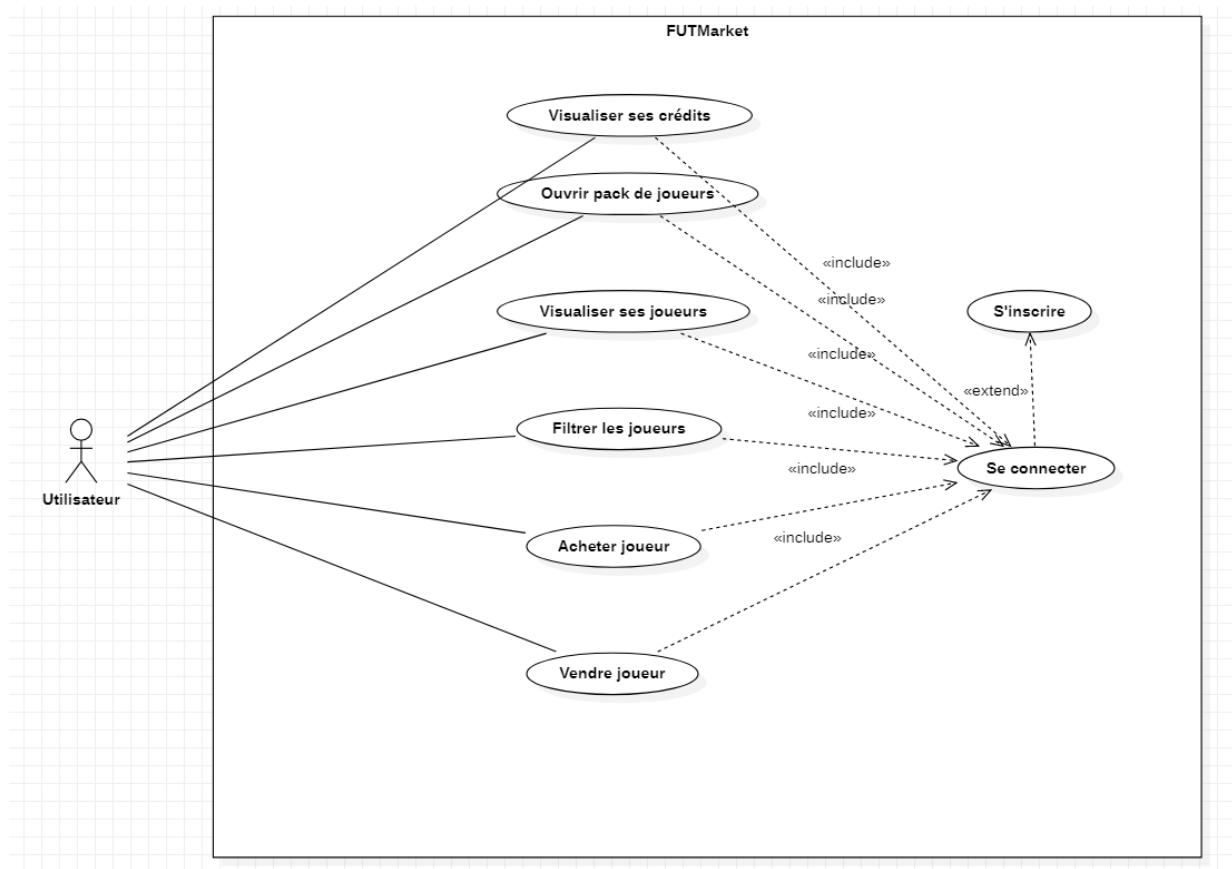


Projet Android

Contexte :

Dans le cadre de notre matière de conception d'application mobile nous avons eu à réaliser une application mobile en duo en 7 semaines. Nous avons choisi comme sujet, le football, et plus particulièrement les joueurs de football. Nous avons voulu créer une application du style « Ligue fantasy » ou du mode Fifa Ultimate Team de Fifa qui est un jeu où les participants endossent le rôle de propriétaires d'équipes sportives. Le joueur a donc l'objectif de collectionner tous les footballeurs qui sont représentés par des cartes de jeu. Pour cela, la principale méthode pour acquérir des joueurs est d'ouvrir des packs ou paquet de cartes. Evidemment les paquets de cartes sont donnés en échange de crédits. Ces crédits peuvent être ensuite obtenus en revendant les joueurs possédés. De plus, si vous recherchez un joueur en particulier, il peut être retrouvé dans la boutique. Il vous sera donné en échange de crédits. Il vous faudra donc ouvrir des paquets, pour ensuite vendre les joueurs obtenus pour acheter les joueurs de vos rêves et collectionner tous les joueurs en ouvrant encore une fois des paquets. Tout cela pour un objectif, être le plus grand collectionneur de FUT Market.

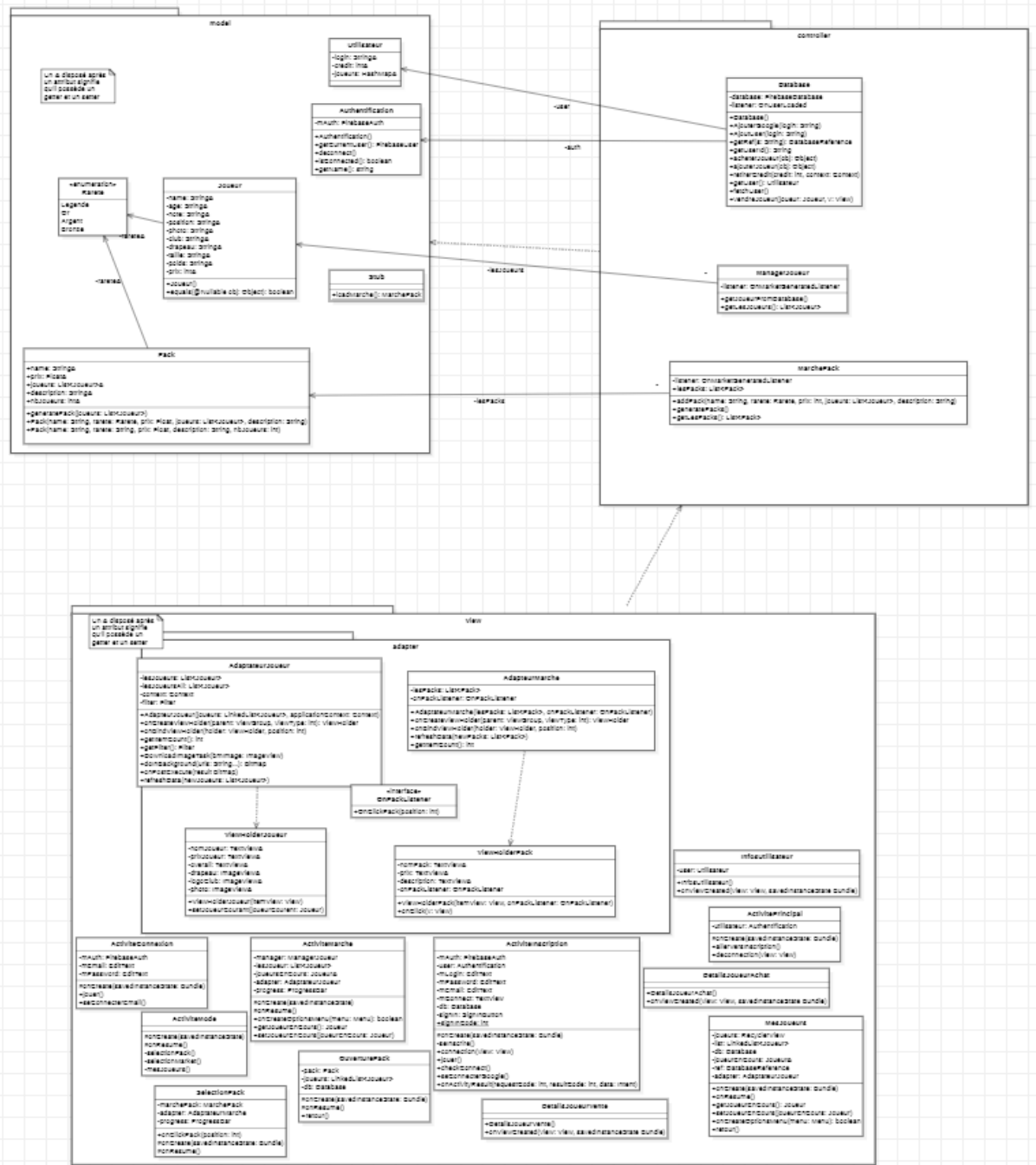
Voici le cas d'utilisation correspondant aux fonctionnalités de notre application :



Dans notre projet, nous avons mis en place un modèle MVC, ou modèle métier, vue, contrôleur. Celui-ci consiste en la séparation du code de la vue, du métier et de la place de classe « manager » qui sont représenté par les contrôleurs.

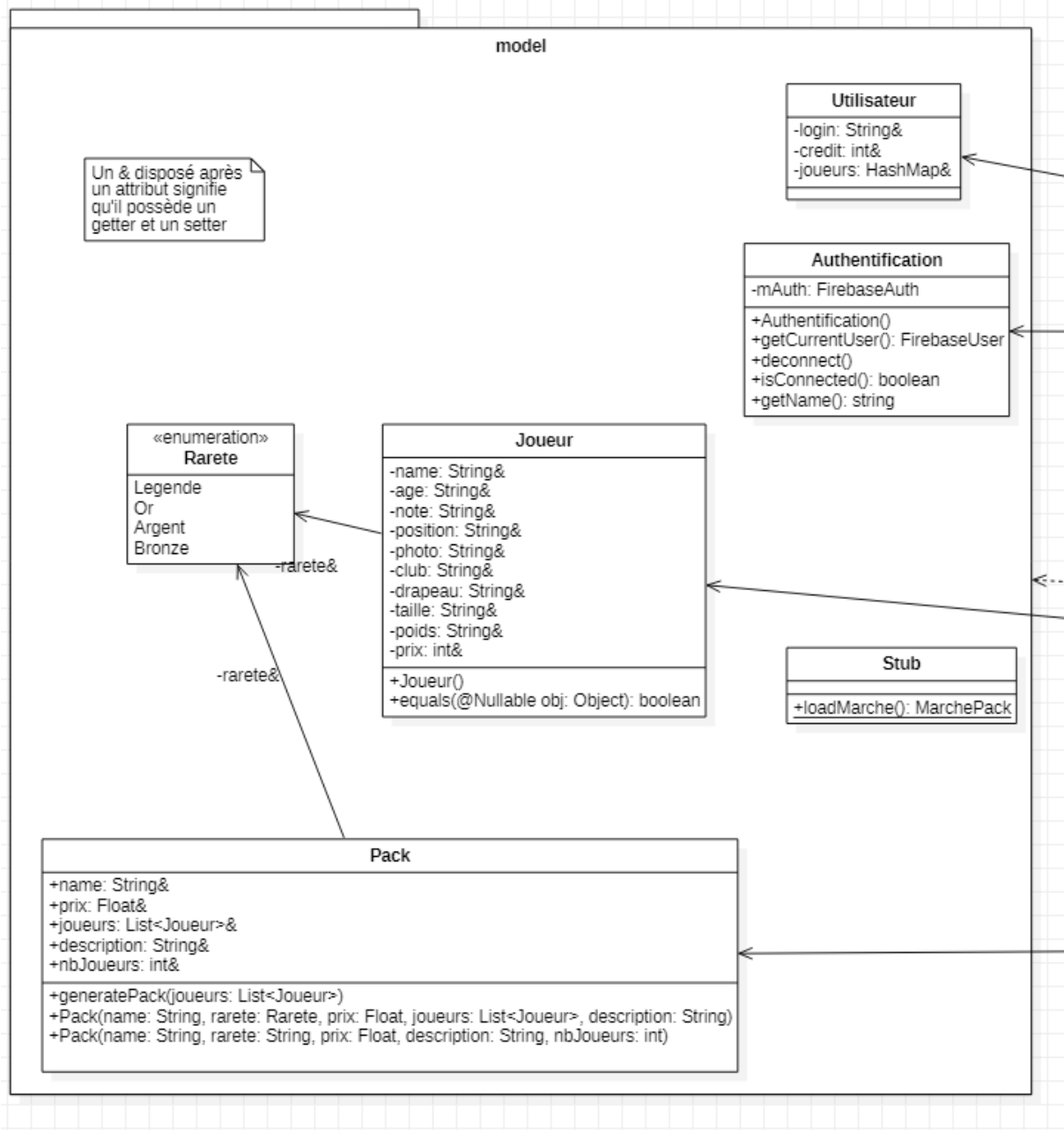
Cette architecture est largement observable avec une vue d'ensemble du diagramme de classe :

On observe en haut à gauche le dossier représentant le modèle, à droite celui représentant les contrôleurs et enfin en bas les code-behind de toutes les vues.

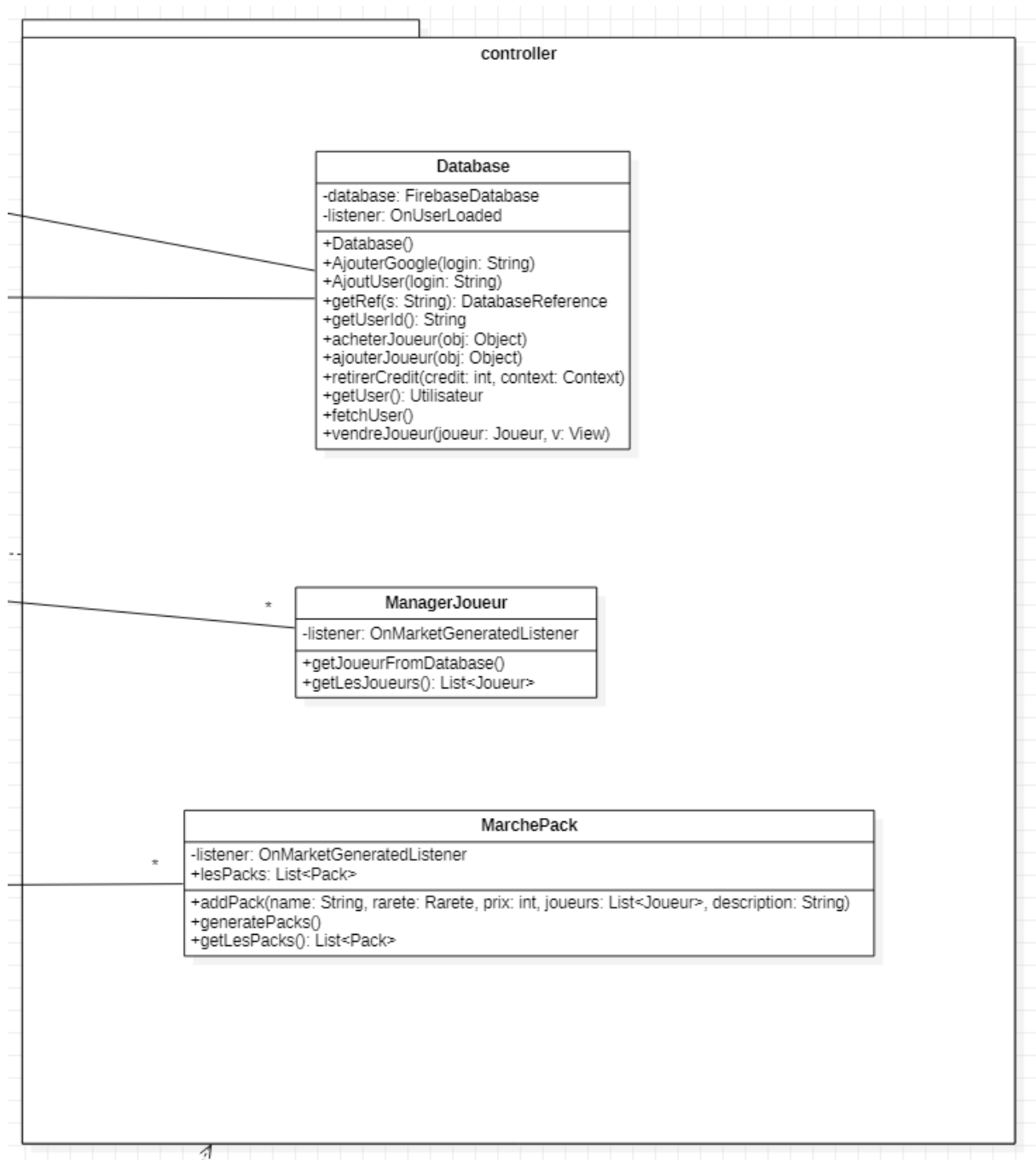


Nous allons donc décrire à un à un l'architecture de notre application :

Nous retrouvons ici le métier, celui-ci est responsable de déterminer les classes nécessaires à l'implémentation de l'application. On retrouve donc la définition du Pack, d'un Utilisateur et d'un Joueur. Ces classes sont alors réutilisées partout dans l'application. Nous avons aussi la présence d'un Stub simulant une persistance et nous permettant de tester notre application en générant des données fictives. Pour finir nous avons la classe d'Authentification qui permet à l'utilisateur de se connecter aussi bien avec son compte google qu'avec une connexion anonyme avec son propre mail.



On retrouve de l'autre côté les contrôleurs de notre application, c'est eux qui sont responsable de faire le pont entre le métier et la vue et ainsi de correctement découpler les deux. On retrouve tout particulièrement la classe responsable des interactions avec base de données Firebase. Par la suite, il y a la présente des manager pour les packs et les joueurs. Ces classes représente le « design pattern » Manager bien connu.



Pour finir nous retrouvons le code-behind de la vue avec la logique implémentée via les contrôleurs observé plus haut. Les vues utilisent tout particulièrement les adapter afin d’afficher et de remplir les RecyclerView.

