Applications

Design of models:

owner: ForeignKey(Account), Required (readonly)
petlisting: ForeignKey(PetListing), Required
shelter: ForeignKey(Account), Required (readonly)
applicant: CharField, Required
email: EmailField, Required
phone1: CharField, Required
phone2: CharField
description: TextField, Required
status: CharField, Required (readonly), must be one of: 'PENDING', 'ACCEPTED', 'DENIED', 'WITHDRAWN'
creation_time: DateTimeField (auto_now_add) (readonly)
last_update_time: DateTimeField (auto_now) (readonly)

Full list of API endpoints:

- applications/
  - Get: ListView of every application for any pet listing made by the currently logged in shelter, or every application made by the currently logged in seeker
    - Query param 'filter' can be set to one of the status values to filter applications with that status. (e.g. ?filter=PENDING)
    - Query param 'sort' can be set to either 'creation_time' or 'last_update_time' to sort results by the given field. (e.g. ?sort=creation_time)
  - Post: CreateView for application. Raises 403 error instead if logged in user is a shelter.
    - Fields: owner, petlisting (id), applicant, email, phone1, phone2 (optional), description
- applications/<int: pk>/
  - Get: DetailView of application with id 'pk' created by currently logged in seeker, or created for a pet created by currently logged in seeker. Raises 404 error if no such application exists.
  - Put: UpdateView of that application where, if currently logged in user is a seeker, can only change status from 'PENDING' or 'ACCEPTED' to 'WITHDRAWN'. Otherwise, if currently logged in user is a shelter, can only change status from 'PENDING' to 'ACCEPTED' or 'DENIED'. There are no allowed updates to be made if the status is not one of these values.
    - Fields: status (or none)

Notifications

Design of models:

owner: ForeignKey, Required (readonly)
state: CharField, Required (readonly), must be one of: ‘read’, ‘unread’
creation_time: DateTimeField (auto_now_add) (readonly)
message: TextField, Required
link: URLField, Required

Full list of API endpoints:

- notifications/
  - Get: ListView of every notification owned by the currently logged in user.
    - Query param ‘filter’ can be set to one of the state values to filter notifications with that state. (e.g. ?filter=read)
    - Query param ‘sort’ can be set to ‘creation_time’ to sort results by the creation time. (e.g. ?sort=creation_time)
  - Post: CreateView for notification.
    - Fields: message, link
- notifications/<int: pk>/
  - Get: DetailView of notification with id ‘pk’ owned by currently logged in user. Raises 404 error if no such application exists.
  - Put: UpdateView of that notification where, if state is currently ‘unread’, can change to ‘read’. If state is ‘read’, no changes can be made.
    - Fields: staat
  - Delete: DeleteView of that notification.

Listings

design of models:

class PetListing:
name = TextField, **Required**
breed = TextField
age =.PositiveIntegerField
gender = CharField
size = CharField
description = TextField
avatar =ImageField, (upload_to='avatars/')
created_date = DateTimeField, **Required (auto_now_add)**

status = CharField, **Required**, Must be one of: 'PENDING', 'AVAILABLE', 'ADOPTED', 'WITHDRAWN'

shelter = Foreign Key, **Required (auto_now_add)**

full list of all API endpoints:

listings/

- all/
  - Description: Will list all of the available pet listings
  - **Methods:** GET
- ''
  - Description: Will list all of the available pet listings for the logged-in shelter and allow the user to create a new one
  - **Methods:** GET, POST
  - **Fields/payload:** status, name, age, breed, gender, size, description, avatar
- <listing_id>
  - Description: Allows user to Retrieve Update and Delete a Listing of the specified ID
  - **Methods:** GET, PUT,DELETE
  - **Fields/payload:** status, name, age, breed, gender, size, description, avatar
- filters/<filter1>/<filter2>/<filter3>/<filter4>/<keyword>/results/<sort1>/<sort2>
  - Description: Allows the user to filter and sort by categories. Filter must be one of shelter, status, breed, or gender, and Sort must be one of age or name. The endpoint can consist of 1-4 filters (inclusive) if "filters" is specified and 1-2 filters (inclusive) if "results" is specified. If "filters" isn't specified then 0 filters and no keyword is allowed (will just sort). If "results" isn't specified then 0 sorts is allowed (will just filter).
  - **Methods:** GET

Account

Account Model

Inherited from Abstractuser

is_active =BooleanField, default is True

seeker_or_shelter = BooleanField to check if account is pet seeker or shelter

phone_num = CharField

location = CharField

preferences = CharField

profile_pic =ImageField

mission_statement = TextField

full list of all API endpoints:

accounts/

- users/
    - Description: Create a pet seeker
    - **Methods:** POST
    - **Fields/payload:** username, email, first name, last name, password
- shelters/
    - Description: Create a shelter
    - **Methods:**POST
    - **Fields/payload:** username, email, first name, last name, password
- user/<int:pk>/
    - Description: Can access, update and delete a pet seeker
    - **Methods:** GET, PUT, DELETE
    - **Fields/payload:** username, email, first name, last name, password
- shelter/<int:pk>/
    - Description: Can access, update and delete a shelter
    - **Methods:** GET, PUT, DELETE
    - **Fields/payload:** username, email, first name, last name, password

Comment

Comment Model

Text = TextField
User = ForeignKey(Account)
Create_time = DateTimeField(auto_now_add)
Shelter = ForeignKey(Account), optional
Application = ForeignKey(Application), optional
Reply = ForeignKey(Comment), optional

full list of all API endpoints:

comments/

- app/<int:pk>/comments/
    - Description: will list all comments on the given application (identified by pk) and allow the user to create a new one.
    - **Methods:** GET, POST
    - **Fields/payload:** text
- <int:pk>/
    - Description: will allow you to create a reply to the comment (identified by pk)
    - **Methods:** POST
    - **Fields/payload:** text

- shelter/<int:pk>/reviews/
    - Description: will list all review comments on the given shelter (identified by pk) and allow any user or shelter to create a new one.
    - **Methods:** GET, POST
    - **Fields/payload:** text