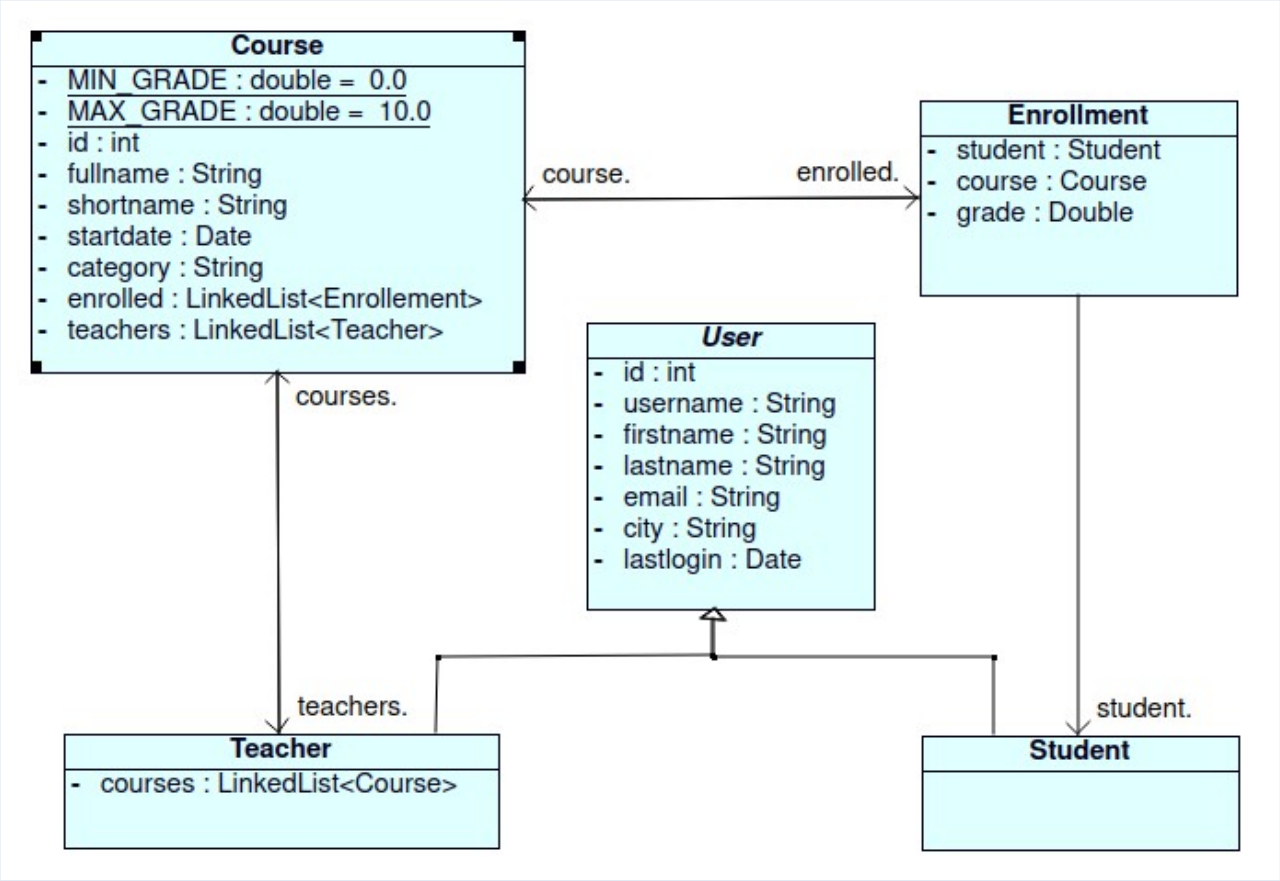
DAM2/DAW2. UF6 Pe1

L'aplicació «**AppMoodle**» permet gestionar un entorn d'aprenentatge organitzat en cursos, als que s'hi poden afegir professors alhora que inscriure alumnes i avaluar-los. A continuació es mostra el Diagrama de classes UML.



Aquestes classes s'inclouen dins el projecte adjunt a l'enunciat amb tots els seus mètodes implementats.

Al mateix projecte també s'inclou una **base de dades ObjectDB** (moodle.odb), les **classes de l'esquema** i la classe «**AppMoodle**» amb el mètode «main» i la resta de mètodes que cal implementar buits.

Exercici 1. (3 punts) Afegir les anotacions a les classes que permetin:

- Fer persistents les instàncies de les classes del diagrama.
- Establir les relacions entre les classes en cascada per totes les operacions.
- Definir la comprovació de camps nuls, els índexs i consultes amb nom «**Named Query**» de la taula següent:

(Si no saps què és una consulta «Named Query» no cal ni que ho preguntis!!)

Classe	**Atributs no opcionals**	**Índexs**	**Únic?**	***Named Query***	**Descripció consulta**
«Course»	fullname, shortname, startdate, category	id	Si	*Course.findAll*	Consulta tots els cursos ordenats per nom complet «fullname»
				Course.findById	Consulta del curs amb «id» indicat com a paràmetre
«User»	username, firstname, lastname, email	id	Si	--	--
		username	Si		
		email	Si		
«Teacher»	--	--	--	*Teacher.findAll*	Consulta tots els professors ordenats per nom d'usuari «username»
				Teacher.findById	Consulta del professor amb «id» indicat com a paràmetre
«Student»	--	--	--	*Student.findAll*	Consulta tots els estudiants ordenats per nom d'usuari «username»
				Student.findById	Consulta de l'estudiant amb «id» indicat com a paràmetre
«Enrollment»	student, course	--	--	*Enrollment.findBy Student*	Consulta totes les inscripcions d'un estudiant ordenades per nom complet del curs «fullname»

Exercici 2. (3 punts) Cal implementar els mètodes corresponents de la classe «**AppMoodle**» amb les operacions indicades i en el mateix ordre.

A dins de cada mètode totes les actualitzacions han d'anar dins una única transacció. Per fer persistents tots els canvis que s'hi produeixen **de la manera més simple possible**.

En cas d'error cal:

1. Capturar l'excepció,
2. mostrar el missatge d'error de l'excepció i
3. desfer els canvis a la base de dades des de l'inici de la transacció.

Mètode estàtic «**gestionarDades1**»:

- Afegeix un nou estudiant amb les següents dades:

  id: 2201, username: nul, email: "student2201@gmail.com",
  firstname: "Anna", lastname: "Garcia", city: "Sant Boi", lastlogin: null

Mètode estàtic «**gestionarDades2**»:

- Consulta el curs amb «id» valor 2 fent servir la consulta amb nom «Course.findById».
- Canvia la categoria a «DAM2/DAW2».
- Afegeix un nou estudiant al curs amb les següents dades:

  id: 2202, username: "student2202", email: "student2202@gmail.com",
  firstname: "Joan", lastname: "Planas", city: "Sant Boi", lastlogin: null

- Posa una qualificació de 7.5 a l'alumne anterior.

Mètode estàtic «**gestionarDades3**»:

- Consulta les inscripcions a cursos («Enrollment») de l'estudiant amb «id» valor 63.
- Esborra l'estudiant dels cursos («Course#removeStudent») on encara no tingui cap qualificació («grade» valor nul).

Mètode estàtic «**gestionarDades4**»:

- Consulta el curs amb «id» valor 314 fent servir la consulta amb nom «Course.findById».
- Esborra aquest curs i totes les dades relacionades: professors, inscripcions i alumnes si escau.

Mètode estàtic «**gestionarDades5**»:

- Consulta el professor amb «id» valor 2447 fent servir la consulta amb nom «Teacher.findById».
- Canvia la el nom d'usuari per «qres».

Exercici 4. (4 pts) Cal implementar les següents consultes als mètodes corresponents de la classe «**AppMoodle**».

El resultat de totes les consultes ha d'anar paginat, la **mida de la pàgina és 10**, i dins el mateix mètode cal mostrar el resultat de la pàgina indicada fent servir el mètode estàtic «**mostrarResultatConsulta(titol, dades)**» que mostra el resultat de la consulta en columnes.

Per exemple el codi per mostrar el resultat d'una consulta podria ser com el següent:

```
List<Object[]> resultat = query.getResultList();

Vector<String[]> dades = new Vector<String[]>();
for (Object[] item : resultat) {
	dades.add(new String[]{ item[0]+"", item[2]+"" });
}

System.out.println(mostrarResultatConsulta("Consulta Dades X", dades));
```

Mètode «**consultaDades1**»:

- Pàgina 2 de la consulta dels estudiants amb última entrada («lastlogin») posterior a una data i nom d'usuari que comenci per un cert text. Els dos valors indicats per paràmetre.
- Cal mostrar l'identificador, el nom d'usuari, el nom i cognoms, la data de la darrera entrada i el correu.
- Ordenar per nom i cognoms ascendent.

Mètode «**consultaDades2**»:

- Pàgina 5 de la consulta dels cursos i les qualificacions dels seus estudiants, només per a cursos amb més d'un cert nombre de professors i d'una categoria determinada. Els dos valors indicats per paràmetre.
- Cal mostrar l'identificador, la categoria i el nom complet dels cursos, i per cada alumne inscrit, el seu nom, cognoms, i la qualificació.
- Ordenar les dades per nom complet del curs ascendent i qualificació dels estudiants descendent.

Mètode «**consultaDades3**»:

- Pàgina 1 de la consulta per cada curs del total de professors, el nombre total d'estudiants, la nota mínima, mitjana i màxima d'aquests.
- Només per a cursos de les categories que continguin un cert text indicat per paràmetre.
- Cal mostrar l'identificador, la categoria i el nom complet dels cursos, el total de professors, el nombre total d'estudiants i la nota mínima, mitjana i màxima d'aquests.
- Ordenar pel total d'estudiants de cada curs.