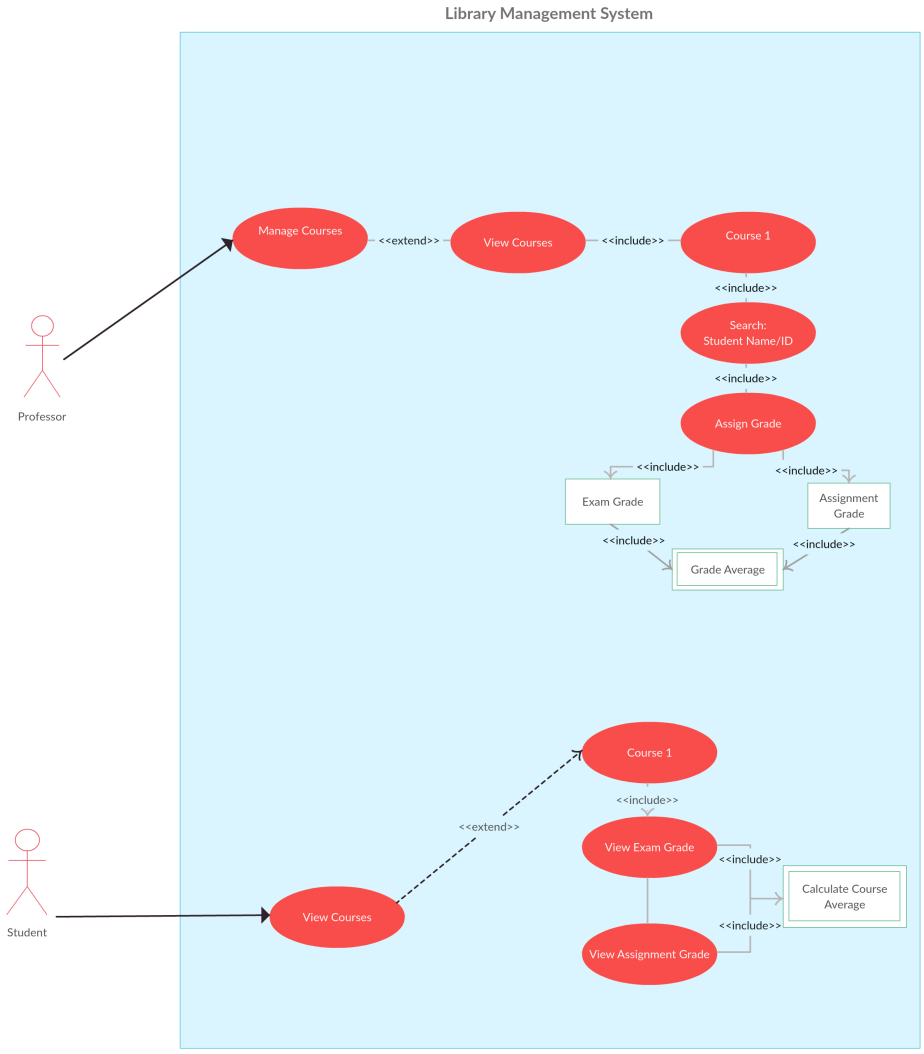
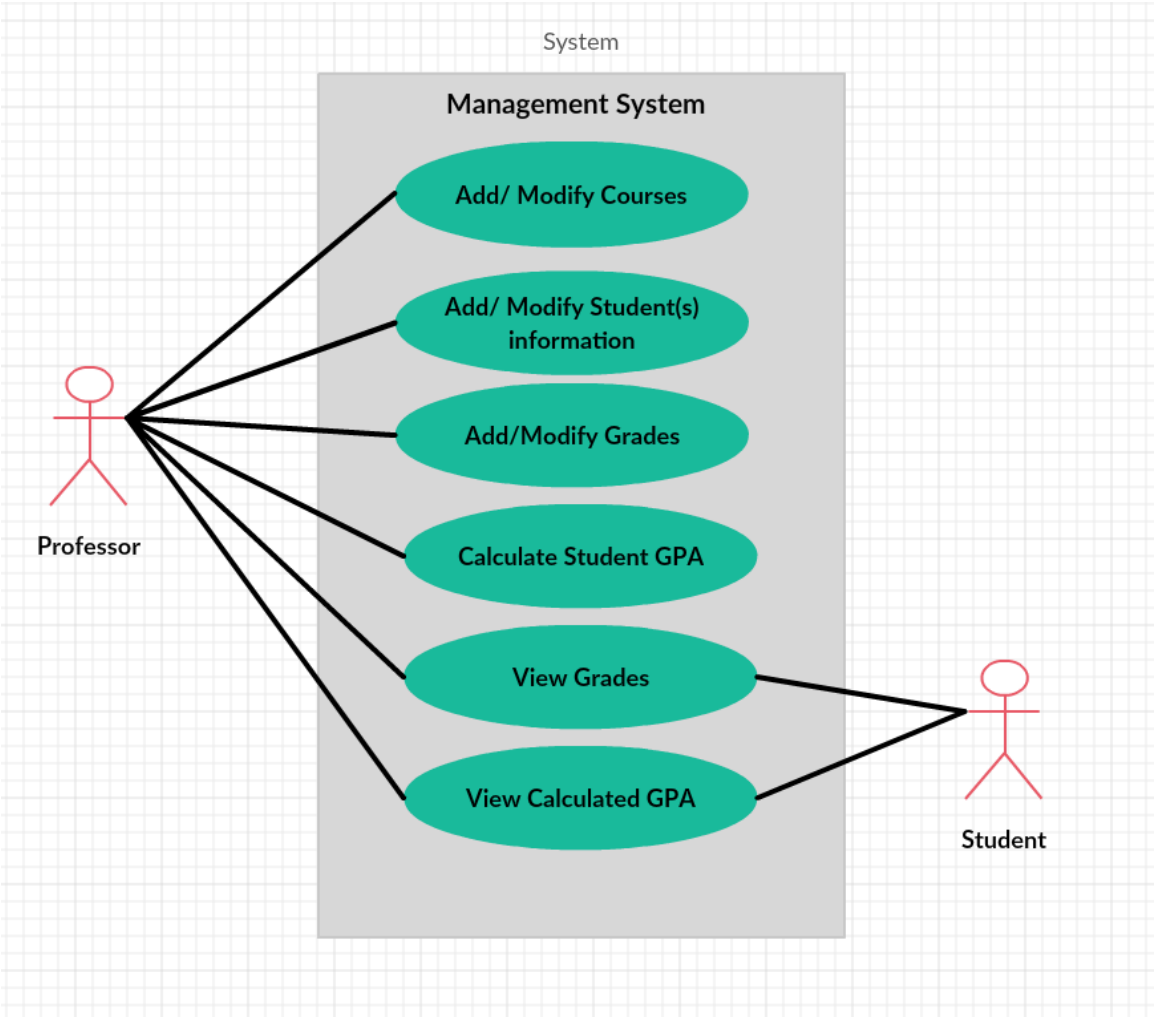
Project Report
Team consisting of 3 members.
Luis Moreno – team leader
Esteban Mejia- programmer
Rana Sadat – assistant team leader.

Type of team used : Modern Hierarchical Team.
UML diagrams used:





The following comes the functional modeling which represents scenarios

1. Professor wishes to add a new course.
2. Professor presses the add course(s) button.
3. New window appears.
4. Professor types course id number.
5. Course information unsaved until confirmation.

1. Professor wishes to delete an existing course.
2. Professor presses the delete course(s) button.
3. New window appears.
4. Professor makes selection of course to delete.
5. Course information unsaved until confirmation.

1. Professor wishes to add new student into a course.
2. Professor presses the add student(s) button.
3. New window appears.
4. Professor enters student information.
5. Student information unsaved until confirmation.

1. Professor wishes to delete student from course.
2. Professor presses the delete student(s) button.
3. New window appears.
4. Professor makes selection of student to delete.
5. Student information unsaved until confirmation.

1. Professor wishes to modify student(s) information.
2. Professor presses the modify student button.
3. New window appears.
4. Professor makes changes to student information.
5. Student information unsaved until confirmation.

1. Professor wishes to see the calculated GPA.
2. Professor presses the course in which student is in.
3. New window appears.
4. Professor selects the student.
5. New window appears.
6. Professor clicks the calculate GPA button.
7. Calculated GPA score appears in GPA box.

1. User wishes to see exam score.
2. User presses the course button in which exam score is desired.
3. New window appears.
4. Exam score appears next to the corresponding exam.

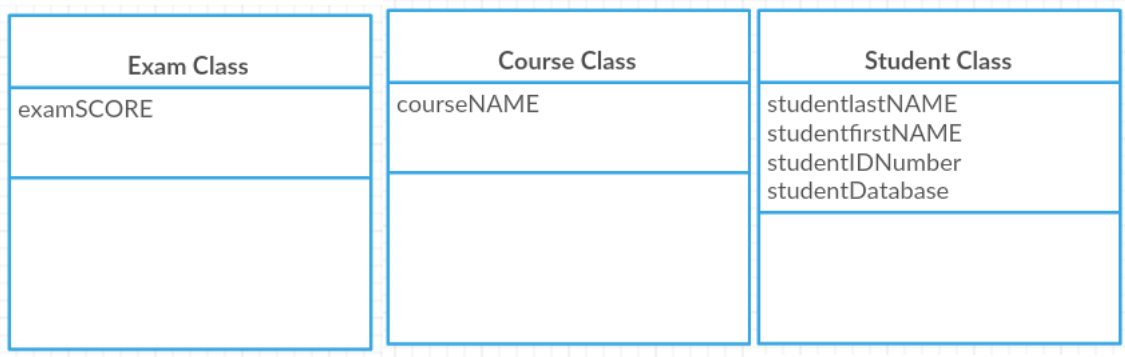
1. User wishes to see calculated GPA.
2. Use presses the course button in which GPA is desired.
3. New window appears.
4. GPA score appears in GPA box.

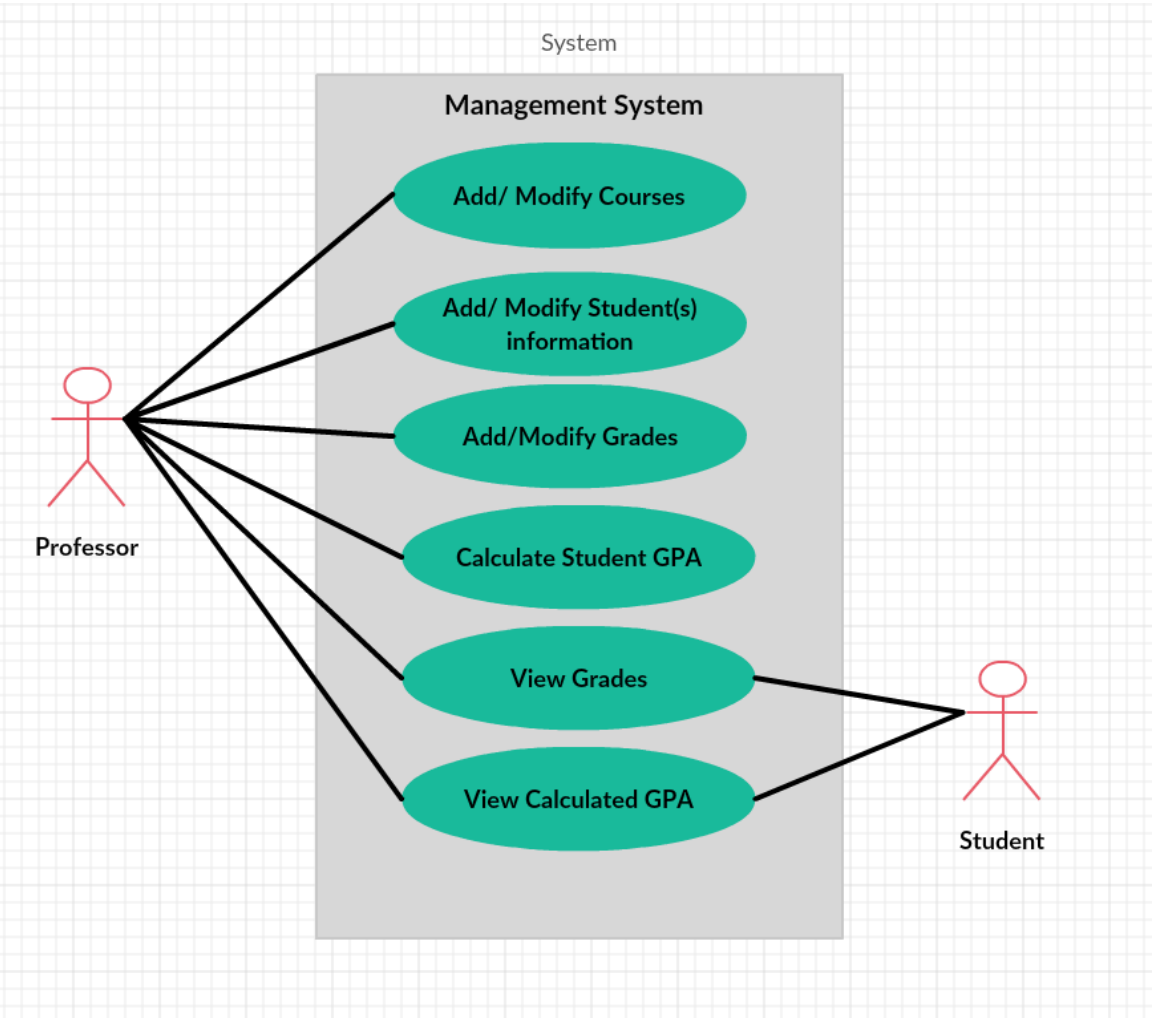
1. Professor wishes to modify exam score.
2. Professor presses the edit exam button.
3. New window appears.
4. Professor enters new exam score.
1. Score information unsaved until confirmation.

The software system only stores and retrieves students' partial information in the current semester and other basic information including **student's name**, **student's ID**, **registered courses in the current semester**, **each exam's score in one course**, **GPA calculation in the current semester**

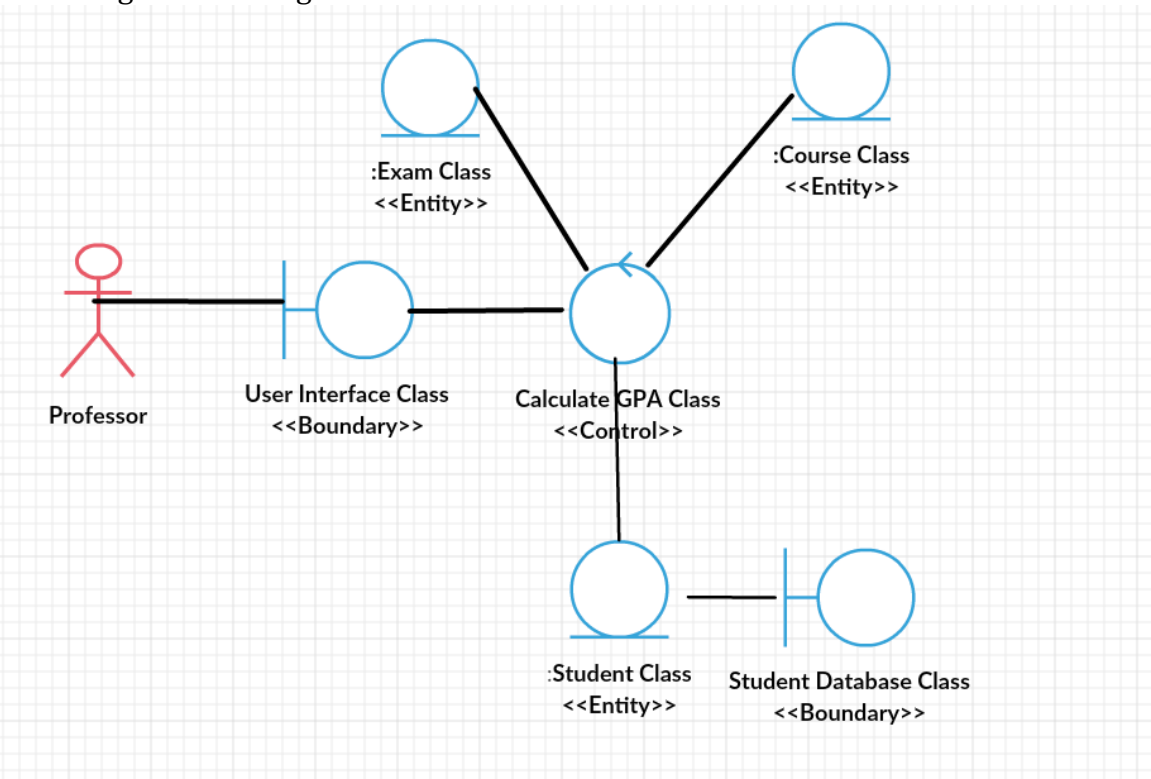
-NOUN EXTRACTION-

Initial Entity Classes: Student Class, Course Class, and Exam Class.
Initial Boundary Classes: User Interface Class, Student Database Class.
Initial Control Classes: Calculate GPA Class.

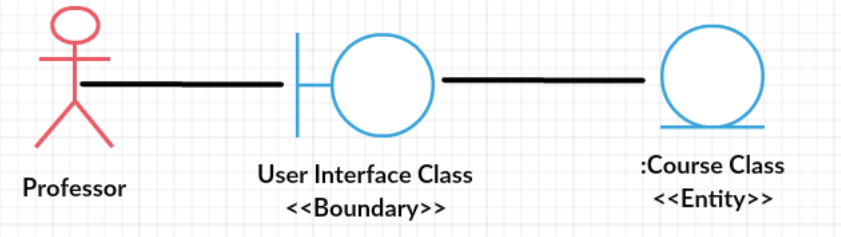

Exam Class
examSCORE
Course Class
courseNAME
Student Class
studentlastNAME
studentfirstNAME
studentIDNumber
studentDatabase


System
Management System
Add/ Modify Courses
Add/ Modify Student(s) information
Add/Modify Grades
Calculate Student GPA
View Grades
View Calculated GPA
Professor
Student

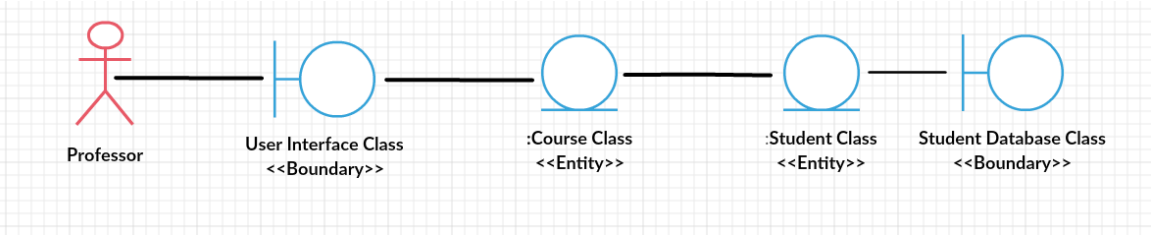
Class diagram showing the classes that realize the Calculate GPA from use case.



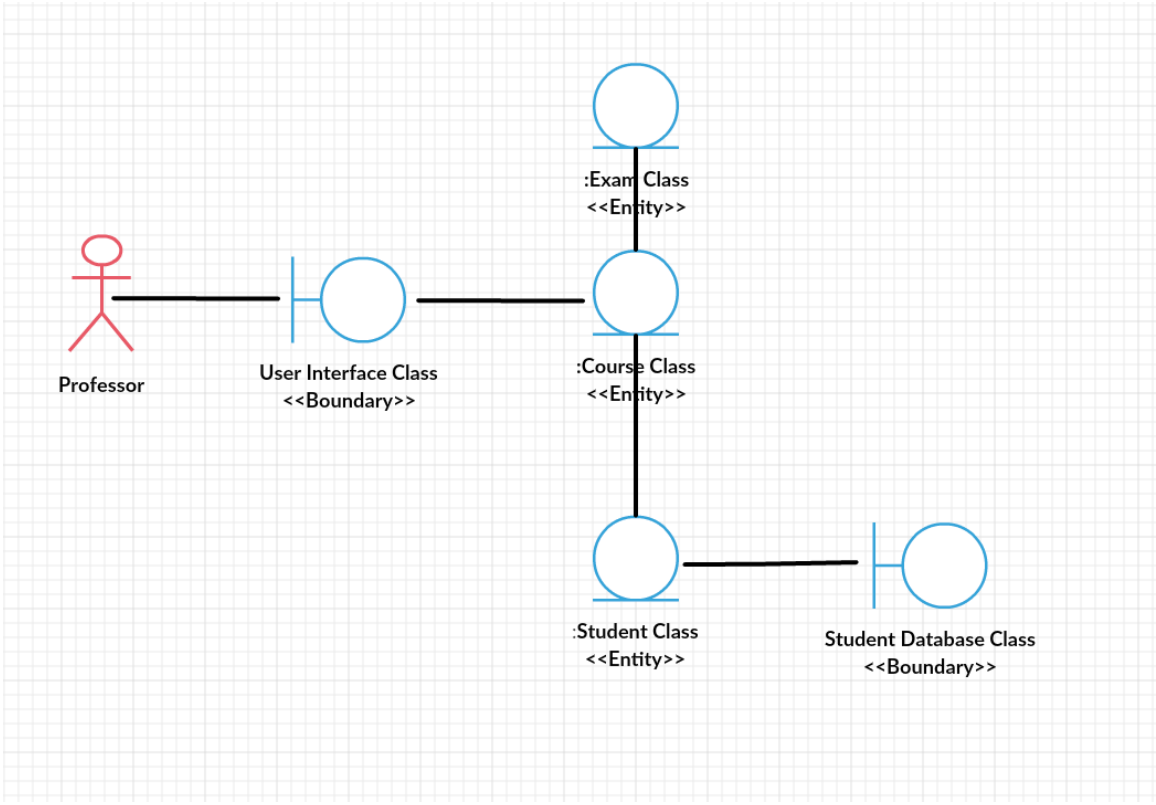
Class diagram showing the classes that realize the Add/Modify Courses.



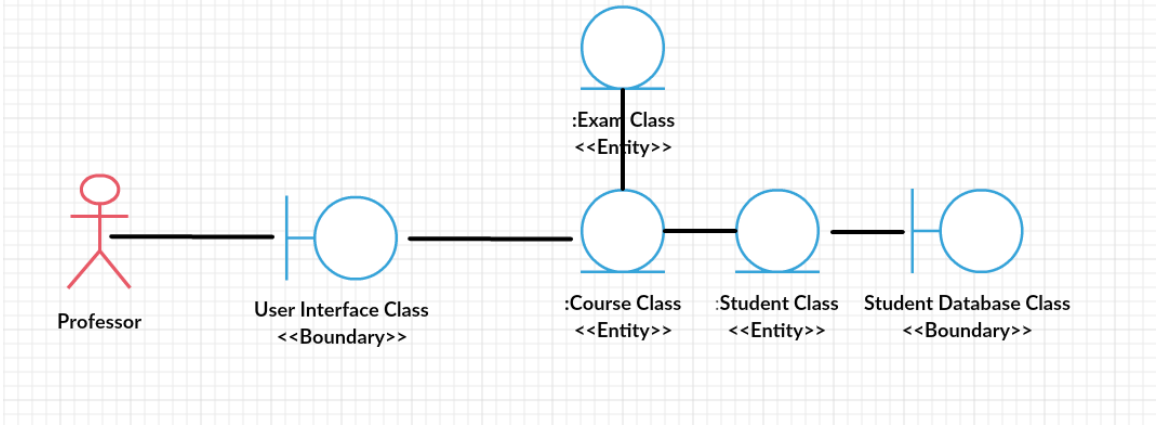
Class diagram showing the classes that realize the Add/Modify Student(s) information.



Class diagram showing the classes that realize the Add/Modify Grades.



Class diagram showing the classes that realize the View Grade.



Design architecture