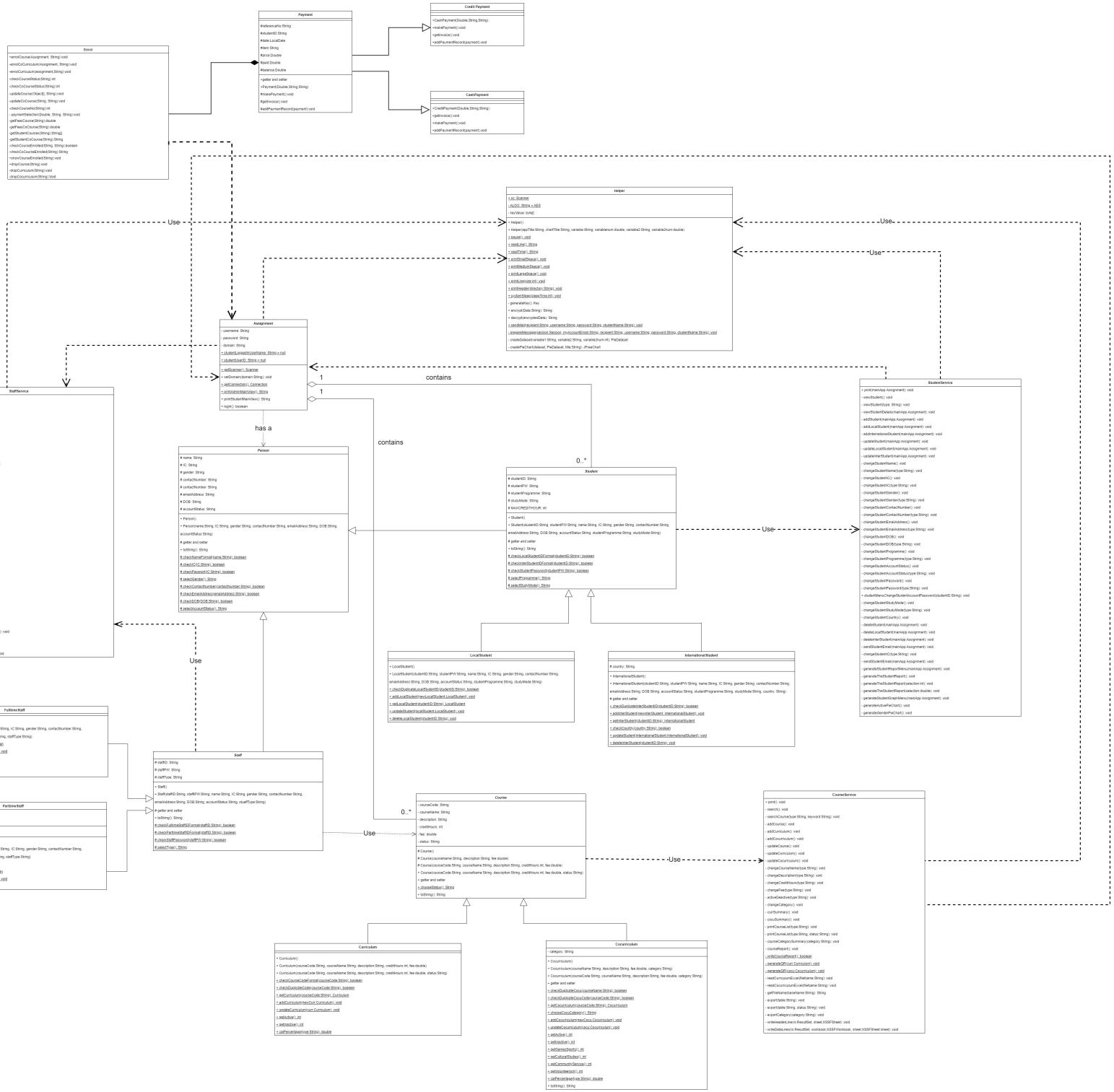
Payment
#referenceNo:String
#studentID:String
#date:LocalDate
#item:String
#price:Double
#paid:Double
#balance:Double
+getter and setter
+Payment(Double,String,String)
#makePayment():void
#getInvoice():void
#addPaymentRecord(payment):void
Credit Payment
+CashPayment(Double,String,String)
+makePayment():void
+getInvoice():void
+addPaymentRecord(payment):void
CashPayment
+CreditPayment(Double,String,String)
+getInvoice():void
+makePayment():void
+addPaymentRecord(payment):void
Enrol
+enrolCourse(Assignment, String):void
+enrolCoCurriculum(assignment, String):void
+enrolCurriculum(assignment,String):void
+checkCourseStatus(String):int
+checkCoCourseStatus(String):int
+updateCourse(Date[0], String):void
+updateCoCourse(String, String):void
+checkCourseNo(String):int
-paymentSelection(Double, String, String):void
-getFeesCoCourse(String):double
-getFeesCurriculum(String):double
-getStudentCoCourse(String):String
-checkCourseEnrolled(String, String):boolean
-checkCoCourseEnrolled(String):String
+withdrawCourseEnrolled(String):void
+dropCourse(String):void
-dropCurriculum(String):void
-dropCocurriculum(String):void
Helper
+ sc: Scanner
- ALGO: String = AES
- keyValue: byte[]
+ Helper()
+ Helper(appTitle:String, chartTitle:String, variable:String, variable1num:double, variable2:String, variable2num:double)
+ sleep(int):void
+ readLine():String
+ readTime():String
+ printSmallSpace():void
+ printMediumSpace():void
+ printLargeSpace():void
+ print(Line(int n):void
+ printHeader(title:String):void
+ printSelectionSpace(String[]):void
- generateKey():Key
+ encrypt(Data:String):String
+ decrypt(encryptedData):String
+ sendMail(recipient:String, username:String, password:String, studentName:String):void
+ prepareMessage(session:Session, myAccountEmail:String, recipient:String, username:String, password:String, studentName:String):void
- createDataset(variable1:String, variable2:String, variable2num:int):PieDataset
- createPieChart(dataset, PieDataset, title:String):PieChart
Use
Use
Use
Assignment
- username: String
- password: String
- domain: String
+ student_loggedIn(userName, String) = null
+ studentUserID: String = null
+ getScanner():Scanner
+ setDomain(domain:String):void
+ getConnection():Connection
+ printAdminMainView():String
+ printStudentMainView():String
+ login():boolean
1
contains
1
has a
StaffService
+ print(mainApp:Assignment):void
+ viewStaff():void
+ viewStaff(type:String):void
+ viewStaffDetails(mainApp:Assignment):void
+ addStaff(mainApp:Assignment):void
+ addFulltimeStaff(mainApp:Assignment):void
+ addPartimeStaff(mainApp:Assignment):void
+ updateStaff(mainApp:Assignment):void
+ updateFulltimeStaff(mainApp:Assignment):void
+ updatePartimeStaff(mainApp:Assignment):void
+ changeStaffName():void
+ changeStaffName(type:String):void
+ changeStaffID():void
+ change StaffCitype:String):void
+ changeStaffGender():void
+ changeStaffGender(type:String):void
+ changeStaffContactNumber():void
+ changeStaffContactNumber(type:String):void
+ changeStaffEmailAddress():void
+ changeStaffEmailAddress(type:String):void
+ changeStaffDOB():void
+ changeStaffDOB(type:String):void
+ changeStaffType():void
+ changeStaffType(type:String):void
+ changeStaffAccountStatus():void
+ changeStaffAccountStatus(type:String):void
+ changeStaffPassword():void
+ changeStaffPassword(type:String):void
+ deleteStaff(mainApp:Assignment):void
+ deleteFulltimeStaff(mainApp:Assignment):void
+ deletePartimeStaff(mainApp:Assignment):void
+ changeStaffCi(type:String):void
+ generateStaffReportMenu(mainApp:Assignment):void
+ generateTheStaffReport():void
+ generateTheStaffReport(selection:int):void
+ generateTheStudentReport(selection:double):void
Person
name: String
IC: String
gender: String
contactNumber: String
contactNumber: String
emailAddress: String
DOB: String
accountStatus: String
+ Person()
+ Person(name:String, IC:String, gender:String, contactNumber:String, emailAddress:String, DOB:String, accountStatus:String)
# getter and setter
+ toString():String
# checkNameFormat(name:String):boolean
# checkICIC:String):boolean
# checkPassportIC:String):boolean
# selectGender():String
# checkContactNumber(contactNumber:String):boolean
# checkEmailAddress(emailAddress:String):boolean
# checkDOB(DOB:String):boolean
# selectAccountStatus():String
contains
0..*
Student
studentID: String
studentPW: String
studentProgramme: String
studyMode: String
MAXCREDITHOUR: int
+ Student()
+ Student(studentID:String, studentPW:String, name:String, IC:String, gender:String, contactNumber:String, emailAddress:String, DOB:String, accountStatus:String, studentProgramme:String, studyMode:String)
# getter and setter
+ toString():String
# checkLocalStudentIDFormat(studentID:String):boolean
# checkInternationalStudentIDFormat(studentID:String):boolean
# checkStudentPassword(studentPW:String):boolean
# selectProgramme():String
# selectStudyMode():String
Use
StudentService
+ print(mainApp:Assignment):void
+ viewStudent():void
- viewStudent(type:String):void
- viewStudentDetails(mainApp:Assignment):void
+ addStudent(mainApp:Assignment):void
+ addLocalStudent(mainApp:Assignment):void
+ addInternationalStudent(mainApp:Assignment):void
+ updateStudent(mainApp:Assignment):void
+ updateLocalStudent(mainApp:Assignment):void
+ updateInterStudent(mainApp:Assignment):void
+ changeStudentName():void
+ changeStudentName(type:String):void
+ changeStudentID():void
+ changeStudentID(type:String):void
+ changeStudentGender():void
+ changeStudentGender(type:String):void
+ changeStudentContactNumber():void
+ changeStudentContactNumber(type:String):void
+ changeStudentEmailAddress():void
+ changeStudentEmailAddress(type:String):void
+ changeStudentDOB():void
+ changeStudentDOB(type:String):void
+ changeStudentProgramme():void
+ changeStudentProgramme(type:String):void
+ changeStudentAccountStatus():void
+ changeStudentAccountStatus(type:String):void
+ changeStudentPassword():void
+ changeStudentPassword(type:String):void
+ studentMenuChangeStudentAccountPassword(studentID:String):void
+ changeStudentStudyMode():void
+ changeStudentStudyMode(type:String):void
+ changeStudentCountry():void
+ deleteStudent(mainApp:Assignment):void
+ deleteLocalStudent(mainApp:Assignment):void
+ deleteInterStudent(mainApp:Assignment):void
+ sendStudentEmail(mainApp:Assignment):void
+ changeStudentCi(type:String):void
+ sendStudentEmail(mainApp:Assignment):void
+ generateStudentReportMenu(mainApp:Assignment):void
+ generateTheStudentReport():void
+ generateTheStudentReport(selection:int):void
+ generateTheStudentReport(selection:double):void
+ generateStudentGraphMenu(mainApp:Assignment):void
+ generateActivePieChart():void
+ generateGenderPieChart():void
LocalStudent
+ LocalStudent()
+ LocalStudent(studentID:String, studentPW:String, name:String, IC:String, gender:String, contactNumber:String, emailAddress:String, DOB:String, accountStatus:String, studentProgramme:String, StudyMode:String)
+ checkDuplicateLocalStudentID(studentID:String):boolean
+ addLocalStudent(mainLocalStudent:LocalStudent):void
+ getLocalStudent(studentID:String):LocalStudent
+ updateStudent(mainLocalStudent:LocalStudent):void
+ deleteLocalStudent(studentID:String):void
InternationalStudent
country: String
+ InternationalStudent()
+ InternationalStudent(studentID:String, studentPW:String, name:String, IC:String, gender:String, contactNumber:String, emailAddress:String, DOB:String, accountStatus:String, studentProgramme:String, StudyMode:String, country: String)
# getter and setter
+ checkDuplicateInterStudent(studentID:studentID:String):boolean
+ addInterStudent(newInterStudent:InternationalStudent):void
+ getInterStudent(studentID:String):InternationalStudent
+ checkCountry(country:String):boolean
+ updateStudent(mainInternationalStudent:InternationalStudent):void
+ deleteInterStudent(studentID:String):void
FulltimeStaff
+ FulltimeStaff()
+ FulltimeStaff(staffID:String, staffPW:String, name:String, IC:String, gender:String, contactNumber:String, emailAddress:String, DOB:String, accountStatus:String, staffProgramme:String)
+ checkDuplicateFulltimeStaff(staffID:String):boolean
+ addFulltimeStaff(newFulltimeStaff:FulltimeStaff):void
+ getFulltimeStaff(staffID:String):FulltimeStaff
+ updateStaff(fulltimeStaff:FulltimeStaff):void
+ deleteFulltimeStaff(staffID:String):void
Staff
staffID: String
staffPW: String
staffType: String
+ Staff()
+ Staff(staffID:String, staffPW:String, name:String, IC:String, gender:String, contactNumber:String, emailAddress:String, DOB:String, accountStatus:String, staffTypeStr:String)
# getter and setter
+ toString():String
# checkFulltimeStaffID(Format(staffID:String):boolean
# checkPartimeStaffID(Format(staffID:String):boolean
# checkStaffPassword(staffPW:String):boolean
# selectTitle():String
Use
PartimeStaff
+ updateStaff(partimeStaff:PartimeStaff):void
+ deletePartimeStaff(staffID:String):void
# type: String
+ PartimeStaff()
+ PartimeStaff(staffID:String, staffPW:String, name:String, IC:String, gender:String, contactNumber:String, emailAddress:String, DOB:String, accountStatus:String, staffType:String)
# getter and setter
+ checkDuplicatePartimeStaff(staffID:String):boolean
+ addPartimeStaff(newPartimeStaff:PartimeStaff):void
+ getPartimeStaff(staffID:String):PartimeStaff
Course
- courseCode: String
- courseName: String
- description: String
- creditHours: int
- fee: double
- status: String
+ Course()
# Course(courseName:String, description:String, fee:double)
# Course(courseCode:String, courseName:String, description:String, creditHours:int, fee:double)
+ Course(courseCode:String, courseName:String, description:String, creditHours:int, fee:double, status:String)
+ getter and setter
+ chooseStatus():String
+ toString():String
0..*
Use
Use
CourseService
+ print():void
- search():void
- search(Course(type:String, keyword:String):void
+ addCourse():void
- addCoCurriculum():void
- addCocurriculum():void
+ updateCourse():void
- updateCurriculum():void
- updateCocurriculum():void
- changeCourseName(type:String):void
- changeDescription(type:String):void
- changeCreditHours(type:String):void
- changeFee(type:String):void
- activeOrDeactive(type:String):void
- changeCategory():void
- courSummary():void
- cocurSummary():void
- printCourseList(type:String):void
- printCourseList(type:String, status:String):void
- courseCategorySummary(category:String):void
- courseReport():void
- writeCourseReport():boolean
- generateGPCurr(Curriculum):void
- generateGPCoco(Cocurriculum):void
- readCurriculumExcel(fileName:String):void
- readCocurriculumExcel(fileName:String):void
- getFileName(pageName:String):String
- exportTable(String):void
- exportTable(String, status:String):void
- exportCategory(category:String):void
- writeHeadersLine(rs:ResultSet, sheet:XSSFSheet):void
- writeDataLine(rs:ResultSet, workbook:XSSFWorkbook, sheet:XSSFSheet sheet):void
Curriculum
+ Curriculum()
+ Curriculum(courseCode:String, courseName:String, description:String, creditHours:int, fee:double)
+ Curriculum(courseCode:String, courseName:String, description:String, creditHours:int, fee:double, status:String)
+ checkCourseCodeFormat(courseCode:String):boolean
+ checkDuplicateCode(courseCode:String):boolean
+ addCurriculum(courseCode:String):Curriculum
+ addCurriculumNewCur:Curriculum):void
+ updateCurriculum(cur:Curriculum):void
+ setAcademic():int
+ setInterest():int
+ setPercentage(type:String):double
Cocurriculum
- category: String
+ Cocurriculum()
+ Cocurriculum(courseName:String, description:String, fee:double, category:String)
+ Cocurriculum(courseCode:String, courseName:String, description:String, fee:double, category:String)
+ getter and setter
+ checkDuplicateCocu(courseName:String):boolean
+ checkDuplicateCocuCode(courseCode:String):boolean
+ addCocurriculum(courseCode:String):Cocurriculum
+ addCocurriculumNewCoco:Cocurriculum):void
+ updateCocurriculum(coco:Cocurriculum):void
+ setAcademic():int
+ setInterest():int
+ setGameSports():int
+ setCulturalStudies():int
+ setCommunityService():int
+ setVolunteerism():int
+ calPercentage(type:String):double
+ toString():String