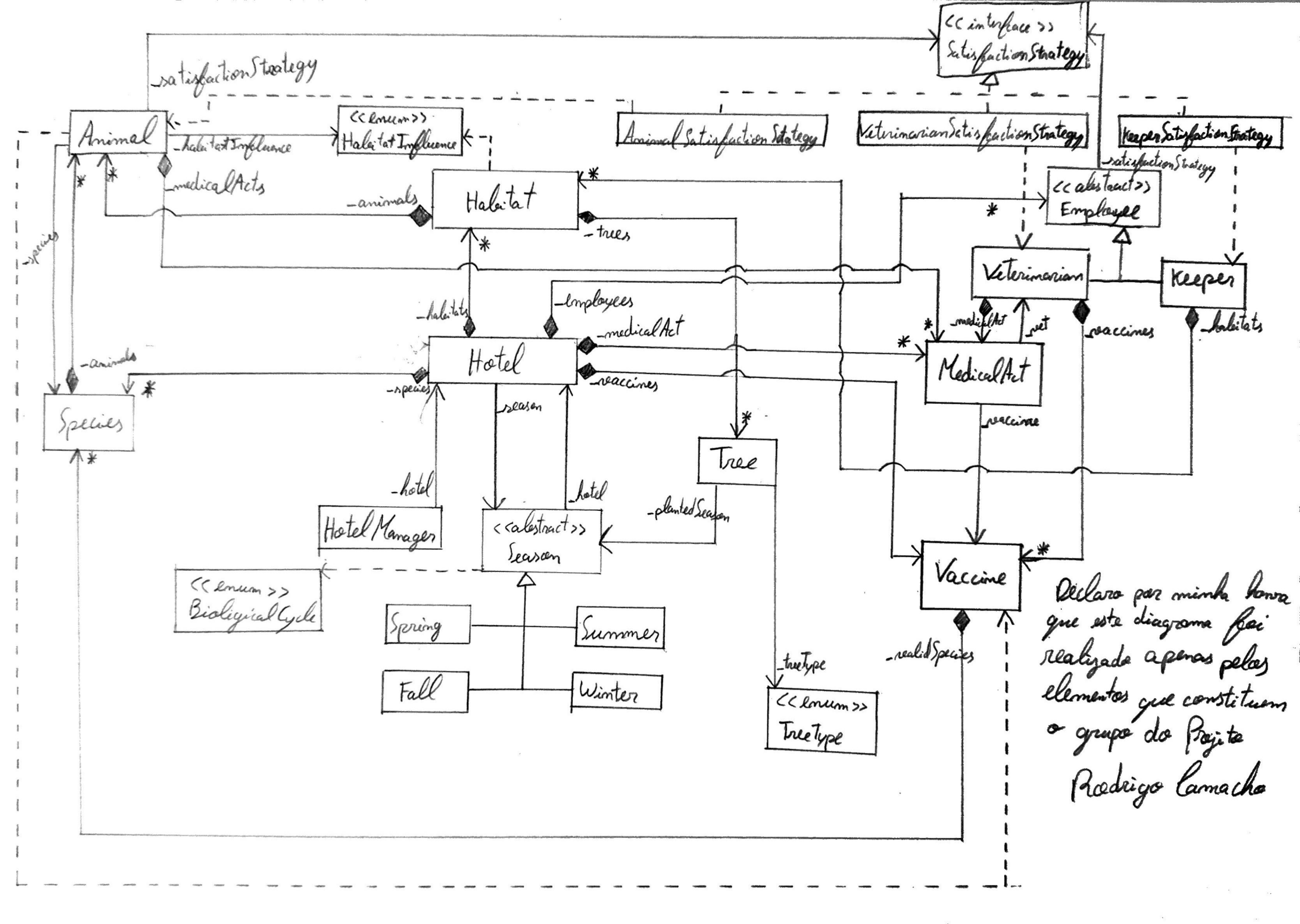
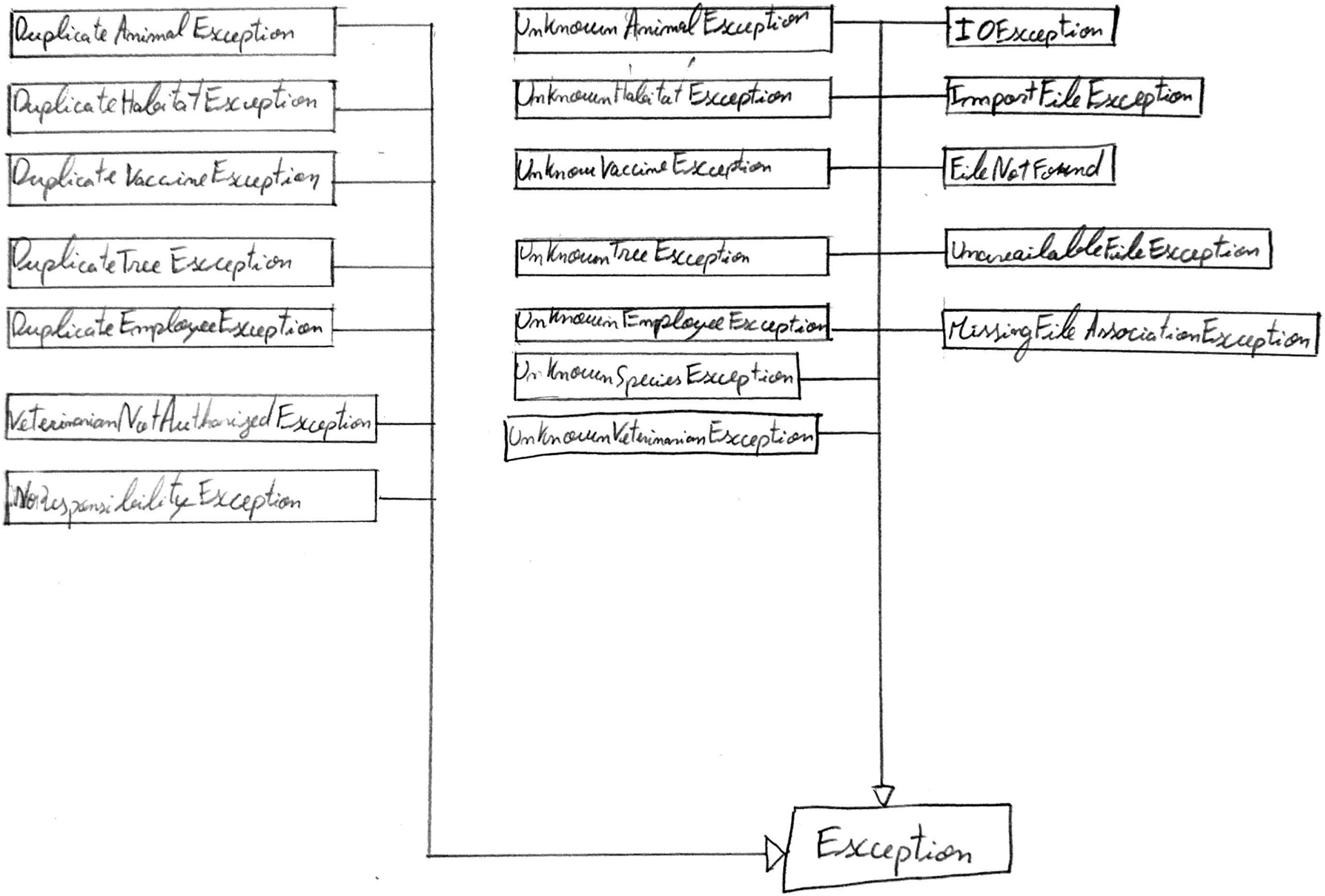

Animal
Habitat
Hotel
Species
Tree
Employee
Veterinarian
Keeper
MedicalAct
Vaccine
HotelManager
Season
Spring
Summer
Fall
Winter
TreeType
BiologicalCycle
HabitatInfluence
SatisfactionStrategy
AnimalSatisfactionStrategy
VeterinarianSatisfactionStrategy
KeeperSatisfactionStrategy
_satisfactionStrategy
_habitatInfluence
_medicalActs
_animals
_trees
_species
_habitats
_employees
_medicalAct
_vaccines
_season
_hotel
_plantedSeason
_treeType
_vet
_vaccine
_realidSpecies
Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo do Projeto
Rodrigo Camacho

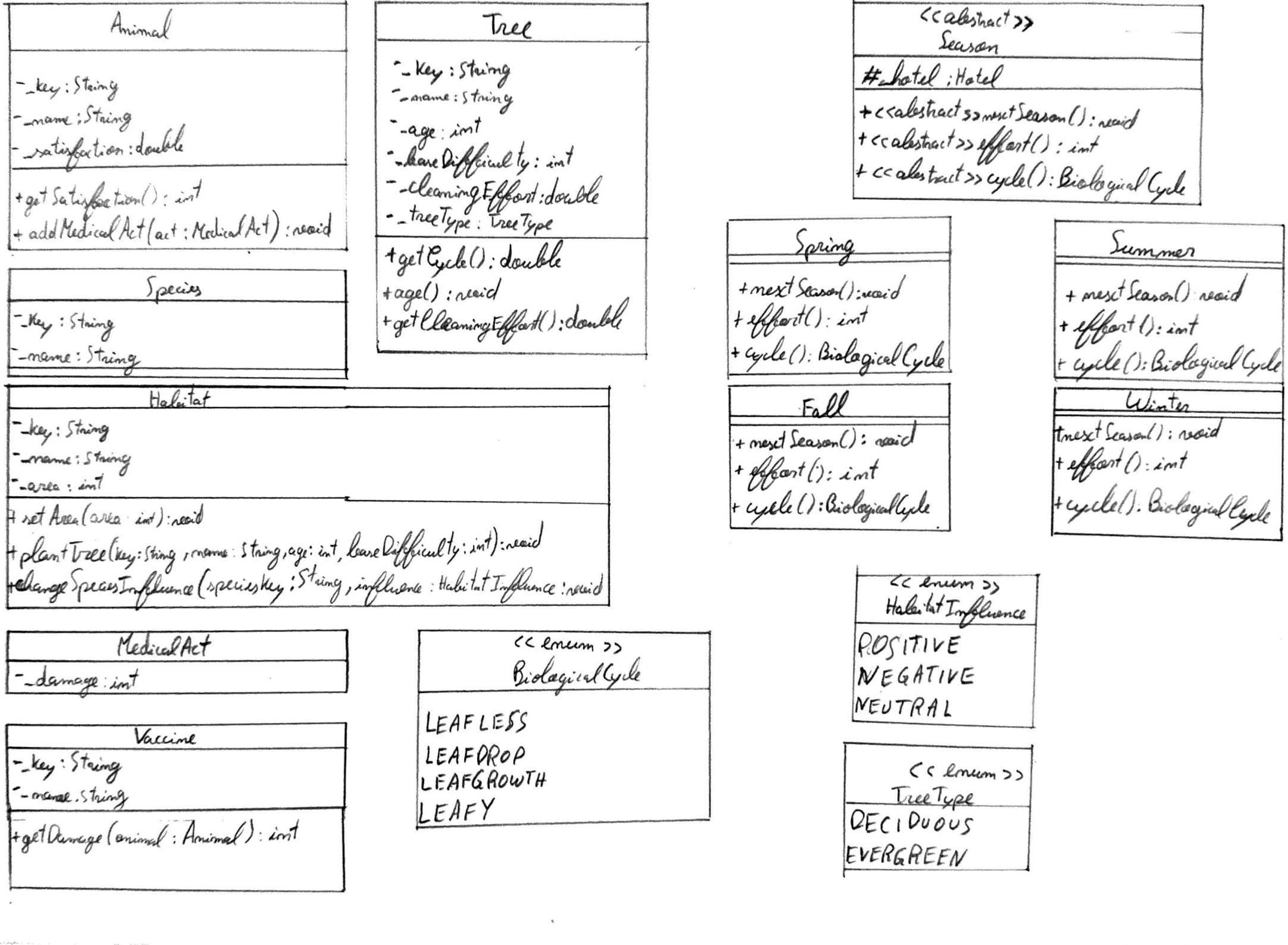
Hotel

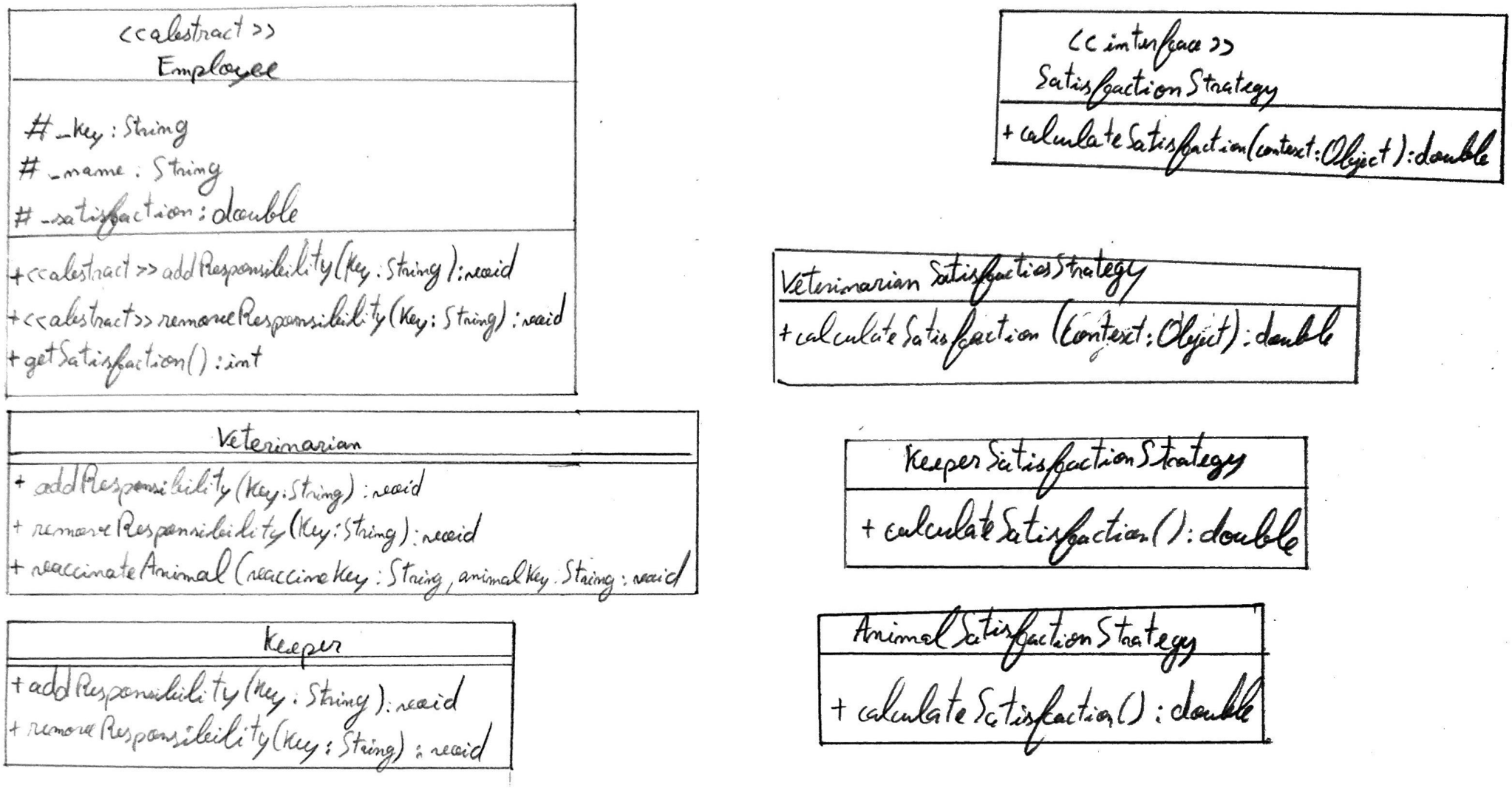
~ importFile(filename : String) : void
+ advanceSeason() : void
+ getGlobalSatisfaction() : int
+ getAllAnimals() : Collection<Animal>
+ registerSpecies(key : String, name : String) : void
+ registerAnimal(animalKey : String, name : String, speciesKey : String, habitatKey : String) : void
+ transferAnimal(animalKey : String, habitatKey : String) : void
+ getAllEmployees() : Collection<Employee>
+ registerVeterinarian(key : String, name : String) : void
+ registerKeeper(key : String, name : String) : void
+ registerHabitat(key : String, area : int) : void
+ getAllTreesInHabitat(key : String) : Collection<Tree>
+ getAllVaccines() : Collection<Vaccine>
+ registerVaccine(key : String, name : String, validSpeciesKeys : String[])
+ getAllMedicalActs() : Collection<MedicalAct>
+ lookupAnimalsByHabitat(habitatKey : String) : Collection<Animal>
+ lookupMedicalActsByAnimal(animalKey : String) : Collection<MedicalAct>
+ lookupMedicalActsByVeterinarian(veterinarianKey : String) : Collection<MedicalAct>
+ lookupWrongMedicalActs() : Collection<MedicalAct>
+ newMedicalAct(vaccineKey : String, veterinarianKey : String, animalKey : String) : void

HotelManager

+ save() : void
+ saveAs(filename : String) : void
+ load(filename : String) : void
+ importFile(filename : String) : void
+ getHotel() : Hotel


DuplicateAnimalException
DuplicateHabitatException
DuplicateVaccineException
DuplicateTreeException
DuplicateEmployeeException
VeterinarianNotAuthorizedException
NoResponsibilityException
UnknownAnimalException
UnknownHabitatException
UnknownVaccineException
UnknownTreeException
UnknownEmployeeException
UnknownSpeciesException
UnknownVeterinarianException
IOException
ImportFileException
FileNotFound
UnavailableFileException
MissingFileAssociationException
Exception


Animal
-_key : String
-_name : String
-_satisfaction : double
+getSatisfaction() : int
+addMedicalAct(act : MedicalAct) : void
Species
-_key : String
-_name : String
Tree
-_key : String
-_name : String
-_age : int
-_leaveDifficulty : int
-_cleaningEffort : double
-_treeType : TreeType
+getCycle() : double
+age() : void
+getCleaningEffort() : double
<<abstract>>
Season
#_hotel : Hotel
+<<abstract>>nextSeason() : void
+<<abstract>>effort() : int
+<<abstract>>cycle() : BiologicalCycle
Spring
+nextSeason() : void
+effort() : int
+cycle() : BiologicalCycle
Summer
+nextSeason() : void
+effort() : int
+cycle() : BiologicalCycle
Fall
+nextSeason() : void
+effort() : int
+cycle() : BiologicalCycle
Winter
+nextSeason() : void
+effort() : int
+cycle() : BiologicalCycle
Habitat
-_key : String
-_name : String
-_area : int
+setArea(area : int) : void
+plantTree(key : String, name : String, age : int, leaveDifficulty : int) : void
+changeSpecesInfluence(speciesKey : String, influence : HabitatInfluence) : void
MedicalAct
-_damage : int
Vaccine
-_key : String
-_name : String
+getDamage(animal : Animal) : int
<<enum>>
BiologicalCycle
LEAFLESS
LEAFDROP
LEAFGROWTH
LEAFY
<<enum>>
HabitatInfluence
POSITIVE
NEGATIVE
NEUTRAL
<<enum>>
TreeType
DECIDUOUS
EVERGREEN


<<abstract>>
Employee
_key : String
_name : String
_satisfaction : double
+ <<abstract>> addResponsibility(key : String) : void
+ <<abstract>> removeResponsibility(key : String) : void
+ getSatisfaction() : int
<<interface>>
SatisfactionStrategy
+ calculateSatisfaction(context : Object) : double
VeterinarianSatisfactionStrategy
+ calculateSatisfaction(context : Object) : double
Veterinarian
+ addResponsibility(key : String) : void
+ removeResponsibility(key : String) : void
+ reaccinateAnimal(reaccineKey : String, animalKey : String) : void
KeeperSatisfactionStrategy
+ calculateSatisfaction() : double
Keeper
+ addResponsibility(key : String) : void
+ removeResponsibility(key : String) : void
AnimalSatisfactionStrategy
+ calculateSatisfaction() : double