**A. Identify the Entities and the relationship in the Community Portal Database**

Portal komunitas adalah sebuah sistem yang dapat menemukan antara programmer dengan perusahaan yang sedang membutuhkan programmer. Pada sistem ini memiliki dua buah user, yaitu Administrator dan Software Programmer. **Administrator** sendiri memiliki rule yaitu, Admin dapat mengadministrasi data user, serta mengirimkan email invitation kepada programmer untuk melakukan registrasi. **Software Programmer** sendiri memiliki rule berupa manajemen akun masing-masing, melihat profile user lain, mempost pekerjaan baru, mengirim pesan dan lain-lain.

Dari penjabaran tersebut, aplikasi portal komunitas ini terdapat lima buah entitas, dimana masing-masing entitas memiliki relasi antar entitasnya. Setiap entitas juga memiliki atributnya masing-masing yaitu :

**1. Administrator :**

* Admin\_ID
* First\_Name
* Last\_Name
* Username
* Password

**2. Software Programmer**

* User\_ID
* First\_Name
* Last\_Name
* Username
* Password
* Company\_Name
* City
* Country

**3. Message**

* Message\_ID
* Room\_ID
* Sender\_ID
* Message

**4. Thread**

* Thread\_ID
* User\_ID
* Title
* Text
* Hastag

**5. Job Vacancy**

* Jobs\_ID
* User\_ID
* Title
* Description
* Category
* Created\_Date
* Salary
* Status

**Dan juga terdapat enam buah relasi, Yaitu :**

Pada entitas **Administrator** dengan entitas **Software Programmer** terdapat relasi berupa pembuatan akun dari Software Programmer. Pada relasi antara kedua entitas ini bersifat *one to many*, dimana satu admin dapat membuat atau mengadministrasi lebih dari satu software programmer. Relasi ini bersifat wajib dikarenakan untuk membuat suatu akun baru, harus ada admin yang terlibat dalam proses pembuatan.

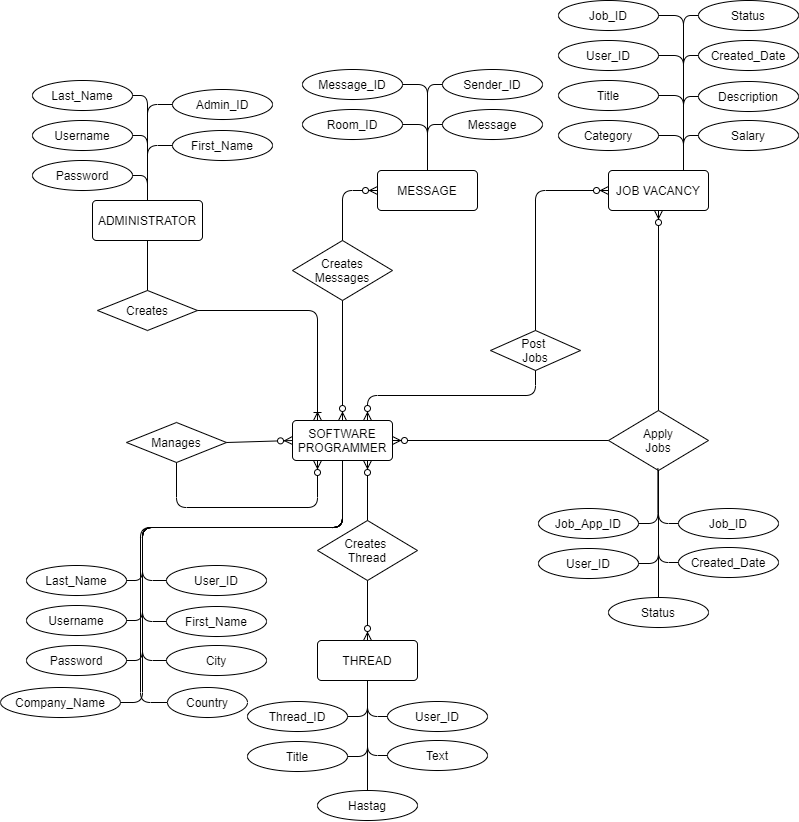
Pada entitas **Message** terdapat relasi dengan **Software Programmer** berupa pembuatan pesan. Relasi ini bersifat *many to many*, hal ini dikarenakan bahwa lebih dari satu software programmer yang bisa membuat lebih dari satu pesan, dan satu pesan dapat dilihat oleh lebih dari satu user yang berada pada room yang sama. Setiap pesan akan memiliki satu Id Sender dan satu Id Room.

Pada entitas **Thread** dengan entitas **Software Programmer** terdapat relasi berupa pembuatan thread. Relasi ini juga bersifat *many to many*, hal ini dikarenakan satu software programmer dapat membuat lebih dari satu thread dan satu thread juga dapa di akses oleh lebih dari satu software programmer. Setiap thread memiliki satu user Id dan sekumpulan thread dapat di filter berdasarkan hastag atau kategori.

Pada entitas **Software Programmer** terdapat relasi terhadap entitas **Software Programmer** sendiri yaitu manajemen data software programmer, relasi ini bersifat *many to many*, diakarenakan satu software programmer dapat melihat atau mengunjungi publik profile lebih dari satu software programmer. Manajemen data software programmer disi melingkupi update profil, reset password, login, mengunjungi profil, melakukan pencarian, dan lain-lain.

Pada entitas **Job Vacancy** dengan entitas **Software Programmer** terdapat dua buah relasi. Relasi yang pertama adalah post jobs. Relasi ini bersifat *many to many*, dikarenakan software programmer yang juga termasuk perusahaan yang membutuhkan jasa programmer dapat memposting lebih dari satu jobs dan satu jobs dapat di akses oleh lebih dari satu software programmer. Setiap jobs akan memiliki satu user Id pembuat jobs, dan sekumpulan dari data jobs dapat difilter berdasatkan kategori. Relasi yang kedua adalah apply jobs. Relasi ini bersifat *many to many,* dikarenakan satu software programmer dapat melamar pada lebih dari satu jobs, dan satu jobs dapat menerima lamaran lebih dari satu software programmer. Pada relasi ini menghasilkan atribut baru yaitu job\_application\_ID, Job\_ID, User\_ID, Created\_Date dan status. Atribut user\_id akan menyimpan id dari software programmer sedangkan job\_id menyimpan id dari job yang dilamar oleh user tersebut.

**B. Create ER Diagram for the Community Portal Database**

****