## Step 1: Listing the Nouns :

Consider an application such as Blackboard, an online learning management system (LMS) that provides tools for faculty and students to create and share a learning experience. Faculty can author courses that contain learning modules broken up into lessons. Modules and lessons can be rearranged into a different order based on the calendar schedule. An LMS should provide a set of rich content widgets to build each of the various topics in a particular lesson. Widgets come in variety of types: youtube videos, slides, text documents, raw HTML, evaluations, and many more. Evaluation widgets can be a simple essay assignment, a submission assignment, or an exam. Exams are used to evaluate the student's progress as they answer the various types of questions in an exam, such as essay questions, multiple choice questions, fill in the blank questions, and many more types of questions. Based on the popularity of courses, the registrar's office creates several sections for a course for a given semester. There are 5 types of semesters: fall, spring, full summer, summer 1 and summer 2. Some of the less popular courses are only taught in particular semester in a given academic year. Students enroll in different sections for a course. When registering for a course, students can see a section's seat capacity, and the faculty assigned to teach the section for a particular course. Undergrad students tend to enroll for many more courses than graduate students. The registrar's office keeps track of student progress in the enrollment, such as the final grade, letter grade, and student feedback. From time to time, the University asks everyone to verify their personal profile information such as username, password, first name, last name, emails, phones, and addresses. Users can provide multiple emails, addresses and phones. Faculty additionally need to update their benefits, tenure status, parking, and bank account inn Students have to verify their financial aid info, work-study, and scholarship. Students with scholarships are always keeping an eye on their gpa that it does not drop below a certain threshold. Students can see their final grades for a particular course they were enrolled in. Grades are neatly broken by the various assessments such as assignments and exams. Even down to the points they lost on a particular question based on a rubric that keeps track on how much each question was worth on an exam or how much a particular part of an assignment was worth. Students often go to office hours to review an evaluation with their instructor or teaching assistant. They go question by question reviewing their answers and where they might have gone wrong.

List of nouns that are candidate classes or attributes:

Blackboard, faculty, students, courses, learning modules, lessons, calendar schedule, rich content widgets, topics, youtube videos, slides, text documents, raw HTML, evaluations, simple essay assignment, a submission assignment, or an exam, answer, questions, essay questions, multiple choice questions, fill in the blank questions, the registrar's office, sections, semester, fall, spring, full summer, summer 1 and summer 2, seat capacity, Undergrad students, graduate students, student progress , final grade, letter grade, and student feedback, personal profile information, username, password, first name, last name, emails, phones, and addresses. benefits, tenure status, parking, and bank account info. financial aid info, work-study, and scholarship.gpa, assessments, assignments, exams, rubric instructor, teaching assistant

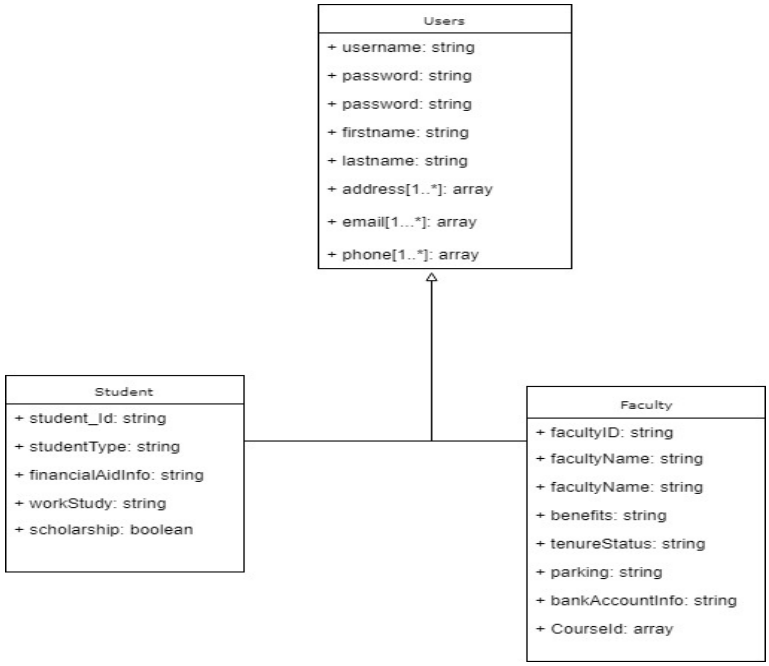
## Step2: List of verbs as candidate relations between classes

1. Faculty **author** courses
2. courses **contain** learning modules
3. learning modules **broken up into** lessons
4. Modules **rearranged** based on calendar schedule
5. lessons **rearranged** based on calendar schedule
6. rich content widgets **build** topics
7. topics **are part of** lesson
8. youtube videos, slides, text documents, raw HTML, evaluations **are types of** widgets
9. Evaluation widgets **represent** simple essay assignment, a submission assignment, or an exam
10. Exams **evaluate** the student's progress
11. essay questions, multiple choice questions, fill in the blank questions **are types of** questions
12. the registrar's office **creates** sections
13. course **has** sections
14. semester **has** courses
15. fall, spring, full summer, summer 1 and summer 2 **are types of** semesters
16. courses **are taught in** semester
17. Students **enroll in** different sections
18. Students **register** for course
19. Students **see** seat capacity of course
20. Faculty **is assigned for** course
21. registrar's office **keeps track of** student progress
22. users **verify** personal profile information
23. personal profile information **can be** username, password, first name, last name, emails, phones, and addresses.
24. Faculty **update** benefits, tenure status, parking, and bank account inn.
25. Students **verify** financial aid info, work-study, and scholarship.
26. Students with scholarships **keep an eye** on gpa
27. Students **see** final grades for courses
28. Grades **are based on** assessments
29. Assessments **can be** exam or assignment
30. rubric **keeps track of** mark for each question or assignment
31. Students **review** evaluation with their instructor or teaching assistant.

## Step 3: Generalization/specialization (inheritance, if applicable, explain) - show parts of your diagram that specifically illustrates the use of inheritance

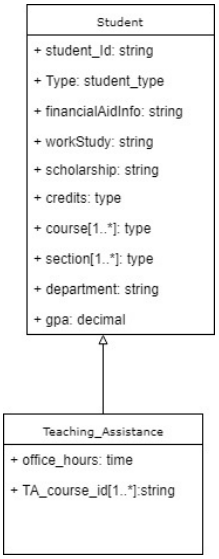
1. Student and Faculty generalized to Users

Since the main users of black board are students and faculty and all the users have some common attributes like username, password, first name, last name, emails, phones, and addresses, we can generalize student and faculty to Users class.

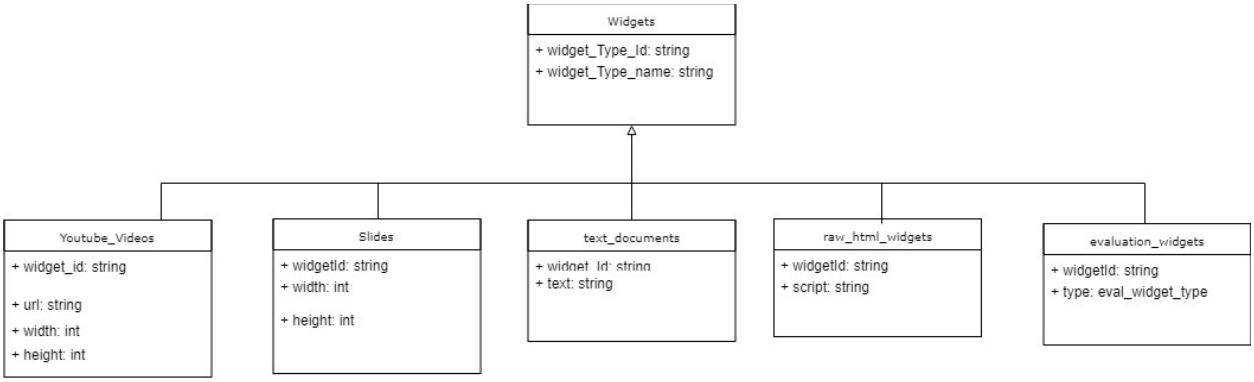



2. Teaching_assistant class inherits/is a specialization of Student class

Since teaching assistant is a student who has some additional attributes like office hours.



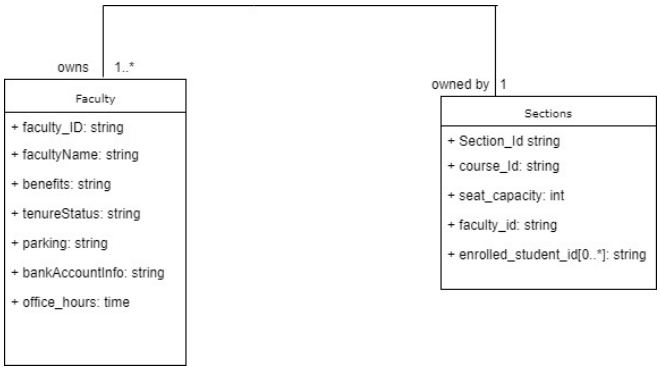

3. YouTube videos, slides, text documents, raw HTML, evaluations inherit from widget class.



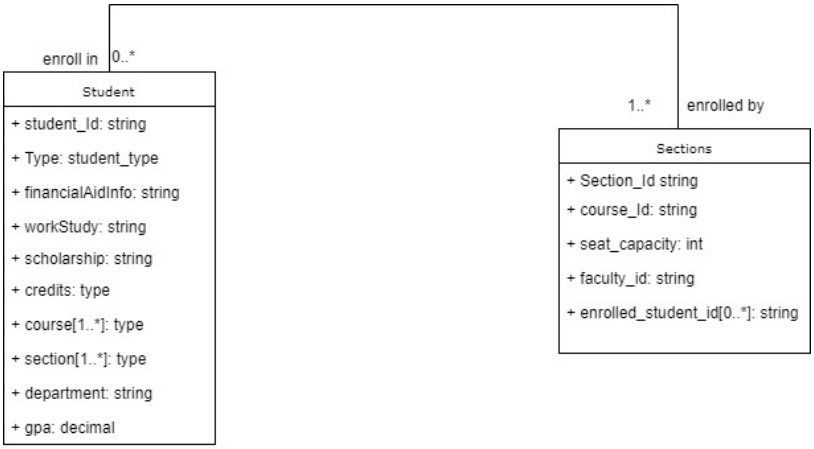
# Step 4: Associations, aggregation and/or composition

## Associations:

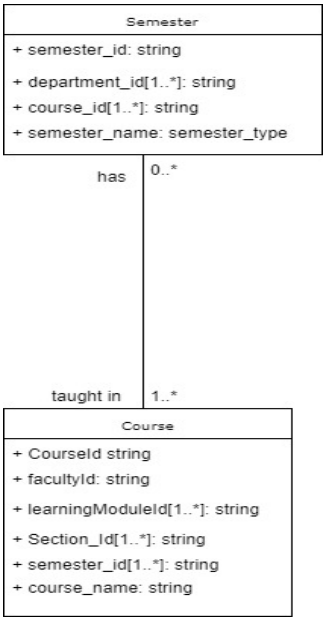
1. Faculty owns section / Section is owned by a faculty. One faculty can own one or more sections[1..*] but one section can be owned by one faculty [1].



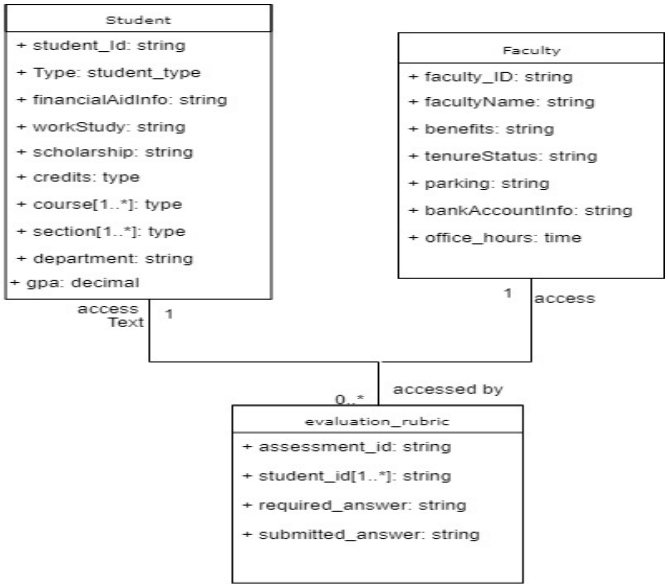

3. Students enroll in sections. Sections are enrolled by students. One student can enroll in one or more sections from different courses [1..*]and one section can have zero or more students[0..*].



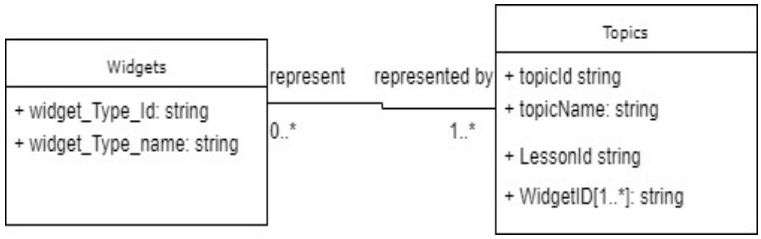

4. Course taught in Semester/Semester has courses. One course can be taught in zero or more semesters [0..*] and one semester can have multiple courses[1..*].



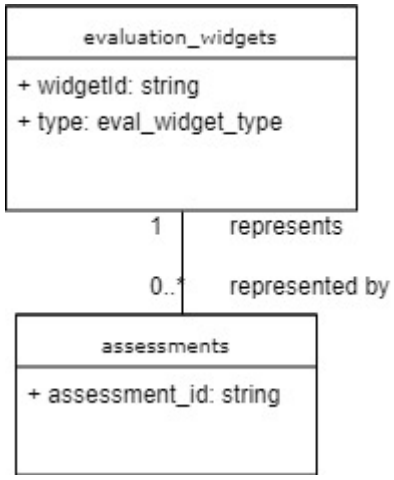

5. Student/Faculty access evaluation_rubric / evaluation_rubric is accessed by Student/Faculty. Zero or more students[0..*] can access the evaluation rubric[1].




6. Widget represent the topics or topics are represented by widgets. One widget might represent multiple topics and zero or more topics can be represented by one widget.

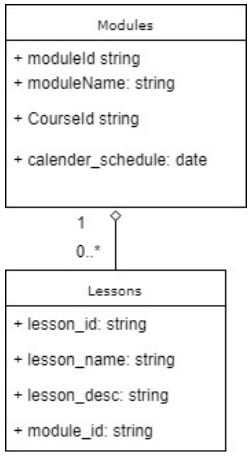


7. Evaluation widgets represent assessments or assesmnets are reprsesnted by evaluation widgets. One assessment can be reprsesnted by one more evakuation widgets and evaluation widgets can reprsesnt zero or more assessments.

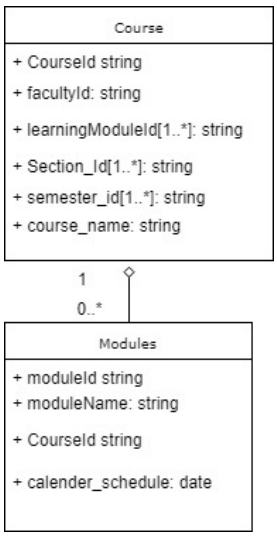



## Aggregation:

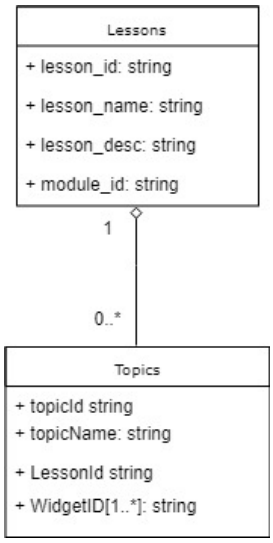
1. Modules and lessons : Modules consists of Lessons. This is an aggregation relation since existence of lessons will not be affected by the non-existence or deletion of the modules.



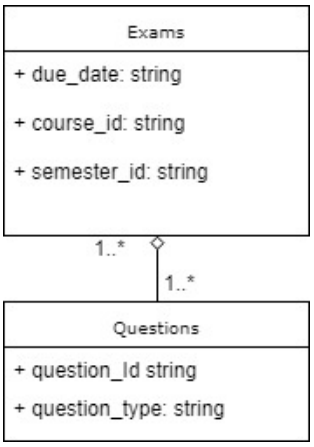

2. Course and modules: Each Course consists of modules. But if courses are deleted, modules still persist. Hence its an aggregation.




3. Lessons and topics: Each lesson has topics. But if lessons are deleted, topics still persist. Hence its an aggregation.

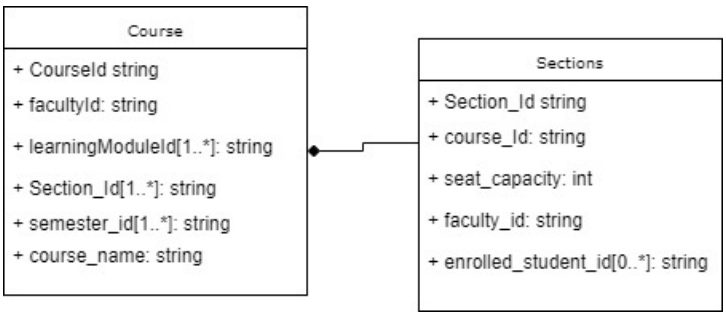



4. Exams and Questions: Exams consist of different questions. But if exams are deleted, questions still persist. Hence its an aggregation.

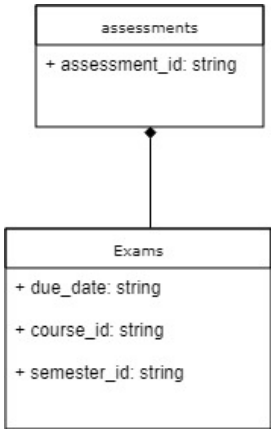



**Composition:**

1. Course is composed of sections. If course is removed, it doesn't make sense if sections class remains and cascade deletion happens. Hence it's a composition.



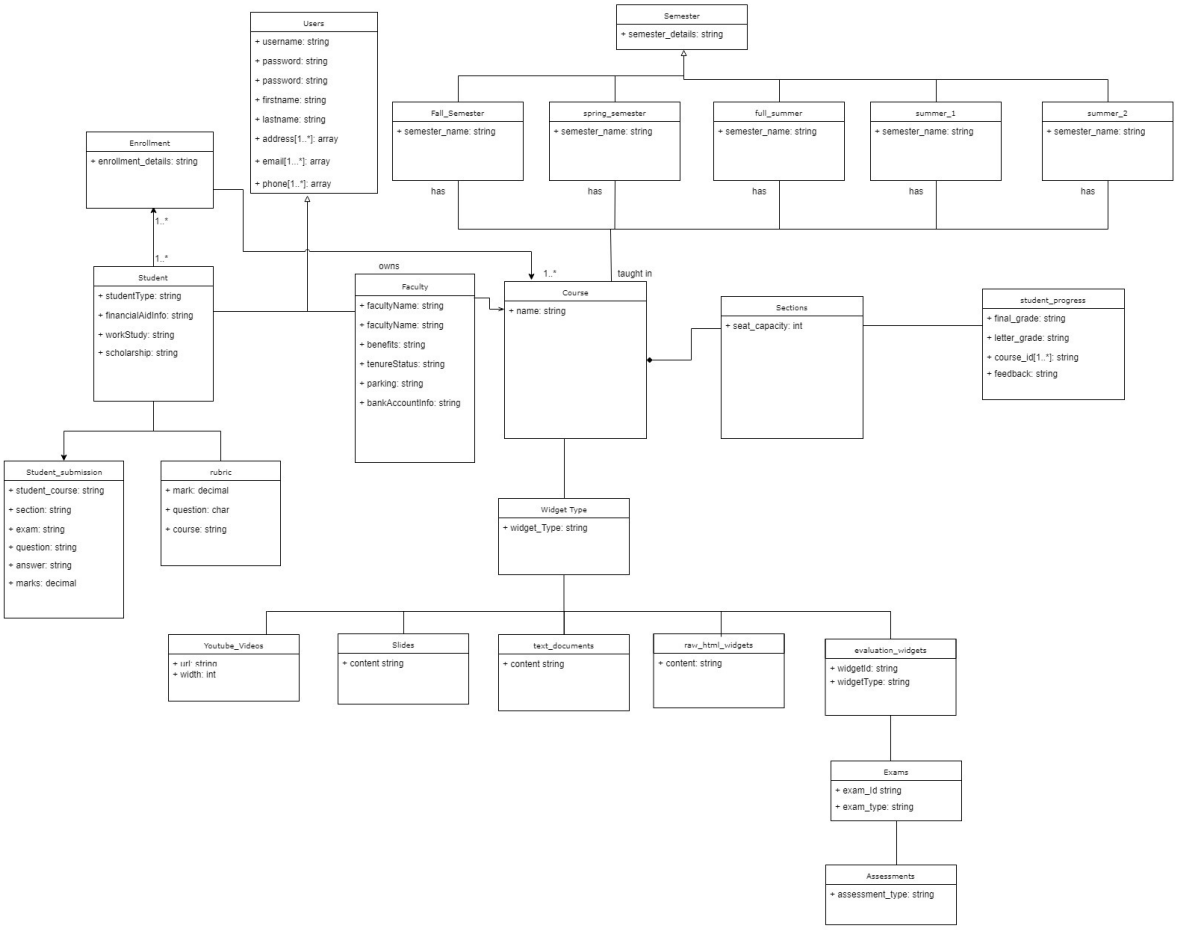

2. Exams are a part of Assessments. If assessments class is removed, it doesn't make sense if exams class remains and cascade deletion happens. Hence it's a composition.




## Step 5: Classes vs. Attributes Analysis

**Naïve Class Diagram:**

(Attached separate jpg file)




The following changes were made during the transition of the class diagram from naïve to final.

- Fall,Spring,Full_Summer,Summer1,Summer2: These are types of the Semester and has same attributes with difference in their values. Hence these can be considered as an enum datatype-semester_type. Further Semester class can have attributes semester_id,department_id,course_id and semester_name as attributes where semester_name has snum data type ie semester_type.
- Undergrad and grad student types can be implemented as another enum. Adding an attribute ‘type’ into the Student class with enum datatype student_type.
- Some of the students can be teaching assistants. This can be made a separate class which inherits from the Student class with an additional attribute office_hours.
- Assessments class can be represented as a generalisation of the exams or assignments which inturn is represented using evaluation_widgets in blackboard. Hence this relationship can be denoted with an association.

- Questions class is added and it has an enum attribute ‘type’ which has the values- fill_in_the_blanks, multiple_choice and essay_questions.
- Evaluation_rubric class was modified adding attributes- assessment_id, student_id,required_answer,submitted_answer and marks_allotted. Student_progress class has a dependancy with this class.
- Student_Progress class is inturn dependant on Course class and Student class for the course details and the student details.
- Student_Submission was a separate class in the naïve diagram which had the attributes related to the students submission of assignments or answers to question. This has been merged with the Evaluation_Rubric class which has attributes for the answer_submitted.

## Step 6: Redundancy Check

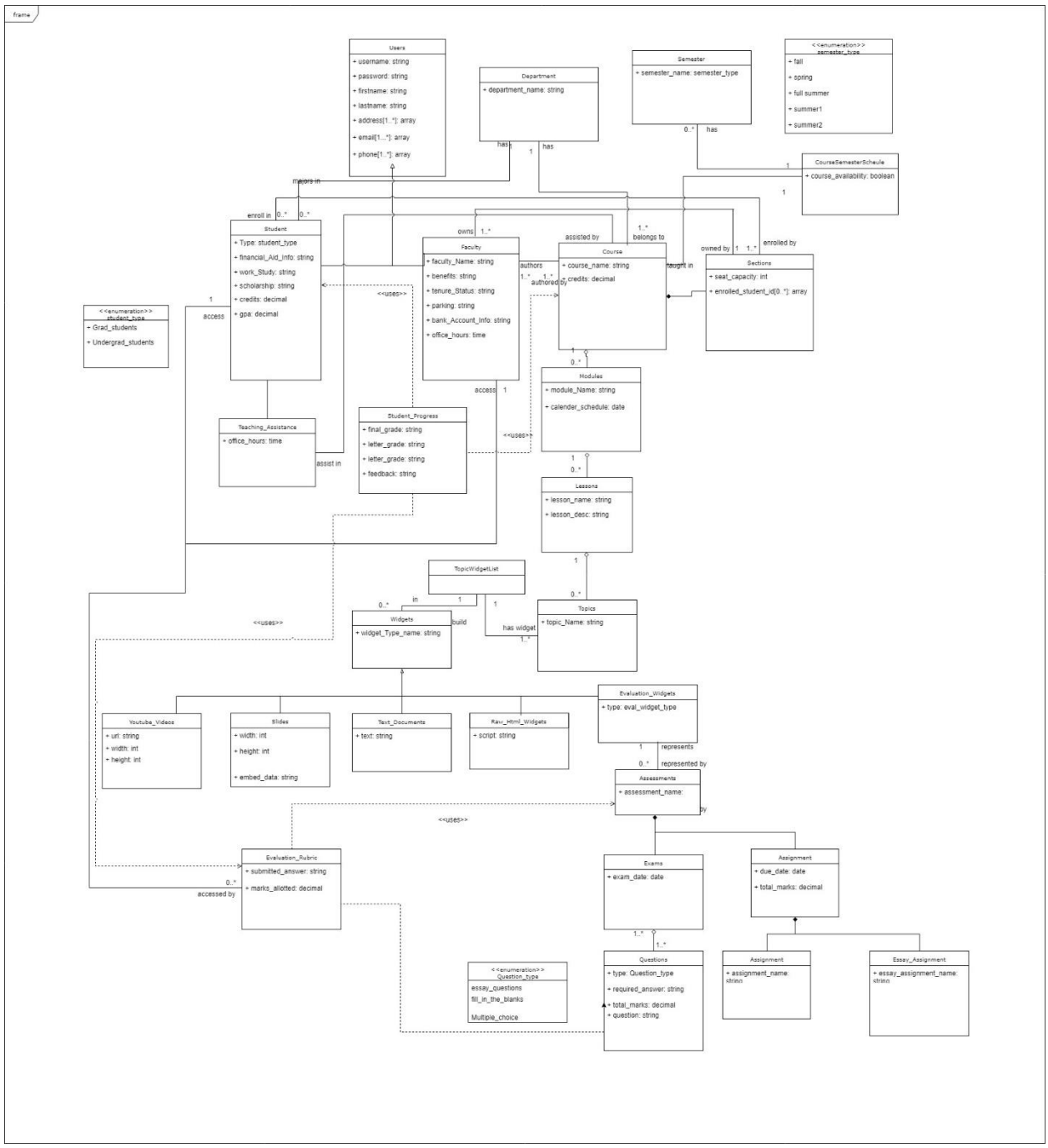
- In the naïve diagram there were two classes, Student_Submission and Rubric, both of which had students submission and grading data. Student_Submission was removed and attributes assessment_id, student_id, required_answer, submitted_answer and marks_allotted are added to the class diagram.
- Each of the types of Semester like Fall, Spring etc. were considered to be separate classes in the naïve diagram and Semester being another class. This led to redundant data in classes with attributes like course_id. Hence semester_type enum data type is declared and there is just one class Semester containing all details of the respective semesters on the basis of the semester_type attribute
- Each Widget type had different attributes and a few common attributes. So this couldn’t be converted to enum. But the common attributes were repeated. So, generalization/inheritance is introduced where the common attributes from Widget class is inherited by the different Widget type classes like text.
- User profile information is common for Student and faculty. Naïve diagram, if implemented could have led to redundant data. Hence inheritance or specialization is enforced were student and faculty extends the User class.
- Instructor and Faculty refer to the same entity. So separate classes are not required for the same.
- Teaching Assistant noun can be interpreted as a special case of Student Class. Hence this can be implemented as a class which inherits from the Student class with an extra attribute – office_hours.
- Graduate and Undergraduate students need not be separate classes since this leads to redundancy as all the attributes required are same for both. Instead a new attribute is added to the student class that has Student_type enum to understand whether the student is grad or undergrad.
- Scholarship need not be implemented as a separate class since we donot have any other data about this. Hence this is implemented as an attribute in the Student class with True or False value(Boolean datatype).
- Questions and type of questions like Fill in the Blanks, Essay Questions and Multiple Choice Questions were separate classes but had redundant attributes. This is represented as enum data

type in the final class diagram adding an attribute with Question_type enum data type in the Questions Class.

## Step 7: Reification

- Enrollment of a student to a course was a multi relation between several classes. Instead enrollment table was removed and department table was added, the relationship is depicted without many-many relation.
- Faculty can author multiple courses and one course can be authored by multiple faculties, hence we use sections class to depict this relation which removes the many-many relation.
- Multiple courses can be taught in a semester and one course can be taught in multiple semesters. This was a many many relation. Hence class CourseSemesterSchedule was added in between that contains the course availability details for each sem. Thereby removing the many many relation.
- One widget might represent multiple topics and zero or more topics can be represented by one widget. Hence class TopicWidgetList is added which ahs the widgets under a topic and hence converting to many -many relation to one to many and many to one.

## Final Class Diagram


frame
Users
+ username: string
+ password: string
+ firstname: string
+ lastname: string
+ address[1..*]: array
+ email[1...*]: array
+ phone[1..*]: array
Department
+ department_name: string
Semester
+ semester_name: semester_type
<<enumeration>>
semester_type
+ fall
+ spring
+ full summer
+ summer1
+ summer2
CourseSemesterSchedule
+ course_availability: boolean
has
has
has
majors in
enroll in
Student
+ Type: student_type
+ financial_Aid_Info: string
+ work_Study: string
+ scholarship: string
+ credits: decimal
+ gpa: decimal
owns
Faculty
+ faculty_Name: string
+ benefits: string
+ tenure_Status: string
+ parking: string
+ bank_Account_Info: string
+ office_hours: time
assisted by
belongs to
Course
+ course_name: string
+ credits: decimal
owned by
enrolled by
Sections
+ seat_capacity: int
+ enrolled_student_id[0..*]: array
taught in
authors
authored by
<<enumeration>>
student_type
+ Grad_students
+ Undergrad_students
access
<<uses>>
Modules
+ module_Name: string
+ calender_schedule: date
Teaching_Assistance
+ office_hours: time
assist in
Student_Progress
+ final_grade: string
+ letter_grade: string
+ letter_grade: string
+ feedback: string
Lessons
+ lesson_name: string
+ lesson_desc: string
TopicWidgetList
Topics
+ topic_Name: string
Widgets
+ widget_Type_name: string
in
build
has widget
Youtube_Videos
+ url: string
+ width: int
+ height: int
Slides
+ width: int
+ height: int
+ embed_data: string
Text_Documents
+ text: string
Kavu_Html_Widgets
+ script: string
Evaluation_Widgets
+ type: eval_widget_type
represents
represented by
Assessments
+ assessment_name:
Evaluation_Rubric
+ submitted_answer: string
+ marks_allotted: decimal
accessed by
Exams
+ exam_date: date
Assignment
+ due_date: date
+ total_marks: decimal
Questions
+ type: Question_type
+ required_answer: string
+ total_marks: decimal
+ question: string
<<enumeration>>
Question_type
essay_questions
fill_in_the_blanks
Multiple_choice
Assignment
+ assignment_name: strinn
Essay_Assignment
+ essay_assignment_name: strinn