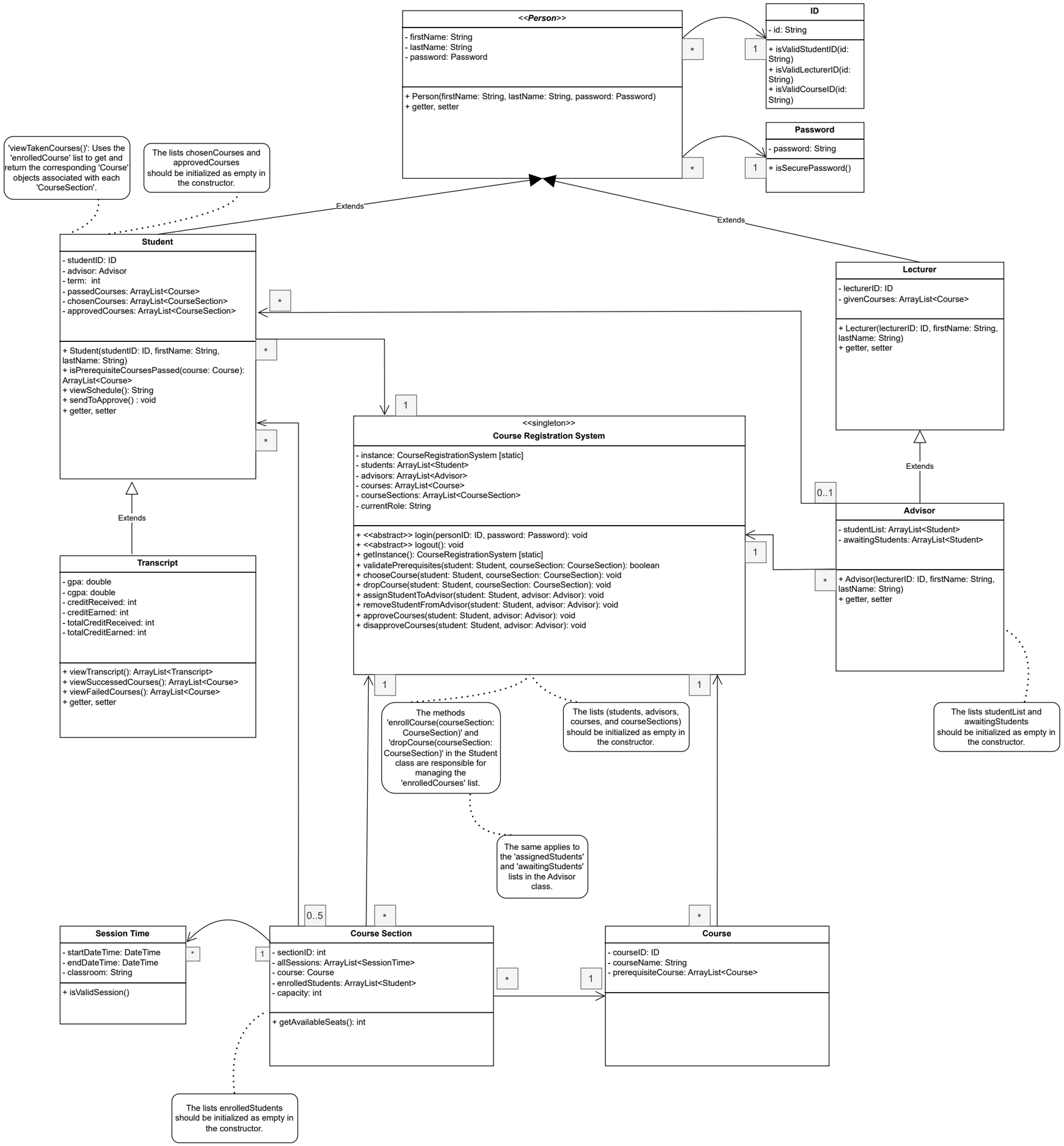
<<Person>>
- firstName: String
- lastName: String
- password: Password
+ Person(firstName: String, lastName: String, password: Password)
+ getter, setter
ID
- id: String
+ isValidStudentID(id: String)
+ isValidLecturerID(id: String)
+ isValidCourseID(id: String)
Password
- password: String
+ isSecurePassword()
'viewTakenCourses()': Uses the 'enrolledCourse' list to get and return the corresponding 'Course' objects associated with each 'CourseSection'.
The lists chosenCourses and approvedCourses should be initialized as empty in the constructor.
Extends
Extends
Student
- studentID: ID
- advisor: Advisor
- term: int
- passedCourses: ArrayList<Course>
- chosenCourses: ArrayList<CourseSection>
- approvedCourses: ArrayList<CourseSection>
+ Student(studentID: ID, firstName: String, lastName: String)
+ isPrerequisiteCoursesPassed(course: Course): ArrayList<Course>
+ viewSchedule(): String
+ sendToApprove() : void
+ getter, setter
Lecturer
- lecturerID: ID
- givenCourses: ArrayList<Course>
+ Lecturer(lecturerID: ID, firstName: String, lastName: String)
+ getter, setter
Extends
Extends
Transcript
- gpa: double
- cgpa: double
- creditReceived: int
- creditEarned: int
- totalCreditReceived: int
- totalCreditEarned: int
+ viewTranscript(): ArrayList<Transcript>
+ viewSuccessedCourses(): ArrayList<Course>
+ viewFailedCourses(): ArrayList<Course>
+ getter, setter
<<singleton>>
Course Registration System
- instance: CourseRegistrationSystem [static]
- students: ArrayList<Student>
- advisors: ArrayList<Advisor>
- courses: ArrayList<Course>
- courseSections: ArrayList<CourseSection>
- currentRole: String
+ <<abstract>> login(personID: ID, password: Password): void
+ <<abstract>> logout(): void
+ getInstance(): CourseRegistrationSystem [static]
+ validatePrerequisites(student: Student, courseSection: CourseSection): boolean
+ chooseCourse(student: Student, courseSection: CourseSection): void
+ dropCourse(student: Student, courseSection: CourseSection): void
+ assignStudentToAdvisor(student: Student, advisor: Advisor): void
+ removeStudentFromAdvisor(student: Student, advisor: Advisor): void
+ approveCourses(student: Student, advisor: Advisor): void
+ disapproveCourses(student: Student, advisor: Advisor): void
Advisor
- studentList: ArrayList<Student>
- awaitingStudents: ArrayList<Student>
+ Advisor(lecturerID: ID, firstName: String, lastName: String)
+ getter, setter
The methods 'enrollCourse(courseSection: CourseSection)' and 'dropCourse(courseSection: CourseSection)' in the Student class are responsible for managing the 'enrolledCourses' list.
The lists (students, advisors, courses, and courseSections) should be initialized as empty in the constructor.
The lists studentList and awaitingStudents should be initialized as empty in the constructor.
The same applies to the 'assignedStudents' and 'awaitingStudents' lists in the Advisor class.
Session Time
- startDateTime: DateTime
- endDateTime: DateTime
- classroom: String
+ isValidSession()
Course Section
- sectionID: int
- allSessions: ArrayList<SessionTime>
- course: Course
- enrolledStudents: ArrayList<Student>
- capacity: int
+ getAvailableSeats(): int
Course
- courseID: ID
- courseName: String
- prerequisiteCourse: ArrayList<Course>
The lists enrolledStudents should be initialized as empty in the constructor.