

# Cahier de charge :

## Cahier des Charges pour un Site Web de Booking de Films en Ligne

### 1. Introduction

Le présent cahier des charges décrit les spécifications fonctionnelles, techniques et organisationnelles pour la création d'un site web permettant la réservation de places pour des projections de films en ligne. Ce document a pour objectif de servir de référence pour l'équipe de développement.

### 2. Objectifs du Projet

- Fournir une plateforme intuitive pour les utilisateurs afin de consulter les films à l'affiche, les horaires des séances et réserver des billets.
- Proposer une expérience utilisateur agréable et adaptée à tous les supports (ordinateurs, tablettes, smartphones).

### 3. Fonctionnalités Principales

#### 3.1. Fonctionnalités Utilisateur

- **Inscription et Connexion** : Possibilité de créer un compte utilisateur avec email, mot de passe, et profils personnalisés.
- **Consultation des Films** : Affichage des films disponibles avec leurs détails (titre, synopsis, bande-annonce, acteurs, réalisateur, etc.).
- **Réservation de Billets** : Choix du film, de la salle, de la date et de l'horaire, ainsi que du nombre de places.
- **Paiement en Ligne** : Intégration d'une solution de paiement sécurisée (ex : Stripe, PayPal).
- **Historique des Réservations** : Consultation des billets réservés et possibilité d'annulation.

- **Support Client** : Formulaire de contact ou chat pour demander de l'aide.

## 4. Spécifications Techniques

### 4.1. Architecture

- **Technologie Backend** : .NET MVC avec C#.
- **Base de Données** : SQL Server pour la gestion des données (films, utilisateurs, réservations).
- **Frontend** : HTML5, CSS3, JavaScript avec Bootstrap ou un framework comme React.js pour une meilleure interactivité.
- **API** : Création d'API REST pour la gestion des réservations et des interactions frontend-backend.

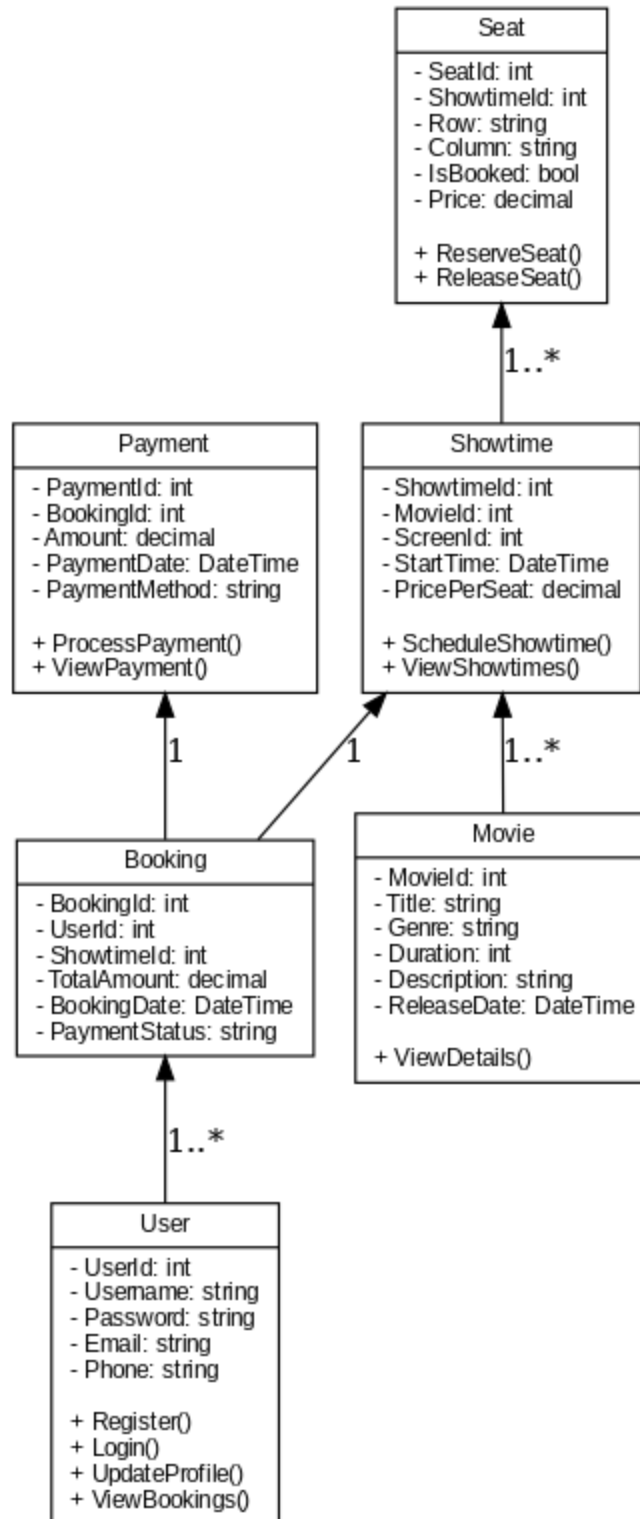
### 4.2. Sécurité

- **Authentification** : Implémentation de l'authentification via ASP.NET Identity.
- **Protection des Données** : Utilisation de protocoles HTTPS pour sécuriser les échanges.

---

## 8. Conception et Diagramme de Classes

Voici le diagramme de classes représentant les principales entités du système et leurs relations :



## 1. User (Utilisateur)

Le rôle de cette classe est de gérer les interactions liées aux utilisateurs.

- **Scénario** :Un utilisateur s'inscrit sur la plateforme en utilisant la méthode `Register()` . Après avoir créé un compte, il peut se connecter via `Login()` . Une fois connecté, il peut mettre à jour ses informations de profil ( `UpdateProfile()` ) et consulter son historique de réservations grâce à `ViewBookings()` .
- 

## 2. Booking (Réservation)

Cette classe gère les réservations des utilisateurs.

- **Scénario** :L'utilisateur sélectionne un film et un horaire pour réserver. Une nouvelle entrée est créée dans la classe `Booking` avec les détails de l'utilisateur, de la séance et le montant total à payer. Le système utilise les données comme le statut de paiement ( `PaymentStatus` ) et la date de réservation ( `BookingDate` ) pour suivre les réservations.
- 

## 3. Payment (Paiement)

Cette classe assure le traitement des paiements.

- **Scénario** :Après avoir effectué une réservation, l'utilisateur effectue le paiement via `ProcessPayment()` . Le système enregistre les informations, notamment le montant payé, la méthode de paiement ( `PaymentMethod` ) et la date ( `PaymentDate` ). L'utilisateur peut également consulter les détails de ses paiements via `ViewPayment()` .
- 

## 4. Movie (Film)

Cette classe représente les films disponibles sur la plateforme.

- **Scénario** :Les administrateurs ajoutent des films avec leurs informations (titre, genre, durée, description, etc.). Les utilisateurs peuvent consulter ces informations via `ViewDetails()` pour décider quel film réserver.
- 

## 5. Showtime (Séance)

Cette classe gère les horaires de projection des films.

- **Scénario** :L'administrateur planifie les séances pour un film spécifique en utilisant `ScheduleShowtime()` . Les utilisateurs peuvent consulter les horaires

disponibles via `ViewShowtimes()` et choisir une séance à réserver.

---

## 6. Seat (Siège)

Cette classe gère la sélection et la disponibilité des sièges dans une salle.

- **Scénario** :Lorsqu'un utilisateur réserve une place pour une séance, le système vérifie les sièges disponibles dans la salle via `IsBooked` . Une fois confirmé, le système réserve le siège via `ReserveSeat()` . Si la réservation est annulée, le siège est libéré avec `ReleaseSeat()` .
- 

### Exemple Pratique :

1. **Inscription et connexion** : L'utilisateur utilise `User.Register()` puis `User.Login()` pour accéder à la plateforme.
  2. **Sélection du film** : Il consulte les films disponibles via `Movie.ViewDetails()` et sélectionne un horaire avec `Showtime.ViewShowtimes()` .
  3. **Réservation** : Il choisit des sièges via `Seat.ReserveSeat()` et confirme la réservation en créant une entrée `Booking` .
  4. **Paielement** : Il paie le montant dû via `Payment.ProcessPayment()` pour finaliser sa réservation.
- 

Ce scénario montre comment les classes interagissent pour offrir une expérience utilisateur complète.