

# Cahier de charge AeroConnect

**Nom:** AeroConnect

**Classe:** Lim3A

**Membres:** Iyed Aloui / Wassim Kedidi / Elyes Charaa

## Nature du projet

Développement d'une **application web** de gestion d'aéroport permettant :

- aux **passagers** de consulter les vols et effectuer des réservations,
- aux **agents** de gérer les vols, les avions et les réservations,
- et d'afficher les **départs et arrivées** de manière claire et dynamique

## Objectif principal

Faciliter la **consultation et la gestion des vols aériens** à travers une application web moderne, responsive et sécurisée, basée sur une architecture **Spring Boot (backend)** et **Angular (frontend)** avec une base de données **SQL**.

## Description

L'application AeroConnect doit permettre :

- à un **visiteur** de consulter les vols et leurs horaires,
- à un **utilisateur connecté (passager)** de réserver un vol et consulter ses réservations,
- à un **administrateur/agent** de gérer les vols, les avions et les réservations.

# **Objectifs du projet**

- Créer une interface web fluide et intuitive (Angular).
- Développer un backend REST fiable et sécurisé (Spring Boot).
- Assurer une communication fluide entre le frontend et le backend via API JSON.
- Gérer toutes les données dans une base SQL (MySQL ou PostgreSQL).
- Respecter la logique d'authentification et de rôles.

# **Fonctionnalités principales**

## **Côté utilisateur :**

- Page d'accueil avec la liste des vols
- Barre de recherche (destination, date, compagnie, etc.)
- Détails d'un vol sélectionné
- Système de réservation simple (nom, prénom, nombre de places, etc.)
- Historique des réservations

## **Côté admin :**

- Tableau de bord
- Gestion des vols (CRUD complet)
- Gestion des avions
- Consultation des réservations
- Mise à jour du statut des vols

## **Côté public :**

- Tableau d'affichage dynamique des vols (arrivées / départs)

# Technologies

- **Frontend** : Angular
- **Backend** : Spring Boot (Java)
- **Base de données** : MySQL ou PostgreSQL
- **Communication** : API REST (JSON)

## Besoins fonctionnels

### 4.1. Rôles et accès

Rôle	Description	Accès
Visiteur	Utilisateur non connecté	Consulter les vols publics
Utilisateur (passager)	Utilisateur connecté	Réserver un vol, consulter ou annuler ses réservations
Administrateur / Agent	Personnel aéroport	Gérer les vols, les avions, les utilisateurs et les réservations

## Besoins non fonctionnels

Catégorie	Exigence
Performance	L'application doit charger les pages en moins de 3 secondes.
Sécurité	Authentification sécurisée (JWT), mots de passe hashés (BCrypt).
Ergonomie	Interface claire, responsive, compatible PC/tablette/mobile.
Fiabilité	Sauvegarde et intégrité des données assurées par le SGBD.
Maintenance	Code organisé (MVC + API REST), facilement extensible.
Disponibilité	L'application doit être accessible en local (localhost) et déployable sur un serveur.
Langue	Interface en français (possibilité d'évolution multilingue).
Interopérabilité	Communication standard JSON entre backend et frontend

## Livrables attendus

- Code source complet (frontend + backend)
- Script SQL de création de la base
- Documentation technique (API + structure)
- Manuel utilisateur (capture d'écran et guide)
- Présentation PowerPoint / démonstration