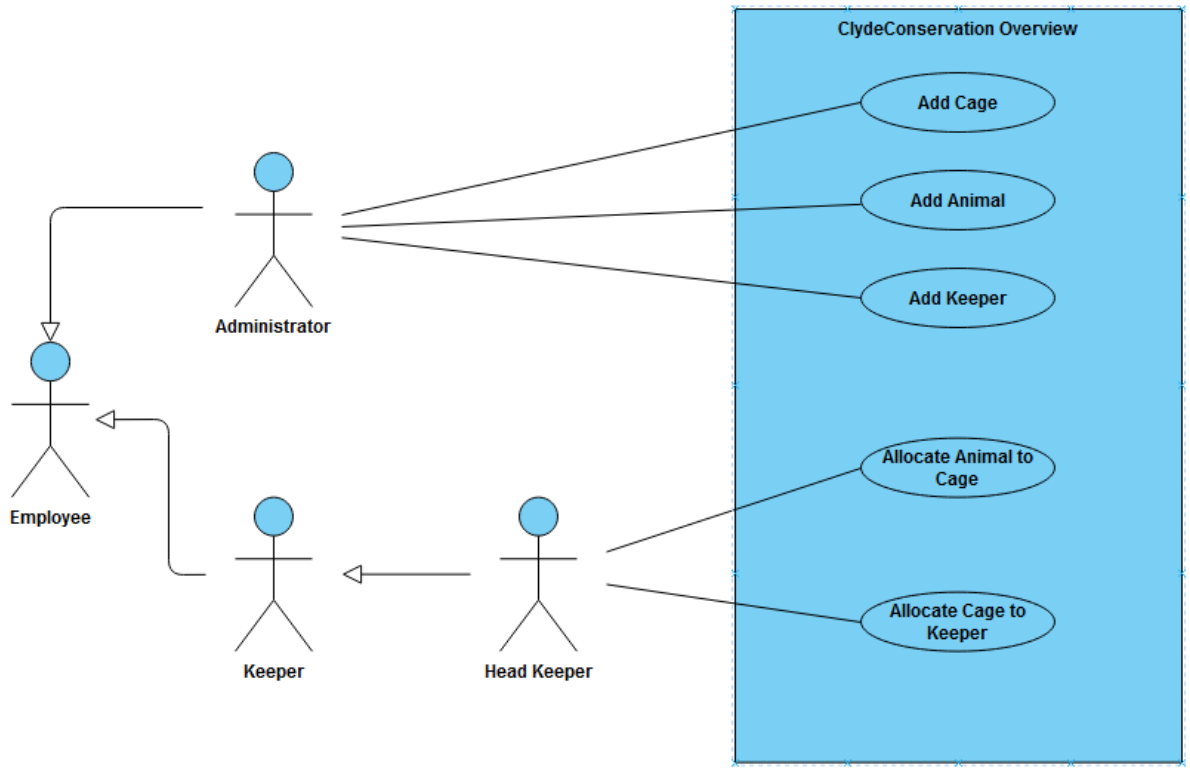
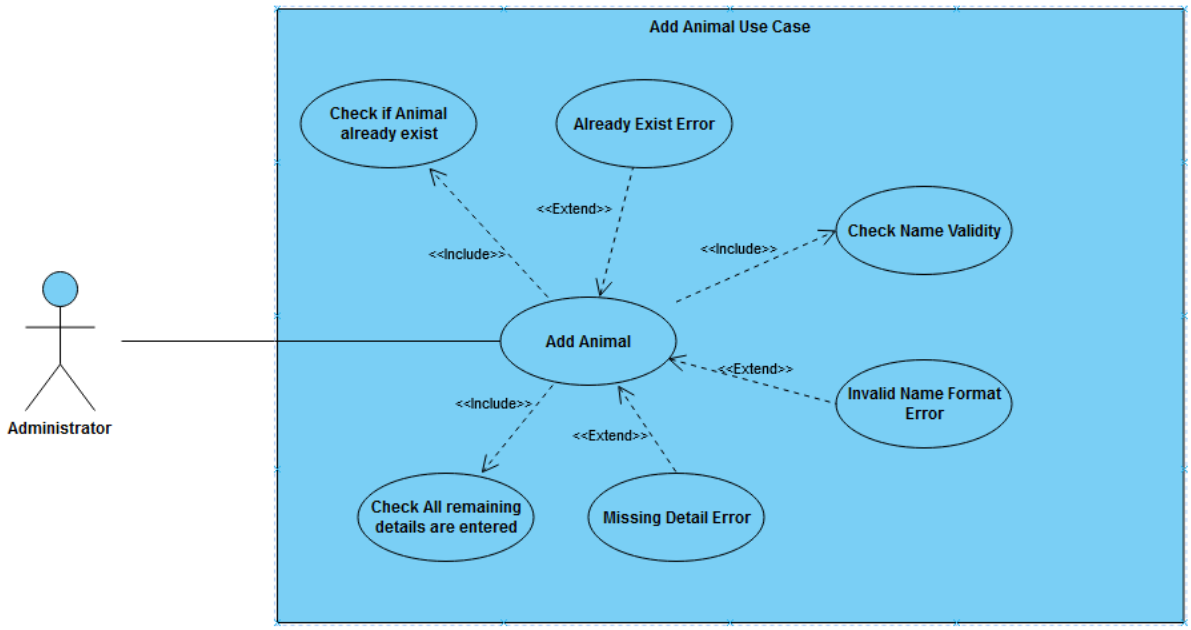
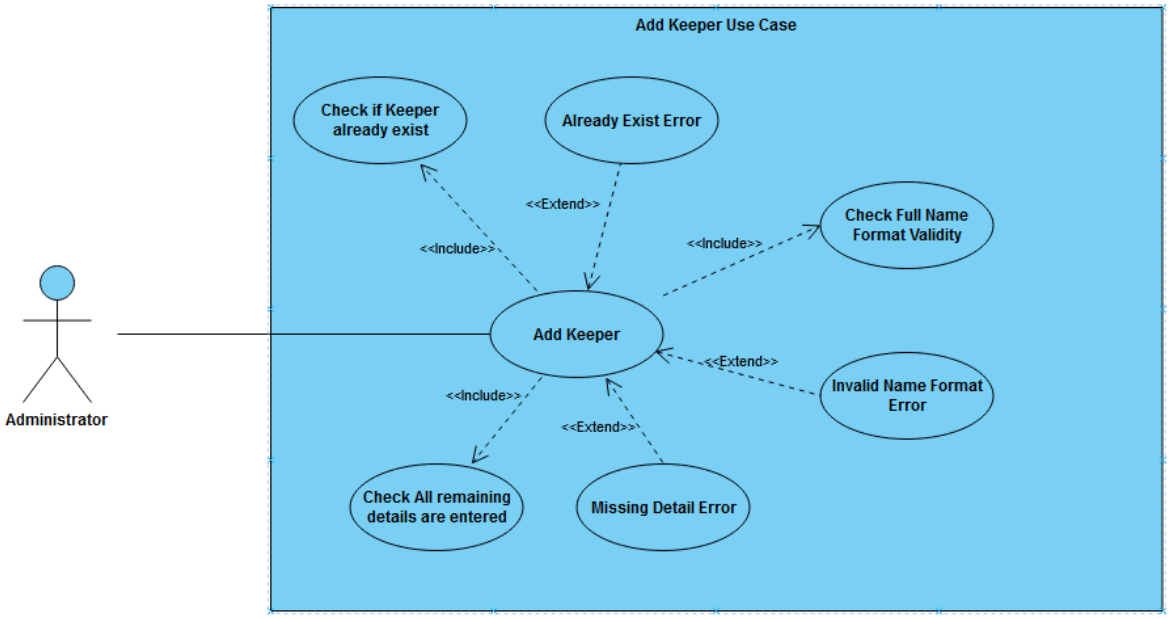
Use Case Documentation

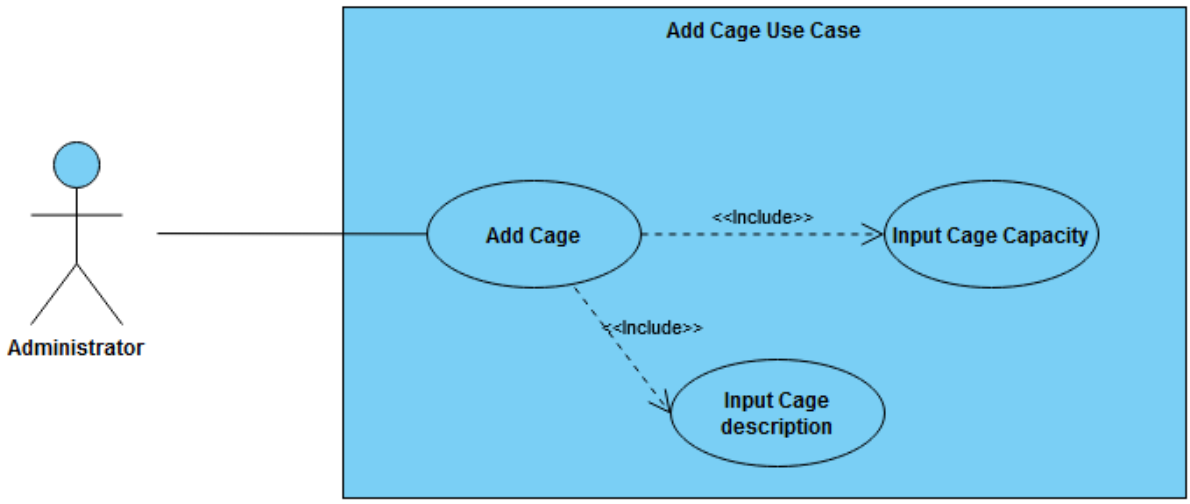
Task One – copy your detailed use case diagram into the space below.

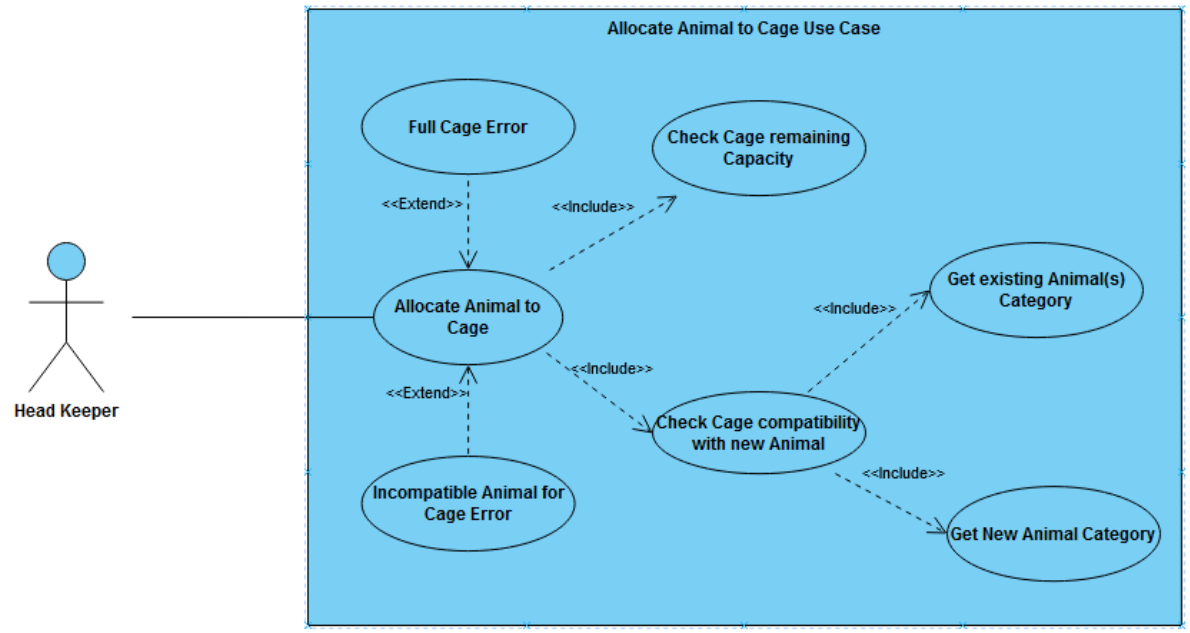
Detailed Use Case diagram:

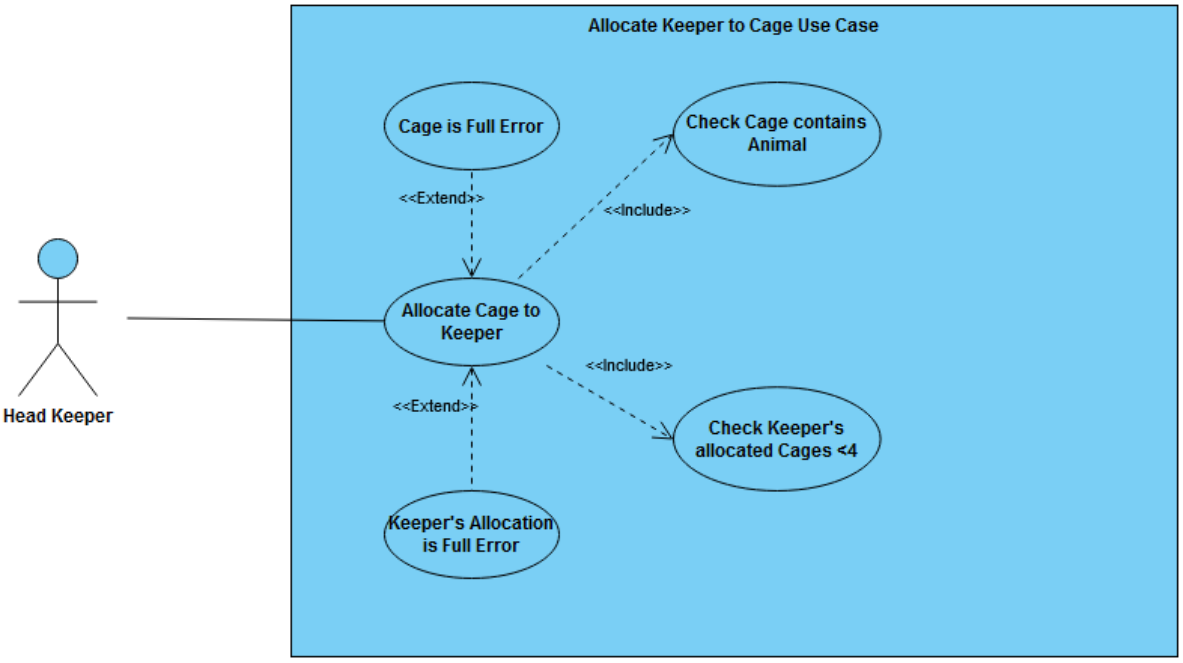





Add Keeper Use Case
Check if Keeper already exist
Already Exist Error
<<Extend>>
<<Include>>
Check Full Name Format Validity
<<Include>>
Add Keeper
<<Extend>>
Invalid Name Format Error
<<Include>>
<<Extend>>
Check All remaining details are entered
Missing Detail Error
Administrator


Add Cage Use Case
Add Cage
<<Include>>
Input Cage Capacity
<<Include>>
Input Cage description
Administrator


Allocate Animal to Cage Use Case
Full Cage Error
Check Cage remaining Capacity
<<Extend>>
<<Include>>
Allocate Animal to Cage
Get existing Animal(s) Category
<<Include>>
<<Include>>
<<Extend>>
Check Cage compatibility with new Animal
<<Include>>
Incompatible Animal for Cage Error
Get New Animal Category
Head Keeper


Allocate Keeper to Cage Use Case
Cage is Full Error
Check Cage contains Animal
<<Extend>>
<<Include>>
Allocate Cage to Keeper
<<Include>>
<<Extend>>
Check Keeper's allocated Cages <4
Keeper's Allocation is Full Error
Head Keeper

Object Oriented Design H172 35

Fill in the documentation for each use case. Make sure that you have all extends and includes.

Use Case Documentation

Use Case Name: UC1_Admin-Add Animal

Initiating Actor(s): Administrator	Receiving Actor(s): None

Trigger/Pre-condition(s):
The correct documentation has been provided to the admin (ownership docs, health/safety docs, management approvals..)

Main Flow of Events
1) Admin select the animal's Type
2) Admin enters the chosen animal's name
3) Check if the name format is valid
4) Check if the name/animal already exists in the system
5) Admin select animal's category(Prey or Predator)
6) Admin input Date of Birth
7) Admin input Date of Acquisition
8) Admin select animal sex
9) validate the remaining data
10) Use Case end

Alternate Flows
2a. Invalid Name Format
    1) Input a new valid name.
    2) Continue to step 3.
4a. Name already exists for this type of animal.
    1) Input a new name and continue to step 3.
    2) Or Animal already exists, exit Add Animal function.
6-9a. Missing Data
    1) Input missing Data.
    2) Continue to the next step.

Assumptions (optional):
Animal names within a type must be unique. The combination of name + type will determine if an animal already exists in the system.

Post-conditions: A new animal has been entered in the system.

Use Case Documentation

Use Case Name: UC2_Admin-Add Keeper

Initiating Actor(s): Administrator	**Receiving Actor(s):** None

Trigger/Pre-condition(s):
Correct documentation and approvals for a new Keeper confirmed.

Main Flow of Events
1) Admin inputs Keeper's name.
2) Admin enters Keeper's surname
4) Admin inputs Date of Birth
5) Check if Name and Surname in valid format
6) Check if Keeper already in the system
7) Admin inputs address
8) Admin enters contact number
9) Admin selects Keeper's position
10) End of Use Case

Alternate Flows
5a. Invalid Data Format
    1) Display error message.
    2) Enter correct information.

3) Continue to next step.

6a. Duplicate Keeper

1) Exit Add Keeper function.

Assumptions (optional):

Combination of Name, Surname and DOB will determine if the Keeper entry already exists.

Post-conditions: Keeper record added to the system.

Use Case Documentation

Use Case Name: UC3-Admin-Add Cage

Initiating Actor(s): Administrator	**Receiving Actor(s):**

Trigger/Pre-condition(s):

Management approvals and budget confirmed.

Main Flow of Events

1) Admin selects Cage capacity
2) Admin enters Cage Description

Alternate Flows

No alternate flow

Assumptions (optional):

Post-conditions: New Cage created in the system.

Use Case Documentation

Use Case Name: UC1_HKeeper-Allocate Animal to Cage

Initiating Actor(s): Head Keeper	**Receiving Actor(s):**

Trigger/Pre-condition(s):
Animal has been created in the system by the Admin. If needed, a new cage has also been created.

Main Flow of Events
1) Keeper selects the animal to allocate
2) Keeper selects the cage to allocate
3) Check Cage capacity
4) Check Cage compatibility with selected Animal
5) System allocate Animal to the selected Cage
6) End of Use Case

Alternate Flows
3a. Cage Full
    1) Display Error Message.
    2) Return to Step 2.
4a. Incompatible animal for the selected cage
    1) Display Error Message.
    2) Return to step 2 or exit.

Assumptions (optional):

Post-conditions: Animal is now allocated to a Cage.

Use Case Documentation

Use Case Name: UC1_HKeeper-Allocate Keeper to Cage

Initiating Actor(s): Head Keeper	**Receiving Actor(s):**

Trigger/Pre-condition(s):
Cage is not empty.

Main Flow of Events
1) Keeper selects the Cage to allocate
2) Keeper selects the Keeper to allocate
3) Check Keeper Allocation capacity
4) System allocate Animal to the selected Cage
5) End of Use Case

Alternate Flows
3a. Keeper at Full Capacity
    1) Display Error Message.
    2) Return to Step 2 or exit.

Assumptions (optional):

Post-conditions: Keeper is now allocated to a Cage.

Note: Once complete upload to canvas 'Task One Submission'