**ERD**

**Basic Entities :**

▪ **Educational Centers**

**-Center ID (p.k)**

**- CenterName**

**- Location**

**- CenterType**

**- Email**

**-Bio**

▪ **Course**

**- Course ID (p.k)**

**- CourseName**

**- CourseStatus**

**- Description**

▪ **CourseSchedule**

**- Day**

**- Hour**

**- Year**

**-Semester**

▪ **CourseMaterial**

**-Material ID (p.k)**

**- Title**

**-File Type**

**-Description**

▪ **Class Room**

**- ClassID (p.k)**

**- Capacity**

▪ **Student**

**-StudentID (p.k)**

**-StudentName**

**- Email**

**-{Phone}**

**-Date Of Birth**

**- Age()**

**Teacher** **▪**

**-TeacherID (p.k)**

**-TeacherName**

**-Email**

**-Salary**

**-Qulalificiation**

**- {Phone}**

**Admin** ▪

**-AdminID (p.k)**

**-AdminName**

**-Email**

**-Role**

▪ **UserAcount**

**-UserID (p.k)**

**-UserName**

**-Email**

**-{ Phone}**

▪ **Payment**

**-PaymentID(p.k)**

**-Amount**

**-Recived Date**

▪ **Notifications**

**-NotificationID (p.k)**

**-Message**

**-DateSent**

**IsRead-**

▪ **Feadback**

**-FeadbackID (p.k)**

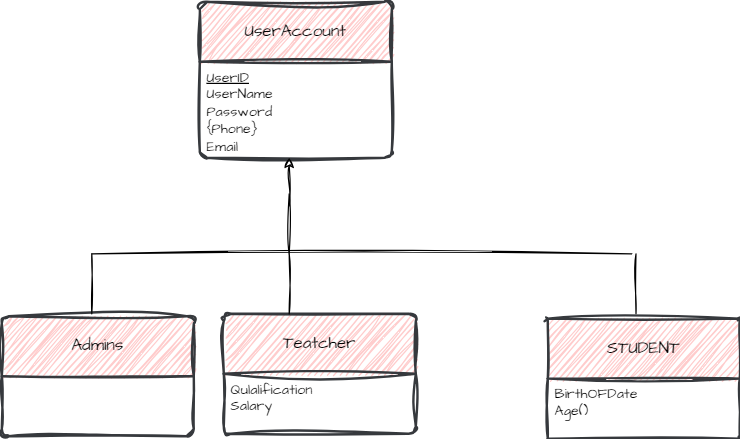
**- Message**

**Relationship**

**On our site, any person who owns an account must be either a teacher, a student, or an admin, so that each entity will have its own relationships in addition to the general relationships common between all these entities. On the other hand, there is a set of common attribute, so we used the concept of Specialization.**

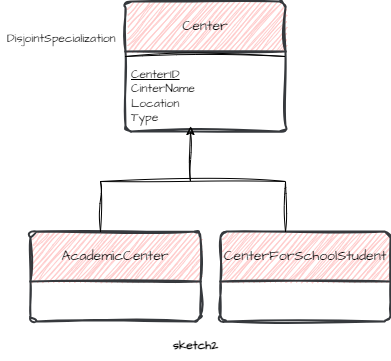
**Each user account belongs to only one subclass, so we will use the Disjoint Specialization.**

SuperClass

****

SubClass

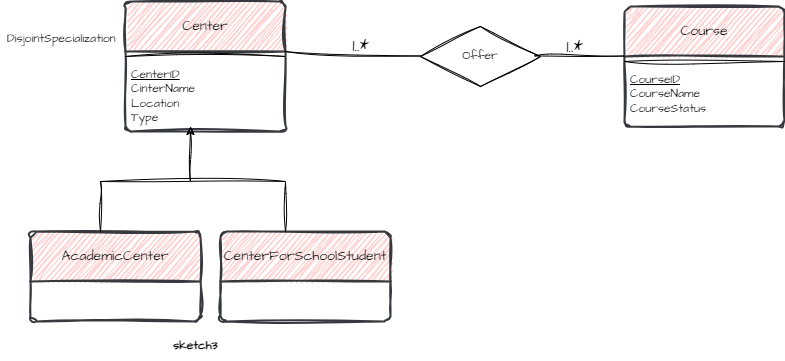
**The centers are divided into two types: Academic centers that offer courses in programming and other fields, and Center for school students ,so we decided to use the concept of Specialization To express these types , Since these center is either academic or for school students, we used the Disjoint Specialization (so we have Higher-Level entity set and Lower-Level entity set ).**

****

**Attribute and relationship inherited**

**The Super entity center has the attribute center-type so all center evaluated on the defining center-type if it Academic so its belong to the Acadimic Center, Otherwise its belong to center for school student.**

**The center offers many courses, and eatch course is offered by more than one center. Therefore, the relationship between the center and the course is :Many to Many.**

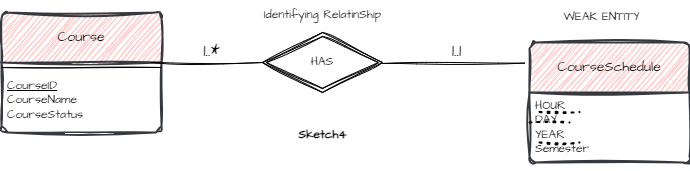
****

**Each center must offer courses, and each course must be linked to a center** **Total Participation .**

**CourseSchedule is a** **weak entity Because its existence depends on the existence of the course itself .**

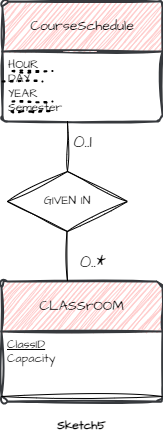
**Each course has more than one Appointment and eatch Appointment link with only one course so the relationship between the Course and CourseSchedule is: one to Many**

**Each Appointment is linked to a Course and each Course is linked to an Appointment Total** **Participation**

****

**The relationship between the** **CourseSchedule and Class Room is: Many to one.**

**Partial Participation on the part of CourseSchedule (online course) and Class Room .**

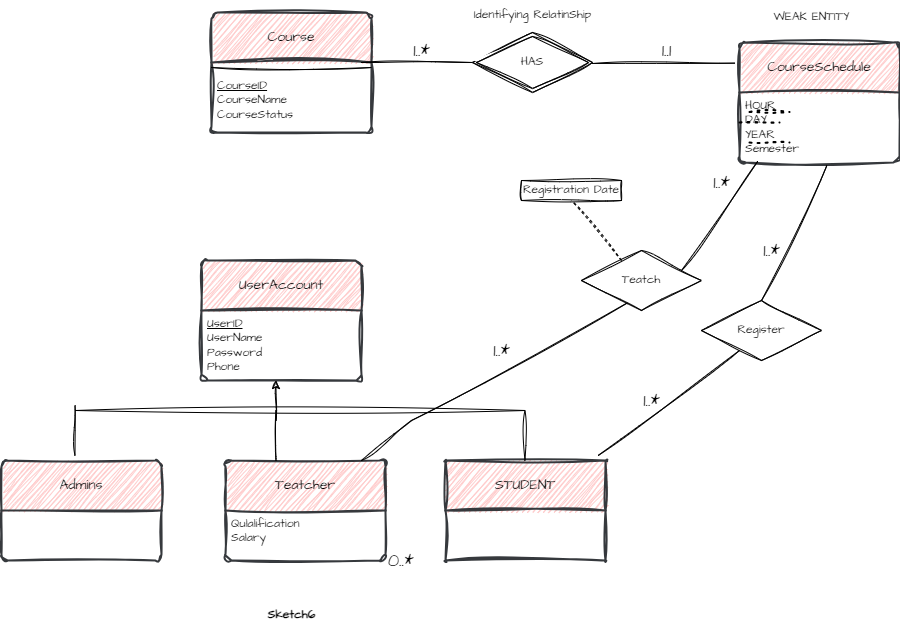
****

**Each course has several students linked to it, and one student is linked to several courses. To obtain a registration table containing the student’s name, course name, and dates, we linked it to the Corse Schedule table and for the same reson we linked Teacher with it.**

**A Teacher gives more than one course, and one course is given by more than one teacher, so the relationship is Many to Many.**

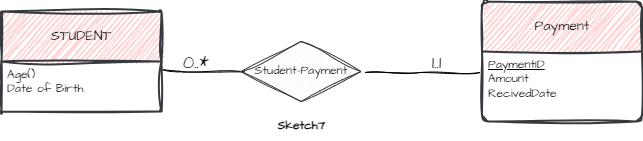
**Each appointment is assigned of Teacher so Total Participation.**

**In the Student's relationship with the Course schedule we used descriptive Attributes .**

****

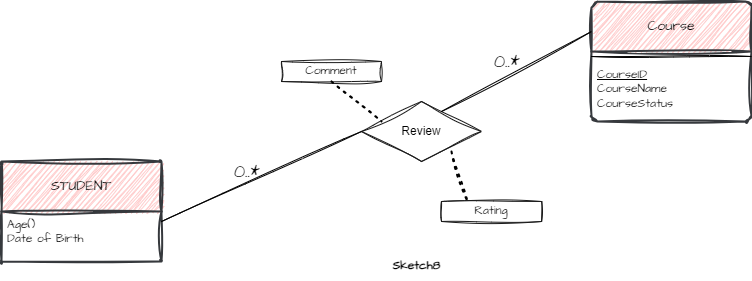
**The relationship between the Student and Payment is one to Many .**

**Partial** **participation on the part of Student and** **Total participation on the part of Payment.**

****

**The Student can add several comments on a Course, and one Course is linked to several comments, hence the relationship between the Student and Course is Many to Many.**

**Partial Participation on both Student and Course.**

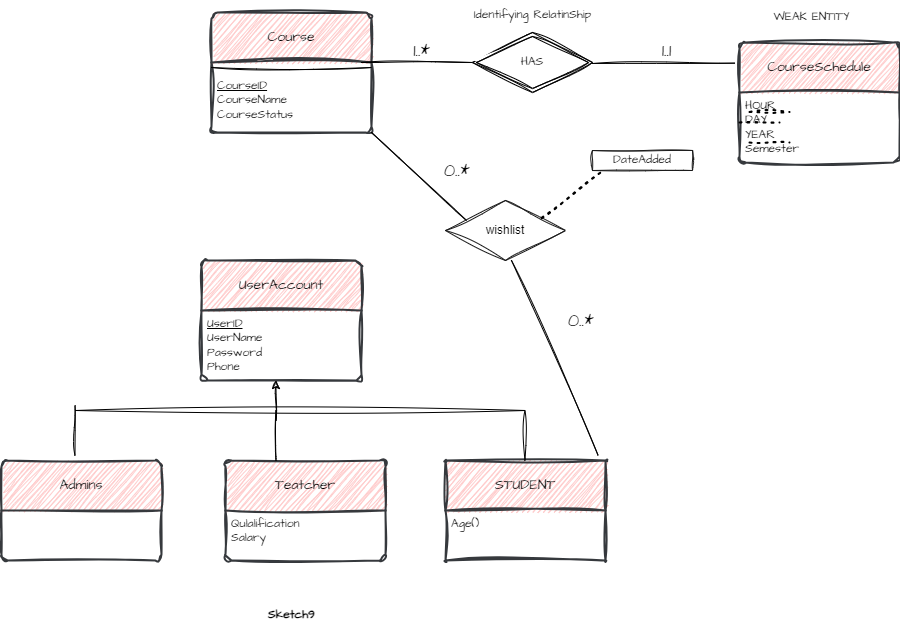
****

**A student can have a group of courses he is interested in.**

**Partial** **Participation on both Student and Course.**

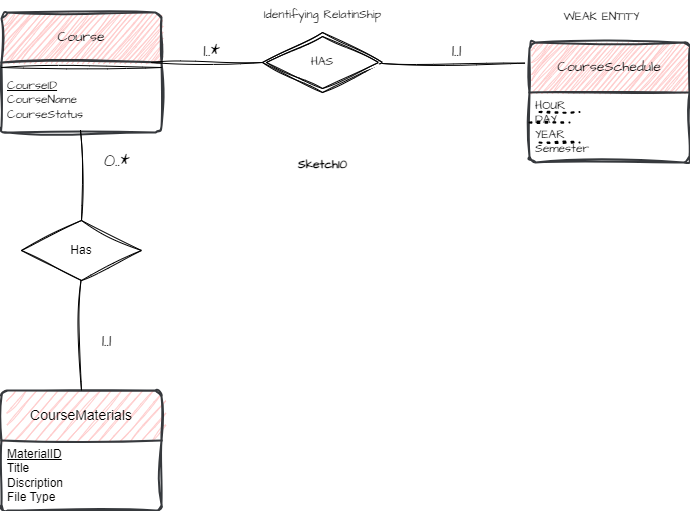
**The relation ship between Student and Course is Many to Many.**

**In the Student's relationship with the Course we used descriptive Attributes.**

****

**The relationship between Course and Material is one to Many.**

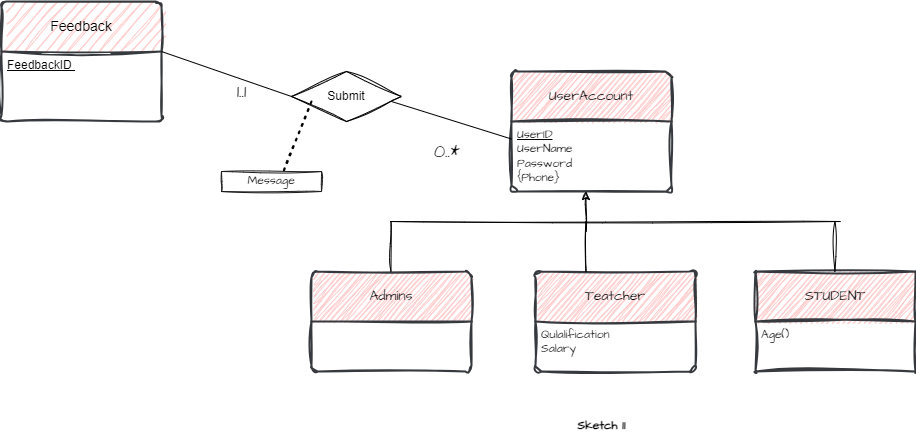
**Total Participation on the side of Material and Partial on Course.**

****

**Anyone who has an account, whether a student, teacher or admin, can give us feedback.**

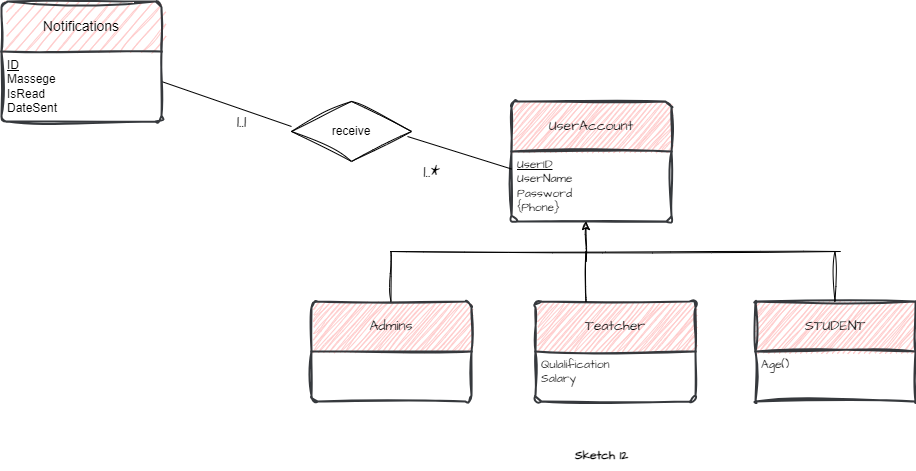
**Partial participation on the side of user.**

**The relationship between user and Feedback is one to Many.**

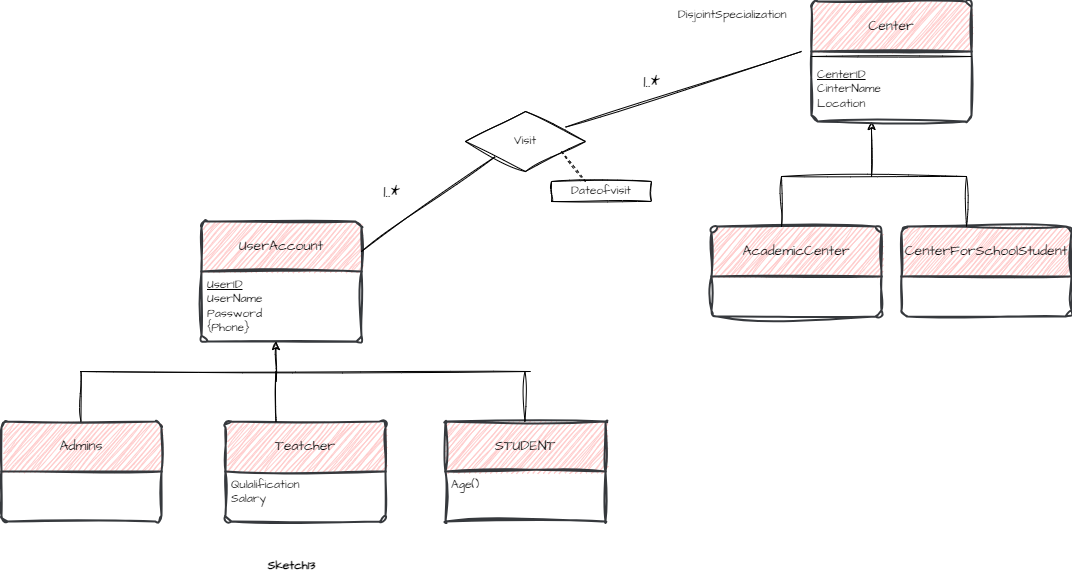
****

**A User Account can receive multiple Notifications and each**

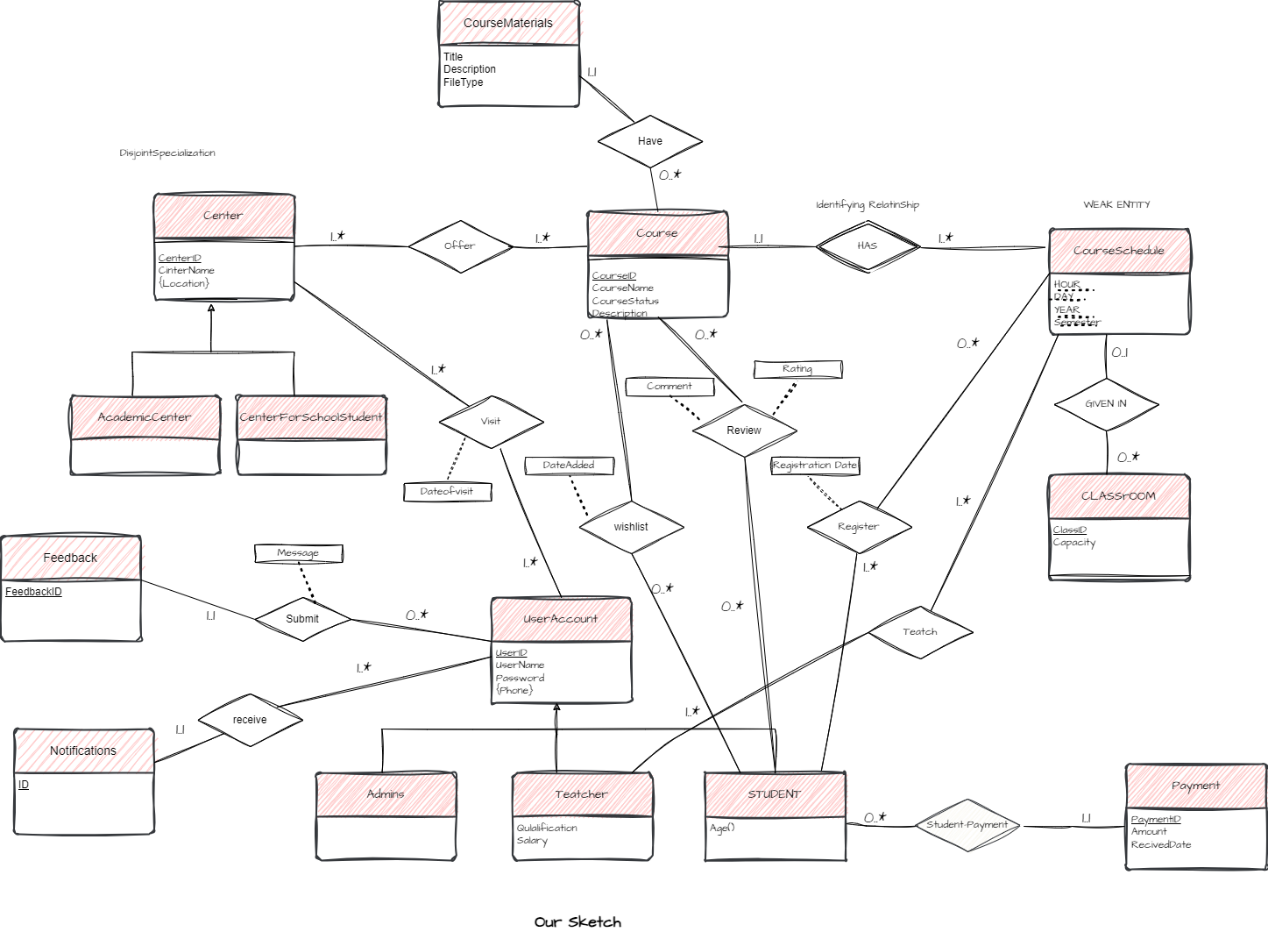
**Notifications linked with only one User.**

****

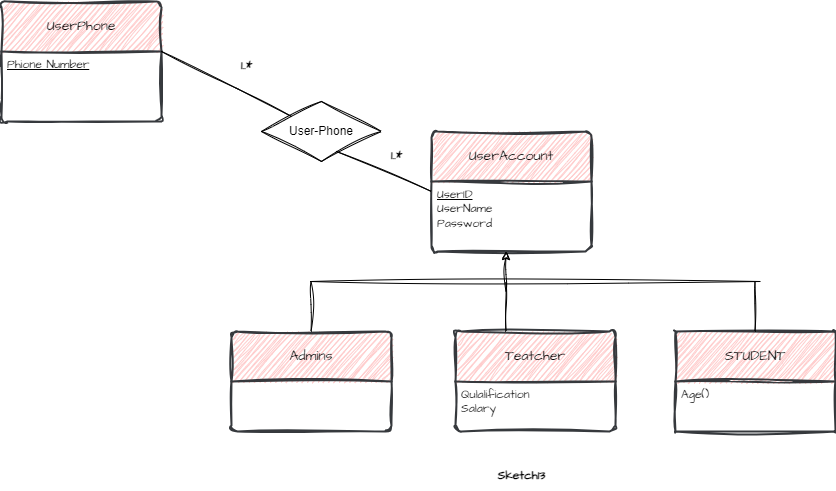
**Any person who has an account visits any center, and at the same time the center is accessed by more than one user, so the relationship between User and Center is Many to Many.**

****

**ERD Sketch**

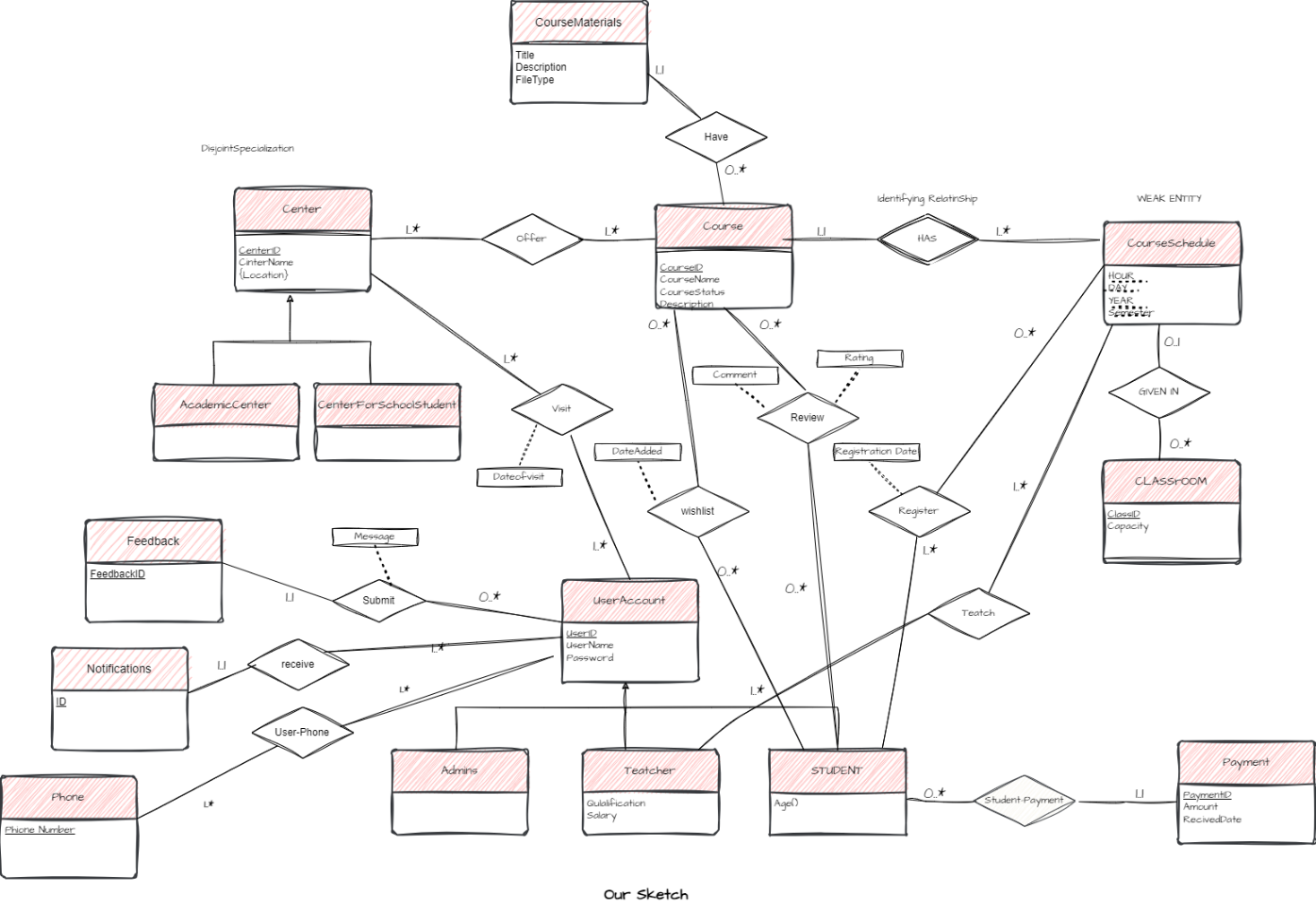
****

**The phone number can contain more than one value, so it can be separated into another table that contains the user number and the phone number.**

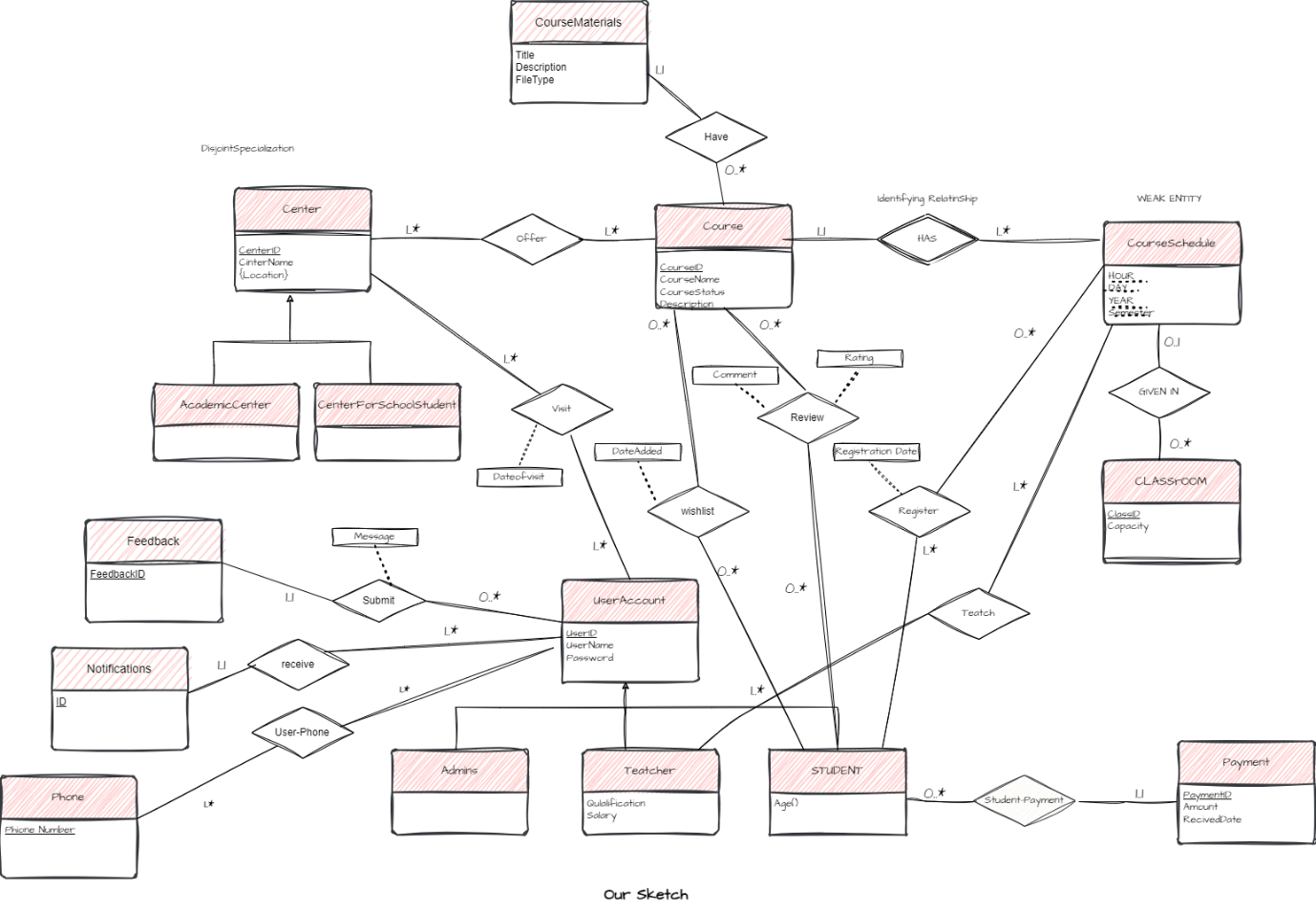
****

**We can keep phone in our ERD, even though it holds more than one value, but we must separate it from our table after the process of converting the ERD to a relational table, because in order for us to achieve a good design, we must achieve the first Normlization.**

**That is, in all cases, it will be separated, whether when we design the ERD or when converting the ERD to relational tables.**

****

**Final ERD Sketch**

****