JSONFileManag

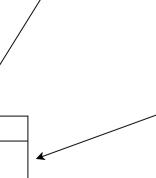- advisors: ArrayList<Advisor>
- lecturers: ArrayList<Lecturer>
- studentAffairsStaff: ArrayList<Stuc
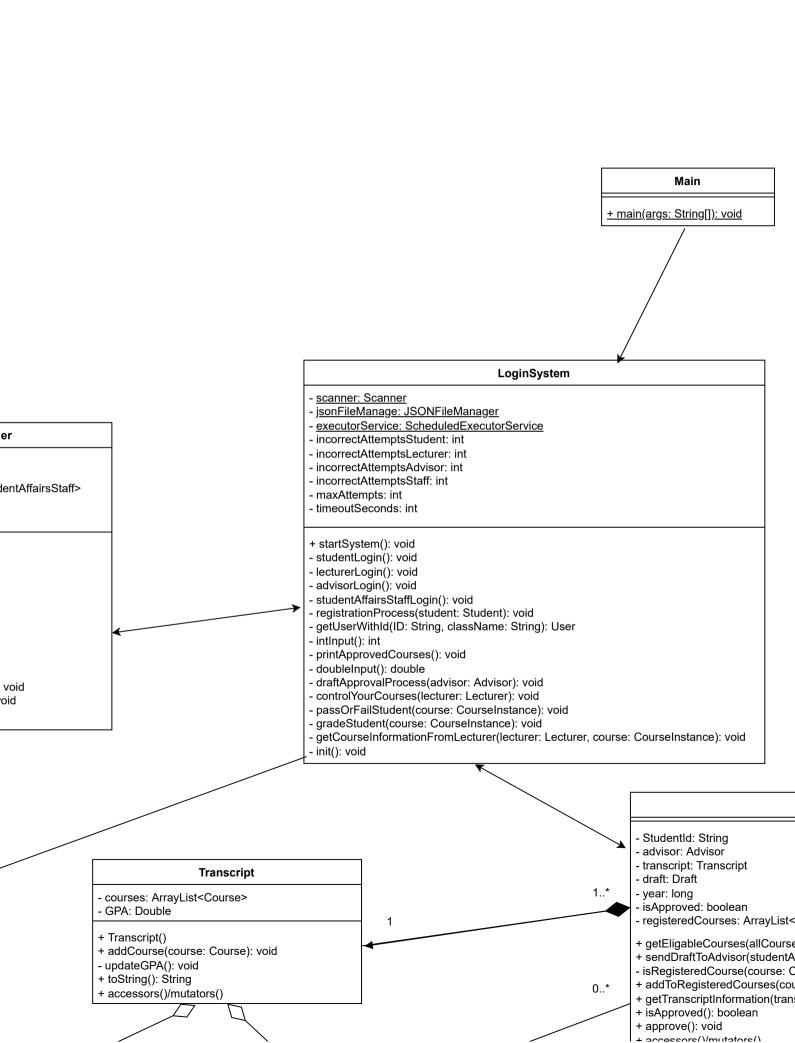- student: ArrayList<Student>
- courses: ArrayList<Course>

+ JSONFileManager()
+ writeAllDataToJSON(): void
- writeAllCoursesData(): void
- getAllCoursesData(): void
- writeAllStudentsdata(): void
- writeAllAdvisorsData(): void
- getAllAdvisorsData(): void
- getAllStudentsData(): void
- getAllLecturersData(): void
- writeAllLecturersData(): void
- writeAllStudentAffairsStaffsData():
- getAllStudentAffairsStaffsData(): v
+ accessors()/mutators()

CourseInstance

- maxNumberStudents: long
- courseCode: String
- courseName: String
- courseLecturerID: String
- registeredStudents: ArrayList<Student>

+ accessors()/mutators()

Main
+ <u>main(args: String[]): void</u>

LoginSystem

- <u>scanner: Scanner</u>
- <u>jsonFileManage: JSONFileManager</u>
- <u>executorService: ScheduledExecutorService</u>
- incorrectAttemptsStudent: int
- incorrectAttemptsLecturer: int
- incorrectAttemptsAdvisor: int
- incorrectAttemptsStaff: int
- maxAttempts: int
- timeoutSeconds: int

- + startSystem(): void
- studentLogin(): void
- lecturerLogin(): void
- advisorLogin(): void
- studentAffairsStaffLogin(): void
- registrationProcess(student: Student): void
- getUserWithId(ID: String, className: String): User
- intInput(): int
- printApprovedCourses(): void
- doubleInput(): double
- draftApprovalProcess(advisor: Advisor): void
- controlYourCourses(lecturer: Lecturer): void
- passOrFailStudent(course: CourseInstance): void
- gradeStudent(course: CourseInstance): void
- getCourseInformationFromLecturer(lecturer: Lecturer, course: CourseInstance): void
- init(): void

...er
...dentAffairsStaff>

... void
... oid

Transcript

- courses: ArrayList<Course>
- GPA: Double

+ Transcript()
+ addCourse(course: Course): void
- updateGPA(): void
+ toString(): String
+ accessors()/mutators()

- StudentId: String
- advisor: Advisor
- transcript: Transcript
- draft: Draft
- year: long
- isApproved: boolean
- registeredCourses: ArrayList<

+ getEligableCourses(allCourse
+ sendDraftToAdvisor(studentA
- isRegisteredCourse(course: C
+ addToRegisteredCourses(cou
+ getTranscriptInformation(trans
+ isApproved(): boolean
+ approve(): void
+ accessors()/mutators()

1..*

1

0..*

User

- name: String
- lastName: String
- birthDate: String
- address: String
- ssn: String
- email: String
- password: String

+ getInfo(): String {abstract}
+ accessors()/mutators()

Extends

Extends

Student

Course>

es: ArrayList<Course>): ArrayList<Course>
dvisor: Advisor): void
Course): boolean
urse: Course): void
script: Transcript): String

- staffID: String
- department: St

+ Staff()
+ Staff(staffID: S
+ accessors()/m
+ getInfo(): Strir

0..*

Extends

0..*

Staff

tring

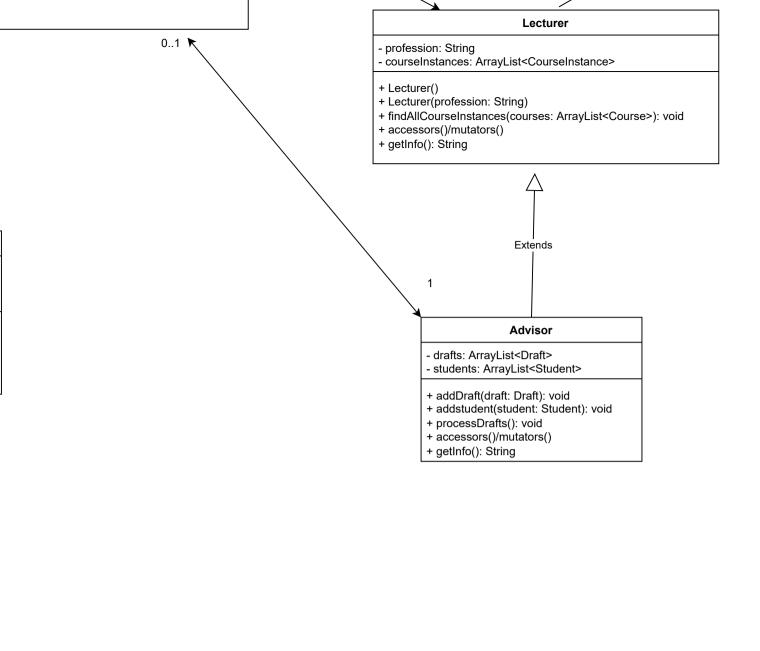String, department: String)
utators()
g

Extends

C
- student: Student
- advisor: Advisor
- courseLecturer: Lectur
- grade: Grade
- courseName: String
- courseCode: String
- prequisite: String
- credits: long
- year: long
- isPrerequisitesComple
- isCompleted: boolean
- isApproved: boolean
+ isPrerequisitesComple
+ accessors()/mutators()

Course

er

ted: boolean

eted(course: String): boolean

1..*

0..*

Grade

- outOfHundred: double
- gano: double
- letterGrade: String

+ Grade()
+ Grade(outOfHunderd: double, gano: double, letterGrade: String)
+ convertHunderdToGano(hunderdNot: double): double
+ convertHunderdToLetterGrade(hunderd: double): void
+ accessors()/mutators()

Draft

- courses: ArrayList<Course>
- student: Student

+ hasCourse(course: Course): boolean
+ hasCourse(courseCode: String): boolean
+ removeCourse(courseCode: String): boolean
+ isEmpty(): boolean
+ getNumberOfClasses(): int
+ clearDraft(): void
+ addCourse(course: Course): void
+ accessors()/mutators()

Lecturer

- profession: String
- courseInstances: ArrayList<CourseInstance>

+ Lecturer()
+ Lecturer(profession: String)
+ findAllCourseInstances(courses: ArrayList<Course>): void
+ accessors()/mutators()
+ getInfo(): String

0..1

Extends

1

Advisor

- drafts: ArrayList<Draft>
- students: ArrayList<Student>

+ addDraft(draft: Draft): void
+ addstudent(student: Student): void
+ processDrafts(): void
+ accessors()/mutators()
+ getInfo(): String

StudentAffairsStaff

- workingField: String

+ StudentAffairsStaff()
+ StudentAffarisStaff(workingField: String)
+ studentUpdate(updateStudent Student,studentID String
+ accessors()/mutators()
+ getInfo(): String