**Name: Zijie Yu**

**Database Management Systems (CSC434)**

**Final Project**

**1) - Define the information content of your database.**

**a)-Define a set of entities and appropriate attributes for each entity. Minimum 10 entities.**

Note: The first attributes are primary key, the Blue color entity means entity created automaticly by Visual Paradigm because of many to many relationship (intermediate table).

* **Students:** Student\_ID; First\_Name; Last\_Name; Phone; Year
* **Major**: Major\_Name; Department; DepartmentId
* **Students\_Major:** Students\_StudentId; Major\_MajorId
* **Course:** CourseId; Course\_Name; Seats Avaliable; Professor\_ProfessorId; SubjectId; ClassSceduleId;
* **Student\_Course:** Students\_StudentId; Course\_CourseId
* **Classroom:** ClassroomId; Name; Location; Capacity
* **Classroom\_Course:** Classroom\_ClassroomId; Course\_CourseId
* **Course\_Scedule**: CourseSceduleId; Meeting\_Day; Time\_Start; Time\_End
* **Subject:** SubjectId; Subject\_Name; DepartmentId
* Professors: ProfessorId; First\_Name; Last\_Name; Email; Phone
* **Subject\_Professors:** Subject\_SubjectId; Professors\_ProfessorId
* **Titles:** TitleId; Type
* **Titles:** Titles\_Titled; Professors\_ProfessorId
* **Manager:** ManagerId; First\_Name; Last\_Name; Phone
* **Advisor:** AdvisorId; First\_Name; Last\_Name; Phone
* **Department:** DepartmentId; Name; AdvisorId; ManagerId

**b)-Define a set of relationships that might exist between/among entities and attributes. Such relationships may include one-to-one, one-to-many and many-to-many associations.**

**Students – Course:** Many to many

(One student can take many course, and one course can have many students)

**Students – Major:** Many to many

(One student can have one or more majore, and one major can have many students.)

**Major – Department:** Many to one

( One major must belong to one department, but one department can have many major)

**Department – Manager:** One to one

(One department can only have one manager, and one manager can only manage one specific department)

**Department – Advisor:** One to one

(One department can only have one advisor, and one advisor can only manage one specific department)

**Department – Subject:** One to many

(One deparment can have many subjects, but one subject can only belong to one department.)

**Subject – Course:** One to many

(One subject can have many different course, but one course must have one subject)

**Course – Course\_Schedule:** Many to one

(One course\_Schedule can have many different course, but one course can only have on specific schedule.)

**Course – Classroom:** Many to many

(One course may have one or more classroom, and one classroom can have many different course.)

**Professor – titles:** Many to many

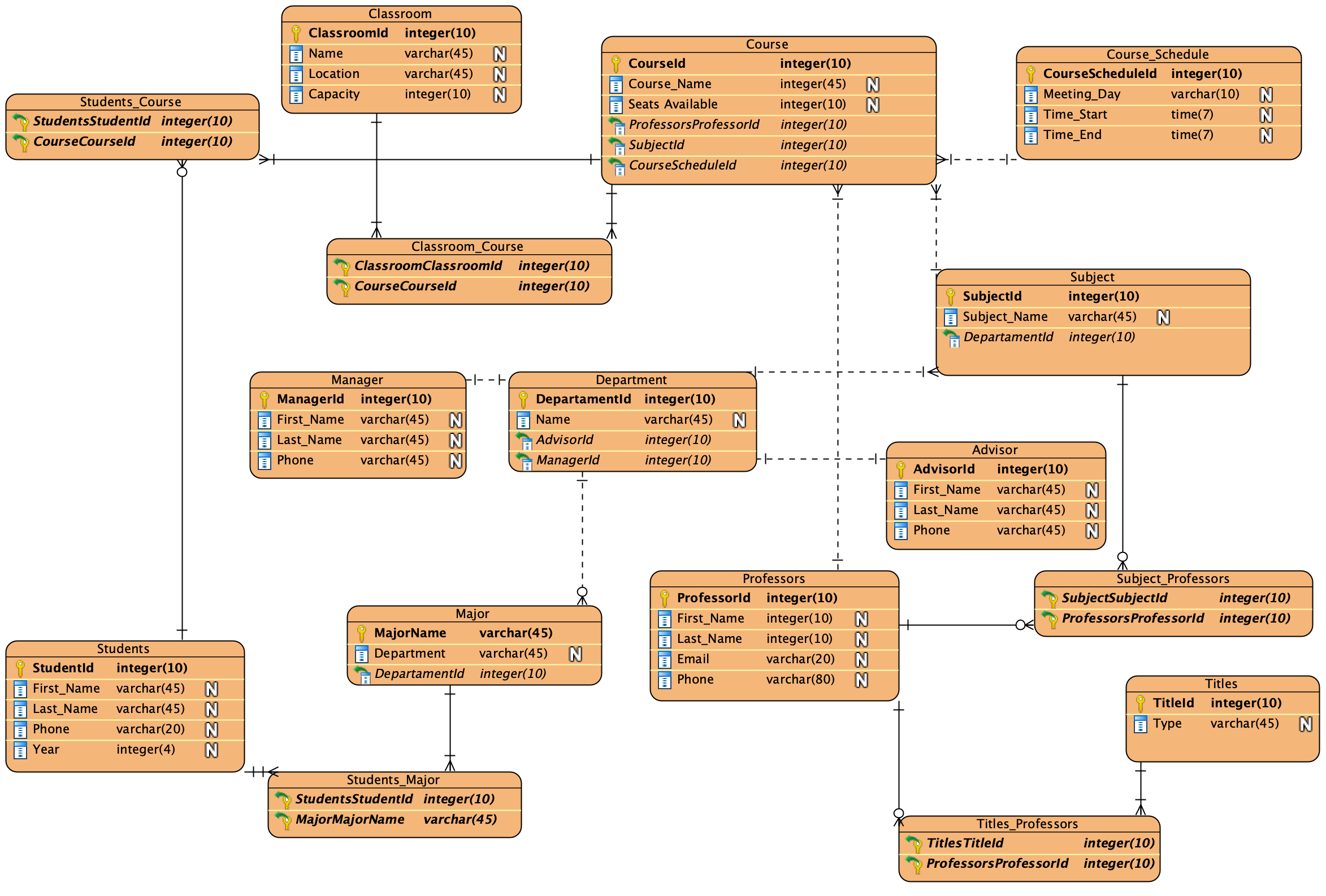
(One professor can have one or more titles, and one titles can have many different professor.)

**Professor – subject:** Many to many

(One professor can teach many different subject course, and one subject course can have many different professor.)

**c)-Define a set of constraints that may be imposed on data.**

**2) - Define an E-R Diagram for your database design.**

****

**3) - Define a relational schema for your database design.**