

## Projet Tuteuré - Description des parties ajoutées

Notre partie ajoutée porte sur un égaliseur de son qui utilise la dll Naudio.

Celle-ci permet de récupérer la valeur du pic de son en sortie du périphérique de son par défaut, que nous utilisons pour fixer la valeur de multiples ProgressBar verticale placées dans un Dictionary<int, ProgressBar> : la Key étant le numéro identifiant la Progressbar dans le Dictionnaire.

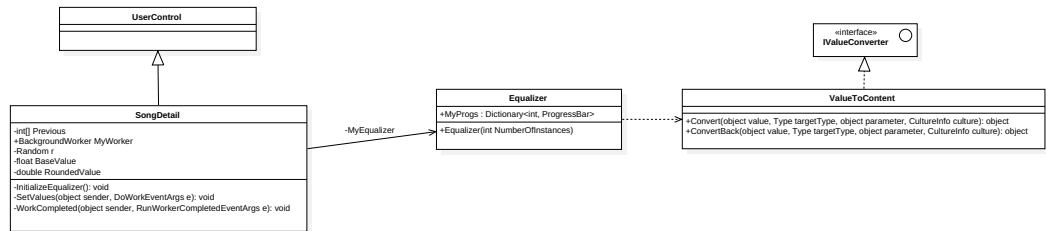
Comme l'actualisation est coûteuse en ressource, celle-ci est réalisée de manière asynchrone pour optimiser le fonctionnement global de l'application.

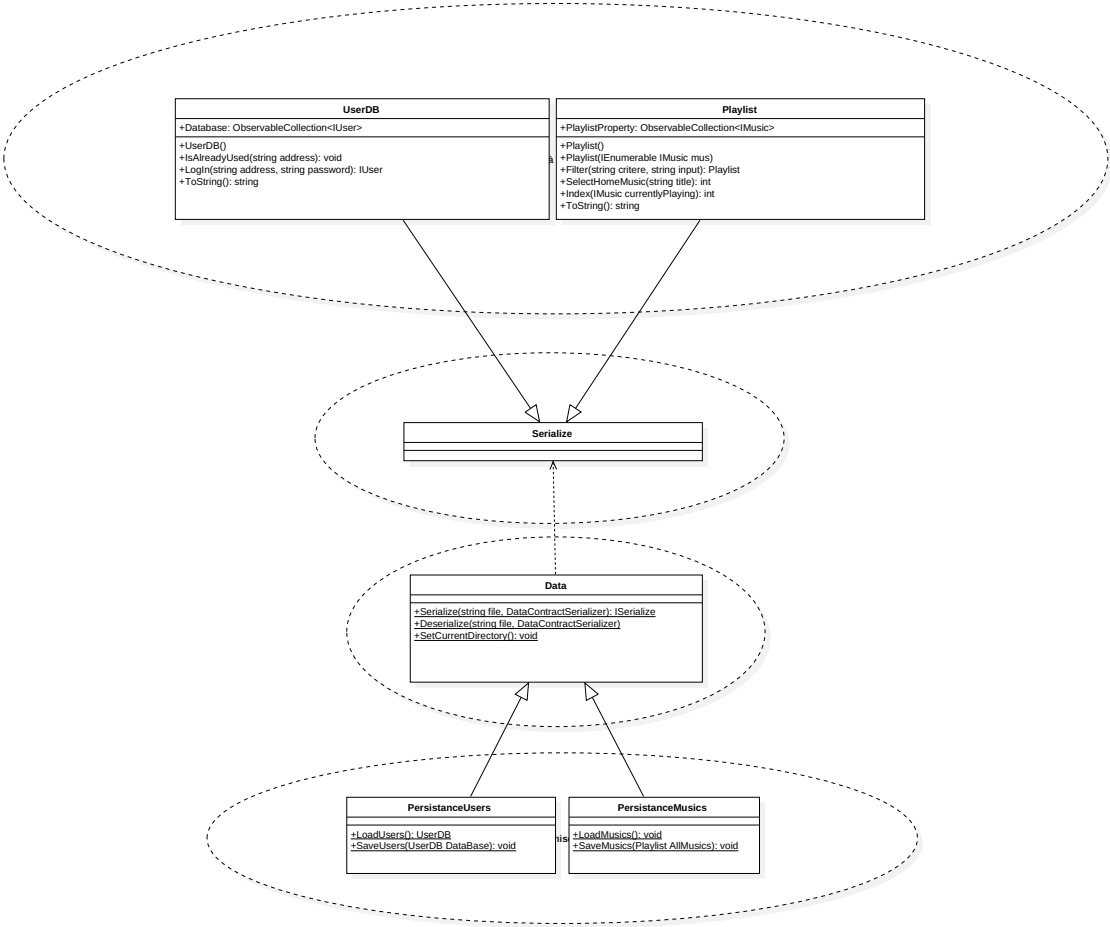
## Projet Tuteuré - Diagramme de classe mettant en avant la partie persistance

Au niveau de la persistance, la classe abstraite Serialize, dont Playlist et UserDB héritent, permet d'avoir un objet commun de (dé)sérialisation, ce qui simplifie les implémentations de stratégie de (dé)sérialisation. Data nous permet de n'avoir qu'une seule classe à modifier si nous décidons de changer de méthode de (dé)sérialisation.

## Projet Tuteuré - Diagramme de package mettant en avant la partie persistance

Au niveau de la persistance, les packages montrent l'emboîtement de notre partie persistance : notre utilitaire aide à l'uniformisation des classes à (dé)sérialiser, la stratégie contient la méthode de (dé)sérialisation et enfin l'implémentation contient les mises en oeuvre spécifiques.





Projet Tutoré::Diagramme de paquetage mettant en avant la partie persistance

