

SystemClass
- domain: SystemDomain - currentUser : Person - userInterface : UserInterface
+ __init__(self, u) + run (self) + login(self , userInfo) + logout(self) + exit(self) + updateCourseJSON(self) + updateLecturerJSON(self) + updateAdvisorJSON(self) + updateStudentJSON(self) + studentToJsonArray(self , students) + transcriptCourses(self , transcriptCoursesList) + listenUserInterface(self , sm) + getCurrentUser(self) + setCurrentUser(self , currentUser)

SystemDomain
- lecturerCreator - advisorCreator - courseCreator - studentCreator - pageCreator
+ __init__(self) + getLecturerCreator(self) + getAdvisorCreator(self) + getCourseCreator(self) + getStudentCreator(self) + getPageCreator (self)
SystemMessage
- functionType - nextPageType - input
+ __init__(self , functionType, pageType, input) + getFunctionType(self) + setFunctionType(self , functionType) + getPageType(self) + setPageType(self , pageType) + getInput(self) + setInput(self , input)

<<enumeration>> FunctionType
LOGIN EXIT CHANGE_PAGE LOGOUT CHANGE_PASSWORD READ_NOTIFICATIONS SELECT_COURSE DROP_COURSE SEND_APPROVE SELECET_STUDENT APPROVE_REQUEST DISAPPROVE_REQUEST SELECT_MY_COURSE
<<enumeration>> CourseType
MANDATORY NONTECHNICAL TECHNICAL FACULTY
<<enumeration>> Day
MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY

<<enumeration>> Grade
AA BA BB CB CC DC DD FD FF DZ
<<enumeration>> Hour
H_08_30_09_20 H_09_30_10_20 H_10_30_11_20 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

Course
- courseId - courseName - quota - term - 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) +getter setter methods

<<Abstract>> Person
- firstName - lastName - password - syllabus - personID
+ __init__(self, firstName, lastName, personId, password) + equals(self, obj) + login(self, user_info, persons) + createSyllabus(self, courses) +getter setter methods
Student
- advisor - transcript - selectedCourseCredit - marks - unreadNotifications - readNotifications - selectableCourse - selectedCourses - approvedCourses - curriculum - request
+ __init__(self, firstName, lastName, personId, password, advisor, transcript, curriculum) + filterCourses(self) + login(self, userInfo, persons) + isSelectedCourse(self, course) + isPassedCourse(self, course) + isPrerequisiteCoursesPassed(self, course) + isUnderQuota(self, course) + isFailedCourse(self, course) + addSelectableCourse(self, course) + addApprovedCourse(self, course) + dropAllSelectedCourses(self) + sendToApproval(self) + addSelectedCourse(self, i) + setMarks(self) + setMarksInitial(self) + finalCheckSelectedCourse(self, course) + checkCourseType(self, course) + exceed(self, courseType, limit) + exceedTerm(self, courseType) + clearUnreadNotification(self) + addUnreadNotification(self, notification) + dropCourse(self, i) + addAllSessions(self, course) + removeAllSessions(self, course, selectableCourses) + createSyllabus(self, courses) + getter setter methods

Lecturer
- givenCourses - selectedCourse
+ __init__(self, FirstName,LastName , LecturerId, password) + login(self, user_info, persons) + selectCourse(self, index) + createSyllabus(self, givenCourses) + getter setter methods
Advisor
- students - awaitingStudents - selectstudent
+ __init__(self, FirstName, LastName, lecturerID, password) + login(self, user_info, persons) + findAwaitingStudents(self) + selectStudent(self, index) + approve(self) + disapprove(self) + sendNotification(self, message, type) + removeAwaitingStudent(self, student) + getter setter methods

Password
- password
+ __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
- id
+ __init__(self, id) + getter setter methods

Transcript
- GPA_100 - term - 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, course) + fillSyllabus(self, selectedCourses) + removeCourseFromSyllabus(self, course) + 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 ) + courseId(self , courseTable) + courseListForContent(self , courses, type) + allCourseLabels(self , labelIndex) + returnHour(self , hourIndex) + blankAfterStr(self , str , space) + blankAfterI(self , i)

CreateStudent
-students -fileName -studentsFile
+ __init__(self, fileName , studentsFile , courses , advisors ) + createStudents (self , courses , advisors) + setTranscriptCourses (self , transcriptCourses , grades , termPassed, courses) + setStudentCourses (self , studentCoursesAr , courses) + findAdvisor (self, advisorID , advisors ) + assingStudentsToAdvisor(self , advisors) + fillStudentListCourse(self, courses) + courseExists(self , course , transcriptCourseList ) + calculateGPA(self, passedCourses , failedCourses) + letterToGrade(self, grade) + getCourseGrade(self , strGrade ) + getStudents(self)

CreateCourse
- courses - fileName
+ __init__(self , fileName , lecturers) + createCourses(Iself , ecturers) + assignCoursesToLecturer(self , lecturers ) + fillCourseSchedule(self , dayJsonArr , hourJsonArr , courseSchedules ) + getCourseDay(self , strDay) + getCourseHour (self , strHour) + setCourseType(self , courseTypeStr ) + getCourses(self) + setCourses(self , courses)

CreateLecturer
- lecturers - fileName
+ __init__(self , fileName) + createLecturers(self)

CreateAdvisor
- advisors - fileName
+ __init__(self,fileName, lecturers) + createAdvisors(self , lecturers) + getAdvisors(self) + getFileName(self) + setFileName(self , fileName)