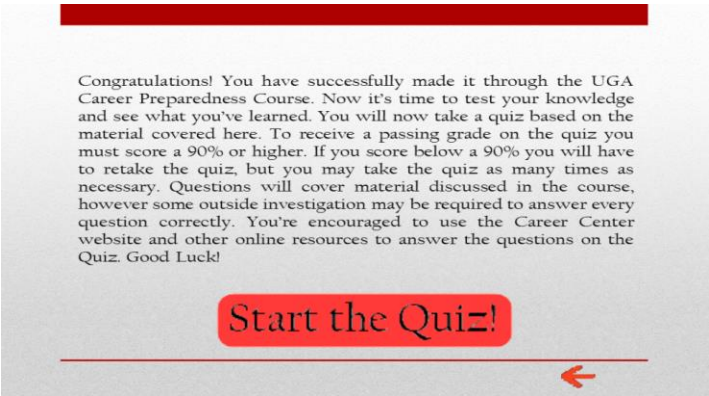
Once a student is registered by the system as flagged, they will receive notifications of that they need to take the course or a hold will be put on their registration for the next semester. The Student would then go and take the course which would be hosted on a website. To begin, the student must log in with their UGA MyID and Password.

Once a student is registered by the system as flagged, they will receive notifications of that they need to take the course or a hold will be put on their registration for the next semester. The Student would then go and take the course which would be hosted on a website. To begin, the student must log in with their UGA MyID and Password. Once it has been verified, the student is taken to a set of slides that contain the content of the course. The slides consist of text and images that guide the student through the material. There are forward and back buttons located in the bottom right of the screen to be used for navigation.

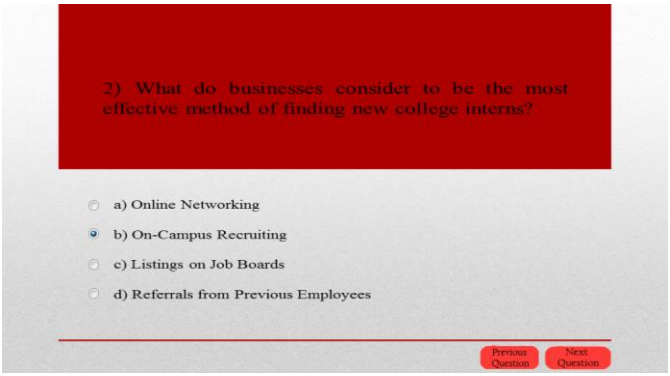




After going through all of the slides, the student is informed of the upcoming quiz and notified that passing the quiz is required to become unflagged and a score of 90% or higher is considered to be passing.



Once the quiz is begun, the slides close out and become inaccessible. The student then starts the quiz, which consists of multiple choice questions about the material covered. The student can navigate through the quiz by using the next question and previous question buttons in the bottom right of the screen.

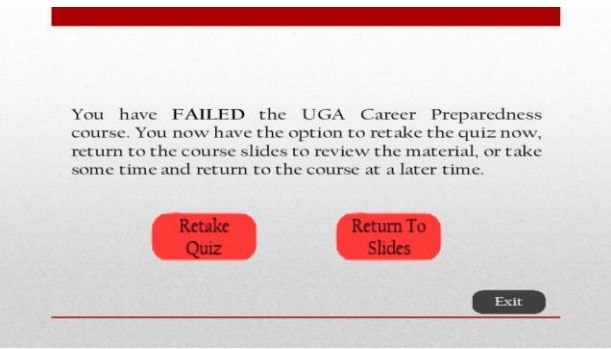


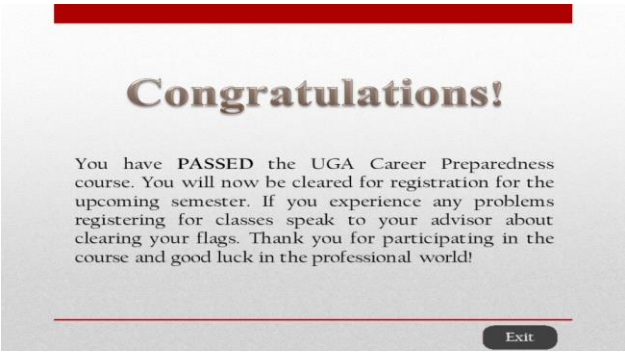
On the last question, the next question button is replaced with a submit answers button. If a student presses this button a dialog box comes up asking if they really want to submit.

An important aspect to note about the quiz is that not all of the questions come directly from the slides. Some questions will only be answerable if the student actually goes to the resources described in the course and explores them further on their own. This was designed to

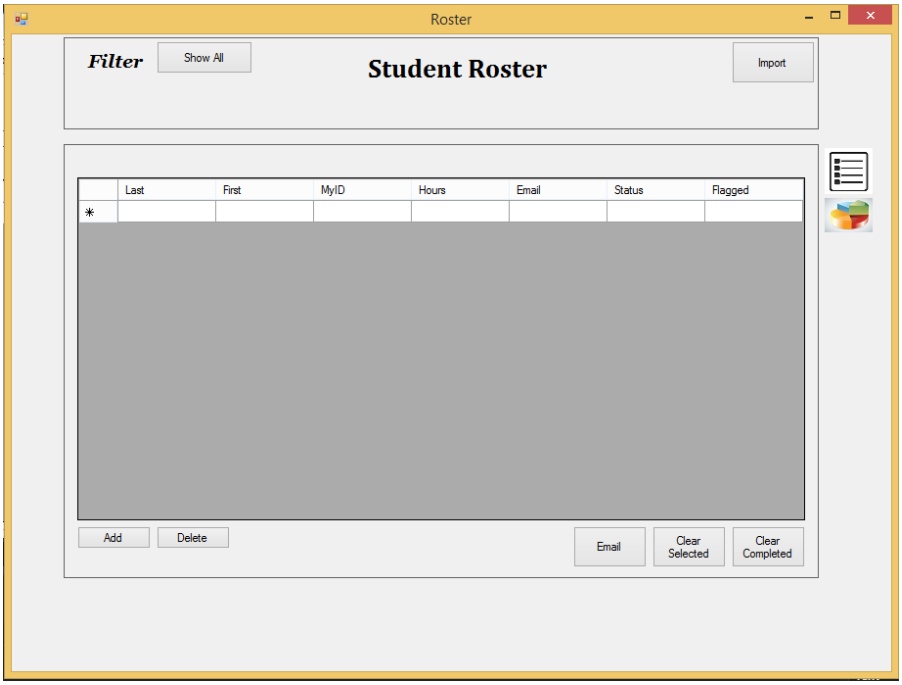
encourage students to go out and actually look at the resources presented, instead of just flying through the quiz and ignoring the material.

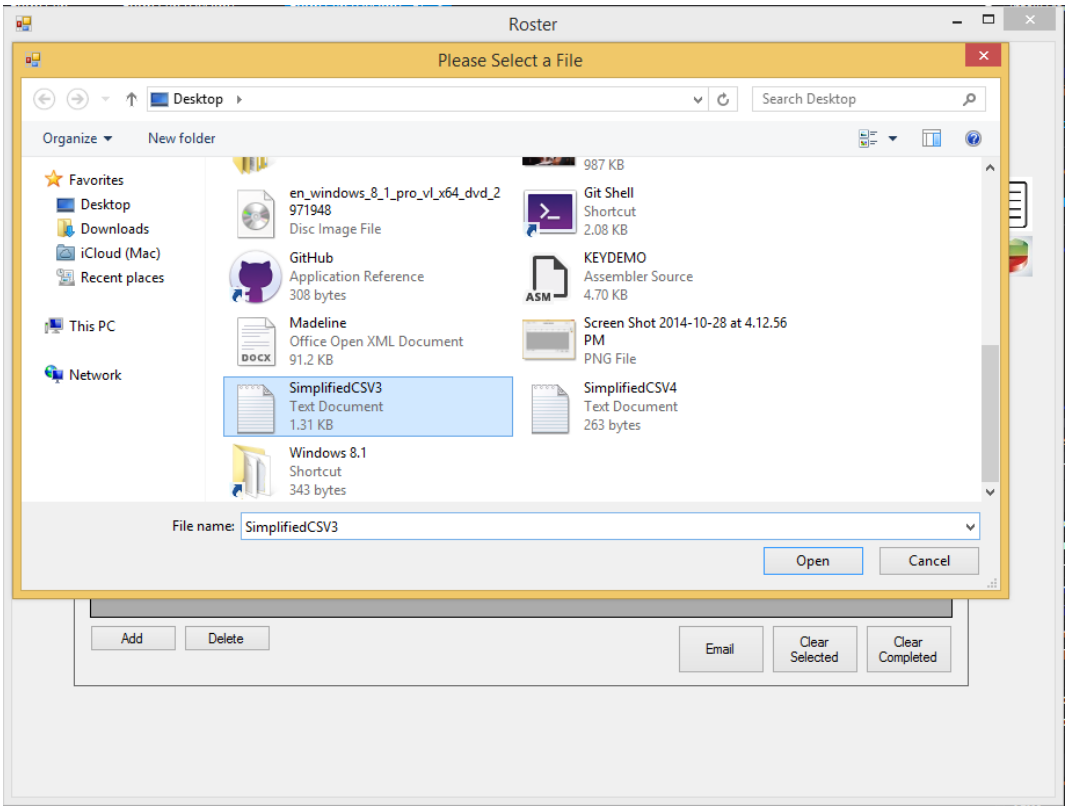
Once their answers are submitted, students will either see the pass screen of fail screen. The pass screen informs the student that they have passed the course and that their flag will be removed. The fail screen informs them that they have failed and gives them the option to retake the quiz now, return to the slides, or exit and return to the course later.



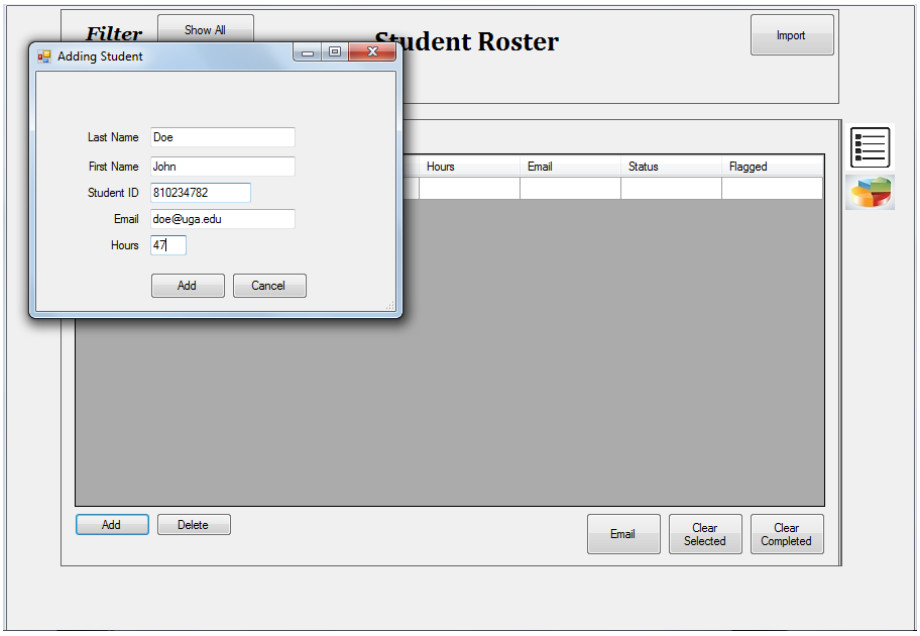


The other main component of our system is the administrator side of the program. This is side of the system is for use by a department faculty member and keeps track of students who have and have not completed the course. When the system is initially accessed for the first time, the user is met with an empty student list section. The administrator can import a student list that they have received from the registrar to populate the list. The system takes in a CSV file containing the student data and creates cells in the list. If a student has more than 30 hours, they are flagged, if they have less than 30 hours, they are not.



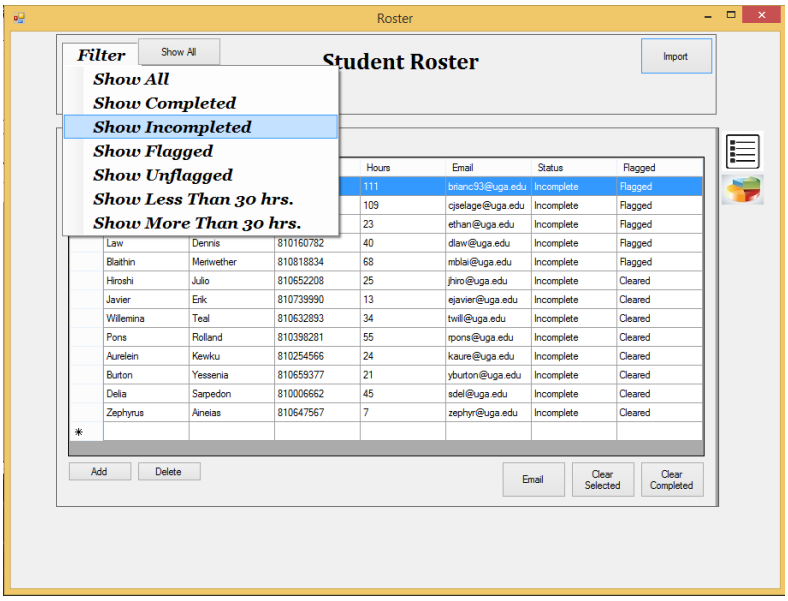


An alternative way for administrators to add in students is the add button, located in the bottom right of the screen. This button pulls up a form for the user to fill in and adds a student based on that data.

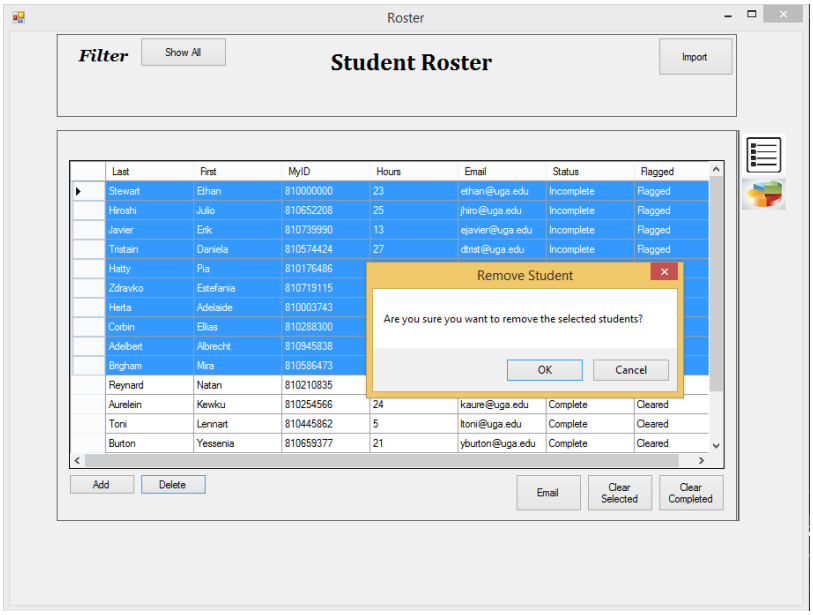


The user also has the option to filter the list based on several parameters, including students who have and have not completed the course, students who are flagged and unflagged, and students who have more than or less than 30 hours completed. There is a button just to the

right of the filter that clears it to show the whole list again. There is also a search feature that allows the user to search for a student by name, e-mail, or student ID number.

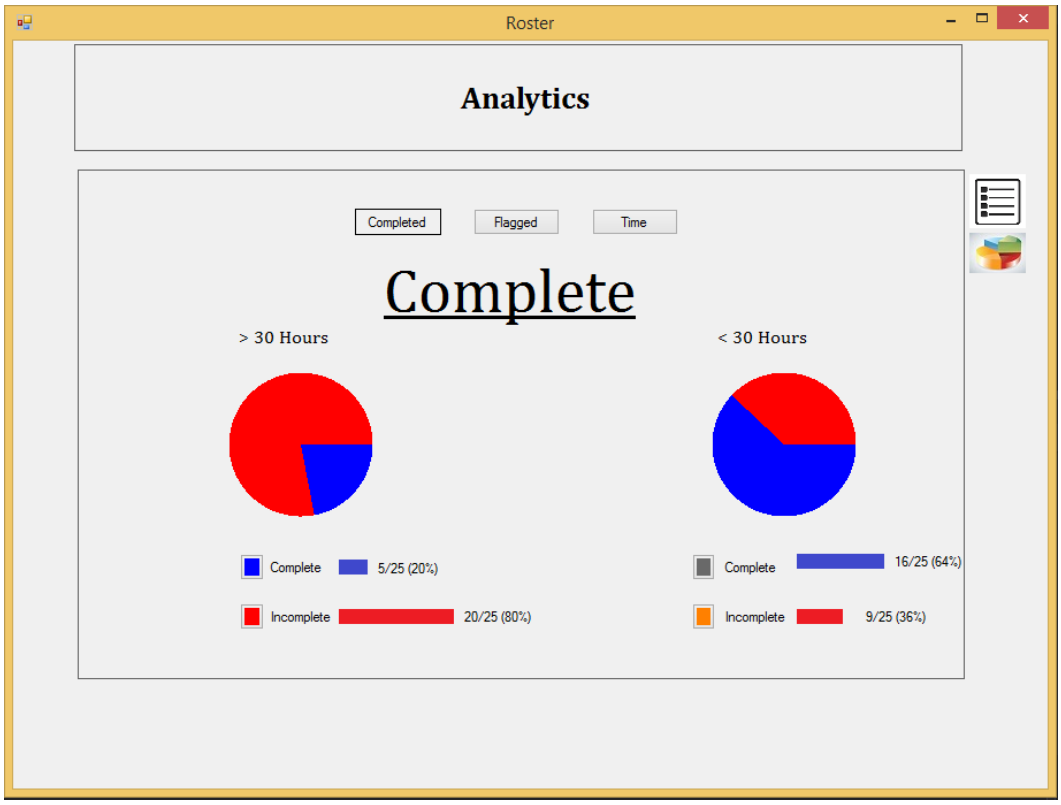


A user also has the ability to select certain students in the table by clicking in their respective cell. Multiple students can be selected by using ctrl+click or by clicking and dragging across the student list. Once a student has been selected, the user has two options. The first is to delete a student from the list. When the delete button is pressed, a dialog pops up to confirm that this is the desired action. This is useful for situations where a student changes majors during the semester and their new department does not require the course. This would be putting a registration hold on someone who hasn't done anything wrong. The other option is to clear the student manually if the system has had an error or they have extenuating circumstances for whatever reason.



The clear completed button in the bottom right unflags all students who have completed the course. The final button on the student list view is the email button, which notifies all flagged students that they must complete the course or a hold will be put on their registration for the upcoming semester.

The alternative view of the system is the analytics view. This view takes data from the student list and displays information on things like number of people who have completed the course and number of people who are flagged. This view allows the user to see visualize the data and get a good read on how many people have completed the course. To access the analytics view the user can click on the pie chart icon on the right of the screen. To swap back to list view, all it takes is a click on the small list icon above it. These two buttons can be used to swap between the screens at will. When first pressed, the analytics view displays data on how many people above and below 30 hours have completed the course. It is displayed as a pie chart and next to the key for the chart it displays the raw numerical data. To swap between the different data metrics the user can click on the three buttons located above the title of the screen.



Part 3:

Implementation Challenges:

Most of the implementation challenges we faced were caused by silly mistakes. For example, for a long time we could not get the delete button to actually delete a student off the list, and we later found out that we were checking one column off the data we were using to identify the student. In another instance, we could not get the graphs to draw on the correct

panel, but only because we forgot we were declaring the graph on the background instead of the panel. One particularly outstanding example was when implementing searching, a function provided by Visual Basic was not robust enough to do what we wanted, so we had to rewrite half of the code of a class.

Another obstacle was that we could not get checkboxes to function correctly, so we had to fall back to selection by row.

The only aspect we would have liked to build, but couldn't was communication between the two applications. Say, for example if a hypothetical student completed the course and passed the quiz, we would have liked for the program containing the course to communicate that that student had completed the course to the administrator program.

Part 4:

Design Justifications:

We decided to use this design over the others because it addressed our problem in the most direct manner while still being a system we could build. Our first idea, integration into eLC was a bust for 2 main reasons. First of all, after looking into the solution, fleshing out the ideas, and speaking with our instructor, we decided that it would not really increase student knowledge. The system would act more as a hub for receiving information about internships rather than informing students about any resources. While something like this would be nice, it doesn't really address our problem and would function basically like a pared down DAWGLink. The other big problem with this was logistics, as eLC was recent taken over by the board of regents and any changes we would have wanted to make to it would be very difficult to actually implement.

Our second idea, the mobile application, was scrapped because it was a solution looking for a problem. There was nothing we found that really suggested that this was wanted, and we didn't have a strong direction we were going with it. It started to become bloated by feature creep and ended up being a reiteration of the already existing Career Center app. We included some new functionality in that design, such as making appointments with Career Counselors, but at the end of the day, there was no real motivation for student to be using the application.

Our final, and current, idea is this course. This design was chosen because it directly addressed the problem of informing students of the resources available and it was possible for us to make this a reality. Obviously, people would not want to take this course on their own, but that is why we have the hold on registration. We decided that this method of compulsion was ideal, since it prevents students from waiting until just before they graduate to do it, but it's not too drastic that not doing it will be catastrophic. Since a series of slide and a quiz are not terribly interactive systems, we also implemented the administrator side of the system. We wanted to design both ends of the system here to show how it could operate in practice. This portion of the project handles student rosters and simulates a system that places hold on registration to force

student to take the course. This part is more interactive, since we have options to add and delete students, send out email notifications, etc. There is also the analytics view, which displays the data in the student list in a more visual manner. It will primarily be used for the administrators to keep track of how many people have completed the course and track data across time.

This design sets itself apart because of its direct solution to the problem. People don't know something, so they take a course on it to learn. Since people probably won't want to take it, we incentivize. Its specialty lies in its straightforward approach to the issue. Lack of focus and abstraction of the problem is what caused us to have trouble with the other two solutions. That is what sets this design apart.