

# The Design and implementation of an alternative Coursera platform

Dure date: 10/27 at 11:59PM

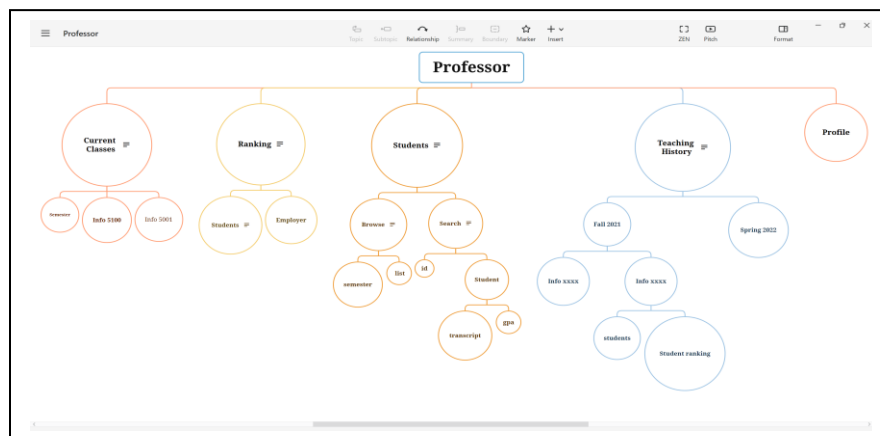
**Format:** Team project involving 2-3 of your peers.

The objective of this assignment is to instill in you the engineering techniques for turning an object model into a machine for information gathering and data aggregation and analytics. We want to use software engineering techniques to increase access to education, and reduce tuition cost while improving the quality of education anywhere in the world. We want to hold people accountable for improving the quality of life for the masses through education, learning to learn, and feedback.

Consider the following digital educational platform where the professor is at the center of students' learning (not the university). The platform will be coursera-like with the following requirements:

The system is totally decentralized where professors own and benefit from their teaching talent and experience not the educational institution.

- The professor manages their own courses as well as what they want to teach and when to teach them. Students sign up as they please.
- The professor has full autonomy. Likely the professor will be offering courses in their specialty.
- A reputation index will be available to help aspiring students decide which courses fit best.
- A professor will join the service and operate remotely from anywhere in the world.
- Professor services are visible and accessible from anywhere in the world, using tablets, browsers, smart devices, etc.
- To earn their degree, students must fulfill course requirements which means courses must be taken from many professors (residing anywhere in the world)



## Uses Cases:

1. System startup with all the initial configurations all worked out. These include universal student directory, a universal certification authority to say if a student is ready to graduate. A student is ready to graduate with an MS degree if they take 8 courses max. A directory of available courses and associated search capabilities will be in place to serve students.
2. A professor can sign up to the service and then able to manage their course catalog (1 or 2 courses) and semester-based course schedule.
3. Students would sign up to use the system. A transcript is creating accordingly.

4. Professors create course schedule every term with the courses they plan to offer for that term
5. Students browse the courses that are available from any professor anywhere in the world. They can search by professor name, topic, region, language, etc.
6. Students decide to register for classes. They will search for topics and register the courses they want to take. They should be able to see the professor rating before they register.
7. Students declare they want to graduate. The platform would review their transcript and indicate if the student satisfy graduation requirements.
8. A dashboard that enables platform owners to collect performance data of different varieties.
9. Any additional innovative idea you see appropriate (bonus points as below).

### Deliverables

1. Report outlining your proposed solution.
2. Architecture diagram showing the new business model and how it is different from traditional university setup.
3. A class diagram showing the changes to the university model to support the new capabilities. This diagram must include the additional methods and attributes required to deliver the results.
4. A complete running application for all the different use cases outlined above.
5. Overall assessment and opinion on whether such digital educational platforms of the future will make education easily accessible and affordable to the less fortunate.

### Grading:

All implementations must use the java package for role management as foundation. **Strict adherence to the approach outlined in the java package is a MUST. Projects that do not adhere to the this restriction will not be considered for evaluation.**

1. [Basic, (B-, B+)]
  - a. System configuration, populating the model with data. A population of 10 professors and their students. An average of 10 students per class. Successful integration of the university model with the UI reference architecture given. Good look and feel. Organized code that is easy to understand. A minimum of 8 user-driven use-cases must be implemented.
2. [Extras (B+, A-)]
  - a. In addition to the Basic features above, integrate employers and loop back their feedback to the professors. Implement some extensions to show how your new features will improve the quality of education for students and society benefit. You must implement 12 user-driven use cases.
3. [Innovative (A-, A)]
  - a. In addition to the extras in case 2 above, implement creative ideas and innovations that go beyond the requirements. For example: 1) you might want to focus on employer engagement since this could have huge impact on the quality of education the students receive. 2) The underlying digital platform could benefit from third party certification authority to approve the degree.