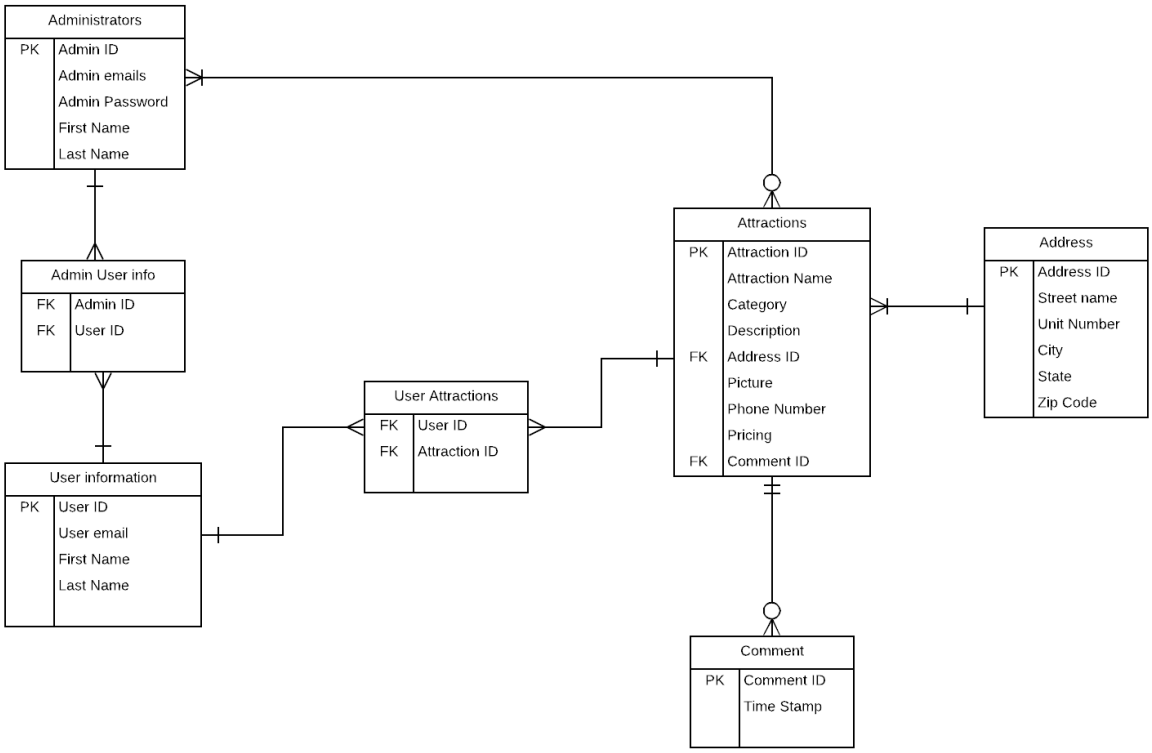
Administrators
PK
Admin ID
Admin emails
Admin Password
First Name
Last Name
Admin User info
FK
Admin ID
FK
User ID
User information
PK
User ID
User email
First Name
Last Name
User Attractions
FK
User ID
FK
Attraction ID
Attractions
PK
Attraction ID
Attraction Name
Category
Description
FK
Address ID
Picture
Phone Number
Pricing
FK
Comment ID
Address
PK
Address ID
Street name
Unit Number
City
State
Zip Code
Comment
PK
Comment ID
Time Stamp

Entity Relational Diagram Explanation

Administrators' Entity stores the Admin ID, Admin Emails, Admin Passwords, First Name, and Last Name. With Admin ID as the primary key where we can access all of the other attributes. The cardinalities attached to that table say that administrators have control over zero to many attraction in which attractions don't have to be administrated all the time, however, attraction are administrated are controlled by one to many administrators.

User Information entity stores the userID, user email, First name, and last name. With UserID as the primary key which allows us to retrieve all of the other attributes. This helps us capture the first name to send a confirmation email to the users emails.

Since there was a many-to-many relationship between Administrators and User Information we needed an associative entity and we called it Admin User Info, it merges the needed information between administrators and users. This entity helps us with security, in which users are unable to view passwords for admins.

Attractions Entity contains all of the information stored about our attractions including, the name, category, description, address, picture, phone number, pricing, & comments.
We have a primary key called Attraction ID which contains all of the attributes.
Again we had a mandatory many-to-many relationship between Attractions and User information we created an associative entity called User Attraction which allows users to read information about attractions and allows us to store information about which users are subscribed to which attractions.

Each Attraction has a comment feature, in which users are able to write comments about each attraction. We want to be able to track the users by their comments and so every comment is related to the users and to the attractions by the foreign keys in the user attraction associative entity. In the comment table we are storing a comment ID and a time stamp.
Our website has an enforced rule that only allows those who are subscribed to be able to leave a comment on an attraction, that way we are able to capture user ID's from our users and not have to store every comment ID written by anyone, to make it easier to manage our database.

The address entity was part of our normalization process, since Address attribute has multiple values, and so we created a new table for it to store each part of the address, Address ID (which is our primary key), Street name, unit number, city, state, and zip code.

GitHub link
https://github.com/YasminAlshafai/JustGo-Boulder/commit/067719af9c60817b47a0b8c0b164c3afdf5c9821