Last modified 6 months ago

This page is intended to list all entities that we work with. The descriptions should contain the purpose of the object, to which component it belongs to and how the entity is related to others.

NOTE: is this distinction between Persons,Other Entities necessary and useful?

Example

Entityname

here goes the verbal description of what this entity is and what it is doing, for what is it good and so on.

variables === variables this entity may actually have or should have

belongs to === to which component this entity belongs to, e.g. scheduler, login, ... multiple are possible

relations with === the entity has relations to the other entities listed here. describe how the entities are related

---

---

persons

A natural person is someone who uses MyCourses in order to interact with the system.

Variables

- Name, including surname
- ID, which he is identified by
- E-mail address, used as login
- Password, used as login

Belongs to

--

Relations with

---

lecturer

A lecturer is a person that can be assigned for a course instance. That implies making him/her the main lecturer of this course instance. He/She can get involved in other course instances, how is not completly clear (The main lecturer of another course instance may add him to the teaching stuff or similar).

The main lecturer defines the details about the course instance he is assigned for: Adds or edits a description for the course instance, which are other people from the teaching staff involved, which are the course elements (lectures, tutorials, labs, projects...), the way of possible course execution (a number of lectures and other elements per week, preferred days and times (the scheduler determines at what time they actually take place), expected number of students, and similar). Hey may wish to (smart) copy all data from a previous instance of the course. He/She can kick, ban or add students from his/her course instance.

Comment$_{Konrad}$: We might want to call this entity Teacher.

variables

- a name, including surname
- ID, which he is identified by

belongs to

relations with

- he is a person
- he is teaching Courses
- he is assigned (the main lecturer) for a course instance
- he chooses the course elements
- he can kick, ban or add students from course instances if he is assigned for it

---

student

A student is a person that might attend courses. The student uses MyCourses to view the individual course schedule, after the student has defined the courses he attends.

variables

- a name, including surname
- ID, which he is identified by

belongs to

relations with

- is a person
- attends courses

---

program administrator

A program administrator who is responsible for management of the programs at the university defines programs. She identifies the program, its running period (starting and ending year), and a

program manager who will have the overall responsibility for the program. Typically when defining a new program, the same program from the previous year would be copied, with some data changed afterwards.

variables

- a name, including surname
- ID, which she is identified by

belongs to

relations with

- defines programs
- identifies a program manager

---

program manager

A program manager, when enabled, can enter all details about the program: Which courses it includes, which of these are obligatory and which optional, etc. The courses may already exist, and in that case she creates a new course instance and enters the details. If the course is new, then the program manager creates it first, and then creates a new instance of it.

variables

- a name, including surname
- ID, which he is identified by

belongs to

relations with

- is a person
- enters details about the program
- may create courses
- creates new instances

---

role

A role is a bundle of permissions granted to aperson. For instance there is a permission "login" which is bundled with the role "user". As a matter of fact every user of the MyCourses system would have this role.

belongs to

relations with

- persons have/get roles

Other Entities

---

program

A program consists of all course instances of one specific department of an university.

variables

- starting and ending year (date)

belongs to

relations with

- defined by a program administrator
- has an program manager

---

course element

A course element is a part of a course. There are different kinds of course elements, for instance lectures, tutorials, labs or projects. The course elements are determined by a program manager which is a role.

belongs to

relations with

---

permission

A permission is part of the user right management. A permission enables a person to access a functionality of the system.

belongs to

relations with

---

course

A course is teached by a lecturer. students might attend the course.

variables

- name
- ID which identifies the course
- duration

belongs to

relations with

---

courseinstance

A courseinstance is an instance of a course with a period of execution, a main lecturer and some general information about the course, all this entered by a program administrator.

variables

- period of execution
- main lecturer
- general information

belongs to

relations with

- instance of course
- has an attended main lecturer
- created by a program administrator

---

room

A room is a place where course instances can be hold. A room may have different features with a certain degree in quality (amount of). For example a room may have the features "capacity" (having an integral quality), "has-overhead-projector" (having a boolean quality) or "suitable-for-exams" (having no quality at all, i.e. void). course instances may be defined as dependent on such features, the most prominent one being "capacity". A common use-case may also be "workstations" of which a certain amount may be required. Some course instances will be scattered across many rooms until a lower bound of capacity is reached (consider a programming course taking place in multiple pool rooms).

variables

- name
- ID which identifies the room
- features

belongs to === sheduler

relations with === Feature

---

feature

Every feature describes a special property a room can have. The properties have different types, like booleans for "suitable-for-exams" or integral types like "number-of-seats" or "workstations".

belongs to

relations with

- feature of a room

---

studentsgroup

A student group bundles students with the same interest for instance distinguished by their major they have chosen. Student groups are identified by an ID. The amount of total students participating in this group is also stored. Furthermore they attend course elements.

variables

- ID which identifies the group
- size of group

belongs to

relations with

---

locality

A locality may or may not be a bundle of rooms located near each other

variables

- name

belongs to

relations with

- may bundle rooms