



Rapport Technique

Sommaire :

- 1/Identification du projet
- 2/Présentation de l'équipe
- 3/Description du fonctionnement
- 4/Cahier des charges
- 5/Création du code
- 6/Création de la base de données
- 7/La liaison de JAVA à MYSQL
- 8/Démonstration
- 9/Guide d'utilisation du programme
- 10/Conclusion
- 11/Sources

1-Identification du projet

Nous avons choisi avec les membres de groupes de prendre le projet de l'aéroport car dans le groupe la plupart des membres veulent faire du développement dans leur futur métier. D'autre part, nous voulons approfondir nos connaissances sur les bases de données. Pour affiner nos connaissances dans les bases de données et dans le développement de Java.

Énoncé du projet:

L'objectif de ce projet est de réaliser un logiciel d'échange entre les différents avions qui circulent au niveau de l'aéroport de Toulouse. Ce programme va intégrer la gestion des lignes, la gestion des avions, la gestion du parcage et la gestion des retards. Dans un autre but est d'automatiser tout le système pour qu'il soit le plus autonome possible.

Le projet est à réaliser en groupe de trois. Un rapport décrivant les fonctionnalités et justifiant le choix des méthodes devra être remis à la fin du projet. De plus, un rapport technique sera demandé plus une présentation orale sera requise.

2/Présentation de l'équipe

Pour ce projet,l'équipe est composée de Maxime Lambert,Axel Graziani et Goncalo Filipe Goncalves Monteiro qui sera le chef de projet.


3/Description du fonctionnement du Projet

Dans cette partie, nous allons parler du fonctionnement que nous avons procédé pour le projet.Dans un premier temps, nous allons utiliser la méthode agile Kanban.

De ce fait,on a utilisé Trello pour nous répartir le travail et avoir un contact visuel pour bien savoir qu'on était positionné par les objectifs donnés par le chef de projet.



Exemple de description de la première séance.

 **Description** Modifier

Premier séance compréhension les rôles etc.
Définir les objectifs les rôles

 **Description** Modifier

Compréhension du sujet. Début du cahier des charges avec Axel.

En effet, après avoir bien compris le sujet avec les différents membres du groupe. Nous avons décidé de réaliser un Trello pour se donner des tâches à faire, surtout des objectifs .

Également nous avons pris connaissance dans le groupe des points forts de chacun ou il avait plus de compétences que d'autres pour mieux comprendre les besoins de chacun. Certes le but est de le faire le travail demandé mais également de s'entraider pour améliorer les compétences de chacun.

C'est pour cela que nous avons déroulé étape par étape pour bien appréhender les différentes difficultés.

- La première étape consistait à la création d'un cahier de charge en adéquation avec le projet.
- La deuxième étape était de faire tous les diagrammes nécessaires aux processus d'une création de base de données. Pour ensuite, faire une base de données.
- Notre troisième étape, la plus rude, était de transformer la partie théorique en une idée concrète en une solution technique.
- L'ultime étape est de réaliser un rapport technique puis un diaporama pour notifier tout ce que nous avons réalisé lors de ce P.P.E.2.
-

4/Cahier des charges

Pour la réalisation du cahier des charges nous sommes allés au plus simple, nous l'avons réalisé en répartissant les différentes parties qui comportent des sous parties.

Pour notre exemple, nous avons réalisé les tirets dit "les plus important" nous avons:

1.Le contexte

2.Exigences

3.Recette

4.Prestation associées

Dans ses différents tirets nous avons des sous parties qui viennent éclairer le titre comme pour **Le Contexte** nous avons:

-1.1Présentation de l'entité

-1.2Situation Actuelle

-1.3Acteurs

-1.4Objectif du projet

Comme seconde partie, nous retrouvons **Les Exigences:**

-2.1Exigences fonctionnelles

-2.2Exigences matérielles

-2.3Exigences Logicielles

Pour **La Recette** nous avons comme sous parties

-3.1Documents

-3.2Délais

Pour finir notre dernière tiret est **La prestations associées:**

-4.1Logicielle

-4.2Livraison

Dans les sous-parties, le but est de répondre à un sujet précis contrairement aux tirets qui restent quant à eux assez vastes. Pour avoir un vrai support vous retrouverez le cahier des charges en annexes.

5/Création du code

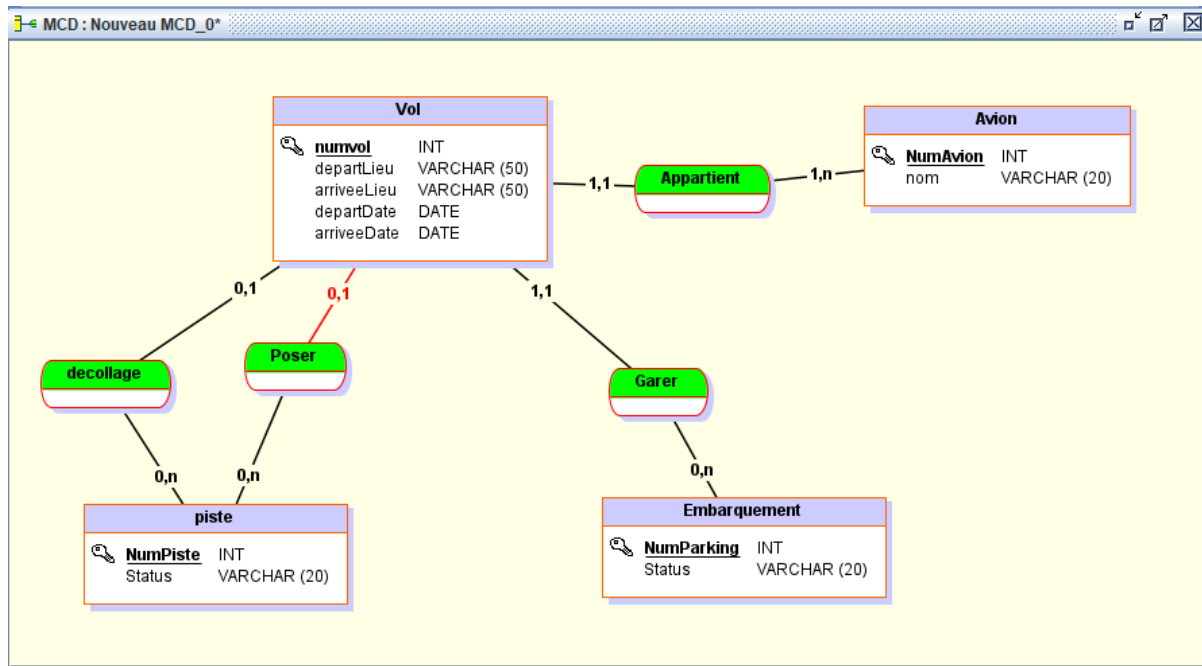
Pour la création du code nous avons choisi d'utiliser du Java qui restait à nos yeux l'option la plus plausible et la plus simple pour mettre en œuvre le projet dans de bonnes conditions. Pour ce faire, nous avons utilisé l'application Netbeans pour coder notre programme. Dans notre dossier P.P.E2, vous pouvez trouver les différentes versions du code à l'avancée de chaque cours de projet vous pouvez voir l'évolution du code. Il faut l'avouer c'était l'étape avec la création de base de donnée ou on avait le plus d'appréhension.

Malheureusement, c'est l'étape où nous avons eu du mal car malgré nos connaissances dans le langage. Notre projet est différent des cas qu'on a réalisés en classe c'était un projet où il fallait relier la base de donnée et le code Java.

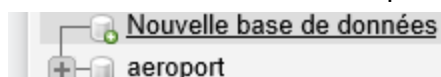
Cela signifie que le code dans certains détails diffère mais malgré nos difficultés nous avons pris nos courage à deux mains. Après plusieurs versions du code nous avons réussi à développer un programme. Mais nous n'avons pas réussi à finaliser le programme malgré nos efforts. Il faut l'admettre nous avons eu du mal dans le code à se retrouver. Manque de compréhension entre l'étape pseudo code et le code. Le passage au code fut dur. Cependant nous avons essayé de faire au mieux. De plus, maintenant on comprend la complexité du code pour tout mettre en œuvre. Nous ressortons de cette expérience grandit avec plus de connaissance et d'expérience dans la création d'un programme

6/Création de la base de donnée

Tout d'abord, nous avons créé un MCD afin de construire la base de données de l'aéroport, on a utilisé le logiciel JMerise pour avoir directement le MLD qui permet de construire automatiquement la base de données dans PhpMyAdmin.



Une fois ce MCD fait on transpose le code dans une base de données MySQL "aéroport"



Une fois la base de données mis dans MySQL on obtient cette base de données

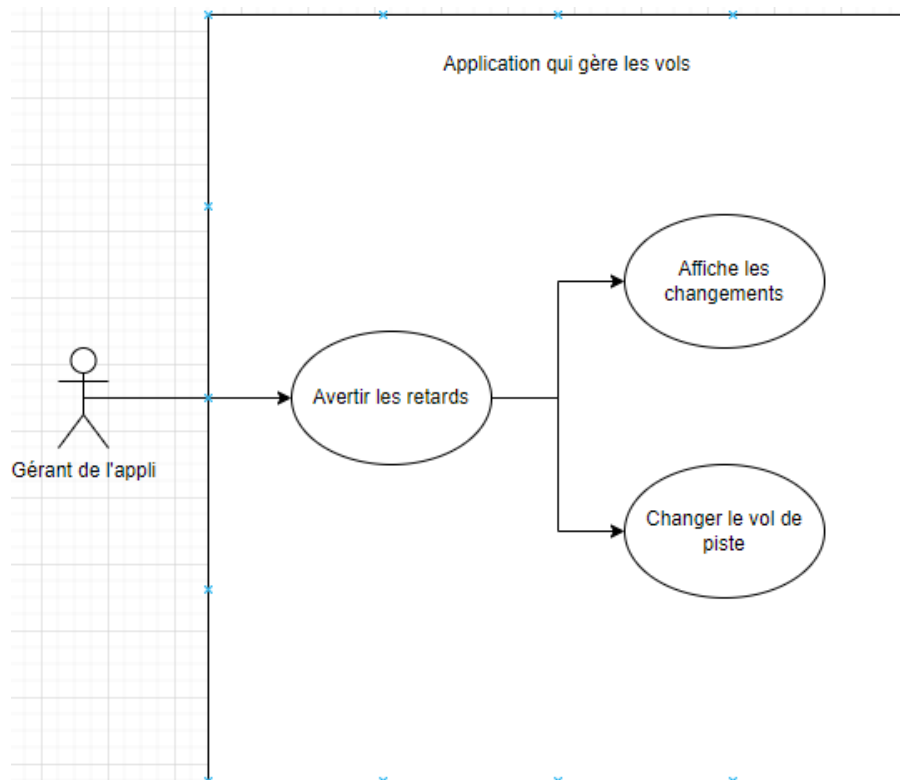
<input type="checkbox"/>	avion	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	2	InnoDB	utf8_general_ci	16,0	kio	-
<input type="checkbox"/>	embarquement	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	2	InnoDB	utf8_general_ci	16,0	kio	-
<input type="checkbox"/>	piste	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	2	InnoDB	utf8_general_ci	16,0	kio	-
<input type="checkbox"/>	vol	★	Parcourir	Structure	Rechercher	Insérer	Vider	Supprimer	2	InnoDB	utf8_general_ci	80,0	kio	-
4 tables		Somme							8	MyISAM	utf8_general_ci	128,0	kio	0 o

Maintenant il n'y a plus qu'à insérer les vols, avion, piste et embarquement afin de pouvoir utiliser ces données

6/Création des fonctionnalités

Dans cette partie, nous avons réalisé différents diagrammes pour faciliter la tâche au niveau de la base de données. En effet, nous avons réalisé différents diagrammes. Dans un premier temps, le diagramme de cas d'utilisation. Puis, dans un second temps un diagramme de séquences, pour finir un diagramme de classe.

Diagramme de cas d'utilisation:



Diagrammes de séquences:

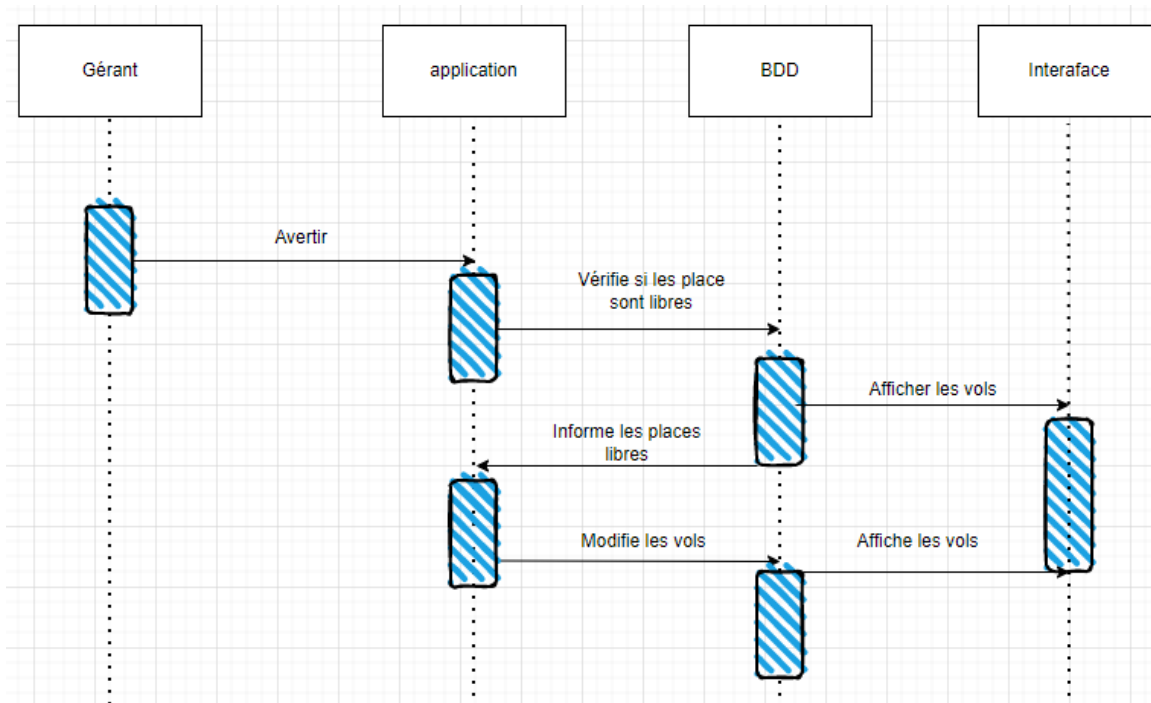
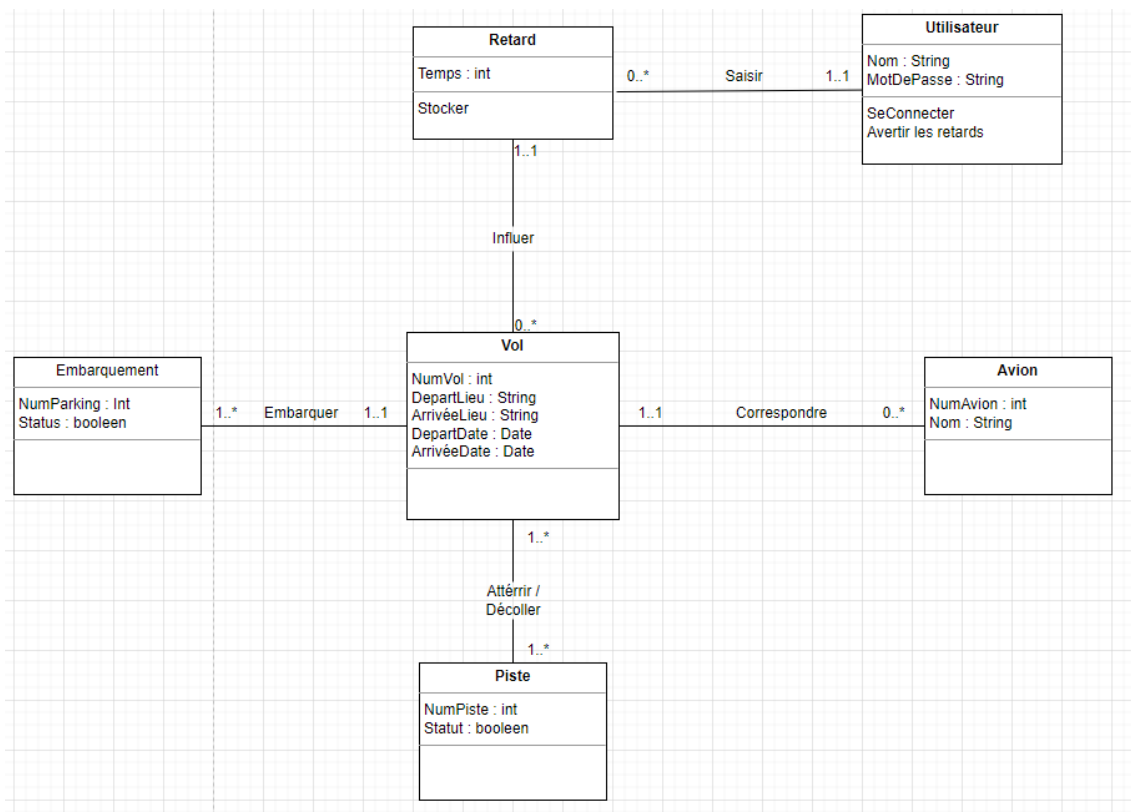


Diagramme de classes:



7/La liaison de Java et Mysql

Dans cette partie, nous allons parler de la liaison de Java et mysql. En effet, dans notre programme, il fallait utiliser une base de données pour stocker des informations comme le nom des avions et le vol puis l'heure de départ.

Pour cela, on s'est aidé du cours vu en classe et les tp réalisés en classe.

Pour créer une base de données nous avons utilisé comme application phpmyadmin.

On peut le voir dans la capture d'écran, nous avons créé une classe java où on a mis les informations de connexion pour que le programme se connecte. Nous avons téléchargé une librairie de Mysql pour bien avoir la connexion.

```
}  
public static Connection connecterDB() {  
    try{  
        Class.forName("com.mysql.jdbc.Driver");  
        String url="jdbc:mysql://localhost/aéroport";  
        String user="root";  
        String password="";  
        Connection cnx=DriverManager.getConnection(url,user,password);  
        System.out.println("Connexion bien établie");  
        return cnx;  
    } catch (Exception e) {  
        e.printStackTrace();  
        return null;  
    }  
}
```

8/Démonstration



Dans cette partie nous allons vous montrer les différentes parties du code comme par exemple les classes, les getters, les setters, les constructeurs ainsi que d'autre partie intéressante dans le code.

Dans un premier temps, nous avons pris un temps pour se poser les bonnes questions pour savoir où on se dirige, les grandes étapes du code. Après s'être concerté nous avons mis comme étapes les suivantes : Créer les classes, créer les constructeurs, créer les getters et setters, la liaison de la base de données avec le code, création des requêtes Sql, Les boucles.

Créer les classes

Ici dans notre cas nous avons créé les classes piste, vol, avion, embarquement.



Créer les constructeurs

Pour chaque classe il fallait créer un constructeur pour chaque classe que nous avons créé précédemment.

```
public class Connect {
    private avion[] avions=new avion[0];
    private Piste[] pistes=new Piste[0];
    private Vol[] vols=new Vol[0];
    private embarquement[] embarquements=new embarquement[0];
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
    }

    public avion(double NumAvion, String Nom) {
        this.NumAvion = NumAvion;
        this.Nom = Nom;
    }

    /**
    public class Vol {
        private double NumVol;
        private String Avion;
        private String Depart;
        private String Arrivee;
        private double NumAvion;

        public Vol(double NumVol, String Avion, String Depart, String Arrivee, double NumAvion) {
            this.NumVol = NumVol;
            this.Avion = Avion;
            this.Depart = Depart;
            this.Arrivee = Arrivee;
            this.NumAvion = NumAvion;
        }
    }
```

Créer les getters et setters

Comme pour les constructeurs nous avons créé pour chaque classe des getters et setters.

```
public String getNom() {  
    return Nom;  
}  
  
public void setNom(String Nom) {  
    this.Nom = Nom;  
}  
  
public void setNumAvion(double NumAvion) {  
    this.NumAvion = NumAvion;  
}  
  
public embarquement(String libre, String occupé) {  
    this.libre = libre;  
    this.occupé = occupé;  
}  
  
public String getLibre() {  
    return libre;  
}  
  
public void setLibre(String libre) {  
    this.libre = libre;  
}  
  
public String getOccupé() {  
    return occupé;  
}  
  
public void setOccupé(String occupé) {  
    this.occupé = occupé;  
}
```

La liaison de la base de données avec le code

Ici, vous trouverez le lien entre la base de donnée et Php mais si vous voulez plus de précision vous avez une partie sur l'explication de comment nous avons réussi.

```
}  
public static Connection connecterDB() {  
    try{  
        Class.forName("com.mysql.jdbc.Driver");  
        String url="jdbc:mysql://localhost/aeroporto";  
        String user="root";  
        String password="";  
        Connection cnx=DriverManager.getConnection(url,user,password);  
        System.out.println("Connexion bien établie");  
        return cnx;  
    }catch(Exception e){  
        e.printStackTrace();  
        return null;  
    }  
}
```

Des requêtes Sql, Les boucles

Ici nous avons des boucles qui permettent d'ajouter une piste. C'est ici que nous avons terminé le code. Ensuite, si on avait plus de temps, on aurait créé les boucles pour toutes les situations possibles puis finir le programme.

```
public void AjouterPiste(Piste Piste){  
    pistes = addPiste(pistes.length,pistes,Piste);  
}  
public static Piste[] addPiste(int n,Piste liste[], Piste x){  
    int i;  
    Piste nouvelleliste[] = new Piste[n + 1];  
    for (i=0; i<n; i++)  
        nouvelleliste[i] = liste[i];  
    nouvelleliste[n] = x;  
    return nouvelleliste;  
}
```

```
public void AjouterAvion(avion avion){
    avions = addavion(avions.length,avions,avion);
}
public static avion[] addavion(int n,avion liste[], avion x){
    int i;
    avion nouvelleliste[] = new avion[n + 1];
    for (i=0; i<n; i++)
        nouvelleliste[i] = liste[i];
    nouvelleliste[n] = x;
    return nouvelleliste;
}
```

9/Guide d'utilisation du programme

Pour le guide d'utilisation du programme nous voulions montrer le résultat comme nous n'avons pas réussi à finaliser le programme, nous avons réalisé une maquette pour vous montrer notre rendu final idéal.

Ca aurait été comme un tableau ou il y a toutes les informations des avions comme l'heure de départ, l'heure de d'arrivée, la destination, le nom de piste etc.

Avec le code le programme aurait fonctionné en totale autonomie. La tour de contrôle doit notifier dans le programme si les avions sont bien arrivés ou sont en retard et automatiquement cela va montrer dans le programme libre ou en

Boeing	1	17h30	Paris	Libre/ Retard
Airbus	2	9h20	Tokyo	Libre/ Retard
Antonov	1	16h	Seoul	Libre/ Retard
Boeing	2	12h15	Toulouse	Libre/ Retard
Airbus	1	6h30	Paris	Libre/ Retard
Antonov	2	20h30	Paris	Libre/ Retard

Avion Piste Heure Destination Libre

Boeing	1	17h30	Paris	Libre/ Retard
Airbus	2	9h20	Tokyo	Libre/ Retard
Antonov	1	16h	Seoul	Libre/ Retard
Boeing	2	12h15	Toulouse	Libre/ Retard
Airbus	1	6h30	Paris	Libre/ Retard
Antonov	2	20h30	Paris	Libre/ Retard

retard.

10/Conclusion

Pour conclure ce projet nous a permis de découvrir des facettes comme la liaison entre phpmyadmin et netbean. Même si notre programme n'est pas fini on a mit tous nos efforts cela montre à quel point le métier de développeur est complexe ou il faut avoir beaucoup de compétences techniques. Nous ressortons de ce projet très consciencieux du travail d'un codeur. Ou un différent problème peut prendre du temps à se résoudre.

Cependant nous avons effectué une grande avancée dans nos connaissance dans le code et java mais aussi dans les base de donné qui n'est pas également une tâche facile. Même si dans le domaine de la base de données nous avons réussi a réalisé la base de données puis le Mcd et le Mld. On comprend que pour qu'un programme fonctionne avec une base de données cela demande des compétences techniques dans plusieurs domaines.



CONCLUSION

11/Sources

Pour les sources nous avons utilisé généralement les cours de M.Armstrong pour le code Java.

Pour la base de données nous avons utilisé les cours de M.Monsteills.

Nous avons très peu utilisé des sources internet.

Pour connecter Mysql à netbeans nous avons utilisé un tuto comme sur Delft Stack

<https://www.delftstack.com/fr/howto/java/connect-mysql-database-in-java/>

ANNEXES 1:CAHIER DES CHARGES

LAMBERT.Maxime GONCALVES.Filipe GRAZIANI.Axel

Cahier des charges

1.Contexte

1.1 Présentation de l'entité:

L'aéroport de Toulouse-Blagnac est un aéroport international français. Avec une fréquentation de plus de 9 millions de personnes et en moyenne 270 avions par jour qui desservent plus de 70 destinations.

1.2. Situation Actuelle:

Le projet est à réaliser en groupe de trois. Un rapport décrivant les fonctionnalités et justifiant le choix des méthodes devra être remis à la fin du projet. De plus, un rapport technique sera demandé plus une présentation orale sera requise.

1.3 Acteurs

M.MÉLENCHON, Directeur de l'aéroport de Toulouse.

M.LAMBERT,M,GRAZIANI,M.GONCALVES MONTEIRO responsable du cahier des charges.

1.4. Objectif du projet:

L'objectif de ce projet est de réaliser un logiciel d'échange entre les différents avions qui circulent au niveau de l'aéroport de Toulouse. Ce programme va intégrer la gestion des lignes, la gestion des avions, la gestion du parcage et la gestion des retards. Dans un autre but est d'automatiser tout le système pour qu'il soit le plus autonome possible.

2. Exigences

2.1. Exigences fonctionnelles

Le logiciel doit être fonctionnel avec les ordinateurs de l'aéroport. De plus, on ne peut pas avoir 2 avions sur la même place à la même heure. La place doit être libre quelques minutes avant l'arrivée d'un avion.

Si il y a un retard, les autres avions doivent signaler pour informer les autres avions.

2.2. Exigences matérielles

L'aéroport de Toulouse souhaite un hébergement des données du logiciel sur des serveurs. Cependant, elle souhaiterait en acheter puisqu'elle ne dispose actuellement pas d'hébergement pour ses données.

2.3. Exigences Logicielles

Le logiciel doit être compatible avec les systèmes d'exploitation suivants:

- Windows
- Mac OS
- Linux

3. Recette

3.1. Documents

Lors de la recette un document technique, un document décrivant comment utiliser le logiciel.

3.2. Délais

Le 07/05/2022 sera remis la documentation sur les techniques qui doivent être utilisées pour avoir une utilisation du logiciel le plus performant possible.

4. Prestations associées

4.1. Logicielle

Le logiciel devra être installé sur chacune des machines servant à la gestion du logiciel, soit un total de 5 machines.

4.2. Livraison

Le logiciel sera installé le 07/04/2022 dans les différents postes qui ont une nécessité.

● Définition des objectifs

- Créer un affichage de vol d'avion* et nombre de place par vol (ex: 192/200 place prise)
- Gestion de l'aéroport et parcage (si retard alors décaler les heures de départ d'avion sur la place)

- **Analyse des besoins et faisabilité**

- Information sur les vols
- 6 semaines
- Réussir à avoir la base de données ainsi que l'interface
- Automatique

- **Spécifications (fonctionnelles, techniques)**

Conditions :

- On ne peut pas avoir 2 avion sur la même place à la même heure
- La place doit être libre quelques minutes avant l'arrivée d'un avion
- Si il y a un retard, les autres avions doivent signaler un retard aussi
- Décaler l'avion sur une autre piste si emplacement pris

- **Conception (architecturales, détaillée)**

Parking	Piste (n°)	id avion	Heure de départ	heure d'arrivée	retard	Place libre ?
1	1	1	10h	9h55	a l'heure / En retard	oui / non
...
...

- **Codage**

- Java/MySQL

- **Tests (unitaires, intégration, validation)**

- **Déploiement (Livraison, Mise en production)**

- Avril 2022

- interface de vol

- **Maintenance**

- Modification des vols par un employé

*affichage de vol d'avion (Départ/Arrivé, date/heure de départ/arrivé, terminal, emplacement (porte d'embarquement, nom de l'avion / n° vol), nombre de place prise (annulation))