



**Ecole Leader**

RÉPUBLIQUE DU BÉNIN

\*\*\*\*



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
(MESRS)

\*\*\*\*

DIRECTION DES ÉTABLISSEMENTS PRIVÉS D'ENSEIGNEMENT SUPÉRIEUR (DGES)

\*\*\*\*

HAUTE ECOLE DE COMMERCE ET DE MANAGEMENT (HECM)

\*\*\*\*

MÉMOIRE DE FIN DE FORMATION POUR L'OBTENTION DU DIPLÔME DE LICENCE  
PROFESSIONNELLE

\*\*\*\*

**Filière :** *Système Informatique et Logiciel (SIL)*

**Option :** *Sciences et Techniques*

## ***THÈME***

*Conception et développement d'une application mobile de  
gestion des tickets d'événements culturels et de  
divertissements*

***Réalisé et présenté par :***

***Mlle M. Josiane E. GANSA***

***&***

***Mlle Y. Irfane O. RAFIOU***

***Sous la supervision de :***

***Tuteur de stage***

***Directeur de mémoire***

***M. Samuel DJIDOHOKPIN***

***M. Cyriaque NASCIMENTO***

***(Développeur Full Stack)***

***(Enseignant à HECM)***

***Mai 2025***

## **ENGAGEMENT**

La Haute Ecole de Commerce et de Management (HECM) n'entend donner ni approbation, ni improbation aux opinions émises dans ce mémoire. Ces opinions doivent être considérées comme propres à leurs auteurs.

## **DEDICACE**

à :

- ★ mon père, Christophe GANSA ;
- ★ ma mère, Rolande ADJALLO.

**Josiane Emmanuella GANSA**

## **DEDICACE**

à :

- ★ mon père, Mohamed RAFIOU ;
- ★ ma mère, Nadia FAGBEMY.

**Youssoufou Irfane Omo-Labakè RAFIOU**

## REMERCIEMENTS

Ce travail est le résultat d'un engagement collectif, et nous tenons à exprimer notre profonde reconnaissance à toutes les personnes qui y ont contribué. Nous adressons nos remerciements tout spécialement à :

- l'Honorable Natondé AKE, pour avoir mis à notre disposition le cadre de formation ;
- Nadine AKOGBETO, Directrice des études de la **Haute École de Commerce et de Management (HECM)**, pour sa simplicité et son dévouement, ainsi que pour ses nombreuses interventions en notre faveur, garantissant ainsi un cadre d'apprentissage optimal ;
- notre directeur de mémoire, M. Cyriaque NASCIMENTO, pour son enseignement, ses conseils et l'intérêt qu'il nous porte ;
- l'ensemble des **enseignants** de la Haute École de Commerce et de Management, et plus particulièrement à ceux de la filière **Systeme Informatique et Logiciel (SIL)**, pour la qualité de l'enseignement dispensé et leur engagement dans notre formation ;
- monsieur Nicolas SACKREUTER, Directeur du Service Informatique d' EREVAN BÉNIN SA pour sa disponibilité ;
- notre tuteur de stage, M. Samuel DJIDOHOKPIN pour sa disponibilité ;
- tout le personnel de EREVAN BÉNIN SA ;
- nos parents respectifs pour leurs soutiens moraux et financiers de tous les jours ;
- les honorables membres du jury qui ont bien voulu accepter d'apprécier notre travail ;
- tout ceux qui, d'une manière ou d'une autre, ont contribué à la réalisation de ce travail.

## **LISTE DES SIGLES, ACRONYMES ET ABRÉVIATIONS**

API : Application Programming Interface

BPEL : Business Process Execution Language

EMT : Erevan Mobile Tools

GLPI : Gestionnaire Libre de Parc Informatique

HECM : Haute Ecole de Commerce et de Management

HTTP : HyperText Transfer Protocol

MERISE : Méthode d'Etude et de Réalisation Informatique pour les Schémas d'Entreprise

MySQL : My Structured Query Language

OMT : Object Modeling Technique

OOSE : Object-Oriented Software Engineering

PHP : HyperText Preprocessor

RESTful : Representational State Transfer

SOBEPEC : Société Béninoise de Peinture et Colorant

SODAPEC : Société Dahoméenne de Peinture et Colorant

SGBD : Système de Gestion de Base de Données

UML : Unified Modeling Language

VIP : Very Important Person

VVIP : Very Very Important Person

WAMP : Windows, Apache, MySQL, PHP

## LISTE DES TABLEAUX

Tableau 1 : Identification des classes et attribut .....	18
Tableau 2 : Dictionnaire des données .....	21
Tableau 3 : Scénario nominal du cas « Inscription utilisateur » .....	27
Tableau 4 : Scénario nominal du cas « Payer un ticket » .....	31
Tableau 5 : Matériels utilisés .....	38

## LISTE DES FIGURES

Figure 1 : Situation géographique de Erevan .....	4
Figure 2 : Organigramme de Erevan .....	6
Figure 3 : Site officiel de Canal Olympia (Page A L’AFFICHE) .....	10
Figure 4 : Site officiel de Canal Olympia (Page ESCAPE GAME) .....	10
Figure 5 : Site officiel de Canal Olympia (Page ESCAPE PRO) .....	11
Figure 6 : Diagramme de cas d’utilisation .....	17
Figure 7 : Diagramme de classe .....	25
Figure 8 : Dessin de l’interface d’inscription .....	28
Figure 9 : Diagramme de séquence du cas « S’inscrire » .....	29
Figure 10 : Dessin de l’interface de réservation de ticket .....	32
Figure 11 : Diagramme de séquence du cas « Payer un ticket » .....	33
Figure 12 : Diagramme d’activité du cas d’utilisation « S’inscrire » .....	34
Figure 13 : Diagramme d’activité du cas d’utilisation « Payer un ticket » .....	35
Figure 14 : Extrait du code Backend (constructeur API) pour la connexion d’un utilisateur .....	41
Figure 15 : Extrait du code Frontend pour l’inscription utilisateur .....	42
Figure 16 : Interface de bienvenue .....	42
Figure 17 : Interface d’inscription (client) .....	43
Figure 18 : Interface d’inscription (responsable centre) .....	43
Figure 19 : Interface d'accueil .....	44

## RÉSUMÉ

Ce mémoire porte sur la conception et le développement de **Tickngo**, une application mobile dédiée à la gestion des tickets pour les événements culturels et les divertissements, avec un focus particulier sur **Canal Olympia**. L'objectif principal de ce projet est de moderniser l'accès aux événements en remplaçant les méthodes traditionnelles de vente de tickets physiques par une solution numérique, accessible via smartphone. **Tickngo**, développée à l'aide du framework **Flutter** et de l'environnement **Android Studio**, permet aux utilisateurs d'acheter et de réserver leurs tickets en ligne, offrant ainsi une expérience plus fluide et pratique. Côté serveur, l'application repose sur un back-end conçu avec **Laravel** et une base de données My Structured Query Language (**MySQL**), interconnectés grâce à une architecture Application Programming Interface (**API RESTful**). La modélisation et la conception du système ont été réalisées selon les standards de la méthode Unified Modeling Language (**UML**), afin d'assurer une structure claire et cohérente. L'application simplifie également la gestion des tickets pour les organisateurs, leur permettant de suivre les ventes et les flux de spectateurs en temps réel. Ce projet vise ainsi à améliorer l'expérience des spectateurs tout en fournissant aux organisateurs un outil efficace et moderne pour la gestion d'événements.

**Mots clés** : Tickngo, Application mobile, Tickets, Canal Olympia.

## ABSTRACT

This thesis focuses on the design and development of **Tickngo**, a mobile application dedicated to managing tickets for cultural events and entertainment, with a particular focus on **Canal Olympia**. The main objective of this project is to modernize access to events by replacing traditional physical ticket sales methods with a digital solution accessible via smartphone. **Tickngo**, developed using the **Flutter** framework and **Android Studio** environment, allows users to purchase and reserve their tickets online, offering a smoother and more convenient experience. On the server side, the application relies on a back-end designed with **Laravel** and a **MySQL** database, interconnected through a **RESTful API** architecture. The system modeling and design were carried out according to **Unified Modeling Language (UML)** standards to ensure a clear and coherent structure. The application also simplifies ticket management for organizers, allowing them to track sales and audience flow in real-time. This project aims to improve the spectator experience while providing organizers with an efficient and modern tool for event management.

**Keywords**: Tickngo, Mobile Application, Tickets, Canal Olympia.

# SOMMAIRE

<b>Introduction .....</b>	<b>1</b>
<b>Chapitre 1 : Cadre institutionnel de l'étude .....</b>	<b>3</b>
Section 1.1 : Présentation de la structure d'accueil .....	4
Section 1.2 : Déroulement de stage .....	7
Section 1.3 : Présentation du projet d'étude .....	8
<b>Chapitre 2 : Analyse et conception du système à mettre en place .....</b>	<b>13</b>
Section 2.1 : Analyse fonctionnelle .....	14
Section 2.2 : Analyse statique .....	18
Section 2.3 : Analyse dynamique .....	26
<b>Chapitre 3 : Réalisation du système .....</b>	<b>36</b>
Section 3.1 : Présentation des outils et technologies .....	37
Section 3.2 : Présentation du résultat .....	41
Section 3.3 : Mesure de sécurité et confidentialité .....	45
<b>Conclusion .....</b>	<b>47</b>

# **INTRODUCTION**

À l'ère où le numérique redessine les contours de notre quotidien, la gestion des événements connaît une transformation majeure grâce aux technologies mobiles. L'achat de billets, la gestion des accès et l'expérience utilisateur globale sont aujourd'hui fortement influencés par les applications mobiles, qui offrent des solutions pratiques et efficaces aussi bien aux organisateurs qu'aux spectateurs.

Ce mémoire s'inscrit dans cette dynamique en proposant la conception et le développement d'une application mobile dédiée à la gestion des tickets d'événements culturels, en prenant comme cas d'étude Canal Olympia. Ce choix repose sur le fait que cette chaîne de salles de spectacle, de jeux et de cinéma, présente dans plusieurs pays africains, notamment le Bénin, accueille régulièrement des événements culturels et artistiques de grande envergure.

L'objectif principal de cette application est de faciliter l'achat et la validation des billets, tout en optimisant la fluidité des entrées et en réduisant les fraudes. Pour cela, nous avons adopté une approche alliant analyse des besoins, conception ergonomique et mise en œuvre technique, en nous appuyant sur des technologies adaptées aux usages mobiles.

Dans ce travail, nous commencerons par une revue des systèmes existants de billetterie numérique et des défis associés. Ensuite, nous détaillerons le processus de conception et de développement de notre application, en expliquant les choix technologiques et les fonctionnalités mises en place. Enfin, nous présenterons les résultats obtenus et discuterons des perspectives d'amélioration possibles.

Ainsi, cette étude ambitionne non seulement d'apporter une solution innovante pour la gestion des tickets d'événements culturels, mais aussi de contribuer à l'amélioration de l'expérience utilisateur et de l'organisation des spectacles, des jeux et projections de Canal Olympia.

# **Chapitre 1 : Cadre institutionnel de l'étude**

## Section 1.1 : Présentation de la structure d'accueil

### 1.1.1 Historique

Le 25 septembre 2009 a été inauguré avec faste l'un des plus grands centres commerciaux d'Afrique Occidentale et porte le nom EREVAN (Nom de la capitale d'Arménie). Un centre commercial de plus de 13000 mètre carré pour un investissement de plusieurs milliards de francs CFA.

Le président Directeur Général, Monsieur Marcel TCHIFTEYAN, né à Marseille en 1931 et arrivé au Bénin dès 1953, après son Brevet élémentaire. Il a exercé, depuis lors, de nombreuses activités sur le sol béninois. Tout d'abord employé de diverses entreprises, il s'est ensuite illustré en créant notamment la société Matériaux Dahomey en 1958, qui deviendra quelques années plus tard Matériaux Bénin-SA puis la Société Dahoméenne de Peinture et Colorant (SODAPEC) en 1971, qui est devenu par la suite SOBEPEC (Société Béninoise de Peinture et de Colorants).

Premier opérateur économique à doter le Bénin d'une grande surface, (le supermarché EREVAN), Marcel TCHIFTEYAN a également joué un rôle prépondérant dans la création, à Ouidah, d'un complexe hôtelier de 60 chambres face à la mer : Casa del Papa par son fils aîné, M. Jean Luc TCHIFTEYAN actuel Directeur Général d'EREVAN-BENIN SA.

Les autorités béninoises ont reconnu que l'entrepreneur français avait remarquablement apporté sa pierre au développement socio-économique et touristique du pays, ce qui lui a valu de devenir Commandeur de l'Ordre National du Bénin.

### 1.1.2. Situation géographique



Figure 1 : Situation géographique de Erevan

### 1.1.3. Services

EREVAN BENIN SA, conçu comme l'un des plus grands centres commerciaux d'Afrique de l'Ouest, offre une infrastructure moderne en bord de mer avec un hypermarché de 4000 m<sup>2</sup>, un snackbar (Malongo), et 18 boutiques incluant des enseignes renommées. Associé à l'enseigne française U via une franchise, EREVAN a également ouvert quatre nouveaux magasins : U Express à Ganhi, U café à Ganhi, Supermarché U à Calavi et Supermarché U à Akpakpa. Le centre est facilement accessible, dispose d'un parking sécurisé pour 200 véhicules et propose plus de 40 000 références de produits variés : alimentation, textile, électroménager, cosmétique, multimédia, etc. Un personnel qualifié accueille les clients tous les jours avec des horaires adaptés.

### 1.1.4. Projets et Objectifs

Son objectif principal est d'atteindre l'excellence en matière de satisfaction clientèle, en plaçant les besoins et les attentes de ses consommateurs au cœur de sa stratégie commerciale. Dans cette optique, son ambitieux projet d'expansion vise à implanter progressivement des centres commerciaux EREVAN dans chacun des douze départements du Bénin, offrant ainsi une couverture nationale complète.



## Section 1.2 : Déroulement de stage

Tout au long de ses trois (03) mois de stage à Erevan Benin, nous avons mené plusieurs activités dans un climat de travail assez convivial, ce qui nous a permis d'acquérir de nouvelles connaissances pratiques en développement mobile. Cette section mettra en lumière les différentes activités menées, les compétences développées ainsi que les défis rencontrés au cours de cette expérience.

### 1.2.1. Activités menées

Dans le cadre de notre mémoire de fin de parcours en licence professionnelle, nous avons débuté notre stage à **EREVAN BÉNIN** en mars 2025, au sein du service informatique, où nous avons réalisé plusieurs tâches comme :

- apprendre à utiliser Flutter et laravel grâce à l'aide de notre encadreur de stage et des cours sur Internet ;
- créer des bases de données avec mysql ;
- faire des applications mobiles ;
- mettre en place le module "Stock Dormant" de l'application mobile EMT (Erevan Mobile Tools) ;
- Générer des tickets GLPI en guise de test du nouveau système de gestion libre du parc informatique de **EREVAN BÉNIN**;

### 1.2.2. Compétences développées

Au cours de notre stage à **EREVAN BÉNIN SA** nous avons eu à acquérir de nouvelles compétences et aussi à approfondir nos connaissances antérieures. Nous avons appris à gérer notre temps, à organiser notre travail afin d'être plus productifs. Ce stage nous a également permis d'avoir une idée de la vie professionnelle. Nous avons pu acquérir une meilleure compréhension des processus de développement, notamment le cycle de vie des applications et le déploiement des solutions informatiques, de résolution des problèmes techniques en temps réel. Ce stage nous a permis de perfectionner notre approche des projets informatiques et de mieux appréhender les défis techniques rencontrés dans un environnement professionnel.

## Section 1.3 : Présentation du projet d'étude

### 1.3.1. Problématique et présentation du thème

L'accès aux événements et aux instants de divertissement repose encore largement sur des méthodes traditionnelles qui, bien que fonctionnelles, peuvent présenter certaines limites. L'attente prolongée, l'incertitude quant à la disponibilité des accès, ainsi que l'absence de solutions de réservation à l'avance compliquent l'organisation aussi bien pour les participants que pour les responsables.

Dans de nombreux cas, la gestion des tickets et des accès impliquent des intermédiaires ou des ventes physiques, ce qui peut entraîner des risques d'erreurs, de pertes ou encore un manque de contrôle sur les flux d'entrées. Cette situation peut engendrer des désagréments pour le public et nuire à l'efficacité organisationnelle.

Face à ces constats, l'adoption d'une application mobile dédiée apparaît comme une réponse pertinente pour moderniser et fluidifier l'ensemble du processus. Une solution mobile offrirait plus de flexibilité, de sécurité et de confort, tant pour les organisateurs que pour les spectateurs.

### 1.3.2. Les objectifs

#### 3.2.1. Objectif principal

Notre objectif principal est de développer une solution mobile et intuitive visant à révolutionner la gestion des accès aux événements, tout en optimisant l'expérience utilisateur et la performance opérationnelle des organisateurs.

#### 1.3.2.2. Objectifs spécifiques

Pour atteindre cet objectif principal et offrir une solution véritablement transformative dans le domaine de la billetterie événementielle, notre projet s'articule autour des objectifs spécifiques suivants :

- **Concevoir une plateforme d'acquisition digitale** : Créer une application mobile permettant l'achat et la réservation de tickets avec une interface utilisateur intuitive et personnalisable selon les types d'événements.
- **Développer un tableau de bord analytique avancé** : Fournir aux organisateurs une interface de suivi en temps réel présentant des données détaillées sur les transactions, les flux d'entrée, et les tendances de fréquentation pour faciliter la prise de décision stratégique.
- **Faciliter la gestion des événements** : Doter les organisateurs de fonctionnalités complètes pour ajuster les tarifs dynamiquement et suivre précisément le taux de remplissage par zone ou catégorie de places.

- **Intégrer des services complémentaires** : Incorporer des options de personnalisation comme la réservation de services additionnels (restauration, merchandising, vestiaire) directement depuis l'application pour augmenter la valeur ajoutée de la solution.

Cette approche holistique vise à transformer la gestion événementielle traditionnelle en une expérience digitale fluide et sécurisée, répondant aux attentes des utilisateurs modernes tout en offrant aux organisateurs des outils performants pour optimiser leurs opérations et maximiser leur rentabilité.

### 1.3.3. Hypothèse

Nous pensons qu'une application mobile bien faite va rendre l'achat et la gestion des tickets plus faciles pour les utilisateurs. Avec des fonctions comme la réservation, les QR codes et les rappels, les gens pourront accéder aux événements sans stress. Si l'application est rapide, sécurisée et compatible avec tous les téléphones, elle pourra conquérir une large base d'utilisateurs.

### 1.3.4. Spécifications fonctionnelles

L'application mobile devra intégrer plusieurs fonctionnalités essentielles pour assurer une gestion efficace des tickets et des accès :

- **Création de compte utilisateur** : Inscription et authentification sécurisée des utilisateurs.
- **Achat et réservation de tickets** : Interface intuitive permettant de sélectionner un événement, un jeu ou un film et effectuer un paiement en ligne.
- **Gestion des centres par les directeurs** : Interface dédiée aux organisateurs pour ajouter, modifier ou supprimer des événements, des films ou des jeux.
- **Suivi des ventes en temps réel** : Statistiques et tableaux de bord permettant de surveiller l'évolution des ventes.
- **Tableau de bord admin** : Pour la validation des centres, et la gestion des utilisateurs
- **Notifications et rappels** : Alertes envoyées aux utilisateurs pour les informer des événements à venir ou des changements éventuels.

### 1.3.5. Spécifications non fonctionnelles

Outre les spécifications fonctionnelles, l'application devra respecter plusieurs autres critères de qualité pour assurer une utilisation optimale :

- **Performance** : Temps de réponse rapide et fluidité dans l'utilisation.
- **Sécurité** : Chiffrement des données, protection des transactions et authentification robuste.
- **Compatibilité** : Disponible sur Android et iOS pour toucher un large public.

- **Scalabilité ou évolutivité** : Capacité à gérer un nombre d'utilisateurs évolutif et d'événements sans perte de performance.
- **Fiabilité** : Disponibilité continue et résilience face aux pannes

### 1.3.6. Etude de l'existant

Dans le secteur du divertissement, la gestion des tickets reste parfois complexe pour les utilisateurs. Les plateformes existantes ne répondent pas toujours aux besoins de simplicité, d'accessibilité et de sécurité.

#### 1.3.6.1. Présentation de l'existant

##### Site officiel de canal Olympia

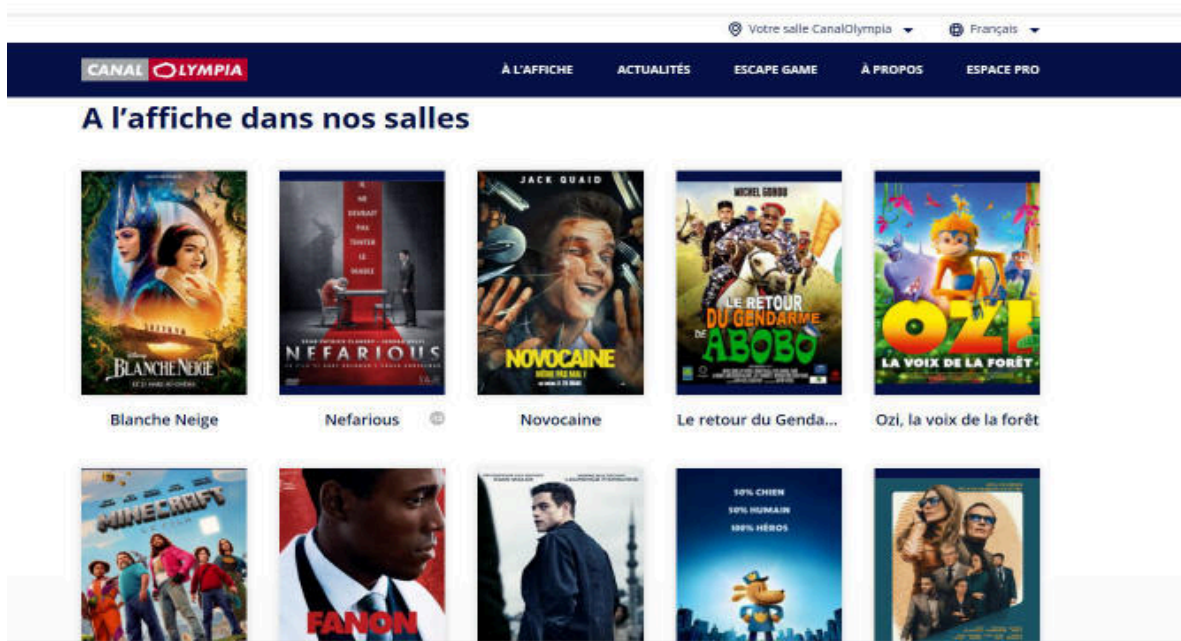


Figure 3 : Site officiel de Canal Olympia (Page A L'AFFICHE) Source : <https://www.canalolympia.com/>



Figure 4 : Site officiel de Canal Olympia (Page ESCAPE GAME) Source : <https://www.canalolympia.com/>



Figure 5 : Site officiel de Canal Olympia (Page ESCAPE PRO) Source : <https://www.canalolympia.com/>

Le site officiel de Canal Olympia présente une interface élégante aux couleurs bleu-marine et rouge, avec une navigation claire et intuitive en haut de page. La section principale «À l'affiche» met en valeur les films actuellement projetés dans un format visuel attrayant. Chaque film est présenté par une affiche de qualité accompagnée de son titre, permettant aux visiteurs d'identifier rapidement

l'offre cinématographique variée incluant des films internationaux, africains et d'animation. L'organisation en grille des affiches offre une présentation moderne et aérée qui valorise le contenu visuel tout en facilitant la navigation.

Au-delà du cinéma, le site diversifie son offre avec des sections dédiées comme «Escape Game», proposant des expériences immersives d'énigmes en équipe à résoudre en moins de 60 minutes, et «Espace Pro» qui cible les entreprises en présentant les salles comme lieux d'événements corporate (séminaires, team building, projections privées). Cette approche multifonctionnelle reflète la stratégie de Canal Olympia de se positionner non seulement comme réseau de cinémas mais comme véritable espace de divertissement et d'événements à travers l'Afrique.

#### 1.3.6.2. Critiques

En dépit d'une présentation visuelle soignée, le site Canal Olympia souffre de plusieurs lacunes fonctionnelles qui limitent considérablement l'expérience utilisateur. L'absence criante d'un système d'achat en ligne oblige les visiteurs à se déplacer physiquement pour acheter leurs tickets. La personnalisation fait également défaut, avec un contenu générique qui ne s'adapte pas à la localisation de l'utilisateur. Plus problématique encore, le site n'affiche pas clairement les événements spécifiques ni les jeux disponibles de façon détaillée, offrant seulement des descriptions générales sans horaires précis. Cette impossibilité d'acheter directement depuis le site web représente un inconvénient majeur qui va à l'encontre des attentes numériques actuelles des utilisateurs, et limite l'accessibilité des services proposés par Canal Olympia.

#### 1.3.7. Intérêt de notre solution par rapport à l'existant

Par rapport au site actuel de Canal Olympia, cette étude apporte une solution plus complète et interactive. L'application mobile corrige les limites existantes comme le manque d'achat en ligne, l'absence de calendrier personnalisé et la faible accessibilité mobile. Elle propose une expérience plus moderne, adaptée aux besoins des utilisateurs d'aujourd'hui, tout en renforçant la gestion des événements et la fidélisation du public.

## **Chapitre 2 : Analyse et conception du système à mettre en place**

## Section 2.1 : Analyse fonctionnelle

### 2.1.1. Présentation des outils d'analyse

Les différentes techniques de modélisation permettent de simplifier des situations complexes en mettant l'accent sur les aspects clés. Avec les progrès technologiques et l'expérience accumulée, les méthodologies de modélisation ont évolué, passant d'approches parfois trop contraignantes à des pratiques plus souples et adaptées aux besoins des développeurs. De nombreuses méthodes d'analyse et de conception existent, telles que UML, MERISE, OMT, Booch, OOSE et BPEL. Toutefois, dans le cadre de notre formation, l'accent a été mis sur l'utilisation d'UML pour la modélisation et la conception du système.

#### 2.1.1.1. UML



UML ou Unified Modeling Language, développé en 1997, est un langage standardisé permettant de modéliser visuellement tous les aspects d'un système logiciel. Il est particulièrement apprécié pour sa flexibilité et sa polyvalence.

UML propose différents types de diagrammes. Les **diagrammes structurels** incluent le **diagramme de classes**, qui définit l'architecture statique du système ; le **diagramme d'objets**, qui représente des instances spécifiques à un moment donné ; et le **diagramme de composants**, qui illustre l'organisation physique du code. Les **diagrammes comportementaux** se concentrent sur l'interaction et le flux au sein du système. Par exemple, le **diagramme de cas d'utilisation** décrit les interactions entre utilisateurs et système, le **diagramme de séquence** présente l'ordre chronologique des messages échangés, le **diagramme d'activité** décrit les flux de contrôle, et le **diagramme d'état-transition** montre les différents états possibles d'un système et leurs transitions.

Depuis son apparition, UML a évolué. La version UML 1.x (1997-2004) proposait un formalisme relativement souple. La version UML 2.x (2005 à nos jours) a enrichi ses fonctionnalités et introduit des possibilités comme la génération automatique de code.

Les avantages d'UML incluent sa reconnaissance universelle, sa polyvalence, sa flexibilité d'utilisation et l'existence d'outils facilitant son adoption. Cependant, son utilisation présente aussi des limites, telles qu'une courbe d'apprentissage assez importante, un risque de sur-modélisation, une divergence possible entre les modèles et le code produit, ainsi qu'une inadaptation à certaines approches spécifiques.

### 2.1.1.2. MERISE



MERISE est une méthodologie française, créée en 1979, qui vise à structurer la conception des systèmes d'information. Elle repose sur une distinction fondamentale entre les données et les traitements, ce qui permet une approche méthodique et rigoureuse.

La modélisation des données dans MERISE se fait en trois étapes : le **Modèle Conceptuel des Données (MCD)**, qui offre une vision métier pure et abstraite ; le **Modèle Logique des Données (MLD)**, qui organise ces données en structures relationnelles compatibles avec les bases de données ; et enfin, le **Modèle Physique des Données (MPD)**, qui ajuste ces structures aux spécificités du système de gestion de base de données (SGBD) choisi.

La modélisation des traitements suit une logique semblable. Elle débute avec le **Modèle Conceptuel des Traitements (MCT)**, qui décrit la logique métier de manière pure et indépendante. Ensuite, le **Modèle Organisationnel des Traitements (MOT)** intègre des considérations organisationnelles et pratiques. Enfin, le **Modèle Opérationnel des Traitements (MPT)** traduit ces concepts en instructions programmables.

MERISE présente des avantages tels qu'une rigueur méthodologique, un langage commun facilitant la communication entre acteurs, une séparation claire des préoccupations, ainsi qu'une indépendance technologique. Cependant, elle souffre de certaines limites, comme une adaptation complexe à l'approche orientée objet, un formalisme parfois jugé lourd et une prise en compte limitée des interfaces utilisateur.

### 2.1.2. Choix du langage de modélisation

Afin d'assurer le bon déroulement de notre projet, nous avons opté pour **UML** comme outil d'analyse, principalement pour sa souplesse dans la création de diagrammes. Conçu comme un langage de modélisation visuelle universel, **UML** se distingue par sa richesse sémantique et syntaxique, facilitant ainsi la représentation des systèmes et l'organisation des processus de développement.

### 2.1.3. Modélisation fonctionnelle

Il faut noter que cette étude se fera à travers ces différentes étapes à savoir :

- l'identification des acteurs ;
- l'élaboration du diagramme de cas d'utilisation.

### 2.1.3.1 Identification des acteurs

Un acteur représente une entité extérieure (comme une personne, un processus ou un objet) qui interagit avec un système. Il joue un rôle bien défini en intervenant directement sur l'état du système, soit en envoyant, soit en recevant des messages pouvant contenir des données. Pour bien identifier les acteurs, il faut s'intéresser aux rôles que ces entités extérieures remplissent vis-à-vis du système étudié. On distingue plusieurs types d'acteurs à savoir :

✓ **Acteurs principaux** : ce sont les utilisateurs qui interagissent directement avec les fonctionnalités principales du système.

✓ **Acteurs secondaires** : ils regroupent les personnes chargées des tâches de maintenance, d'administration, ainsi que les systèmes externes qui interviennent dans l'exécution d'une fonction.

Les acteurs identifiés dans le cadre de notre étude sont les suivants :

- Les utilisateurs ;
- Les responsables de centre ;
- Le système de paiement ;
- L'administrateur.

### 2.1.3.2. Le diagramme de cas d'utilisation

Un **cas d'utilisation** (ou *use case*) représente une fonctionnalité bien définie, visible depuis l'extérieur du système. Il correspond à un service complet rendu à un acteur, avec un point de départ, un déroulement logique et une finalité. Ce cas d'utilisation décrit **ce que fait le système**, sans préciser **comment** il le fait techniquement. Pour les identifier, il faut analyser les interactions entre les acteurs et le système, ainsi que les services fonctionnels attendus décrits dans le cahier des charges. On peut donc dire qu'un cas d'utilisation reflète une **fonction métier** du système.

Dans le cadre de notre étude, nous avons identifié les cas d'utilisation suivants, répartis en fonction des différents acteurs impliqués :

#### ❖ **Responsables centre :**

- S'inscrire ;
- Se connecter ;
- Gérer films / jeux / événements ;
- Voir statistique ;
- Voir profil centre.

❖ **Utilisateurs :**

- S'inscrire ;
- Consulter divertissement (Films, Jeux, Evénements) ;
- Voir profil ;
- Consulter offre tickets ;
- Payer un ticket ;
- Télécharger tickets ;
- Annuler ticket.

❖ **Administrateur :**

- S'inscrire ;
- Se connecter ;
- Gérer les utilisateurs.

Le diagramme de cas d'utilisation se représente alors comme ceci :

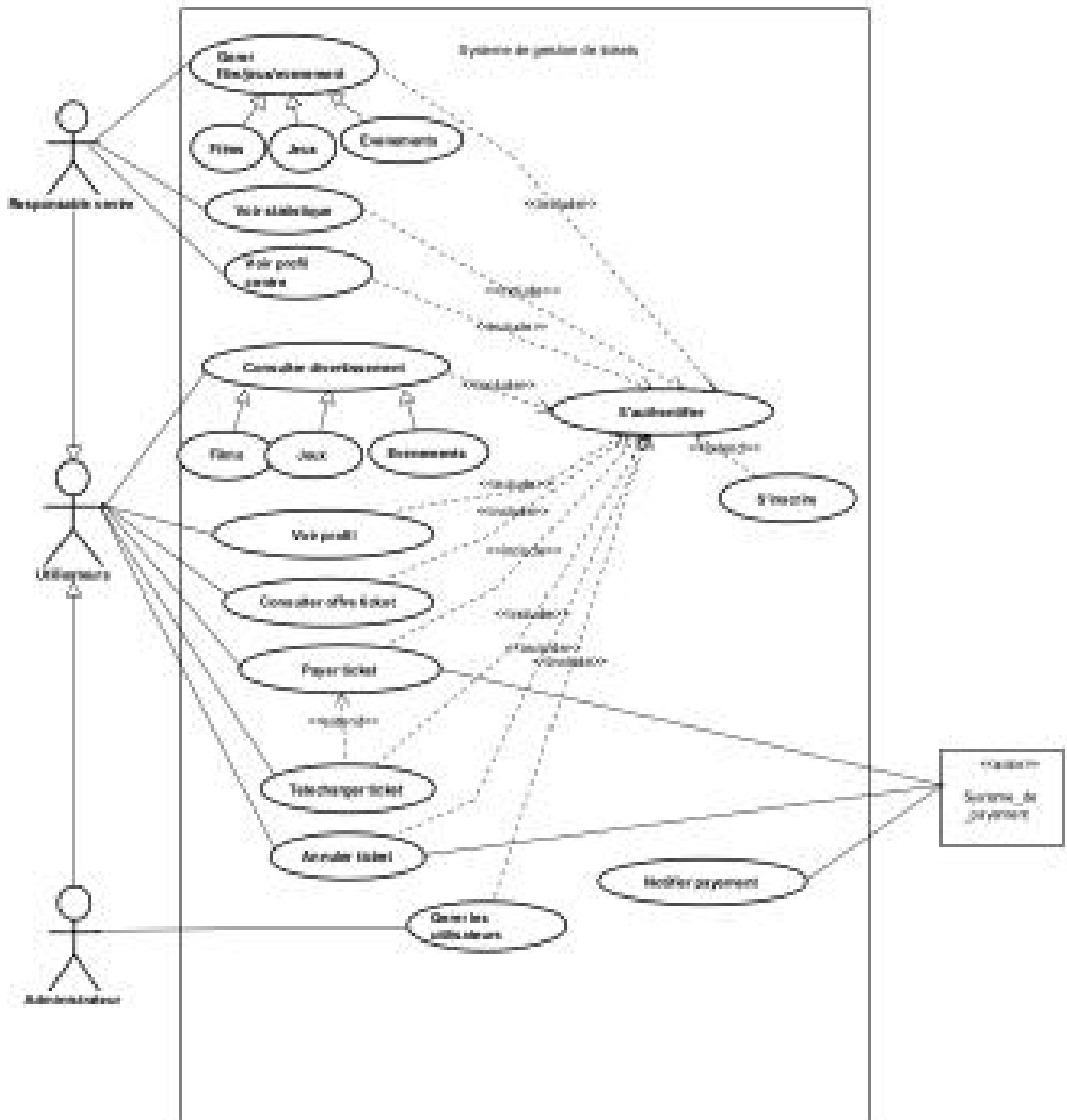


Figure 6 : Diagramme de cas d'utilisation

## Section 2.2 : Analyse statique

La modélisation statique est une étape importante pour comprendre comment un système est construit. Elle montre les différentes parties du système et comment elles sont liées entre elles, sans s'occuper de ce qui change avec le temps.

Notre démarche méthodologique pour l'étude statique s'articulera autour de trois axes complémentaires :

- L'identification des classes, attributs et méthodes ;
- l'élaboration du dictionnaire des données ;
- la construction du diagramme de classes.

### 2.2.1. Identification des classes, attributs et méthodes

Une **classe** est un modèle qui décrit un groupe d'objets similaires.

Par exemple, la classe **Utilisateur** décrit tous les utilisateurs de notre application.

Un **objet** est un exemple concret d'une classe. Si **Utilisateur** est la classe, alors "John" pourrait être un objet spécifique de cette classe.

Les **attributs** sont les informations que chaque objet doit connaître sur lui-même. Pour un utilisateur, ce pourrait être son nom, son email ou son mot de passe.

Les **méthodes** sont les actions qu'un objet peut faire. Pour un utilisateur, ce pourrait être **se connecter**, **acheter un billet** ou **modifier son profil**.

Dans notre tableau qui, nous allons lister chaque classe avec ses attributs et méthodes correspondants, ce qui nous donnera une vue claire de la structure de notre application.

CLASSES	ATTRIBUTS	MÉTHODES
Centre	- id_centre - nom_centre - ville - logo - num_centre - description - documents	+ s'inscrire () + se_connecter () + gerer_offres ()
Offre	- id_offre - lib_offre	-

Jeux	<ul style="list-style-type: none"> <li>- id_jeux</li> <li>- nom_jeux</li> <li>- lieu_jeux</li> <li>- tarif_enf_jeux</li> <li>- tarif_adu_jeux</li> <li>- logo_jeux</li> <li>- age_mini</li> </ul>	-
Evénements	<ul style="list-style-type: none"> <li>-id_event</li> <li>-nom_event</li> <li>-image_event</li> <li>-lieu_event</li> <li>-tarif_stand</li> <li>-tarif_VIP</li> <li>_tarif_VVIP</li> </ul>	-
Films	<ul style="list-style-type: none"> <li>- id_film</li> <li>- nom_film</li> <li>- image</li> <li>- duree_film</li> <li>- tarif_enf_film</li> <li>- tarif_adu_film</li> <li>- tarif_première</li> </ul>	-
Categorie_film	<ul style="list-style-type: none"> <li>- id_cat_film</li> <li>- lib_cat_film</li> </ul>	-
Format	<ul style="list-style-type: none"> <li>- id_format</li> <li>- lib_format</li> </ul>	-
Genre	<ul style="list-style-type: none"> <li>- id_genre</li> <li>- lib_genre</li> </ul>	-
Langue	<ul style="list-style-type: none"> <li>- id_langue</li> <li>- lib_langue</li> </ul>	-
Tranche_age	<ul style="list-style-type: none"> <li>- id_tranche</li> <li>- lib_tranche</li> </ul>	-
Programme	<ul style="list-style-type: none"> <li>- id_programme</li> <li>- heure</li> </ul>	-
Jour	<ul style="list-style-type: none"> <li>- id_jour</li> <li>- lib_jour</li> </ul>	-

Tickets	- id_ticket - date_ach - date_annul - montant_ticket	-
Statut_ticket	- id_statut - lib_statut	-
Payement	- id_pay - date_pay	+ se_connecter () + gerer_offres ()
Users	- id_user	+ s'inscrire ()
Client	- nom_client - num_client - mot_de_passe_client	+ se_connecter () + acheter_ticket ()
Administrateur	-nom_admin -num_admin -mot_de_passe_admin	+ se_connecter () + gerer_utilisateur ()

Tableau 1 : Identification des classes et attribut

### 2.2.2. Elaboration du dictionnaire des données

Le dictionnaire des données constitue un référentiel central indispensable à la construction rigoureuse d'une base de données relationnelle. Il s'agit d'un inventaire méthodique et exhaustif qui documente chaque élément informationnel en précisant sa définition sémantique, sa provenance, ses spécifications techniques, ses modalités d'exploitation et ses interconnexions avec les autres composantes du système.

Formalisé sous forme tabulaire, ce référentiel doit respecter un ensemble de principes structurants pour garantir son intégrité et son efficacité :

- il ne doit y avoir d'homonymes ni de synonymes ;
- les données doivent être regroupées par entité ;
- les noms des données doivent être monovalués et non décomposables ;
- les identifiants doivent être complètement précisés ;
- les commentaires doivent être pertinents.

Le dictionnaire des données de modélisation se présente comme suit :

DONNÉES	DESCRIPTION	TYPE	TAILLE	NATURE	COMMENTAIRE
id_centre	L'identifiant de centre	N	-	E	-
nom_centre	Le nom du centre	AN	30	E	-
ville	La localisation du centre	AN	30	E	-
logo	Le logo du centre	AN	60	E	-
num_centre	Le numéro du centre	N	-	E	-
description	Une description du centre	AN	100	E	-
documents	Le document fourni par le centre	AN	60	E	-
id_offre	L'identifiant des offres de notre application	N	-	E	-
lib_offre	Le libellé des offres de notre application	AN	20	E	-
id_jeux	L'identifiant du jeu proposé	N	-	E	-
nom_jeux	Le nom du jeu proposé	AN	30	E	-
lieu_jeux	Le lieu/centre du jeu proposé	AN	30	E	-
duree_jeux	La durée du jeu	AN	10	E	-
tarif_enf_jeux	Le tarif enfant par jeu	N	-	E	-
tarif_adu_jeux	Le tarif adulte par jeu	N	-	E	-
logo_jeux	Le logo/image du jeu	AN	60	E	-
age_mini	L'âge minimal pour un jeu	N	-	E	-
id_film	L'identifiant du film proposé	N	-	E	-
nom_film	Le nom du film proposé	AN	30	E	-

image	L'image/affiche du film	AN	60	E	-
duree_film	La durée du film	AN	10	E	-
tarif_enf_film	Le tarif enfant par film	N	-	E	-
tarif_adu_film	Le tarif adulte par film	N	-	E	-
tarif_première	Le tarif première par film	N	-	E	-
id_cat_film	L'identifiant de la catégorie du film	N	-	E	-
lib_cat_film	Le libellé de la catégorie du film	AN	20	E	-
prix	Le prix par catégorie de film	N	-	E	-
id_format	L'identifiant du format du film	N	-	E	-
lib_format	Le libellé du format du film	AN	30	E	-
id_genre	L'identifiant du genre du film	N	-	E	-
lib_genre	Le libellé du genre du film	AN	30	E	-
id_langue	L'identifiant de la langue de diffusion du film	N	-	E	-
lib_langue	Le libellé de la langue de diffusion du film	AN	30	E	-
id_tranche	L'identifiant de la tranche d'âge du film	N	-	E	-
lib_tranche	Le libellé de la tranche d'âge du film	AN	30	E	-
id_event	L'identifiant de l'événement proposé	N	-	E	-
nom_event	Le nom de l'événement proposé	AN	30	E	-
image_event	L'image/affiche de l'événement	AN	60	E	-

lieu_event	Le lieu de l'événement	AN	30	E	-
tarif_stand	Le tarif standard de l'événement	N	-	E	-
tarif_VIP	Le tarif VIP de l'événement	N	-	E	-
tarif_VVIP	Le tarif VVIP de l'événement	N	-	E	-
id_statut	L'identifiant du statut d'un ticket	N	-	E	-
lib_statut	Le libellé du statut d'un ticket	AN	30	E	-
id_ticket	L'identifiant du ticket	N	-	E	-
date_ach	La date d'achat du ticket	D	10	E	jj/mm/aaaa
date_annul	La date d'annulation du ticket	D	10	E	jj/mm/aaaa
montant_ticket	Le montant à payer	N	-	E	-
id_programme	L'identifiant du programme	N	-	E	-
heure	L'heure rattachée au programme	AN	10	E	-
id_jour	L'identifiant du jour rattaché au programme	N	-	E	-
lib_jour	Le libellé du jour rattaché au programme	A	15	E	-
id_user	L'identifiant de l'utilisateur	N	-	E	-
nom_client	Le nom de l'utilisateur (client)	AN	30	E	-
num_client	Le numéro de l'utilisateur (client)	N	-	E	-
mot_de_passe_client	Le mot de passe de l'utilisateur (client)	AN	10	E	-
nom_admin	Le nom de l'administrateur	AN	30	E	-

num_admin	Le numéro de l'administrateur	N	-	E	-
mot_de_passe_admin	Le mot de passe de l'administrateur	AN	10	E	-
id_pay	L'identifiant du paiement	N	-	E	-
date_pay	La date de paiement	D	10	E	jj/mm/aaaa

Tableau 2 : Dictionnaire des données

### 2.2.3. Diagramme de classes

Le diagramme de classes est un élément clé de la modélisation orientée objet. Contrairement au diagramme des cas d'utilisation, qui montre comment les utilisateurs interagissent avec le système, le diagramme de classes s'intéresse à la structure interne du système. Il montre les différentes classes (ou éléments importants) du système, ainsi que les liens entre elles (comme les relations, les héritages ou les dépendances). Ce diagramme permet donc de mieux comprendre comment les objets du système vont travailler ensemble pour accomplir les fonctions prévues. Le diagramme que nous présentons ci-dessous est une base technique importante du projet. Il offre une vue d'ensemble claire des éléments du système et de leurs relations, ce qui est essentiel pour bien développer la solution et garder une logique cohérente tout au long du projet.

Voici donc le diagramme de classe réalisé :

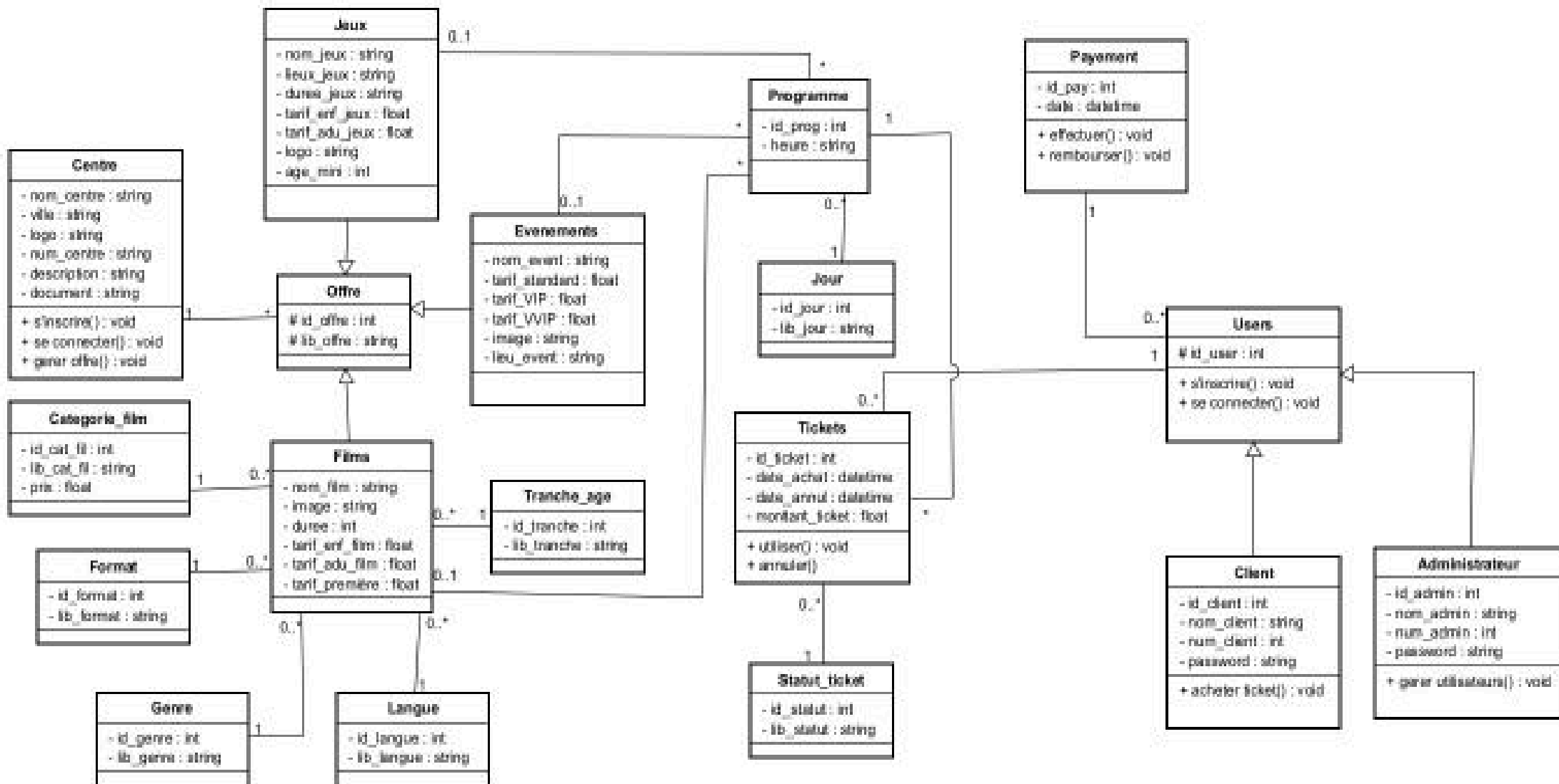


Figure 7 : Diagramme des classes

## Section 2.3 : Analyse dynamique

À cette étape de notre analyse, nous déployons une approche structurée en trois volets complémentaires permettant de capturer avec précision les aspects dynamiques et interactionnels du système :

- la description textuelle des cas d'utilisation ;
- la conception des diagrammes de séquence ;
- l'élaboration des diagrammes d'activités.

Cette triple perspective nous permet d'appréhender de manière complète et cohérente les dimensions comportementales du système, en assurant l'alignement entre les attentes fonctionnelles exprimées et leur traduction technique en termes d'interactions et de processus.

### 2.3.1. Description textuelle des cas d'utilisation et conception des diagrammes de séquence

La description textuelle d'un cas d'utilisation explique comment une action se déroule dans le système. Elle comprend un scénario principal (le déroulement normal), des alternatives et des erreurs possibles. Chaque étape est numérotée et décrit une interaction entre l'utilisateur et le système. On précise aussi ce qui doit être en place avant l'action (préconditions) et ce qu'on obtient après (postconditions).

#### A. Cas d'utilisation : S'inscrire en tant qu'utilisateur

##### SOMMAIRE D'IDENTIFICATION

- **Titre** : S'inscrire en tant qu'utilisateur
- **Résumé** : Ce cas permet à un visiteur de s'inscrire à l'application afin d'accéder à ses fonctionnalités.
- **Acteurs** : Visiteurs (futur utilisateur)
- **Responsables** : Emmanuella, Irfane
- **Date de création** : 18/04/2025
- **Version** : 1.0

## SCÉNARIOS DU CAS D'UTILISATION « S'inscrire en tant qu'utilisateur »

### Préconditions :

- L'application est installée
- La connexion internet est fonctionnelle
- L'utilisateur accède à l'application.

### Scénario nominal :

Étape	Visiteurs	Systeme
1	Le visiteur clique sur "Client"	
2		Le système affiche le formulaire d'inscription
3	Le visiteur remplit les champs requis et valide	
4		Le système vérifie si tous les champs sont remplis
5		Le système vérifie si les informations sont valides
6		Le système crée un compte et redirige l'utilisateur vers la page d'accueil

Tableau 3 : Scénario nominal du cas « Inscription utilisateur »

### Enchaînements alternatifs :

- **E1 : Champs vides**
  - Démarre au point 4
  - Le système affiche un message d'erreur : "Veuillez remplir tous les champs"
  - Retour à l'étape 3
- **E2 : Données invalides (mot de passe trop court, etc.)**
  - Démarre au point 5

- Le système affiche un message d'erreur : "Le mot de passe doit contenir au moins 8 caractères"
- Retour à l'étape 3

Postconditions :

- L'utilisateur est inscrit et connecté.

Le diagramme illustre l'interface d'inscription sous le titre "Créer votre compte". Elle est contenue dans un cadre arrondi et comprend les éléments suivants :

- Un champ de saisie pour le "Nom complet".
- Un champ de saisie pour le numéro de téléphone, divisé en deux sections : "+229" et "01".
- Un champ de saisie pour le "Mot de passe".
- Un champ de saisie pour la "Confirmation mot de passe".
- Un bouton "S'inscrire" en bas.

Figure 8 : Dessin de l'interface d'inscription

Diagramme de séquence « S'inscrire en tant qu'utilisateur »

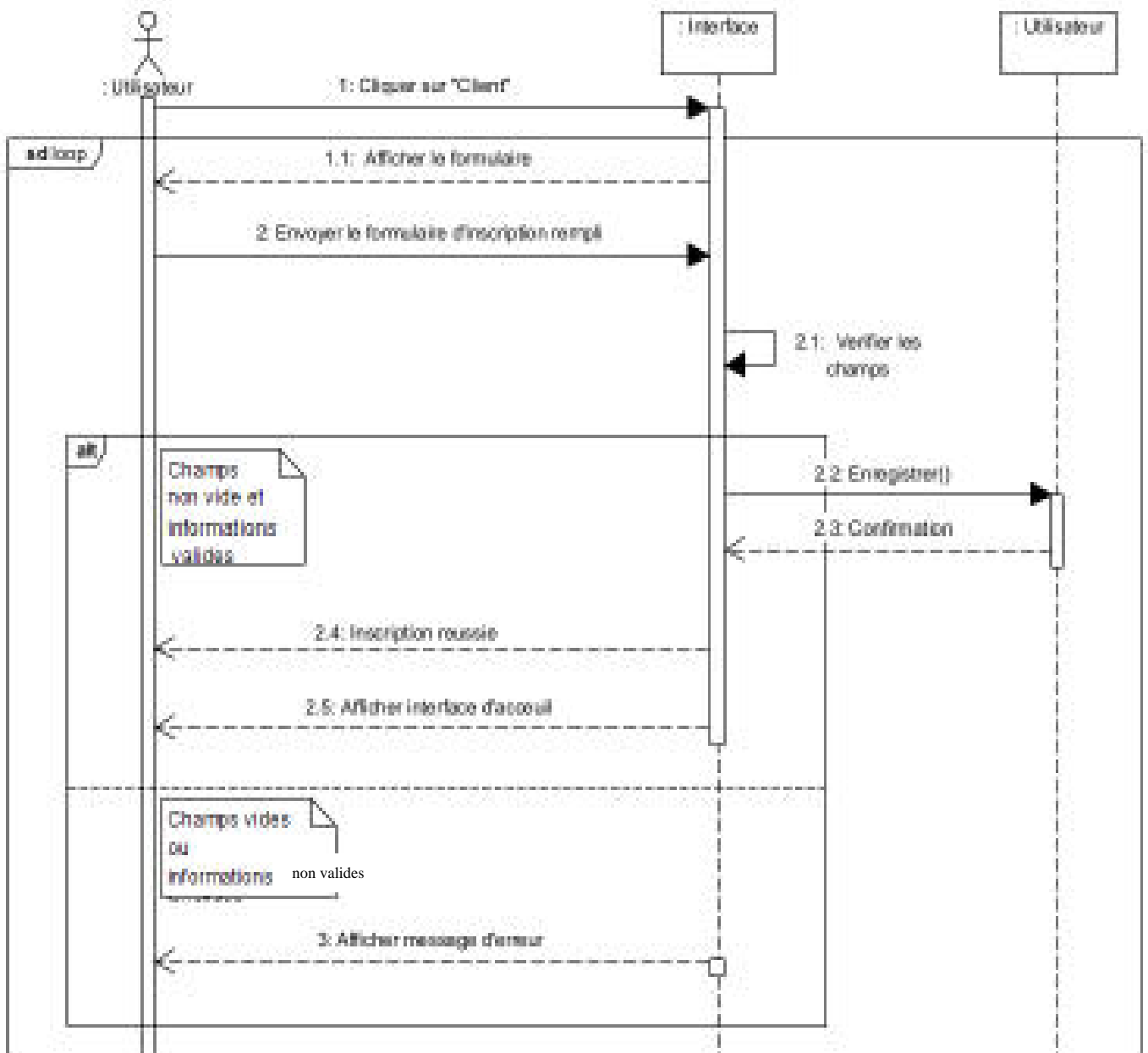


Figure 9 : Diagramme de séquence du cas « S'inscrire »

## B. Cas d'utilisation : Payer un ticket

### SOMMAIRE D'IDENTIFICATION

- **Titre** : Payer un ticket
- **Résumé** : Permet à un utilisateur d'acheter un ticket pour un film, un jeu ou un événement dans un centre.
- **Acteurs** : Utilisateurs
- **Responsables** : Emmanuella, Irfane
- **Date de création** : 18/04/2025
- **Version** : 1.0

### SCÉNARIOS DU CAS D'UTILISATION « Payer un ticket »

#### Préconditions :

- L'utilisateur est connecté.
- Il a sélectionné une activité (cinéma, jeu ou événement).

Scénario nominal :

Étape	Utilisateurs	Système
1	L'utilisateur sélectionne une activité (cinéma, jeu ou événement)	
2		Le système affiche la liste disponible selon le choix de l'utilisateur
3	L'utilisateur choisit une option (ex : un film dans un centre)	
4		Le système affiche les détails de l'activité ( Nom, durée, etc)
5	L'utilisateur clique sur "Procéder au paiement"	
6		Le système affiche le formulaire réservation de tickets
7	L'utilisateur remplit le formulaire et valide le paiement	
8		Le système traite le paiement
9		Le système enregistre le ticket avec statut « payé »
10		Le système envoie une notification de confirmation
11		L'utilisateur peut ensuite télécharger le ticket

Tableau 4 : Scénario nominal du cas « Payer un ticket »

Enchaînements alternatifs :

- **E1 : Paiement échoué (ex : solde insuffisant)**

Démarre au point 6

Le système affiche un message d'erreur « Paiement refusé »

Le scénario reprend au point 3

- **E2 : Ticket non utilisé et non annulé**

Le statut reste "valide" jusqu'à expiration

Postconditions :

- Le ticket est soit valide ou expiré.

**Réservation**

PS4  
Cotonou  
15min  
Type de ticket  
Adulte  
Catégorie  
Standard  
Quantité  
- 1 +  
Total : 2000.00f  
Procéder au paiement

Figure 10 : Dessin de l'interface de réservation de ticket

Diagramme de séquence « Payer un ticket »

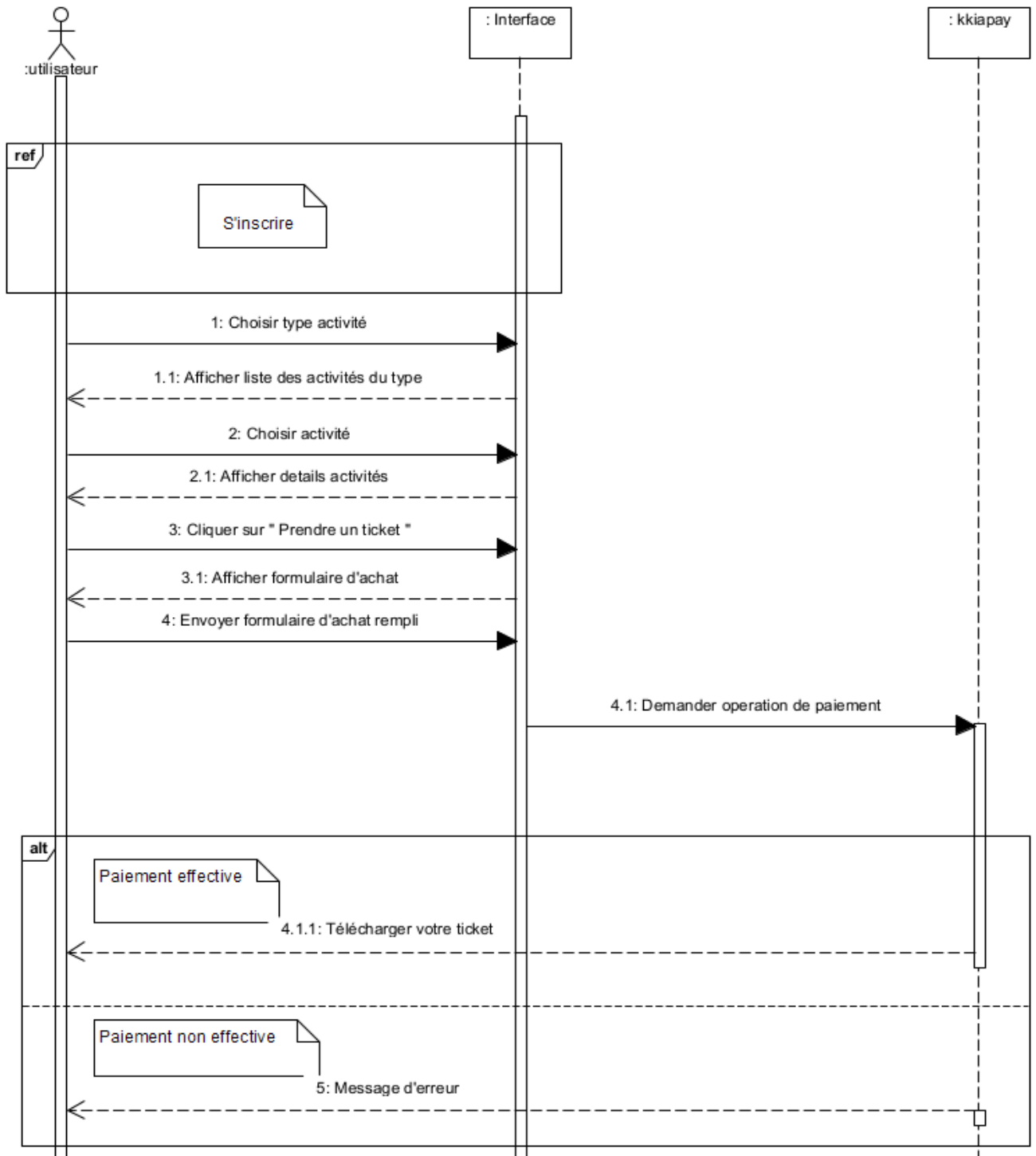


Figure 11 : Diagramme de séquence du cas « Payer un ticket »

### 2.3.2. Elaboration des diagrammes d'activités

Le diagramme d'activité est un diagramme comportemental d'UML, permettant de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables. Le diagramme d'activité est également utilisé pour décrire un flux de travail.

- Diagramme d'activité du cas d'utilisation « S'inscrire »

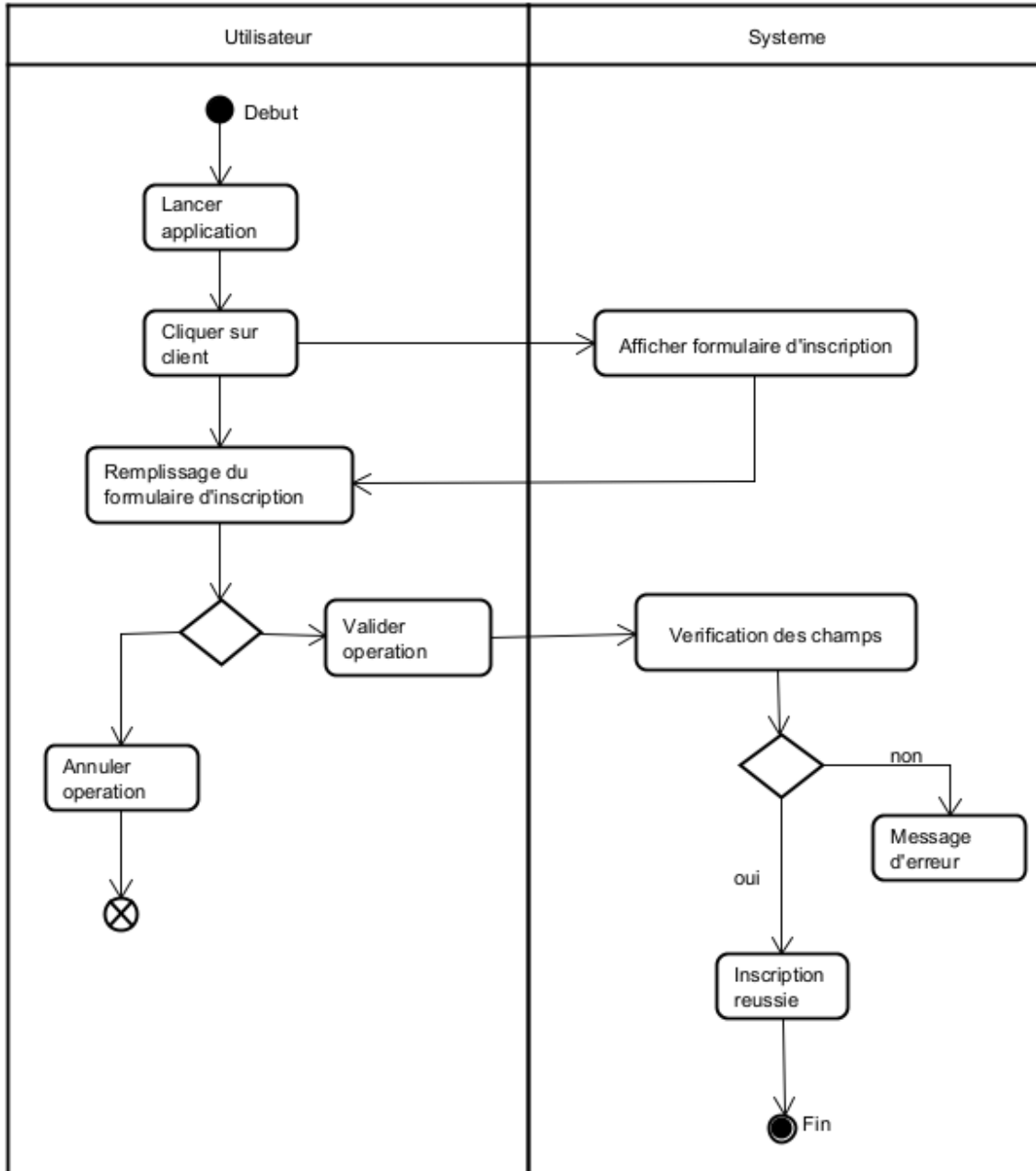


Figure 12 : Diagramme d'activité du cas d'utilisation « S'inscrire »

• Diagramme d'activité du cas d'utilisation « Payer un ticket »

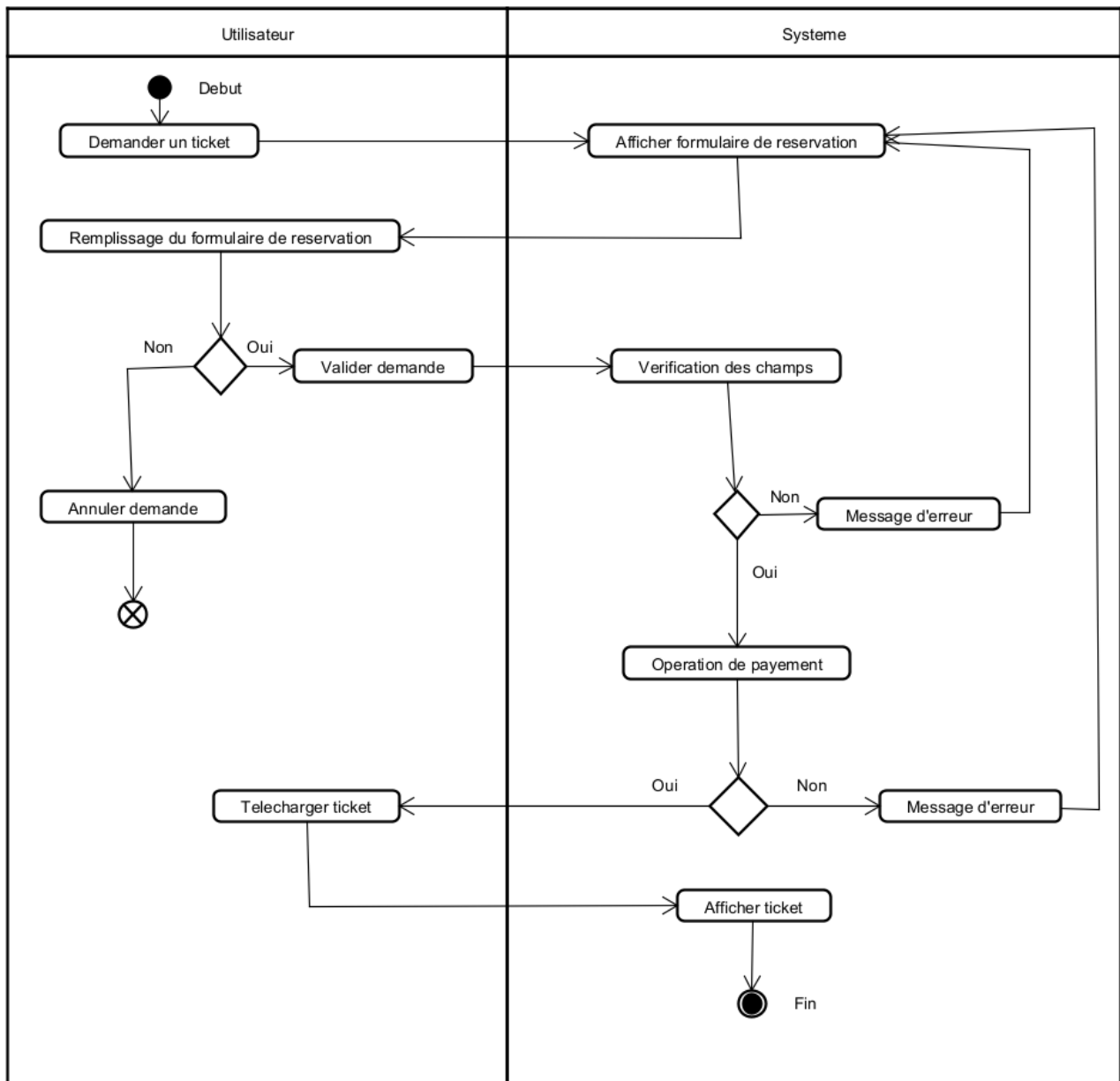


Figure 13 : Diagramme d'activité du cas d'utilisation « Payer un ticket »

## **Chapitre 3 : Réalisation du système**

## Section 3.1 : Présentation des outils et technologies

L'implémentation du système a nécessité l'utilisation d'un certain nombre d'outils énumérés ci-dessous :

### 3.1.1. Outils de programmation

#### 3.1.1.1. Langages utilisés

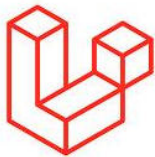
- **Dart**



Dart est un langage de programmation développé par Google en 2011. Il a été conçu pour offrir une alternative à JavaScript, particulièrement adapté au développement d'applications web modernes et performantes. Dart est également utilisé avec Flutter, un framework populaire pour créer des applications mobiles multiplateformes.

#### 3.1.1.2. Framework utilisés

- **Laravel**



Laravel est un Framework du langage de programmation PHP. Il a été créé par Taylor Otwell et regroupe les meilleures bibliothèques utiles pour la création d'un site web. L'utilisation de ce Framework permet d'avoir accès à des fonctionnalités préprogrammées telles que l'authentification, le routage et la création de modèle HTML

- **Flutter**



Flutter est un framework open-source développé par Google qui permet de créer des applications mobiles, web et desktop avec un seul code source. Il utilise le langage Dart et son propre moteur de rendu graphique Skia, offrant ainsi une interface utilisateur fluide et personnalisable. Grâce à la fonctionnalité Hot Reload, les développeurs peuvent voir instantanément les modifications sans redémarrer l'application. Flutter est apprécié pour sa rapidité, sa flexibilité et sa compatibilité multiplateforme, ce qui en fait un excellent choix pour le développement moderne d'applications.

### 3.1.1.3. SGBD

MySQL est l'un des systèmes de gestion de bases de données les plus populaires au monde. Il est open-source et largement utilisé dans le développement web et logiciel. MySQL prend en charge de nombreux types de données, y compris les chaînes de caractères, les nombres, les dates et les binaires. Il offre une grande performance, une évolutivité et une fiabilité, ce qui en fait un choix privilégié pour les applications de toutes tailles, des petites applications web aux grandes entreprises. MySQL est également connu pour sa compatibilité avec de nombreuses plateformes et langages de programmation.



### 3.1.1.4. Serveur

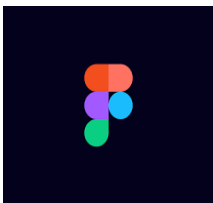
WAMP (Windows, Apache, MySQL, PHP) est un logiciel serveur permettant de créer un environnement de développement web sur un ordinateur Windows. Il regroupe Apache (serveur web), MySQL (gestionnaire de base de données) et PHP (langage de programmation) pour exécuter des sites et applications web en local. WAMP est largement utilisé par les développeurs pour tester leurs projets avant de les mettre en ligne, grâce à une interface simple facilitant la gestion des services.



### 3.1.2. Outils de design

#### **Figma**

Figma est un outil de conception d'interfaces utilisateur (UI) et d'expérience utilisateur (UX) en ligne, permettant de créer des maquettes, des prototypes interactifs et de collaborer en temps réel. Il facilite le travail d'équipe entre designers et développeurs grâce à sa compatibilité multiplateforme, sa gestion de composants réutilisables et ses fonctionnalités de prototypage rapide.



### 3.1.3. Environnement de développement

Notre choix s'est porté sur le développement d'une application mobile multiplateforme car, contrairement aux applications natives, elle n'est pas restreinte à un seul système d'exploitation. En effet, une application multiplateforme est développée à l'aide de technologies comme Flutter ou React Native, permettant son utilisation sur plusieurs systèmes d'exploitation tels qu'Android et iOS.

### 3.1.3.1. Environnement matériel

Marque d'ordinateur	Processeur	Système d'exploitation	Mémoire Vive	Disque Dur
HP	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz	Windows 11	8 Go	SSD 512 Go
Dell	11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz 1.80 GHz	Windows 11Pro	16 Go	SSD 512 Go

Tableau 5 : Matériels utilisés

### 3.1.3.2. Environnement logiciel

- **Visual Studio Code**



Visual Studio Code (VS Code) est un éditeur de code source développé par Microsoft. Il est largement utilisé par les développeurs pour écrire et déboguer du code dans divers langages de programmation. VS Code offre une interface utilisateur intuitive, des fonctionnalités puissantes telles que la coloration syntaxique, l'achèvement automatique du code, le débogage intégré, les extensions personnalisables et une intégration étroite avec les outils de gestion de versions comme Git. Son architecture légère en fait un choix populaire pour les projets de toutes tailles et pour une variété de plateformes.

- **Astah professional**



Astah Professional est un outil de modélisation UML puissant et intuitif, utilisé pour concevoir et documenter des systèmes logiciels. Il permet de créer divers types de diagrammes comme les diagrammes de classes, de cas d'utilisation, de séquence ou encore les cartes heuristiques. Grâce à ses fonctionnalités avancées, il facilite la visualisation, l'analyse et la communication des architectures logicielles, tout en offrant la possibilité de générer du code et d'effectuer l'ingénierie inverse.

- **Android Studio**



Android Studio est l'environnement de développement intégré (IDE) officiel de Google pour créer des applications Android. Basé sur IntelliJ IDEA, il offre des outils puissants comme un éditeur de code intelligent, un émulateur rapide, une interface graphique pour la conception d'applications, ainsi qu'un débogueur avancé. Il prend en charge plusieurs langages, dont Java, Kotlin et C++, et permet un développement efficace grâce à des fonctionnalités comme Gradle pour la gestion des projets et le profiler pour optimiser les performances des applications.

- **Postman**



Postman est un outil utilisé pour tester, développer et déboguer des API. Il permet d'envoyer des requêtes HTTP (GET, POST, PUT, DELETE, etc.) et d'analyser les réponses sous divers formats comme JSON ou XML. Postman simplifie l'interaction avec les API sans avoir à écrire de code, tout en offrant des fonctionnalités pour automatiser les tests, gérer différents environnements, simuler des requêtes avec des paramètres, en-têtes et authentifications, et collaborer facilement avec des équipes en partageant des collections d'API. C'est un outil clé pour les développeurs travaillant avec des API.

- **Git**



Git est un système de contrôle de version distribué, largement utilisé dans le développement logiciel. Il permet de suivre l'historique des modifications faites sur un projet, de collaborer efficacement en équipe, et de travailler en parallèle sans écraser le travail des autres. Grâce à Git, chaque membre de l'équipe peut travailler sur sa propre partie du projet, puis partager ses changements avec les autres.

#### 3.1.4. Moyen de paiement



KKiaPay est une solution de paiement développée par la fintech béninoise OpenSi en 2017. Elle agit comme un agrégateur de moyens de paiement, permettant aux entreprises et e-commerçants d'accepter des paiements via Mobile Money (MTN, Moov, Celtics etc.), cartes bancaires. Principalement utilisée en Afrique de l'Ouest francophone, KKiaPay se distingue par sa simplicité, sa sécurité et sa conformité aux normes internationales comme PCI DSS. Son nom, inspiré du yoruba, signifie "paie vite vite".

#### 3.1.5. Architecture du système

L'architecture de notre système est basée sur le modèle Modèle-Vue-Contrôleur (MVC), une structure largement utilisée dans le développement d'applications interactives. À l'instar d'autres approches comme Modèle-Vue-Présentation (MVP) ou Présentation-Abstraction-Contrôle (PAC), le modèle MVC vise à **organiser l'application en couches distinctes**. Cette séparation des responsabilités permet de **simplifier la maintenance, favoriser l'évolutivité et améliorer la lisibilité du code**.

Le pattern MVC est structuré autour de trois composants principaux :

- **Le Modèle** : il gère les données de l'application, en interaction avec la base de données. Dans notre cas, il regroupe l'ensemble des données relatives à la plateforme, ainsi que les classes métier dont les instances sont manipulées.
- **La Vue** : elle correspond à l'interface utilisateur. Sa fonction est de présenter les données fournies par le modèle, permettant ainsi à l'utilisateur d'interagir visuellement avec le système.
- **Le Contrôleur** : il orchestre les interactions entre la vue et le modèle. Il contient la logique métier nécessaire pour interpréter les actions de l'utilisateur, les traiter, et mettre à jour les données ou les vues en conséquence.

L'un des principaux avantages de cette architecture est qu'elle favorise une meilleure organisation du code, tout en assurant la cohérence des données et en simplifiant les opérations de maintenance.

## Section 3.2 : Présentation du résultat

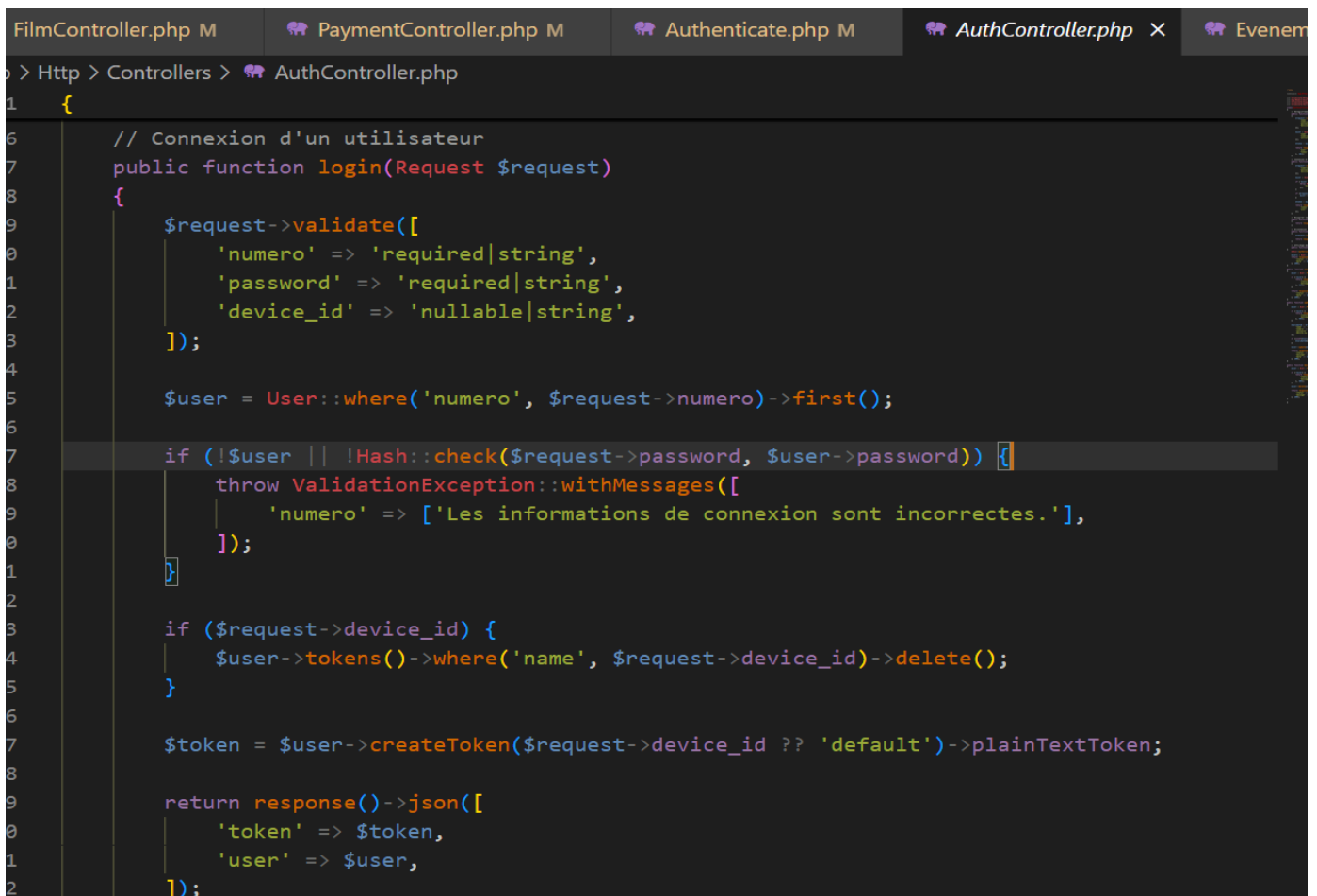
### 3.2.1. Présentation du travail Backend (API RESTful)

Avant de présenter les interfaces de l'application, il est important de souligner le rôle crucial joué par le backend développé sous forme d'API RESTful. Ce backend a été conçu avec le framework Laravel, dans une architecture séparée du frontend afin de faciliter la gestion des données, l'authentification des utilisateurs, et la communication sécurisée entre le client (application mobile ou web) et le serveur.

L'API permet notamment :

- la gestion des utilisateurs et responsables centres (inscription, connexion) ;
- la gestion des tickets ;
- le contrôle des accès selon les rôles (clients, responsable centre, administrateurs) ;
- la fourniture de données en temps réel aux interfaces utilisateur.

L'image suivante illustre un extrait du code du contrôleur Laravel qui gère l'inscription d'un utilisateur.



```

FilmController.php M | PaymentController.php M | Authenticate.php M | AuthController.php X | Evenem
> Http > Controllers > AuthController.php
1  {
6  // Connexion d'un utilisateur
7  public function login(Request $request)
8  {
9      $request->validate([
10         'numero' => 'required|string',
11         'password' => 'required|string',
12         'device_id' => 'nullable|string',
13     ]);
14
15     $user = User::where('numero', $request->numero)->first();
16
17     if (!$user || !Hash::check($request->password, $user->password)) {
18         throw ValidationException::withMessages([
19             'numero' => ['Les informations de connexion sont incorrectes.'],
20         ]);
21     }
22
23     if ($request->device_id) {
24         $user->tokens()->where('name', $request->device_id)->delete();
25     }
26
27     $token = $user->createToken($request->device_id ?? 'default')->plainTextToken;
28
29     return response()->json([
30         'token' => $token,
31         'user' => $user,
32     ]);

```

Figure 14 : Extrait du code backend (contrôleur API) pour la connexion d'un utilisateur



### 3.2.3. Interface d'Inscription (clients)

Cette interface permet aux clients des centres de créer leur compte et d'avoir leur profil utilisateur

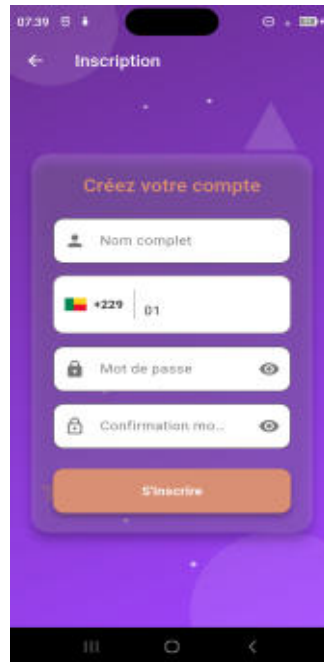


Figure 17 : Interface d'Inscription (clients)

Inscription clients : L'utilisateur voulant utiliser notre application doit remplir le formulaire et cliquer sur « s'inscrire » pour accéder à la page d'accueil.

### 3.2.4. Interface d'Inscription (administrateur)

Cette interface permet aux administrateurs (responsable des centres) de créer leur compte et d'avoir leur profil centre



Figure 18 : Interface d'Inscription (responsable centre)

Inscription administrateur : Un centre voulant utiliser notre application pour vendre ses tickets de divertissement (jeux, cinéma, événements) doit remplir le formulaire et cliquer sur « s'inscrire » pour accéder au tableau de bord.

### 3.2.5. Interface d'Accueil



Figure 19 : Interface d'Accueil

L'écran d'accueil de notre application présente un carrousel d'images qui permet aux utilisateurs de découvrir les offres en cours et de naviguer directement vers les sections correspondantes. La page propose trois catégories principales (Cinéma, Événement et Jeux) accessibles via des boutons.

## Section 3.3: Mesure de sécurité et confidentialité

### 3.3.1. Confidentialité

La confidentialité vise à garantir que seules les personnes autorisées peuvent accéder aux données. Elle protège les informations sensibles contre toute divulgation non autorisée, grâce à des mécanismes comme le chiffrement et le contrôle d'accès. Bien qu'aucun système ne soit totalement inviolable, ces mesures permettent de limiter les risques. Toutefois, la confidentialité repose sur cinq grands principes fondamentaux :

- **Authenticité** : Il s'agit de vérifier l'identité des utilisateurs à l'aide de codes d'accès. L'authenticité repose sur l'association entre un identifiant visible (public) et un élément secret. Cela permet d'attribuer les droits d'accès appropriés et d'instaurer un climat de confiance dans les échanges.
- **Intégrité** : Les données doivent rester conformes à leur état attendu, sans modification accidentelle, illégale ou malveillante. Cela signifie que les informations doivent être exactes et complètes. Des techniques comme les fonctions de hachage ou les sommes de contrôle (*checksum*) sont souvent utilisées pour garantir ce principe.
- **Confidentialité** : Seules les personnes autorisées doivent pouvoir accéder aux informations qui leur sont destinées. Tout accès non autorisé doit être bloqué à l'aide de mécanismes de contrôle d'accès basés sur les droits et permissions.
- **Disponibilité** : Les utilisateurs doivent pouvoir accéder aux services et aux ressources du système de manière continue et fiable pendant les périodes prévues d'utilisation. L'objectif est d'assurer un accès fluide et constant.
- **Traçabilité (ou preuve)** : Toutes les actions, y compris les tentatives d'accès, doivent être enregistrées et conservées. Ces traces doivent être exploitables pour permettre un suivi précis des événements.
- **Non-répudiation et imputation** : Chaque utilisateur doit assumer ses actes et ne pas pouvoir les nier par la suite. De même, aucune action ne doit pouvoir être attribuée à un autre utilisateur que celui qui l'a réellement effectuée.

### 3.3.2. Sécurité

La sécurité informatique est une discipline qui vise à protéger l'authenticité et l'intégrité des informations stockées dans un système informatique. Bien qu'aucune technique ne puisse garantir l'invulnérabilité absolue d'un système, la cybersécurité s'articule autour de six objectifs fondamentaux :

- **Authenticité** : Les utilisateurs doivent prouver leur identité via des mécanismes d'authentification. La correspondance entre un identifiant public et une donnée secrète garantit l'authenticité de l'utilisateur, permettant ainsi une gestion efficace des droits d'accès et le maintien de la confiance dans les échanges numériques.
- **Intégrité** : Les données doivent demeurer exactes, complètes et inaltérées, qu'il s'agisse de modifications fortuites ou malveillantes. Cette protection s'appuie sur des algorithmes de hachage cryptographique et des mécanismes de vérification d'intégrité comme les checksums.
- **Confidentialité** : L'accès aux informations est strictement limité aux personnes autorisées selon une politique de droits précise. Le chiffrement des données en transit et au repos constitue une mesure essentielle pour garantir cette confidentialité.
- **Disponibilité** : Les ressources et services du système d'information doivent rester accessibles de manière continue et performante pendant les plages d'utilisation prévues, avec des mécanismes de redondance et de reprise après incident.
- **Traçabilité** : Toutes les actions et tentatives d'accès sont enregistrées dans des journaux sécurisés, horodatés et inaltérables, permettant des analyses forensiques en cas d'incident.
- **Non-répudiation** : Ce principe empêche un utilisateur de nier avoir effectué une action et protège contre l'usurpation d'identité, généralement via des signatures électroniques et des mécanismes d'horodatage certifiés.

Parmi les technologies et méthodes essentielles à la cybersécurité moderne, on compte :

- **Protocole TLS/SSL** : Sécurise les communications sur Internet en établissant un canal chiffré entre client et serveur, assurant confidentialité, intégrité et authentification des échanges.
- **Pare-feu nouvelle génération** : Au-delà du filtrage basique, ces systèmes intègrent l'inspection approfondie des paquets, l'analyse comportementale et la prévention des menaces avancées.
- **Systèmes IDS/IPS** : Combinent la détection d'intrusion par analyse de signatures et comportementale avec des capacités de blocage automatique des attaques en temps réel.
- **Authentification multifactorielle (MFA)**: Étend la 2FA en permettant l'utilisation de multiples facteurs d'authentification (connaissance, possession, biométrie), réduisant considérablement les risques de compromission des accès.
- **Gestion proactive des vulnérabilités** : Approche cyclique combinant scans automatisés, tests d'intrusion, correction priorisée et validation continue des correctifs.
- **Cryptographie avancée** : Emploie des algorithmes à clé publique/privée, des fonctions de hachage résistantes aux collisions et des protocoles d'échange de clés sécurisés pour protéger les données sensibles et garantir leur intégrité.

### 3.3.3. Perspectives

Dans le but de garantir une meilleure expérience utilisateur et de satisfaire pleinement notre clientèle, nous nous engageons à améliorer continuellement notre plateforme de gestion des tickets de divertissement. C'est dans cette dynamique d'évolution que nous avons envisagé, pour les prochaines versions de l'application **Tickngo**, d'intégrer les améliorations suivantes :

- modification des profils clients et centres ;
- ajout de la page “paramètre profil” ;
- ajout de la page “notification client” ;
- mise à jour de la page “statistique” ;
- création de l'interface admin ;
- génération du QR Code sur les tickets ;

**CONCLUSION**

Dans ce mémoire ayant pour thème « **Conception et développement d'une application mobile de gestion des tickets d'événements culturels et de divertissements** », nous avons conçu et développé l'application Tickngo, une solution numérique destinée à simplifier la réservation et la gestion des tickets pour les projections cinématographiques, les jeux interactifs et les autres événements organisés dans les centres.

L'analyse fonctionnelle nous a permis d'identifier les différents acteurs du système ainsi que leurs besoins spécifiques. À partir de cette base, nous avons élaboré une architecture technique cohérente, puis procédé au choix des technologies les plus adaptées au contexte. Le développement de l'application s'est appuyé sur le modèle MVC avec Laravel et flutter afin de garantir une structure claire, modulable et évolutive.

Dans une démarche d'amélioration continue, plusieurs perspectives ont été envisagées. Il s'agira notamment de mettre en place un système de scannage de QR code pour le contrôle des tickets à l'entrée des projections, des jeux et des autres événements, d'intégrer des notifications automatiques pour la confirmation et le rappel des séances et parties, de proposer un espace personnalisé aux utilisateurs pour consulter leur historique de réservation et d'ajouter un tableau de bord plus complet destiné aux directeurs de centre et à l'administrateur.

À travers cette solution, nous aspirons à doter des centres d'un outil moderne, performant et adapté aux réalités du secteur du divertissement (cinéma, jeux et événements) tout en offrant aux utilisateurs une expérience fluide, fiable et agréable.



# **ANNEXES**

- Extrait du code backend (contrôleur API) pour l'inscription centre

```

1  [?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Centre;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8  use Illuminate\Support\Facades\Validator;
9
10 class CentreController extends Controller
11 {
12     /**
13      * Afficher la liste de tous les centres.
14      */
15     public function index()
16     {
17         $centre = Centre::with(['jeux', 'films', 'evenement'])
18             ->where('statut', true)
19             ->get();
20
21         $total = $centre->count();
22
23         return response()->json([
24             'total' => $total,
25             'message' => $total > 0 ? 'Liste des centres récupérée avec succès' : 'Aucun centre trouvé',
26             'data' => $centre
27         ], $total > 0 ? 202 : 404 );
28     }
29
30     /**
31      * Afficher un centre spécifique.

```

```

10 class CentreController extends Controller
30     /**
31      * Afficher un centre spécifique.
32      */
33     public function show($id)
34     {
35         $centre = Centre::with(['jeux', 'film', 'evenement'])->find($id);
36
37         if (!$centre) {
38             return response()->json([
39                 'total' => 0,
40                 'message' => 'Centre non trouvé',
41                 'data' => null
42             ], 404);
43         }
44
45         return response()->json([
46             'total' => 1,
47             'message' => 'Détails du centre récupérés avec succès',
48             'data' => $centre
49         ], 200);
50     }
51
52     /**
53      * Créer un centre.
54      */
55     public function store(Request $request)
56     {
57         $validator = Validator::make($request->all(), [
58             'nom_centre' => 'required|string|max:255',
59             'ville' => 'required|string|max:255',
60             'logo' => 'nullable|image|max:2048',
61             'num_centre' => [

```

- Extrait du code frontend pour l'inscription centre (inscription\_centre\_screen)

```

import ...

class RegistrationScreen extends StatefulWidget {...}

class _RegistrationScreenState extends State<RegistrationScreen> {
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  final TextEditingController _nomController = TextEditingController();
  final TextEditingController _numeroController = TextEditingController();
  final TextEditingController _descriptionController = TextEditingController();
  final TextEditingController _mdpController = TextEditingController();
  final TextEditingController _confirmMdpController = TextEditingController();
  final TextEditingController _phoneController = TextEditingController();

  String? _selectedVille;
  PlatformFile? _selectedLogo;
  List<PlatformFile> _selectedDocs = [];
  bool _isPasswordVisible = false;
  bool _isConfirmPasswordVisible = false;
  bool _isLoading = false;
  final ApiService _apiService = ApiService();

  List<String> villes = ["Cotonou", "Porto-Novo"];

```

```

List<String> villes = ["Cotonou", "Porto-Novo"];

Future<void> _pickLogo() async {
  final result = await FilePicker.platform.pickFiles(
    type: FileType.image,
  );
  if (result != null && result.files.single.path != null) {
    setState(() => _selectedLogo = result.files.first);
  }
}

Future<void> _pickDocuments() async {
  final result = await FilePicker.platform.pickFiles(
    type: FileType.custom,
    allowedExtensions: ['pdf', 'docx'],
    allowMultiple: true,
  );
  if (result != null && result.files.isNotEmpty) {
    setState(() => _selectedDocs = result.files);
  }
}

```

- Extrait du code service pour l'inscription centre (inscription\_centre\_service)

```
Project ▾
  > Cinema
  > Dashboard
  > Evenement
  > interfaces
  ▾ Jeu
    jeu_scre
  ▾ login
    connexion
    inscripti
    inscripti
  ▾ Services
    cinema_ser
    connexion_
    inscription_s
    inscription_
    kkiapay_ser
  > themes
  > widgets
  main.dart
  > linux
  > macos

ajoutfilm_screen.dart welcome_screen.dart inscription_centre_screen.dart ins

1 > import ...
2
3
4
5
6 class ApiService {
7   final Dio _dio = Dio();
8
9   > Future<void> registerCenter(...) async {
10    try {
11      // Trim des mots de passe avant comparaison
12      final trimmedPassword = password.trim();
13      final trimmedConfirmPassword = confirm_password.trim();
14
15      if (trimmedPassword.isEmpty || trimmedConfirmPassword.isEmpty) {
16        throw Exception("Les champs mot de passe ne peuvent pas être vides.");
17      }
18
19      if (trimmedPassword != trimmedConfirmPassword) {
20        throw Exception("Les mots de passe ne correspondent pas.");
21      }
22
23      const url = '${ApiConfig.baseUrl} + ApiConfig.registerCentrePath';
24
25      _dio.options = BaseOptions(
26        validateStatus: (status) => status! < 500,
27      );
28    }
29  }
30 }
31
32
33
34
35
```

## WEBOGRAPHIE

- <https://www.canalolympia.com/> consulté le 11/04/2025 a 17:13min;
- [https://www.google.com/search?q=android+studio&oq=&gs\\_lcrp=EgZjaHJvbWUqBggCEEUYOzIGCAAQRRg5MgcIARAUgIAEMgYIAhBFGDsyCggDEAAysQMYgAQyBggEEUYPDIGCAUQRRg8MgYIBhBFGDwyBggHEEUYPNIBCDMzODFqMGo3qAIIsAIB8QU1DpS82QxXsg&sourceid=chrome&ie=UTF-8](https://www.google.com/search?q=android+studio&oq=&gs_lcrp=EgZjaHJvbWUqBggCEEUYOzIGCAAQRRg5MgcIARAUgIAEMgYIAhBFGDsyCggDEAAysQMYgAQyBggEEUYPDIGCAUQRRg8MgYIBhBFGDwyBggHEEUYPNIBCDMzODFqMGo3qAIIsAIB8QU1DpS82QxXsg&sourceid=chrome&ie=UTF-8) consulté le 24/03/2025 a 10:53min;
- <https://www.youtube.com/watch?v=Y0sXY-Ucy4> consulté le 24/03/2025 a 09:02min;
- <https://www.youtube.com/watch?v=dOVW7MLur1I> consulté le 21/03/2025 a 09:22min;
- [https://flutter.dev/?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=brand\\_sem&utm\\_content=emea\\_emea&gad\\_source=1&gclid=CjwKCAjw8IfABhBXEiwAxRHlsNz9J6W3-qbJb6k3ZSfE4r4ZMPFi2mjGz4KtkfJDWbbBUi32oKMo5BoCZWsQAvD\\_BwE&gclsrc=a.w.ds](https://flutter.dev/?utm_source=google&utm_medium=cpc&utm_campaign=brand_sem&utm_content=emea_emea&gad_source=1&gclid=CjwKCAjw8IfABhBXEiwAxRHlsNz9J6W3-qbJb6k3ZSfE4r4ZMPFi2mjGz4KtkfJDWbbBUi32oKMo5BoCZWsQAvD_BwE&gclsrc=a.w.ds) consulté le 18/03/2025 a 08:32min;
- <https://stackoverflow.com/questions/51101178/android-studio-device-list-stuck-on-loading> consulté le 16/03/2025 a 18:22min;
- <https://github.com/> consulté le 16/03/2025 a 10:22min.

## **BIBLIOGRAPHIE**

- CONCEPTION ET RÉALISATION D'UNE PLATEFORME DE GESTION DES SALONS DE BEAUTÉ.

Réalisé par : NONFODJI Ronel & TOVIEZON Ornélia en 2024, de la filière Systèmes Informatiques et Logiciels de l'école La Haute École de Commerce et de Management (HECM Calavi).

- CONCEPTION D'UNE APPLICATION DECENTRALISÉE BASÉE SUR LA BLOCKCHAINS : CAS DU VOTE EN LIGNE

Réalisé par : Iréné M. AHOUANTO & Hope H. SAGBO en 2024, de la filière Systèmes Informatiques et Logiciels de l'école La Haute École de Commerce et de Management (HECM Porto-Novvo).

# TABLE DES MATIÈRES

DEDICACE ..	iii
DEDICACE .....	iv
REMERCIEMENTS .....	v
LISTE DES SIGLES, ACRONYMES ET ABRÉVIATIONS .....	vi
LISTE DES TABLEAUX.....	vii
LISTE DES FIGURES.....	viii
RÉSUMÉ .....	ix
ABSTRACT.....	ix
SOMMAIRE .....	x
Section 1.1 : Présentation de la structure d'accueil .....	4
1.1.1 Historique .....	4
1.1.2. Situation géographique .....	4
1.1.3. Services .....	5
1.1.4. Projets et Objectifs .....	5
1.1.5. Organigramme.....	6
Section 1.2 : Déroulement de stage.....	7
1.2.1. Activités menées.....	7
1.2.2. Compétences développées.....	7
Section 1.3 : Présentation du projet d'étude .....	8
1.3.1. Problématique et présentation du thème .....	8
1.3.2. Les objectifs .....	8
3.2.1. Objectif principal .....	8
1.3.2.2. Objectifs spécifiques.....	8
1.3.3. Hypothèse.....	9
1.3.4. Spécifications fonctionnelles.....	9
1.3.5. Spécifications non fonctionnelles.....	9

1.3.6. Etude de l'existant .....	10
1.3.6.1. Présentation de l'existant .....	10
1.3.6.2. Critiques.....	12
1.3.7. Intérêt de notre solution par rapport à l'existant .....	12
Section 2.1 : Analyse fonctionnelle .....	14
2.1.1. Présentation des outils d'analyse.....	14
2.1.2. Choix du langage de modélisation .....	15
2.1.3. Modélisation fonctionnelle.....	15
2.1.3.1 Identification des acteurs .....	16
2.1.3.2. Le diagramme de cas d'utilisation .....	16
Section 2.2 : Analyse statique .....	19
2.2.1. Identification des classes, attributs et méthodes.....	19
2.2.2. Elaboration du dictionnaire des données.....	21
2.2.3. Diagramme de classes .....	25
Section 2.3 : Analyse dynamique.....	27
2.3.1. Description textuelle des cas d'utilisation et conception des diagrammes de séquence .....	27
A. <b>Cas d'utilisation : S'inscrire en tant qu'utilisateur</b> .....	27
B. <b>Cas d'utilisation : Payer un ticket</b> .....	31
2.3.2. Elaboration des diagrammes d'activités .....	35
• Diagramme d'activité du cas d'utilisation « S'inscrire» .....	35
• Diagramme d'activité du cas d'utilisation «Payer un ticket» .....	36
Section 3.1 : Présentation des outils et technologies .....	38
3.1.1. Outils de programmation.....	38
3.1.1.1. Langages utilisés.....	38
3.1.1.2. Framework utilisés.....	38
3.1.1.3. SGBD.....	39
3.1.1.4. Serveur.....	39
3.1.2. Outils de design.....	39

3.1.3. Environnement de développement .....	39
3.1.3.1. Environnement matériel.....	40
3.1.3.2. Environnement logiciel.....	40
3.1.4. Moyen de paiement .....	42
3.1.5. Architecture du système .....	42
Section 3.2 : Présentation du résultat .....	43
3.2.1. Présentation du travail Backend (API RESTful).....	43
3.2.2. Interface de Bienvenue .....	44
3.2.3. Interface d’Inscription (clients) .....	45
3.2.4. Interface d’Inscription (administrateur) .....	45
3.2.5. Interface d'Accueil.....	46
Section 3.3: Mesure de sécurité et confidentialité .....	47
3.3.1. Confidentialité.....	47
3.3.2. Sécurité.....	47
3.3.3. Perspectives .....	49
WEBOGRAPHIE.....	xv
BIBLIOGRAPHIE .....	xvi
TABLE DES MATIÈRES .....	xvii