Model::Main

DatabaseItem
+add_to_database(): bool
+remove_from_database(): bool
+update_in_database(new_props: Map): bool

Facultry
-administrator: string
-instructors: string[*]
-courses: string[*]
+students: string[*]
-dean: string
-name: string

Database
-connection_key: string
+initialize_db(key: string): void
-Database()
+remove_item(item: DatabaseItem): bool
+add_item(item: DatabaseItem): bool
+get_instance(): Database&
+get(string, string)

User
#userId: string
#username: string
#password_hash: string
#email: string
#faculty: string
#name: string
+get_username(): string
+get_email(): string
+set_email(): void
+set_username(): string
+check_password(password: string): bool
+User(username: string, email: string, password: string)

Course
-ID: string
-credit_hours: int
-professor: string
-teaching_assistants: string[0..3]
-textBook: string
+get_credit_hours(): int
+set_credit_hours(): void
+get_professor(): Instructor
+get_teaching_assistants(): Instructor
+get_students(): Student[*]
+set_textbook(): string
+set_professor(professor: Instuctor): bool
+get_assignment(): string

Student
-courses_registartions: string[0..6]
+Student(username: string, email: string, password: string)
+getCourses(): Course[*]
+register_course(course_id: string): bool
+drop_course(course_id: string): bool
+view_total_grade(course_id: string): double
+get_average_grade_in_all_courses(): double
+get_GPA(): double

Administrator
+add_user(user: User): bool
+Administrator(username: string, email: string, password: string)
+register_course_for_student(student: string, course: string): bool
+remove_user(user: User): bool
+add_course()

Assignment
-ID: string
-course: string
-submissions: string[*]
-type: AssignmentType
+getCourse(): Course
+getSubmissions(): AssignmentSubmission[*]
+getMinimumGrade(): double
+getMaximumGrade(): double
+getAverageGrade(): double

AssignmentSubmission
-ID: string
-student: string
-assignment: string
-grade: double
+getStudent(): student
+getAsignment(): Assignment
+getGrade(): Grade
+setGrade(grade: double): bool

«enumeration»
AssignmentType
Paper
Online

Instructor
-Courses: string[0..5]
+Instructor(username: string, email: string, password: string)
+add_course(course: Course): bool
+remove_course(course: Course): bool
+modify_grade(submission: AssignmentSubmission): bool
+modify_grade(student: Student, Assignment: Assignment): bool
+get_maximum_grade(course_id: string)
+get_minimum_grade(course_id: string)
+get_average_grade(course_id: string)
+add_teaching_assistant(course_id: string, TA: Instructor): bool
+add_student(student: Student, course: Course)

CourseRegistration
-Course: string
-Student: string
-startDate: Date
-endDate: Date
+state: CourseRegistrationState

endDate will be equal to startDate or zero in case the course is still being taken, otherwise, when finished or droped, It will take another value

«enumeration»
CourseRegistrationState
Current
Finished
Dropped
AwaitingApproval

The method modify_grade will either add of update a grade if existent

add_teaching_assistant will work in case the instructor is the professer of the course chosen and return true, otherwise return false