

-students

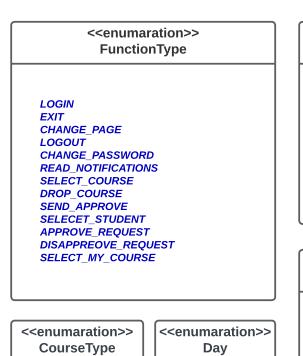
-fileName

-studentsFile

+ setPageType(self , pageType)

+ getInput(self)

+ setInput(self , input)



TUESDAY

WEDNESDAY

THURSDAY

FRİDAY

MANDATORY

TECHNICAL

FACULTY

- __init__(self, fileName , studentsFile , courses , advisors)

+ setStudentCourses (self , studentCoursesAr , courses)

F courseExists(self , course , transcriptCourseList)

calculateGPA(self, passedCourses, failedCourses)

+ createStudents (self , courses , advisors)

findAdvisor (self, advisorID, advisors)

fillStudentListCourse(self, courses)

+ getCourseGrade(self , strGrade)

F letterToGrade(self, grade)

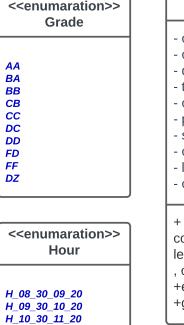
getStudents(self)

assingStudentsToAdvisor(self, advisors)

NONTECHNICAL

CreateStudent

+ setTranscriptCourses (self , transcriptCourses , grades , termPassed, courses)



H_11_30_12_20 H_13_00_13_50

H_14_00_14_50

H_15_00_15_50

H_16_00_16_50

H_17_00_17_50

H_18_00_18_50

H_19_00_19_50

H_20_00_20_50

H_21_00_21_50

courseld courseName quota courseType prerequisiteCourses studentList courseSchedules lecturer credit + __init__(self , courseId, courseName , quota , term, lecturer, courseSchedules , credit, courseType) +enrollStudent(self , student) +getter setter methods CourseSession sessionId - __init__(self , courseID, courseName ,quota , term, lecturer, sessionId,courseSchedules,credit,

courseType)

- courseDay

- courseHours

courseHours)

- course: Course

grade: Grade

term: int

+ __init__(self , courseDay,

+ getter setter methods

+getter setter methods

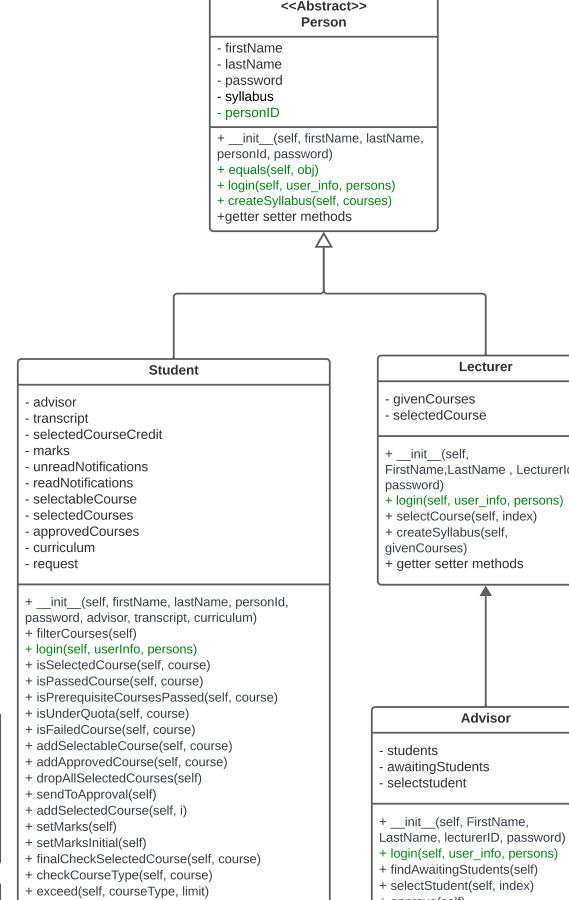
CourseSchedule

GradeClass

+ __init__(self, course, grade)

getter setter methods

Course



+ exceedTerm(self, courseType)

+ addUnreadNotification(self, notification)

+ clearUnreadNotification(self)

addAllSessions(self, course)

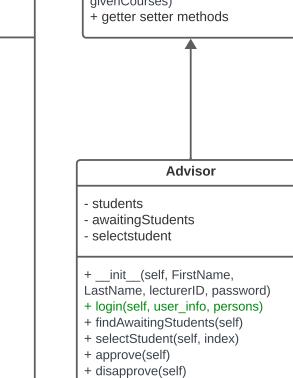
+ createSyllabus(self, courses)

getter setter methods

removeAllSessions(self, course,

+ dropCourse(self, i)

selectableCourses)



+ removeAwaitingStudent(self,

+ getter setter methods

student)

- __init__(self, password) + compareCurrentPassword(self, enteredPassword) + checkPasswordCond(self, newPassword) + isContainsSpecialChar(self, newPassword) + lengthCond(self, newPassword) + checkUpperCaseCond(self, newPassword) + checkLowerCaseCond(self, newPassword) + checkNumberCaseCond(self, newPassword) + equals(self, obj) + getter setter methods Id + __init__(self, id) getter setter methods Transcript FirstName,LastName, LecturerId,

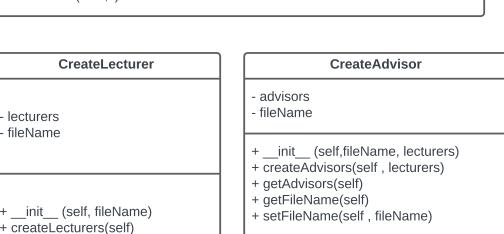
- password

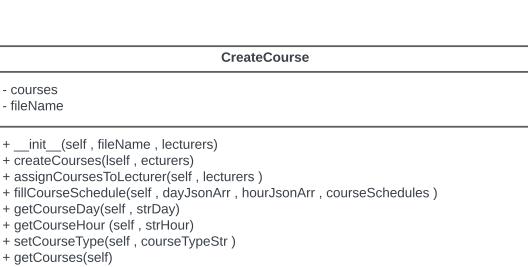
Password

- GPA _100 totalCredit passedCourses failedCourses + __init__(self, GPA_100, term, passedCourses, failedCourses) + calculateTotalCredit(self) + getter setter methods

Syllabus - syllabus + __init__(self) + returnIndexDay(self, day) + returnIndexHour(self,hour) addCourseToSyllabus(self, course) + checkConflict(self, course) + findConflictedCourseName(self, + fillSyllabus(self, selectedCourses) + removeCourseFromSyllabus(self, + isEmpty(self, rowIndex, columnIndex) getter setter methods

CreatePage - pages + createPages(self , currentUser) + createMainMenuPageStudentContent (self , numberOfUnreadNotifications) + createMainMenuPageAdvisorContent(self) + createMainMenuPageLecturerContent(self) + createSelectedStudentsRequestPageContent(self , student) + createEvaluateRequestPageContent(self , student) + createMyStudentsPageContent(self , student) + createSelectableCoursesPageContent(self , courses) + createSelectedCoursesPageContent(self , courses) + createAllCoursesPageContent (self , courses) + createApprovedCoursesPageContent (self , courses) + createProfilePageContent(self , user) + createChangePasswordPage(self) + createMyCoursesPageContent(self , lecturer) + createSelectedMyCoursePage(self , course) + createSyllabusPageContent(self , syllabus) + createMyNotificationsPageContent(self, unreadNotifications, readNotifications) + createTranscriptPageContent(self , student) + courseIds(self , courseTable) + courseListForContent(self, courses, type) + allCourseLabels(self , labelIndex) + returnHour(self , hourIndex) + blankAfterStr(self , str , space) + blankAfterI(self , i)





CreateCourse
courses fileName
init(self , fileName , lecturers) createCourses(lself , ecturers) assignCoursesToLecturer(self , lecturers) fillCourseSchedule(self , dayJsonArr , hourJsonArr , courseSchedules) getCourseDay(self , strDay) getCourseHour (self , strHour) setCourseType(self , courseTypeStr) getCourses(self) setCourses(self , courses)