

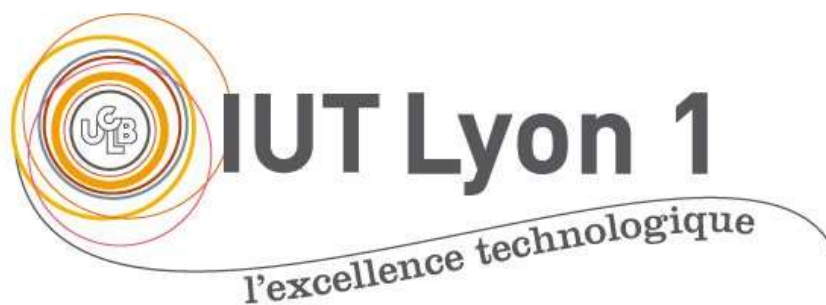
Cahier d'analyse



Oubah Karim – Bolot Martin – Fischer Remy – Bronner Loïc - Merle Yoann

DUT INFORMATIQUE – Année spéciale 2015/2016

Enseignant : **Benouaret Karim**



Sommaire

I - Définition du sujet.....	3
II - Dictionnaire des termes utilisés.....	8
III - Descriptif de l'équipe	9
1) Les acteurs	9
Martin Bolot (Chef de projet)	9
Loïc Bronner	9
Rémy Fischer	9
Yoann Merle.....	9
Karim Oubah	9
2) Répartition des lots	10
IV - Outils choisis	11
1) NetBeans	11
2) Oracle.....	11
3) ArgoUML.....	11
4) PowerAMC.....	11
5) Git / GitHub	11
V - Diagrammes	12
1) Diagramme UML de classes.....	12
2) Diagramme de Gantt	13
3) Diagramme de séquence.....	14
4) Diagramme MCD (modèle conceptuel de données).....	15
5) Maquettes de l'application	16
Interface générale.....	16
Interface européenne	18
Interface nationale.....	19
6) Diagramme de cas d'utilisation	20
7) Planning et rôle des intervenants sous forme de carte	21

I - Définition du sujet

Ce cours d'application de synthèse va nous permettre de concevoir une application dans son ensemble en passant par toutes les phases de conception et de développement nécessaires.

Le sujet est le suivant : Nous devons assurer la conception et la mise en œuvre d'un logiciel de gestion des fédérations qui ont en charge l'organisation et la gestion des différentes compétitions de football en Europe.

Ne pouvant pas éditer une base de donnée très complète avec tous les clubs européens ni toutes les divisions, nous avons décidé de nous focaliser sur huit fédérations. Celles-ci possèdent deux divisions et disputent une coupe nationale selon le modèle suivant :

- Allemagne
 - 1. Bundesliga
 - 18 équipes
 - 2. Bundesliga
 - 18 équipes
 - Coupe nationale
 - 36 équipes
 - Tour Préliminaire avec 8 équipes
 - Système classique démarrant aux 16èmes de finale avec les 4 équipes vainqueurs du tour préliminaire mélangées avec les équipes restantes

- Angleterre
 - Premier League
 - 20 équipes
 - Championship
 - 24 équipes
 - Coupe nationale
 - 44 équipes
 - Tour Préliminaire avec 24 équipes
 - Système classique démarrant aux 16èmes de finale avec les 12 équipes vainqueurs du tour préliminaire mélangées avec les équipes restantes
- Espagne
 - Liga
 - 20 équipes
 - Liga Adelante
 - 22 équipes
 - Coupe nationale
 - 42 équipes
 - Tour Préliminaire avec 20 équipes
 - Système classique démarrant aux 16èmes de finale avec les 10 équipes vainqueurs du tour préliminaire mélangées avec les équipes restantes

- Italie
 - Serie A
 - 20 équipes
 - Serie B
 - 22 équipes
 - Coupe nationale
 - 42 équipes
 - Tour Préliminaire avec 20 équipes
 - Système classique démarrant aux 16èmes de finale avec les 10 équipes vainqueurs du tour préliminaire mélangées avec les équipes restantes
- France
 - Ligue 1
 - 20 équipes
 - Ligue 2
 - 20 équipes
 - Coupe nationale
 - 40 équipes
 - Tour Préliminaire avec 16 équipes
 - Système classique démarrant aux 16èmes de finale avec les 8 équipes vainqueurs du tour préliminaire mélangées avec les équipes restantes

- Portugal
 - Liga NOS
 - 18 équipes
 - Segunda Liga
 - 24 équipes
 - Coupe nationale
 - 42 équipes
 - Tour Préliminaire avec 20 équipes
 - Système classique démarrant aux 16èmes de finale avec les 10 équipes vainqueurs du tour préliminaire mélangées avec les équipes restantes
- Pays-Bas
 - Eredivisie
 - 18 équipes
 - Eerste Divisie
 - 20 équipes
 - Coupe nationale
 - 38 équipes
 - Tour Préliminaire avec 12 équipes
 - Système classique démarrant aux 16èmes de finale avec les 6 équipes vainqueurs du tour préliminaire mélangées avec les équipes restantes

- Russie
 - Division 1
 - 16 équipes
 - Division 2
 - 14 équipes
 - Coupe nationale
 - 42 équipes
 - Tour Préliminaire avec 28 équipes
 - Système classique démarrant aux 8èmes de finale avec les 14 équipes vainqueurs du tour préliminaire mélangées avec les équipes restantes

Nous avons également introduit la Ligue des Champions et la Ligue Europa. On considère que les quatre premiers de chaque première division de chaque fédération accèdent aux phases de poule de la Ligue des champions. Un classement entre la 5ème à la 10^{ème} place (incluses) permet d'intégrer la phase de poule de la Ligue Europa. Le reste de ces deux compétitions fonctionne de la manière suivante :

- Ligue des Champions
 - Phase de poule
 - 8 groupes constitués de 4 équipes
 - Matches allés/retours
 - Les 2 premiers accèdent à la phase finale
 - Le troisième est basculé en phase finale de Ligue Europa
 - Le quatrième est éliminé
 - Phase finale
 - Système de matches allés/retours démarrant au stade des 8èmes de finale.
 - Pas de match retour pour la finale

- Ligue Europa
 - Phase de poule
 - 12 groupes constitués de 4 équipes
 - Matches allés/retours
 - Les 2 premiers accèdent à la phase finale
 - Le troisième et le quatrième sont éliminés
 - Phase finale
 - Système de matches allés/retours démarrant au stade des 16èmes de finale.
 - Pas de match retour pour la finale

II - Dictionnaire des termes utilisés

POO : « Programmation Orientée Objet », modèle de programmation sur lequel est basé Java, le langage que nous allons utiliser pour ce projet.

UML : « Unified Modeling Language », langage de modélisation des données grâce auquel nous réalisons, entre autres, notre diagramme de classes.

MCD : « Modèle Conceptuel de Données », un modèle permettant de représenter des entités de données ainsi que leurs relations.

IDE : « Intelligent Development Environment », environnement de développement intelligent. Un exemple d'IDE est NetBeans.

SGBD : Système de Gestion de Bases de Données, logiciel qui gère la manipulation de données stockées sur disque et offre une interface utilisateur pour les définir et les manipuler.

III - Descriptif de l'équipe

1) Les acteurs

Par ordre alphabétique :

Martin Bolot (Chef de projet)

Disposant d'une expérience de trois ans dans le domaine du développement Web, il a été choisi comme chef de projet car il a déjà pu observer dans un cadre professionnel le cycle de vie d'une application. Il s'occupera de la coordination de l'équipe, de la rédaction de certains documents fonctionnels, mais aussi du développement d'une partie de l'application.

Loïc Bronner

A la fois grand amateur de football et spécialiste des interfaces graphiques, Loïc possède une connaissance exhaustive du fonctionnement des championnats ciblés par notre application. Ainsi, il aura pour rôle de gérer la partie interactive du logicielle et de guider les acteurs du projet ayant le moins de connaissance sur le sujet du football afin d'arriver à une structure logique cohérente.

Rémy Fischer

Rémy est issu du milieu scientifique et plus précisément d'une filière mathématique, son domaine de compétence porte vers l'algorithme. Il aura donc pour rôle, entre autres, de définir en amont les couches logiques qui seront utilisés dans les différentes parties de l'application.

Yoann Merle

Très familier avec la POO et avec le langage UML, il s'occupera de la définition du diagramme de classes et de son implémentation concrète dans l'application, mais aussi de veiller à la cohérence de la gestion des classes par rapport au MCD.

Karim Oubah

Karim a précédemment suivi un cursus en GEA, possède de solides connaissances en gestion et sera en charge d'une partie de la gestion des plannings. Son projet tuteuré est basé sur le Java, il sera donc le référent technique pour toutes les questions ayant trait au développement de l'application Java et à la couche d'accès aux données.

2) Répartition des lots

Deux sous-groupes ont été formés pour la répartition des tâches :

- Martin, Yoann et Karim s'occuperont des championnats nationaux
- Loïc et Rémy s'occuperont des championnats internationaux

Les différents lots ont été répartis de la manière suivante (en plus des tâches prédéfinies dans la partie description de l'équipe).

Martin devra rédiger le cahier d'analyse et produire le diagramme de séquences de l'application.

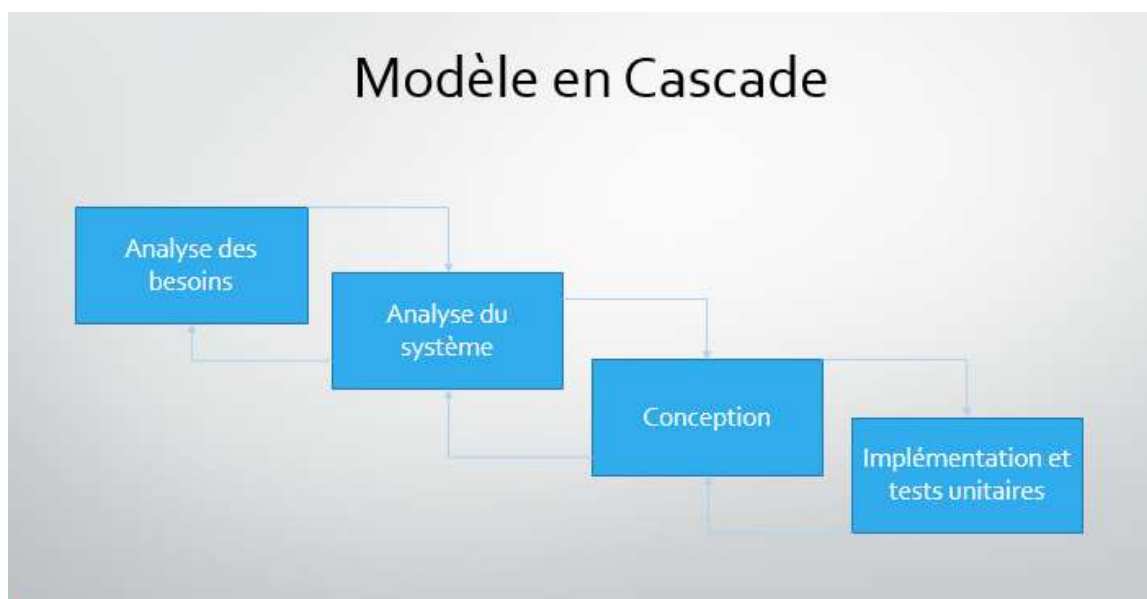
Loïc sera en charge de la conception des maquettes puis de la réalisation de l'interface graphique.

Rémy aura pour rôle de créer les algorithmes nécessaires au traitement des données de l'application telles que la génération d'un classement par exemple.

Karim réalisera les diagrammes de Gantt ainsi que les diagrammes de cas d'utilisations (use cases) avant de se charger de réaliser le modèle conceptuel de données (MCD). C'est, à terme, lui qui gèrera la persistance physique des données de l'application.

Yoann se chargera quant à lui de produire le diagramme de classes puis de fournir le code Java des différentes classes qui composeront notre programme.

Nous avons utilisé le modèle en cascade car il s'agit d'un petit projet.



IV - Outils choisis

1) NetBeans

Nous avons choisi NetBeans comme environnement de développement pour notre application Java. En effet, il s'agit d'un IDE gratuit et performant, installé sur toutes les machines sur lesquelles nous allons travailler.



2) Oracle

Nous allons utiliser Oracle comme SGBD pour la gestion des données persistantes de l'application. Ce logiciel sera utilisé pour deux raisons : premièrement l'équipe en a une bonne connaissance car il a été étudié en cours via deux modules, et deuxièmement car il s'intègre facilement dans un environnement Java.



3) ArgoUML

ArgoUML est un outil gratuit qui nous servira à effectuer une partie de nos diagrammes UML.



4) PowerAMC

PowerAMC est un outil payant mais disponible sur les machines de l'IUT qui nous permettra de réaliser les diagrammes UML les plus complexes que nous aurons à produire (notamment le diagramme de classes), plus facilement faisable que sous ArgoUML.



5) Git / GitHub

Nous utiliserons Git comme logiciel de gestion de version durant le développement de ce programme Java, ainsi que la plateforme afin d'héberger notre projet en open-source. Les outils utilisés pour Git seront la ligne de commande et l'interface graphique. Seules les sources du programme seront « versionnées » (dossier « src/ ») afin que chaque acteur du projet puisse participer en créant son propre projet NetBeans sur sa machine.



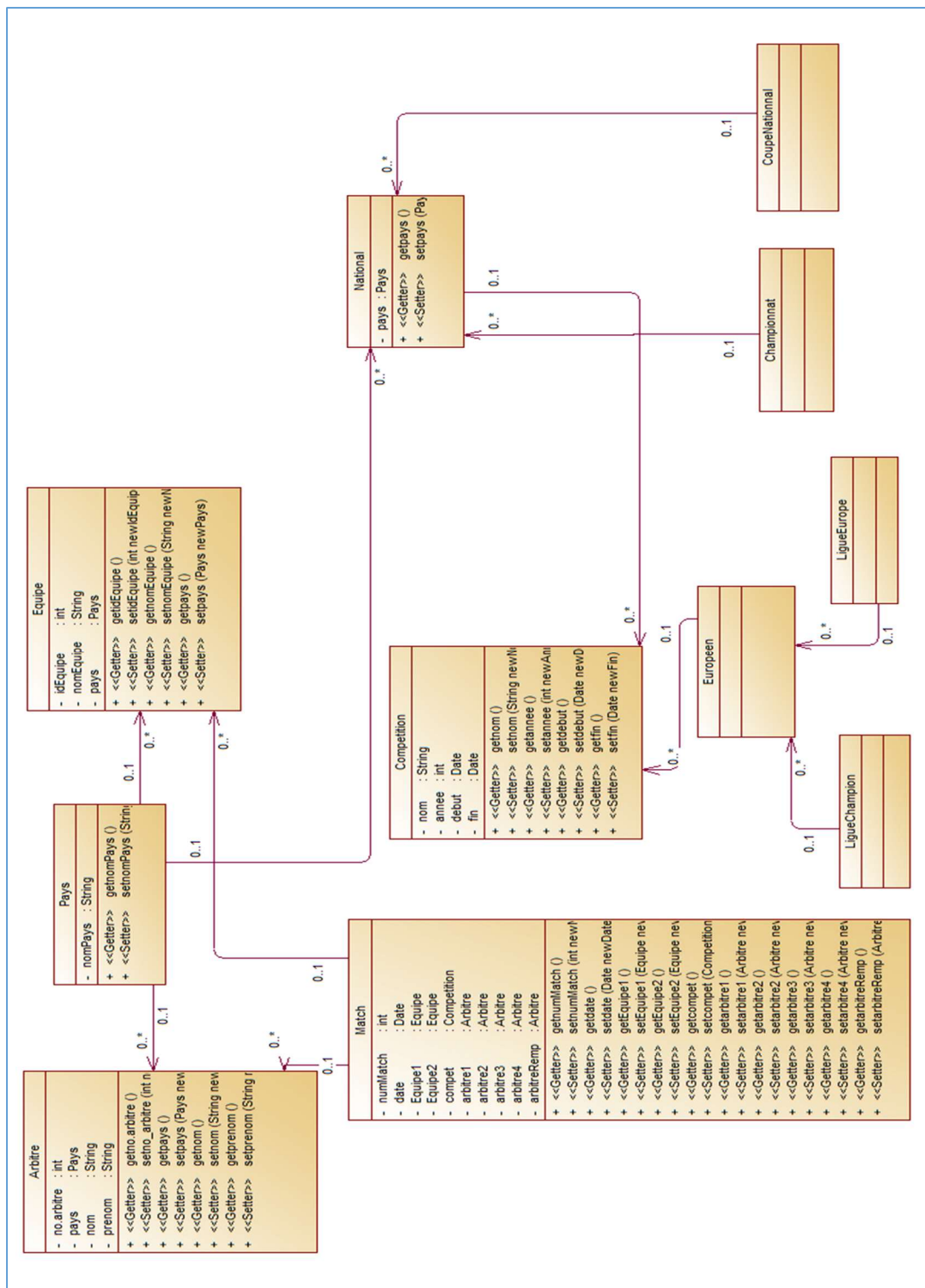
GitHub

V - Diagrammes

1) Diagramme UML de classes

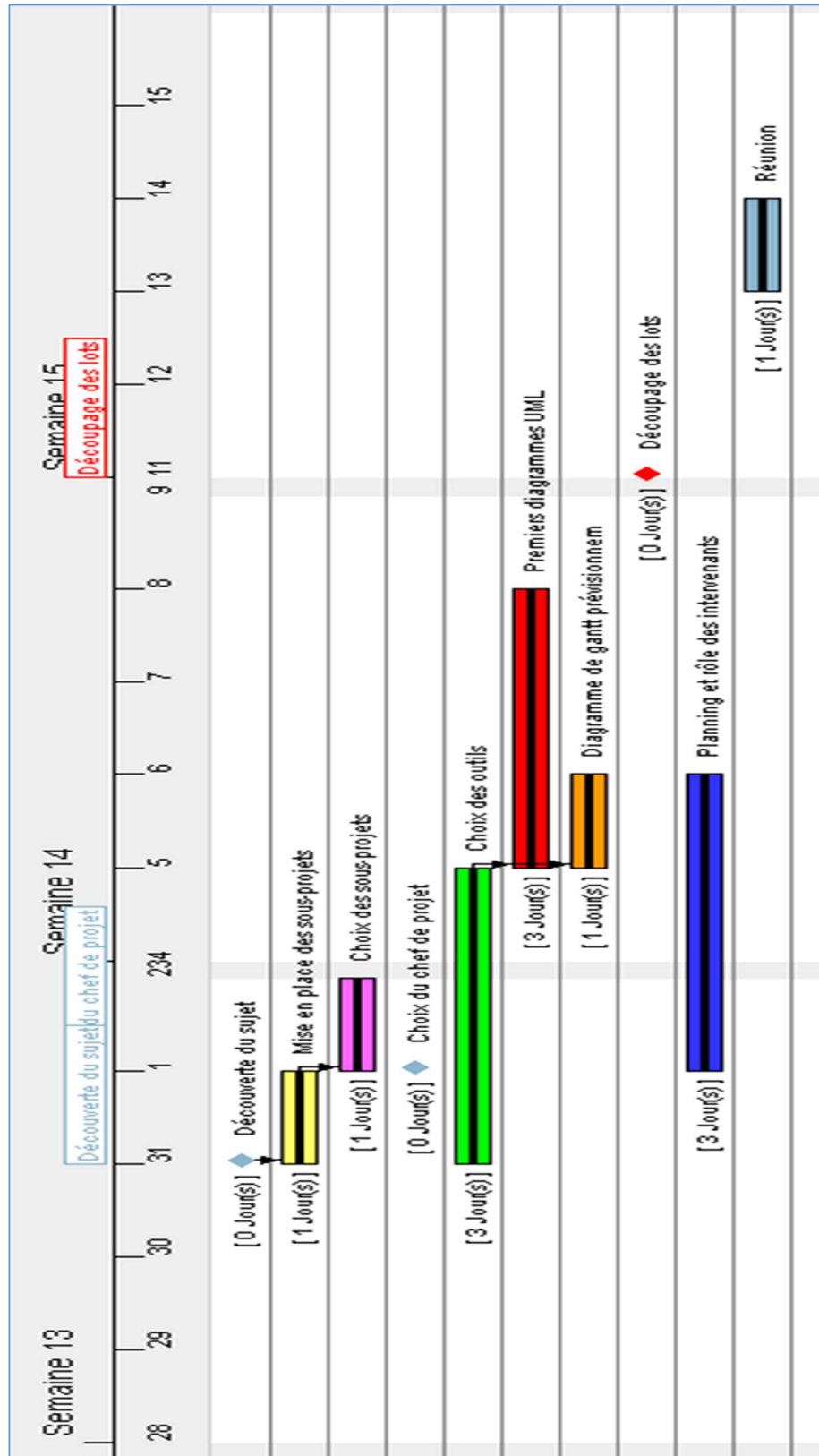
Le diagramme de classe définit les interactions et relations entre les différentes classes.

(Faire « ctrl alt + » pour changer l’affichage en mode paysage idem pour revenir)



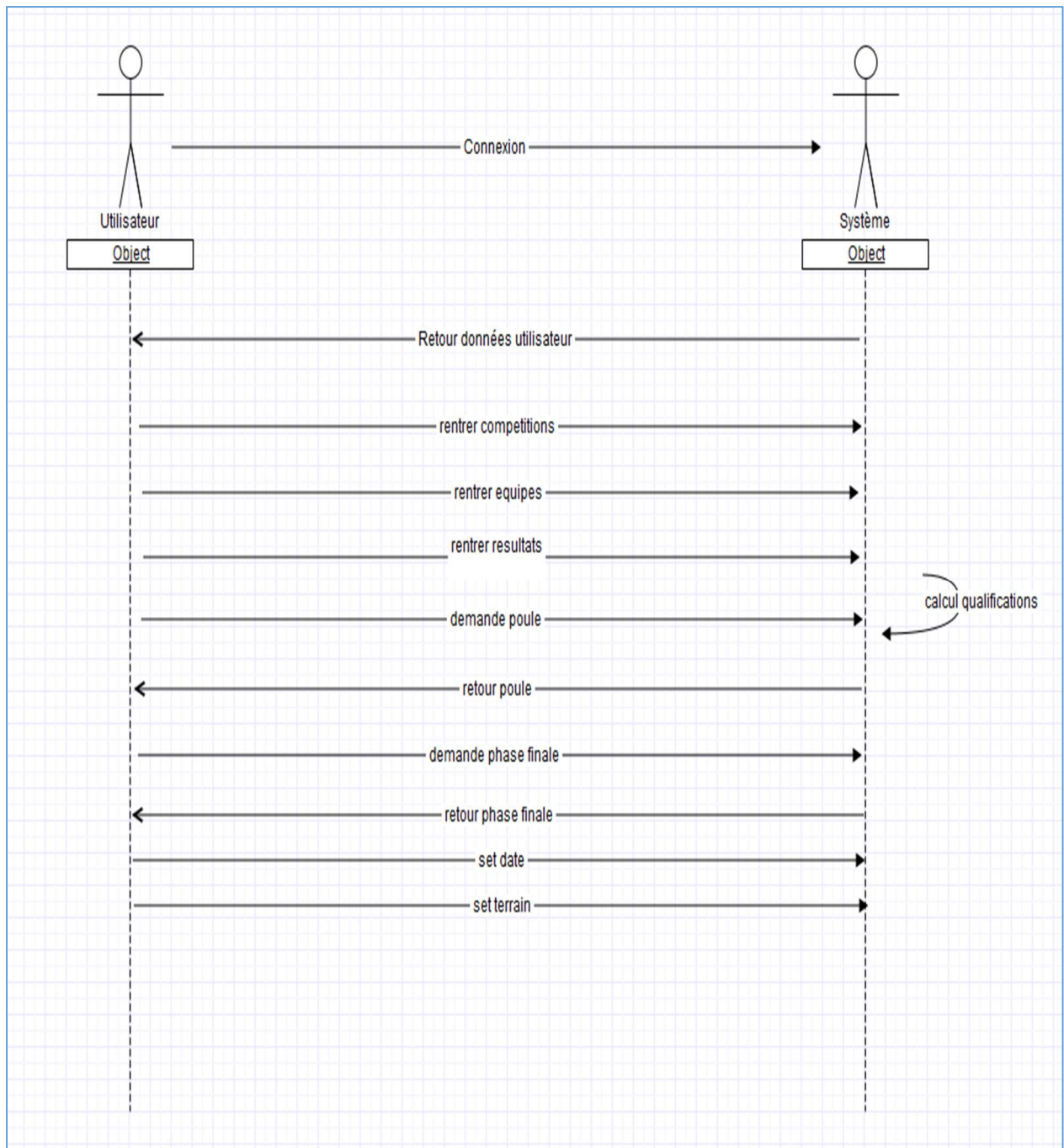
2) Diagramme de Gantt

Le diagramme de Gantt permet de représenter toutes les tâches définies par l'équipe et leur déroulement prévisionnel dans le temps (date de début et date de fin).



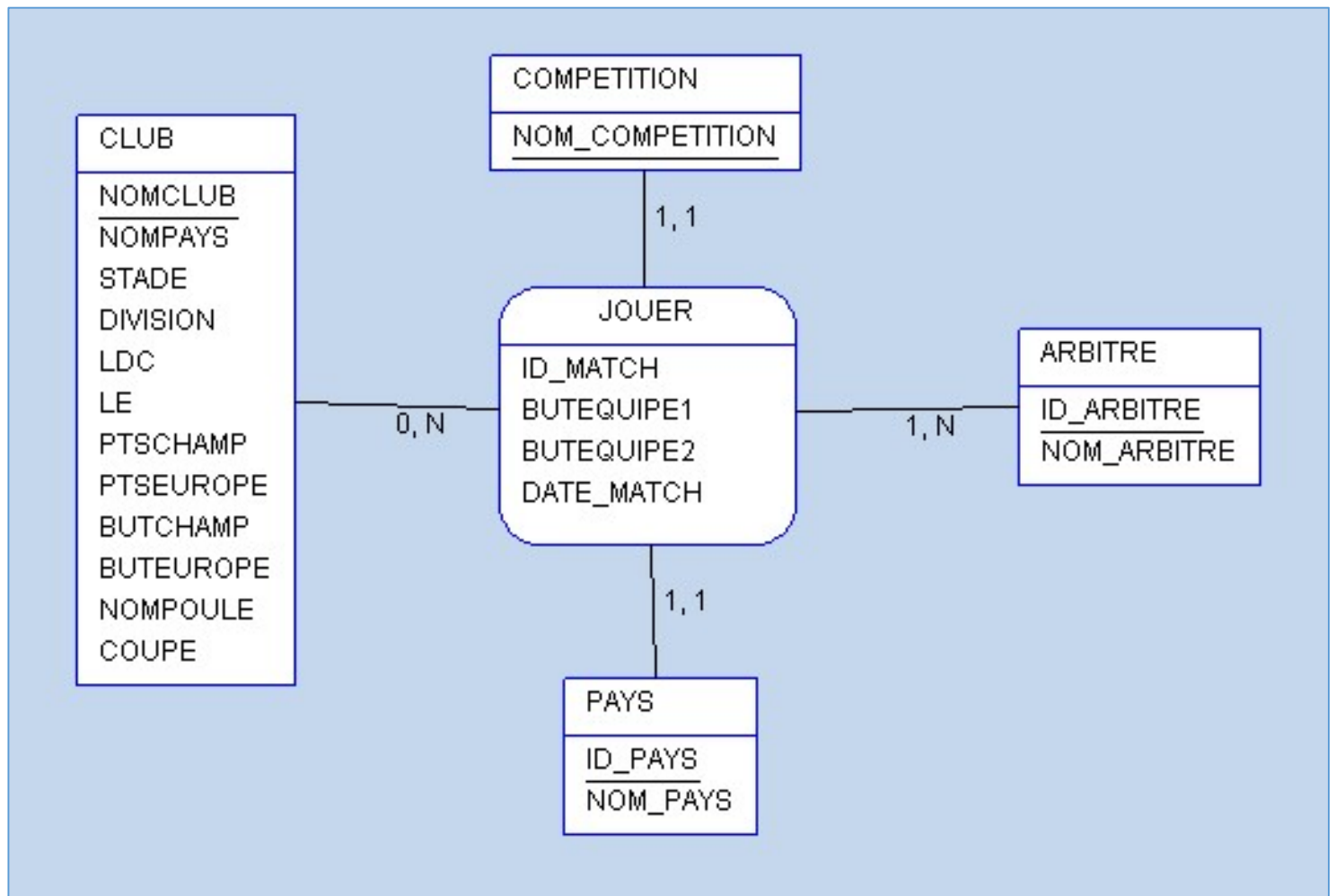
3) Diagramme de séquence

Le diagramme de séquence permet d'illustrer les acteurs de l'application et leurs interactions.



4) Diagramme MCD (modèle conceptuel de données)

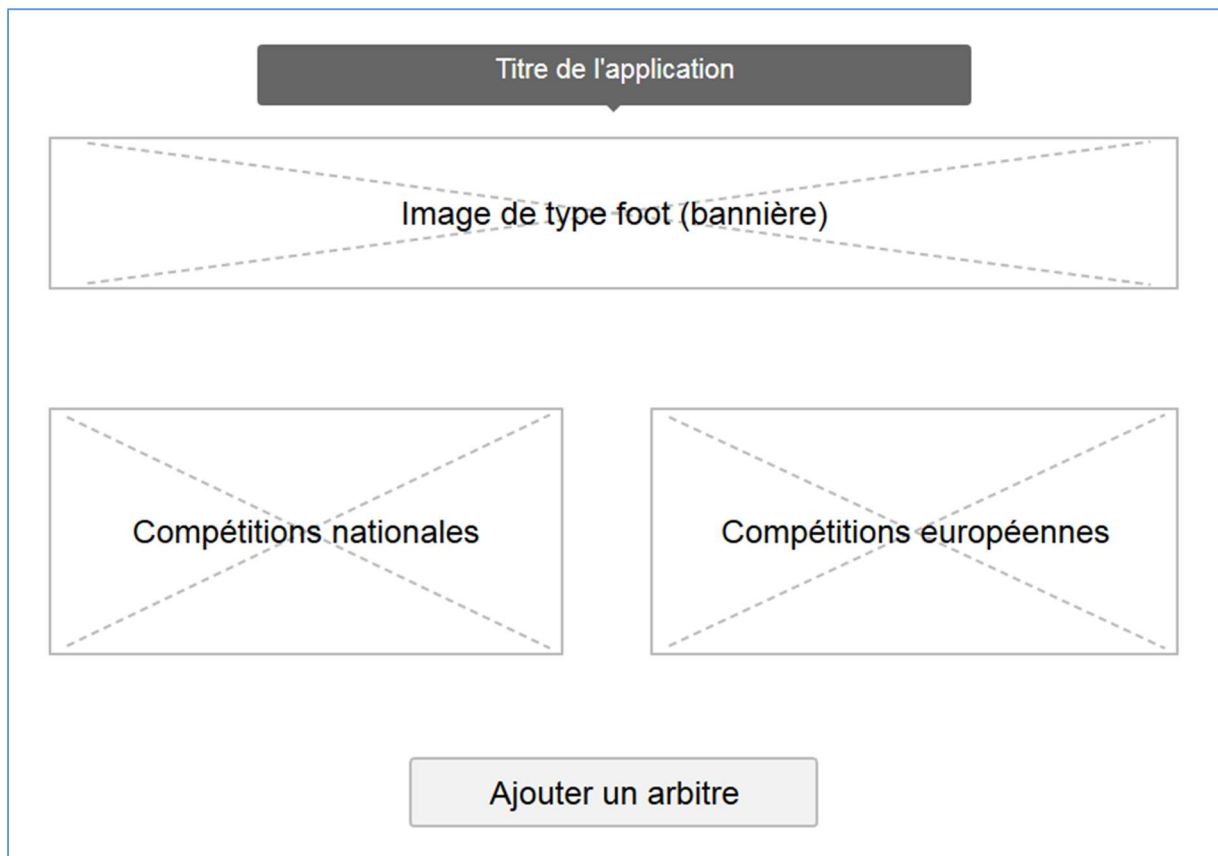
Le diagramme MCD nous permet de définir la structure de notre base de données ainsi que les relations entre les différentes tables qui la composeront et leurs attributs.



5) Maquettes de l'application

Les maquettes de l'application vont servir aux concepteurs de l'interface graphique du programme pour gérer les dispositions des fenêtres et les différentes étapes d'interaction entre l'utilisateur et l'application.

Interface générale



Maquette 1 - Page d'accueil de l'application

Championnat - Actions

Générer classement

Générer calendrier

Championnat - Ajout match

mm / dd / yyyy

Equipe 1

Equipe 2

Arbitre 1

Arbitre 2

Arbitre 3

Arbitre 4

Arbitre remp

Ajouter match

Championnat - Ajout résultat

Match

Score Equipe 1

20

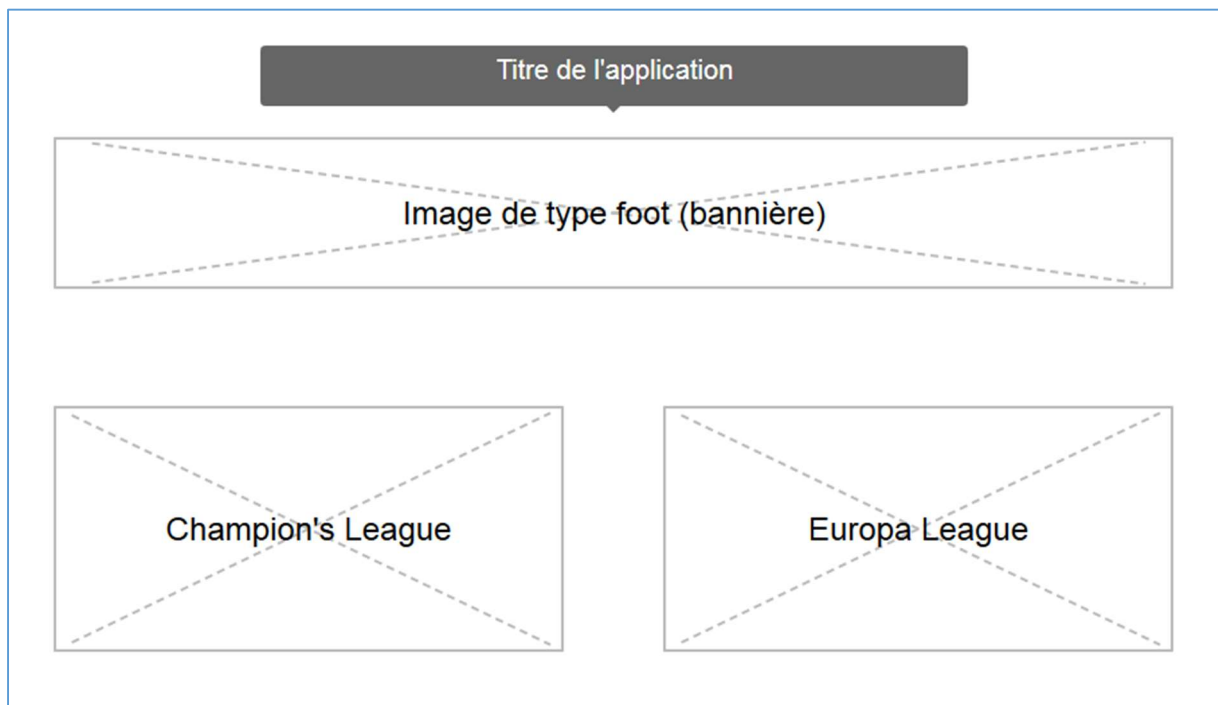
Score Equipe 2

20

Ajouter résultat

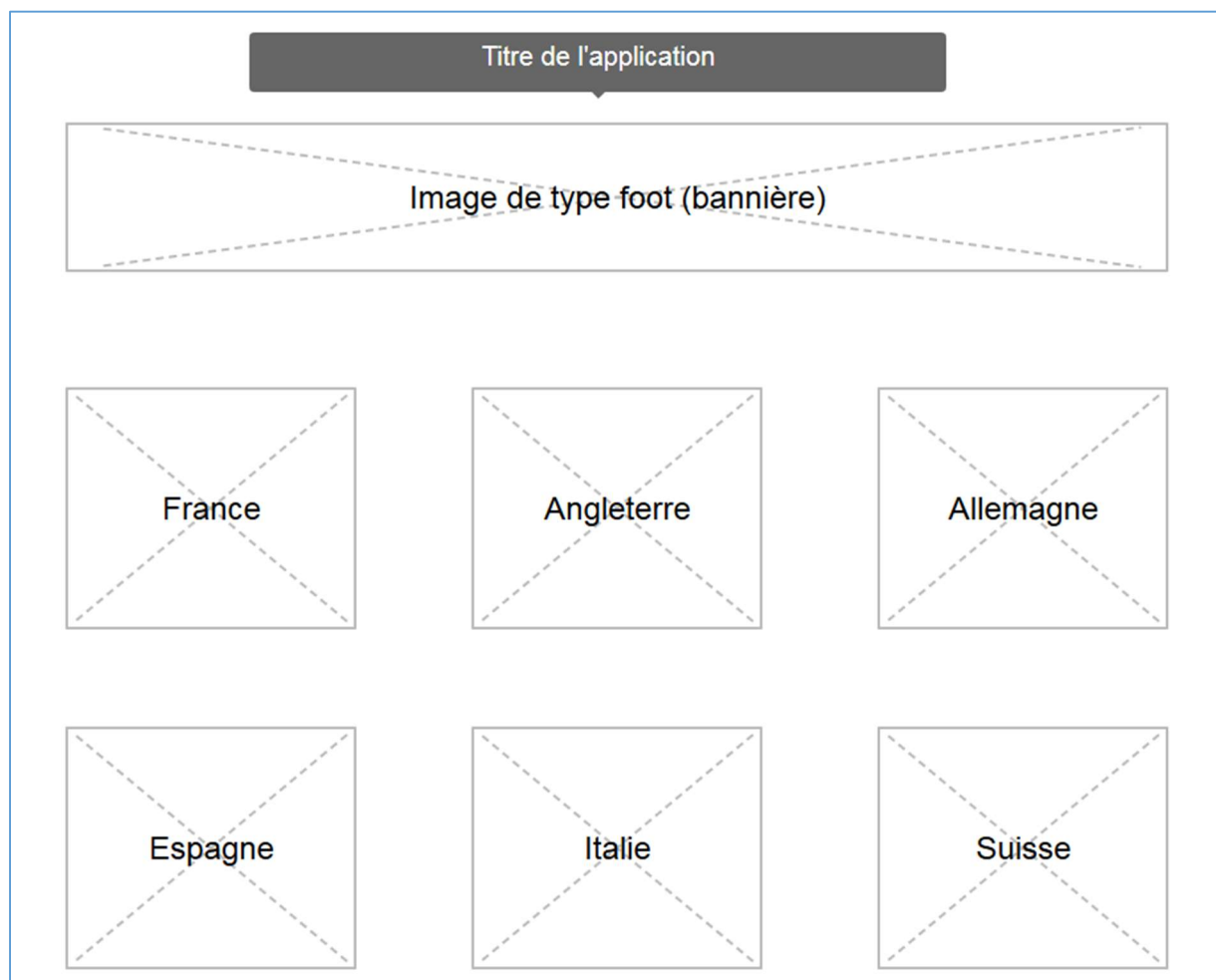
Maquette 2 - Interface pour la gestion des matches

Interface européenne



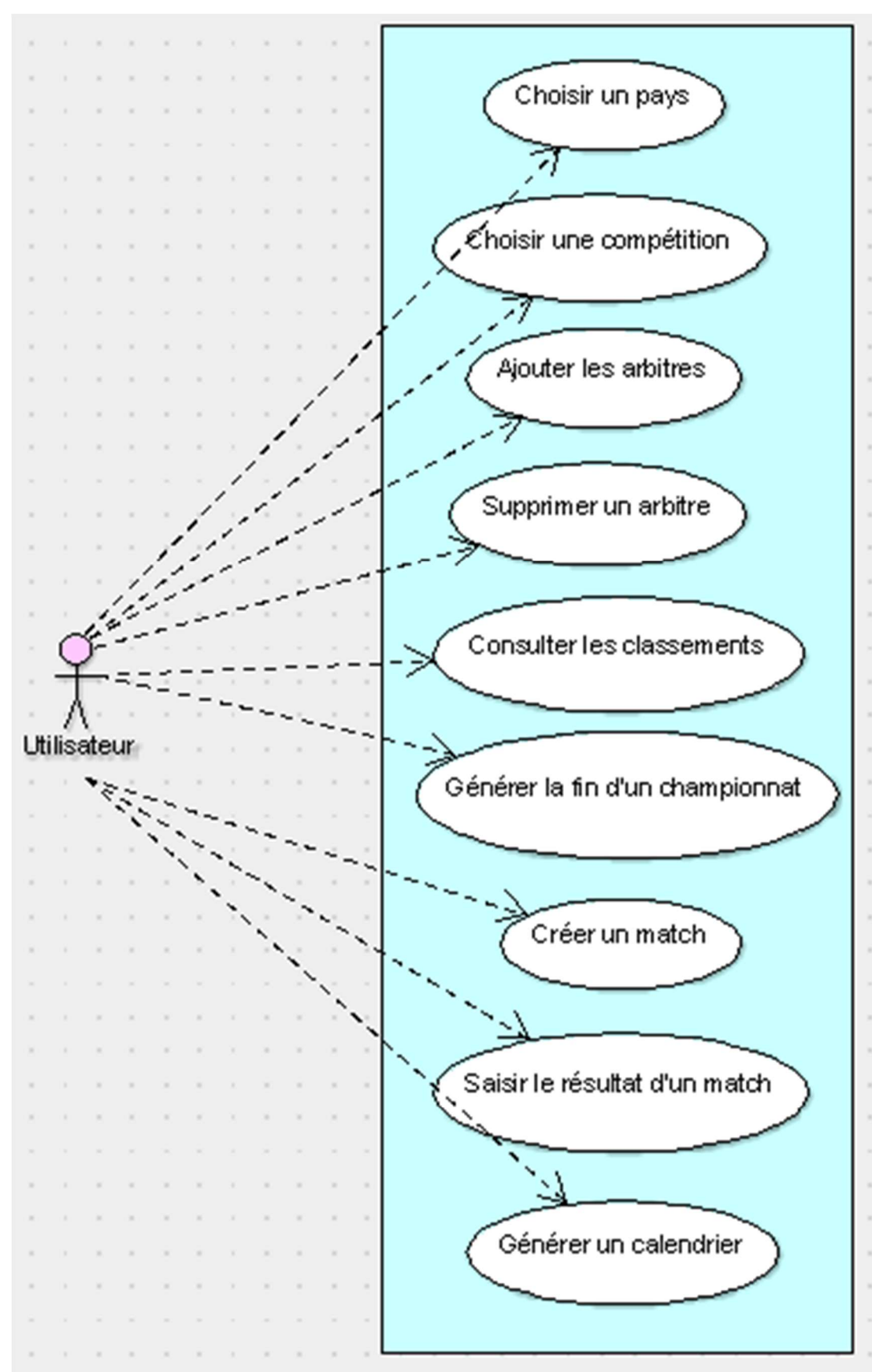
Maquette 3 - Interface européenne

Interface nationale



Maquette 4 - Interface nationale

6) Diagramme de cas d'utilisation



7) Planning et rôle des intervenants sous forme de carte

Les diagrammes de Gantt ne reflètent que l'état d'avancement du projet. Ce qui est certes nécessaire d'un point de vue gestion de projet mais aussi insuffisant car ils n'indiquent pas la répartition des tâches très clairement. Il n'est d'ailleurs pas très interactif et difficile à manipuler en groupe, c'est pourquoi nous avons aussi utilisé Trello. Trello est un outil de gestion de projet en ligne. Il est basé sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement. Cet outil est donc beaucoup plus interactif que le diagramme de Gantt, c'est la raison pour laquelle il a aussi été utilisé afin d'évaluer l'avancement et surtout pour répartir les tâches. De plus, le fait qu'il soit en ligne et gratuit est intéressant car on peut s'y connecter à partir de n'importe quel ordinateur étant donné que l'on dispose d'un compte utilisateur. Une fonctionnalité ingénieuse permet d'envoyer des mails de rappel à ses équipiers pour les avertir d'un retard, de la fin d'une tâche ou d'une nouvelle tâche.

Appli synthèse ☆ Privé

A faire

Ajouter une carte...

En cours

IHM

LB

Diagrammes de séquence

MB

Diagramme de Gantt

K

Merise

K

Ajouter une carte...

En tests

Tests fonctionnels

RF

YM

Tests unitaires

K

LB

RF

Génération du classement

RF

Génération du calendrier

RF

Base de données

K

Ajouter une carte...

Fait

Diagrammes de classe

YM

Diagramme de cas d'utilisation

K

Cahier des charges

MB

Ajouter une carte...

Ajouter une liste...