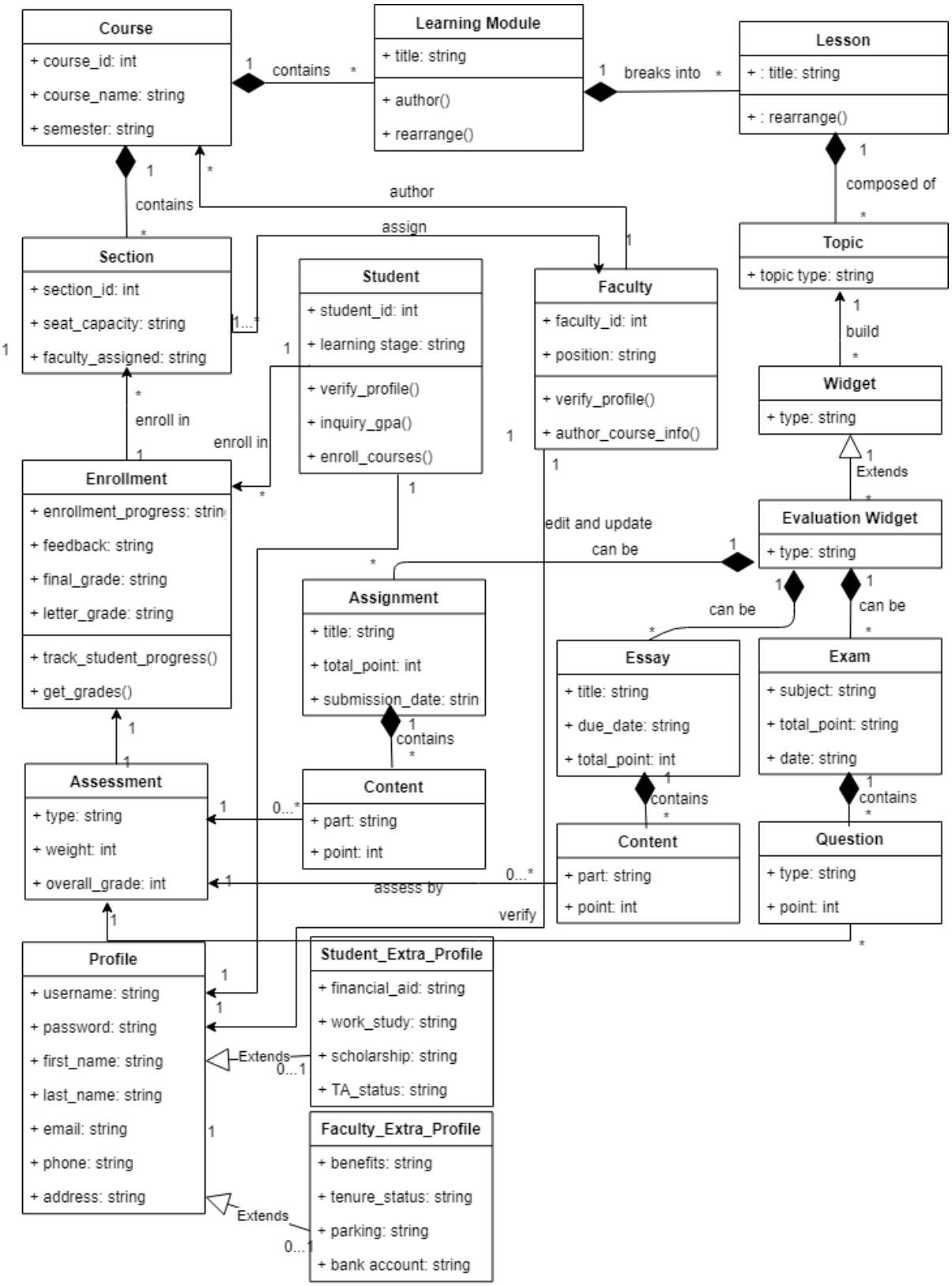
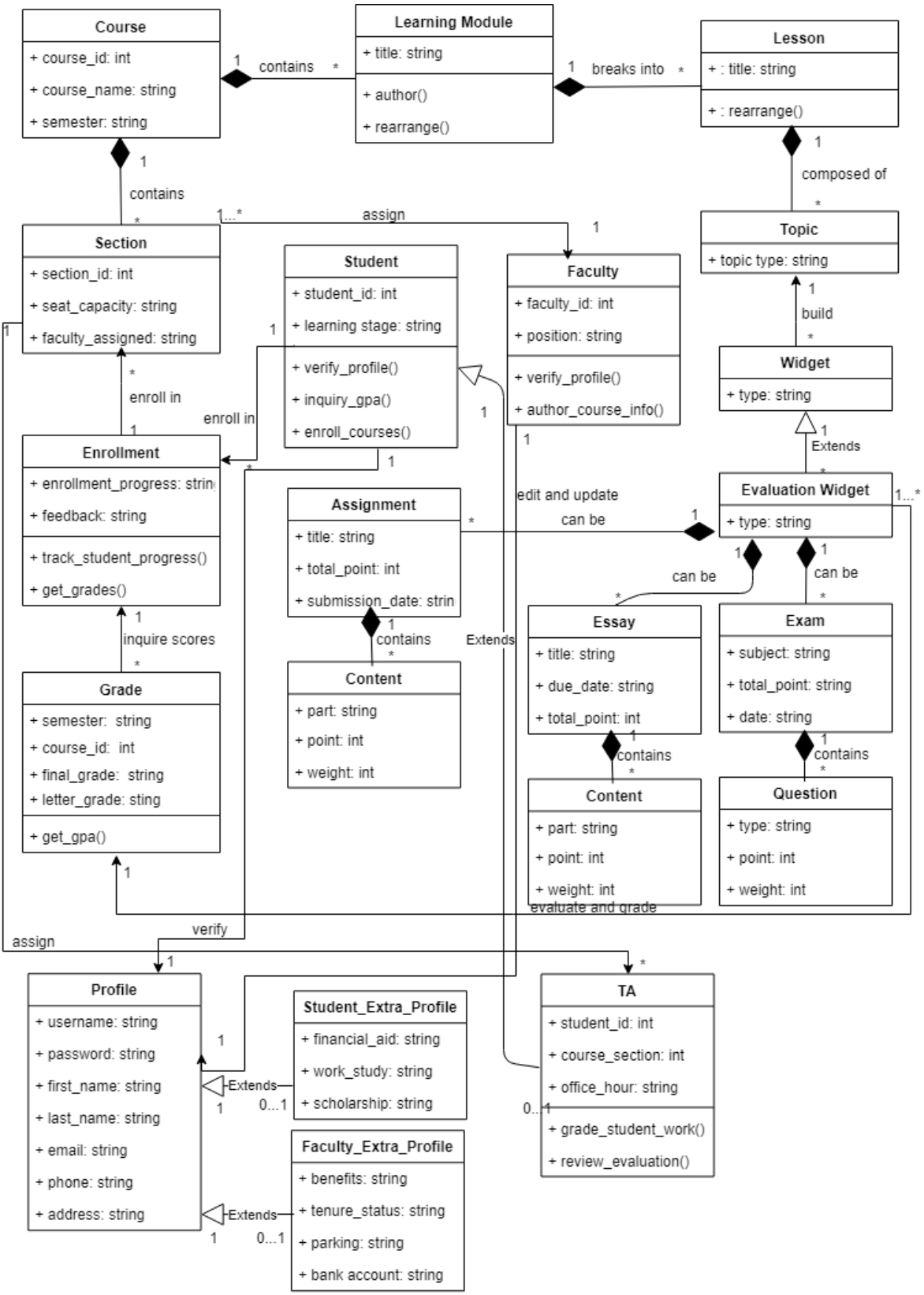
HW1

Chenxin Shi

The **naïve** diagram for LMS:



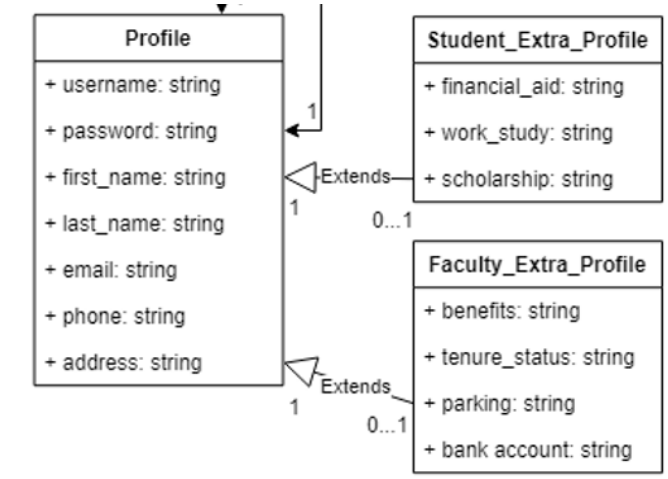
The **final** diagram:

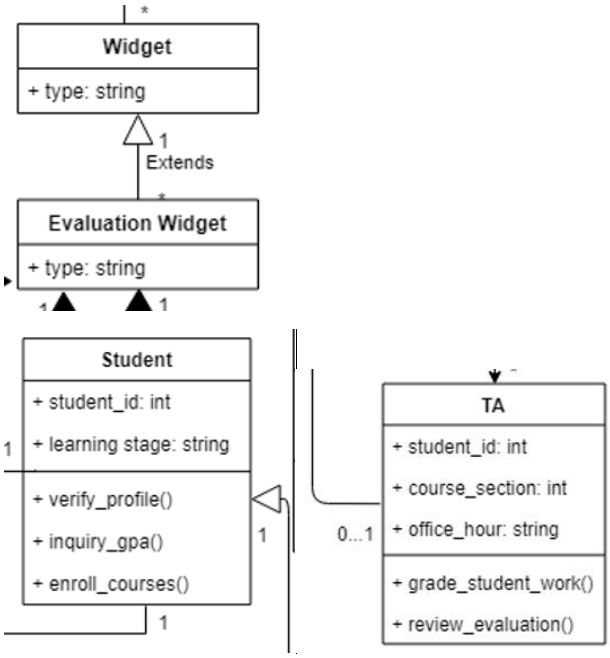


1. The candidate classes include courses, faculty, students, learning modules, lessons, widgets, topics, evaluation, exams, questions, enrollment, profile, etc. The candidate attributes include name, id,

type, content, date, email, phone number, address, etc.

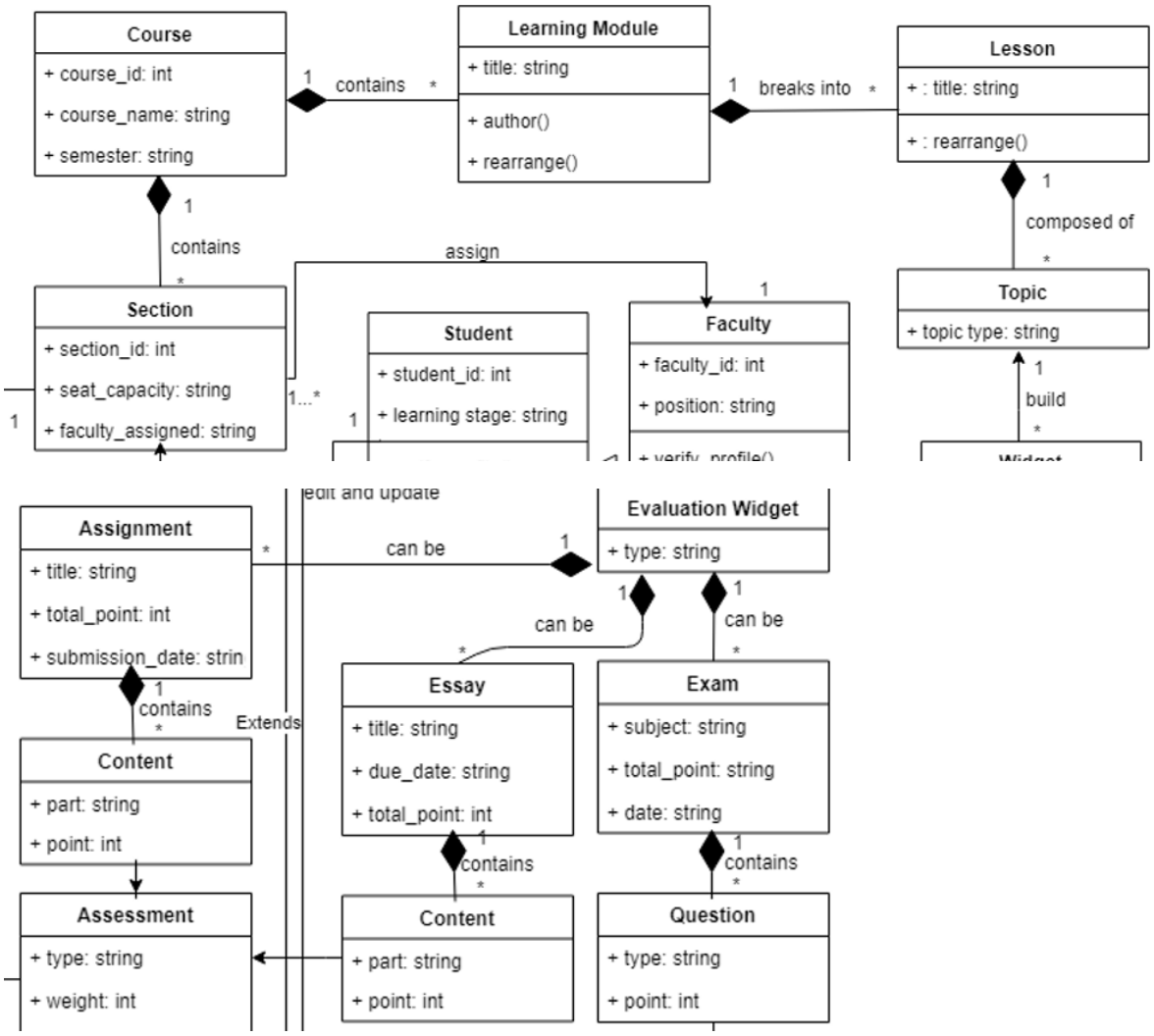
2. The candidate relation verbs include author, teach, contain, verify, update, enroll in, assign, break into, etc.
3. The use of inheritance in the diagram:





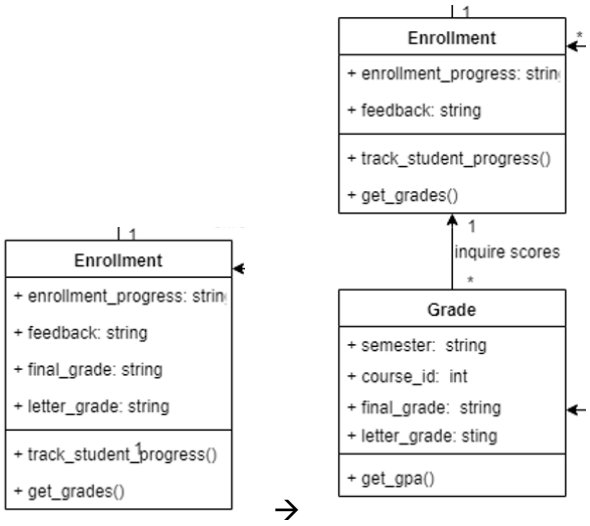
I think the profiles that students and faculty separately need to add are extensions to the general files. The evaluation widget is one of the types of widgets. And the TAs belong to students. They have the attributes of the parents. And they can still exist without the general profile. So the relationships are all generalization.

4. The use of composition:



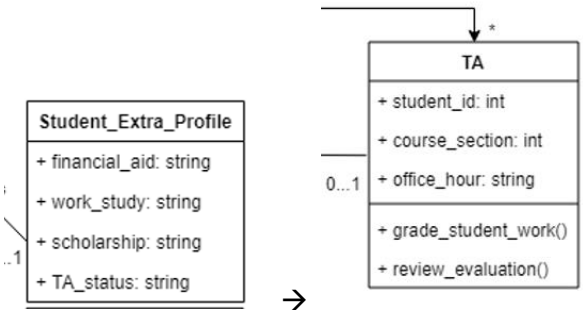
In the diagram, each course is composed of learning modules, a learning module is composed of lessons. A course can also be composed of sections. In the second screenshot, a evaluation widget can be an assignment, an essay, or an exam. An assignment or an essay include several parts, which have different points, and an exam is composed of different questions with different points. The subclasses cannot exist if the parent classes are destroyed. So I think the relationship should be composition.

5. In the naïve diagram shown above, I first think that the information of TA is an attribute of the class Student or the Student_Extra_Profile, in the work_study attribute. Then I think it should be a class other than an attribute since it contains other attributes and relations with other classes like Section.



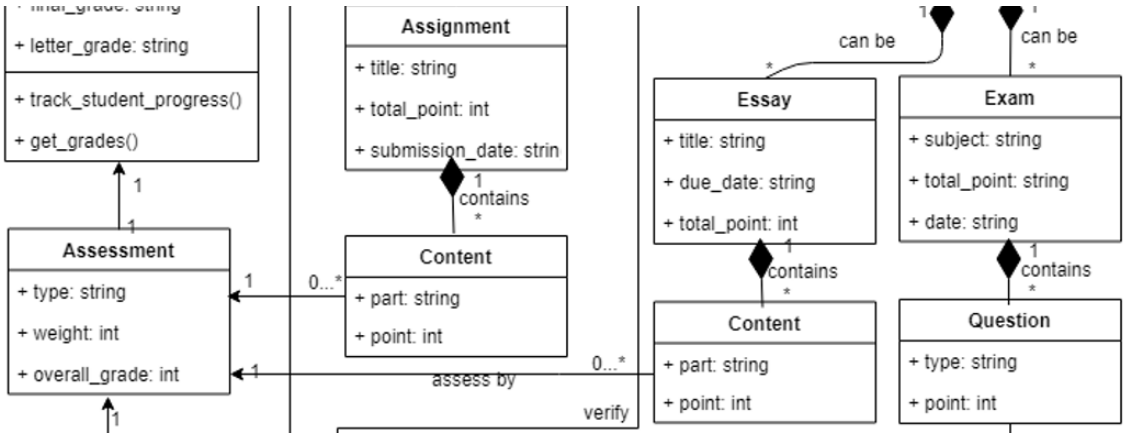
naïve final

So is Grade. I first set the final_grade and letter_grade as attributes of the class Enrollment, then I find the grades have relations with evaluation widgets. So it should be a class with its own attributes, and be connected with evaluation.



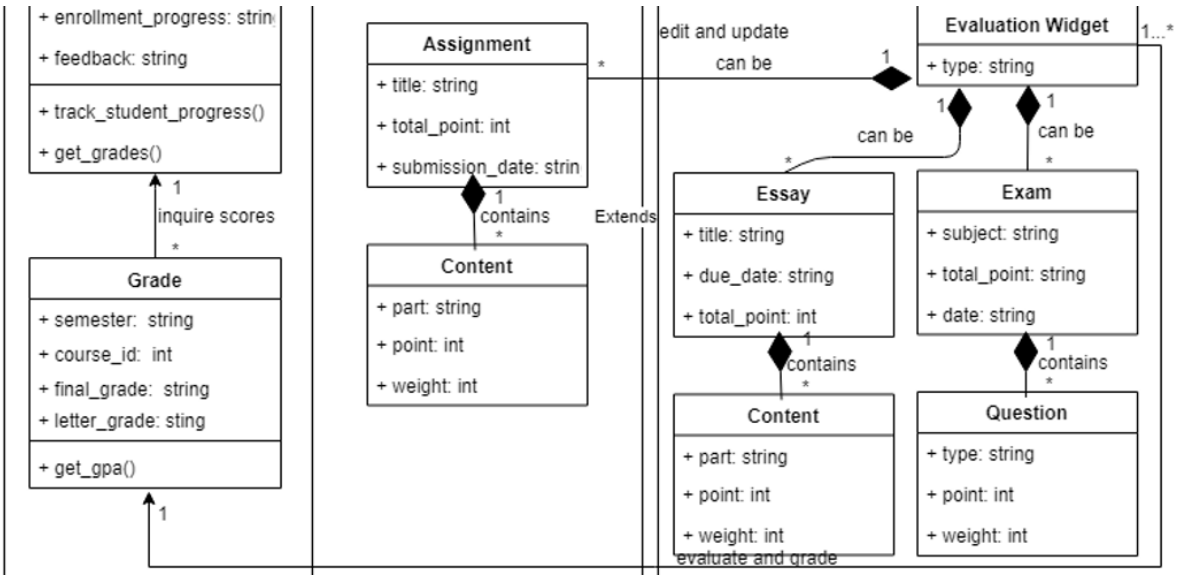
naïve final

Besides, I used a class Assessment between the classes Exam, Assignment and Essay, cause I think there should be some method to compute the weight. But then I think it would be better to convert this class to an attribution. So I add the attribute weight to the three classes, omit the class Assessment and connect the class Evaluation Widgets with Grade.



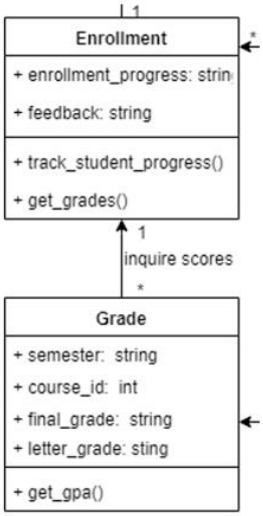
Naïve

->

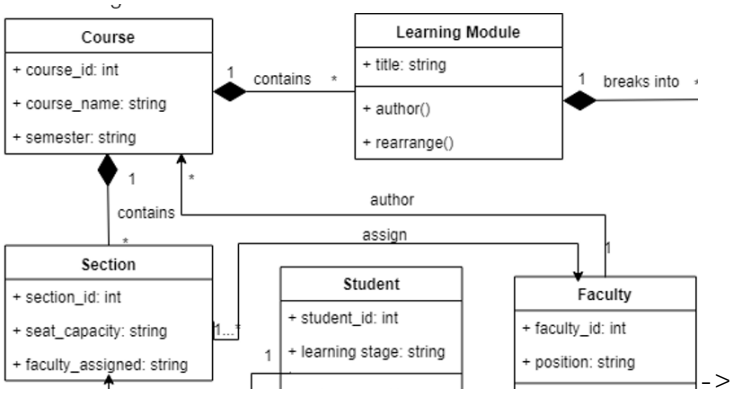


final

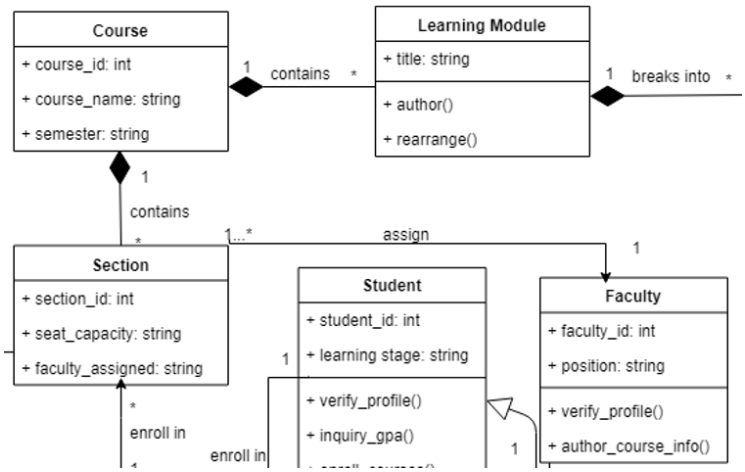
6. The data types are marked in the diagrams as following:
   + student_id: int
7. The cardinalities are marked in the diagrams as following:



8. In the naïve diagram, I first connect the class Faculty with the class Course since the teachers can author the course information. But then I find out that the Faculty class has been related to Course through the class Section. So I remove the connection as shown below.



naive



final

The relationships between the Exam, Assignment, Essay and Grade have been removed too because of the redundancy. So is the noun Assessment.

9. At first, I directly connect the class Student with Section to show the relationship that the students enroll in the particular section. Then I find out that it is a many-to-many relation, which should be avoided. So I add the class Enrollment to fix it. It is a one-to-many problem now just as shown in the diagram.