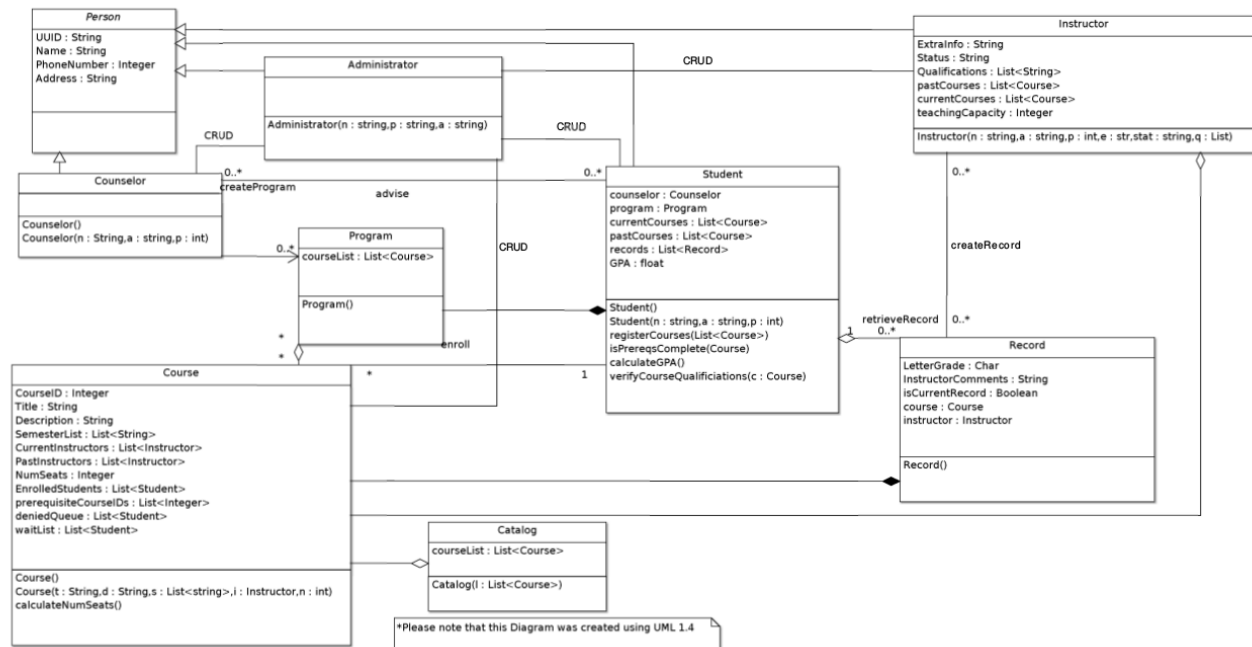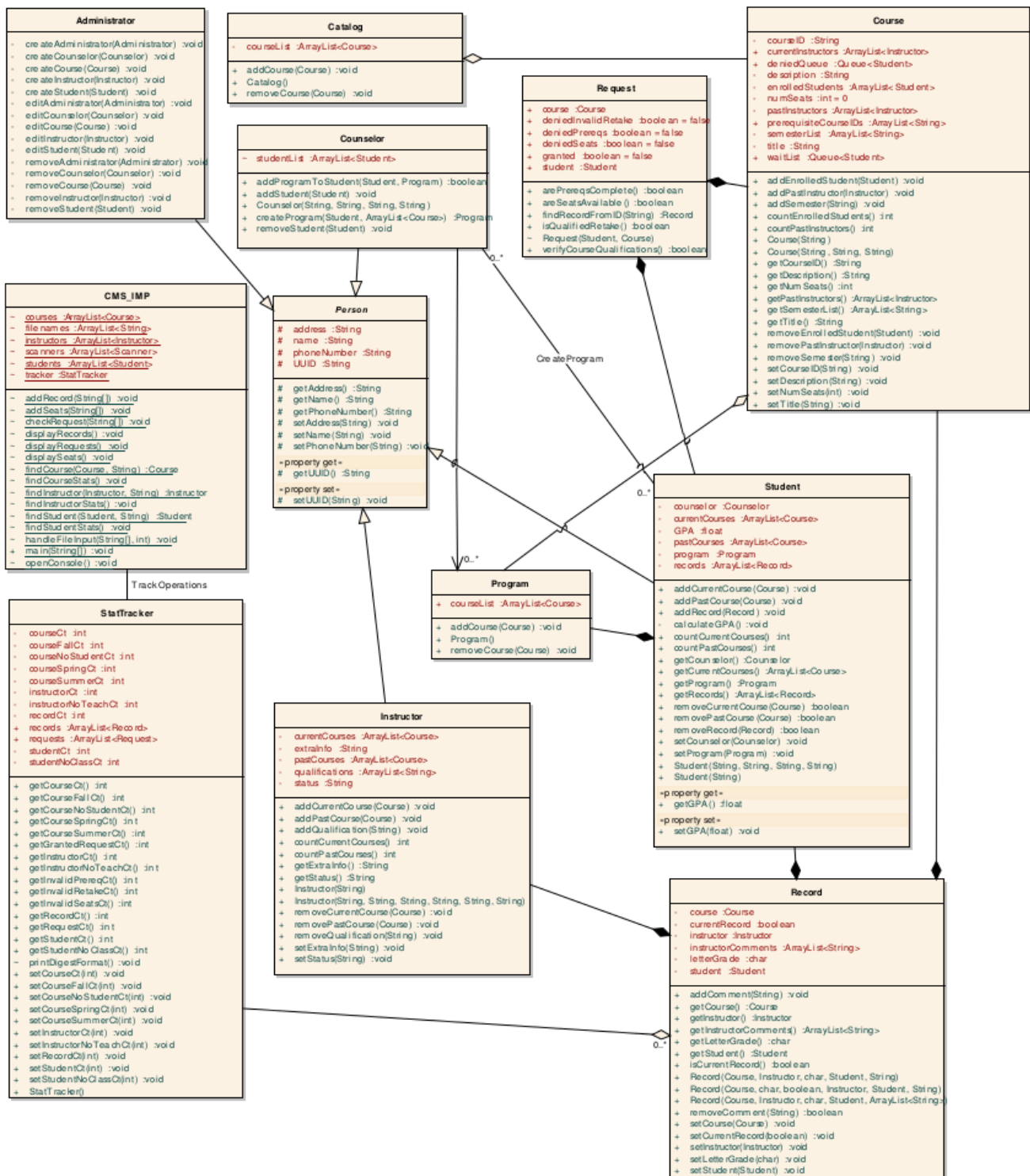Author: Matthew McDermott
Date: October 25,2016
Assignment: #6 - Course Management System – Next Design Phase (v3)

Previous UML class diagram from Assignment 5 (UML 1.4):

Updated UML class diagram (UML 2.0):

...next page

UML Class Diagram

Administrator
- createAdministrator(Administrator) :void
- createCounselor(Counselor) :void
- createCourse(Course) :void
- createInstructor(Instructor) :void
- createStudent(Student) :void
- editAdministrator(Administrator) :void
- editCounselor(Counselor) :void
- editCourse(Course) :void
- editInstructor(Instructor) :void
- editStudent(Student) :void
- removeAdministrator(Administrator) :void
- removeCounselor(Counselor) :void
- removeCourse(Course) :void
- removeInstructor(Instructor) :void
- removeStudent(Student) :void

Catalog
- courseList :ArrayList<Course>
- + addCourse(Course) :void
- + Catalog()
- + removeCourse(Course) :void

Course
- + courseID :String
- + currentInstructors :ArrayList<Instructor>
- + deniedQueue :Queue<Student>
- + description :String
- + enrolledStudents :ArrayList<Student>
- + numSeats :int = 0
- + pastInstructors :ArrayList<Instructor>
- + prerequisiteCourseIDs :ArrayList<String>
- + semesterList :ArrayList<String>
- + title :String
- + waitList :Queue<Student>
- + addEnrolledStudent(Student) :void
- + addPastInstructor(Instructor) :void
- + addSemester(String) :void
- + countEnrolledStudents() :int
- + countPastInstructors() :int
- + Course(String)
- + Course(String, String, String)
- + getCourseID() :String
- + getDescription() :String
- + getNumSeats() :int
- + getPastInstructors() :ArrayList<Instructor>
- + getSemesterList() :ArrayList<String>
- + getTitle() :String
- + removeEnrolledStudent(Student) :void
- + removePastInstructor(Instructor) :void
- + removeSemester(String) :void
- + setCourseID(String) :void
- + setDescription(String) :void
- + setNumSeats(int) :void
- + setTitle(String) :void

Request
- + course :Course
- + deniedInvalidRetake :boolean = false
- + deniedPrereqs :boolean = false
- + deniedSeats :boolean = false
- + granted :boolean = false
- + student :Student
- + arePrereqsComplete() :boolean
- + areSeatsAvailable () :boolean
- + findRecordFromID(String) :Record
- + isQualifiedRetake() :boolean
- - Request(String)
- - Request(Student, Course)
- + verifyCourseQualifications() :boolean

Counselor
- studentList :ArrayList<Student>
- + addProgramToStudent(Student, Program) :boolean
- + addStudent(Student) :void
- + Counselor(String, String, String, String)
- + createProgram(Student, ArrayList<Course>) :Program
- + removeStudent(Student) :void

CMS_IMP
- courses :ArrayList<Course>
- filenames :ArrayList<String>
- instructors :ArrayList<Instructor>
- scanners :ArrayList<Scanner>
- students :ArrayList<Student>
- tracker :StatTracker
- addRecord(String[]) :void
- addSeats(String[]) :void
- checkRequest(String[]) :void
- displayRecords() :void
- displayRequests() :void
- displaySeats() :void
- findCourse(Course, String) :Course
- findCourseStats() :void
- findInstructor(Instructor, String) :Instructor
- findInstructorStats() :void
- findStudent(Student, String) :Student
- findStudentStats() :void
- handleFileInput(String[], int) :void
- main(String[]) :void
- openConsole() :void

Person
- # address :String
- # name :String
- # phoneNumber :String
- # UUID :String
- # getAddress() :String
- # getName() :String
- # getPhoneNumber() :String
- # setAddress(String) :void
- # setName(String) :void
- # setPhoneNumber(String) :void
- «property get»
- # getUUID() :String
- «property set»
- # setUUID(String) :void

Program
- + courseList :ArrayList<Course>
- + addCourse(Course) :void
- + Program()
- + removeCourse(Course) :void

CreateProgram

TrackOperations

StatTracker
- courseCt :int
- courseFallCt :int
- courseNoStudentCt :int
- courseSpringCt :int
- courseSummerCt :int
- instructorCt :int
- instructorNoTeachCt :int
- recordCt :int
- + records :ArrayList<Record>
- + requests :ArrayList<Request>
- studentCt :int
- studentNoClassCt :int
- + getCourseCt() :int
- + getCourseFallCt() :int
- + getCourseNoStudentCt() :int
- + getCourseSpringCt() :int
- + getCourseSummerCt() :int
- + getGrantedRequestCt() :int
- + getInstructorCt() :int
- + getInstructorNoTeachCt() :int
- + getInvalidPrereqCt() :int
- + getInvalidRetakeCt() :int
- + getInvalidSeatsCt() :int
- + getRecordCt() :int
- + getRequestCt() :int
- + getStudentCt() :int
- + getStudentNoClassCt() :int
- - printDigestFormat() :void
- + setCourseCt(int) :void
- + setCourseFallCt(int) :void
- + setCourseNoStudentCt(int) :void
- + setCourseSpringCt(int) :void
- + setCourseSummerCt(int) :void
- + setInstructorCt(int) :void
- + setInstructorNoTeachCt(int) :void
- + setRecordCt(int) :void
- + setStudentCt(int) :void
- + setStudentNoClassCt(int) :void
- + StatTracker()

Instructor
- currentCourses :ArrayList<Course>
- extraInfo :String
- pastCourses :ArrayList<Course>
- qualifications :ArrayList<String>
- status :String
- + addCurrentCourse(Course) :void
- + addPastCourse(Course) :void
- + addQualification(String) :void
- + countCurrentCourses() :int
- + countPastCourses() :int
- + getExtraInfo() :String
- + getStatus() :String
- + Instructor(String)
- + Instructor(String, String, String, String, String, String)
- + removeCurrentCourse(Course) :void
- + removePastCourse(Course) :void
- + removeQualification(String) :void
- + setExtraInfo(String) :void
- + setStatus(String) :void

Student
- + counselor :Counselor
- + currentCourses :ArrayList<Course>
- + GPA :float
- + pastCourses :ArrayList<Course>
- + program :Program
- + records :ArrayList<Record>
- + addCurrentCourse(Course) :void
- + addPastCourse(Course) :void
- + addRecord(Record) :void
- + calculateGPA() :void
- + countCurrentCourses() :int
- + countPastCourses() :int
- + getCounselor() :Counselor
- + getCurrentCourses() :ArrayList<Course>
- + getProgram() :Program
- + getRecords() :ArrayList<Record>
- + removeCurrentCourse(Course) :boolean
- + removePastCourse(Course) :boolean
- + removeRecord(Record) :boolean
- + setCounselor(Counselor) :void
- + setProgram(Program) :void
- + Student(String, String, String, String)
- + Student(String)
- «property get»
- + getGPA() :float
- «property set»
- + setGPA(float) :void

Record
- + course :Course
- + currentRecord :boolean
- + instructor :Instructor
- + instructorComments :ArrayList<String>
- + letterGrade :char
- + student :Student
- + addComment(String) :void
- + getCourse() :Course
- + getInstructor() :Instructor
- + getInstructorComments() :ArrayList<String>
- + getLetterGrade() :char
- + getStudent() :Student
- + isCurrentRecord() :boolean
- + Record(Course, Instructor, char, Student, String)
- + Record(Course, char, boolean, Instructor, Student, String)
- + Record(Course, Instructor, char, Student, ArrayList<String>)
- + removeComment(String) :boolean
- + setCourse(Course) :void
- + setCurrentRecord(boolean) :void
- + setInstructor(Instructor) :void
- + setLetterGrade(char) :void
- + setStudent(Student) :void

Changes made:

1. Getters and setters were added to the diagram for every single class. This was required in order to more clearly depict all methods included in the program.

2. All attributes and methods were given the proper access modifier indications. This was required in order to properly designate which attributes and methods should be easily modified by outside classes.

3. All methods now provide a return type. This was provided for clarity an easier user readability.

4. The Request class was added to the diagram. This class is comprised of a Student and a Course and has methods to handle checking requirements for the provided Student to enroll in the provided Class before enrolling them.

5. The Record class now has associations directly relating to instructor, student, and course. The Student class no longer contains an arraylist of Record objects, and now the STatTracker actually contains that arraylist for more realistic interaction.

6. The StatTracker class was added in order to keep statistics for all records and requests. It contains all of the counters as well as arraylists for Records and Requests, which makes for easier manipulation and interaction from a user prospective.

7. The Administrator class was added to handle all CRUD functions in the system itself.