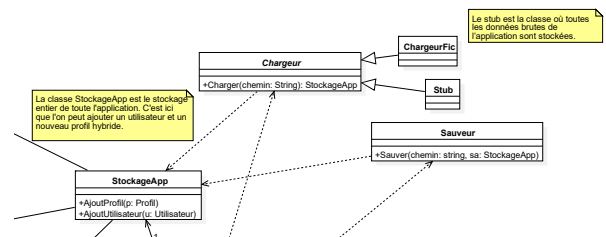


Documentation Projet tuteuré :

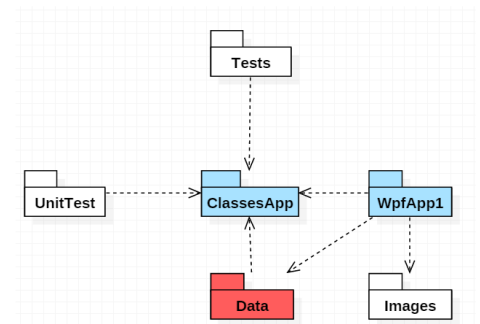
1. Persistance

La présence de la classe `StockageApp` est justifiée par la persistance. En effet, puisque cette classe contient toutes les données, cela permet de n'avoir besoin de sérialiser qu'une instance de cette classe pour avoir toutes les données. C'est le travail de la classe `Sauveur`, qui sérialise toutes les données dans un fichier binaire.



L'interface `Chargeur` est implémentée par deux classes : `ChargeurFic` et `Stub`. Dans le `Stub` se trouvent toutes les données « en dur » de l'application (les profils de base et les boîtes de jeu). C'est lui qui est appelé lors du premier lancement de l'application. Ensuite, si l'utilisateur a sauvegardé de nouvelles données, alors c'est la classe `ChargeurFic` qui est appelée, et qui va chercher dans le fichier binaire toutes les données.

Le paquet `Data` est celui dans lequel se trouvent les classes liées uniquement à la gestion des données. On y trouve donc le `Stub`, le `ChargeurFic`, le `Chargeur` et le `Sauveur`. Ce paquet est référencé par le paquet `WpfApp1`, dans lequel se trouvent toutes les pages de l'application. C'est grâce à cela que l'application peut obtenir ses données, ainsi que les sauvegarder pour les utilisations suivantes. Enfin, la référence vers `ClassesApp` est présente pour permettre aux classes de `Data` d'accéder au `StockageApp` afin de pouvoir gérer facilement toutes les données de l'application.



2. Ajouts personnels

Plusieurs options de cette application sont des ajouts personnels.

Le premier qui est visible sur la **MainWindow** est la possibilité de se faire rediriger sur une page qui vend une boîte du jeu Chroniques Oubliées. Dans l'événement d'un des trois boutons (ex : **LienBoite1_Click**), on instancie un processeur d'un navigateur puis on lui donne le lien du site. Donc lorsque l'utilisateur cliquera sur une des images des boîtes de jeu il sera directement envoyé sur le site de vente.

Lorsqu'un utilisateur se connecte, il peut créer un profil hybride. Ce profil hybride, il le constitue à partir des 14 profils de base du jeu. Pour créer un profil hybride, il doit appuyer sur le bouton **Créer hybride** qui se rapporte à l'événement **LBprofils_MouseDoubleClick**. Cet événement appelle la fenêtre **CréationProfil.xaml**. Ce profil est ensuite stocké dans sa liste **ProfilsHybrides**. Cette dernière est ensuite sérialisée lorsque l'utilisateur appuie sur le bouton **Sauvegarder**. Il peut aussi mettre des profils de base ou ses profils hybride en profils favoris. Ces profils favoris sont stockés dans sa liste **ProfilsFavoris** qui elle aussi est sérialiser lors de l'activation du bouton **Sauvegarder**.

Utilisateur
+MotDePasse: string
+AjoutProfilHybride(p: Profil) +AjoutProfilFavori(p: Profil) +SupprimerProfilFavori(p: Profil) +SupprimerProfilHybride(p: Profil)