UEFA Euro 2016

Architecture technique

EPSI Arras i4

Nicolas Janssoone / Pierre Menuge / Anthony Leloire / Armand Szypura
15/04/2016

Avant-propos

Ce document a comme but la définition de l'architecture technique du système d'information dans le cadre du projet UEFA Euro 2016.

Suivi du livrable

Responsable du livrable :	Pierre Menuge
Validation du livrable le :	15/04/2016

Table des matières

1. Objet du projet – Contexte

Le projet a pour but de proposer une application de paris dans le cadre de l'UEFA Euro 2016.

2. Architecture technique

2.1. Les critères de l'architecture

L'architecture technique pour l'UEFA Euro 2016 doit répondre à un certains nombres de critères provenant directement des besoins déterminées pour le client.

Elle doit être capable de :

- gérer la connexion des utilisateurs ;
- enregistrer un nouvel utilisateur ;
- permettre de réaliser un pari sur un match ;
- gérer les matchs ;

2.2. Liste des fonctions

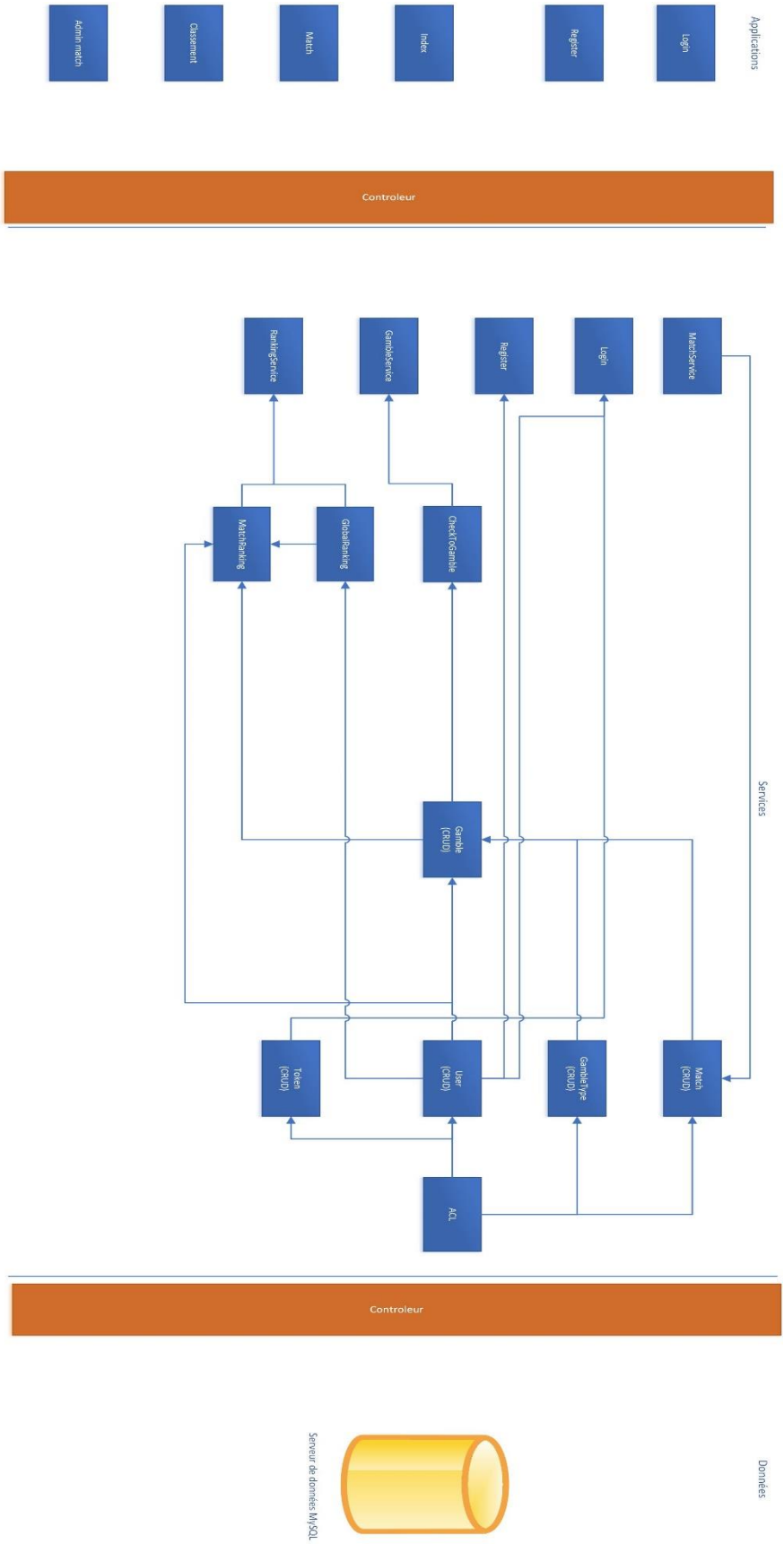
2.2.1. Fonction d'un utilisateur

- Se connecter/déconnecter ;
- S'inscrire ;
- Voir la liste des matchs ;
- Voir le classement des joueurs ;
- Accéder à un match ;
- Parier sur un match.

2.2.2. Fonction d'un administrateur

- Se connecter/déconnecter ;
- Voir la liste des matchs ;
- Voir le classement des joueurs ;
- Accéder à un match ;
- Modifier le score d'un match.

2.3. Schéma de l'architecture technique


Applications
Admin match
Classement
Match
Index
Register
Login
Controleur
Services
RankingService
GambleService
Register
Login
MatchService
MatchRanking
GlobalRanking
CheckToGamble
Gamble (CRUD)
Token (CRUD)
User (CRUD)
GambleType (CRUD)
Match (CRUD)
ACL
Controleur
Données
Serveur de données MySQL

3. Hiérarchie d'appel des services

- Connexion
  - ConnectUser (Login, Password)
  - GetIdUser(élément de l'utilisateur)
  - IsUserIdMatchWithPassword(Login, Password)
  - GenerateToken (Login)

- Inscription
  - RegisterUser (Name, LastName, BirthDate, Email,Login, Password)
  - IsLoginExist(Login)
  - IsEmailExist(Email)

- Liste des Matches
  - GetMatches()
  - IsTokenExist(UserToken)

- Matchs
  - GetMatch(MatchId)
  - IsTokenExist(UserToken)

- Gamble
  - SetGambleMatch(MatchId, Gamble, userToken)
  - IsTokenExist(UserToken)
  - IsMatchExist(MatchId)
  - IsMatchEnd(MatchId)

- RankingGlobal
  - GetGlobalRanking(UserToken)
  - IsTokenExist(UserToken)

- RankingMatch
  - GetMatchRanking(UserToken, MatchId, userToken)
  - IsTokenExist(UserToken)
  - GetMatch(MatchId)

- CreateMatch
  - CreateMatch(FirstEquipe, SecondEquipe, Date, userToken)
  - IsTokenAdmin(UserToken)
  - IsTokenExist(UserToken)
  - IsMatchAlreadySet(FirstEquipe, SecondEquipe, Date)

- CloseMatch
  - CloseMatch(MatchId, score, userToken)
  - IsTokenAdmin(UserToken)
  - IsTokenExist(UserToken)
  - SetScore(MatchId, score)

- UpdateMatch
  - UpdateMatch(MatchId, FirstEquipe, SecondEquipe, Date, userToken)
  - IsTokenAdmin(UserToken)
  - IsTokenExist(UserToken)
  - IsMatchExist(MatchId)

- DeleteMatch
  - DelteMatch(MatchId, userToken)
  - IsTokenAdmin(UserToken)
  - IsTokenExist(UserToken)
  - IsMatchExist(MatchId)

- Déconnexion
  - DeleteToken()
  - IsTokenExist(UserToken)

4. Technologies utilisées

Le choix des technologies, conformément aux critères de l'architecture technique définis plus haut, est fait en considérant les compétences techniques de chacun. Elles serviront à la bonne conception de l'application pour l'UEFA Euro 2016.

- PHP 5
- MySQL
- Web Service REST en JSON
- MySQL Workbench
- Visio

5. Description du Code

5.1. Historique de la solution

- Création d'un MDC ;
- Création d'une base de données basée sur MySQL ;
- Création des vues avec HTML et CSS ;
- Création du front office ;
- Création des Web services avec PHP ;
- Création du back office.

5.2. Mécanismes d'identification et d'authentification

Les utilisateurs et les administrateurs doivent s'authentifier grâce à un login et un mot de passe sécurisé lorsque que la demande d'authentification est réaliser avec les services d'authentification qui attaque le serveur de base de données.

6. Normes et standards

L'application doit répondre aux normes et standards en vigueur au sein de l'Union Européenne.

Si certaines directives n'ont pu être mises en œuvre, cela est dû au fait que l'application est encore en court de réalisation et n'est pas encore disponible pour le public.

7. Cas d'utilisations

1. Connection d'un utilisateur (prérequis : 3)
    - 1.1 l'utilisateur entre ses identifiants
        - 1.1.1 l'utilisateur n'a pas de compte -> 3.
        - 1.1.2 les identifiants sont incorrects -> 1.1
        - 1.1.3 les identifiants sont corrects
            - 1.1.3.1 l'utilisateur est administrateur -> 1.3
            - 1.1.3.2 l'utilisateur est joueur -> 1.2
    - 1.2 le joueur est connecté -> 2
    - 1.3 l'administrateur est connecté -> 4

2. Pari sur un match (prérequis : 1.2)
    - 2.1 le joueur consulte la liste des matchs
    - 2.2 le joueur choisit un match
    - 2.3 le joueur veut ajouter un pari
        - 2.3.1 le match est clôturé -> ERREUR
        - 2.3.2 l'utilisateur est administrateur -> ERREUR
    - 2.4 le joueur ajoute le pari

3. Enregistrement d'un utilisateur
    - 3.1 l'utilisateur accède au formulaire d'inscription
        - 3.1.1 l'utilisateur remplit le formulaire
            - 3.1.1.1 le formulaire est incorrect -> 3.1
    - 3.2 le compte de l'utilisateur est créé -> 1.2
        - 3.2.1 l'utilisateur reçoit un mail pour confirmer sont inscription

4. Gestion des matchs (prérequis: 1.3)
    - 4.1 l'administrateur consulte la liste des matchs
    - 4.2 l'administrateur veut créer un match -> 5
    - 4.3 l'administrateur accède à la fiche d'un match

4.3.1 l'administrateur veut modifier un match -> 6

4.3.2 l'administrateur veut clôturer un match -> 7

4.3.3 l'administrateur veut supprimer un match -> 8

5. Création d'un match (prérequis 4)

5.1 l'administrateur accède au formulaire de création de match

5.1.1 le formulaire est erroné -> 5.1

5.2 le match est créé -> 4.3

6. Modification d'un match (prérequis 5)

6.1 l'administrateur accède au formulaire de modification

6.1.1 le match est clôturé -> ERREUR

6.2 l'administrateur remplit le formulaire

6.3 l'administrateur valide le formulaire

6.3.1 le formulaire est erroné -> 6.2

6.4 le match est modifié

7. Clôture d'un match (prérequis 5)

7.1 l'administrateur valide la clôture du match

7.1.1 la date du match n'est pas dépassée (inférieur ou égal) -> ERREUR

7.2 les paris sont impactés

7.2.1 les joueurs ayant parié gagnent ou perdent des points selon leur pari

8. Suppression d'un match (prérequis 5)

8.1 l'administrateur accède au formulaire de suppression

8.1.1 le match est clôturé -> ERREUR

8.2 l'administrateur renseigne une raison de suppression (liste de choix ?)

8.2.1 la raison est vide -> 8.1

8.3 l'administrateur valide la suppression du match

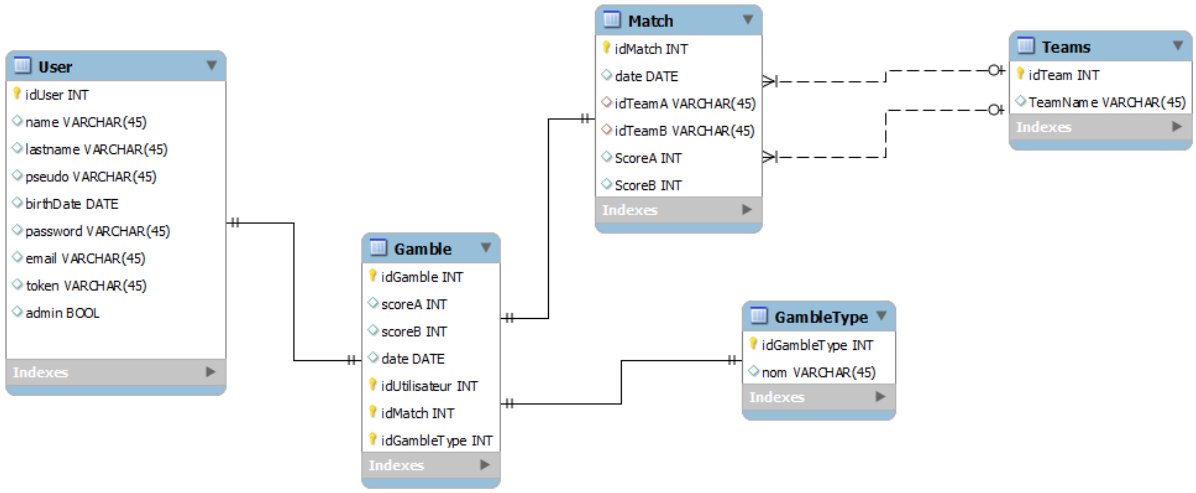
8.4 les paris sont annulés

8.5 les joueurs ayant parié sont avertis (notification ? mail ?)

8.6 le match est supprimé

8. Modèle conceptuel de données et base de données

8.1. Modèle conceptuel de données



8.2. Base de données

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

USE euro2016;


-- -----------------------------------------------------
-- Table `euro2016`.`User`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `User` (
  `idUser` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NULL,
  `lastname` VARCHAR(45) NULL,
  `pseudo` VARCHAR(45) NULL,
  `birthDate` DATE NULL,
  `password` VARCHAR(45) NULL,
  `email` VARCHAR(45) NULL,
  `token` VARCHAR(45) NULL,
  `admin` TINYINT(1) NULL,
  PRIMARY KEY (`idUser`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `euro2016`.`Teams`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Teams` (
  `idTeam` INT NOT NULL AUTO_INCREMENT,
  `TeamName` VARCHAR(45) NULL,
  PRIMARY KEY (`idTeam`))
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `euro2016`.`Match`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Match` (
  `idMatch` INT NOT NULL AUTO_INCREMENT,
  `date` DATE NULL,
  `idTeamA` VARCHAR(45) NULL,
  `idTeamB` VARCHAR(45) NULL,
  `ScoreA` INT NULL,
  `ScoreB` INT NULL,
  PRIMARY KEY (`idMatch`),
  INDEX `fk_Match_Teams1_idx` (`idTeamA` ASC),
  INDEX `fk_Match_Teams2_idx` (`idTeamB` ASC),
  CONSTRAINT `fk_Match_Teams1`
    FOREIGN KEY (`idTeamA`)
    REFERENCES `euro2016`.`Teams` (`TeamName`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Match_Teams2`
    FOREIGN KEY (`idTeamB`)
    REFERENCES `euro2016`.`Teams` (`TeamName`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `euro2016`.`GambleType`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `GambleType` (
  `idGambleType` INT NOT NULL AUTO_INCREMENT,
  `nom` VARCHAR(45) NULL,
  PRIMARY KEY (`idGambleType`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `euro2016`.`Gamble`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `euro2016`.`Gamble` (
  `idGamble` INT NOT NULL AUTO_INCREMENT,
  `scoreA` INT NULL,
  `scoreB` INT NULL,
  `date` DATE NULL,
  `idUtilisateur` INT NOT NULL,
  `idMatch` INT NOT NULL,
  `idGambleType` INT NOT NULL,
  PRIMARY KEY (`idGamble`, `idUtilisateur`, `idMatch`, `idGambleType`),
  INDEX `fk_Gamble_Utilisateur_idx` (`idUtilisateur` ASC),
  INDEX `fk_Gamble_Match1_idx` (`idMatch` ASC),
  INDEX `fk_Gamble_GambleType1_idx` (`idGambleType` ASC),
  CONSTRAINT `fk_Gamble_User`
    FOREIGN KEY (`idUtilisateur`)
    REFERENCES `euro2016`.`User` (`idUser`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Gamble_Match`
    FOREIGN KEY (`idMatch`)
    REFERENCES `euro2016`.`Match` (`idMatch`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Gamble_GambleType`
    FOREIGN KEY (`idGambleType`)
    REFERENCES `euro2016`.`GambleType` (`idGambleType`)
    ON DELETE NO ACTION
```

```
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```