Course Management subsystem

The course management subsystem is responsible for any operations that are related to the courses as well as its impact on the schedule itself. Each section inherits a course to know all its information. Each course has at least one section. Also each course may have prerequisites which have to be met in order to be able to add it to the schedule. Each section for a specific course may have a tutorial and a laboratory component. Note that each section, tutorial and laboratory all have one or more room blocks. This is necessary because each of those classes may have different time and room allocations during the week. The roomBlock class inherits the timeBlock class to know at which time and during which semester the room is going to be occupied. The timeBlock class inherits the preferences to know which times are preferred by the user. The current schedule may have different sections for which the user is already registered to.

**Schedule:** is a views class that provides data about the courses the user wants to add.

Attributes:

* N/a

Methods:

* addSection(s : section) will temporarily add the section s, which is the parameter, to the schedule.
* removeSection(s : section) will remove the section s, which is the parameter, from the schedule.
* addTutorial(Tutorial) will add the tutorial that is associated to the section (if any)
* addLaboratory(Laboratory) will add the laboratory that is associated to the section (if any)

**Section:** is a views class for a specific course. Depending on the course, it may have one or more sections.

Attributes:

* instructor: string

Methods:

* getTutorials(): Tutorials[\*] will return an array of tutorials for a specific section of a course.
* getLaboratories(): Laboratories[\*] will return an array of laboratories for a specific section of a course.

**Tutorial:** is a views class which represents the tutorial for a specific section of a course (if applicable.

Attributes:

* TA: string

Methods:

* N/A

**Laboratory:** is a views class which represents the laboratory for a specific section of a course (if applicable).

Attributes:

* Lab\_instructor: string

Methods:

* N/A

RoomBlock: is a views class which represents the room(s) in which a class is taking place.

Attributes:

* id: string
* room: string

Methods:

* N/A

TimeBlock: is a views class that represents the time for which the room is needed.

Attributes:

* start: Date
* end: Date
* semester: int

Methods:

* overlaps(TimeBlock timeBlock): bool will look at timeBlock (which is the parameter) and checks if there is any conflict in terms of time. If there is a conflict, then the method will return true, otherwise false.

Preference: is a views class which represents the user’s desired options when generating a schedule.

Attributes:

* N/A

Methods:

* N/A

Course: is a views class which represents a subject that needs to be completed in order to graduate. It has at least one section.

Attributes:

* name: string
* subject: string
* courseNumber: int
* credits: double
* courseDescription: string

Methods:

* getSections(): Section[\*] will return an array of different sections for a specific course.
* isAvailable(string semester): bool will take a string semester as parameter and check if the course is available for that semester. If it is available, the method will return true, otherwise it will return false.
* getPrerequisites(): Course[\*] will return an array of different courses that are prerequisites for a specific course.
* arePrerequisitesMet(): bool will check if the prerequisites are met for a specific course. If the prerequisites are met, then the method will return true, else it will return false.