|  |
| --- |
| Application de planification de vols aériens |

Gianinetti Lucas

Avenue de la gare 2

1450 Ste-Croix

[1 Analyse préliminaire 3](#_Toc499021832)

[1.1 Introduction 3](#_Toc499021833)

[1.2 Objectifs 3](#_Toc499021834)

[1.3 Planification initiale 4](#_Toc499021835)

[2 Analyse / Conception 5](#_Toc499021836)

[2.1 Concept 5](#_Toc499021837)

[2.2 Stratégie de test 13](#_Toc499021838)

[2.3 Risques techniques 14](#_Toc499021839)

[2.4 Planification 14](#_Toc499021840)

[2.5 Dossier de conception 14](#_Toc499021841)

[3 Réalisation 17](#_Toc499021842)

[3.1 Dossier de réalisation 26](#_Toc499021843)

[3.2 Description des tests effectués 28](#_Toc499021844)

[3.3 Erreurs restantes 33](#_Toc499021845)

[3.4 Liste des documents fournis 35](#_Toc499021846)

[4 Conclusions 36](#_Toc499021847)

[5 Annexes 38](#_Toc499021848)

[5.1 Résumé du rapport du TPI / version succincte de la documentation 38](#_Toc499021849)

[5.2 Sources – Bibliographie 38](#_Toc499021850)

[5.3 Journal de travail 39](#_Toc499021851)

[5.4 Manuel d'Installation 42](#_Toc499021852)

[5.5 Manuel d'Utilisation 41](#_Toc499021853)

[5.6 Archives du projet 52](#_Toc499021854)

*NOTE L’INTENTION DES UTILISATEURS DE CE CANEVAS:  
Toutes les parties en italiques sont là pour aider à comprendre ce qu’il faut mettre dans cette partie du document. Elles n’ont donc aucune raison d’être dans le document final.*

*De plus, en fonction du type de projet, il est tout à fait possible que certains chapitres ou paragraphes n’aient aucun sens. Dans ce cas il est recommandé de les retirer du document pour éviter de l’alourdir inutilement.*

# Analyse préliminaire

## Introduction

Ce projet est réalisé dans le cadre du TPI. Mon choix du sujet était de générer un horaire car la logique qui se trouve derrière m’intéresse. Le cahier des charges que j’ai reçu correspond à mon choix.

Le projet consiste à réaliser une application de planification de vols aériens :

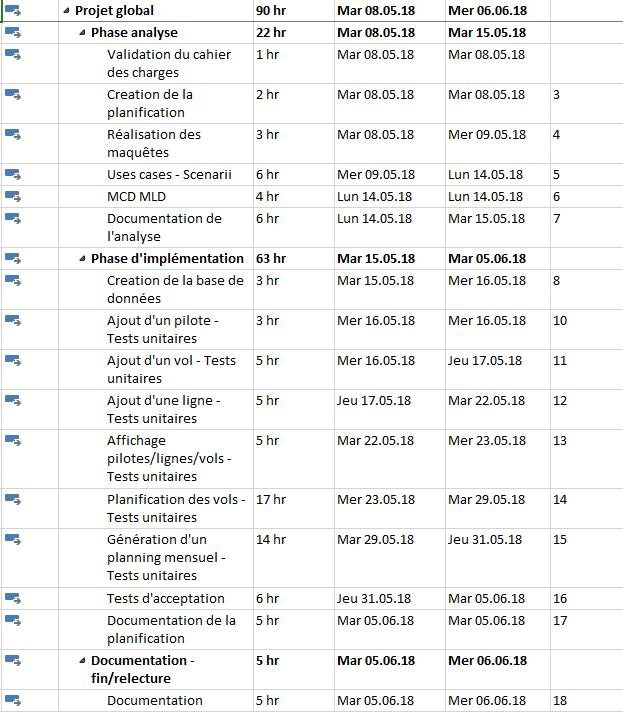
L’application permet à l’utilisateur de créer des pilotes, des lignes et des vols. Puis il peut affecter des pilotes à des vols, planifier des vacances aux pilotes et générer un horaire de travail pour les pilotes.

Aucun travail n’a déjà été effectué pour ce projet.

## Objectifs

* Gestion des pilotes, des lignes et des vols : l’utilisateur pourra ajouter des pilotes, des lignes et des vols par l’intermédiaire d’une interface graphique.
  + Un pilote est caractérisé notamment par son nom, son prénom, son aéroport d’affectation et le nombre d’heures de vol à son actif
  + Un vol est caractérisé, entre autres, par une date de départ et une date d’arrivée.
  + Une ligne est caractérisée notamment par un lieu de départ et un lieu d’arrivée
* Affichage des pilotes, des lignes et des vols : l’utilisateur pourra afficher la liste des pilotes, des lignes et des vols existants
* Planification des vols : une fois ces informations entrées, l’utilisateur pourra demander à gérer l’affectation d’un vol à un pilote, l’utilisateur pourra chercher un pilote disponible après avoir sélectionné un vol. Les contraintes suivantes devront être prises en compte :
  + Un pilote peut voler jusqu’à 10h consécutives, il bénéficie ensuite de minimum 12h de repos sur le lieu d’arrivée.
  + Chaque semaine, le pilote doit bénéficier de 2 jours de repos consécutifs sur le lieu de son aéroport d’affectation
  + Un pilote pourra être rapatrié vers son aéroport d’affectation soit en tant que passager soit en tant que pilote, selon les besoins.
  + Un pilote a droit à 5 semaines de vacances par an.
* Génération d’un planning mensuel par employé
  + L’utilisateur, sous la forme d’un fichier CSV, pourra générer pour chaque pilote et pour le mois en cours ou le mois suivant un planning de ses activités.
* Toutes les informations concernant les pilotes, les lignes, les vols, la planification des vols seront stockées dans une base de données.

## Planification initiale

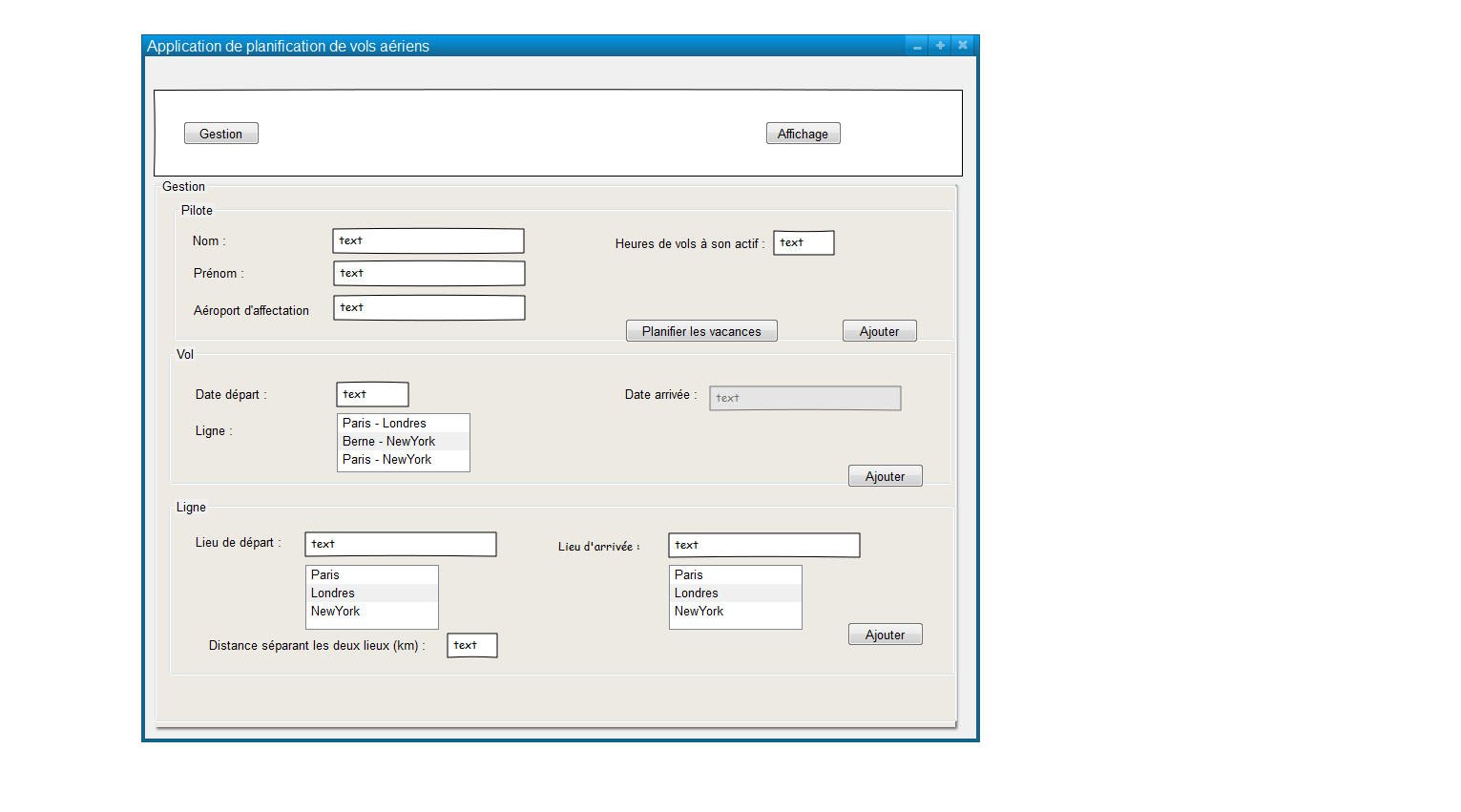


Planification initiale

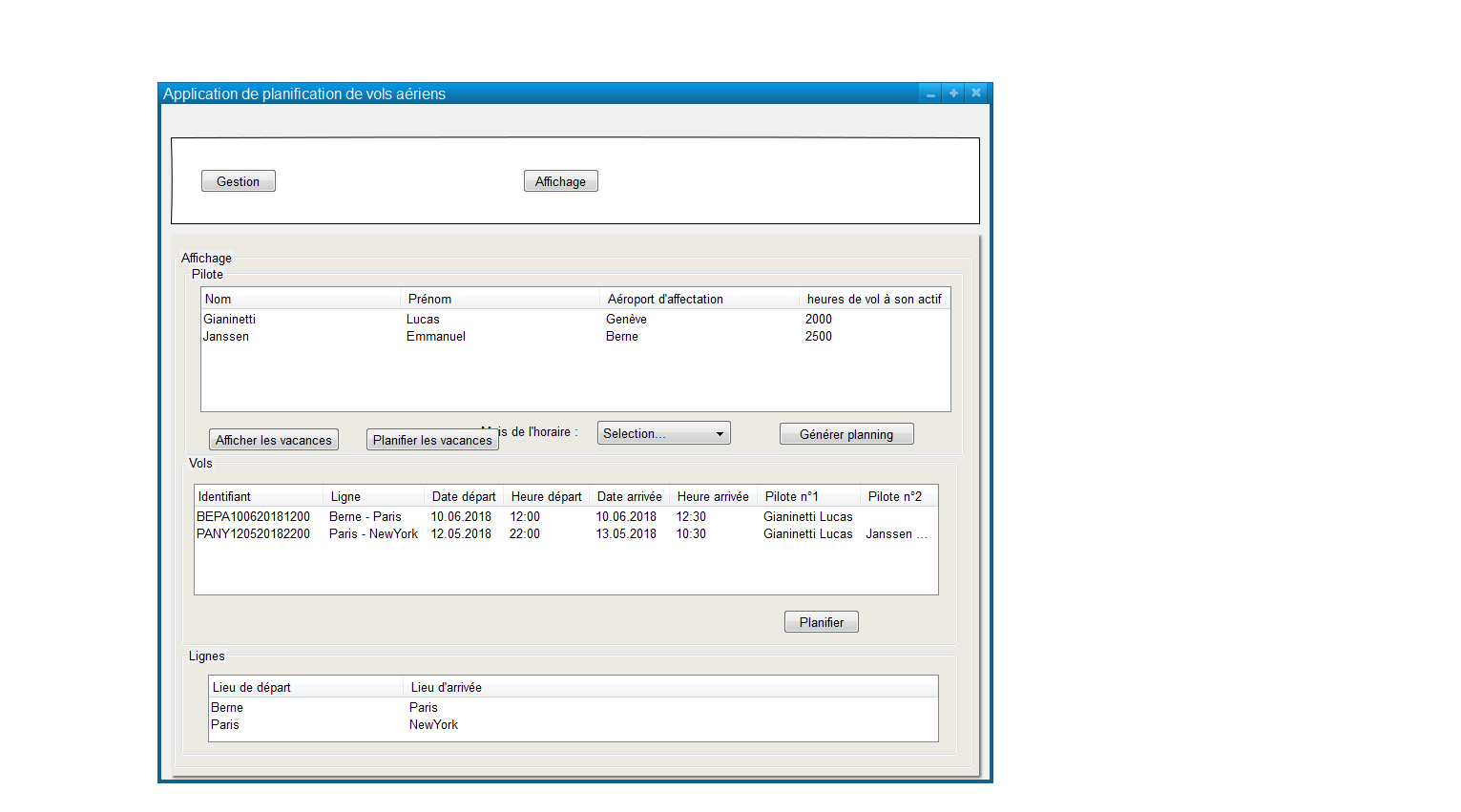
# Analyse / Conception

## Concept

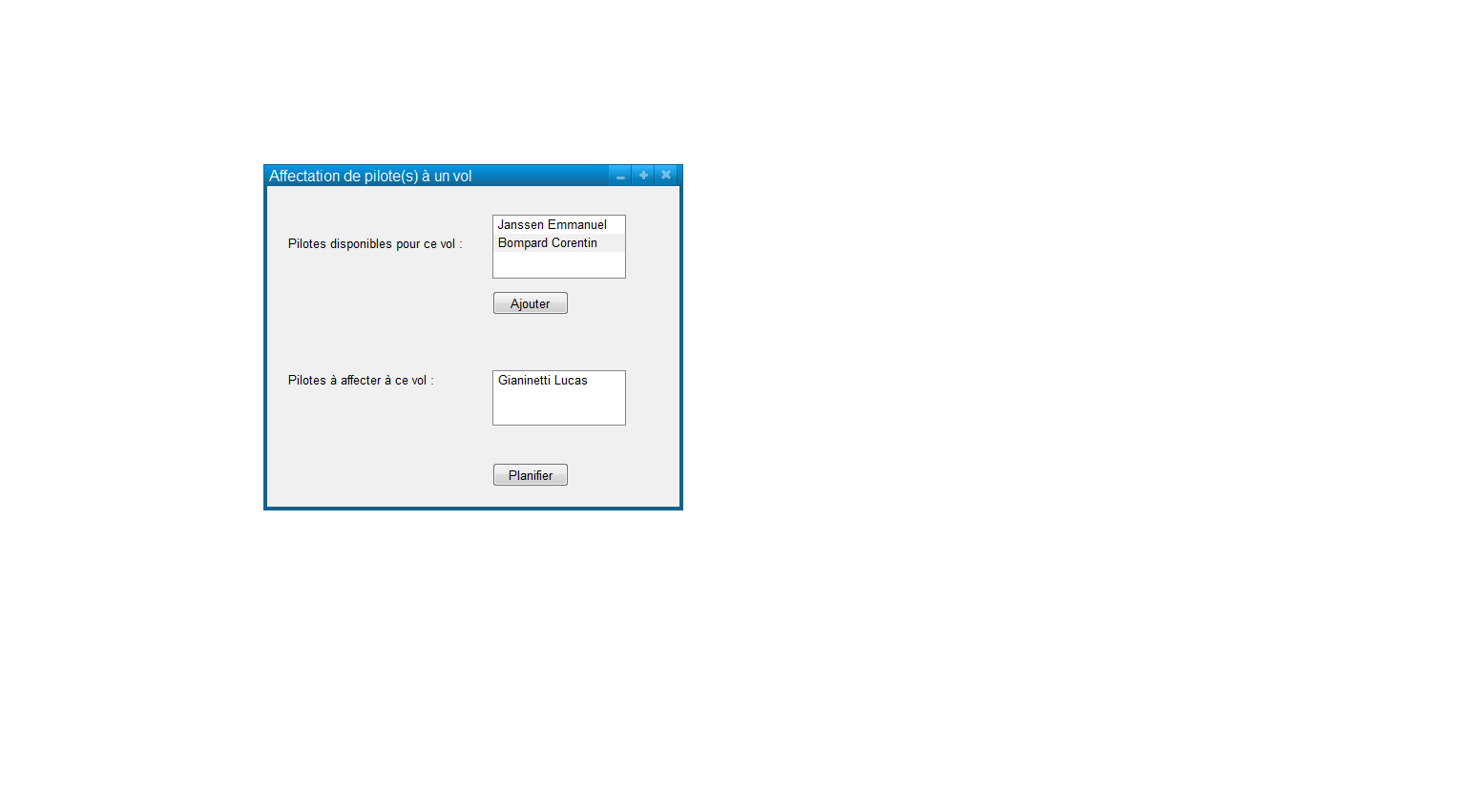
### Maquettes



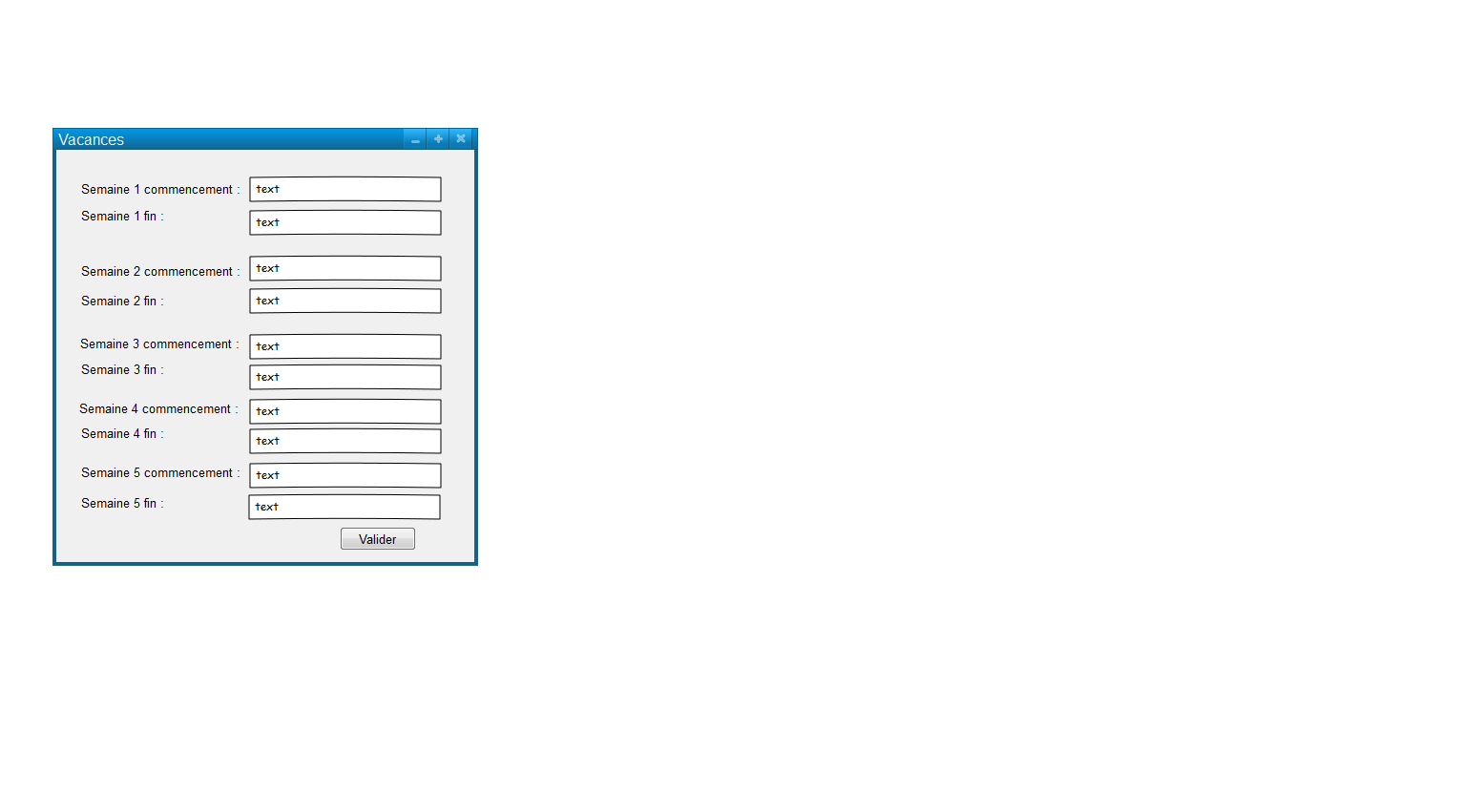
Maquette correspondant au formulaire « Gestion ». C’est dans ce formulaire que l’on pourra ajouter des pilotes, des vols et des lignes.



Maquette correspondant au formulaire « Affichage ». C’est dans ce formulaire que les informations concernant les pilotes, lignes et vols existants apparaitront. Ce panel permettra aussi d’affecter un pilote à un vol et de générer un planning pour un pilote spécifique

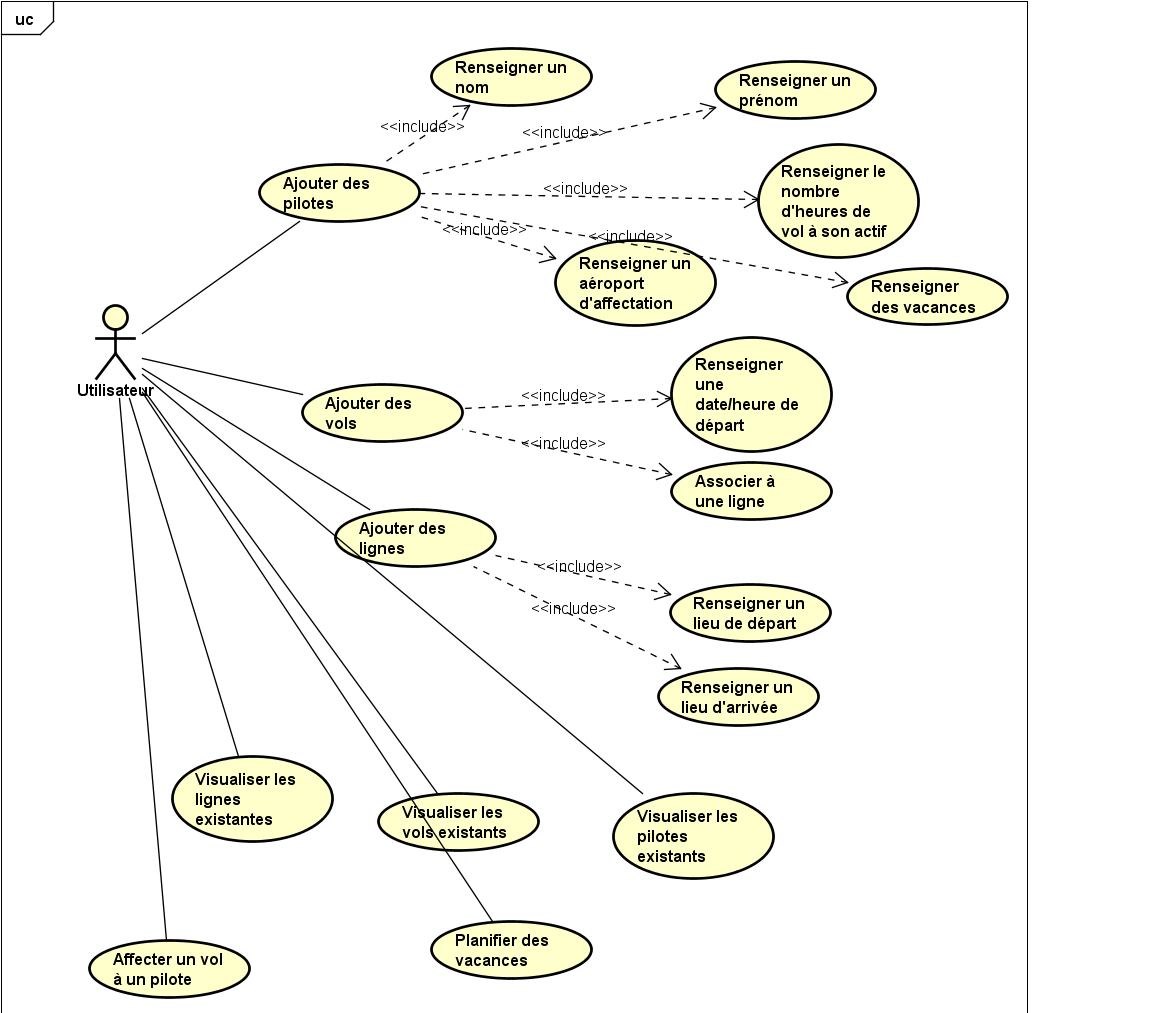


Maquette correspondant à la fenêtre qui apparait lors de l’affectation d’un pilote à un vol. Elle permet de choisir le(s) pilote(s) à affecter à un vol.



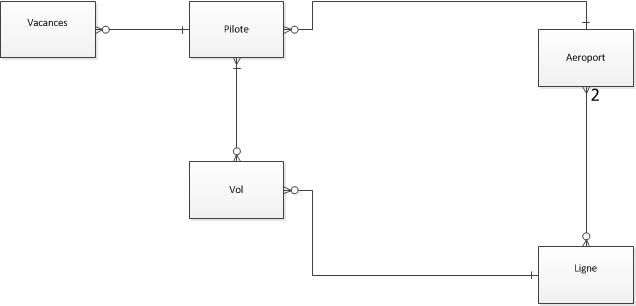
Maquette correspondant au formulaire « Vacances » qui apparait lorsqu’on planifie les vacances pendant la création d’un pilote, lorsqu’on affiche les vacances d’un pilote, lorsqu’on ajoute des vacances aux pilotes lors de la nouvelle année civique. Elle permet de rentrer les dates des 5 semaines de vacances.

### Uses case



Uses case

### Modèle conceptuel de données initial



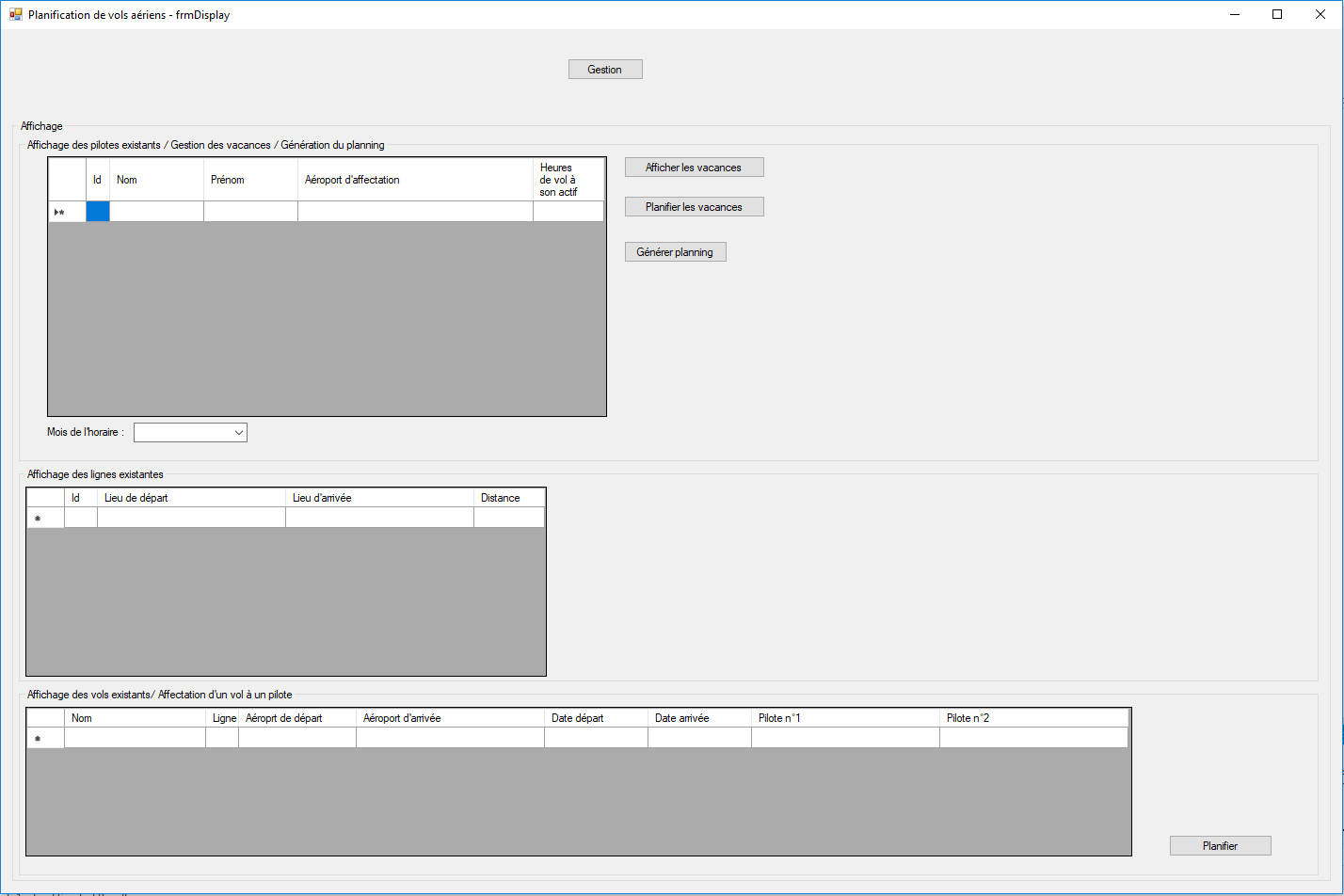
MCD

### Modèle logique de données initial

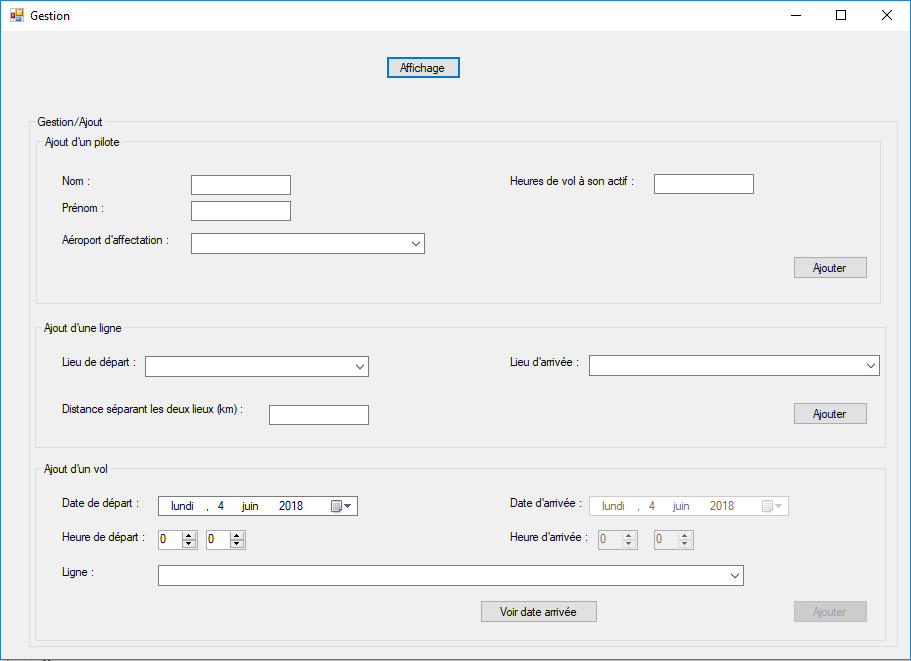
### 

MLD

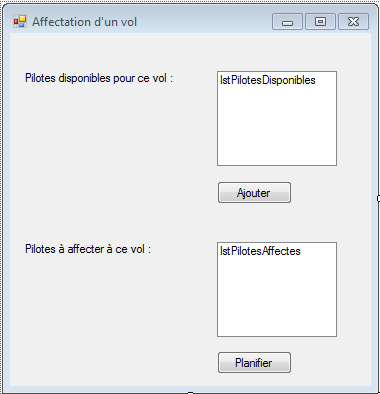
### Interfaces graphiques



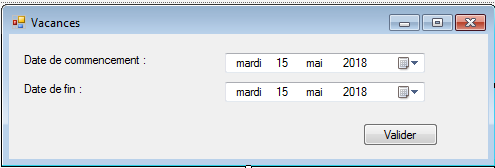
Interface du formulaire « Affichage »



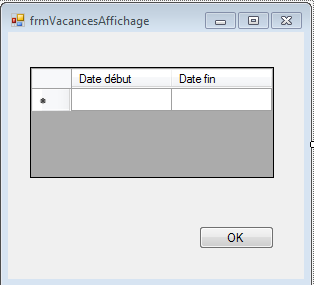
Interface du formulaire « Gestion »

**

Interface du formulaire « Affectation d’un vol »



Interface du formulaire « Vacances »



Interface du formulaire « VacancesAffichage »

## Stratégie de test

Des tests unitaires seront effectués après chaque implémentation d’une nouvelle fonctionnalité.

Dès la 1ère fonctionnalité implémentée, des tests d’intégration seront fait à l’ajout d’une nouvelle fonctionnalité afin de vérifier si cette dernière n’a pas affecté les autres fonctionnalités existantes.

Une fois le développement de l’application terminé des tests d’acceptation seront effectués afin de valider le bon fonctionnement du programme et/ou d’en ressortir les erreurs persistantes. Ses tests seront effectués par Mme. Andolfatto qui représente la cliente et par moi-même.

Un test automatique sera implémenté pour vérifier l’affectation d’un vol à un pilote.

Un test « d’utilisation » sera réalisé, un membre de ma famille va essayer d’utiliser l’application à l’aide du manuel d’utilisation afin de savoir si l’interface est intuitive.

## Risques techniques

La difficulté principale du projet concerne la planification des vols : les pilotes disponibles pour un vol devront remplir certaines conditions.

C’est pourquoi j’ai consacré une grande partie de mon implémentation au développement de cette fonctionnalité.

## Planification

## Dossier de conception

Le projet sera réalisé sur la machine mise à disposition dans ma salle de classe. C’est un ordinateur Dell Optiplex 9020 avec comme système d’exploitation Windows 7 Entreprise SP1.

Les logiciels suivants seront utilisés :

* Visual Studio Entreprise 2015
  + Programmation, tests automatique.
* Evolus Pencil version 2.0.5
  + Réalisation des maquettes
* Microsoft Visio Professionnel 2010 version 14.0.6023.1000
  + Réalisation du MCD
* MySQL Workbench
  + Réalisation du MLD, création du script de base de données
* Wamp Server 3.0.6
  + Serveur local pour utiliser la base de données
* Doxygen 1.8.13
  + Génération de la documentation du code en format HTML
  + Export du code source en format XML pour créer le diagramme de classe avec Astah Professional
* Astah Professional 7.2.0
  + Réalisation du Use Cases et du diagramme de classes
* GitHub Desktop 3.3.6.1
  + Versionning
* Microsoft Word 2016
  + Rédaction de la documentation
* Microsoft Excel 2016
  + Réalisation des scenarii et des tests unitaires, journal de travail et journal de bord
* Microsoft Project 2016
  + Planification initiale et finale

***!***

### Diagramme de classe :

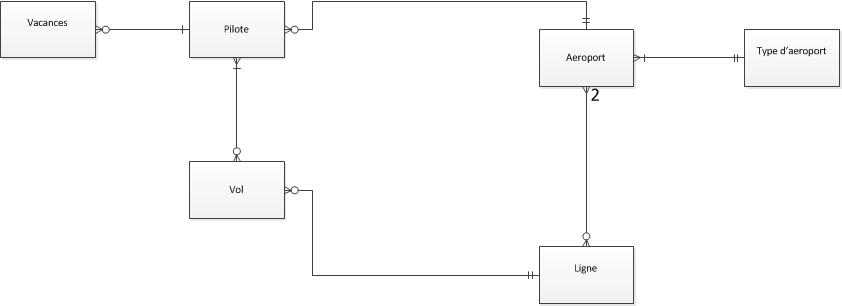
### 

Classes et leurs méthodes

### 

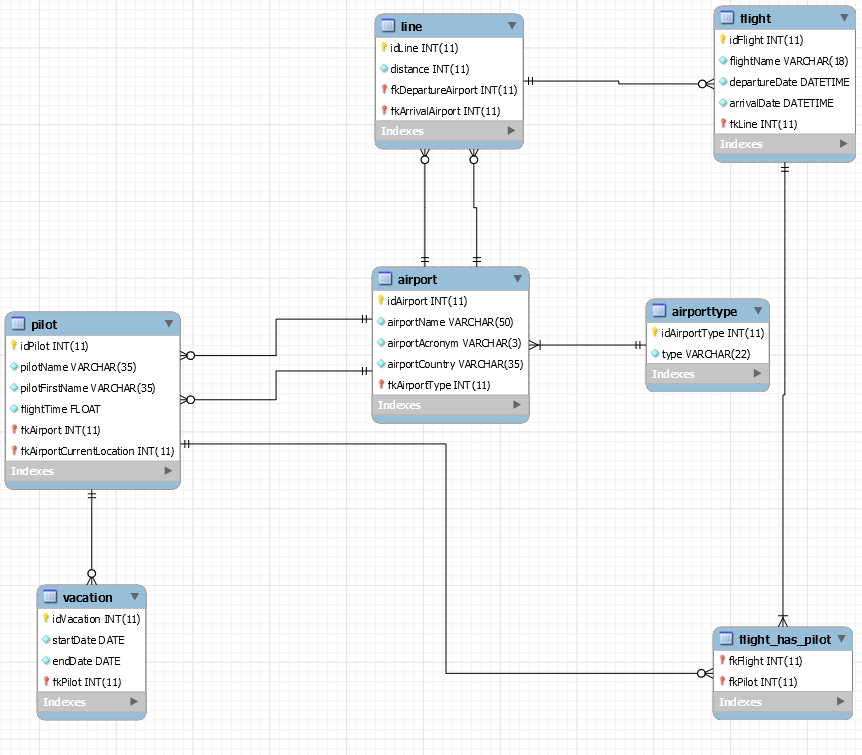
Diagramme de classes

### Modèle conceptuelle de données



MCD final

### Modèle logique de données

**

MLD final

Explication des relations :

La méthode Chen a été utilisée pour représenter les relations entre les tables.

 Ce symbole implique une relation « au moins un à plusieurs ».

Ce symbole implique une relation « un seul et unique ».

Ce symbole implique une relation « zéro à plusieurs ».

Exemple pour la relation qui relie line à flight :

Une ligne a zéro à plusieurs vols. Un vol a une seule et unique ligne

Exemple pour la relation qui relie airport à airportType

Un aéroport est défini par un seul et unique type d’aéroport. Un type d’aéroport définit au moins un à plusieurs aéroports.

Explication des tables :

* pilot : contient les informations concernant le pilote.
  + pilotName représente le nom du pilote. Le type est un varchar de taille 35 car c’est la taille conseillée par le *UK Government Data Standards Catalogue.*
  + pilotFirstName représente le prénom du pilote. Le type est un varchar de taille 35 car c’est la taille conseillée par le *UK Government Data Standards Catalogue.*
  + flightTime représente les heures de vol à son actif d’un pilote. Le type est un float car les heures peuvent avoir des décimales.
  + fkAirport représente l’aéroport d’affectation du pilote.
  + fkAirportCurrentLocation représente l’aéroport ou est actuellement le pilote
* flight : contient les informations d’un vol.
  + flightName sera composé de l’acronyme de l’aéroport de départ + acronyme de l’aéroport d’arrivée + la date de départ du vol (exemple : GVACDG140520181125, un vol partant de Genève le 14.05.2018 à 11h25 jusqu’à Paris). C’est un varchar de taille 18 car les noms des vols contiennent 18 caractères avec la convention que j’ai choisi.
  + departureDate représente la date de départ du vol.
  + arrivalDate représente la date d’arrivée du vol.
* vacation : contient les informations des vacances
  + startDate représente la date de début des vacances.
  + endDate représente la date de fin des vacances.
* airport : contient les informations des aéroports
  + airportName représente le nom de l’aéroport. C’est un varchar de taille 50 car il permet de contenir tous les noms d’aéroports que j’ai trouvé[[1]](#footnote-1).
  + airportAcronym représente un code de trois lettres permettant d’identifier un aéroport. C’est un varchar de taille 3 car il correspond au code AITA[[2]](#footnote-2).
  + airportCountry représente le pays dans lequel se situe l’aéroport. C’est un varchar de taille 35 car il permet de contenir tous les noms de pays en « version courte ».
* line : contient les informations sur les lignes
  + distance représente la distance de la ligne en kilomètres
  + fkArrivalAirport représente l’aéroport d’arrivée.
  + fkDepartureAirport représente l’aéroport de départ.
* airportType : contient les informations sur les différents types d’aéroports
  + type représente le type d’aéroport (ex : aéroport international, aérodrome…). C’est un varchar de taille 22 car le type le plus long est « Aeroport international », ce qui représente 22 caractères.

### Scenarii

|  |  |
| --- | --- |
| Identifiant | 1 |
| En tant que | Utilisateur |
| Je veux | Ajouter un pilote |
| Pour | Pouvoir l’affecter à un vol |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Je lance le programme | La formulaire gestion apparait |
| Je remplis le champ « nom » du groupBox « Pilote » | Les caractères que j’ai tapé sont inscrits dans le champ |
| Je remplis le champ  « prénom » du groupBox « Pilote » | Les caractères que j’ai tapé sont inscrits dans le champ |
| Je remplis le champ « Aéroport d’affectation » du groupBox « Pilote » | Les caractères que j’ai tapé sont inscrits dans le champ |
| Je remplis le champ « Heures de vols à son actif » du groupBox « Pilote » | Les caractères que j’ai tapé sont inscrits dans le champ |
| Je clique sur le bouton « Ajouter » du groupBox « Pilote » | Si tous les champs ont été renseignés correctement et les vacances planifiées, le pilote est ajouté dans la base de données. Si les champs n’ont pas été renseignés correctement, un message d’erreur averti l’utilisateur. |

|  |  |
| --- | --- |
| Identifiant | 2 |
| En tant que | Utilisateur |
| Je veux | Ajouter un vol |
| Pour | Pouvoir l’associer à un pilote |

|  |  |
| --- | --- |
| Action | Réaction |
| Je lance le programme | La formulaire gestion apparait |
| Je remplis le champ « Date départ » du groupBox « Vol » | Les caractères que j’ai tapé sont inscrits dans le champ |
| Je sélectionne une ligne dans le groupBox « Vol » | La ligne est sélectionnée et le champ « Date arrivée » du groupBox « vol » se remplit en fonction de la date de départ et de la ligne sélectionnée. |
| Je clique sur le bouton « Ajouter » du groupBox « Vol » | Si tout est renseigné correctement le vol est ajouté dans la base de données. Si tout n’est pas renseigné correctement, un message d’erreur averti l’utilisateur. |

|  |  |
| --- | --- |
| Identifiant | 3 |
| En tant que | Utilisateur |
| Je veux | Ajouter une ligne |
| Pour | Pouvoir y associer des vols |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Je lance le programme | La formulaire gestion apparait |
| Je sélectionne une case de la liste « Lieu de départ » du groupBox « Ligne » | La case sélectionnée est sélectionnée. |
| Je sélectionne une case de la liste « Lieu d’arrivée» du groupBox « Ligne » | La case sélectionnée est sélectionnée |
| Je clique sur le bouton « Ajouter » du groupBox « Ligne » | Si une case est sélectionnée dans les deux listes du groupBox « Ligne » la ligne est ajoutée dans la base de données. Si ce n’est pas le cas un message d’erreur averti l’utilisateur |

|  |  |
| --- | --- |
| Identifiant | 4 |
| En tant que | Utilisateur |
| Je veux | Afficher les vols, lignes, pilotes |
| Pour | Pouvoir visualiser les vols, lignes, pilotes existants |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Je lance le programme | La formulaire  « Gestion » apparait |
| Je clique sur le bouton « Affichage » | * Le panel « Affichage » remplace le panel « Gestion » * Les informations concernant les pilotes existants sont affichées dans un tableau dans le groupBox « Pilote »   + Les vacances ne sont pas affichées directement. * Les informations concernant les vols existants sont affichées dans un tableau dans le groupBox « Vols » * Les informations concernant les lignes existantes sont affichées dans un tableau dans le groupBox « Lignes » |
| Je clique sur le bouton « Afficher les vacances » | Si un pilote est sélectionné : Le formulaire AfficherVacances s’ouvre et on peut voir les dates de début et de fin des vacances du pilote.  Si aucun pilote n’est sélectionné : un message d’erreur averti l’utilisateur |

|  |  |
| --- | --- |
| Identifiant | 5 |
| En tant que | Utilisateur |
| Je veux | Affecter un vol à un pilote |
| Pour | Dans le futur généré un horaire pour ce pilote |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Je lance le programme | La formulaire « Gestion » apparait |
| Je clique sur le bouton « Affichage » | La formulaire « Affichage » apparait, le formulaire « Gestion » se ferme |
| Je sélectionne une ligne du tableau du groupBox « Vols » | La ligne est correctement sélectionnée |
| Je clique sur planifier | Si une ligne a été précédemment sélectionnée, la fenêtre « Affectation de pilote(s) à un vol » s’ouvre. Si ce n’est pas le cas un message d’erreur averti l’utilisateur. |
| Dans la fenêtre « Affectation de pilote(s) à un vol » je sélectionne une case de la liste « Pilotes disponibles pour ce vol » | La case est correctement sélectionnée |
| Dans la fenêtre « Affectation de pilote(s) à un vol » je clique sur le bouton « Ajouter » | * Si une case de la liste est sélectionnée, le pilote est ajouté dans la liste « Pilotes à affecter à ce vol ». * Si aucune case de la liste n’est sélectionnée, un message d’erreur averti l’utilisateur. * Si le vol fait moins de 10h le bouton « Ajouter » devient désactivé. * Si le vol fait plus de 10h le bouton « Ajouter » est toujours afin d’ajouter un 2ème pilote à ce vol. |
| Dans la fenêtre « Affectation de pilote(s) à un vol » je clique sur le bouton « Planifier » | * Si aucun pilote n’est présent dans la liste « Pilotes à affecter à ce vol » un message d’erreur averti l’utilisateur. * Si un pilote est présent dans la liste « Pilotes à affecter à ce vol » et que le vol fait moins de 10h l’affectation du pilote à ce vol est enregistrée dans la base de données et la fenêtre se ferme * Si un pilote est présent dans la liste « Pilotes à affecter à ce vol » et que le vol fait plus de 10h, un message d’erreur averti l’utilisateur qu’un 2ème pilote est nécessaire pour ce vol * Si deux pilotes sont présents dans la liste « Pilotes à affecter à ce vol » l’affectation des pilotes à ce vol est enregistrée dans la base de données et la fenêtre se ferme. |

|  |  |
| --- | --- |
| Identifiant | 6 |
| En tant que | Utilisateur |
| Je veux | Générer un planning pour un pilote |
| Pour | Voir ses prochains vols durant ce mois ou le mois prochain |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Je lance le programme | La formulaire « Gestion » apparait |
| Je clique sur le bouton « Affichage » | La formulaire « Affichage » apparait, le formulaire « Gestion » se ferme |
| Je sélectionne une ligne du tableau du groupBox « Pilotes » | La ligne sélectionnée est correctement sélectionnée |
| Je sélectionne le « Mois de l’horaire » (mois actuel ou mois prochain) | Le champ que j’ai sélectionné est correctement sélectionné. |
| Je clique sur le bouton « Générer planning » du groupBox « Pilotes » | * Si aucune ligne du tableau n’est pas sélectionnée ou aucun « Mois de l’horaire » n’est sélectionné, un message d’erreur averti l’utilisateur. * Si une ligne du tableau est sélectionnée et un « Mois de l’horaire » est sélectionné, un planning en format CSV correspondant au pilote sélectionné est enregistré sur l’ordinateur de l’utilisateur. |

|  |  |
| --- | --- |
| Identifiant | 7 |
| En tant que | Utilisateur |
| Je veux | Planifier des vacances |
| Pour | Ajouter des vacances à un pilote |

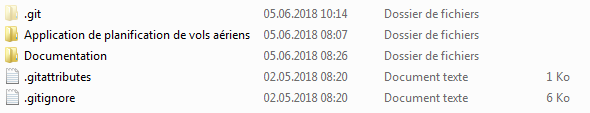
|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Je lance le programme | La formulaire « Gestion » apparait |
| Je clique sur le bouton « Affichage » | La formulaire « Affichage » apparait, le formulaire « Gestion » se ferme |
| Je sélectionne une ligne du tableau du groupBox « Pilotes » | La ligne sélectionnée est correctement sélectionnée |
| Je clique sur le bouton « Planifier les vacances » | Le formulaire vacances s’ouvre |
| Je renseigne une date de début et une date de fin | Les DatesTimesPickers affiche les dates renseignées. |
| Je clique sur le bouton « Valider » | Si tous les DatesTimesPickers sont rempli correctement et que le pilote a encore des vacances disponibles, les vacances du pilote sont ajoutées dans la base de données et le formulaire se ferme.  Si le pilote ne possède plus de vacances disponible un message d’erreur averti l’utilisateur. |

# Réalisation

## Dossier de réalisation

### Arborescence du projet

Le dossier principal se nomme « TPI », il contient les dossiers et fichiers suivants :



Contenu du dossier principal « TPI »

* « .git », « .gitattributes », « .gitignore »
  + Sont des fichiers/dossiers utilisés par le gestionnaire de version GitHub
* « Application de planification de vols aériens »
  + Contient la solution Visual Studio de l’application

Le dossier documentation contient toute la documentation relative au projet, il se présente ainsi :



Contenu du dossier « Documentation »

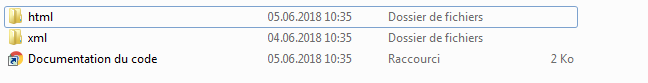
Le dossier « Base de données » se présente ainsi :



Contenu du dossier « Base de données »

* « VolsAeriens.sql »
  + Script de création de la base de données

Le dossier « Code » se présente ainsi :



Contenu du dossier « Code »

* « xml » a été généré à partir de Doxygen afin de pouvoir générer automatiquement le diagramme de classe avec Astah Professional.
* « html » contient la documentation du code générée à partir de Doxygen.
* « Documentation du code »
  + Est un raccourci pour accéder à la page web de la documentation du code.

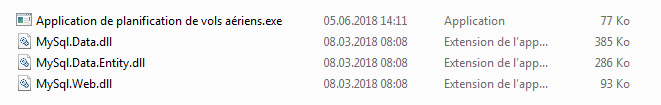
Le dossier « Projet » se présente ainsi :



Contenu du dossier « Projet »

* « Application de planification de vols aériens.docs »
  + Documentation du projet

Le dossier « Exécutable » se présente ainsi :



* « Application de planification de vols aériens.exe »
  + Exécutable de l’application
* « MySql.Data.dll », « MySql.Data.Entity.dll », « MySql.Web.dll »
  + Extensions d’application pour la librairie « MySql.dll »

### Convention de nommage des éléments Windows Forms

|  |  |
| --- | --- |
| Form | frm |
| GroupBox | grp |
| Label | lbl |
| TextBox | txt |
| DataGridView | dgv |
| Column (DataGridView) | col |
| Combobox | cbo |
| Button | cmd |
| NumericUpDown | nud |
| DateTimePicker | dtp |
| ListBox | lst |

Les différents éléments ont été nommés en anglais. Par exemple pour le formulaire « Gestion » son nom est donc « frmManagement ».

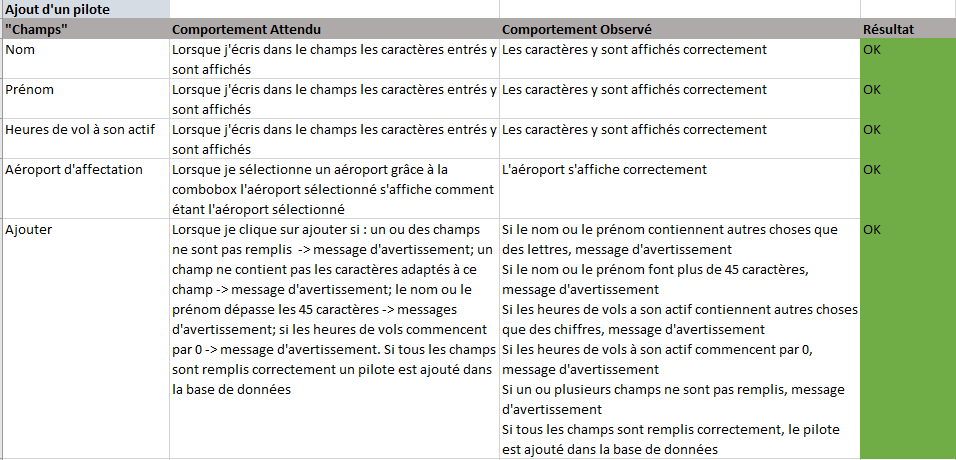
### Librairie utilisée

Afin de pouvoir communiquer avec la base de données, la libraire « MySql.dll » a été utilisée.

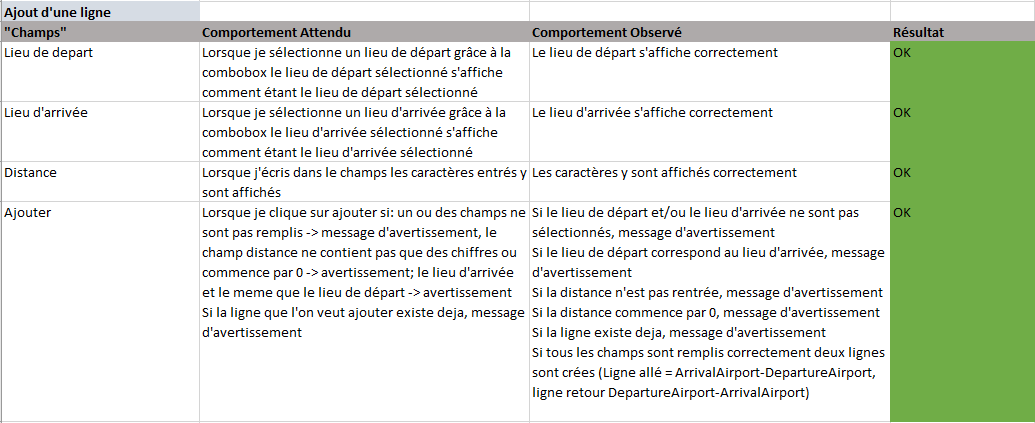
## Description des tests effectués

### Tests unitaires

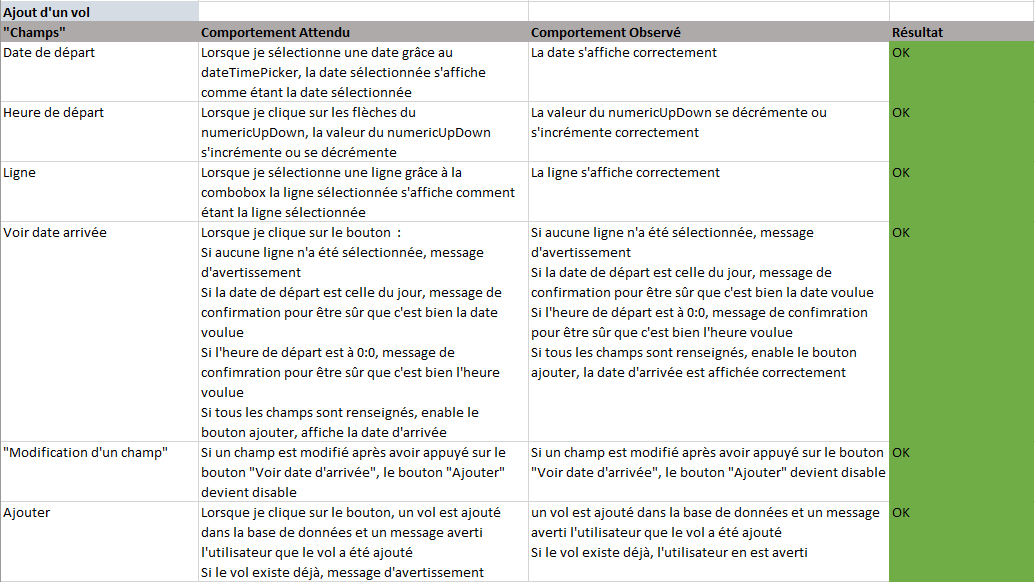
Les tests unitaires sont effectués sur la même machine qui a servi au développement de l’application. Ils ont tous été effectués par moi-même.



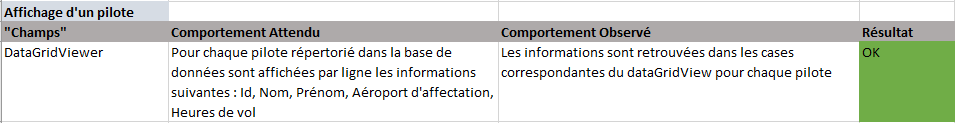
Tests unitaires pour l’ajout d’un pilote



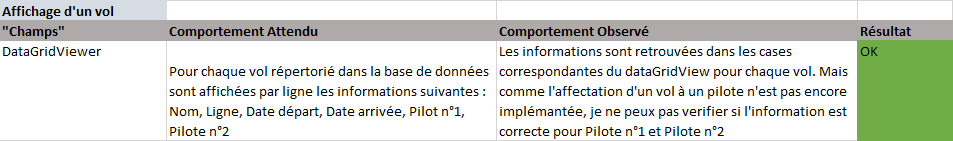
Tests unitaires pour l’ajout d’une ligne



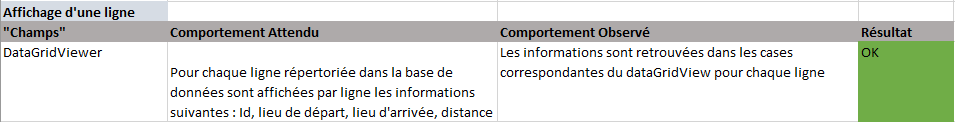
Tests unitaires pour l’ajout d’un vol



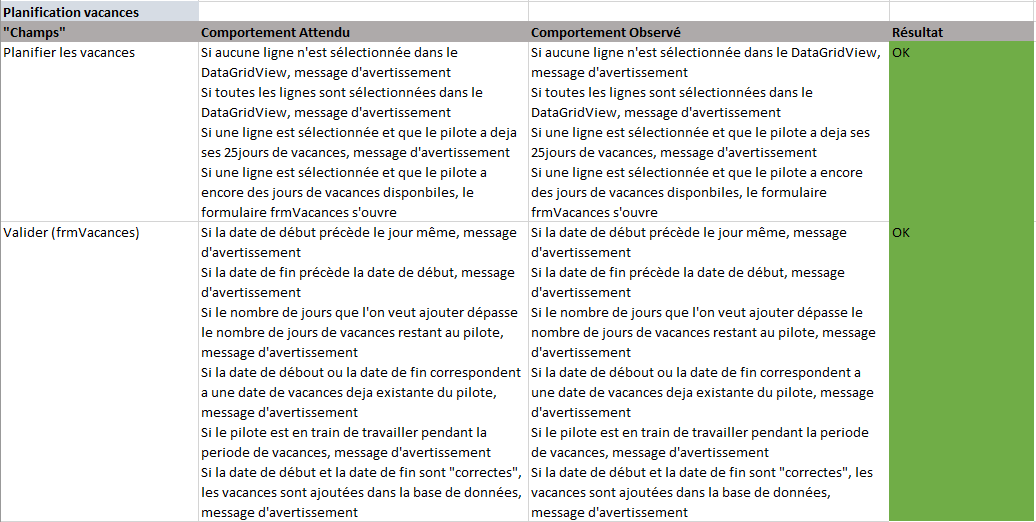
Test unitaire pour l’affichage d’un pilote



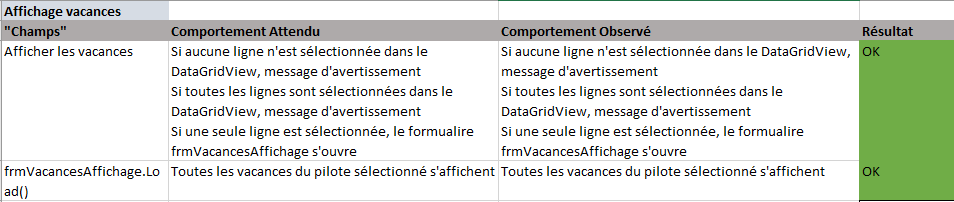
Test unitaire pour l’affichage d’un vol



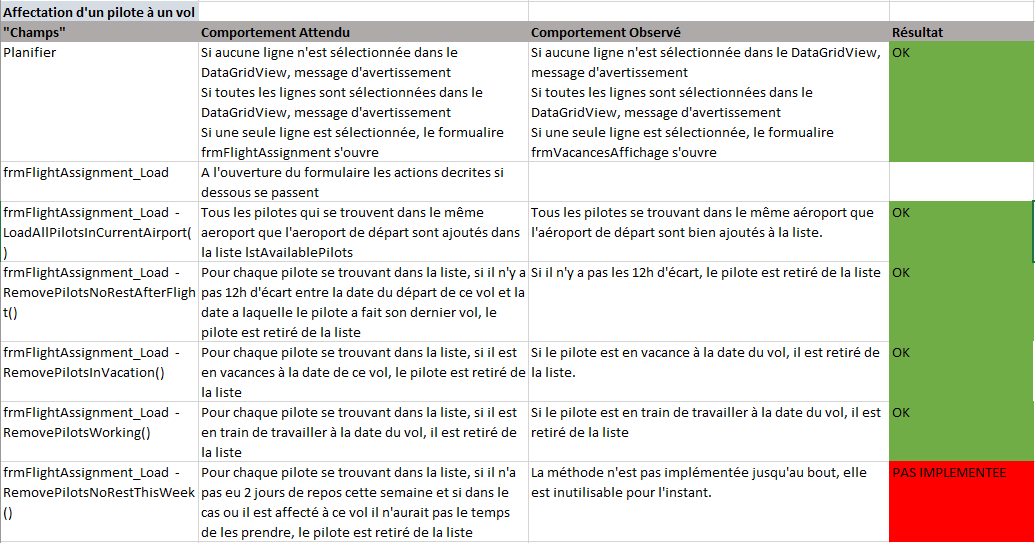
Test unitaire pour l’affichage d’une ligne



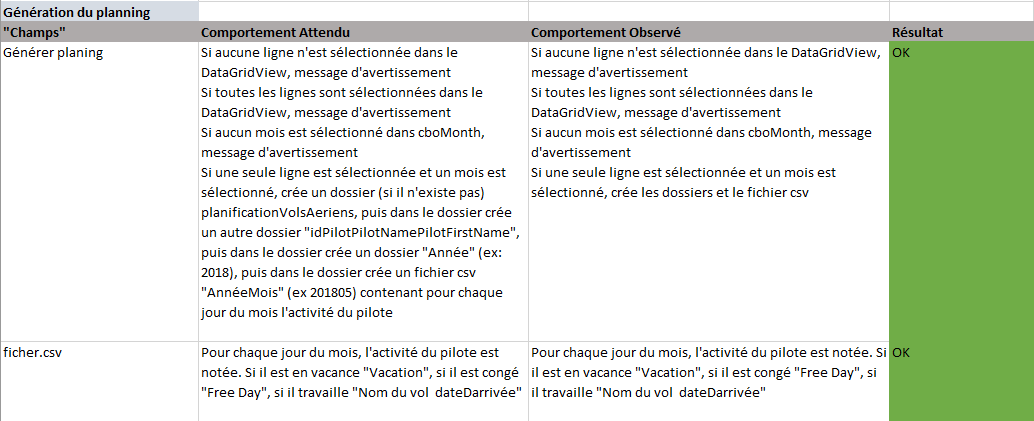
Tests unitaires pour la planification des vacances



Tests unitaires pour l’affichage des vacances



Tests unitaires pour l’affectation d’un pilote à un vol

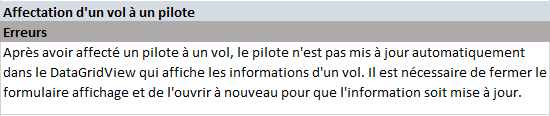


Tests unitaires pour la génération du planning

### Tests d’acceptation

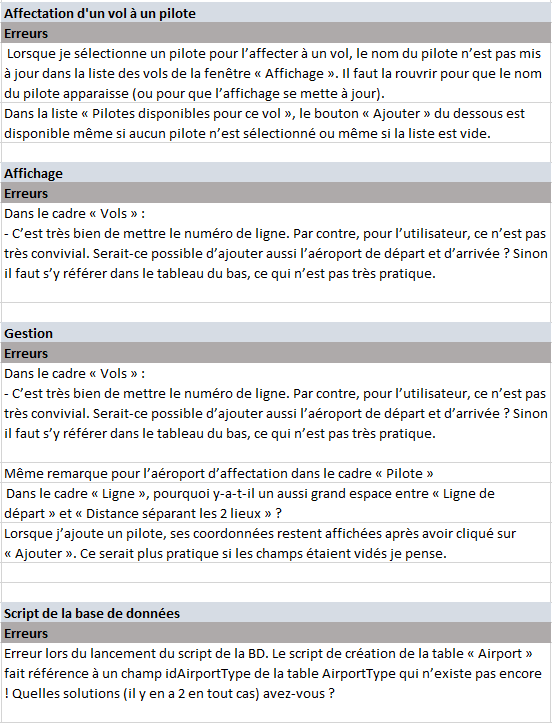
Mme. Andolfatto et moi-même avons réalisé des tests d’acceptation.

Résultat de mes tests d’acceptation réalisés sur la même machine qui m’a servie à développer l’application :



Tests d’acceptation par Lucas Gianinetti

Résultats des tests d’acceptation réalisés par Mme. Andolfatto sur une machine Windows 7 avec Visual Studio 2015 et MySql 5.7.14:



Tests d’acceptation réalisés par Mme. Andolfatto

### Tests d’utilisation

Les tests d’utilisation ont été effectués par M. Gianinetti sur une machine Windows 10 avec Visual Studio 2015 et MySql 6.3.

D’abord sans l’aide du manuel d’utilisation, puis avec le manuel d’utilisation, M. Gianinetti a essayé d’utiliser l’application.

Il a remarqué trois points qui faciliterai la facilité d’utilisation :

* Mettre des noms plus parlant pour les GroupBox, afin de savoir exactement pour quoi ils sont utilisés.
* Changer l’ordre des GroupBox dans le formulaire gestion, car l’ajout du vol se situait avant l’ajout d’une ligne et que l’ajout d’une ligne est nécessaire pour pouvoir ajouter un vol.
* Il n’y a pas d’information qui permet de savoir dans quel aéroport se trouve actuellement le pilote. Il n’est donc pas facile d’affecter des pilotes à des vols lorsqu’on ne sait pas où sont ces pilotes.

### Tests automatiques

Le test automatique pour vérifier l’affectation d’un vol à un pilote est implémenté dans la solution grâce au Unit Test de Visual Studio.

## Erreurs restantes

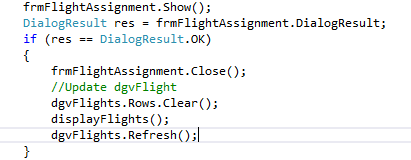
* La contrainte « Chaque semaine, le pilote doit bénéficier de 2 jours de repos consécutifs sur le lieu de son aéroport d’affectation » de l’objectif « Planification des vols » n’a pas été implémentée.
  + Cela a une conséquence majeure sur l’utilisation de l’application, l’application ne tient pas en compte cette contrainte lorsqu’elle propose la liste des pilotes disponibles pour un vol.
  + J’ai commencé à développer cette fonctionnalité, mais pas manque d’efficacité et du coup de temps n’ai pas réussi à la finir. J’arrive à savoir si un pilote a eu 2 jours de repos dans la semaine. Quel jour de la semaine nous sommes.

Mais il reste à savoir si les 2 jours se situent avant ou après la date du vol sélectionné :

* + - S’ils se situent avant, le pilote serait disponible pour le vol.
    - S’ils se situent après :
      * Si le pilote est dans son aéroport d’affectation, il faudrait savoir si le pilote à le temps de faire le vol allé + les 12h de repos + le vol retour + les 2 jours de repos. Si c’est le cas il serait disponible pour le vol.
      * Si le pilote n’est pas dans son aéroport d’affectation et que le vol retourne dans son aéroport d’affectation, il faudrait savoir si le pilote à le temps de faire le vol retour + les 2 jours de repos. Si c’est le cas il serait disponible pour le vol
      * Si le pilote n’est pas dans son aéroport d’affection et que le vol ne retourne pas dans son aéroport d’affectation, il faudrait savoir si le pilote a le temps de faire le vol allé + les 12h + un vol jusqu’à son aéroport d’affectation + les 2 jours de repos. Si c’est le cas il serait disponible pour le vol.
* La contrainte « Un pilote pourra être rapatrié vers son aéroport d’affectation soit en tant que passager soit en tant que pilote, selon les besoins » de l’objectif « Planification des vols » n’a pas été implémentée car elle découle de la contrainte ci-dessus.
  + Cela a une conséquence majeure sur l’utilisation de l’application, les pilotes restent dans l’aéroport d’arrivée après un vol.
  + Je n’ai pas du tout développé cette fonctionnalité car elle découle d’une fonctionnalité que je n’ai pas réussi à développer.

La solution que j’envisage est la suivante : Après un vol lorsque le pilote arrive à destination.

* Si le pilote a déjà eu ses jours de congés, il peut rester dans l’aéroport de destination en attendant qu’on lui affecte un vol retour.
* Si le pilote n’a pas déjà eu ses jours de congés :
  + S’il a juste le temps de se reposer les 12h puis de retourner dans son aéroport d’affectation et de prendre ses 2 jours de congés, le pilote serait rapatrié automatiquement en tant que passager dans un vol fictif
  + S’il a plus de temps que se reposer 12h puis de retourner dans son aéroport d’affectation et de prendre ses 2 jours de congés, le pilote resterai dans l’aéroport en attendant d’être affecté à un vol pour retourner à son aéroport d’affectation.
  + Si le pilote est en attente d’être affecté à un vol pour retourner à son aéroport d’affectation et qu’il n’a pas été affecté à un vol. Lorsque le temps restant est juste suffisant pour le vol jusqu’à son aéroport d’affectation + les 2 jours de repos, le pilote est rapatrié automatiquement en tant que passager dans un vol fictif.
* Lorsqu’un pilote est affecté à un vol, le nom du pilote n’est pas mis à jour automatiquement dans la liste des vols de la fenêtre « Affichage » :
  + Cela a une conséquence mineure sur l’utilisation de l’application, les utilisateurs doivent rouvrir la fenêtre « Affichage » pour le nom du pilote apparaisse.
  + Malgré les solutions trouvées sur internet je n’arrive pas à mettre à jour le DataGridView dgvFlight à la fermeture du formulaire qui permet d’affecter un pilote à un vol :



## Liste des documents fournis

Dans le dossier « Application de planification de vols aériens »

* Le code source du projet.

Dans le dossier « Documentation »

* Dans le dossier « Projet »
  + La documentation du projet « Application de planification de vols aériens.docs ».
* Dans le dossier « Code »
  + Un raccourci internet « Documentation du code » pour la documentation du code.
* Dans le dossier « Base de données »
  + Le script de création de la base de données « VolsAeriens.sql ».
* Dans le dossier « Exécutable »
  + L’exécutable du programme « Application de planification de vols aériens.exe »

# Conclusions

Les objectifs atteints sont les suivants :

* Gestion des pilotes, des lignes et des vols : l’utilisateur pourra ajouter des pilotes, des lignes et des vols par l’intermédiaire d’une interface graphique.
* Affichage des pilotes, des lignes et des vols : l’utilisateur pourra afficher la liste des pilotes, des lignes et des vols existants.
* Planification des vols : une fois ces informations entrées, l’utilisateur pourra demander à gérer l’affectation d’un vol à un pilote, l’utilisateur pourra chercher un pilote disponible après avoir sélectionné un vol.

Les contraintes suivantes devront être prises en compte :

* Un pilote peut voler jusqu’à 10h consécutives, il bénéficie ensuite de minimum 12h de repos sur le lieu d’arrivée.
* Un pilote a droit à 5 semaines de vacances.

Les objectifs non atteints sont les suivants :

* Planification des vols : une fois ces informations entrées, l’utilisateur pourra demander à gérer l’affectation d’un vol à un pilote, l’utilisateur pourra chercher un pilote disponible après avoir sélectionné un vol.

Les contraintes suivantes devront être prises en compte :

* Chaque semaine, le pilote doit bénéficier de 2 jours de repos consécutifs sur le lieu de son aéroport d’affectation.
* Un pilote pourra être rapatrié vers son aéroport d’affectation soit en tant que passager soit en tant que pilote, selon les besoins.

Le point négatif du projet est que je n’ai pas réussi à remplir le cahier des charges dans son entier pour le temps qui m’est imparti, et qu’il reste une erreur dans ce qui a été implémenté.

Le point positif du projet est que dans ce qui a été implémenté, il n’y a pas d’erreur qui pose problème au bon fonctionnement de l’application.

La plus grosse difficulté du projet concerne les deux points de l’objectif non atteints. J’ai mis trop de temps à trouver une solution à mon problème et il me restait plus assez de temps pour l’implémenter et la tester.

Pour l’avenir de ce projet, voici diverses améliorations/évolutions possibles :

* Implémenter les 2 points de l’objectif non atteints.
* Calculer automatiquement la distance qui sépare deux aéroports, afin que l’utilisateur n’ait pas à rentrer lui-même cette distance. Les distances seraient donc exactes et plus réalistes.
* Dans l’affichage des informations des pilotes, afficher la localisation actuelle du pilote.
* Pouvoir sélectionner des types d’avions pour les vols. Chaque avion aurait des spécificités différentes.
* Le temps de vol ne soit pas calculé à partir de la distance de la ligne est d’une valeur constante (ici 900km/h). Mais par rapport au type d’avion, à la vitesse d’atterrissage et de freinage.
* Ajout de copilotes et gestion de leurs emplois du temps comme pour les pilotes.
* Les pilotes puissent faire parti d’une compagnie aérienne ce qui auraient d’éventuelles conséquences. (Par exemple pour une certaine compagnie le pilote devrait être présent 1h avant le vol et pour une autre 2h, ou que le temps de repos entre deux vols varie selon la compagnie, …).
* Gestion salariale.
* …

# Annexes

## Résumé du rapport du TPI / version succincte de la documentation

## Sources – Bibliographie

Sites internet :

* Création du diagramme de classes :
  + <http://astah.net/features/csharp-plugin>
* Aide pour la programmation :
  + https://stackoverflow.com/questions/7657137/datagridview-full-row-selection-but-get-single-cell-value
  + <https://stackoverflow.com/questions/1607336/calculate-difference-between-two-dates-number-of-days>
  + https://stackoverflow.com/questions/5672862/check-if-datetime-instance-falls-in-between-other-two-datetime-objects
  + <https://stackoverflow.com/questions/38039/how-can-i-get-the-datetime-for-the-start-of-the-week>
  + <https://stackoverflow.com/questions/18757097/writing-data-into-csv-file-in-c-sharp>
  + <https://stackoverflow.com/questions/25032523/c-sharp-asp-net-selecting-multiple-column-from-same-mysql-command-and-use-them-i>
  + <https://stackoverflow.com/questions/3744882/datagridview-clear>
* Convention de nommage :
  + <https://gist.github.com/andyyou/3052671>
* Convention pour la taille des données :
  + <http://webarchive.nationalarchives.gov.uk/+/http://www.cabinetoffice.gov.uk/media/254290/GDS%20Catalogue%20Vol%202.pdf>

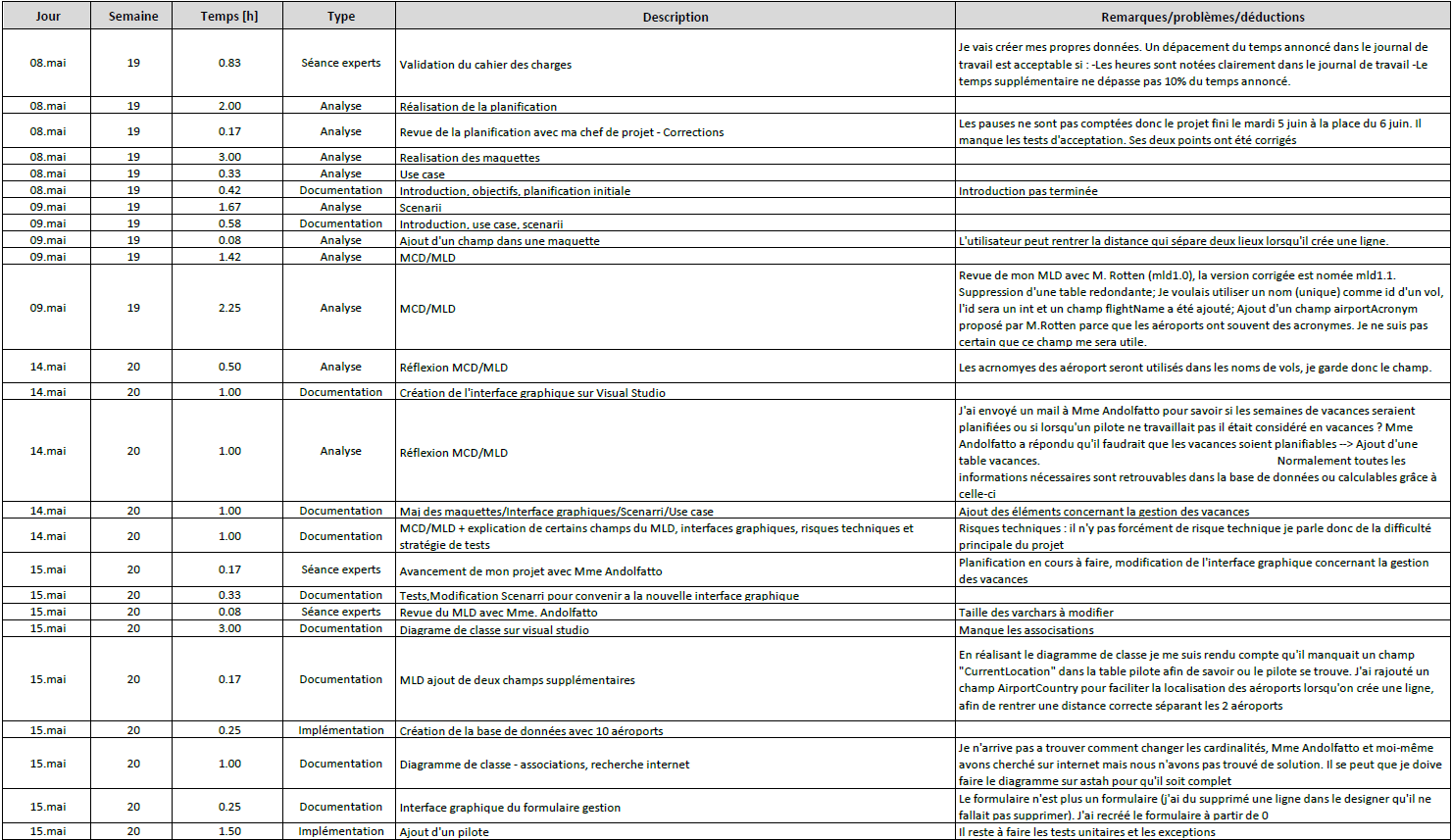
Personnes :

Mme. Andolfatto pour les tests d’acceptation et différents retours et aide durant le projet.

M. Gianinetti pour les tests d’utilisation.

Mme. Gianinetti pour l’orthographe.

## Journal de travail



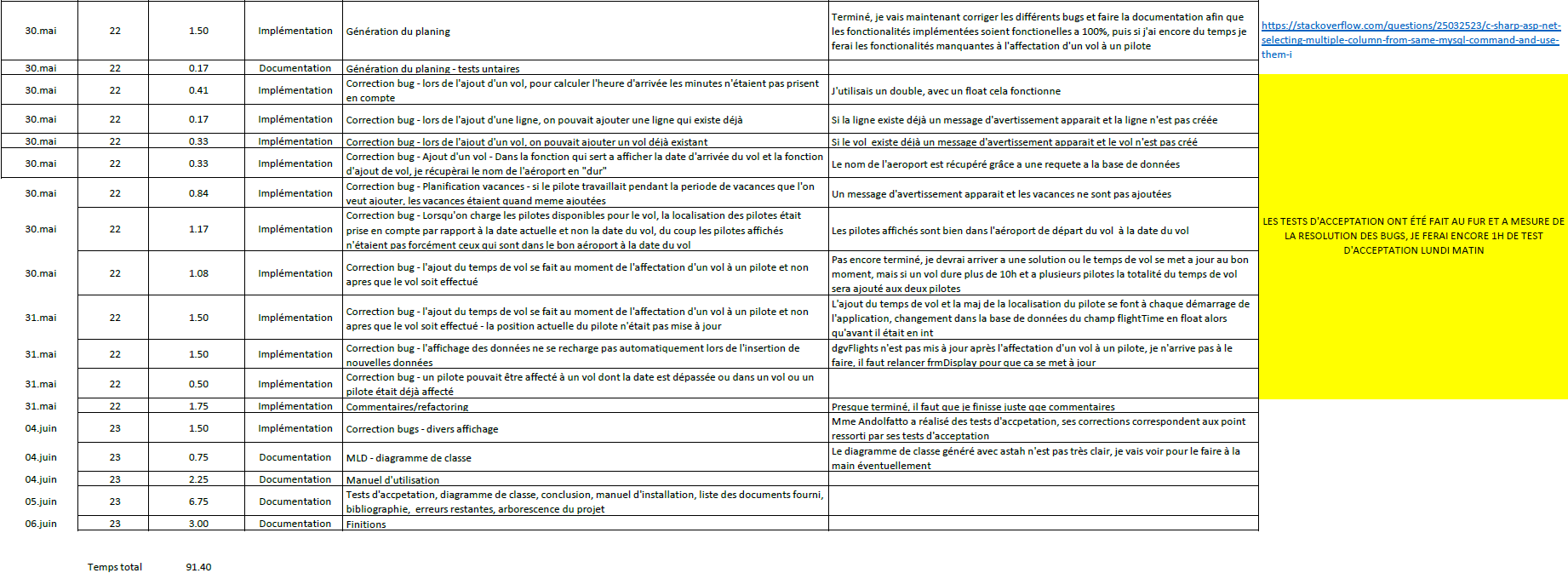
Journal de travail 1ère partie



Journal de travail 2ème partie

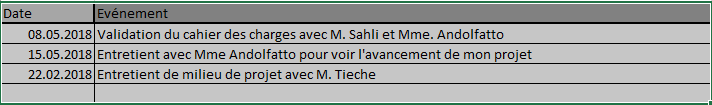


Journal de travail 3ème partie



Journal de travail 4ème partie

## Journal de bord



Journal de bord

## Manuel d'installation

Vous aurez besoin d’installer un programme avant d’utiliser l’application :

* Wamp Server, que vous pouvez télécharger ici <http://www.wampserver.com>

Démarrez Wamp. Si Wamp n’est pas démarré vous ne pourrez exécuter le script

Puis il faudra exécuter le script « planificationVolsAeriens.sql » qui permet de créer la base de données à l’aide d’un client de base de données (par exemple MySQL Workbench).

Il faut ensuite lancer l’exécutable « Application de planification de vols aériens.exe », le programme se lancera.

Il vous reste plus qu’à utiliser le programme, vous pouvez vous aidez du manuel d’utilisation présent en annexe.

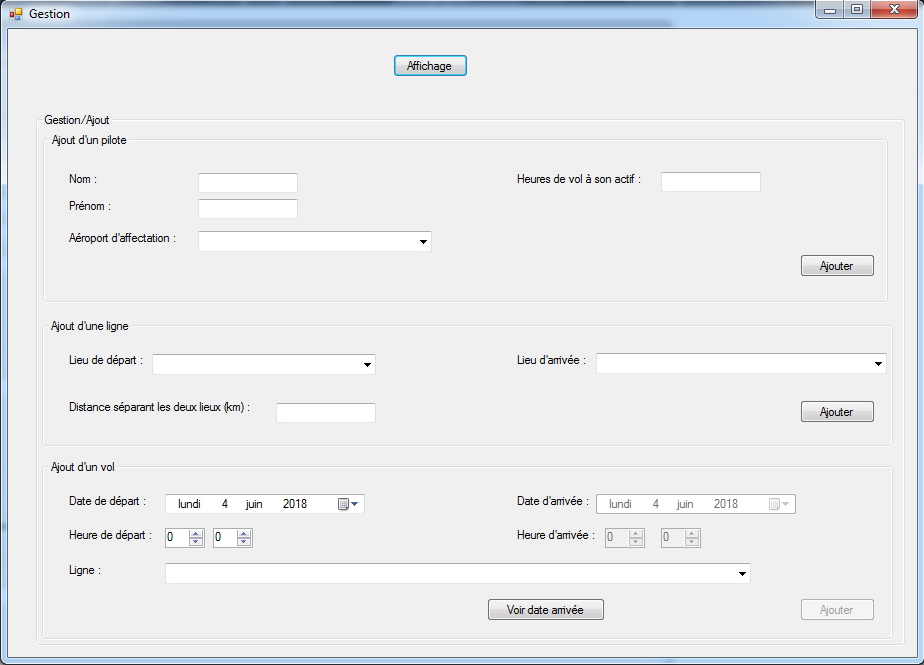
## Manuel d'utilisation

### Gestion :

La fenêtre gestion permet de créer :

* Un/des pilote(s)
* Une/des ligne(s)
* Un/des vol(s)

Et d’accéder à la fenêtre affichage en cliquant sur le bouton « Affichage ».

****

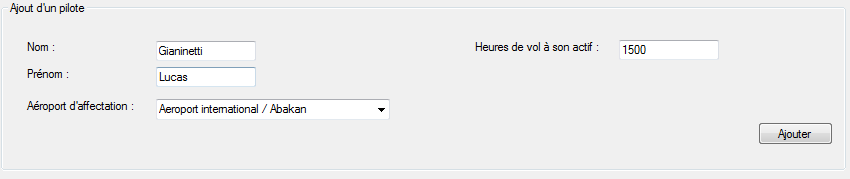
Fenêtre « Gestion »

### Créer un pilote :

Pour créer un pilote, l’utilisateur doit renseigner tous les champs du groupbox « Ajout d’un pilote » :

* Nom : Le nom de famille du pilote que l’on veut créer.
* Prénom : Le prénom du pilote que l’on veut créer.
* Heures de vol à son actif : Le nombre d’heures de vols que le pilote a déjà à son actif. Les heures de vols n’auront pas d’influence sur la suite du programme. Elles augmenteront juste après chaque vol que le pilote effectue.
* Aéroport d’affectation : L’utilisateur doit sélectionner dans la liste l’aéroport auquel sera affecté ce pilote.

Une fois ces 4 informations rentrées, l’utilisateur doit cliquer sur le bouton « Ajouter ». Si une information est manquante ou incorrecte un message avertira l’utilisateur. Si toutes les informations sont correctes un message informera l’utilisateur que le pilote a bien été ajouté.



Bloc qui permet de créer un pilote

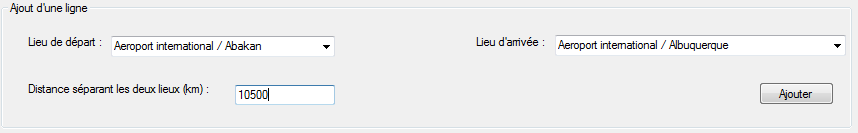
Pour l’exemple ci-dessus, le pilote qui sera créé s’appellera « Lucas Gianinetti » il aura « 1500 » heures de vol à son actif et sera affecté à « l’aéroport international d’Abakan ».

### Créer une ligne :

Pour créer une ligne, l’utilisateur doit renseigner tous les champs du groupbox « Ajout d’une ligne » :

* Lieu de départ : L’utilisateur doit sélectionner dans la liste l’aéroport de départ de la ligne.
* Lieu d’arrivée : L’utilisateur doit sélectionner dans la liste l’aéroport d’arrivée de la ligne.
* Distance séparant les deux lieu (km) : La distance qui sépare l’aéroport de départ et l’aéroport d’arrivée.

Une fois ces 3 informations rentrées, l’utilisateur doit cliquer sur le bouton « Ajouter ». Si une information est manquante ou incorrecte, un message avertira l’utilisateur. Si toutes les informations sont correctes, un message informera l’utilisateur que la ligne a bien été créée.

****

Bloc qui permet de créer une ligne

Pour l’exemple ci-dessus, la ligne qui sera créée part de l’aéroport international d’Abakan arrive à l’aéroport international d’Albuquerque pour une distance de 10500 km.

### Créer un vol :

Avant de créer un vol, il faut avoir créer une ligne sinon la liste « Ligne » ne contiendra aucune ligne et le vol ne pourra être créé.

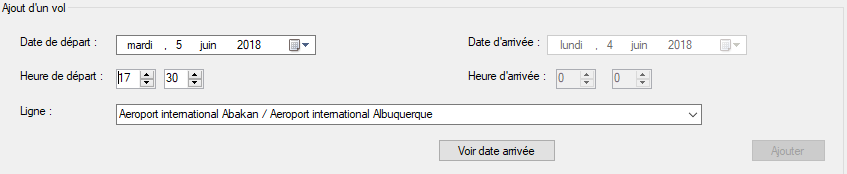
Pour créer un vol, l’utilisateur doit renseigner les champs suivant du groupbox « Ajout d’un vol » :

* Date de départ : L’utilisateur doit sélectionner grâce au DateTimePicker la date de départ du vol.
* Heure de départ : L’utilisateur doit sélectionner l’heure de départ. Le premier numericUp&Down représente les heures et le deuxième les minutes
* Ligne : L’utilisateur doit sélectionner dans la liste, la ligne sur laquelle il veut créer le vol.

Une fois ces trois informations rentrées, l’utilisateur doit cliquer sur le bouton «Voir date arrivée ». ». Si une information est manquante ou incorrecte, un message avertira l’utilisateur. Si toutes les informations sont correctes, la date d’arrivée et l’heure d’arrivée vont être mis à jour. L’utilisateur doit ensuite cliquer sur le bouton « Ajouter », un message informera que le vol a bien été ajouté.

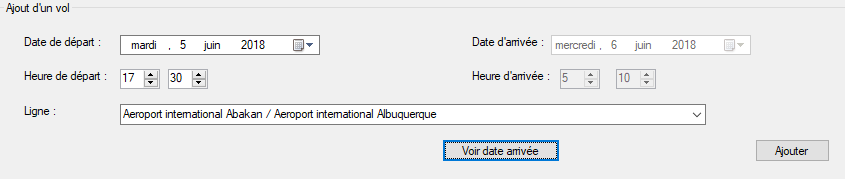
Si la date de départ, l’heure de départ ou la ligne ont été modifiés après avoir cliqué sur le bouton « Voir date arrivée », il faudra à nouveau cliquer sur ce bouton pour mettre à jour l’arrivée du vol.

Avant avoir appuyé sur « Voir date arrivée » :



Bloc qui permet de créer un vol

Après avoir appuyé sur « Voir date arrivée » :



Bloc qui permet de créer un vol

Pour l’exemple ci-dessus, le vol qui sera créé part le 5 juin 2018 à 17h30 et arrive le 6 juin 2018 à 5h10 sur la ligne Abakan – Albuquerque.

### Affichage :

Dans la fenêtre affichage, les informations concernant les pilotes, lignes et vols existants y sont affichés.

Il permet aussi d’ajouter des vacances à un pilote, d’affecter un pilote à un vol et de générer un planning de l’horaire d’un pilote.

### Affichage des informations concernant les pilotes existants :

On retrouve dans le tableau les informations suivantes pour chaque pilote existant :

* Id : il permet de différencier deux pilotes ayant le même nom et prénom
* Nom
* Prénom
* Aéroport d’affectation
* Heures de vol à son actif

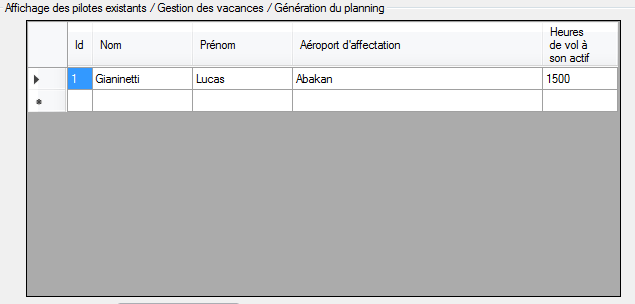
****

Tableau dans lequel sont affichés les pilotes existants

### Affichage des informations concernant les lignes existantes :

On retrouve dans le tableau les informations suivantes pour chaque ligne :

* Id : numéro de la ligne
* Lieu de départ
* Lieu d’arrivée
* Distance

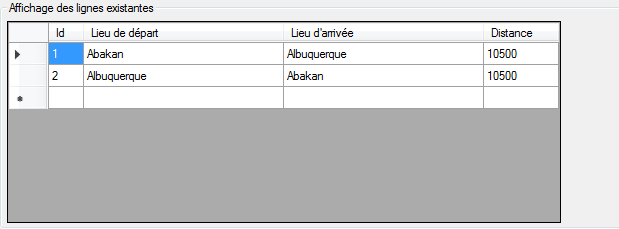


Tableau dans lequel sont affichées les lignes existantes

### Affichage des informations concernant les vols existants :

On retrouve dans le tableau les informations suivantes pour chaque vol :

* Nom du vol
* Ligne : numéro de la ligne
* Aéroport de départ
* Aéroport d’arrivée
* Date de départ
* Date d’arrivée
* Pilote n°1, le pilote affecté au vol
* Pilote n°2, si un deuxième pilote est affecté au vol

Lorsqu’aucun pilote n’a été affecté à un vol, il est normal que les informations « Pilote n°1 » et « Pilote n°2 » soient vide.

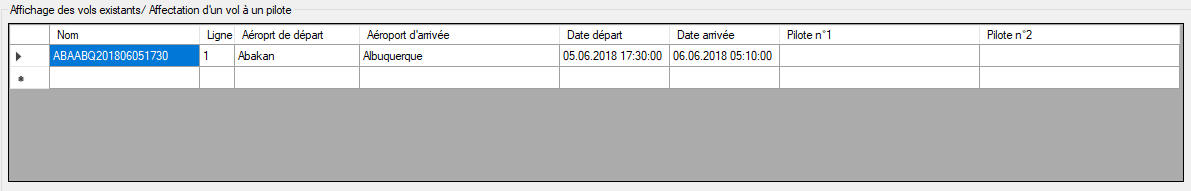
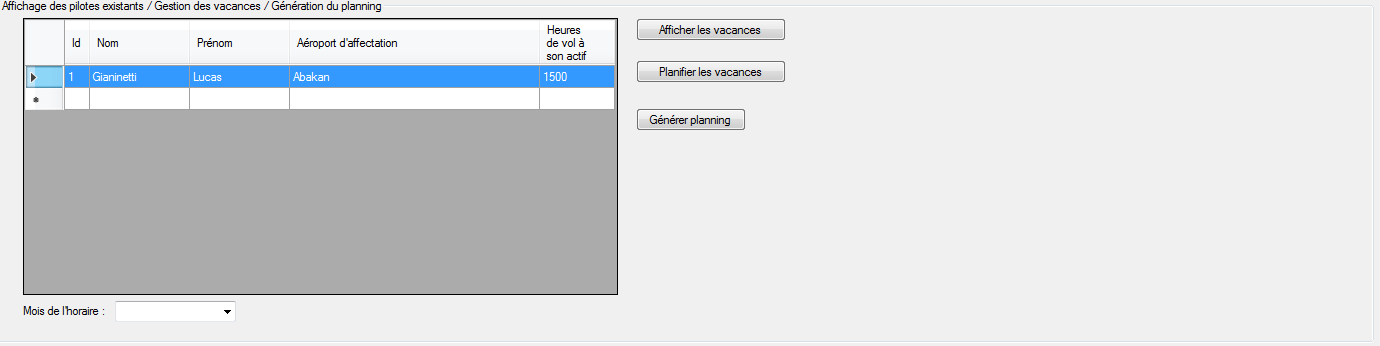


Tableau dans lequel sont affichés les vols existants

### Ajouter des vacances à un pilote :

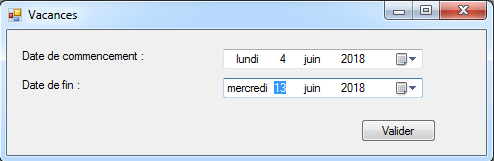
Pour ajouter des vacances à un pilote, l’utilisateur doit sélectionner une ligne dans le tableau grâce à la flèche noire qui se situe tout à gauche de la ligne :



Bloc qui permet l’affichage des pilotes, l’ajout/affichage des vacances et de générer un planning

Une fois la ligne sélectionnée, l’utilisateur doit cliquer sur le bouton « Planifier les vacances ». Si la ligne n’est pas sélectionnée correctement, un message averti l’utilisateur. Si la ligne est sélectionnée correctement, une nouvelle fenêtre dans laquelle les vacances pourront être planifiées s’ouvre :

Dans cette nouvelle fenêtre, l’utilisateur doit indiquer la date de commencement et la date de fin des vacances. Une fois le choix fait, l’utilisateur doit cliquer sur « Valider ». Si le pilote est en train de travailler, est déjà en vacances ou s’il a déjà pris ses 25jours de vacances, un message avertira l’utilisateur. Sinon les vacances seront ajoutées et un message avertira l’utilisateur de cet ajout.

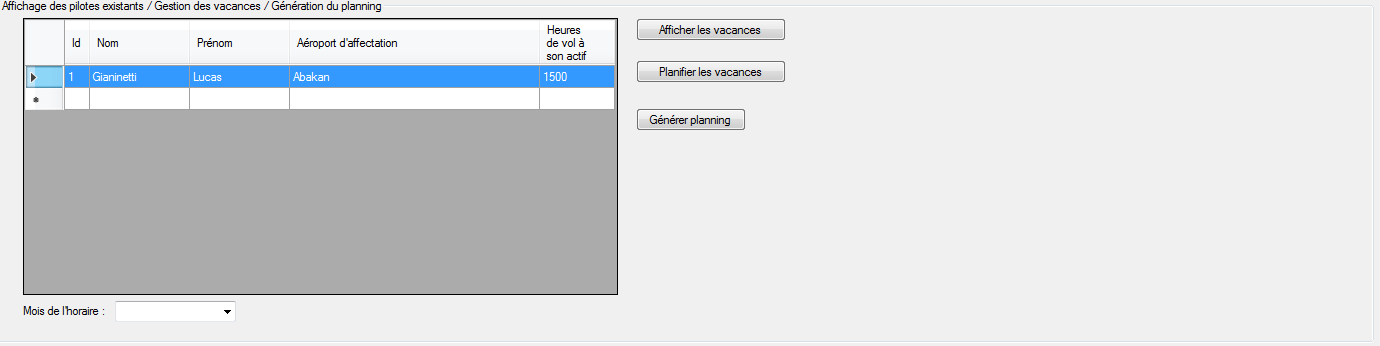


Fenêtre qui permet d’ajouter des vacances

Pour l’exemple ci-dessus, il sera ajouté au pilote numéro 1 « Gianinetti Lucas » une période de vacances du 4 juin 2018 au 13 juin 2018.

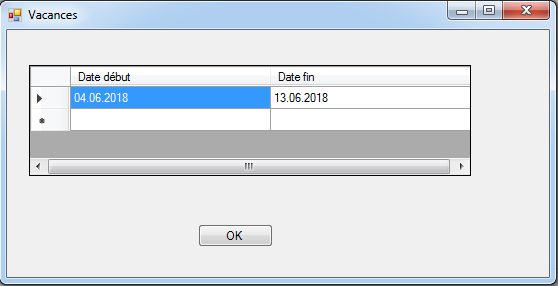
### Afficher les vacances d’un pilote :

Pour afficher les vacances d’un pilote, l’utilisateur doit sélectionner une ligne dans le tableau grâce à la flèche noire qui se situe tout à gauche de la ligne :



Bloc qui permet l’affichage des pilotes, l’ajout/affichage des vacances et de générer un planning

Une fois la ligne sélectionnée, l’utilisateur doit cliquer sur le bouton « Afficher les vacances ». Si la ligne n’est pas sélectionnée correctement, un message averti l’utilisateur. Si la ligne est sélectionnée correctement, une nouvelle fenêtre dans laquelle sont affichées les vacances du pilote s’ouvre :

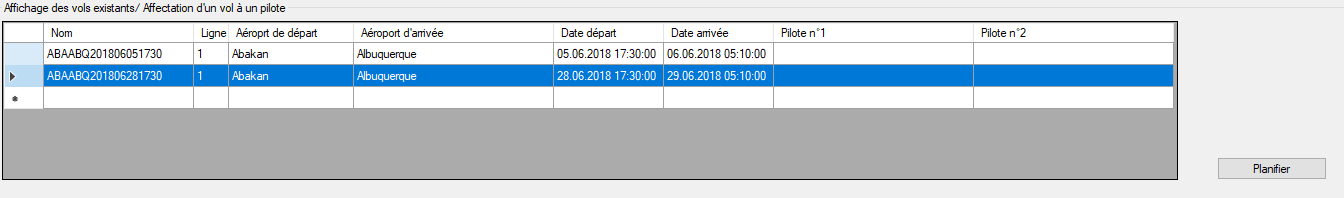


Fenêtre qui permet d’afficher les vacances

On y voit les vacances précédemment ajoutées.

### Affecter un pilote à un vol :

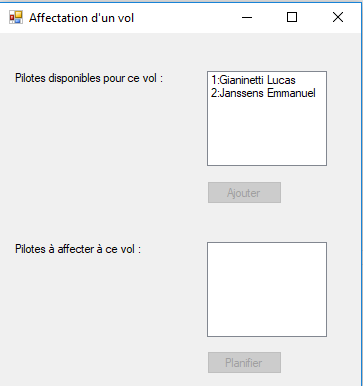
Pour affecter un pilote à un vol, l’utilisateur doit sélectionner une ligne dans le tableau grâce à la flèche noire qui se situe tout à gauche de la ligne :



Bloc qui permet d’afficher les vols et d’affecter un pilote à un vol

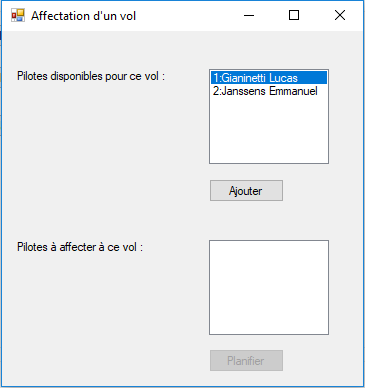
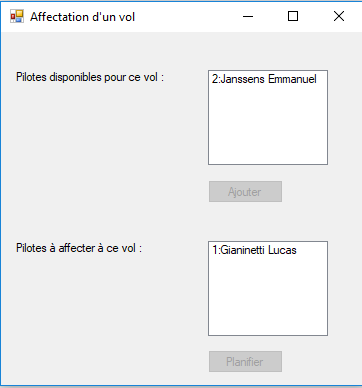
Une fois la ligne sélectionnée, l’utilisateur doit cliquer sur le bouton « Planifier ». Si la ligne n’est pas sélectionnée correctement, un message averti l’utilisateur. Si la ligne est sélectionnée correctement, une nouvelle fenêtre dans laquelle des pilotes pourront être affectés à ce vol apparait :

Dans la liste « Pilotes disponibles pour ce vol », tous les pilotes étant dans le même aéroport que l’aéroport de départ y sont affichés. Ces pilotes ne sont ni en vacances, ni déjà en train de travailler et ont eu une pause de plus de 12h depuis leur dernier vol.

****

Fenêtre qui permet d’affecter un pilote à un vol

L’utilisateur doit alors sélectionner dans la liste « Pilotes disponibles pour ce vol » le pilote qu’il veut affecter à ce vol. Une fois le pilote sélectionné, il faut cliquer sur ajouter et le pilote apparait maintenant dans la liste « Pilotes à affecter à ce vol » :



Fenêtres qui permettent d’affecter un pilote à un vol

Puis l’utilisateur doit cliquer sur « Planifier », si le vol dure plus de 10h il sera demandé à l’utilisateur d’ajouter un 2ème pilote dans la liste « Pilotes à affecter à ce vol ». S’il y a suffisamment de pilotes à affecter à ce vol, un message averti l’utilisateur et la fenêtre se ferme.

Malheureusement, le tableau ne se met pas à jour automatiquement. Pour cela, l’utilisateur doit cliquer sur le bouton « Gestions », puis quand le fenêtre « Gestions » s’ouvre cliquer sur le bouton « Affichage » afin de revenir sur la fenêtre d’affichage. Une fois de retour sur la fenêtre d’affichage on pourra voir que le tableau a été mis à jour :

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevé

Bloc qui permet d’afficher les vols et d’affecter un pilote à un vol

**Générer un planning :**

Pour pouvoir générer le planning horaire d’un pilote, l’utilisateur doit sélectionner une ligne dans le tableau grâce à la flèche noire qui se situe tout à gauche de la ligne. Et il doit aussi indiquer le mois pour lequel générer le planning grâce à la liste « Mois de l’horaire » :

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevé

Bloc qui permet l’affichage des pilotes, l’ajout/affichage des vacances et de générer un planning

Une fois la ligne sélectionnée, l’utilisateur doit cliquer sur le bouton « Générer planning ». Si la ligne n’est pas sélectionnée correctement ou si le mois de l’horaire n’est pas renseigné, un message averti l’utilisateur. Sinon un message averti que le planning a été créé, le message indique aussi dans quel dossier le planning a été créé.

Pour chaque horaire généré, un dossier est créé au nom du pilote pour lequel l’horaire a été généré (ex : 1GianinettiLucas). Les horaires créés sont situés à l’intérieur de ce dossier, dans des dossiers différents selon les années. Les horaires sont créés sous forme d’un fichier « csv », le nom du fichier est composé de l’année et du mois de l’horaire (ex : 201806.csv).

Pour pouvoir lire le fichier horaire de manière visible il faut :

* Sélectionner le fichier (en cliquant dessus)
* Faire un clic droit de la souris
* Sélectionner « Ouvrir avec »
* Sélectionner « Excel »

**Informations complémentaires :**

La distance d’une ligne ne peut dépasser les 18000km, ce qui représente un vol de 20h.

Le pilote change de localisation après avoir effectué un vol. S’il n’y a aucun pilote disponible pour le vol que vous voulez créer c’est peut-être parce que le pilote n’est plus dans l’aéroport.

S’il y a deux pilotes pour un même vol, le temps de vol sera ajouté aux deux pilotes.

## Archives du projet

Un repository GitHub a été créé pour ce projet : <https://github.com/L-Gianinetti/TPI>

Un cd contenant le projet a été remis à ma chef de projet et aux deux experts.

## Lexique

Code AITA : Code de trois lettres désignant de nombreux aéroports à travers le monde, défini par l’Association internationale du transport aérien (AITA).

UK Government Data Standards Catalogue : Catalogue qui définit les standards des données pour le gouvernement Anglais.

Windows Form : Outil de création et de gestion d’interface graphique dans un programme.

1. <https://fr.wikipedia.org/wiki/Liste_des_codes_AITA_des_a%C3%A9roports/A> [↑](#footnote-ref-1)
2. https://fr.wikipedia.org/wiki/Code\_AITA\_des\_a%C3%A9roports [↑](#footnote-ref-2)