

PROJET Épreuve E5 - Documentation réalisé par MELOTTI Simone

Cahier des Charges

Objectif du Projet :

L'objectif principal du projet A-4MENIC est de développer une application de gestion pour les salles de cinéma indépendantes en Île-de-France. Cette application doit permettre aux gérants de gérer les demandes de séances, l'aménagement des salles pour les événements, et de visualiser les données et statistiques liées au cinéma. L'application facilitera également la demande de subventions en offrant des fonctionnalités d'analyse et de reporting.

Fonctionnalités Demandées :

- Gestion des Comptes Utilisateurs :

Inscription et Connexion : Les administrateurs doivent pouvoir créer un compte, se connecter et gérer leurs informations personnelles.

Vérification de Compte : Les administrateurs doivent vérifier leur compte avant d'accéder aux services de gestion des séances et des salles.

- Gestion des Séances et des Salles :

Demande et Réservation de Séances : Les administrateurs doivent pouvoir planifier des séances de films.

Aménagement des Salles : Gestion de l'aménagement des salles pour différents types d'événements (projections, conférences, etc.).

Confirmation des Réservations : Les réservations de salles doivent être confirmées par les gérants.

- Connexion aux API :

API internationale de films (<https://www.themoviedb.org/>) : Intégration pour obtenir une liste des films disponibles et leurs détails.

API officielle de cinéma (<https://data.culture.gouv.fr/>) : Utilisation pour accéder aux archives des données concernant les cinémas par région.

- Visualisation des Données et Statistiques :

Rapports et Analyses : Génération de rapports détaillés et analyses statistiques pour faciliter la demande de subventions.

Tableaux de Bord : Tableau de bord interactif pour visualiser les données clés en temps réel.

- Base de Données :

Accessibilité et Sécurité : La base de données des comptes utilisateurs et des réservations doit être accessible en tout temps, avec des mesures de sécurité robustes pour protéger les données sensibles.

Performance Optimisée : Optimisation des requêtes pour assurer une performance optimale.

Réalisations :

Pour répondre aux besoins exprimés dans le cahier des charges, les réalisations suivantes ont été effectuées :

- *Développement de l'Interface Utilisateur :*

Création de l'application avec une mise en page claire et intuitive en JAVA, avec l'aide de JavaFX.

- *Intégration des API :*

Connexion à l'API officielle de cinéma et à l'API internationale de films pour récupérer et afficher les données des films et des cinémas.

- *Développement du Système de Gestion des Comptes :*

Création des formulaires d'inscription et de connexion. Mise en place de la vérification des comptes utilisateurs à la connexion grâce au code d'accès unique donnée dans l'invitation du département.

- *Mise en Place du Système de Réservation :*

Développement des fonctionnalités de recherche et de réservation de séances et de salles. Gestion des confirmations de réservation par les gérants.

- *Base de Données MySQL :*

Conception et implémentation de la base de données pour stocker les informations des utilisateurs, des cinémas, des séances, et des réservations.

Méthodes et Technologies Utilisées :

Modèle MVC (Model-View-Controller) : Sépare le projet en trois composants : le modèle gère la logique métier et les données, la vue s'occupe de l'affichage et l'interface utilisateur, et le contrôleur gère les interactions de l'utilisateur et met à jour le modèle et la vue. Cette architecture facilite la maintenance, la testabilité et la scalabilité des applications.

CRUD (Create, Read, Update, Delete) : La méthode CRUD en développement logiciel représente les quatre opérations de base pour la gestion des données : Create (créer), Read (lire), Update (mettre à jour), et Delete (supprimer). Ces opérations permettent de manipuler les enregistrements dans une base de données de manière complète et structurée.

Langages et Outils : Java, JavaFX, MySQL, Visual Studio Code, GitHub.

Tests : Les comportements de l'API ont été testés à l'aide de tests unitaires et de tests d'intégration manuels via JUnit. Ces tests ont permis de garantir que chaque fonctionnalité fonctionne correctement de manière isolée et intégrée.