

PROJET DE FIN D'ANNÉE

3ème Année en Ingénierie Informatique et Réseaux

Développement d'une plateforme de streaming de contenu multimédia

Réalisé par :

Douaa Lakhsadi

Amal Jawhari

Malak Chabli

Encadrant : Mr Ahmed Azouaoui

Année universitaire : 2024/2025

Dédicaces

Nous, Douaa Lakhsadi, Amal Jawhari et Malak Chabli, dédions ce travail avec beaucoup de reconnaissance et d'amour :

À nos familles respectives, pour leur soutien inconditionnel, leurs encouragements constants et leur amour sans limites. Grâce à leur patience, leurs sacrifices et leur présence, nous avons pu avancer sereinement dans notre parcours.

À nos professeurs et encadrants, pour la qualité de leur enseignement, leur bienveillance et leurs précieux conseils tout au long de ce projet. Leur accompagnement a été pour nous une source d'inspiration et de rigueur.

À nos amis, qui ont su être à nos côtés durant les moments de stress comme ceux de réussite. Leur soutien moral, leur écoute et leurs encouragements ont été d'une grande importance pour nous.

Enfin, à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce projet.

Ce travail est le fruit d'un effort collectif, mais aussi d'un environnement humain et bienveillant.

Remerciements

Nous souhaitons exprimer nos plus sincères remerciements à M. Ahmed Azouaoui, enseignant à l'École Marocaine des Sciences de l'Ingénierie (EMSI), pour son encadrement exemplaire tout au long de la réalisation de ce projet. Sa grande disponibilité, sa patience constante, ainsi que ses conseils pertinents et avisés ont grandement contribué à orienter nos efforts dans la bonne direction. Son accompagnement pédagogique rigoureux et motivant a été un véritable moteur dans la progression et la réussite de notre travail.

Nous lui sommes profondément reconnaissants pour le temps qu'il nous a consacré, ainsi que pour sa capacité à nous transmettre non seulement des connaissances techniques mais également des valeurs professionnelles essentielles.

Nous adressons également nos remerciements à l'ensemble du corps professoral de l'EMSI, dont l'expertise, la passion pour l'enseignement et le soutien ont largement contribué à la qualité de notre formation. Leur engagement et leur disponibilité ont favorisé un cadre d'apprentissage stimulant et enrichissant, nous permettant d'acquérir les compétences nécessaires pour mener à bien ce projet.

Nous tenons aussi à remercier chaleureusement l'administration de l'EMSI, pour son appui logistique et organisationnel tout au long de notre parcours. Grâce à leur efficacité et leur soutien, nous avons pu évoluer dans un environnement propice à l'étude et à la recherche.

Enfin, nous exprimons notre profonde gratitude à toutes les personnes, qu'elles soient proches ou éloignées, qui nous ont encouragés, soutenus et accompagnés moralement durant cette période intense de travail. Leur présence, leurs conseils et leurs encouragements ont été des sources précieuses de motivation et de réconfort, sans lesquelles ce projet n'aurait pas pu aboutir avec autant de succès.

Table des matières

Dédicaces	1
Remerciements	2
Introduction Générale	5
1 Présentation du cadre de projet	7
1.1 Introduction	7
1.2 Présentation du contexte académique et des objectifs du PFA	7
1.3 Étude de l'existant	7
1.3.1 Plateformes de streaming existantes	7
1.3.2 Limites et problèmes constatés	7
1.3.3 Opportunité du projet NETFILM	8
1.4 Objectifs du projet	8
1.5 Choix du modèle de développement	8
1.6 Planning prévisionnel	8
1.7 Technologies utilisées	8
1.8 Conclusion	9
2 Spécification des besoins	11
2.1 Introduction	11
2.2 Besoins fonctionnels	11
2.3 Besoins non fonctionnels	11
2.3.1 Gestion des utilisateurs	11
2.3.2 Gestion des films	12
2.3.3 Interface utilisateur fluide et responsive	12
2.4 Besoins non fonctionnels	12
2.5 Cas d'utilisation	12
2.6 Besoins non fonctionnels	12
2.6.1 Acteurs principaux	12
2.6.2 Description des cas d'utilisation	13
2.7 Conclusion	13
3 Conception du système	15
3.1 Introduction	15
3.2 Modélisation UML	15
3.2.1 Diagrammes de cas d'utilisation	15
3.2.2 Diagrammes de séquence	16
3.2.3 Diagrammes d'activités	17
3.2.4 Diagrammes d'états-transitions	17

3.3	Conception statique	17
3.3.1	Diagramme de classes	17
3.3.2	Modèle relationnel de la base de données MySQL	18
3.4	Architecture de l'application	18
3.4.1	Architecture logicielle (Django MVC)	18
3.4.2	Architecture matérielle	18
3.5	Dictionnaire des données	18
3.6	Conclusion	18
4	Réalisation du système	24
4.1	Introduction	24
4.2	Environnement de développement	24
4.2.1	Configuration matérielle	24
4.2.2	Outils et technologies utilisés	24
4.3	Interfaces graphiques développées	24
4.3.1	Page d'accueil	24
4.3.2	Connexion / inscription	25
4.3.3	Catalogue des films	25
4.3.4	Lecture vidéo	25
4.3.5	Tableau de bord utilisateur	25
4.3.6	Apports personnels	25
4.3.7	Perspectives et améliorations futures	25
4.4	Conclusion	25
	Conclusion Générale	26
	Bibliographie et Nétographie	27

Introduction Générale

Dans un monde où la consommation de contenu audiovisuel est en constante évolution, les plateformes de streaming comme Netflix, Disney+ ou Amazon Prime Video se sont imposées comme des références incontournables. Ces services ont transformé notre manière d'accéder aux films et séries, en offrant une expérience personnalisée, accessible à tout moment et depuis n'importe quel appareil connecté.

Dans ce contexte, nous avons choisi de développer une application web appelée NET-FILM, un clone simplifié de Netflix, permettant aux utilisateurs de visionner des films en ligne.

Ce projet s'inscrit dans le cadre de notre stage de fin d'études au sein du département Technologies de l'Informatique. Il représente à la fois un défi technique et une opportunité de mettre en pratique nos compétences en développement web, conception logicielle et modélisation UML. La problématique à laquelle nous souhaitons répondre est la suivante : comment concevoir une plateforme de streaming fonctionnelle, intuitive et sécurisée, permettant de visualiser des films en ligne sans gestion d'abonnements ?

Pour atteindre cet objectif, nous avons opté pour l'utilisation des technologies Django pour le backend, et HTML, CSS, JavaScript pour le développement du frontend. L'approche méthodologique inclut également une phase de modélisation à l'aide du langage UML, afin de garantir une architecture claire et évolutive.

Le présent rapport s'articule autour de plusieurs chapitres. Dans un premier temps, nous analyserons les besoins fonctionnels et non fonctionnels de l'application. Ensuite, nous présenterons la conception UML, incluant les diagrammes essentiels à la compréhension du système.

Le chapitre suivant portera sur l'implémentation technique de l'application, en mettant en évidence les choix technologiques, les structures de données et les principales fonctionnalités réalisées. Enfin, nous conclurons par une évaluation du projet et proposerons des pistes d'amélioration possibles.

C hapitre 1

**Présentation
du cadre de
projet**

Chapitre 1

Présentation du cadre de projet

1.1 Introduction

Dans le cadre de notre Projet de Fin d'Année (PFA), nous avons choisi de concevoir et développer une application web de streaming de films, intitulée NETFILM. Inspirée des plateformes telles que Netflix, cette application vise à offrir une expérience simple et fluide de visionnage de contenus vidéo.

Le développement de ce projet nous permet de mettre en œuvre les compétences acquises durant notre formation, en mobilisant à la fois des outils de conception, de développement web full-stack et de gestion de base de données.

1.2 Présentation du contexte académique et des objectifs du PFA

Ce PFA s'inscrit dans le cadre de notre parcours académique en informatique. Il constitue une étape clé dans l'application pratique des notions vues en cours, telles que la conception UML, le développement web (front-end et back-end), ainsi que l'implémentation d'une base de données relationnelle avec MySQL.

L'objectif principal est de développer une application complète, depuis la phase de spécification jusqu'à sa réalisation et son déploiement.

1.3 Étude de l'existant

1.3.1 Plateformes de streaming existantes

Des plateformes populaires telles que Netflix, Amazon Prime Video, Disney+ ou encore YouTube dominent aujourd'hui le marché du streaming. Ces services offrent un large catalogue de contenus audiovisuels et intègrent des fonctionnalités avancées : recommandations personnalisées, visionnage multi-écran, téléchargements hors ligne, etc.

1.3.2 Limites et problèmes constatés

Bien que performantes, ces plateformes ne sont pas exemptes de limitations. Certaines fonctionnalités sont accessibles uniquement par abonnement, l'interface peut parfois être

surchargée, et les algorithmes de suggestion peuvent manquer de pertinence. Par ailleurs, la complexité technique de ces systèmes les rend peu accessibles aux débutants souhaitant apprendre à créer un tel service.

1.3.3 Opportunité du projet NETFILM

Notre projet NETFILM s'inscrit comme un clone simplifié de ces plateformes. Il permet d'explorer les aspects techniques d'une application de streaming, en nous concentrant uniquement sur les films. L'objectif est de proposer une solution légère, intuitive, et totalement développée par nous, avec une architecture claire et évolutive.

1.4 Objectifs du projet

Les objectifs du projet NETFILM sont multiples :

- Concevoir une plateforme web dédiée à la visualisation de films.
- Offrir une interface utilisateur fluide, responsive et simple à utiliser.
- Implémenter un système d'authentification sécurisé.
- Permettre la lecture de films via un lecteur vidéo intégré.
- Utiliser Django pour le back-end, HTML/CSS/JS pour le front-end, et MySQL comme base de données relationnelle.

1.5 Choix du modèle de développement

Le modèle de développement adopté pour ce projet est le cycle en V. Ce choix est motivé par la clarté de ses phases : spécification, conception, développement, puis tests. Ce modèle permet un meilleur suivi du projet et une validation progressive des livrables à chaque étape.

1.6 Planning prévisionnel

Le développement du projet a été organisé selon plusieurs phases : étude de l'existant, spécification des besoins, conception, développement, tests, et rédaction du rapport final. Ce découpage nous a permis de structurer notre travail de manière rigoureuse.

Semaine Etape	Février				Mars				Avril				Mai			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
<i>Etude préalable</i>																
<i>Conception</i>																
<i>Réalisation</i>																
<i>Test et Validation</i>																

1.7 Technologies utilisées

Voici les principales technologies utilisées :

- Django (Python) pour la gestion du back-end et des modèles de données.
- HTML5, CSS3 et JavaScript pour le front-end.

Résumé des technologies utilisées (avec WAMP)	
Composant	Technologies
Front-end	HTML, CSS, JavaScript
Back-end	Django (Python)
Base de données	MySQL (via phpMyAdmin dans WAMP)
Streaming	Lecteur HTML5 avec lien sécurisé
Authentification	Django Auth, hachage des mots de passe
Hébergement local	WAMP (Windows, Apache, MySQL, PHP/Python)

- MySQL pour la base de données relationnelle.
- Bootstrap pour le design responsive.
- Visual Studio Code comme éditeur de code.
- GitHub pour la gestion de versions.

1.8 Conclusion

Ce premier chapitre a permis de poser les bases de notre projet NETFILM. Nous avons présenté son contexte académique, l'étude des solutions existantes, ainsi que les objectifs fixés. Dans le prochain chapitre, nous allons détailler les spécifications fonctionnelles et non fonctionnelles du système, ainsi que les différents cas d'utilisation.



C2

hapitre

Spécification
des besoins

Chapitre 2

Spécification des besoins

2.1 Introduction

Ce chapitre a pour objectif de présenter les besoins du système NETFILM. Il est divisé en deux grandes parties : les besoins fonctionnels, qui décrivent les fonctionnalités attendues de l'application, et les besoins non fonctionnels, qui concernent les critères de qualité du système. Une section est également dédiée aux cas d'utilisation, afin d'illustrer les interactions entre les utilisateurs et le système.

2.2 Besoins fonctionnels

2.3 Besoins non fonctionnels

Besoins fonctionnels	
Fonctionnalité	Description
Gestion des utilisateurs	Inscription, connexion, gestion du profil
Gestion des films	Ajout, modification, suppression, affichage
Streaming	Lecture des films via une URL sécurisée
Interface utilisateur	Navigaton fluide et intuitive

2.3.1 Gestion des utilisateurs

Inscription / Authentification

- L'utilisateur doit pouvoir créer un compte via un formulaire d'inscription.
- L'utilisateur doit pouvoir se connecter à son compte avec son email et son mot de passe.
- Un mécanisme de vérification des identifiants (avec messages d'erreur en cas d'échec) est requis.
- Une option de réinitialisation de mot de passe est souhaitée.

Gestion du profil

- Une fois connecté, l'utilisateur peut accéder à son profil.
- Il peut modifier certaines informations personnelles (nom, mot de passe).
- Il peut consulter son historique de visionnage.

2.3.2 Gestion des films

Parcourir les films par catégorie

- L'utilisateur peut consulter les films disponibles classés par genre (Action, Comédie, Drame, etc.).
- Un moteur de recherche permet de retrouver rapidement un film par son titre ou par mots-clés.

Lecture des films en streaming

- L'utilisateur peut lancer la lecture d'un film directement via l'interface.
- Le lecteur vidéo intégré permet de mettre en pause, de reprendre, de modifier la qualité de lecture et de régler le volume.

2.3.3 Interface utilisateur fluide et responsive

- L'interface doit être intuitive et accessible depuis différents types d'appareils (PC, tablette, smartphone).
- L'expérience utilisateur doit être fluide, sans temps de chargement excessif.

2.4 Besoins non fonctionnels

Besoins non fonctionnels	
Critère	Description
Performance	Chargement rapide des pages et vidéos
Sécurité	Protection des données, mots de passe hashés
Compatibilité	Responsive sur mobile, tablette et desktop
Scalabilité	Possibilité d'ajouter de nouveaux contenus sans refonte

- **Performance** : Le temps de réponse doit être inférieur à 2 secondes pour les requêtes principales.
- **Compatibilité** : L'application doit être compatible avec les navigateurs modernes (Chrome, Firefox, Edge, Safari).
- **Sécurité** : Les mots de passe doivent être stockés de manière sécurisée (hachage). Les communications doivent être sécurisées (HTTPS).
- **Extensibilité** : Le système doit permettre d'ajouter facilement de nouveaux films et catégories.
- **Maintenabilité** : Le code doit être organisé, commenté et documenté pour faciliter les mises à jour futures.

2.5 Cas d'utilisation

2.6 Besoins non fonctionnels

2.6.1 Acteurs principaux

- **Visiteur** : Utilisateur non connecté qui peut consulter les films disponibles, mais ne peut pas lancer de lecture.

Description des cas d'utilisation		
Acteur	Cas d'utilisation	Description
Utilisateur	S'inscrire	Créer un nouveau compte sur la plateforme
Utilisateur	Se connecter	Accéder à son compte avec email/mot de passe
Utilisateur	Parcourir les films	Voir les films disponibles selon la catégorie
Utilisateur	Lire un film	Regarder un film en streaming
Utilisateur	Gérer son profil	Modifier ses informations personnelles
Administrateur	Ajouter un film	Insérer un nouveau film dans la base
Administrateur	Modifier un film	Mettre à jour les informations d'un film
Administrateur	Supprimer un film	Supprimer un film de la base
Administrateur	Gérer les utilisateurs	Voir/modifier les comptes utilisateurs

- **Utilisateur connecté** : Peut visionner les films, gérer son profil, consulter son historique.
- **Administrateur (optionnel)** : Gère la base de données de films (ajout, modification, suppression).

2.6.2 Description des cas d'utilisation

Voici les cas d'utilisation principaux :

- S'inscrire
- Se connecter
- Parcourir les films
- Rechercher un film
- Lire un film en streaming
- Modifier le profil
- Consulter l'historique de visionnage

2.7 Conclusion

Ce chapitre a présenté de manière structurée les fonctionnalités que devra proposer l'application NETFILM. Ces spécifications permettent de définir le périmètre du projet et serviront de base pour la conception UML détaillée dans le chapitre suivant.

C

hapitre

3

Conception
du système

Chapitre 3

Conception du système

3.1 Introduction

Dans ce chapitre, nous présentons la conception du système de notre application de streaming en ligne, développée avec Django pour le back-end, et HTML, CSS, et JavaScript pour le front-end.

L'application est inspirée de la plateforme Netflix et vise à permettre aux utilisateurs de s'inscrire, se connecter, consulter des films et lancer leur lecture en ligne. Une interface d'administration est également mise en place via l'administration Django pour gérer les contenus, les utilisateurs et les abonnements.

La modélisation UML nous a permis de visualiser et structurer les interactions entre les acteurs et le système à travers différents diagrammes : cas d'utilisation, séquence, activité, et états-transitions.

3.2 Modélisation UML

3.2.1 Diagrammes de cas d'utilisation

Ce diagramme décrit les fonctions principales offertes par l'application NETFILM, en distinguant deux acteurs principaux :

Utilisateur : personne naviguant sur le site pour consulter et regarder des films.

Administrateur : responsable de la gestion du contenu et des comptes via une interface web spécifique (pas l'admin Django).

L'utilisateur peut :

- S'inscrire ou se connecter.
- Gérer son compte utilisateur.
- Rechercher un film ou filtrer par catégorie.
- Consulter les détails d'un film (bande-annonce, description, durée).
- Regarder un film.
- Ajouter aux favoris et consulter ses favoris.
- étendre vers la modification ou la suppression d'un utilisateur («extend»).
- Gérer les abonnements, avec des extensions possibles, voir l'historique des paiements et désactiver un abonnement («extend»).

L'administrateur peut :

- Se connecter.

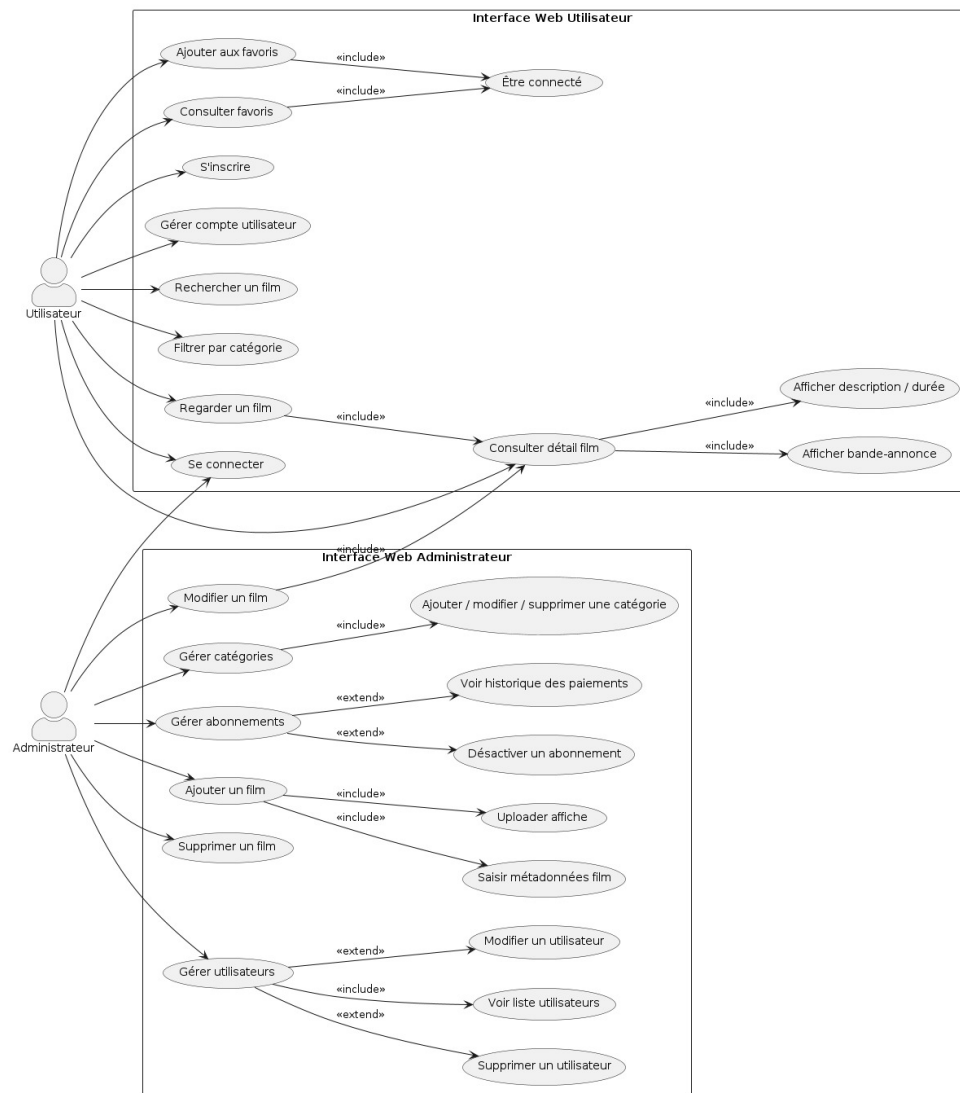


FIGURE 3.1 – Diagramme de Cas d'Utilisation

- Ajouter, modifier, supprimer un film.
- Gérer les utilisateurs (voir, modifier, supprimer).
- Gérer les catégories.
- Gérer les abonnements.

3.2.2 Diagrammes de séquence

Le diagramme de séquence illustre les interactions chronologiques entre les utilisateurs (standard et administrateur) et le système.

Pour l'utilisateur :

1. Connexion avec vérification d'identifiants.
2. Navigation dans le catalogue.
3. Sélection d'un film.
4. Lecture en streaming.
5. Ajout aux favoris.

Pour l'administrateur :

1. Connexion sécurisée.
2. Ajout/modification/suppression de films.
3. Gestion des utilisateurs et abonnements.

3.2.3 Diagrammes d'activités

Ce diagramme montre le parcours de l'utilisateur et de l'administrateur à travers l'interface.

Utilisateur :

- Accès à la plateforme.
- Connexion ou inscription.
- Navigation dans le catalogue.
- Lecture ou ajout aux favoris.

Administrateur :

- Connexion à l'espace admin.
- Gestion des films.
- Gestion des abonnements et utilisateurs.

3.2.4 Diagrammes d'états-transitions

Ce diagramme illustre les états du système :

Utilisateur :

- État initial : NonConnecté.
- Connexion : passe à Connecté.
- État Connecté : accès aux films, favoris, lecture.
- Déconnexion : retour à NonConnecté.

Administrateur :

- État initial : TableauDeBordAdmin.
- Sous-états : AjouterFilm, ModifierFilm, GérerUtilisateurs, etc.
- Déconnexion : sortie de l'espace admin.

3.3 Conception statique

3.3.1 Diagramme de classes

Les classes principales sont :

Utilisateur : nom, email, mot de passe, listes, favoris.

Film : titre, description, durée, genre.

Genre : Action, Drame, etc.

Favoris : films aimés par l'utilisateur.

ListeFilms : collections personnalisées de films.

Relations :

- Un utilisateur peut avoir plusieurs favoris et plusieurs listes.
- Un film peut appartenir à plusieurs genres.

3.3.2 Modèle relationnel de la base de données MySQL

- **Utilisateur** (id, nom, email, mot_de_passe)
- **Film** (id, titre, durée, genre_id, résumé)
- **Genre** (id, nom)
- **ListeFilms** (id, utilisateur_id, films_id)
- **Favoris** (id, utilisateur_id, film_id)
- **Admin** (id, email, mot_de_passe)

3.4 Architecture de l'application

3.4.1 Architecture logicielle (Django MVC)

- **Modèle** : ORM Django, entités du système.
- **Vue** : gestion des requêtes, rendu HTML/JSON.
- **Contrôleur** : logique métier et routage.
- **Template** : affichage visuel, pages utilisateur.

3.4.2 Architecture matérielle

- **Développement** : serveur local Django.
- **Production** : déploiement cloud (Heroku, AWS, etc.).
- **Sécurité** : HTTPS, gestion des API keys, sauvegardes.

3.5 Dictionnaire des données

Toutes les entités de la base (Utilisateur, Film, Genre, etc.) sont définies avec leurs attributs et types. Ce dictionnaire facilite la compréhension du schéma de données pour tout développeur ou administrateur.

3.6 Conclusion

La modélisation UML et la conception statique de NETFILM ont permis de structurer le projet avant son implémentation. Grâce aux diagrammes et à l'architecture logique, nous avons une vision claire du fonctionnement global du système.

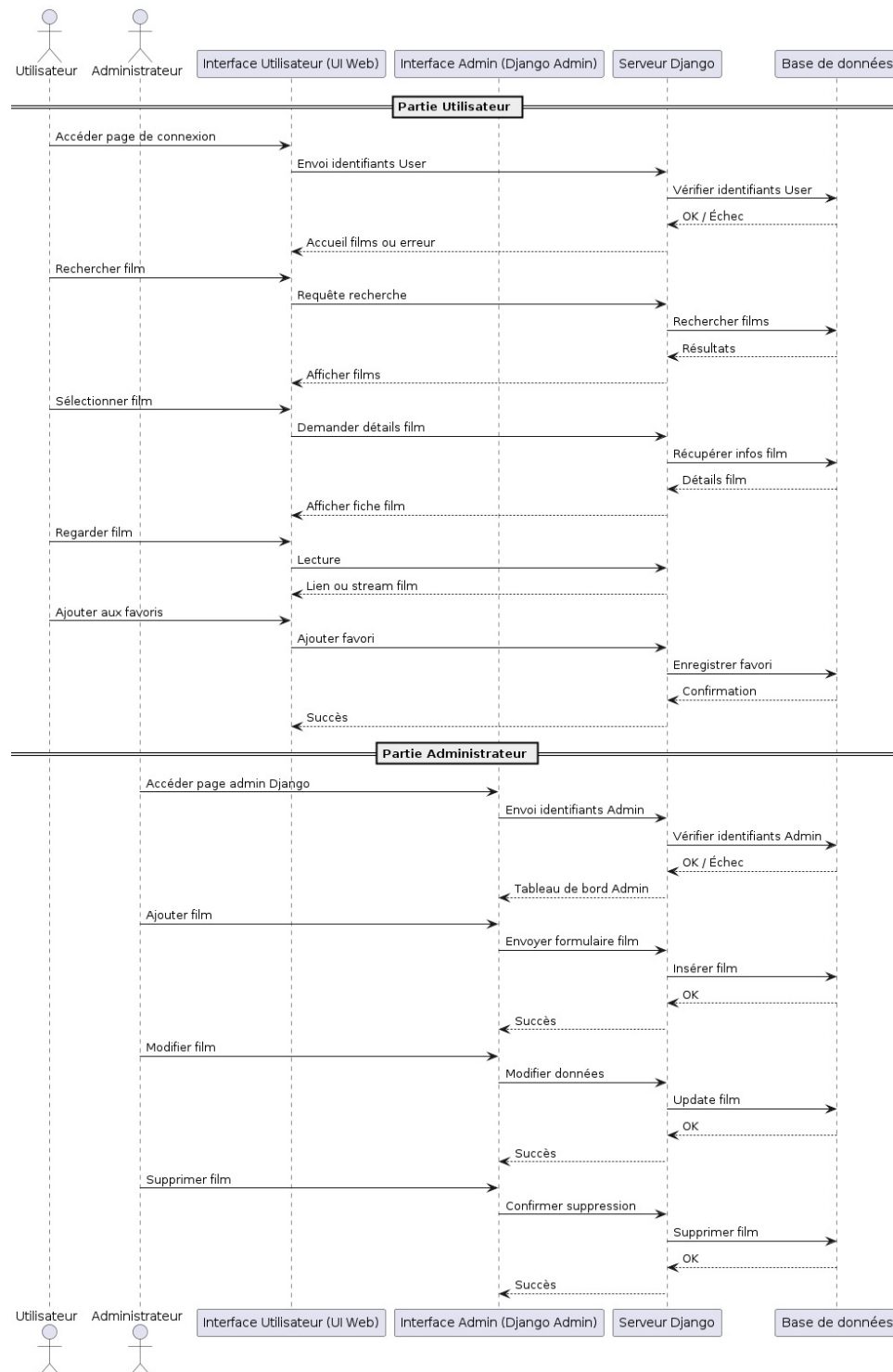


FIGURE 3.2 – Diagramme de Séquence

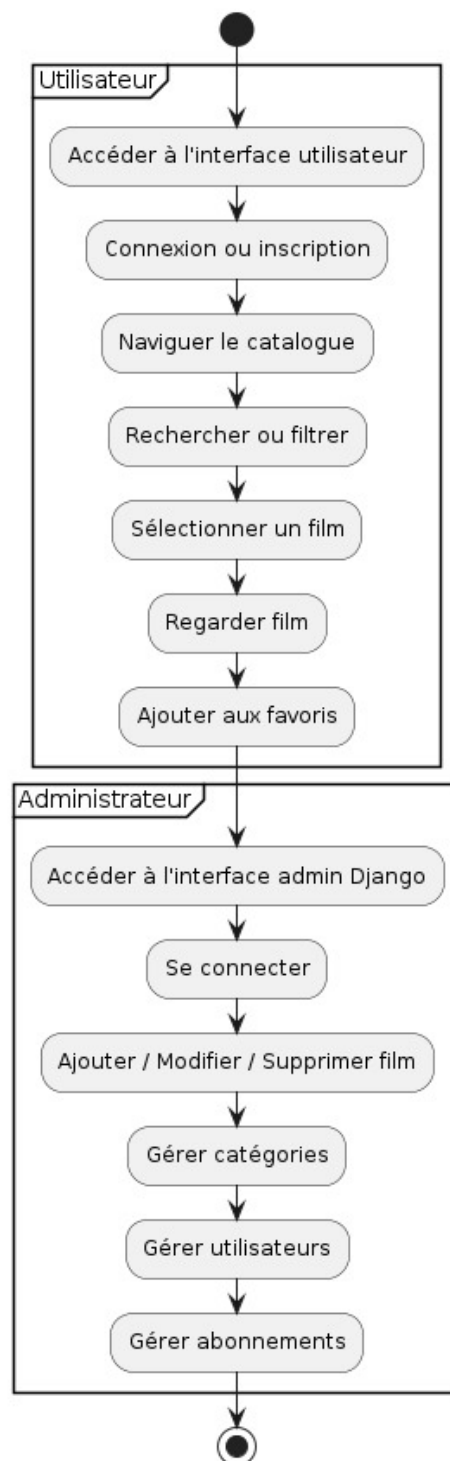


FIGURE 3.3 – Diagramme d'Activité

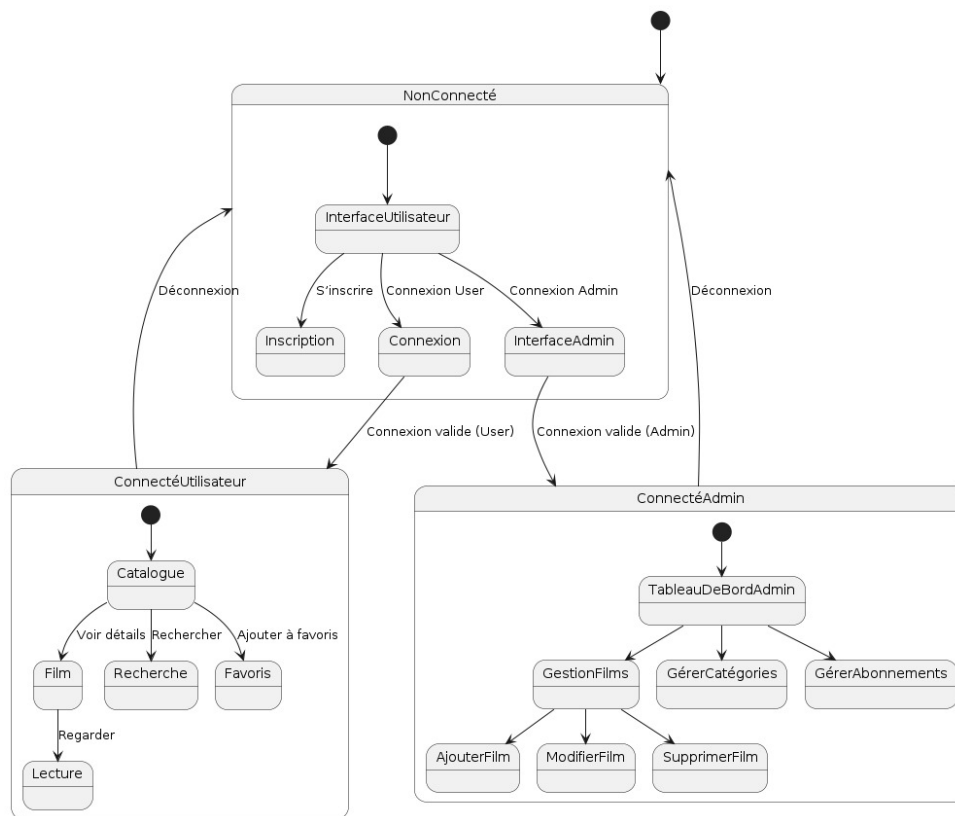


FIGURE 3.4 – Diagramme d'États-Transitions

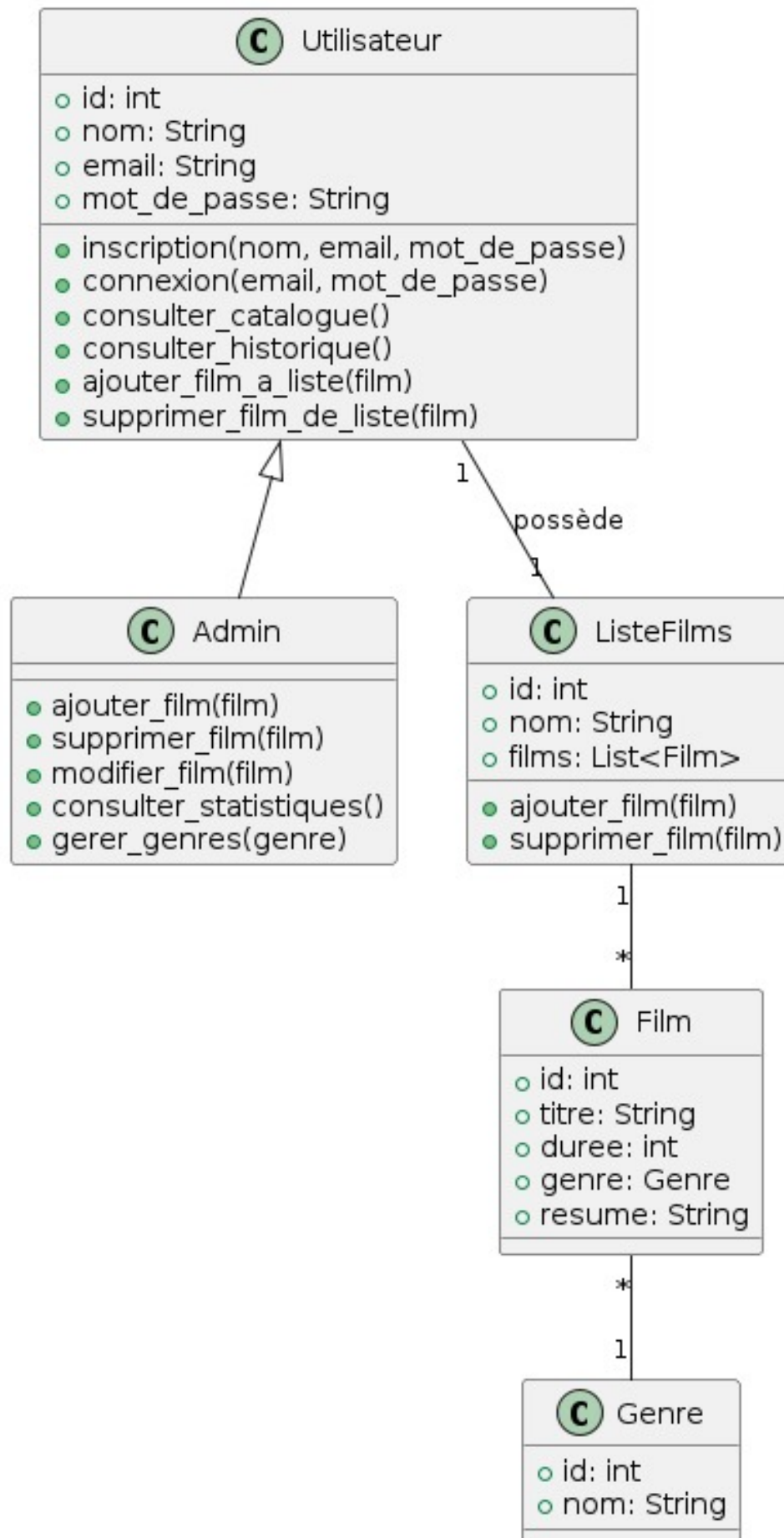


FIGURE 3.5 – Diagramme de Classes

C

hapitre

4

Réalisation du
système

Chapitre 4

Réalisation du système

4.1 Introduction

Dans ce chapitre, nous décrivons la phase de réalisation du système, qui constitue l'étape concrète de transformation des modèles conceptuels en une application fonctionnelle.

Le système développé est une plateforme de streaming en ligne inspirée de Netflix, offrant aux utilisateurs la possibilité de s'inscrire, de se connecter, de parcourir un catalogue de films, de rechercher et visionner des contenus, et de gérer leurs favoris.

Du côté administrateur, la plateforme permet de gérer les films, les catégories et les utilisateurs via une interface d'administration sécurisée.

4.2 Environnement de développement

4.2.1 Configuration matérielle

Le projet a été développé sur un ordinateur personnel avec processeur Intel Core i5, 8 Go de RAM, SSD de 256 Go, système d'exploitation Windows 10 ou Ubuntu 20.04 LTS, et une connexion Internet stable.

4.2.2 Outils et technologies utilisés

- **Django** (backend)
- **HTML5** / **CSS3** / **JavaScript** (frontend)
- **MySQL** (base de données)
- **Bootstrap** (design responsive)
- **Visual Studio Code**, **GitHub**, **Postman**, **XAMPP/WAMP**

4.3 Interfaces graphiques développées

4.3.1 Page d'accueil

Accueil attrayant avec logo, menu, carrousel de films récents et sections thématiques.

4.3.2 Connexion / inscription

Formulaires intuitifs pour l'authentification. JavaScript utilisé pour les validations côté client.

4.3.3 Catalogue des films

Affichage en cartes avec moteur de recherche par titre, genre, ou année.

4.3.4 Lecture vidéo

Lecteur intégré avec lecture/pause, volume, et informations du film.

4.3.5 Tableau de bord utilisateur

Historique de visionnage, favoris, informations personnelles, paramètres de profil.

4.3.6 Apports personnels

Renforcement des compétences full-stack, gestion de projet, autonomie dans la résolution d'erreurs, rigueur et collaboration via GitHub.

4.3.7 Perspectives et améliorations futures

- Ajout de commentaires et notes.
- Système de recommandation.
- Profils multiples.
- Optimisation du lecteur vidéo.
- Notifications et newsletter.
- Déploiement sur hébergement cloud.

4.4 Conclusion

Ce projet a permis la mise en œuvre d'une application de streaming complète, de la conception à la réalisation. Le respect des principes MVC, le design responsive, l'authentification sécurisée, et le lecteur vidéo sont les points forts de NETFILM.

Conclusion Générale

Le projet Netfilm avait pour objectif de développer une plateforme web de streaming de films, accessible à tous les utilisateurs via une interface claire, simple et fonctionnelle. Ce projet a été réalisé en utilisant le framework Django pour le développement back-end, et MySQL pour la gestion de la base de données. L'ensemble des fonctionnalités a été implémenté manuellement, sans recourir à l'interface d'administration native de Django, ce qui nous a permis de maîtriser en profondeur la structure du framework, les vues, les modèles, les formulaires et les templates. Tout au long du projet, nous avons mis en place les principales fonctionnalités d'un site de streaming, telles que : L'inscription et la connexion des utilisateurs, L'affichage dynamique des films selon les catégories, La consultation des détails de chaque film, Un tableau de bord personnalisé permettant aux utilisateurs de gérer leur profil, Le système CRUD (ajout, modification, suppression) des films via une interface web dédiée. Contrairement aux plateformes professionnelles, nous n'avons pas intégré de restrictions d'accès ou de système d'abonnement. Ainsi, tous les utilisateurs, une fois connectés, ont un accès libre au contenu disponible sur la plateforme. Ce choix nous a permis de nous concentrer sur la mise en place des bases fonctionnelles du site, tout en assurant une navigation fluide et cohérente. La réalisation de Netfilm nous a offert une opportunité concrète de mettre en œuvre nos connaissances en développement web full-stack, en particulier sur la gestion de base de données, les interactions utilisateur-serveur, et la création d'interfaces dynamiques. Elle a également renforcé notre capacité à développer un projet sans dépendre des outils préintégrés, en nous responsabilisant sur chaque aspect du développement. En conclusion, Netfilm constitue une première version fonctionnelle d'un site de streaming. Ce projet pourrait être amélioré à l'avenir par l'ajout de nouvelles fonctionnalités comme un moteur de recherche, une section de recommandations, un système de notation ou encore une gestion des accès avancée. Il représente une étape importante dans notre parcours de développeur, en alliant rigueur, autonomie et créativité.

Bibliographie et Nétographie

Bibliographie et Nétographie : Django Software Foundation. Django Documentation (version 4.x). Disponible en version imprimée ou sur le site officiel de Django.

2. Grinberg, Miguel. Flask Web Development with Python. O'Reilly Media, 2018. Bien que centré sur Flask, ce livre fournit de solides bases sur les architectures web de type MVC.

3. Freeman, Eric et Robson, Elisabeth. Head First HTML and CSS. O'Reilly Media, 2012. Un ouvrage visuel et interactif pour comprendre la structure des pages web.

4. Duckett, Jon. JavaScript and JQuery : Interactive Front-End Web Development. Wiley, 2014. Guide illustré pour apprendre JavaScript et jQuery en contexte web.

5. Beighley, Lynn. Learning MySQL : Get a Handle on Your Data. O'Reilly Media, 2007. Introduction claire à la gestion des bases de données relationnelles avec MySQL.

Documentation officielle de Django - <https://docs.djangoproject.com/en/stable/> Référence officielle pour comprendre et utiliser le framework Django.

Documentation officielle de MySQL - <https://dev.mysql.com/doc/> Guide technique complet sur la gestion de bases de données MySQL.

Mozilla Developer Network (MDN) - <https://developer.mozilla.org/> Ressource professionnelle pour HTML, CSS et JavaScript (normes web officielles).

SBootstrap (documentation officielle) - <https://getbootstrap.com/docs/> Si tu utilises Bootstrap pour le responsive design (optionnel).

W3C - World Wide Web Consortium - <https://www.w3.org/> Organisation de standardisation du web. Références officielles sur HTML, CSS et accessibilité.

Stack Overflow - <https://stackoverflow.com/> Plateforme professionnelle de questions/réponses pour résoudre des erreurs de code spécifiques.

Real Python – Tutoriels professionnels sur Django et Python - <https://realpython.com/> Articles rédigés par des experts, souvent utilisés pour des cas concrets en Django.

GitHub - <https://github.com/> Utilisé pour héberger et collaborer sur du code. Utile pour s'inspirer ou partager un projet comme NETFILM.