**So far will write it in English, then**

**DATABASE DISCRIPTION**

Roles: Admin, Student, Educator

Some roles are also related to separate classes:

**ENTITIES**

**User(inherits IdentityUser)**

FirstName: string

FamilyName: string

*For later*

Picture – optional – how to do it?

**Student (one to one with? LMSAppUser)**

User

Student number – int - will leave it non-unique since different faculties might have identical student numbers

Major - enum

Group (many to one)

Last seen – Datetime

List of StudentCourse (for many to many relationship)

List of StudentLecturecise (for many to many relationship)

List of StudentEvent (many to many)

List of Student Assignment (for many to many relationship)

Task completion rating? – decimal in db? As a part

Task grade rating – decimal in db? As a part

List of StudentBadge

Total grade in lectures – with get – out of db?

Total grade in exercise – with get – out of db?

*For later- Or in another library!!! To be related to this one?*

List of StudentPracticeSet (for many to many relationship)

Practice rating? – как да се смятат finish rating, grade rating

**Group:** Made for simpler browsing – otherwise everything is accessible through other entities

List of students

List of Events

**Educator (one to one with? LMSAppUser)**

User

Additional info visible to students: string optional

Link to personal page: url, optional

List of EducatorCourses (for many to many relationship)

List of EducatorLecturecise (for many to many relationship)

List of EducatorEvents (for many to many)

List of Assignment

List of Materials – да избира дали са свободни за всички или само за група?

**Course:**

Name - string

Semester - enum

UniYear: string, To be selected automatically in the form with an option to edit – or to verify it with a regex

List of Majors

List of Lecturecise

List of StudentCourses(many to many)

List of Materials

**StudentCourses:**

Student

Course

**EducatorCourses**

Educator

Course

TeachingRole

**Lecturecise:**

LectureciseType

List of WeekTimes

List of EducatorLecturecise (many to many)

List of StudentLecturecise (many to many)– should have functionality to be added by group

List of Events – should be automatically populated for the semester?

List of Student Assignment

**StudentLecturecise:**

Student

Lecturecise

**EducatorLecturecise:**

Educator

Lecturecise

**WeekTime:**

Day of the week - DayOfWeek

StartHour – int

EndHours - int

**Event:**

StartTime - DateTime

Lecturecise – optinal, could be additional event

List of StudentEvents(many to many)

List of EducatorEvents(many to many) – option to have events only for educators

List of Materials

**StudentEvents:**

Student

Event

**EducatorEvent:**

Educator

Event

EventRole

**Assignment:**

Name: string

Lecturecise:

AssignmentType:

Content: string, optional

Date created: DateTime

Educator giving:

Max grade: decimal (optional)

List of Materials

List of StudentTask

List of Groups

**StudentAssignment:**

Student

Assignment

Lecturecise:

Date due: DateTime?

AssignmentStatus:

Expired: bool

Grader:

Grade: decimal, optional

Grade Comment: string, optional

**Badge:**

Name: string

Descripton: string, optional

Assignment: (optional)

List of StudentBadge

*For later*

Badge Image

**StudentBadge:**

Student

Badge

**Material:**

Name – string

Content – string, optional

Link to repo: url - ??? is this enough?

Educator:

Course: optional

Assignment: optional

Event: optional

List of Students: optional

StudentMaterial:

Student

Material

**ENUMS:**

**Major:** Molecular Biology, Biotechnology

**LectureciseType:** Lecture, Exercise

**Semester:** Winter, Summer

**TeachingRole:** Lecturer, Assistant, Other

**EventRole:** Instructor, Examiner, Observer, DecisionMaker, PartyAnimal, Other

AssignmentStatus**:** Unfinished, UnderRevision, ToRemake, Finished, Other

AssignmentType**:** Current, Final, Other

**OOP Principles and Security**

Make all lists ReadOnly or some? Or just make input models for everything and be careful with get and post

**FUNCTIONALITY DISCRIPTION**

**GUESTS:**

**-** Sees some homepage

-Registers – includes to mark the role, student number field is required if student role is marked, same with major and group number

- Login – by email/or other username and password,

**ADMIN:**

Adds Users:

* Manually
* From an Excell file

Deletes Users – delete is real and not only by Flag

Assigns Roles to users

Approves Roles to users

**EDUCATOR:**

To Copy and Edit Assignments from previous years

**APP:**

Sending emails

Automatic email confim

Emails admin for hanging role requests – once per 24 hours