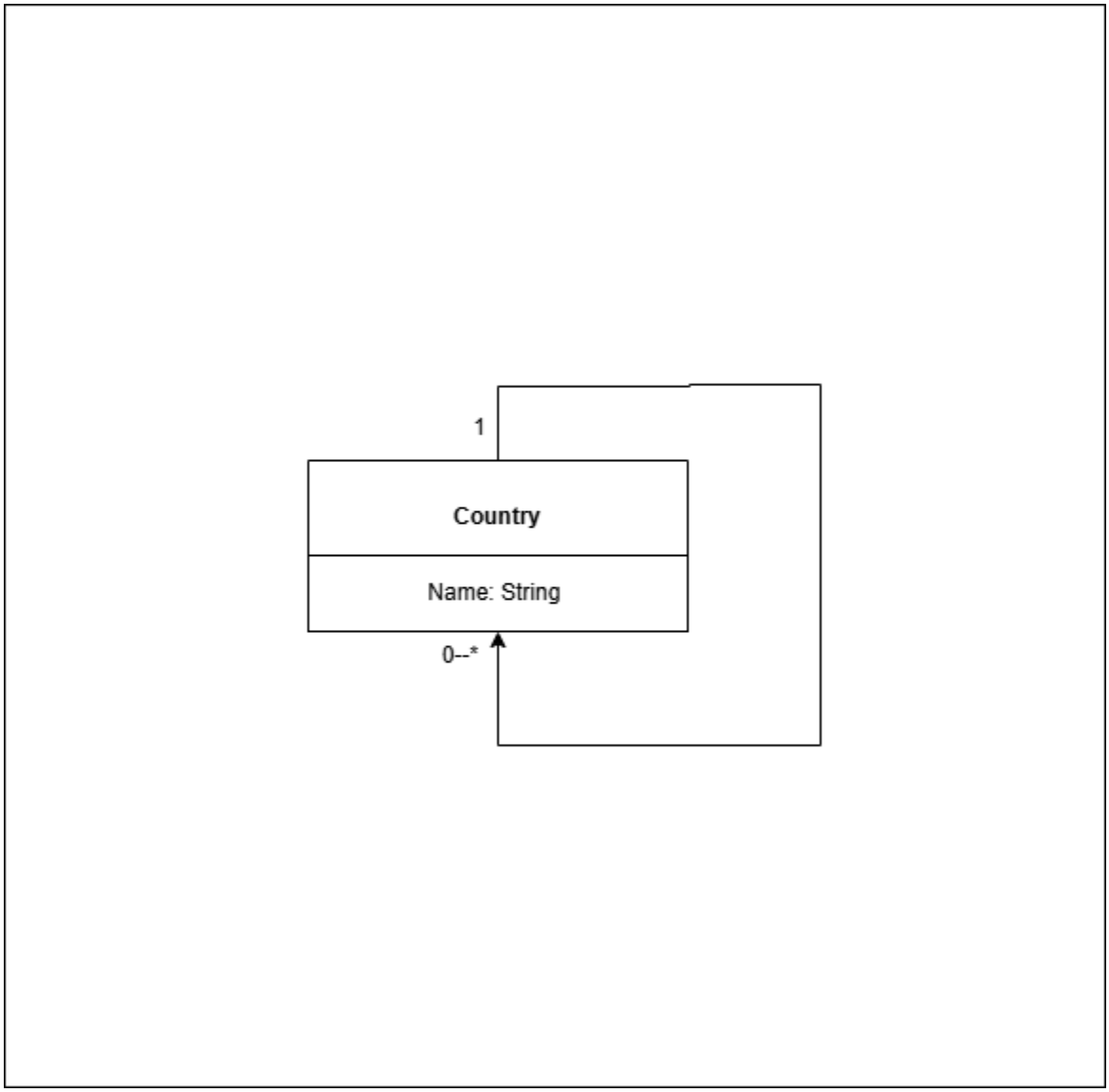
Software Engineering
Lab_4

Student I'd - 202201152
Name - Ragan Patel

Q.1 Prepare a class diagram for the following object diagram that shows a portion of Europe.



Q.2 Prepare a class diagram for the object diagram given in Figure -2. Explain your multiplicity decisions.
What is the smallest number of points required to construct a polygon? Does it make a difference
whether or not points may be shared between polygons? Your answer should address the fact that
points are ordered.

Sub Question 1: Elaborate on the variety of decisions involved.

Multiplicity Decisions:

A polygon is made up of several points. The minimum number of points required to create a polygon (such as a triangle) is 3.

The connection between a Polygon and its Points is one-to-many. In other words, a single Polygon contains several Points.

A Point can be part of one or multiple Polygons (this can be illustrated using a many-to-many relationship if necessary).

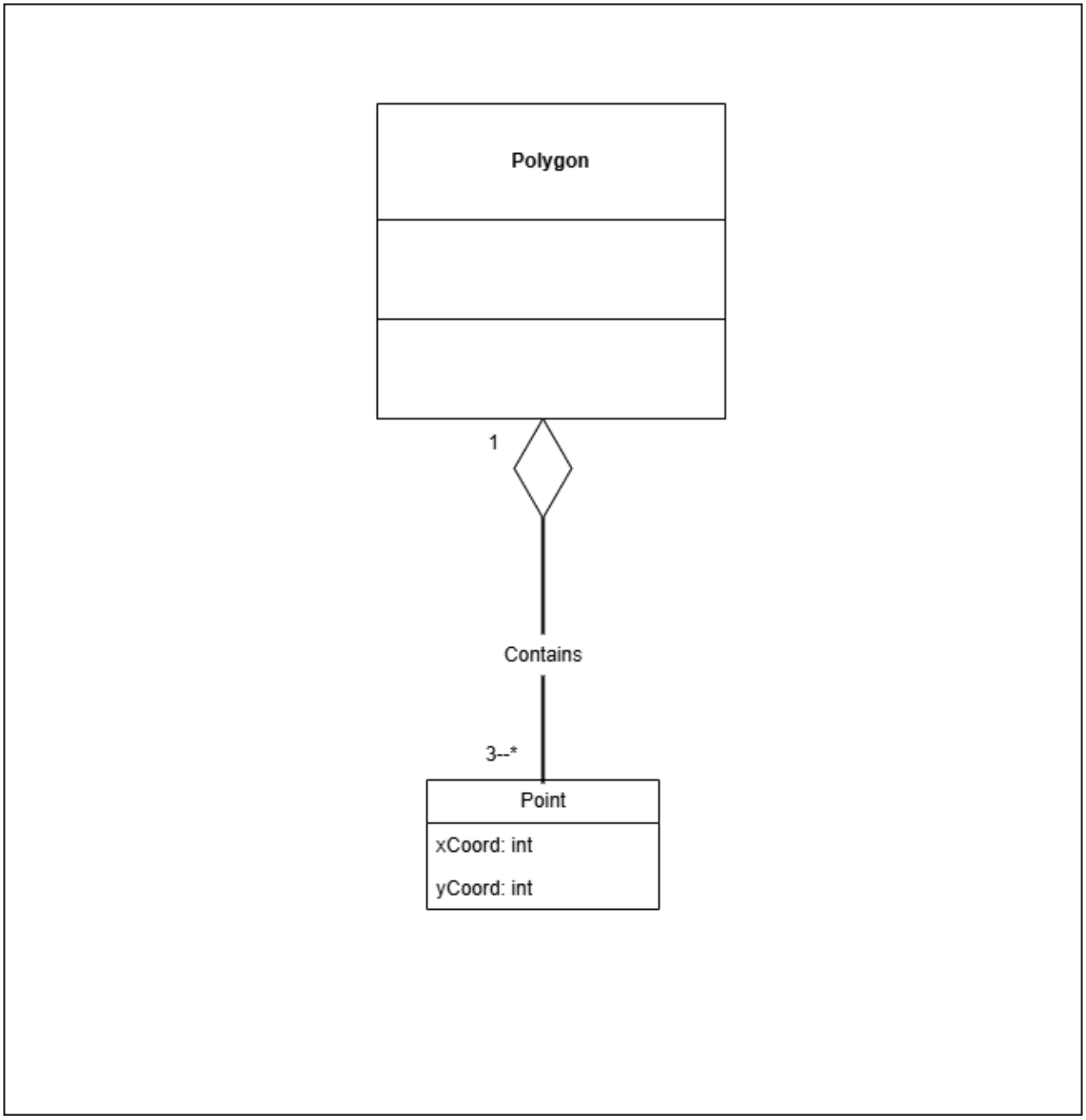
Sub Question 2: What is the least number of points needed to create a polygon?

The least number of points needed to form a polygon is 3 (as seen in a triangle).

Sub Question 3: Does sharing points between polygons affect the structure?

If points can be shared among polygons, a Point can be linked to multiple polygons, forming a many-to-many relationship between Points and Polygons.

The sequence of points is crucial as it determines the form of the polygon.


Polygon
1
Contains
3--*
Point
xCoord: int
yCoord: int

Q.3 Figure 3 is a partially completed class diagram of an air transportation system. Add multiplicities in the diagram. Also add association names to unlevelled associations.

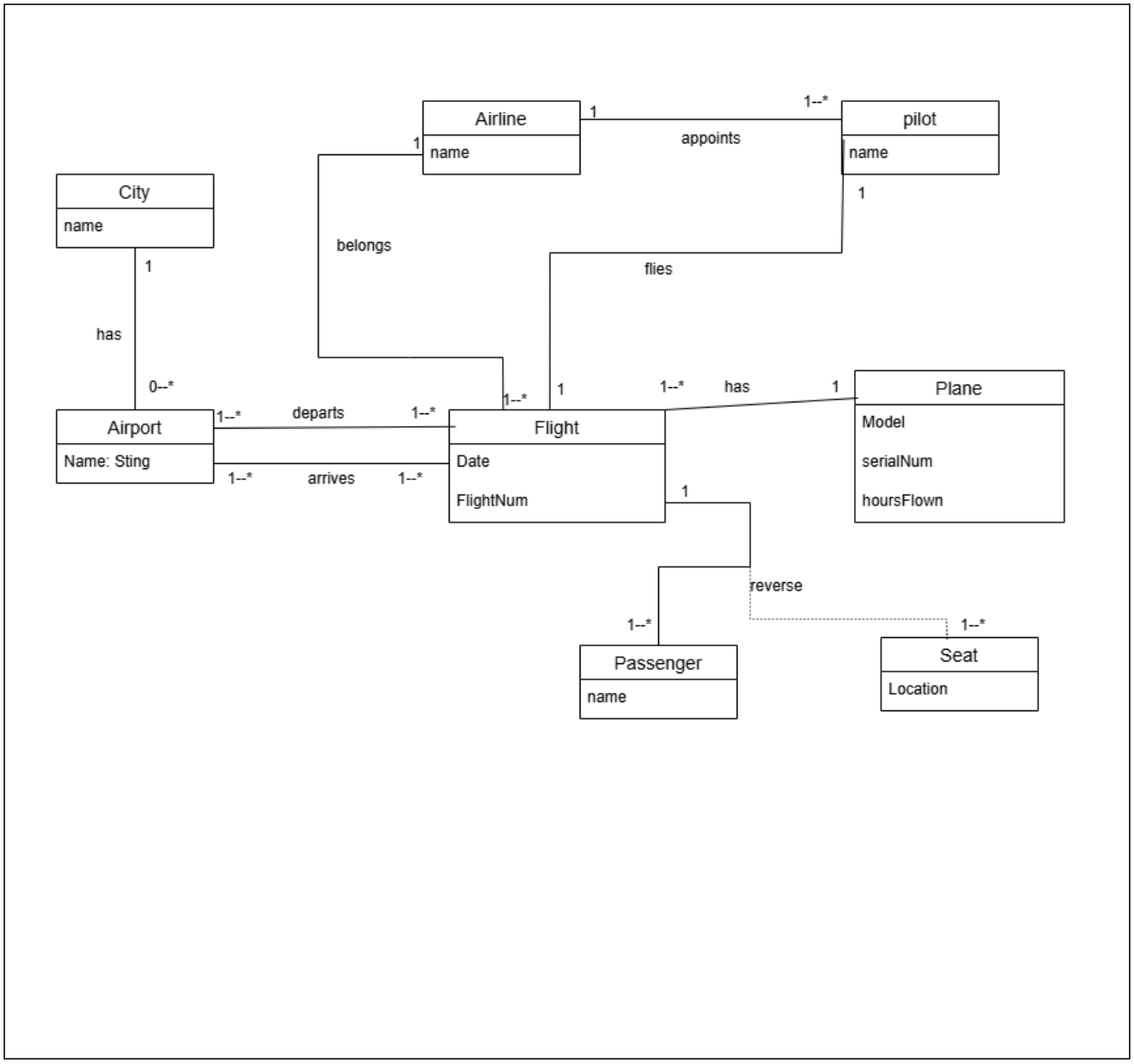
Answer:

Association Names:

- City to Airport: A City Has one or more Airports.
- Airport to Flight: A Flight Departs from or *Arrives* at an Airport.
- Airline to Flight: An Airline Operates multiple Flights.
- Airline to Pilot: An Airline Employs multiple Pilots.
- Pilot to Flight: A Pilot Pilots one or more Flights.
- Flight to Plane: A Flight Uses one Plane.
- Plane to Seat: A Plane HasSeats multiple Seats.
- Flight to Seat: A Flight *Contains* multiple Seats.
- Seat to Passenger: A Seat is *OccupiedBy* one Passenger.

Completed Diagram with Associations:

- City 0..* —— 1 Airport (*Has*)
- Airport 1 —— 0..* Flight (*Depart*, *Arrive*)
- Airline 1 —— 0..* Flight (implicitly related via *Operates*)
- Airline 1 —— 0..* Pilot (*Employs*)
- Pilot 0..* —— 1..* Flight (implicitly related via *Pilots*)
- Flight 1 —— 1 Plane (*Uses*)
- Plane 1 —— 1..* Seat (*HasSeats*)
- Flight 1 —— 1..* Seat (*Contains*)
- Seat 1 —— 0..1 Passenger (*OccupiedBy*)


Airline
name
pilot
name
appoints
1
1--*
City
name
belongs
flies
has
1
0--*
Airport
Name: Sting
departs
arrives
1--*
Flight
Date
FlightNum
Plane
Model
serialNum
hoursFlown
has
reverse
Passenger
name
Seat
Location

Q.4 We want to model a system for management of flights and pilots. An airline operates flights. Each airline has an ID. Each flight has an ID a departure airport and an arrival airport: an airport as a uniqueidentifier. Each flight has a pilot and a co-pilot, and it uses an aircraft of a certain type; a flight also has a departure time and an arrival time. An airline owns a set of aircrafts of different types. An aircraft can be in a working state or it can be under repair. In a particular moment an aircraft can be landed or airborne. A company has a set of pilots: each pilot has an experience level: 1 is minimum, 3 is maximum. A type of airplane may need a particular number of pilots, with a different role (e.g.:captain, co-pilot, navigator): there must be at least one captain and one co-pilot, and a captain must have a level 3.