Class Diagram Documentation

Task 1

Class descriptions

Class name	Description
Employee	Interface to describe Clyde conservation employees
Administrator	Implements Employee Interface. Will have functions to add cages, animals or keepers
Keeper	Implements Employee Interface. Will be used as a parent class for the two types of Keepers.
HeadKeeper	Extends Keeper class. Have methods to allocate animals to cages and allocate cages to assistant Keepers.
AssistantKeeper	Extends Keeper class. Are allocated cages to care for
AllocationsCollection	This will contain the Allocation tables and provide some search/display functions
EmployeeRoster	Will contains the employees of the charity
Animalcollection	Will contains the animals owned by the charity
AllocationTable	An allocation table will contain a Keeper and all the cages allocated to him.
Animal	Will hold the details of an animal
Cage	Container for animals, comes in 3 sizes: L, M, S

Object Oriented Design H172 35

Task 2

Menu Interface:

- **Selecting a profile:**

```
----- Welcome to Clyde Conservation System -----

Please select an option
1) Head Keeper Application
2) Administrator Application
3) Exit
-->
```

- **Administrator Menu:**

```
----- Welcome to Clyde Conservation System -----

Please select an option
1) Head Keeper Application
2) Administrator Application
3) Exit
-->
2

Please select an option
1) Add an Animal
2) Add a Cage
3) Add a Keeper
4) Return
```

- **Head keeper Menu:**

```
----- Welcome to Clyde Conservation System -----

Please select an option
1) Head Keeper Application
2) Administrator Application
3) Exit
-->
1

Please select an option
1) Create an assigment
2) Assign an Animal to a Cage
3) Assign a cage to a Keeper
4) Display the existing Assignments
5) Return
-->
```

Object Oriented Design H172 35

Screen Interfaces (include 2):

Headkeepers options

```
Please select an option
1) Create an assigment
2) Assign an Animal to a Cage
3) Assign a cage to a Keeper
4) Display the existing Assignments
5) Return
-->
2
-------- Animal assignment --------
***** Animal List *****

Type: Bird
Name: Robin
Category: Prey
Sex: Male
Date of Birth: 02/02/2021
Date of Acquisition: 05/01/2021
Animal ID: 200
```

Admin adding an animal:

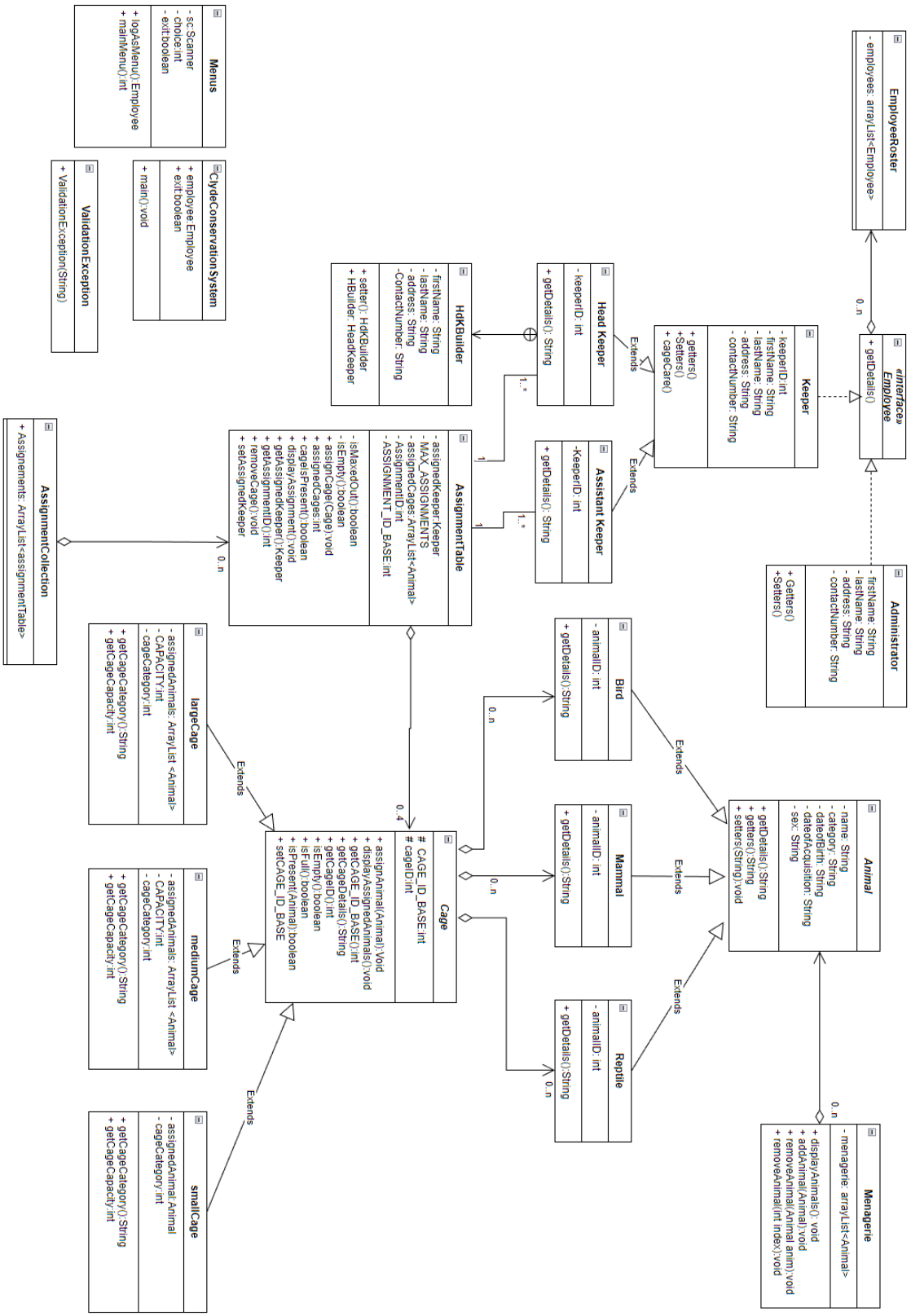
```
----- Welcome to Clyde Conservation System -----

Please select an option
1) Head Keeper Application
2) Administrator Application
3) Exit
-->
2

Please select an option
1) Add an Animal
2) Add a Cage
3) Add a Keeper
4) Return
-->
1
----- Add an Animal -----
Please select an option
1) Add a Bird
2) Add a Mammal
3) Add a Reptile
4) Return
-->
1

Please enter the animal's name
(Must contains at least one character)
-->
```

Object Oriented Design H172 35

Task 3 (a and b)

Class diagram

See the attached drawio file on canvas for a better view.



Task 3 (c)

Data dictionary of 4 classes

Class name	Description (attributes and Operations)
Cage	**Attributes:** - cageID: Int - cageCategory: String - CAGE_ID_BASE: Int **Methods:** - getCageCategory() - getCAGE_ID_BASE() - setCAGE_ID_BASE(): Int - getCageID(): Int - isEmpty(): Boolean - assignAnimal(Animal) - isFull(): Boolean - getCageDetails() - isPresent() - displayAssignedAnimals() - getCageSize()
AssignmentTable	**Attributes**: - assignmentID: Int - MAX_ASSIGNMENTS: Int - assignedKeeper: Keeper - assignedCages: ArrayList<Cage> - ASSIGNMENT_ID_BASE: Int **Methods**: - isMaxedOut: Boolean - getAssignedKeeper: Keeper - getAssigmentID: Int - assignCage(Cage): Boolean - removeCage(): Boolean - assignedCages(): Int - isEmpty(): Boolean - cageIsPresent(Cage): Boolean - setAssignedKeeper(Keeper): void - displayAssignment(): void
Keeper	**Attributes:** - contactNumber: String - keeperID: Int

	- firstName: String - address: String - lastName: String - KEEPER_ID_BASE: int **Methods**: - getLastName(): String - getContactNumber(): String - getName(): String - gutted(): int - getDetails(): String - setLastName(String): void - setAddress(String): void - getFirstName(): String - cageCare(): void - setFirstName(String): void - getAddress(): String - getSurname(): String - getKeeperID(): Int - setContactNumberString): void
Menagerie	**Attributes**: - Filename: String - Menagerie: ArrayList<Animal> **Methods**: - saveMenagerie(): void - getAnimal(Int): Animal - countUnassignedAnimals(): Int - addAnimal(Animal): void - removeanimal(String): void - displayAllAnimals(): void - displayUnassignedAnimals(): void - removeAnimal(int): void - isPresent(Animal): Boolean