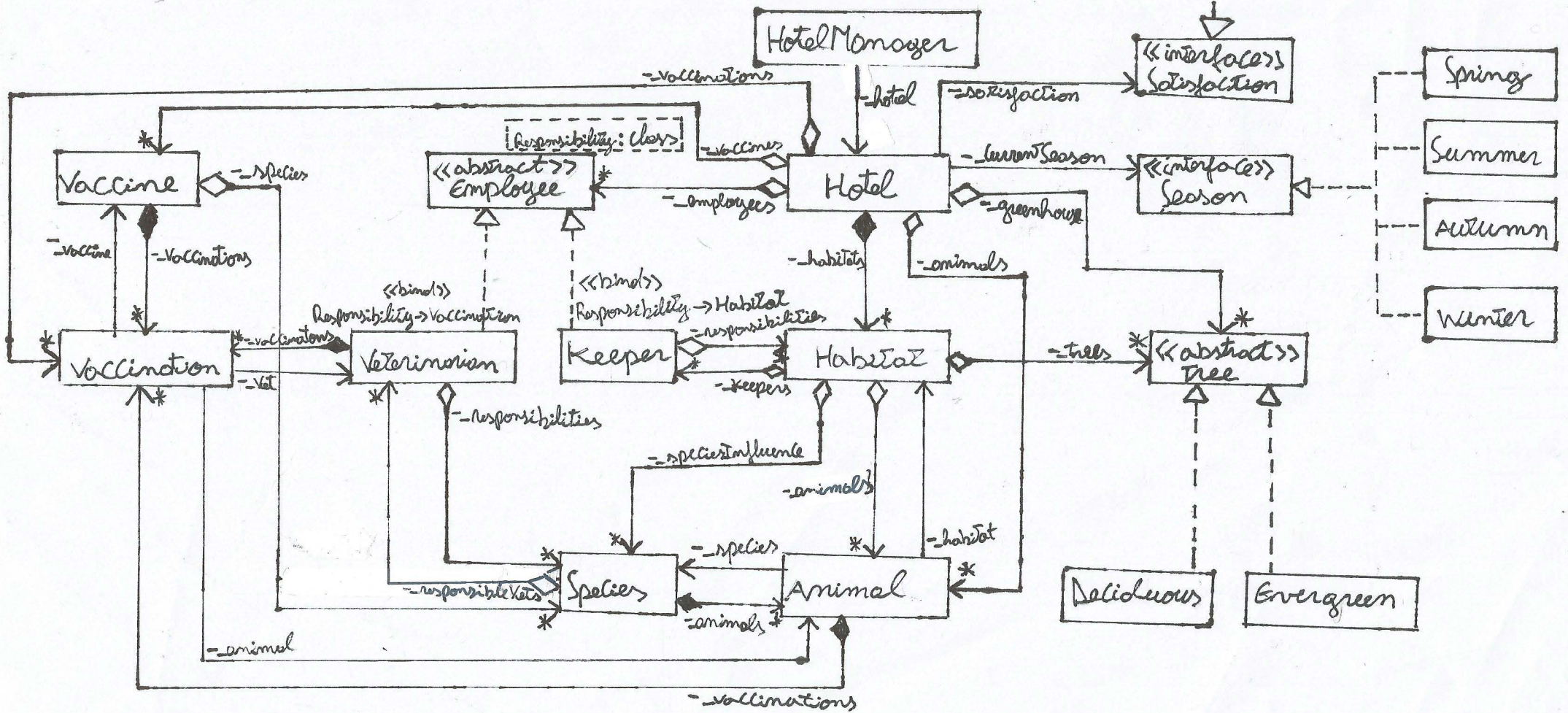
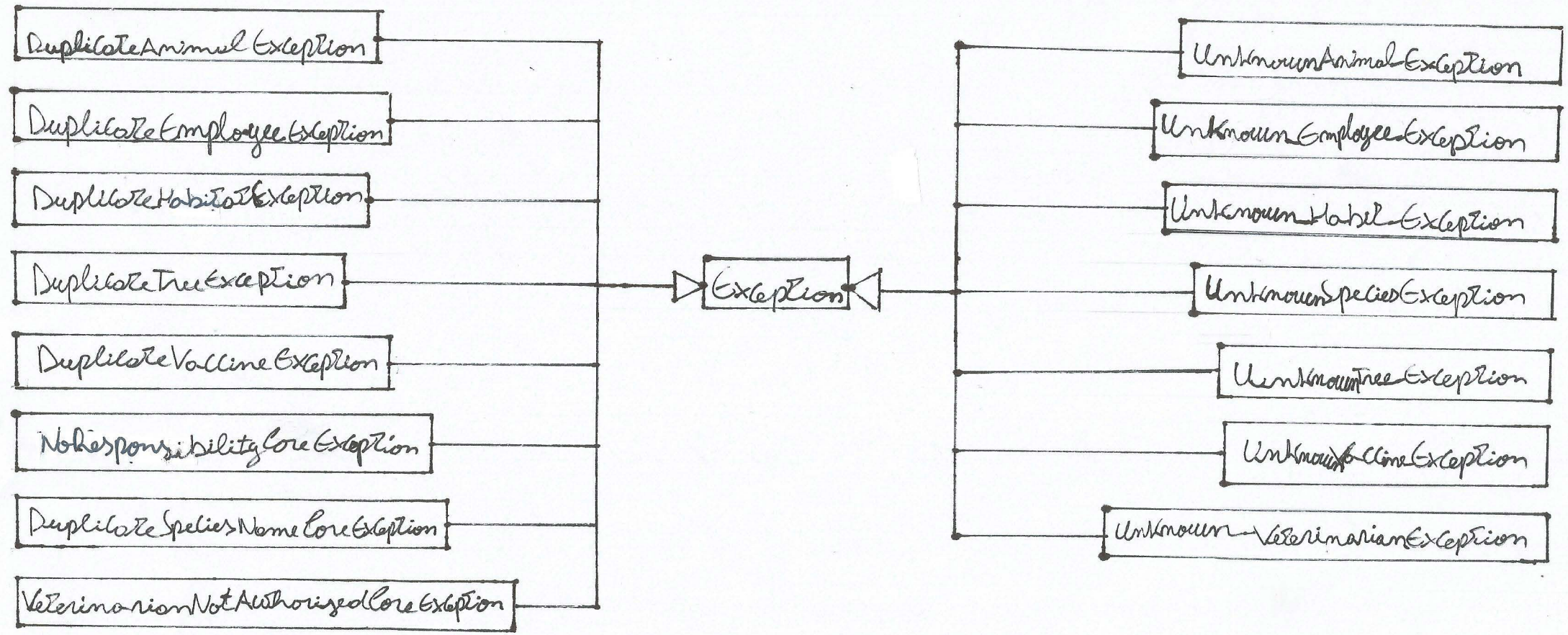


Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto. *Guilherme*
 Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto. *Ema Ferrão*

All classes implement Serializable except exceptions, Satisfaction, Season and their implementations. Class descriptions and exceptions are represented in other pages. Methods "equals" and "toString" have been omitted since they are inherited from Object. Getters and setters have also been omitted.



Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo do projeto.
Ema Ferrão
Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo do projeto.
Aguilhera



Hotel

```

- <<final>> serialVersionUID: Long
- -animals: Map<String, Animal>
- -habitats: Map<String, Habitat>
- -greenhouse: Map<String, Trees>
- -employees: Map<String, Employee>
- -vaccines: Map<String, Vaccine>
- -vaccinations: Collection<Vaccination>
- -currentSeason: Season
- -satisfaction: Satisfaction

```

```

+ importFile(filename: String): void
+ showAllAnimals(): String
+ registerAnimal(idAnimal: String, name: String, idSpecies: String, idHabitat: String): void
+ transferAnimalToHabitat(idAnimal: String, idHabitat: String)
+ showSatisfactionOfAnimal(idAnimal: String): int
+ showAllEmployees(): String
+ registerEmployee(idEmployee: String, name: String, employeeType: String): void
+ addResponsibility(idEmployee: String, idResponsibility: String): void
+ removeResponsibility(idEmployee: String, idResponsibility: String): void
+ showSatisfactionOfEmployee(idEmployee: String): int
+ showAllHabitats(): String
+ registerHabitat(idHabitat: String, name: String, area: int): void
+ changeHabitatArea(idHabitat: String, area: int): void
+ changeHabitatInfluence(idHabitat: String, idSpecies: String, influence: String): void
+ addTreeToHabitat(idHabitat: String, idTree: String, name: String, age: int,
    bloomFrequency: int, type: String): void
+ makeAgeTrees(): void
+ showAllTrees(): void
+ showAllTreesInHabitat(idHabitat: String): void
+ showAllVaccines(): String
+ registerVaccine(idVaccine: String, name: String, idSpeciesList: String[]): void
+ vaccinateAnimal(idVaccine: String, idVet: String, idAnimal: String): void
+ showVaccinations(): String
+ showAnimalsInHabitat(idHabitat: String): String
+ showMedicalAidsOnAnimal(idAnimal: String): String
+ showMedicalAidsByVeterinarian(idVet: String): String
+ showWrongVaccinations(): String
+ searchAnimal(idAnimal: String): Animal
+ searchHabitat(idHabitat: String): Habitat
+ searchEmployee(idEmployee: String): Employee
+ searchVaccines(idVaccines: String): Vaccine
+ nextSeason(): int
+ animalExists(idAnimal: String): boolean
+ habitatExists(idVaccines: String): boolean
+ employeeExists(idEmployee: String): boolean
+ treeExists(idTree: String): boolean
+ vaccineExists(idSpecies: String): boolean
+ duplicateSpeciesName(speciesName: String): boolean
+ noResponsibility(idResponsibility: String): boolean

```

Dado por minha banca que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto. Eu não sei.

Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto: equipe
Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto: Emc Ferrão

Habitat

- <<final>> - id Habitat: String
- name: String
- area: int
- trees: Collection < Tree >
- keepers: Collection < Keeper >
- animals: Collection < Animal >
- speciesInfluence: Map < Species, String >

- + Habitat (id Habitat: String, name: String, area: int)
- + add Tree (tree: Tree): String
- + add Animal (animal: Animal): void
- + remove Animal (animal: Animal): void
- + change Area (area: int): void
- + change Influence (species: Species, influence: String): void
- + total Animals(): int
- + total Animal Of Species (species: Species): int
- + total Trees(): int
- + total Keepers(): int
- + work In Habitat (season: Season): double
- + same Species (animal: Animal): int
- + adequation (animal: Animal): int
- + show Habitat(): String

<<abstract>> Tree

- <<final>> - id Tree: String
- name: String
- age: int
- base Clean Difficulty: double

- + Tree (id Tree: String, name: String, base Clean Difficulty: double)
- + biologic Cycle (season: Season): String
- + age(): int
- + seasonal Effort (season: Season): int
- + cleaning Effort (season: Season): double
- + show Tree(): String

Deciduous

Evergreen

Animal

- <<final>> - id Animal: String
- name: String
- species: Species
- habitat: Habitat
- vaccinations: Collection <Vaccination>

+ Animal(id Animal: String, name: String, species: Species, habitat: Habitat)
+ vaccinate Animal (vaccination: Vaccination): void
+ show Health History(): String
+ transfer To Habitat (habitat: Habitat): void
+ show Animal(): String

Species

- <<final>> - id Species: String
- name: String
- animals: Collection <Animal>
- responsible Vets: Collection <Veterinarian>

+ Species(id Species: String, name: String)
+ add Animal (animal: Animal): void
+ total Animals(): int
+ add Responsible Vet (vet: Veterinarian): void
+ remove Responsibility Vet (vet: Veterinarian): void
+ total Responsible Vets(): int

<<abstract>> Employee

Responsibility: Class

- <<final>> - id Employee: String
- name: String
- responsibilities: Collection <Responsibility>

+ Employee(id Employee: String, name: String)
+ add Responsibility (id Responsibility: String): void
+ remove Responsibility (id Responsibility: String): void
+ work(): double
+ show Employee(): String

Veterinarian

- vaccinations: Collection <Vaccination>

+ add Vaccination (vaccination: Vaccination): void
+ is Authorized (species: Species): boolean

Keeper

<<interface>> Satisfaction

+ get Satisfaction(): int

Animal Satisfaction

Vet Satisfaction

Keeper Satisfaction

<<interface>> Season

+ next (hotel: Hotel): int

Spring

Summer

Autumn

Winter

Vaccine

- @@final @@ - id Vaccine: String
- name: String
- species: Collection < Species >
- vaccinations: Collection < Vaccination >

- + Vaccine(id Vaccine: String, name: String, species: Collection < Species >)
- + damage(Animal: Animal): int
- + namesSize(s1: Species, s2: Species): int
- + commonChars(s1: Species, s2: Species): int
- + incVaccinationsGiven(): void
- + isAdequate(Animal: Animal): boolean
- + vaccinateAnimal(Animal: Animal, vet: Veterinarian): Vaccination
- + showVaccine(): String

Vaccination

- vaccine: Vaccine
- animal: Animal
- vet: Veterinarian
- damage: int

- + Vaccination(vaccine: Vaccine, animal: Animal, vet: Veterinarian, damage: int)
- + showVaccination(): String

Hotel Manager

- hotel: Hotel

- + save(): void
- + saveAs(filename: String): void
- + load(filename: String): void
- + importFile(filename: String): void
- + advanceSeason(): int
- + showGlobalSatisfaction(): int