Requirements Documentation for Grader

Team 12: Ruoshi, Jinghu, Chunar

INDEX

Introduction

For our project, our group focuses on, first, having the core functionalities of a grading system, and secondly, introducing easy to use features for the teacher. We include a login screen for security. After logging in, our application displays a selection of semester to add courses in. For a course, we separate it into assignments, students and sections. Each component of the course then has their own summary feature and customizability by the teacher. All of our information is stored inside a MySQL database and queried for depending on the tab the teacher opens. The following document explains our choices in design and requirements for such.

User Requirements Specifications

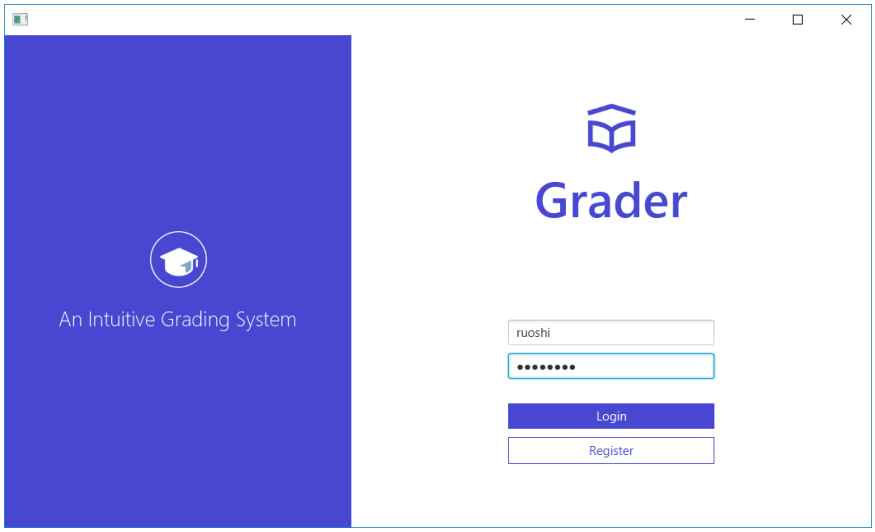
Following are the requirements specified by the user for the project :

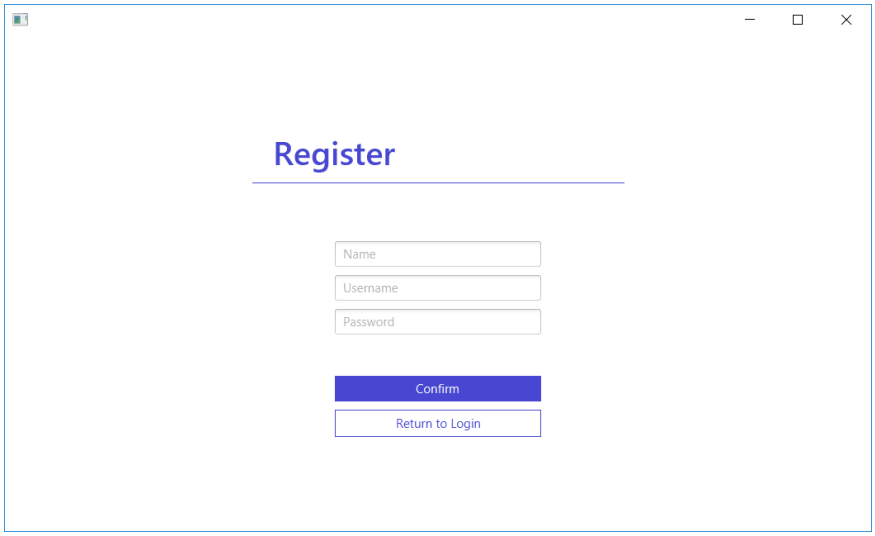
- A system to create and manage data/grades of students more efficiently.
- A security/login page to authenticate user before accessing the data.
- Easy to navigate from courses to assignments to students information.
- Easy to grade assignments for students.
- Pull up statistics about students.
- Pull up statistics about assignments such as average, weights etc.

Workflow of Project

Login & Register:

Users can register for new accounts and use the username and password to login next time.





Main Page:

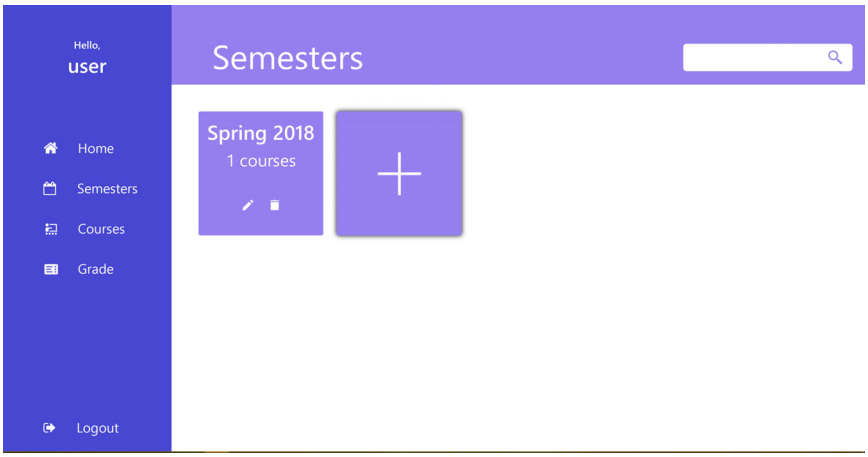
Consists of a dashboard and a display pane. Users can choose different functionalities and enter the corresponding detail page on the dashboard. Information and query result are displayed in the right portion of the screen.

1. *Home Tab:*
   A default page which shows ungraded homeworks and history record.

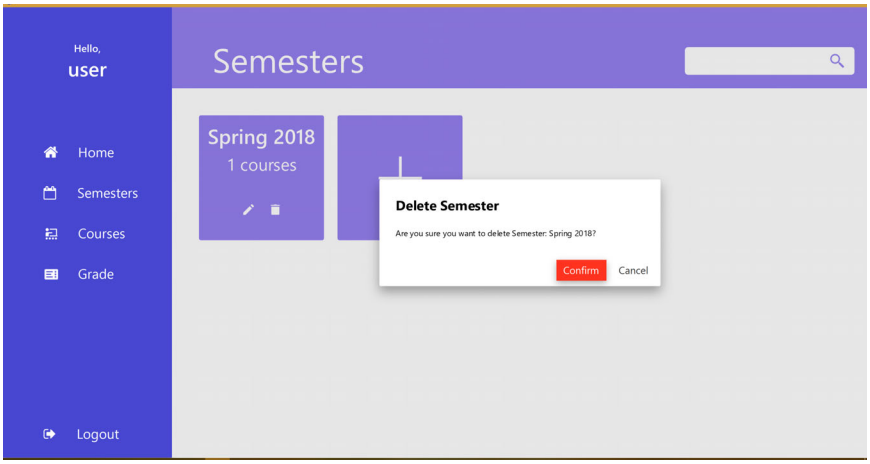
2. *Semester Tab:*

   By clicking on this tab user can search information in 3 steps.

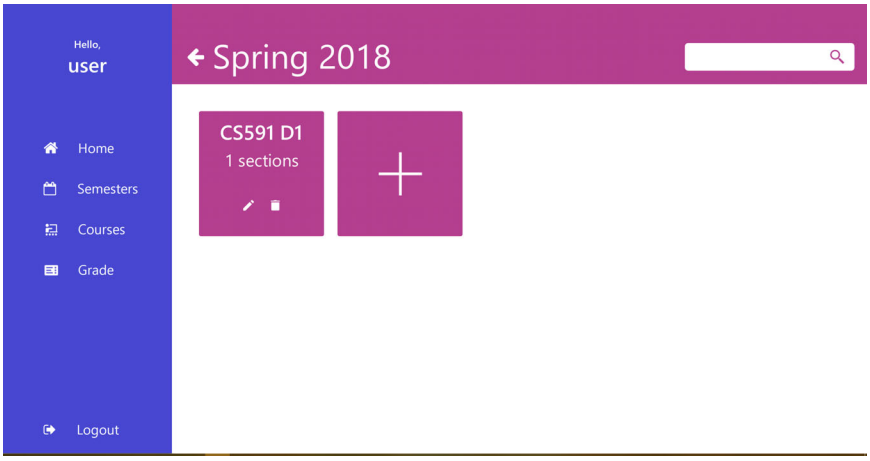
   - Choose an existing semester or add a new one like the following figure.

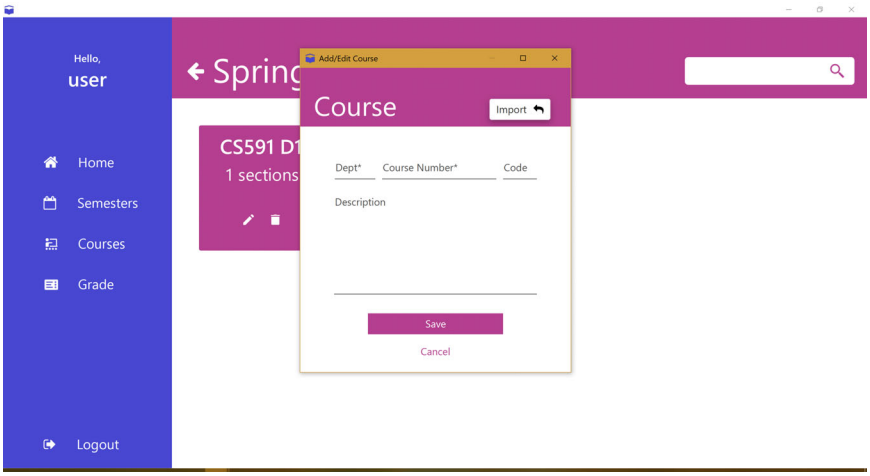


   - Delete and existing semester.



   - In the courses page of above semester, choose an existing course or add a new one.





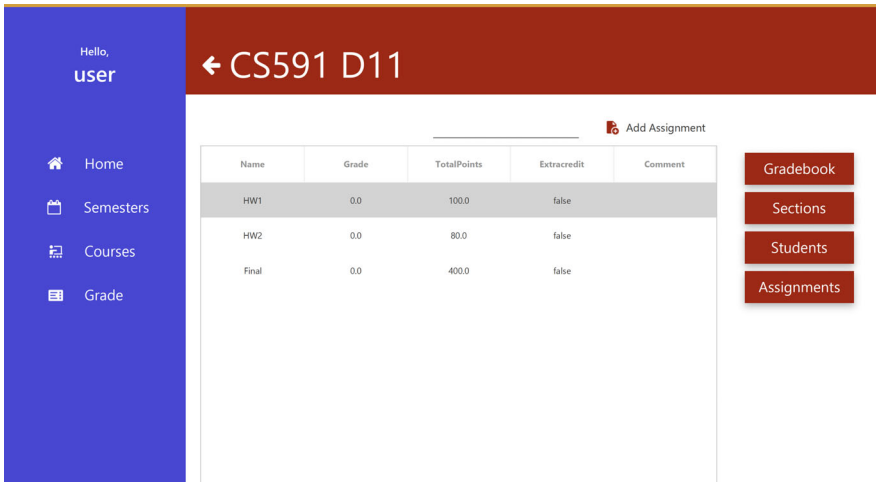
- Access information by choosing different tabs on the navigation bar.

3. *Course Tab:*
   By clicking on this tab user can search information in 3 steps.
   - Choose an existing course or add a new one.
   - Choose a semester from a list which shows the semesters offered above course. Otherwise add a new semester for this course.
   - Access information by choosing different tabs on the navigation bar.

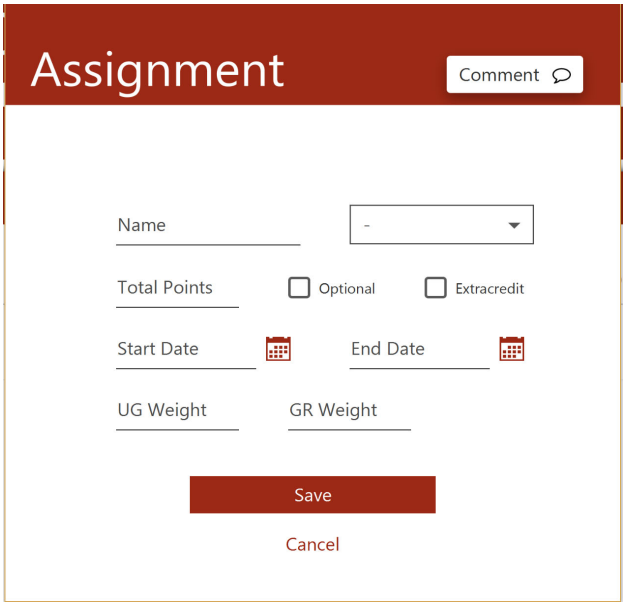
Course Page:

After entering the detail page of course in different ways, users can access information by choosing different tabs on the navigation bar.

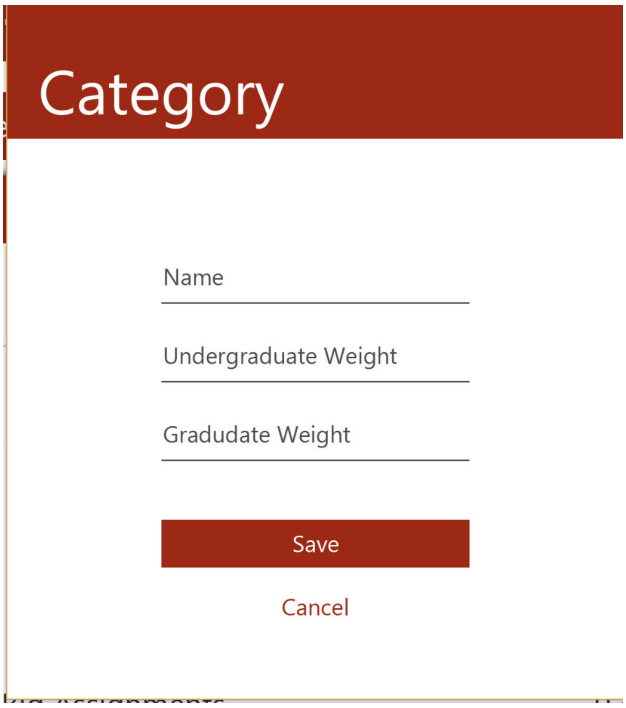
1. *Assignments Tab:*
   A list of assignments in this course with their information such as category, weights and statistics is showed in the display pane.



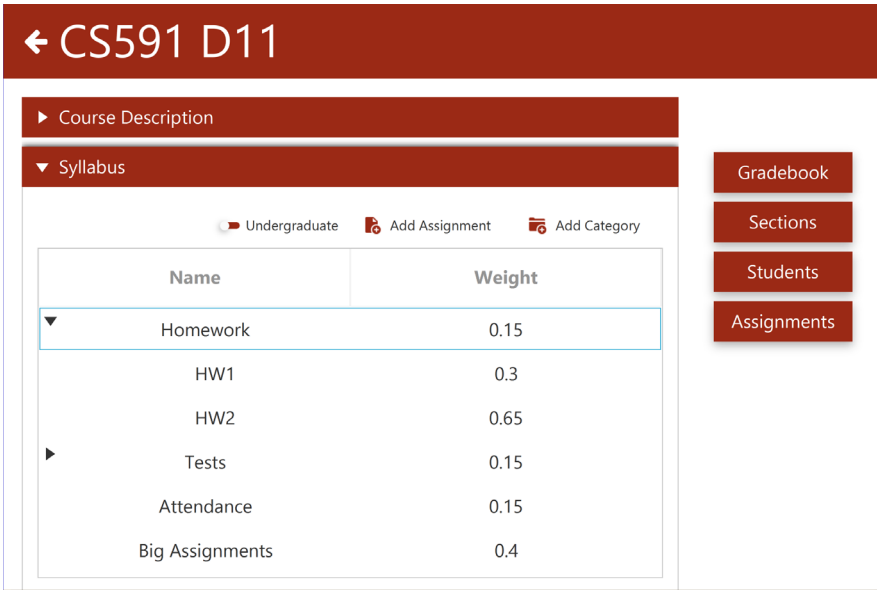
- User can add a new assignment by clicking on the ‘add assignment’ button.



- User can add a new category by clicking on the ‘add category’ button.

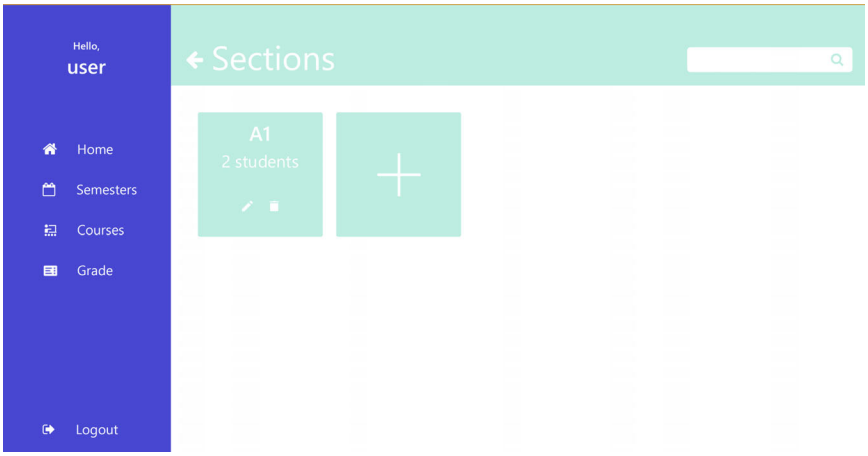


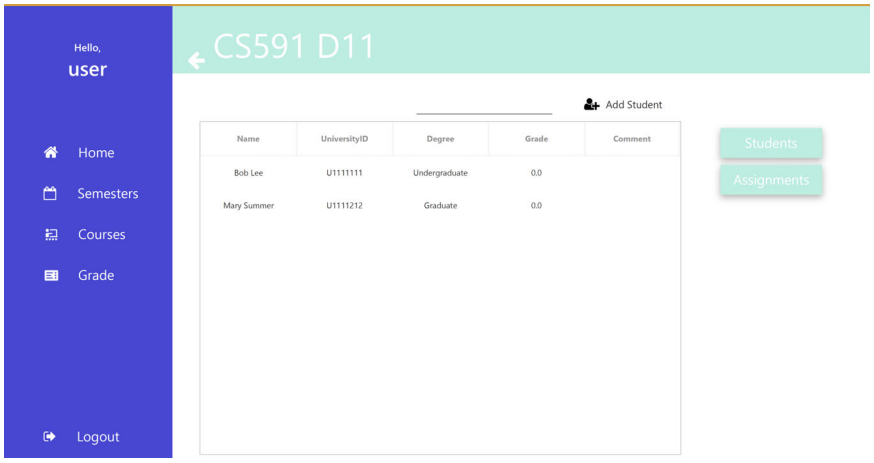
- User can change weights for categories or a courses by editing them.



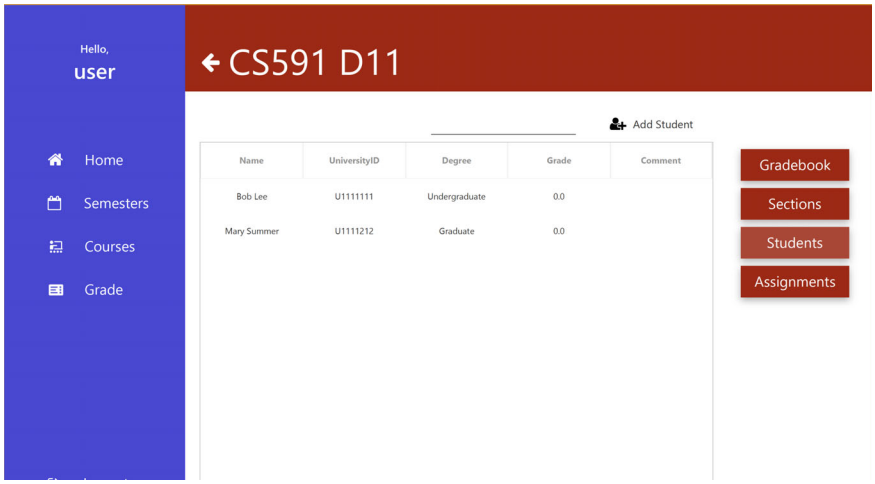
- User can choose an existed assignment to see statistics of all the students. The functionalities in assignment detail page are introduced in next part.
- User can search a specific assignment in this page.

2. *Sections Tab:*
   Sections are created to manage students in a class. It's convenient because if teacher has many graders of one course, she can let every grader take charge of students of each section.





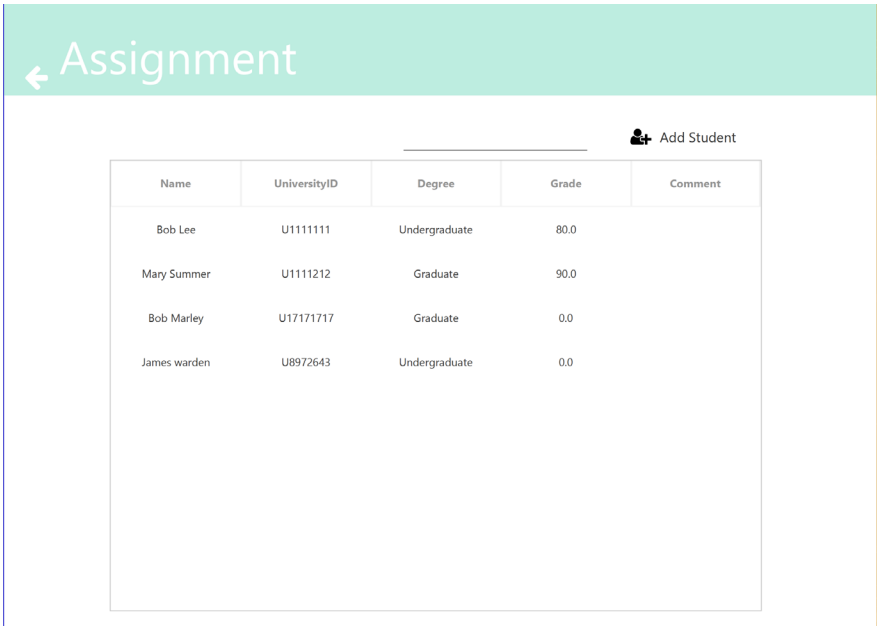
3. *Students Tab:*
   A list of students in this course with their information such as college, number of submitted homeworks and statistics is showed in the display pane.



- User can add a new student by clicking on the ‘add student’ button.
- User can choose an existed student to see statistics of all the assignments.The functionalities in student detail page are introduced in next part.
- User can search a specific student in this page.

Assignment Detail Page:

After entering the assignment detail page from above, a statistic graph and the grade list of students are showed in the display pane.



1. *Tool bar*
   User can search information by clicking the search tab. “Add” button allows user to add student and corresponding score.
2. *Grade List*
   Scores of all the student in this course.

Student Detail Page:

After entering the student detail page from above, a statistic graph and the grade list of this student are showed in the display pane.

1. *Tool bar*
   User can search information by clicking the search tab. “Add” button allows user to add assignment and corresponding score.
2. *Grade List*
   Scores of all the assignments of this student

Functionality

Add a new Assignment

To add a new assignment, the user should navigate to the right course from the course tab on the left-side panel. Click “Add Assignment” button. A window pops up and the user has to enter all the relevant information and the assignment is created.

Categories

Users can add new categories by click on the “add category” button and enter the properties of this category. When a new category is created, teachers can add new assignments for them or add existed assignments for them. For the former, users can choose the category and click on the “add assignment” button to add assignments. For the later, teachers can click “Add Assignment” button and just choose the right category for it in the choosing box.

Sections and Students

After adding assignments and make the assignment well formed. Users need to add students to the course and record their grade for that course. Sections are created to manage students in a class. It’s convenient because if teacher has many graders of one course, she can let every grader take charge of students of each section. When users enter the course page, they can click on the “add section” button and enter the information of this section. If they want to add students for sections, they can click on the section line and enter the section page. The only thing to do is clicking on the “add student” button in this page.

Grades

Now users have semester, course, assignment, section and student. What they want to do is giving grade for students that take courses. Users will go back to the course page and they will choose a specific assignment. When they enter the assignment detail page

they will see the students that submitted this assignment. And they can click on “grade” of every student and just enter the score of them.

Implementation

View - UI

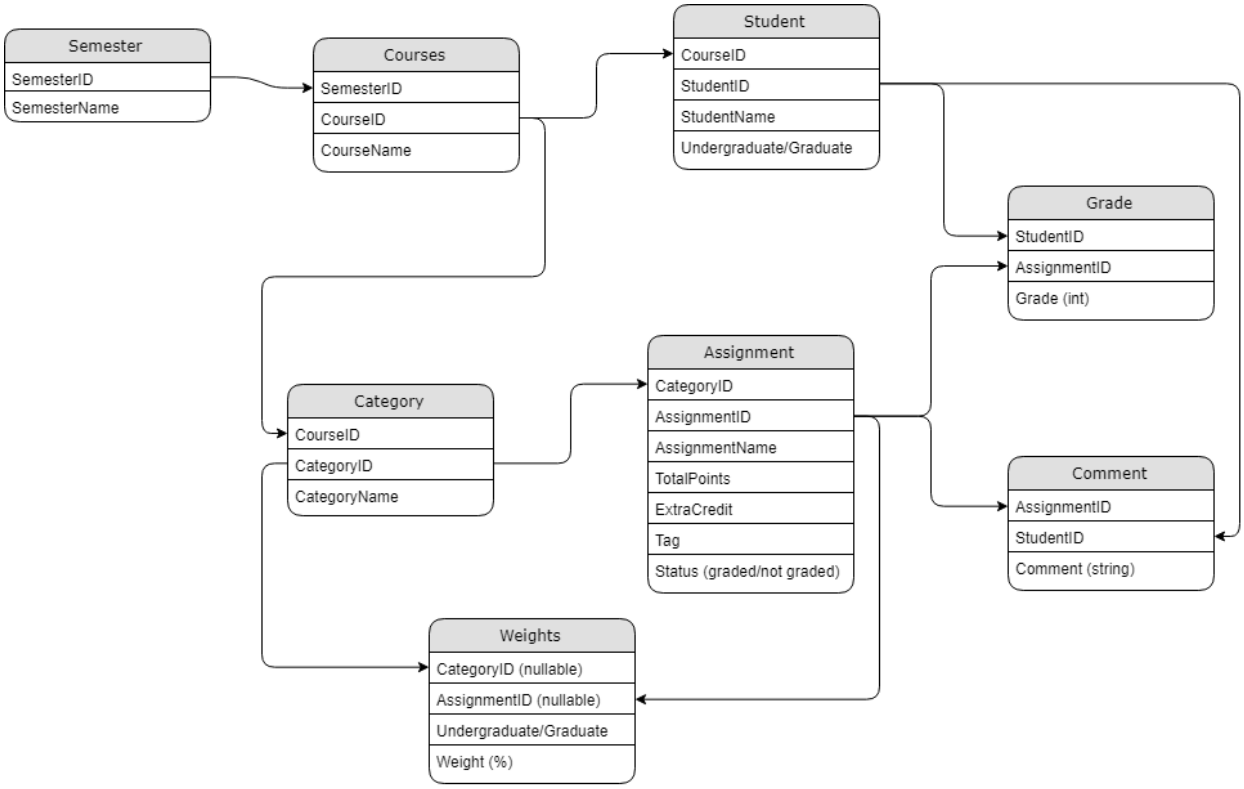
We designed an user friendly interface by using JavaFX for the grading system according the design before. One of our main goals was for the user to have an actual experience with the program rather than feel like he or she is staring at a spread sheet. We planned around with colors as well as modern styles to leave a lasting impression on the user.

To do this, we first packaged our code in for reusability. This is a concept that’s everywhere in design as themes and patterns commonly repeat. We very strictly followed the ideas of Modularization when possible.

Modularization not only cleans up our code by also makes it incredibly easy to make general themes for the user. We made custom components such as the TileButton and PlusButton you saw on many of our application’s screens. Besides components, there was also a lot of reusability in our pages and layouts. We split up the different pages into their functionality (either add, view, or delete). This led to many of the pages being similar, constituting an use case of inheritance. Each page will be encapsulated with their own object. Then other pages can call upon the page to load their information when needed. Navigation will happen by switching pages and through a singleton class called Router. This keeps track of the backstack and adds new panes into position.

Compatibility is another feature we kept in mind. Our users may use this system in different window sizes and on different desktops, so we use gridpane to ensure our UI gracefully adapts for each size of screens and windows. Furthermore, we use ScrollViews and other flexible components to build an responsive and *lively* user interface.

Database



For our Database, we relied upon MySQL in addition to the provided JDBC package to keep our persistent information.

Future Work

- Finalized gradebook to display all of the assignments that need grading.
- Sort feature that allow users to sort the list by scores.
- Magnified search feature to search all courses and semesters.

We chose to stub some features in the last week due to the time constraint. However, if given more time, the above bullets are our top three we would finish first. The gradebook feature was something we came up with early on. It would be a table with Assignments as columns and student's as rows (like) a gradebook. However, the implementation of this was just extremely difficult as it meant a lot of intertwined queries and expanding tableviews. However, it is something that we would want to work out.

What We Learned

What we learned during this time of development, if how much you still don’t expect regardless of your amount of planning. This project seemed very manageable when we first started and our progress was looking really great. However, when it came down to merging the UI and backend together, it was brutal torture. Merging these two seperate pieces of code together was not only lengthy but confusing. JavaFX packages would often conflict with some of our naming conventions and result in more bug bashing. However, through this progress, we believe we learned to plan better; to expect the unexpected. Although we did have this one week for bug fixing, it was definitely not enough to get rid of all of them.

Conclusion

This project was very difficult but also rewarding in forcing us to think in an object oriented manner. Planning the complex object and data relationships was not an easy task. More so, we learned to both respect the programming techniques you taught us in class as it carried us through some of the development.