Cupet Backend CRC

Andrew Qiu
Daniel Baek
Fangyi Li
Harry Xu
Katherine Jelich
Kenneth Tran

Legend:

🟦 Frameworks & Drivers

🟩 Interface Adapters

🟥 Application Business Rules

🟨 Enterprise Business Rules

Backend Specifications

Establishes database, setup API endpoints for the frontend

ServerApplication

Responsibilities:

- Boot up the backend server

Collaborators:

UserRepository

Responsibilities:

- Connect to the User collection/repository/table in database framework
- Implement methods defined in IUserRepository

Collaborators:

- IUserRepository
- HTTPSGateway

PetRepository

Responsibilities:

- Connect to the Pet collection/repository/table in database framework
- Implement methods defined in IPetRepository

Collaborators:

- IPetRepository
- HTTPSGateway

HTTPGateway

Responsibilities:

- Implement abstract methods in APIGateway by calling methods in UserController, PetController, and SessionController
- Uses HTTP API endpoints

Collaborators:

- APIGateway
  - UserController
  - PetController
  - SessionController
- UserRepository
- PetRepository

CmdLineGateway

Responsibilities:

- Implement abstract methods in APIGateway by calling methods in UserController, PetController, and SessionController
- Uses command line commands

Collaborators:

- APIGateway
  - UserController
  - PetController
  - SessionController

UserController

Responsibilities:

- Implements methods defined in IUserController:
    - Create a new user
    - Fetch a user’s profile (via their id)
    - Fetch a user’s account information (via their id)
    - Edit an existing user’s profile (via their id)
    - Edit an existing user’s account (via their id)

Collaborators:

- IUserController
- UserCreaterRequestModel
- UserCreaterResponseModel

IUserController (Interface)

Responsibilities:

- Defines methods that will call use cases:
    - Create a new user
    - Fetch a user’s profile (via their id)
    - Fetch a user’s account information (via their id)
    - Edit an existing user’s profile (via their id)
    - Edit an existing user’s account (via their id)

Collaborators:

- UserController

PetController

Responsibilities:

- Implements methods defined by IPetController:
  - Create a new pet given a user id, pet name, pet pictures, etc.
  - Edit an existing pet

Collaborators:

- IPetController

IPetController (Interface)

Responsibilities:

- Defines methods that will call use cases:
  - Create a new pet given a user id, pet name, pet pictures, etc.
  - Edit an existing pet

Collaborators:

- PetController

MatchController

Responsibilities:

- Return a list of potential matches for a given pet
- Indicate that a user wants their pet to match with another pet (swiping right)

Collaborators:

- IMatchController

SessionController

Responsibilities:

Stores:

- Handles the logging in of users via email and password. Returns a session id for that user
- Return a user id given a session id.

Collaborators:

- ISessionController

APIGateway (Interface)

Responsibilities:

- Defines methods for each of the API endpoints we need. For instance,
  - Creating a user
  - Editing a user’s profile
  - Editing a user (private account details)
  - Creating a pet
  - Etc.

Collaborators:

- IPetController
- IUserController
- ISessionController
- HTTPSGateway
- CmdLineGateway

IUserRepository (abstract)

Responsibilities:

- Define methods to:
    - Retrieve stored information from a user given their ID
    - Return a list of users in a latitude and longitude range

Collaborators:

- UserRepository
- UserCreator
- UserProfileFetcher
- UserProfileEditor
- UserAccountFetcher
- UserAccountEditor

IPetRepository (abstract)

Responsibilities:

- Define methods to:
  - Retrieve stored information from a pet given their ID
  - Return a list of pets in a latitude and longitude range

Collaborators:

- PetRepository
- PetCreater
- PetProfileFetcher
- PetMatcher
- PetEditor

UserFromPetFetcher

Responsibilities:

- Fetch a user’s id from a pet’s id

Collaborator:

- IPetRepository

UserProfileFetcher

Responsibilities:

Fetch a user’s profile, which includes their name, pictures, biography, and (optionally) contact information.

Collaborator:

- IUserRepository
- UserProfileFetcherResponseModel
- UserController

UserProfileEditor

Responsibilities:

Edit a user’s profile, allowing them to change their pictures and/or biography.

Collaborator:

- IUserRepository
- UserProfileEditorResponseModel
- UserController

UserCreator

Responsibilities:

Create a user given a first and last name, email address, and home address, and password.

Collaborators:

- IUserRepository
- UserCreatorResponseModel
- UserCreatorRequestModel

UserCreatorRequestModel

Responsibilities:

- Input Boundary for the UserCreator use case
- Models the request object to pass as input into the UserCreator

Collaborators:

- UserCreator
- UserController

UserCreatorResponseModel

Responsibilities:

- Output Boundary for the UserCreator use case
- Models the response object to return as output from the UserCreator:

Collaborators:

- UserCreator
- User

UserAccountEditor

Responsibilities:

Edit a user, allowing them to change their name, email address, home address, and password.

Collaborators:

- IUserRepository
- UserAccountEditorResponseModel
- UserController

UserAccountFetcher

Responsibilities:

Fetch a user’s account details, including their name, email address, and home address.

Collaborators:

- IUserRepository
- UserAccountFetcher
- UserController

PetProfileFetcher

Responsibilities:

Fetch a pet’s profile, which includes its name, type of animal, pictures, type of animal, its owner’s profile, and (optionally) its owner’s contact information.

Collaborators:

- IPetRepository
- PetProfileFetcherResponseModel
- PetController

PetEditor

Responsibilities:

Edit a pet and its profile, allowing them to change its name, type of animal, biography, and pictures.

Collaborators:

- IPetRepository
- PetEditorResponseModel
- PetController

PetCreator

Responsibilities:

Create a pet given its name, type of animal, pictures, and its owner (user).

Collaborators:

- IPetRepository
- PetCreaterResponseModel
- PetController

PetMatcher

Responsibilities:

- Return a list of matches (with other pets) given a user's pet
  - The matches are based on the both users' preferences and locations
  - For instance, a user's pet should only be able to match with another pet nearby
  - The process should also take into account what kind of pets a user wants to match with (ex. Dogs only, cats only, or a mix of cats and dogs)

Collaborators:

- IPetRepository
- PetMatcherResponseModel
- PetController

CheckRequestAuthenticated

Responsibilities:

- For a request that requires that only the owner of a pet be able to make a request,
  - Given a session id and pet id
  - Return whether or not the user id corresponding to the session id matches the user id of the owner of the pet

Collaborators:

- CheckRequestAuthenticatedResponseModel
- SessionController

LoginUser

Responsibilities:

- Log-in a user/create a new session for a user given their email address and password. Return the session id of the user.

Collaborators:

- IUserRepository
- SessionController
- LoginUserResponseModel

GetUserIdFromSession

Responsibilities:

- Given a valid session id, get a user id

Collaborators:

- SessionController
- GetUserIdFromSessionResponseModel

User

Responsibilities:

- Store (instance variables):
    - Name: String
    - Address: String
    - Biography: String
    - Pictures: List<Image>
    - Contact Information: ContactInfo
    - List<Pet>
    - Reports
    - ID

Collaborators:

- ContactInfo
- UserCreatorResponseModel
- UserProfileFetcherResponseModel
- UserProfileEditorResponseModel
- UserAccountFetcher
- UserAccountEditorResponseModel

Pet (abstract)

Responsibilities

- Store:
  - Name
  - Biography
  - Pictures
  - Age
  - ID
- Methods
  - getBreed (abstract)

Collaborators:

- Dog
- Cat
- PetCreatorResponseModel
- PetProfileFetcherResponseModel
- PetMatcherResponseModel
- PetEditorResponseModel

Cat extends Pet

Responsibilities:

- Store:
  - Breed
- Methods:
  - getBreed

Collaborators:

- Pet

Dog extends Pet

Responsibilities:

- Store:
  - Breed
- Methods:
  - getBreed

Collaborators:

- Pet

ContactInfo

Responsibilities

- Store:
  - PhoneNumber: String
  - Email: String
  - Instagram: String
  - Facebook: String

Collaborators:

- User

SessionState

Responsibilities:

- Stores all active user sessions

Collaborators:

- LoginUserResponseModel
- CheckRequestAuthenticatedResponseModel
- GetUserFromSessionResponseModel