Rent-A-Pup

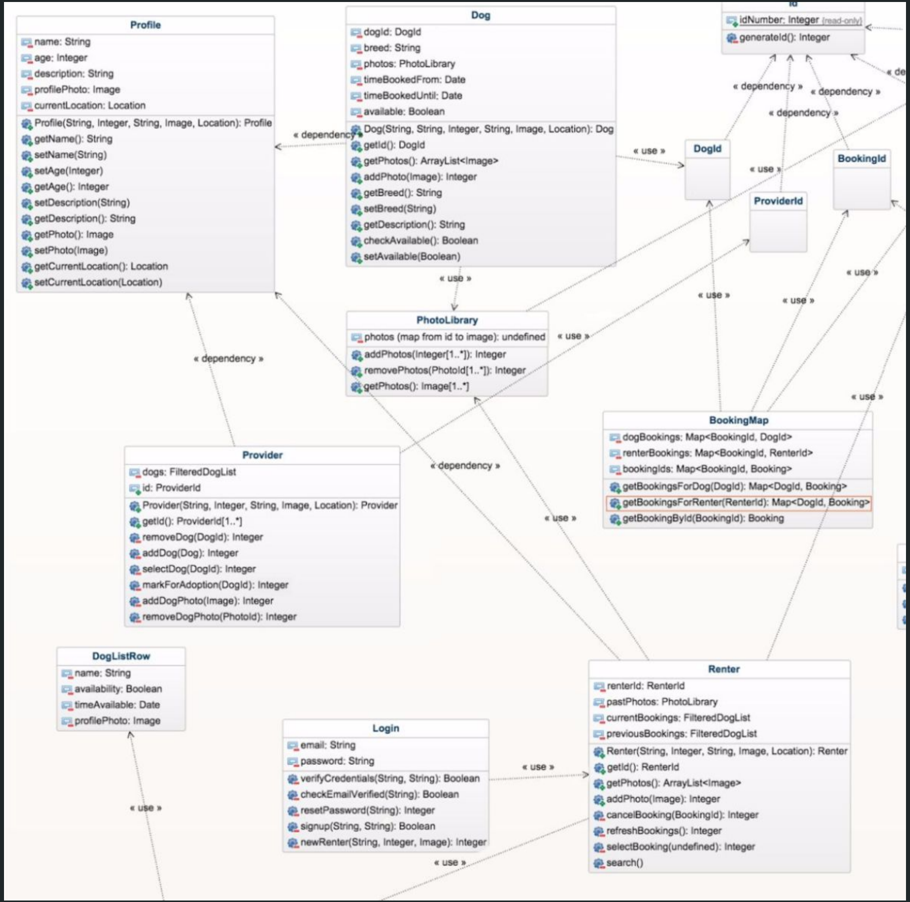
Elijah, Callie, Elias, Calvin

Demo

Design Pattern : Template

- Profile superclass allows for code reuse

- Easy integration for future profile types



Profile
name: String
age: Integer
description: String
profilePhoto: Image
currentLocation: Location
Profile(String, Integer, String, Image, Location): Profile
getName(): String
setName(String)
setAge(Integer)
getAge(): Integer
setDescription(String)
getDescription(): String
getPhoto(): Image
setPhoto(Image)
getCurrentLocation(): Location
setCurrentLocation(Location)

Dog
dogId: DogId
breed: String
photos: PhotoLibrary
timeBookedFrom: Date
timeBookedUntil: Date
available: Boolean
Dog(String, String, Integer, String, Image, Location): Dog
getId(): DogId
getPhotos(): ArrayList<Image>
addPhoto(Image): Integer
getBreed(): String
setBreed(String)
getDescription(): String
checkAvailable(): Boolean
setAvailable(Boolean)

Id
idNumber: Integer (read-only)
generateId(): Integer

DogId

ProviderId

BookingId

PhotoLibrary
photos (map from id to image): undefined
addPhotos(Integer[1..*]): Integer
removePhotos(PhotoId[1..*]): Integer
getPhotos(): Image[1..*]

BookingMap
dogBookings: Map<BookingId, DogId>
renterBookings: Map<BookingId, RenterId>
bookingIds: Map<BookingId, Booking>
getBookingsForDog(DogId): Map<DogId, Booking>
getBookingsForRenter(RenterId): Map<DogId, Booking>
getBookingById(BookingId): Booking

Provider
dogs: FilteredDogList
id: ProviderId
Provider(String, Integer, String, Image, Location): Provider
getId(): ProviderId[1..*]
removeDog(DogId): Integer
addDog(Dog): Integer
selectDog(DogId): Integer
markForAdoption(DogId): Integer
addDogPhoto(Image): Integer
removeDogPhoto(PhotoId): Integer

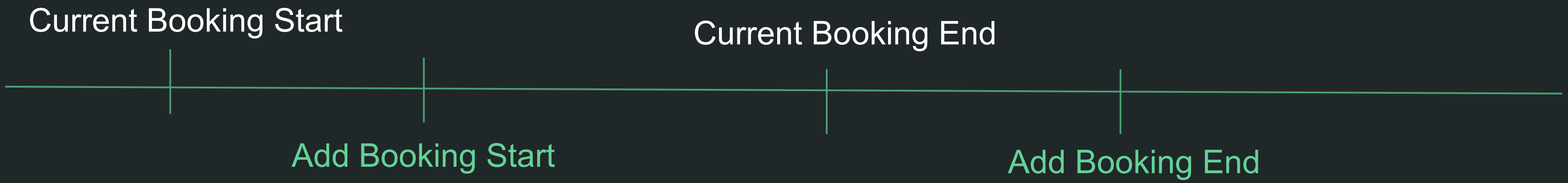
DogListRow
name: String
availability: Boolean
timeAvailable: Date
profilePhoto: Image

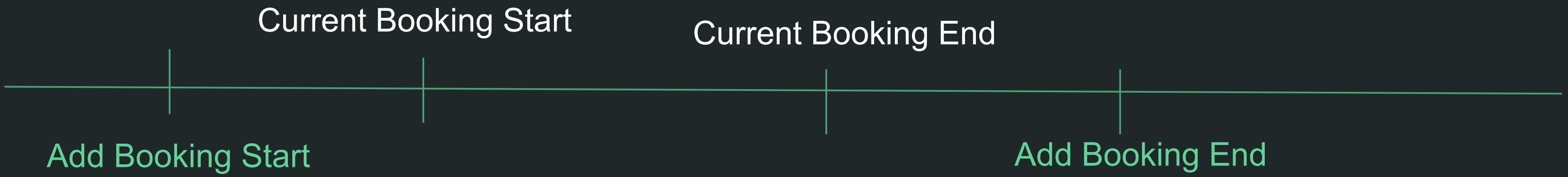
Login
email: String
password: String
verifyCredentials(String, String): Boolean
checkEmailVerified(String): Boolean
resetPassword(String): Integer
signup(String, String): Boolean
newRenter(String, Integer, Image): Integer

Renter
renterId: RenterId
pastPhotos: PhotoLibrary
currentBookings: FilteredDogList
previousBookings: FilteredDogList
Renter(String, Integer, String, Image, Location): Renter
getId(): RenterId
getPhotos(): ArrayList<Image>
addPhoto(Image): Integer
cancelBooking(BookingId): Integer
refreshBookings(): Integer
selectBooking(undefined): Integer
search()

Design Pattern : Command

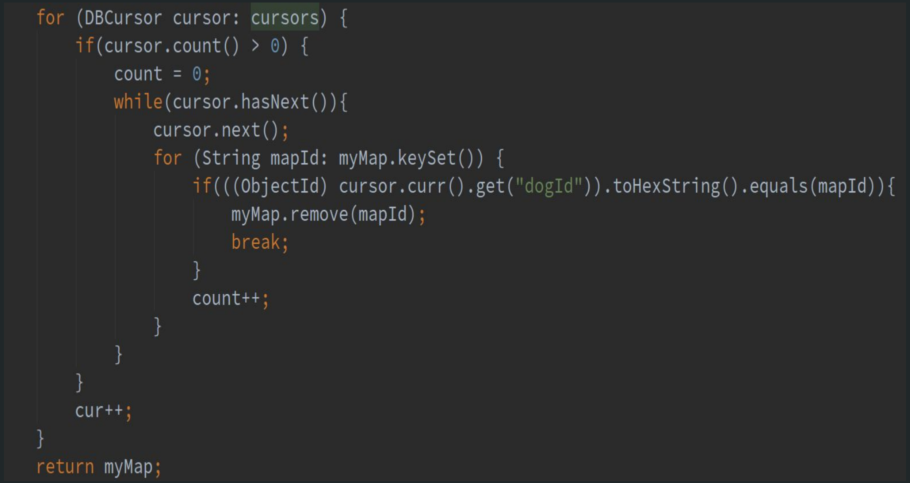
- When a user cancels a booking, cancelBooking() must undo what addBooking() did.



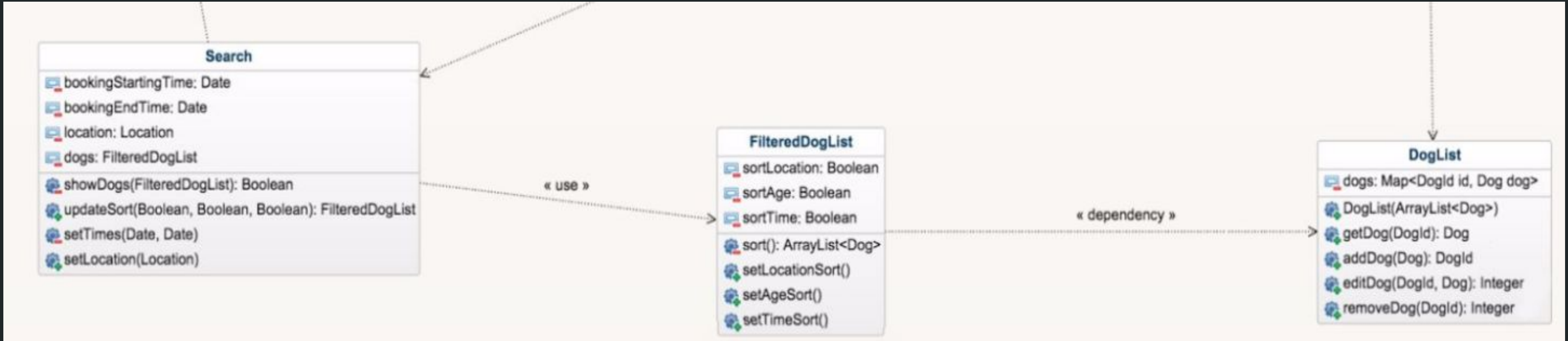


Design Pattern : Observer

- Filtered list of dogs when searched by time



```
for (DBCursor cursor: cursors) {
    if(cursor.count() > 0) {
        count = 0;
        while(cursor.hasNext()){
            cursor.next();
            for (String mapId: myMap.keySet()) {
                if(((ObjectId) cursor.curr().get("dogId")).toHexString().equals(mapId)){
                    myMap.remove(mapId);
                    break;
                }
                count++;
            }
        }
    }
    cur++;
}
return myMap;
```



Search

- bookingStartingTime: Date
- bookingEndTime: Date
- location: Location
- dogs: FilteredDogList
- showDogs(FilteredDogList): Boolean
- updateSort(Boolean, Boolean, Boolean): FilteredDogList
- setTimes(Date, Date)
- setLocation(Location)

« use »

FilteredDogList

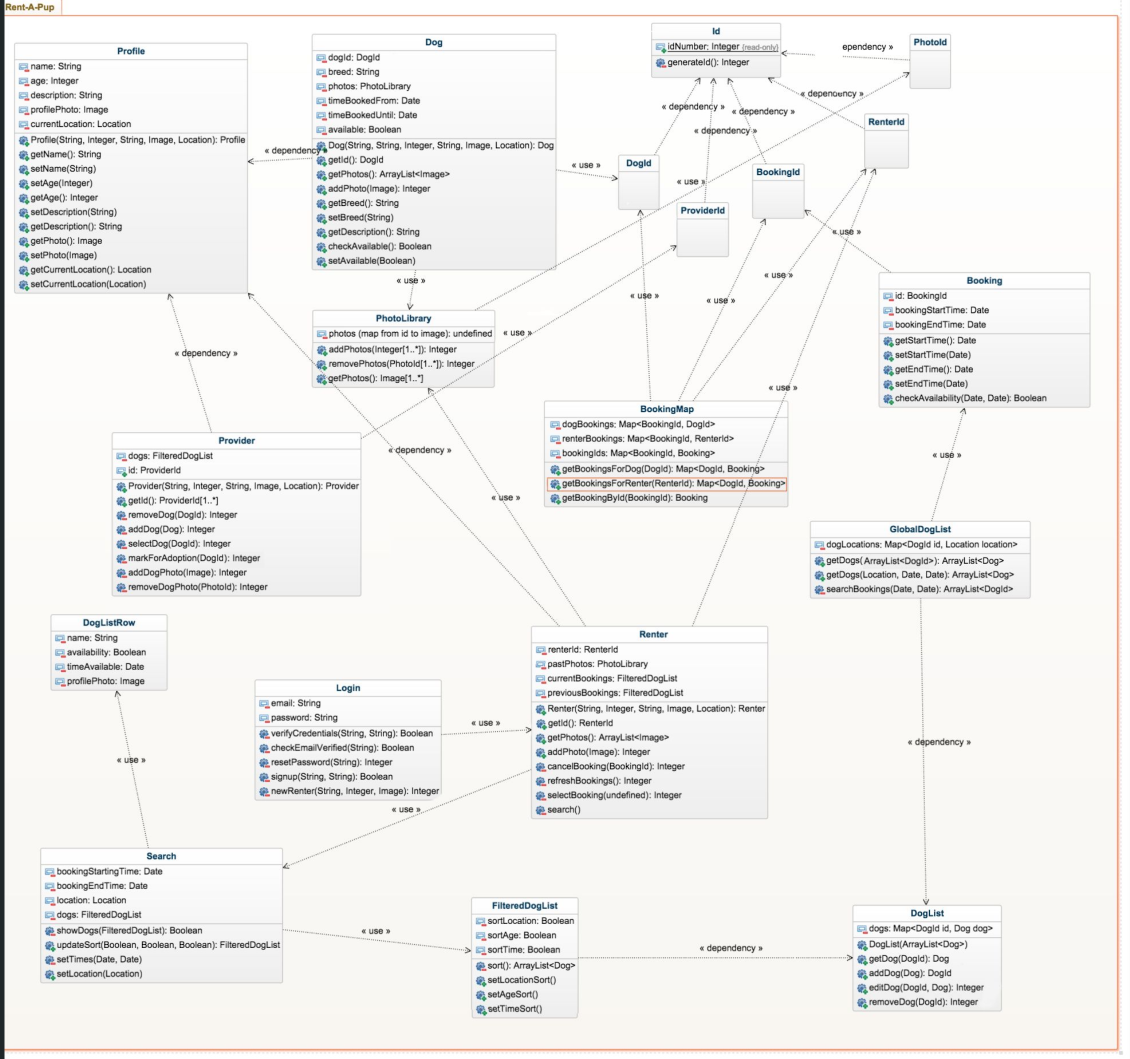
- sortLocation: Boolean
- sortAge: Boolean
- sortTime: Boolean
- sort(): ArrayList<Dog>
- setLocationSort()
- setAgeSort()
- setTimeSort()

« dependency »

DogList

- dogs: Map<DogId id, Dog dog>
- DogList(ArrayList<Dog>)
- getDog(DogId): Dog
- addDog(Dog): DogId
- editDog(DogId, Dog): Integer
- removeDog(DogId): Integer

Class Diagram



Id
- idNumber: Integer {read-only}
- generateId(): Integer

PhotoId

DogId

ProviderId

BookingId

RenterId

Profile
- name: String
- age: Integer
- description: String
- profilePhoto: Image
- currentLocation: Location
- Profile(String, Integer, String, Image, Location): Profile
- getName(): String
- setName(String)
- setAge(Integer)
- getAge(): Integer
- setDescription(String)
- getDescription(): String
- getPhoto(): Image
- setPhoto(Image)
- getCurrentLocation(): Location
- setCurrentLocation(Location)

Dog
- dogId: DogId
- breed: String
- photos: PhotoLibrary
- timeBookedFrom: Date
- timeBookedUntil: Date
- available: Boolean
- Dog(String, String, Integer, String, Image, Location): Dog
- getId(): DogId
- getPhotos(): ArrayList<Image>
- addPhoto(Image): Integer
- getBreed(): String
- setBreed(String)
- getDescription(): String
- checkAvailable(): Boolean
- setAvailable(Boolean)

PhotoLibrary
- photos (map from id to image): undefined
- addPhotos(Integer[1..*]): Integer
- removePhotos(PhotoId[1..*]): Integer
- getPhotos(): Image[1..*]

Booking
- id: BookingId
- bookingStartTime: Date
- bookingEndTime: Date
- getStartTime(): Date
- setStartTime(Date)
- getEndTime(): Date
- setEndTime(Date)
- checkAvailability(Date, Date): Boolean

BookingMap
- dogBookings: Map<BookingId, DogId>
- renterBookings: Map<BookingId, RenterId>
- bookingIds: Map<BookingId, Booking>
- getBookingsForDog(DogId): Map<DogId, Booking>
- getBookingsForRenter(RenterId): Map<DogId, Booking>
- getBookingById(BookingId): Booking

Provider
- dogs: FilteredDogList
- id: ProviderId
- Provider(String, Integer, String, Image, Location): Provider
- getId(): ProviderId[1..*]
- removeDog(DogId): Integer
- addDog(Dog): Integer
- selectDog(DogId): Integer
- markForAdoption(DogId): Integer
- addDogPhoto(Image): Integer
- removeDogPhoto(PhotoId): Integer

GlobalDogList
- dogLocations: Map<DogId id, Location location>
- getDogs(ArrayList<DogId>): ArrayList<Dog>
- getDogs(Location, Date, Date): ArrayList<Dog>
- searchBookings(Date, Date): ArrayList<DogId>

DogListRow
- name: String
- availability: Boolean
- timeAvailable: Date
- profilePhoto: Image

Renter
- renterId: RenterId
- pastPhotos: PhotoLibrary
- currentBookings: FilteredDogList
- previousBookings: FilteredDogList
- Renter(String, Integer, String, Image, Location): Renter
- getId(): RenterId
- getPhotos(): ArrayList<Image>
- addPhoto(Image): Integer
- cancelBooking(BookingId): Integer
- refreshBookings(): Integer
- selectBooking(undefined): Integer
- search()

Login
- email: String
- password: String
- verifyCredentials(String, String): Boolean
- checkEmailVerified(String): Boolean
- resetPassword(String): Integer
- signup(String, String): Boolean
- newRenter(String, Integer, Image): Integer

Search
- bookingStartingTime: Date
- bookingEndTime: Date
- location: Location
- dogs: FilteredDogList
- showDogs(FilteredDogList): Boolean
- updateSort(Boolean, Boolean, Boolean): FilteredDogList
- setTimes(Date, Date)
- setLocation(Location)

FilteredDogList
- sortLocation: Boolean
- sortAge: Boolean
- sortTime: Boolean
- sort(): ArrayList<Dog>
- setLocationSort()
- setAgeSort()
- setTimeSort()

DogList
- dogs: Map<DogId id, Dog dog>
- DogList(ArrayList<Dog>)
- getDog(DogId): Dog
- addDog(Dog): DogId
- editDog(DogId, Dog): Integer
- removeDog(DogId): Integer

Class Diagram Changes

- Including all of the arrow types
  - Cannot verify design patterns without inheritance, and association.
- Changing some arrows
  - Most associations aren’t explicit enough.
- Refine classes for mongoDB compatibility

Link to Demo:

https://youtu.be/1xinxctJq40