Software Project Management Plan

for

Learning Management System

**Prepared by:**

**Luis Llanas**

**Rafael Navarro**

**Diana Rodarte**

**Eva Ruiz**

**Software Engineering CS3321**

**November 26, 2019**

Table of Contents

[1.0](#_heading=h.gjdgxs) Introduction 3

[1.1](#_heading=h.30j0zll) Project Overview 3

[1.2](#_heading=h.1fob9te) Project Deliverables 3

[1.3](#_heading=h.3znysh7) Evolution of the SPMP 3

[1.4](#_heading=h.2et92p0) Reference Materials 3

[1.5](#_heading=h.tyjcwt) Definitions and Acronyms 3

[2.0](#_heading=h.3dy6vkm) Project Organization 4

[2.1](#_heading=h.1t3h5sf) Process Model 4

[2.2](#_heading=h.4d34og8) Organizational Structure 4

[2.3](#_heading=h.2s8eyo1) Organizational Interfaces 4

[2.4](#_heading=h.17dp8vu) Project Responsibilities 5

[3.0](#_heading=h.3rdcrjn) Managerial Process 5

[3.1](#_heading=h.26in1rg) User Interfaces/Documentation 5

[3.2](#_heading=h.35nkun2) Management Objectives and Priorities 9

3.3 Assumptions, Dependencies, and Constraints 9

[3.3](#_heading=h.1ksv4uv) Risk Management 10

[3.4](#_heading=h.44sinio) Communication Interferences 10

[4.0](#_heading=h.z337ya) Technical Process 10

[4.1](#_heading=h.3j2qqm3) Methods, Tools, and Techniques 10

[4.2](#_heading=h.1y810tw) Software Documentation 10

[4.2.1 Software Requirements Specification (SRS)](#_heading=h.4i7ojhp) 11

[4.2.2 Software Test Plan](#_heading=h.2xcytpi) 11

[4.3](#_heading=h.3whwml4) Unified Modeling Language 11

[5.0](#_heading=h.qsh70q) Work Packages, Schedule, and Budget 13

[5.1](#_heading=h.3as4poj) Work Packages 13

[5.2](#_heading=h.1pxezwc) Resource Requirements 13

[5.3](#_heading=h.49x2ik5) Budget and Resource Allocation 13

[5.4](#_heading=h.2p2csry) Schedule 13

[6.0](#_heading=h.3o7alnk) Additional Components 14

[6.1](#_heading=h.23ckvvd) UML Class Diagrams 14

1. **Introduction**

* 1. **Project Overview**

This document has the purpose of describing all the technical aspects concerning the development of a Learning Management System for the Software Engineering class at the University of Houston Downtown. All the content is directed to the members of the group for planning and scheduling purposes, serving as a summary document about the evolution of the project and member’s activities.

* 1. **Project Deliverables**

The project will develop an LMS software which will have three different kind of users: professors, students, and administrators. The system will allow professors to include theirs classes, manage their content, include exam questions and answers, and track student’s progress. The students will be able to view an exam, submit answers to the questions on the exam, and be able to keep track of their grades. The administrator will be able to manage all aspects of the system.

* 1. **Evolution of the SPMP**

The software development utilizes the GitHub version control system available online. Proposed changes and new versions will be available on GitHub, and this document was properly updated.

* 1. **Reference Materials**

No reference codes were used in the development of the Learning Management System. All code was done by individual developers and previous knowledge of Python language. The Graphical User Interface design was generated by the PyQt Creator.

* 1. **Definitions and Acronyms**
* LMS – Learning Management System
* GUI – Graphics User Interface
* UML - Unified Modeling Language
* UI – User Interface
* SPMP- Software Project Management Plan
* SDLC - Software Development Life Cycle

1. **Project Organization**
   1. **Process Model**

The project was initiated on August, 2019 and with the team using Django to develop the Learning Management System. However, after careful thought and consideration, it was confirmed to transition to a different platform due to short amount of time to implant. We decided to switch to using PyQt Designer. The Software Development Life Cycle model used was the agile model. The agile model separates the product cycles and delivers a working product very quickly. This method also allowed for the production of a succession of releases. Testing of each releases feedback info that was incorporated into the next versions of the LMS.

The project uses an object-oriented design methodology and has a team organization of a Democratic Team Approach. At this initial phase, each member of the team has one activity that has to be submitted to the platform GitHub. Each team members will interact with each other and produce the software prototype utilizing the GitHub as a version control tool. This also goes hand in hand with the Agile SDLC model where the team members are encouraged to give back feedback throughout the whole project, so that lessons learned are used to improve future iterations.

* 1. **Organizational Structure**

The client:

Dr. Yuchou Chang.

The Project managers is:

Rafael Navaro

The infrastructure team consists of:

Luis Llanas, Eva Ruiz

The Software Project Management Planning (documentation) is:

Diana Rodarte

* 1. **Organizational Interfaces**

Meetings times:

* 09/04/2019 – 2:30PM till 4:00PM
* 09/11/2019 – 2:30PM till 4:00PM
* 09/18/2019 – 2:30PM till 4:00PM
* 11/13/2019 – 2:30PM till 4:00PM
* 11/16/2019 – 1:00PM till 5:00PM
* 11/25/2019 – 12:45PM till 2:00PM
  1. **Project Responsibilities**

Diana Rodarte: UML and documentation (SPMP)

Eva Ruiz: Software Development

Luis Llanas: GUI, Software Development

Rafael Navarro: Software Development

1. **Managerial Process**
   1. **User Interfaces/Documentation**

Figure 1, demonstrates the GUI for the Login app screen where the user can select the checkbox if they are a “Student” or “Professor”. Then if they would like to create an account, they can click the “Signup” button, after they entered the desired username and password. Once the user creates their credentials then they may click the “Login” button. If the user already had an account initially, they may just enter their username and password and click the “Login” button. If the user desires to not “Login” or “Signup” they may quit the interface by clicking the “Quit” button. If the user is an admin they do not select a check box when logging in.

All buttons have a black border and a font of Arial. This user interface will not support a 4k screen resolution but anything under it will.

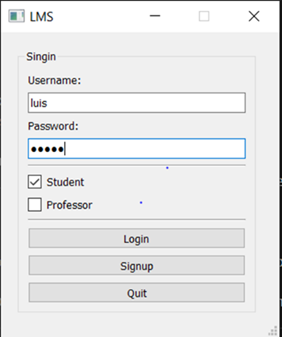


Figure 1. Login app screen

Figure 2, is the UI for the MainWindow for the Student screen. The student is only able to view their exams for each class and their GPA. The student is not able to modify the grades or GPA at all and the textboxes are voided for that reason. If the student wants to quit the interface they may click the “Quit” button and it will quit them from the interface. This UI also does not support a 4k resolution screen; however, anything below a 4k resolution it will.

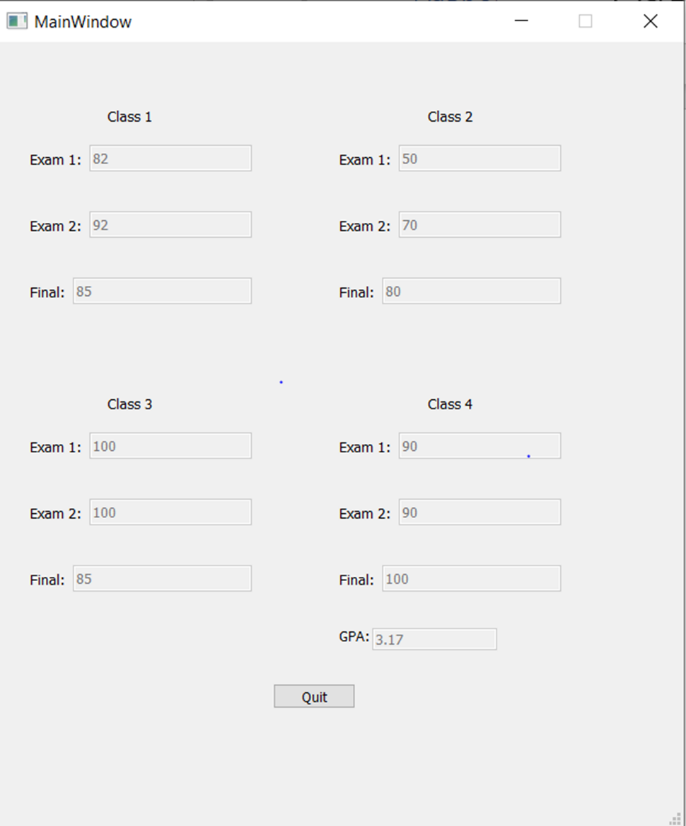


Figure 2. MainWindow for student screen

Figure 3, the represents the UI for the Professor Menu screen. Once the professor logs into the system they may click the “Exams” button if they wish to open up the exam questions and modify the questions and answers. The professor may also click the “Grades” button if they want to update the students’ exam grades. This UI also does not support a 4k resolution screen; however, anything below a 4k resolution it will.

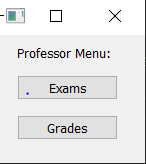


Figure 3. Professor Menu screen

Figure 4, is the UI for the MainWindow for the Professor screen. If the professor clicks the “Exams” button in Figure 3, then it will take them to this UI where the professor may select what class to modify grades by selecting the dropdown option “Classes.” Then selecting which student by selecting the dropdown option “Student.” Once the student is selected from the proper class the professor may update the exam grades by typing in the grade in the proper textbox. The correct text boxes will be in a white background versus gray, so the professor does not confuse a student with a different class. To save changes the professor clicks the “Save” button at the bottom and then the “Ok” button to confirm. If the professor wishes to then get out the interface, they may click the “Quit” button.

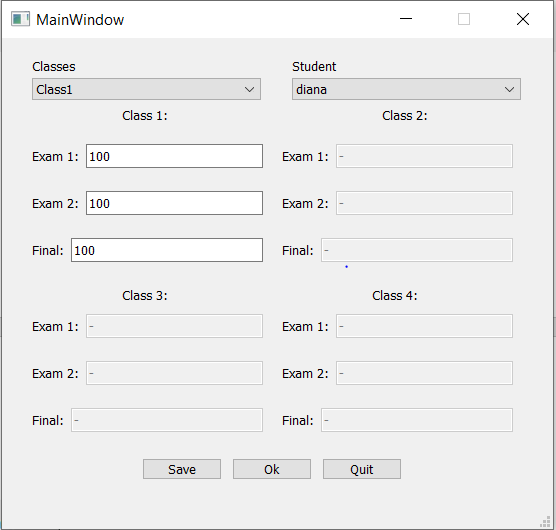


Figure 4. MainWindow for Professor screen

In figure 5, the UI represents the form where the questions and answers are inputted by the professor. The professor selects the class, exam and question by selecting the dropdown option corresponding accordingly. The professor then types in the question that wants the student to answer by typing it in the “Question” textbox. Then the professor creates and inputs the answer choices for each textbox. The professor then clicks the “update” then the “save” button when everything is entered. If the professor wishes to add or create a new question then they will click the “add” or “new” button. The “Answer” button is not enabled because that button is reserved for the student to submit their answer to the question asked.

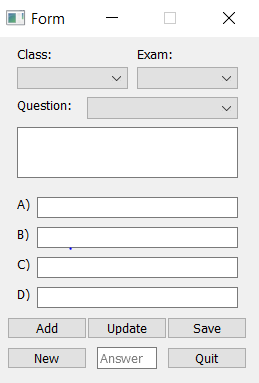


Figure 5. Form for Exam Question screen

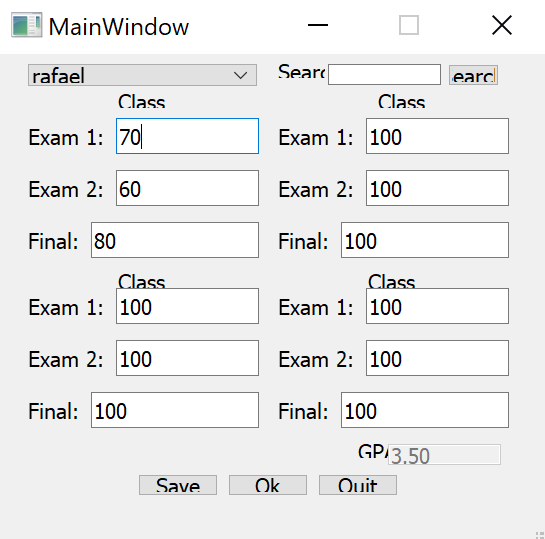
Figure 6, represents the MainWindow screen for the Admin. The admin does have access to mostly anything that needs to be modified. The admin has a search bar compared to the other UI. If the admin wishes to change any grades for the students they may do so by putting in the correct grade and clicking the “save” and then the “Ok” button. The admin however, cannot change a student’s GPA. That is why the “GPA” textbox is not enabled and has a gray background.

Figure 6. MainWindow for Admin screen

Figure 7. shows how the students are able to answer questions from an exam.They select a class, exam number and question number corresponding by selecting the dropdown option. Then the student may select the correct answer to the question by clicking the circular button next to the answer letter on the right. It will be colored black when clicked. The student may then continue with answering the exam questions by clicking the “Next” button. The student may also go back and see an exam question by clicking the “Previous” button. If the student wishes to open the exam and view all the questions at the same time, then they may click the “Open” button. When the student is done answering the questions, they may quit the UI by clicking the “Quit” button, which will terminate the interface.

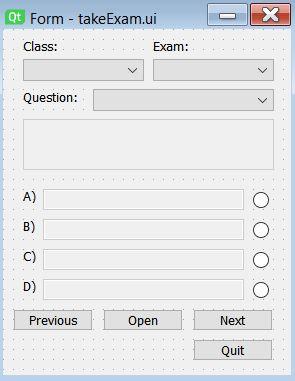


Figure 7. TakeExam screen

* 1. **Management Objectives and Priorities**

The initial objective is to define all entities and its attributes. It will help the project team to develop the objects (classes) and prepare a database. This is set as a high priority since this is a necessary information, understand the data flow, in order to proceed with further steps. As well as finding an efficient platform to create the UI’s and develop the LMS through python.

* 1. **Assumptions, Dependencies, and Constraints**

Python 3.8 must be installed in order to run the application since it has not been deployed. In addition, SQLite must be installed as well in order to have access to the database and win32api must be imported in order to view the features on a few of the windows from the LMS. Furthermore, a reasonable time frame was a dependency/constraint. The time frame to complete the LMS was a determining factor when deciding what features to focus on and implement.

* 1. **Risk Management**

Since the project is being developed for a class project, there is no risk involved.

* 1. **Communications Interfaces**

The team used “GroupMe” to communicate and set up meetings when needed. It was quick and efficient.

1. **Technical Process**
   1. **Methods, Tools, and Techniques**

The project was developed using PyQt Designer to create the graphical UL’s more efficient from the Qt GUI framework. It allowed for a simple drag and drop interface for laying out component such as buttons, text fields, combo boxes and more. Python was used to run the application since it has not been deployed. In addition, SQLite was installed as well in order to have access to the database. Win32api was used to view the features on a few of the windows from the LMS.

* 1. **Software Documentation**

The project is a GUI- based Learning Management System built in Python 3.8. The application uses SQL lite commands together with several GUI objects in order to display student information from a small student database. A user can login and select whether they are a Professor or a Student. Depending on what the user selects, the program will check against the database to make sure the user information is authentic, and provided access based on that. In the event of valid credentials for a student user, the program will take them to a menu where they can see their grade for their 4 classes as well as viewing their cumulative GPA. Furthermore, they can take professor assigned test. A professor user can login and view all the students grades for their class, as well as make edits to individual exam scores from their menu. This throws update commands to the mySQLlite database to make sure the professor can update each student grades individually for their classes. The professors also can write and administer test from the professor menu. Lastly, there is an administrator user, who can login by unselecting both the “student” and “professor” checkboxes, as well as entering valid administrator data. This type of user has access to each student records and can edit them as they desire. They are not restricted to classes they teach as the professor is and can make all the edits from a single screen. However, the admin cannot administer test since they do not actually have any courses assigned to them. The functionality of the application fully depends on the type of “user” that is logging into the application.

* + 1. **Software Requirements Specification (SRS)**

The software being developed is a Learning Management System. Thus, this system should allow faculty to create their classes/courses, include their content, include the assignments for each class, and track students’ progress. From students’ perspective, the system should allow them to visualize the classes’ content, submit their assignments, and follow their grades posted. The admin has access mostly to both the students’ and professors’ systems. However, they do not have the ability to change a student’s GPA, answer questions, or create questions.

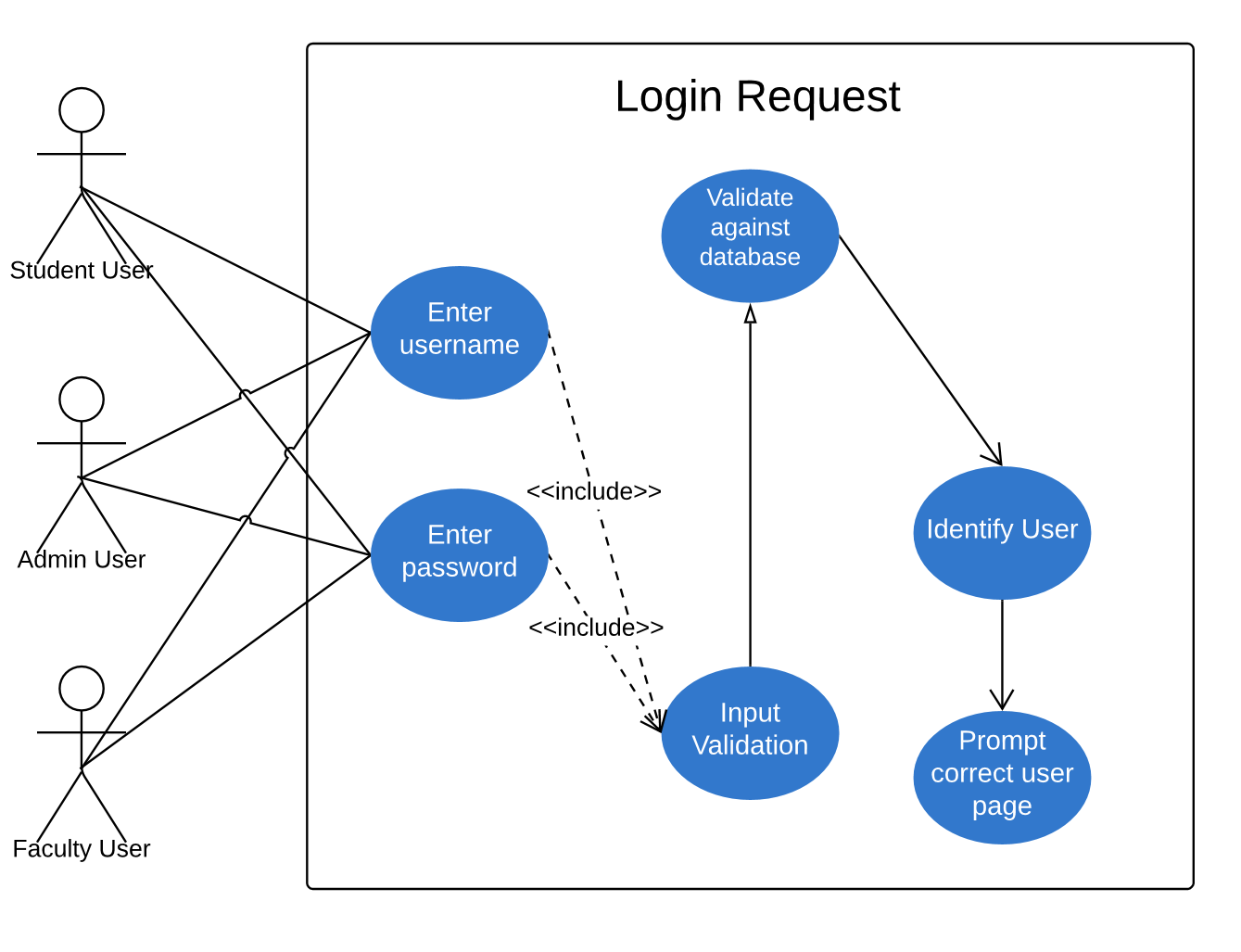
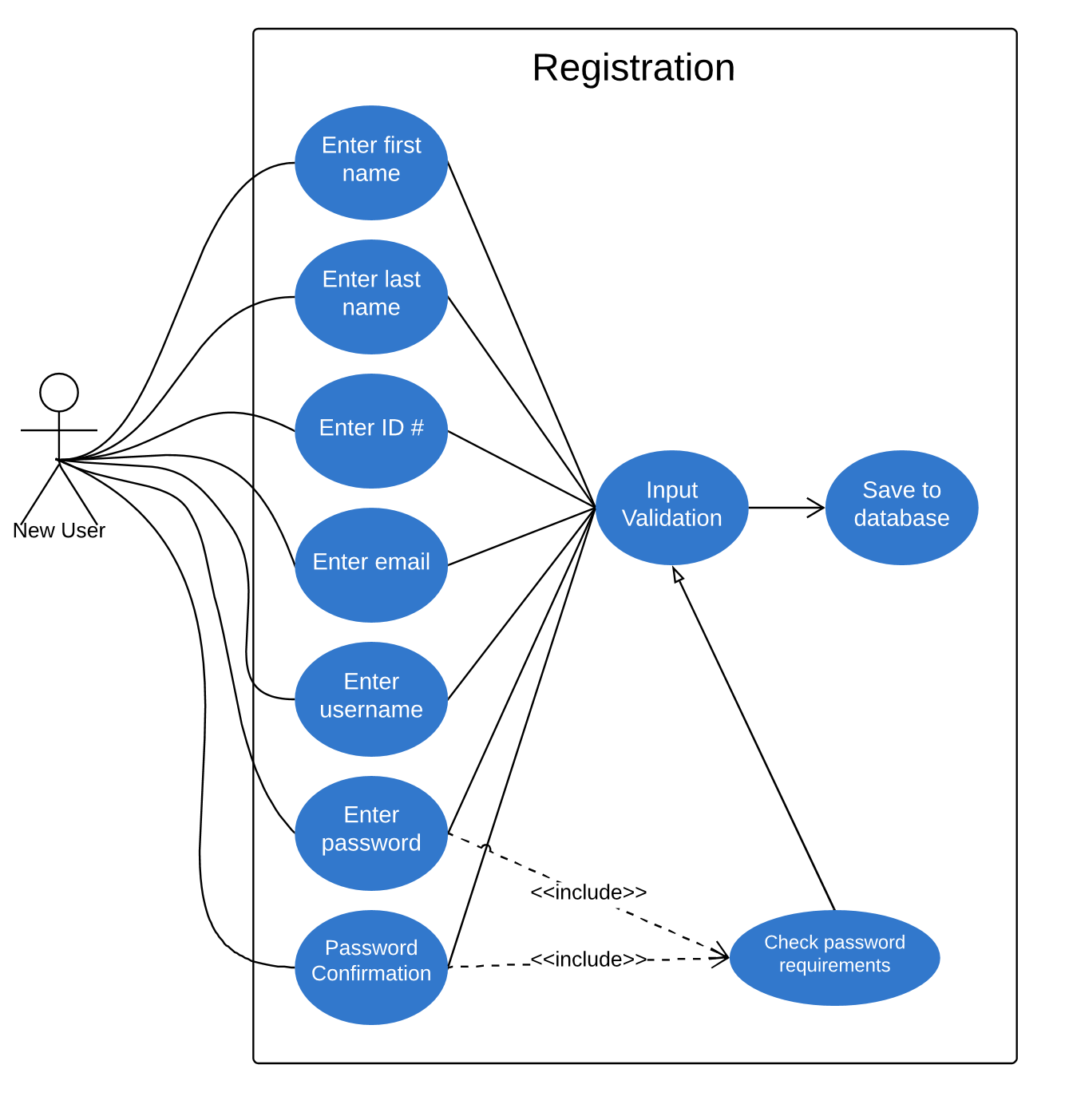
* + 1. **Software Test Plan**

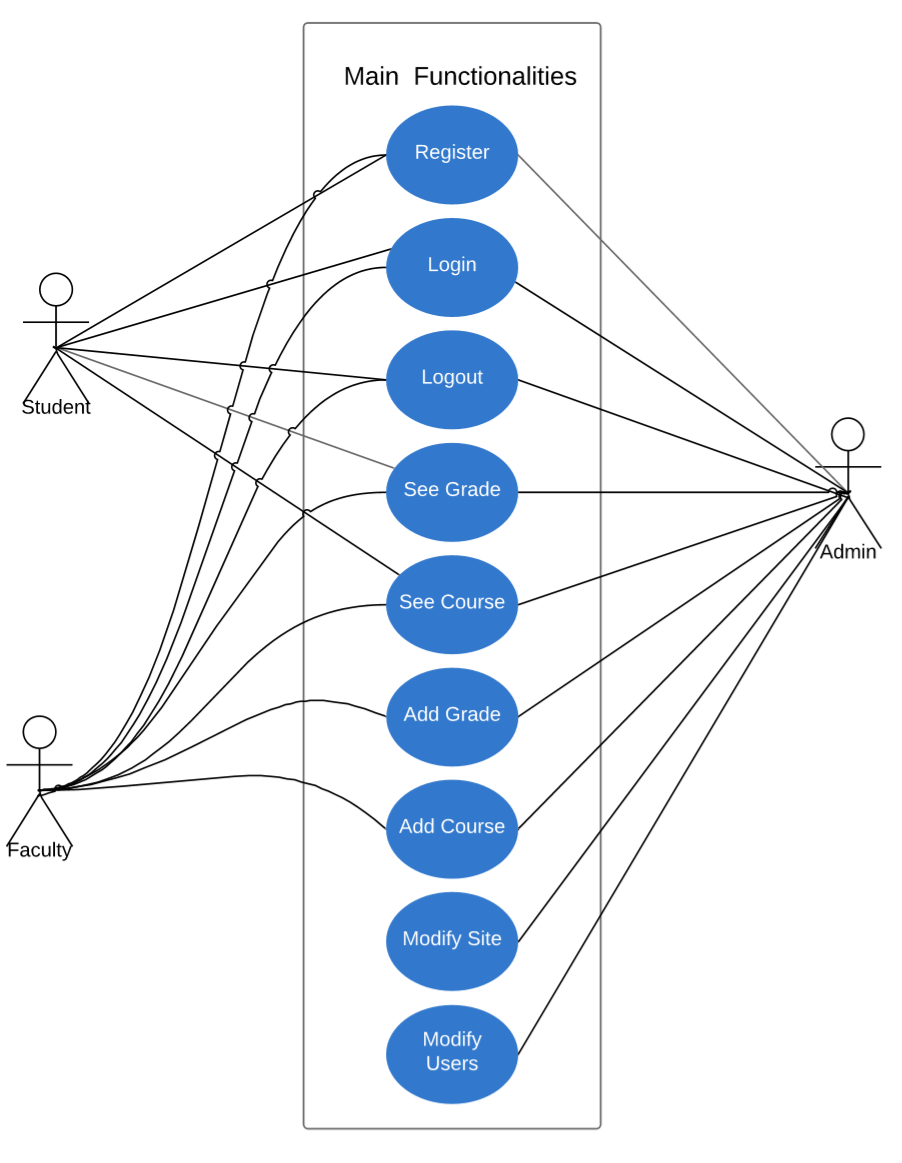
The software testing plan used was unit testing. Instead of detecting a breakpoint in the LMS, then firing up the GUI to provide a few inputs that will hopefully hit the code and that it resolves the bug, the team used unit testing. The developer wrote the test, the code and ran the test. Writing tests took some time but it was compensated with the less amount of time it took to run the tests. The team opened the UI and provides all those inputs and tested them. For example, the validation of the password and username of the Login GUI. Debugging the software was made easier because when a test failed, only the latest changes needed to be debugged. With testing at higher levels, changes were made in the span of a couple of days.

* 1. **Unified Modeling Language**

Figure 8, is representing the UML’s for registration, login request, the main functionalities, and displaying courses. This assists enabling the software developers to specify, visualize, construct and document the artifacts of the software system. Thus, these UML’s make these artifacts scalable, secure and robust in execution. Some of the execuations may not be in the final LMS; however, this is what the team had in mind to finish if more time permits. In the Registration UML, the user may enter their first and last name, their ID #, email, username, password. Then there will be a password confirmation where the LMS asks the user to enter a password twice again to confirm it is the same password and not a typo. The login request UML has three different users. All three users (student, professor and admin) has the ability to enter their username created when registration, enter their password. This will then allow the LMS to validate the inputted credentials, against the database. Once confirmed, it will identify the user and prompt their correct UI screen.

The main functionalities UML shows how the student and faculty has access to certain actions. However, the admin has access to mostly everything in the LMS. When displaying the course, the student and faculty is able to view their name and course they are registered with the help of the database. Again, something to be noted is that some of the functionalities were not fully implemented in the final LMS; however, if the project was a year long project versus a 4 month long project, the remaining functionalities could be implemented in the LMS.





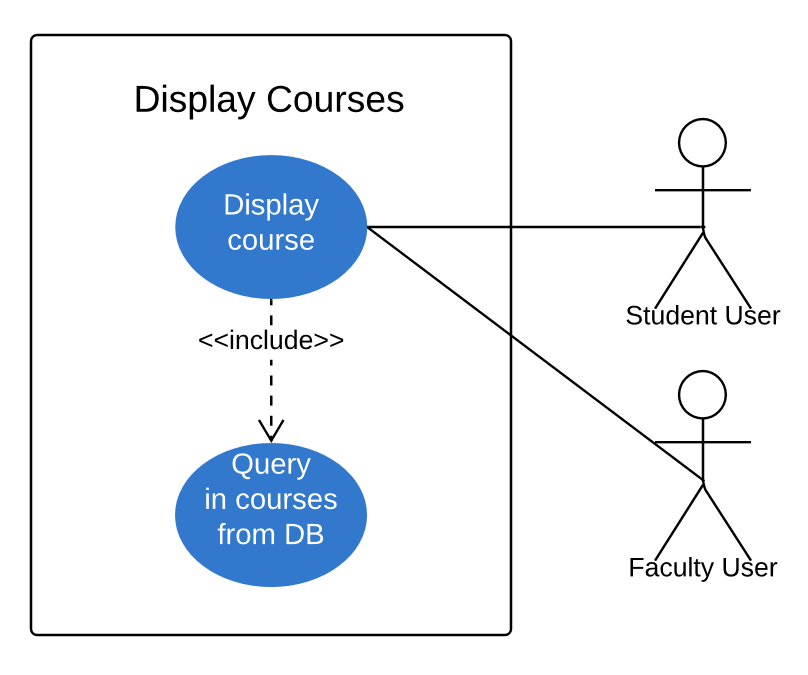


Figure 8. Unified Modeling Language

1. **Work Packages, Schedule, and Budget**
   1. **Work Packages**

The project manager assigns roles to each team member. Then the chosen software developers are assigned with developing the backend of the LMS. The person in charge of the documentation records any changes, onto the SPMP.

* 1. **Resource Requirements**

We did not use any source codes. All code was done by individual developers and previous knowledge of Python language. The Graphical User Interface design was generated by the PyQt Creator.

* 1. **Budget and Resource Allocation**

Since this project is being developed in a form of class project, there is no budget associated.

* 1. **Schedule**
* August 20, 2019 - Create and solodate team project members
* August 29, 2019 - Introduce and discuss the LMS and its components
* September 2, 2019 - Complete individual research on the most efficient way to conduct this LMS Team project. As well, as the which team cycle to best use
* September 12, 2019 - Discuss on the top ways to carry out the project
* September 19, 2019 - Solidate the IDEs, languages, platforms, team cycle, assign roles for each team member, etc. that will be used
* September 23, 2019 - Design and implement the UMLs. Begin document on the SPMP
* September 25, 2019 - Starting designing the GUI on the chosen platform with the chosen languages. Continue Documentation as well
* September 26, 2019- Present the midterm LMS project and any artifacts created through out the project, so far to Dr.Chang and the class
* October 5, 2019 - Continue development and documentation
* November 7, 2019 - Correct any bugs detected in the system and document
* November 12, 2019 - Continue development and documentation
* November 14, 2019 - Start preparing presentation for final presentation. Continue documening on SPMP
* November 19, 2019- Improve any UI screen if necessary and enough time. Continue developing and documentation
* November 25, 2019 - Present final presentation of LMS with corresponding artifacts. As well as a GitHub and the LMS Demo
* November 25, 2019 - Make any final adjustments to LMS and SPMP. Then compress the project and turn in into BlackBoard.

1. **Additional Components**

Figure 9, shows the UMLS Class diagrams that were used in the LMS. They are designed to represent the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects. There are 4 classes: grades, grade, answeredExam, and examsTest. Each class has their corresponding attributes and operations as follows.

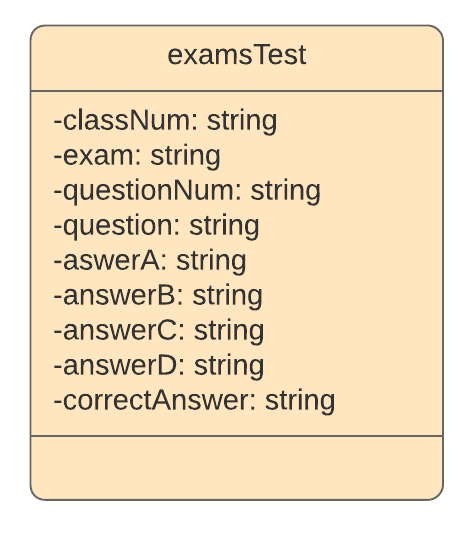
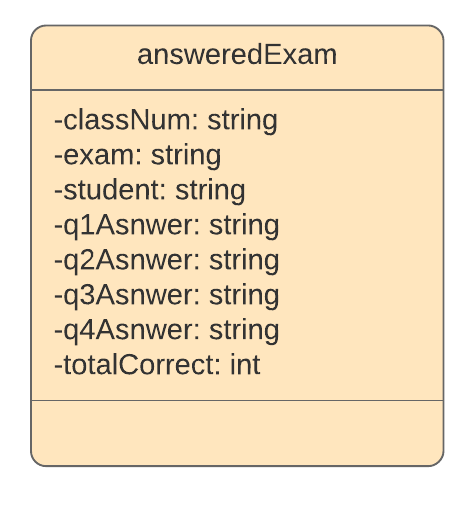
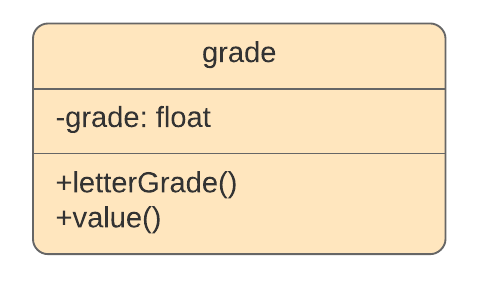
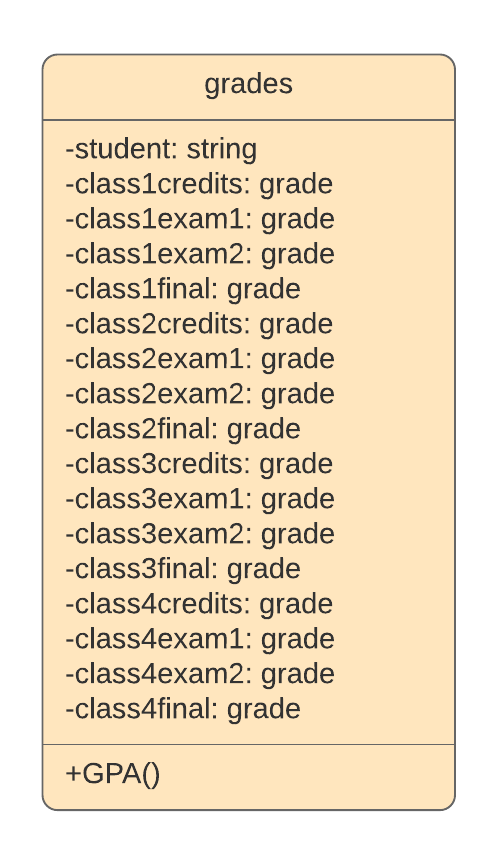


Figure 9. UML Class Diagrams