

TeacherService

+createTeacher(String name, String surname, String contact): void

+readTeacher(int idTeacher): Teacher

+searchTeacher(String searchTerm): List<Teacher>

+updateTeacher(Integer id, String name, String surname, String contact): void

+assignmentCourseToTeacher(Teacher teacher, Course course): void

+readTeacherList(): List<Teacher>

Teacher

-idTeacher: int

-login: String

-password: String

-name: String

-surname: String

-contact: String

-courseList: List<Course>

Course

-idCourse: int

-name: String

-teacher: Teacher

-studentList: List<Student>

CourseService

+createCourse(String name): void

+updateCourse(int idCourse, String courseName): void

+deleteCourse(int idCourse): void

+readCourseList(): List<Course>

+readCourse(int idCourse): Course

+calculateCourseAverage(int courseId): double

+calculateStudentAverageInCourse(int courseId, int studentId): double

+searchCourse(String searchTerm): List<Course>

Administrator

-idAdministrator: int

-login: String

-password: String

AdministratorService

+generateUniqueLoginTeacher(String firstName, String lastName): String

+generateUniqueLoginStudent(String firstName, String lastName): String

+generatePassword(): String

+isLoggedInTeacher(String login): boolean

+isLoggedInStudent(String login): boolean

(calculerMinMaxGrades)  
mettre List<Assessment>  
assessments en parametre  
retourne : Map<Integer,  
Map<String, Float>>

Assessment

-idAssessment: int

-name: String

-gradeList: List<Grade>

-average: float

-idCourse: int

AssessmentService

+createAssessment(Course course, String name): void

+createGrade(Student student, int idAssessment, float gradeValue): void

+updateAssessmentAverage(int idAssessment): void

+checkAssessmentExists(Course course, String name): boolean

+calculateMinMaxGrades()

Student

-idStudent: int

-login: String

-password: String

-name: String

-surname: String

-dateBirth: Date

-contact: String

-schoolYear: String

-courseList: List<Course>

GradeService

+readGrade(int idStudent, int idCourse): List<Grade>

+calculateAverage(int idStudent, int idCourse): float

+readGradeStudent(Course): List<float>

+readTranscript(int idStudent): List<Grade>

EmailSenderService

+sendEmail(String recipientEmail, String subject, String htmlContent): void

StudentService

+createStudent(String name, String surname, Date dateBirth, String contact, String schoolYear)

+updateStudent(Integer id, String name, String surname, Date dateBirth, String contact, String schoolYear): void

+deleteStudent(int idStudent): void

+readStudentList(): List<Student>

+readStudent(int idStudent): Student

+searchStudent(String searchTerm): List<Student>

+getStudentsByPromo(String[] promos): List<Student>

+registrationCourse(Course course , Student student): void

(createStudent)  
renvoi Map<String,  
String>