

Rugbyscope

➤ Contexte

Si Comme nous vous êtes mordu de rugby alors l'application Rugbyscope est faite pour vous. En effet elle va vous permettre de consulter les résultats, les matchs à venir ou encore la fiche des joueurs de chaque équipe.

Prenons un exemple, vous êtes entrain de regarder le match ASM-Racing mais vous ne connaissait pas tous les joueurs de l'équipe du Racing (en tant que supporter de l'ASM vous connaissait les joueurs sur le bout des doigts), pas de soucis un tour sur Rugbyscope et vous aller rapidement découvrir le nom et la description du joueur.

Ou alors vous ne vous rappelé pas du score du dernier match des français lors du Six nations ? Pas de problème Rugbyscope vous le donne en un clic.

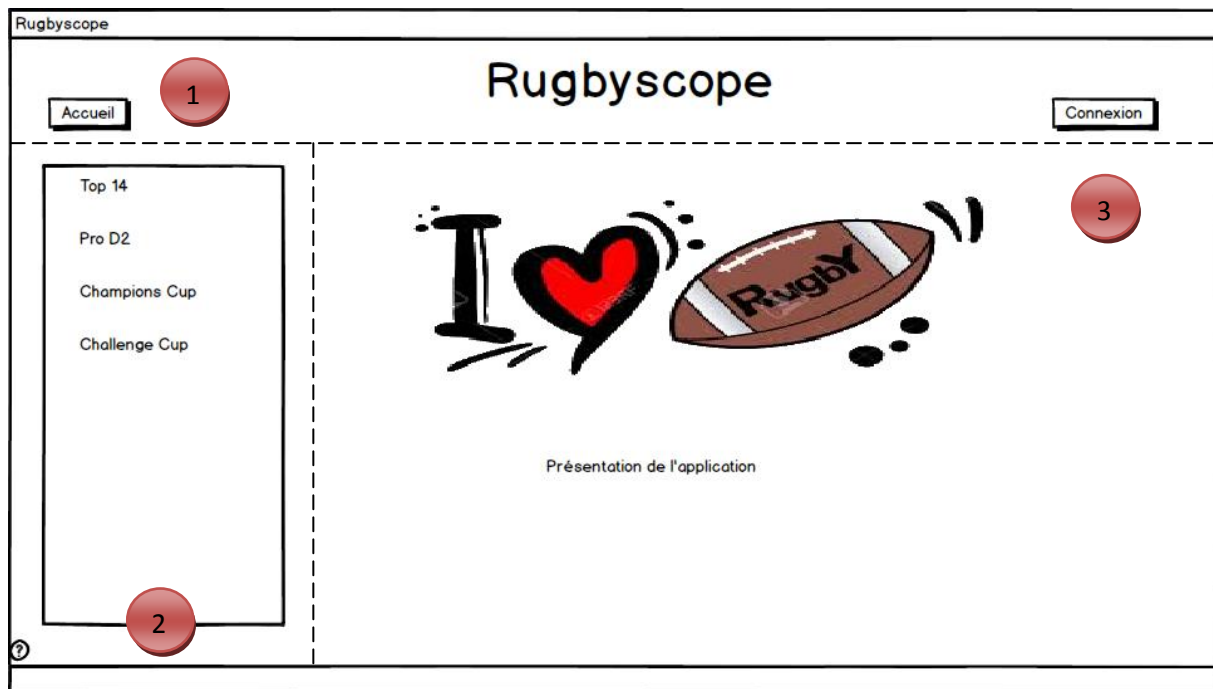
A quel type de personnes cette application est-elle destinée ?

Rugbyscope s'adresse essentiellement à un public amateur de rugby même si d'autres personnes qui souhaitent juste obtenir quelques renseignements sur un joueur, une équipe ou une compétition ont ici une application qui est tout à fait approprié. En effet via cette application l'utilisateur pourra suivre les différents résultats d'une compétition à laquelle il s'intéresse tout particulièrement, ou tout simplement les résultats de son équipe favorite. Mais l'utilisateur pourra aussi consulter les palmarès des équipes et l'historique des vainqueurs d'une compétition.

A quoi sert Rugbyscope ?

Comme il est indiqué ci-dessus, cette application permet de suivre les différents résultats que ce soit d'une équipe ou d'une compétition qui est bien entendu présente sur l'application. Cependant RugbyScope possède aussi une partie « information » qui permet aux utilisateurs de consulter le palmarès d'une équipe, l'historique d'une compétition ou encore les caractéristique d'un joueur comme par exemple son nombre de sélections, sa taille, son poids...

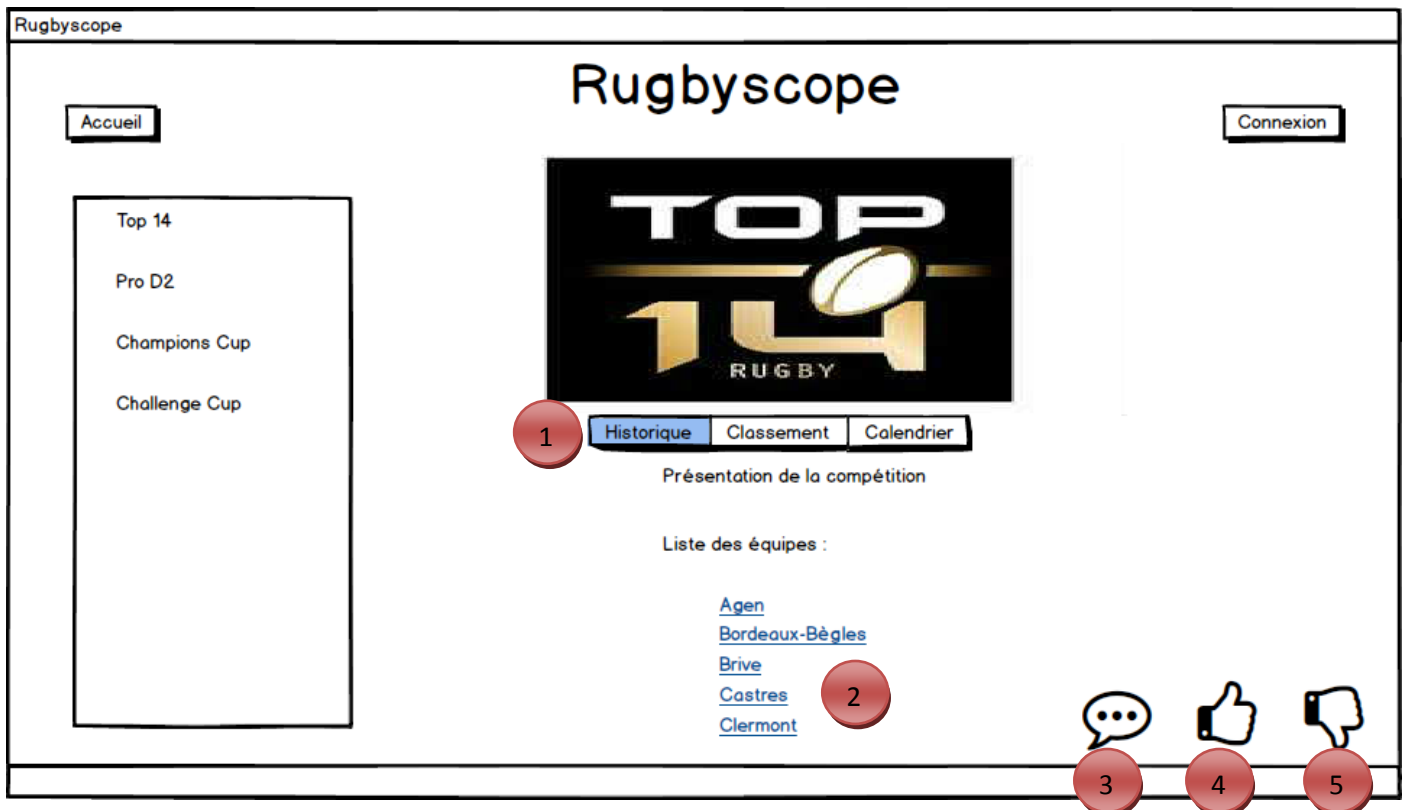
➤ Sketch



Comme on peut le voir, notre application est divisée en trois parties :

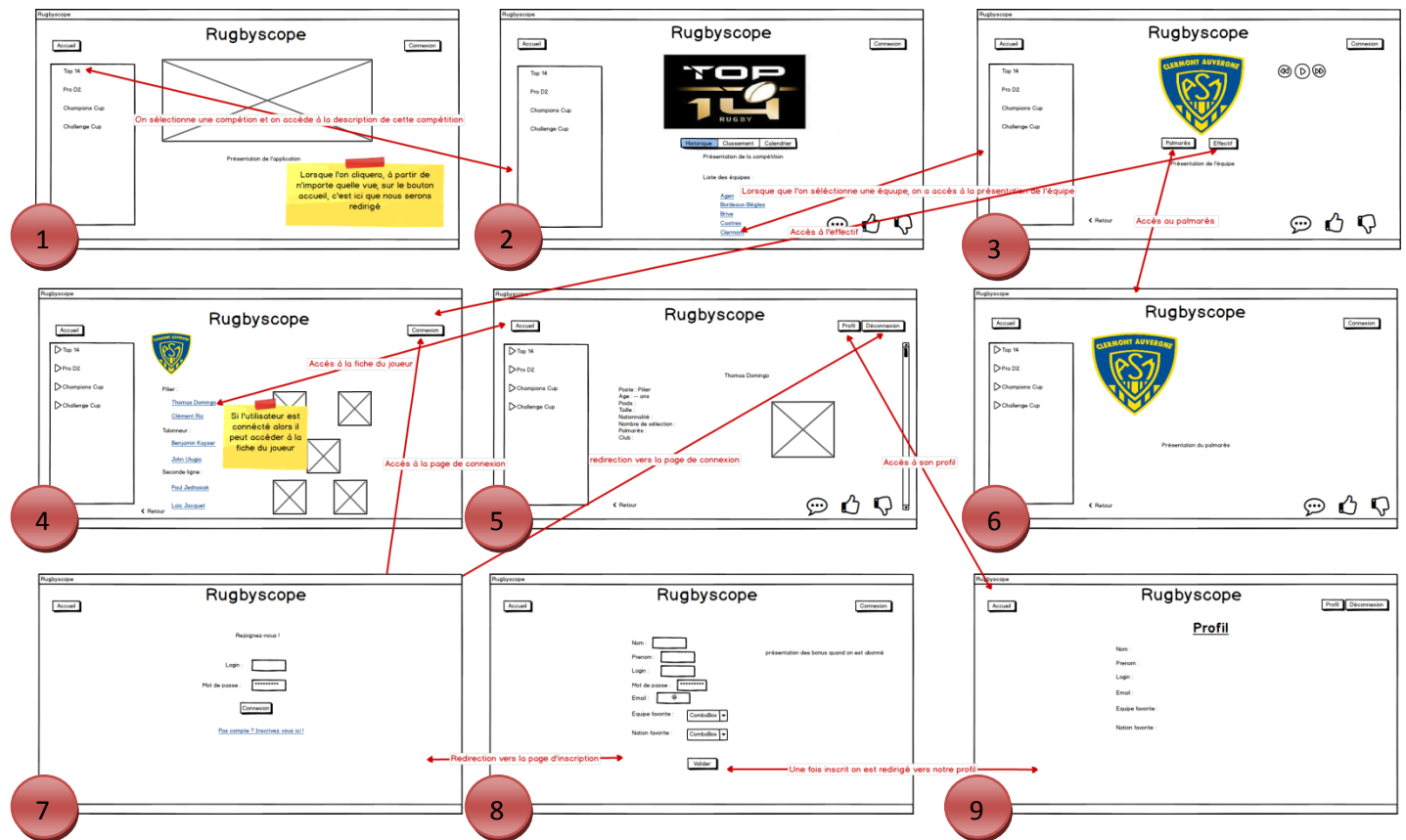
- 1- Cette partie correspond au bandeau de notre application. On y retrouve le titre « Rugbyscope » et un bouton de chaque côté du titre. Le bouton Accueil permet de revenir à la page d'accueil qui est celle qui est présentée ci-dessus. On le retrouve sur toutes les pages de notre application car à partir de n'importe quelle page on peut revenir sur l'accueil. Le bouton Connexion permet à l'utilisateur de se connecter ce qui va lui donner plus d'accessibilité à l'application. Lorsque l'on clique sur ce bouton on est redirigé vers une page de connexion où l'on peut renseigner notre login et notre mot de passe ou alors s'inscrire.
- 2- La deuxième partie se situe sur la gauche de la fenêtre et est composée d'une liste. Cette liste contient toutes les compétitions que notre application gère.
- 3- Cette partie qui prend la place restante permet d'afficher le contenu. Lorsque l'on va choisir une compétition dans la liste de la partie 2, le contenu de cette compétition s'affiche ici. De plus à partir de ce contenu on peut accéder à d'autres contenus qui seront eux aussi d'afficher dans la partie 3.

Ci-dessous, un sketch montrant ce que l'on voit quand on clique sur une compétition. La fenêtre est divisée de la même façon, seul la partie centrale est modifiée grâce à un contrôle utilisateur.



- 1- Sous la photo de la compétition on a trois boutons qui permettent d'accéder à l'historique de la compétition c'est-à-dire à une liste des équipes qui ont remporté la compétition, le classement actuel de la compétition et le calendrier actuel.
- 2- On a là une liste des équipes qui participent à la compétition.
- 3- Ce bouton permet d'accéder aux commentaires qui correspondent à la compétition
- 4- Bouton « j'aime » qui lorsque l'on clic dessus rajoute un « j'aime » à la compétition.
- 5- Bouton « je n'aime pas » qui est le contraire du bouton j'aime. Les deux boutons ne peuvent pas être activés en même temps.

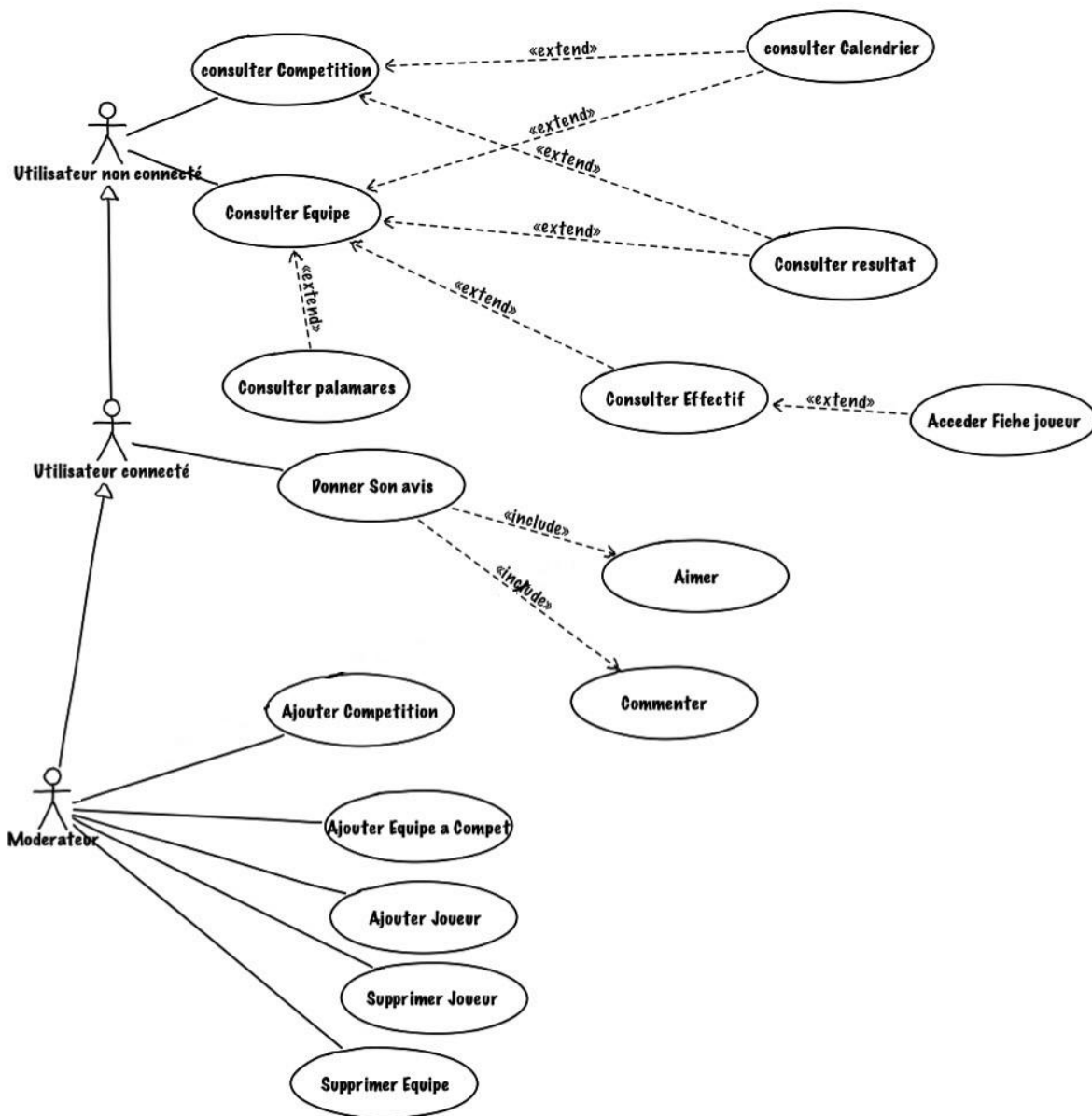
➤ Story-boards



- 1- Page d'accueil de l'application (voir description au dessus).
- 2- Lorsque que l'on sélectionne une compétition, le contrôle utilisateur est modifié et on obtient la compétition correspondante (voir au dessus).
- 3- Le contrôle utilisateur compétition est composé de la liste des équipes de la compétition. En cliquant sur une de ces équipes, le contrôle utilisateur est à nouveau modifié. En effet, il affiche l'équipe en question. Il est composé du logo de l'équipe, de deux boutons (palmarès et effectif), en dessous on a la description de l'équipe. En haut à droite on a un moyen pour équiper l'hymne de l'équipe si celle-ci en possède une. En bas, on retrouve le bouton « commenter », « j'aime » et « je n'aime pas » ainsi qu'un bouton retour qui permet de retourner à la compétition.
- 4- Si on clique sur le bouton « effectif » du sketch précédent le contrôle utilisateur affiche l'effectif de l'équipe. On a une liste des noms de joueurs selon les différents postes. Il ya la possibilité d'accéder à la fiche du joueur en cliquant sur son nom mais dans ce cas l'utilisateur doit être connecté.
- 5- Comme cela a été dit au dessus, si l'utilisateur est connecté alors il accède à la fiche du joueur qui s'affiche toujours dans le contrôle utilisateur. Sur ce dernier on a le nom et prénom du joueur suivi de différents attributs lui correspondant ainsi qu'une photo (pour certain). On a toujours en bas les différents boutons d'appréciation et le bouton de retour qui permet de revenir en arrière soit sur l'effectif complet de l'équipe en question.
- 6- A partir du sketch n° 3, on peut accéder au palmarès de l'équipe. Le contrôle utilisateur est modifié de façon à ce qu'il affiche la photo de l'équipe et le palmarès s'il y en a.
- 7- On a ici la page qui permet de se connecter en renseignant son login et son mot de passe. On y a accès à partir du bouton connexion mais on y est aussi redirigé lorsque l'on se déconnecte. Cette fois, lorsque que l'on clique sur connexion, c'est une nouvelle fenêtre qui s'ouvre mais qui ferme la fenêtre précédente. On y trouve un lien pour être redirigé vers un page d'inscription.

- 8- Voici la page d'inscription. On y retrouve différents champs qu'il faut remplir ainsi qu'un bouton pour valider l'inscription.
- 9- Une fois que l'on est inscrit on est dirigé vers notre profil qui donne les différentes informations données lors de l'inscription.

➤ Description Diagramme de Cas



Tout d'abord, nous avons décidé de distinguer trois types d'utilisateurs au sein de notre application, les Utilisateurs non-inscrits, les utilisateurs Inscrit et les Modérateurs qui auront des cas d'utilisations différents au sein de l'application sachant que l'utilisateur Inscrit peut faire tout ce que l'Utilisateur non inscrit peut faire mais l'inverse n'est pas vrai, et le modérateur aura un total contrôle (ajouts, modifications) de l'application.

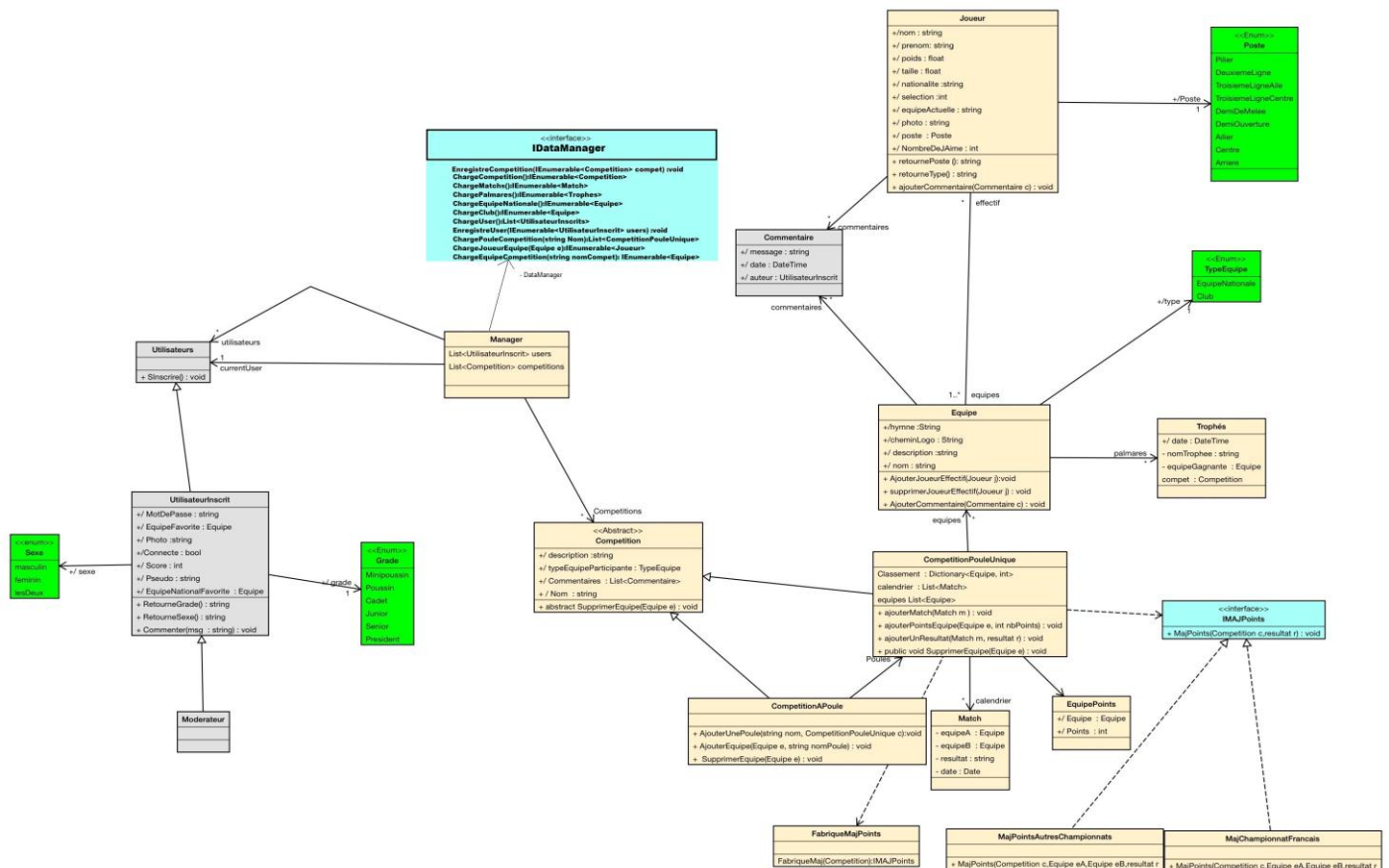
Un utilisateur qu'importe son type pourra Consulter une compétition à partir de l'accueil à partir de là plusieurs cas d'utilisation lui seront possible :

- Consulter en cliquant sur le bouton approprié le classement de cette compétition.
- Consulter le calendrier de la compétition.
- Consulter l'historique des gagnants de la Compétition.
- Consulter une équipe en cliquant sur le nom de celle-ci dans la liste des équipes de la compétition. A partir de la consultation d'une équipe, différents cas d'utilisation lui seront de nouveaux possibles.
- Consulter le calendrier d'une équipe (de toutes les compétitions dans laquelle l'équipe est engagée).
- Consulter le palmarès de l'équipe.
- Consulter son calendrier.
- Consulter l'effectif (la liste des joueurs). A partir de cet effectif, l'utilisateur pourra accéder à la fiche d'un joueur qu'il aura préalablement sélectionné en cliquant sur son nom.

Dès lors qu'un utilisateur est inscrit il pourra en plus d'effectuer tous les cas retranscrit ci-dessus Donner son avis, ce qui signifie qu'il pourra commenter et/ou aimer les différents onglets.

Lorsque l'utilisateur est un modérateur, il aura le pouvoir de modifier le contenu de l'application en ajoutant/supprimant des compétitions, des équipes au sein des compétitions, des résultats et des joueurs dans l'effectif d'une équipe.

➤ Description Diagramme de Classe



La figure ci-dessus correspond à notre diagramme de classe. On constate qu'il est assez imposant et il paraît complexe à première vue. C'est pour cela que l'on va le découper en partie pour permettre une meilleure explication. Il faut tout d'abord savoir que notre classe principale est la classe Manager car elle permet une liaison entre le métier, la persistance et la vue. On va donc voir un peu plus en détail cette classe.

1

Manager
Classe

Champs

- admins
- AllEquipe
- AllTrophes
- Competitions
- Effectif
- EquipeCompet
- Equipenat
- Matches
- MatchesTries
- PouleCompet
- ResultatsEquipe
- Trophes
- users

2

Propriétés

- AllEquipeROC
- AllTrophesROC
- CompetitionsR...
- CurrentUser
- DataManager
- EffectifROC
- EquipeCompet...
- EquipeNatROC
- MatchesROC
- MatchesTriesROC
- PouleCompetR...
- ResultatsEquipe...
- TrophesROC

3

Méthodes

- AddEquipeCom...
- AddEquipeCom...
- AjouterCompet...
- AjouterJoueur
- AjouterMatch
- AjouterResultat
- ChargeAllEquipe
- ChargeCompeti...
- ChargeCompeti...
- ChargeEffectif
- ChargeEquipe
- ChargeEquipeN...
- ChargeMatch
- ChargeMatches...
- ChargePouleCh...
- ChargerResulta...
- ChargeTropheC...
- ChargeTropheE...
- EnregistreCom...

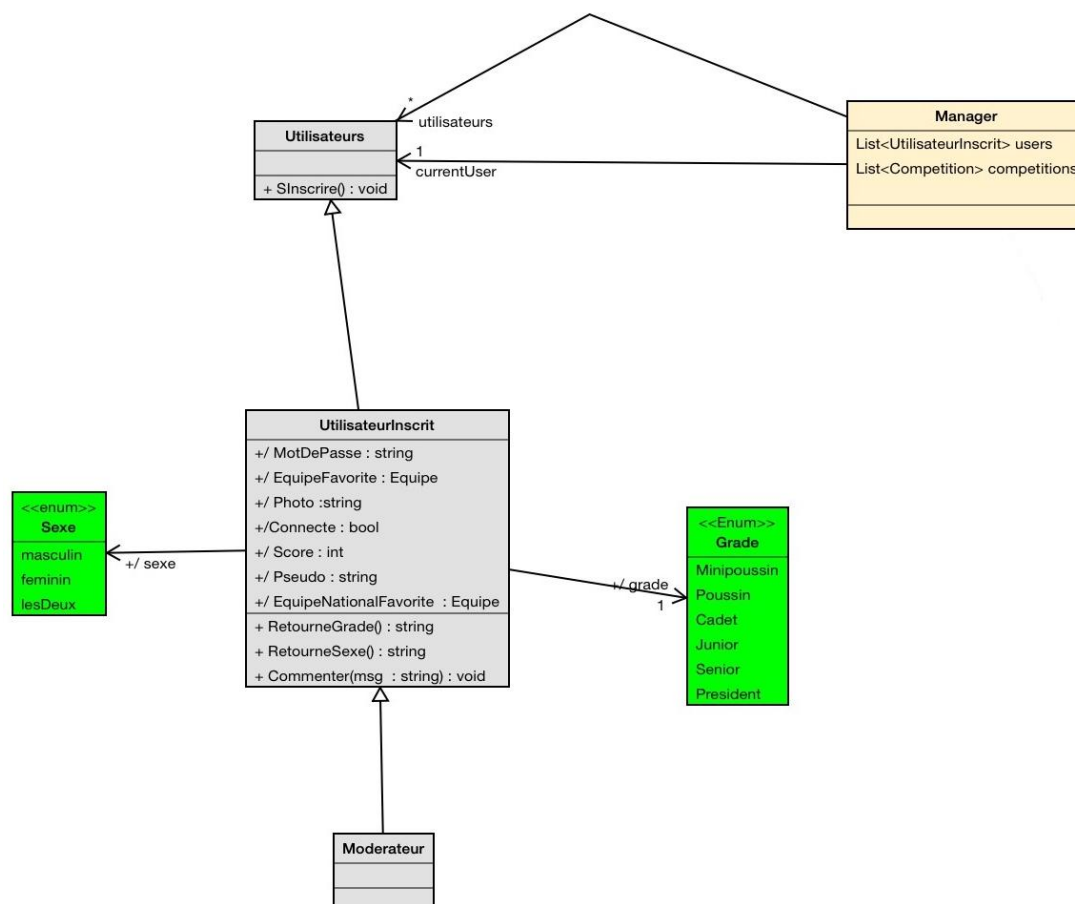
4

Méthodes

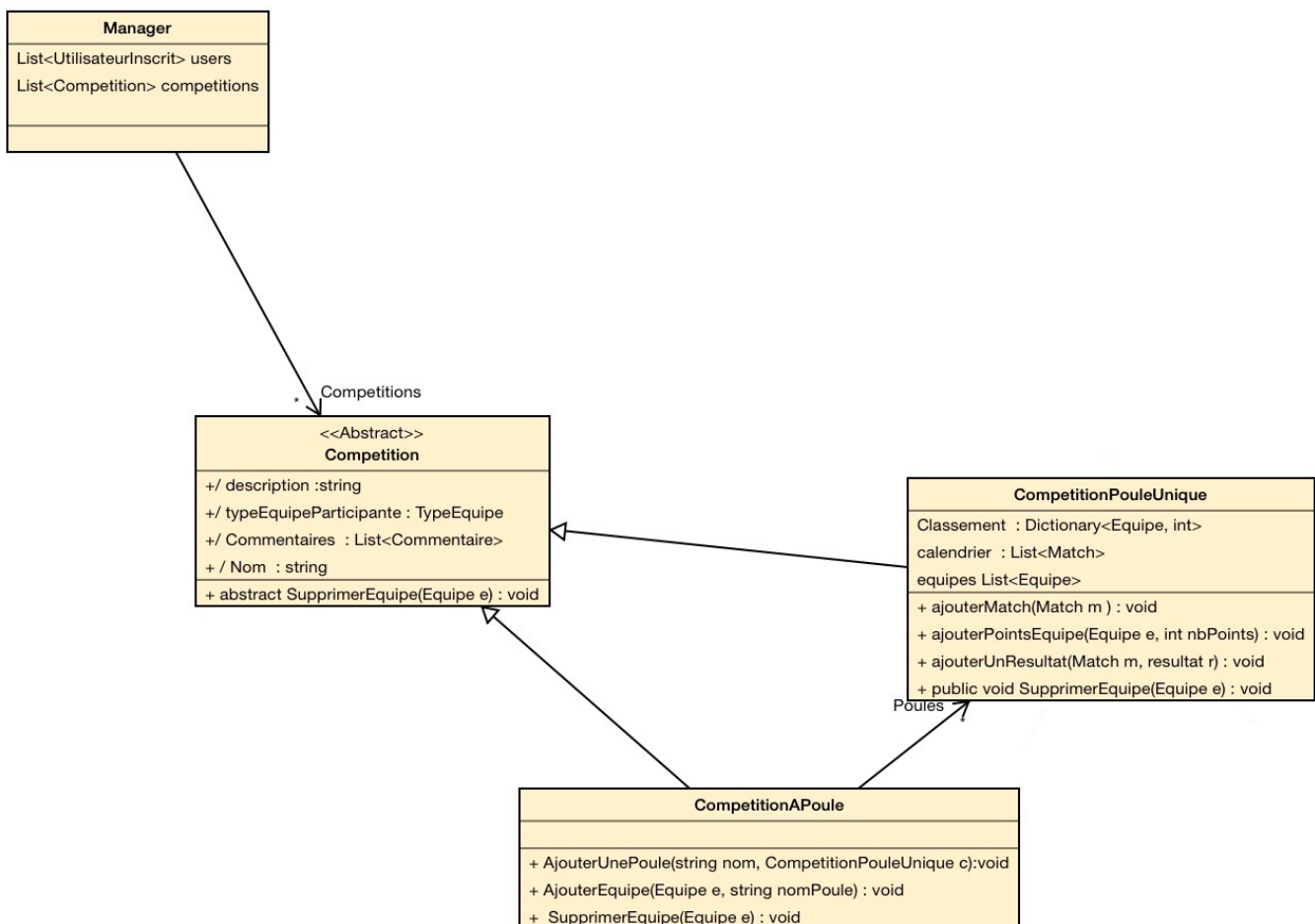
- getCalendrierE...
- getClendrierCo...
- getEffectifEquipe
- getResultatsCo...
- getResultatEqui...
- InscriptionUser
- Manager
- Retourneutilisa...
- seconnecter
- SupprimerCom...
- SupprimerEqui...
- SupprimerEqui...
- SupprimerJoueur

Les quatre figures présentes ci-dessus nous détaillent la classe Manager. En effet on peut y voir sur la figure 1 les champs de cette classe, sur la 2 les propriétés, sur la 3 et la 4 les méthodes de Manager. Les champs de la classe sont composés en partie de listes et on retrouve dans les propriétés les « ReadOnlyCollection » qui correspondent et encapsulent à ces listes. Par exemple AllEquipe est une liste qui comprend toutes les équipes que l'on trouve dans notre application et AllEquipeROC est la « ReadOnlyCollection » qui correspond. On trouve aussi une propriété nommée DataManager et qui est du type IDataManager qui est l'interface qui va permettre d'accéder au stub qui se trouve dans le package persistance. CurrentUser est du type Utilisateur et cette propriété correspond à l'utilisateur courant c'est-à-dire celui qui utilise actuellement l'application. Pour ce qui est des méthodes, on en trouve beaucoup car c'est dans le Manager que se fait le chargement des données à partir du stub et en direction de la vue. Pour prendre un exemple, ChargeAllEquipe permet de charger dans la liste AllEquipe, dont on a parlé précédemment, les données situées dans le stub. Par conséquent, toutes les méthodes qui commencent par Charge... permettent donc de charger les données du stub. La méthode nommée Manager correspond au constructeur du Manager et qui prend en paramètre un élément du type IDataManager. On y trouve aussi des méthodes qui permettent de se connecter à l'application ou de s'inscrire.

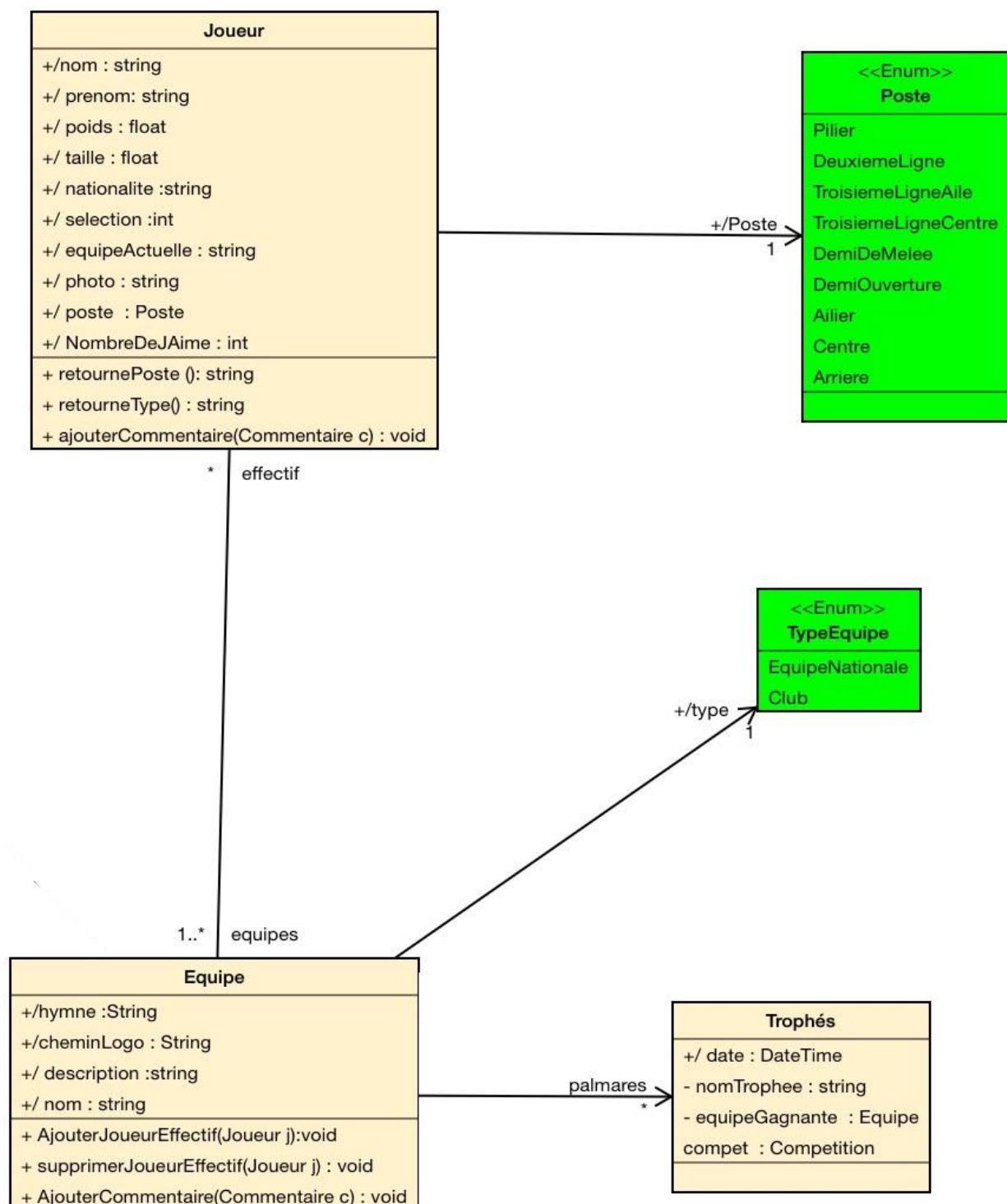
On va maintenant décrire la partie de gauche qui correspond à la partie « connexion » de notre application. Sur la figure suivante on retrouve uniquement cette partie. On y trouve une classe nommée UtilisateurInscrit qui hérite de la classe Utilisateur. C'est cette même classe Utilisateur qui va être utilisée dans le Manager car ce dernier possède une liste d'Utilisateur et un utilisateur courant (CurrentUser dont on a déjà parlé). On trouve aussi deux énumération : L'énumération Sexe qui permet de définir le sexe de l'utilisateur et l'énumération Grade qui correspond au grade attribué à l'utilisateur (on notera que l'utilisation du grade risque de ne pas être faite car par manque de temps, on a préféré négliger la partie Utilisateur et se concentrer sur le reste du métier). On peut aussi voir qu'il existe la classe Modérateur qui hérite d'UtilisateurInscrit mais pour la même raison nous avons préféré laisser de coté cette idée.



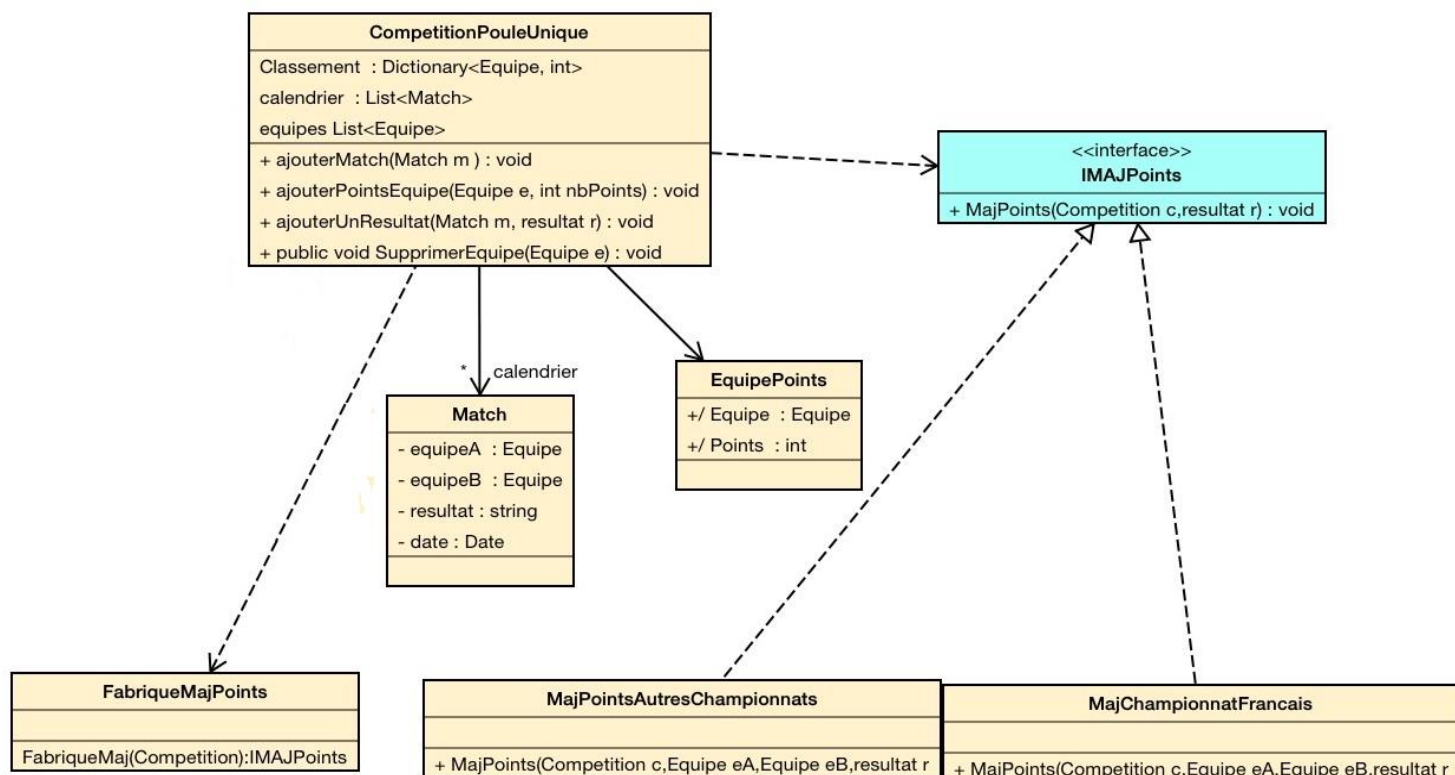
On va maintenant passer à la partie compétition. En effet le Manager possède une liste de compétition. Mais il existe deux types de compétition qui forment ainsi deux classes héritant de la même classe abstraite : Compétition. Les deux types de compétitions sont : compétition à poule et compétition à poule unique. On remarque aussi une liaison entre les deux classes représentant ces deux types. En effet, une compétition à poule possède une liste de compétition à poule unique car chaque poule de la compétition est au final une petite compétition. Chaque compétition peut importer le types possède un nom, une description, le type des équipes qui participe à la compétition (le type est choisi dans une énumération entre club et équipe nationale) et une liste de commentaire (nous reviendrons sur les commentaires plus tard). Une compétition à poule unique possède d'autres paramètres comme une map pour le classement (clé = une équipe et valeur = int correspondant au nombre de points), une liste de match pour le calendrier et la liste des équipes qui participe à la compétition.



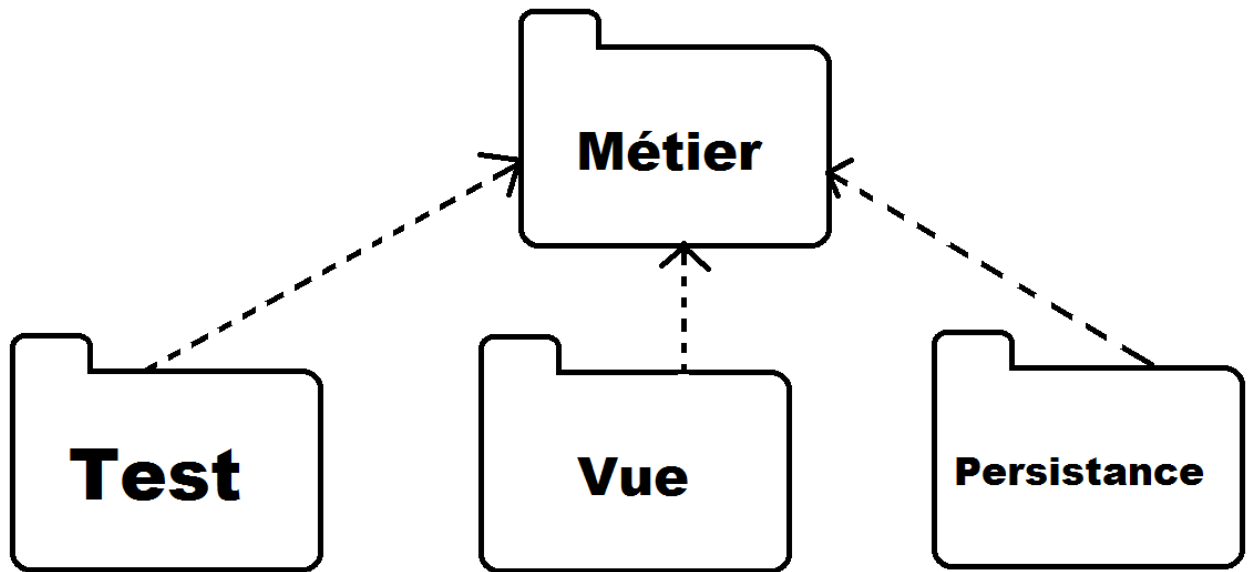
Ci-dessous, on retrouve la partie équipe de notre diagramme de classe. En effet une compétition à poule unique possède une liste d'équipe. Une équipe est composée d'un nom, d'une description, d'un chemin vers un logo et d'un chemin vers une hymne (Par manque de temps nous n'avons pas encore incorporé cette fonctionnalité à notre application). Une équipe comporte aussi un type qui provient de l'énumération TypeEquipe (deux choix possible : club ou Equipe nationale). Une équipe a aussi un effectif c'est-à-dire une liste de joueur. Un joueur possède un nom, un prénom, une nationalité, un nombre de sélection en équipe nationale, un poids et une taille qui sont facultatifs, une équipe actuelle, un nombre de j'aime, une photo (chemin) et un poste. Le poste est issu de l'énumération Poste qui regroupe tout les postes possibles au rugby. Par un souci de commodité, on a choisi qu'un joueur ne pouvait avoir qu'un seul poste. On constate aussi qu'une équipe a une liste de trophées qui forment le palmarès de l'équipe.



La dernière partie de notre diagramme de classe représente la partie qui va permettre de gérer les résultats et les classements des compétitions. C'est donc pour cette raison que l'on retrouve la classe `CompétitionPouleUnique`. En effet cette dernière possède une liste de match qui forme le calendrier. On trouve la classe `EquipePoint` qui possède en paramètre une `Equipe` et un `int` correspondant aux points de cette équipe. Cette classe va permettre dans la classe `CompétitionPouleUnique` de retourner le classement correspondant. On trouve une « fabrique » qui permet de mettre à jour les points d'une équipe. Mais elle ne met pas elle-même les points à jour, elle possède une méthode qui va regarder la compétition passée en paramètre et retourner une instance de la bonne classe à utiliser. En effet, elle va soit retourner une instance de la classe `MajPointsAutresChampionnats` soit celle de `MajPointsChampionnatsFrancais`. Ces deux classes ont la même origine car elles héritent de l'interface `IMAJPoints`. On a créé deux classes pour mettre à jour car les deux championnats français (Top 14 et Pro d2) sont différents que les autres championnats dans le monde. En effet, dans les championnats français, les points de bonus sont différents que dans les autres championnats. De plus, si une nouvelle règle attribution des points est inventée, il nous suffit de créer une nouvelle classe héritant de `IMAJPoints` qui réécrit la méthode de l'interface selon la nouvelle règle.



➤ Description Diagramme de Paquetage



Comme on peut le distinguer sur la figure ci-dessus notre projet est constitué de 4 Paquetages. Nous allons maintenant décrire quelle est le rôle de chaque paquetage.

1) Le Métier

Le paquetage Métier constitue l'épine dorsale du projet, en effet comme on peut le voir sur le diagramme tous les autres paquetages l'utilisent. Le métier contient toutes les classes qui constituent l'application, comme par exemple Joueur, Equipe ou encore Compétition, c'est à partir de là que sont créés tous ces objets. Il contient aussi toutes les méthodes permettant d'agir sur ces objets (les modifier, leurs ajouter quelque chose...)

2) La Vue

Le Paquetage Vue contient lui les différents éléments graphiques de l'application. Ainsi on retrouve dans la Vue les différents Contrôles Utilisateur et page de base mais aussi les éléments qui permettent à la Vue d'utiliser le paquetage métier pour permettre l'affichage des bons éléments.

3) Test

Le Paquetage Test permet comme son nom l'indique d'effectuer des tests sur le métier afin de vérifier que tous les éléments du métier (attributs et méthodes) fonctionnent correctement et ne génèrent pas d'exception et que les différentes méthodes effectuent bien la ou les actions que l'on souhaite.

4) Persistance

Le Paquetage Persistance contient les éléments de Sauvegarde et de chargement de l'application. En effet ce package permet à chaque utilisation de charger les éléments du métier (obtenir la dernière sauvegarde faite sur les éléments du métier) et après l'utilisation de sauvegarder ces éléments. Il existe différents types de Persistance (binaire, SQL...), notre package en contient deux types qui sont :

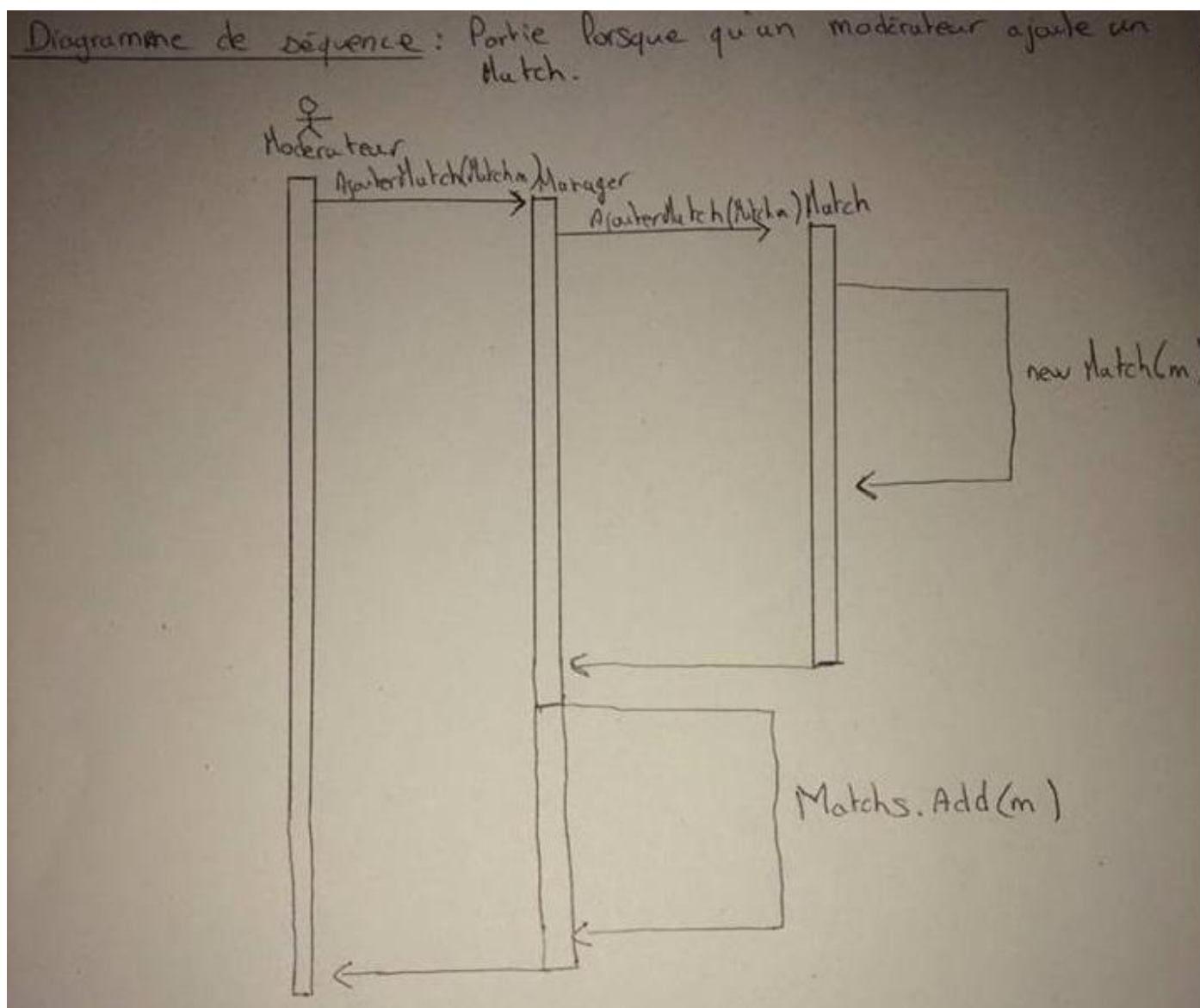
Le StubData qui contient des données en « brut » ce qui signifie que l'on peut juste charger les éléments et non les sauvegarder.

La persistance binaire qui permet le chargement et la sauvegarde sur des fichiers en binaire.

Pour pouvoir afficher les données, la vue doit avoir une référence sur la persistance mais aussi sur le métier pour permettre l'utilisation des fonctionnalités du métier. Les tests doivent avoir une référence au métier car ils permettent de tester le métier. Il doit aussi avoir en référence sur la persistance pour avoir accès aux données qu'elle contient.

Le métier possède une classe Manager qui est ce que l'on nomme la façade (patron de conception nommé « Façade »). Cette classe possède un attribut de type IDataManager. Cette interface impose l'utilisation de méthodes spécifiques permettant la sauvegarde et la lecture des données. Cette interface est implémentée par les classes de la persistance. Ceci est le patron de conception « stratégie ».

➤ Diagramme de séquence



On a ci-dessus un diagramme de séquence qui correspond à l'ajout d'un Match par un modérateur. On voit que lorsque le Modérateur ajoute un Match, c'est le Manager qui est appelé et qui crée un nouvel objet de type Match qu'il ajoute ensuite à la liste qui possède tous les Matches de l'application.

Login et mot de passe pour se connecter :

Modérateur : login = admin et mot de passe = admin

Simple utilisateur : login = antho et mot de passe = 1105