**UHD Proposed Learning Management Software**

**Software Project Management Plan**

By: Suhaib Chaudhry, Javier Ramirez, Anh Huy Truong

**Table of Contents**

1. Introduction
   1. Project Overview
   2. Project Deliverables
   3. Evolution of this Document
   4. References
   5. Definitions, Acronyms, and Abbreviations
2. Project Organization
   1. Class Diagram
   2. Activity Diagram
   3. Use Case Diagram
   4. Organizational Structure
   5. Organizational Boundaries and Interfaces
   6. Project Responsibilities
3. Managerial Process
   1. Management Objectives and Priorities
4. Technical Process
   1. Use of Version Control
5. Software Details
6. Conclusion

**Introduction**

**Project Overview**

The main purpose for this project is to create a student information management system, For example UHD’s Blackboard system as a reference but a much more simple design when comparing it to Blackboard. The software system will mainly store and retrieve a student’s basic information such as : student name, student ID, registered courses, exam scores, add/delete students, and add/delete courses etc. The software will be a user interface and will begin with a login menu for either student, faculty, or admin once a specific user logs on they will be prompted with a few features that are exclusive to either the student, faculty or admin dashboard once the user is logged in he will be able to pick from the specific options available for him/her.

**Project Deliverables**

1. Preliminary Project Plan

2. Requirements Specification

3. Analysis [Object model, Dynamic model, and User interface]

4. Architecture Specification

5. Component/Object Specification

6. Source Code

7. Test Plan

8. Final Product w/ Demo

**Evolution of this Document**

This document will be updated as the project progresses. Updates should be expected in the following sections:

* **References** **-** updated as the project progresses.
* **Definitions, acronyms, and abbreviations** **-** updated as the project progresses.
* **Organizational Structure -** will be updated as the team leaders are assigned for each phase.
* **Technical Process *-*** this section will be revised appropriately and frequently as the requirements and design decisions become clearer.
* **Schedule *–*** as the project progresses, the schedule will be updated accordingly.

**References**

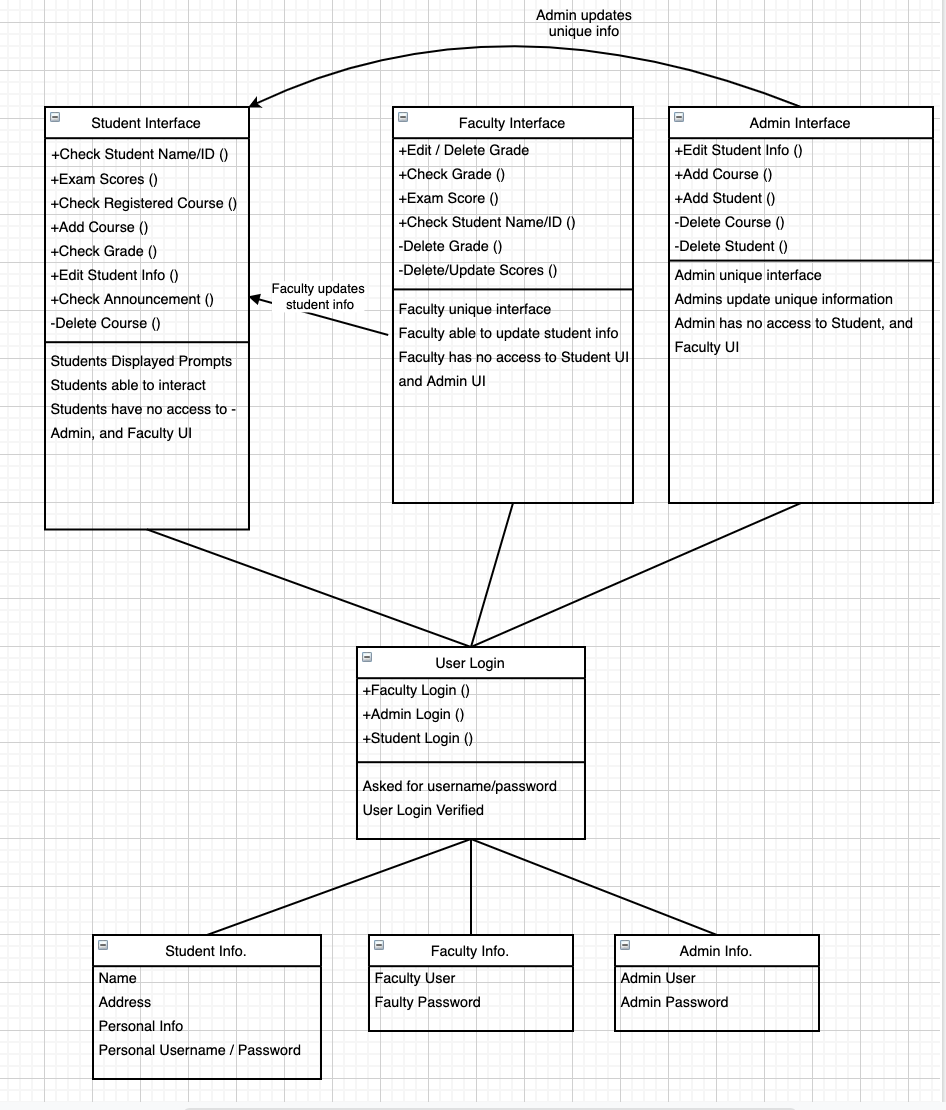
* <http://www.codebind.com/category/cpp-tutorial/qt-tutorial/>
* <https://tallyfy.com/uml-diagram/>
* <https://creately.com/blog/diagrams/uml-diagram-types-examples/#ClassDiagram>
* [https://draw.io](https://draw.io/)

**Definitions, Acronyms, and Abbreviations**

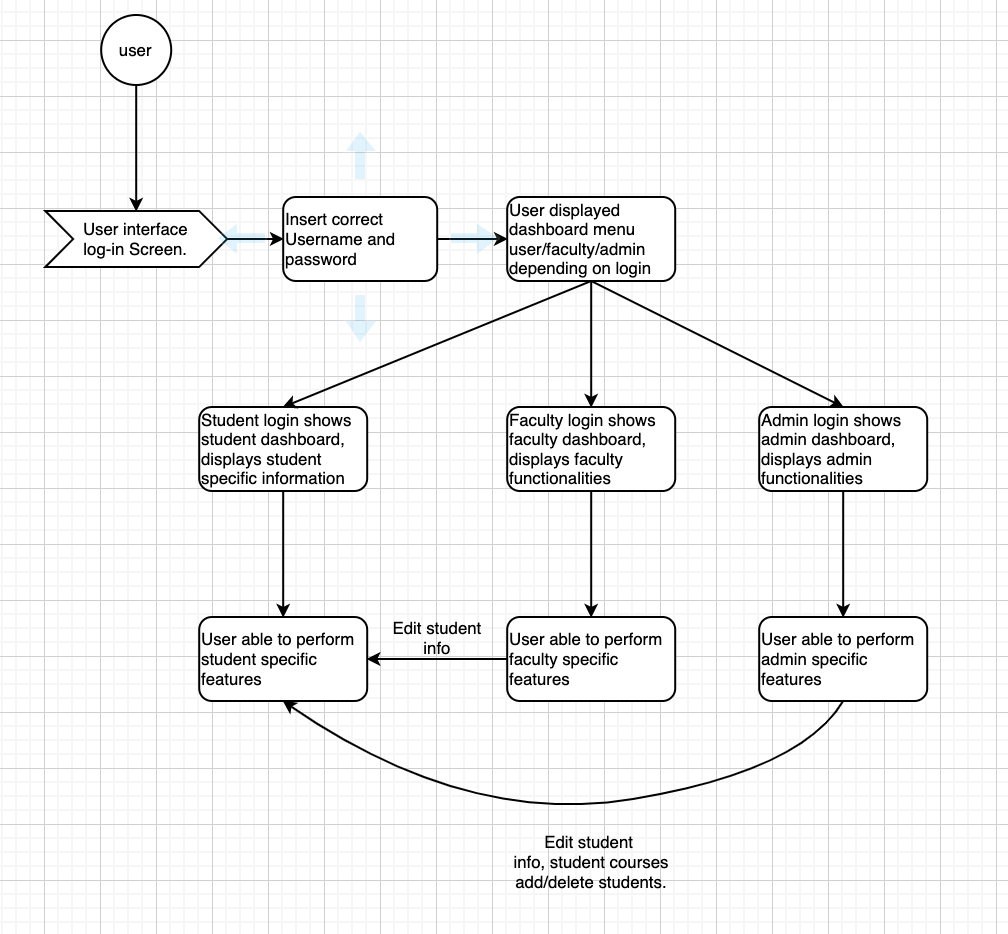
* UML - Unified Modeling Language
* AD - Architectural Design
* ADD - Architectural Design Documentation
* CD - Class Diagram
* UD - Use Case Diagram
* LCM - Life cycle model
* TM - Team Model
* Users - Student, Faculty, Admin
* PM - Project Manager
* SM - Senior Management SPMP Software Project Management Plan
* SQA - Software Quality Assurance
* SR - Software Requirements
* SRD - Software Requirements Document
* SUM - Software User Manual
* UR - User Requirements
* URD - User Requirements Document

**Project Organization**

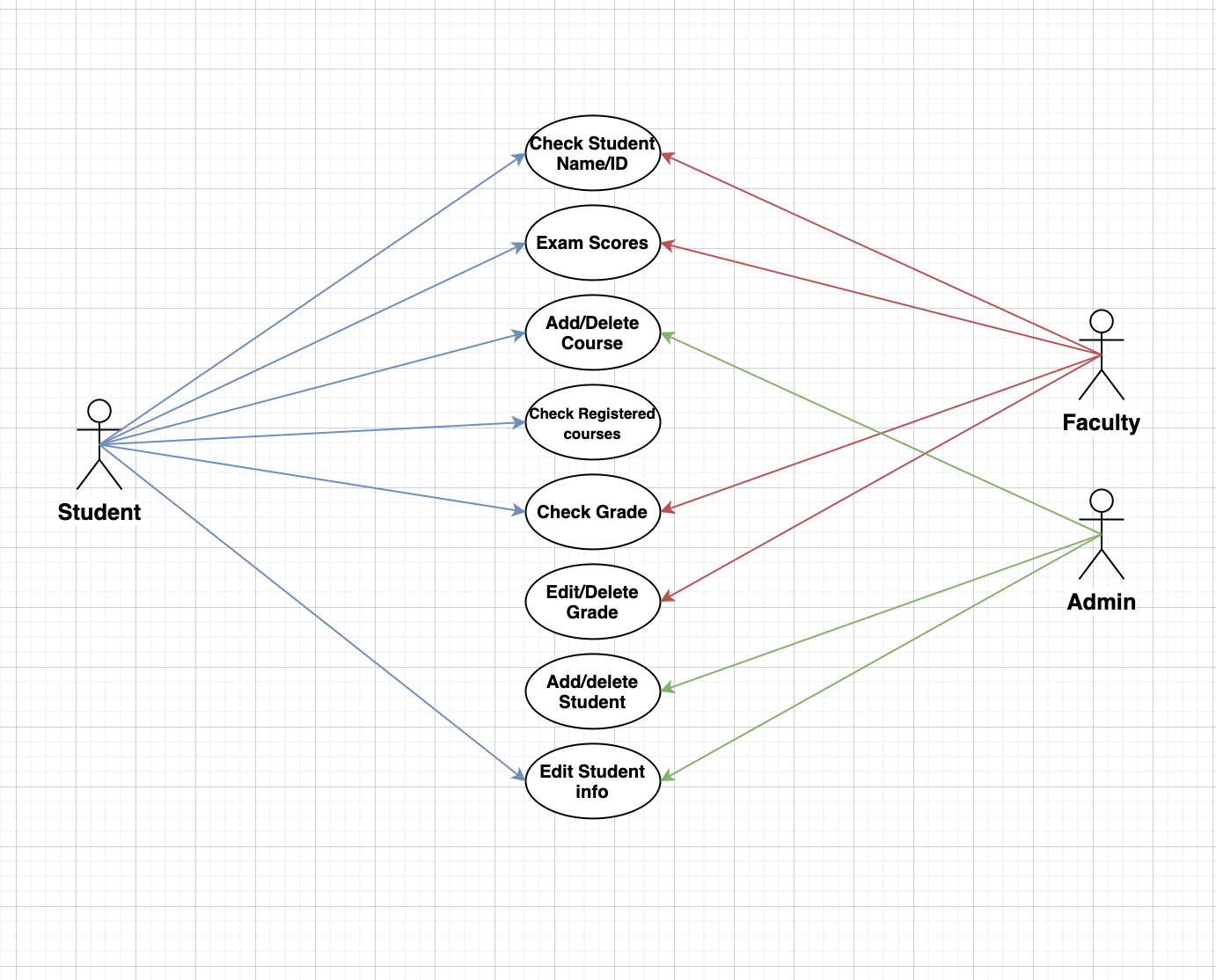
**Class Diagram:**



**Activity Diagram:**

****

**Use Case Diagram:**



**Organizational Structure**

* Team Members:
* Suhaib Chaudhry
* Javier Ramirez
* Anh Huy Truong

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **Team Leader** | **Deliverable Description** |
| 1 | All Team Members | Project Plan |
| 2 | Suhaib Chaudhry & Javier R. | Requirements Specification |
| 3 | Anh Huy Truong | Analysis |
| 4 | Javier Ramirez | Architecture Spec |
| 5 | Suhaib Chaudhry | Component/Object Specification |
| 6 | All Team Members | Source Code |
| 7 | All Team Members | Test Plan |
| 8 | All Team Members | Final Deliverable |

**Organizational Boundaries and Interfaces**

Team leaders throughout each development of the phases will be responsible for coordinating team meetings, updates, communications, and team deliverables.

**Project Responsibilities**

Ultimately the whole complete development of the project team is responsible for the successful delivery of the product. The team member tasks per deliverable according to expertise and the phases below:

* Project Plan – Whole Team
* Requirements Specification – Suhaib Chaudhry & Javier Ramirez
* Analysis – Anh Huy Truong
* Architecture Spec – Javier Ramirez
* Component/Object Specification – Suhaib Chaudhry
* Source Code – All Team Members
* Test Plan – All Team Members
* Final Deliverable – Entire Team

**Managerial Process**

**Management Objectives and Priorities**

The management objective is to deliver the product in time and of high quality. The Project manager and the whole team works together to achieve this by checking that progress is made as planned and monitoring the quality of the product at various stages. Our first step was what program to decide on using, we went with Qt using C++ to begin our project. Next, we decided how we wanted to structure our branches, the branches were split up into each team member and worked on altogether. If there were any bugs, we would join a voice chat and share each other’s code. We were able to work swiftly as a team communicating our problems and issues to resolve them and be able to accomplish our goal.

**Technical Process**

**Use of Version Control**

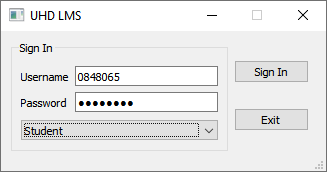
We used GitHub to track history of the software. We started by creating a repository that held the master branch containing the main window of our program. In order to work collaboratively using GitHub we decided to create three branches named student\_dashboard, faculty\_dashboard, and admin\_dashboard and assigned them to each of the team members as shown below:

* Suhaib Chaudhry – student\_dashboard
* Javier Ramirez – admin\_dashboard
* Anh Huy Truong – faculty\_dashboard

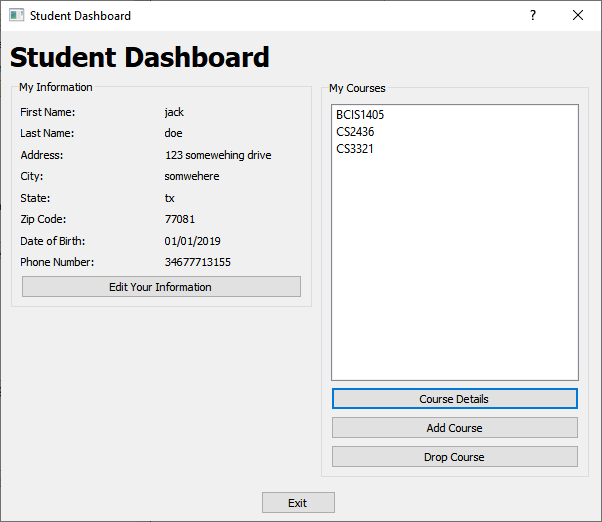
This allowed us to work on each separate feature at the same time; once we all completed our own features/dashboards, we merged the branches back into master and resolved the conflicts that arose due to the merge. At the end our code is all published at the GitHub repository below:

**GitHub Repository:** <https://github.com/suhaibchaudhry/lms_project>

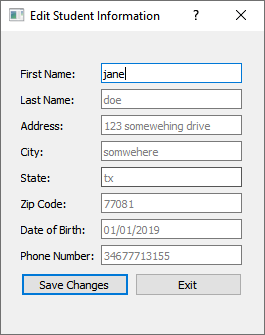
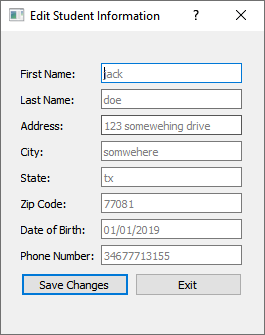
**Software Details**



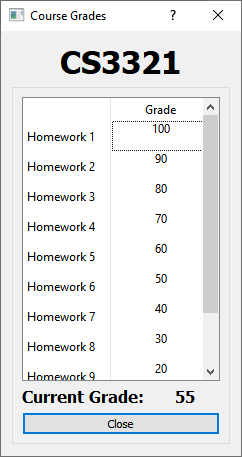
**Login Screen:** This screen allows the user to login with his/her login information which is checked with the database. The user then selects what type of user they are (Admin, Teacher, Student) and proceeds to click ‘Sign In” to enter the specific dashboard for that user. The ‘Exit’ button closes out of the program entirely.



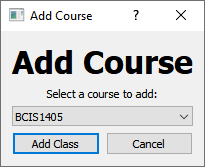
**Student Dashboard:** This screen is meant for the student and in it their basic information is displayed from the database. When the student clicks ‘Edit Your Information’, it opens a new window that allows their information to be edited and saved to the database. Under the ‘My Course’ section, the student can see the currently enrolled courses he/she is in and is able to click ‘Course Details’ after selecting a specific course to see their grades and assignments list for that specific course. When the student clicks on ‘Add Course’, a new window opens that allows the student to add a course to their list of enrolled students which is then in turn saved in the database. When the student clicks on ‘Drop Course’, the currently selected course is removed from the database and the list is refreshed to show the change. The ‘Exit’ button closes out of the program entirely.



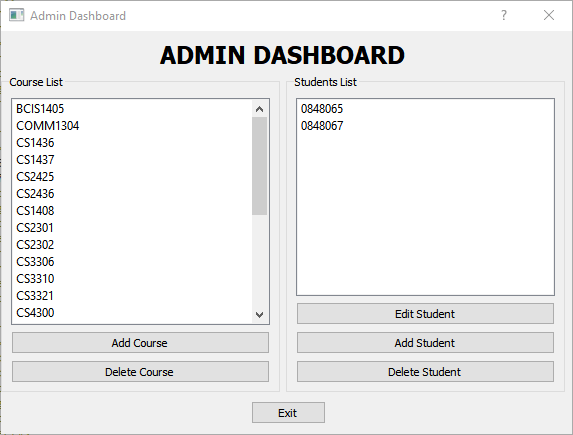
**Edit Student Information (From Student Dashboard):** This screen allows the student to edit their information and save the changes to the database. The placeholder text for each textbook reads the current data of the student from the database so he/she can see what they are changing. When clicking ‘Save Changes’, the program saves the information to the database and returns the user back to the Student Dashboard. The ‘Exit’ button closes out of the current screen without making any changes to the database.



**Course Details (From Student Dashboard):** This screen allows the student to view their current grades in the program and shows them the current grade they have in that class. The ‘close’ returns the student to the Student Dashboard. The data for the assignments list and the grades are all read from the database and the current grade is calculated based on those grades.

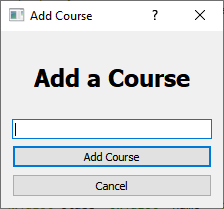


**Add Course (From Student Dashboard):** This screen allows the student to add a course from a full course list drop down menu. This drop down menu is populated from the database and is a premade list that can only be changed by an Admin. When the student clicks ‘Add Class’ the class currently selected in the drop down menu is added to that student's currently enrolled courses list and saved in the database. If the student presses ‘Cancel’, the user is returned back to the Student Dashboard.

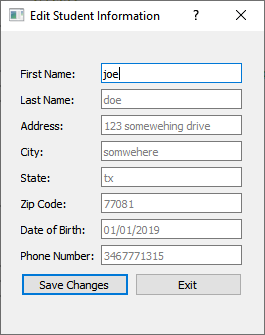
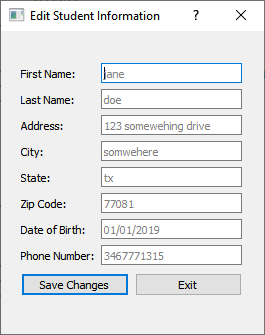


**Admin Dashboard:**

This dashboard is only seen by an administrator user when they are authenticated. On the left side they have access to a full course list and clicking ‘Add Course’ opens a new window allowing them to add a course to this list. If they click ‘Delete Course’, the currently selected course is removed from the list and the database and the list in the Admin Dashboard is refreshed. On the right side, there’s a students list in which all the currently enrolled students are listed. The Admin is able to edit the students information by clicking ‘Edit Student’ and entering the new student information in the new window. If the administrator clicks on ‘Add Student’, another window opens allowing the administrator to specify new information for a new student including their username and password and saves them to the database. Lastly, if the administrator clicks ‘Delete Student’, the currently selected student in the list is removed from the database and the list is refreshed. Clicking on ‘Exit’ closes the program completely.

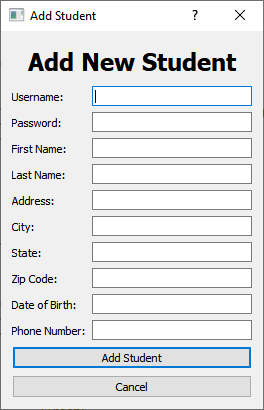


**Add Course (From Admin Dashboard):** This screen allows the admin to add a course name. When the administrator clicks ‘Add Course’ the course name currently typed in the textbox is added to the full list of available courses and saved in the database. If the student presses ‘Cancel’, the user is returned back to the Student Dashboard.

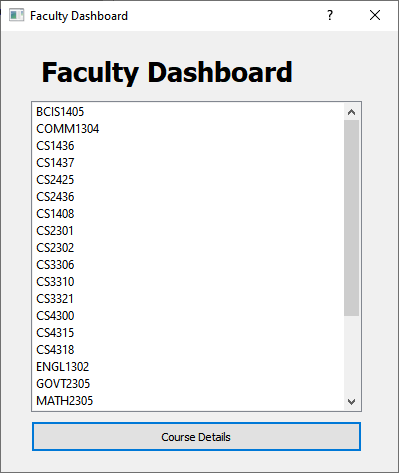


**Edit Student Information (From Admin Dashboard):**

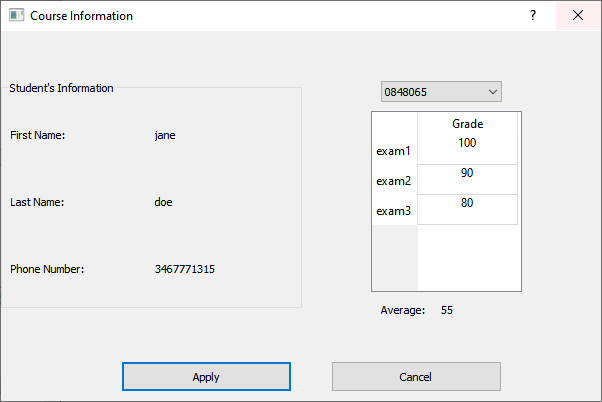
This screen allows the admin to edit student information and save the changes to the database. The placeholder text for each textbook reads the current data of the student from the database so the admin can see what they are changing. When clicking ‘Save Changes’, the program saves the information to the database and returns the user back to the Admin Dashboard. The ‘Exit’ button closes out of the current screen without making any changes to the database.



**Add New Student:** This screen allows the administrator to create a new student in the database. Upon clicking ‘Add Student’, the fields ‘First Name’ to ‘Phone Number’ are saved in the database as student information and the fields ‘Username’ and ‘Password’ are saved in the database for student authentication. If the administrator clicks on ‘Cancel’, no changes are saved to the database and the user is returned back to the Admin Dashboard.



**Faculty Dashboard:** This dashboard is only seen by a teacher when they are authenticated. It shows a full list of courses from the database and allows the teacher to click on a course and edit the information for that course when clicking on ‘Course Details’ after selecting a course from the list.



**Course Information:** Left side is information of the student with the little combo box on the right side for teachers to choose which student information/grade he wants to see. Anh was not able to make it so that the teacher could edit the grade so the ‘Apply’ button is of no use. Pressing ‘cancel’ will take the user back into the faculty dashboard.

**Conclusion:**

Overall as a team we managed to learn many things, communication became a huge factor when making our project and shared information crucial to achieving our goal. We worked as a team on this project and structured each project into part to make this project easier to work with. We managed to get the results we wanted and as a team we learned many new things. Not only were we exposed to a new program but we were able to solve the various bugs and issues that arose; this project was a learning experience from the start.