

AUZANNEAU Maxime

CAFFY Cédric

G32

Création d’un logiciel de gestion d’aéroclub



*Année 2015-2016*

Sommaire

**Aucune entrée de table des matières n'a été trouvée.**

# Introduction

Dans le cadre de notre projet de première année de cycle d’ingénieur en informatique et modélisation, nous avons souhaité faire un logiciel de gestion d’aéroclub.

Un aéroclub est une association à but non lucratif qui a pour but de permettre la pratique du pilotage d’avions légers.

Notre logiciel doit alors permettre de gérer tout ce qui est en rapport avec les avions et les membres qui composent l’aéroclub.

En première partie, nous présenterons la partie analyse de notre projet. En deuxième partie nous expliquerons les étapes du développement de notre application (élaboration des interfaces graphiques, base de données…). Enfin, nous aborderons les difficultés rencontrées au sein du projet.

# La partie « analyse » du projet

L’analyse est une phase très importante au sein d’un projet. Elle permet de faire un lien entre les développeurs et l’utilisateur final, mais permet également d’avoir une vue d’ensemble du projet afin que les développeurs puissent avoir un même socle de départ.

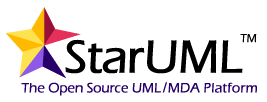
## Cahier des charges

A FAIRE

## L’analyse UML

L’analyse UML (Unified Modeling Language) permet de fournir une méthode normalisée afin de visualiser la conception d’un système. Nous avons opté pour ce type d’analyse car nous connaissons son efficacité de par notre expérience antérieure.

Nous avons réalisé notre analyse UML à l’aide du logiciel Star UML.



### Le diagramme de Use Case

Ce diagramme permet de décrire les cas d’utilisations du logiciel. Il permet de répondre de manière synthétique aux questions « Que peut faire le logiciel ? »,  « Qui peut faire telle ou telle action ? », « Qu’est-il nécessaire de faire pour effectuer une action ? ».

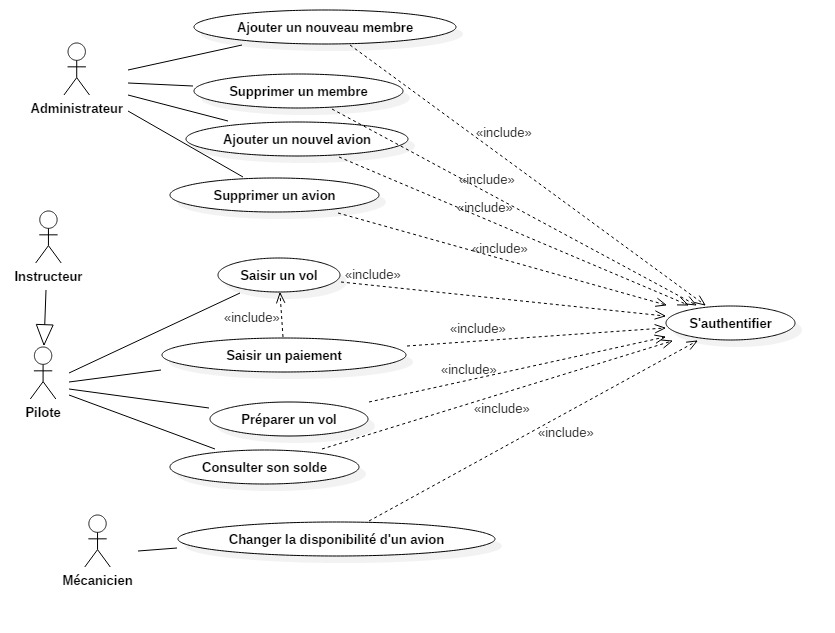
Voici le diagramme de Use Case que nous avons réalisé :

Figure 1:Le diagramme de Use Case de notre logiciel

#### Description détaillée de chaque Use Case

##### Use case « Ajouter un nouveau membre »

Objectif : Inscription d’un nouveau membre dans le logiciel

Acteur principal : l’administrateur

Préconditions : L’administrateur doit être authentifié

Post conditions : Un nouveau membre sera ajouté

##### Use case « ajouter un nouvel avion »

Objectif : Ajouter un nouvel avion dans le logiciel

Acteur principal : l’administrateur

Préconditions : L’administrateur doit être authentifié

Post conditions : Un nouvel avion sera ajouté à la flotte de l’aéroclub

##### Use case « Saisir un vol »

Objectif : Saisie d’un vol lors du retour d’un pilote

Acteur principal : le pilote

Préconditions : le pilote doit être authentifié

Post conditions : Le compte pilote sera débité du montant de l’heure de vol

##### Use case « Saisir un paiement

Objectif : Saisie d’un paiement pour réapprovisionner le compte pilote du membre

Acteur principal : le pilote

Préconditions : le pilote doit être authentifié

Post conditions : le compte pilote sera crédité du montant de l’heure de vol

##### Use case « Supprimer un membre »

Objectif : Supprimer un membre de la base de données

Acteur principal : l’administrateur

Préconditions : l’administrateur doit être authentifié et le membre doit exister

Post conditions : Toutes les informations relatives au membre seront supprimées.

##### Use case « Supprimer un avion »

Objectif : Suppression d’un avion de la base de données

Acteur principal : l’administrateur

Préconditions : l’administrateur doit être authentifié et l’avion doit exister

Post conditions : Toutes les informations relatives à l’avion seront supprimées

##### Use case « Consulter un solde »

Objectif : Consulter le solde d’un membre

Acteur principal : le membre

Préconditions : Le membre doit être authentifié

Post conditions : aucune

##### Use case « Préparer un vol »

Objectif : Aider le pilote à préparer son futur vol

Acteur principal : le pilote

Préconditions : Le pilote doit être authentifié

Post conditions : aucune

##### Use case « Changer la disponibilité d’un avion

Objectif : Permet au mécanicien de changer l’état d’un avion (en état de vol, en réparation, en révision)

Acteur principal : le mécanicien

Préconditions : le mécanicien doit être authentifié

Post conditions : aucune

### Le diagramme de classe

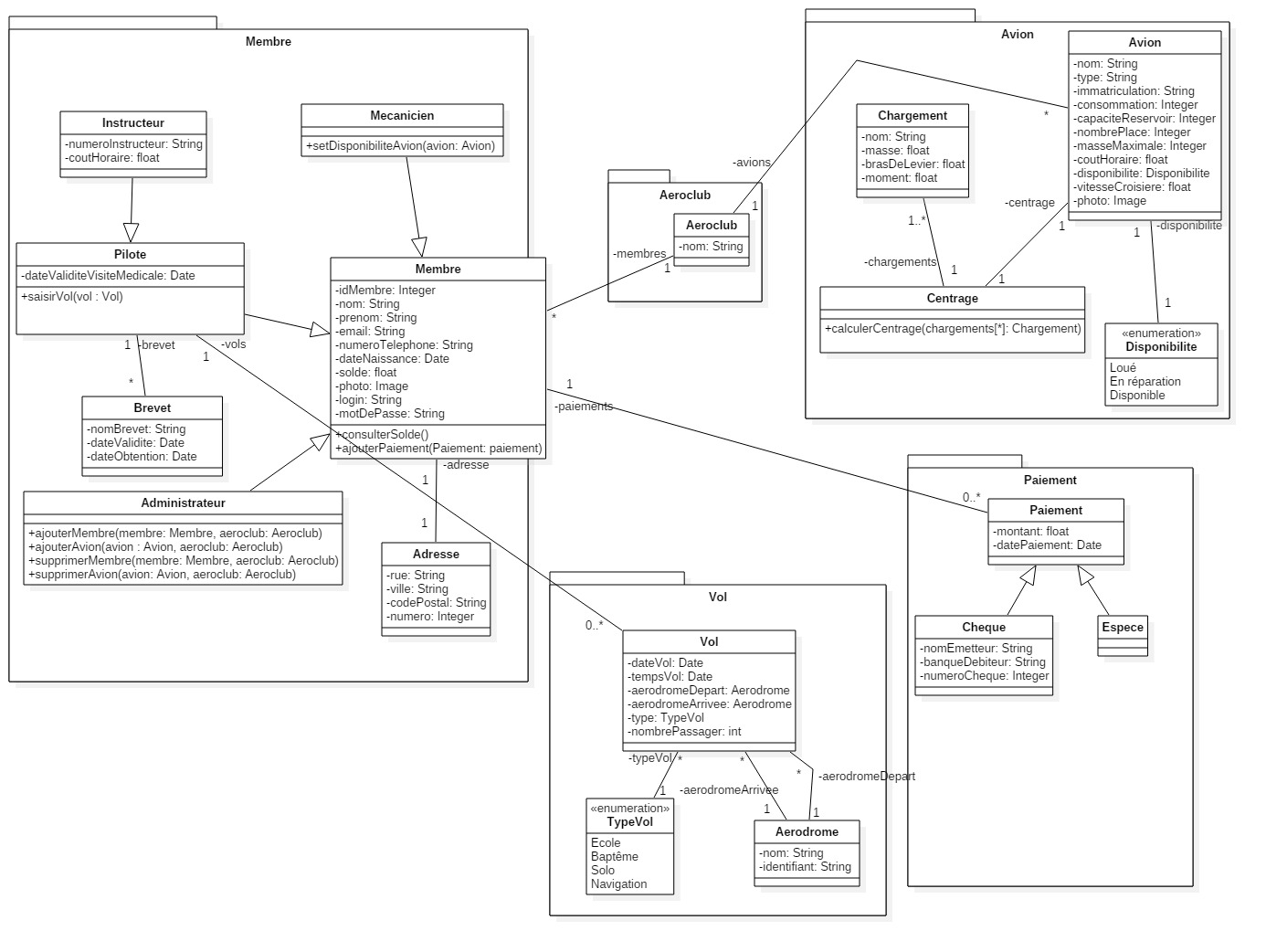
Le diagramme de classe permet d’avoir une vue d’ensemble des différentes classes métiers de notre application. Il permet également de faciliter l’implémentation de ces classes car les liaisons entre chaque classe sont visibles d’un seul coup d’œil.

Figure 2:Le diagramme de classe de notre application

A CONTINUER