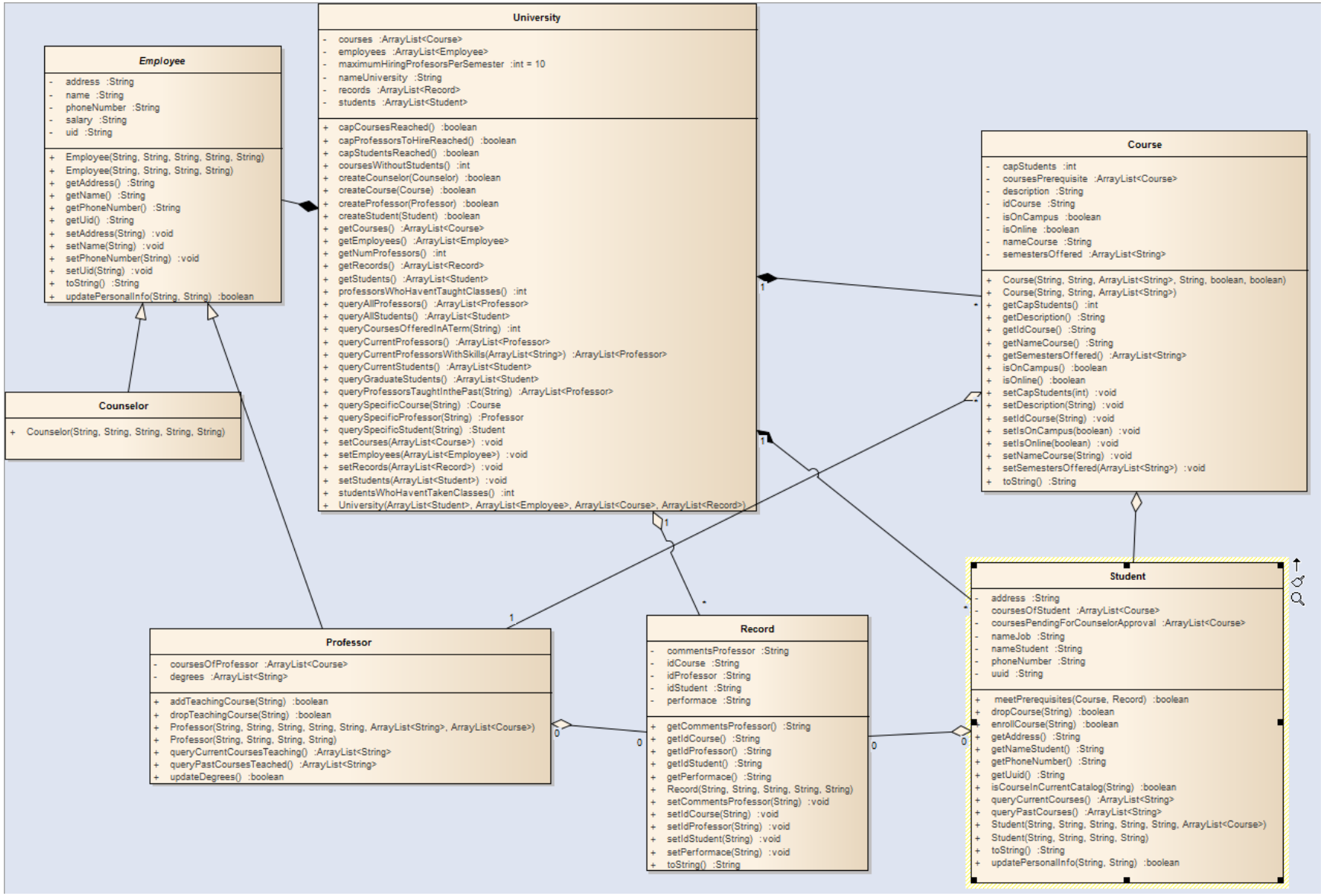
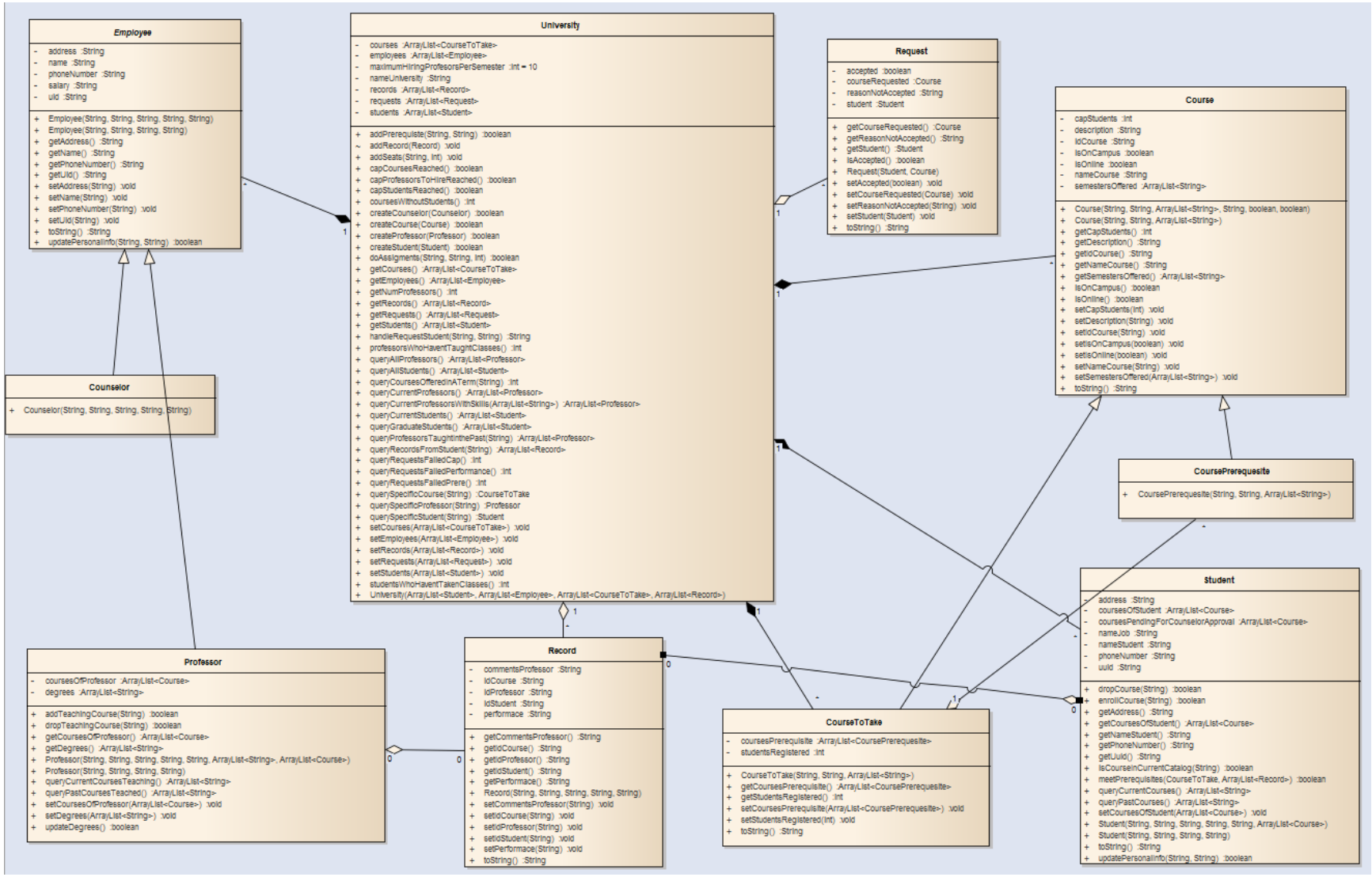
OLD UML:


University
- courses :ArrayList<Course>
- employees :ArrayList<Employee>
- maximumHiringProfesorsPerSemester :int = 10
- nameUniversity :String
- records :ArrayList<Record>
- students :ArrayList<Student>
+ capCoursesReached() :boolean
+ capProfessorsToHireReached() :boolean
+ capStudentsReached() :boolean
+ coursesWithoutStudents() :int
+ createCounselor(Counselor) :boolean
+ createCourse(Course) :boolean
+ createProfessor(Professor) :boolean
+ createStudent(Student) :boolean
+ getCourses() :ArrayList<Course>
+ getEmployees() :ArrayList<Employee>
+ getNumProfessors() :int
+ getRecords() :ArrayList<Record>
+ getStudents() :ArrayList<Student>
+ professorsWhoHaventTaughtClasses() :int
+ queryAllProfessors() :ArrayList<Professor>
+ queryAllStudents() :ArrayList<Student>
+ queryCoursesOfferedInATerm(String) :int
+ queryCurrentProfessors() :ArrayList<Professor>
+ queryCurrentProfessorsWithSkills(ArrayList<String>) :ArrayList<Professor>
+ queryCurrentStudents() :ArrayList<Student>
+ queryGraduateStudents() :ArrayList<Student>
+ queryProfessorsTaughtInthePast(String) :ArrayList<Professor>
+ querySpecificCourse(String) :Course
+ querySpecificProfessor(String) :Professor
+ querySpecificStudent(String) :Student
+ setCourses(ArrayList<Course>) :void
+ setEmployees(ArrayList<Employee>) :void
+ setRecords(ArrayList<Record>) :void
+ setStudents(ArrayList<Student>) :void
+ studentsWhoHaventTakenClasses() :int
+ University(ArrayList<Student>, ArrayList<Employee>, ArrayList<Course>, ArrayList<Record>)
Employee
- address :String
- name :String
- phoneNumber :String
- salary :String
- uid :String
+ Employee(String, String, String, String, String)
+ Employee(String, String, String, String)
+ getAddress() :String
+ getName() :String
+ getPhoneNumber() :String
+ getUid() :String
+ setAddress(String) :void
+ setName(String) :void
+ setPhoneNumber(String) :void
+ setUid(String) :void
+ toString() :String
+ updatePersonalInfo(String, String) :boolean
Course
- capStudents :int
- coursesPrerequisite :ArrayList<Course>
- description :String
- idCourse :String
- isOnCampus :boolean
- isOnline :boolean
- nameCourse :String
- semestersOffered :ArrayList<String>
+ Course(String, String, ArrayList<String>, String, boolean, boolean)
+ Course(String, String, ArrayList<String>)
+ getCapStudents() :int
+ getDescription() :String
+ getIdCourse() :String
+ getNameCourse() :String
+ getSemestersOffered() :ArrayList<String>
+ isOnCampus() :boolean
+ isOnline() :boolean
+ setCapStudents(int) :void
+ setDescription(String) :void
+ setIdCourse(String) :void
+ setIsOnCampus(boolean) :void
+ setIsOnline(boolean) :void
+ setNameCourse(String) :void
+ setSemestersOffered(ArrayList<String>) :void
+ toString() :String
Counselor
+ Counselor(String, String, String, String, String)
Professor
- coursesOfProfessor :ArrayList<Course>
- degrees :ArrayList<String>
+ addTeachingCourse(String) :boolean
+ dropTeachingCourse(String) :boolean
+ Professor(String, String, String, String, ArrayList<String>, ArrayList<Course>)
+ Professor(String, String, String, String)
+ queryCurrentCoursesTeaching() :ArrayList<String>
+ queryPastCoursesTeached() :ArrayList<String>
+ updateDegrees() :boolean
Record
- commentsProfessor :String
- idCourse :String
- idProfessor :String
- idStudent :String
- performace :String
+ getCommentsProfessor() :String
+ getIdCourse() :String
+ getIdProfessor() :String
+ getIdStudent() :String
+ getPerformace() :String
+ Record(String, String, String, String, String)
+ setCommentsProfessor(String) :void
+ setIdCourse(String) :void
+ setIdProfessor(String) :void
+ setIdStudent(String) :void
+ setPerformace(String) :void
+ toString() :String
Student
- address :String
- coursesOfStudent :ArrayList<Course>
- coursesPendingForCounselorApproval :ArrayList<Course>
- nameJob :String
- nameStudent :String
- phoneNumber :String
- uuid :String
+ meetPrerequisites(Course, Record) :boolean
+ dropCourse(String) :boolean
+ enrollCourse(String) :boolean
+ getAddress() :String
+ getNameStudent() :String
+ getPhoneNumber() :String
+ getUuid() :String
+ isCourseInCurrentCatalog(String) :boolean
+ queryCurrentCourses() :ArrayList<String>
+ queryPastCourses() :ArrayList<String>
+ Student(String, String, String, String, String, ArrayList<Course>)
+ Student(String, String, String, String)
+ toString() :String
+ updatePersonalInfo(String, String) :boolean
1
*
0

New UML:


Employee
- address :String
- name :String
- phoneNumber :String
- salary :String
- uid :String
+ Employee(String, String, String, String, String)
+ Employee(String, String, String, String)
+ getAddress() :String
+ getName() :String
+ getPhoneNumber() :String
+ getUid() :String
+ setAddress(String) :void
+ setName(String) :void
+ setPhoneNumber(String) :void
+ setUid(String) :void
+ toString() :String
+ updatePersonalInfo(String, String) :boolean
University
- courses :ArrayList<CourseToTake>
- employees :ArrayList<Employee>
- maximumHiringProfesorsPerSemester :int = 10
- nameUniversity :String
- records :ArrayList<Record>
- requests :ArrayList<Request>
- students :ArrayList<Student>
+ addPrerequisite(String, String) :boolean
~ addRecord(Record) :void
+ addSeats(String, int) :void
+ capCoursesReached() :boolean
+ capProfessorsToHireReached() :boolean
+ capStudentsReached() :boolean
+ coursesWithoutStudents() :int
+ createCounselor(Counselor) :boolean
+ createCourse(Course) :boolean
+ createProfessor(Professor) :boolean
+ createStudent(Student) :boolean
+ doAssignments(String, String, int) :boolean
+ getCourses() :ArrayList<CourseToTake>
+ getEmployees() :ArrayList<Employee>
+ getNumProfessors() :int
+ getRecords() :ArrayList<Record>
+ getRequests() :ArrayList<Request>
+ getStudents() :ArrayList<Student>
+ handleRequestStudent(String, String) :String
+ professorsWhoHaventTaughtClasses() :int
+ queryAllProfessors() :ArrayList<Professor>
+ queryAllStudents() :ArrayList<Student>
+ queryCoursesOfferedInATerm(String) :int
+ queryCurrentProfessors() :ArrayList<Professor>
+ queryCurrentProfessorsWithSkills(ArrayList<String>) :ArrayList<Professor>
+ queryCurrentStudents() :ArrayList<Student>
+ queryGraduateStudents() :ArrayList<Student>
+ queryProfessorsTaughtInThePast(String) :ArrayList<Professor>
+ queryRecordsFromStudent(String) :ArrayList<Record>
+ queryRequestsFailedCap() :int
+ queryRequestsFailedPerformance() :int
+ queryRequestsFailedPrere() :int
+ querySpecificCourse(String) :CourseToTake
+ querySpecificProfessor(String) :Professor
+ querySpecificStudent(String) :Student
+ setCourses(ArrayList<CourseToTake>) :void
+ setEmployees(ArrayList<Employee>) :void
+ setRecords(ArrayList<Record>) :void
+ setRequests(ArrayList<Request>) :void
+ setStudents(ArrayList<Student>) :void
+ studentsWhoHaventTakenClasses() :int
+ University(ArrayList<Student>, ArrayList<Employee>, ArrayList<CourseToTake>, ArrayList<Record>)
Request
- accepted :boolean
- courseRequested :Course
- reasonNotAccepted :String
- student :Student
+ getCourseRequested() :Course
+ getReasonNotAccepted() :String
+ getStudent() :Student
+ isAccepted() :boolean
+ Request(Student, Course)
+ setAccepted(boolean) :void
+ setCourseRequested(Course) :void
+ setReasonNotAccepted(String) :void
+ setStudent(Student) :void
+ toString() :String
Course
- capStudents :int
- description :String
- idCourse :String
- isOnCampus :boolean
- isOnline :boolean
- nameCourse :String
- semestersOffered :ArrayList<String>
+ Course(String, String, ArrayList<String>, String, boolean, boolean)
+ Course(String, String, ArrayList<String>)
+ getCapStudents() :int
+ getDescription() :String
+ getIdCourse() :String
+ getNameCourse() :String
+ getSemestersOffered() :ArrayList<String>
+ isOnCampus() :boolean
+ isOnline() :boolean
+ setCapStudents(int) :void
+ setDescription(String) :void
+ setIdCourse(String) :void
+ setIsOnCampus(boolean) :void
+ setIsOnline(boolean) :void
+ setNameCourse(String) :void
+ setSemestersOffered(ArrayList<String>) :void
+ toString() :String
Counselor
+ Counselor(String, String, String, String, String)
CoursePrerequesite
+ CoursePrerequesite(String, String, ArrayList<String>)
Professor
- coursesOfProfessor :ArrayList<Course>
- degrees :ArrayList<String>
+ addTeachingCourse(String) :boolean
+ dropTeachingCourse(String) :boolean
+ getCoursesOfProfessor() :ArrayList<Course>
+ getDegrees() :ArrayList<String>
+ Professor(String, String, String, String, String, ArrayList<String>, ArrayList<Course>)
+ Professor(String, String, String, String)
+ queryCurrentCoursesTeaching() :ArrayList<String>
+ queryPastCoursesTeached() :ArrayList<String>
+ setCoursesOfProfessor(ArrayList<Course>) :void
+ setDegrees(ArrayList<String>) :void
+ updateDegrees() :boolean
Record
- commentsProfessor :String
- idCourse :String
- idProfessor :String
- idStudent :String
- performace :String
+ getCommentsProfessor() :String
+ getIdCourse() :String
+ getIdProfessor() :String
+ getIdStudent() :String
+ getPerformace() :String
+ Record(String, String, String, String, String)
+ setCommentsProfessor(String) :void
+ setIdCourse(String) :void
+ setIdProfessor(String) :void
+ setIdStudent(String) :void
+ setPerformace(String) :void
+ toString() :String
CourseToTake
- coursesPrerequisite :ArrayList<CoursePrerequesite>
- studentsRegistered :int
+ CourseToTake(String, String, ArrayList<String>)
+ getCoursesPrerequisite() :ArrayList<CoursePrerequesite>
+ getStudentsRegistered() :int
+ setCoursesPrerequisite(ArrayList<CoursePrerequesite>) :void
+ setStudentsRegistered(int) :void
+ toString() :String
Student
- address :String
- coursesOfStudent :ArrayList<Course>
- coursesPendingForCounselorApproval :ArrayList<Course>
- nameJob :String
- nameStudent :String
- phoneNumber :String
- uuid :String
+ dropCourse(String) :boolean
+ enrollCourse(String) :boolean
+ getAddress() :String
+ getCoursesOfStudent() :ArrayList<Course>
+ getNameStudent() :String
+ getPhoneNumber() :String
+ getUuid() :String
+ isCourseInCurrentCatalog(String) :boolean
+ meetPrerequisites(CourseToTake, ArrayList<Record>) :boolean
+ queryCurrentCourses() :ArrayList<String>
+ queryPastCourses() :ArrayList<String>
+ setCoursesOfStudent(ArrayList<Course>) :void
+ Student(String, String, String, String, String, ArrayList<Course>)
+ Student(String, String, String, String)
+ toString() :String
+ updatePersonalInfo(String, String) :boolean

For this ASsigment , I had to break Course into 3 classes to be able to handle the prerequisites in an organized way, the list of prerequisite courses that used to be of type Course inside itself now is type CoursePrerequisite, which do not have a list of its own.

Also the need of keeping a list of requests made me create a request class.

Everything else stayed the same.

UML VERSION:2.0.