

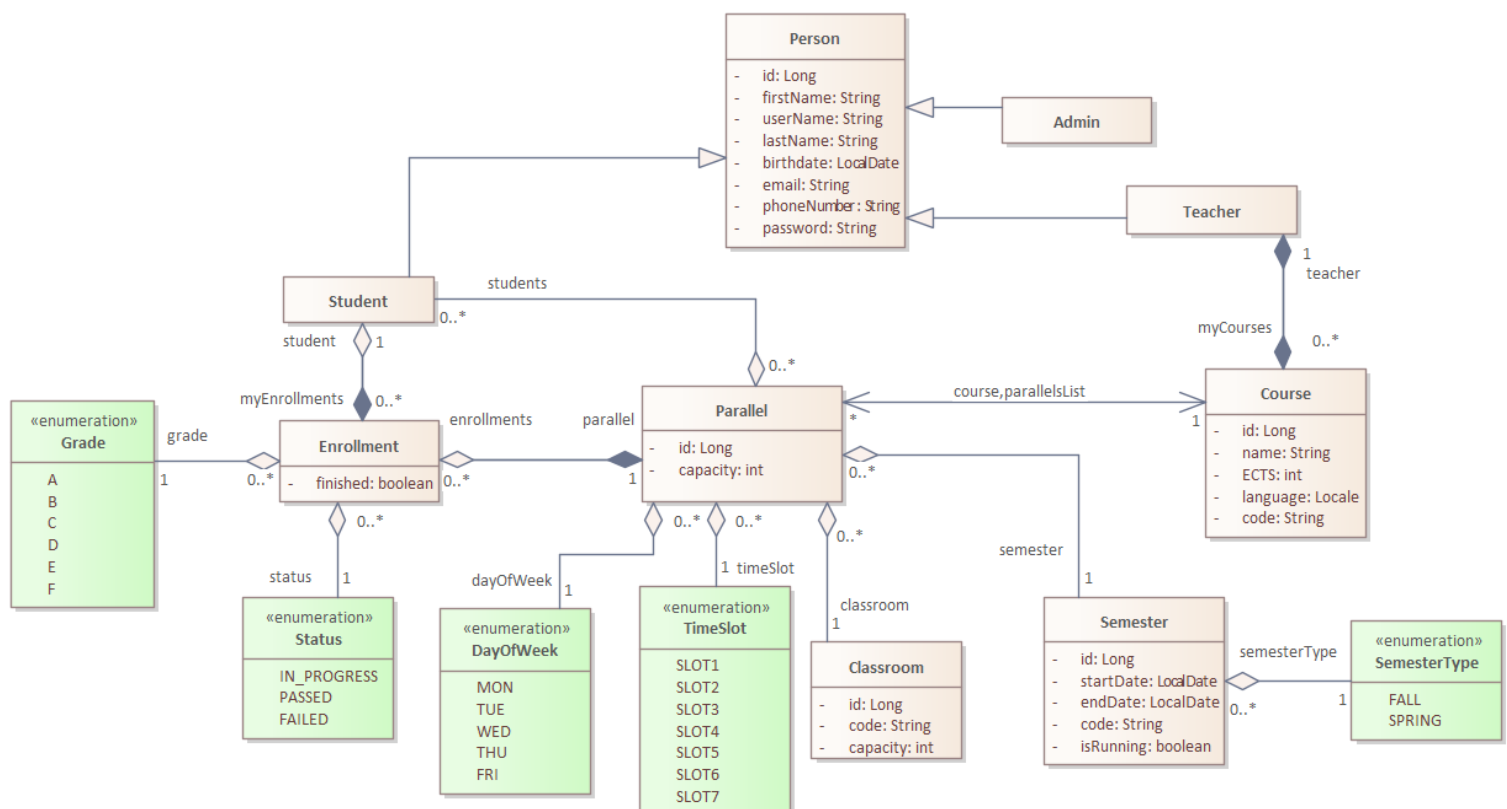
Software Requirements Specification

The SIS is a Spring Boot Application designed to facilitate university management relating to teachers, courses, enrollments, grades, etc.

Core Entities and Their Usage:

- **Person**: The foundational entity from which all user types derive, capturing essential details like name, contact information, and credentials. It serves as a single source of truth for all personal data within the system.
- **Admin**: Inherits from `Person`, representing administrative users with access to management of semesters and classrooms.
- **Student**: Extends `Person`, with relationships to `Enrollment`, it's the central entity for studentspecific operations, like course enrollments and academic tracking.
- **Teacher**: Also a type of `Person`, linking to `Course`, representing faculty members responsible for delivering course content and managing academic progress.
- **Course**: A key entity that defines academic courses, associated with `Teacher` and `Parallel`, including details like course code, ECTS credits, and language.
- **Parallel**: Represents individual class sections within a course, containing scheduling details such as time slots and days, along with classroom and student enrollments.
- **Enrollment**: Captures the association between `Student` and `Parallel`, tracking students' course registrations, academic grades, and enrollment status.
- **Semester**: Defines the academic terms, marked by start and end dates, and identified by a unique code reflecting the semester type and year.
- **Classroom**: Physical spaces where courses are conducted, identified by a unique code and defined by their capacity.

class SIS



User Roles

Administrator: Sets academic terms, adds classrooms, designates active semesters.

Teacher: Lists courses, schedules classes, grades students, avoids booking conflicts.

Student: Enrolls in classes, views schedules, withdraws from courses, checks grades.

System Constraints

No Parallel Types: No differentiation between lectures and practices.

No Study Plans: Students are not required to follow specific course plans.

Classification by Grade: Each enrollment is assigned one final grade.

No Multiple Roles: Users are restricted to a single role within the system.

One Teacher per Class: Only one teacher is assigned to each class section.

Minimal Personal Data: The system does not store detailed personal information.

Business Requirements

BR0: Sign-up will be possible only for one role determined by a keypass and only users older or equal to 18 years of age can sign up.

BR1: Administrators are responsible for creating academic semesters.

BR2: Administrators manage the creation and maintenance of classroom resources.

BR3: Administrators designate the active semester for the institution.

BR4: Teachers may list class sections ("parallels") only for the upcoming semester, anytime during the current semester.

BR5: Teachers can evaluate students enrolled in their class sections within the current semester.

BR6: Teachers set the capacity of a class section up to the maximum capacity of the assigned classroom.

BR7: Teachers must ensure no scheduling conflicts arise due to room reservations for the given semester.

BR8: Students enroll directly into specific class sections, not preliminarily into courses.

BR9: Students can enroll only in class sections listed for the next semester and only during the current semester.

BR11: Students can withdraw their enrollment at any time before the semester begins.

BR12: Students must adhere to a maximum of 50 credits per semester.

BR13: Students cannot enroll in the same class section more than once per semester.

BR14: Students are prohibited from enrolling in class sections that overlap in schedule.

BR15: Students who pass a class cannot enroll in the same class again.

BR16: Students are permitted to re-enroll in a class they previously failed, with a maximum of one retake allowed.

BR17: The system will enforce additional constraints to prevent actions like enrolling in conflicting class sections, grading non-existent students, etc. These will be handled through appropriate exception mechanisms based on business logic validations.

Communication Interface

Type	Endpoint	Meaning
The following 2 endpoints will cause: - AlreadyLoggedInException in case user is logged in		
POST	/signup	to create a new account
POST	/login	to log in and start the session with person type and their id
All of the following endpoints will cause: - NotLoggedInException in case user is not logged in		
POST	/logout	to log out and remove the session data
GET	/me	to list user info
The following 5 endpoints will cause: - NotAdminException in case the logged in user is not of type admin		
POST	/admin/semester	Create a new semester
GET	/admin/semester	List all created semesters
POST	/admin/semester/{semesterId}	Will set semester's isActive field with the given id as true, and the other isActive semester as false
POST	/admin/room	Create a new room
GET	/admin/room	List all created rooms
The following 6 endpoints will cause: - NotTeacherException in case the logged in user is not of type admin		
POST	/teacher/course	Create a new course
GET	/teacher/course	List all of my courses
POST	/teacher/parallel/{courseId}	Create a new parallel for a given course
GET	/teacher/parallel/{courseId}	List all of parallels for my course id
GET	/teacher/students/{parallelId}	List all signed up students for my parallelId
POST	/teacher/grade/{parallelId}/{studentId}	Grade a student within my parallel
The following 6 endpoints will cause: - NotStudentException in case the logged in user is not of type admin		
GET	/student/course	List all courses with parallels the next

		semester
GET	/student/parallel/{courseId}	List all parallels for a given course available the next semester
POST	/student/enroll/{parallelId}	Enroll in a next semester's parallel
DELETE	/student/enroll/{parallelId}	Revert such enrollment
GET	/student/schedule/{semesterCode}	See all of parallels (and inherently the schedule, course, etc) that I am enrolled in for a given semester
GET	/student/report	See report of all my enrollments including, semester, courseName, grade and status

There are more exceptions to be defined for bad requests, like signing up for a conflicting parallel, grading a nonexistent student, etc. Above are only guarding exceptions defining which user type has access to what endpoints.