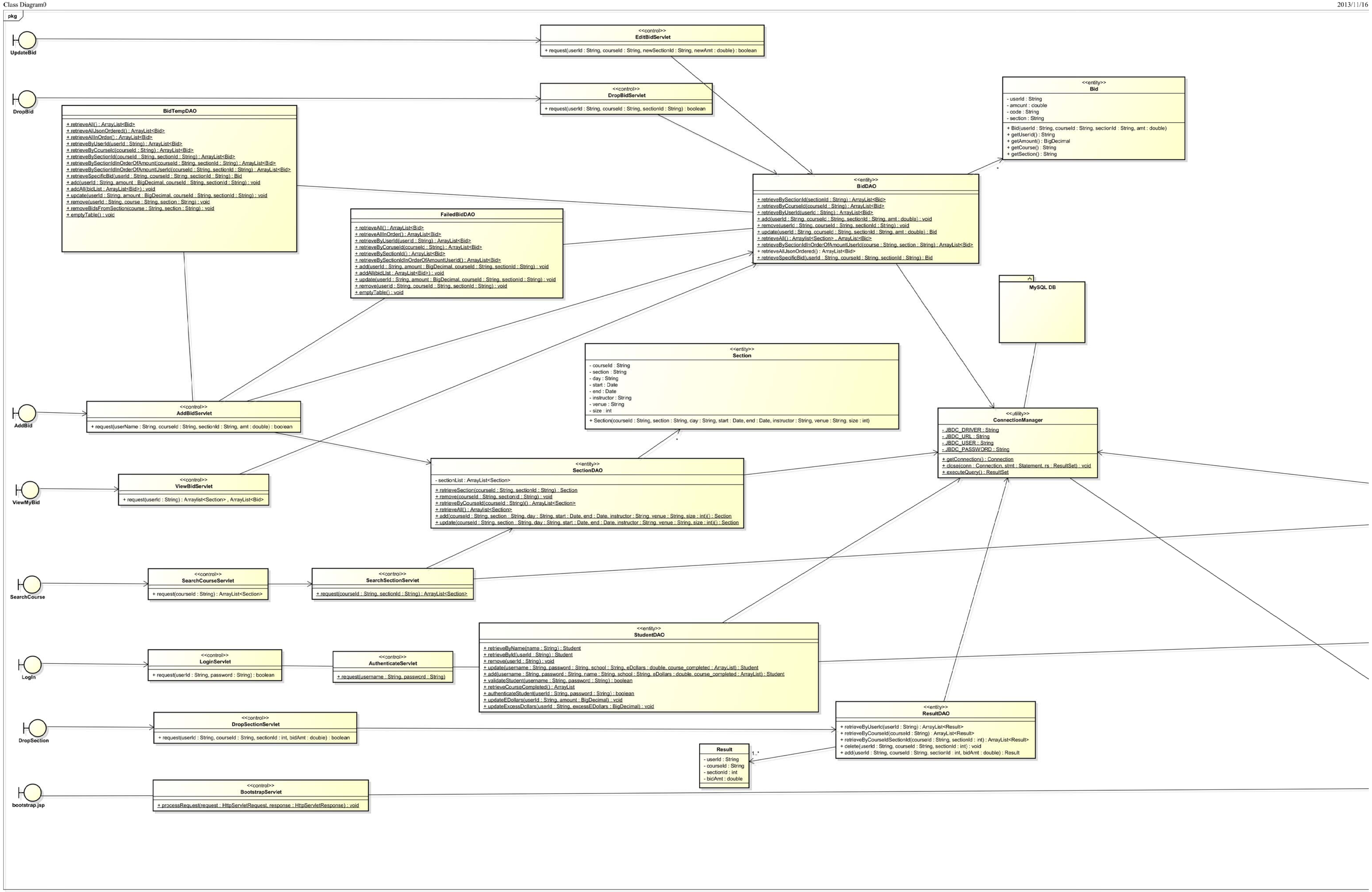
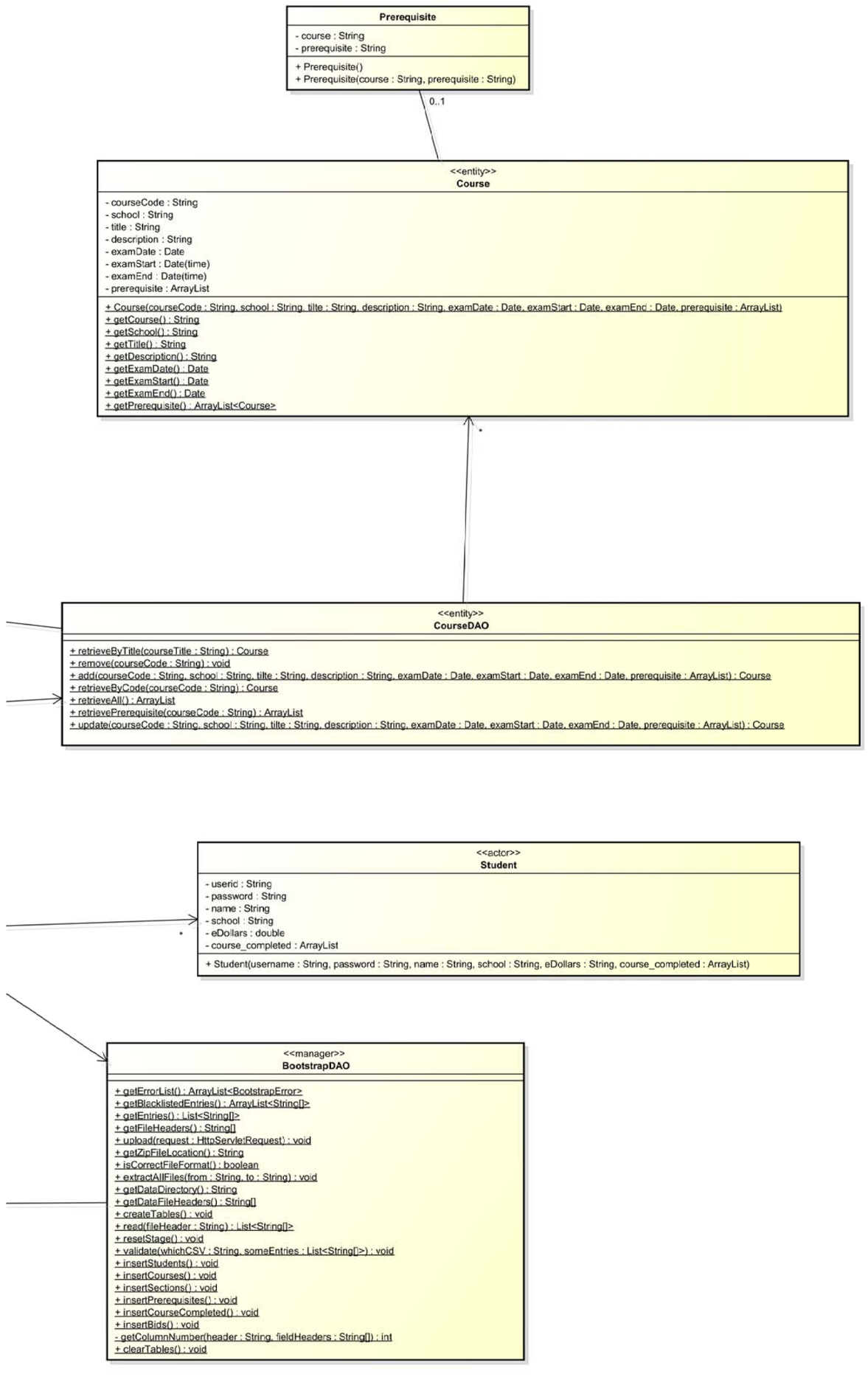

pkg
UpdateBid
<<control>>
EditBidServlet
+ request(userId : String, courseId : String, newSectionId : String, newAmt : double) : boolean
DropBid
<<control>>
DropBidServlet
+ request(userId : String, courseId : String, sectionId : String) : boolean
BidTempDAO
+ retrieveAll() : ArrayList<Bid>
+ retrieveAllJsonOrdered() : ArrayList<Bid>
+ retrieveAllInOrder() : ArrayList<Bid>
+ retrieveByUserId(userId : String) : ArrayList<Bid>
+ retrieveByCourseId(courseId : String) : ArrayList<Bid>
+ retrieveBySectionId(courseId : String, sectionId : String) : ArrayList<Bid>
+ retrieveBySectionIdInOrderOfAmount(courseId : String, sectionId : String) : ArrayList<Bid>
+ retrieveBySectionIdInOrderOfAmountUserId(courseId : String, sectionId : String) : ArrayList<Bid>
+ retrieveSpecificBid(userId : String, courseId : String, sectionId : String) : Bid
+ add(userId : String, amount : BigDecimal, courseId : String, sectionId : String) : void
+ addAll(bidList : ArrayList<Bid>) : void
+ update(userId : String, amount : BigDecimal, courseId : String, sectionId : String) : void
+ remove(userId : String, course : String, section : String) : void
+ removeBidsFromSection(course : String, section : String) : void
+ emptyTable() : void
<<entity>>
Bid
- userId : String
- amount : double
- code : String
- section : String
+ Bid(userId : String, courseId : String, sectionId : String, amt : double)
+ getUserId() : String
+ getAmount() : BigDecimal
+ getCourse() : String
+ getSection() : String
FailedBidDAO
+ retrieveAll() : ArrayList<Bid>
+ retrieveAllInOrder() : ArrayList<Bid>
+ retrieveByUserId(userId : String) : ArrayList<Bid>
+ retrieveByCourseId(courseId : String) : ArrayList<Bid>
+ retrieveBySectionId() : ArrayList<Bid>
+ retrieveBySectionIdInOrderOfAmountUserId() : ArrayList<Bid>
+ add(userId : String, amount : BigDecimal, courseId : String, sectionId : String) : void
+ addAll(bidList : ArrayList<Bid>) : void
+ update(userId : String, amount : BigDecimal, courseId : String, sectionId : String) : void
+ remove(userId : String, courseId : String, sectionId : String) : void
+ emptyTable() : void
<<entity>>
BidDAO
+ retrieveBySectionId(sectionId : String) : ArrayList<Bid>
+ retrieveByCourseId(courseId : String) : ArrayList<Bid>
+ retrieveByUserId(userId : String) : ArrayList<Bid>
+ add(userId : String, courseId : String, sectionId : String, amt : double) : void
+ remove(userId : String, courseId : String, sectionId : String) : void
+ update(userId : String, courseId : String, sectionId : String, amt : double) : Bid
+ retrieveAll() : ArrayList<Section>, ArrayList<Bid>
+ retrieveBySectionIdInOrderOfAmountUserId(course : String, section : String) : ArrayList<Bid>
+ retrieveAllJsonOrdered() : ArrayList<Bid>
+ retrieveSpecificBid(userId : String, courseId : String, sectionId : String) : Bid
MySQL DB
<<entity>>
Section
- courseId : String
- section : String
- day : String
- start : Date
- end : Date
- instructor : String
- venue : String
- size : int
+ Section(courseId : String, section : String, day : String, start : Date, end : Date, instructor : String, venue : String, size : int)
AddBid
<<control>>
AddBidServlet
+ request(userName : String, courseId : String, sectionId : String, amt : double) : boolean
<<utility>>
ConnectionManager
- JBDC_DRIVER : String
- JBDC_URL : String
- JBDC_USER : String
- JBDC_PASSWORD : String
+ getConnection() : Connection
+ close(conn : Connection, stmt : Statement, rs : ResultSet) : void
+ executeQuery() : ResultSet
ViewMyBid
<<control>>
ViewBidServlet
+ request(userId : String) : Arraylist<Section> , ArrayList<Bid>
<<entity>>
SectionDAO
- sectionList : ArrayList<Section>
+ retrieveSection(courseId : String, sectionId : String) : Section
+ remove(courseId : String, sectionId : String) : void
+ retrieveByCourseId(courseId : String)() : ArrayList<Section>
+ retrieveAll() : Arraylist<Section>
+ add(courseId : String, section : String, day : String, start : Date, end : Date, instructor : String, venue : String, size : int)() : Section
+ update(courseId : String, section : String, day : String, start : Date, end : Date, instructor : String, venue : String, size : int)() : Section
SearchCourse
<<control>>
SearchCourseServlet
+ request(courseId : String) : ArrayList<Section>
<<control>>
SearchSectionServlet
+ request(courseId : String, sectionId : String) : ArrayList<Section>
<<entity>>
StudentDAO
+ retrieveByName(name : String) : Student
+ retrieveById(userId : String) : Student
+ remove(userId : String) : void
+ update(username : String, password : String, school : String, eDollars : double, course_completed : ArrayList) : Student
+ add(username : String, password : String, name : String, school : String, eDollars : double, course_completed : ArrayList) : Student
+ validateStudent(username : String, password : String) : boolean
+ retrieveCourseCompleted() : ArrayList
+ authenticateStudent(userId : String, password : String) : boolean
+ updateEDollars(userId : String, amount : BigDecimal) : void
+ updateExcessDollars(userId : String, excessEDollars : BigDecimal) : void
LogIn
<<control>>
LoginServlet
+ request(userId : String, password : String) : boolean
<<control>>
AuthenticateServlet
+ request(username : String, password : String)
DropSection
<<control>>
DropSectionServlet
+ request(userId : String, courseId : String, sectionId : int, bidAmt : double) : boolean
<<entity>>
ResultDAO
+ retrieveByUserId(userId : String) : ArrayList<Result>
+ retrieveByCourseId(courseId : String) : ArrayList<Result>
+ retrieveByCourseIdSectionId(courseId : String, sectionId : int) : ArrayList<Result>
+ delete(userId : String, courseId : String, sectionId : int) : void
+ add(userId : String, courseId : String, sectionId : int, bidAmt : double) : Result
Result
- userId : String
- courseId : String
- sectionId : int
- bidAmt : double
1..*
bootstrap.jsp
<<control>>
BootstrapServlet
+ processRequest(request : HttpServletRequest, response : HttpServletResponse) : void


Prerequisite
- course : String
- prerequisite : String
+ Prerequisite()
+ Prerequisite(course : String, prerequisite : String)
0..1
<<entity>>
Course
- courseCode : String
- school : String
- title : String
- description : String
- examDate : Date
- examStart : Date(time)
- examEnd : Date(time)
- prerequisite : ArrayList
+ Course(courseCode : String, school : String, title : String, description : String, examDate : Date, examStart : Date, examEnd : Date, prerequisite : ArrayList)
+ getCourse() : String
+ getSchool() : String
+ getTitle() : String
+ getDescription() : String
+ getExamDate() : Date
+ getExamStart() : Date
+ getExamEnd() : Date
+ getPrerequisite() : ArrayList<Course>
*
<<entity>>
CourseDAO
+ retrieveByTitle(courseTitle : String) : Course
+ remove(courseCode : String) : void
+ add(courseCode : String, school : String, title : String, description : String, examDate : Date, examStart : Date, examEnd : Date, prerequisite : ArrayList) : Course
+ retrieveByCode(courseCode : String) : Course
+ retrieveAll() : ArrayList
+ retrievePrerequisite(courseCode : String) : ArrayList
+ update(courseCode : String, school : String, title : String, description : String, examDate : Date, examStart : Date, examEnd : Date, prerequisite : ArrayList) : Course
<<actor>>
Student
- userid : String
- password : String
- name : String
- school : String
- eDollars : double
- course_completed : ArrayList
+ Student(username : String, password : String, name : String, school : String, eDollars : String, course_completed : ArrayList)
*
<<manager>>
BootstrapDAO
+ getErrorList() : ArrayList<BootstrapError>
+ getBlacklistedEntries() : ArrayList<String[]>
+ getEntries() : List<String[]>
+ getFileHeaders() : String[]
+ upload(request : HttpServletRequest) : void
+ getZipFileLocation() : String
+ isCorrectFileFormat() : boolean
+ extractAllFiles(from : String, to : String) : void
+ getDataDirectory() : String
+ getDateFileHeaders() : String[]
+ createTables() : void
+ read(fileHeader : String) : List<String[]>
+ resetStage() : void
+ validate(whichCSV : String, somaEntries : List<String[]>) : void
+ insertStudents() : void
+ insertCourses() : void
+ insertSections() : void
+ insertPrerequisites() : void
+ insertCourseCompleted() : void
+ insertBids() : void
- getColumnNumber(header : String, fieldHeaders : String[]) : int
+ clearTables() : void