

CoFiApp
L'appli des films Indé !



Allavena Clément - Barland Estéban
Année 2017-2018
DUT Informatique - IUT Des Cézeaux

- Qu'est-ce-que c'est ?

CoFiApp est une application gratuite qui stocke des films avec leurs informations principales. La différence avec une simple base de données de films, c'est que tous les films présents dans l'application sont ajoutés par les utilisateurs de l'application. Tout le monde peut consulter la liste des films qui ont été ajoutés dans l'application. Pour chaque film, il y a différentes informations :

- Le titre
- L'affiche
- Le réalisateur ou les réalisateurs
- La date de sortie
- La liste des étiquettes associées au type du film
- Le synopsis
- La liste des acteurs avec le nom, prénom et une photo de profile

Il n'y a pas besoin de se créer de compte pour pouvoir ajouter un film dans l'application. Seuls les administrateurs de l'application ont besoin d'un compte. Il sera utilisé pour modifier ou supprimer des films.

Chaque utilisateur pourra signaler un ou plusieurs films si le contenu ne correspond pas. Pareil avec les réalisateurs. Un mail sera envoyé aux administrateurs afin qu'ils puissent modifier ça à tout prix.

- Pour qui ?

Nous avons pensé CoFiApp comme une application accessible à tous et pour tous. Ainsi chaque personne souhaitant découvrir de nouveaux films pourra venir, sans besoin de donner ses informations personnelles, utiliser notre application.

CoFiApp vise le tout public, néanmoins, l'application est dotée d'une partie entièrement dédiée aux films indépendants, chaque utilisateur pourra alors découvrir des films indépendants. Les réalisateurs peuvent aussi aspirer à cette particularité, et ainsi pouvoir profiter d'une meilleure visibilité de leurs travaux.

- Les Personas



"Si tu peux le rêver, tu peux le faire"

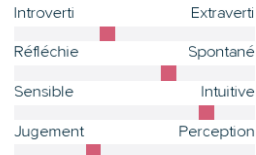
Age: 20

Travail: Étudiante en Droit

Relation: Célibataire

Lieu de Vie: Paris, France

Personnalité



Curieuse

Persévérante

Rêveuse

Objectifs

- Devenir Avocate au barreau de Paris.
- Fonder une famille.

Frustrations

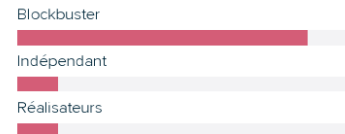
- Elle déteste ne pas finir une tâche.

Bio

Sarah est une jeune femme étudiant le droit à Paris. Originnaire de Paris et de nature curieuse elle a toujours voulu aidé les autres. C'est après un stage dans un cabinet d'avocat qu'elle trouve sa vocation, depuis ce jours elle persévère dans cette voie la.

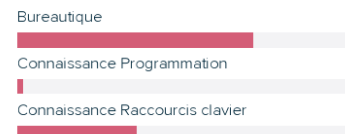
Intérêts

Cinématographique



Compétences

Informatique



Habitudes Informatique

- Web (élevée)
- Logiciel Office
- Réseau sociaux



"Rien de grand ne s'accomplit dans le monde sans passion"

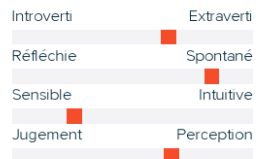
Age: 32

Travail: Intermittent du Spectacle

Relation: En couple, Sans enfants

Lieu de Vie: Avignon, France

Personnalité



Optimiste

Passionné

Audacieux

Objectifs

- Écrire une pièce de Théâtre.
- Voyager.
- Faire une tournée de spectacle mondiale.

Frustrations

- Déteste le travail mal fait
- N'aime pas donné ces informations personnelles

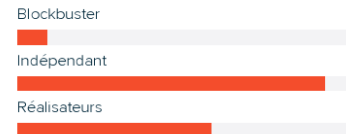
Biographie

Connor est un Irlandais, vivant en France à Avignon depuis plus de 15 ans, il est venu en France poursuivre ces études dans l'informatique. Passionné de théâtre et de spectacle, il a ensuite rejoint une troupe indépendante de comédien à Avignon, chez qui il joue depuis maintenant 11 ans.

Il est atteint d'une malformation du poignet droit qui l'empêche de ce servir majoritairement d'une souris, sans quoi il subirait des douleurs et est sujet à des tendinites. Il connaît les raccourcis claviers car il a appris à s'en servir. Se sert très peu de sa souris.

Intérêts

Cinématographique



Compétences

Informatique



Habitudes Informatique

- Web (faible)
- Réseau sociaux

³⁵₁₇ Leur utilisation de l'application

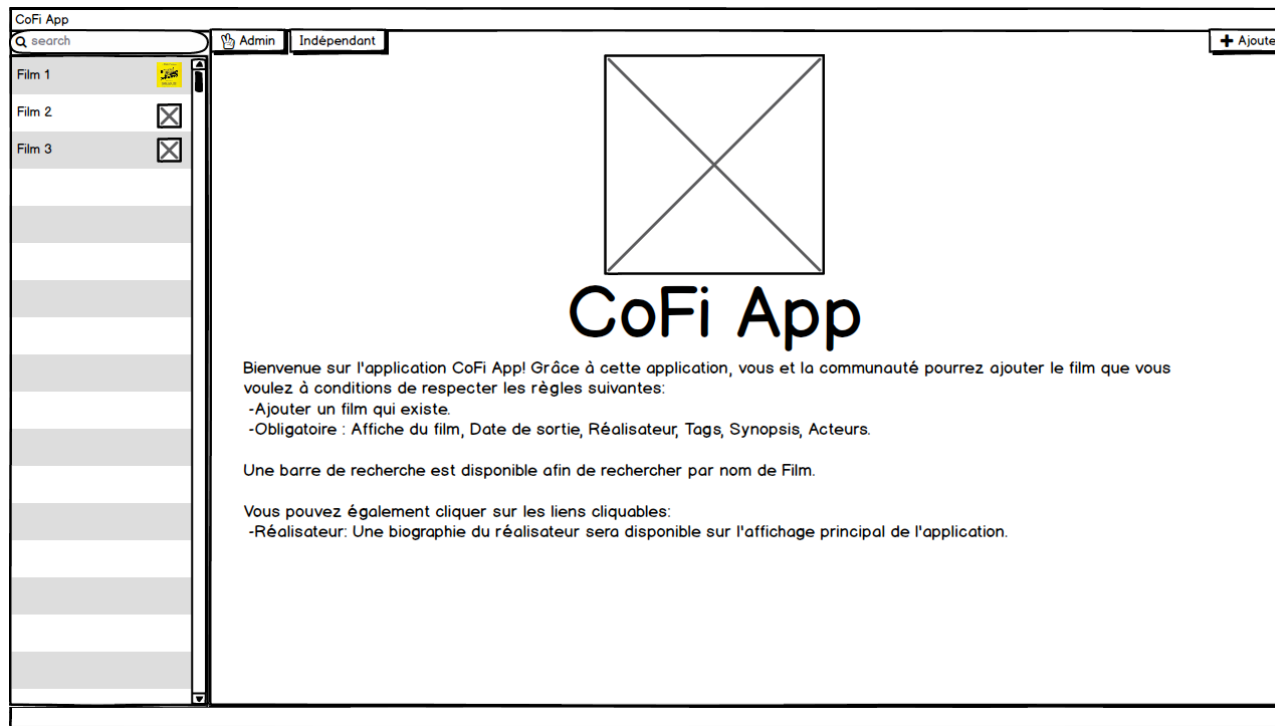
-Sarah voit en l'application un moyen pour elle de découvrir des Films de type Blockbuster qu'elle ne connaîtrait pas encore. Sa nature curieuse la forcerait à rechercher tout type d'informations sur les films en question. Elle y découvrirait ainsi toutes les informations qui lui sont nécessaires pour découvrir un film. Sarah aimerait bien qu'il soit possible d'intégrer un lien cliquable qui permettrait d'acheter le film via un site externe.

-Connor voit en l'application un moyen pour lui de découvrir, et surtout de faire découvrir aux autres personnes l'existence de grands nombres de films indépendants. L'ergonomie de l'application lui permet de très peu utilisé sa souris, et donc de lui permettre d'utiliser l'application comme les autres sans tenir compte de son handicap. Si jamais sa troupe se met au long-métrage il sera ravi de la faire découvrir aux autres via cette application !

Description des Sketchs

Éléments redondants lors des descriptions :

- Situés en haut de la page, les différents boutons permettent d'accéder à diverses fonctionnalités. Le bouton “admin” permet aux gérants de se connecter sur compte spécial qui a plus de droits qu’un simple utilisateur sur l’application. Une fois connecté en tant qu’administrateur, le bouton “supprimer” apparaît dans la fenêtre et il permet de supprimer des films ou des réalisateurs. Le bouton “ajouter” permet d’ajouter dans l’application des films ou des réalisateurs. Le bouton “indépendants” va permettre de mettre en avant les courts ou longs métrages de studios indépendants.
- Sur la partie gauche, on trouve la liste des tous les films disponibles avec une barre de recherche pour une navigation plus simple.



Accueil :

- Cette vue contient le logo ainsi que le nom de l'application et en dessous de ces éléments, il y a une brève description du fonctionnement de l'application avec les différentes règles à suivre.
- Le bouton Admin permet d'accéder à la page de connexion Administrateur de l'application
- Le bouton Indépendant permet d'accéder à la page Indépendant de l'application
- Le bouton Ajouter ouvre une boîte de dialogue permettant à l'utilisateur de choisir ce qu'il souhaite ajouter (Film, Réalisateur).

Add Box :

- Cette fenêtre est ouverte lorsque l'on clique sur le bouton "Ajouter" et elle permet d'ajouter un film ou un réalisateur.
- Si le bouton Film est activé, l'utilisateur sera redirigé vers La page Ajout Film.

- Si le bouton Réalisateur est activé, l'utilisateur sera redirigé vers La page Ajout Réalisateur.
- Le bouton Annuler permet de revenir à la page précédente.

CoFi App

search

Film 1

Film 2

Film 3

Accueil

Ajouter un Réalisateur

Nom:

Prénom:

Date naissance:

Date mort:

Nationalité:

Biographie:

Photo:

Add Director :

- C'est la partie qui va permettre d'ajouter un réalisateur en fonction des films déjà existants dans l'application.
- On a différents champs à remplir avec du texte contenant le nom, le prénom, la date de naissance, la date de mort (laissé le champ vide s'il n'y en a pas), la nationalité et la biographie.
- On peut ajouter une image qui va servir de photo de profil pour le réalisateur. Si cette option est ignorée, le réalisateur aura une photo de profil par défaut qui pourra être modifiée par un administrateur.
- Pour finir le bouton "ajouter" qui permet de valider et d'enregistrer le réalisateur dans la base de données.
- Le bouton Accueil permet de revenir à la page d'accueil. **Attention : Cliquer sur ce bouton avant d'avoir**

confirmé l'ajout annule l'ajout du film !

CoFi App

search

Accueil

Ajouter un film

Choisissez parmi la liste de Réalisateur connus:

Réalisateur:

Titre:

Date de sortie:

Tags:

Acteurs (Nom, Prénom):

Synopsis:

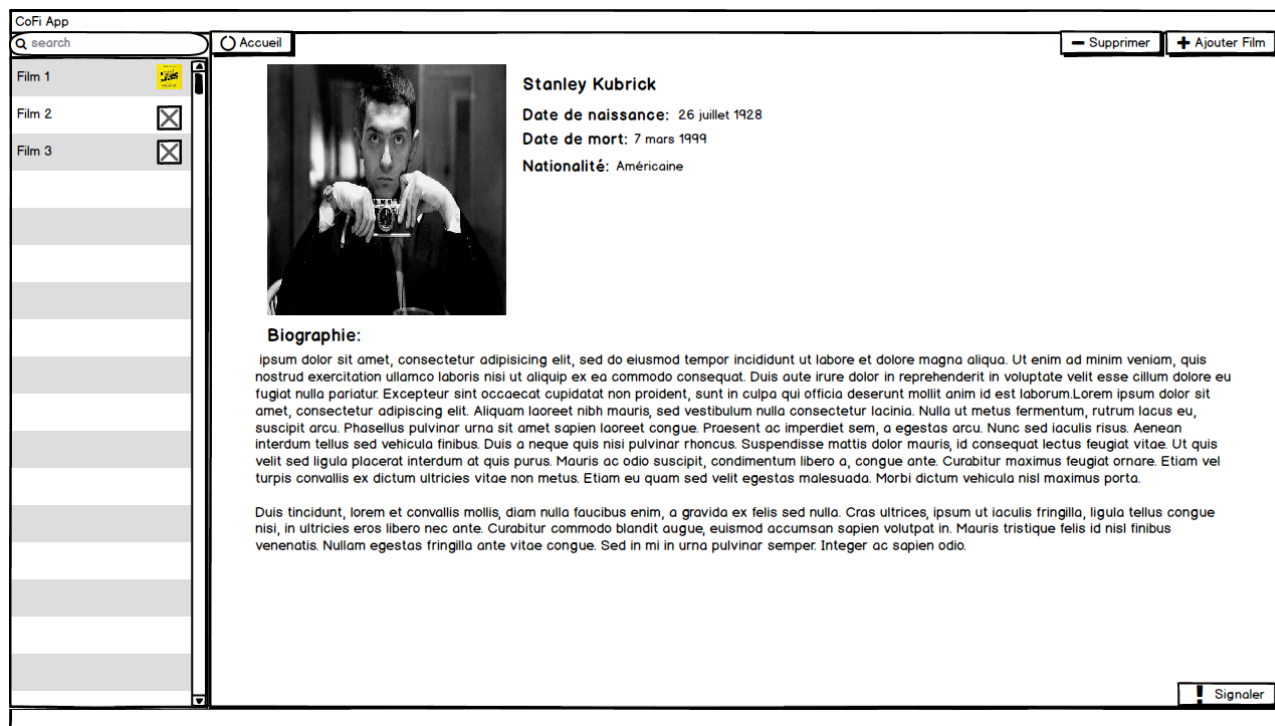
Affiche du film:

Add Movies :

- Cette page permet d'ajouter un film dans la collection de films. Il y a différents champs à remplir pour pouvoir ajouter un film.
- Pour associer un réalisateur au film, il faut qu'il soit déjà enregistré.
- On a aussi le titre, la date de sortie, le synopsis et la liste des tags qui représentent les thèmes du film à ajouter. La liste des tags peut être modifiée à tout moment, pour ajouter un Tag il suffit de sélectionner un tag dans la liste de droite et de l'ajouter. Pour en supprimer un il suffit de sélectionner un tag dans la liste des tags du film et de le supprimer.
- On peut aussi choisir la liste des acteurs principaux du film et en ajouter s'ils ne sont pas déjà présents.
- Enfin une image pour le film peut être ajoutée et en cliquant sur ajouter le film sera maintenant présent dans la liste des films.

Delete Box :

- Cette petite fenêtre apparaît lorsque l'on appuie sur le bouton supprimer.
- Une boîte de dialogue pour afficher une erreur s'affiche si l'utilisateur n'est pas connecté en tant qu'administrateur.
- Si l'utilisateur est un administrateur, alors, une boîte de dialogue est alors ouverte pour demander une confirmation de la suppression du dît film ou réalisateur, si Oui alors le film, ou réalisateur, sélectionné est supprimé de la base de donnée de l'application.



Ex Real :

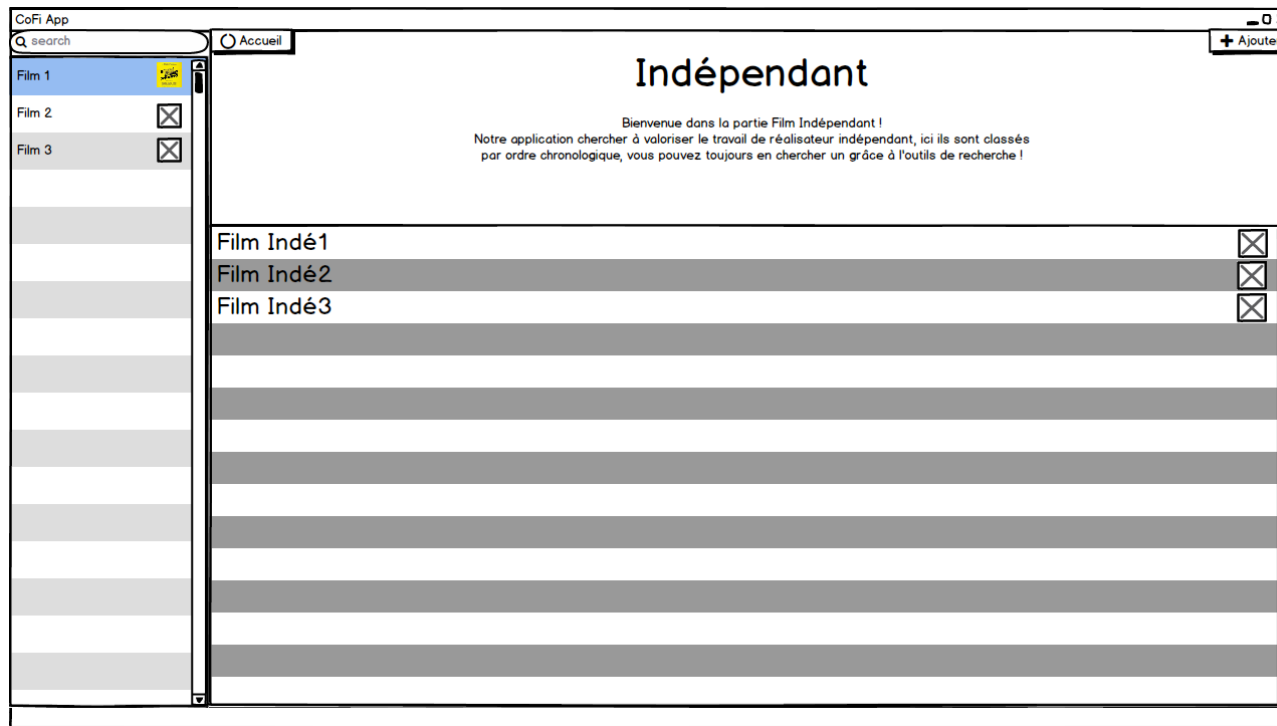
- Cette page contient toutes les informations nécessaires concernant un réalisateur.
- Il y a les informations principales avec son nom, son prénom, sa date de naissance, sa date de mort (si nécessaire), sa nationalité et sa biographie.

- Les boutons ajout et suppression fonctionne de la même manière que sur les autres pages.
- Le bouton Signaler permet de signaler un réalisateur, il envoie un mail d'avertissement aux administrateurs de l'application. Ainsi les administrateurs seront mis au courant et pourrons modifier le réalisateur ou le supprimer si besoins est._



Ex Select :

- Cette page sera disponible lorsqu'on cliquera sur un film présent dans la liste de gauche.
- On peut voir l'image du film, le titre, le réalisateur, la date de sortie, les différents tags et le synopsis.
- En dessous des diverses informations du film est présente la liste des acteurs principaux du film.
- Comme avec la page du réalisateur, le bouton « Modifier » est disponible seulement pour les administrateurs et permet de changer les informations du film.



Indépendant :

- Cette page permet de recenser tous les films créés par des réalisateurs indépendants. Cela permet de mettre ces films en avant.
- On a donc une liste de films classés par ordre chronologique.

CoFi App

search

Accueil

Modif

Film 1

Film 2

Film 3

Modifier un film (Admin)

Réalisateur: InfoRéal

Titre: InfoTitre

Tags: Liste Tag du Film - Supprimer Indépendant + Ajouter

Date de Sortie: InfoDateDeSortie

Acteurs: Acteur1 - Supprimer Acteur2 Acteur3 + Ajouter

Synopsis: InfoSynopsis

Affiche: Ajouter image

Modify Movie :

- Comme pour la page pour modifier un réalisateur, on pourra aussi changer les informations d'un film. Le bouton Modifier doit être enfoncé afin de débloquent les cases à modifier. Un tag peut être supprimé en le sélectionnant dans la liste des tags du film, et en appuyant sur le bouton de suppression. Un tag peut être ajouté en sélectionnant un tag existant et en appuyant sur le bouton ajouter.

CoFi App

search

Accueil

Film 1

Film 2

Film 3

Username

PassWord

Se connecter

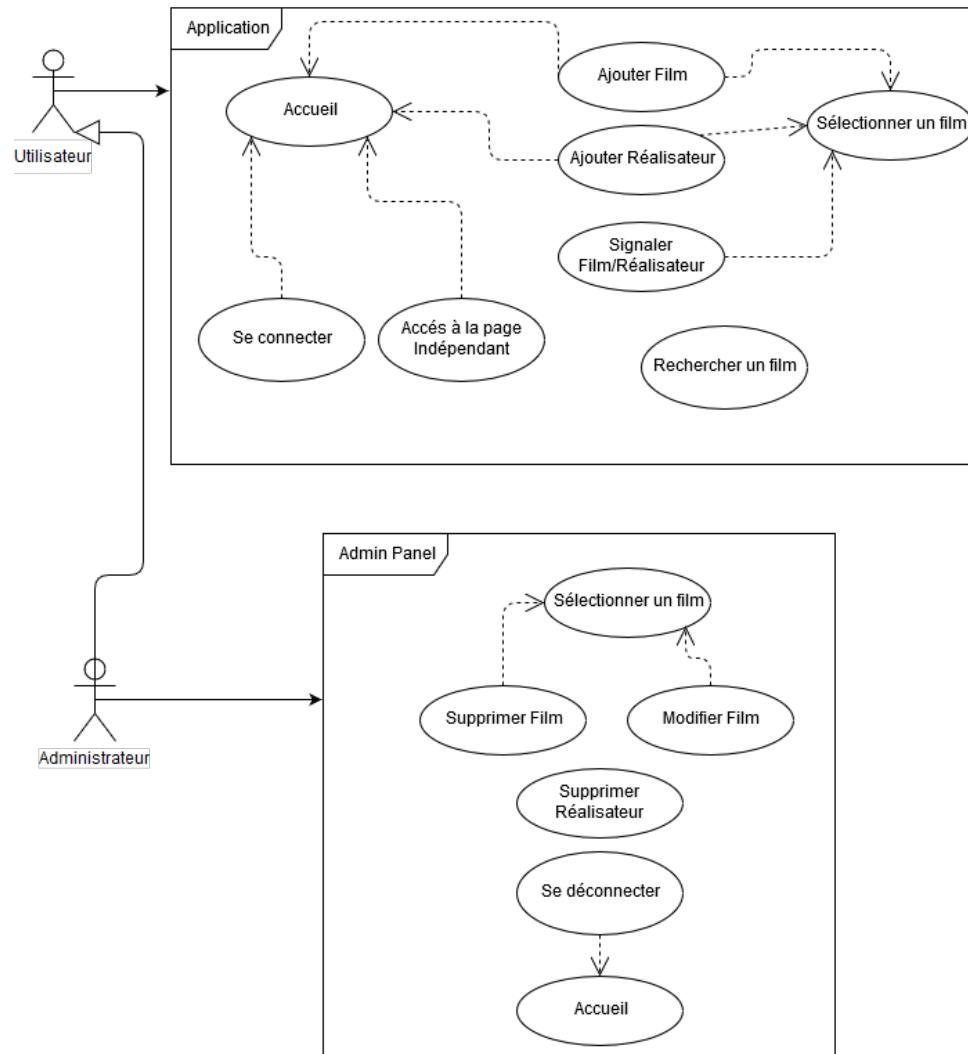
Quitter

Se Déconnecter

Panel Admin :

- Cette petite fenêtre apparaît quand on clique sur le bouton « admin » situé à l'accueil de l'application.
- Avec un nom d'utilisateur et un mot de passe, on pourra se connecter en tant qu'administrateur à l'application afin de modifier les informations de film, et aussi de pouvoir supprimer un film ou un réalisateur.
- L'administrateur revient sur cette page pour se déconnecter.

Description du Diagramme de Cas d'utilisation



Notre application prend en compte 2 types d'utilisateurs, un utilisateur lambda, et un Administrateur qui dérivent de l'utilisateur Lambda car l'Administrateur possède tous les droits de l'utilisateur plus ceux Administrateur.

Coté Utilisateur :

L'utilisateur n'a pas besoin de se connecter pour accéder à notre application. Il a juste à la lancer pour pouvoir pleinement l'utiliser. Il peut ajouter un film, ou un réalisateur, directement via l'accueil sans passer par une page spécifique. Nous avons laissé la possibilité à l'utilisateur de Signaler un film, ou un réalisateur, afin de signaler aux Administrateurs un contenu inapproprié. Pour ce faire l'action Signaler dépend de la sélection d'un film ou d'un réalisateur. L'utilisateur ne peut signaler s'il n'a pas au préalable sélectionné un film.

L'utilisateur peut également directement et à tout moment, rechercher un film spécifique. Il n'est pas obligé de se souvenir totalement du nom du film, ni même respecter les majuscules, il a juste à écrire sa demande, partiellement ou pleinement, pour que la liste de films se mette à jour.

L'utilisateur peut également accéder à la page Indépendant, accessible seulement depuis l'accueil.

Si l'utilisateur possède un compte administrateur, il peut alors se connecter avec son identifiant et mot de passe depuis la page Admin accessible seulement depuis l'accueil.

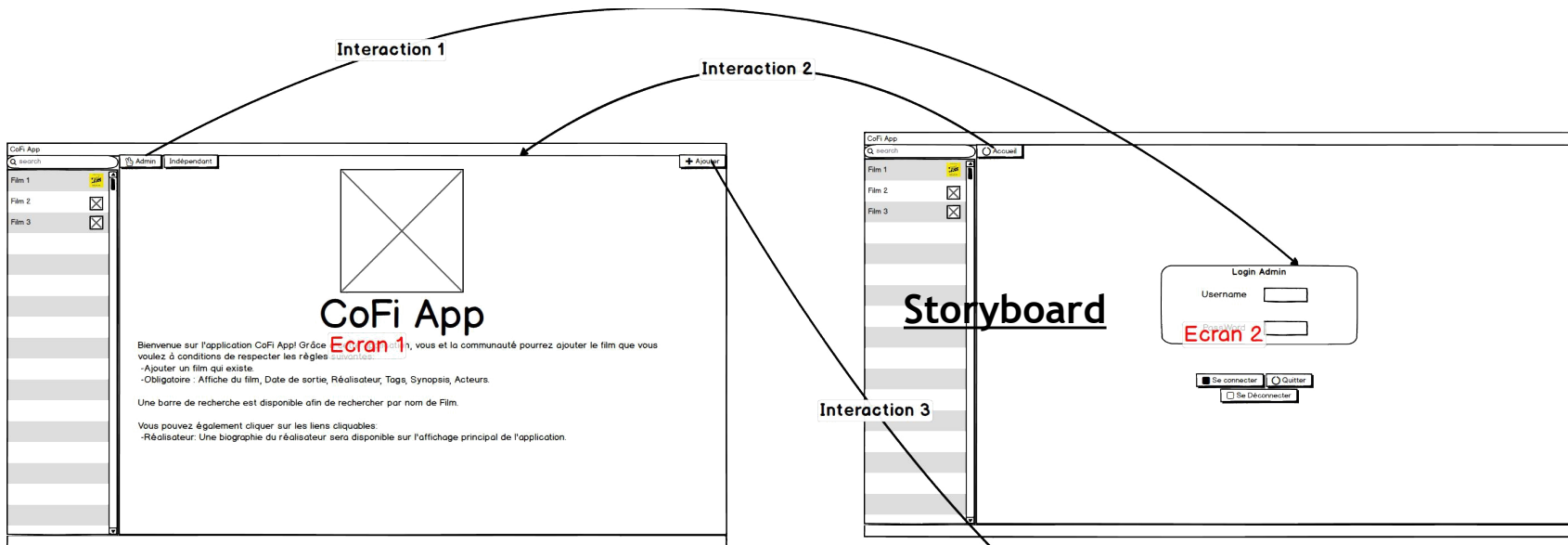
Coté Administrateur :

L'administrateur possède les mêmes droits et dépendances que l'utilisateur.

Contrairement à l'utilisateur il possède en plus :

- Le droit de suppression d'un film, à condition d'en avoir sélectionné un avant.
- Le droit de modification d'un film, à condition d'en avoir sélectionné un avant.
- Le droit de suppression d'un réalisateur, à condition d'en avoir sélectionné un avant.

L'administrateur peut se déconnecter seulement sur la page « Admin » accessible seulement depuis l'accueil.

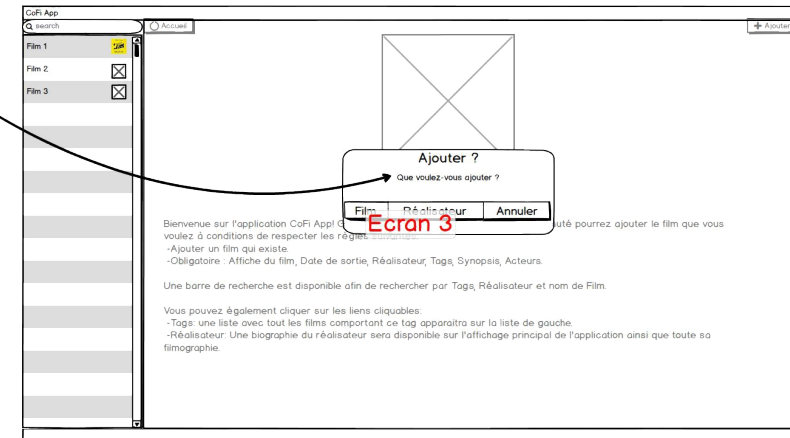


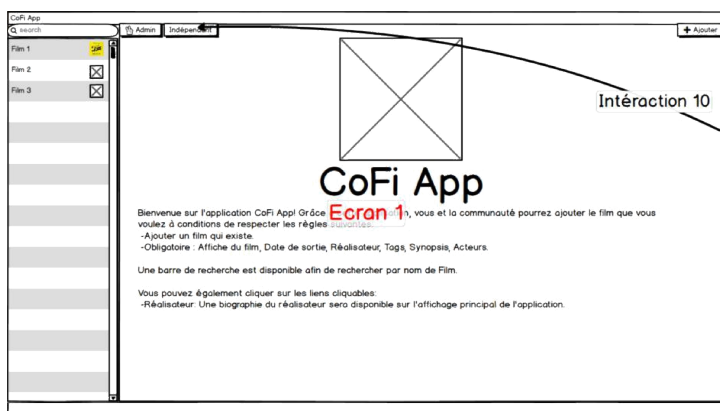
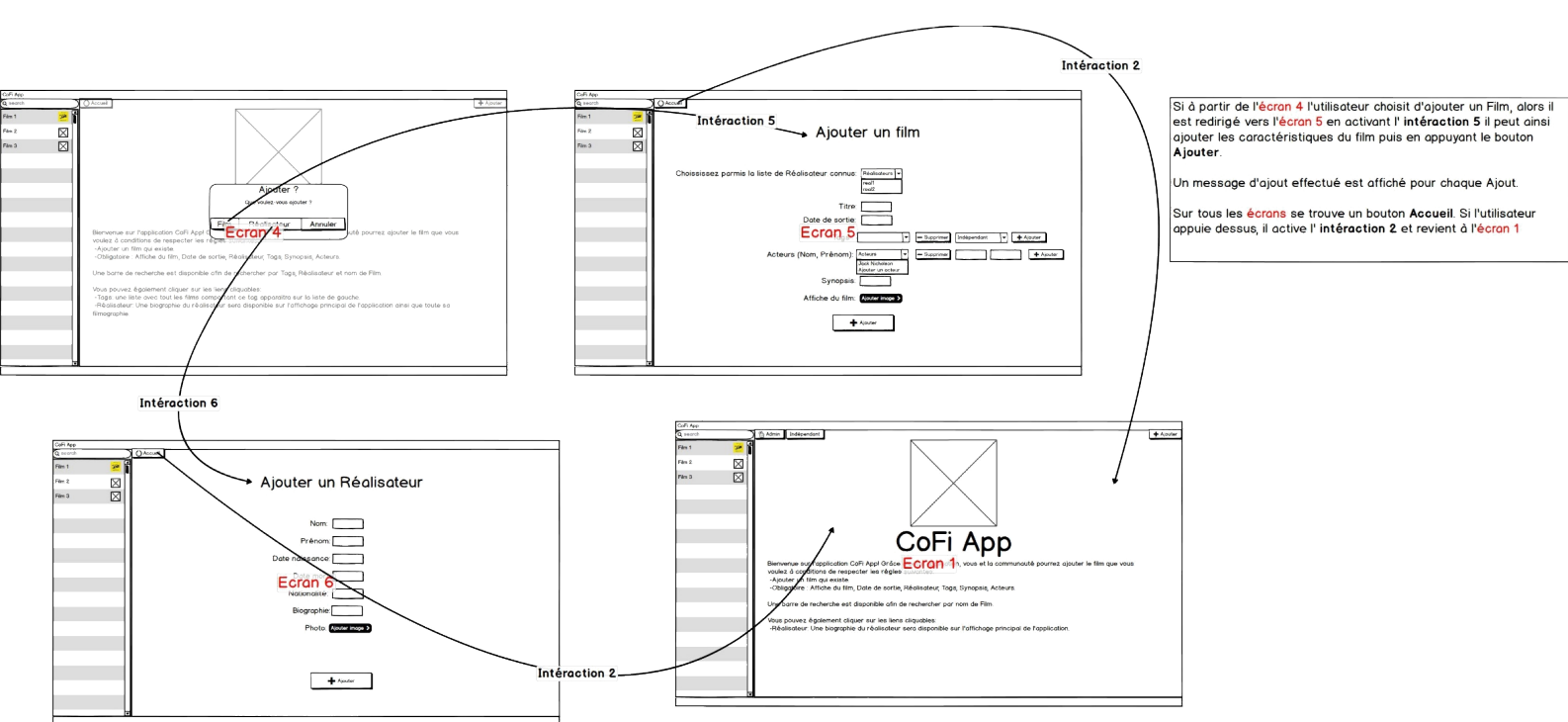
Notre application démarre à l'**écran 1**, à partir de là, plusieurs possibilités de choix:

Si l'utilisateur décide de cliquer sur le bouton "Admin", alors l'**interaction 1** emmène l'utilisateur vers l'**écran 2**. La possibilité pour l'utilisateur alors de se connecter en tant qu'Administrateur de l'application.

Si à partir de cette **écran 2** l'utilisateur décide de valider ou d'appuyer sur le bouton Accueil, alors l'**interaction 2** ramène notre utilisateur vers la page d'accueil, cf **Ecran 1**.

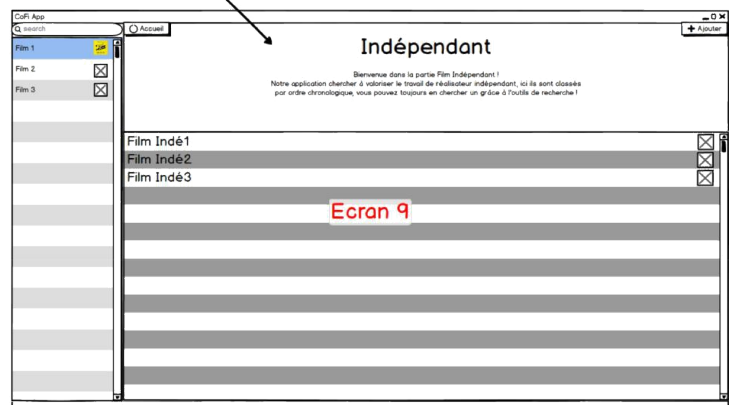
Si l'utilisateur arrivant sur l'**écran 1** décide d'appuyer sur le bouton ajouter (cf **Interaction 3**): Alors il est redirigé vers l'**écran 3** où il peut ainsi choisir la nature de son ajout: Film ou Réalisateur. Sinon, il peut annuler et il est redirigé vers l'**écran 1**.

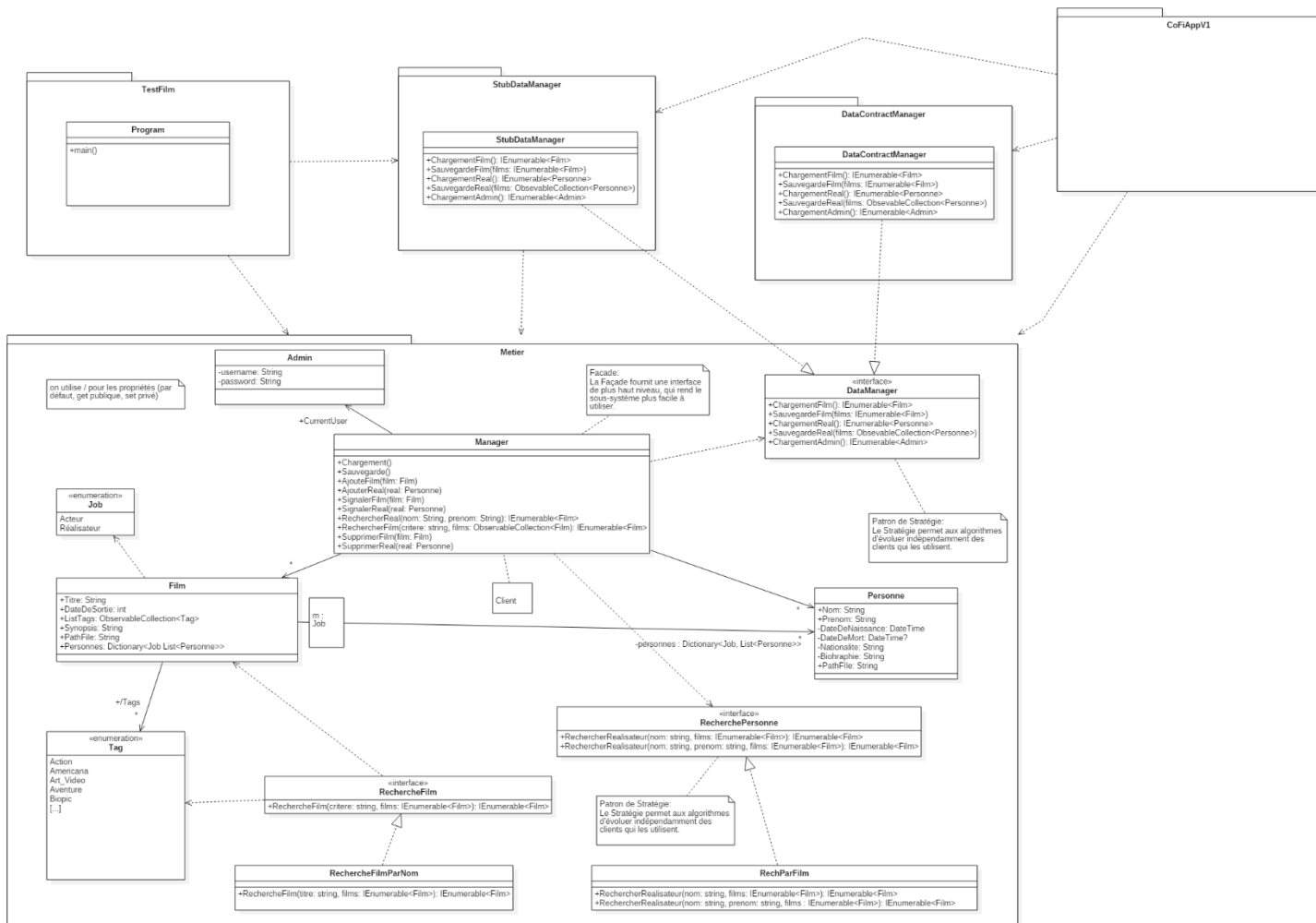




A partir de l'écran 1, l'utilisateur peut appuyer sur le bouton **Indépendant**, cela activera l'interaction 10 et amènera l'utilisateur à l'écran 9.

Un film est Indépendant lorsqu'il contient le Tag "Indépendant". Un film avec ce tag sera automatiquement dans cette page.





Description de l'architecture

Patron de Conceptions de l'application :

Patron de Stratégie :

Notre application utilise des interfaces, ces interfaces possèdent des méthodes, et ces méthodes sont réécrites dans une classe implémentant cette interface.

Le Stratégie permet aux algorithmes d'évoluer indépendamment des clients qui les utilisent.

Nous utilisons ce patron car il permet de rendre notre programmation beaucoup plus indépendante et évolutive.

Patron de Façade :

La classe Manager de notre application, fait office de Façade de l'application. Le Manager est lié à toute les classes de l'application, ce qui rend l'application unifié est plus facile d'accès.

Tout d'abord, la classe la plus importante est le "Manager". Il est le cœur de l'application car c'est lui qui dirige toutes les actions effectuées dans l'application. Il contient les méthodes Chargement() et Sauvegarde() qui vont appeler les méthodes de chargements et de sauvegardes de l'interface "IDataManager". Cela va permettre de charger la liste de films et la liste des réalisateurs présentes dans l'application à l'ouverture de celle-ci. Pareil pour la sauvegarde mais dans le sens inverse, elle va stocker les données présentes dans les listes afin que les données puissent persister.

Ensuite les méthodes "AjouterFilm" et "AjouterReal" sont similaires, la seule différence entre les deux, c'est qu'elles vont ajouter une instance différente dans une liste différente. La méthode "AjouterFilm" va donc ajouter un film à la collection de films en vérifiant qu'il n'existe pas déjà. Pareil pour un réalisateur qui sera ajouté dans la liste des réalisateurs en vérifiant toujours s'il n'existe pas déjà.

Après les méthodes de signalements. Elles permettent d'envoyer un mail aux administrateurs de l'application afin qu'ils puissent modifier ou supprimer l'objet signalé. Les méthodes sont les mêmes, seul le paramètre d'entrée change en fonction que ce soit un film ou un réalisateur.

Ensuite, nous avons les méthodes de recherches pour trouver un réalisateur ou un film. Chaque méthode délègue ce travail à l'interface IRecherchePersonne pour le réalisateur et IRechercheFilm pour le film.

Pour finir la suppression d'un film ou d'un réalisateur. Ce sont des simples méthodes qui vont supprimer les objets passés en paramètres des listes de réalisateurs ou de films.

Passons ensuite à la classe "Film", c'est la structure qui va permettre de représenter un film avec différents attributs. Il contient tout d'abord un titre de type string. Il a aussi une date de sortie de type int qui ne représentera que l'année

de sortie du film. Ensuite, une ObservableCollection de Tag qui sera la liste des différentes étiquettes du film. On utilise une ObservableCollection pour éviter d'implémenter l'interface INotifyPropertyChanged, le type ObservableCollection l'implémente de base, ce qui permet d'actualiser l'attribut sans avoir à appeler un événement de notification. "Tag" est une classe énumération qui va permettre de stocker toutes les étiquettes différentes que l'on peut associer à un film (par exemple film d'aventure, d'action, comédie, etc). De plus on a le Synopsis qui est stocké dans une variable de type string qui représente un résumé du film. On a aussi le PathFile qui représente le chemin d'accès à l'image du film. Pour finir, nous avons un dictionnaire de "Job" et d'une liste de "Personne". Elle va servir dans notre cas à stocker les différents acteurs et réalisateurs d'un film. On aurait pu faire deux simples listes pour les stocker mais on a pensé une façon évolutive d'enregistrer les personnes qui participent à la réalisation d'un film. En clé pour le dictionnaire, on a une énumération qui représente le rôle de la personne dans le film. Pour l'instant nous n'avons que réalisateur et acteur mais on pourrait en rajouter plus par la suite. Associé à chaque clé représentant un rôle dans le film, on a une liste de "Personne" qui sont donc assignés à un travail. On peut donc avec différentes listes de personnes avec des métiers différents par rapport au film réalisé.

La classe Personne représente donc une personne comme son nom l'indique. Elle a donc un prénom et un nom qui sont de type string. Elle a aussi une date de naissance de type DateTime et une date de mort de type DateTime?. On a fait en sorte que la date de mort soit "nullable" afin de pouvoir créer des personnes qui n'ont pas de date de mort. Ensuite nous avons une partie nationalité et biographie pour chacune d'elles. Et pour finir le chemin d'accès de type string stocké dans la variable "PathFile" afin de leur mettre une photo de profile.

Présentons rapidement la classe Admin. Cette classe représente un administrateur de l'application avec un nom d'utilisateur et un mot de passe. Dans un futur développement, on pourrait y associer d'autres propriétés comme une vérification d'ip ou d'autres sécurités. Tous ces administrateurs sont stockés dans une liste d'administrateurs présents dans le Manager.

Passons maintenant aux recherches de Films. Il y a donc une interface "IRechercheFilm" qui contient une méthode pour rechercher un film. Une autre classe implémente cette interface, c'est "RechercheFilmParNom". De cette façon, on pourra créer différentes manières de rechercher un film en ajoutant simplement une classe qui implémente l'interface "IRechercheFilm". De la même manière on a "IRecherchePersonne" qui fonctionne pareil mais pour rechercher des personnes.

Pour finir, nous allons parler de la persistance. On a donc une interface "IDataManager" qui contient des méthodes de chargements et de sauvegardes pour

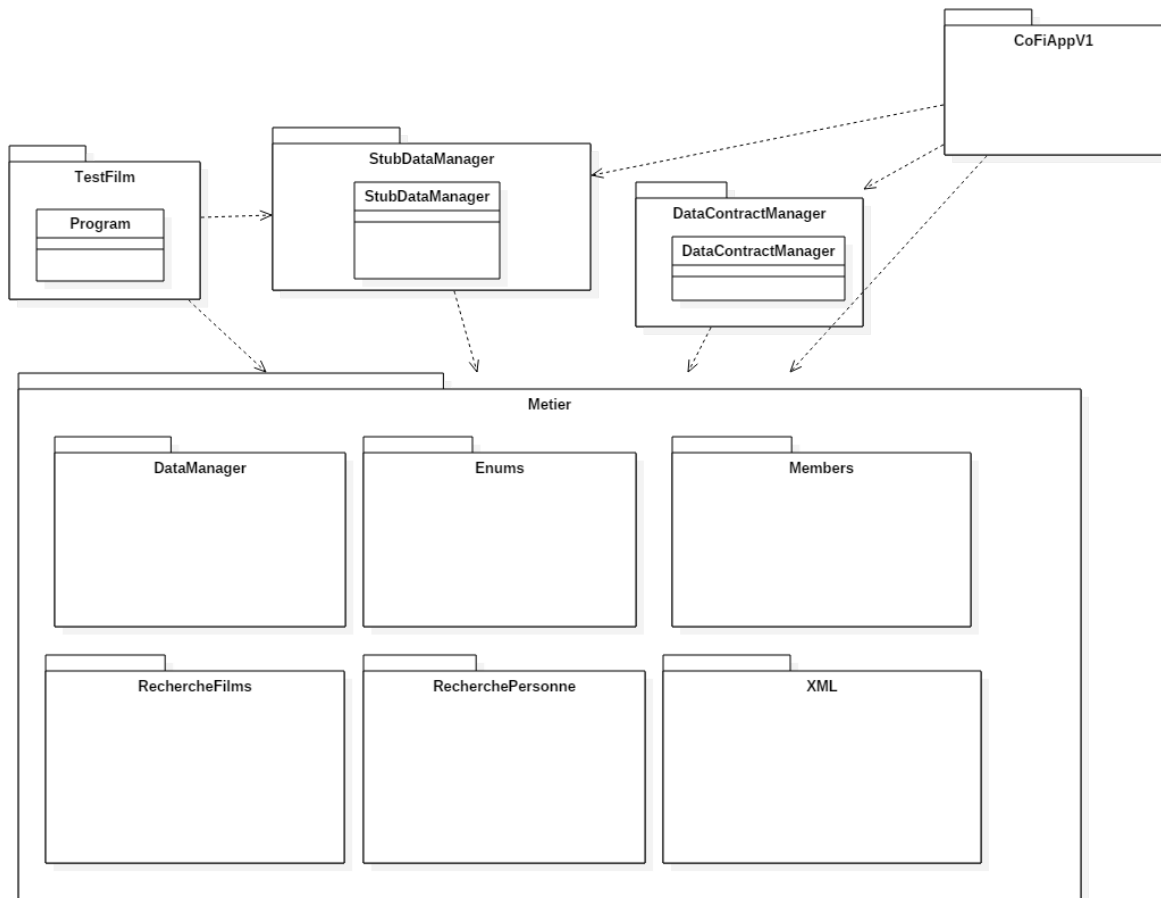
les films, les réalisateurs et les administrateurs. Il faut donc créer des classes qui implémentent celle-ci afin d'avoir de la persistance dans l'application. Le problème est qu'il y a une multitude de façon de persister des données que ce soit dans des fichiers XML, JSON, dans des bases de données ou autres. C'est pour ça que cette méthode est très pratique. On a une injection de dépendance dans le constructeur du Manager lui dit qu'il doit prendre en paramètre une classe qui implémente l'interface "IDataManager" mais il ne sait pas laquelle c'est. Comme ça on peut créer autant de façon de persister les données tant que l'on veut tant qu'on implémente l'interface "IDataManager". Il faudra bien évidemment changer aussi la classe que l'on veut utiliser lorsque l'on construit le Manager. Dans notre exemple, on a créé deux autres classes.

La première est le "StubDataManager" qui va juste utiliser les méthodes de chargement de données. Dans un premier temps, cela nous a permis de tester notre application avec des données en brutes en attendant d'avoir une vraie classe pour faire de la persistance depuis un fichier.

Deuxièmement, nous avons le "DataContractManager" qui lui implémente toutes les méthodes de chargements et de sauvegardes. On utilise un data contract pour pouvoir stocker et charger nos données depuis un fichier XML. On a donc deux fichiers qui sont "List_Films" et "List_Reals" qui stockent la liste de films et la liste de réalisateurs. Les administrateurs sont chargés en dur.

Nous avons un paquet TestFilm qui nous a permis de réaliser nos premiers tests dans des applications consoles afin d'avoir une idée de notre travail.

Diagramme de paquetage



La bibliothèque de classe Métier, comporte plusieurs paquetages :

-DataManager :

Contient l'interface IDataManager, qui comporte des méthodes de sauvegarde et de chargement.

-Enums :

Contient tous les types Enum de l'application.

-Members :

Contient les Membres objets de l'application : Film, Admin et Personne.

-RechercheFilms :

Contient une Interface de Recherche, et une Classe de recherche de film par nom.

-RecherchePersonne :

Contient une Interface de Recherche de personnes, et une classe de recherche de personne par film.

-XML :

Le paquetage XML contient les fichiers XML de sauvegarde de l'application, ce sont les fichiers de persistance de l'application.

-CoFiAppV1 :

C'est la partie XAML donc qui contient toutes les vues de l'application. Il y a aussi le code behind qui permet de relier le Metier avec les vues.

-TestFilm :

Dans ce paquet, il y a tous les tests d'application console que nous avons effectué avant de voir si notre Metier était fonctionnel avant de l'utiliser dans les vues.

-StubDataManager :

Ce paquet contient une classe et cette classe est composée de méthodes de chargement et de sauvegarde car elle implémente l'interface IDataManager. Ce sont donc des données brutes qui nous ont permis de tester les fonctionnalités que nous avons créer dans le métier et de faire un premier pas pour pouvoir utiliser l'application avec des données.

-DataContractManager :

Comme le StubDataManager, elle implémente l'interface IDataManager. Elle contient donc les méthodes de chargement et de sauvegarde mais d'une façon différente. Ce ne sont plus des données brutes mais des données stockées dans des fichiers XML présents dans le paquet XML du Métier.

Pour finir avec ce projet, nous avons réaliser une vidéo qui permet de présenter toutes les fonctionnalités de notre application en moins de trois minutes. Voici le lien : [cliquez ici](#)