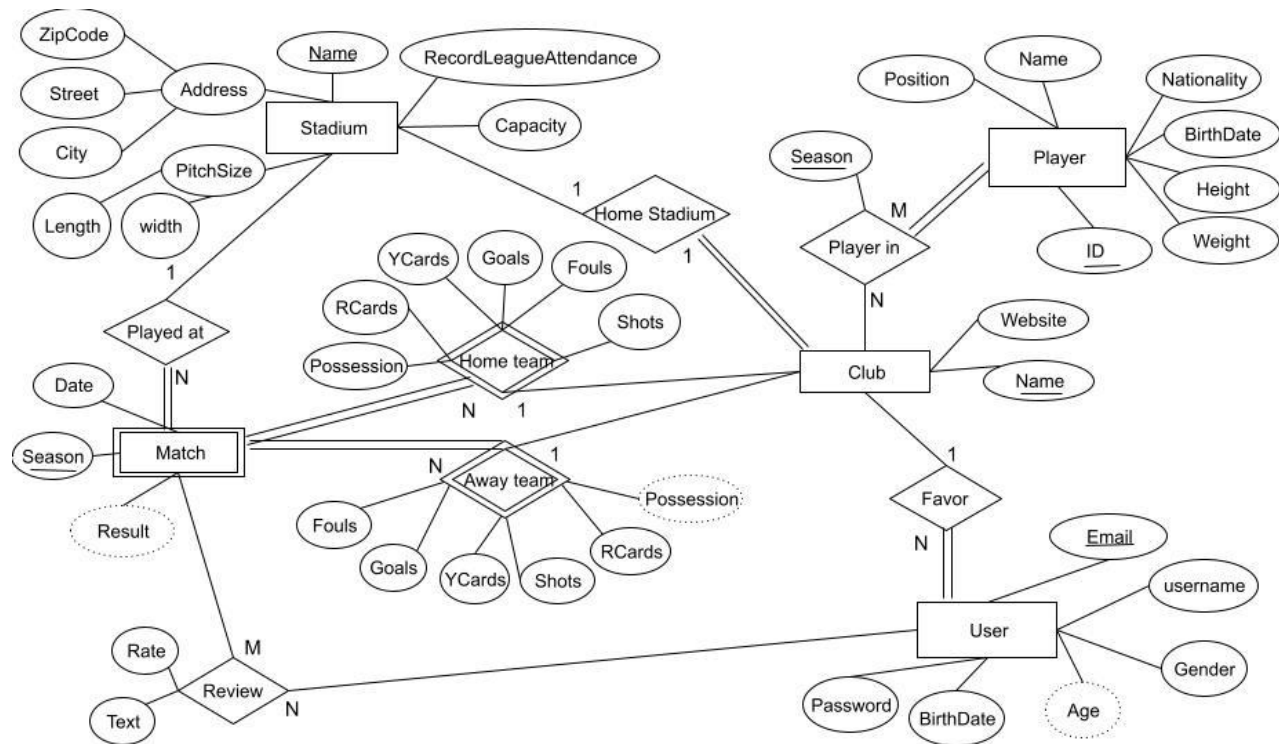


**Note:** I used this pattern for describing the schemas for consistency: **primary key**, **foreign key** [Table, Column], **both** (PK, FK).

### ERD:



### Relational Model:

Stadium(**Name**, RecordLeagueAttendance, Capacity, ZipCode, Street, City, LengthMeter, WidthMeter)

Club(**Name**, Website, HomeStadium [Stadium, Name])

Player(**ID**, Name, BirthDate, Position, Nationality, Weight, Height)

ClubPlayers(ClubName [Club, Name], PlayerID [Player, ID], **Season**)

Match(**Season**, HomeClub [Club, Name], AwayClub [Club, Name], Stadium [Stadium, Name], Date, HomePossessions, HomeRCards, HomeYCards, HomeGoals, HomeFouls, HomeShots, AwayRCards, AwayYCards, AwayGoals, AwaysFouls, AwayShots)

User(**Email**, UserName, Password, BirthDate, Gender, FavoriteClub [Club, Name])

MatchReviews(UserEmail [User, Email], Season [Match, Season], HomeClub [Match, HomeClub], AwayClub [Match, AwayClub], Text, Rate)

### Design Decisions:

For the player names, there were many ambiguities since players did not all have the same number of names (some had only first, other first and last, others had 4 names). Also, some players had the exact same name such as Aaron Ramsey.

Therefore:

- ➔ Full name, is not considered a candidate key.
- ➔ The primary key chosen is player id, since that is what is represented in the website where each player has a unique id that can be scrapped.
- ➔ Only one attribute called name is created for the name in player table. For querying by first or last names, wild cards or regex can be used.