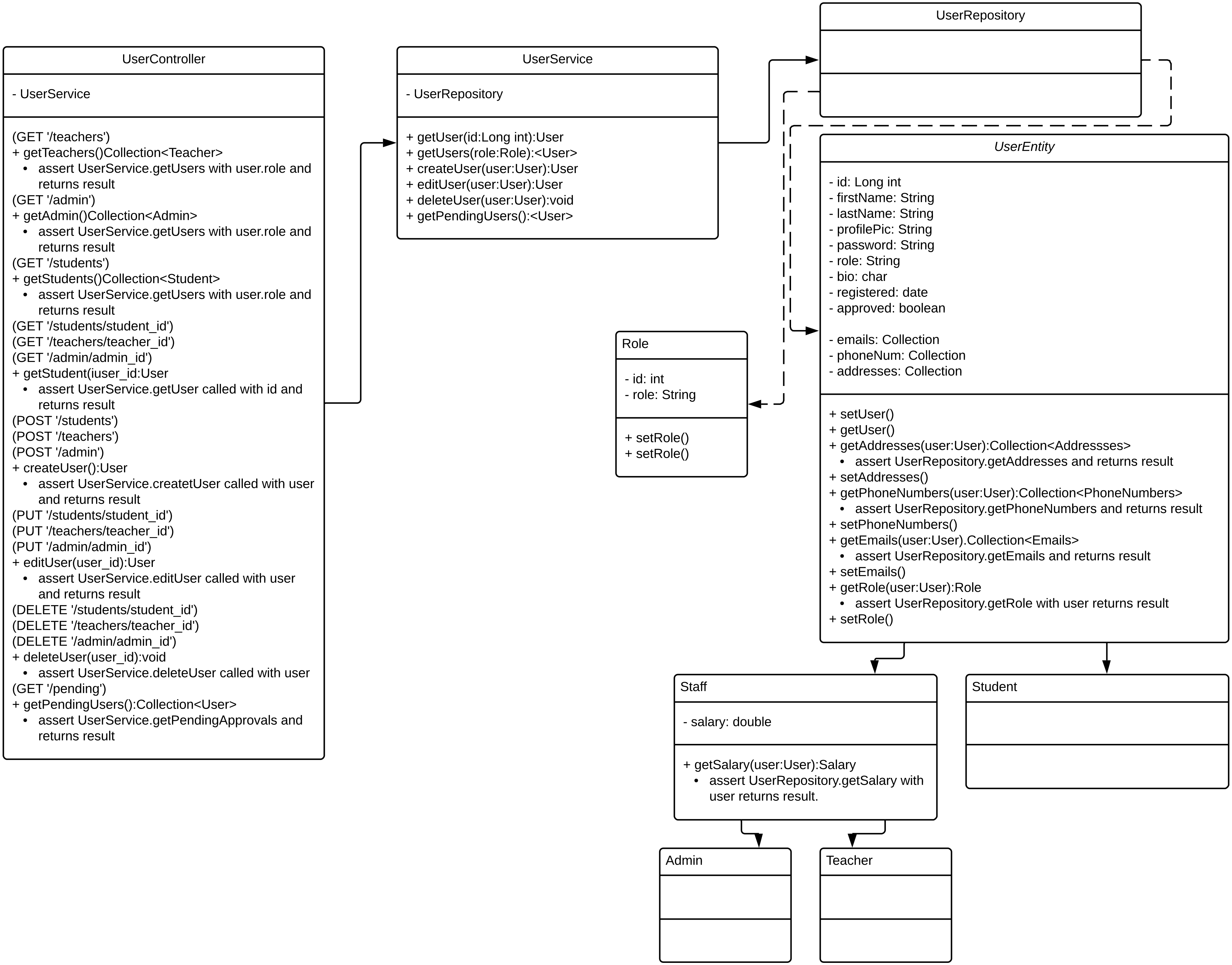


UserController

- UserService

(GET '/teachers')
+ getTeachers()Collection<Teacher>
- assert UserService.getUsers with user.role and returns result

(GET '/admin')
+ getAdmin()Collection<Admin>
- assert UserService.getUsers with user.role and returns result

(GET '/students')
+ getStudents()Collection<Student>
- assert UserService.getUsers with user.role and returns result

(GET '/students/student_id')
(GET '/teachers/teacher_id')
(GET '/admin/admin_id')
+ getStudent(iuser_id:User
- assert UserService.getUser called with id and returns result

(POST '/students')
(POST '/teachers')
(POST '/admin')
+ createUser():User
- assert UserService.createtUser called with user and returns result

(PUT '/students/student_id')
(PUT '/teachers/teacher_id')
(PUT '/admin/admin_id')
+ editUser(user_id):User
- assert UserService.editUser called with user and returns result

(DELETE '/students/student_id')
(DELETE '/teachers/teacher_id')
(DELETE '/admin/admin_id')
+ deleteUser(user_id):void
- assert UserService.deleteUser called with user

(GET '/pending')
+ getPendingUsers():Collection<User>
- assert UserService.getPendingApprovals and returns result

UserService

- UserRepository

+ getUser(id:Long int):User
+ getUsers(role:Role):<User>
+ createUser(user:User):User
+ editUser(user:User):User
+ deleteUser(user:User):void
+ getPendingUsers():<User>

UserRepository

UserEntity

- id: Long int
- firstName: String
- lastName: String
- profilePic: String
- password: String
- role: String
- bio: char
- registered: date
- approved: boolean

- emails: Collection
- phoneNum: Collection
- addresses: Collection

+ setUser()
+ getUser()
+ getAddresses(user:User):Collection<Addressses>
- assert UserRepository.getAddresses and returns result

+ setAddresses()
+ getPhoneNumbers(user:User):Collection<PhoneNumbers>
- assert UserRepository.getPhoneNumbers and returns result

+ setPhoneNumbers()
+ getEmails(user:User).Collection<Emails>
- assert UserRepository.getEmails and returns result

+ setEmails()
+ getRole(user:User):Role
- assert UserRepository.getRole with user returns result

+ setRole()

Role

- id: int
- role: String

+ setRole()
+ setRole()

Staff

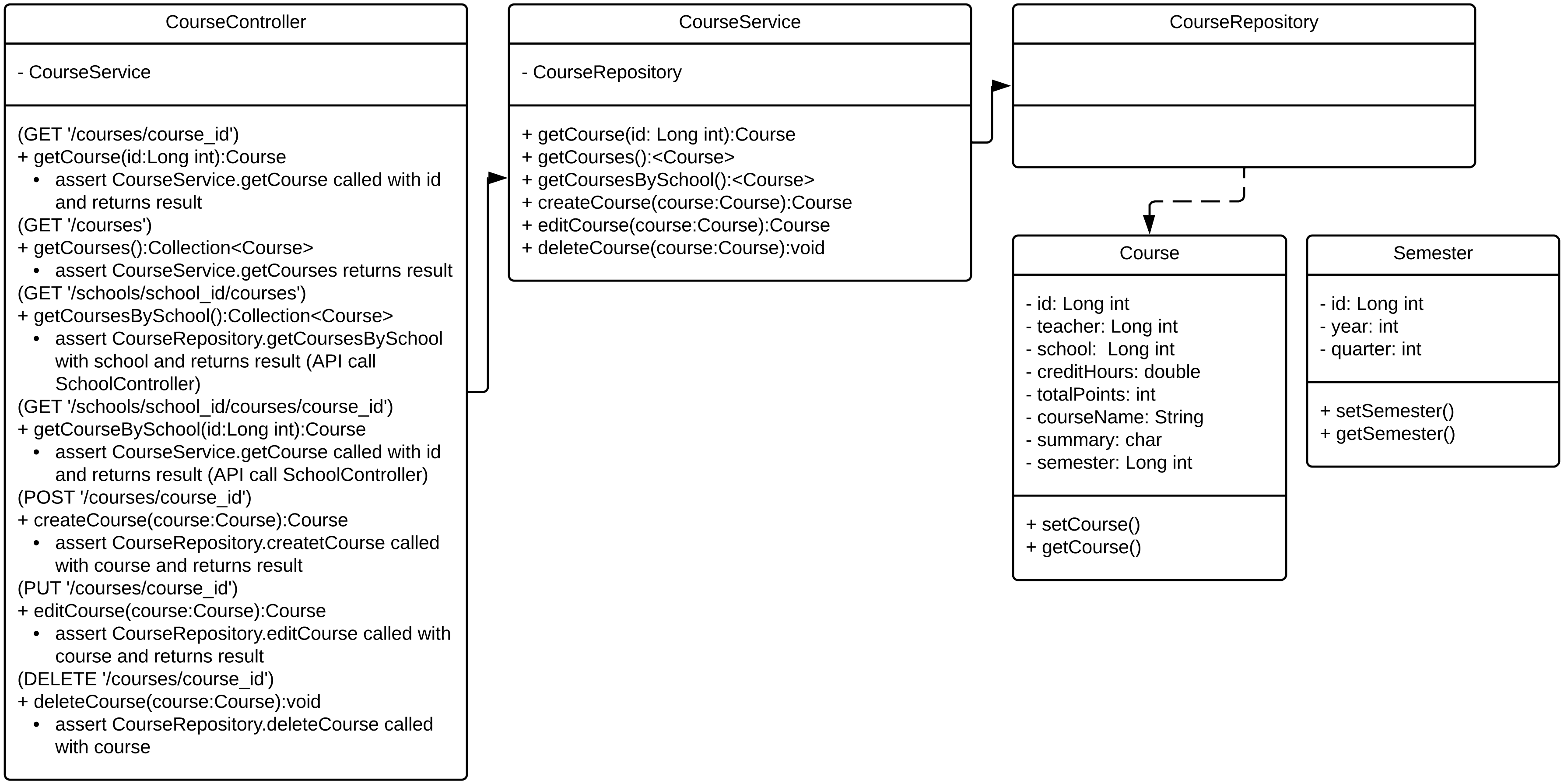
- salary: double

+ getSalary(user:User):Salary
- assert UserRepository.getSalary with user returns result.

Student

Admin

Teacher



CourseController

- CourseService

(GET '/courses/course_id')
+ getCourse(id:Long int):Course
  - assert CourseService.getCourse called with id and returns result

(GET '/courses')
+ getCourses():Collection<Course>
  - assert CourseService.getCourses returns result

(GET '/schools/school_id/courses')
+ getCoursesBySchool():Collection<Course>
  - assert CourseRepository.getCoursesBySchool with school and returns result (API call SchoolController)

(GET '/schools/school_id/courses/course_id')
+ getCourseBySchool(id:Long int):Course
  - assert CourseService.getCourse called with id and returns result (API call SchoolController)

(POST '/courses/course_id')
+ createCourse(course:Course):Course
  - assert CourseRepository.createtCourse called with course and returns result

(PUT '/courses/course_id')
+ editCourse(course:Course):Course
  - assert CourseRepository.editCourse called with course and returns result

(DELETE '/courses/course_id')
+ deleteCourse(course:Course):void
  - assert CourseRepository.deleteCourse called with course

CourseService

- CourseRepository

+ getCourse(id: Long int):Course
+ getCourses():<Course>
+ getCoursesBySchool():<Course>
+ createCourse(course:Course):Course
+ editCourse(course:Course):Course
+ deleteCourse(course:Course):void

CourseRepository

Course

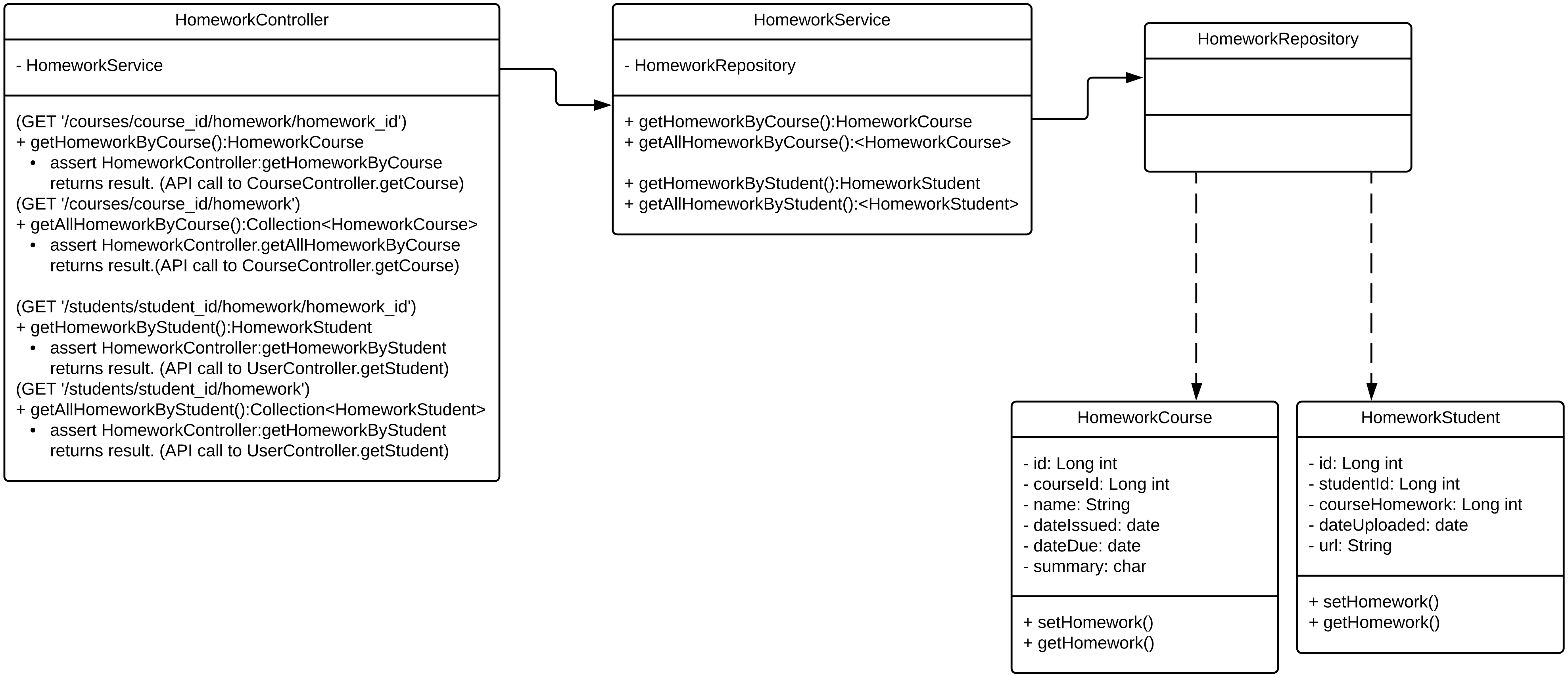
- id: Long int
- teacher: Long int
- school: Long int
- creditHours: double
- totalPoints: int
- courseName: String
- summary: char
- semester: Long int

+ setCourse()
+ getCourse()

Semester

- id: Long int
- year: int
- quarter: int

+ setSemester()
+ getSemester()



HomeworkController

- HomeworkService

(GET '/courses/course_id/homework/homework_id')
+ getHomeworkByCourse():HomeworkCourse
- assert HomeworkController:getHomeworkByCourse returns result. (API call to CourseController.getCourse)

(GET '/courses/course_id/homework')
+ getAllHomeworkByCourse():Collection<HomeworkCourse>
- assert HomeworkController.getAllHomeworkByCourse returns result.(API call to CourseController.getCourse)

(GET '/students/student_id/homework/homework_id')
+ getHomeworkByStudent():HomeworkStudent
- assert HomeworkController:getHomeworkByStudent returns result. (API call to UserController.getStudent)

(GET '/students/student_id/homework')
+ getAllHomeworkByStudent():Collection<HomeworkStudent>
- assert HomeworkController:getHomeworkByStudent returns result. (API call to UserController.getStudent)

HomeworkService

- HomeworkRepository

+ getHomeworkByCourse():HomeworkCourse
+ getAllHomeworkByCourse():<HomeworkCourse>

+ getHomeworkByStudent():HomeworkStudent
+ getAllHomeworkByStudent():<HomeworkStudent>

HomeworkRepository

HomeworkCourse

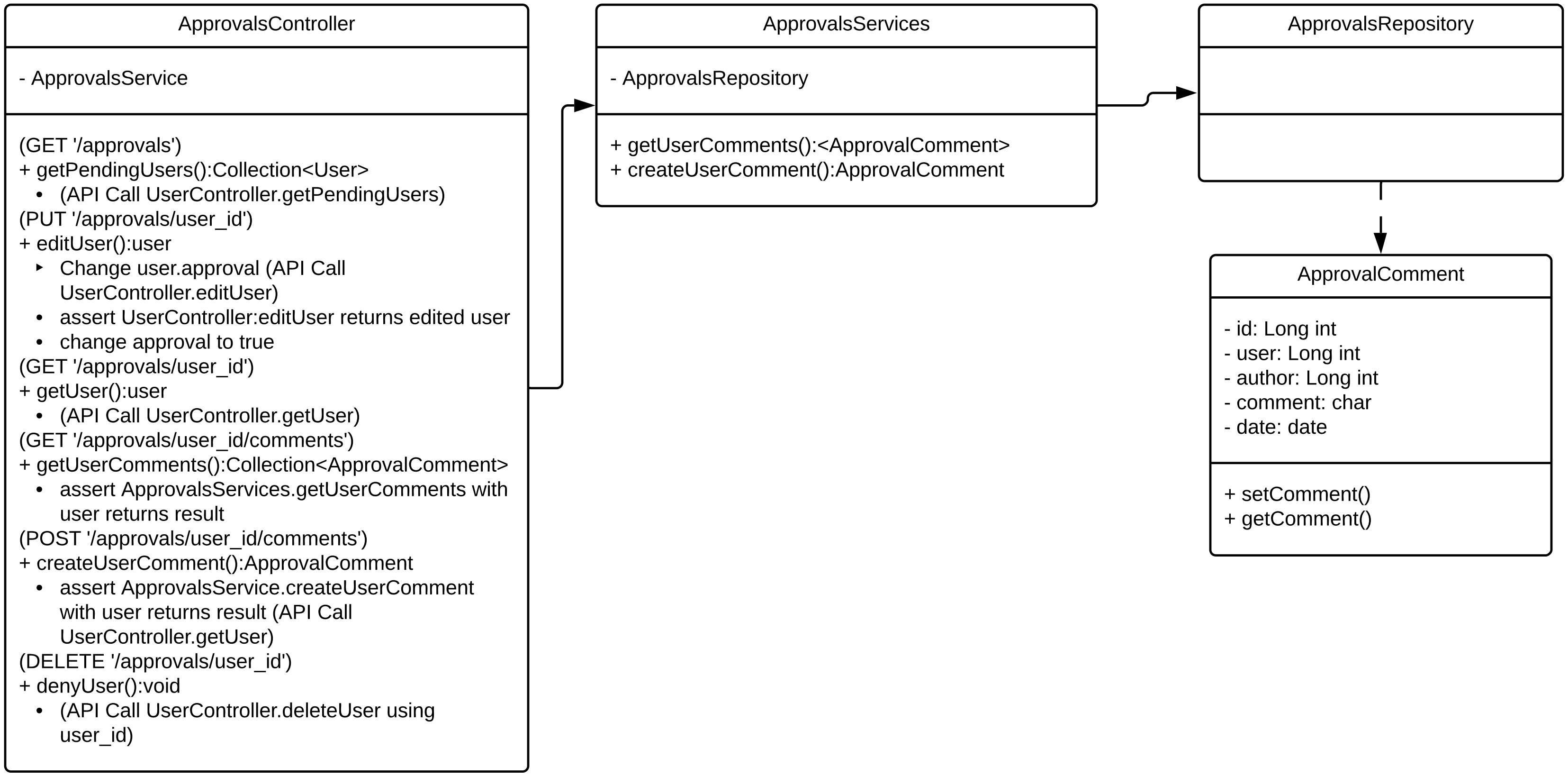
- id: Long int
- courseId: Long int
- name: String
- dateIssued: date
- dateDue: date
- summary: char

+ setHomework()
+ getHomework()

HomeworkStudent

- id: Long int
- studentId: Long int
- courseHomework: Long int
- dateUploaded: date
- url: String

+ setHomework()
+ getHomework()



ApprovalsController

- ApprovalsService

(GET '/approvals')
+ getPendingUsers():Collection<User>
- (API Call UserController.getPendingUsers)

(PUT '/approvals/user_id')
+ editUser():user
- ‣ Change user.approval (API Call UserController.editUser)
- assert UserController:editUser returns edited user
- change approval to true

(GET '/approvals/user_id')
+ getUser():user
- (API Call UserController.getUser)

(GET '/approvals/user_id/comments')
+ getUserComments():Collection<ApprovalComment>
- assert ApprovalsServices.getUserComments with user returns result

(POST '/approvals/user_id/comments')
+ createUserComment():ApprovalComment
- assert ApprovalsService.createUserComment with user returns result (API Call UserController.getUser)

(DELETE '/approvals/user_id')
+ denyUser():void
- (API Call UserController.deleteUser using user_id)

ApprovalsServices

- ApprovalsRepository

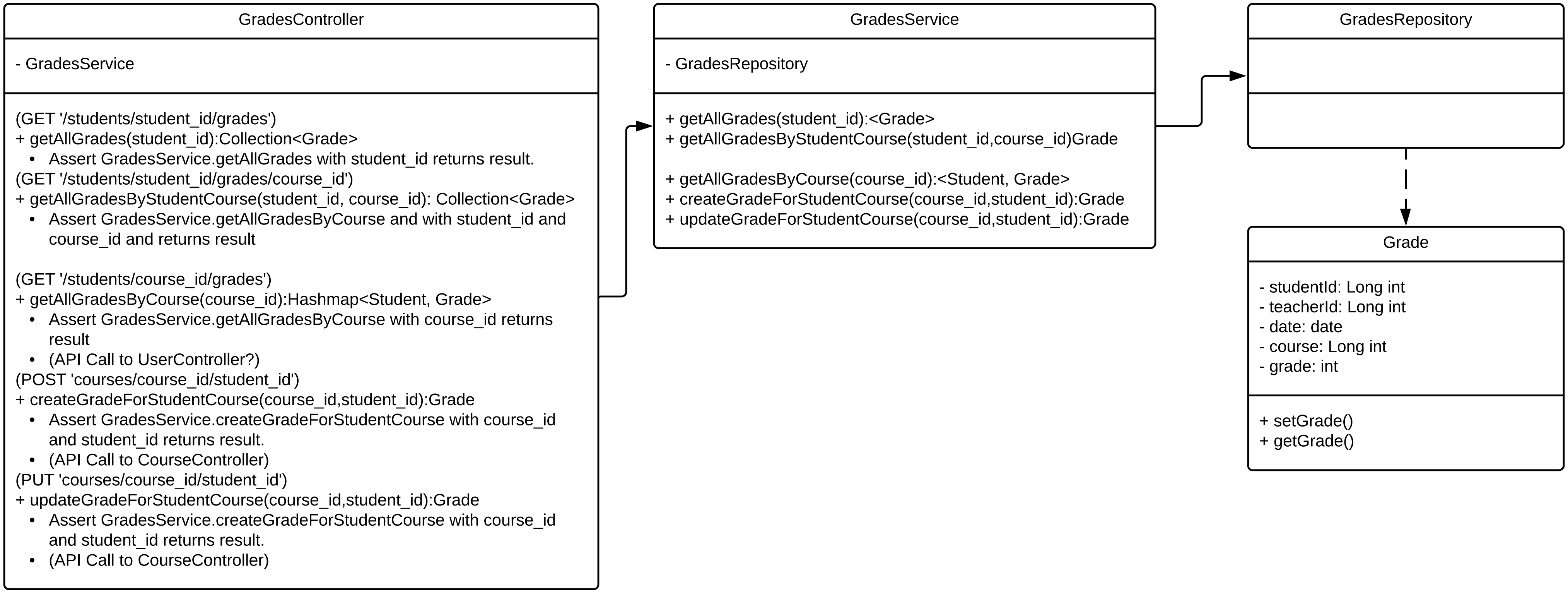
+ getUserComments():<ApprovalComment>
+ createUserComment():ApprovalComment

ApprovalsRepository

ApprovalComment

- id: Long int
- user: Long int
- author: Long int
- comment: char
- date: date

+ setComment()
+ getComment()



GradesController

- GradesService

(GET '/students/student_id/grades')
+ getAllGrades(student_id):Collection<Grade>
- Assert GradesService.getAllGrades with student_id returns result.

(GET '/students/student_id/grades/course_id')
+ getAllGradesByStudentCourse(student_id, course_id): Collection<Grade>
- Assert GradesService.getAllGradesByCourse and with student_id and course_id and returns result

(GET '/students/course_id/grades')
+ getAllGradesByCourse(course_id):Hashmap<Student, Grade>
- Assert GradesService.getAllGradesByCourse with course_id returns result
- (API Call to UserController?)

(POST 'courses/course_id/student_id')
+ createGradeForStudentCourse(course_id,student_id):Grade
- Assert GradesService.createGradeForStudentCourse with course_id and student_id returns result.
- (API Call to CourseController)

(PUT 'courses/course_id/student_id')
+ updateGradeForStudentCourse(course_id,student_id):Grade
- Assert GradesService.createGradeForStudentCourse with course_id and student_id returns result.
- (API Call to CourseController)

GradesService

- GradesRepository

+ getAllGrades(student_id):<Grade>
+ getAllGradesByStudentCourse(student_id,course_id)Grade

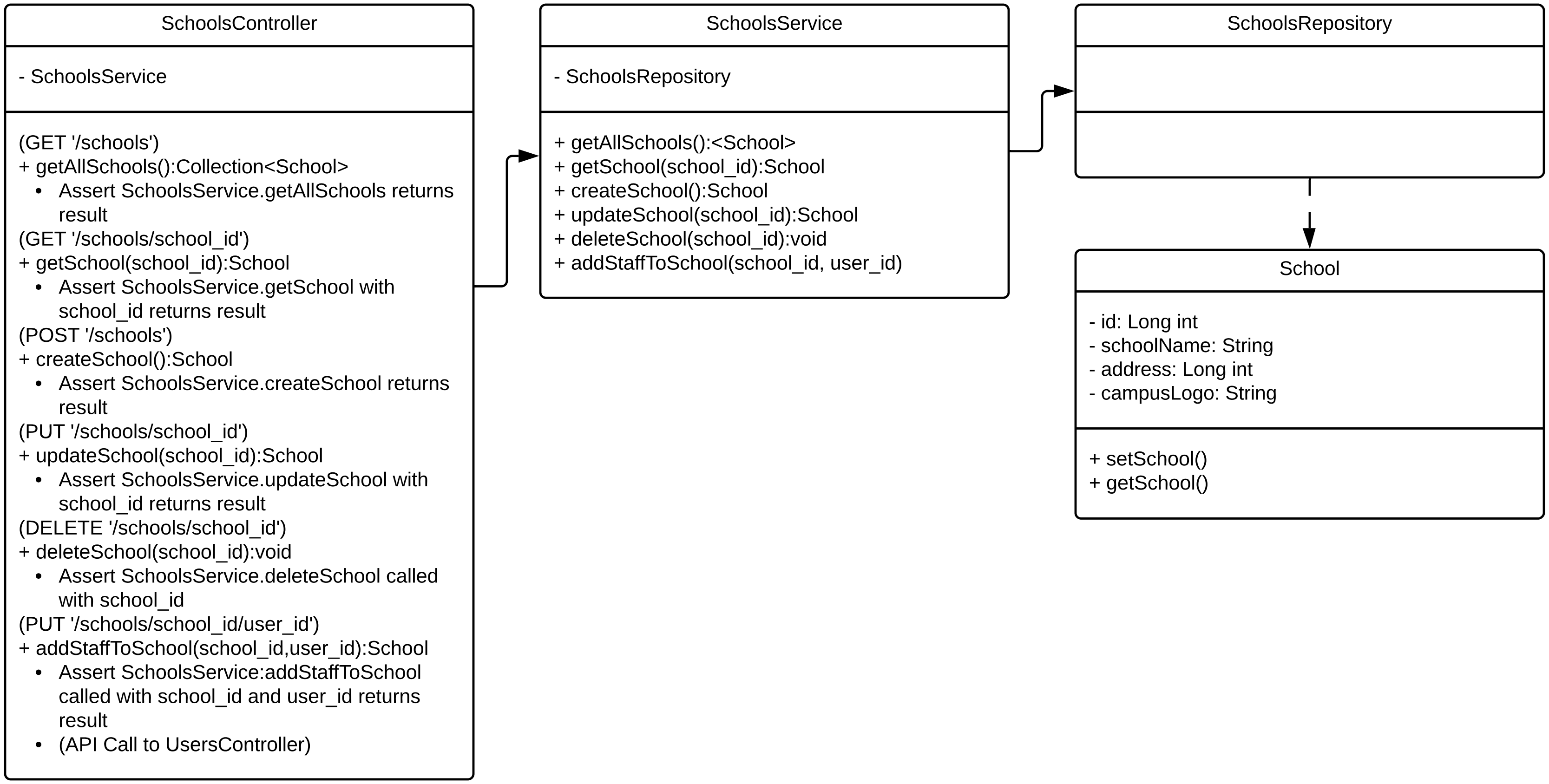
+ getAllGradesByCourse(course_id):<Student, Grade>
+ createGradeForStudentCourse(course_id,student_id):Grade
+ updateGradeForStudentCourse(course_id,student_id):Grade

GradesRepository

Grade

- studentId: Long int
- teacherId: Long int
- date: date
- course: Long int
- grade: int

+ setGrade()
+ getGrade()



SchoolsController

- SchoolsService

(GET '/schools')
+ getAllSchools():Collection<School>
- Assert SchoolsService.getAllSchools returns result

(GET '/schools/school_id')
+ getSchool(school_id):School
- Assert SchoolsService.getSchool with school_id returns result

(POST '/schools')
+ createSchool():School
- Assert SchoolsService.createSchool returns result

(PUT '/schools/school_id')
+ updateSchool(school_id):School
- Assert SchoolsService.updateSchool with school_id returns result

(DELETE '/schools/school_id')
+ deleteSchool(school_id):void
- Assert SchoolsService.deleteSchool called with school_id

(PUT '/schools/school_id/user_id')
+ addStaffToSchool(school_id,user_id):School
- Assert SchoolsService:addStaffToSchool called with school_id and user_id returns result
- (API Call to UsersController)

SchoolsService

- SchoolsRepository

+ getAllSchools():<School>
+ getSchool(school_id):School
+ createSchool():School
+ updateSchool(school_id):School
+ deleteSchool(school_id):void
+ addStaffToSchool(school_id, user_id)

SchoolsRepository

School

- id: Long int
- schoolName: String
- address: Long int
- campusLogo: String

+ setSchool()
+ getSchool()