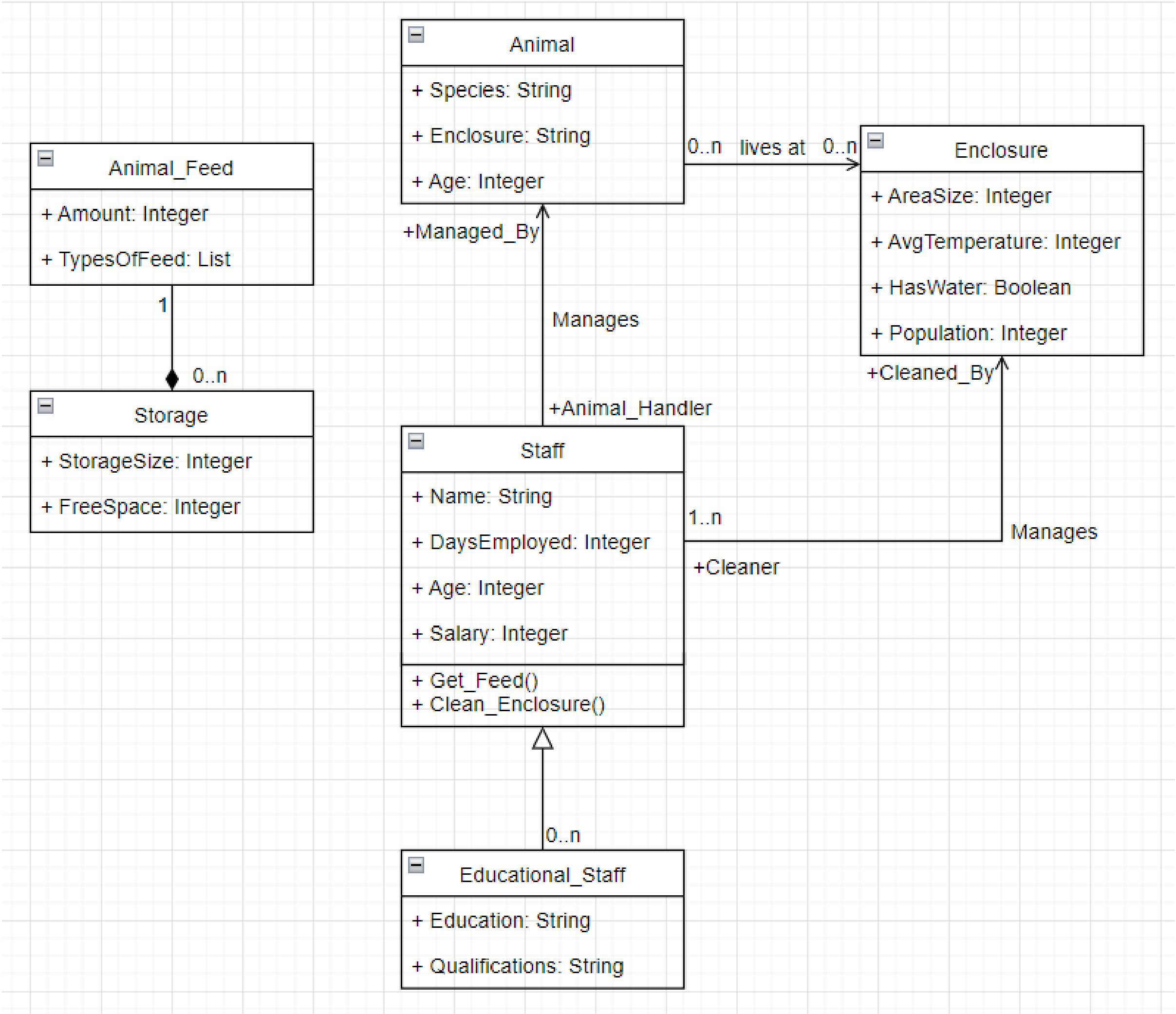
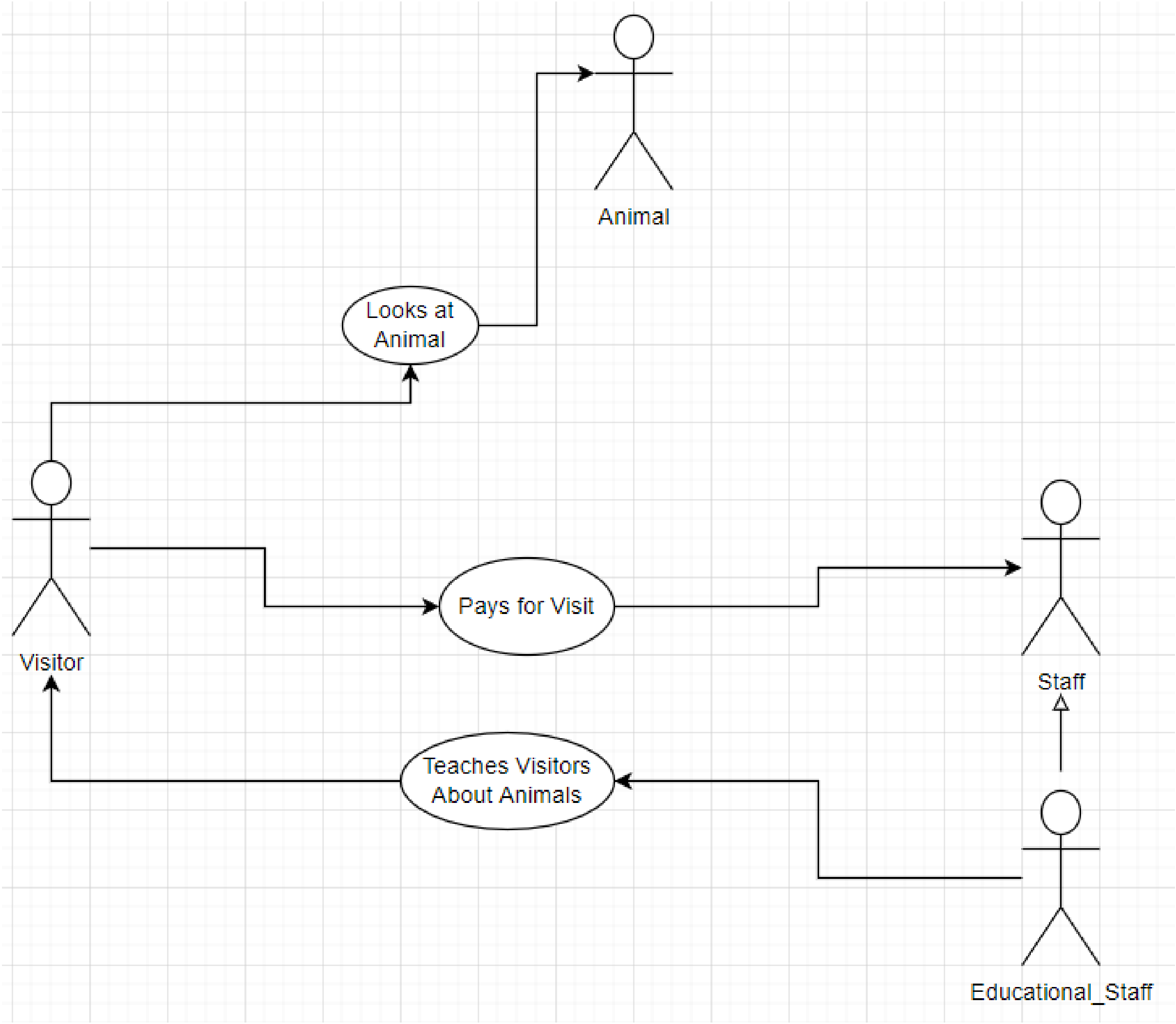
Sample 1

Question 1

(a)



(b)



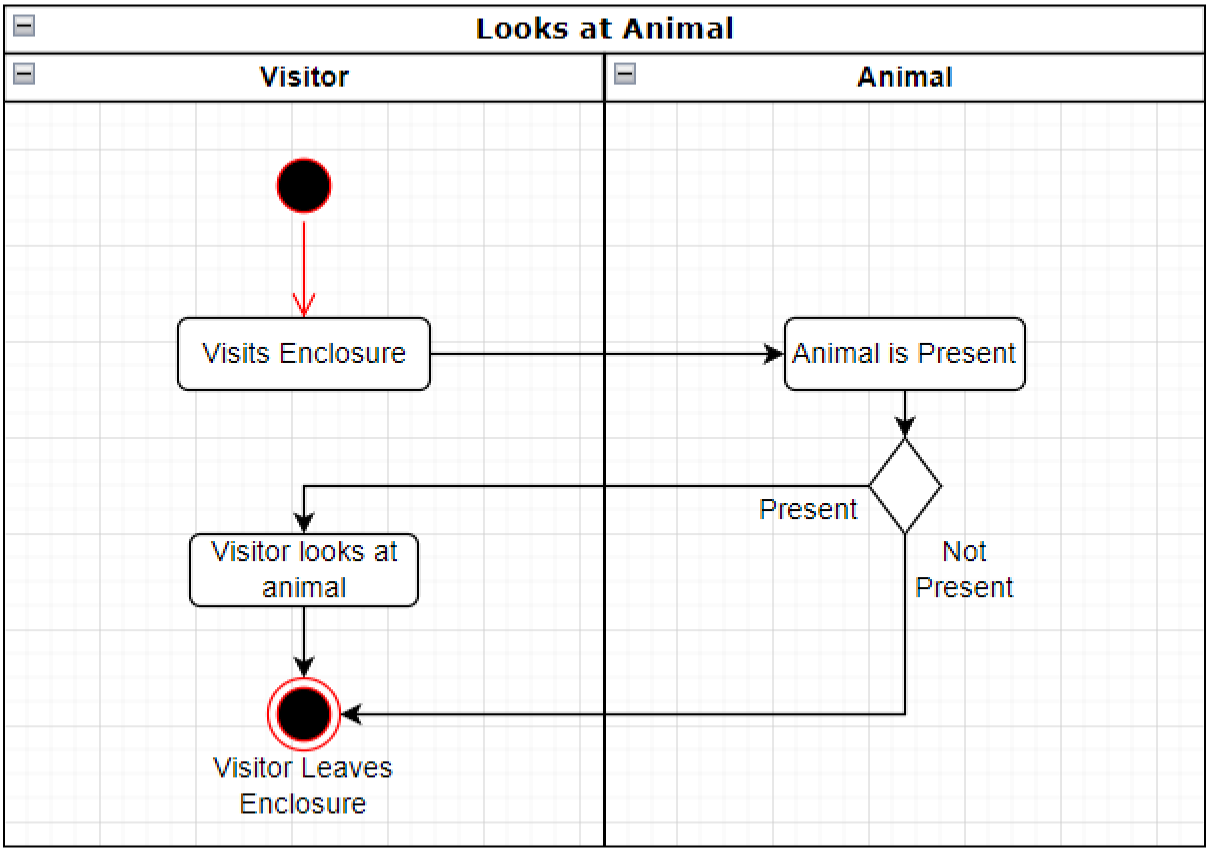
Use Case Descriptions:

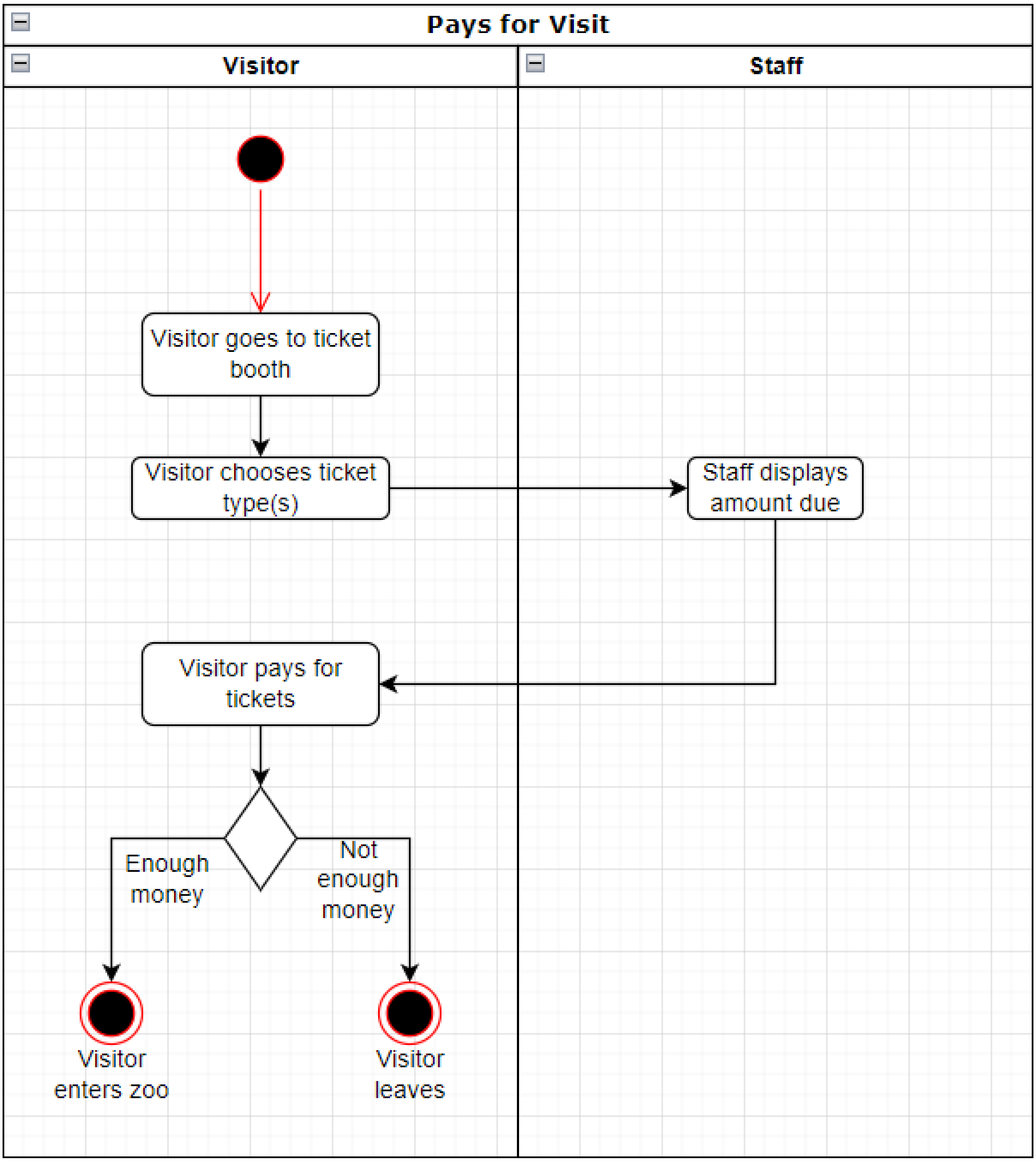
1. **Name:** Pays for Visit
2. **Participating Actors:** Visitor, Staff
3. **Entry Condition:**
   Visitor Stands in front of ticket booth with staff
   Visitor has enough money for ticket
4. **Exit Condition:** Visitor has ticket
5. **Normal Scenario:**
   Visitor chooses type of ticket(s) (adult, child, student, education)
   Staff communicated the amount due
   Visitor gives money to staff at ticket booth
   Staff issues ticket to visitor
6. **Error Scenario:**
   Visitor does not have sufficient money
   Zoo is at maximum capacity

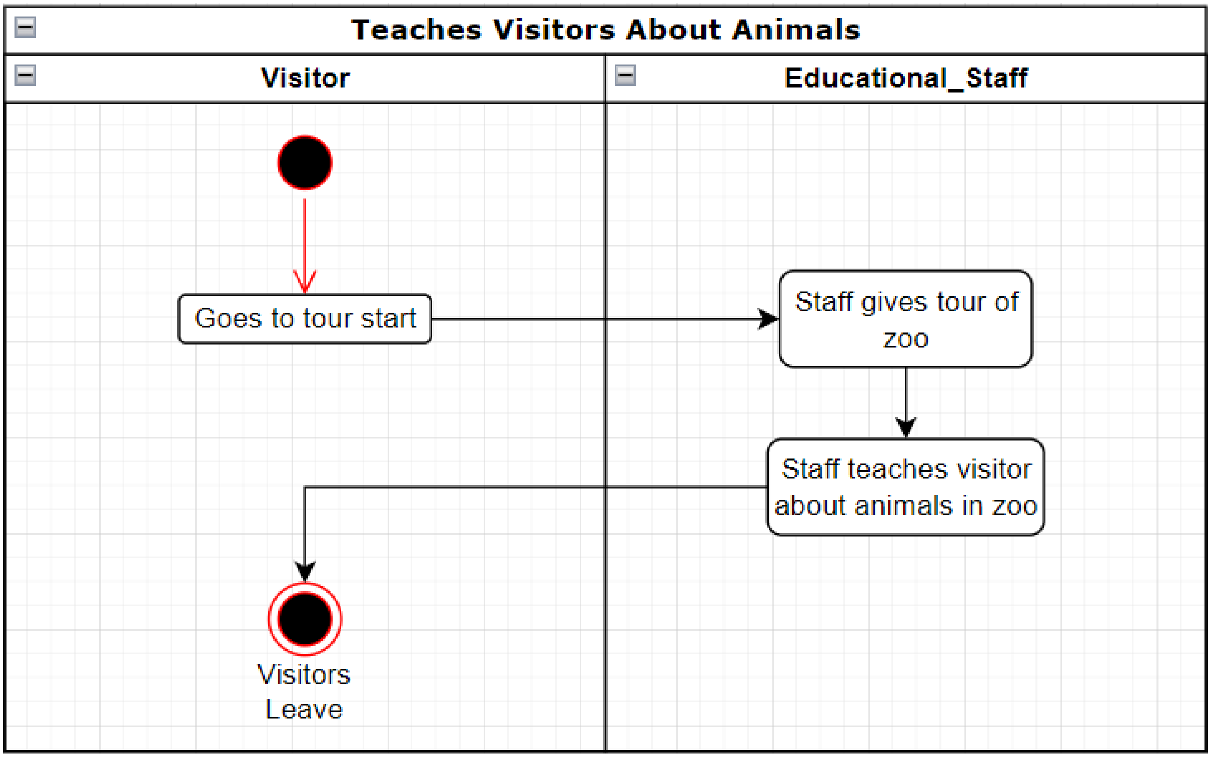
1. **Name:** Looks at Animal
2. **Participating Actors:** Visitor, Animal
3. **Entry Condition:**
   Visitor has paid for ticket
   Visitor goes to enclosure
4. **Exit Condition:** Visitor moves on to next animal
5. **Normal Scenario:**
   Visitor chooses enclosure to visit
   Visitor goes to enclosure
   Visitor sees animal in enclosure
6. **Error Scenario:**
   Animal is not present at enclosure

1. **Name:** Teaches Visitors About Animals
2. **Participating Actors:** Visitor, Educational_Staff
3. **Entry Condition:**
   Visitor has paid for education ticket
4. **Exit Condition:** Visitor has finished the education tour
5. **Normal Scenario:**
   Educational staff provides tour of zoo
   Educational staff provides information and facts to visitors about animals and enclosures
6. **Error Scenario:**
   Not enough educational staff is present at the zoo

(c)


Looks at Animal
Visitor
Animal
Visits Enclosure
Animal is Present
Present
Not Present
Visitor looks at animal
Visitor Leaves Enclosure


Pays for Visit
Visitor
Staff
Visitor goes to ticket booth
Visitor chooses ticket type(s)
Staff displays amount due
Visitor pays for tickets
Enough money
Not enough money
Visitor enters zoo
Visitor leaves



(d) Due to time constraints, I only chose to make the most obvious and necessary classes for running a zoo, along with very basic use case ovals.

(e) One major ethics problem with running a zoo is the display of animals in limited enclosures. This may upset animal rights groups that are against zoos.

Question 2

(a) A valid XML document uses DTD's that declare element types, attributes, cardinality and entities within the XML file, whether that be externally or internally. A valid XML document also requires XML declarations. Non-empty elements require closing tags which are matching and attribute values must always be quoted within the document.
DTD example:

```
<!DOCTYPE XML_Name [
  <!ELEMENT XML_Name (XML_Name*)>

    <!ELEMENT AddXML_Name(element1+, element2+)>

    <!ELEMENT element1 (#PCDATA)>
    <!ELEMENT element2 (#PCDATA)>
  ]>
```

Element tag and attribute example:

```
<element1> content </element1>
<element2> content </element2>
```

(b)

```
<!DOCTYPE salespersondirectory [
  <!ELEMENT salespersondirectory (salesperson*)>

    <!ELEMENT salesperson (name, telephone+)>
    <!ELEMENT name (firstname+, lastname+)>
    <!ELEMENT firstname (#PCDATA)>
    <!ELEMENT lastname (#PCDATA)>
    <!ELEMENT telephone (number)>
    <!ELEMENT number (#PCDATA)>
    <!ATTLIST salesperson area CDATA #IMPLIED>
    <!ATTLIST salesperson identification CDATA #IMPLIED)>
    <!ATTLIST telephone Type CDATA #REQUIRED>
  ]>
```

(c) (i)

```
let $x :=
doc("salesdirectory.xml")/salespersondirectory/salesperson/name
return
  <Surnames>
    {fn:string-join(($x/lastname), "+")}
  </Surnames>
```

(ii)

```
For $x in doc(“test.xml”)/salespersondirectory/salesperson/@area
return
  <AreasCovered>
  {$x}
  </AreasCovered>
```

(iii)

```
for $x in doc("test.xml")/salespersondirectory/salesperson/name
return
  ($x/firstname)[1]
```