

# Projet : "Plateforme de Réservation de Tickets d'Événements"

## *Objectif*

Créer une application en Java pour gérer la réservation de tickets pour différents types d'événements (concerts, spectacles, conférences). Le projet vise à introduire les concepts fondamentaux de la programmation orientée objet (POO) tout en intégrant une interface graphique simple avec JavaFX.

## Fonctionnalités principales

### **Épic 1 : Gestion des utilisateurs**

- **Connexion utilisateur :**
  - Les utilisateurs peuvent se connecter avec un identifiant et un mot de passe.
  - Deux types de comptes :
    - **Client** : Peut réserver des tickets et consulter son historique.
    - **Organisateur** : Peut créer et gérer des événements.
  - Message d'erreur si les informations de connexion sont incorrectes.
- **Création de compte :**
  - Les utilisateurs peuvent créer un compte en fournissant nom, email, et type de compte.
  - Validation des champs avant enregistrement.

### **Épic 2 : Gestion des événements**

- **Création d'événements (organisateur) :**
  - Saisie des informations : nom, date, lieu, catégories de places (ex. : Standard, VIP), prix et nombre de places disponibles.
  - Les événements créés apparaissent dans une liste accessible aux clients.
- **Consultation des événements (client) :**
  - Liste des événements triée par date.
  - Filtres disponibles : type d'événement, lieu, artiste/intervenant.
- **Statistiques des événements (organisateur) :**
  - Total des tickets vendus.
  - Taux de remplissage par catégorie de place.
  - Chiffre d'affaires total généré.

### **Épic 3 : Gestion des réservations**

- **Réservation de tickets (client) :**

- Les clients peuvent réserver des tickets pour un événement et choisir une catégorie de place.
- Le système vérifie la disponibilité des places avant confirmation.
- **Gestion des erreurs lors de la réservation :**
  - Une exception `PlacesInsuffisantesException` est levée si le nombre de places demandé est supérieur au disponible.
  - Message d'erreur clair pour l'utilisateur.
- **Historique des réservations (client) :**
  - Les clients peuvent consulter leurs réservations passées avec les détails : nom de l'événement, date, nombre de tickets, total payé.
- **Annulation de réservation (client) :**
  - Les clients peuvent annuler une réservation.
  - Si l'annulation a lieu moins de 24 heures avant l'événement, une exception `AnnulationTardiveException` est levée.

## **Épic 4 : Paiement**

- **Saisie des informations de paiement :**
  - Les clients doivent fournir les détails de paiement (nom, numéro de carte).
  - Une exception `PaiementInvalideException` est levée si les informations sont incorrectes.
- **Confirmation de paiement :**
  - Un message de succès s'affiche avec les détails de la transaction.
  - Les tickets réservés sont enregistrés dans l'historique du client.

## **Épic 5 : Interface graphique (JavaFX)**

- **Ecran de connexion :**
  - Permet de se connecter ou de créer un compte.
- **Dashboard client :**
  - Liste des événements disponibles avec options de recherche et filtrage.
  - Formulaire de réservation avec choix de la catégorie de place et saisie du nombre de tickets.
  - Historique des réservations avec possibilité d'annuler.
- **Dashboard organisateur :**
  - Formulaire de création d'événement.
  - Consultation des statistiques pour chaque événement.
- **Notifications et messages :**
  - Les erreurs (ex. : paiement invalide, places insuffisantes) et succès (ex. : réservation validée) sont affichés via des boîtes de dialogue.

## Concepts techniques à intégrer

- **Héritage et classes abstraites :**
  - Classe abstraite Evenement et sous-classes Concert, Spectacle, et Conference.
  - Classe abstraite Utilisateur et sous-classes Client et Organisateur.
- **Interfaces :**
  - Interface Reservable pour gérer les réservations.
  - Interface Payable pour gérer les paiements.
- **Exceptions personnalisées :**
  - PlacesInsuffisantesException pour les réservations impossibles.
  - PaiementInvalideException pour les erreurs de paiement.
  - AnnulationTardiveException pour les annulations hors délai.
- **Polymorphisme :**
  - Manipulation des événements via des références de type Evenement.
  - Manipulation des utilisateurs via des références de type Utilisateur.
- **Collections :**
  - Utiliser des listes pour stocker les événements et réservations.
  - Trier et filtrer les événements selon des critères (date, type).

## Livrables attendus

1. **Code source :**
  - a. Bien commenté, structuré et conforme aux bonnes pratiques.
2. **Interface utilisateur :**
  - a. Fonctionnelle et intuitive via JavaFX.
3. **Document de présentation :**
  - a. Inclure un fichier README.md expliquant le fonctionnement de l'application.
4. **Démonstration :**
  - a. Présentation d'une application fonctionnelle, avec exemples de scénarios de réservation et de gestion.