Use Case Name and Actors	REST Endpoint	Request Body (Typescript Interface)	Response Body (Typescript Interface)	Associated Java Classes
Manage his/her own account **All actors**	**GET** **/v1/me**		**IAccount**	**AcademicPersonnel** **AcademicPersonnelRepository** **AcademicPersonnelController** **Student** **StudentRepository** **StudentController**
	PATCH **/v1/me**	**profilePicture?:** string, **oldPassword:** string, **newPassword?:** string	**IAccount**	
Manage accounts **Admin**	**POST** **/v1/accounts**	**IAccountRequest**	**IAccount**	**AcademicPersonnel** **AcademicPersonnelRepository** **AcademicPersonnelController** **Student** **StudentRepository** **StudentController**
	GET **/v1/accounts/:id**?type=*student*|*instructor*|*departmentManage*r		**IAccount**	
	PATCH **/v1/accounts/:id**	**oldType**: instructor|departmentManager **newType**:instructor|departmentManager	**IAccount**	
	DELETE **/v1/accounts/:id**			

Create/Get/Start Semester **Admin**	**POST /v1/semesters**	**ISemesterRequest**	**ISemester**	**Semester** **SemesterRepository** **SemesterController**
	GET /v1/semesters:id		**ISemester**	
	PATCH /v1/semesters/:id	**startDate?**: number **endDate?**: number **isActive?**: boolean	**ISemester**	
Manage Courses **Admin**	**POST /v1/courses**	**ICourseRequest**	**ICourse**	**Course** **CourseRepository** **CourseController**
	GET /v1/courses/:id		**ICourse**	
Manage Sections **Admin**	**POST /v1/sections**	**ISectionRequest**		**Section** **SectionRepository** **SectionController**
	GET /v1/sections/:id		**ISection**	
Get Sections **Student**	**GET /v1/student/:id/sections**		**ISection[]**	
Get Sections **Instructor**	**GET /v1/instructor/:id/sections**		**ISection[]**	
Add instructor to the course **Department Manager**	**PATCH /v1/courses/:id**	**instructorID**: number	**ICourse**	**Course** **CourseRepository** **CourseController**
Enroll in a course section **Student**	**PATCH /v1/students/:id/sections/:id**	**is_enrolled:** true		
Create survey **Instructor**	**POST /v1/surveys**	**ISurveyRequest**	**ISurvey**	**Survey** **SurveyRepository** **SurveyController**
Access survey **Admin**	**GET /v1/surveys/:id**		**ISurvey[]**	
Access surveys **Student**	**GET /v1/students/:id/surveys**		**ISurvey[]**	
Access surveys **Instructor**	**GET /v1/instructors**		**ISurvey[]**	

	/:id/surveys			
Create questions in the questions pool **Instructor, Department Manager**	**POST /v1/questions**	**IQuestionRequest**	**IQuestion**	**Question QuestionRepository QuestionController**
Access questions from the question pool **All actors**	**GET /v1/questions/:id**		**IQuestion**	
Access questions of a survey **All actors**	**GET /v1/surveys/:id/questions**		**IQuestion[]**	
Insert a question in a survey **Instructor**	**POST /v1/surveys/:id/questions**	**question_id:** number **question_order:** number	**IQuestion[]**	
Reorder a question in a survey **Instructor**	**PATCH /v1/surveys/:id/questions/:id**	**question_order:** number		
Access evaluations of his/her surveys **Student**	**GET /v1/students/:id/evaluations**		**IEvaluation[]**	**Evaluation EvaluationRepository EvaluationController**
Create an evaluation of a survey **Student**	**POST /v1/students/:id/evaluations**	**survey_id:** number	**IEvaluation**	
Modify an evaluation of a survey **Student**	**PATCH /v1/students/:id/evaluations/:id**	**survey_id:** number **do_it_later_count:** number **is_submitted:** number		
Answer a question in an evaluation form **Student**	**POST /v1/students/:id/evaluations/:id/answers**	**IEvaluationAnswerRequest**		**Answer AnswerRepository AnswerController**
Access answers in an evaluation form **Student**	**GET /v1/students/:id/evaluations/:id/answers**		**IEvaluationAnswer**	
Change an answer in an evaluation form **Student**	**PATCH /v1/students/:id/evaluations/:id/answer/:id**	**is_choiceA_selected**: boolean **is_choiceB_selected**: boolean	**IEvaluationAnswer**	

		is_choiceC_selected : boolean **is_choiceD_selected** : boolean **is_choiceE_selected** : boolean **open_ended_answer : **string		

TYPESCRIPT INTERFACES (FRONT-END)

```
interface IDepartment {
  department_id: number,
  name: string
}

interface IAccountRequest {
  account_type: “student” | “instructor” | “departmentManager”,
  department_id: number,
  manager_of?: number,
  username: string,
  name: string,
  surname: string,
  email: string,
  profile_picture_url: string
}

interface IAccount extends IAccountRequest {
  id: number,
  department_name: string,
  is_banned?: boolean
}

interface ISemesterRequest {
  name: string,
  start_date: date,
  end_date: date,
  is_active: boolean
}

interface ISemester extends ISemesterRequest {
  semester_id: number
}

interface ICourseRequest {
  instructor_id: number,
  department_id: number,
  name: string,
  credits: number
}

interface ICourse extends ICourseRequest {
  course_id: number,
  instructor_name: string,
  instructor_surname: string,
  department_name: string
}

interface ISectionRequest {
  course_id: number,
  instructor_id: number,
  semester_id: number,
  section_no: number
}

interface ISection extends ISectionRequest {
  section_id: number,
  course_name: number,
  instructor_name: string,
  instructor_surname: string,
  semester_name: string
}

interface IQuestionRequest {
  written_by_instructor: number,
  written_by_department: number,
  course_id: number,
  question_type: string,
  is_for_course: boolean,
  is_required: boolean,
  question_title: string,
  question_text: string,
  question_image_url: string,
  choiceA: string,
  choiceB: string,
  choiceC: string,
  choiceD: string,
  choiceE: string
}

interface IQuestion extends IQuestionRequest {
  question_id: number,
  written_by_instructor_name: string,
  written_by_instructor_surname: string,
  written_by_department_name: string,
  course_name: string
}

interface ISurveyRequest {
  section_id: number
  is_for_course: boolean,
  is_submitted: boolean,
  do_it_later_count: number,
  is_reevaluate_request_sent: boolean,
  start_date: date,
  end_date: date
}

interface ISurvey extends ISurveyRequest {
  survey_id: number,
  course_name: string,
  instructor_name: string,
  instructor_surname: string,
  semester_name: string
}

interface IEvaluationAnswerRequest {
  evaluation_id: number,
  question_id: number,
  is_choiceA_selected: boolean,
  is_choiceB_selected: boolean,
  is_choiceC_selected: boolean,
  is_choiceD_selected: boolean,
  is_choiceE_selected: boolean,
  open_ended_answer: string
}

interface IEvaluationAnswer extends IEvaluationAnswerRequest {
  answer_id: number,
  question_title: string,
  question_text: string,
  section_id: number,
  is_for_course: boolean
}
```