# Accounts

* + Role: (What role a profile has at the club. Example: Player and/or Administrator)
    - role = text field.
      * Add England Volleyball Administrator
  + Position: (What position a profile has in a role. Example: Wing Forward and/or Club President)
    - Position = text field
  + Profile: (This is where a person’s account information is stored. Everything relating to an individual’s account is stored here)
    - user =OneToOneField(User)
    - club =ForeignKey('clubs.Club')
    - role =ManyToManyField(Role)
    - position =ManyToManyField(Position)
    - bio =TextField
    - birth\_date =DateField
    - ADD qualification (should be multi choice)
    - ADD first aid yes/no
    - ADD first aid expiration date.
  + Qualification (Should be a CHOICE\_LIST and relate back to profile)

# Articles:

* + Article: (Stores the
    - user =ForeignKey(User)
    - title =CharField)
    - post\_slug =SlugField
    - short\_description =CharField
    - body = models.TextField()
    - image =ImageField
    - image\_credit =CharField
    - posted =DateTimeField
    - ADD the ability to declare what teams are involved in the article. Team A should always be (request.user.club). Team B should be a ManyToMany(Team) incase a story has multiple teams selected in it.

# Clubs (DONE)

* + Club: (Contains all club data)
    - name =CharField
    - address1 =CharField
    - address2 =CharField
    - town =CharField
    - post\_code =CharField
    - logo =ImageField()
  + Team (should contain squad info, possible use of a “Through” model?)
    - name = models.CharField()
    - club = models.ForeignKey(Club)
    - logo = models.ImageField()
    - gender = ChoiceField(Returns Men, Women, Mixed)
  + Squad
    - Profile
    - Member\_joined (time\_date add\_now =True)
  + Club Infomation

# Fixtures

* + Match
    - League (ForeignKey=League)
    - Date (TimeDateField)
    - Team\_A (ForeignKey, associate with the request.profile.team on the HTML)
    - Team\_B(ForeignKey)
  + Match\_Data
    - Match (ForeignKey = Match)
    - Team\_A\_Lineup (many to many field = Clubs.Squad)
    - Team\_B\_Lineup (many to many field)
    - Team\_A\_Score
    - Team\_B\_Score
  + Scorer\_Data (ask if this is required, if it is: One entry per player in match, not per score. Score time replaced by “Score = IntField” )
    - Match (ForeignKey = Match)
    - Player = (foreginkey = Clubs.Squad)
    - Score\_time (time\_data = add\_now.True)

# Leagues

(contains all the league related data)

* + League (only EV admins have permission to add to this)
    - League Sponsor (blank=true)
    - League Name(default=”none”)
    - League Contact (ForeignKey = Profile)
    - Age\_group(choice\_field)
    - Gender(choice\_field)
  + League\_Table
    - Year (Date Field)
    - League = ForeignKey = League
    - Team (many to many = Clubs.Team)
  + League\_Table\_Data
    - Team (ForgineKey = League\_Table.Team)
    - Win
    - Loss
    - Draw
    - SF?
    - SA?
    - SQ?
    - PA?
    - PF?
    - PQ?
    - Points

# Sponsors

(DONE)

* + Club (ForgineKey = Club)
  + Name
  + Logo (image Field)
  + Info(long text field)
  + Deal\_end\_date (datefield)

# Things to add:

## Admin Area:

* Fixtures. MatchData: need to adda filter so that teamA/teamB is automatically selected when you select the match you want to assign data for.
* Leagues. LeagueTableData: need to filter teams based on the league selected. Chained Foreign key should work.

## Permissions:

* Need to add permissions:
  + VE \_admin permissions: A user should be able to interact with everything. Full access to API
  + Club\_admin permission: A user should only be able to see and interact with a club he is a member of and VE directly. No access to API
  + API permissions. A user can interact with most Data. Some is off limits, contact data maybe? Can only delete its own objects.
  + Player/official permissions. Should only be able to interact with its own objects.