

Documentation projet

C#INEMA

Par Antoine DUTEYRAT et Samuel SIRVEN

Sommaire

Documentation projet C#INEMA	1
I – Contexte	2
II – Ergonomie et accessibilité	10
III – Diagramme de cas d'utilisation	13
III – Sketchs et Storyboards	19
IV – Wireframes	33
V – Mock-ups	38
VI – Diagramme de paquetage	43
VII – Diagramme de classe	46
VIII – Diagramme de séquence	57
IX – Explications des patrons	60
X – Annexe	65

I – Contexte

Partie 1 : L'introduction

Vous êtes un magnat du 7^e art ou vous souhaitez le découvrir ? Alors, l'application C#INEMA est faite pour vous !

Finie l'époque où vous deviez vous renseigner sur plusieurs sites différents, vous perdre et ne plus savoir où donner de la tête. Grâce à l'application C#inema tout vous sera mis à disposition de façon simple et efficace !

Vous souhaitez obtenir des informations détaillées sur les films et séries que vous aimez ou encore sur vos acteurs préférés ? Tout ceci sera désormais disponible au même endroit ! Vous en avez assez de devoir changer d'application ou de site pour pouvoir regarder en streaming ce que vous voulez ?

Le problème est réglé, nous vous proposons de regarder tous les films et séries du moment sur une seule et même plateforme, dès lors que vous possédez un compte et un abonnement sur un des sites concernés.

Si ça ne suffit pas à vous convaincre, vous pourrez même noter et commenter les différents films et séries pour y laisser une empreinte ou un conseil aux autres utilisateurs.

Bien sûr, nous avons aussi pensé à vos enfants avec un mode famille qui bloquera l'accès à certains films et séries considérés comme non-appropriés pour leur âge.

Pour résumer, cette application vous offre la possibilité de rechercher, noter et visionner vos films et séries préférés, le tout en un seul et même endroit.

Partie 2 : Les Personas



<i>Prénom :</i>	Ginette
<i>Nom :</i>	Dauchamp
<i>Âge :</i>	84 ans
<i>Profession :</i>	Retraitée, travaillait en tant que contrôleuse de titre de transport SNCF (pfffff)
<i>Couleur préférée :</i>	Dark Salmon
<i>Plat préféré :</i>	Le bœuf bourguignon au vin rouge de campagne
<i>Situation :</i>	Seule et délaissée, veuve depuis la semaine dernière
<i>Revenus :</i>	31 200€
<i>Profil technique :</i>	Intéressée par le cinéma, aussi bien récent que celui du siècle dernier. Elle utilise assez peu son ordinateur (environ 1h/jour) pour Facebook (évidemment), elle utilise un LENOVO sous win10 et un smartphone Android de 2013.
<i>Sites préférés :</i>	Facebook Marmiton Wikipédia

Ginette cherche un logiciel simple pour se renseigner et voir des films qui ont marqué son enfance.

Elle est à la retraite et s'ennuie comme pas possible dans sa maison sur la côte ouest (sur le littoral) puisque ses enfants et petits-enfants ne font aucunement attention à elle.

Le seul contact humain qu'elle trouve dans la journée est celui du facteur ou du buraliste.

Elle est donc prise de nostalgie et veut s'occuper sans chercher à savoir comment se créer un compte Netflix ou OCS (elle utilise celui de ses enfants).

“À mon époque, les films étaient de véritables œuvres d’art, cette partie de ma vie me manque.”



<i>Prénom :</i>	Chloé
<i>Nom :</i>	Du Bois
<i>Âge :</i>	32 ans
<i>Profession :</i>	Directrice marketing de H&M
<i>Couleur préférée :</i>	Rouge bordeaux
<i>Plat préféré :</i>	Tagliatelles au pesto avec escalope milanaise
<i>Situation :</i>	Célibataire, 2 enfants insupportables sous le bras
<i>Revenus :</i>	45 000€
<i>Profil technique :</i>	À l'aise avec les nouvelles technologies, Ordinateurs portable Mac, smartphone iOS. Utilisation des deux environ 10h/j pour le travail et le loisir.
<i>Site préféré :</i>	H&M Rotten Tomatoes OCS

Chloé est frustrée de ne jamais trouver les acteurs en lien avec le film qu'elle cherche.

Avec le peu de temps qu'elle a dans la semaine elle aime pouvoir avoir accès à ce qu'elle veut rapidement.

Son ex étant confiné à l'autre bout de la France elle doit garder les deux enfants pendant encore au moins un mois. Elle espère secrètement qu'ils partent au plus vite pour avoir la paix à la maison.

Ses enfants n'ont absolument pas les mêmes goûts cinématographiques qu'elle et sont encore jeunes. Il est donc compliqué de pouvoir regarder ce qu'elle veut sans que les enfants soient à côté ou puissent accéder aux films qui leur sont interdits.

“Je n’arrive jamais à trouver des films en lien avec les thèmes que je choisis.”



Prénom :	Arthur
Nom :	Bontuyau
Âge :	15 ans
Profession :	Jeune lycéen
Couleur préférée :	Lime green
Plat préféré :	Burger King et O'Tacos
Situation :	Célibataire car très timide et peu drôle
Revenus :	360€
Profil technique :	Intéressé de très loin par le cinéma, seulement le récent. Il utilise assez souvent son ordinateur (environ 7h/jour) pour traîner sur CS:GO où Twitter ou encore LoL, il utilise un Alienware fixe sous win10 et un smartphone ROGphone.
Sites préférés :	Twitter Netflix La chaîne youtube de Lama Faché

Arthur cherche à se faire des amis en passant par les films pour surpasser sa timidité.

Il est en seconde générale et a donc très peu de travail à la maison, étant très timide, il n'a pas d'ami et se retrouve par-dessus le marché confiné le 5 avril 2021.

Il veut se sortir de son addiction à LoL qui ronge sa santé mentale et s'intéresser à quelque chose de plus sains dont il pourrait parler avec ses parents.

Il trouve donc l'idée du Cinéma et se met à regarder toutes sortes de films pour trouver son film préféré (pour ensuite le défendre corps et âme sur Twitter).

"Je suis un peu trop timide, peut-être qu'être calé en cinéma me permettrait des relations plus faciles avec de potentiels amis ?"

Partie 3 : Les User-Stories

Ginette Dauchamp

Une douce journée d'automne, il est 16 h et Ginette regarde un téléfilm de très mauvaise qualité en sirotant un thé vert.

Soudain, il lui prend l'envie de regarder de nouveau "Boulevard du crépuscule", film qui l'a accompagné durant sa préadolescence. Elle se décide donc à dégainer son ordinateur portable.

Elle ouvre l'application C#INEMA et écrit dans la barre de recherche le titre du film. Apparaît alors sous ses yeux le film qu'elle recherchait, elle clique dessus et tombe sur la page du film, elle s'empresse alors de cliquer sur un des liens de streaming et elle est aussitôt renvoyée sur la page Netflix du film. C'est une belle fin d'après-midi qui s'annonce pour notre mamie silencieuse...

Notre adorable mamie a toujours eu du mal à se souvenir sur quelles plateformes ses petits-fils lui ont prêté un compte. Donc à chaque film qu'elle veut regarder, elle se retrouve à perdre du temps à toujours cliquer sur tous les liens, jusqu'à avoir trouvé le bon.

Mais elle a réussi à se souvenir que sur sa nouvelle application C#INEMA, elle pouvait cocher les plateformes auxquelles elle était abonnée. Ce qu'elle s'empresse de faire, pour plus tard se rendre compte du temps qu'elle a gagné à avoir directement ses comptes mis en avant lors du choix de ses films.

Chloé Du Bois

Profitant des quelques heures de tranquillité qu'elle peut obtenir le week-end, elle se rue sur le web pour chercher des informations sur certains films ou bien ses acteurs préférés. Malheureusement, elle ne trouve pas les informations qu'elle souhaite et se redirige donc sur sa nouvelle application C#INEMA.

Elle se met donc à utiliser la barre de recherche en sélectionnant le filtre "Thème", pour rechercher tous les films d'action. Une fois dans cette section, elle peut maintenant naviguer et rechercher le film qu'elle souhaite, pour en obtenir plus de détails avec son synopsis ou bien les acteurs qui sont dedans.

Chloé vient de finir de visionner la série *Breaking Bad* qu'elle a beaucoup appréciée, mais ne veut pas que ses enfants puissent la regarder. Elle se rappelle donc qu'un mode Famille est disponible sur l'application. Elle décide donc de se déconnecter, pour se connecter sur le compte des enfants. Une fois connectée, elle va donc modifier les préférences utilisateurs pour l'activer. Pour en être sûr, elle tape le nom de la série dans la barre de recherche et miracle, elle s'aperçoit qu'aucun résultat n'apparaît comme si la série n'existe pas.

Arthur Bontuyeau

Après avoir fini sa très dure journée de cours de 4h en distanciel, Arthur décide d'aller prendre un peu de bons temps. Après avoir passé beaucoup trop de temps sur la chaîne incroyablement nulle de Lame Fâché, il lui vient soudain l'envie de retrouver l'actrice de Harley Queen dans *Suicide Squad*, mais n'arrive vraiment pas à la retrouver.

Il décide donc d'aller sur sa nouvelle application C#INEMA. Il tape le nom du film et arrive sur sa page.

Et là, tout content, Arthur aperçoit le nom de l'actrice qui interprète le personnage parmi les acteurs mentionnés et peut même accéder à sa page en cliquant dessus.

Il décide avant de fermer son ordinateur pour la journée, de laisser son avis sur ce film, il clique donc sur le « + ». Une petite fenêtre s'ouvre alors, Arthur écrit quelques mots : « Une daube insipide », laisse une étoile et valide, fier d'avoir rendu service à tous les autres utilisateurs, il peut s'endormir paisiblement.

Partie 4 : Les fonctionnalités principales

L'application C#INEMA vous permettra de rechercher films et séries pour les visionner, mais aussi de pouvoir parcourir la section acteurs, ou bien laisser une note et un commentaire. Regardons maintenant plus en détail, ce qu'il vous sera possible de faire avec cette application...

Recherche (cf. user-story n°1.1, 2.1) :

La recherche se divisera en deux parties principales :

Une première partie où vous pouvez rechercher directement un acteur, ou bien un film à l'aide de deux filtres différents. Une fois la recherche effectuée, l'application vous mènera directement sur la page de l'acteur, ou du film concerné.

Si votre recherche n'est pas assez précise, ou s'il y a plusieurs résultats, il vous sera proposé tous les films et acteurs qui comportent votre requête, pour vous laisser le choix final. Par exemple, si vous tapez *Star Wars*, les six films vous seront proposés (les trois derniers n'existent pas).

Pour ce qui est de la seconde partie, vous pourrez afficher tous les films ou acteurs disponibles, sur notre application.

Il y aura également la possibilité d'afficher tous les thèmes de films disponibles, pour avoir une recherche un peu plus affinée et pour ne voir que les films de cette catégorie.

Les Films et séries (cf. user-story n°3.1) :

Chaque film ou série vous donnera accès à un certain nombre d'informations utiles, comme son titre, son synopsis, ainsi que les acteurs les plus importants. Vous pourrez donc obtenir toutes ses informations et naviguer sur les différentes pages des acteurs, à partir d'un film ou d'une série.

Vous pourrez aussi le regarder en cliquant sur un lien. Les pages de streaming Netflix, Prime Vidéo, ou OCS, vous permettront de vous y connecter et de directement le visionner.

Pour finir, il vous sera possible de l'ajouter à votre liste d'envie, mais aussi de laisser un avis, comprenant au moins, une note de zéro à cinq étoiles et si vous le voulez, vous pourrez ajouter un commentaire pour exprimer plus en détail ce que vous avez aimé ou non, sur ce film ou cette série.

Les Acteurs :

Il vous sera également possible d'obtenir les informations les plus utiles sur l'acteur de votre choix. Comme son nom, son prénom, sa date de naissance, ainsi qu'une brève biographie. Bien sûr, tous les films et séries dans lequel cet acteur est présent, seront eux aussi accessible depuis la page de celui-ci.

Les Films/Séries/Acteurs récemment consultés :

Si vous possédez un compte et que vous l'utilisez, il vous sera possible de retrouver les pages récemment consultées des différents films, séries et acteurs.

Ceci vous aidera si vous aussi vous avez quelques problèmes de mémoire et que vous ne vous souvenez plus de ce que vous avez regardé hier.

Le compte (cf. user-story n°1.2, 2.2) :

Pour finir, vous pourrez (comme il a été mentionné ci-dessus) avoir un compte, pour pouvoir laisser des notes, des commentaires, ainsi qu'avoir accès à la liste des pages récemment consultées. Il vous sera donc possible de choisir un pseudo, ainsi que de personnaliser votre photo de profil. Un mode "famille" est aussi disponible, vous permettant de bloquer l'accès aux films et séries non appropriés à vos enfants.

Vous pourrez aussi cocher la liste des plateformes de streaming auxquelles vous êtes abonnées, pour qu'elles soient mises en avant lors de la sélection du lien de streaming, si celui-ci est disponible.

Il vous sera également possible de consulter toutes les notes que vous avez laissées aux différents films et séries, ou bien jeter un coup d'œil à votre liste d'envies à partir de cette page.

II – Ergonomie et accessibilité

Ce sont deux notions importantes dans un projet tel qu'une application : l'ergonomie et l'accessibilité.

Elles permettent à l'application d'inclure dans son fonctionnement et son aspect toutes les personnes dont les capacités cognitives sont réduites voire inexistantes.

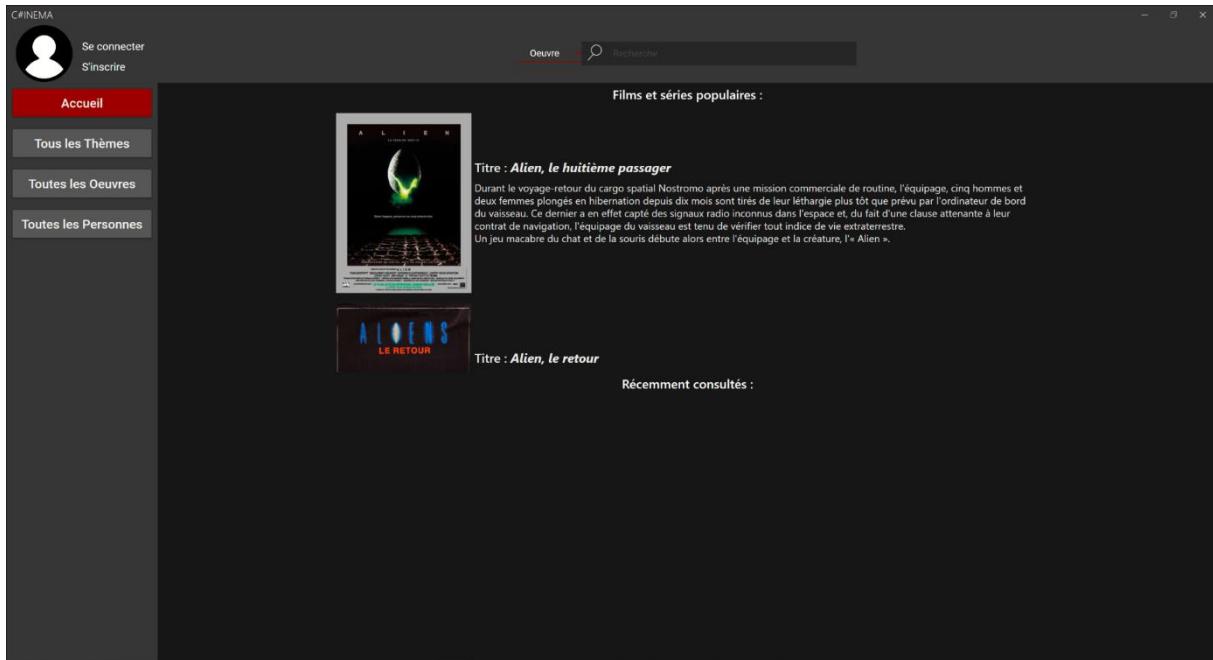
Dans un premier temps, parlons de l'ergonomie. C'est un aspect à ne pas négliger, elle permet notamment de rendre l'application facile d'usage et ce pour n'importe qui.

Dans l'application, plusieurs choix ont été effectués selon ce principe :

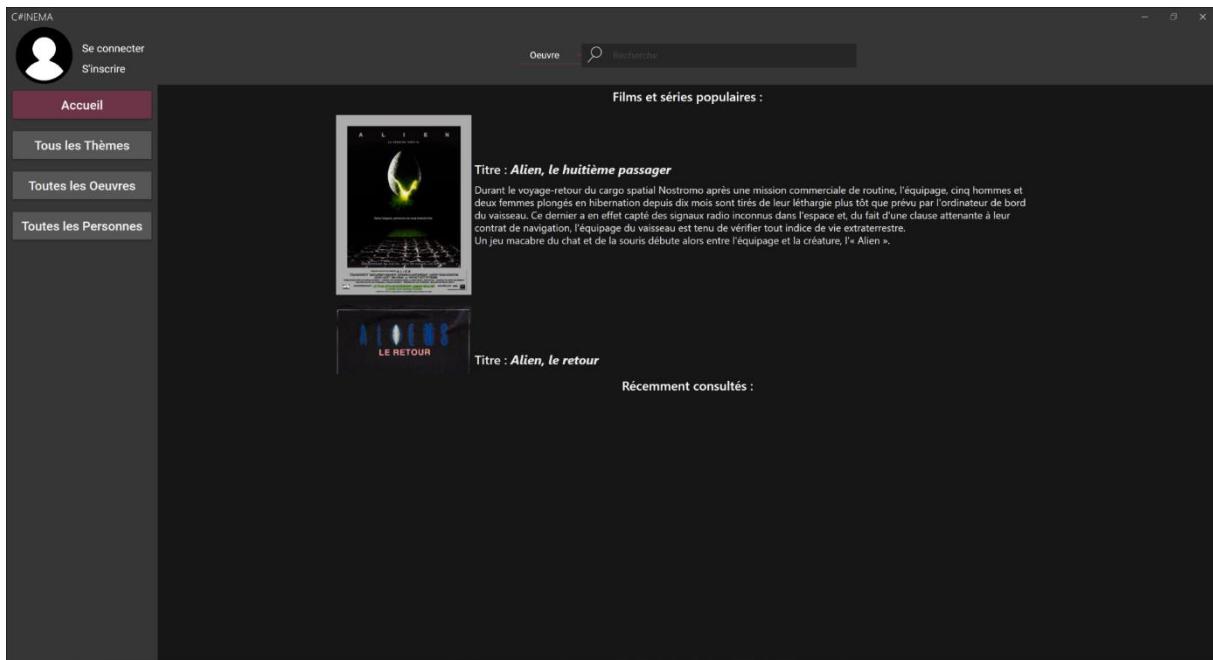
- Un thème sombre qui met en valeur les affiches des œuvres publiées sur l'application, ainsi que tous les textes qu'on peut y retrouver.
- Des boutons larges, dont les intitulés sont clairs et explicites pour ne pas perdre les utilisateurs lors de leur utilisation. Tous les boutons ayant des utilisations antinomiques sont séparés par de larges distances.
- Des positions de bouton efficientes et suivants quelques principes de base (pas de bouton près du bouton 'Fermer' ou encore boutons de navigation sur les côtés)
- Des logos et icônes explicites servent aussi dans l'interface Administrateur.
- L'incitation : Nous avons rendu les boutons importants colorés pour inciter l'utilisateur à cliquer dessus plutôt que sur d'autres.
De plus, nous avons, pour la plupart des champs d'écriture (ou ComboBox) accessibles aux utilisateurs lambdas, inséré des labels par défaut pour guider l'utilisateur sur la démarche à suivre.
- Nous avons également considéré important de renvoyer un feedback immédiat lors d'une mauvaise utilisation de l'application (principalement en Administrateur). Ceci a été fait par le biais de la fonction ShowMessageAsync().
- Nous avons également optimisé les temps de chargement de façon à ne pas laisser d'écran vide, pouvant laisser l'utilisateur se distraire.
- Nous avons également, pour privilégier la lisibilité, préféré écrire en minuscules et faire des paragraphes clairs.

En second lieu, on trouve l'accessibilité, cet aspect n'a pas été mis de côté non plus dans l'application.

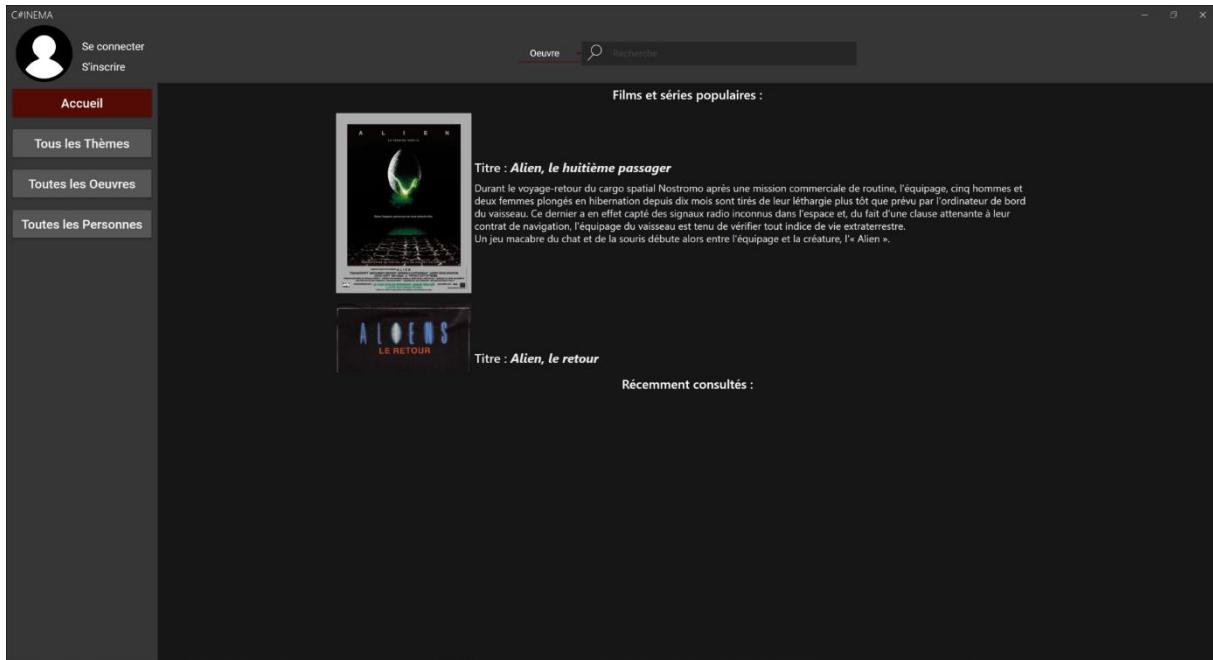
En effet, le choix des couleurs gris foncé et rouge sang pour accentuer les points clés de la fenêtre permet à des personnes daltoniennes de percevoir le contraste entre les éléments, ce choix permet également de préserver les yeux de l'utilisateur de douleurs ou de fatigue lors d'un usage prolongé.



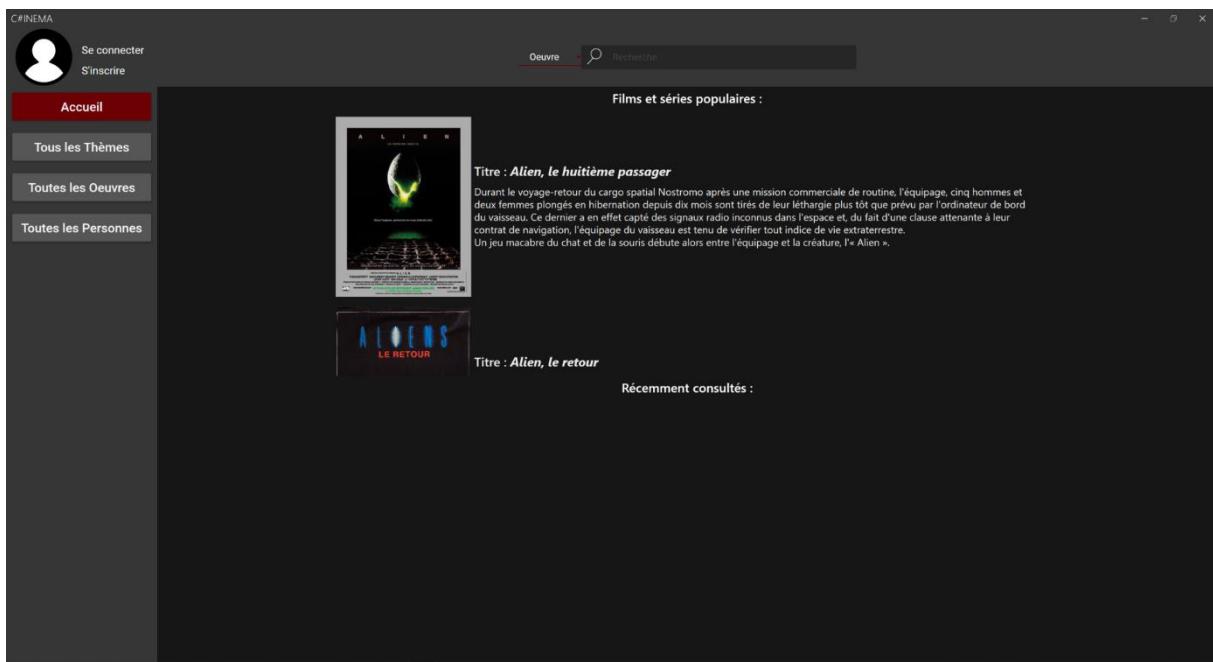
Vue de base de l'application



Vue d'un deutéranope de l'application



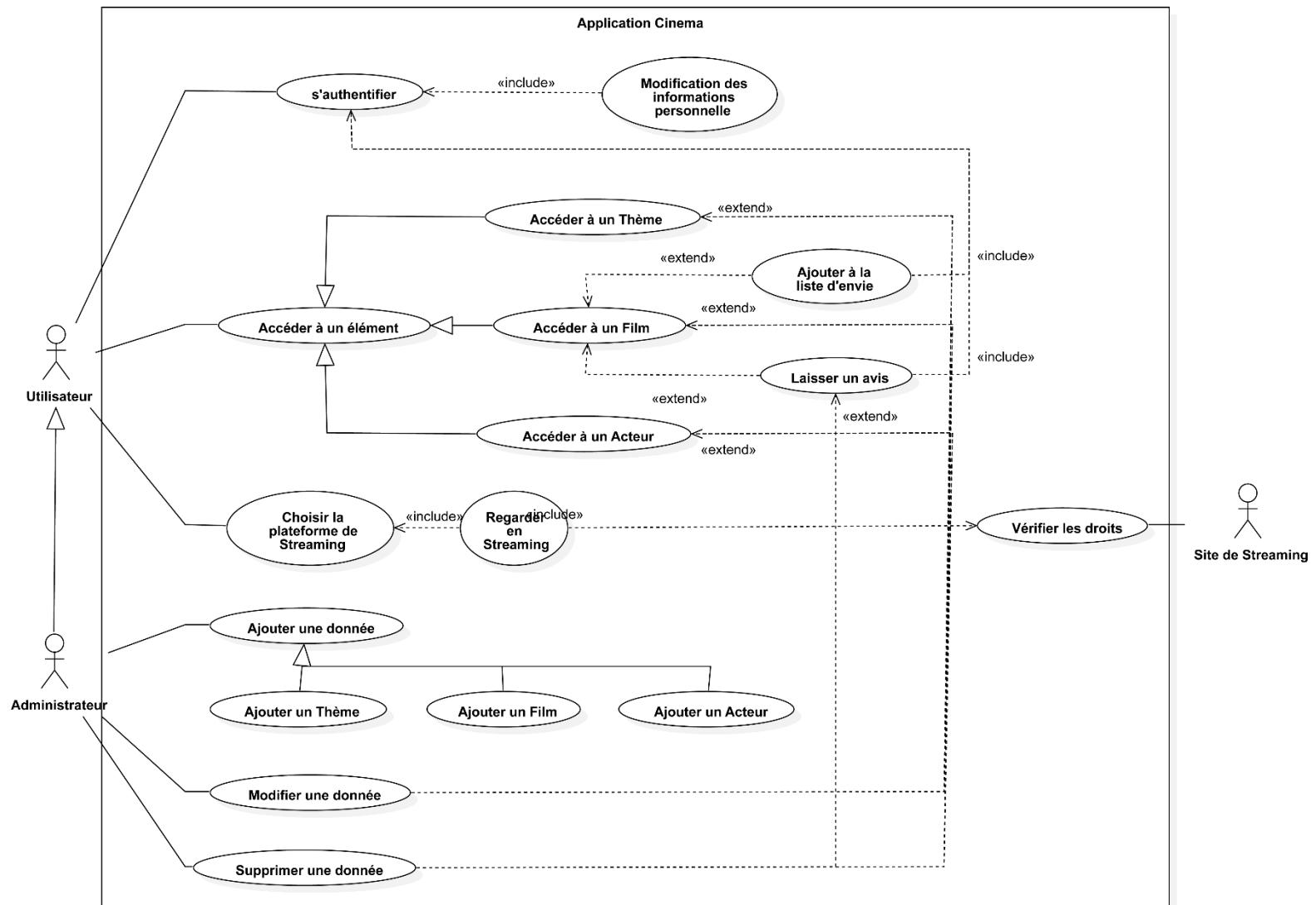
Vue d'un protanope de l'application



Vue d'un tritanope de l'application

Ensuite, les boutons sont larges, la police est assez grande et ce tout permet également aux personnes malvoyantes d'utiliser l'application dans de meilleures conditions.
Enfin, le choix de la police d'écriture s'est porté sur « David », une police claire et lisible.
Cette police possède un espace assez important entre ses lettres pour une bonne dissociation lors de la lecture ainsi que des formes claires comme on a l'habitude d'en voir au quotidien (police classique dans l'utilisation d'un ordinateur).

III – Diagramme de cas d'utilisation



Cas « S'authentifier »

Nom :	S'authentifier
Objectif :	Se connecter
Acteurs principaux :	Utilisateur/Administrateur
Conditions initiales :	<ul style="list-style-type: none">- L'utilisateur n'est pas connecté- L'utilisateur possède un compte
Scénario d'utilisation :	<ul style="list-style-type: none">- L'utilisateur rentre ses identifiants- Le système vérifie la validité des identifiants (sinon conditions de fin 2)- Les identifiants sont validés (conditions de fin 1)
Conditions de fin :	<ol style="list-style-type: none">1) L'utilisateur est connecté avec succès2) L'utilisateur n'est pas connecté

*Le système propose bien entendu la possibilité de créer un compte si l'utilisateur n'en possède pas un

Cas « Modification des informations personnelles »

Nom :	Modification des informations personnelles
Objectif :	Personnaliser ses préférences utilisateur
Acteurs principaux :	Utilisateur/Administrateur
Conditions initiales :	<ul style="list-style-type: none">- Être connecté à un compte
Scénario d'utilisation :	<ul style="list-style-type: none">- L'utilisateur modifie ses préférences utilisateur (pseudo/mot de passe /mode famille, etc)- L'utilisateur sauvegarde ses modifications en entrant son mot de passe- Le système vérifie la validité du mot de passe (sinon conditions de fin 2)- Le mot de passe est validé (conditions de fin 1)
Conditions de fin :	<ol style="list-style-type: none">1) Les modifications sont sauvegardées2) Les modifications ne sont pas sauvegardées

Cas « Accéder à un Thème »

Nom : Accéder à un Thème
Objectif : Obtenir un thème
Acteurs principaux : Utilisateur/Administrateur

Conditions initiales : - Aucune

Scénario d'utilisation : - L'utilisateur choisi un thème

Conditions de fin : - Le thème est fourni

Cas « Accéder à un Acteur »

Nom : Accéder à un Acteur
Objectif : Obtenir un acteur
Acteurs principaux : Utilisateur/Administrateur

Conditions initiales : - Aucune

Scénario d'utilisation : - L'utilisateur choisit un acteur

Conditions de fin : - L'acteur est fourni

*Il sera aussi possible d'y accéder avec une recherche

Cas « Accéder à un Film »

Nom : Accéder à un Film
Objectif : Obtenir un film/série
Acteurs principaux : Utilisateur/Administrateur

Conditions initiales : - Aucune

Scénario d'utilisation : - L'utilisateur choisi un film

Conditions de fin : - Le film est fourni

*Il sera aussi possible d'y accéder via une recherche

Cas « Ajouter à la liste d'envie »

Nom :	Ajouter à la liste d'envie
Objectif :	Ajouter un(e) film/série à la liste d'envie de l'utilisateur
Acteurs principaux :	Utilisateur/Administrateur
Conditions initiales :	<ul style="list-style-type: none">- L'utilisateur doit être sur la page d'un film- L'utilisateur doit être connecté
Scénario d'utilisation :	<ul style="list-style-type: none">- L'utilisateur ajoute le film/série souhaité à sa liste d'envie
Conditions de fin :	<ul style="list-style-type: none">- Le film/série est ajouté à la liste d'envie

Cas « Laisser un avis »

Nom :	Laisser un avis
Objectif :	Permettre à un utilisateur de laisser un avis sur le film/série de son choix
Acteurs principaux :	Utilisateur/Administrateur
Conditions initiales :	<ul style="list-style-type: none">- L'utilisateur doit être sur la page d'un film- L'utilisateur doit être connecté
Scénario d'utilisation :	<ul style="list-style-type: none">- L'utilisateur laisse un avis sur le film/série de son choix
Conditions de fin :	<ul style="list-style-type: none">- L'avis est publié

Cas « Choisir la plateforme de Streaming »

Nom :	Choisir la plateforme de Streaming
Objectif :	Choisir la plateforme de Streaming sur laquelle l'utilisateur va regarder son film/série
Acteurs principaux :	Utilisateur/Administrateur
Conditions initiales :	<ul style="list-style-type: none">- L'utilisateur doit être sur la page d'un film
Scénario d'utilisation :	<ul style="list-style-type: none">- L'utilisateur choisit un site parmi la liste des plateformes proposées
Conditions de fin :	<ul style="list-style-type: none">- L'utilisateur est redirigé vers la plateforme choisie

Cas « Vérifier les droits »

Nom :	Vérifier les droits
Objectif :	Vérifier les identifiants de l'utilisateur sur une plateforme de streaming
Acteurs principaux :	Utilisateur/Administrateur
Acteur secondaire :	Site de Streaming
Conditions initiales :	<ul style="list-style-type: none">- L'utilisateur doit être sur la page d'un film- L'utilisateur doit avoir choisi un site de Streaming
Scénario d'utilisation :	<ul style="list-style-type: none">- L'utilisateur rentre les identifiants de la plateforme de Streaming choisi- Le site de Streaming vérifie la validité des identifiants (si valide conditions de fin 1)- Les identifiants ne sont pas valides (conditions de fin 2)
Conditions de fin :	<ol style="list-style-type: none">1) L'utilisateur est connecté à la plateforme2) L'utilisateur n'est pas connecté à la plateforme

Cas « Regarder en Streaming »

Nom :	Regarder en Streaming
Objectif :	Visionner le film/série de son choix sur une plateforme de streaming choisie
Acteurs principaux :	Utilisateur/Administrateur
Conditions initiales :	<ul style="list-style-type: none">- L'utilisateur doit être sur la page d'un film- L'utilisateur doit avoir choisi un site de Streaming- L'utilisateur est connecté à la plateforme de Streaming
Scénario d'utilisation :	<ul style="list-style-type: none">- L'utilisateur démarre son film/série
Conditions de fin :	<ul style="list-style-type: none">- Le film/série est en cours

Cas « Ajouter un Thème »

Nom :	Ajouter un Thème
Objectif :	Permettre d'ajouter un nouveau thème
Acteurs principaux :	Administrateur
Conditions initiales :	<ul style="list-style-type: none">- L'utilisateur doit être connecté en tant qu'administrateur
Scénario d'utilisation :	<ul style="list-style-type: none">- L'utilisateur rentre les informations du nouveau thème
Conditions de fin :	<ul style="list-style-type: none">- le nouveau thème est ajouté

Cas « Ajouter un Film »

Nom :	Ajouter un Film
Objectif :	Permettre d'ajouter un nouveau film
Acteurs principaux :	Administrateur
Conditions initiales :	- L'utilisateur doit être connecté en tant qu'administrateur
Scénario d'utilisation :	- L'utilisateur rentre les informations du nouveau film
Conditions de fin :	- le nouveau film est ajouté

Cas « Ajouter un Acteur »

Nom :	Ajouter un Acteur
Objectif :	Permettre d'ajouter un nouvel acteur
Acteurs principaux :	Administrateur
Conditions initiales :	- L'utilisateur doit être connecté en tant qu'administrateur
Scénario d'utilisation :	- L'utilisateur rentre les informations du nouvel acteur
Conditions de fin :	- le nouvel acteur est ajouté

Cas « Modifier une donnée »

Nom :	Modifier une donnée
Objectif :	Permettre de modifier un thème/film/acteur
Acteurs principaux :	Administrateur
Conditions initiales :	- L'utilisateur doit être connecté en tant qu'administrateur
Scénario d'utilisation :	- L'utilisateur modifie les informations d'un thème/film/acteur
Conditions de fin :	- Le thème/film/acteur est modifié

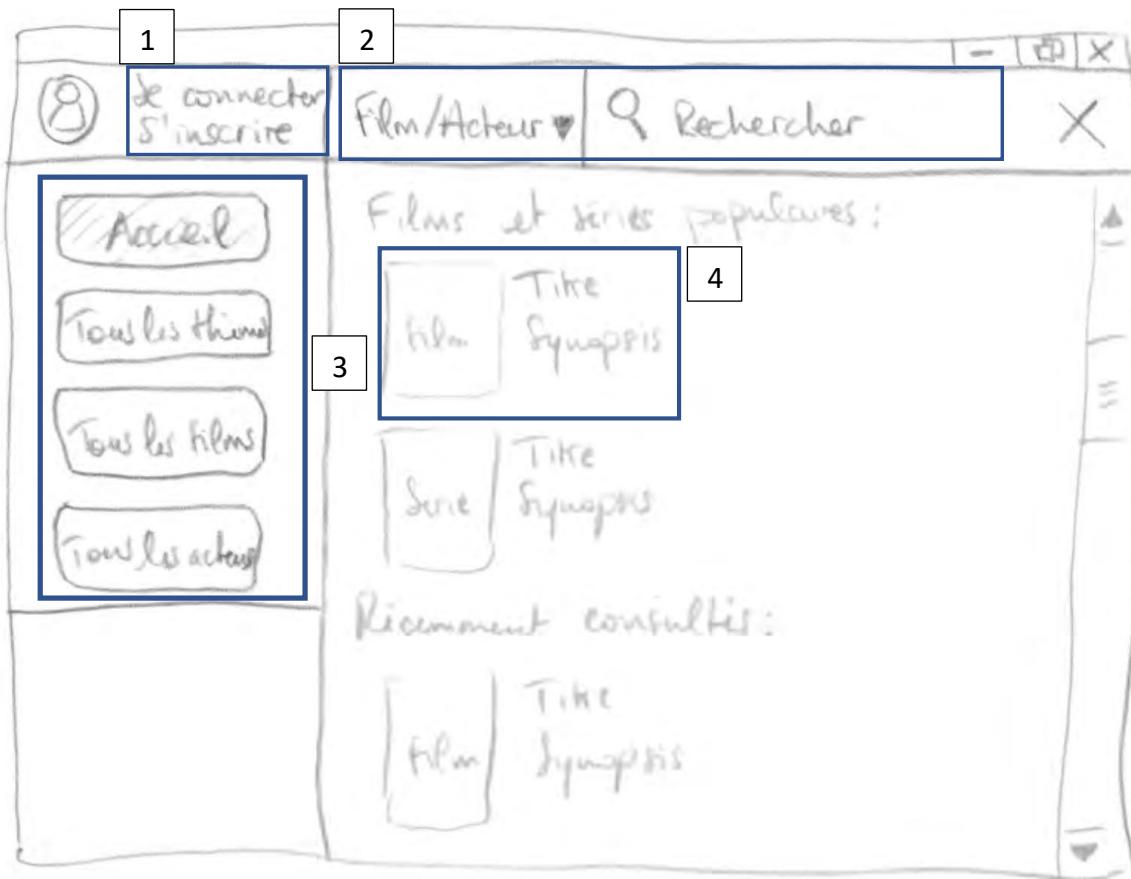
Cas « Supprimer une donnée »

Nom :	Supprimer
Objectif :	Permettre de supprimer un thème/film/acteur
Acteurs principaux :	Administrateur
Conditions initiales :	- L'utilisateur doit être connecté en tant qu'administrateur
Scénario d'utilisation :	- L'utilisateur supprime un thème/film/acteur
Conditions de fin :	- Le thème/film/acteur est supprimé

III – Sketchs et Storyboards

Partie 1 : Sketchs

Page d'accueil



- 1) Ces boutons servent à l'utilisateur pour se connecter ou s'inscrire sur l'application.
- 2) Cette combo-box a pour fonction d'orienter la recherche de l'utilisateur, il sélectionne soit film soit acteur puis écrit sa recherche dans la barre et tape entrée. Les résultats qui lui seront renvoyés comporteront la chaîne de caractères de sa requête.
- 3) Ces quatre boutons servent à accéder aux quatre pages principales (vues) de l'application comme l'indiquent leurs libellés.
- 4) L'utilisateur a également la possibilité de cliquer sur la couverture, le titre ou le synopsis du film pour se rendre sur la page de celui-ci.

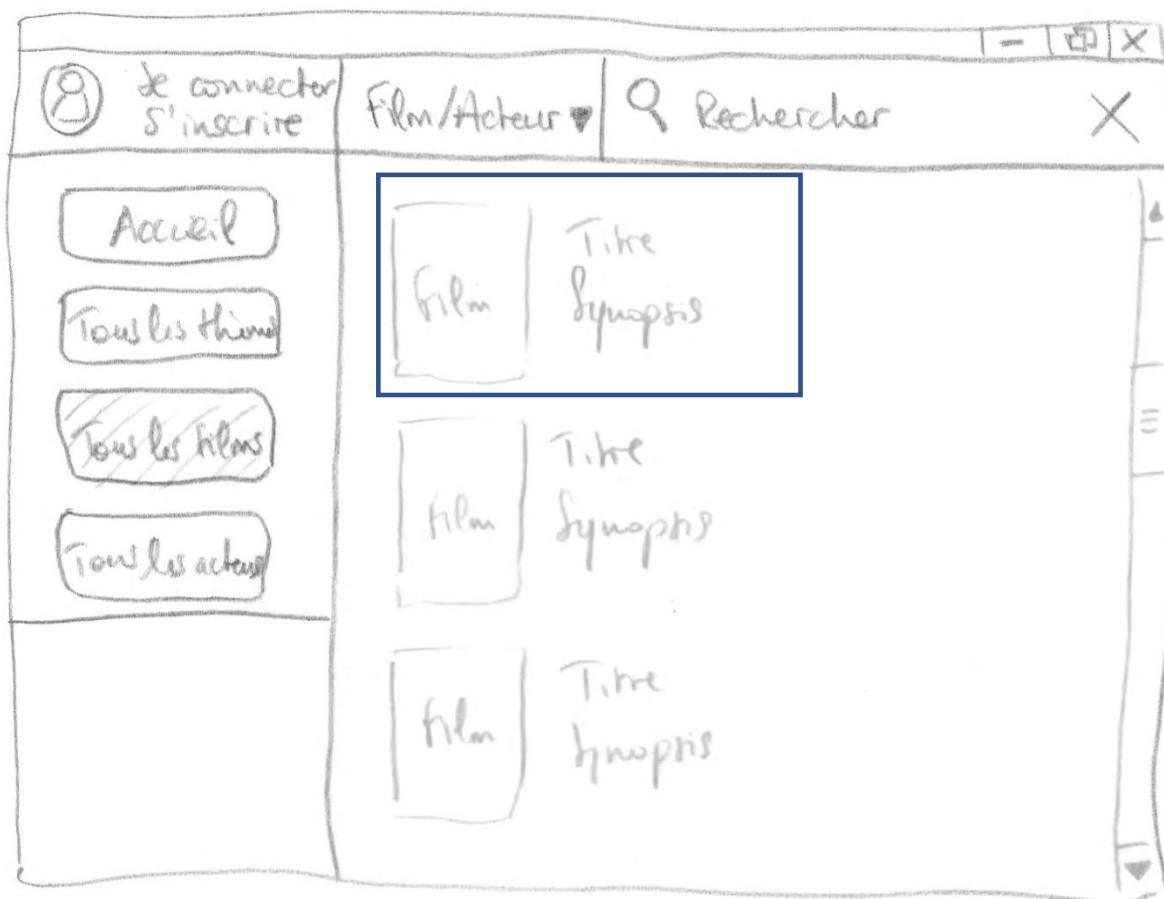
Cette vue est celle d'un utilisateur qui démarre l'application.

Page contenant tous les thèmes



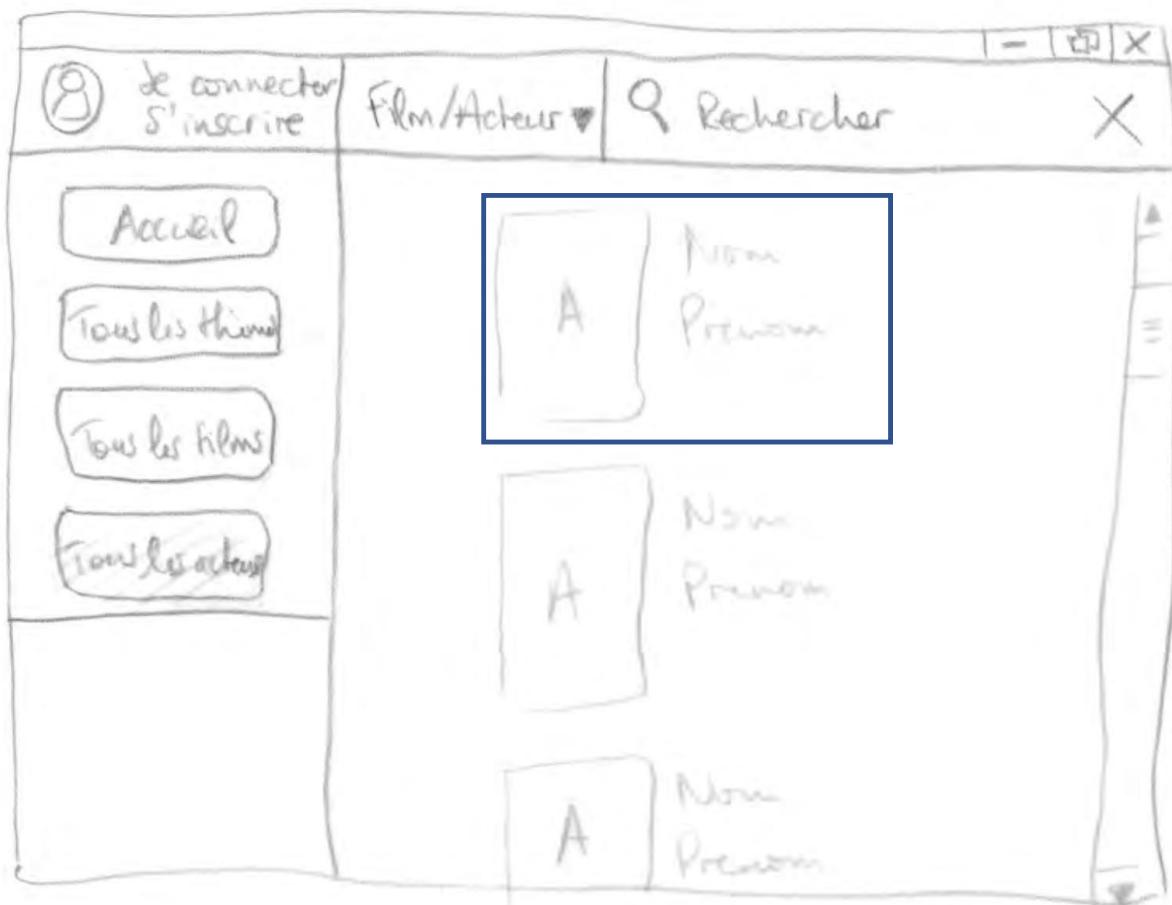
Ici l'utilisateur a les mêmes possibilités qu'à l'accueil, à l'exception des boutons « thème » sur lesquels il peut cliquer pour se rendre sur la page d'un thème, sur laquelle seront répertoriés tous les films qui y correspondent.

Page contenant tous les films



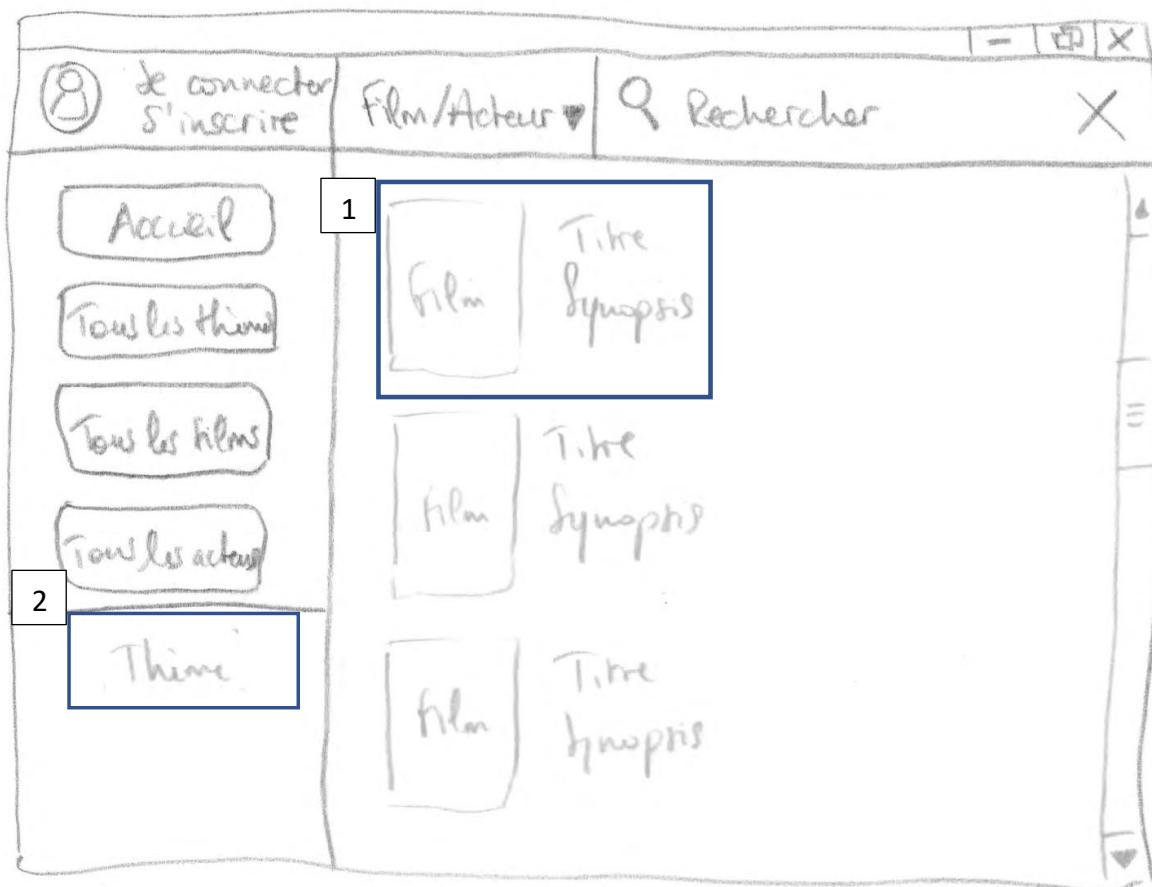
Ce bouton film a les mêmes fonctions qu'à l'accueil, avec les mêmes parties cliquables.

Page contenant tous les thèmes



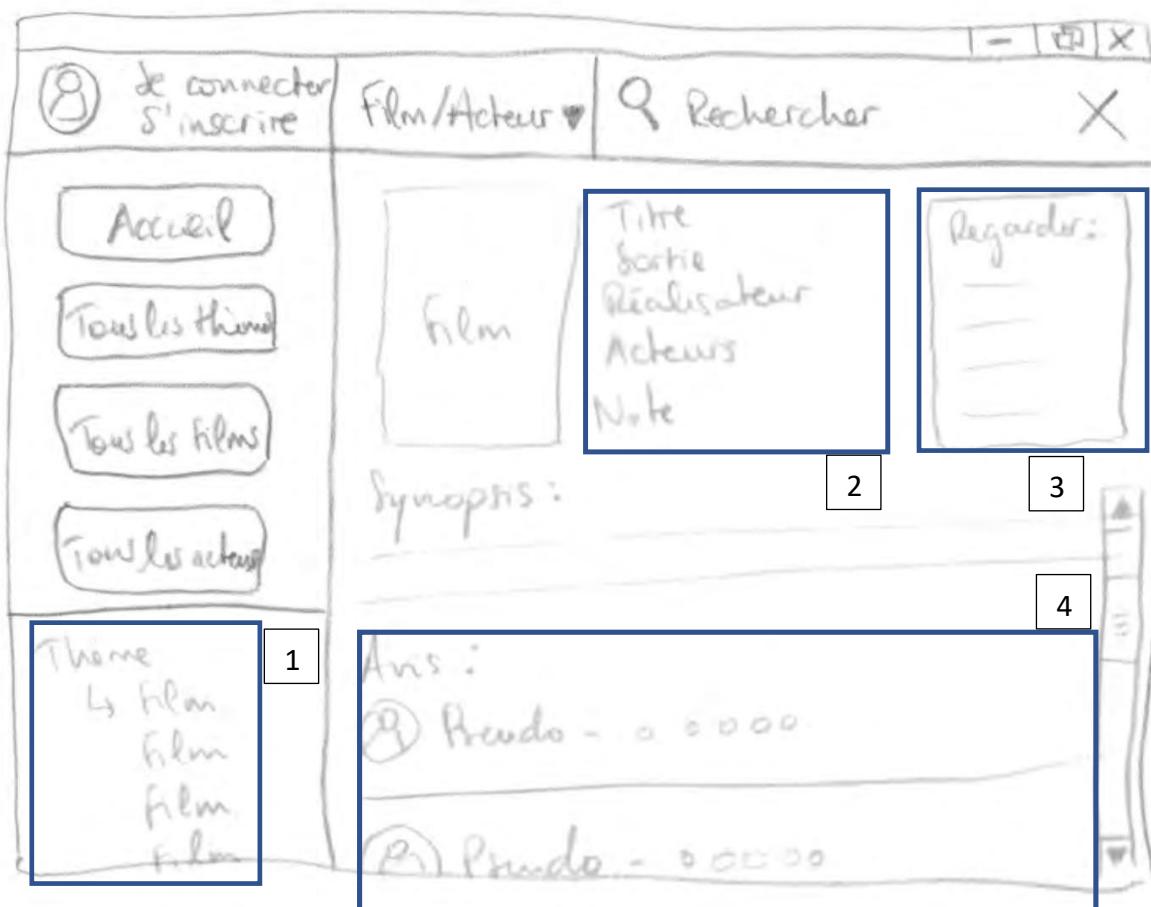
Chaque item « Acteur » (photo, nom et prénom) sera également cliquable à la manière des items « Film » pour se rendre sur sa page (photo, nom et prénom cliquables).

Page contenant tous les films d'un même thème



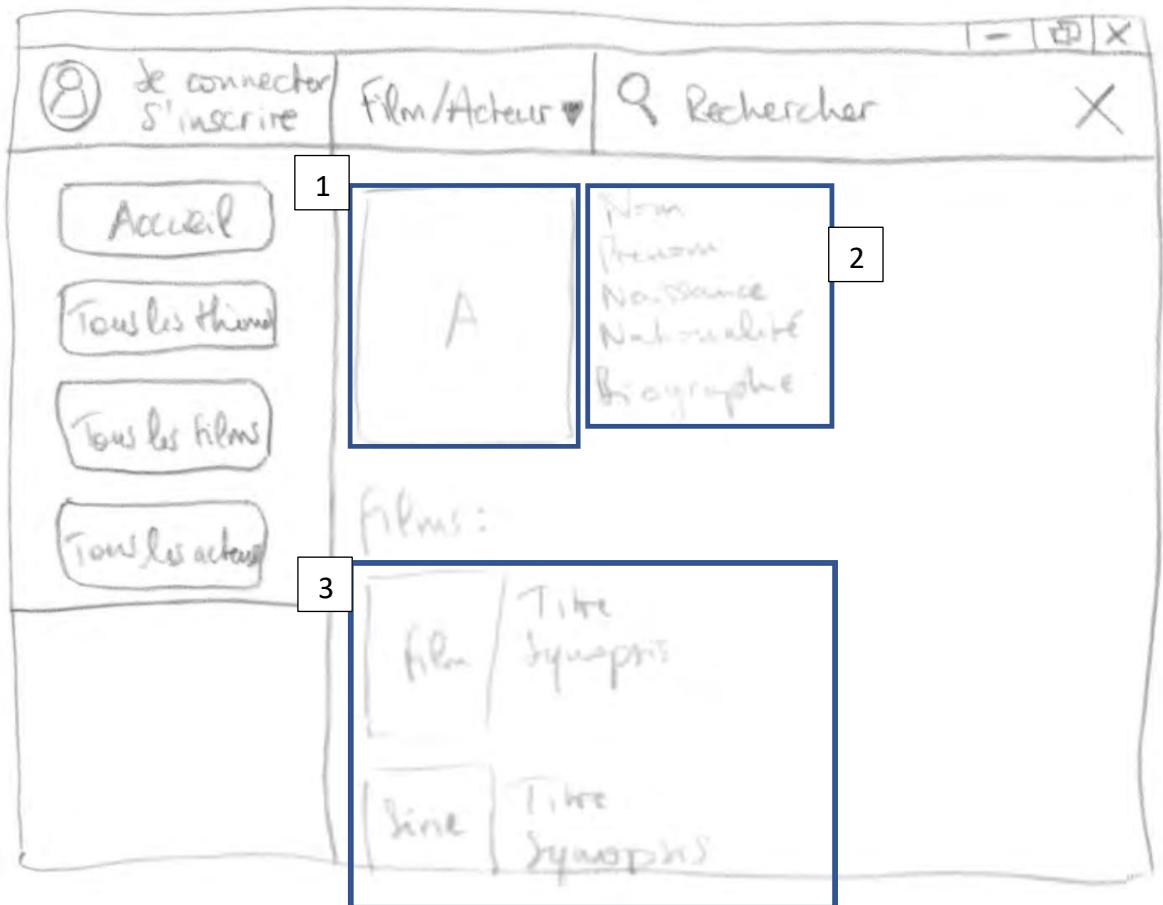
- 1) À la manière des pages « Accueil » et « Tous les thèmes », les films seront ici cliquables pour obtenir plus de renseignements.
- 2) La mention « Thème » encadrée aura un but indicatif : celui de rappeler le thème sur lequel on a cliqué.

Page contenant les détails d'un film



- 1) La section ci-dessus présentera le thème du film sur lequel on se trouve et sous cette mention figureront tous les noms des films associés à ce thème.
- 2) Cette partie de la page donnera tous les renseignements nécessaires sur le film :
 - Titre
 - Date de sortie
 - Réalisateur
 - Note générale attribuée par les utilisateurs
 - *Liste des acteurs principaux*, on trouvera cette section une liste verticale des noms des acteurs qui seront eux aussi cliquables pour se rendre sur leurs pages respectives.
- 3) Cette box présentera à l'utilisateur (en fonction de ses comptes) des liens pour regarder le film en streaming.
- 4) Enfin, cette section contiendra tous les commentaires des utilisateurs sur le film, les photos de profil et les pseudos des utilisateurs ayant laissé ces avis seront cliquables également et permettront de se rendre sur les profils utilisateurs de ces personnes.

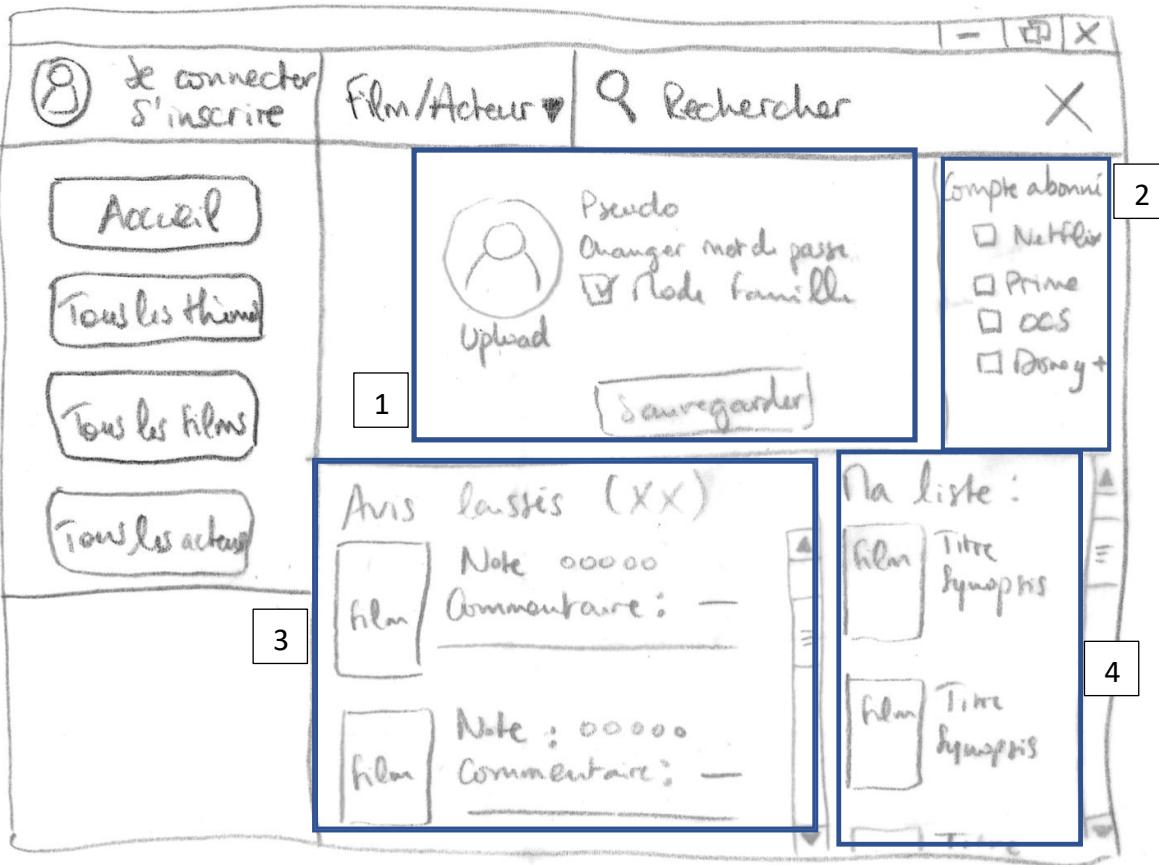
Page contenant les détails d'un acteur



Ici la page d'un acteur :

- 1) Une image de l'acteur en question
- 2) Quelques informations sur lui :
 - Nom
 - Prénom
 - Date de naissance
 - Nationalité
 - Biographie en quelques lignes
- 3) Et les films auxquels il a participé sur lesquels on pourra cliquer de la même manière que sur les pages précédentes (cf. sketch « Accueil »).

Page contenant les détails et paramètres du compte de l'utilisateur



1) Quelques informations sur votre compte :

- Votre photo de profil que vous avez la possibilité de changer à tout moment grâce au bouton « Uploader ».
- Votre pseudo, que vous pouvez changer en cliquant dessus.
- Un bouton pour *changer votre mot de passe*. Il déclenchera l'ouverture d'une petite fenêtre avec deux champs pour écrire votre ancien et votre nouveau mot de passe.
- Une checkbox libellée « Mode famille », que l'on peut cocher si on désire bloquer le contenu déconseillé aux enfants.

Enfin, le bouton *sauvegarder* qui vous permettra de sauvegarder vos modifications en ouvrant une petite fenêtre « prompt » pour entrer votre mot de passe.

- 2) Ici il suffit de cocher les comptes que l'on possède sur les différentes plateformes de streaming.
- 3) On trouve dans cette section les commentaires et les notes que l'on a laissées sur différents films récemment.
- 4) Enfin, cette section répertorie les films que l'on a mis en favoris pour les regarder plus tard ou pour ce faire une sorte de vidéothèque virtuelle.

Pages de connexion et inscription

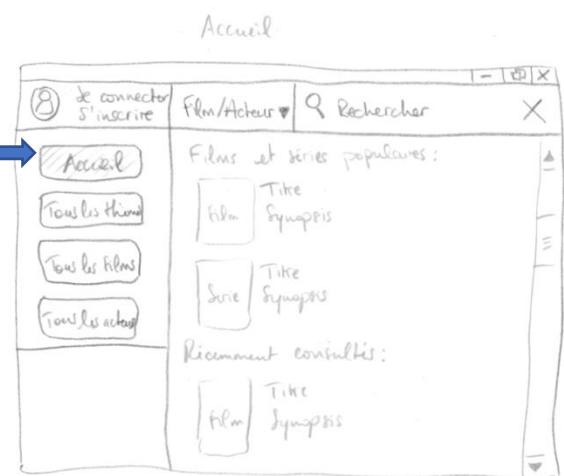


Voici les fenêtres « Connexion » et « Inscription ».

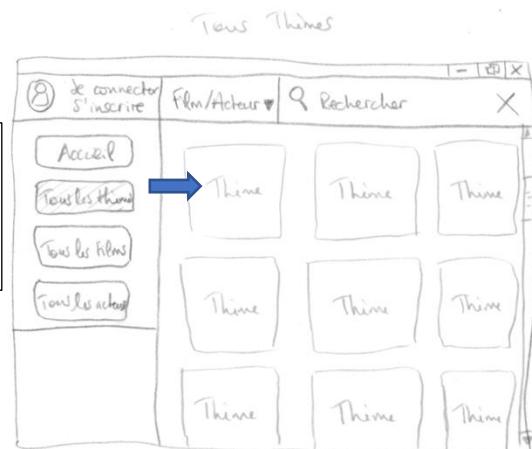
Partie 2 : Storyboards

RECHERCHER UN FILM PAR THEME :

L'utilisateur peut cliquer sur ce bouton pour se rendre sur la page où tous les thèmes sont présentés.



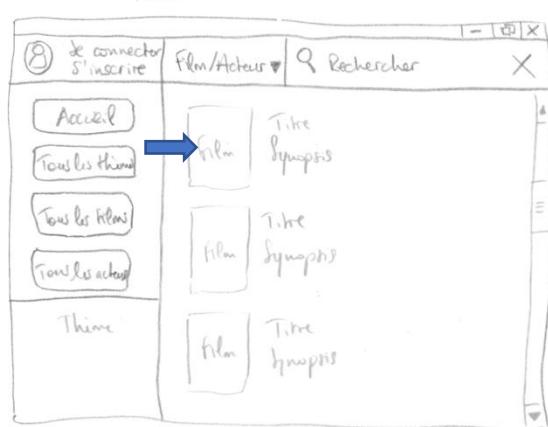
Il peut ensuite cliquer sur un thème pour afficher tous les films qui y correspondent.



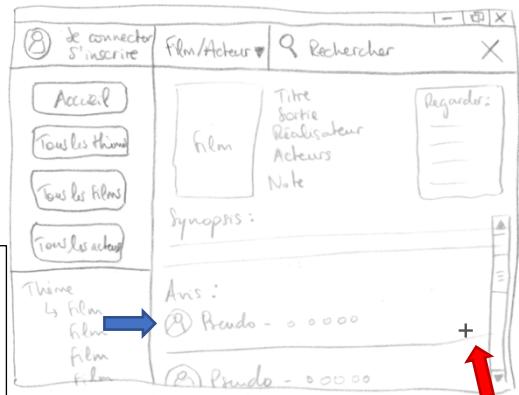
Puis l'utilisateur cliquera sur un film pour arriver sur sa page.



Il arrive enfin sur la page du film sur lequel il a cliqué.



Film / Série



CONSULTER LES AVIS LAISSES SUR UN FILM,
PUIS ACCÉDER AU PROFIL D'UN MEMBRE :

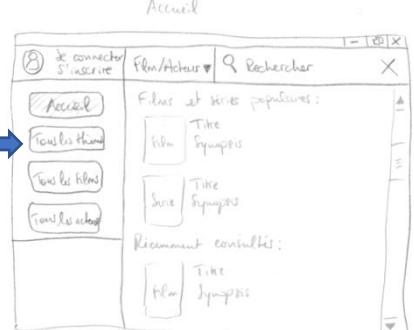
Sur la page d'un film,
l'utilisateur pourra
consulter le profil d'un
autre membre en cliquant
sur sa photo de profil ou
son pseudo.

A hand-drawn wireframe of a member profile page. The header is identical to the movie details page. The sidebar on the left has the same links. The main content area shows a profile card for 'Pseudo'. Below the card is a section titled 'Avis laissés (XX)' containing two review snippets. A large blue arrow points from this profile page down to the main movie details page.

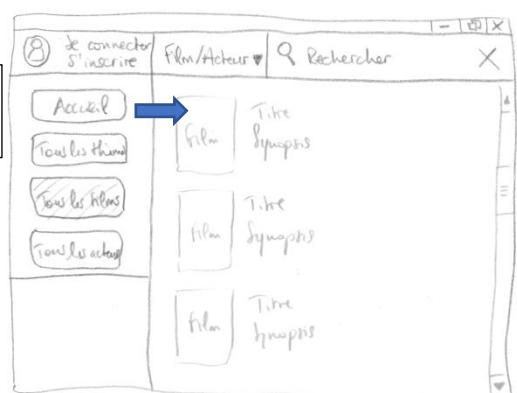
Ou bien ajouter un
commentaire, une petite
fenêtre prompt s'ouvrira
alors sur la fenêtre
principale.

RECHERCHER UN FILM EN PASSANT PAR UN AUTRE FILM, PUIS UN ACTEUR :

Ici il est possible de se
rendre sur la page qui
contient tous les films.



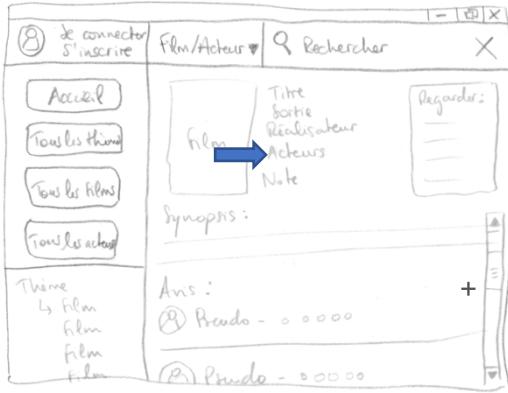
L'utilisateur cliquera alors
sur le film qui l'intéresse.



Film / Série

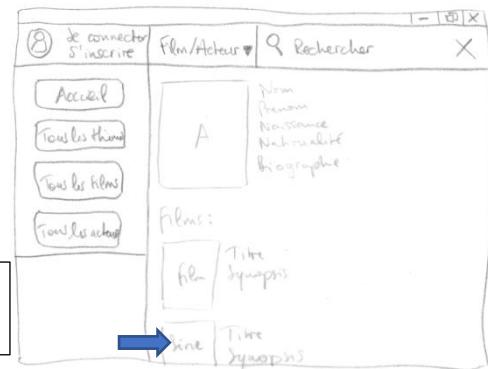
A cet endroit, on trouve une liste des noms des acteurs.

L'utilisateur clique ensuite sur le nom de l'un d'entre eux (un lien) pour se rendre sur la page de l'acteur en question.



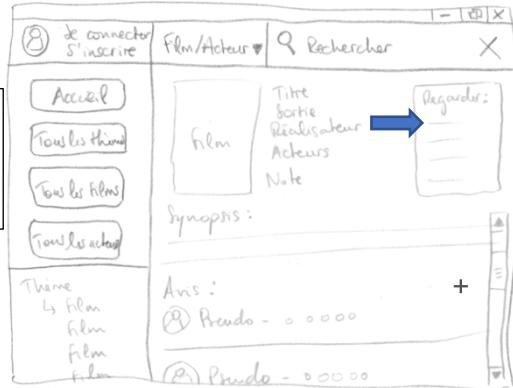
Acteur

Puis clique sur un film auquel l'acteur a participé.



Film / Série

Et enfin il clique sur un lien de streaming, la page web s'ouvrira dans le logiciel.



RECHERCHE D'UN ACTEUR VIA 'TOUS LES ACTEURS' :

L'utilisateur peut cliquer sur le bouton pour se rendre sur la page « tous les acteurs ».

Accueil

⑧ de connecter s'inscrire Film/Acteur Rechercher X

Films et séries populaires:

- Film Titre Synopsis
- Série Titre Synopsis

Récemment consultés:

- Film Titre Synopsis

Et ainsi se rendre sur la page d'un acteur.

Acteur

⑧ de connecter s'inscrire Film/Acteur Rechercher X

A

Nom Prénom

Nom Prénom

Nom Prénom

Nom Prénom

Films:

- Film Titre Synopsis
- Film Titre Synopsis

Tous Acteurs

⑧ de connecter s'inscrire Film/Acteur Rechercher X

A

Nom Prénom

Nom Prénom

Nom Prénom

CONNEXION :

Ici on peut se connecter (Ouverture d'une petite fenêtre par-dessus).

Accueil

⑧ de connecter s'inscrire Film/Acteur Rechercher X

Films et séries populaires:

- Film Titre Synopsis
- Série Titre Synopsis

Récemment consultés:

- Film Titre Synopsis

Accueil

⑧ de connecter s'inscrire Film/Acteur Rechercher X

Accueil

Tous les thèmes

Tous les films

Tous les acteurs

Films et séries populaires:

- Film Titre Synopsis
- Série Titre Synopsis

Récemment consultés:

- Film Titre Synopsis

Après avoir rentré ses informations, l'utilisateur est renvoyé sur la page d'accueil.

Connexion

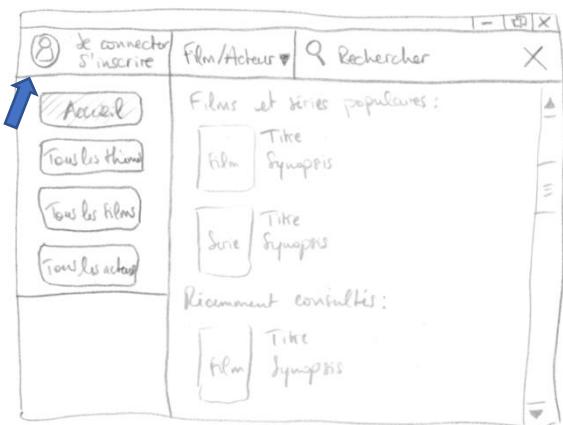
Pseudo

Password

Connexion

MODIFICATION DE SES INFORMATIONS PERSONNELLES :

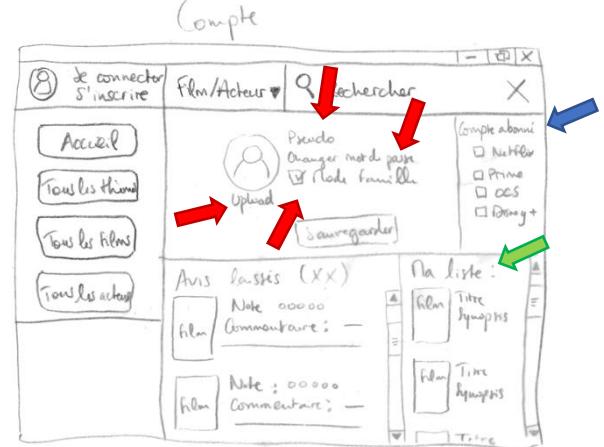
Ici, l'utilisateur peut, s'il est connecté, jeter un œil à son compte en cliquant sur sa photo de profil.



A partir d'ici, l'utilisateur peut changer son mot de passe, son pseudo, sa photo de profil ou bien activer le mode famille.

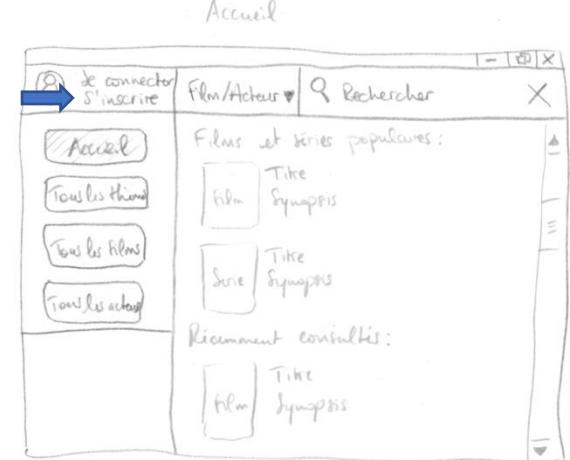
Toutes ces opérations s'effectueront dans une petite fenêtre juxtaposée sur la fenêtre principale.

Mais également consulter sa liste de films favoris ou modifier son répertoire de comptes connectés.

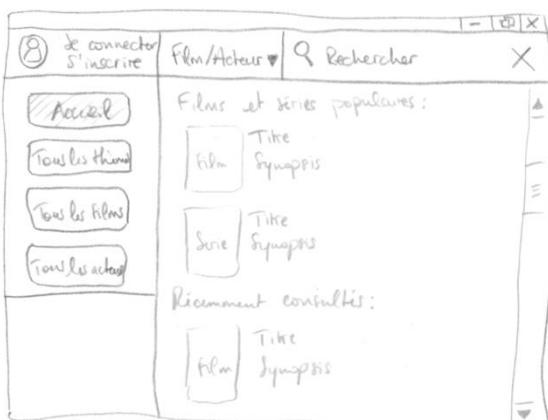


INSCRIPTION :

Ici on peut s'inscrire (Ouverture d'une petite fenêtre par-dessus).



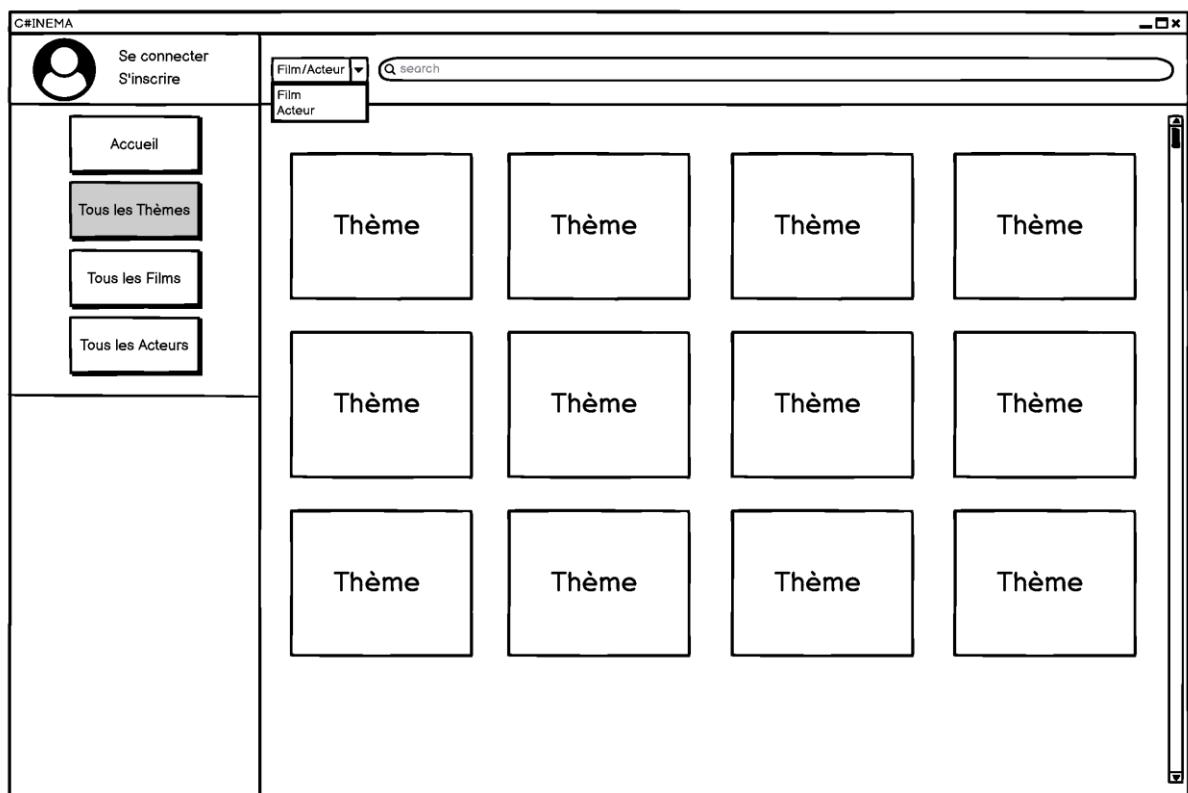
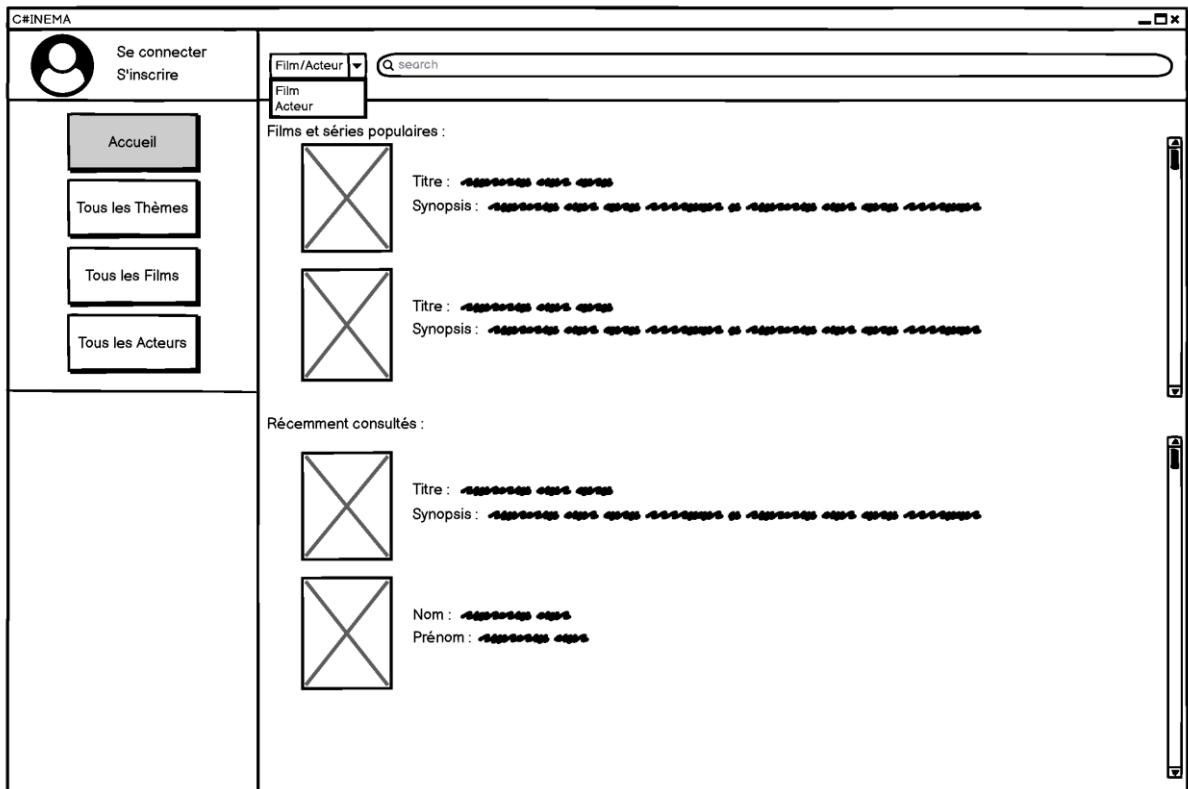
Accueil



Après avoir rentré ses informations, l'utilisateur est renvoyé sur la page d'accueil.



IV – Wireframes



C#INEMA

The wireframe shows a desktop application window titled "C#INEMA". On the left is a sidebar with a user icon, "Se connecter", "S'inscrire", "Accueil", "Tous les Thèmes", "Tous les Films", and "Tous les Acteurs". The main area has a search bar ("search") with dropdown menus for "Film/Acteur" and "Film Acteur". Below the search bar are five entries, each consisting of a square icon with an X, followed by "Titre :", and "Synopsis :".

Se connecter
S'inscrire

Film/Acteur ▾ search

Film
Acteur

Accueil

Tous les Thèmes

Tous les Films

Tous les Acteurs

Titre :
Synopsis :

This wireframe shows the same application window as above, but the main area displays four entries for actors, each with a square icon containing an X and fields for "Nom :" and "Prénom :" followed by placeholder text.

Se connecter
S'inscrire

Film/Acteur ▾ search

Film
Acteur

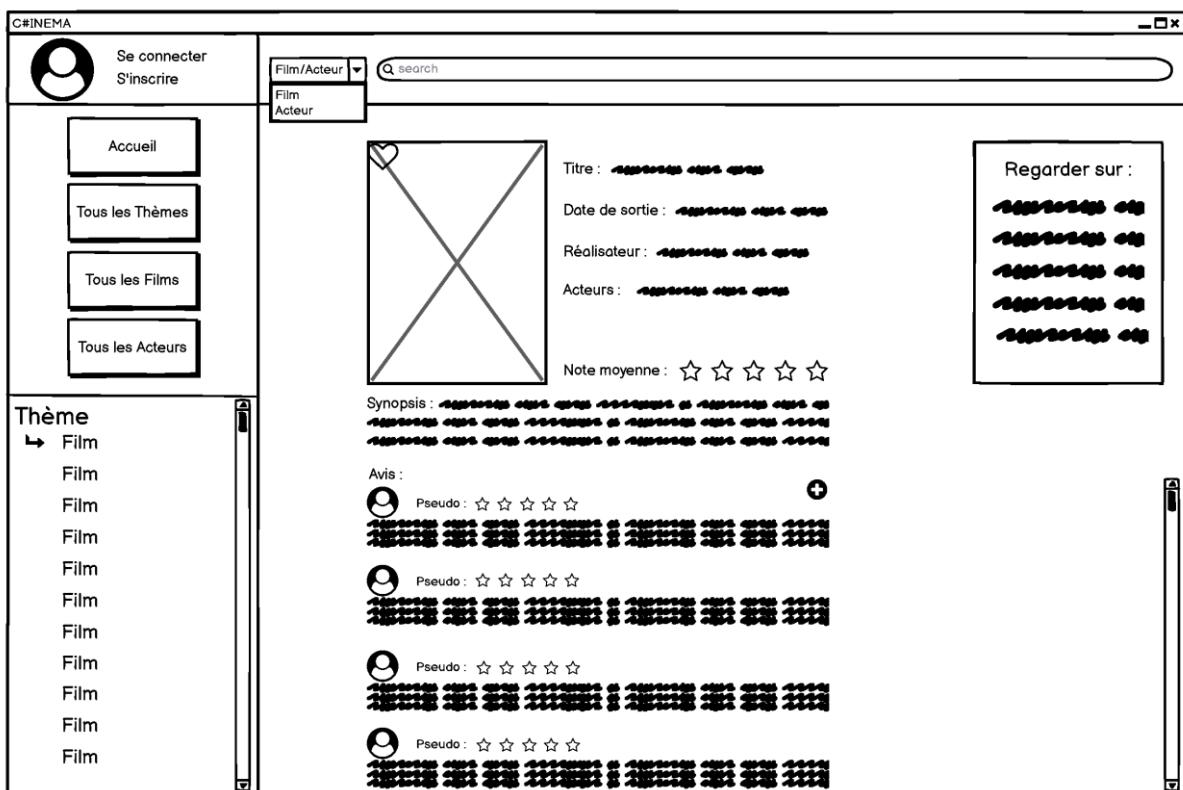
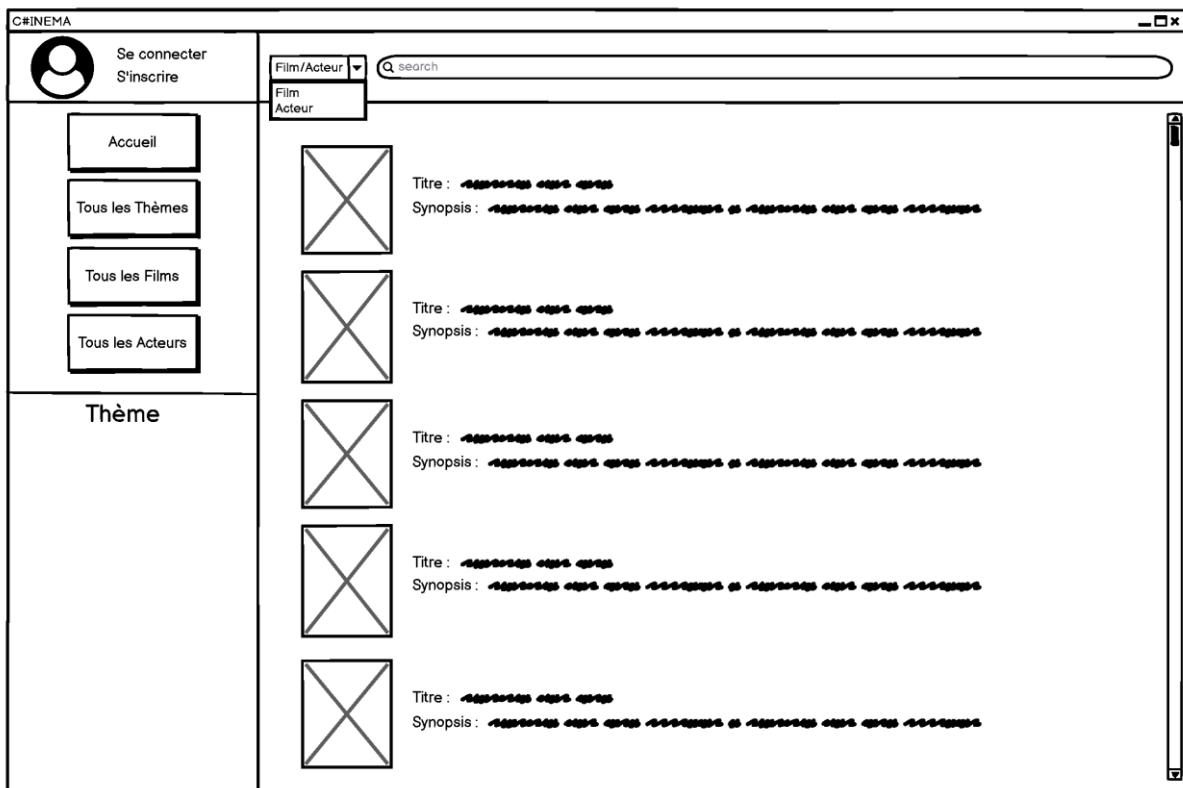
Accueil

Tous les Thèmes

Tous les Films

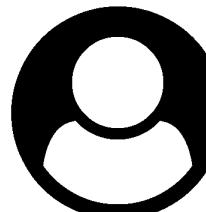
Tous les Acteurs

Nom :
Prénom :



The image shows a window titled "Connexion" with a black header bar. In the top right corner are standard window control buttons: a minus sign for window size, a square for window position, and an X for close. The main area contains a large circular icon with a white silhouette of a person's head and shoulders, representing a user profile. To the left of this icon is a black arrow pointing to the left. Below the icon are two rectangular input fields with black borders. The top field contains the text "Pseudo" and the bottom field contains the text "Mot de passe". At the bottom of the window is a single rectangular button with a black border containing the text "Connexion".

Créer son compte



<

Pseudo

Mot de passe

Confirmation

Créer

C#INEMA

Pseudo
Se déconnecter

Film/Acteur ▾ search

Film Acteur

Accueil
Tous les Thèmes
Tous les Films
Tous les Acteurs

Pseudo
Changer mot de passe :
 Mode Famille

Compte abonné :

Netflix
 Prime Video
 OCS
 Disney +

Sauvegarder

Avis laissés (XX) :

Note : ★ ★ ★ ★ ★
Commentaire :


Note : ★ ★ ★ ★ ★
Commentaire :


Note : ★ ★ ★ ★ ★
Commentaire :


Liste d'envie :

Titre :
Synopsis :


Titre :
Synopsis :


Titre :
Synopsis :


C#INEMA

Se connecter
S'inscrire

Film/Acteur ▾ search

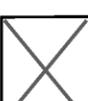
Film Acteur

Accueil
Tous les Thèmes
Tous les Films
Tous les Acteurs

Pseudo :

Avis laissés (XX) :

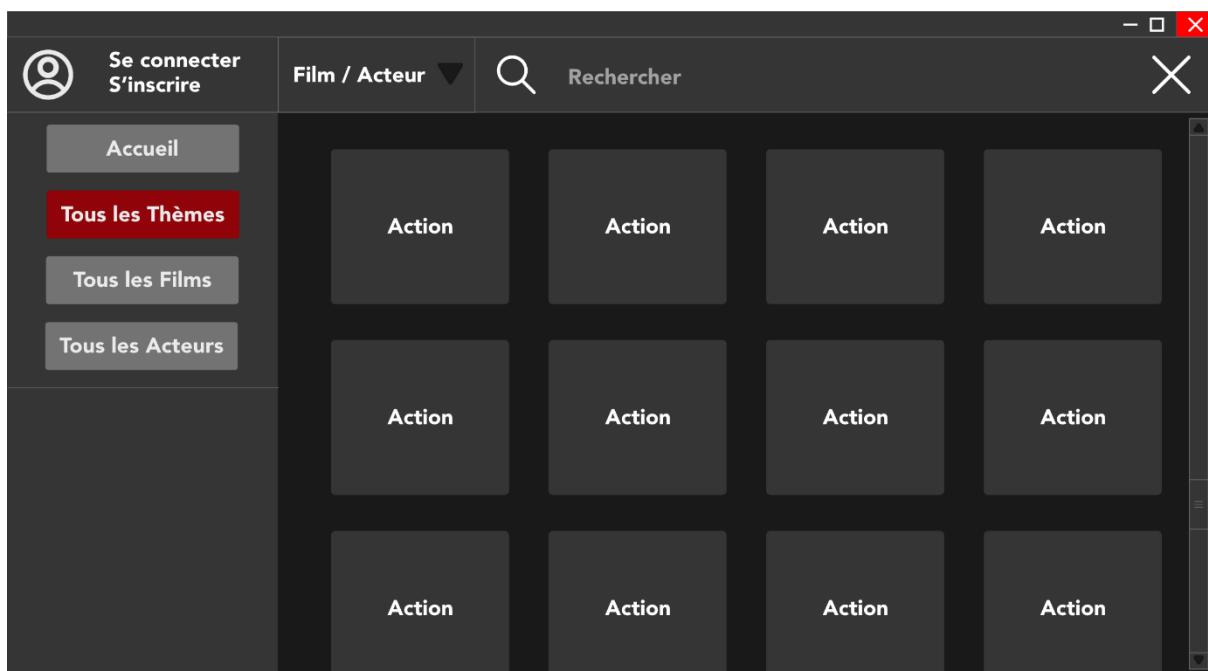
Note : ★ ★ ★ ★ ★
Commentaire :


Note : ★ ★ ★ ★ ★
Commentaire :


Note : ★ ★ ★ ★ ★
Commentaire :


V – Mock-ups

This mock-up shows a user interface for a movie database. The top navigation bar includes a user icon, links for "Se connecter" and "S'inscrire", a search dropdown set to "Film / Acteur", a search icon, and a "Rechercher" button. A close button (X) is in the top right. The left sidebar has buttons for "Accueil" (highlighted in red), "Tous les Thèmes", "Tous les Films", and "Tous les Acteurs". The main content area displays two sections: "Films et séries populaires:" and "Récemment consultés:". Each section contains a placeholder card for a "Film" or "Acteur" with fields for "Titre:" and "Synopsis:", followed by another card with similar fields.



Se connecter
S'inscrire

Film / Acteur

Rechercher

X

Accueil

Tous les Thèmes

Tous les Films

Tous les Acteurs

Titre : Fight Club
Synopsis : Le narrateur, sans identité précise, vit seul, travaille seul, dort seul, mange seul.
C'est pourquoi il va devenir membre du Fight club...

Titre : Fight Club
Synopsis : Le narrateur, sans identité précise, vit seul, travaille seul, dort seul, mange seul.
C'est pourquoi il va devenir membre du Fight club...

Titre : Fight Club
Synopsis : Le narrateur, sans identité précise, vit seul, travaille seul, dort seul, mange seul.
C'est pourquoi il va devenir membre du Fight club...

Titre : Fight Club
Synopsis : Le narrateur, sans identité précise, vit seul, travaille seul, dort seul, mange seul.
C'est pourquoi il va devenir membre du Fight club...

Titre : Fight Club
Synopsis : Le narrateur, sans identité précise, vit seul, travaille seul, dort seul, mange seul.

Se connecter
S'inscrire

Film / Acteur

Rechercher

X

Accueil

Tous les Thèmes

Tous les Films

Tous les Acteurs

 Nom : Pitt
Prénom : Brad

 Nom : Pitt
Prénom : Brad

 Nom : Pitt
Prénom : Brad

 Nom : Pitt
Prénom : Brad

Se connecter
S'inscrire

Film / Acteur Rechercher

Accueil

Tous les Thèmes

Tous les Films

Tous les Acteurs

Thème


Titre : Fight Club
Synopsis : Le narrateur, sans identité précise, vit seul, travaille seul, dort seul, mange seul.
C'est pourquoi il va devenir membre du Fight club...


Titre : Fight Club
Synopsis : Le narrateur, sans identité précise, vit seul, travaille seul, dort seul, mange seul.
C'est pourquoi il va devenir membre du Fight club...


Titre : Fight Club
Synopsis : Le narrateur, sans identité précise, vit seul, travaille seul, dort seul, mange seul.
C'est pourquoi il va devenir membre du Fight club...


Titre : Fight Club
Synopsis : Le narrateur, sans identité précise, vit seul, travaille seul, dort seul, mange seul.
C'est pourquoi il va devenir membre du Fight club...


Titre : Fight Club
Synopsis : Le narrateur, sans identité précise, vit seul, travaille seul, dort seul, mange seul.

Se connecter
S'inscrire

Film / Acteur Rechercher

Accueil

Tous les Thèmes

Tous les Films

Tous les Acteurs

Thème

↳ Film

Film

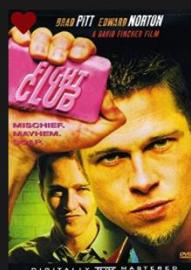
Film

Film

Film

Film

Film


Titre : Fight Club
Date de sortie : 10/11/1999
Réalisateur(s) : David Fincher
Acteurs : Brad Pitt
Edward Norton
Helena Bonham Carter

Regarder :
[Lien](#)
[Lien](#)
[Lien](#)
[Lien](#)

Note : ★★★★☆

Synopsis :
Le narrateur, sans identité précise, vit seul, travaille seul, dort seul, mange seul.
C'est pourquoi il va devenir membre du Fight club...

Avis :
 thurarDu28 - ★★★★☆

+

- □ X

 Se connecter
S'inscrire

Film / Acteur  Rechercher 

Accueil

Tous les Thèmes

Tous les Films

Tous les Acteurs



Nom : Pitt
Prénom : Brad
Date de naissance : 18/12/1963
Nationalité : Américain

Biographie : William Bradley Pitt, dit Brad Pitt, est un acteur et producteur de cinéma américain, né le 18 décembre 1963 à Shawnee. Repéré dans une publicité pour Levi's, Brad Pitt sort de l'anonymat grâce à un petit rôle dans le film *Thelma et Louise* de Ridley Scott.

Films :


Titre : Fight Club
Synopsis : Le narrateur, sans identité précise, vit seul, travaille seul, dort seul, mange seul. C'est pourquoi il va devenir membre du Fight club...


Titre : Fight Club
Synopsis : Le narrateur, sans identité précise, vit seul, travaille seul, dort seul, mange seul. C'est pourquoi il va devenir membre du Fight club...

- □ X

< 

Pseudo

Mot de passe

Connexion

Inscription

Pseudo

Mot de passe

Confirmation

S'inscrire

Se connecter

Film / Acteur

Rechercher

Accueil

Tous les Thèmes

Tous les Films

Tous les Acteurs

Pseudo : thurarDu28

[Changer le mot de passe](#)

Mode famille

Uploader une photo

Sauvegarder

Comptes abonné :

- Netflix
- OCS
- Prime Video
- Disney+

Avis laissés (XX)

Note : ★★★★☆

Commentaire : Une belle bouse sans nom...

Note : ★★★★☆

Commentaire : Une belle bouse sans nom...

Ma liste :

Titre : Fight Club
Synopsis : Le narrateur, sans identité précise, vit seul, travaille seul, dort seul, mange seul.

Titre : Fight Club
Synopsis : Le narrateur, sans identité précise, vit seul, travaille seul, dort seul, mange seul.

S'inscrire

Se connecter

Film / Acteur

Rechercher

Accueil

Tous les Thèmes

Tous les Films

Tous les Acteurs

Pseudo : thurarDu28

Avis laissés (XX)

Note : ★★★★☆

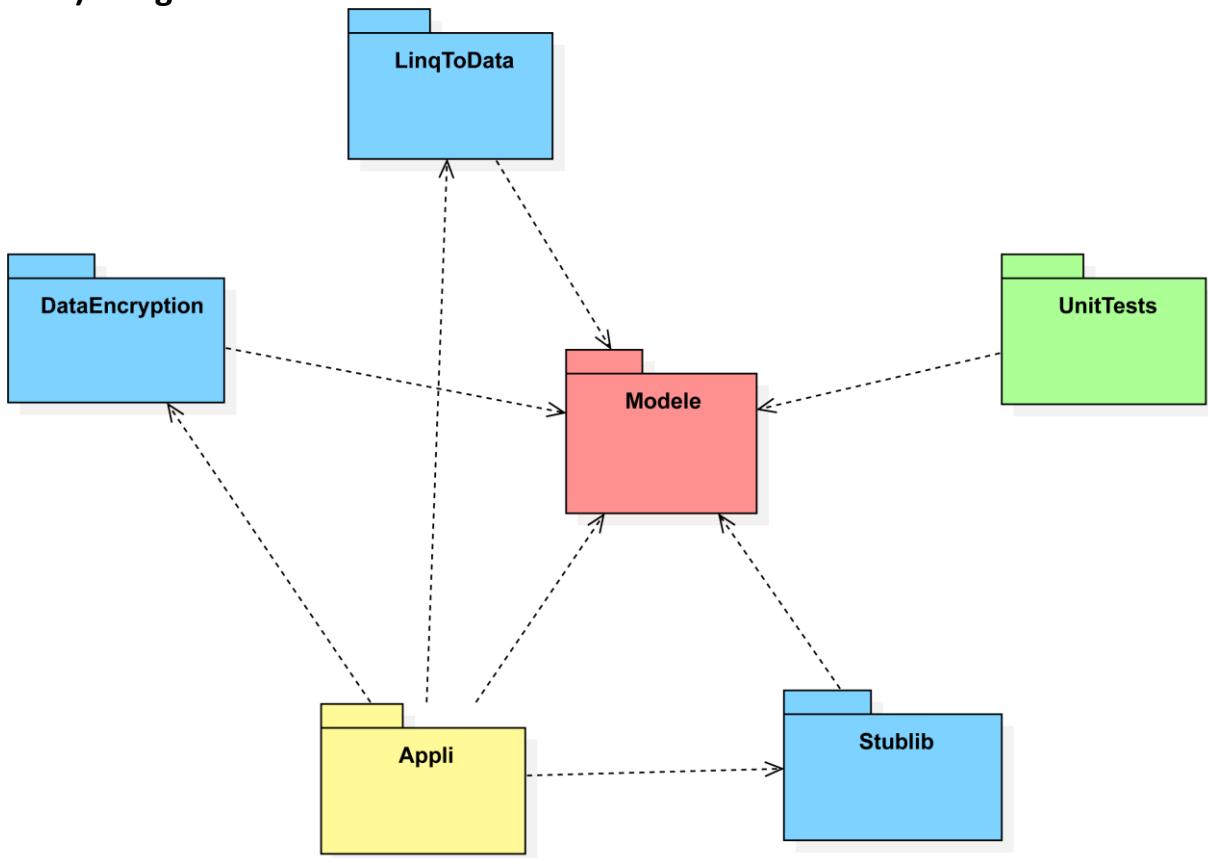
Commentaire : Une belle bouse sans nom...

Note : ★★★★☆

Commentaire : Une belle bouse sans nom...

VI – Diagramme de paquetage

a) Diagramme



b) Explications

On voit sur le diagramme six packages différents :

- **Appli**, package qui contient toutes les vues de l'application :
 - o Son rôle : Rassembler toutes les vues et contenir la partie « visuelle » de notre application.
 - o Son contexte : Il s'agit du premier assembly qu'on ait eu à créer et pour cause, sans vues, il n'y a pas d'application ni même de liaison à faire avec le code-behind. C'est un assembly clé du projet.
 - o Nous avons décidé de séparer cet assembly du reste car il ne s'agit que des vues et du binding et que le fonctionnement de l'application se fait via les autres assemblies de la solution.
- **Modele**, package contenant toutes les classes :
 - o Son rôle : Gérer toutes les « entités » présentes dans l'application (comme les films ou séries, acteurs y figurant ou même les utilisateurs) en comportant des classes instanciables qui permettent de les concrétiser.

- Son contexte : Cet assembly trouve également une position clé dans le projet car il permet à l'application de prendre vie en faisant fonctionner les vues et en concrétisant les entités vues dans le XAML, en leur attribuant des classes et des propriétés.
 - Nous avons séparé cet assembly du reste du projet pour le rendre indépendant et ainsi l'utiliser dans d'autres assemblies.

- UnitTests, où l'on trouve tous les tests unitaires des classes de Modele :
 - Son rôle : Tester toutes les classes, propriétés et méthodes de l'assembly Modele indépendamment. On trouve dans les fichiers de cet assembly des tentatives de création de bug lors de l'instanciation de classes en tous genres pour essayer de sécuriser au maximum l'utilisation de l'application.
 - Son contexte : Cet assembly a été créé dans le même laps de temps que Modele du fait de sa dépendance à celui-ci. On lui trouve un rôle peut-être moins primordial mais pas moins nécessaire que les deux autres assembly.
 - Pour des raisons de propreté du code, nous avons séparé le code des tests. Ainsi, le code n'interfère pas dans les compilations et générations.
- StubLib, qui instancie une IFactory et implémente l'interface IPersistance :
 - Son rôle : Créer l'entièreté des instances nécessaires au bon remplissage de l'application.
 - Son contexte : Ce projet a été créé pour remplir le Manager de données factices pour démarrer le binding et les modifications. Son utilisation s'est arrêtée lors de la création des fichiers de persistance.
 - Nous l'avons séparé du reste des projets pour organiser le code plus proprement. De plus, son utilisation n'est pas obligatoire par l'application.
- LinqToData, qui contient les classes de persistance des données :
 - Son rôle : Implémenter les interfaces IFactory et IPersistance et leurs méthodes Sauvegarder, Charger et celles qui servent à la création des instances.
 - Son contexte : Nous avons eu à créer ce projet pour faire persister les données à travers les différentes exécutions de l'application. Ainsi nous n'avions qu'à compléter un fichier qui remplirait l'application et changerait en fonction des modifications apportées durant l'utilisation.
 - Nous l'avons séparé du reste des projets pour organiser le code plus proprement. De plus, son utilisation n'est pas obligatoire par l'application.
- DataEncryption, qui contient une interface IEncryption et quatre classes qui l'implémente :
 - Son rôle : être utilisé par LinqToData pour encrypter les données sensibles liées à l'application sur l'ordinateur des utilisateurs.

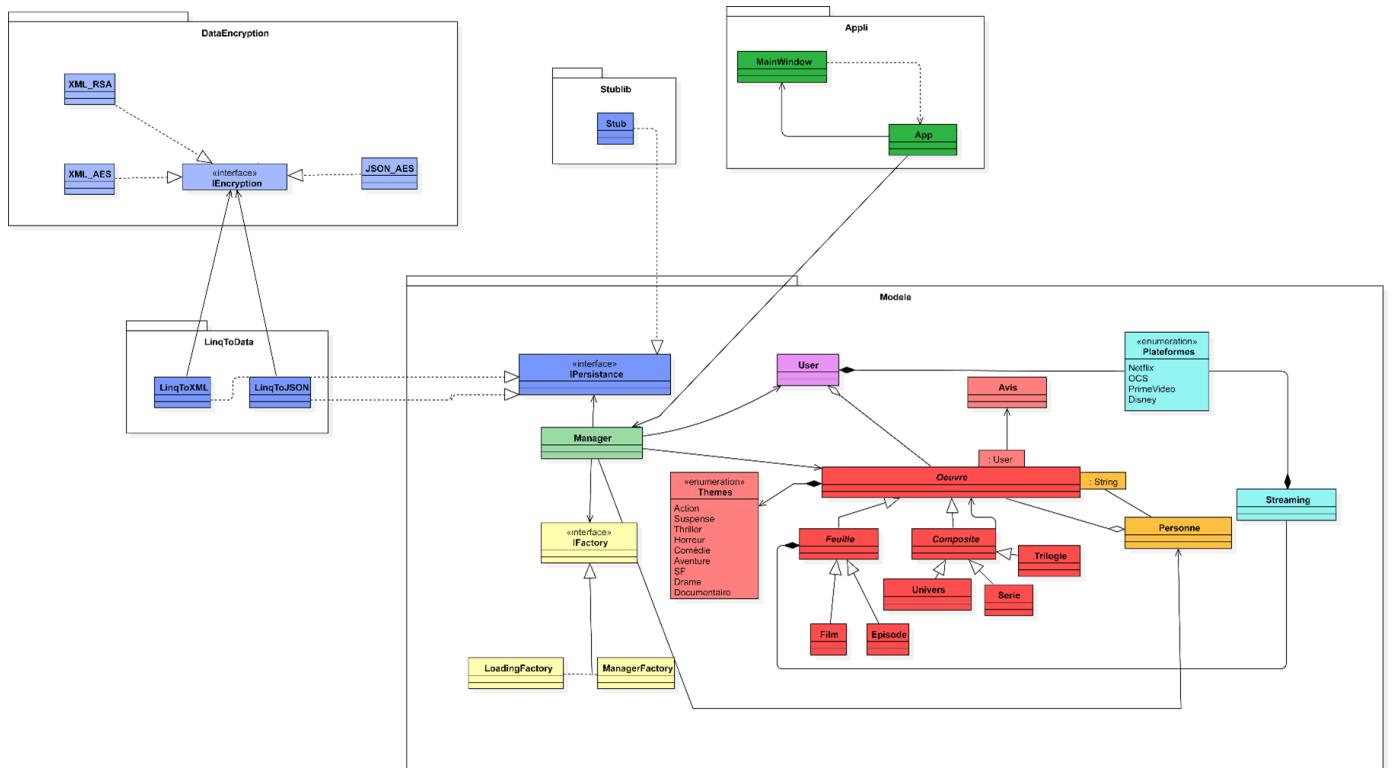
- Son contexte : Nécessaire en termes de sécurité du fait de l'existence de profils utilisateurs et de mots de passe.
- Nous l'avons séparé car encore une fois son utilisation n'est pas indispensable par le reste de l'application.

Figurent également sur ce diagramme deux dépendances :

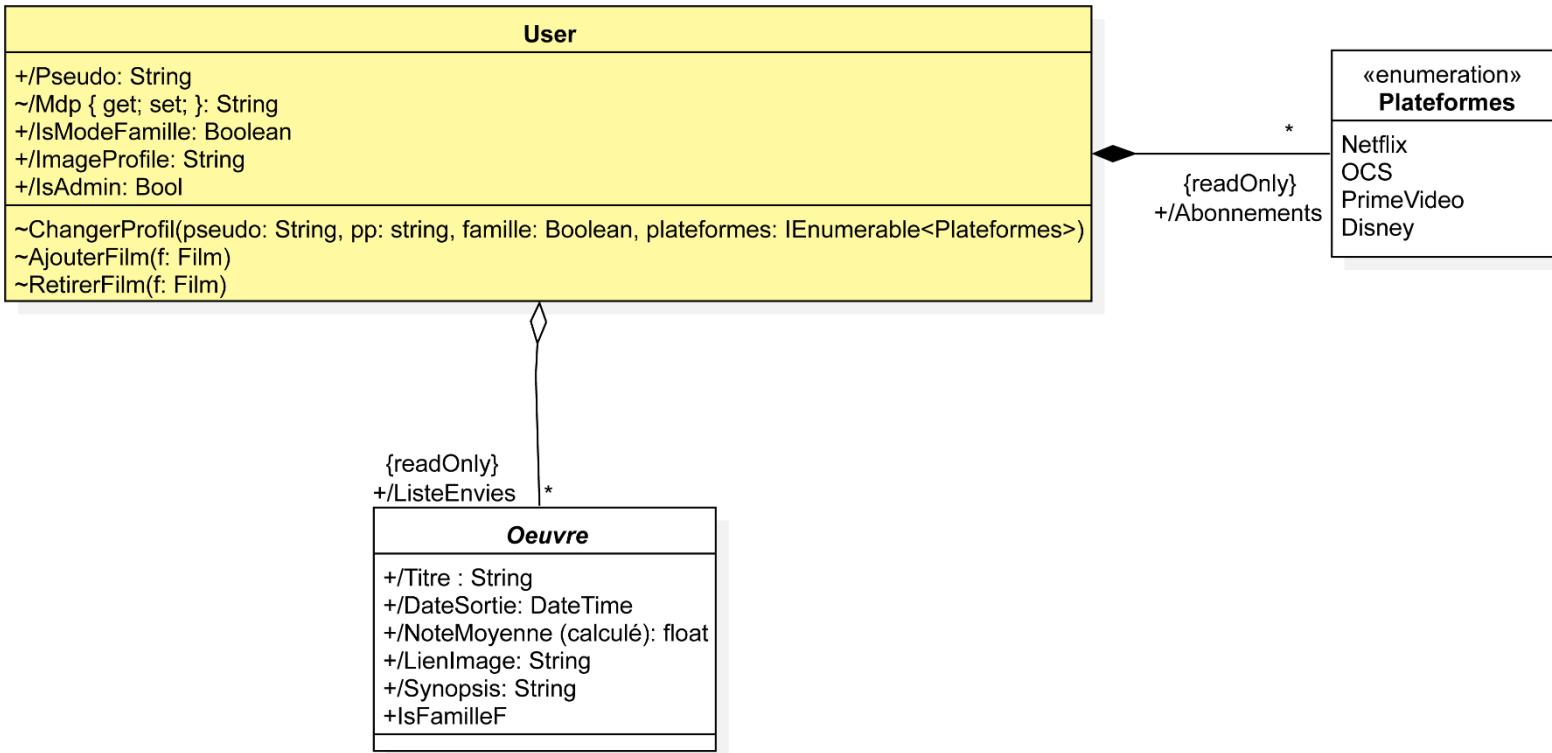
- Une première de UnitTests à Modele, qui permet aux tests unitaires d'accéder aux classes pour reconnaître les types, constructeurs et méthodes qu'il est nécessaire d'appeler pour s'assurer du bon fonctionnement des classes.
- Et une seconde dépendance de Appli à Modele. Cette dépendance servira dans le futur à la liaison des vues et du C#.

VII – Diagramme de classe

1) Diagramme version simplifiée



2) User

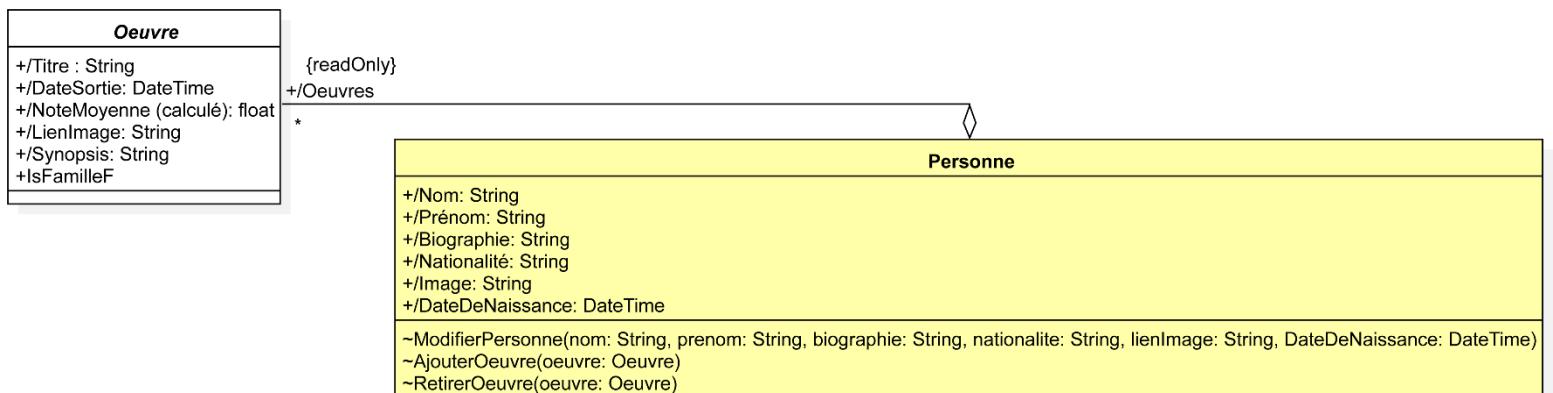


Commençons par la classe User. Un User est une classe publique, implémentant IEquatable<User>.

On trouve notamment dans cette classe les attributs :

- Mdp : string (internal), est internal car modifiable seulement par le biais du Manager, qui doit y avoir accès.
- IsModeFamille : bool (public), permet de savoir si le mode famille est activé sur un compte (donc un User) et donc de restreindre l'accès à certains films.
- IsAdmin : bool (public), permet de savoir si un utilisateur est admin, et donc s'il a le droit de créer, modifier ou supprimer des Œuvres ou User.
- ListePlateforme : ReadOnlyCollection<Plateforme> (public), est une collection de Plateforme, remplie en fonction des abonnements de l'User.
- RecemmentConsultes : ReadOnlyCollection<Object> (public), est une collection de Personne et d'œuvre que le User a récemment consulté.

3) Personne



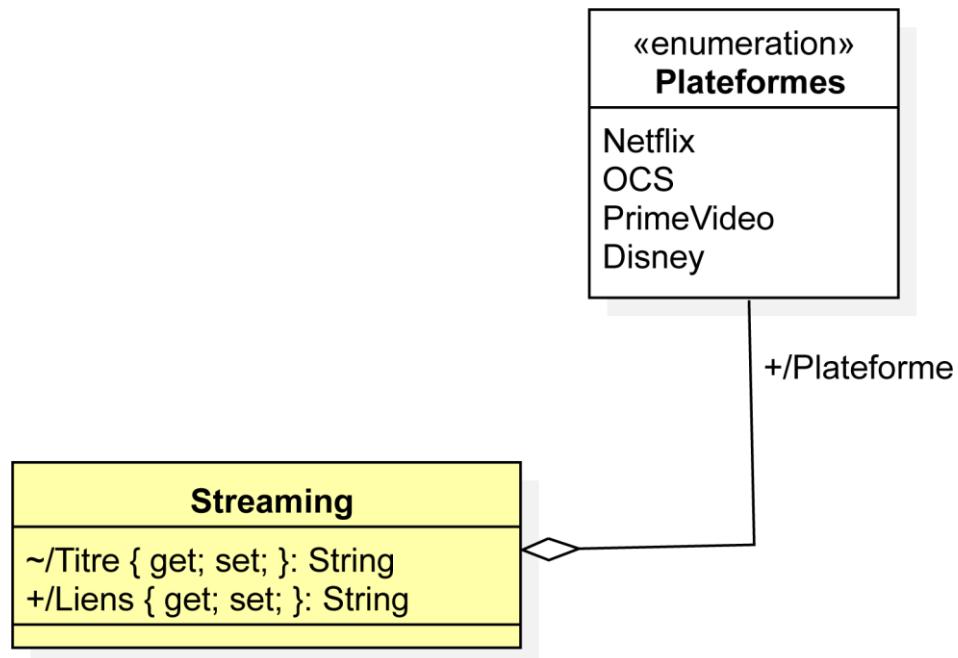
Une Personne est une classe publique, implémentant `IEquatable<Personne>`.

Cette classe représente une personne ayant travaillé sur une Œuvre (comme acteur ou réalisateur).

Elle possède un attribut Œuvres qui est une Collection d'œuvres sur lesquelles la Personne figure. Elle possède en conséquence deux méthodes pour ajouter et retirer une Œuvre de la Collection.

On trouve notamment dans cette classe les attributs une `ReadOnlyCollection<Œuvre>` œuvres (public), qui contient toutes les œuvres auxquelles cette Personne a participé.

4) Streaming

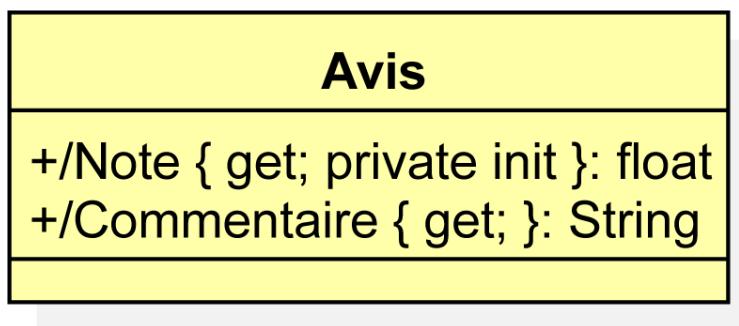


Un `Streaming` est une classe publique, implémentant `IEquatable<Streaming>`.

On trouve notamment dans ses attributs :

- Un Titre : string (internal), de l'œuvre qui le contient.
- Un Lien : string (public), vers l'œuvre en streaming.
- Une Plateforme : Plateforme (public), venant d'un enum, indiquant la plateforme vers laquelle renvoie le Lien.

5) Avis

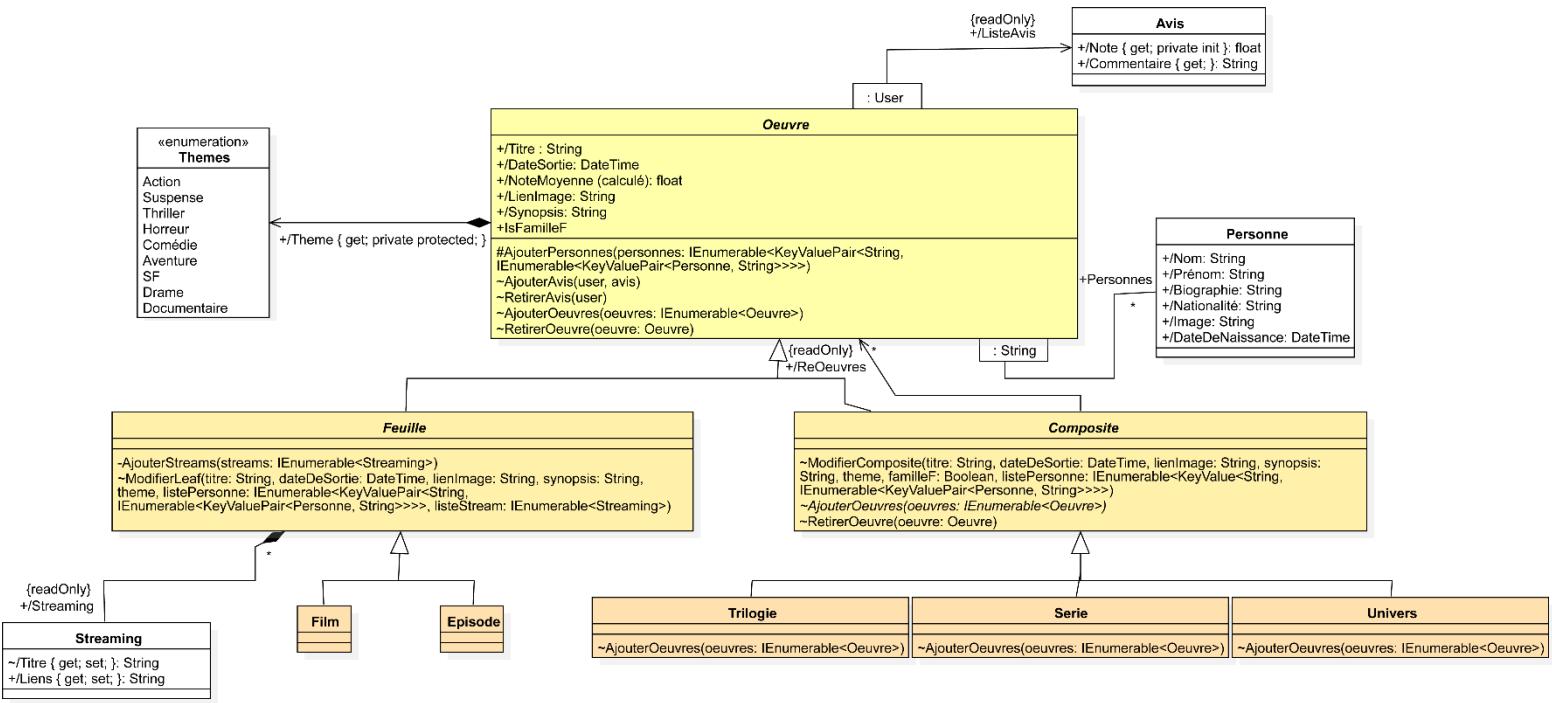


Un `Avis` est une classe publique, implémentant `IEquatable<Avis>`.

On trouve notamment parmi ses attributs :

- Une Note : float (public), avec un private init, qu'on ne peut pas changer après la création.
- Un Commentaire : string (public), écrit par le User.

6) Œuvre



Une Œuvre est une classe publique abstraite, implémentant IEquatable<Œuvre>.

On trouve notamment dans cette classe les attributs :

- Une ListeAvis : `ReadOnlyDictionary<User, Avis>` (public), qui fait correspondre chaque Avis à l'User qui l'a posté.
Son utilisation a pour avantages d'empêcher un utilisateur de poster deux commentaires sur une même Œuvre et de pouvoir retrouver pour chaque utilisateur sa liste d'avis postés simplement en parcourant la liste d'avis de chaque œuvre. Cela impose de redéfinir le protocole d'égalité pour un User que nous avons alors basé sur le pseudo. Un utilisateur ne peut donc pas posséder le même pseudo qu'un autre, et cela grâce à des vérifications effectuées dans Modele.
- Une NoteMoyenne : `float` (public), propriété calculée, à partir de toutes les notes des Avis de la ListeAvis.
- Un Theme : `Theme` (public), unique et provenant de l'enum correspondant.
- Une ListePersonnes : `ReadOnlyDictionary<string, Dictionary<Personne, string>>` (public), Sert à l'Œuvre pour répertorier ses acteurs et réalisateurs, regroupés sous la classe Personne.

Le premier string (key du premier dictionary) est le métier de la personne (dans l'application soit acteur soit réalisateur, la personne (clé du deuxième dictionary) est la personne dont il est question et le deuxième string (value du deuxième dictionary) est le rôle de la personne dans l'œuvre, vide s'il s'agit d'autre chose que d'un acteur. Le dictionnaire nous permet ici encore une fois de retrouver facilement les acteurs ayant participé

Plusieurs sous-classes dérivent de la classe Œuvre, on trouve ici en réalité un patron de conception : le Composite. Nous expliquerons son utilité en dernière partie de ce document.

- Une Leaf est une classe publique abstraite, implémentant Œuvre.

Nous avons rajouté à Leaf un attribut ListeStreams : Collection<Streaming> (public), lui permettant de stocker des instances de Streaming. Nous avons décidé d'en faire une composition car les instances de streaming doivent dépendre d'une Leaf, autrement dit, si une Leaf disparaît, ses streamings aussi. Cet attribut n'apparaît que sur Leaf pour la simple et bonne raison qu'un Œuvre (donc une liste d'Episodes, ou de Films) répertorie déjà des Leafs qui, elles, possèdent cet attribut. Cette classe représente soit un Film soit un Episode :

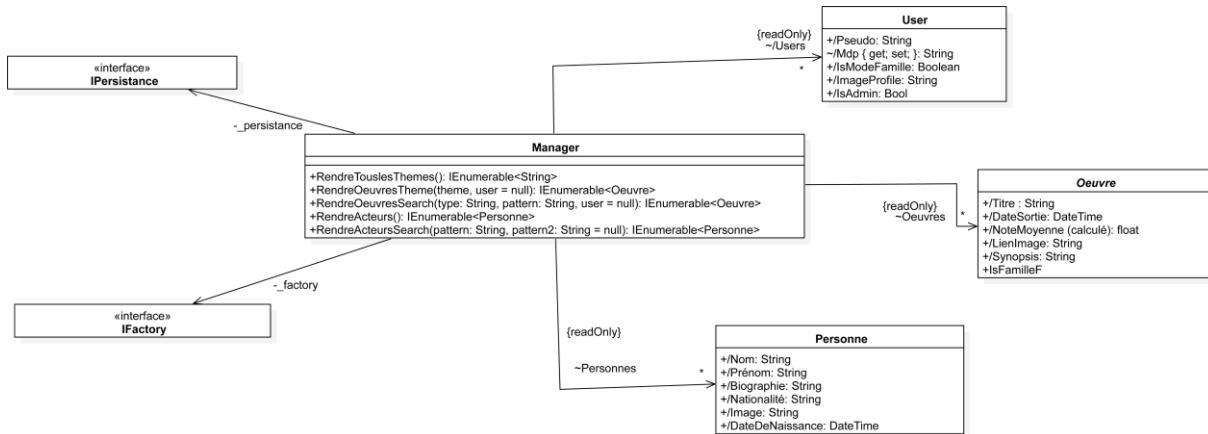
- o Un Film est une classe publique, implémentant Leaf.
Cette classe comporte tous les attributs de ses classes mères.
- o Un Episode est une classe publique, implémentant Leaf.
Cette classe comporte tous les attributs de ses classes mères.

- Un Composite est une classe publique abstraite, implémentant Œuvre.

Nous avons ajouté à Composite un attribut Œuvres : Collection<Œuvre> (public) lui permettant de stocker des instances d'œuvre. Cette classe représente au choix une Trilogie, une Série ou un Univers :

- o Une Trilogie est une classe publique, implémentant Composite.
Cette classe comporte tous les attributs de ses classes mères.
Cette classe représente un ensemble de trois Films.
- o Une Série est une classe publique, implémentant Composite.
Cette classe comporte tous les attributs de ses classes mères.
Cette classe représente un ensemble d'un nombre indéfini d'Episode.
- o Un Univers est une classe publique, implémentant Composite.
Cette classe comporte tous les attributs de ses classes mères.
Cette classe représente un ensemble de taille indéfinie de Trilogies, Films et Séries qui constituent un seul et même Univers (exemple : Star Wars, 2 trilogies, des séries spin-off etc.).

7) Manager



Un Manager est une classe publique.

Elle possède trois attributs primaires :

- Users : `ReadOnlyCollection<User>` (internal), la liste de tous les Users de l'application.
- Œuvres : `ReadOnlyCollection<Œuvre>` (internal), la liste de toutes les Œuvres de l'application.
- Personnes : `ReadOnlyCollection<Personne>` (internal), la liste de toutes les Personnes de l'application.

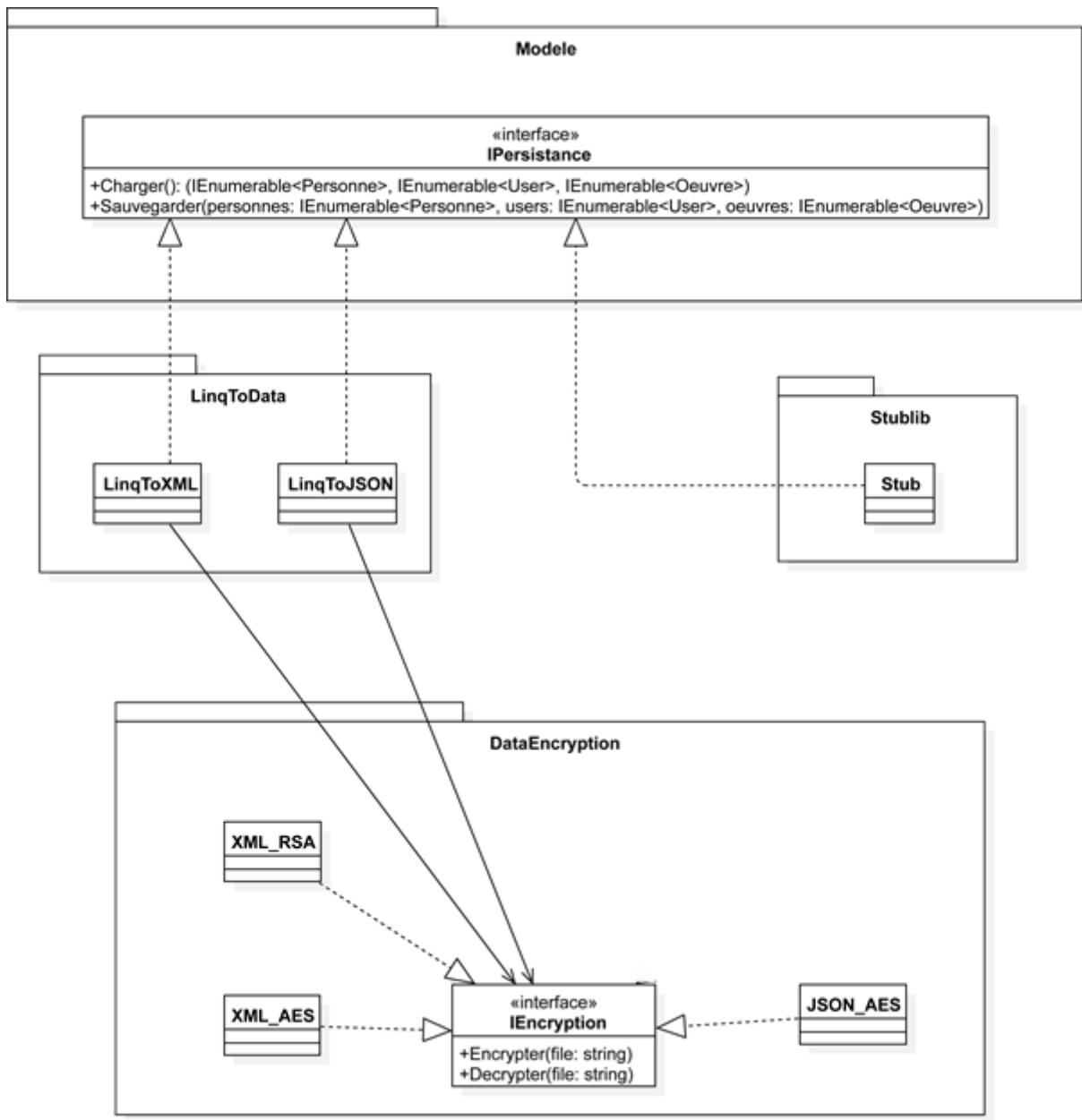
Ses autres attributs secondaires (Current...) servent à faciliter le binding lorsqu'un utilisateur modifie son profil avant de sauvegarder.

On trouve également dans le Manager deux attributs des types de deux interfaces :

- `_factory` du type `IFactory` : une instance d'une classe fille de l'interface qui constitue une partie d'un patron de conception que nous détaillerons en dernière partie.
- `_persistance` du type `IPersistence` : une instance d'une classe fille de l'interface qui constitue notre persistance que nous détaillerons plus bas.

Cette classe sert uniquement d'intermédiaire entre les classes énumérées ci-dessus et les vues de l'application, elle comporte toutes les méthodes permettant aux admins de modifier, ajouter ou supprimer une Œuvre, une Personne ou un User. On y trouve également toutes les méthodes utiles dans l'utilisation de l'application (pour la création d'un compte ou le post d'un avis par exemple).

8) Persistance



Se trouve donc dans la solution, et plus particulièrement dans le package Modele, une interface IPersistence, comportant deux méthodes à implémenter : Charger et Sauvegarder.

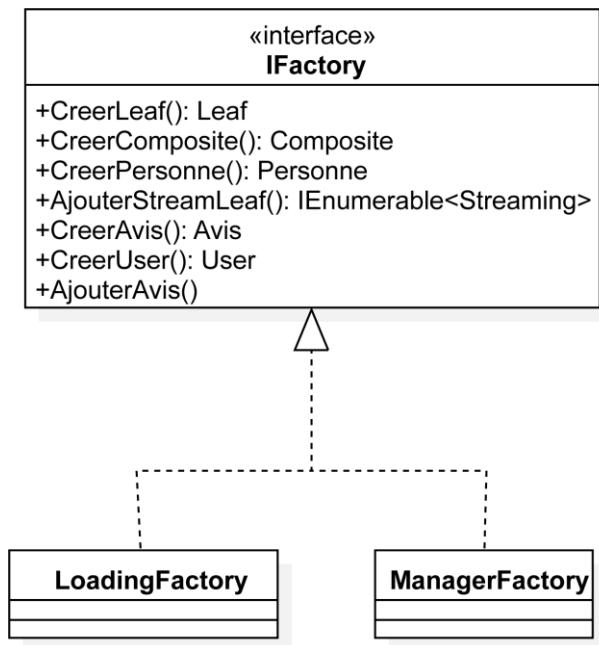
Comme expliqué dans la première partie, nous avons trois packages destinés à la persistance des données.

Parmi eux, on trouve :

- StubLib : Contient le fichier, désormais inutile, Stub qui, jusqu'à la création d'une persistance, servait à remplir l'application de fausses données que nous avons rentrées afin de travailler sur le Data Binding et autres évènements et liaisons. Cette classe hérite de l'interface IPersistence.

- LinqToData : Comporte LinqToXML et LinqToJson, tous deux implémentants l'interface IPersistence. Ces deux classes remplacent la classe Stub, détaillée ci-dessus. En effet, elles permettent, à l'aide de plusieurs méthodes que nous n'avons pas affichées par souci de simplicité, de créer un fichier de sauvegarde avec l'extension correspondante d'y écrire les données lors de la fermeture de l'application, ainsi que de les lire lors de l'ouverture.
- DataEncryption : Package qui contient une interface et quatre classes l'implémentant. L'interface IEncryption comporte deux méthodes que sont Encrypter et Decrypter. Nous utiliserons, pour le fichier XML les encryptages RSA et AES, et pour le fichier JSON le chiffrement AES. Ces trois chiffrements seront effectués par des classes respectives.

9) Factory



On voit sur ce schéma la représentation de l'interface **IFactory**. Cette interface comporte plusieurs méthodes servant à créer ou ajouter différents composants à l'application via le Manager.

Petite précision concernant **AjouterStreamLeaf** : elle sert à créer un **IEnumerable** de **Streamings** suivant un **Dictionary<Plateforme, string>**.

Deux classes héritent de cette interface :

- **LoadingFactory** : Classe utilisée dans la persistance, implémentant toutes les méthodes de l'interface mais ne faisant aucune vérification. Elle est simplement une auxiliaire servant à charger toutes les informations en mémoire.
- **ManagerFactory** : De même, elle implémente toutes les méthodes de l'interface dont elle hérite. En revanche, elle effectue toutes les vérifications nécessaires concernant les ajouts ou créations (par exemple pour éviter deux utilisateurs ayant le même pseudo).

10) Conclusion

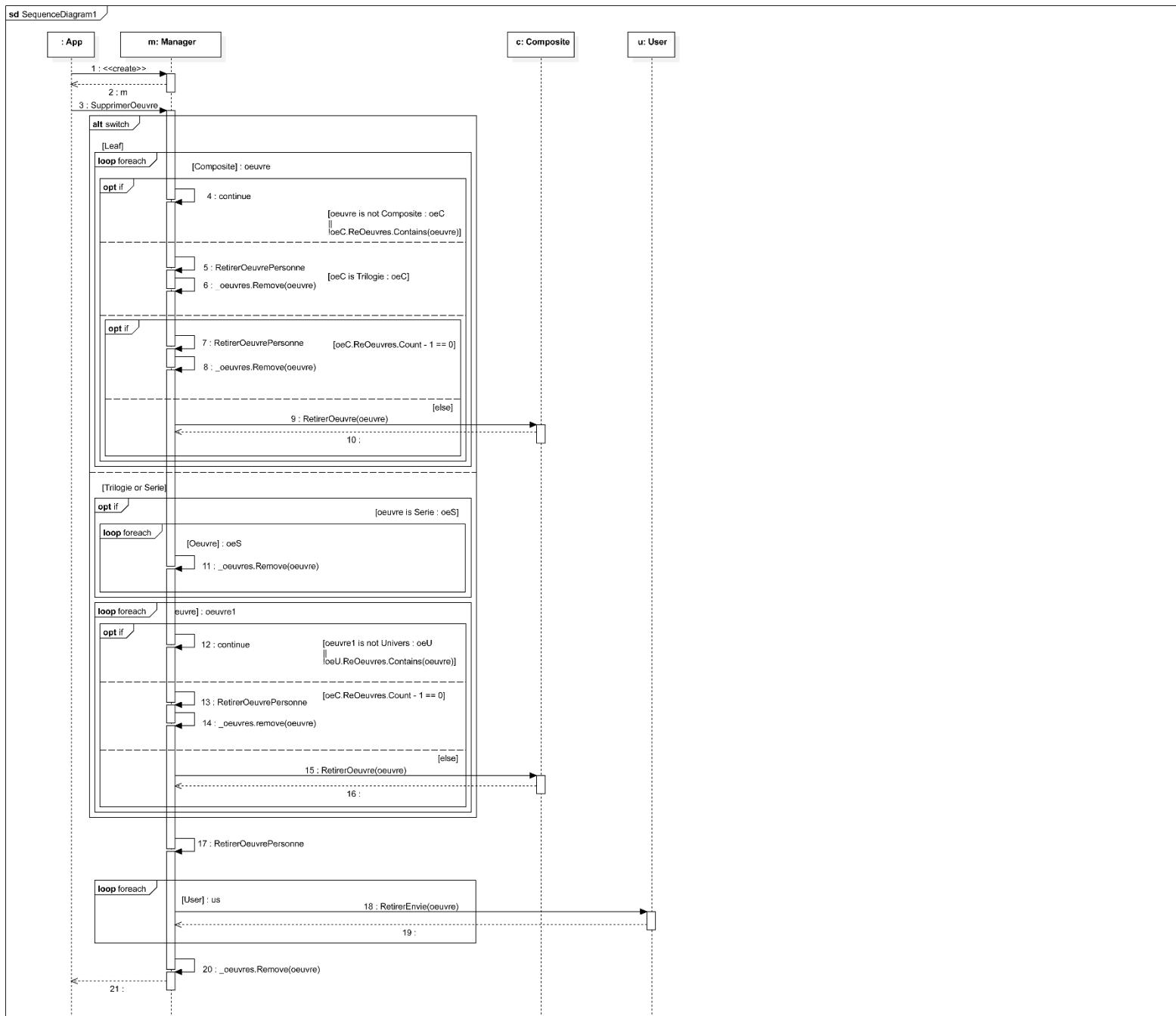
En conclusion, l'architecture de l'application est en partie pensée pour être « réutilisable ».

En revanche, quelques points faibles sont à déplorer. Le manque de temps pour perfectionner nous a empêché de réaliser :

- Des plateformes plus « malléables » : Un enum de plateformes entraîne une maintenabilité assez médiocre s'il s'agissait d'ajouter de nouvelles plateformes de streaming (ce qui arrive très certainement).
- Enfin et surtout, un simple booléen IsAdmin est assez manichéen dans sa vision des choses dans la mesure où une application a besoin d'être gérée par beaucoup de personnes. On aura donc besoin de beaucoup de rôles différents en fonction du rôle de modération de chacun. Un attribut correspondant à une interface IRole serait certainement préférable.

VIII – Diagramme de séquence

a) Diagramme



b) Explications

Nous avons choisi de représenter dans ce diagramme la suppression d'une Œuvre depuis l'Appli.

L'Appli commence par instancier un Manager « m » qui lui est ensuite attribué.

Cette même Appli appelle ensuite la méthode « SupprimerŒuvre » du Manager qui prend en paramètres :

- Un User (qui fait la requête) ;
- Une Œuvre (à retirer).

De là, deux cas de figure majeurs ont été représentés :

- ***L'œuvre passée en paramètre est une Leaf***

On parcourt la liste d'œuvres du Manager en effectuant plusieurs tests :

Si l'œuvre est Composite, on la renomme oeC, ou alors si oeC ne contient pas l'œuvre passée en paramètre alors on *continue*.

Sinon si oeC est une Trilogie, alors on la retire des listes d'œuvres des Personnes qui y participent, puis on supprime la Trilogie de la liste d'œuvres du Manager.

Sinon, on effectue un autre test :

Si la longueur de la liste d'œuvres du composite est 1, alors on le retire des listes d'œuvres des Personnes qui y participent, puis on le supprime de la liste d'œuvres du Manager.

Sinon, on retire simplement la Leaf de la liste d'œuvres du Manager et du Composite.

- ***L'œuvre passée en paramètre est une Trilogie ou une Serie***

On effectue plusieurs tests :

Si l'œuvre est Serie, pour chaque œuvre dans la Serie, on la supprime de la liste du Manager.

Sinon on parcourt la liste d'œuvres du Manager et pour chaque œuvre :

Si l'œuvre est Univers, on la renomme oeU, ou alors si oeU ne contient pas l'œuvre passée en paramètre alors on *continue*.

Sinon si la longueur de la liste d'œuvres de l'univers est 1, alors on le retire des listes d'œuvres des Personnes qui y participent, puis on le supprime de la liste d'œuvres du Manager.

Sinon, on retire simplement la Serie ou la Trilogie de la liste d'œuvres du Manager et du Composite.

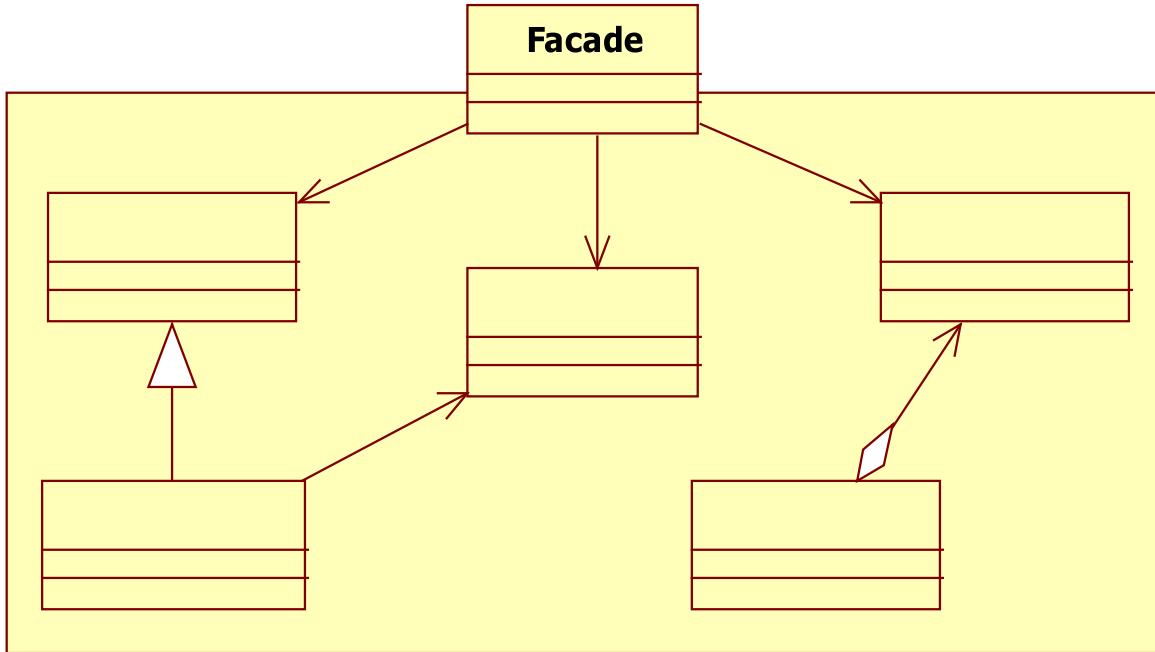
Après tout cela, on retire l'œuvre en question des attributs des personnes y ayant participé.

On retire l'œuvre des listes d'envies des utilisateurs.

Enfin, on supprime l'œuvre de la liste du Manager puis on rend la main à App.

IX – Explications des patrons

1) La façade



Le principe de fonctionnement d'une façade est simplement d'offrir une interface unifiée au client (s'il ne s'aventure pas à aller plus loin que celle-ci) pour unir les sous-systèmes potentiellement complexes d'une application.

Plus particulièrement, elle sert à notre projet à simplifier l'interaction avec des sous-systèmes qui se sont compliqués au fur et à mesure du projet et à rendre le code plus portable.

Notre projet comporte une façade unique, les sous-systèmes du Modèle communiquent directement entre eux.

La façade fait partie intégrante de notre application, et ce pour beaucoup de raisons. Nous tenterons d'en exposer quelques-unes :

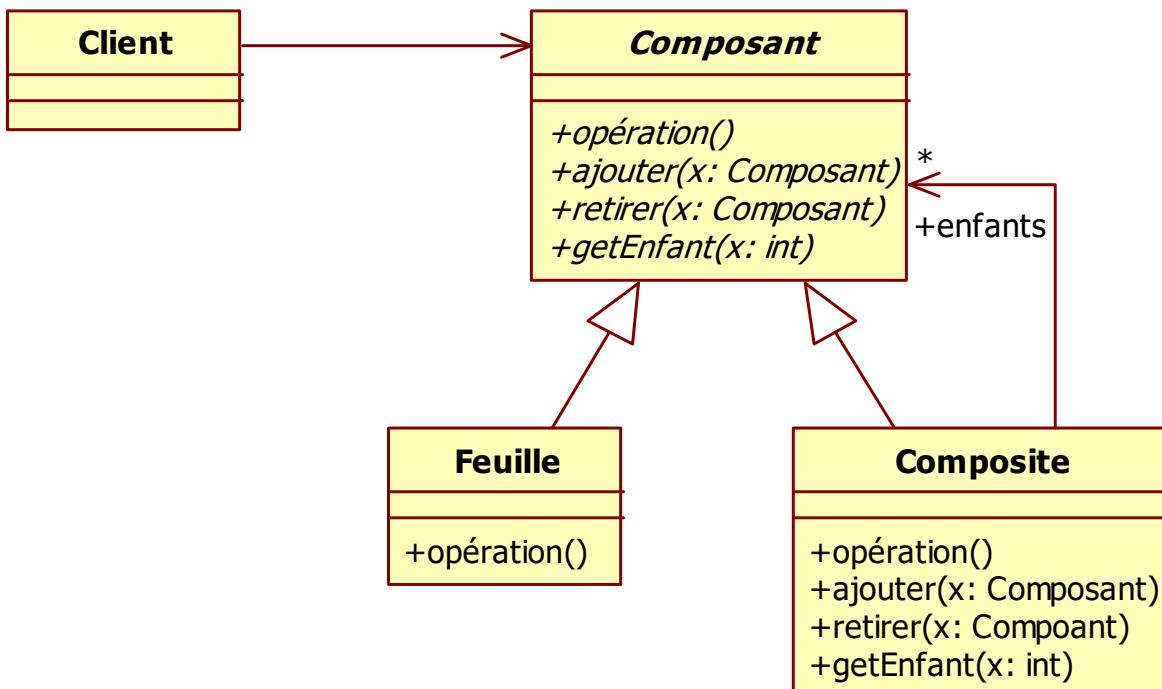
- **La propreté du code** : La façade permet de concentrer toutes les méthodes qui seront utilisées dans l'application (en appuyant sur les boutons des vues XAML par exemple). On évite ainsi de disperser ces mêmes méthodes dans toutes les autres classes, plus pratique pour la maintenance en cas de bug.
- **Une façade partielle** : En plus d'utiliser ce patron, nous avons décidé de le diviser en plusieurs parties en utilisant les partial class. Ainsi, nous avons pu découper la façade en :
 - o Une façade Générale
 - o Une façade Oeuvre
 - o Une façade Streaming
 - o Une façade User

Ils comportent chacun les méthodes associées aux classes qui se trouvent dans leurs noms, pour encore plus de clarté dans les fichiers.

- **Une encapsulation efficace** : Enfin, la façade permet aux projets qui l'utilisent, d'acquérir une sécurité supplémentaire en encapsulant toutes les méthodes des classes dans des méthodes de celui-ci, on contrôle ainsi les usages que les futurs utilisateurs ou les futurs collaborateurs auront de l'application en les empêchant de faire de grosses bêtises.
C'est pourquoi on voit dans le projet que la plupart des méthodes des classes autres que les Managers sont en internal.

Enfin, le fonctionnement de ce patron de conception est prouvé par le diagramme de séquence (voir page 6). On y montre notamment que la suppression d'une Œuvre de l'application est grandement facilitée. Ceci repose sur le fait que le Manager comporte toutes les Œuvres, Users et Personnes de l'application parmi ses attributs, on appelle donc ses méthodes pour supprimer ses propres attributs. Dans ces méthodes, des vérifications ont lieu et donnent naissance au fonctionnement que l'on peut observer dans le diagramme de séquence.

2) Le Composite



Ce patron de conception permet au projet de gérer d'une unique façon les composants et listes de composants (Trilogies, Univers, Séries, Films et Episodes) en utilisant une arborescence d'héritage.

Nous avons utilisé le Composite pour représenter une hiérarchie de trois niveaux de l'objet œuvre.

Nous avons assigné la classe œuvre au Composant, elle implémente toutes les variables et méthodes communes à tous ses descendants.

La classe Leaf constitue la Feuille, elle possède deux enfants : Episode et Film. Elle est abstract et constraint ses enfants à contenir certaines méthodes et attributs (leur permettant entre autres de posséder une liste d'instances de Streamings et de la gérer).

La classe Composite a été nommée selon son assignation, elle possède des enfants : Trilogie, Serie et Univers. Elle stocke les instances d'œuvre (composant) dans une liste attribut. Elle n'implémente pas les méthodes liées à la gestion de ses enfants. En effet, nous avons jugé plus pratique de créer ces méthodes dans ses classes enfants, pour nous permettre de bloquer l'ajout de certaines œuvres dans la liste d'œuvre.

De la même façon que le Manager, le Composite est assez présent dans le projet pour plusieurs raisons :

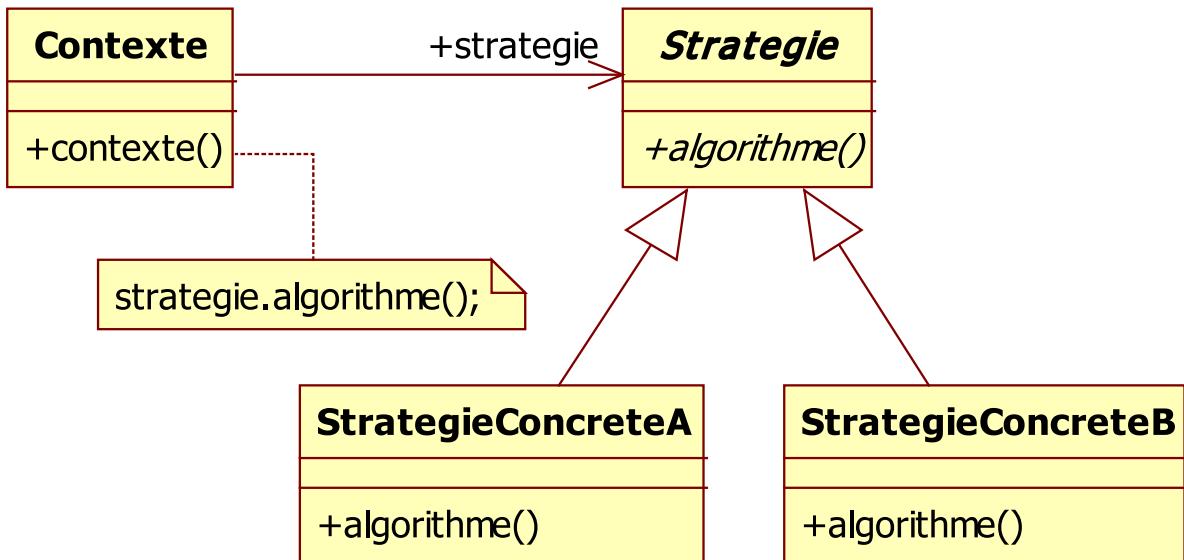
- **La maintenance** : Si à l'avenir il est nécessaire d'ajouter une Leaf ou un Composite il suffira de créer une classe héritant de la classe abstraite correspondante.
- **La lisibilité** : De nouveau, séparer les fichiers permet une meilleure lisibilité. On sépare les propriétés de chaque classe pour mieux gérer les possibilités qu'elles offrent.

- **La viabilité de ce patron de conception :** En effet, dans notre projet, ce patron convient particulièrement bien par l'architecture de l'application elle-même. Le Composite (Trilogie, Univers ou Série) s'apparente à tout ce qui peut contenir une Leaf (Episode ou Film). L'analogie paraît donc évidente.
Le fonctionnement de ce patron est lui aussi en partie prouvé dans le diagramme de séquence (voir page 6).

Enfin, ce patron de conception permet de factoriser considérablement les traitements des Œuvres en général. Les méthodes communes à toutes les sous-classes se trouvent dans Œuvre et peuvent être utilisées par chacune d'elles.

D'un autre côté, les méthodes nécessitant d'être présentes sur certaines classes uniquement ou d'effectuer des traitements différents en fonction de la nature de la classe appelante ont pu être dissociées sans problème grâce à ce patron.

3) La Stratégie



Trois interfaces sont intégrées à la solution :

- IEncryption
- IPersistance
- IFactory

Ces trois interfaces font partie de trois stratégies différentes avec des classes filles qui en héritent. Nous avons décidé d'utiliser ce patron à plusieurs reprises pour les raisons suivantes :

- **La clarté et la maintenance** : En effet, ce patron nous permet de gérer le code (encore une fois) de façon plus propre. De plus, si une maintenance venait à être faite, les classes filles des interfaces ne pourraient pas perdre leurs méthodes nécessaires et obligatoires au fonctionnement de l'application. En revanche, ces mêmes méthodes pourraient être modifiées et ainsi le bon fonctionnement de l'application serait perturbé, ce qui est un point faible.
- **Les attributs du type des interfaces** : Nous avons décidé de créer deux classes filles à l'interface IFactory. De cette façon, il nous a été possible de dissocier le fonctionnement de l'interface dans la persistante de celui qu'elle a dans le fonctionnement de l'application.
- **La viabilité de ce patron de conception** : Comme vu plus haut, ce patron se révèle relativement viable en ce qui concerne la maintenance et l'extensibilité de l'application.

Dans ce patron a été intégrée une partie d'un autre : La Fabrique. Nous avons mêlé ces deux propositions d'architecture en créant l'interface IFactory et en la concrétisant en deux classes. En revanche, aucun produit concret n'a été assigné attribut d'aucune des deux classes filles.

X – Annexe

Pour la création d'œuvres ou autre vous devez être en administrateur. Pour cela 2 comptes existent déjà avec ces privilèges :

Pseudo : Samuel Mdp : Admin
Pseudo : Antoine Mdp : Admin

Pour ce qui est des différents types d'encryptions, une fois sélectionné il n'est plus possible d'en changer. Néanmoins si vous le souhaitez dans les fichiers source il existe un dossier Data qui contient une version non cryptée des données Json ou Xml. Si vous avez effectué des modifications entre temps elles ne seront pas sauvegardées dans ce fichier, il contient uniquement les données original fournies par la release actuelle de l'application.

Pour finir, pour ce qui est de regarder vos films et séries, si vous êtes sur Netflix, veuillez choisir votre profil une fois connecté pour être redirigé par la suite vers votre épisode. Cependant si vous êtes déjà sur le profil souhaité lors de votre connexion, veuillez rafraîchir la page avec F5 pour vous faire rediriger.