LMS PROJECT

Software Engineering

Alex Anthony

Juan Reyes

John Baer

Lucy Tea

Table of Contents

1 Intro

1.1 Purpose

1.2 Model

1.3 Scope

2 Description

2.1 Goals

2.2 Functions

2.3 Use Cases

2.4 Operating Environment

2.5 Design Constraints

3. Interface Requirements

3.1 User Interface

4. System Design

4.1 UML Diagram

4.2 Use Case Design

4.3 Database Design

5. Team Coordination

5.1 Github

5.2 Other Communications

6. Software Product Management Plan

6.1 Software Product Management Plan

1. Introduction

1.1 Purpose

The purpose of this document is to describe our software with unique details under the long process of software development and the software engineering process that our group has embarked into ultimately developing a user-friendly Learning Management System desktop application.

The various methods and strategies used in the project have guided us to fully develop a learning management system program that varies in its utilizations in according to the user.

1.2 Model Used for the Project:

Our project utilized the Democratic approach, with no designated leader. Our project positively exploited the organic ability for various individuals to shine in their respective fields of expertise.

Additionally, our project used a combination of the Rapid Prototype life cycle model and the Iterative and Incremental life cycle model. The model was used due to the nature of the goals within the limited scope of the project itself. Specifically, this models' goal was to build a rapid prototype to let the client and future users interact and experiment with it. Once the initial prototype was created, then we were able to test certain features and make adjustments where needed to fit the clients needs, and also to fit any other constraints such as time, working knowledge of the programming language, etc.

1.1 Scope:

The scope of this project was not only to improve our skills as future computer programmers, but to also to teach us how to work together on major programs and projects with others and to learn the software engineering process throughout the life of a product.

2. Project Description

2.1 Product Goals

The Goal for the manufacturing of the software has been to develop a learning management system. The functionalities of this system would work under the umbrella of a desktop application. The program would purposefully bifurcate the targeted users into an administrator and a student. Each who would have their respective and limited possibilities delineated by various levels of access, based on the appropriate responsibilities.

2.2 Product Functions

The main function is to pragmatically let students and administrators manage the sequential and tiresome bureaucratic reality of test-taking into a user interface user-friendly way of managing grades. The Administrator would be able to without restriction, add, delete, and update the student as well as his grades. The Student on the other hand, will be able with some restrictions to see his grades, and his Grade point Average (GPA).

1) Log-In:

1.1) If User is Admin: Can log-in to edit students and courses

1.2) If User is Student Can log-in to view grades courses and GPA.

2) Add-Student

2.1) Once logged in only the admin can add a student.

3) Remove Student:

3.1) Once logged in only the admin can remove a student

4) Update Student:

4.1) Once logged in only the admin can update a student.

5) View Students:

5.1) Once logged in, the Admin can view all the students and their respective grades.

6) View Grades and Courses:

6.1) The user can view his exam grades and courses.

7) View GPA

7.1) The user can view and calculate his GPA of a particular course.

2.3 USE Cases

STUDENT: View Student; View Course; View GPA; View Grades

ADMIN: View Student; View Course; View GPA; View Grades; Calculate GPA; Modify Grades; Modify Student; Modify Course

2.4 Operating Environment

The Software project was created by the utilization of Visual Studio as a tool to write our program in the C# language, with the interface WPF, to ultimately store the data in a Microsoft SQL server. This means that our product can only be operated in Windows Machines only.

More Specifically

- Operating System: Windows
- Database: SQL
- Platform: .NET/C#

2.5 Design and Implementation Constraints.

1. Ideally the Database would be centralized.
2. Limited SQL knowledge

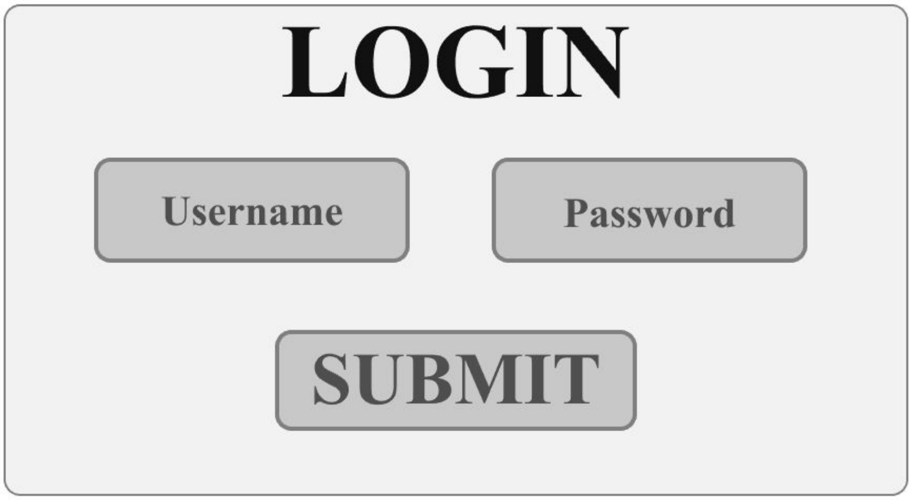
3. Interface Requirements

3.1 User Interface

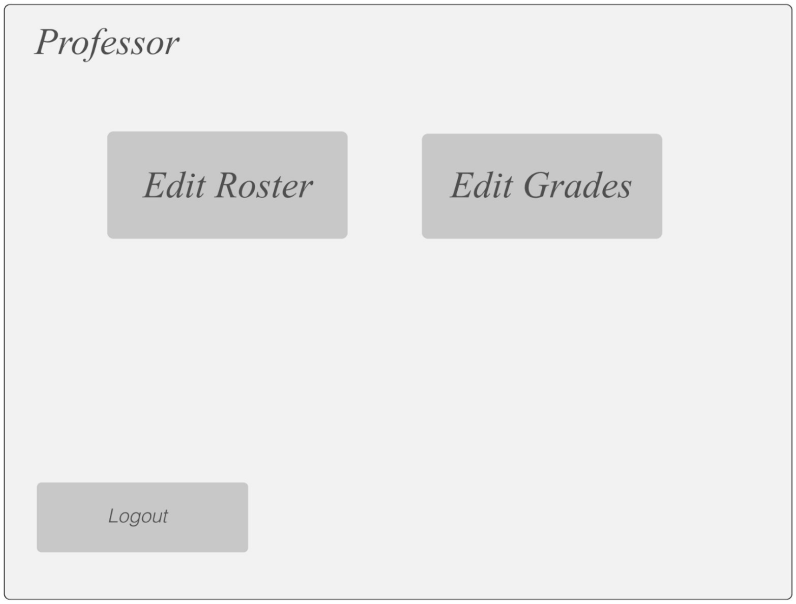
For the user interface, there simply needed to be an professor view (admin), and a student view. The student view only has the means to see his grades, classes, and averages, while the professor has the ability to view grades, view student IDs, view/add/delete from the class roster, and view/add/delete/update an exam. Then, once those changes are made. the student can view the updated changes.

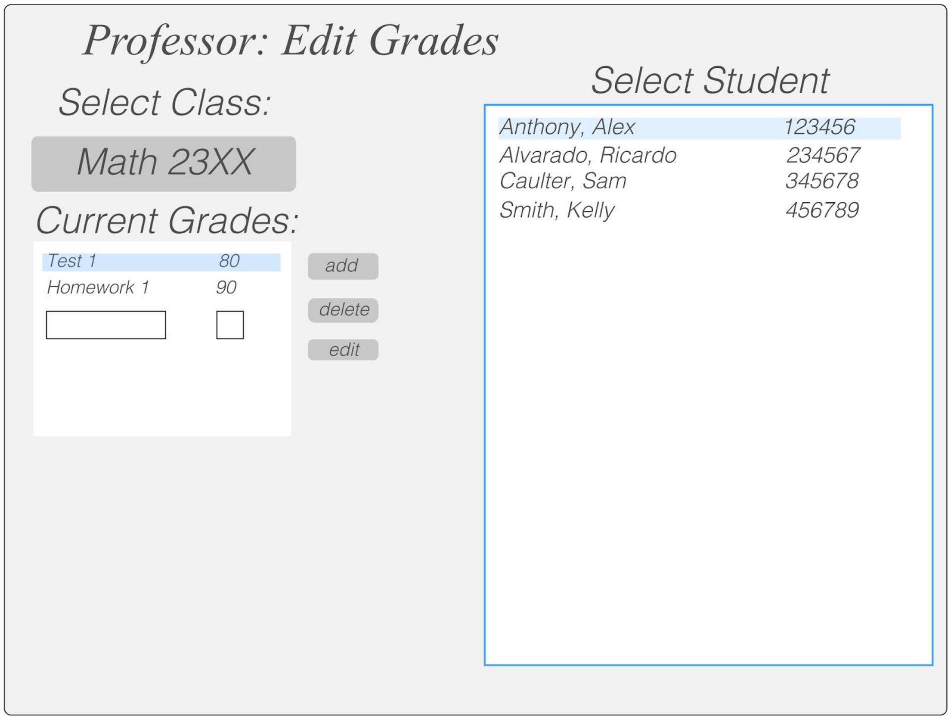
While in the design workflow, in addition to working on the the logical design, we also planned out how we wanted our user interface to work and perform.

Below are the mock ups for each of the views:



Professor dashboard:





Professor: Edit Grades

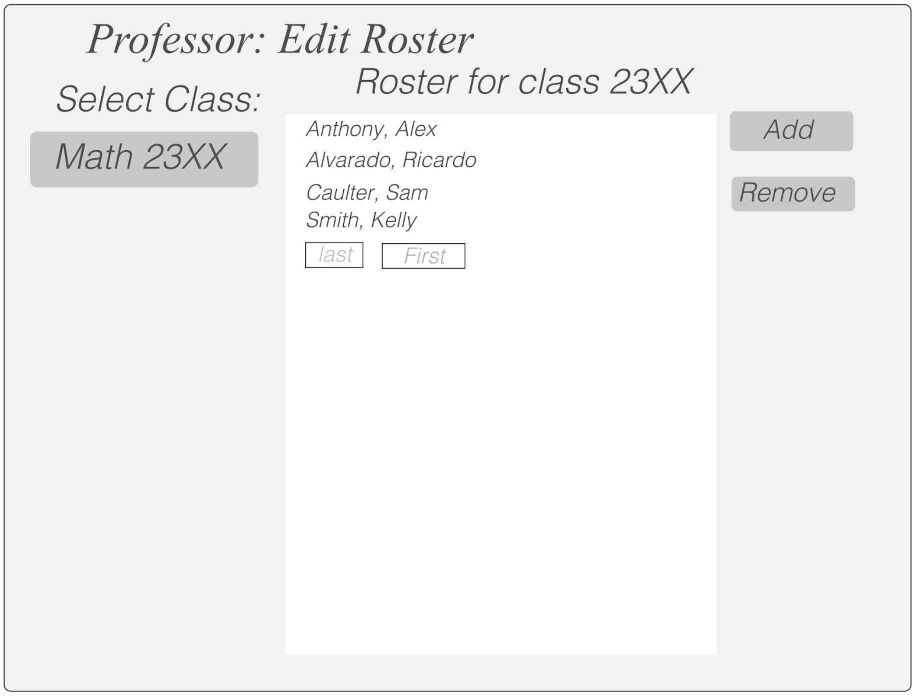
Select Class:

Current Grades:

Test 1	80
Homework 1	90

Select Student

Anthony, Alex	123456
Alvarado, Ricardo	234567
Caulter, Sam	345678
Smith, Kelly	456789



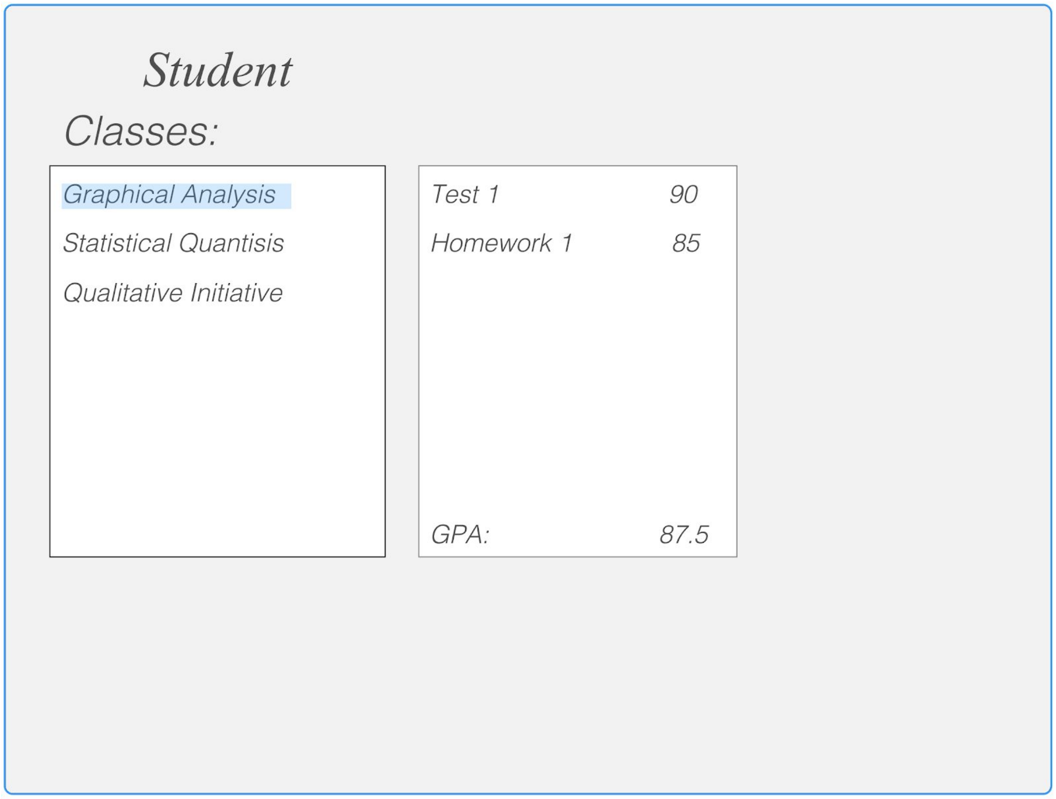
Professor: Edit Roster

Select Class:

Roster for class 23XX

Anthony, Alex
Alvarado, Ricardo
Caulter, Sam
Smith, Kelly

Student Dashboard:



Student

Classes:

Graphical Analysis

Statistical Quantisis

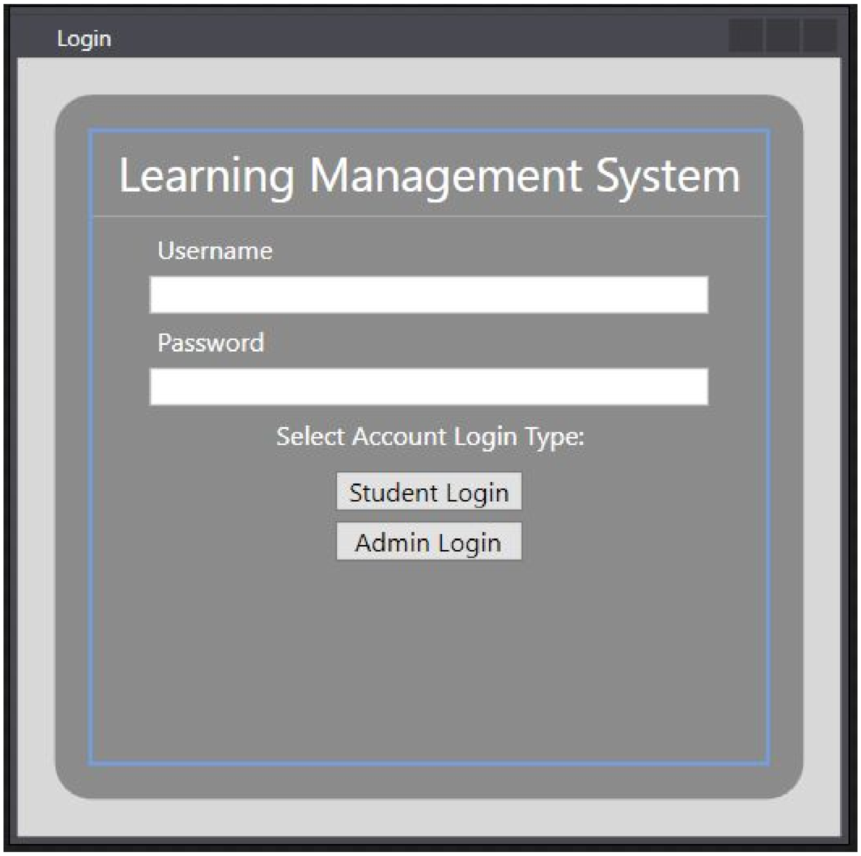
Qualitative Initiative

Test 1	90
Homework 1	85
GPA:	87.5

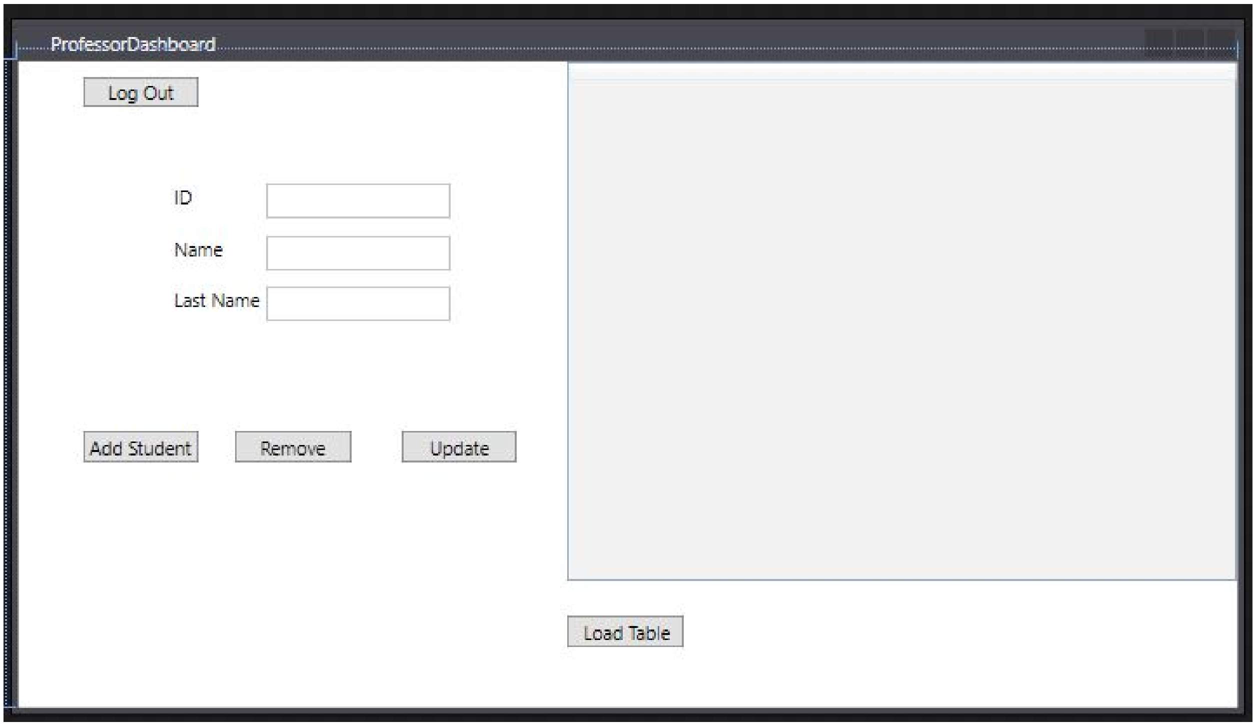
Now, of course, the mock ups would be the fully polished version of our program, but due to time constraints and lack of expertise in the area of GUIs, we had to revise our design and functionality to meet the time limitations of the project.

Here is the result of our work:

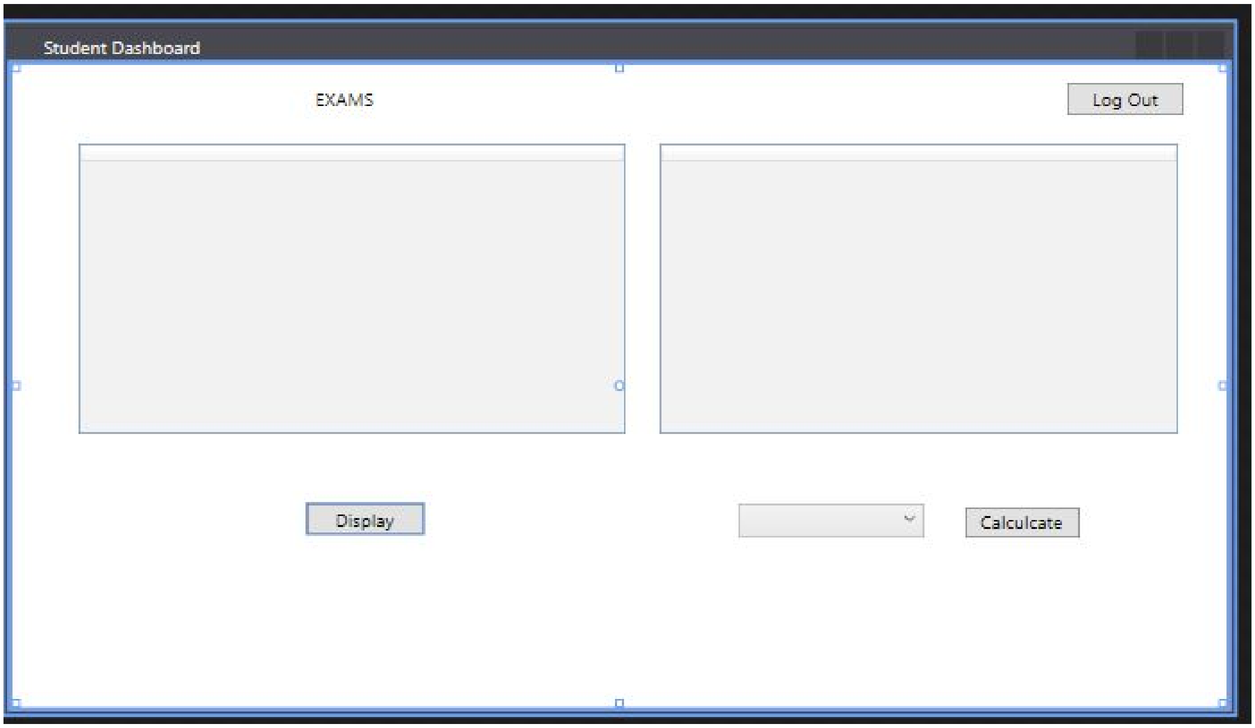
Login Screen:



The professor dashboard:



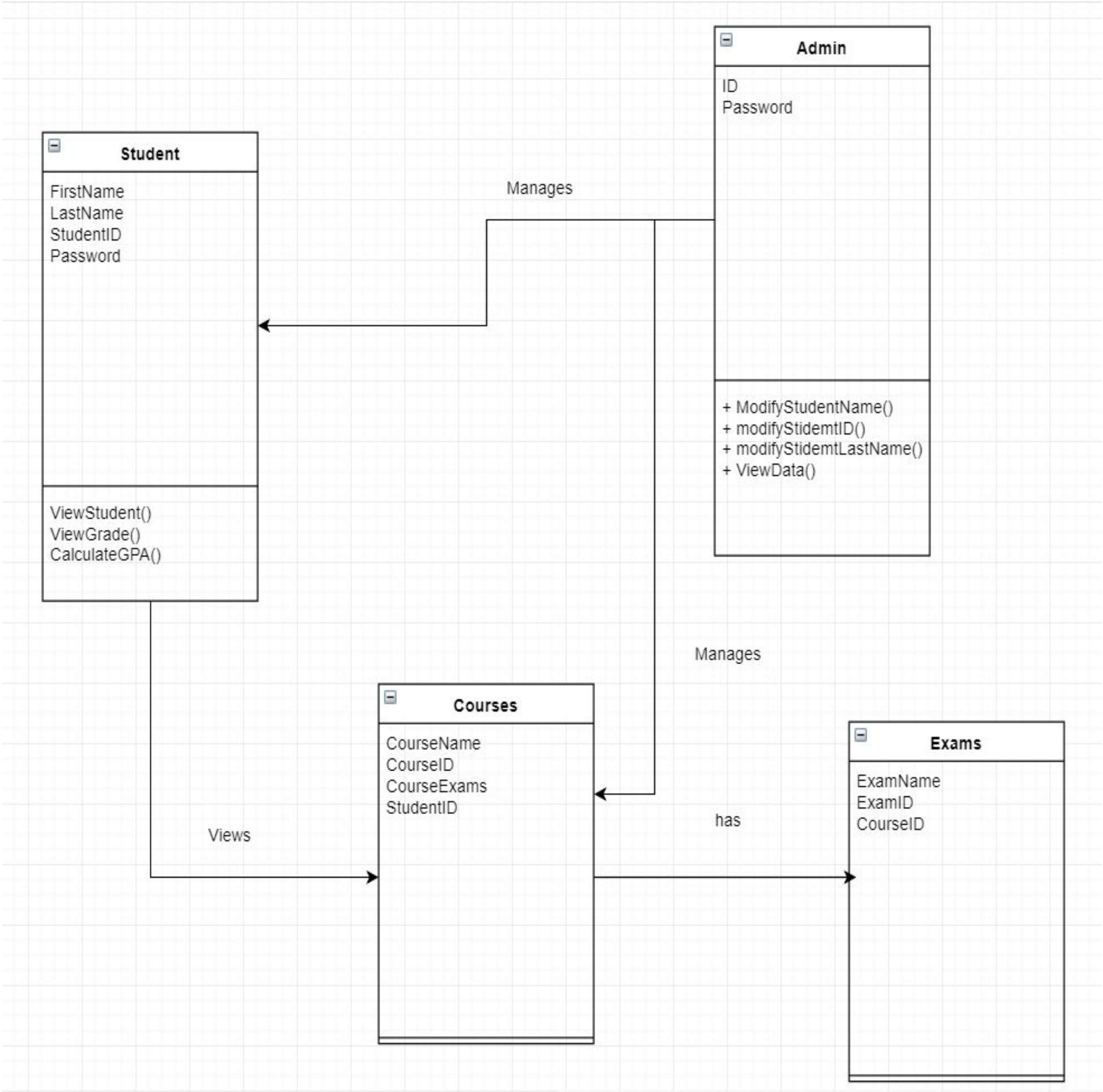
The student dashboard:



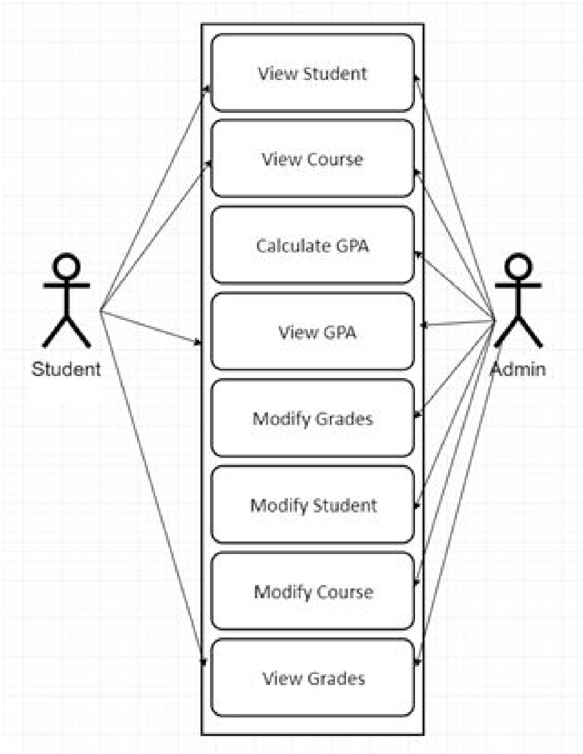
Although not as polished, the working prototype still displays all requirements for the LMS.

4. System Design

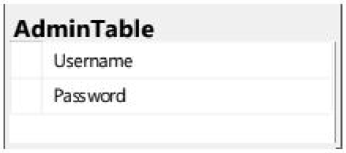
4.1 UML Design



4.2 Use Case Design



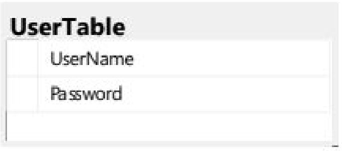
4.3 Database Design



AdminTable

Username

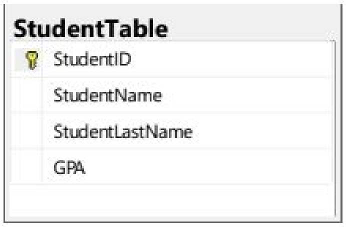
Password



UserTable

UserName

Password



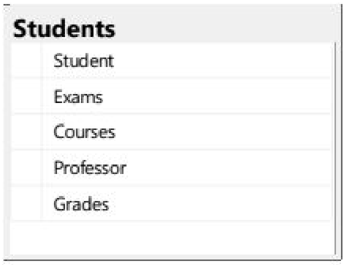
StudentTable

StudentID

StudentName

StudentLastName

GPA



Students

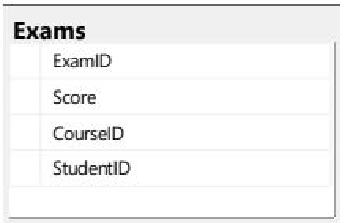
Student

Exams

Courses

Professor

Grades



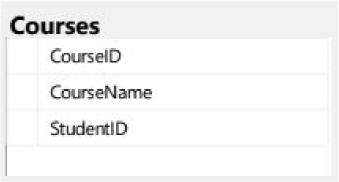
Exams

ExamID

Score

CourseID

StudentID



Courses

CourseID

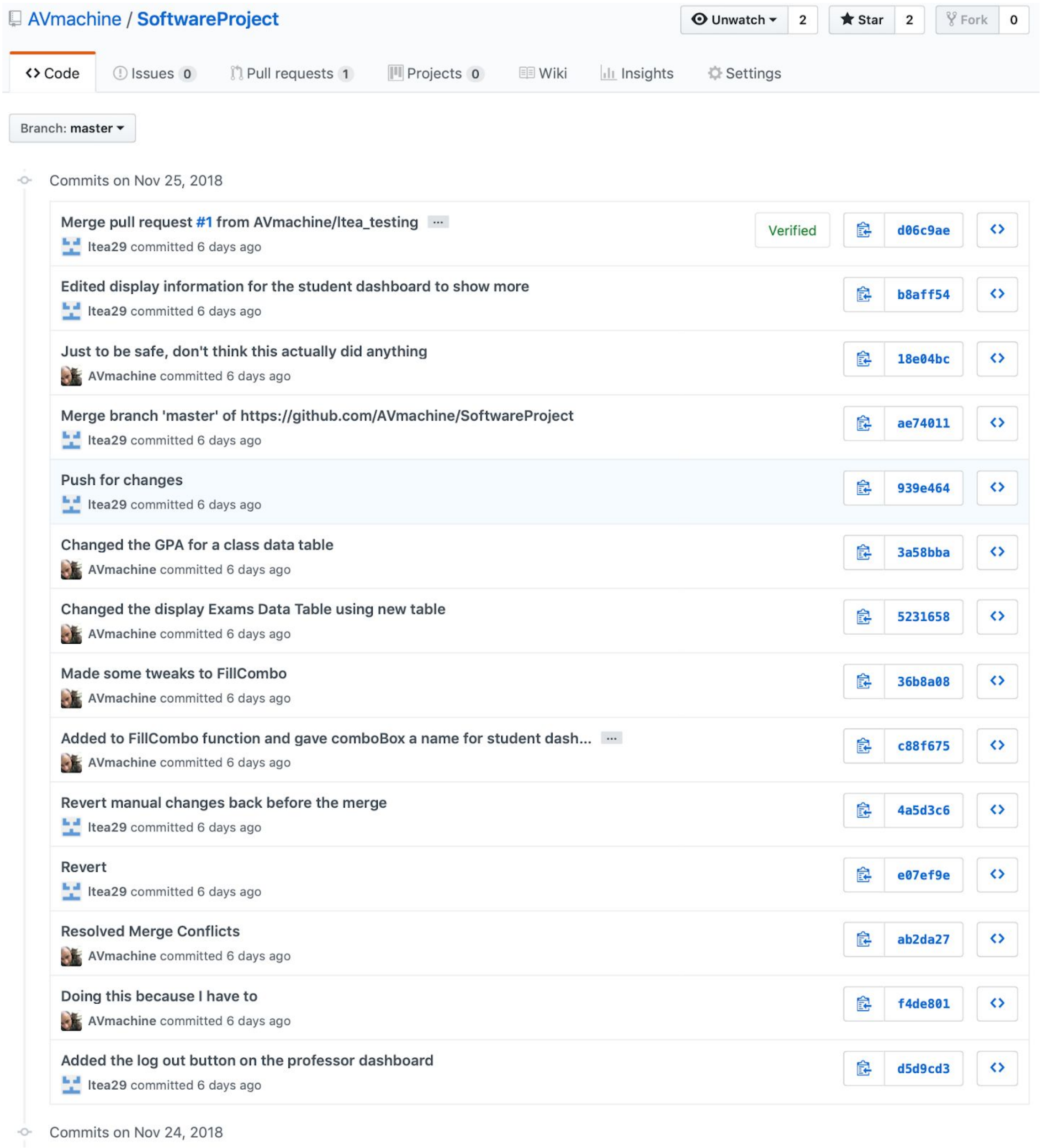
CourseName

StudentID

5. Team Coordination

5.1 Github

Our team utilized Github as our version control system. Using Github as our VCS served to our advantage in the fact that each of us was able to work on part of the project separately, and then merge all of our changes together and resolve any disputes.



Commits on Nov 25, 2018

Merge pull request #1 from AVmachine/Itea_testing …
Itea29 committed 6 days ago — Verified — d06c9ae

Edited display information for the student dashboard to show more
Itea29 committed 6 days ago — b8aff54

Just to be safe, don't think this actually did anything
AVmachine committed 6 days ago — 18e04bc

Merge branch 'master' of https://github.com/AVmachine/SoftwareProject
Itea29 committed 6 days ago — ae74011

Push for changes
Itea29 committed 6 days ago — 939e464

Changed the GPA for a class data table
AVmachine committed 6 days ago — 3a58bba

Changed the display Exams Data Table using new table
AVmachine committed 6 days ago — 5231658

Made some tweaks to FillCombo
AVmachine committed 6 days ago — 36b8a08

Added to FillCombo function and gave comboBox a name for student dash… …
AVmachine committed 6 days ago — c88f675

Revert manual changes back before the merge
Itea29 committed 6 days ago — 4a5d3c6

Revert
Itea29 committed 6 days ago — e07ef9e

Resolved Merge Conflicts
AVmachine committed 6 days ago — ab2da27

Doing this because I have to
AVmachine committed 6 days ago — f4de801

Added the log out button on the professor dashboard
Itea29 committed 6 days ago — d5d9cd3

Commits on Nov 24, 2018

5.2 Other Communications

In addition to collaborating on Github, we made use of a group messaging app to communicate, in addition to telephone calls and meeting in person work on the project. Staying on top of communication really helped us to keep an overall flow of the project.

6 Software Product Management Plan

6.1 Software Product Management Plan

Deliverable	**Due Date**
Use Case Diagram	10/28/2018
UML Design	11/04/2018
Github Setup	11/11/2018
First Prototype	11/18/2018
Revision 1	11/25/2018
Final Product	11/30/2018