

Architecture des Applications Réparties
Rapport de Projet
Gestion d'un tournoi de football

Geoffrey CROCHET, Zo RABARIJAONA, Willy FRANÇOIS

2 décembre 2013

Table des matières

1	Analyse UML	3
1.1	Diagrammes de classes	3
1.1.1	Le modèle	3
1.1.2	Les EJB	5
1.1.3	Les Value Objects	9
1.2	Diagramme de cas d'utilisation	10
1.3	Schéma synthétique de l'application	11
2	Répartition du travail	12
2.1	Gestion du temps de travail	12
2.2	Répartition des tâches	12
3	Guide d'utilisation	13

Introduction

Le projet demandé consiste à développer une application permettant la gestion d'un tournoi de football. Pour ce faire, nous nous appuierons sur la technologie Java EE et l'utilisation des EJB 3.0 afin de pouvoir persister des objets dans la base de données interne du serveur d'application JBoss.

Ce rapport présentera tout d'abord une analyse UML de l'ensemble de notre application en visualisant les diagrammes de classes, le diagramme de cas d'utilisation ainsi qu'un schéma synthétique de l'architecture du projet. Nous évoquerons ensuite comment le temps de travail a été géré et la répartition des différentes tâches au sein du groupe. Enfin, un guide d'utilisation sera présenté expliquant le fonctionnement général de l'application.

Chapitre 1

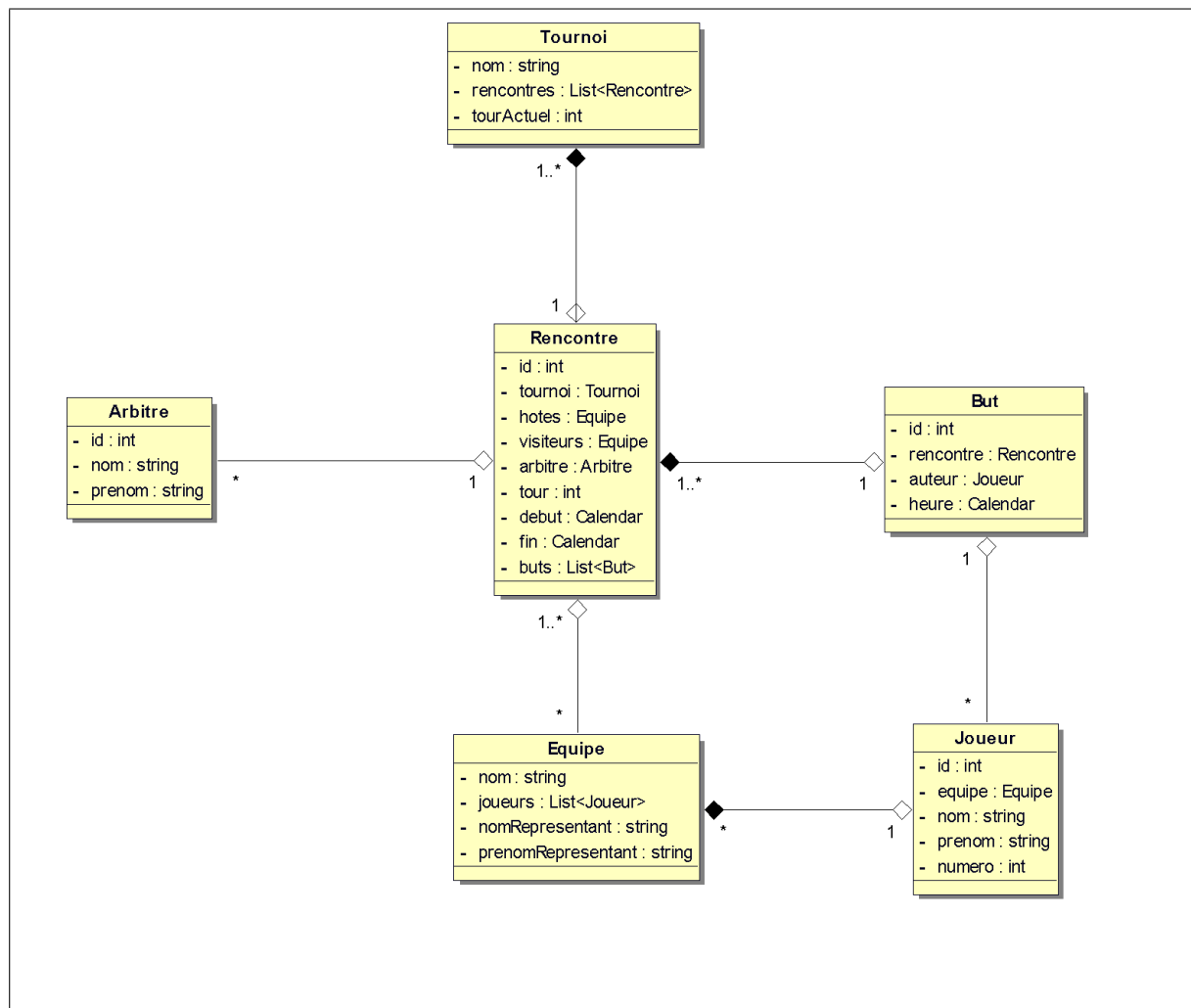
Analyse UML

1.1 Diagrammes de classes

Notre application se compose de trois principaux packages : le modèle contenant des objets POJO, les EJB permettant la persistance des objets et les Value Objects utilisés dans les vues de l'application.

1.1.1 Le modèle

Le schéma ci-dessous représente le diagramme de classes de notre modèle.



blabla

1.1.2 Les EJB

Le schéma ci-dessous représente le diagramme de classes de l'EJB Utilisateur créé.

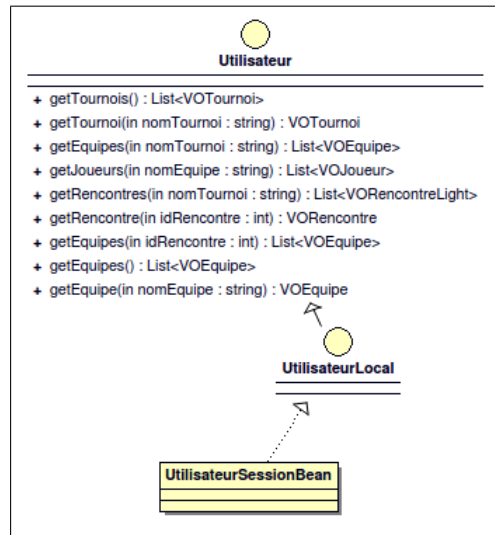
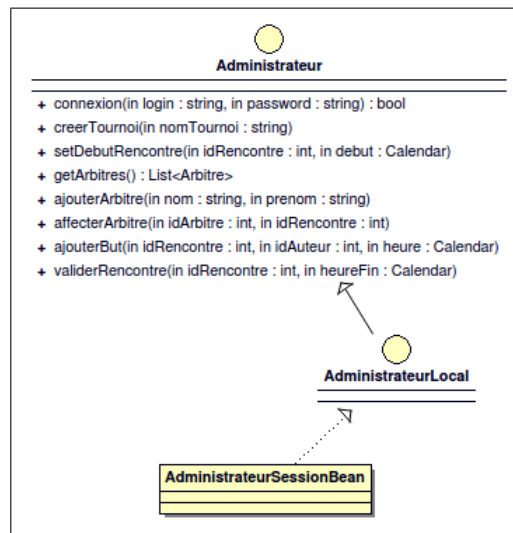


Diagramme de classes de l'EJB Utilisateur

L'EJB Utilisateur se compose de deux interfaces (*Utilisateur* et *UtilisateurLocal*) et d'une classe d'implémentation (*UtilisateurSessionBean*). L'interface *Utilisateur* définit les profils des méthodes implémentées par la classe *UtilisateurSessionBean*. L'interface *UtilisateurLocal* correspond à l'interface locale de l'EJB utilisée par la facade de notre application. L'ensemble des méthodes définies dans cet EJB retournent un Value Object ou une liste de Value Objects. En effet, l'utilisateur ne doit pas être en mesure de connaître l'ensemble des attributs et méthodes des classes du modèle.

Le schéma ci-dessous représente le diagramme de classes de l'EJB Administrateur créé.



Diagrammes de classes de l'EJB Administrateur

L'EJB Administrateur se compose de deux interfaces (*Administrateur* et *AdministrateurLocal*) et d'une classe d'implémentation (*AdministrateurSessionBean*). L'interface *Administrateur* définit les profils des méthodes implémentées par la classe *AdministrateurSessionBean*. L'interface *AdministrateurLocal* correspond à l'interface locale de l'EJB utilisée par la facade de notre application. Les méthodes définies concernent la connexion de l'administrateur à l'application, la création d'un tournoi ou encore le déroulement des matchs (indication des horaires, affectation des arbitres).

Le schéma ci-dessous représente le diagramme de classes de l'EJB Représentant créé.

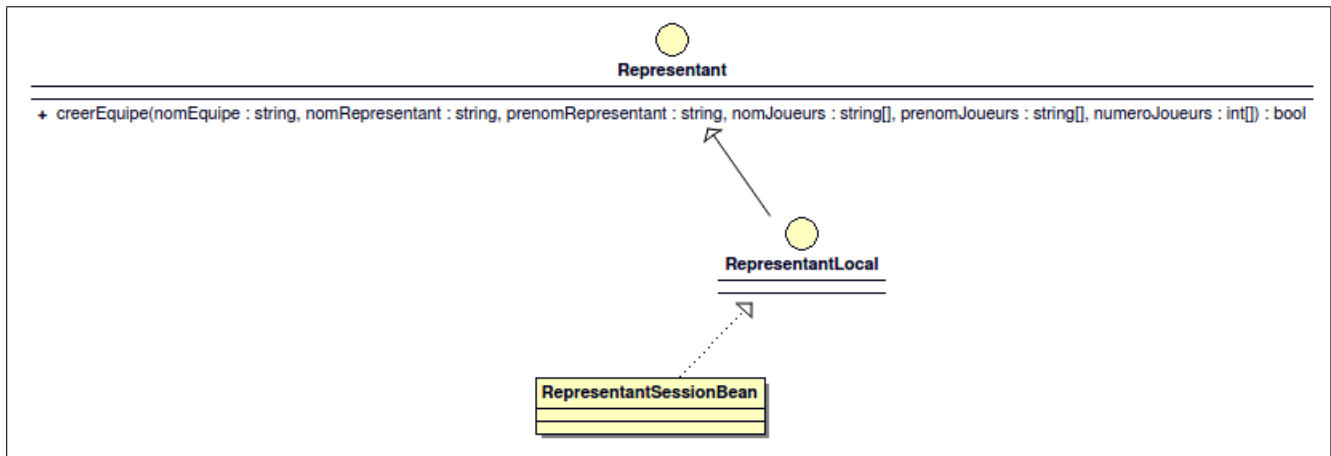


Diagramme de classes de l'EJB Représentant

L'EJB Représentant se compose de deux interfaces (*Representant* et *RepresentantLocal*) et d'une classe d'implémentation (*RepresentantSessionBean*). L'interface *Representant* définit le profil de la méthode implémentée par la classe *RepresentantSessionBean*. L'interface *RepresentantLocal* correspond à l'interface locale de l'EJB utilisée par la facade de notre application. La méthode définie consiste à la création d'une équipe au sein de l'application.

Le schéma ci-dessous représente le diagramme de classes de l'EJB session Facade créé.

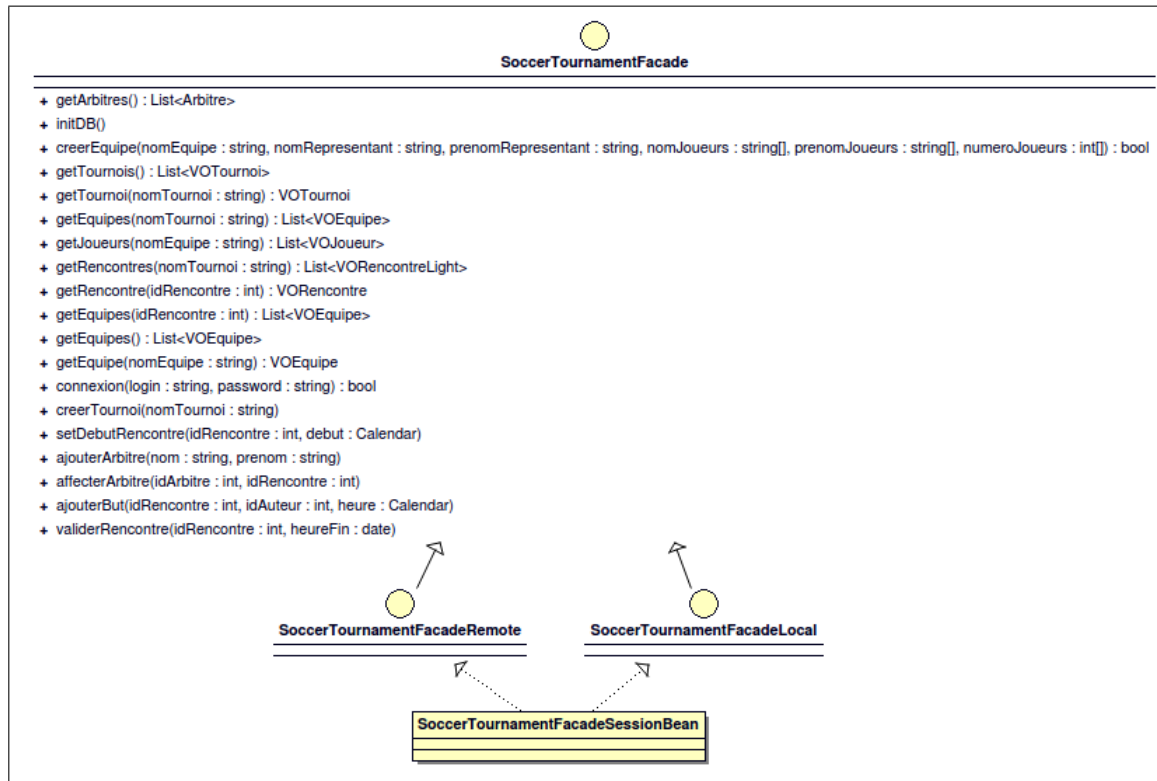


Diagramme de l'EJB session Facade

L'EJB session Facade se compose de trois interfaces (*SoccerTournamentFacade*, *SoccerTournamentFacadeLocal* et *SoccerTournamentFacadeRemote*) et d'une classe d'implémentation (*SoccerTournamentFacadeBean*). L'interface *SoccerTournamentFacadeLocal* correspond à l'interface locale de l'EJB utilisée dans notre application. Nous avons également créé l'interface *SoccerTournamentFacadeRemote* dans le cas où nous devions atteindre la façade à distance. Cet EJB regroupe l'ensemble des méthodes définies dans les EJB créés ainsi qu'une méthode *InitDB()* nous permettant d'initialiser la base de données au lancement de l'application. La base de données se charge ainsi, à l'aide de fichiers XML préalablement remplis.

1.1.3 Les Value Objects

Le schéma ci-dessous représente le diagramme de classes des Value Objects créés. blabla

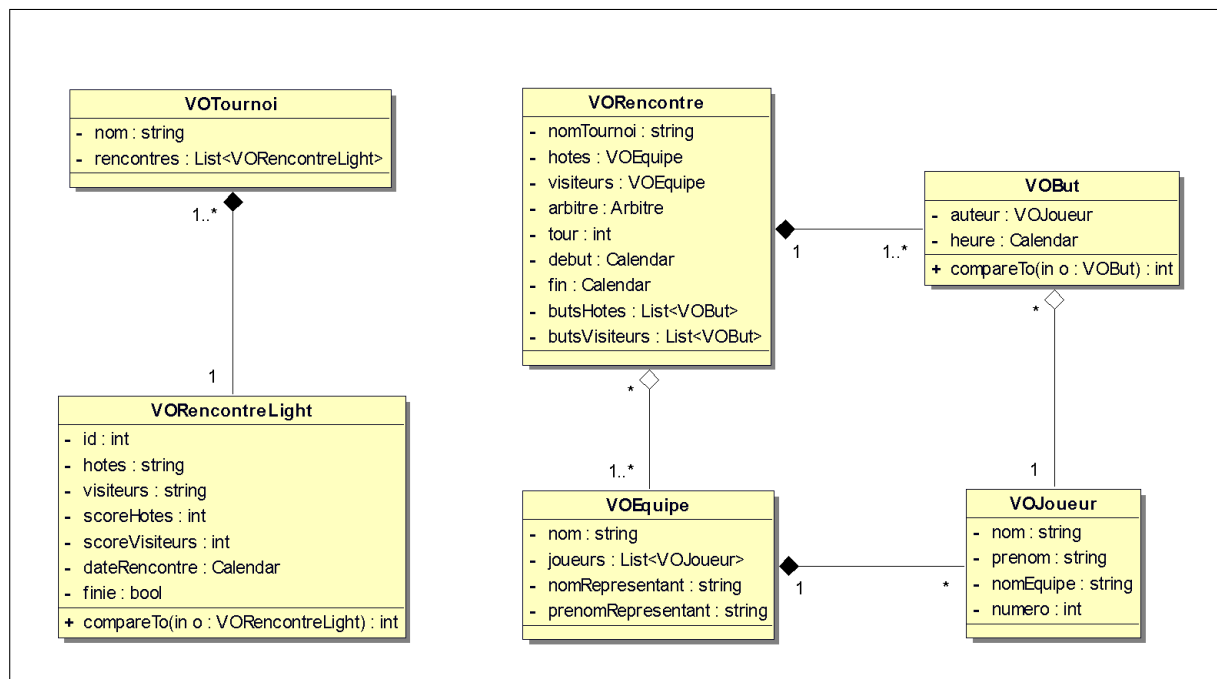


Diagramme de classes des Value Objects

1.2 Diagramme de cas d'utilisation

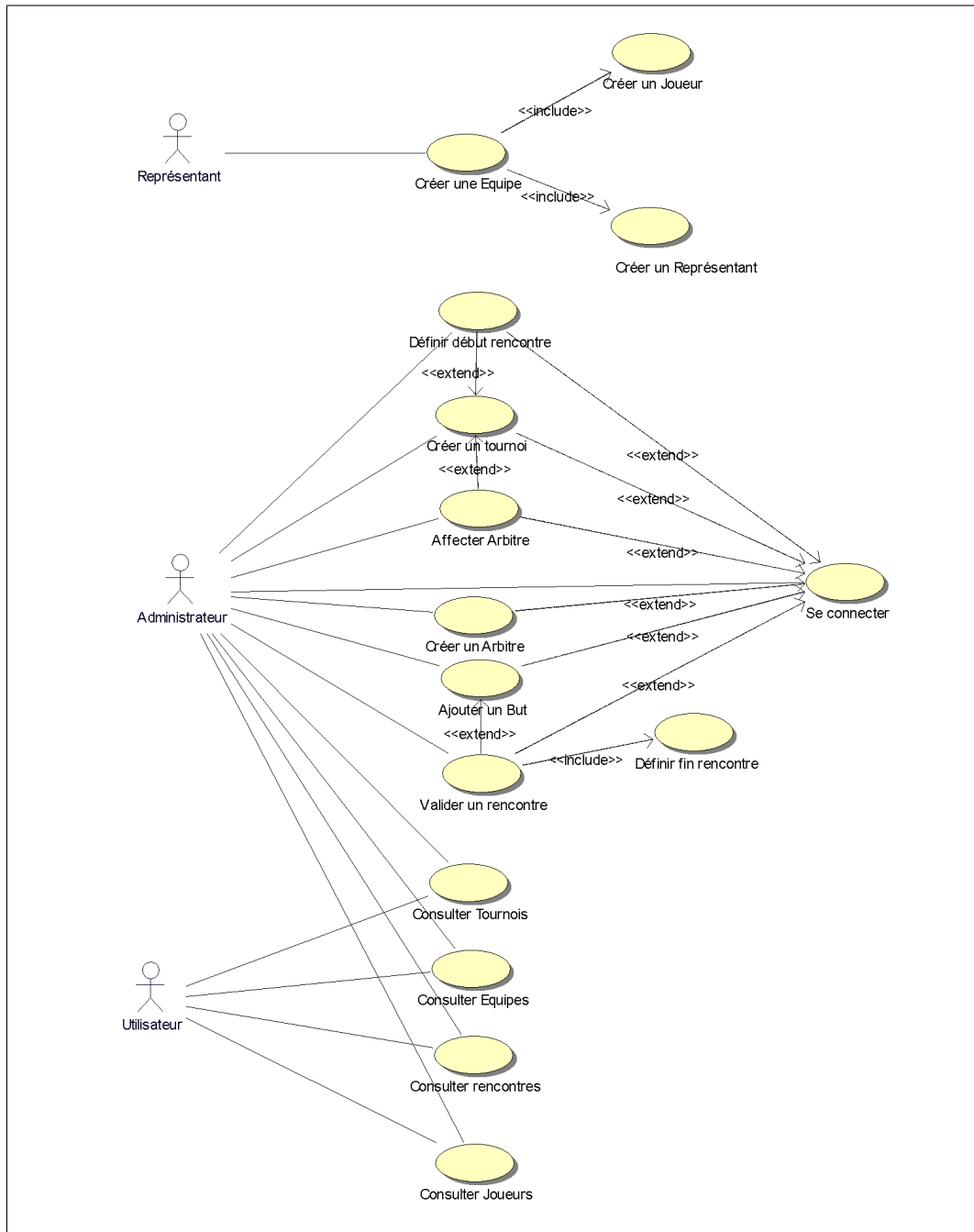


Diagramme de cas d'utilisation

blabla

1.3 Schéma synthétique de l'application

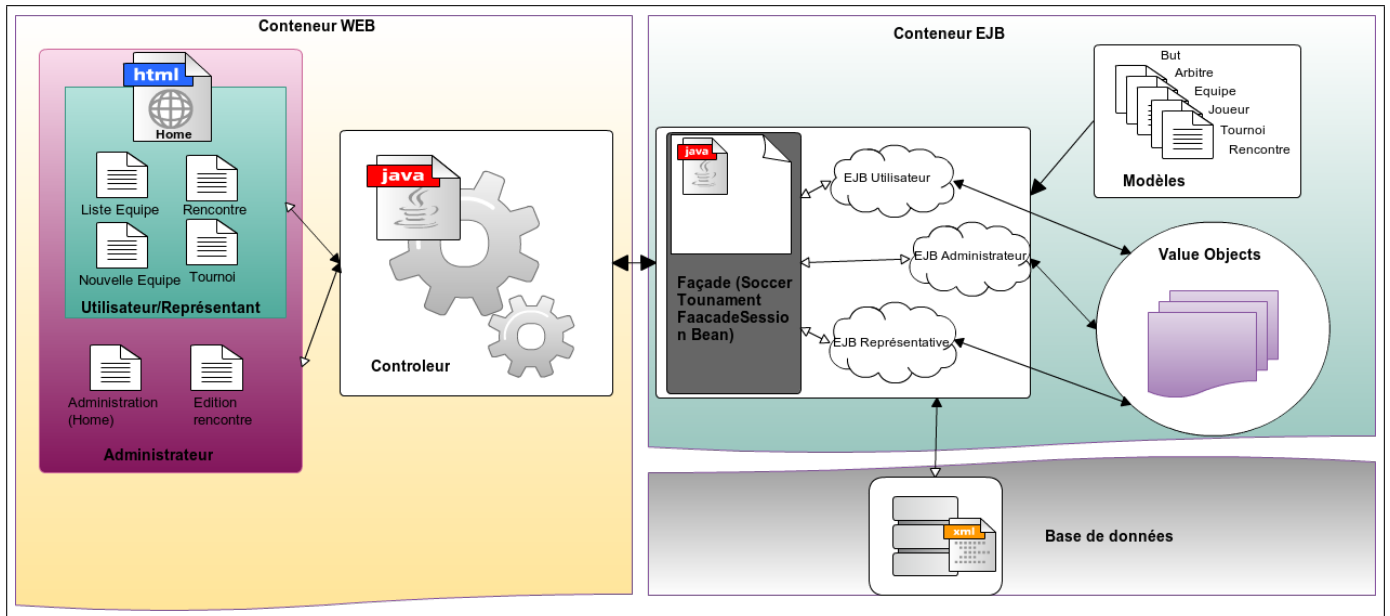


Schéma synthétique de l'application

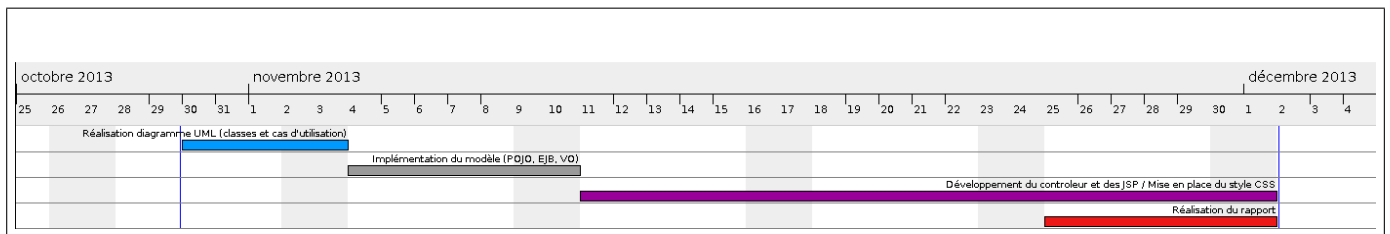
blabla

Chapitre 2

Répartition du travail

2.1 Gestion du temps de travail

Le diagramme de Gantt ci-dessous présente la répartition du travail tout au long du projet.



Gestion du temps de travail

La première semaine de travail fut consacrée à l'analyse UML et à l'élaboration des diagrammes de classes et des diagrammes de cas d'utilisation. Une fois cela fait, nous avons débuté l'implémentation du modèle de l'application à savoir les objets POJO, les EJB ainsi que les Value Objects. Le développement du contrôleur et des vues utilisateurs et administrateurs a été réalisé ensuite jusqu'à la fin du projet. Le rapport, quant à lui, a été rédigé dans la dernière semaine de travail.

2.2 Répartition des tâches

Chapitre 3

Guide d'utilisation

Conclusion

Ce projet aura permis de mettre en oeuvre les connaissances vues tout au long du semestre. Le groupe a pu améliorer ses compétences dans l'utilisation de la technologie Java EE et a acquis les principales notions de persistance d'objets, dans une base de données, avec l'usage des EJB. L'intérêt des Value Objects a été compris lors de ce projet et, les membres du groupe ont également renforcés leurs compétences dans le domaine du Web en travaillant sur un contrôleur et ses JSP.