

# PetVetSet Data Requirements

## 1. Entities specification

NOTE: All attributes are non-null unless otherwise specified.

- A USER has a unique user\_id, a unique email address (string), is\_admin (Boolean: attribute to differentiate between normal users and admins), a profile photo, join\_date (date), password (string, may be null)
- ~~Not all USERS have passwords stored in the system. [NOT YET CONFIRMED – see issue #1]~~
  - ~~For example, Gmail users will use google OAuth service to sign up/ log in, and therefore their passwords will not be stored. Non-google user passwords are stored in the system.~~
  - ~~So, a NATIVE\_AUTH\_USER is just a USER but with a password (for users who do not use external authentication services)~~
- A POST can either be a PET, a PRODUCT, or a SERVICE.
- A POST has a unique id, a post\_date (date), a description (string), an image, is\_approved (Boolean: the entity must be reviewed by an admin before viewing it to other users) and a price (float).
  - NOTE: any POST attribute is automatically inherited by its sub entities.
- A PET has ~~a unique pet\_id~~, an owner (the USER offering the PET), animal\_type (string), breed (string) and a name (string).
- A PRODUCT (food/accessories) has ~~a unique product\_id~~, name of brand/company (string), [a quantity (integer): LATER – see issue #3], category\_name (string: “cat food”, “dog food”, “cleaning product”, etc.), target\_animal (string: for which animal is this product targeted?), a product\_name (string), and a product\_review (float – **derived from all posts that contain that product** – see issue #2)
- A SERVICE has a provider\_name (string), category (string: “vet”, “hair cut”, etc.), ~~a unique service\_id~~, and a [service\_rating (float – **derived??**) – see issue #2]
- A REVIEW is identified by the USER who made the review and the POST reviewed. A REVIEW has a value (integer 0:5), date of review (date).

## 2. Relationships

- An admin USER can post any POST.
- A USER can post a PETs and PRODUCTS, but not SERVICES (application constraint, not shown in ERD)
- A USER can make a REVIEW.

- A REVIEW is made on a specific POST. The overall review (rating) of a POST is a **derived** attribute.
- A USER can bookmark a POST.
- A USER can put in cart a POST.
- A POST is a PET.
- A POST is a PRODUCT.
- A POST is a SERVICE.
- ~~A NATIVE\_AUTH\_USER is a USER.~~

### 3. Multiplicities

- ~~A NATIVE\_AUTH\_USER corresponds to one USER. A USER may correspond to one NATIVE\_AUTH\_USER.~~
- A USER can post many POSTs, A POST is offered by exactly one user.
- A USER can bookmark multiple POSTs. A POST is bookmarked by multiple USERS.
- A USER can put\_in\_cart multiple POSTs. A POST can be in the cart of multiple USERS.
- A USER can make multiple REVIEWS. A REVIEW is made by exactly one USER.
- A REVIEW is made on exactly one POST. A POST has many REVIEWS.
- A POST has many REVIEWS
- A POST has either one PRODUCT, one PET, or one SERVICE.
- Each PRODUCT, PET, SERVICE is a to one POST. [see issue #2]

### Undecided Issues

1. Google OAuth: should users of OAuth not have passwords stored in system?
2. ~~Should PETs, PRODUCTS, and SERVICES be separated from POSTs, so that the same entity can appear in multiple posts (is\_a vs has\_a)?~~
  - a. ~~If POSTs are separate entities from (PETs, PRODUCTS, and SERVICES), then each POST should have its own derived overall review, and each (PET PRODUCT, and SERVICE) should have another overall review derived from the reviews of all posts that it belongs to.~~
3. How to handle product "quantity"?