

# Rent-A-Pup

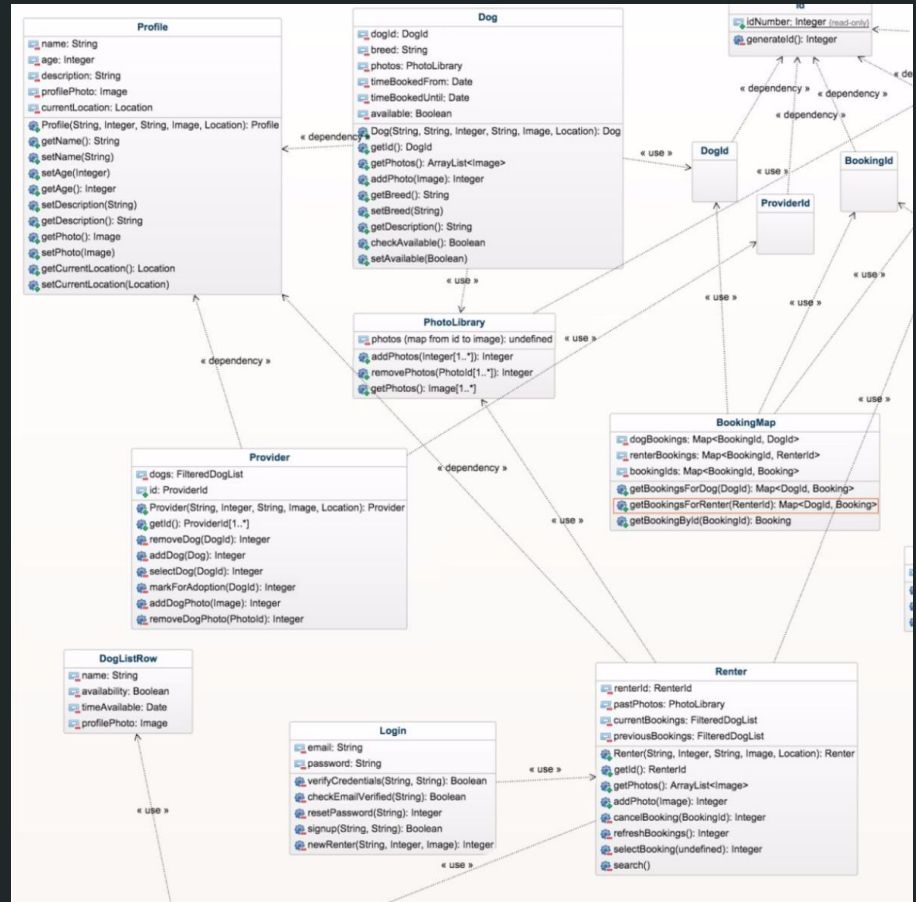
---

Elijah, Callie, Elias, Calvin

Demo

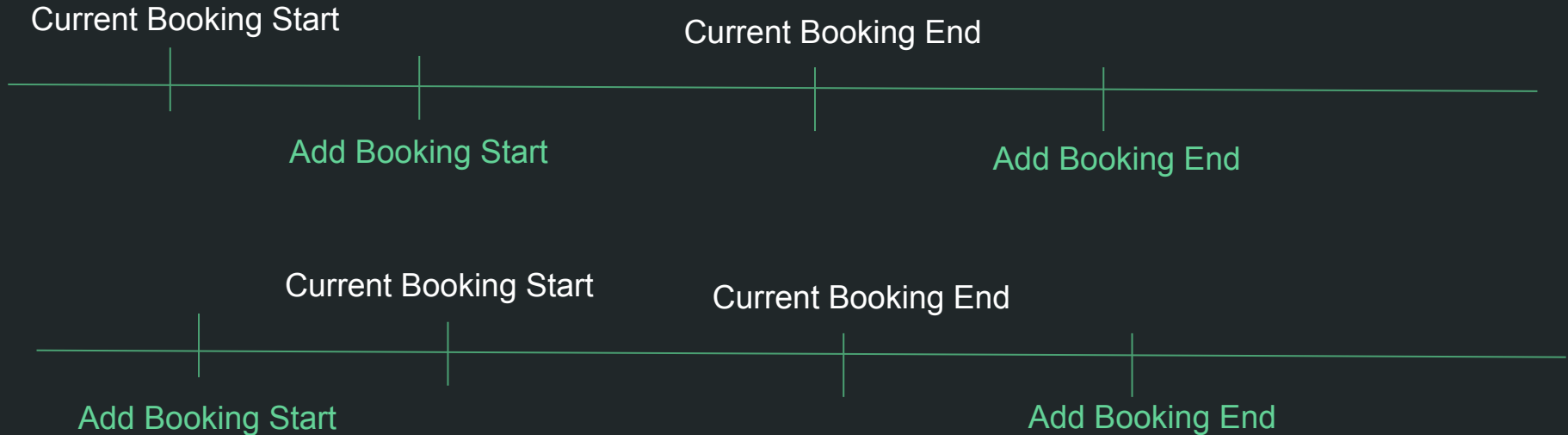
# Design Pattern : Template

- Profile superclass allows for code reuse
- Easy integration for future profile types



# Design Pattern : Command

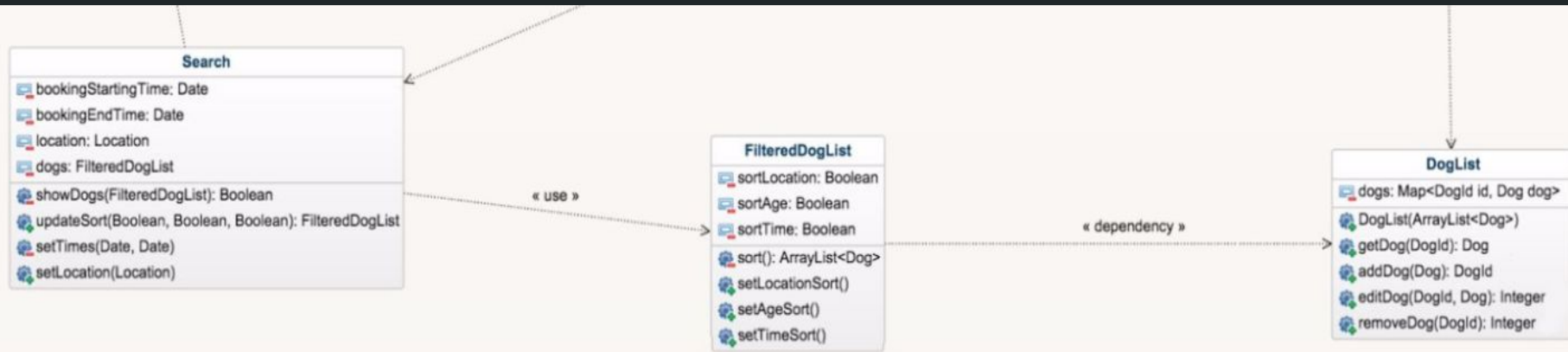
- When a user cancels a booking, `cancelBooking()` must undo what `addBooking()` did.



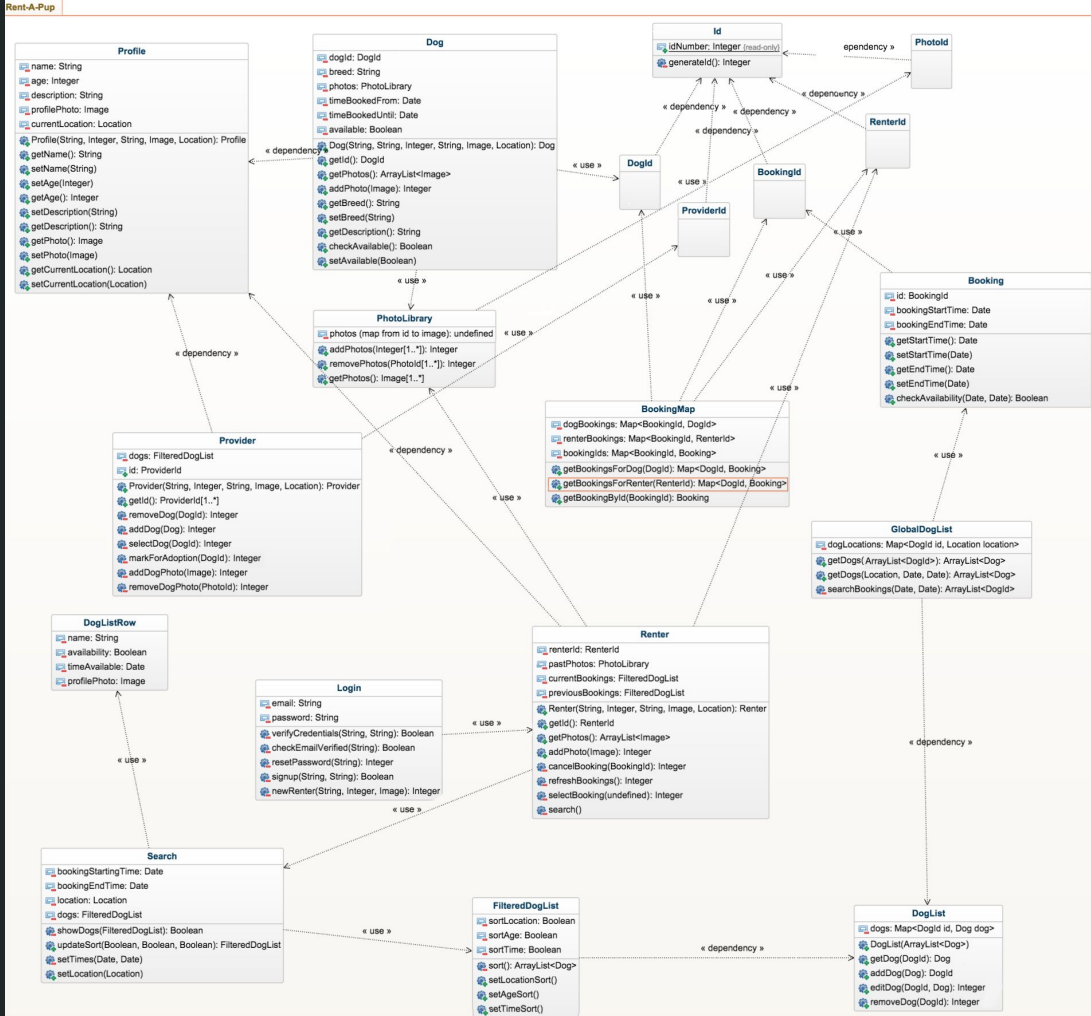
# Design Pattern : Observer

- 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;
```



# Class Diagram



# 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