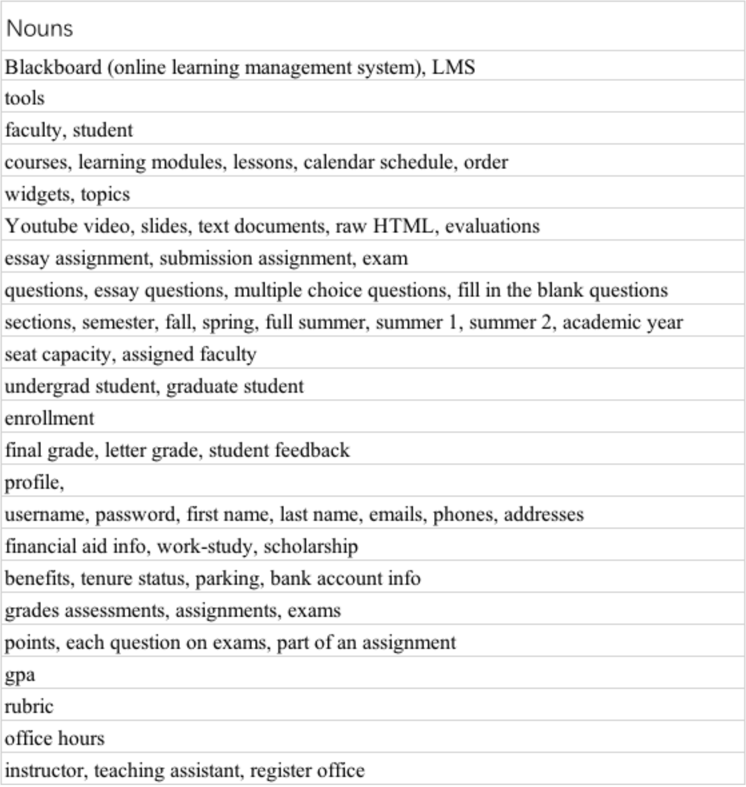
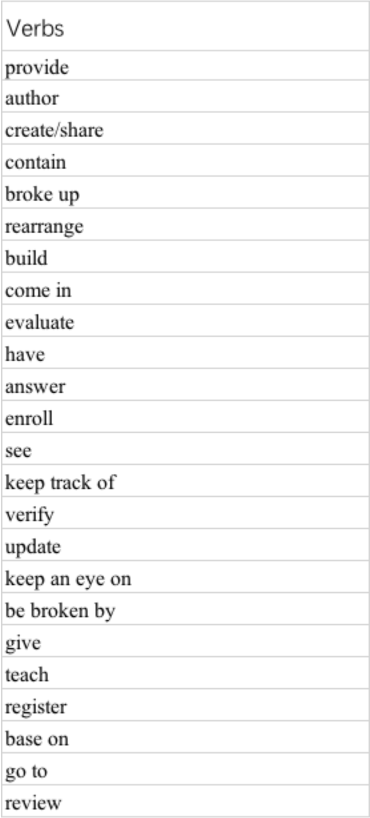
1. **List nouns that are candidate classes or attributes.**



Nouns
Blackboard (online learning management system), LMS
tools
faculty, student
courses, learning modules, lessons, calendar schedule, order
widgets, topics
Youtube video, slides, text documents, raw HTML, evaluations
essay assignment, submission assignment, exam
questions, essay questions, multiple choice questions, fill in the blank questions
sections, semester, fall, spring, full summer, summer 1, summer 2, academic year
seat capacity, assigned faculty
undergrad student, graduate student
enrollment
final grade, letter grade, student feedback
profile,
username, password, first name, last name, emails, phones, addresses
financial aid info, work-study, scholarship
benefits, tenure status, parking, bank account info
grades assessments, assignments, exams
points, each question on exams, part of an assignment
gpa
rubric
office hours
instructor, teaching assistant, register office

2. **List verbs as candidate relations between classes**



Verbs
provide
author
create/share
contain
broke up
rearrange
build
come in
evaluate
have
answer
enroll
see
keep track of
verify
update
keep an eye on
be broken by
give
teach
register
base on
go to
review

3. **Generalization/specialization (inheritance, if applicable, explain) - show parts**

of your diagram that specifically illustrates the use of inheritance

Answer:

In UML modeling, a generalization relationship is a relationship in which one model element (the child) is based on another model element (the parent). Generalization relationships are used in class, component, deployment, and use-case diagrams to indicate that the child receives all of the attributes, operations, and relationships that are defined in the parent. To comply with UML semantics, the model elements in a generalization relationship must be the same type.

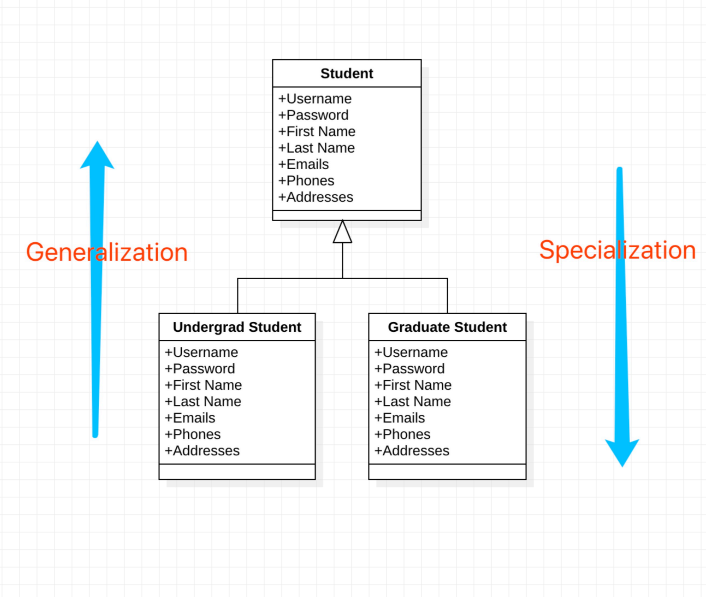
Generally, terms generalization and specialization both refers to inheritance but the way in which they are implemented make them different.

When we create a super class by extracting all common characteristics (attributes and behavior) from two or more similar type of objects, then this process is known as Generalization.

In contrast to Generalization if new classes are created from existing class to perform or implement a specific feature this process is called Specialization.

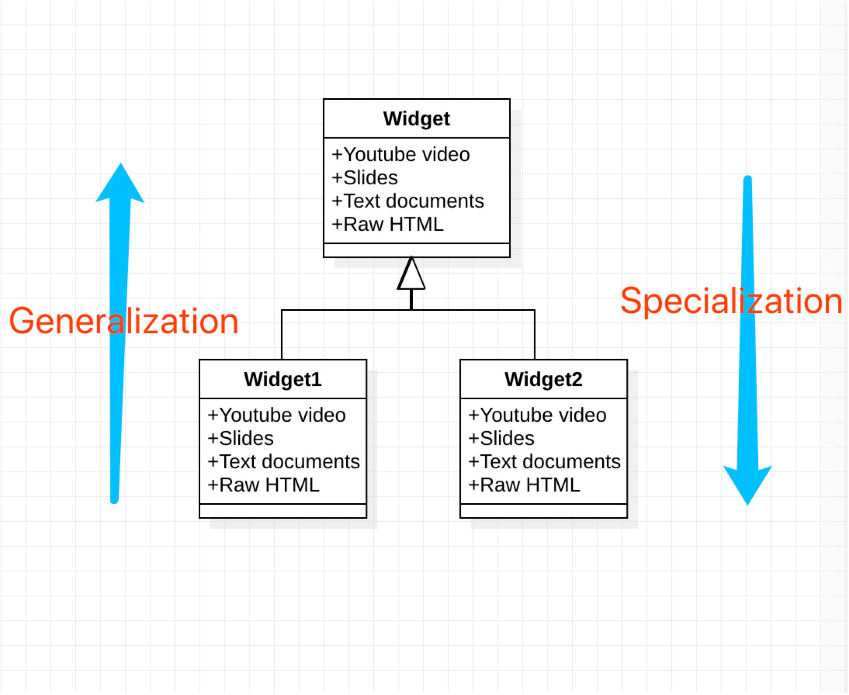
Both the concept generalization and specialization are implemented using inheritance but it’s only the order of creation of the subclass and super-class drives the concept name.

For example, in the following diagram, Undergrad student and Graduate student are inherit the same types of elements of Student, such as Username, Password and etc.



There is another example. Widget1 and Widget2 inherit the same types of elements of

Widget, such as Slides, Raw HTML and etc.



4. **Associations, aggregation and/or composition, e.g., empty or filled in diamonds (1 to * or 1 to 1..*, if applicable, explain) - capture any lifecycle dependencies between classes using aggregation or composition.**

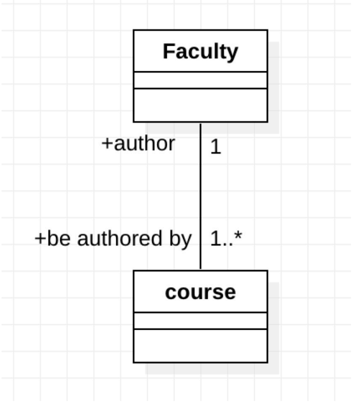
Answer:

1)Association

In UML models, an association is a relationship between two classifiers, such as classes or use cases, that describes the reasons for the relationship and the rules that govern the relationship.

An association represents a structural relationship that connects two classifiers. Like attributes, associations record the properties of classifiers.

Association relationship is any logical connection between classes. For example, faulty author course and courses are authored by faculty. One faulty could author one or more courses and one course could be authored by one faculty, so they should be one-to-many relationship for this example.

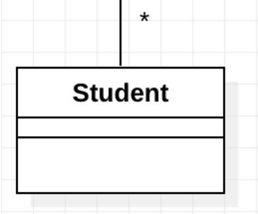


2）Aggregation

In UML models, an aggregation relationship shows a classifier as a part of or subordinate to another classifier.

An aggregation is a special type of association in which objects are assembled or configured together to create a more complex object. An aggregation describes a group of objects and how you interact with them. Aggregation protects the integrity of an assembly of objects by defining a single point of control, called the aggregate, in the object that represents the assembly. Aggregation also uses the control object to decide how the assembled objects respond to changes or instructions that might affect the collection.

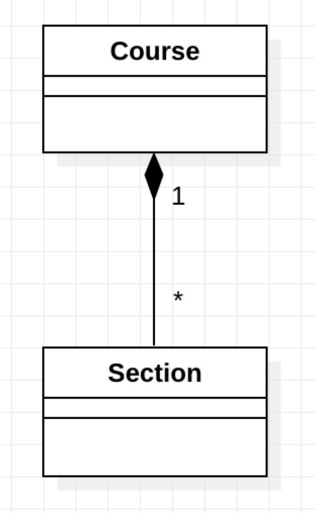
Aggregation relationship refers to that one class is aggregated or built by another class, but they are independent by each other. For example, Section contains Student, but Student is not dependent on the lifecycle of the section. If we remove Section, Student will remain. And we remove Student, Section will remain. Besides, one Section may contain many Students, and one Student could be in different sections. Thus, they are many-to-many relationship.



3)Composition

Composition relationship refers to that one class contains the other class, and the lifecycle of the contained class depend on the container class, which means the contained class will be obliterated if we remove the container class.

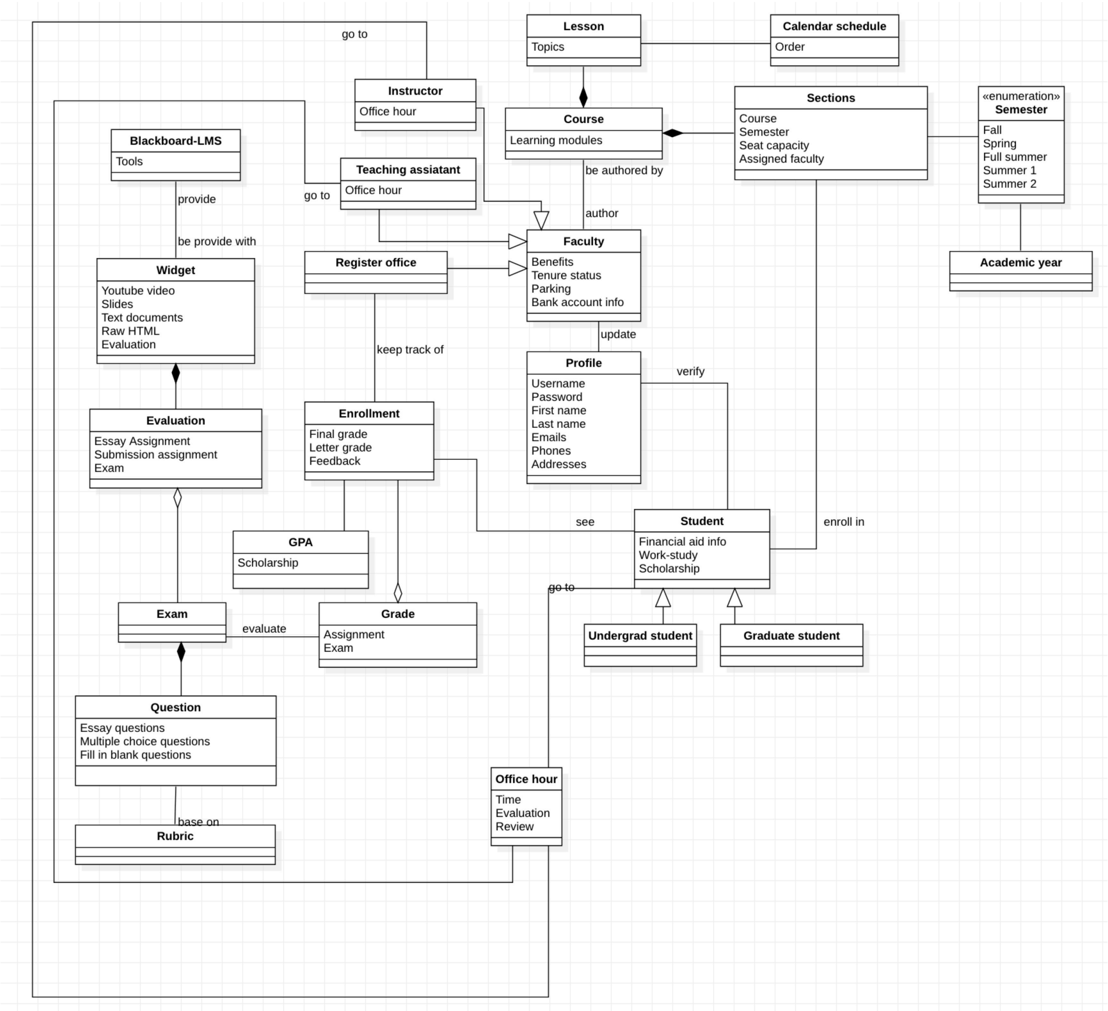
For example, Course is made of different Sections and is we remove the Course, Sections related to it will be removed too.



5. **Classes vs. attributes analysis - if you are not familiar with a particular domain, e.g., a particular industry such as education, or nuclear energy, or telecommunication, or astrophysics, you might be unfamiliar with the vocabulary and so any noun might be a potential class. You might create a 'naive' class diagram that makes no distinction between classes and attributes. As you research the domain and learn more about it you realize the relationships between nouns where some might just be describing other nouns or some are compositions of other nouns. Create a naive class diagram and diagram your process of identifying certain classes as actually being attributes of other classes. Or vice versa, where some attributes might actually be better modeled as classes. Document how the class diagram evolved from a naive first approximation to the final result. Explain your decisions and support them with relevant portions of the class diagram. Show your final class diagram as one single diagram.**

Answer:

1) Naïve diagram



a. Widget contains evaluation, but evaluation also contains several attributes, then Widget needs to be a class.

b. Since exam has several relationships with other classes, so it is better to be a class than an attribute.

c. Rubric is related with just one class, so it could be an attribute of questions, which could make the diagram clearer.

d. Academic year is related with just one class, but it is will not have direct effect on semester, then it may be a class or an attribute of sections.

e. Fall, spring, full summer, summer 1, and summer 2 semesters consist of semesters, so it is better to be enumeration of semester.

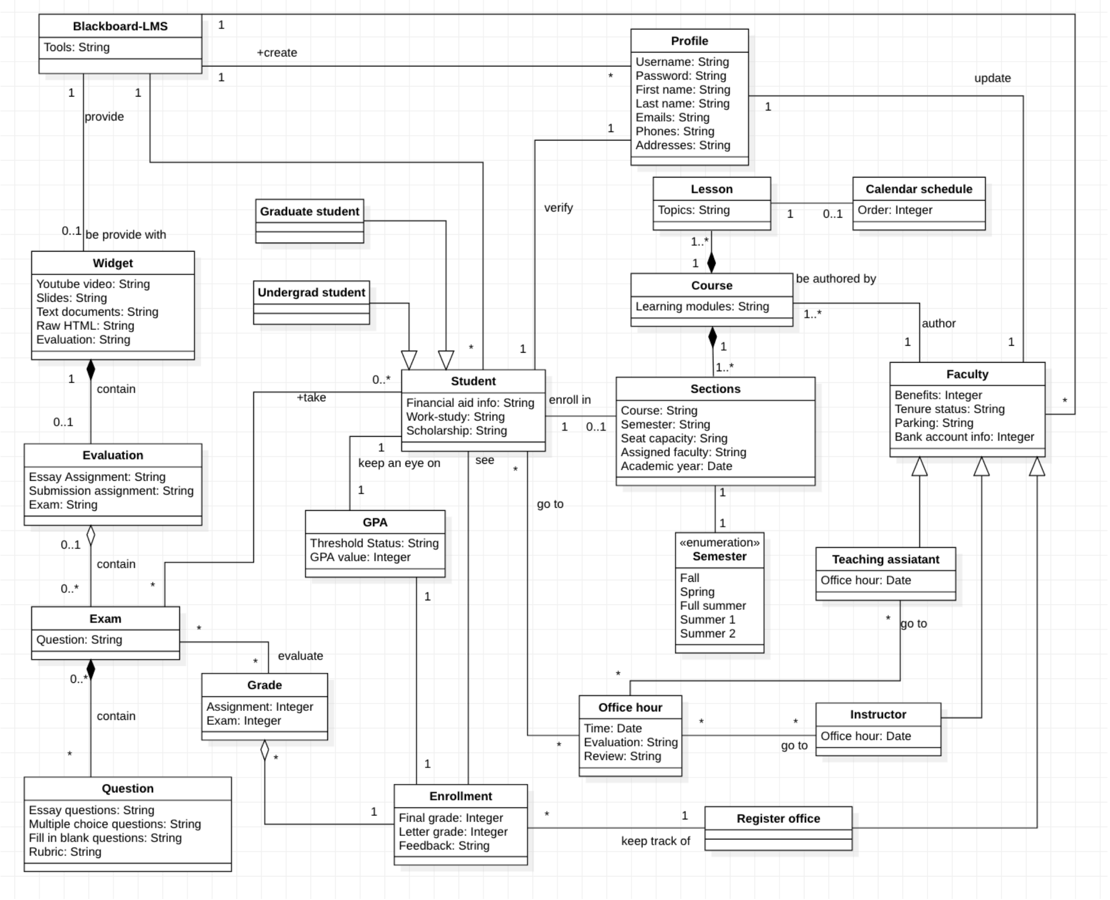
f. Faculty, student, question, profile, section, course, lesson have a lot of detailed information and the information could be attribute, then they could be classes.

g. Order of Calendar schedule will determine the Lesson, then Calendar schedule should be class.

h. Undergrad student and Graduate student both inherit element and data type of Student, then they have generalization/specialization relationship with Student.

i. Instructor, Teaching assistant and Register office inherit element and data type of Faculty, then they have generalization/specialization relationship with Faculty.

2) Final Diagram



6. Correct data types, e.g., Date, String, Integer, List, Array, Enumeration, etc.

Answer:

Data types are shown in the Final Diagram.

7. Cardinality - for every single association, show the number of instances participating in a relation

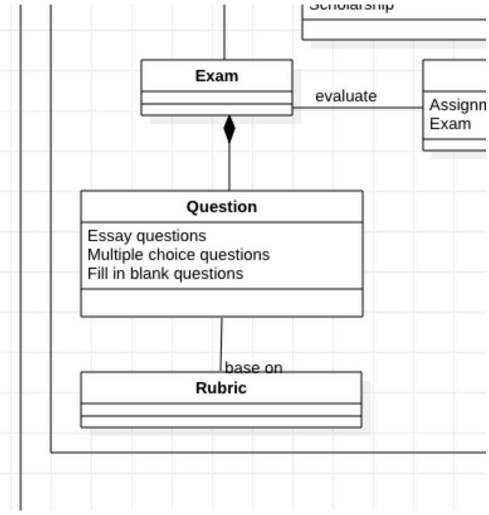
Answer:

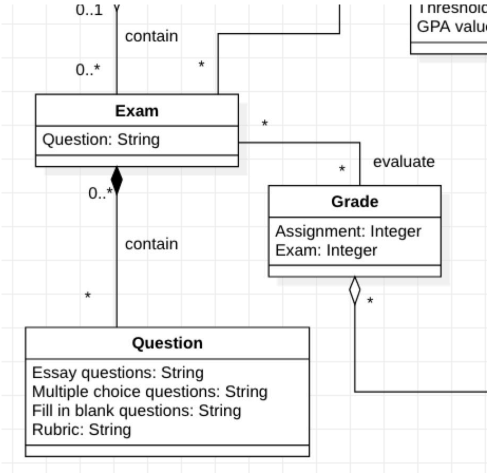
Cardinality are shown in the Final Diagram.

8. **Remove any inadequate or redundant relationships, entities or attributes (if applicable, explain) - if you identify redundant associations, entities, or attributes, explain how/why you removed it. For instance, the problem statement might have irrelevant information that you might need to ignore. Also, the text might describe contradictory or ambiguous descriptions. Finally, the statement might use different terms to refer to the same thing. Make sure you make a compelling argument for your decisions to ignore a particular noun or verb as irrelevant or redundant or an overloaded term.**

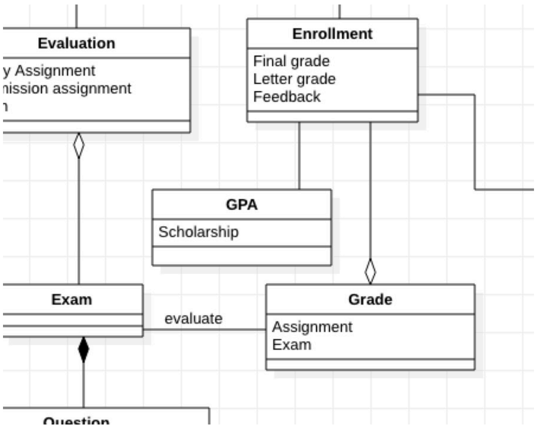
Answer:

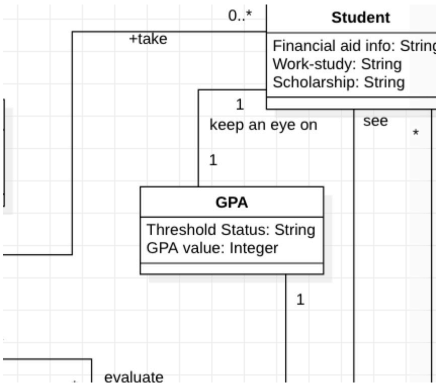
1) For Rubric, it only has relationship with Question and it could be one attribute of Question.



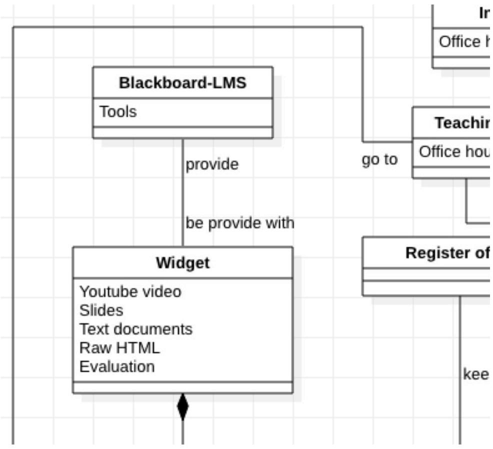


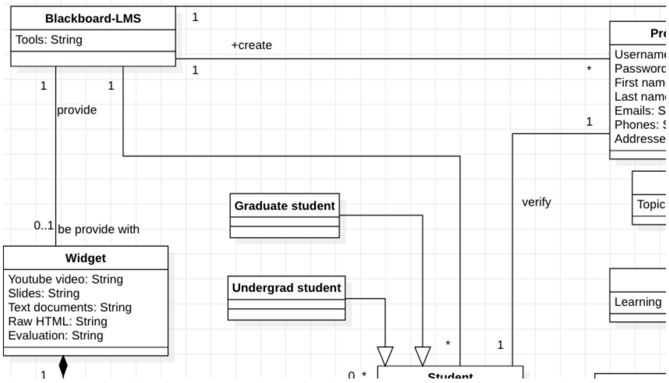
2) For GPA, it is used for evaluation of Scholarship but it does not contain Scholarship, then I change the attribute in it.





3) The biggest change is Blackboard-LMS. Because at the beginning, I thought it is just the platform to display toolsm, but actually it will create a profile for each Student and Faculty. Beside, the lines to display the relations need to be rearranged, then I change the position of many classes on the diagram.





9. Reify (if applicable, explain) - if you have association classes or other UML artifacts that don't readily map to relational schemas, explain how you transformed it to a concrete relational schema representation

Answer：

A student can enroll in a section of a course only if the student has not taken that course before. But we could not track this situation. Thus, I decide to use Id for course (CId) and Id for student (SId) to solve this problem. CId will be used in class Section and SId will be used in class Student. It will avoid the conflict on the Grade, Exam and etc.