UNIT DESCRIPTIONS

COURSES SUBSYSTEM

|  |
| --- |
| **Class Course** |
| Description: Course will provide the system the courses available for the student |
| Attributes:   * id: int * name: String * credits: double * description: String * faculty: Faculty * requisites: Course List |
| Methods:   * getname(): String   Returns a string of the name of the course   * getFaculty(): int   Returns an integer of the faculty of the course   * getCredits(): double   Returns a double of the credits of the course   * getRequisites(): Course List   Returns a Course List of the requisites of a given course |

|  |
| --- |
| Class Faculty |
| Description: Faculty provides the names of faculties in courses |
| Attributes:   * id: int * name: String |
| Methods:   * **N/A** |

|  |
| --- |
| Class ScheduledCourse |
| Description: ScheduledCourse is a general class that provides courses well scheduled in the system for students to choose from |
| Attributes:   * id: int * sessionID: int * course: Course |
| Methods:   * **N/A** |

|  |
| --- |
| **Class TimeSlot** |
| Description: A more specific class that is able to schedule courses |
| Attributes:   * id: int * scheduledCourse: ScheduledCourse * room: String * timeStart: Date * timeEnd: Date * dayOfWeekID: int * courseTypeID: int |
| Methods:   * getScheduledCouseID(): int   Returns an integer of a Schedule Course for its identification   * getCourseInfo(course: Courses): void   Returns a void in which its parameters is course, where it provides a list of information about the course   * isFull(): Boolean   Returns a Boolean to specify if the course is full or not   * addStudent(student: Student)   Adds a student in the selected course, using a parameter student   * addSection(room: String, timeStart:Date, timeEnd:Date, dayOfWeekID:int)   Adds a desired section of the course, by providing the room, the time start, the time end and the day of the week |

STUDENT SUBSYSTEM

|  |
| --- |
| Class Student |
| Description: Student class provides information about the student, including the course record, the name, the preferences and the password |
| Attributes:   * student\_id: int * first\_name: String * last\_name: String * user\_name: String * password: String * studentRecord: CourseRecord List * Preferences: Preferences |
| Methods:   * getID(): int   Returns an integer of the student’s identification   * getFullName: String   Returns a String of the full name’s student   * changePassword(oldPass: String, newPass)   Changes an old password to a new one by passing oldPass and newPass in the parameters |

|  |
| --- |
| Class CourseRecord |
| Description: CourseRecord contains all courses taken by the student and the final grade they got |
| Attributes:   * Course: Course * Grade: Grade |
| Methods:   * **N/A** |

|  |
| --- |
| Class Grade |
| Description: Grade contains the grades received by the student for the course taken by him or her |
| Attributes:   * Id: int * Value: String |
| Methods:   * **N/A** |

|  |
| --- |
| Class ScheduledCourse |
| Description: ScheduledCourse is imported from Courses Subsystem. This provides the student the ability to schedule a course without the courses already taken |
| Attributes:   * Id: int * sessionID: int * Course: Course |
| Methods:   * **N/A** |

|  |
| --- |
| Class Preferences |
| Description: Preferences provides the student the ability to change his or her preferences |
| Attributes:   * id: int * day\_of\_week\_id: int * start: time * end: time |
| Methods:   * addPreference(day: int, start: time, end: time)   Adds a preference, by providing the day, start time and end time in the parameters   * editPreference(day: int, start: time, end: time)   Edits an already available preference by proving the day, start time and end time in the parameters |