XML Report

Group 14

Adam Mulvihill
Caoilfhionn Ní Dheoráin
Emmet Morrin
Emer Murphy

XML Files

PassengerList XML

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<!-- This document holds information about Passengers for flights -->
<!DOCTYPE Passengerlist [
<!-- The root element PassengerList holds multiple Passengers. It can
hold zero or more passengers. -->
<!ELEMENT Passengerlist (Passenger*)>
<!-- Each passenger can have this information about them in the
document. Each person may have a title, but it is not necessary. A
Passenger must have a last name, and one or more first names. They must
also have a date of birth, an age attribute. They may be boarding no
flights, one flight, or multiple. -->
<!ELEMENT Passenger (title?, firstname+, lastname, dateofbirth,
boards)>
<!ATTLIST Passenger age CDATA #REQUIRED>
<!ELEMENT dateofbirth (Date)>
<!ELEMENT Date (day, month, year)>
<!ELEMENT boards (Flight*)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT day (#PCDATA)>
<!ELEMENT month (#PCDATA)>
<!ELEMENT year (#PCDATA)>
]>

<Passengerlist>

  <Passenger age="26" xmi.idref="p1">
    <Passenger.title>Mr</Passenger.title>
    <Passenger.firstname>Andrew</Passenger.firstname>
    <Passenger.lastname>Burke</Passenger.lastname>
    <Passenger.dateofbirth>
      <Date>
        <Date.day>13</Date.day>
        <Date.month>June</Date.month>
        <Date.year>1994</Date.year>
      </Date>
    </Passenger.dateofbirth>
    <Passenger.boards>
      <Flight xmi.idref="f1" />
    </Passenger.boards>
  </Passenger>

  <Passenger age="20" xmi.idref="p2">
    <Passenger.title>Ms</Passenger.title>
```

```
    <Passenger.firstname>Andrea</Passenger.firstname>
    <Passenger.lastname>O'Shea</Passenger.lastname>
    <Passenger.dateofbirth>
      <Date>
        <Date.day>21</Date.day>
        <Date.month>September</Date.month>
        <Date.year>2000</Date.year>
      </Date>
    </Passenger.dateofbirth>
    <Passenger.boards>
      <Flight xmi.idref="f1" />
    </Passenger.boards>
  </Passenger>

  <Passenger age="43" xmi.idref="p3">
    <Passenger.firstname>Jonathan</Passenger.firstname>
    <Passenger.lastname>Griffin</Passenger.lastname>
    <Passenger.dateofbirth>
      <Date>
        <Date.day>3</Date.day>
        <Date.month>Feburary</Date.month>
        <Date.year>1977</Date.year>
      </Date>
    </Passenger.dateofbirth>
    <Passenger.boards>
      <Flight xmi.idref="f2" />
    </Passenger.boards>
  </Passenger>

  <Passenger age="41" xmi.idref="p4">
    <Passenger.firstname>Lisa</Passenger.firstname>
    <Passenger.lastname>Griffin</Passenger.lastname>
    <Passenger.dateofbirth>
      <Date>
        <Date.day>15</Date.day>
        <Date.month>May</Date.month>
        <Date.year>1975</Date.year>
      </Date>
    </Passenger.dateofbirth>
    <Passenger.boards>
      <Flight xmi.idref="f2" />
    </Passenger.boards>
  </Passenger>

</Passengerlist>
```

PassengeList UML to XML Changes

- The first change made was to make age an attribute rather than an element
- Another change was adding boards as an element to indicate what flight the passenger is taking. Boards is a relation on the UML diagram.
- I also added three elements to the Data element. These were day, month and year.

FlightList XML

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<!-- This document holds information about flights that would be necessary for a passenger to know, as well as information important to the airline company -->
<!DOCTYPE Flightlist [
<!-- The root element FlightList holds multiple Flights. It can hold zero or more Flights. -->
<!ELEMENT Flightlist (Flight*)>
<!-- Each Flight has a code, information on its arrival and departure, and a list of passengers for the flight -->
<!ELEMENT Flight (code, departure, arrival, passengers)>
<!-- Each Flight also has a number of seats. Number of Passengers is also added as an attribute -->
<!ATTLIST Flight noseats CDATA #REQUIRED>
<!ATTLIST Flight nopassengers CDATA #REQUIRED>
<!-- Each flight departs from an airport at a given time and arrives at an airport at a given time -->
<!ELEMENT departure (time, airport)>
<!ELEMENT arrival (time, airport)>
<!ELEMENT passengers (Passenger+)>
<!-- Since flights can be international and may be at any time, the time tag contains both timezone, date, and the local time in the given timezone -->
<!ELEMENT time (timezone, localtime, date)>
<!ELEMENT date (day, month, year)>
<!ELEMENT code (#PCDATA)>
<!ELEMENT timezone (#PCDATA)>
<!ELEMENT localtime (#PCDATA)>
<!ELEMENT day (#PCDATA)>
<!ELEMENT month (#PCDATA)>
<!ELEMENT year (#PCDATA)>
]>

<Flightlist>
```

```
<Flight noseats="150" nopassengers="2" xmi.idref="f1">
  <Flight.code>FR2998</Flight.code>
  <Flight.departure>
    <departure.time>
      <time.timezone>GMT</time.timezone>
      <time.localtime>13:30</time.localtime>
      <time.date>
        <date.day>6</date.day>
        <date.month>January</date.month>
        <date.year>2021</date.year>
      </time.date>
    </departure.time>
    <departure.airport>Dublin Airport</departure.airport>
  </Flight.departure>
  <Flight.arrival>
    <arrival.time>
      <time.timezone>CET</time.timezone>
      <time.localtime></time.localtime>
      <time.date>
        <date.day>6</date.day>
        <date.month>January</date.month>
        <date.year>2021</date.year>
      </time.date>
    </arrival.time>
    <arrival.airport>Amsterdam Airport Schiphol</arrival.airport>
  </Flight.arrival>
  <Flight.passengers>
    <Passenger xmi.idref="p1" />
    <Passenger xmi.idref="p2" />
  </Flight.passengers>
</Flight>

<Flight noseats="200" nopassengers="2" xmi.idref="f2">
  <Flight.code>FR383</Flight.code>
  <Flight.departure>
    <departure.time>
      <time.timezone>GMT</time.timezone>
      <time.localtime>08:55</time.localtime>
      <time.date>
        <date.day>11</date.day>
        <date.month>Feburary</date.month>
        <date.year>2021</date.year>
      </time.date>
    </departure.time>
    <departure.airport>Dublin Airport</departure.airport>
  </Flight.departure>
  <Flight.arrival>
    <arrival.time>
      <time.timezone>CET</time.timezone>
      <time.localtime>11:55</time.localtime>
      <time.date>
```

```
          <date.day>11</date.day>
          <date.month>Feburary</date.month>
          <date.year>2021</date.year>
        </time.date>
      </arrival.time>
      <arrival.airport>Frankfurt Airport</arrival.airport>
    </Flight.arrival>
    <Flight.passengers>
      <Passenger xmi.idref="p3" />
      <Passenger xmi.idref="p4" />
    </Flight.passengers>
  </Flight>
</Flightlist>
```

FlightList UML to XML Changes

- ❏ The first change made to Flight was making both noSeats and noPassengers attributes.
- ❏ Departure and Arrival were added to hold both the time of departure/arrival and the airport of departure/arrival, instead of having them as separate elements in flight.
- ❏ I gave time multiple elements: timezone, localtime, and date.

Ticket XML

```
# this xml document describes the information a ticket holds
#such information is useful for both the passenger and the airline company
<?xml version="1.0" ?>

#DTD
<!DOCTYPE Ticket_info[
#The element Ticket_info contains a list of all the tickets
#The cardinality * means that the list can contain multiple tickets
  <!ELEMENT Ticket_info (Ticket*)>
#An element within ticket can contain all of the following
  <!ELEMENT Ticket (Ticket.flight, Ticket.DepartureAirport,
Ticket.CountryOfDeparture, Ticket.CountryOfArrival,
Ticket.timeOfPurchase, Ticket.basePrice, Ticket.addedFees, Type)>
#Here are all the following elements followed be there types
<!ELEMENT Ticket.fight   (#PCDATA)> # flight number so the passenger can identify their flight
<!ELEMENT Ticket.DepartureAirport   (#PCDATA)> #name of departure airport
<!ELEMENT Ticket.ArrivalAirport   (#PCDATA)> #name of arrival airport
<!ELEMENT Ticket.CountryOfDeparture  (#PCDATA)> #name of departure country
<!ELEMENT Ticket.CountryOfArrival  (#PCDATA)> #name of arrival country
```

```
<!ELEMENT Ticket.timeOfPurchase   (#PCDATA)> #time the ticket was
purchased
<!ELEMENT Ticket.basePrice   (#PCDATA)> #the price of the ticket
without any additional fees
<!ELEMENT Ticket.addedFees   (#PCDATA)> #the price of any additional
fees(bagage, priority boarding)
#Ticket contains an attribute called Type, which specifies whether the
ticket is of a single or return type
<!ATTLIST entry Type CDATA #IMPLIED>
<!ELEMENT BoardingPass (BoardingPass+)>
 ]>

#XML
#Ticket List
<Ticket_info>
#Ticket 1 with attribute  type return
  <Ticket type="return">
    <Ticket.fight>FR383</Ticket.fight>
    <Ticket.CountryOfDeparture>Ireland</Ticket.CountryOfDeparture>
    <Ticket.CountryOfArrival>Germany</Ticket.CountryOfArrival>
    <Ticket.DepartureAirport>Dublin Airport</Ticket.DepartureAirport>
    <Ticket.ArrivalAirport>Frankfurt Airport</Ticket.ArrivalAirport>
    <Ticket.timeOfPurchase>10:05</Ticket.timeOfPurchase>
    <Ticket.basePrice>25.99</Ticket.basePrice>
    <Ticket.addedFees>20</Ticket.addedFees>
      #reference number to a boarding pass
      <Ticket.BoardingPass> <BoardingPass xmi.idref= "b1"/>
</Ticket.BoardingPass>
  </Ticket>
#Ticket 2 with attribute type single
  <Ticket type = "single">
    <Ticket.fight>FR2998</Ticket.fight>
     <Ticket.CountryOfDeparture>Ireland</Ticket.CountryOfDeparture>
    <Ticket.CountryOfArrival>Netherlands</Ticket.CountryOfArrival>
    <Ticket.DepartureAirport>Dublin Airport</Ticket.DepartureAirport>
    <Ticket.ArrivalAirport>Amsterdam Airport
Schiphol</Ticket.ArrivalAirport>
    <Ticket.timeOfPurchase>15:30</Ticket.timeOfPurchase>
    <Ticket.basePrice>15.99</Ticket.basePrice>
    <Ticket.addedFees>20</Ticket.addedFees>
      <Ticket.BoardingPass><BoardingPass xmi.idref= "b2"/>
</Ticket.BoardingPass>
  </Ticket>
  #Ticket 3 with attribute type single
   <Ticket type="single">
    <Ticket.fight>FR476</Ticket.fight>
    <Ticket.CountryOfDeparture>Ireland</Ticket.CountryOfDeparture>
    <Ticket.CountryOfArrival>France</Ticket.CountryOfArrival>
    <Ticket.DepartureAirport>Dublin Airport</Ticket.DepartureAirport>
    <Ticket.ArrivalAirport>Lyon-Saint-Exupéry
Airport</Ticket.ArrivalAirport>
```

```
    <Ticket.timeOfPurchase>18:55</Ticket.timeOfPurchase>
    <Ticket.basePrice>46.99</Ticket.basePrice>
    <Ticket.addedFees>40</Ticket.addedFees>
    <Ticket.BoardingPass> <BoardingPass xmi.idref= "b3"/>
</Ticket.BoardingPass>
  </Ticket>
</Ticket_info>
```

Ticket_info UML to XML Changes

- ❏ I changed the boolean isReturn form the UML Class diagram to be an attribute for ticket where it can either be of type single or return
- ❏ I added a boarding pass element to each ticket
- ❏ I changed time of purchase to be in time format
- ❏ I added country of arrival, country of departure, arrival airport and a departure airport for more clarity

Boarding Pass XML

```
# this xml document describes the information a boarding pass holds
#such information is useful for both the passenger and the airline company
#Passengers can identify which gate and seat number they need to arrive at as well as the time the flight will depart

<?xml version="1.0" ?>
#DTD
<!DOCTYPE BoardingPass_info[
#The element BoardingPass_info contains a list of all the boarding passes
#The cardinality * means that the list can contain multiple boarding passes
<!ELEMENT BoardingPass_info (BoardingPass*)>
<!ELEMENT BoardingPass.flight (#PCDATA)> # flight number so the passenger can identify their flight
<!ELEMENT BoardingPass.seat (#PCDATA)> #Which seat the passenger is to sit in
<!ELEMENT BoardingPass.gateNumber (#PCDATA)> #Which gate the passenger must arrive at
<!ELEMENT BoardingPass.Passenger (#PCDATA)> #Type of passenger, priority or non-priority
<!ELEMENT BoardingPass.boarding (#PCDATA)> #Entry for passenger to board through
<!ELEMENT BoardingPass.flyOut (#PCDATA)> #Time of departure
<!ELEMENT BoardingPass (Type, BoardingPass.flight,
BoardingPass.boarding, BoardingPass.gateNumber, BoardingPass.flyOut,
BoardingPass.seat, BoardingPass.Passenger)>
```

```
]>

#XML
#Boarding Pass list
<BoardingPass_info>
#Each boarding pass contains an id so as to identify which ticket it
belongs to
#Boarding Pass 1
  <BoardingPass xmi.id= "b1">
  <BoardingPass.flight>FR383</BoardingPass.flight>
  <BoardingPass.seat>2A</BoardingPass.seat>
  <BoardingPass.gateNumber>10</BoardingPass.gateNumber>
  <BoardingPass.boarding>Front Door</BoardingPass.boarding>
  <BoardingPass.flyOut> 06:10</BoardingPass.flyOut>
  <BoardingPass.Passenger>Priority</BoardingPass.Passenger>
  </BoardingPass>
#Boarding Pass 2
  <BoardingPass xmi.id= "b2">
  <BoardingPass.flight>FR2998</BoardingPass.flight>
  <BoardingPass.seat>13F</BoardingPass.seat>
   <BoardingPass.gateNumber>15</BoardingPass.gateNumber>
   <BoardingPass.boarding>Back Door</BoardingPass.boarding>
   <BoardingPass.flyOut>09:30</BoardingPass.flyOut>
  <BoardingPass.Passenger>Non-Priority</BoardingPass.Passenger>
  </BoardingPass>
 #Boarding Pass 3
   <BoardingPass xmi.id= "b3">
  <BoardingPass.flight>FR476</BoardingPass.flight>
  <BoardingPass.seat>26B</BoardingPass.seat>
   <BoardingPass.gateNumber>3</BoardingPass.gateNumber>
   <BoardingPass.boarding>Front Door</BoardingPass.boarding>
   <BoardingPass.flyOut>16:20</BoardingPass.flyOut>
  <BoardingPass.Passenger>Priority</BoardingPass.Passenger>
  </BoardingPass>
</BoardingPass_info>
```

BoardingPass_info UML to XML Changes

- ❏ I added a gate number, boarding type and a time of departure element to the UML Class diagram for boarding class
- ❏ I find with this extra information it will be easier for the passengers to navigate to their flight

External Services XML

```
<?xml version="1.0" ?>
<!DOCTYPE External_Services_info[
<!ELEMENT External_Services_info (External_Service*)>

<!ELEMENT External_Service (Type, External_Services.Name,
External_Services.Countries*, External_Services.Cities,
External_Services.Telephone, External_Services.Email, Hotel,
Car_Hire_Company)>

<!ELEMENT Hotel (Hotel.Average_cost_Per_Night, Hotel.Room_Available,
Hotel.Breakfast_Included, Hotel.Check_In_Time, Hotel.Check_Out_Time,
Hotel.Review*)>

<!ELEMENT Car_Hire_Company (Cars*, Associated_Companies*)>

<!ATTLIST entry Type CDATA #IMPLIED>
<!ELEMENT External_Services.Name   (#PCDATA)>
<!ELEMENT External_Services.Countries   (#PCDATA)>
<!ELEMENT External_Services.Cities  (#PCDATA)>
<!ELEMENT External_Services.Telephone   (#PCDATA)>
<!ELEMENT External_Services.Email   (#PCDATA)>

<!ELEMENT Hotel.Average_Cost_Per_Night (#PCDATA)>
<!ELEMENT Hotel.Room_Available (#PCDATA)>
<!ELEMENT Hotel.Breakfast_Included (#PCDATA)>
<!ELEMENT Hotel.Check_In_Time (#PCDATA)>
<!ELEMENT Hotel.Check_Out_Time (#PCDATA)>
<!ELEMENT Hotel.Review (#PCDATA)>

<!ELEMENT CarHireCompany.Cars (#PCDATA)>
<!ELEMENT CarHireCompany.Associated_Companies (#PCDATA)>
]>
<External_Services_info>
    <External_Service type="hotel">
    <External_Services.Name>The Airport Hotel</External_Services.Name>
    <External_Services.Countries>Ireland</External_Services.Countries>
    <External_Services.Countries>France</External_Services.Countries>
    <External_Services.City>Dublin</External_Services.City>
    <External_Services.City>Paris</External_Services.City>

<External_Services.Telephone>03-1452-4567</External_Services.Telephone>

<External_Services.Email>TheAirportHotel@gmail.com</External_Services.E
mail>
  <Hotel stars="4">

<Hotel.Average_Cost_Per_Night>60</Hotel.Average_Cost_Per_Night>
          <Hotel.Room_Available>True</Hotel.Room_Available>
          <Hotel.Breakfast_Included>True</Hotel.Breakfast_Included>
```

```
         <Hotel.Check_In_Time>16:00</Hotel.Check_In_Time>
          <Hotel.Check_Out_Time>12:00</Hotel.Check_Out_Time>
         <Hotel.Review>
               A very nice hotel.
        </Hotel.Review>
 </Hotel>
    </External_Service>


    <External_Service type="hotel">
         <External_Services.Name>Fitz Hotel</External_Services.Name>
         <External_Services.Countries>Costa
Rica</External_Services.Countries>
         <External_Services.City>San Jose</External_Services.City>

<External_Services.Telephone>87-58376-874</External_Services.Telephone>

<External_Services.Email>FitzHotelSanJose@gmail.com</External_Services.
Email>
      <Hotel>

<Hotel.Average_Cost_Per_Night>25</Hotel.Average_Cost_Per_Night>
          <Hotel.Room_Available>True</Hotel.Room_Available>
          <Hotel.Breakfast_Included>False</Hotel.Breakfast_Included>
         <Hotel.Check_In_Time>17:00</Hotel.Check_In_Time>
          <Hotel.Check_Out_Time>12:00</Hotel.Check_Out_Time>
         <Hotel.Review>
               Cheap and cheerful.
        </Hotel.Review>
    </Hotel>

    </External_Service>
        <External_Service type="Car_Hire_Company">
        <External_Services.Name>Love_Cars</External_Services.Name>

<External_Services.Countries>Ireland</External_Services.Countries>

<External_Services.Countries>France</External_Services.Countries>
        <External_Services.City>Dublin</External_Services.City>
         <External_Services.City>Cork</External_Services.City>
        <External_Services.City>Paris</External_Services.City>

<External_Services.Telephone>03-1452-4569</External_Services.Telephone>

<External_Services.Email>LoveCars@gmail.com</External_Services.Email>
         <CarHireCompany>
            <CarHireCompany.Cars>BMW</CarHireCompany.Cars>
            <CarHireCompany.Cars>Thesla</CarHireCompany.Cars>

<CarHireCompany.Associated_Companies>Hertz</CarHireCompany.Associated_C
ompanies>
```

```
<CarHireCompany.Associated_Companies>Silly_Cars</CarHireCompany.Associa
ted_Companies>
        </CarHireCompany>

    </External_Service>

</External_Services_info>
```

External_Services_info UML to XML Changes

- I added some attributes to the class External Services in the UML Class diagram
- I added type, this describes what external service it is
- I added cities, this is the list of cities that the external service services
- I add telephone, this is the phone number to contact the external service
- I add email, this is the email address to contact the external service

Hotels XML

```
<?xml version="1.0" ?>

<!DOCTYPE Hotels[
<!ELEMENT Hotels (Hotel*)>
<!ELEMENT Hotel
(External_Services.Name,External_Services.Countries*,External_Services.
City*,External_Services.Telephone,External_Services.Email,
Hotel.Average_Cost_Per_Night,Hotel.Room_Available,Hotel.Breakfast_Inclu
ded,Hotel.Check_In_Time,Hotel.Check_Out_Time,Hotel.review*)>
<!ATTLIST Hotel Stars CDATA #REQUIRED>
<!ELEMENT External_Services.Name   (#PCDATA)>
<!ELEMENT External_Services.Countries   (#PCDATA)>
<!ELEMENT External_Services.City  (#PCDATA)>
<!ELEMENT External_Services.Telephone   (#PCDATA)>
<!ELEMENT External_Services.Email   (#PCDATA)>
<!ELEMENT Hotel.Average_Cost_Per_Night  (#PCDATA)>
<!ELEMENT Hotel.Room_Available  (#PCDATA)>
<!ELEMENT Hotel.Breakfast_Included   (#PCDATA)>
<!ELEMENT Hotel.Check_In_Time  (#PCDATA)>
<!ELEMENT Hotel.Check_Out_Time  (#PCDATA)>
<!ELEMENT Hotel.review   (#PCDATA)>
]>

<Hotels>
    <Hotel Stars="4">
         <External_Services>
        <External_Services.Name>The Airport
Hotel</External_Services.Name>
```

```
<External_Services.Countries>Ireland</External_Services.Countries>

<External_Services.Countries>France</External_Services.Countries>
         <External_Services.City>Dublin</External_Services.City>
         <External_Services.City>Paris</External_Services.City>

<External_Services.Telephone>03-1452-4567</External_Services.Telephone>

<External_Services.Email>TheAirportHotel@gmail.com</External_Services.E
mail>
         </External_Services>

<Hotel.Average_Cost_Per_Night>60</Hotel.Average_Cost_Per_Night>
          <Hotel.Room_Available>True</Hotel.Room_Available>
          <Hotel.Breakfast_Included>True</Hotel.Breakfast_Included>
         <Hotel.Check_In_Time>16:00</Hotel.Check_In_Time>
          <Hotel.Check_Out_Time>12:00</Hotel.Check_Out_Time>
         <Hotel.review>
               A very nice hotel.
        </Hotel.review>
    </Hotel>
    <Hotel Stars="3">
          <External_Services>
         <External_Services.Name>Fitz Hotel</External_Services.Name>
         <External_Services.Countries>Costa
Rica</External_Services.Countries>
         <External_Services.City>San Jose</External_Services.City>

<External_Services.Telephone>87-58376-874</External_Services.Telephone>

<External_Services.Email>FitzHotelSanJose@gmail.com</External_Services.
Email>
         </External_Services>

<Hotel.Average_Cost_Per_Night>25</Hotel.Average_Cost_Per_Night>
          <Hotel.Room_Available>True</Hotel.Room_Available>
          <Hotel.Breakfast_Included>False</Hotel.Breakfast_Included>
         <Hotel.Check_In_Time>17:00</Hotel.Check_In_Time>
          <Hotel.Check_Out_Time>12:00</Hotel.Check_Out_Time>
         <Hotel.review>
               Cheap and cheerful.
        </Hotel.review>
    </Hotel>
    <Hotel Stars="2">
          <External_Services>
         <External_Services.Name>The Ugly
Hotel</External_Services.Name>

<External_Services.Countries>Germany</External_Services.Countries>
         <External_Services.City>Berlin</External_Services.City>
```

```
<External_Services.Telephone>87-493875-45</External_Services.Telephone>

<External_Services.Email>TheUglyHotel@gmail.com</External_Services.Emai
l>
         </External_Services>

<Hotel.Average_Cost_Per_Night>150</Hotel.Average_Cost_Per_Night>
          <Hotel.Room_Available>True</Hotel.Room_Available>
          <Hotel.Breakfast_Included>False</Hotel.Breakfast_Included>
         <Hotel.Check_In_Time>16:00</Hotel.Check_In_Time>
          <Hotel.Check_Out_Time>12:00</Hotel.Check_Out_Time>
         <Hotel.review>
               A very ugly hotel, would not recommend to a friend!
        </Hotel.review>
         <Hotel.review>
               UGLY hotel!
        </Hotel.review>
         <Hotel.review>
               This Hotel needs help from Francis Brennan.
        </Hotel.review>
    </Hotel>
</Hotels>
```

Hotels UML to XML Changes

- ❏ I added and removed some attributes to the class Hotels in the UML Class diagram
- ❏ I removed name, countries and cities because these are all in the class External Services and Hotels is a child class of External Services
- ❏ I added costPerNight, this is the cost of a room for the night
- ❏ I added breakfastIncluded, this is whether breakfast is provided by the hotel and included in the price
- ❏ I added roomAvailable, this is whether there is an empty room
- ❏ I added checkInTime, this is the time from when check in begins
- ❏ I added checkOutTime, this is the time up to when check out ends
- ❏ I added Reviews, this is the reviews of the hotel

XML Queries

Ticket Price XQuery

```
for $i in doc("ticket.xml")/Ticket_info/Ticket,
    $j in doc("boardingPass.xml")/BoardingPass_info/BoardingPass
where $j/BoardingPass.Passenger = "Priority"
where $i/Ticket.BoardingPass/BoardingPass/@xmi.idref = $j/@xmi.id
return
  <Priority_prices_are>
    {$i/Ticket.basePrice + $i/Ticket.addedFees}
  </Priority_prices_are>
```

- ❏ Returns the total price of each ticket if and only if they are priority tickets.
- ❏ This could be useful to a customer who wants to know what the prices of the priority tickets are. The resulting XML output could be used further to get an average price.
- ❏ The expected output would be:

```
<Priority_prices_are>45.989999999999995</Priority_prices_are>
<Priority_prices_are>86.99000000000001</Priority_prices_are>
```

Passenger Headed XQuery

```
declare function local:passengerHeaded($name as xs:string)
{
  for $i in doc("Passengerlist.xml")/Passengerlist/Passenger,
      $j in doc("Flightlist.xml")/Flightlist/Flight
  where $i/Passenger.firstname = $name
  where $j/@xmi.idref = $i/Passenger.boards/Flight/@xmi.idref
  return
    <info>
      {$j/Flight.arrival/arrival.airport/text()}
    </info>
};

<info>
{local:passengerHeaded("Andrew")}
</info>
```

- ❏ This function returns the Airport in which the passenger(s) are headed to, when input a first name.
- ❏ This could be useful to the passenger who's searching their name, to find out exactly where they're headed to.
- ❏ With a name input (in this example, “Andrew”), the expected output would be:

```
<info>Amsterdam Airport Schiphol</info>
```

Passenger Flight XQuery

```
declare function local:passengerSeat($name as xs:string)
{
  for $i in doc("Passengerlist.xml")/Passengerlist/Passenger,
      $j in doc("Flightlist.xml")/Flightlist/Flight
  where $i/Passenger.firstname = $name
  where $j/@xmi.idref = $i/Passenger.boards/Flight/@xmi.idref
  return
    <info>
      {$j/Flight.code/text()}
    </info>
};
<info>
{local:passengerSeat("Andrea")}
</info>
```

- ❏ This function returns the flight code in which the passenger(s) are taking.
- ❏ This could be useful to the passenger who's searching their name, to find out exactly what flight they're taking. This could be expanded further to see if there are any delays with this flight or other information about that flight that the passenger may want to know.
- ❏ With a name input (in this example, “Andrea”), the expected output would be:

```
<info>FR2998</info>
```

Hotel XQuery(1)

```
<results>
{
  let $b:=

doc("hotel.xml")/Hotels/Hotel/External_Services/External_Services.
Countries
(: returns the countries of the hotels in the XML file :)
  return
  <Hotel_Countries>
  { $b }
  </Hotel_Countries>
}
</results>
```

- ❏ This query returns the countries that the hotels in the XML file service.
- ❏ The expected output would be:

```
<results>
  <Hotel_Countries>

<External_Services.Countries>Ireland</External_Services.Countries>

<External_Services.Countries>France</External_Services.Countries>
    <External_Services.Countries>Costa
Rica</External_Services.Countries>

<External_Services.Countries>Germany</External_Services.Countries>
  </Hotel_Countries>
</results>
```

Hotel XQuery(2)

```
<Hotels>
{
  for $b in /Hotels/Hotel/External_Services
  where $b/External_Services.Name = "Fitz Hotel"
  (:returns the contact information for the hotel with the name
given:)
  return
  <External_Services>
  { $b/External_Services.Name }
  { $b/External_Services.Telephone}
  { $b/External_Services.Email}
  </External_Services>
}
</Hotels>
```

- This query returns the contact information of a hotel given the name of the hotel.
- The query returns the name, telephone number and email address.
- The expected output would be:

```
<Hotels>
  <External_Services>
    <External_Services.Name>Fitz
Hotel</External_Services.Name>

<External_Services.Telephone>87-58376-874</External_Services.
Telephone>

<External_Services.Email>FitzHotelSanJose@gmail.com</External
_Services.Email>
  </External_Services>
</Hotels>
```

External Services XQuery

```
let $c:=
doc("ExternalServicess.xml")/External_Services_info/External_Servi
ce/External_Services.Name
return
<External_Service_Names>
{$c}
</External_Service_Names>
```

- ❏ This query returns the names of all the external services in the XML file using let.
- ❏ The expected output would be:

```
<External_Service_Names>
  <External_Services.Name>The Airport
Hotel</External_Services.Name>
  <External_Services.Name>Fitz Hotel</External_Services.Name>
  <External_Services.Name>Love_Cars</External_Services.Name>
</External_Service_Names>
```

Custom Function XQuery

```
declare function local:all_countries()
{
  for $s in
doc("ticket.xml")/Ticket_info/Ticket/Ticket.CountryOfDeparture/tex
t()
    union

doc("ticket.xml")/Ticket_info/Ticket/Ticket.CountryOfArrival/text(
)
  group by $s
  return
    <country>
      {$s}
    </country>
};

<all>
{local:all_countries()}
</all>
```

- ❏ This function gets all of the countries that this airline services, both departing from and arriving in.
- ❏ This function could be useful to the airline itself, to see which countries they currently fly to and from, It could also be used to compare this airline to other airlines, and see which countries the airline should expand their services to.

- ❏ The expected output for this function would be:

```
<country>Ireland</country>
<country>Germany</country>
<country>Frankfort Airport</country>
<country>Netherlands</country>
<country>Amsterdam Airport Schiphol</country>
<country>France</country>
<country>Lyon-Saint-Exupéry Airport</country>
```

Strengths and Weaknesses

Our XML is specifically tailored to be easily understood. We have named all our XML elements and attributes so that they perfectly describe their functionality. Due to the work we put into our UML files, it was easy for us to transfer this to XML. Our XML files thoroughly describe our UML classes. They also contain and categorise a lot of useful information that can be used in the XQueries. We tried to include a lot of links throughout our documents. For example our XQueries test multiple documents and we include associations/relationships between the classes in the XML files. The development of our topic can be seen in the project as we continued to improve and add to our UML Classes. This shows how we were constantly thinking of the best ways to describe our information system and how it can be of maximum use to the customer. We included all necessary clauses and functions.

A weakness is that we didn't have enough time to create more XML files to make even more links between documents. I think we could've expanded the project more so that you could see the association between all the classes. Perhaps we could've added some more unique elements and attributes too.

Listing of Who Did What

XML - Ticket	Caoilfhionn Ní Dheoráin
XML - Boarding Pass	Caoilfhionn Ní Dheoráin
DTD - Ticket	Caoilfhionn Ní Dheoráin
DTD - Boarding Pass	Caoilfhionn Ní Dheoráin
Strengths and Weaknesses	Caoilfhionn Ní Dheoráin
XML - PassengerList	Adam Mulvihill
DTD - PassengerList	Adam Mulvihill
XML - FlightList	Adam Mulvihill
DTD - FlightList	Adam Mulvihill
XML - Hotels	Emer Murphy
DTD - Hotels	Emer Murphy
XML - External Services	Emer Murphy
DTD - External Services	Emer Murphy
XQuery - Hotels	Emer Murphy
XQuery - External Services	Emer Murphy
XQuery- Ticket Price	Emmet Morrin
XQuery - Passenger Headed	Emmet Morrin
XQuery - Passenger Flight	Emmet Morrin
XQuery - List Countries (Custom function)	Emmet Morrin