

**Projet :
1001 Cocktails
Documentation**

**BOISSARD Clément
ROCHA Valentin
SAUGUES Benjamin**

Partie IHM (XAML & WPF)

Contexte du projet :

Qui n'a jamais rêver, lors d'une soirée, avoir accès à une bibliothèque contenant toutes les recettes de cocktails du monde ?

Voilà le but de notre application, mettre à la portée de n'importe qui une bibliothèque de cocktails divers et variés avec en plus des avis postés par des utilisateurs ayant déjà testé le cocktail désiré. Notre application permet aussi aux inventeurs en herbe de poster leur(s) cocktail(s) fraîchement inventé(s) afin d'obtenir des avis de la part d'autres personnes.

Explication du projet :

L'application 1001Cocktails aura pour objectif de permettre à un utilisateur de consulter une liste exhaustive de cocktails dans une base de données modifiable.

L'application permettra aux personnes inscrites de modifier, de supprimer et d'ajouter de nouveaux cocktails dans la base de données préexistantes.

Il sera possible pour n'importe quel utilisateur de laisser une note ainsi qu'un avis sur chacun des cocktails.

Il est tout aussi possible d'utiliser l'application en tant que simple visiteur en n'ayant accès qu'au fonctionnalité de bases (consultation, commentaire). L'utilisateur aura la possibilité de trier la liste de cocktails selon différents critères (nom, couleur, ...).

L'utilisateur aura la possibilité de se créer un compte sur l'application afin d'avoir accès au fonctionnalité avancé (suppression, modification, ajout).

Lorsque l'utilisateur sélectionne un cocktail dans la liste, les détails de ce dernier sont affichés tel que le nom, les ingrédients, la recette ainsi qu'une image et une liste d'avis laissé par des utilisateurs ayant testés ce cocktail

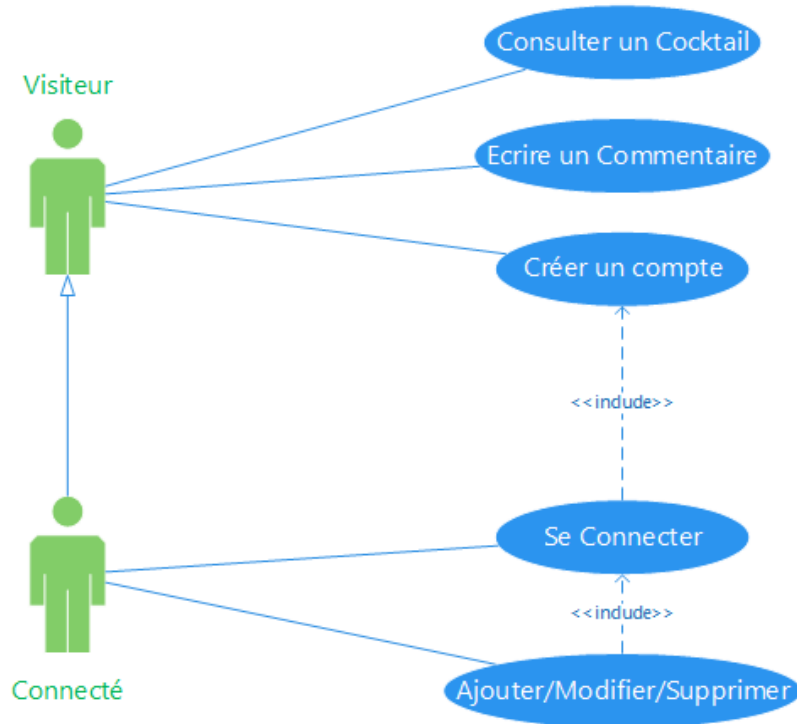
Considérations ergonomiques :

L'interface de l'application a été développé afin de la rendre la plus compréhensible dès la première utilisation. Le design épuré permet de ne pas surcharger l'application au niveau des couleurs et ainsi permettre à l'utilisateur de comprendre instinctivement la fonction de l'application. Les fonctionnalités des boutons sont mises en évidence grâce au fort contraste de leur code couleurs.

Prise en compte de l'accessibilité :

Le code couleur choisi pour notre application n'est pas gênante pour un daltonien car elle ne peuvent pas être confondus par les daltoniens.

Diagramme de cas d'utilisation :



Dès la première utilisation de l'application, l'utilisateur peut utiliser l'application en tant que « Visiteur », il n'aura alors accès qu'aux fonctionnalités de bases de l'application qui sont : « Consulter un cocktail » et « Écrire un commentaire ».

Si l'utilisateur veut avoir accès aux fonctionnalités avancées de l'application il devra se créer un compte. Il aura alors accès à trois nouvelles fonctionnalités : « Ajouter un cocktail », « Modifier un cocktail » et « Supprimer un cocktail »

Sketch :

Fenêtre de connexion à l'application :

Première fenêtre de l'application où l'utilisateur peut si il possède déjà un compte sur l'application entrer ses identifiants et mot de passe (1) puis se connecter (2). Il peut, si il ne possède pas de compte accéder à la fenêtre de création de compte (4) ou tout simplement utiliser l'application en tant que simple visiteur.

The sketch shows a window titled "Connexion". It contains two input fields for username and password, both labeled with a blue "1". Below these fields are three buttons: "Connexion" (labeled with a blue "2"), "Visiter" (labeled with a blue "3"), and "Créer Compte" (labeled with a blue "4").

The sketch shows a window titled "Créer Compte". It contains four input fields for "Pseudo", "Mail", "Mdp", and "Conf.", all labeled with a blue "1". Below these fields are two buttons: "Créer" (labeled with a blue "2") and "Annuler".

Fenêtre de création de compte :

Cette fenêtre permet à un utilisateur ne possédant pas de compte de s'en créer un.

Il doit donc renseigner certaines informations (1).

Après avoir renseigné toutes les informations nécessaires il pourra alors valider la création du compte (2).

Cette action le renverra à la page de connexion où il pourra alors se connecter.

Fenêtre principale de l'application :

Il s'agit de la fenêtre permettant de consulter les différentes recettes de cocktails proposé par l'application. Cette fenêtre comporte quelques différences en fonction du fait que l'utilisateur de l'application possède un compte ou non. Un utilisateur ayant un compte aura accès à toutes les fonctionnalités de l'application, la fenêtre s'affichant à lui lui permettra de toutes les utiliser (1). A l'inverse un utilisateur n'ayant pas de compte n'aura accès qu'à la fonctionnalité de base (2).

Fenêtre de l'utilisateur avec compte (1) :

The sketch shows a window titled "Main Vide". It has a sidebar on the left with a list of cocktail names: "Vodka Orange", "Pina Colada", "Pti Punch", and "B-52". The main area contains a placeholder image (a square with an 'X'), a "Nom" input field, an "Ingrédients" input field, a "Recette" input field, and an "Avis" input field. At the bottom are four buttons: "Ajouter", "Modifier", "Supprimer", and "Commentaire".

Fenêtre de l'utilisateur sans compte (2) :

The sketch shows a window titled "Main Visiteur". It has the same sidebar and main area as the previous window. However, the "Ajouter", "Modifier", and "Supprimer" buttons are replaced by a "Connexion" button, and the "Commentaire" button is replaced by a "Commentaire" button.

Description de la fenêtre principale :

The screenshot shows a window titled "Main Connecte". On the left is a list of cocktails: "Vodka Orange", "Pina Colada", "Pti Punch", and "B-52". The right side displays details for "Vodka Orange", including a placeholder image, the name, ingredients ("5 cl de Vodka", "10 cl de jus d'Orange"), a recipe ("Mélanger la Vodka et le jus d'Orange. Servez bien frais."), and a section for reviews. At the bottom are buttons for "Ajouter", "Modifier", "Supprimer", and "Commentaire".

1. Liste de cocktails :

Liste déroulante de cocktails sur le coté gauche de la fenêtre, chaque item de la liste est composé de l'image du cocktail et de son nom.

2. Le détail du cocktail :

Fiche descriptive du cocktail sélectionné dans la liste (1).

3. L'entête de la description du cocktail :

La partie du haut de la description composé de l'image du cocktail, du nom et de sa liste d'ingrédients.

4. La recette du cocktail :

La marche à suivre afin de réaliser le cocktail sélectionné.

5. La liste des avis :

La liste des avis déposés par les différents utilisateurs ayant testé ce cocktail et ayant laissé un commentaire.

6. Barre de gestion :

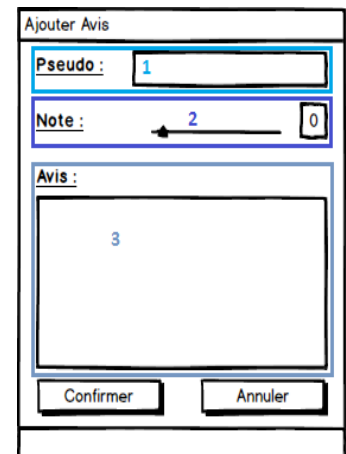
Boutons pour les différentes actions (modifier/ajouter/supprimer/se connecter/commentaire) réalisables suivant le type d'utilisateur.

Fenêtre d'ajout d'avis :

Cette fenêtre permet à un utilisateur de laisser un avis sur un cocktail choisi.

Il devra donc pour cela renseigner son pseudo (1) (si l'utilisateur possède un compte, le pseudo du compte sera automatiquement renseigné).

Il devra aussi choisir une note (2) et écrire le commentaire souhaiter (3).



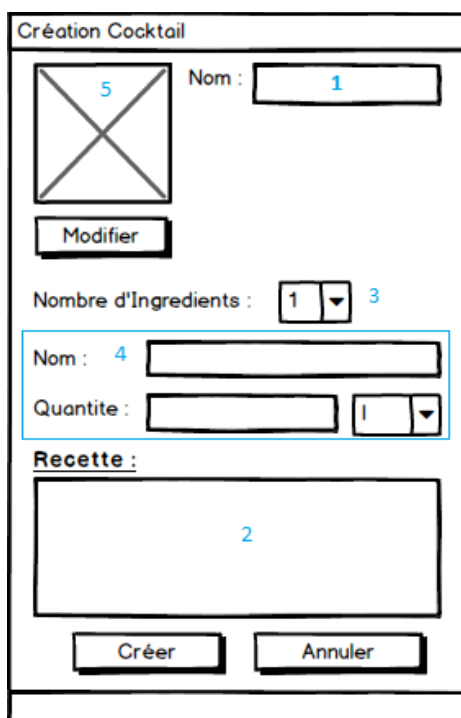
Fenêtre de création/modification d'un cocktail :

Cette fenêtre permet à un utilisateur de créer ou de modifier un cocktail préexistant. L'affichage de la fenêtre sera différents en fonction de l'utilisation quand fait l'utilisateur.

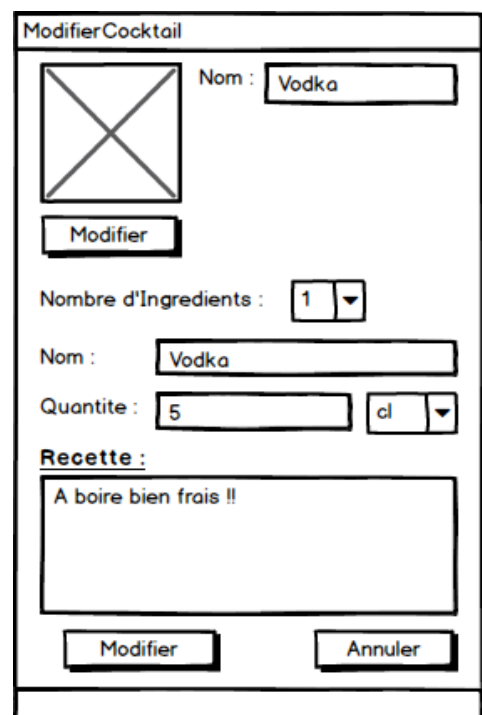
Pour la création du cocktail : l'utilisateur doit renseigner le nom du cocktail (1) , la recette du cocktail (2) et le nombre d'ingrédients nécessaires (3) ainsi qu'une image (5). La partie où l'utilisateur doit renseigner le nom des ingrédients et leurs quantités (4) va se dupliquer en fonction du nombre d'ingrédients sélectionnés par l'utilisateur.

Pour la modification d'un cocktail : les informations préexistantes sur le cocktails sélectionnés seront déjà indiqués sur la fenêtre et l'utilisateur pourra donc soit en modifier certaines soit en supprimer.

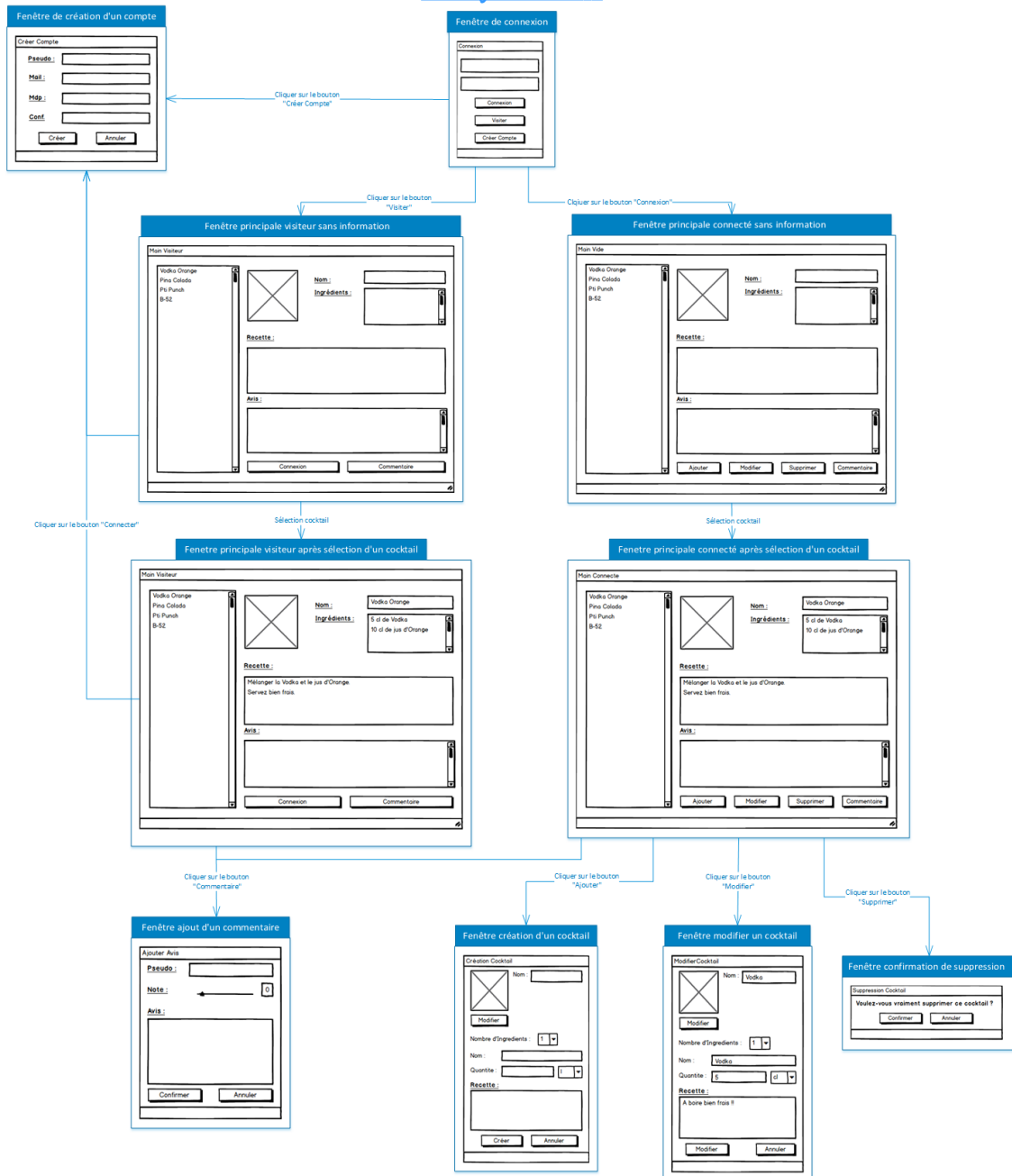
Fenêtre de création d'un cocktail :



Fenêtre de modification d'un cocktail :



Story-board :



Lors de l'arrivée sur l'application trois choix sont disponibles : l'utilisateur peut se connecter en renseignant son login et son mot de passe, il peut continuer sur l'application en tant que simple visiteur ou il peut créer un compte.

Si l'utilisateur choisit de se créer un compte il sera redirigé vers la fenêtre de création de compte où dès qu'il aura renseigné les informations nécessaires et validé sera redirigé vers la fenêtre de connexion afin de se connecter avec ces nouveaux login et mot de passe.

La fenêtre principale comprend quelques différences en fonction si l'utilisateur possède un compte ou non :

- Si l'utilisateur « visite » simplement l'application il pourra consulter et laisser un commentaire sur les cocktails, il pourra aussi choisir de retourner sur la fenêtre de création de compte afin d'avoir accès à des fonctionnalités supplémentaires.
- Si l'utilisateur se connecte à l'application il aura alors accès à toutes les fonctionnalités de l'application, il pourra supprimer/ajouter/modifier un cocktail ainsi qu'ajouter un avis.

Si l'utilisateur quitte l'application en se trouvant sur la fenêtre principale une sauvegarde automatique se fera et sauvegardera les modifications réalisées par l'utilisateur.

Partie COO

Description de l'architecture :

Problème : Toutes nos classes ne doivent et ne peuvent être modifiées par n'importe qui.

Solution : Utilisation du patron de conception « Façade ».

Explication : Création d'une classe « Manager » contenant nos listes et collections et par laquelle il faut passer si l'on veut modifier le contenu de ces listes et collections.

Problème : Mise en place de la persistance.

Solution : Linq To XML.

Explication : Création d'une classe « XMLDataManager » utilisant le Linq to XML. Cette classe est protégée par une interface immuable « IDDataManager ».

Problème : Accès aux constructeurs de « Cocktail » depuis « XMLDataManager » se situant dans un autre package.

Solution : Utilisation du patron de conception « Fabrique Simple ».

Explication : La classe « Cocktail » et la classe « XMLDataManager » se trouvant dans deux packages différents et la classe « Cocktail » étant interne à son package. Nous avons créé une nouvelle classe : la « Fabrique » se situant dans le package de la classe « Cocktail » et implémentant des fonctions faisant appel au constructeur de « Cocktail ». « Fabrique » étant public elle est donc accessible depuis la classe « XMLDataManager » qui peut donc instancier de nouveaux cocktails en appelant les fonctions de la classe « Fabrique ».

Problème : Ne permettre qu'un seul commentaire par utilisateur pour un cocktail précis.

Solution : Utilisation d'un Dictionary et de l'interface IEquatable.

Explication : L'utilisation d'une ObservableDictionary permet d'empêcher un utilisateur de mettre plusieurs commentaires un même cocktail. Si l'utilisateur essaie de commenter une nouvelle fois un cocktail qu'il avait déjà commenté alors le précédent cocktail sera modifié et prendra les valeurs du nouveau.

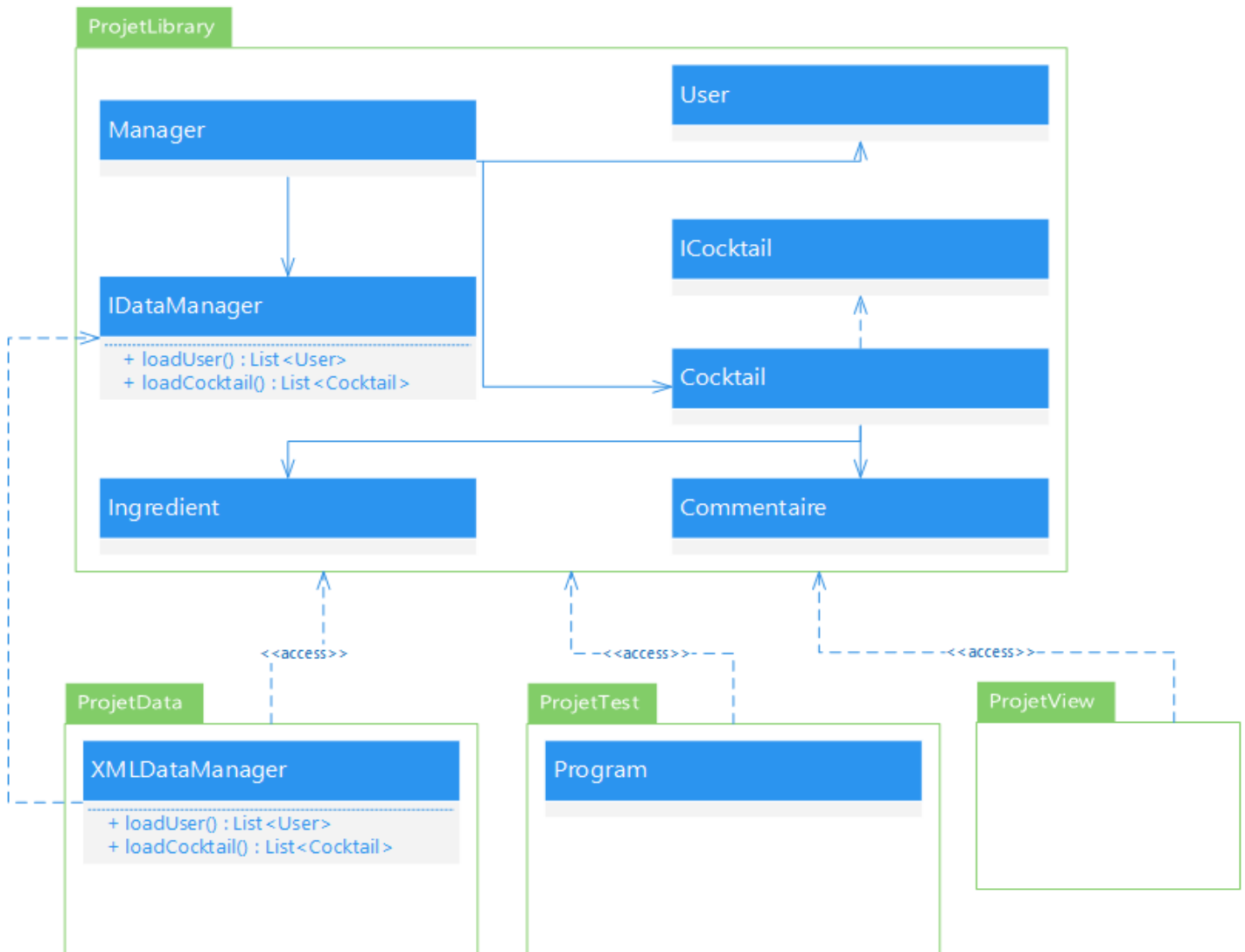
Diagramme de classe :



Notre projet repose sur la classe « Manager » situé au centre du diagramme et qui a accès à la quasi-totalité des classes de la library. Cette classe contient une liste de « User » ainsi qu'une liste de « Cocktail » contenant elle même une liste d'« Ingrédients ». Notre classe manager à aussi accès à l'interface « IDataManager » afin de pouvoir réaliser la persistance du projet. La classe « Cocktail » est protégé par une façade « ICocktail ».

« Cocktail » étant protégé, le « XMLDataManager » utilise la « Fabrique » afin de créer des nouveaux cocktails lorsque cela est nécessaires car « XMLDataManager » n'est pas dans le même package que le « Manager ».

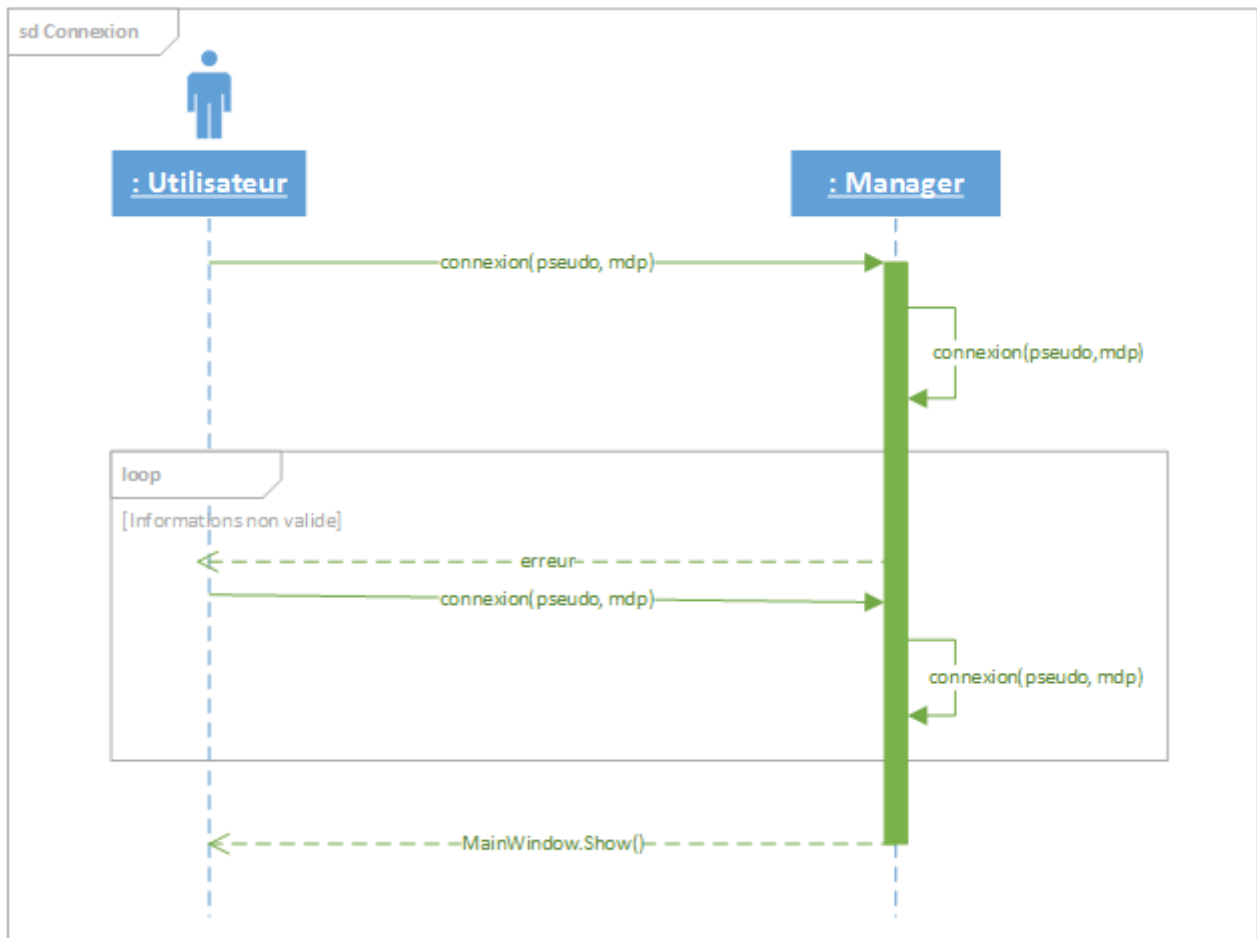
Diagramme de package:



Notre projet contient en tous 4 packages de classes distinct :

- Le package **Projet** qui contient les vues du projet .
- Le package **ProjetTest** qui contient tout nos programme pour testé les classe implémente.
- Le package **ProjetData** qui contient les classes permettant de faire de la persistance de données (dans notre cas du Linq to XML).
- Le package **ProjetLibrary** qui contient la quasi-totalité de la partie métier du projet.

Diagramme de séquence :



Lors de la connexion à l'application, l'utilisateur renseigne son pseudo et son mot de passe au manager afin qu'il puisse vérifier si le compte de l'utilisateur existe réellement dans sa base de données : si le compte est bien présent, le manager autorise à continuer sur l'application en lui affichant la fenêtre principale de l'application, mais si les informations transmises par l'utilisateur ne correspondent à aucun compte présent dans la base de données alors le manager va retourner un message d'erreur et demander à l'utilisateur de ressaisir son pseudo et son mot de passe, il fera cela tant que le pseudo ou le mot de passe seront erronés.