|  |
| --- |
| Colcinet | Colson J. – Ciliberto A. – Minet M. |
| Projet Java : Itération 4 |
| Colcinet Flights |

|  |
| --- |
| HELHa Montignies-sur-Sambre  Section économique  Informatique de Gestion  Année académique 2017 - 2018 |

Table des matières

[1. Analyse globale 2](#_Toc513587332)

[2. Exigences du client (Requirements) 3](#_Toc513587333)

[3. Cas d’utilisation des différentes itérations 4](#_Toc513587334)

[4. Diagramme de cas d’utilisation 5](#_Toc513587335)

[5. Distributions des rôles par itération 6](#_Toc513587336)

[6. Modèle complet 8](#_Toc513587337)

[7. Diagramme de classe simplifié 12](#_Toc513587338)

[8. Diagramme de packages 13](#_Toc513587339)

[9. Schéma relationnel de la base de données 14](#_Toc513587340)

[10. Jeu de tests 15](#_Toc513587341)

# Projet Java : Itération 1

## Analyse globale

* 1. Le nom choisi pour l’application à réaliser sera Colcinet Flights, tiré du nom du groupe de travail et de la fonction de l’application.
  2. Cette application permettra d’enregistrer et modifier les informations des pilotes inscrits, d’enregistrer et modifier les informations concernant les vols effectués, de lister toutes ces informations selon un ordre de tri spécifique et d’en afficher le résultat.
  3. Glossaire des terminologies utilisées
     + Le Gestionnaire : gère le club de vol à voile et est le seul acteur à interagir avec l’application dans le cadre de la gestion du club et des données de ses membres.
     + Le compte pilote : compte propre à chaque pilote, alimenté par le gestionnaire à la réception d’un paiement du pilote concerné. Un solde suffisant sur ce compte est nécessaire afin de permettre à un pilote de prévoir un vol. Ce compte est débité directement à l’enregistrement d’un nouveau vol pour le pilote concerné.
     + La planche de vol : reçue en fin de journée par le Gestionnaire, elle reprend les informations des différents vols effectués sur la journée, ainsi que les informations à sauvegarder les concernant.
  4. Détails des acteurs et de leurs responsabilités :
     + Le seul acteur identifié dans le cadre de ce projet est le *Gestionnaire,* sa responsabilité sera d’interagir avec l’application afin de sauvegarder et modifier toutes les données nécessaires à la bonne gestion du club de vol à voile.

## Exigences du client (Requirements)

* L’application commandée par le client devra être capable d’effectuer les tâches suivantes :

2.1. Permettre au Gestionnaire d’ajouter de nouveaux pilotes et de sauver les informations nécessaires les concernant.

* + - Adresse email, nom, prénom, adresse, numéro de gsm, solde du compte pilote.

2.2. Lister les pilotes déjà ajoutés au préalable dans une liste triée sur le nom et le prénom

2.3. Lister les pilotes dont le solde du compte pilote est en négatif. Cette liste sera triée du solde le plus négatif au solde le moins négatif.

2.4. Sélectionner un pilote depuis les listes des points 2.2 et 2.3 et permettre une modification de leurs données personnelles sauvegardées.

2.5. Permettre au Gestionnaire, à partir de la planche de vol reçue en fin de journée, de sauvegarder les données des vols effectués.

* + - Nom et prénom du pilote, durée de vol, type de planeur utilisé.

2.6. A chaque nouveau vol sauvegardé, le compte pilote du membre concerné sera automatiquement débité du montant calculé.

* + - Le coût d’un vol se calcule comme ceci :
      * Coût de remorquage (fixe) = 25€
      * Coût variable en fonction de la durée du vol et du type de planeur :
        + 17€/h : planeur bois & toile
        + 27€/h : planeur plastique
        + 30€/h : planeur biplace

2.7. Lister les vols effectués à une date précise. Cette liste sera triée en fonction de la durée du vol, du plus long au plus bref.

* Le client a également émis l’exigence de l’utilisation du système de gestion de bases de données PostgreSQL dans l’implémentation de la couche de persistance.

## Cas d’utilisation des différentes itérations

3.1. Itération 2

* + - Use case n°1 : Enregistrer un nouveau pilote (requirement 2.1)
      * Ce use case concerne la persistance des données. Son implémentation devra permettre au gestionnaire l’ajout de nouveaux pilotes et de leurs données personnelles en base de données.

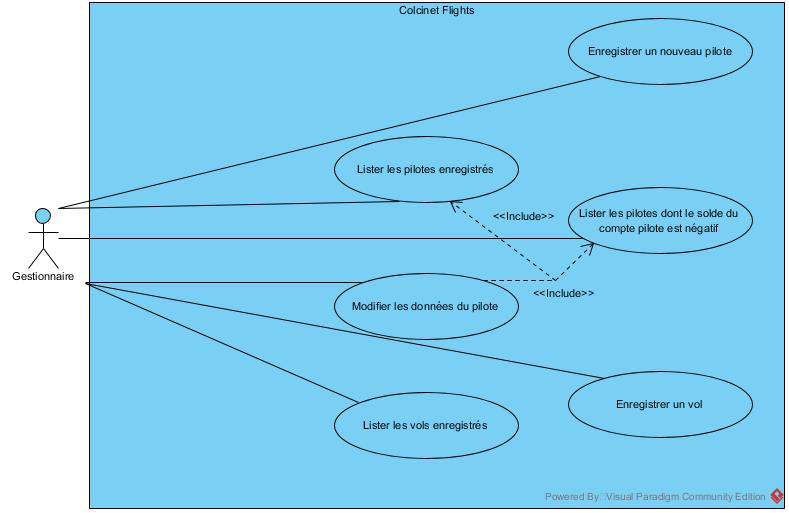
3.2. Itération 3

* + - Use case n°2 : Lister les pilotes enregistrés (requirement 2.2)
      * Permet d’aller rechercher les données personnelles des pilotes sauvées en base de données et de les afficher dans une liste triée sur le nom et le prénom.
    - Use case n°3 : Lister les pilotes dont le solde du compte pilote est négatif (requirement 2.3)
      * Crée une liste composée de tous les pilotes dont le solde du compte pilote est en négatif. Cette liste sera triée sur ce solde, du plus négatif au moins négatif.
    - Use case n°4 : Modifier les données du pilote (requirement 2.4)
      * Permet la modification des données personnelles des pilotes stockées en base de données, ainsi que la modification du solde de leur compte pilote en cas de paiement de leur part.

3.3. Itération 4

* + - Use case n°5 : Enregistrer un vol (requirement 2.5)
      * Permet de stocker les données des vols présents sur la planche de vol en fin de journée. Ces données seront enregistrées en base de données.
    - Use case n°6 : Lister les vols enregistrés (requirement 2.7)
      * Crée une liste à partir des vols préalablement sauvegardés. Cette liste reprendra les vols effectués pendant la journée dont la date aura été entrée par le Gestionnaire.

## Diagramme de cas d’utilisation



On distingue donc un seul acteur : le Gestionnaire, seul à interagir avec l’application. Les différents use cases établis durant l’analyse sont :

* Enregistrer un nouveau pilote – use case n°1
* Lister les pilotes enregistrés – use case n°2
* Lister les pilotes dont le solde du compte pilote est négatif – use case n°3
* Modifier les données du pilote – use case n°4
* Enregistrer un vol – use case n°5
* Lister les vols enregistrés – use case n°6

## Distributions des rôles par itération

5.1. Itération 1

* Ciliberto Angelo :
  + Ecriture de l’analyse globale.
  + Participation aux listings des requirements et des use cases par itération, ainsi qu’aux descriptions du schéma relationnel de la base de données et du diagramme de packages.
  + Réalisation du diagramme de use cases.
  + Distributions des rôles par itération.
  + Relecture et mise en page du contenu.
* Colson Junior :
  + Réalisation du diagramme de classe.
  + Réalisation du diagramme de packages.
  + Réalisation du schéma relationnel de la base de données et de tous les modèles antérieurs nécessaires.
* Minet Maxime :
  + Listing des requirements.
  + Listing des use cases par itération.
  + Ecriture du modèle complet.

5.2. Itération 2

* Ciliberto Angelo :
  + Correction des points 1 à 5 du rapport de l’itération 1.
  + Modification des diagrammes de use cases, de classe, et du schéma relationnel de la base de données.
  + Création et mise en relation des classes de l’IHM.
* Colson Junior :
  + Réalisation des tests d’intégration, unitaires et fonctionnels.
  + Correction du point 9 du rapport de l’itération 1.
  + Réalisation du jeu de tests.
* Minet Maxime :
  + Correction des points 6 et 8 du rapport de l’itération 1.
  + Implémentation des packages dao, usecase, usecaseimpl et domaine.

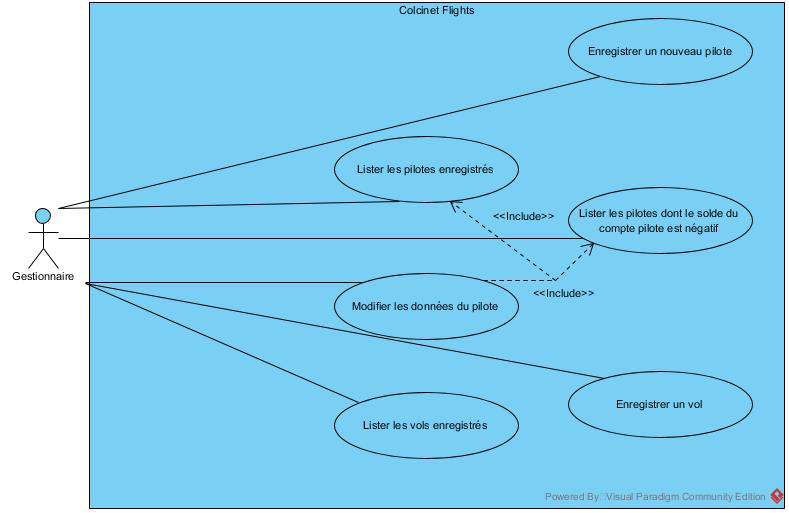
5.3. Itération 3

* Ciliberto Angelo :
  + Correction du rapport pour les itérations 1 et 2.
  + Ecriture d’une partie des jeux de tests.
  + Modifications sur l’IHM.
* Colson Junior :
  + Implémentation des use cases, des mockimpl, des daoimpl ainsi que des tests unitaires et d’intégration.
  + Ecriture d’une partie des jeux de tests.
* Minet Maxime :
  + Implémentation des mock objects de l’itération.

5.4. Itération 4

* A déterminer…

## Modèle complet



**Enregistrer un nouveau pilote – use case n°1**

Acteur : Le Gestionnaire du club.

Pré-condition : On souhaite ajouter un nouveau pilote à la base de données.

Scénario : Le Gestionnaire du club lance le programme ColcinetFlights et clique sur « Gestion des pilotes ». L’application lui affiche alors, dans la partie gauche de la fenêtre, le canevas du tableau permettant d’afficher la liste des pilotes déjà enregistrés. Il clique ensuite sur le bouton « Ajouter un pilote » et le formulaire d’ajout apparait dans la partie droite de la fenêtre. Après avoir renseigné les informations du pilote dans les champs prévus à cet effet (adresse email, nom, prénom, numéro de gsm, adresse et solde du compte pilote), il doit cliquer sur le bouton « Sauver » qui, après confirmation du Gestionnaire, demandera à l’application de procéder à une vérification des données avant de les sauvegarder en base de données.

Post-condition : Si les données renseignées correspondent aux types attendus par l’application, une nouvelle entrée est créée en base de données et les données y sont sauvées.

**Lister les pilotes enregistrés – use case n°2**

Acteur : Le Gestionnaire du club.

Pré-condition : La base de données contient un ou plusieurs pilotes, enregistrés préalablement par le Gestionnaire.

Scénario : Le Gestionnaire du club lance le programme ColcinetFlights et clique sur « Gestion des pilotes ». L’application lui affiche alors, dans la partie gauche de la fenêtre, le canevas du tableau permettant d’afficher la liste des pilotes déjà enregistrés. Il clique ensuite sur le bouton « Lister par nom » qui va permettre d’interroger la base de données et de garnir le tableau de la partie gauche de la fenêtre avec la liste des pilotes déjà enregistrés, triée sur le nom et le prénom.

Scénario alternatif : Dans le cas où il n’existerait encore aucun pilote enregistré dans la base de données, l’applications ouvrira une fenêtre modale, spécifiant qu’aucun pilote n’a encore été sauvé en base de données.

Post-condition : Néant.

**Lister les pilotes dont le solde du compte est négatif – use case n°3**

Acteur : Le Gestionnaire du club.

Pré-condition : Il existe un ou plusieurs pilotes sauvés en base de données dont le solde du compte pilote est négatif.

Scénario : Le Gestionnaire du club lance le programme ColcinetFlights et clique sur « Gestion des pilotes ». L’application lui affiche alors, dans la partie gauche de la fenêtre, le canevas du tableau permettant d’afficher la liste des pilotes déjà enregistrés. Il clique ensuite sur le bouton « Lister les soldes négatifs » qui va permettre d’interroger la base de données et de garnir le tableau de la partie gauche de la fenêtre avec la liste des pilotes dont le solde du compte pilote est négatif, trié du plus négatif au moins négatif.

Scénario alternatif : Dans le cas où il n’existerait aucun pilote enregistré dans la base de données, ou bien que chaque pilote y étant sauvé possède un solde de son compte pilote positif, l’application ouvrira une fenêtre modale spécifiant qu’aucune entrée de la base de données ne correspond à la requête.

Post-condition : Néant.

**Modifier un pilote – use case n°4**

Acteur : Le Gestionnaire du club.

Pré-conditions :

* Les données personnelles de l’un des pilotes ont changé ou un virement visant à créditer le compte pilote d’un des pilotes a été reçu.
* L’un des use cases permettant de lister les pilotes enregistrés en base de données, soit triés sur le nom et prénom (use case n°2) soit pas solde, du plus négatif au moins négatif (use case n°3) a au préalable été réalisé.

Scénario : Une fois toutes les préconditions remplies, le Gestionnaire sélectionne le pilote concerné dans la liste affichée et clique sur le bouton « Modifier pilote ». L’application crée alors un formulaire dont les champs sont garnis des données du pilotes récupérées en base de données, et l’affiche dans la partie droite de la fenêtre. Le Gestionnaire peut alors effectuer toutes les modifications de données nécessaires avant de cliquer sur le bouton « Sauver ». Dans le cas où l’un des champs ne correspond pas au type de données attendu par l’application, une fenêtre modale indique au Gestionnaire quel champ du formulaire est en défaut. Si toutes les données sont conformes aux types attendus, l’application ouvre une fenêtre de confirmation à valider par le Gestionnaire.

Post-condition : Une fois les données valides et la modification confirmée par le Gestionnaire, les informations de la base de données sont mises à jour ou remplacées par les nouvelles entrées.

**Enregistrer un vol – use case n°5**

Acteur : Le Gestionnaire du club.

Pré-condition : Le Gestionnaire reçoit une planche de vol renseignant les différents vols à sauvegarder dans la base de données.

Scénario : Le Gestionnaire du club lance le programme ColcinetFlights et clique sur « Gestion des vols ». L’application lui affiche alors, dans la partie gauche de la fenêtre, le canevas du tableau permettant d’afficher la liste des vols déjà enregistrés. Il clique ensuite sur le bouton « Ajouter un vol » et le formulaire d’ajout apparait dans la partie droite de la fenêtre. Après avoir renseigné les informations du vol dans les champs prévus à cet effet (La date du vol, sa durée, le type de planeur utilisé, le nom du pilote), il doit cliquer sur le bouton « Sauver » qui, après confirmation du Gestionnaire, demandera à l’application de procéder à une vérification des données avant de les sauvegarder en base de données.

Post-condition : Si les données renseignées correspondent aux types attendus par l’application, une nouvelle entrée est créée en base de données et les données y sont sauvées. Le solde du compte pilote attaché au pilote ayant effectué le vol enregistré sera automatiquement mis à jour à la sauvegarde du vol.

**Lister les vols enregistrés – use case n°6**

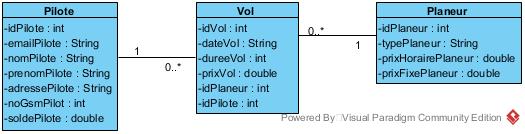
Acteur : Le Gestionnaire du club.

Pré-condition : La base de données contient un ou plusieurs vols, enregistrés préalablement par le Gestionnaire.

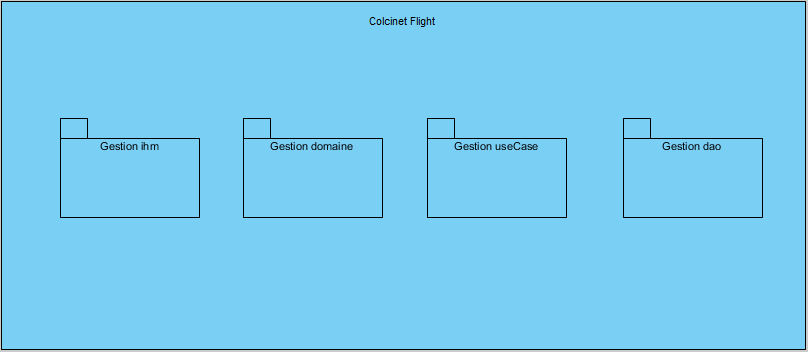
Scénario : Le Gestionnaire du club lance le programme ColcinetFlights et clique sur « Gestion des vols ». L’application lui affiche alors, dans la partie gauche de la fenêtre, le canevas du tableau permettant d’afficher la liste des vols déjà enregistrés. Il clique ensuite sur le bouton « Lister les vols sauvés » qui va permettre d’interroger la base de données et de garnir le tableau de la partie gauche de la fenêtre avec la liste des vols déjà enregistrés, triée sur la durée de vol, du plus long au plus court. Cette fonction demande au Gestionnaire d’entrer une date, afin d’afficher uniquement les vols effectuer à la date renseignée. Dans le cas où il n’entre aucune date, l’application affichera la liste de tous les vols sauvés en base de données.

Post-condition : Néant.

## Diagramme de classe simplifié



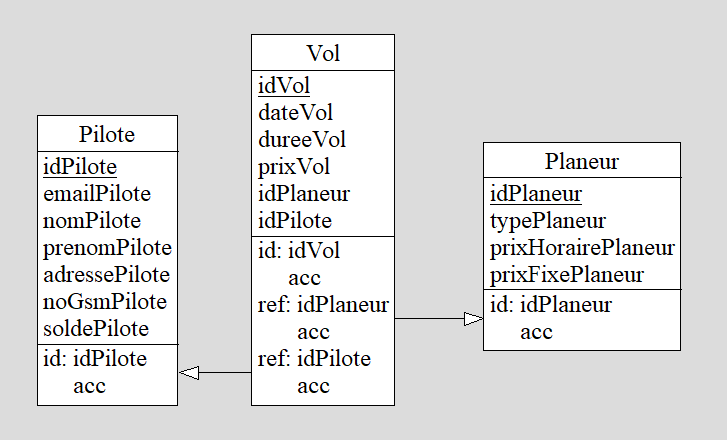
## Diagramme de packages



Nous avons identifié quatre packages dans notre analyse :

* Le package Gestion ihm, qui reprendra toutes les implémentations des interfaces graphiques (Listes, fenêtres, etc…). Ce package représente la couche présentation.
* Le package Gestion domaine, qui reprendra tous les modèles de données de chaque table. Ce package représente la couche logique.
* Le package Gestion useCase, qui reprendra les implémentations des différents use cases relatifs à l’enregistrement et au listage des informations des vols et des pilotes enregistrés par le Gestionnaire. Ce package représente la couche métier.
* Le package Gestion dao, qui reprendra toutes les implémentations des données en « mock objects » ou en base de données. Ce package représente la couche persistance.

## Schéma relationnel de la base de données



Notre analyse nous a conduit à une implémentation de quatre tables en base de données, permettant d’assurer la persistance des données.

Les tables seront :

* + - La table Pilote, reprenant toutes les informations personnelles concernant le pilote, ainsi que le solde de son compte pilote.
    - La table Vol, reprenant tous les vols ayant été effectués et enregistrés dans le système par le Gestionnaire.
    - La table Planeur, reprenant les différents types de planeurs disponibles pour effectuer un vol, ainsi que le tarif horaire y étant associé.

## Jeu de tests

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **Description** | **Use case** | **Directives** | **Résultat attendu** | **Résultat obtenu** |
| 1 | Ajout pilote  avec  informations valides | 1 | * Cliquez sur « Gestion pilotes ». * Cliquez sur « Ajouter un pilote ». | Le formulaire d’ajout apparaît. |  |
| 2 | * Remplissez le formulaire avec les informations suivantes :   + Email : [a.ciliberto@gmail.com](mailto:a.ciliberto@gmail.com) ;   + Nom : Ciliberto ;   + Prénom : Angelo ;   + Gsm : 0491238067 ;   + Adresse : Rue Bascoup, 71 Anderlues ;   + Solde : 200 . * Cliquez sur « Sauver », puis « Valider ». | Après vérification des informations entrées, l’application confirme l’ajout du nouveau pilote en db. |  |
| 3 | Lister les pilotes précédemment enregistrés | 2 | * Cliquez sur « Gestion des pilotes ». * Cliquez sur « Lister par nom ». | L’application affiche un tableau contenant la liste des pilotes enregistrés en db. On constate que le pilote ajouter lors du test précédent figure dans la liste et que celle-ci est bien triée sur les noms et prénoms des pilotes. |  |
| 4 | Ajout pilote  avec  informations non-valides | 1 | * Cliquez sur « Gestion pilotes ». * Cliquez sur « Ajouter un pilote ». | Le formulaire d’ajout apparait. |  |
| 5 | * Remplissez le formulaire avec les informations suivantes :   + Email : colson.com ;   + Nom : Colson ;   + Prénom : Junior ;   + Gsm : 0475445673 ;   + Adresse : Rue du parc, 63 Charleroi ;   + Solde : - 45.30 . * Cliquez sur « Sauver », puis « Valider ». | Après vérification des données entrées, l’application affiche un message d’erreur spécifiant que les données entrées sont invalides pour être sauvées en db. L’adresse email ne correspond en effet pas au format attendu de [\*@\*.\*](mailto:*@*.*) . |  |
| 6 | Lister les pilotes dont le solde du compte pilote est négatif | 3 | * Cliquez sur « Gestion des pilotes ». * Cliquez sur « Lister les soldes négatifs ». | L’application affiche un tableau contenant la liste des pilotes enregistrés en db et dont le solde du compte pilote est négatif. On constate que cette liste est triée du solde le plus négatif, au solde le moins négatif. |  |
| 7 | Modifier les données d’un pilote | 4 | * Sélectionnez le pilote « Ciliberto Angelo » et cliquez sur « Modifier pilote ». * Modifiez le champ solde par la valeur « -10 ».   Cliquez sur « Sauver », puis « Valider ». | Après vérification des informations entrées, l’application confirme la modification des données du pilote en db. |  |
| 8 | Lister les pilotes dont le solde est négatif, afin de vérifier les modifications | 3 | * Cliquez sur « Gestion des pilotes ». * Cliquez sur « Lister les soldes négatifs ». | L’application affiche un tableau contenant la liste des pilotes enregistrés en db et dont le solde du compte pilote est négatif. On constate que le pilote « Ciliberto Angelo » a rejoint cette liste et que son solde est bien de – 10. |  |
| 9 | Tenter de modifier le nom d’un pilote | 4 | * Sélectionnez le pilote « Colson Junior » et cliquez sur « Modifier pilote ». * Tentez de modifier le champ nom. | On constate que la modification du nom d’un pilote est impossible, l’application bloquant la modification du champ. Tous les autres champs sont, eux, modifiables. |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 10 | Enregistrer un nouveau vol avec des données valides | 5 | * Cliquez sur « Gestion des vols ». * Cliquez sur « Ajouter un vol ». | Le formulaire d’ajout apparait. |  |
| 11 | * Entrez les informations suivantes :   + Sélectionnez la date d’aujourd’hui comme date de vol ;   + Durée : 68 (la durée est exprimée en minutes) ;   + Type : plastique (à sélectionner dans la liste proposée) ;   + Pilote : Colson Junior (à sélectionner dans la liste proposée). * Cliquez sur « Sauver », puis sur « Valider ». | Après vérification des informations entrées, l’application confirme la modification des données du vol en db. Le solde du compte du pilote concerné sera automatiquement débité du prix du vol. |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 12 | Enregistrer un nouveau vol avec des données invalides | 5 | * Cliquez sur « Gestion des vols ». * Cliquez sur « Ajouter un vol ». | Le formulaire d’ajout apparait. |  |
| 13 | * Entrez les informations suivantes :   + Sélectionnez la date d’aujourd’hui comme durée de vol ;   + Durée : -12 (la durée est exprimée en minutes) ;   + Type : plastique (à sélectionner dans la liste proposée) ;   + Pilote : Ciliberto Angelo (à sélectionner dans la liste proposée). * Cliquez sur « Sauver », puis « Valider ». | Après vérification des données entrées, l’application affiche un message d’erreur spécifiant que les données entrées sont invalides pour être sauvées en db. On ne peut, en effet pas sauver un vol dont la durée est négative. |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 14 | Lister les vols enregistrés à une date précise | 6 | * Cliquez sur « Gestion des vols ». * Cliquez sur « Lister les vols ». | L’application demande d’entrer une date pour lister les vols de ce jour. |  |
| * Entrez la date du 30-01-18 et cliquez sur « Valider ». | L’application liste les vols sauvés en db et ayant été effectués à la date renseignée. |  |
| 15 | Lister tous les vols enregistrés | * Cliquez sur « Gestion des vols ». * Cliquez sur « Lister les vols ». | L’application demande d’entrer une date pour lister les vols de ce jour. |  |
| * N’entrez aucune date et cliquez sur « Valider ». | L’application liste tous les vols sauvés en db. |  |