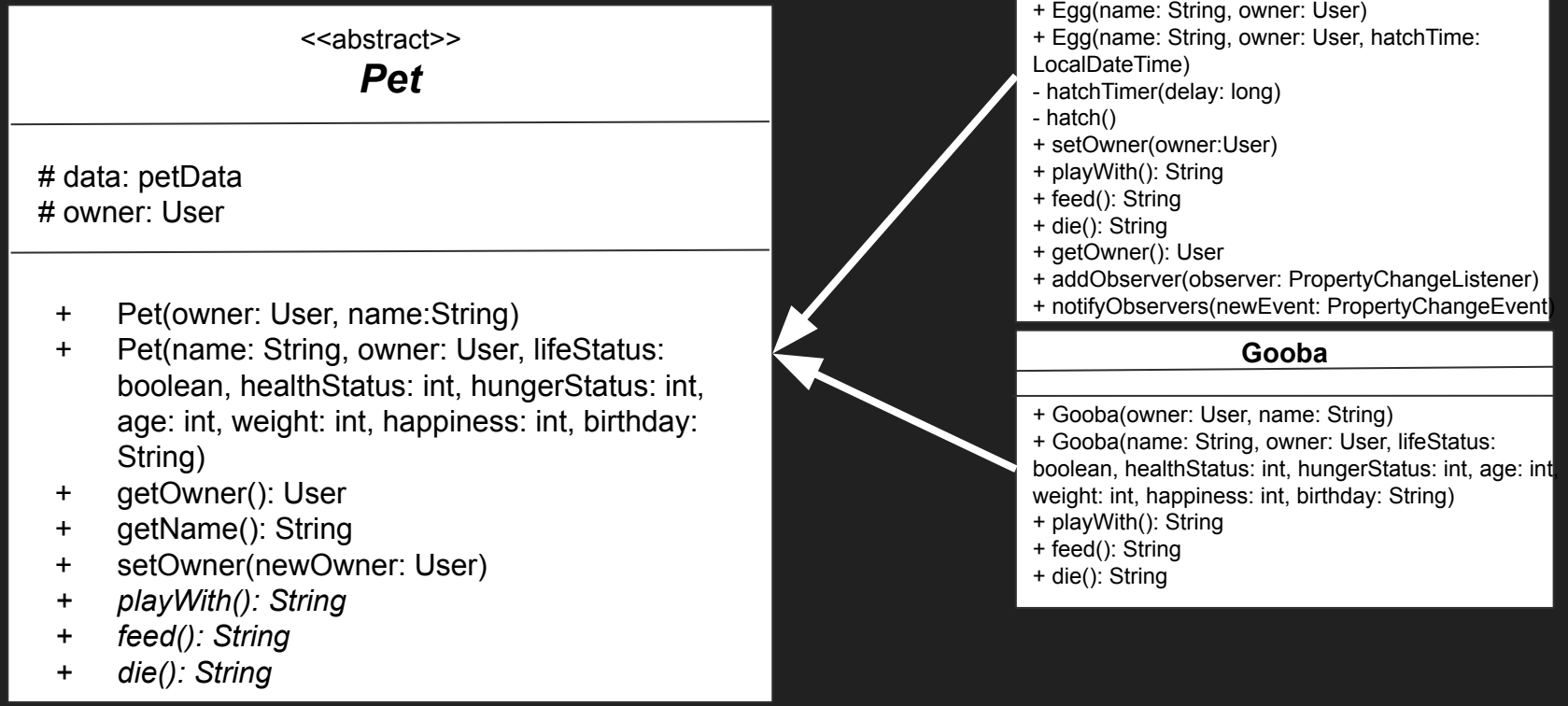
design2

PET SYSTEM


<<abstract>>
Pet
data: petData
owner: User
+ Pet(owner: User, name:String)
+ Pet(name: String, owner: User, lifeStatus: boolean, healthStatus: int, hungerStatus: int, age: int, weight: int, happiness: int, birthday: String)
+ getOwner(): User
+ getName(): String
+ setOwner(newOwner: User)
+ playWith(): String
+ feed(): String
+ die(): String
Egg
- hatchTime: LocalDateTime
- owner: User
- observable: PropertyChangeSupport
+ Egg(name: String, owner: User)
+ Egg(name: String, owner: User, hatchTime: LocalDateTime)
- hatchTimer(delay: long)
- hatch()
+ setOwner(owner:User)
+ playWith(): String
+ feed(): String
+ die(): String
+ getOwner(): User
+ addObserver(observer: PropertyChangeListener)
+ notifyObservers(newEvent: PropertyChangeEvent
Gooba
+ Gooba(owner: User, name: String)
+ Gooba(name: String, owner: User, lifeStatus: boolean, healthStatus: int, hungerStatus: int, age: int, weight: int, happiness: int, birthday: String)
+ playWith(): String
+ feed(): String
+ die(): String

PetData

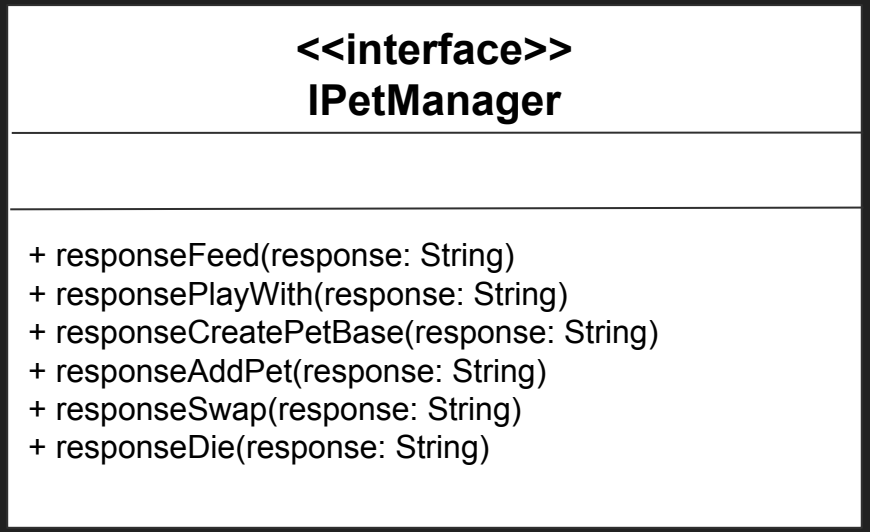
- name: String
lifeStatus: boolean
healthStatus: int
hungerStatus: int
age: int
happiness: int
foodResponse: int
playResponse: int
catchPhrase: String
birthday: String

+ PetData(name: String, owner: User, lifeStatus: boolean, healthStatus: int, hungerStatus: int, age: int, weight: int, happiness: int, birthday: String)
+ Getters and setters

PetController

- pm: PetManager
- presenter: IPresenter

+ PetController(pm: PetManager, presenter: IPresenter)
+ createEgg(name: String, owner: User): String
+ playWith(name: String)
+ feed(name: String)
+ die(name: String)
+ swapPets(currentUsername: String, otherUsername: String): boolean


<<interface>>
IPetManager
+ responseFeed(response: String)
+ responsePlayWith(response: String)
+ responseCreatePetBase(response: String)
+ responseAddPet(response: String)
+ responseSwap(response: String)
+ responseDie(response: String)

PetManager

- Pets: HashMap<String, Pet>
- ub: IUserBase
- am: AccountManager
- observable: PropertyChangeSupport

+ PetManager(am: AccountManager, ub: IUserBase)
+ createPetBase(object: Object, presenter: IPetManager)
+ createEgg(name: String, presenter: IPetManager, owner: User): boolean
+ die(name: String, presenter: IPetManager)
+ playWith(name: String, presenter: IPetManager)
+ feed(name: String, presenter: IPetManager)
+ getPet(name: String): Pet
+ swapPets(currentUsername: String, otherUsername: String, presenter: IPetManager): boolean
+ toString(): String
+ addObserver(observer: PropertyChangeListener)
+ notifyObservers(newEvent: PropertyChangeEvent)
+ updatePets()
+ propertyChange(evt: PropertyChangeEvent)
+ getData(): Object