Project

The crew of an aircraft consists of a pilot, a copilot and several commercial cabin crew (3C). Each of these individuals is identified by name and function. The crew of an aircraft is built for each of its flights.

Each crew member must be operational on both aircraft types (Airbus A320 and Boeing 747). Each aircraft type has a number of 3C in its crew oscillating between a minimum and a maximum. For example we will choose the number of C3 for Airbus A320 between two and three and those of a B747 between three and four.

The objective of the system model is to establish a work duty table. Crew members can visualize flights in which they were assigned. The system administrator can create and delete entities in the system. The manager can add and remove people in a crew for a specific flight (a flight is designated by a flight number and a date), the flight data are archived after flying for a year.

Figure 1 shows an extract from a work duty table of some employees of Air France.

Avion	VOL	Dest	Date	Type	Site	Prénom	Nom	Fonction
13562	AF347	Londres	11/10/06	A320	Orly	Corinne	Lancel	Pilote
13562	AF347	Londres	11/10/06	A320	Orly	Amy	Bosch	Copilote
13562	AF347	Londres	11/10/06	A320	Orly	Maureen	Gates	PNC
13562	AF347	Londres	11/10/06	A320	Orly	Richard	Tata	PNC
13562	AF347	Londres	11/10/06	A320	Orly	Ben	Gamma	PNC
32156	AF545	New-York	12/10/06	B747	Roissy	Jacques	Vlisside	Pilote
32156	AF545	New-York	12/10/06	B747	Roissy	Jean-Louis	Dupont	Copilote
32156	AF545	New-York	12/10/06	B747	Roissy	Ségolène	Orval	PNC
32156	AF545	New-York	12/10/06	B747	Roissy	François	Saadi	PNC
32156	AF545	New-York	12/10/06	B747	Roissy	Nicolas	Harper	PNC

FIGURE 1 – Extrait du tableau de service des vols AF347 et AF545 sur une période

Figure 2 shows some actors and the use cases of the system. Figure 3 shows the proposed class diagram. Figure 4 shows the state machine diagram of the "Vol" class. Figure 5 shows the communication diagram associated with the use case "affect a C3 to a flight."

Questions

1. Write the body of a Java method prototype that constructs the first five lines of the service table in Figure 1
2. Write a class containing two unit tests written using JUnit to test compliance of some classes of the described state machine. Use the setUp method with the @Before annotation.
3. Write with JUnit and using a part of the scenario in Table 1, a validation test. You can use the assertEquals class method of the Assert class with two arguments of type Object.
4. "affect a C3 to a flight." requires interactions with other classes. We want to create at least 2 integration tests to verify these interactions.

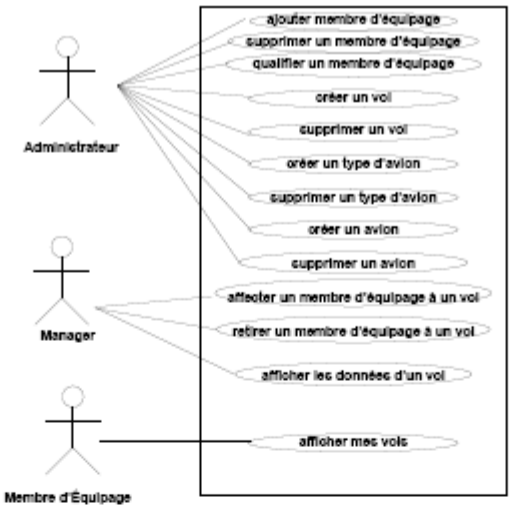

FIGURE 2 – Diagramme de cas d'utilisation

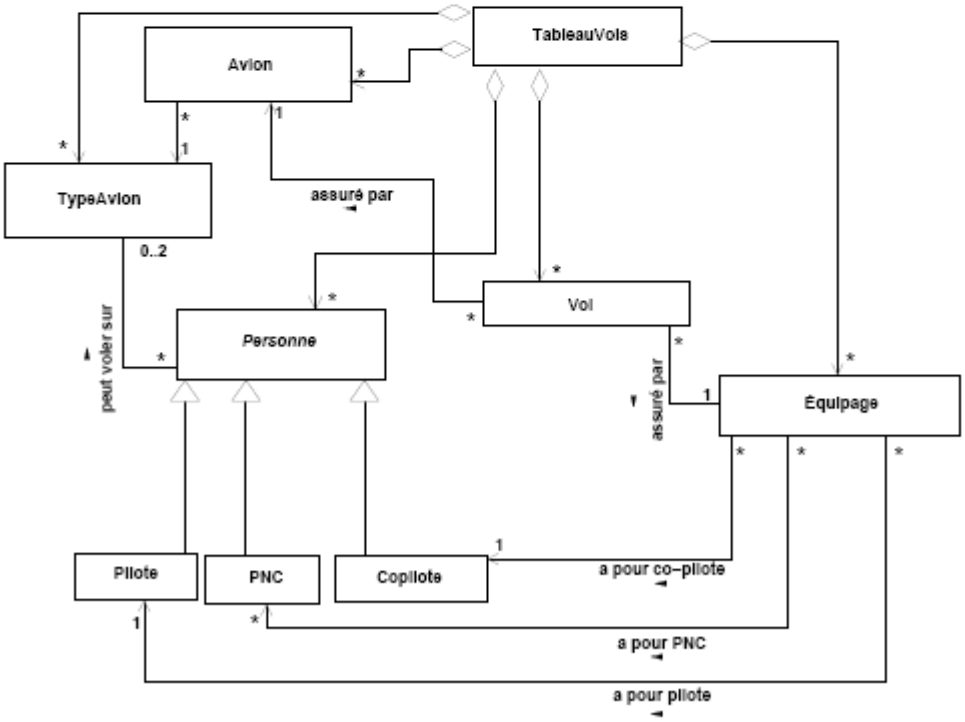

FIGURE 3 – Diagramme de classes

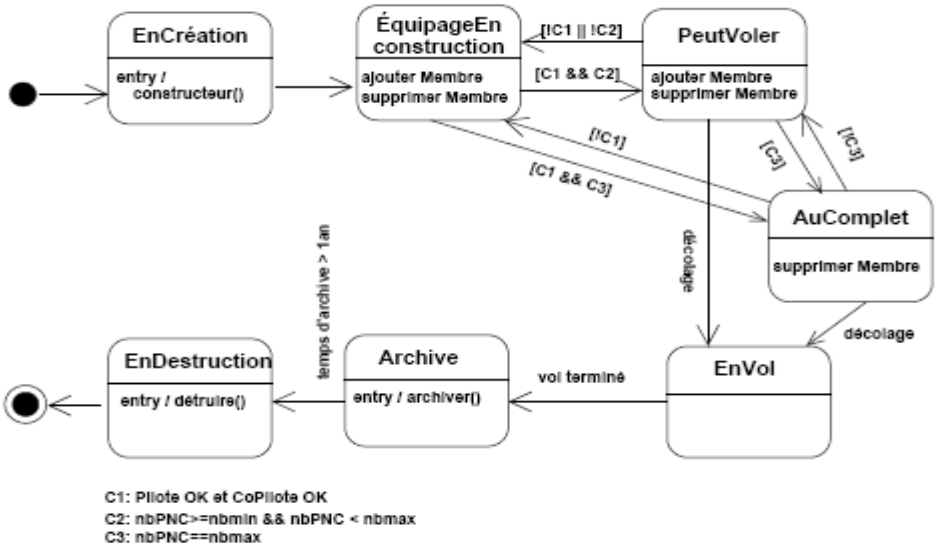

FIGURE 4 – Diagramme de machine à états de la classe « Vol »

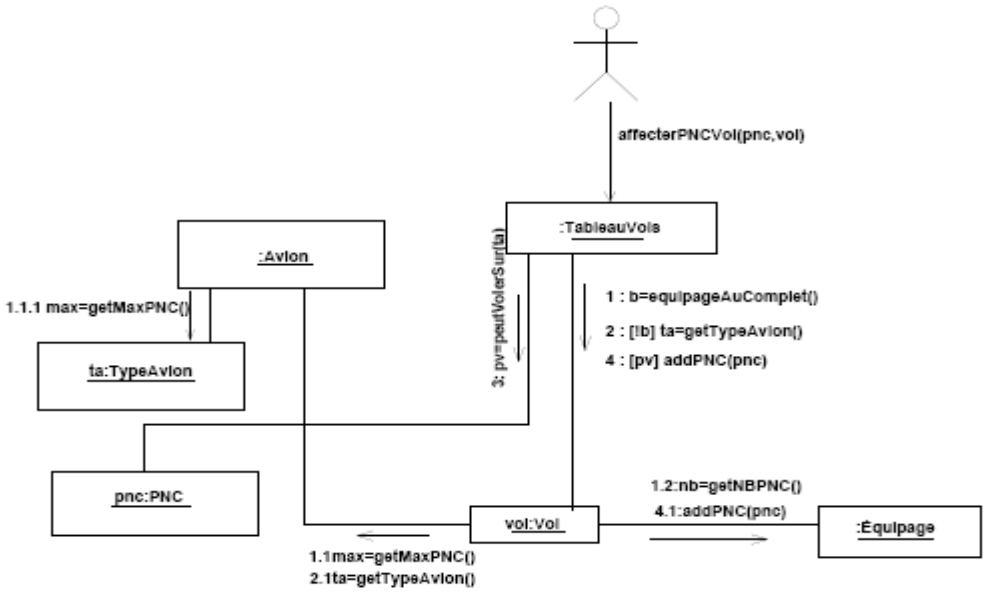

FIGURE 5 – Diagramme de communications du cas d'utilisation « affecter un PNC à un vol »

Formal modeling and verification

We would like to model the flight creation process with a simplified Petri net (based on Figure 4) and to check (at least) the following properties:

1. A flight cannot be considered as created if the number of its 3C does not meet the minimum and maximum values.
2. A flight has necessarily a pilot and a co-pilote
3. A flight cannot be archived when the plane has not landed yet
4. The process of flight creation necessarily terminates

- Use the TINA toolbox to model the system with a Petri net.
- Express the above properties (you can add others) using a temporal logic and check whether your model satisfies them.