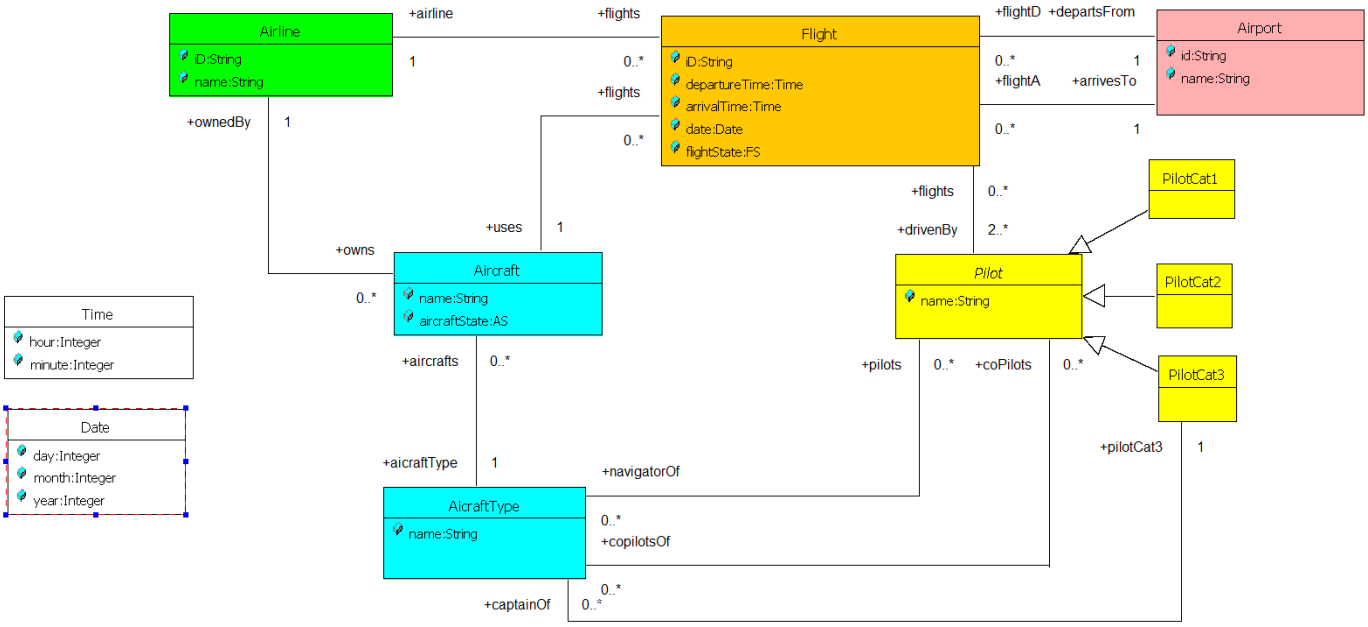
SE evaluation 2023_2

An airline operates flights. Each airline has an ID and a name.
Each flight has an ID a departureAirport and an arrivalAirport, a departureTime, an arrivalTime, a date and a status (onGround, onAir, landed). Each flight has a pilot and a least a coPilot, and uses an aircraft of a certain type, AircraftType. An airline owns a set of aircrafts of different types. Each aircraft ha a name and a state (ready, inRevision, inRepair). A company has a set of pilots: each pilot has an experience level and a set of skils. There are 3 categories of pilots: 1, 2 and 3. Pilots cat 3 are the most experienced pilots. The captain of an aircraft must be a pilot category 3. Each type of aircraft needs a number of pilots. A captain and one or more coPilots. All the pilots of an aircraft must be certified for the aircraftType corresponding to the aircraft.



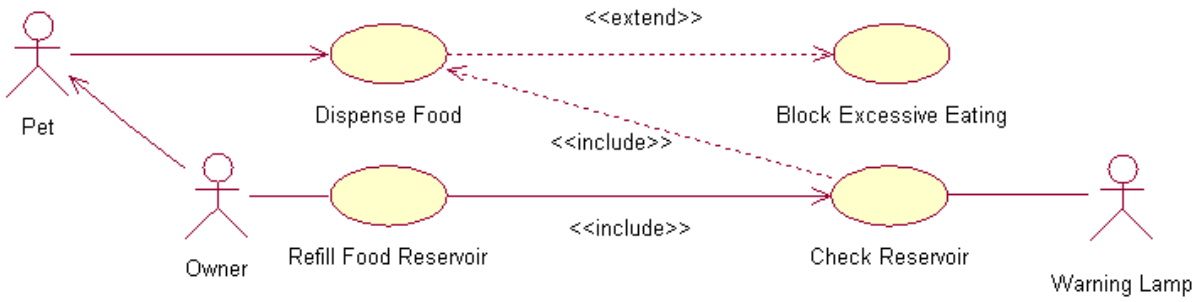
1. Using UML, please specify a class diagram complying with the above requirements. 3p
2. Using OCL, please specify an observer returning the set of tuples (airport, nbOfDepartures) of airline airports having the most departures. 1.5p

```
context Flight
    inv apropiatePilots:
        self.drivenBy->select(p | p.oclIsTypeOf(PilotCat3)).oclAsType(PilotCat3).captainOf->includesAll(self.drivenBy.navigatorOf)
        and (self.drivenBy.navigatorOf->reject(at | at.name=self.uses.aicraftType.name)->isEmpty

context Airline
    def airportsWithMostDepartures:

        let airports:Set(Airport)=self.flights.departsFrom->asSet
        let airportsNbDepartures:Set(TupleType(arp:Airport, fn:Integer))=airports->collect( a | Tuple(arp=a, fn=a.flightD->size))->asSet
        let anAirportWithMostDepartures:TupleType(arp:Airport, fn:Integer)=airports->collect( a | Tuple(arp=a, fn=a.flightD->size))->sortedBy(t | t.fn)->last
        let airportsWithMostDepartures:Set(TupleType(arp:Airport, fn:Integer))=airportsNbDepartures->select(t | t.fn = anAirportWithMostDepartures.fn)
```

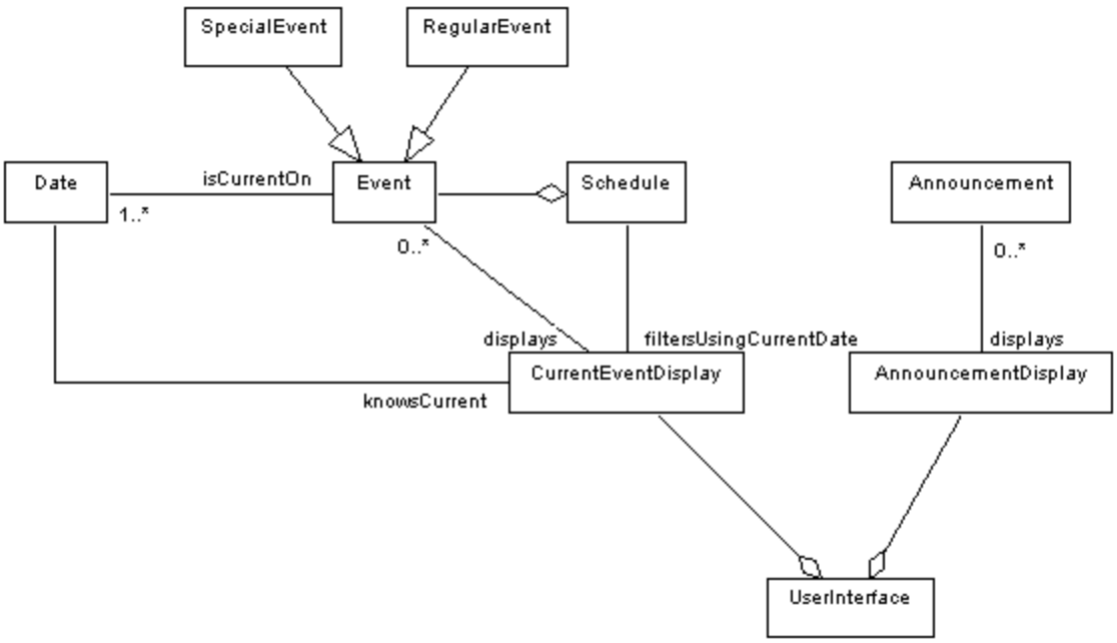
3. Consider the use case diagram below written for an automated pet feeding station. This diagram contain a number of defects. Please identify them and explain shortly. 1.5p



n a UCD between actors only inheritance relationships are represented. The extend relationship represented in the diagram is incorrect oriented. The include relationships is represented in 2 different maners. The correct representation is that using a dotted line. However, the relationship between Dispense Food and Check Reservoir is wrong oriented. The association between Pet and Dispense Food is bidirectional.

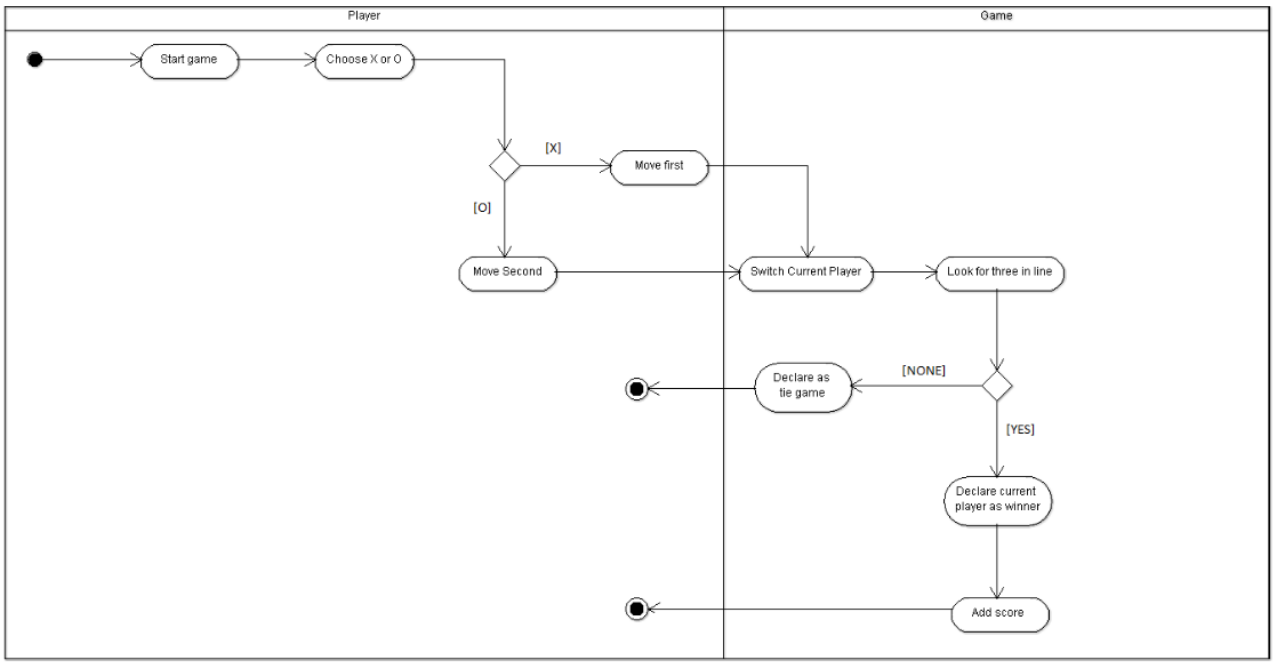
Consider the following UML class diagram. Please explain if this class diagram is correct or not related to UML specification and justify. Beyound this correctenss please consider also the model semantics. Please explain also and justify if Announcement is associated to one or more Date.

1.5p



Event is an abstract class. So its name must be written in italics. An Announcement is associated with Date by means of CurrentEventDispaly, in which case, only a date corresponds to an announcement.

Tic Tac Toe is a two-player game in which the objective is to take turns and mark the correct spaces in a 3x3 (or larger) grid. The first player who places three of their marks in a horizontal, vertical or diagonal row wins the game. Please explain/justify if the diagram below is correct or not. If not please fix the bugs. 1.5p



The activity diagram above is incorect because the number of moves was not checked. When a player chose [x] or [o], the counter c is set to 0. The minimum number of moves to win the game is 5. Moreover, the maximum number of moves is 9. We may have a tie game only at the 9th move.