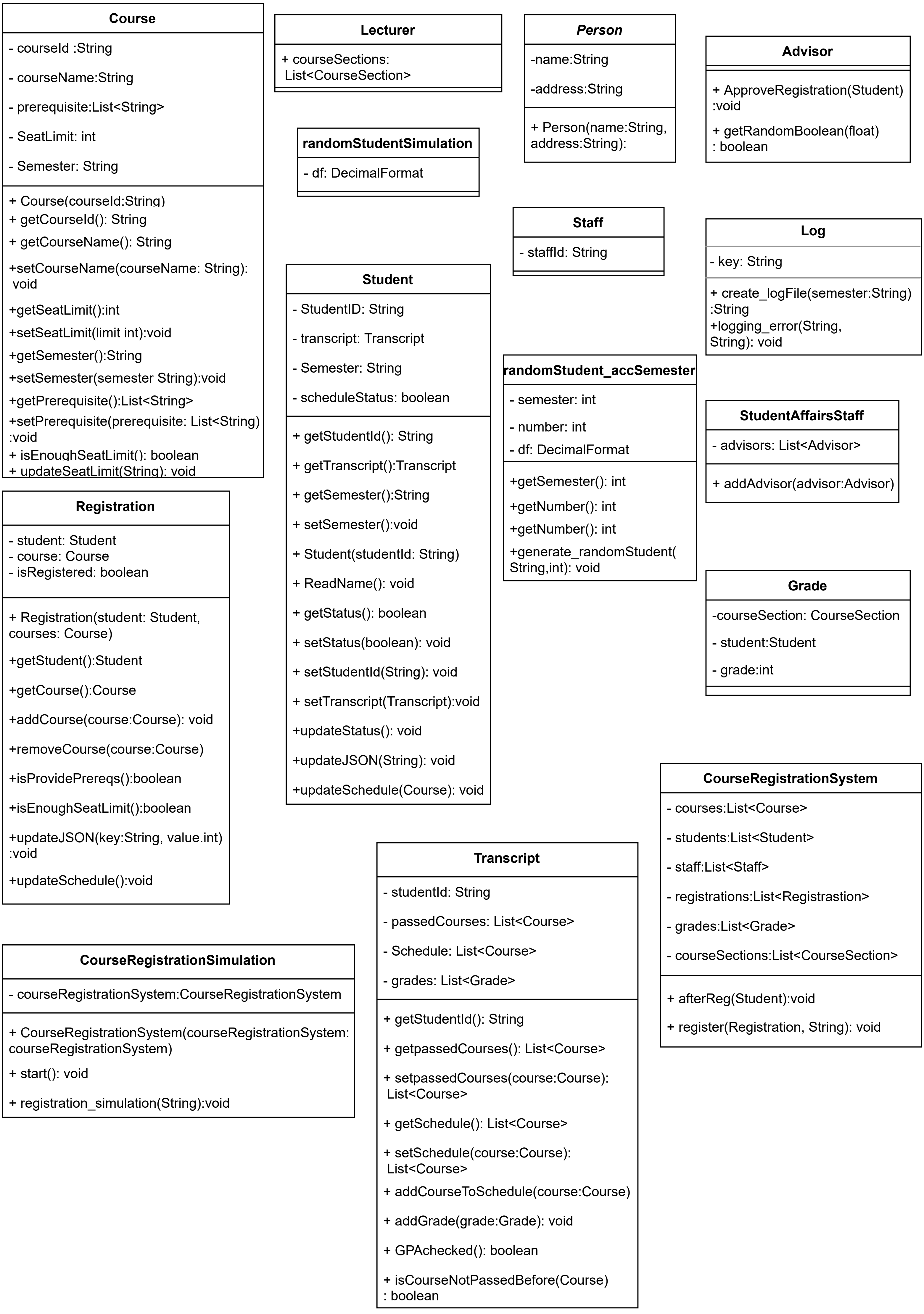

Course

- courseId :String
- courseName:String
- prerequisite:List<String>
- SeatLimit: int
- Semester: String

+ Course(courseId:String)
+ getCourseId(): String
+ getCourseName(): String
+setCourseName(courseName: String): void
+getSeatLimit():int
+setSeatLimit(limit int):void
+getSemester():String
+setSemester(semester String):void
+getPrerequisite():List<String>
+setPrerequisite(prerequisite: List<String>) :void
+ isEnoughSeatLimit(): boolean
+ updateSeatLimit(String): void

Lecturer

+ courseSections: List<CourseSection>

Person

-name:String
-address:String

+ Person(name:String, address:String):

Advisor

+ ApproveRegistration(Student) :void
+ getRandomBoolean(float) : boolean

randomStudentSimulation

- df: DecimalFormat

Staff

- staffId: String

Log

- key: String

+ create_logFile(semester:String) :String
+logging_error(String, String): void

Student

- StudentID: String
- transcript: Transcript
- Semester: String
- scheduleStatus: boolean

+ getStudentId(): String
+ getTranscript():Transcript
+ getSemester():String
+ setSemester():void
+ Student(studentId: String)
+ ReadName(): void
+ getStatus(): boolean
+ setStatus(boolean): void
+ setStudentId(String): void
+ setTranscript(Transcript):void
+updateStatus(): void
+updateJSON(String): void
+updateSchedule(Course): void

randomStudent_accSemester

- semester: int
- number: int
- df: DecimalFormat

+getSemester(): int
+getNumber(): int
+getNumber(): int
+generate_randomStudent(String,int): void

StudentAffairsStaff

- advisors: List<Advisor>

+ addAdvisor(advisor:Advisor)

Registration

- student: Student
- course: Course
- isRegistered: boolean

+ Registration(student: Student, courses: Course)
+getStudent():Student
+getCourse():Course
+addCourse(course:Course): void
+removeCourse(course:Course)
+isProvidePrereqs():boolean
+isEnoughSeatLimit():boolean
+updateJSON(key:String, value.int) :void
+updateSchedule():void

Grade

-courseSection: CourseSection
- student:Student
- grade:int

CourseRegistrationSystem

- courses:List<Course>
- students:List<Student>
- staff:List<Staff>
- registrations:List<Registrastion>
- grades:List<Grade>
- courseSections:List<CourseSection>

+ afterReg(Student):void
+ register(Registration, String): void

Transcript

- studentId: String
- passedCourses: List<Course>
- Schedule: List<Course>
- grades: List<Grade>

+ getStudentId(): String
+ getpassedCourses(): List<Course>
+ setpassedCourses(course:Course): List<Course>
+ getSchedule(): List<Course>
+ setSchedule(course:Course): List<Course>
+ addCourseToSchedule(course:Course)
+ addGrade(grade:Grade): void
+ GPAchecked(): boolean
+ isCourseNotPassedBefore(Course) : boolean

CourseRegistrationSimulation

- courseRegistrationSystem:CourseRegistrationSystem

+ CourseRegistrationSystem(courseRegistrationSystem: courseRegistrationSystem)
+ start(): void
+ registration_simulation(String):void