

## UNIVERSITE ABDELMALEK ESSAADI

### Master IA et Science de Données

#### *Module : Les systèmes multi-agentes*

Sujet : Application SMA pour le contrôle aérien.

Réalisé PAR :

MAHRI AYMANE.

SABBAHI MOHAMED AMINE.

ENCADRE PAR :

Pr. El Mokhtar EN-NAIMI.

---

## TABLE DES MATIÈRES

|   |           |
|---|-----------|
| <b>1. Introduction :</b>  | <b>1</b>  |
| <b>2. Problématique :</b>                                       | <b>2</b>  |
| <b>3. Objectif de Projet :</b>                                  | <b>3</b>  |
| <b>4. Système Multi-Agent pour la Gestion du Trafic Aérien:</b> | <b>4</b>  |
| <b>1.4 C'est quoi un système Multi-Agent :</b>                  | <b>4</b>  |
| <b>2.4 Architecture du SMA pour le projet :</b>                 | <b>4</b>  |
| <b>5. Conception :</b>  | <b>6</b>  |
| <b>1.5 Diagramme de Cas d'Utilisation :</b>                     | <b>6</b>  |
| <b>2.5 Diagramme de Classes :</b>                               | <b>7</b>  |
| <b>3.5 Les diagrammes de séquence :</b>                         | <b>8</b>  |
| <b>4.5 Diagrammes d'État de Transition :</b>                    | <b>10</b> |
| <b>6. Réalisation du projet :</b>                               | <b>12</b> |
| <b>1.6 La structure de projet sur IntelliJ :</b>                | <b>12</b> |
| <b>2.6 Exécution de projet :</b>                                | <b>13</b> |
| <b>3.6 La base de données :</b>                                 | <b>15</b> |
| <b>4.6 Exemple d'exécution pratique :</b>                       | <b>16</b> |
| <b>5.6 Règles et informations à respecter :</b>                 | <b>18</b> |
| <b>7. Outils Utilisés :</b>                                     | <b>19</b> |
| <b>1.7 JADE (Java Agent Développment Framework) :</b>           | <b>19</b> |
| <b>2.7 JavaFX :</b>   | <b>19</b> |
| <b>3.7 IntelliJ IDEA :</b>                                      | <b>19</b> |
| <b>4.7 StarUML :</b>  | <b>20</b> |

---

|                        |           |
|------------------------|-----------|
| <b>5.7 MySQL :</b>     | <b>20</b> |
| <b>6.7 XAMPP :</b>     | <b>20</b> |
| <b>7.7 GitHub :</b>    | <b>20</b> |
| <b>8. Conclusion :</b> | <b>21</b> |
| <b>9. Références :</b> | <b>22</b> |

## TABLE DES FIGURES

|  |                                     |
|--|-------------------------------------|
| Figure 1: Schéma de trois agents dans un environnement communiquant entre eux..... | 4                                   |
| Figure 2: Diagramme de cas d'utilisation.....                                      | 6                                   |
| Figure 3: Diagramme de classes.....  | 7                                   |
| Figure 4: Diagramme de Séquence - Allocation Piste. ....                           | 8                                   |
| Figure 5: Diagramme de Séquence - Demande Atterrissage et Décollage..              | 9                                   |
| Figure 6: Diagrammes d'État de Transition - Allocation de Piste. ....              | 10                                  |
| Figure 7: Diagrammes d'État de Transition - État de l'Avion.....                   | 10                                  |
| Figure 8: Diagrammes d'État de Transition - Demande d'Atterrissage. ....           | 11                                  |
| Figure 9: Diagrammes d'État de Transition - Demande de Décollage.....              | 11                                  |
| Figure 10: La structure de projet. ....  | 12                                  |
| Figure 11: Lancement de Main-Container sur JADE. ....                              | 13                                  |
| Figure 12: Interface d'agent avion. ....   | 13                                  |
| Figure 13: Interface Compagnie Aerienne. ....                                      | 14                                  |
| Figure 14: Interface Directeur Flux.....   | 14                                  |
| Figure 15: Interface Gestionnaire Trafic.....                                      | 15                                  |
| Figure 16: la base de données Allocation.....                                      | 15                                  |
| Figure 17: Exemple de réservation sur l'interface avion.....                       | 16                                  |
| Figure 18: Exemple de réservation sur l'interface Gestionnaire. ....               | 16                                  |
| Figure 19: Exemple de demande d'atterrissage dans l'interface Avion. ....          | 17                                  |
| Figure 20: assigner une piste disponible à l'avion.....                            | 17                                  |
| Figure 21: Mise à jour du statut de l'avion dans la base de données. ....          | 17                                  |
| Figure 22: Exemple de demande de décollage.....                                    | 18                                  |
| Figure 23: Communication entre Gestionnaire et Directeur. ....                     | 18                                  |
| Figure 24: Logo de JADE.....   | 19                                  |
| Figure 24: Logo de JADE.....   | 19                                  |
| Figure 25: Logo JavaFX. ....   | 19                                  |
| Figure 25: Logo JavaFX. ....   | 19                                  |
| Figure 26: Logo de IntelliJ IDEA.....  | 19                                  |
| Figure 26: Logo de IntelliJ IDEA.....  | 19                                  |
| Figure 27: Logo de StarUML. ....   | 20                                  |
| Figure 27: Logo de StarUML. ....   | <b>Error! Bookmark not defined.</b> |
| Figure 28: Logo de MySQL. ....   | 20                                  |

---

|                                 |                                     |
|---------------------------------|-------------------------------------|
| Figure 28: Logo de MySQL. ....  | <b>Error! Bookmark not defined.</b> |
| Figure 29: Logo de XAMPP.....   | 20                                  |
| Figure 29: Logo de XAMPP.....   | <b>Error! Bookmark not defined.</b> |
| Figure 30: Logo de GitHub. .... | 20                                  |
| Figure 30: Logo de GitHub. .... | <b>Error! Bookmark not defined.</b> |

---

## 1. Introduction :

Le contrôle du trafic aérien est une composante cruciale pour assurer la sécurité et l'efficacité dans le ciel. Face à une augmentation continue du nombre de vols partout dans le monde, la gestion efficace du trafic aérien est devenue un défi majeur pour les aéroports et les contrôleurs aériens. Les problèmes de congestion entraînent des retards coûteux et peuvent compromettre la sécurité des passagers. Pour aborder ces défis, les chercheurs et les professionnels du secteur ont exploré diverses stratégies, notamment la gestion de la demande et l'amélioration de la capacité des aéroports.

Dans ce contexte, l'adoption de systèmes multi-agents (SMA) représente une approche prometteuse pour optimiser la gestion du trafic aérien. Les SMA permettent une simulation et une coordination dynamiques entre les agents représentant les avions, les tours de contrôle, et d'autres entités impliquées, favorisant ainsi une gestion plus agile et adaptative du trafic. La plateforme JADE (Java Agent DEvelopment Framework) offre un environnement idéal pour le développement de ces systèmes, grâce à sa robustesse et sa flexibilité.

Ce rapport explore comment les SMA, conçus via la méthodologie AUML (Agent Unified Modeling Language), peuvent être utilisés pour améliorer la gestion de la congestion du trafic aérien. Nous discuterons des principes de la gestion de la demande, tels que l'allocation de créneaux horaires et la réduction de la surcharge pendant les heures de pointe, ainsi que des méthodes pour augmenter la capacité aéroportuaire à travers des séquences optimisées de départs et d'arrivées. L'objectif est de développer un système SMA qui non seulement améliore l'efficacité opérationnelle mais aussi augmente la sécurité globale du trafic aérien.

---

## 2. Problématique :

**Gestion de la congestion du trafic aérien : Quelles stratégies et technologies peuvent optimiser la coordination et l'efficacité du contrôle aérien face à une demande croissante ?**

La gestion de la congestion du trafic aérien est un défi majeur pour les systèmes de contrôle aérien du monde entier. Avec l'accroissement du volume du trafic aérien, exacerbé par une capacité limitée des infrastructures existantes, le risque de retards, de dépenses accrues et de répercussions négatives sur l'environnement s'intensifie. La problématique centrale de ce rapport est de déterminer les méthodes les plus efficaces pour gérer cette congestion. Cette question se décompose en plusieurs sous-problèmes essentiels à explorer :

**Évaluation de la capacité actuelle :** Quelle est la capacité actuelle des systèmes de contrôle du trafic aérien et comment est-elle exploitée ? Quels sont les principaux goulets d'étranglement et les facteurs qui limitent l'efficacité de la gestion du trafic ?

**Stratégies de gestion de la demande :** Comment les stratégies de gestion de la demande, telles que l'allocation de slots horaires et la régulation des départs pendant les heures de pointe, peuvent-elles être optimisées pour réduire la surcharge du système ? Quelles innovations peuvent être introduites pour améliorer ces méthodes ?

**Amélioration de la capacité par la technologie :** Quelles technologies peuvent être mises en œuvre pour augmenter la capacité des aéroports sans nécessiter de nouveaux aménagements physiques ? Comment les systèmes multi-agents et d'autres outils informatiques avancés, tels que l'intelligence artificielle et l'analyse prédictive, peuvent-ils contribuer à une meilleure planification et gestion des flux de trafic ?

---

### **3. Objectif de Projet :**

L'objectif principal de ce projet est de résoudre le problème persistant de la congestion du trafic aérien, qui affecte les aéroports et les espaces aériens du monde entier. Pour y parvenir, le projet vise à développer et à mettre en œuvre un système multi-agent sophistiqué, capable de simuler et de gérer de manière dynamique et efficace les interactions et les mouvements complexes des avions dans les aéroports et lors des phases de vol.

Étapes clés pour atteindre cet objectif :

#### **Conception détaillée du système :**

- Utilisation de la méthodologie AUMML pour une modélisation précise des agents.
- Définition des rôles spécifiques et des comportements de chaque agent au sein du système, incluant les contrôleurs aériens, les avions, et les gestionnaires de terminaux.
- Spécification des interactions et des protocoles de communication entre les agents et leur environnement pour assurer une gestion fluide et coordonnée.

#### **Développement du système multi-agent :**

- Implémentation du système en utilisant la plateforme JADE, reconnue pour sa robustesse et son adaptabilité dans le développement de systèmes basés sur des agents.
- Intégration de bibliothèques et d'outils fournis par JADE pour faciliter la création, la communication, et la coordination des agents.
- Test et validation des performances du système en situations simulées de haute congestion.

#### **Impact attendu :**

- Réduction significative des retards dus à la congestion aérienne.
- Amélioration de la sécurité et de l'efficacité des opérations aéroportuaires.
- Augmentation de la satisfaction des compagnies aériennes et des passagers grâce à une meilleure régularité et prévisibilité des vols.

En mettant en œuvre ce système multi-agent, le projet aspire non seulement à améliorer la fluidité du trafic aérien mais aussi à contribuer à la durabilité à long terme des opérations aéroportuaires et aériennes.



## 4. Système Multi-Agent pour la Gestion du Trafic Aérien:

### 1.4 C'est quoi un système Multi-Agent :

Un système multi-agent (SMA) est un ensemble informatique composé de plusieurs entités autonomes appelées agents. Chacun de ces agents est capable de percevoir son environnement, de prendre des décisions indépendantes et d'agir de manière autonome pour atteindre des objectifs qui peuvent être communs ou individuels à chaque agent. Les agents dans un SMA peuvent posséder des connaissances, des compétences et des objectifs variés, et interagir les uns avec les autres à travers des mécanismes tels que la communication, la négociation, la coopération, ou la concurrence. Cette interaction peut varier en fonction de la nature et des exigences du système. Les SMA sont particulièrement utiles pour modéliser et résoudre des problèmes complexes dans des domaines aussi variés que la logistique, la planification, la robotique, les sciences sociales et la finance, offrant une méthode de résolution flexible et décentralisée qui dépasse les capacités d'un agent centralisé unique.

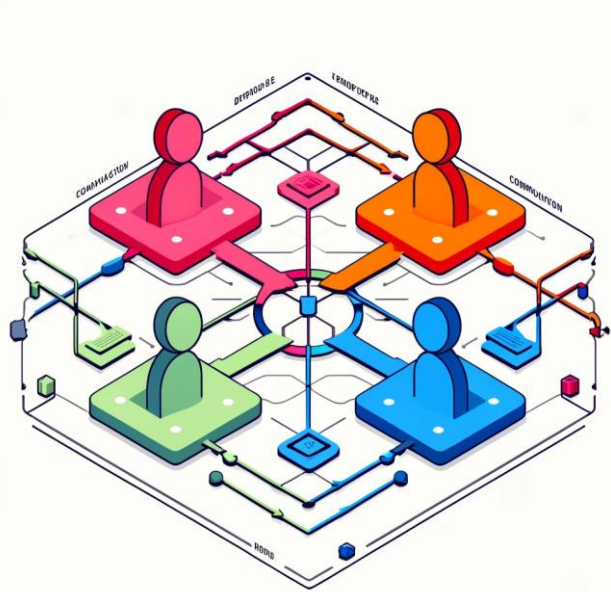


Figure 1: Schéma de trois agents dans un environnement communiquant entre eux.

### 2.4 Architecture du SMA pour le projet :

Le SMA développé pour la gestion du trafic aérien dans notre projet vise à intégrer plusieurs types d'agents spécifiques, chacun jouant un rôle crucial dans la fluidité et la sécurité du trafic aérien. Les agents impliqués dans ce système incluent les avions, les pilotes, les gestionnaires de trafic, les compagnies aériennes et les directeurs de flux. Chaque agent interagit dans un cadre complexe pour optimiser les opérations aéroportuaires et aériennes. Voici une description détaillée des rôles de chaque agent :

---

- **Avion :**

Représente l'entité physique de l'appareil, gérant ses propres paramètres de vol, incluant la vitesse, l'altitude et la position. Cet agent communique activement avec le pilote et le directeur de flux pour ajuster sa trajectoire en fonction des instructions reçues.

- **Pilote :**

Simule les décisions humaines nécessaires au pilotage de l'avion. Cet agent reçoit les données de l'avion et les commandes du gestionnaire de trafic, exécutant les manœuvres de vol et réagissant aux instructions de navigation.

- **Gestionnaire de Trafic :**

Responsable de la surveillance et de la gestion des mouvements d'avions au sein d'un secteur spécifique de l'espace aérien ou de l'aéroport. Cet agent coordonne entre les différents avions et les directeurs de flux pour s'assurer que le trafic est géré de manière sûre et efficace.

- **Compagnie Aérienne :**

Cet agent représente les intérêts commerciaux et opérationnels d'une compagnie aérienne. Il interagit principalement avec les gestionnaires de trafic et les directeurs de flux pour négocier les slots de départs et d'arrivées et pour ajuster les opérations en fonction des besoins de la compagnie.

- **Directeur de Flux :**

Supervise et optimise le flux de trafic aérien sur une échelle plus large, coordonnant les mouvements entre différents secteurs et contrôlant les routes aériennes pour éviter les congestions. Ce rôle est crucial pour la gestion des pics de trafic et pour l'intégration des prévisions et des événements imprévus dans la planification du trafic.

## 5. Conception :

Cette section présente une vue d'ensemble du système multi-agent conçu pour optimiser la gestion du trafic aérien. Nous détaillons les différentes couches architecturales du système, les différents diagrammes, et la manière dont ces choix contribuent à atteindre les objectifs du projet.

### 1.5 Diagramme de Cas d'Utilisation :

Le diagramme de cas d'utilisation est un élément clé de la phase de conception, fournissant une représentation visuelle des interactions entre les acteurs (utilisateurs ou autres systèmes) et le système de contrôle aérien basé sur SMA. Le diagramme ci-joint illustre les principaux cas d'utilisation identifiés et la manière dont chaque acteur interagit avec le système.

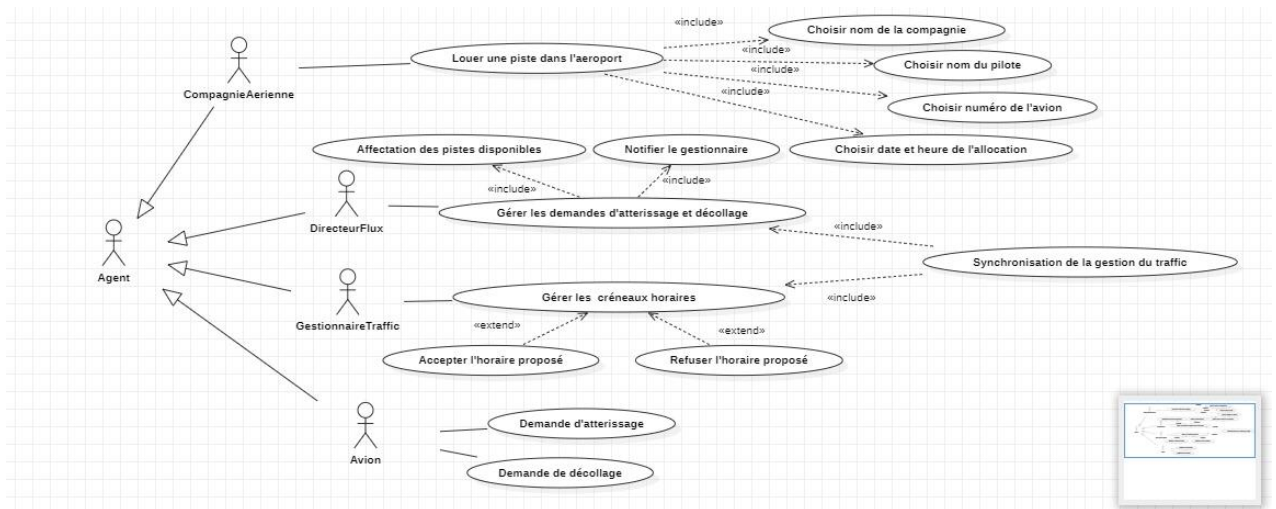


Figure 2: Diagramme de cas d'utilisation.

Les cas d'utilisation principaux sont décrits brièvement ci-dessous :

- **Location d'une piste dans l'aéroport** : Les compagnies aériennes peuvent louer des pistes disponibles, une fonctionnalité essentielle pour la planification des vols.
- **Gestion des demandes d'atterrissage et de décollage** : Le Directeur de Flux reçoit et traite ces demandes, en coordination avec le Gestionnaire de Trafic.
- **Gestion des créneaux horaires** : Un cas d'utilisation critique où le Gestionnaire de Trafic assigne des créneaux horaires pour les atterrissages et les décollages, optimisant ainsi la circulation des avions.
- **Synchronisation de la gestion du trafic** : Ce cas d'utilisation transverse assure que toutes les actions sont coordonnées de manière à garantir une gestion fluide du trafic aérien.

## 2.5 Diagramme de Classes :

Le diagramme de classes offre une représentation détaillée des entités du système multi-agent et de leurs interrelations. Chaque classe est dotée d'attributs spécifiques et de méthodes opérationnelles qui facilitent les interactions complexes au sein du système de gestion du trafic aérien.

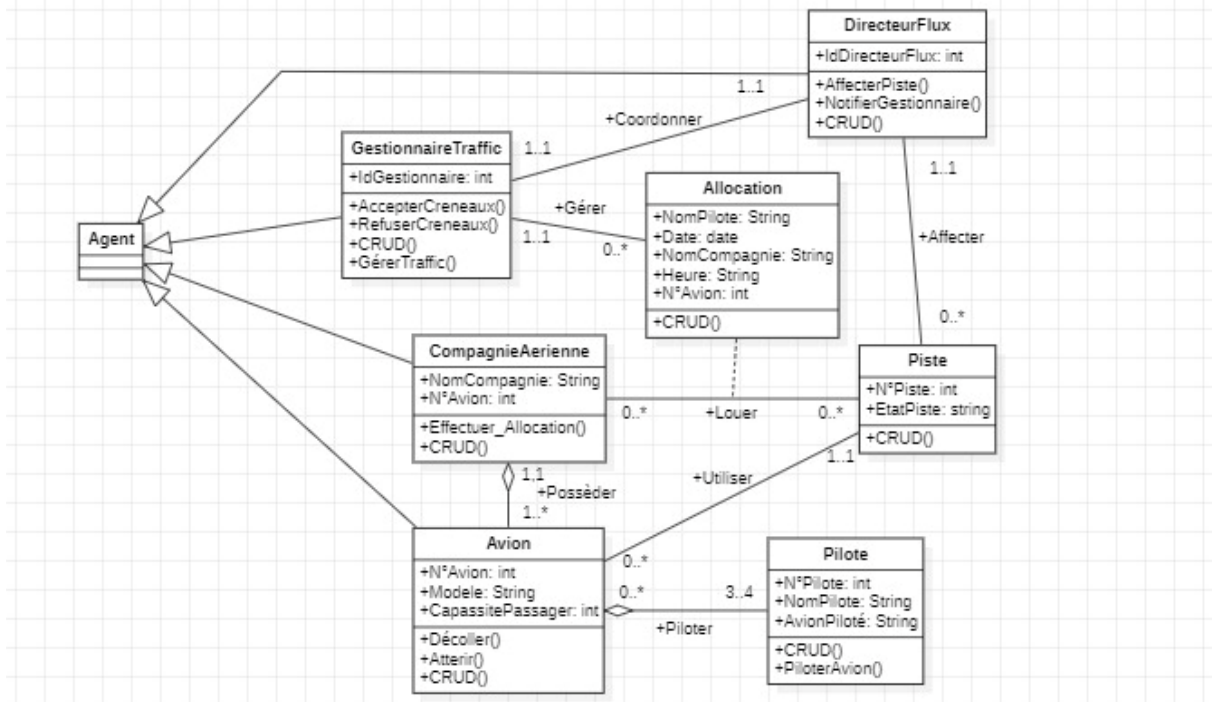


Figure 3: Diagramme de classes.

Les classes principales et leurs fonctions sont décrites ci-dessous :

- **Gestionnaire de Trafic** : Cette classe gère les créneaux horaires et coordonne globale du trafic aérien. Elle est responsable de l'acceptation et de la gestion des créneaux horaires, et de la coordination.
- **Compagnie Aérienne** : Elle détient des informations sur les vols et est responsable de l'exécution des opérations d'allocation. Elle communique avec les gestionnaires de trafic pour les réservations de créneaux et la coordination des vols.
- **Directeur de Flux** : Cette classe joue un rôle clé dans la gestion stratégique du trafic, assurant que les décisions prises par le gestionnaire de trafic sont bien coordonnées avec les exigences opérationnelles de l'aéroport.
- **Avion** : Représentant les aéronefs, cette classe inclut les paramètres essentiels comme le modèle, le nombre de passagers, et les fonctions pour décoller, atterrir, et effectuer les opérations CRU(D).

- **Pilote** : La classe Pilote interagit directement avec les Avions, gérant le pilotage et les opérations associées, tout en communiquant avec le Gestionnaire de Trafic et le Directeur de Flux.
- **Piste** : La classe Piste est utilisée pour gérer les attributs physiques des pistes d'atterrissage et de décollage, et est essentielle pour la logistique au sol de l'aéroport.
- **Allocation** : Cette classe est centrale dans le diagramme et sert de liaison entre les Compagnies Aériennes, les Pistes, et les Directeurs de Flux, gérant les détails des réservations comme les noms des pilotes, la date et l'heure, et les numéros de vol.

### 3.5 Les diagrammes de séquence :

Les diagrammes de séquence fournissent une vue dynamique des interactions entre les agents et les processus au sein du système multi-agent pour la gestion du trafic aérien. Ces diagrammes illustrent les étapes séquentielles qui interviennent dans des scénarios spécifiques, révélant les flux de travail et les décisions prises à chaque étape. Deux processus clés sont décrits :

#### 1. Diagramme de Séquence - Allocation Piste :

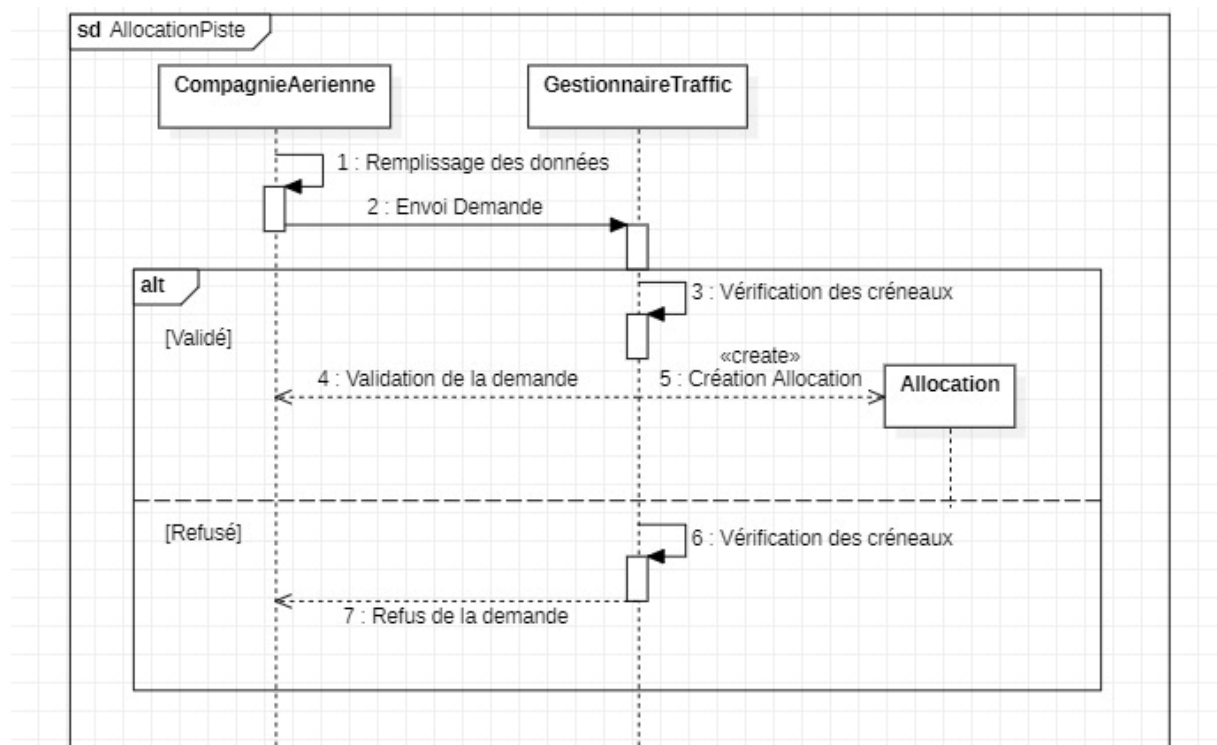


Figure 4: Diagramme de Séquence - Allocation Piste.

Ce diagramme de séquence montre les étapes suivies par la Compagnie Aérienne et le Gestionnaire de Trafic dans le processus d'allocation des pistes :

- La Compagnie Aérienne remplit les données nécessaires et envoie une demande d'allocation de piste au Gestionnaire de Trafic.
- Le Gestionnaire de Trafic vérifie la disponibilité des créneaux horaires et, en fonction de cette vérification, procède soit à la validation de la demande (créant ainsi une allocation) soit au refus de la demande si les conditions ne sont pas remplies.

## 2. Diagramme de Séquence - Demande Atterrissage et Décollage :

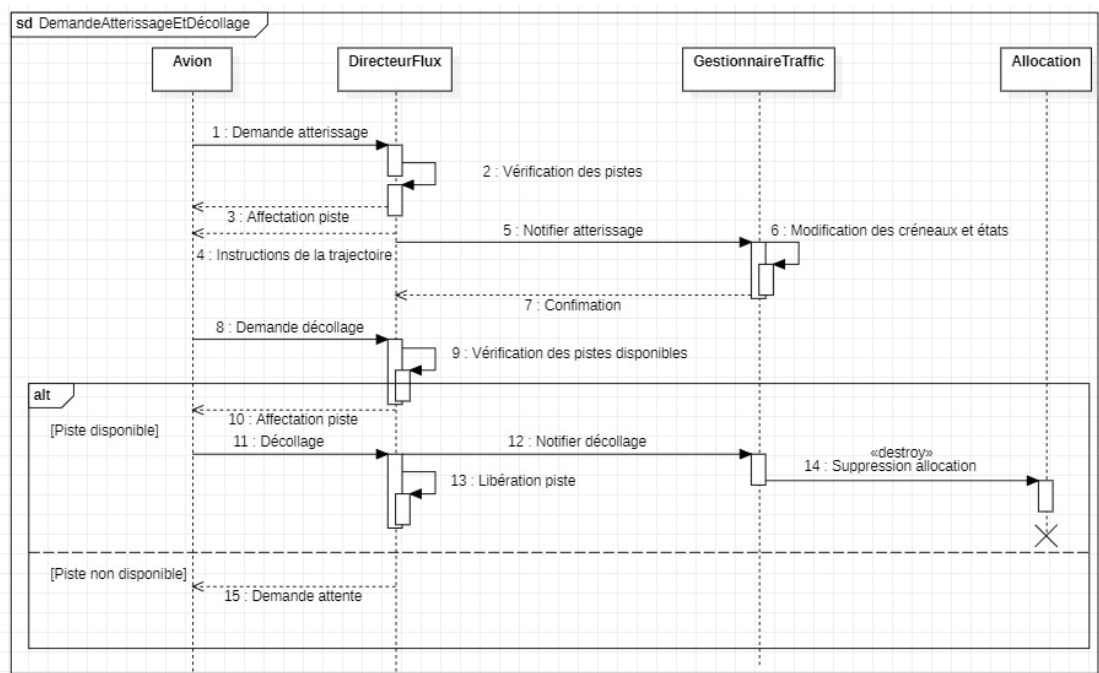


Figure 5: Diagramme de Séquence - Demande Atterrissage et Décollage.

Ce diagramme de séquence décrit les interactions pour les demandes d'atterrissage et de décollage entre l'Avion, le Directeur de Flux, et le Gestionnaire de Trafic :

- L'Avion envoie une demande d'atterrissage, qui est vérifiée par le Directeur de Flux. Une fois la piste affectée, l'Avion reçoit les instructions de trajectoire pour l'atterrissage.
- Suite à l'atterrissage, l'Avion fait une demande de décollage. Si une piste est disponible, elle lui est affectée et le décollage est exécuté, suivi par la libération de la piste et la suppression de l'allocation.
- Si aucune piste n'est disponible, l'Avion est mis en attente jusqu'à ce que la situation change.



## 4.5 Diagrammes d'État de Transition :

Les diagrammes d'état de transition illustrent les changements d'état d'un système ou d'un agent en réponse à des événements. Pour notre système de gestion du trafic aérien, ils décrivent les flux de travail et les conditions requises pour le passage d'un état à un autre. Voici une explication de chaque diagramme :

### 1) Diagrammes d'État de Transition - Allocation de Piste :

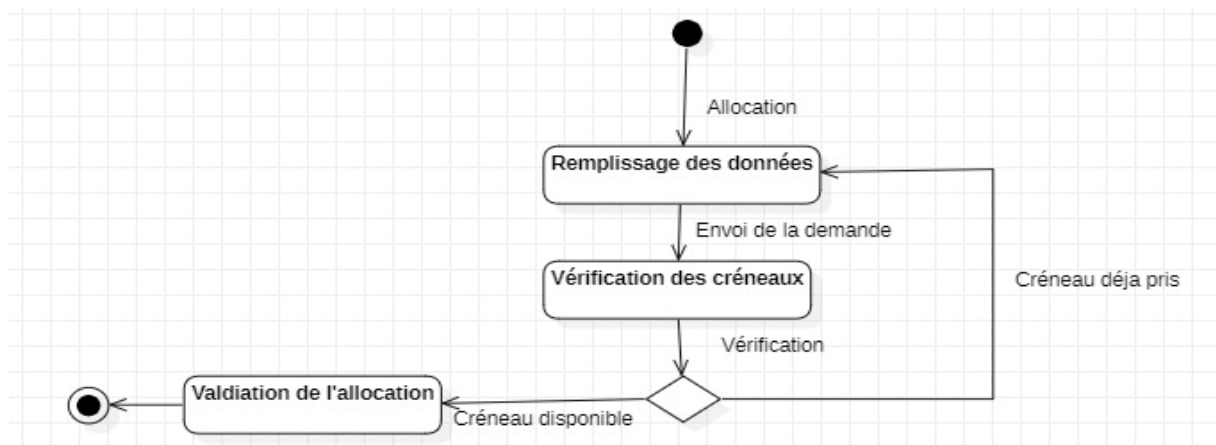


Figure 6: Diagrammes d'État de Transition - Allocation de Piste.

Ce diagramme montre le processus d'allocation des pistes à partir du moment où une compagnie aérienne remplit ses données et envoie une demande, jusqu'à la validation de l'allocation après la vérification des créneaux horaires disponibles.

### 2) Diagrammes d'État de Transition - État de l'Avion:

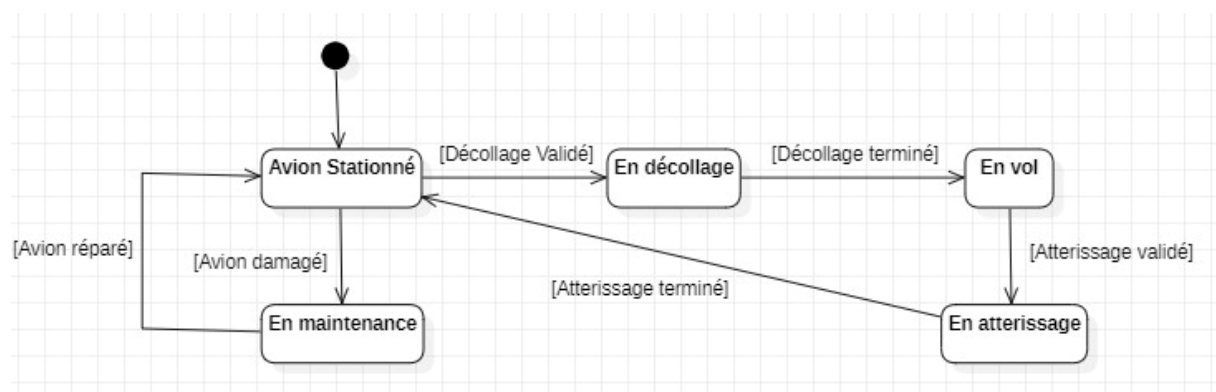


Figure 7: Diagrammes d'État de Transition - État de l'Avion.

Ce diagramme indique les différents états d'un avion, de l'état stationné jusqu'au vol, incluant les transitions vers l'état de maintenance si nécessaire et le retour en état stationné après atterrissage.

### 3) Diagrammes d'État de Transition - Demande d'Atterrissage:

Ici, les étapes sont présentées depuis la demande d'atterrissage par l'avion, en passant par la communication des instructions et la vérification des pistes, jusqu'à l'atterrissage réussi de l'avion.

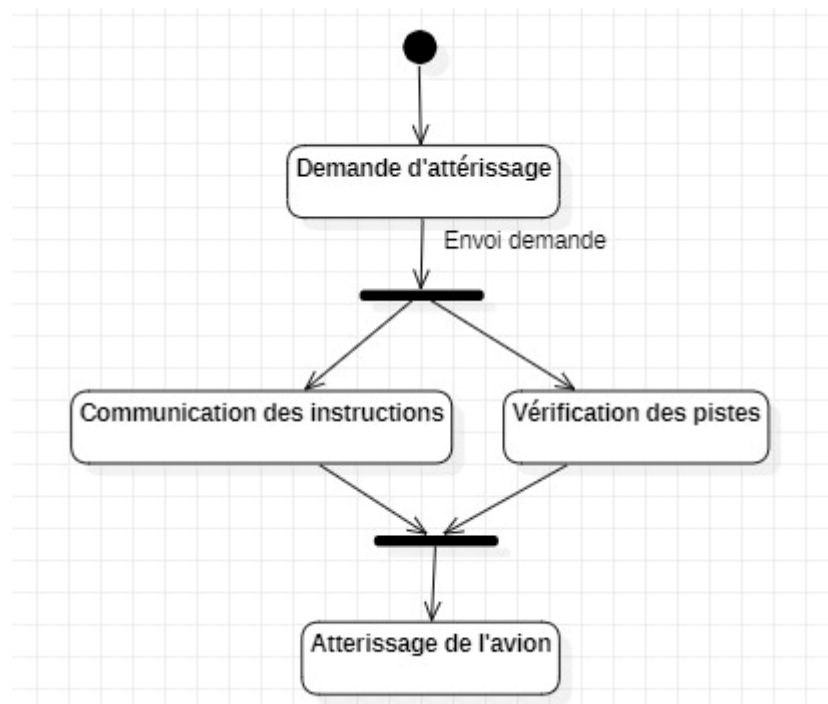


Figure 8: Diagrammes d'État de Transition - Demande d'Atterrissage.

### 4) Diagrammes d'État de Transition - Demande de Décollage :

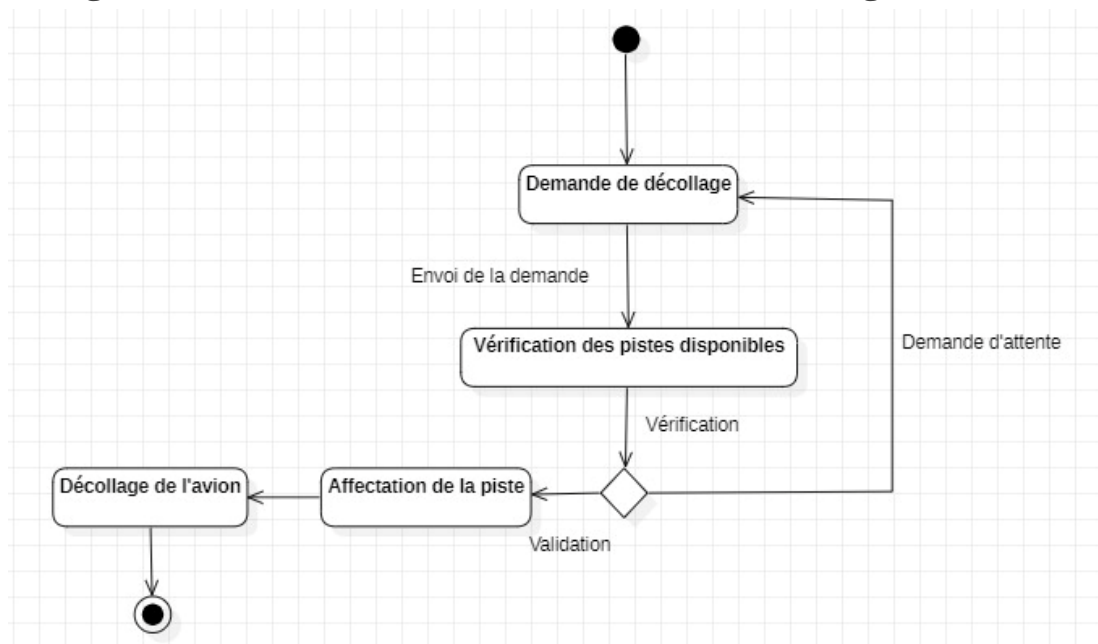


Figure 9: Diagrammes d'État de Transition - Demande de Décollage.

Ce diagramme détaille le processus de demande de décollage, où l'avion envoie sa demande, attend la vérification et l'affectation de la piste disponible, avant de procéder au décollage.



## 6. Réalisation du projet :

La réalisation d'un projet dans IntelliJ implique plusieurs étapes, de la conception initiale à la mise en œuvre et aux tests finaux. Cette section détaille le processus suivi pour développer et concrétiser le projet en utilisant les fonctionnalités et les outils offerts par l'environnement de développement intégré.

### 1.6 La structure de projet sur IntelliJ :

La structure de projet dans IntelliJ joue un rôle crucial dans l'exécution du code. Elle permet d'organiser les fichiers source, les ressources et les dépendances de manière logique et cohérente. Une structure bien définie facilite la navigation et la maintenance du code, tout en assurant une exécution efficace.

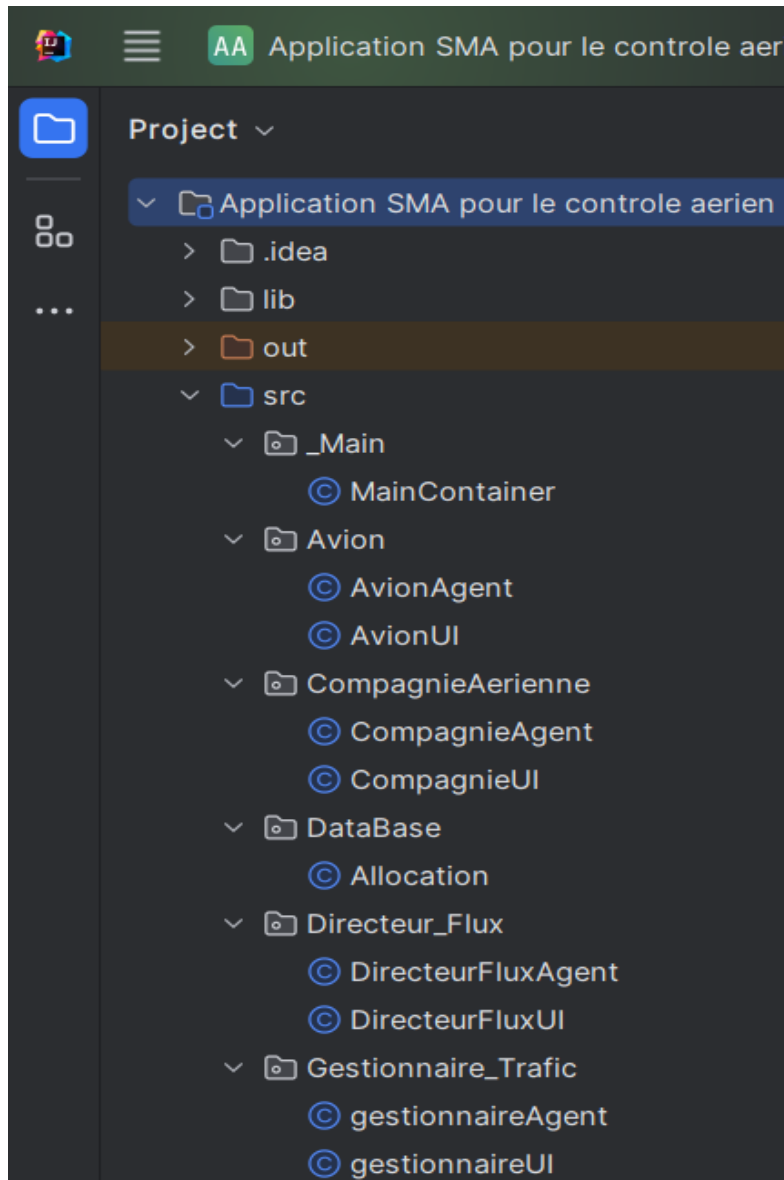


Figure 10: La structure de projet.

## 2.6 Exécution de projet :

Pour exécuter le projet, la première étape consiste à lancer le conteneur principal. Ce conteneur principal abrite trois agents principaux essentiels : DF (Directory Facilitator), responsable de la gestion des services disponibles, AMS (Agent Management System), chargé de la gestion des agents et de leur cycle de vie et RMA (Remote Monitoring Agent), fournissant des fonctionnalités de surveillance et de diagnostic.

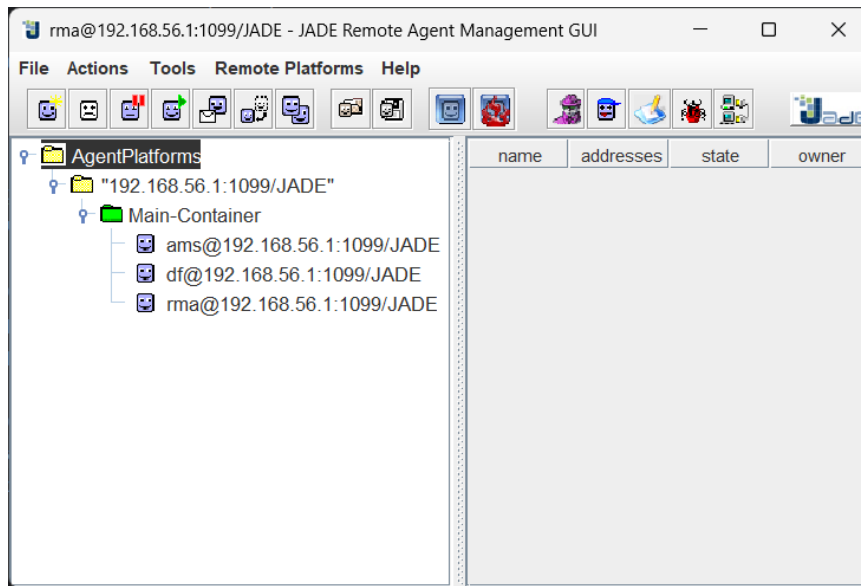


Figure 11: Lancement de Main-Container sur JADE.

après cela, nous commençons à exécuter les autres interfaces des agents une par une, Avion, Compagnie Aérien, Directeur Flux et Gestionnaire de Trafic :

- **Interface Avion :** Interagit avec le Directeur Flux pour obtenir des informations sur les pistes.

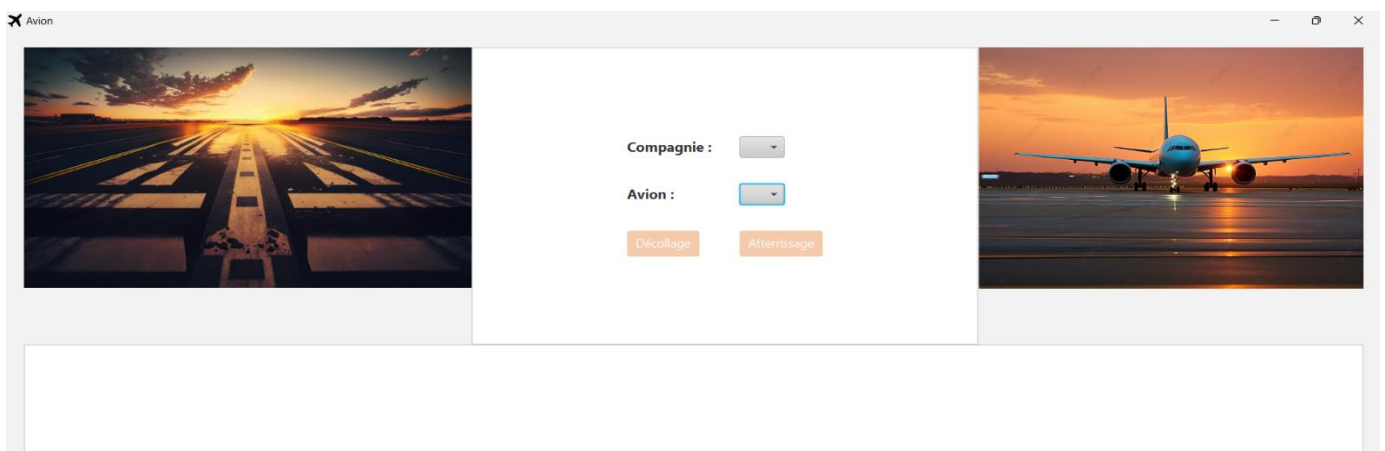


Figure 12: Interface d'agent avion.

- **Interface Compagnie Aérienne :** Demande des créneaux de vol pour ses avions, reçoit des confirmations de créneaux ou des alternatives en cas de conflit.

Figure 13: Interface Compagnie Aérienne.

- **Interface Directeur Flux :** Gère les demandes de pistes et coordonne les mouvements au atterrissage et les décollages.

| DirecteurFlux       |       |        |                  |                 |       |
|---------------------|-------|--------|------------------|-----------------|-------|
| Compagnie           | Avion | Pilote | Date Atterissage | Heure Atteri... | Piste |
| No content in table |       |        |                  |                 |       |

Figure 14: Interface Directeur Flux.

- **Interface Gestionnaire Trafic** : Supervise la distribution des créneaux de vol, surveille le flux de trafic, et ajuste les plans en fonction des besoins.



Figure 15: Interface Gestionnaire Trafic.

### 3.6 La base de données :

Dans notre système de gestion de trafic aérien, une base de données MySQL intitulée **Allocation** a été créée pour assurer un suivi rigoureux des réservations et des allocations de créneaux pour les avions. La table **allocation** au sein de cette base de données comprend plusieurs champs : `compagnie`, `avion`, `pilote`, `date`, `heure`, `minute`, `arrivé`, et `piste`, chacun étant conçu pour stocker des informations spécifiques. Ces champs permettent d'enregistrer avec précision les détails de chaque vol, y compris l'identité de la compagnie aérienne et du pilote, les horaires précis de départ, ainsi que l'état d'arrivée et la piste attribuée, facilitant ainsi la coordination et l'optimisation du flux de trafic aérien.

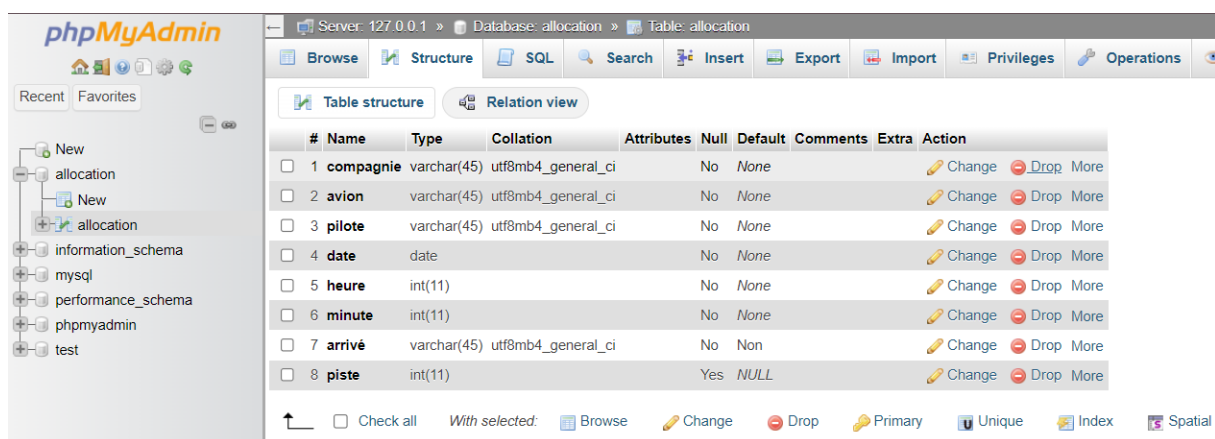


Figure 16: la base de données Allocation.

#### 4.6 Exemple d'exécution pratique :

Dans cette partie, nous illustrons le processus de réservation d'un créneau de vol au sein de notre système de gestion de trafic aérien, à travers un scénario typique impliquant notre base de données MySQL Allocation.

- **Réservation d'un Créneau :**

Pour effectuer une réservation, nous devons remplir les informations dans l'interface Compagnie Aérienne :

Nom Compagnie :

Avion :

Pilote :

Date :

Heure :

Minutes :

Singapore Airlines

Avion 552

AMINE

4/29/2024

15

20

Réserver

RAM: Je voudrais réserver une piste le 2024-04-22 à 11:45 s'il vous plaît !  
Gestionnaire : Votre réservation a bien été effectuée !

Qatar Airways: Je voudrais réserver une piste le 2024-04-23 à 12:30 s'il vous plaît !  
Gestionnaire : Votre réservation a bien été effectuée !

Air France: Je voudrais réserver une piste le 2024-04-24 à 10:15 s'il vous plaît !  
Gestionnaire : Votre réservation a bien été effectuée !

Singapore Airlines: Je voudrais réserver une piste le 2024-04-29 à 15:20 s'il vous plaît !  
Gestionnaire : Votre réservation a bien été effectuée !

Figure 17: Exemple de réservation sur l'interface avion.

Ensuite toutes les données de réservation apparaîtront dans l'interface du Gestionnaire si le temps est libre il confirmera la réservation par « a bien été effectué. » sinon il vous informera de choisir un autre horaire.

[illegible]

Figure 18: Exemple de réservation sur l'interface Gestionnaire.

- **Exemple de demande d'atterrissage :**

Afin de faire une demande d'atterrissage dans l'interface Avion choisissez la compagnie et l'avion d'après notre base de réservation et cliquez sur Atterrissage:

Figure 19: Exemple de demande d'atterrissage dans l'interface Avion.

Lorsqu'un avion arrive, il communique avec le Directeur de Flux pour solliciter une piste d'atterrissage. Le Directeur de Flux analyse la situation du trafic et assigne une piste disponible à l'avion:

| DirecteurFlux |           |        |                   |                    |       |
|---------------|-----------|--------|-------------------|--------------------|-------|
| Compagnie     | Avion ▼   | Pilote | Date Atterrissage | Heure Atterrissage | Piste |
| RAM           | Avion 111 | MAHRI  | 2024-04-22        | 11 : 45            | 1     |
|               |           |        |                   |                    |       |
|               |           |        |                   |                    |       |
|               |           |        |                   |                    |       |

Figure 20: assigner une piste disponible à l'avion.

Suite à cela, le Directeur de Flux informe le Gestionnaire de Trafic de l'arrivée de l'avion, ce qui entraîne la mise à jour du statut de l'avion dans la base de données de la table allocation, la colonne arrivée est alors marquée Oui, indiquant officiellement que l'avion est sur le tarmac :

| Gestionnaire_Trafic |           |         |            |       |        |        |
|---------------------|-----------|---------|------------|-------|--------|--------|
| Compagnie           | Avion     | Pilote  | Date       | Heure | Minute | Arrivé |
| RAM                 | Avion 111 | MAHRI   | 2024-04-22 | 11    | 45     | Oui    |
| Qatar Airways       | Avion 58  | SABBAHI | 2024-04-23 | 12    | 30     | Non    |
| Air France          | Avion 761 | AYMANE  | 2024-04-24 | 10    | 15     | Non    |
| Singapore Airlines  | Avion 552 | AMINE   | 2024-04-29 | 15    | 20     | Non    |

Figure 21: Mise à jour du statut de l'avion dans la base de données.

- **Exemple de demande de décollage :**

Pour demander un décollage, dans l'interface Avion choisissez la compagnie et l'avion concernée et cliquez sur Décollage :



Figure 22: Exemple de demande de décollage.

Après que l'avion ait demandé l'autorisation de décoller, le Directeur de Flux attribue à l'appareil une piste libres pour son envol.

| Communication avec Compagnie et DirecteurFlux                       |
|---|
| Gestionnaire : D'accord c'est noté !                                |
| DirecteurFlux : Avion 761 de Air France vient de décoller !         |
| Gestionnaire : D'accord c'est noté !                                |
| DirecteurFlux : Avion 552 de Singapore Airlines vient d'atterrir !  |
| Gestionnaire : D'accord c'est noté !                                |
| DirecteurFlux : Avion 552 de Singapore Airlines vient de décoller ! |
| Gestionnaire : D'accord c'est noté !                                |

Figure 23: Communication entre Gestionnaire et Directeur.

Une fois le décollage effectué, le Directeur de Flux met à jour le statut de ces zones comme étant disponibles. Il notifie ensuite le gestionnaire de trafic du départ de l'avion, ce qui conduit à la suppression des détails de cette réservation spécifique dans la base de données, libérant ainsi l'espace pour de futures affectations.

## 5.6 Règles et informations à respecter :

- Le temps d'inter arrivé entre les avions est fixé à 20 minutes.
- Aucun avion ne peut rester stationné à l'aéroport pendant plus d'une heure.
- Il y a quatre pistes disponibles pour les opérations de décollage et d'atterrissage.
- La capacité d'accueil permet à quatre avions d'atterrir simultanément, en raison des quatre pistes disponibles.
- La capacité d'accueil maximale par heure est de 12 avions.
- Sur une base quotidienne, l'aéroport peut accueillir jusqu'à 288 avions.



## 7. Outils Utilisés :

Pour le développement et l'implémentation de notre système de gestion du trafic aérien basé sur des agents, nous avons utilisé une gamme d'outils logiciels pour faciliter la conception, le développement et le déploiement. Voici une liste des outils principaux utilisés dans notre projet :

### 1.7 JADE (Java Agent Développment Framework) :

JADE est une plateforme de développement logiciel open source spécialement conçue pour la création d'applications multi-agents en Java. Elle fournit un ensemble d'outils et de bibliothèques qui simplifient la création, le déploiement et la gestion d'agents autonomes qui interagissent entre eux pour réaliser des tâches complexes.



Figure 24: Logo de JADE.

### 2.7 JavaFX :

JavaFX est une plateforme de développement logiciel qui permet de créer des applications client riches en Java. Elle offre une bibliothèque graphique et des outils pour concevoir des interfaces utilisateur graphiques (GUI) interactives et attrayantes. JavaFX est largement utilisé pour développer une variété d'applications, telles que des applications de bureau, des applications mobiles, des applications web et des applications embarquées.



Figure 26: Logo JavaFX.

### 3.7 IntelliJ IDEA :

IntelliJ IDEA est un environnement de développement intégré (IDE) conçu spécifiquement pour le langage de programmation Java. Il est développé par JetBrains et est largement reconnu comme l'un des IDE les plus puissants et les plus efficaces pour le développement Java. IntelliJ IDEA offre une gamme complète de fonctionnalités qui améliorent la productivité des développeurs, y compris un éditeur de code avancé avec une prise en charge de la saisie semi-automatique, une analyse intelligente du code, des outils de refactoring, une intégration avec les systèmes de contrôle de version tels que Git, et des capacités de débogage puissantes.

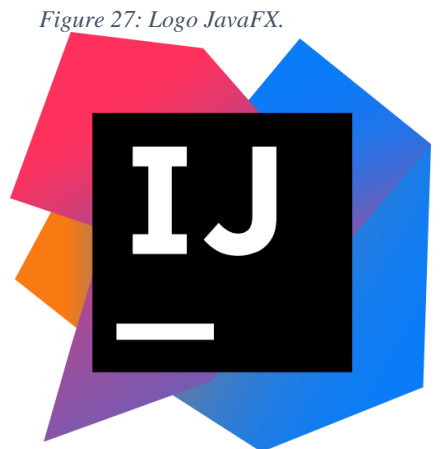


Figure 28: Logo de IntelliJ IDEA.



#### 4.7 StarUML :

StarUML est un logiciel de modélisation UML (Unified Modeling Language) largement utilisé dans le domaine du développement logiciel. Il permet aux développeurs de créer des diagrammes UML pour visualiser, concevoir et documenter les structures et les comportements des systèmes logiciels. StarUML offre une variété de fonctionnalités pour soutenir la modélisation des systèmes, y compris la création de diagrammes de classes, de diagrammes de séquence, de diagrammes de cas d'utilisation, de diagrammes d'activité, et bien plus encore.



Figure 30: Logo de StarUML.

#### 5.7 MySQL :

MySQL est un système de gestion de base de données relationnelle open source largement utilisé dans le développement de logiciels. Il offre un stockage structuré et fiable des données, avec la possibilité d'effectuer des opérations complexes de recherche, d'insertion, de mise à jour et de suppression. MySQL utilise le langage SQL (Structured Query Language) pour interagir avec les données.



Figure 31: Logo de MySQL.

#### 6.7 XAMPP :

XAMPP est un ensemble de logiciels open source qui facilite l'installation et la gestion d'un environnement de développement web. Nous avons utilisé XAMPP pour créer un environnement de développement local, hébergeant MySQL et fournissant un serveur Apache pour le développement de notre application.



Figure 32: Logo de XAMPP.

#### 7.7 GitHub :

GitHub est une plateforme de développement logiciel basée sur Git, largement utilisée pour le contrôle de version et la collaboration de projet. Nous avons utilisé GitHub pour héberger notre code source, gérer les versions et faciliter la collaboration entre les membres de l'équipe.



Figure 33: Logo de GitHub.

---

## 8. Conclusion :

Le projet d'application des Systèmes Multi-Agents (SMA) pour la gestion du trafic aérien a démontré une potentialité élevée pour révolutionner la gestion du trafic dans les aéroports en améliorant significativement la coordination et l'efficacité du contrôle aérien. Grâce à la mise en œuvre de SMA, nous avons pu simuler et optimiser les interactions entre les avions, les tours de contrôle, et d'autres entités, ce qui a permis une gestion plus dynamique et adaptative du trafic aérien.

Les résultats obtenus à partir de ce projet mettent en évidence plusieurs avantages clés des SMA, notamment la réduction des retards de vol, l'amélioration de la sécurité aérienne et une meilleure satisfaction des passagers et des compagnies aériennes. Cela a été possible grâce à une meilleure allocation des pistes et à une gestion optimisée des horaires de vol, rendues possibles par l'agilité des SMA dans la gestion des données et des ressources en temps réel.

Toutefois, il est important de noter que la réussite de l'implémentation de tels systèmes dans un environnement réel dépendra de plusieurs facteurs, tels que la collaboration entre les différents acteurs du secteur aérien, la standardisation des protocoles de communication entre les agents, et la sécurisation des systèmes contre les cyber-attaques, étant donné la nature critique des informations gérées.

En conclusion, bien que des défis subsistent, les SMA offrent une voie prometteuse pour répondre aux besoins croissants de gestion du trafic aérien. Des études supplémentaires et des phases de test plus approfondies seront nécessaires pour valider pleinement l'efficacité de ces systèmes et garantir leur intégration harmonieuse dans les infrastructures aériennes existantes. Avec les progrès continus dans le domaine de l'intelligence artificielle et des systèmes multi-agents, nous nous dirigeons vers une ère où la gestion du trafic aérien pourrait être transformée pour répondre aux demandes du 21<sup>e</sup> siècle.

---

## 9. Références :

- [1] : <https://www.youtube.com/watch?v=aDXg9ilOV84&t=3s>
- [2] : [https://www.youtube.com/watch?v=RPDG\\_Emzx8E&t=2401s](https://www.youtube.com/watch?v=RPDG_Emzx8E&t=2401s)
- [3] : [https://www.youtube.com/watch?v=N2\\_eXT-sBPk](https://www.youtube.com/watch?v=N2_eXT-sBPk)
- [4] : [https://www.youtube.com/watch?v=-OYQ0q\\_yDxc](https://www.youtube.com/watch?v=-OYQ0q_yDxc)
- [5] : [https://www.youtube.com/watch?v=vbn2ptW9E\\_g](https://www.youtube.com/watch?v=vbn2ptW9E_g)
- [6] : <https://www.ecologie.gouv.fr/gestion-des-flux-traffic>
- [7] : <https://www.onisep.fr/ressources/univers-metier/metiers/controleur-aerien-controleuse-aerienne>
- [8] : [github.com/samir-aitabbou/Application-SMA-pour-le-controle-aerien](https://github.com/samir-aitabbou/Application-SMA-pour-le-controle-aerien)
- [9] : <http://leea.recherche.enac.fr/documents/LenoirCongestionSlots.pdf>
- [10] : [https://fr.wikipedia.org/wiki/Gestion\\_du\\_traffic\\_a%C3%A9rien](https://fr.wikipedia.org/wiki/Gestion_du_traffic_a%C3%A9rien)