

# Rapport d'analyse

Grand Prix de Tennis de Lyon

**UNIVERSITE CLAUDE BERNARD LYON 1**

HENRIONNET ANTOINE - LECUTIEZ SIMON - MATHEZ CLAIRE

CPOA – Véronique Deslandres

# Table des matières

I.	PRESENTATION DE L'APPLICATION .....	2
A.	RESUME DU SUJET.....	2
B.	NOTRE ANALYSE DU SUJET .....	2
II.	PACKAGE JAVA .....	3
A.	DIAGRAMME DE CAS D'UTILISATION .....	3
B.	DIAGRAMME DE SEQUENCE.....	4
C.	DIAGRAMME D'ACTIVITES .....	6
D.	DIAGRAMME DE CLASSES JAVA .....	7
III.	PACKAGE WEB.....	8
A.	DIAGRAMME DE CAS D'UTILISATION .....	8
B.	DIAGRAMME DE SEQUENCE.....	9
C.	DIAGRAMME D'ACTIVITES .....	11
D.	DIAGRAMME DE CLASSE WEB .....	12
IV.	IHM APPLICATION WEB .....	13
V.	IHM APPLICATION JAVA .....	16
VI.	CONCEPTION.....	17
A.	DIAGRAMME DE CLASSES DE CONCEPTION .....	17
B.	SCRIPT SQL.....	18
C.	SQUELETTE JAVA.....	20
D.	SCHEMA DE LA BD .....	21

## I. Présentation de l'application

### A. Résumé du sujet

Le grand prix de tennis Lyon est un évènement sportif qui nécessite la mise en place d'une application java, utilisable par les organisateurs de l'évènement et les joueurs, ainsi qu'un site web accessible à tous avec la possibilité pour un organisateur de se connecter et d'obtenir des droits supplémentaires. Pour ce projet nous devons mettre en place deux modules liés entre eux : un module web et un module java.

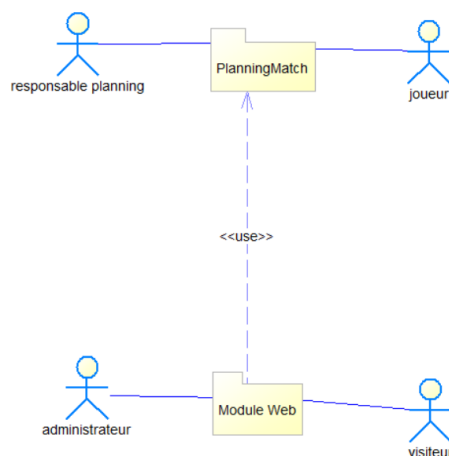
L'application java, permet à un responsable planning de gérer les matchs : joueurs, arbitres, juges de ligne, ramasseurs de balles... Les joueurs peuvent aussi utiliser une partie de cette application afin de réserver des terrains d'entraînement.

Par ailleurs nous devons développer un module web, au choix parmi l'un des sujets suivants : billetterie en ligne, gestion des VIPs, gestion de l'hébergement. Nous avons choisi de traiter la billetterie. Il y aura un site web ouvert au public pour qu'il puisse acheter des billets pour un match. Un administrateur doit pouvoir modifier les prix des billets, appliquer des codes promotionnel.

### B. Notre analyse du sujet

Le module java concerne les responsables des plannings et les joueurs. Les responsables doivent pouvoir gérer les matchs, les joueurs, les horaires, les terrains ainsi que les arbitres tout en respectant certaines contraintes que nous définirons avec les différents diagrammes. Seul un responsable de l'organisation du tournoi peut modifier un match, l'application doit prendre en compte ces contraintes. De plus, l'application doit tenir compte de différents paramètres tel que la nationalité d'un joueur ou encore le gagnant d'un match. Les joueurs peuvent réserver un terrain pour s'entraîner, et nous avons jugé utile que les joueurs puissent consulter les plannings des matchs, afin d'avoir une idée de la disponibilité des terrains. De plus, afin d'aider les joueurs n'étant pas de la région, ils pourront afficher une carte basique, afin de connaître l'emplacement des différents terrains.

Le module web concerne le public et les organisateurs. Le public peut consulter les différents billets, et les acheter en quantité qu'il souhaite. Les organisateurs peuvent, en s'identifiant, modifier le prix des billets, les codes promotionnels, et mettre à jour l'image du planning. Nous avons choisi de ne pas implémenter la fonctionnalité de consultation directe du planning par le public (car nous pensons ne pas avoir le temps) c'est pourquoi nous utiliserons une image que l'administrateur éditera lorsqu'il le souhaite. Le module web utilise la base de données du module java afin de récupérer les horaires des matchs.



## II. Package Java

### a. Diagramme de cas d'utilisation



Pour tous les utilisateurs : Le plan est une image fixe, reprenant les emplacements des terrains. Il s'agit plus d'un schéma que d'une photo aérienne, afin que tous les joueurs puissent se repérer.

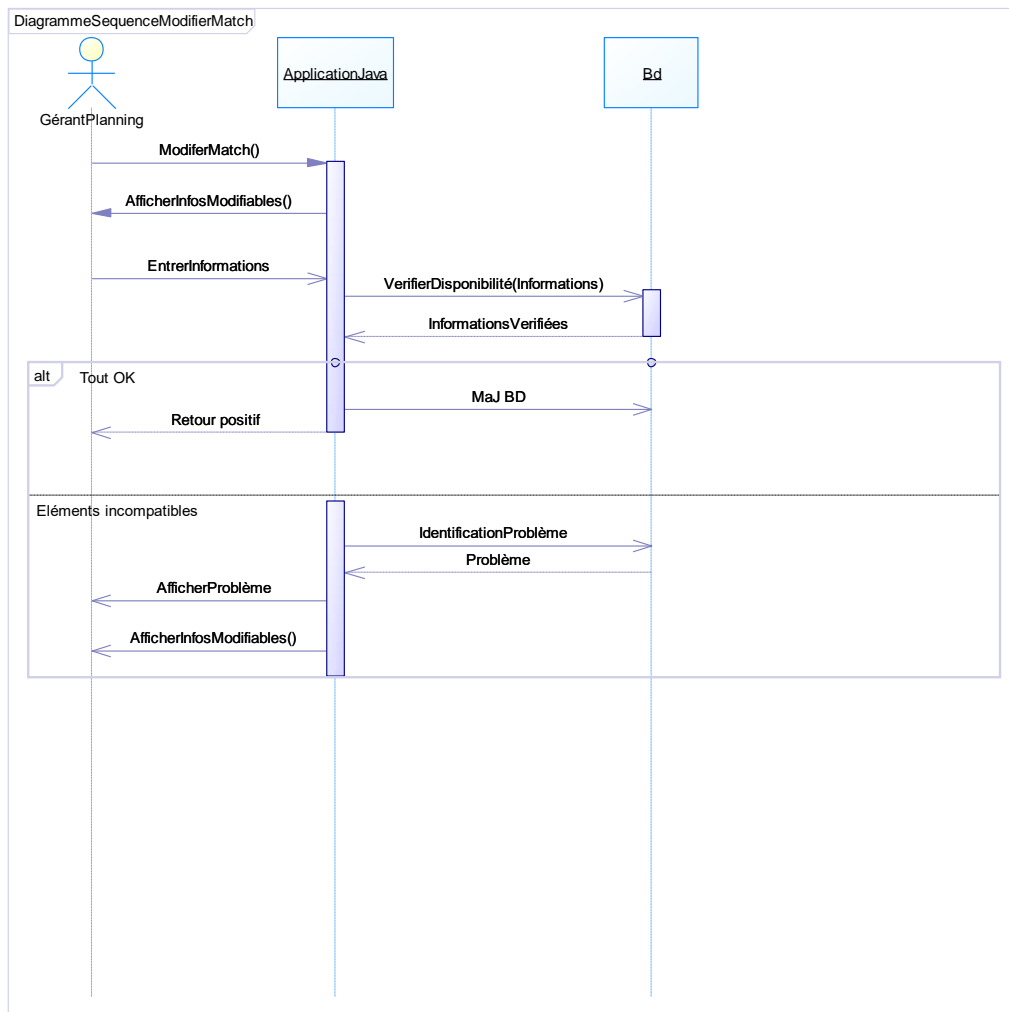
Pour le responsable planning : Le staff d'un match correspond aux arbitres de ligne, de touche et aux ramasseurs de balles.

Les horaires et les terrains peuvent être modifiés à tout moment.

Modifier les terrains signifie définir où aura lieu le match.

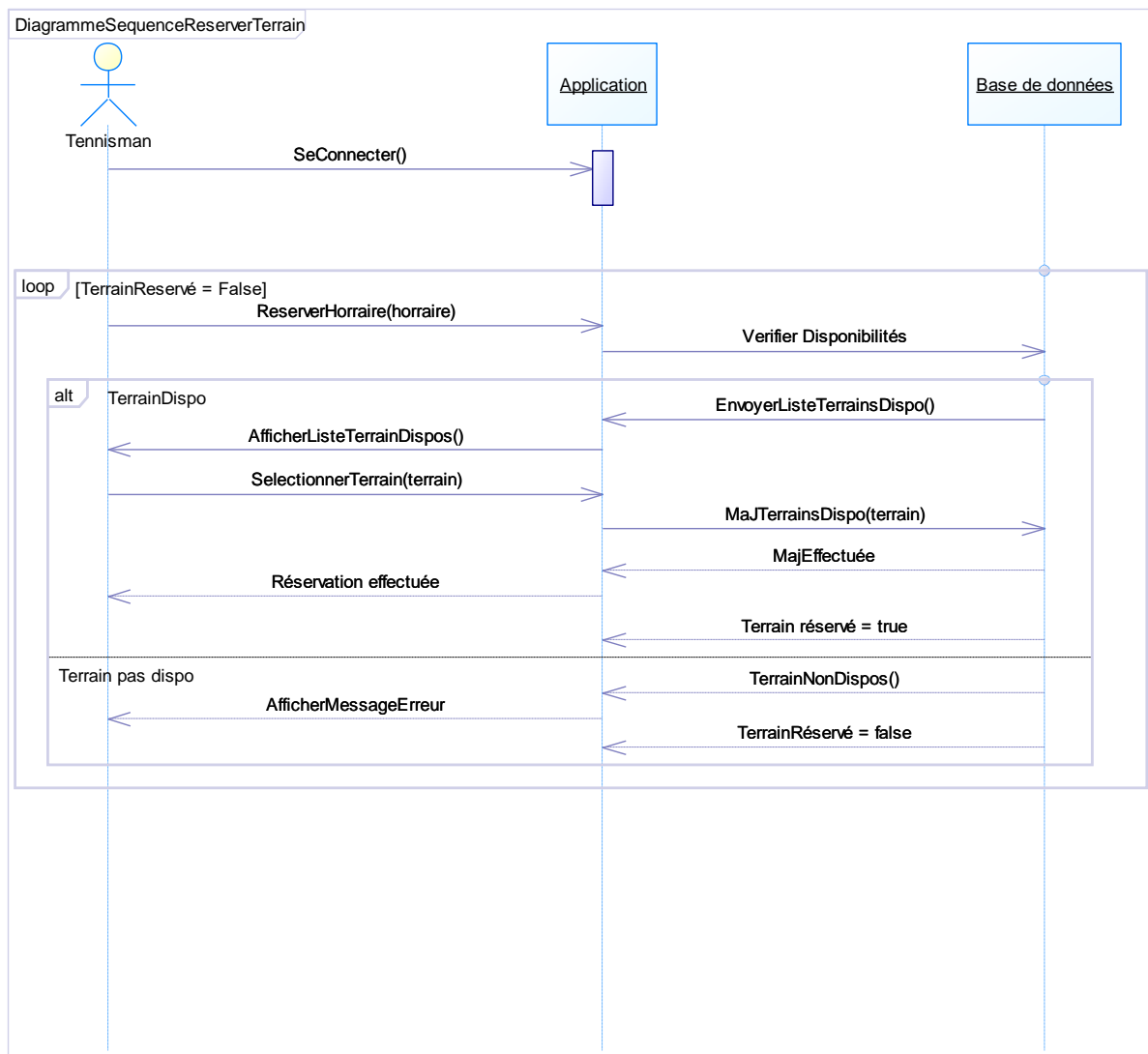
## b. Diagramme de séquence

Diagramme de séquence concernant la modification pour un match :



Dans ce cas on peut rencontrer les problèmes suivants : le terrain est déjà utilisé, l'arbitre ne peut plus arbitrer de matchs, il y a un problème de nationalité avec un des joueurs, l'horaire n'est pas bon, le joueur joue trop de matchs en un jour... On laisse donc la possibilité au responsable planning de remodifier les informations qui sont incohérentes.

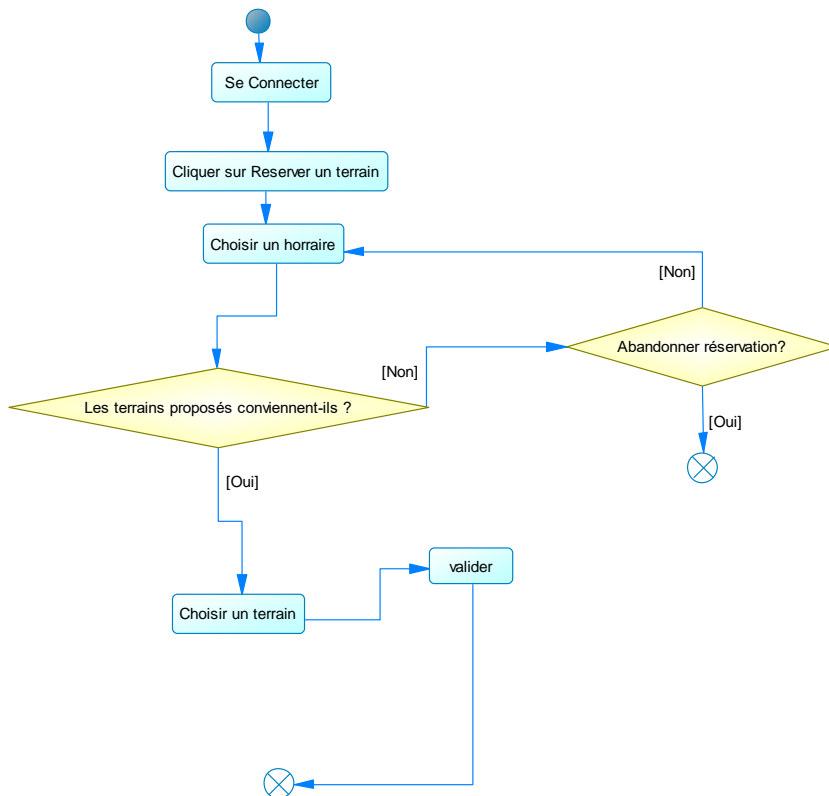
Diagramme de séquence concernant la réservation d'un terrain :



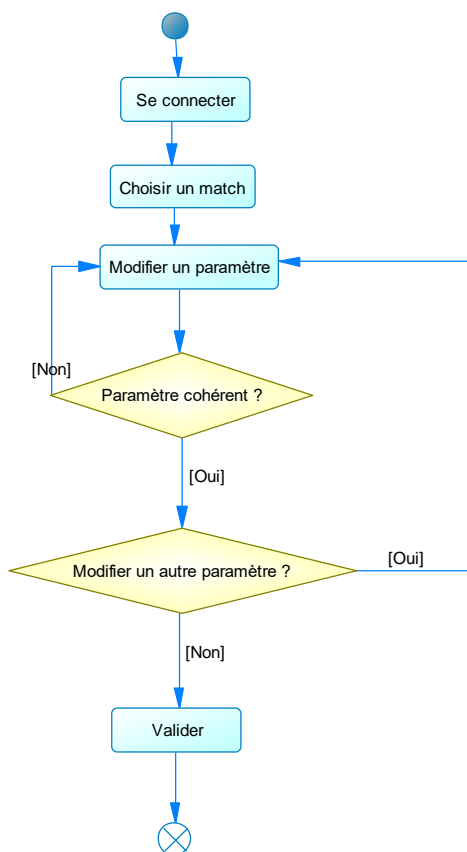
L'utilisateur a la possibilité de réserver un terrain selon des horaires spécifiques. Cependant si aucun terrain n'est disponible à l'heure demandée, il a la possibilité de la changer.

### c. Diagramme d'activités

Ci-dessous le diagramme d'activité pour réserver un terrain :

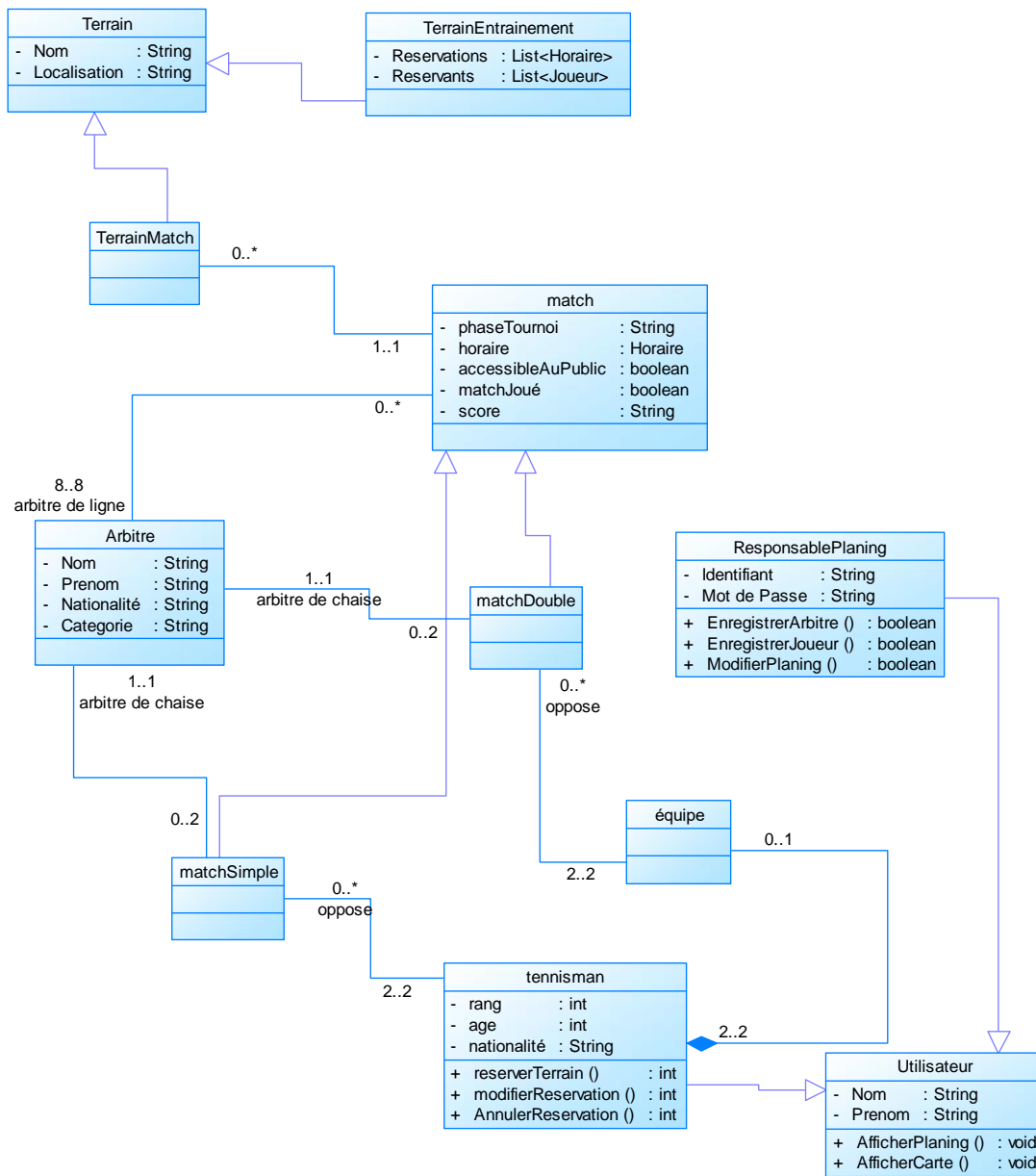


Ci-dessous le diagramme d'activité pour modifier un match :



#### d. Diagramme de classes Java

Ci-dessous le diagramme de classes de l'application Java :



Par soucis de compréhension dans le code, nous avons décidé de faire plusieurs classes pour les matchs et les terrains. Ainsi nous séparons les terrains d'entraînement, qui peuvent être réservés par les joueurs, des terrains de match, bien que les terrains d'entraînement puissent servir de terrain de match.

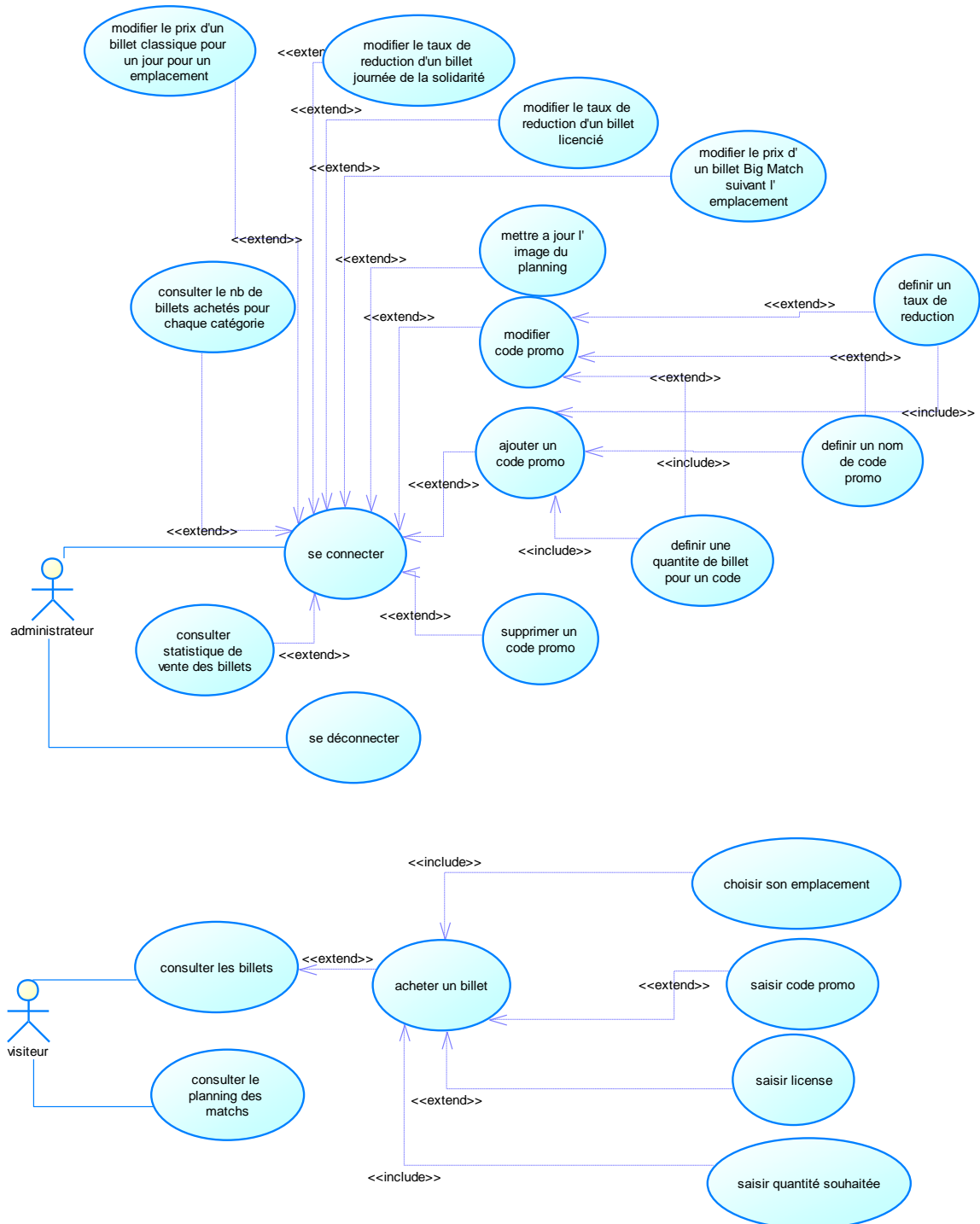
Les matchs sont aussi séparés, avec d'un côté les matchs simples et d'un autre les matchs doubles.

Les classes Utilisateur, tennisman et ResponsablePlaning étant les seules à interagir avec le logiciel, ce sont donc les seules disposantes de fonctions, fonctions qui sont définies dans le cas d'utilisation.



### III. Package Web

#### a. Diagramme de cas d'utilisation

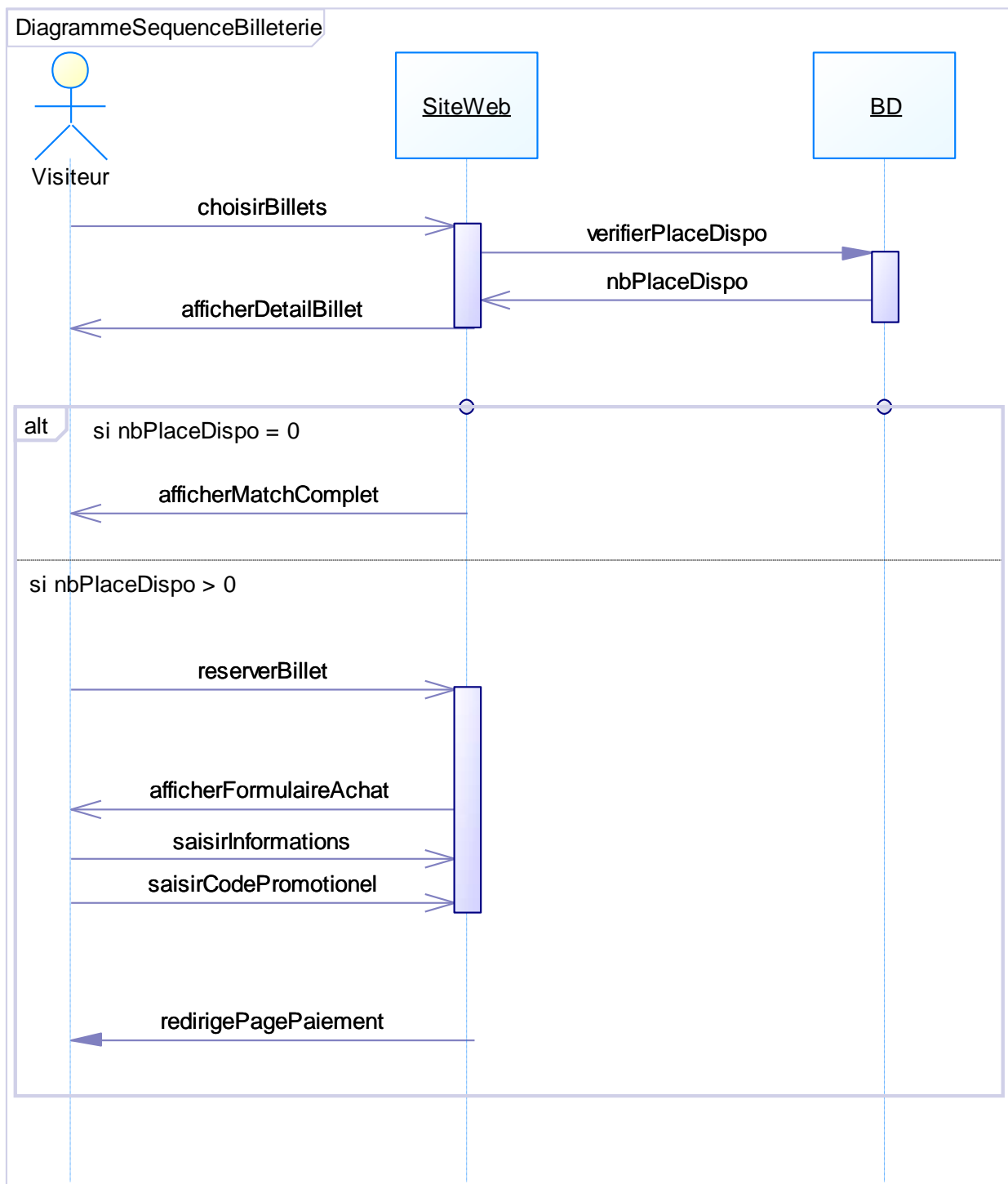


Nous n'avons pas choisi de définir une limite d'emplacement car d'après le sujet il n'y a pas de limitations de places pour un emplacement.

Par ailleurs nous avons choisi de détailler précisément l'éventail d'actions réalisable à chaque fois pour chaque type de billet afin de nous faciliter l'implémentation par la suite.

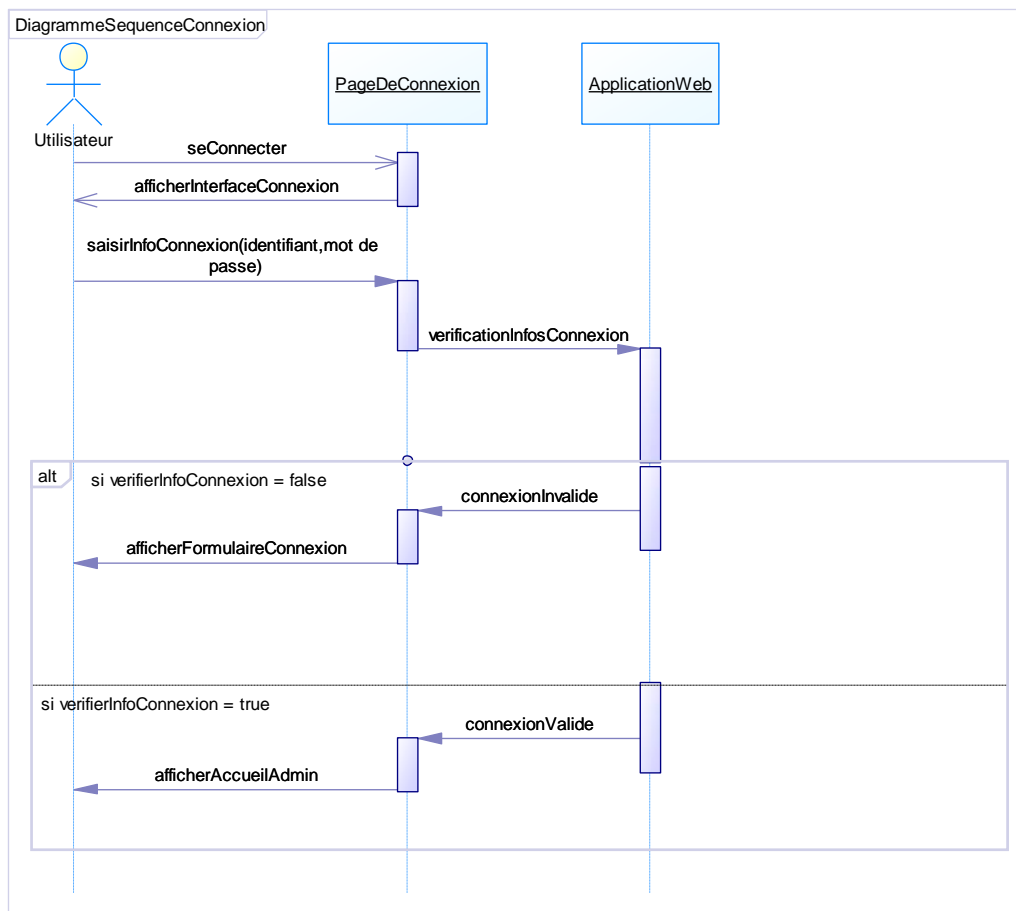
## b. Diagramme de séquence

Le diagramme de séquence concernant la billetterie en ligne :



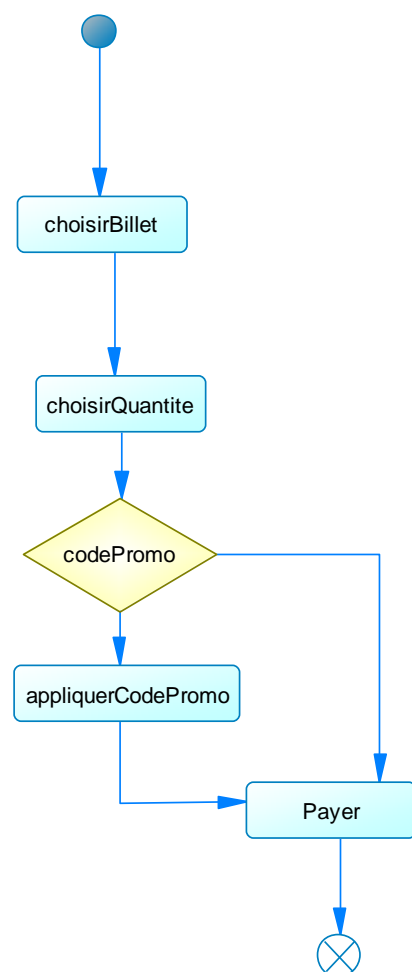
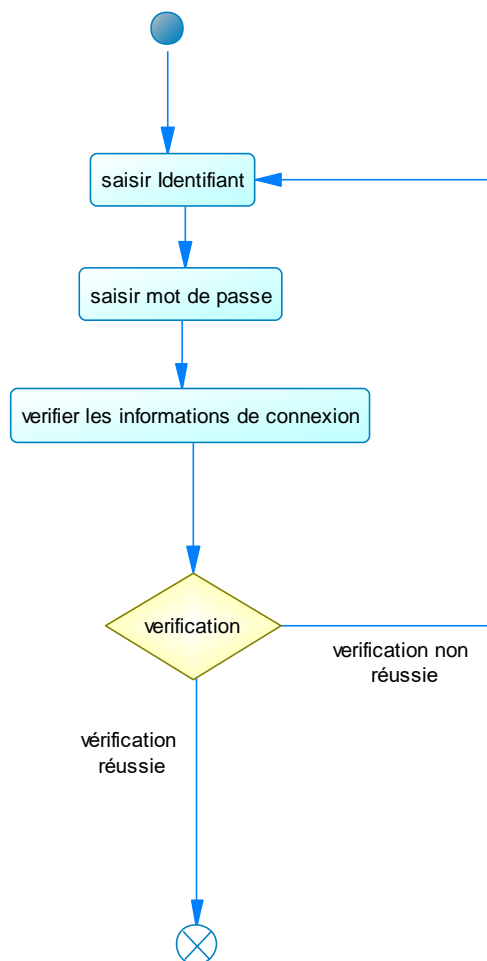
On suppose que le nombre de places disponibles avant tout achat est défini sous la forme d'une constante dans le module Web et est le même pour tous les matchs. On ne gèrera pas une limite de nombre de places par emplacement. C'est pourquoi si le nombre de place est à 0 on indiquera au visiteur que le match est complet.

Le diagramme de séquence pour la connexion à l'application web en tant qu'administrateur :



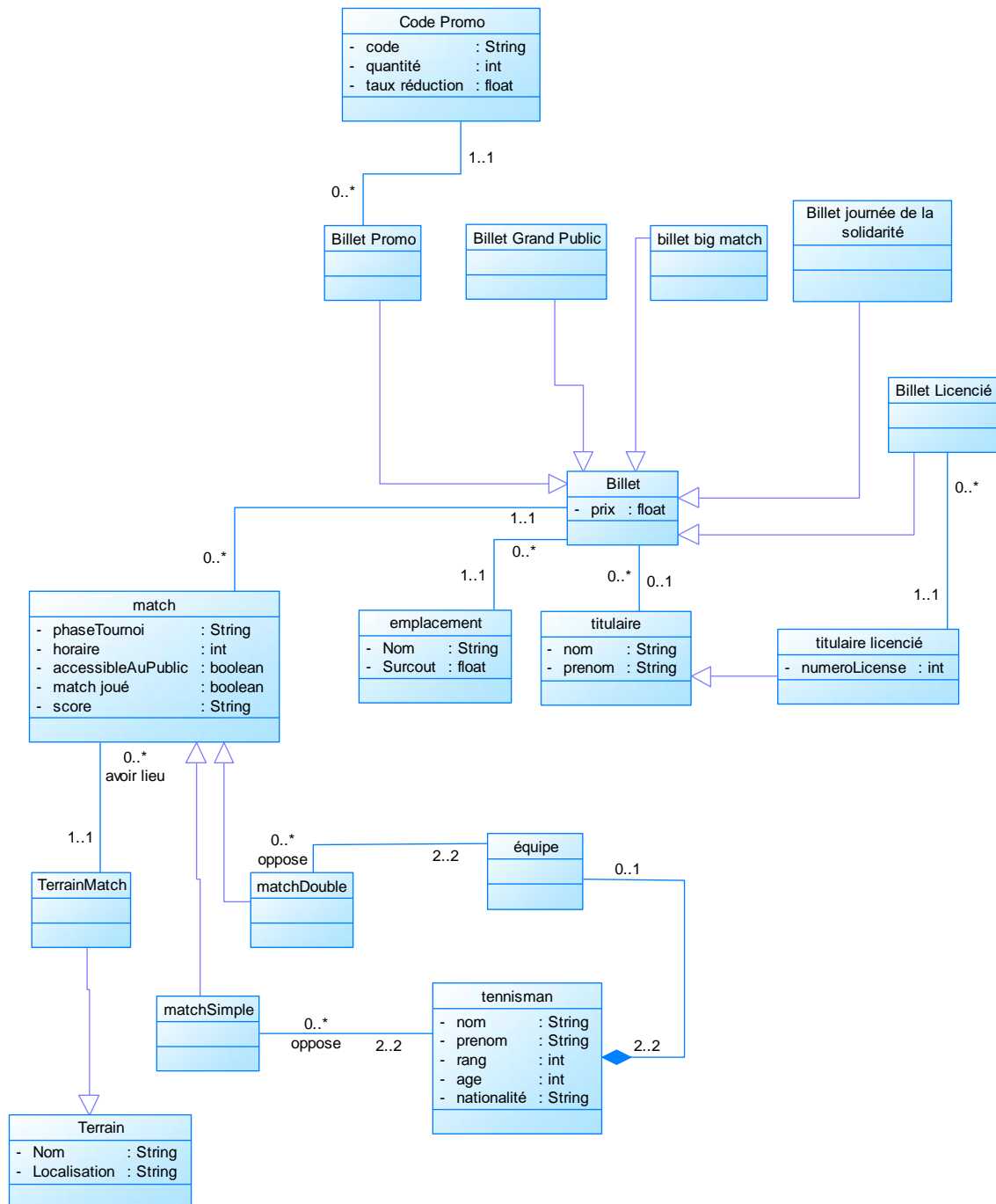
### c. Diagramme d'activités

Ci-dessous les diagrammes d'activités, à droite le diagramme d'activité concernant la connexion en tant qu'administrateur et à droite le diagramme d'activité pour la billetterie en ligne.



#### d. Diagramme de classe Web

Ci-dessous le diagramme de classe pour l'application web :



Nous avons eu l'information qu'un billet donnait accès au site pour toute la journée que trop tard, c'est pourquoi dans le logiciel un billet ne correspond qu'à un match.

#### IV. IHM Application Web

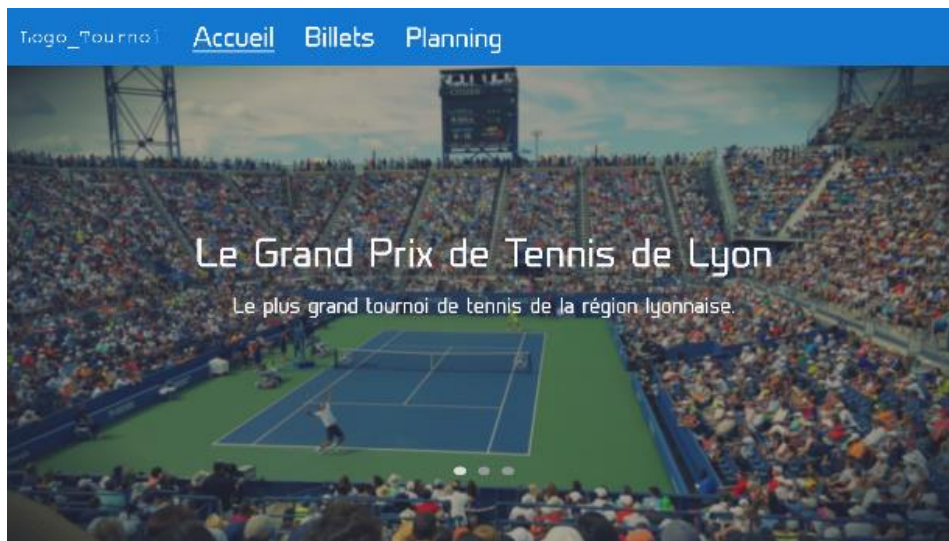


FIGURE 1:PAGE ACCUEIL DU SITE WEB

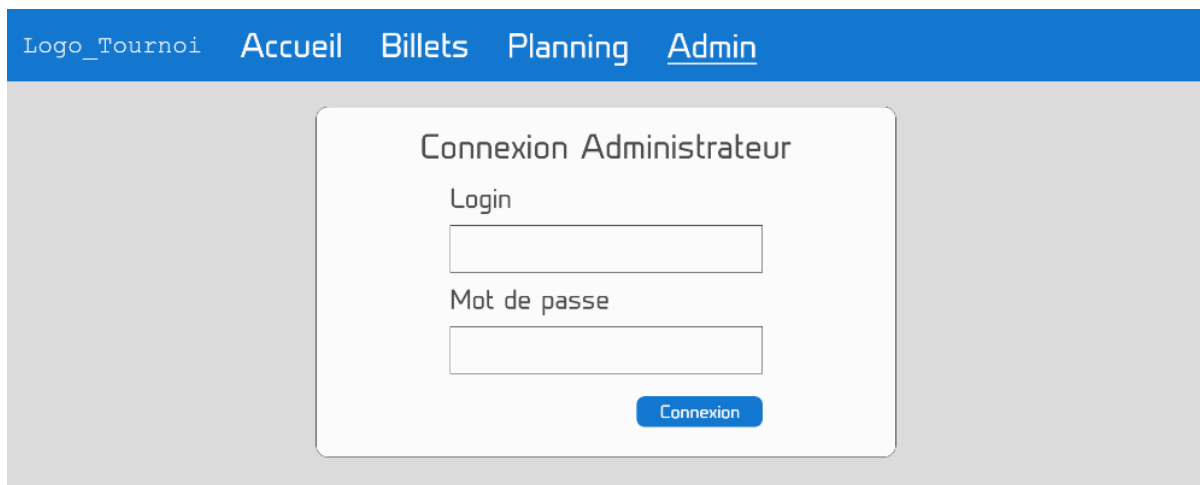


FIGURE 2 : [ADMIN] PAGE DE CONNEXION

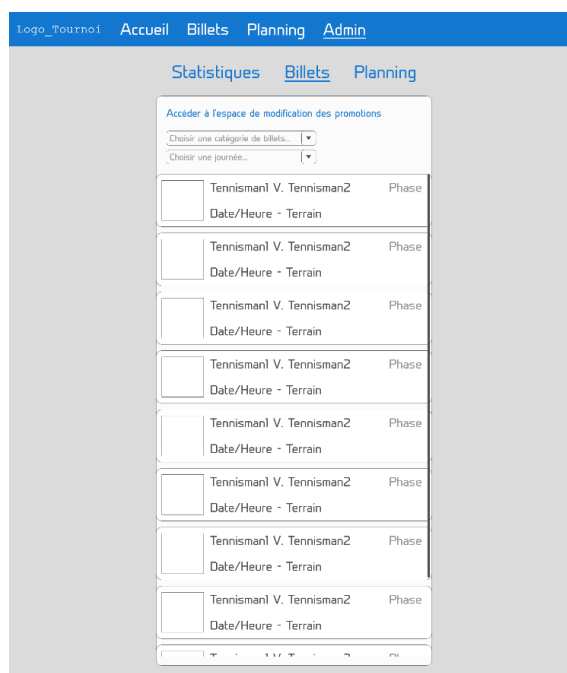


FIGURE 3 : [ADMIN] SELECTION POUR MODIFIER UN BILLET



Logo\_Tournoi Accueil Billets Planning

Objet	Prix Individuel	Nombre	Résultat
Type de billet - N°Match Emplacement	Prix	Nb	XX,XX€
Réduction	Prix	Nb	-XX,XX€
TOTAL			XX,XX€

[Passer au paiement](#)
[Je suis licencié-e](#)

Code promotionnel :  [OK](#)

© 2017 - MATHEZ Claire, HENRIONNET Antoine, LECUTIEZ Simon Connexion Administrateur

FIGURE 7: [VISITEUR] ACHAT BILLET CLASSIQUE

Logo\_Tournoi Accueil Billets Planning

Je suis licencié-e

Nom

Prénom

Numéro de licencié-e

[Valider](#)

© 2017 - MATHEZ Claire, HENRIONNET Antoine, LECUTIEZ Simon Connexion Administrateur

FIGURE 8: [VISITEUR] OBTENIR LE BILLET LICENCIE EN RENSEIGNANT SES INFOS



## V. IHM Application Java

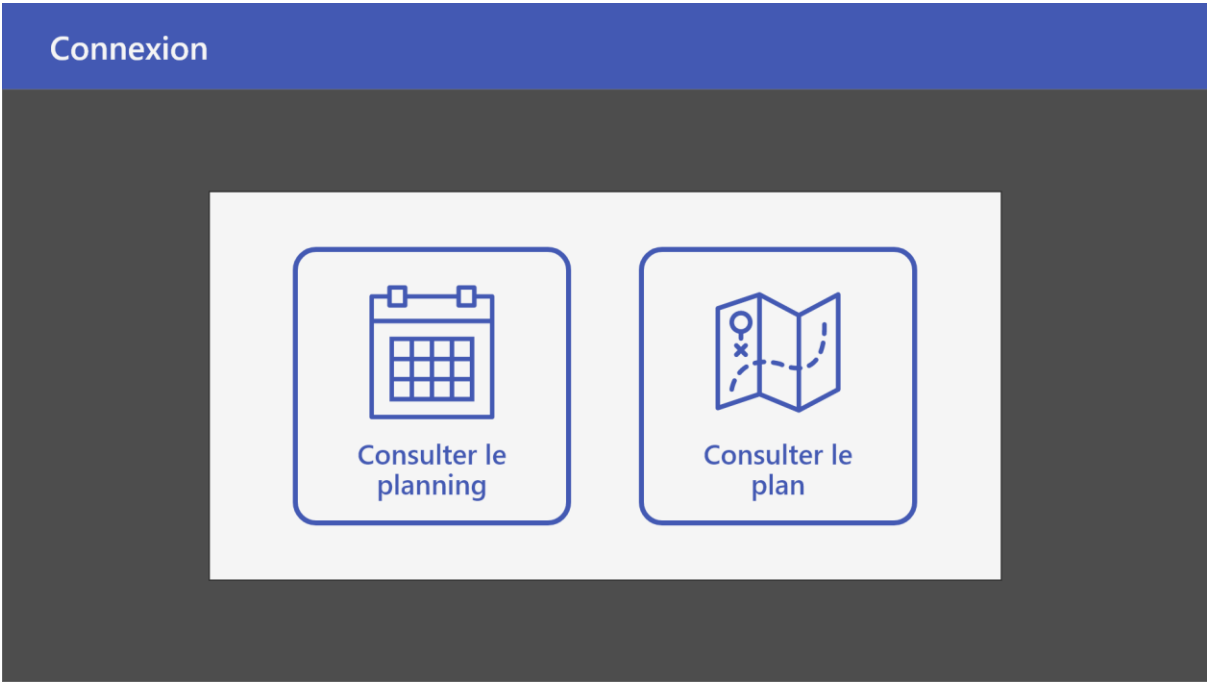


FIGURE 1 : ACCUEIL

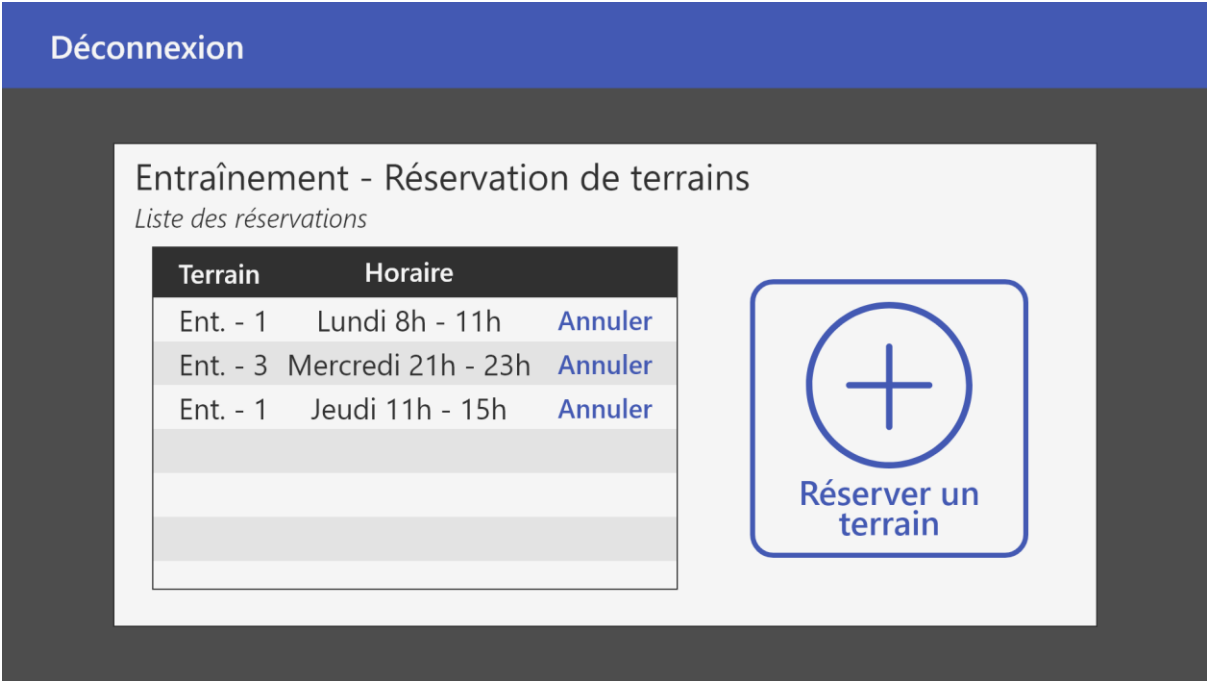
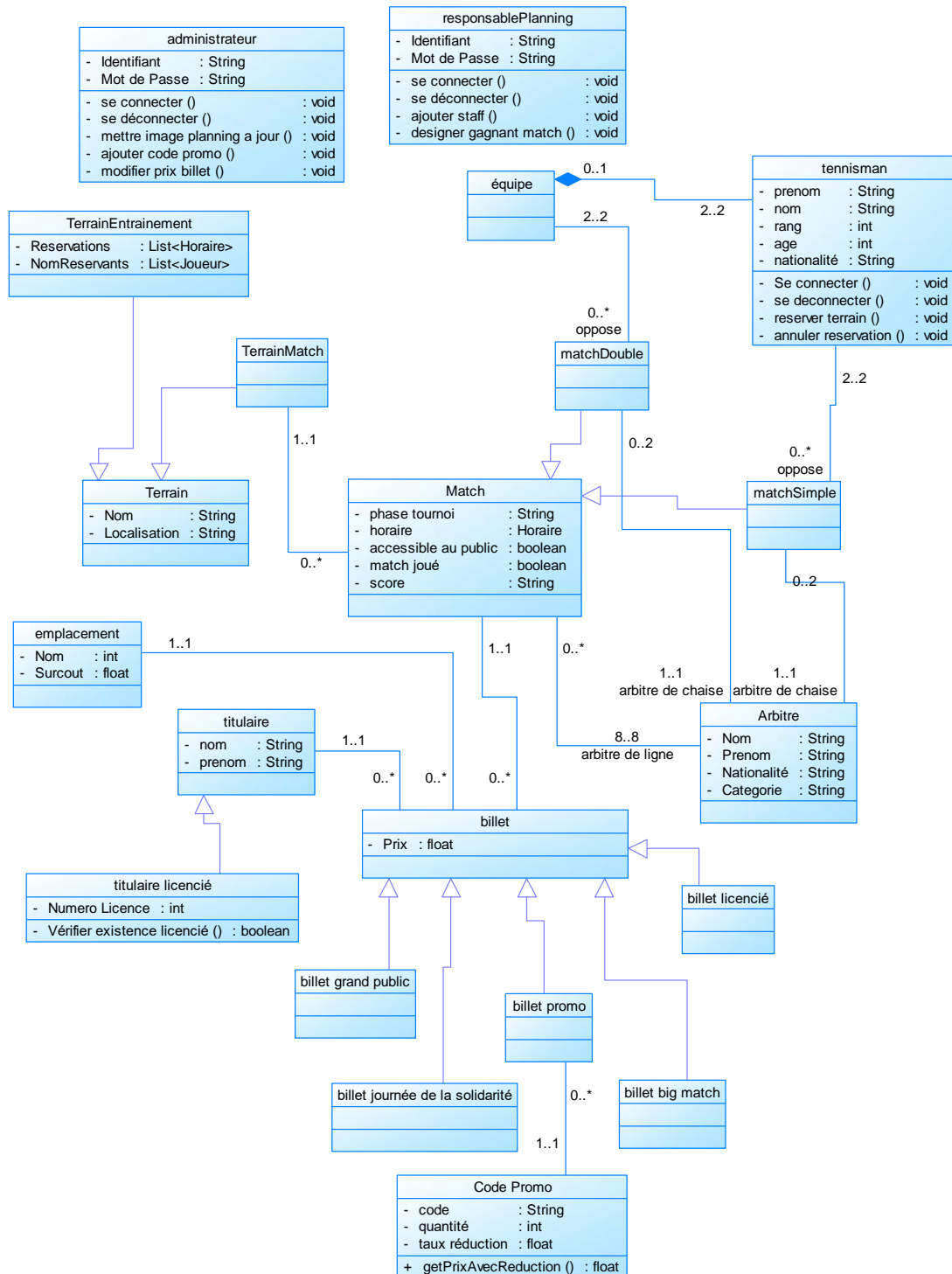


FIGURE 2 : [JOUEUR] RESERVATION DE TERRAINS

## VI. Conception

### a. Diagramme de classes de conception



## b. Script SQL

Table Billet :

```
drop table if exists BILLET;

/*=====*/
/* Table : BILLET */
/*=====*/
create table BILLET
(
    ID_BILLET          int not null,
    ID_MATCH_SIMPLE    int,
    ID_MATCH_DOUBLE    int,
    ID_EMLACEMENT      int not null,
    CODE               text,
    NUMLICENCIE        text,
    PRIX_DE_BASE       float,
    PRENOM_TITULAIRE   text,
    NOM_TITULAIRE      text,
    primary key (ID_BILLET)
);

alter table BILLET add constraint FK_ACCORDE_UNE_REDUCTION foreign
key (NUMLICENCIE)
    references LICENCIE (NUMLICENCIE) on delete restrict on update
restrict;

alter table BILLET add constraint FK_CONCERNE_DOUBLE foreign key
(ID_MATCH_DOUBLE)
    references MATCHDOUBLE (ID_MATCH_DOUBLE) on delete restrict on
update restrict;

alter table BILLET add constraint FK_CONCERNE_SIMPLE foreign key
(ID_MATCH_SIMPLE)
    references MATCHSIMPLE (ID_MATCH_SIMPLE) on delete restrict on
update restrict;

alter table BILLET add constraint FK_REDUCTION foreign key (CODE)
    references CODEPROMO (CODE) on delete restrict on update
restrict;

alter table BILLET add constraint FK_SE_TROUVE_A foreign key
(ID_EMLACEMENT)
    references EMLACEMENT (ID_EMLACEMENT) on delete restrict on
update restrict ;
```

Table MatchDouble :

```
drop table if exists MATCHDOUBLE;

/*=====*/
/* Table : MATCHDOUBLE */
/*=====*/
create table MATCHDOUBLE
(
    ID_MATCH_DOUBLE      int not null,
    ID_TERRAIN            int not null,
    ID_ARBITRE            int not null,
    ID_EQUIPE             int not null,
    HORAIRE_DOUBLE        text,
    PHASE_TOURNOI_DOUBLE int,
    ACCESSIBLE_DOUBLE     bool,
    JOUE_DOUBLE           bool,
    SCORE_DOUBLE          text,
    primary key (ID_MATCH_DOUBLE)
);

alter table MATCHDOUBLE add constraint FK_ARBITRE_CHAISE_DOUBLE
foreign key (ID_ARBITRE)
references ARBITRE (ID_ARBITRE) on delete restrict on update
restrict;

alter table MATCHDOUBLE add constraint FK_A_LIEU_SUR_DOUBLE foreign
key (ID_TERRAIN)
references TERRAIN (ID_TERRAIN) on delete restrict on update
restrict;

alter table MATCHDOUBLE add constraint FK_GAGNE_DOUBLE foreign key
(ID_EQUIPE)
references EQUIPE (ID_EQUIPE) on delete restrict on update
restrict;
```

### c. Squelette Java

Classe CodePromo :

```
/**
 * Module: CodePromo.java
 * Author: Lilou
 * Purpose: Defines the Class CodePromo
 */

import java.util.*;

/** @pdId a71fdf4e-452b-4796-be7b-2bdd2a5f5366 */
public class CodePromo {
    /** @pdId a17a1008-6e5e-44bd-b13e-4947c0d2dd9b */
    private String code;
    /** @pdId b36962be-8957-41e6-b32f-de3aaa1d96ae */
    private int quantite;
    /** @pdId 6c162797-d0d5-45ed-9a64-3f892951ddd8 */
    private float tauxReduction;

    /** @pdId 7759e80f-f210-431b-af2b-360173fff27f */
    public float getPrixAvecReduction() {
        // TODO: implement
        return 0;
    }
}
```

Classe TerrainEntrainement :

```
/**
 * Module: TerrainEntrainement.java
 * Author: Lilou
 * Purpose: Defines the Class TerrainEntrainement
 */

import java.util.*;

/** @pdId 503da013-7d73-4cf5-ad3d-ae2c1962238c */
public class TerrainEntrainement extends Terrain {
    /** @pdId 7ac31fda-fd05-46eb-96f2-a0a36e435f92 */
    private List<Horaire> reservations;
    /** @pdId 9a7534d8-2eb2-4c02-b1f2-a5e8a74db780 */
    private List<Joueur> nomReservants;
}
```

Rapport d'analyse | HENRIONNET ANTOINE - LECUTIEZ SIMON - MATHEZ CLAIRE

