
DatabaseManager
-objectMapper : ObjectMapper
-instance : DatabaseManager
-courseList: List<Course>
-studentList: List<Student>
-advisorList: List<Advisor>
-constraints: Constraint
-adminList :
+getInstance(): DatabaseManager
-initialize(): void
+saveToDatabase(): void
+fetchAdvisedStudents(Advisor advisor): List<Students>
+getConstraints(): HashMap<Integer,String>
+editConstraint(editedAttributes: HashMap<Integer, String>): boolean
+findCourseByCourseCode(courseCode: String): Course
-jsonToStudentList(jsonString: String): List<Student>
-jsonToAdvisorList(jsonString: String): List<Advisor>
-jsonToCourseList(jsonString: String): List<Course>
-getJsonStringList(list: List<T>): <T> String
-jsonToAdminList(jsonString: String): List<Admin>
-readFile(relativePath: String): String
-saveTranscriptsToDatabase(): void
-jsonToConstraints(jsonString:String):List<Constraint>
-writeFile(relativePath: String, jsonString: String): void
Util
+validateNumber(str: String, list: Object[]): boolean
+getRowNumberFromInput(str: String): String
+isValidNumber(str: String): boolean
+checkIfValidRowNumber(str: String, list: Object[]): boolean
+paintText(message: String, color: String): void
+paintTextwithmessage(String, color: String): void
+clearScreen():void
+sendFeedback(message: String, color: String): void
+sendFeedback(message: String): void
+isInputFormatTrueForTimeslist: ArrayList<String>): boolean
+isInputFormatTrueForDaylist: ArrayList<String>): boolean
+makeArrayList(splitPattern: String, textToSplit: String): ArrayList<String>
+animationTimer(duration: int): void
+printTimeTable(timeTable: String[][])
-divideIntoCourses(str: String): List<String>
<<interface>>
Showable
getMyPage(): void
<abstract>
User
-userId: String
-password: String
-firstName: String
-lastName: String
-status: boolean
-notifications: List<String>
+ logout(): void
+ checkCredentials(userId : String, password : String) : boolean
+addNotification(String notification): void
+clearNotifications(): void
Advisor
+getMyPage(): void
+acceptCourse(student: Student, selectedCourse: SelectedCourse) :void
+rejectCourse(student: Student, selectedCourse: SelectedCourse) : void
+fetchAdvisedStudent(): List<Student>
+setNotificationToStudent(student:Student, notification:String):void
+finalizeRegistration(student: Student): void
Admin
+getMyPage(): void
+fetchCourseList(): List<Course>
+fetchConstraint(): HashMap<Integer, String>
+createCourse(course: Course): Course
+editConstraints(editedAttributes: HashMap<Integer, String>): boolean
+deleteCourse(course: Course): boolean
+findCourseByCourseCode(courseCode: String): Course
+setNotificationToStudentsAndAdvisor(student: Student, advisor: Advisor, notification: String)
Student
-email: String
-identityNumber: String
-currentSemester: Int
-selectedCourses: List <SelectedCourse>
-advisorOfStudent: Advisor
-approvalStatus: ApprovalStatus
-transcript: Transcript
+getMyPage() : void
+listAvailableCourseSections(): List<CourseSection>
-findRepeatCourseSections():List<CourseSection>
-checkIfItExistsInSelectedCourses(course: Course):boolean
+addNewCourse(selectedCourse : SelectedCourse) : boolean
+deleteCourse(selectedCourse : SelectedCourse) : boolean
+checkConflictCourses(): List<CourseSection>
+fillTable(): String[][]
+fillTableWithValues(time: String, i: int, day: int, timeTable: String[][])
+sendSelectedCoursesToApproval(): void
-checkIfItWasTaken(course Course): boolean
-checkCourseType(course: Course): boolean
-checkCompulsoryCourses(): boolean
+fetchSelectedCoursesForAdvisor(): List<SelectedCourse>
<<enumeration>>
CourseStatus
DRAFT
PENDING
APPROVED
DENIED
ACTIVE
APPROVED_FINALIZED
DENIED_FINALIZED
<<enumeration>>
CourseResult
PASSED
FAILED
ACTIVE
<<enumeration>>
CourseType
COMPULSORY
NONTECHNICAL_ELECTIVE
TECHNICAL_ELECTIVE
UNIVERSITY_ELECTIVE
FACULTY_ELECTIVE
<<enumeration>>
ApprovalStatus
DONE
PENDING
FINALIZED_REGISTRATION
<<interface>>
Color
RED
GREEN
YELLOW
BLUE
MAGENTA
DEFAULT
CLIAdmin
-adminController: AdminController
+menuPage(): void
-constraintPage():void
-courseListPage():void
-createNewCoursePage():boolean
-findCourseByCourseCode(courseCode: String): Course
CLILogin
-user:User
+loginPage(): User
CLIAdvisor
-advisorController:AdvisorController
+menuPage(): void
-showStudentPage(): boolean
-coursesOfStudentPage(student: Student): boolean
CLIStudent
-studentController: StudentController
-shouldQuit: boolean
+menuPage(): void
-showMyCoursesPage(): void
-showAddCoursePage(): void
-showTimeTablePage(): void
-addCourse(str: String): boolean
-deleteCourse(str: String): boolean
-listSelectedCourses( ): void
-listAvailableCourseSections( ): void
AuthenticateService
+authenticateUser(userID: String, password): String): User
Constraint
-maxNumberOfCoursesStudentTake: int
-isAddDropWeek: boolean
-minRequiredECTSForTermProject: int
-isRegistrationWeek: boolean
+dictToObject(dict): Constraint
+editContrainit(editedAttributes: HashMap<Integer, String>): void
+fetchAttributes(): HashMap<Integer, String>
UserController
-currentUser:User
+loginUser(username: String, password: String): User
StudentController
-currentStudent: Student
+addSelectedCourse(selectedCourse : SelectedCourse): boolean
+removeSelectedCourse(selectedCourse : SelectedCourse): boolean
+sendSelectedCoursesToApproval(): void
+getNotifications(): List<String>
+getTimeTable(): String[][]
+logout():void
+clearNotifications(): void
+getAvailableCourseSections(): List<CourseSection>
+getSelectedCourses():List<SelectedCourse>
+getApprovalStatus(): ApprovalStatus
AdvisorController
-currentAdvisor: Advisor
-studentList: List <Student>
+approveCourse(student: Student, selectedCourse: SelectedCourse): void
+denyCourse(student: Student, selectedCourse: SelectedCourse): void
+getStudentListOrderByStatus(): List<student>
+getNotifications(): List<String>
+logout(): void
+clearNotifications(): void
+finalizeRegistration(student: Student): void
AdminController
-currentAdmin: Admin
+getCourseList(): List<Course>
+getConstraints(): HashMap<Integer, String>
+createCourse(course: Course): Course
+editConstraint(editedAttributes: HashMap<Integer, String>): void
+deleteCourse(course: Course): boolean
+findCourseByCourseCode(courseCode: String): Course
+logout(): void
+setNotificationToStudentAndAdvisor(student: Student, advisor: Advisor, notification: String)
CourseGrade
-course:Course
-letterGrade:String
-courseResult:CourseResult
+convertLetterGradeToScore():double
SelectedCourse
-course: Course
-status: CourseStatus
-courseSection: CourseSection
CourseSection
-studentCapacity: int
-lecturerName: String
-sectionTime: ArrayList<String>
-sectionDay: ArrayList<String>
-classroom: String
-sectionCode: String
-studentCountInsideCourseSection: int
-courseSections: List<CourseSection>
+ checkAvailibility(): boolean
+ incrementStudentCount(): int
+ decrementStudentCount(): int
+findCourseOfCourseSection(): Course
Course
-courseCredit: int
-courseECTS: int
-givenSemester: int
-courseName: String
-courseCode: String
-prerequisiteInformation: Prerequisite
-courseSections: List<CourseSection>
-courseType: CourseType
+ checkPrerequisite(student : Student): boolean
+ acquireAvailableSections() : List<CourseSection>
+ equals(course Course): boolean
Prerequisite
-prerequisiteOfCourses : List <Course>
+ checkPrerequisiteCoursePassed(student : Student, course : Course) : boolean
-isContainTrue(list: ArrayList<Boolean>): boolean
Transcript
-takenCourses: List <CourseGrade>
+addTakenCourse(course: Course): void
+acquirePassedCourses(): List<Course>
+calculateGPA() : double
+calculateCompletedCredits() : double
+checkEngineeringProjectAvailability():boolean
CourseRegistrationSystem
+start(): void