CS5200 A1 Data modeling
Jian Cui

Nouns:
User
Student
Faculty
Course
Module
Lesson
Widget
Evaluation Widget
Exam
User Name
Password
First Name
Last name
Emails
Phone
Address
Financial aid info
Work-study
Scholarship
Benefits
Tenure status
Bank account info

Verb - relations:
Student and Faculty (register office) is a kind of user
Course has different section in a semester
Section(course) got modules
Module broke into lessons: module has lessons
Lesson has widgets
Evaluation widget is a kind of widgets
Exam is a kind of Evaluation widgets
Assignment is a kind of Evaluation widgets
Exam and Assignment produce grade -> one evaluation widget produce one grade
Many evaluation grades can produce a finial grade
Each has many finial grades
A student can view many final grades
A student can enroll many sections of course
A faculty can teach many sections

Inheritance:
Student and Faculty (register office) is a kind of user
Evaluation widget is a kind of widgets
Exam is a kind of Evaluation widgets
Assignment is a kind of Evaluation widgets

Associations: In the graph

Classes: because those nouns got several information
User
Student
Faculty
Course
Module
Lesson
Widget
Evaluation Widget
Exam

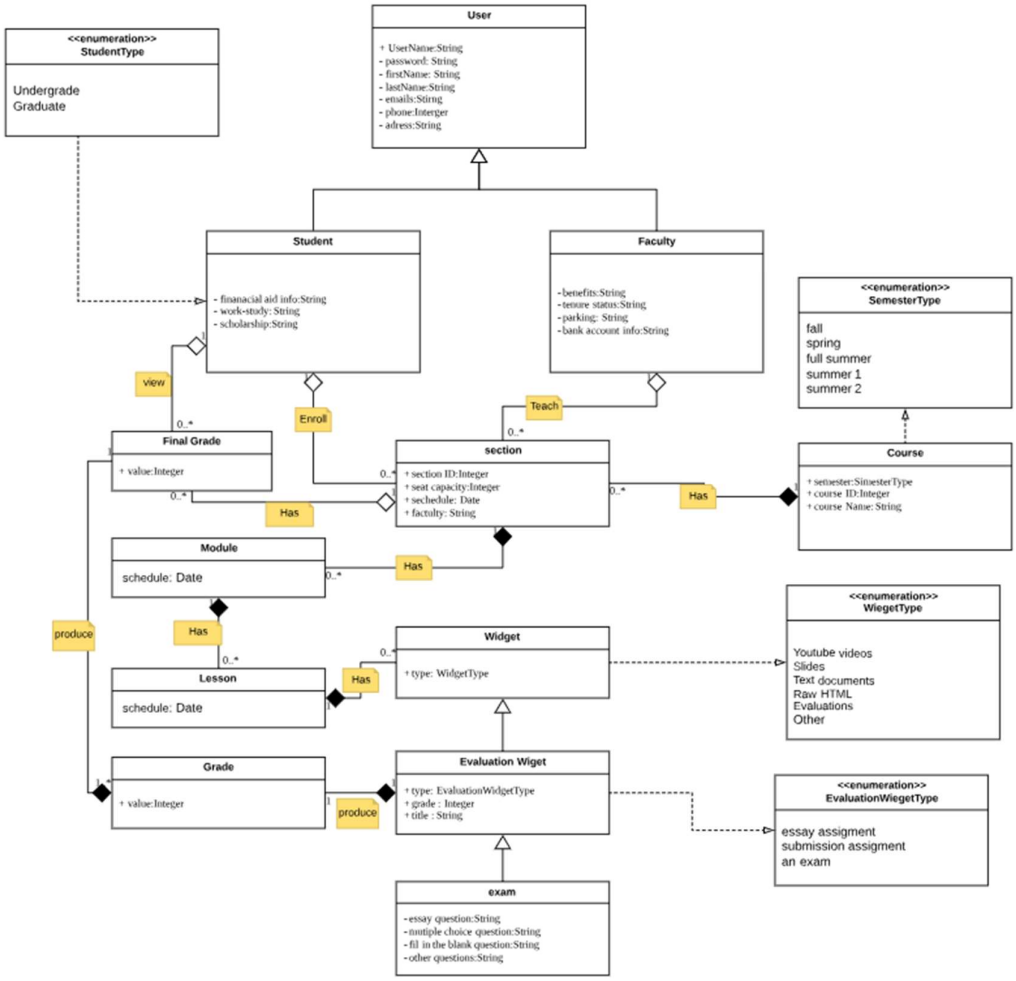
Attributes: because those nouns got single information and they are sub-info of above nouns
User Name
Password
First Name
Last name
Emails
Phone
Address
Financial aid info
Work-study
Scholarship
Benefits
Tenure status
Bank account info

Data type: in the Graph
Enumeration includes:
Student type, I think graduate and under graduate just different in the course number they can choose, but the UML won't consider about that, so just a special type.
Semester type: in article

Widget Type: in article, but evaluation widget is a special widget, it got more information. So I make it become a subclass of widget.
Evaluation widget type: in article, for assignments the title information in the Evaluation widget is enough to describe them. But the exam got more information, So I make it as sub class.

Remove any inadequate or redundant relationships:

I think the register office is not necessary in the graph. The relationship of it really doesn't matter (it can be shown by other class)
In fact, different sections may have different module, so I put module under the sections not course.
The grade is produced by exam and assignments, which is evaluation widget. So, I combine them together.
To avoid many to many relationships. I add a final grade class. One section can have many final grades and one student can have many final grades. Many sub-grades become one final grades.


<<enumeration>>
StudentType
Undergrade
Graduate
User
+ UserName:String
- password: String
- firstName: String
- lastName:String
- emails:Stirng
- phone:Interger
- adress:String
Student
- finanacial aid info:String
- work-study: String
- scholarship:String
Faculty
- benefits:String
- tenure status:String
- parking: String
- bank account info:String
<<enumeration>>
SemesterType
fall
spring
full summer
summer 1
summer 2
view
Enroll
Teach
Final Grade
+ value:Integer
section
+ section ID:Integer
+ seat capacity:Integer
+ sechedule: Date
+ faculty: String
Has
Course
+ semester:SimesterType
+ course ID:Integer
+ course Name: String
Module
schedule: Date
Has
produce
Has
Widget
+ type: WidgetType
<<enumeration>>
WiegetType
Youtube videos
Slides
Text documents
Raw HTML
Evaluations
Other
Lesson
schedule: Date
Has
Grade
+ value:Integer
produce
Evaluation Wiget
+ type: EvaluationWidgetType
+ grade : Integer
+ title : String
<<enumeration>>
EvaluationWiegetType
essay assigment
submission assigment
an exam
exam
- essay question:String
- mutiple choice question:String
- fil in the blank question:String
- other questions:String