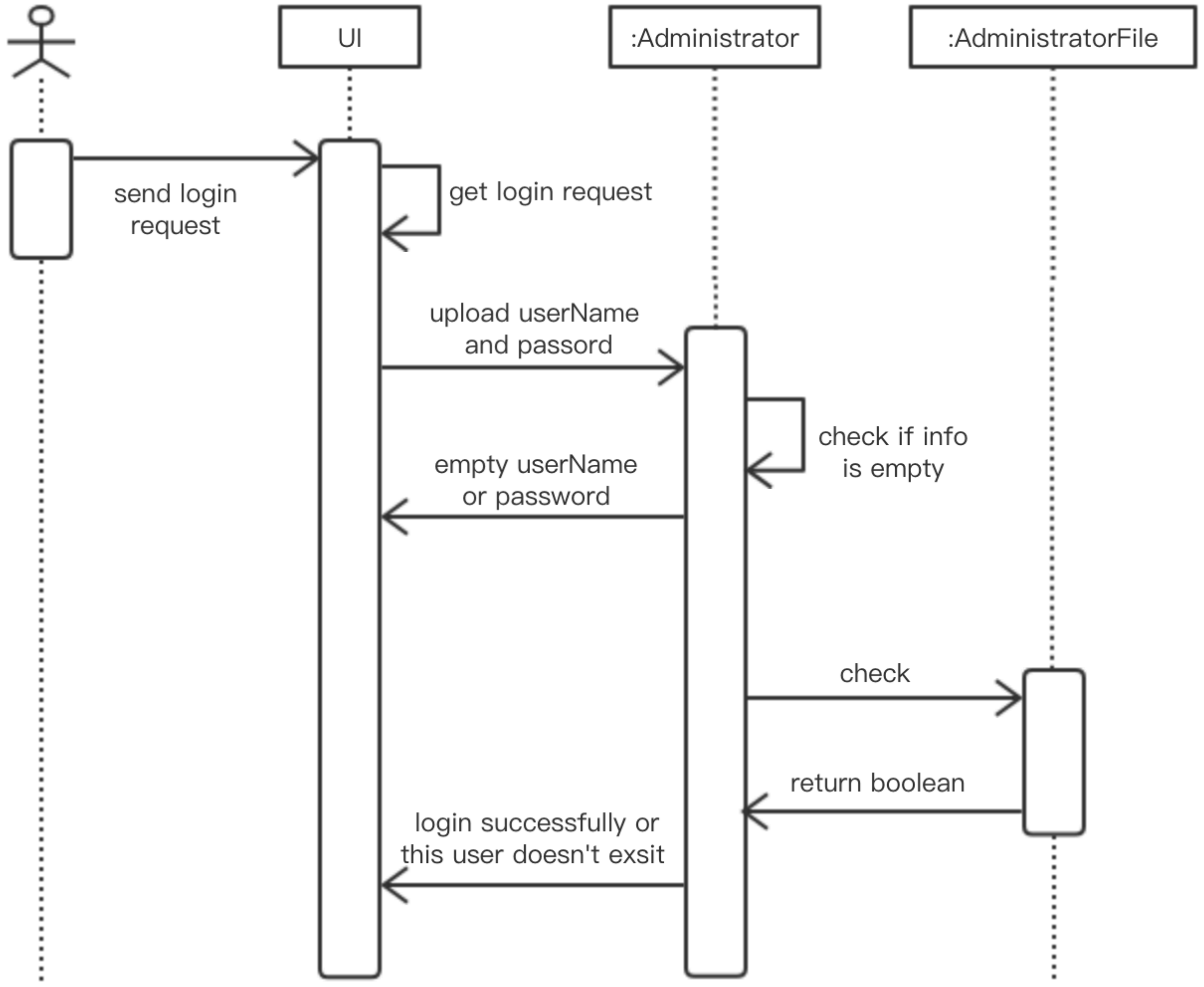
University-Model-Assignment

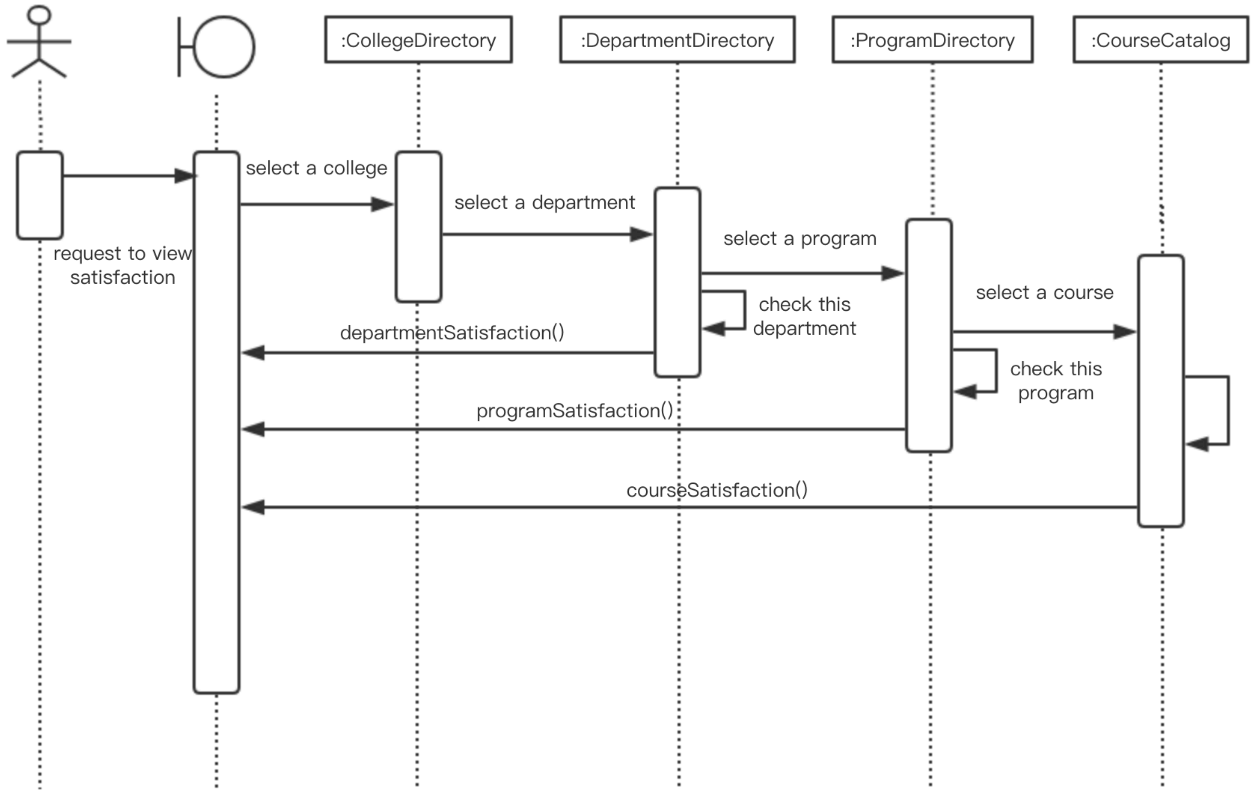
Yuxuan Yang 1389098
Runjie Li 1388946
Yuqing Wang 1443291

In this report, we create an application to help universities improve their performance. This application needs a database to perform functions. So the university should send questionnaires to their alumni every year to collect data. The data we need includes salaries, promotions, jobs and the degree of satisfaction on every academic unit. This application will automatically rank those data. Users can check these indicators (average salary, overall satisfaction, etc.) on any levels (college, department, program, courses).



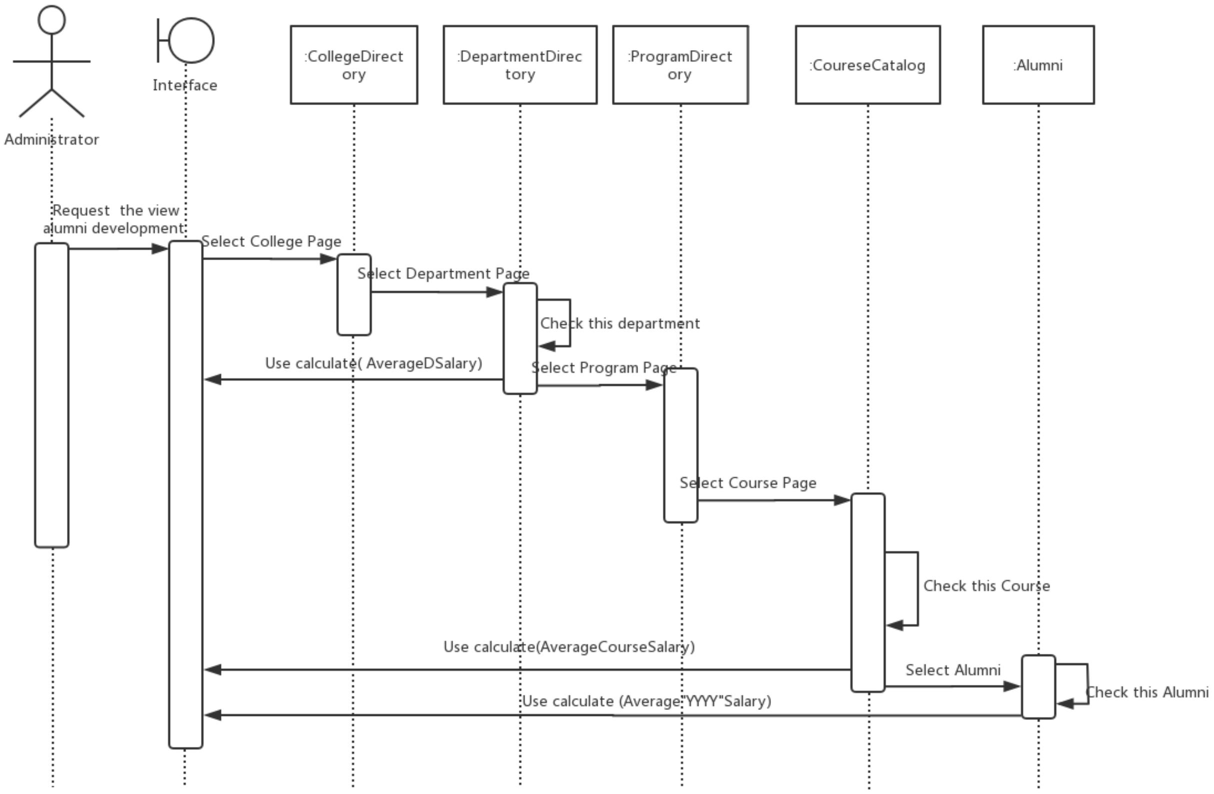
Login Sequence Diagram

In the sequence diagram <login>, the user needs to click the login button to send a request to the UI. When the UI get a login request, it will upload username and password to class Administrator and this class will check if the input is empty. If it is empty, the class will return a warning to the user. If it's not, the Administrator will send this information to AdministratorFile (a database). AdministratorFile will check if this username and password exist, and return a boolean result to the Class Administrator. The administrator will approve or deny the login which depends on the boolean result.



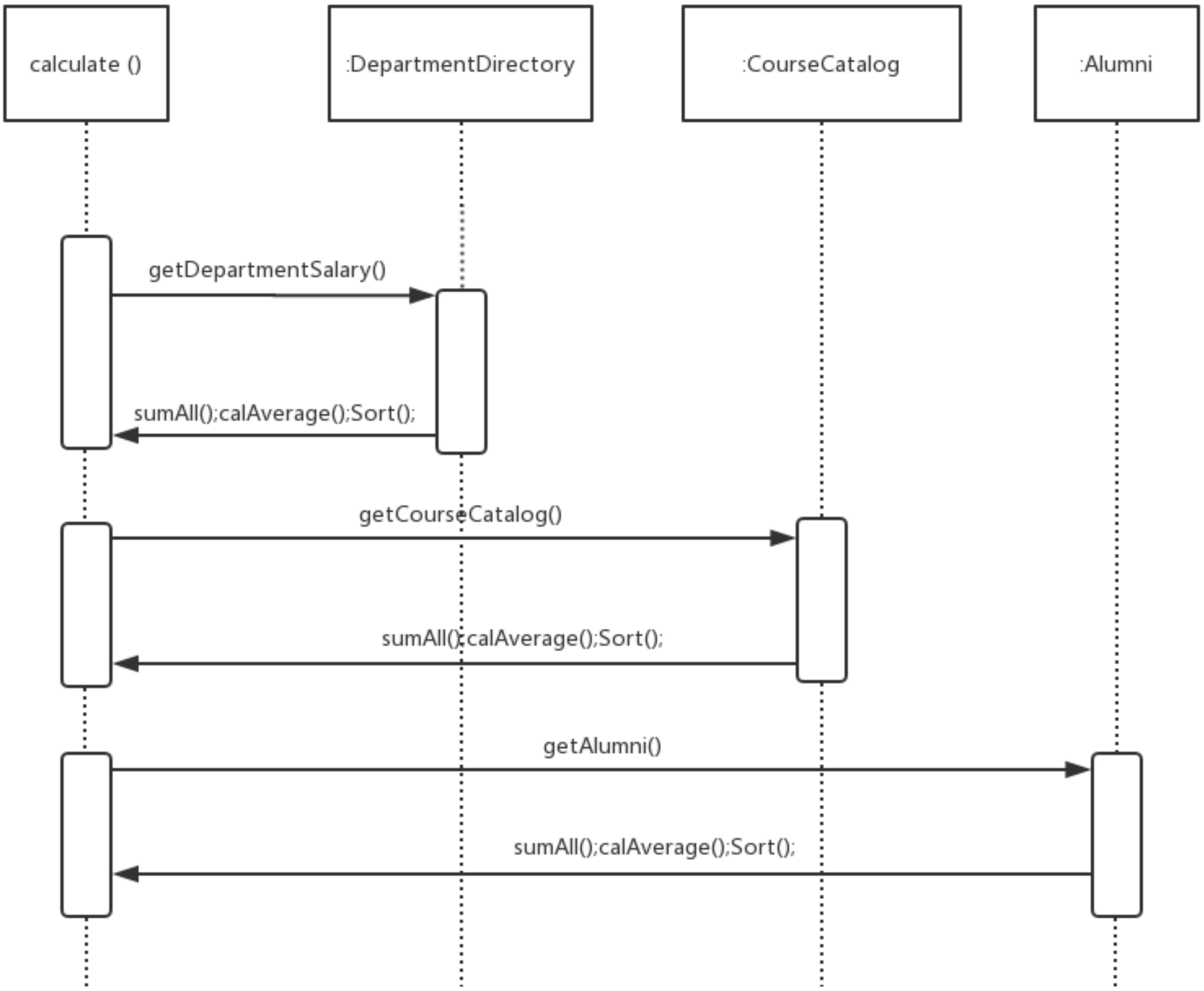
Check Satisfaction Sequence Diagram

In the sequence diagram <check satisfaction>, the user needs to click the view button to send a request to the UI. When the UI get this request, the main page appears. The user can select a college on this page. When a college has been selected, this college page appears. There's no degree of satisfaction for a college, so the user needs to go ahead and select a program under this department. Similarly, when a user gets into a program page, the user can choose to check this program satisfaction or select a course under this department. Finally, the user can check the degree of satisfaction for any courses at this university. And the degree of satisfaction has been ranked in those pages.



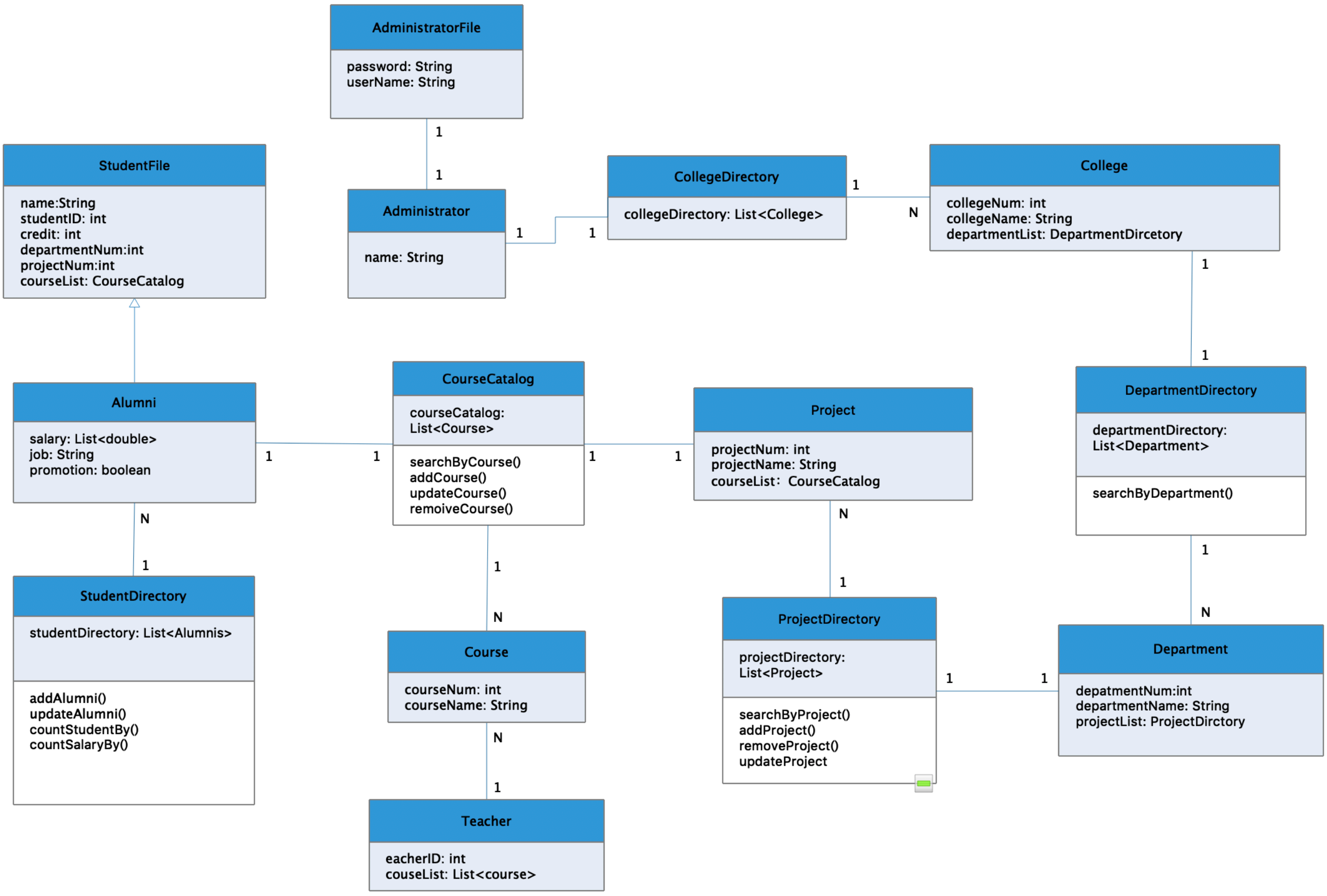
Feedback Sequence Diagram

In the sequence diagram <Feedback>, when the school administrators want to analyze the Alumni's performance in 5 years. They can use the Interface to choose which College they want to view, and in the college, they can choose different department, in department they can use calculate method to get average salary of this department's students. And they can choose different program in which they can select particular course they want to analyze. And after entering different courses, they can use calculate method to get average salary of students who have registered this course. What's more they can also use calculate method to measure graduates' average salary in certain year.



Calculate Sequence Diagram

In the sequence diagram <Calculate>, we use calculate() method to calculate and measure average salary of each student or course or department. We use sumAll() method to add all data we need and use calAverage() to get average salary.And we can use sort() method to rank average salary by different parameters to reflect the performance of this course(or student or department).



Object Model

In the object model, we show all the major attributes and methods. we have fourteen classes, at the top of hierarchy is the CollegeDirectory class, which contain all the colleges in one university. One specific college has a department directory that owns multiple departments. In a same pattern, we also create Program class under the department, and Course class under specific program. We are breaking the whole big object into small piece, in older to help university to measure their education quality in the smallest academic units. Because there are many academic units, so we create different search method under each academic units' catalog and directory. Also, university's administrators can add, remove and update the academic units.

As for the students, the most important data source in the application, we divide into two parts, one of the part may rarely change by time and have important attributes for future evaluation, which is StudentFile class. The other class is Alumni, class Alumni extends from StudentFile, and have four attributes: salary, job, promotion, satisfaction that may change every year. In addition, every student has a course catalog. And a teacher class, show the classes the teacher teaches.

The player of the application is the administrator, administrator can monitor all the data from students and make a decision whether they want to change the course, adjust the program. The class administrator has the attribute collegeList, which is the type of CollegeDirectory. So far, everything components connect together to make a functional system.