

# Entities in myCourses

Conventions in this document: Entities are capitalized always and underlined. The first occurrence of a referenced Entity in the definition of another is **emphasized**.

## Room

A Room is a physical location where **CourseElementInstances** take place. A Room has an *identifier* which uniquely identifies it across the whole university. A Room is located inside a **Building**. Every Room has a *number* which is unique in the Building. A Room has **Features** and it may have a certain quantity of any Feature (e.g. 300 seats, a beamer, 20 workstations). Some Rooms do have a *special name* (“Audimax”) which is unique in the building. A Room may be of a certain *type* (*auditorium*, *lab*, ...). Not all rooms are always available.

Note: If features should have boolean and/or integer quantities: implementation detail. This is only an informal description for clarification of the entities.

## Building

A Building has a unique *street address* and belongs to a Department. Some Buildings have a special name (“Henry-Ford-Building”) which is unique across the whole university.

Note: Sometimes building and room may be the same (some universities do have an auditorium which is the only room in the building). In that case a building needs to be created which consists of one single room. In a possible implementation that special case could be handled differently by the application, but I think it should not be modeled in a special way since it seriously complicates the whole thing.

## Feature

Rooms do have certain Features which may have a certain *quantity*. A Feature is identified by a *name* (“seats”, “beamers”, ...). For example any Room which is suitable to hold a lecture or a seminar in it will have a certain amount of seats, a lab has a certain amount of workplaces. While an auditorium may have 300 seats suitable for doing a lecture it might have only 75 seats suitable for doing an examination.

# Person

A Person is a natural person who is able to login to myCourses. Therefore she is having *login credentials* (e.g. username and password or mail address and password). Every Person has a *first name* and a *last name*. There are many different roles at a university and there are persons who act in more than one role (e.g. there are students doing lectures, while not being professors). Every person needs to have a *unique identifier* regardless of the roles they belong to for administration reasons. Persons may belong to several Groups.

## Staff

Persons belonging to staff do have a *staff id*. Since people belonging to the staff are usually payed for working hours they also do have a certain number of *working hours* they should meet in a given interval (e.g. per week).

Note: The working hours property is an important one. At schools for example teachers do have a certain amount of hours they have to work per week. If our scheduler for example is to allocate course element instances to lecturers (which I think it should be capable of --julian) it should respect that property as an additional constraint.

## Student

A student has a *student id* and a *record of Courses* he already attended and successfully passed. Each Period a student enrolls into certain **CourseInstances** and books several **CourseElementInstances** according to the requirements of the **Course** the CourseInstance belongs to.

## Lecturer

A lecturer is a person that holds a **CourseElementInstance**. A CourseElementInstance belongs to a CourseInstance for which there is a main lecturer, i.e. a Person being responsible for that CourseInstance. However such a main lecturer does not necessarily hold **CourseElementInstances** too. Not all lecturers can hold all Courses, thus a lecturer does only have the ability to hold a certain *set of Courses*. Some lecturers are already tied to certain Courses, CourseInstances or CourseElementInstances. A lecturer may have preferences on certain Courses or CourseElements.

A lecturer is not necessarily available at all times and may favor certain times (e.g. a lecturer may be out of town every wednesday or favor to work in the early morning).

A lecturer is not necessarily part of the staff, since for example a guest lecturer could hold a CourseElementInstance or even a whole CourseInstance too.

## Other roles

Most universities will have many different roles and will therefore require defining their own ones.

# Group

**Persons** may be organized into different Groups, e.g. Students may belong to certain Classes. Groups may be attached to a certain Year. A Group is identified by a *name*. For some universities it might be desirable to define certain kinds of groups sharing certain special properties.

## Year

Typically Students start studying sometime and end studying sometime later and their course of studies has a regular length. Of course **Courses** a Student has to enroll in in the same year must not overlap. That is what Year is for. Courses may be assigned to a certain Year (e.g. *first year*, *second year*), but they do not necessarily have to.

## Department

A logical group of several entities. Persons, Buildings and Courses belong to a Department. Buildings may belong to several Departments.

Note: Buildings may belong to several Departments: consider the Silberlaube at Freie Universität Berlin.

## Period

All Programs happen in a Period. A Period is has a *start* and *end date*. No start date of a Program may be set before the periods start Program and no end date of Program may be set after the Periods end date. One can say: A period is the program of the university, where the university consists of all departments.

## Program

A Program is a set of **CourseInstances**. It has a running period (*start* and *end date*). A program manager is responsible for a Program. A Program belongs to a Department and to one Department only.

## Course

A Course is a well-defined set of **CourseElements**. For example there may be a Course which consists of two CourseElements of which one is a mandatory lecture and the other one is an optional seminar.

### CourseInstance

A CourseInstance is created within a **Program**. It is a Course actually taking place. There can be several CourseInstances of the same **Course** within a program. There is a main lecturer being primary responsible for any CourseInstance.

## CourseElement

A CourseElement is a type of event at a school or university, like a lecture or a seminar. A CourseElement has a *duration* and it may take place *several times in a week* (of which each occurrence has a certain duration). A CourseElement may have some *dependencies* regarding the Features the Room it is held in has. A CourseElement is part of a Course. It is of a certain *type* (e.g. lecture, seminar, discussion, meeting, ...).

### CourseElementInstance

A **CourseElement** actually taking place. Of course it imposes the same constraints like the corresponding CourseElement and Course.