

RAPPORT D'ANALYSE & CONCEPTION

Projet Grand Prix de Tennis de Lyon

Rapport réalisé par CANAQUE Alan, DUTHOIT Pierre-Louis, LOURIKI Omar
Projet réalisé dans un cadre universitaire : IUT Informatique Lyon 1 (CPOA)

I. PRESENTATION DU GRAND PRIX DE TENNIS DE LYON	3
II. PRESENTATION DU PROJET	3
1. SYSTEME DE BILLETTERIE	3
2. PLANNING DES MATCHS	4
3. ERGONOMIE LOGICIELLE	4
III.DESCRPTION COMPLETE DU MODELE.....	5
1. DIAGRAMME DE PACKAGES GPTL	5
2. PACKAGE PLANNING DES MATCHS	6
a. Diagramme de classe d'analyse	6
b. Diagramme de cas d'utilisation	7
c. Diagramme d'activité Déplacer match	8
d. Diagramme de séquence système Déplacer match	9
e. Diagramme d'activité Réservation du terrain d'entrainement	10
f. Diagramme de séquence système Réservation du terrain d'entrainement	11
g. Diagramme de classe	12
h. Schéma de la base de données (Modèle physique de données)	14
i. Squelettes de classes Java	15
j. Interface Homme-Machine Système de Billetterie (web)	17
3. PACKAGE SYSTEME DE BILLETTERIE	22
a. Diagramme de classe d'analyse	22
b. Diagramme de cas d'utilisation	23
c. Diagramme d'activité Création des codes promotionnels	24
d. Diagramme de séquence système Création des codes promotionnels	25
e. Diagramme d'activité Déclarer possession	26
f. Diagramme de séquence système Déclarer possession	28
g. Diagramme de classe	30
h. Schéma de la base de données (Modèle physique de données)	32
i. Script de génération de la base de donnée	33
j. Squelette de classes Java	34
k. Interface Homme-Machine Planning des matchs	35

I. Présentation du Grand Prix de Tennis de Lyon

Le Grand Prix de Tennis de Lyon (GPTL) est un tournoi international comptant au classement de l'ATP (Association of Tennis Professionals). Il fait partis des trois tournois les plus renommés de France. Chaque gagnant d'un match se voit attribué une somme d'argent.

Le GPTL est composé de deux tournois : Simples et Doubles, les deux exclusivement masculins.

Le tournoi comporte une phase de qualification, le samedi et le dimanche, à laquelle participent 32 joueurs en simple et 64 joueurs en tournoi double, puis un système de huitième, de quart, de demi, et de finale, qui se dérouleront les 7 jours de la semaine suivante.

Le Palais des Sports dispose d'un court Central en trois niveaux, et de cinq courts d'entraînements.

Différents types de billets sont mis à disposition : Grand Public, Licenciés, Promotions, Invitations, Finale.

Le tournoi est constitué de 4 comités:

- Comité de Pilotage
- Service Informatique
- Comité de Finances
- Comité OTG (on the ground)

II. Présentation du projet

Pour faciliter la gestion de ce projet, nous allons réaliser deux applications. Une application de gestion de billets qui permet aux utilisateurs de réserver leurs places, et une application Planning des matchs qui permet aux joueurs de réserver des courts de tennis pour leur entraînement et au comité de pilotage d'organiser le planning des matchs de la compétition, avec la possibilité de modifier, de supprimer ou d'ajouter des matchs selon les circonstances.

1. Système de Billeterie

Cette application sera développée au format Web, à l'aide du langage PHP et d'une base de données MySQL qui permettra de stocker les différents types de billets et leur nombre. Cette application permettra aux internautes d'acheter leurs places, en spécifiant éventuellement un type de billet.

On doit gérer les différents types de billets :

- Billet Grand Public, dont le prix dépend des jours et des places.
- Billet Promo, dont les prix dépendent de la réduction sur le prix de base.
- Billet Licencié, pour les joueurs, dont les prix dépendant du pourcentage appliqué.
- Billet Journée de la Solidarité.

Pour chaque billet, l'utilisateur choisira son emplacement et entrera éventuellement ses codes de réductions. Ces deux facteurs influant sur le prix final du billet.

Il faudra prévoir le nombre de billets par catégories de billets.

L'utilisateur sera alors redirigé sur la page de paiement, que nous n'implémenterons pas dans l'application.

2. Planning des Matches

Cette application en Java sera développée en Java, à l'aide de l'IDE NetBeans, langage orienté objet. Elle permettra d'une part aux joueurs de réserver des terrains d'entraînement, et d'autre part au Comité de Pilotage, d'organiser le planning des matchs.

On doit gérer les conditions émises :

- Un arbitre ne peut arbitrer que quatre matchs (deux Doubles, et deux Simples) dans le tournoi
- Un arbitre ne peut pas arbitrer des joueurs de même nationalité que lui.
- Un joueur ne peut pas être en match simple et en match double dans un même créneau.

Le planning peut être généré automatiquement, mais le comité de pilotage a l'autorisation de modifier, de supprimer, ou d'ajouter un match selon les circonstances.

Modifier signifie modifier l'arbitre, les joueurs, le court ou le créneau.

La suppression est envisageable en cas d'annulation, et sera compensée par l'option d'insertion qui permettra de remettre le match supprimé dans un créneau libre.

L'application gèrera automatiquement les contraintes citées plus haut.

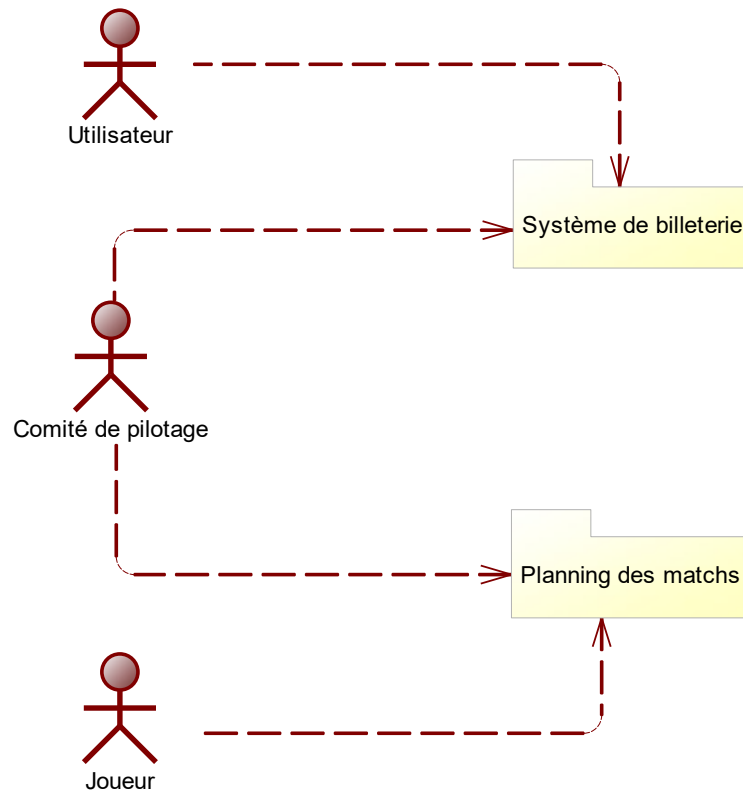
Et elle permettra d'afficher les créneaux disponibles.

3. Ergonomie Logicielle

On veillera à avoir une interface épurée, claire, et ergonomique. Le joueur, le comité, ou le client doivent pouvoir utiliser l'application de manière naturelle et intuitive.

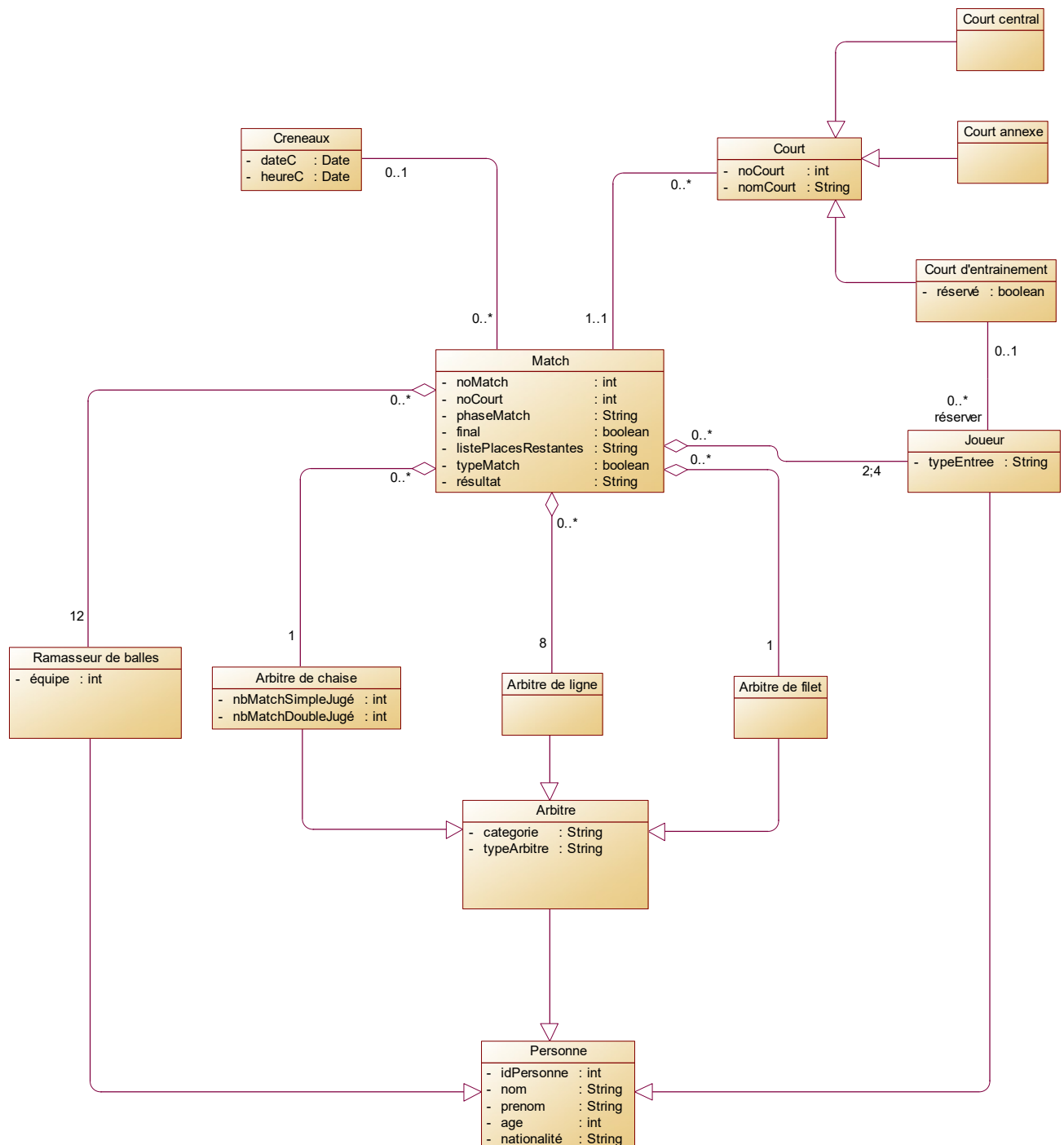
III. Description complète du modèle

1. Diagramme de packages GPTL

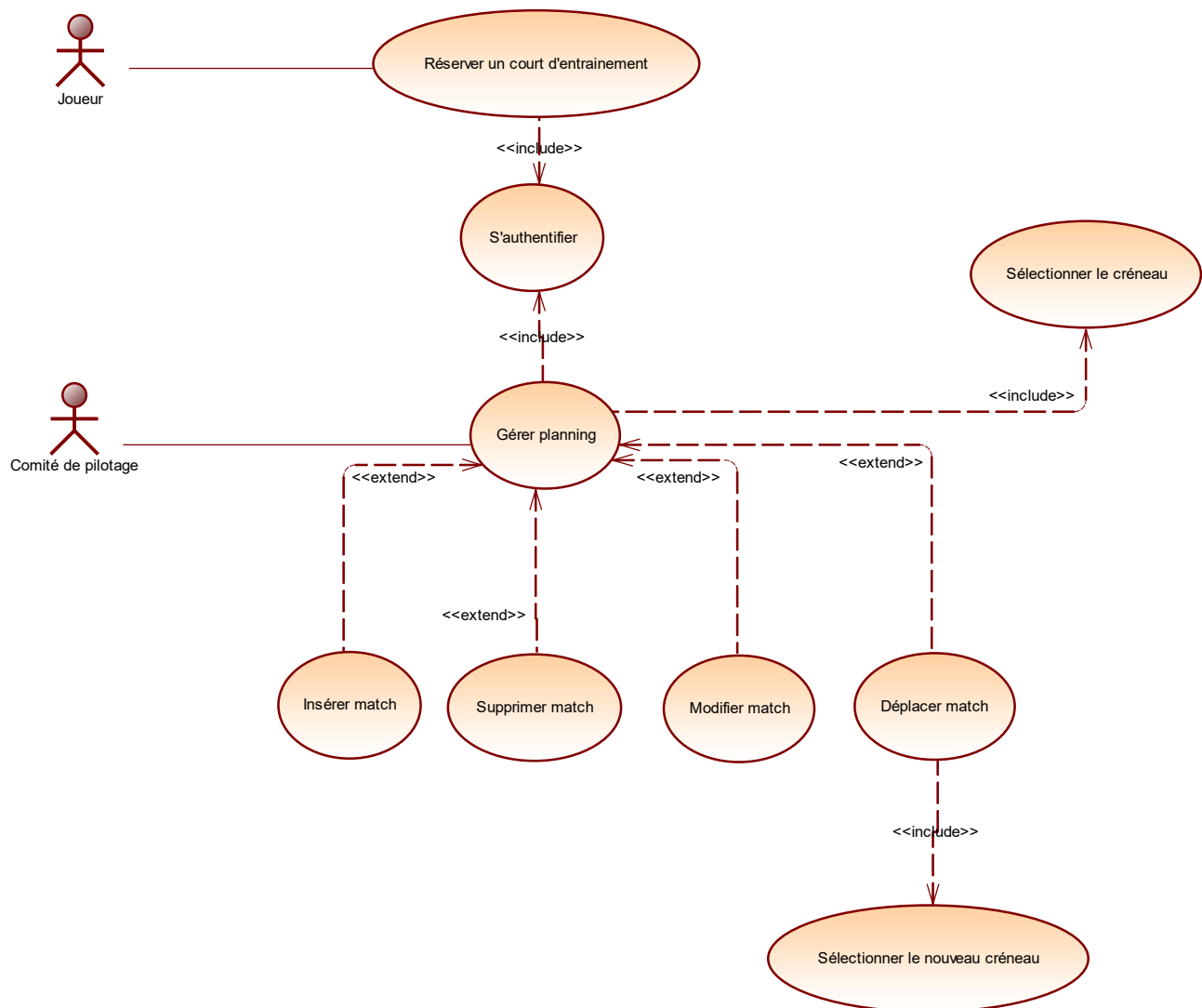


2. Package Planning des matchs

a. Diagramme de classe d'analyse



b. Diagramme de cas d'utilisation



Nous distinguons deux cas. Celui du **Joueur** et celui du **Comité de pilotage**.

Le **Joueur** réserve un terrain d'entraînement. Cette action inclut le fait de s'authentifier, car seul un joueur peut choisir un terrain d'entraînement, et la réservation doit être au nom de quelqu'un.

Le **Comité de pilotage** lui a pour rôle de gérer le planning. Ceci inclut le fait de s'authentifier, pour que seules les personnes qui possèdent les identifiants administrateur puissent gérer le planning.

Le planning est généré automatiquement.

Cette action inclut le fait de sélectionner un créneau dans lequel saisir les résultats des matchs.

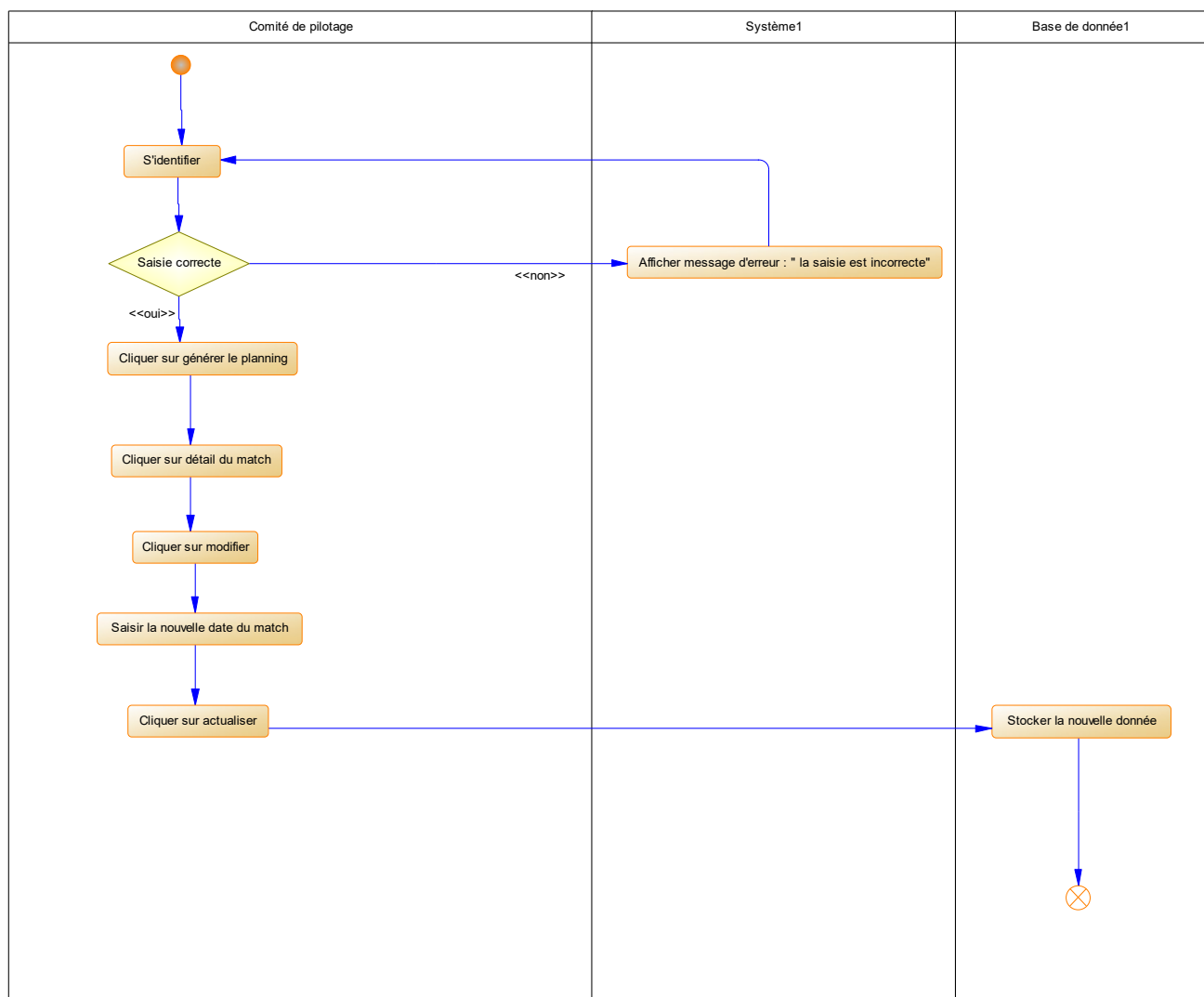
Elle étend aussi le fait d'insérer, de supprimer, de modifier ou de déplacer un match.

Par insérer un match, on entend le fait d'insérer un match que l'on aurait préalablement supprimé en attente de pouvoir le replacer. Il n'est pas question ici d'insérer tous les matchs, car comme dit plus haut, le planning est généré automatiquement.

Modifier match sous-entend : modifier les joueurs, les arbitres, le terrain, et ceci est différent de déplacer match qui ici concerne le fait de déplacer un match d'un créneau à un autre.

On gèrera automatiquement les contraintes imposées de sorte à ce que le **Comité de pilotage** ne puisse pas effectuer une action qui irait à l'encontre des contraintes imposées. (ex : un arbitre de même nationalité qu'un joueur, ou le déplacement sur un créneau déjà occupé)

c. Diagramme d'activité Déplacer match



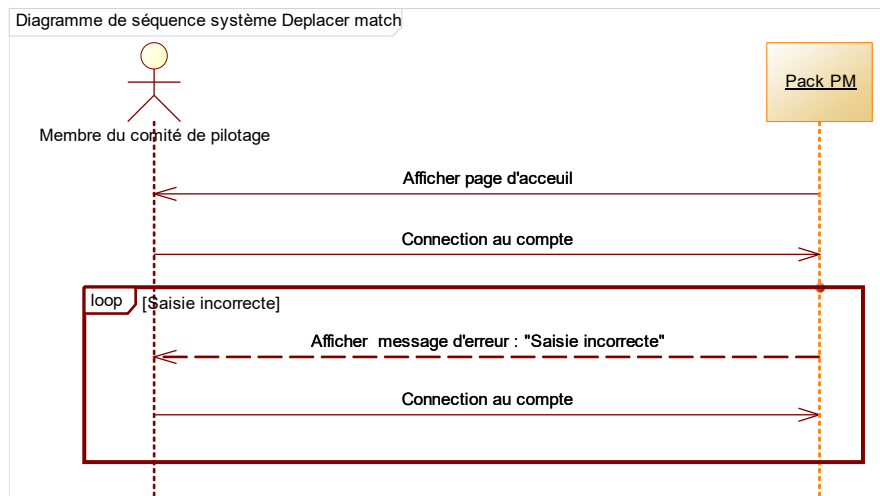
Le **Comité de pilotage** s'identifie. Il saisit ses identifiants, si ceux-ci sont faux, alors un message d'erreur le lui indique et le renvoie de nouveau à la fenêtre d'authentification.

Une fois identifié, il clique sur la fonction "Générer le planning", qui générera automatiquement un planning respectant les contraintes spécifiées.

Le comité cliquera alors sur "Détail du match", afin d'accéder au détail d'un match, puis sur "Modifier", et saisira la nouvelle date du match, puis validera.

Enfin, ces nouvelles informations seront stockées dans la base de données.

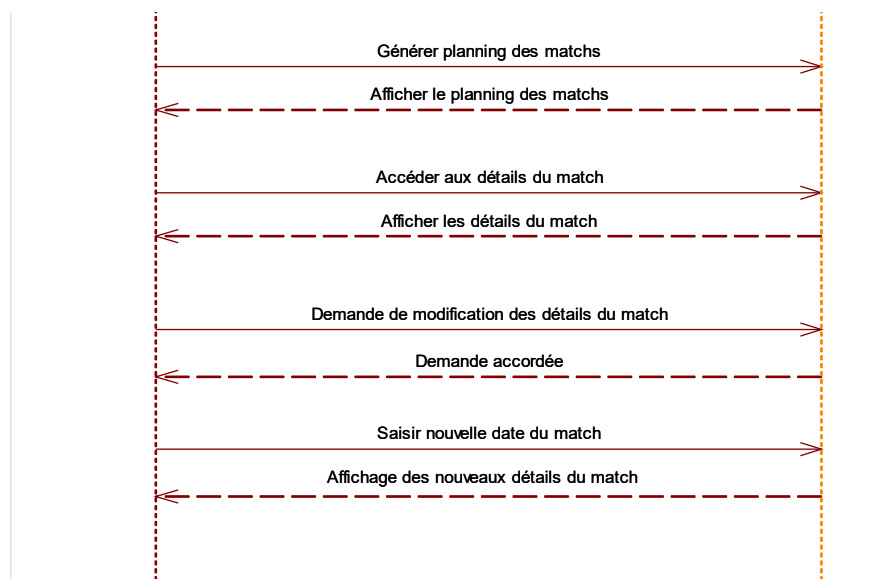
d. Diagramme de séquence système Déplacer match



Le **Pack PM** affiche la page d'accueil au **Membre du comité de pilotage**.

Le membre du comité se connecte à son compte en envoyant les données au Pack PM.

On ouvre une boucle while. Si la saisie est incorrecte, le Pack PM affiche un message d'erreur, et le membre se reconnecte.



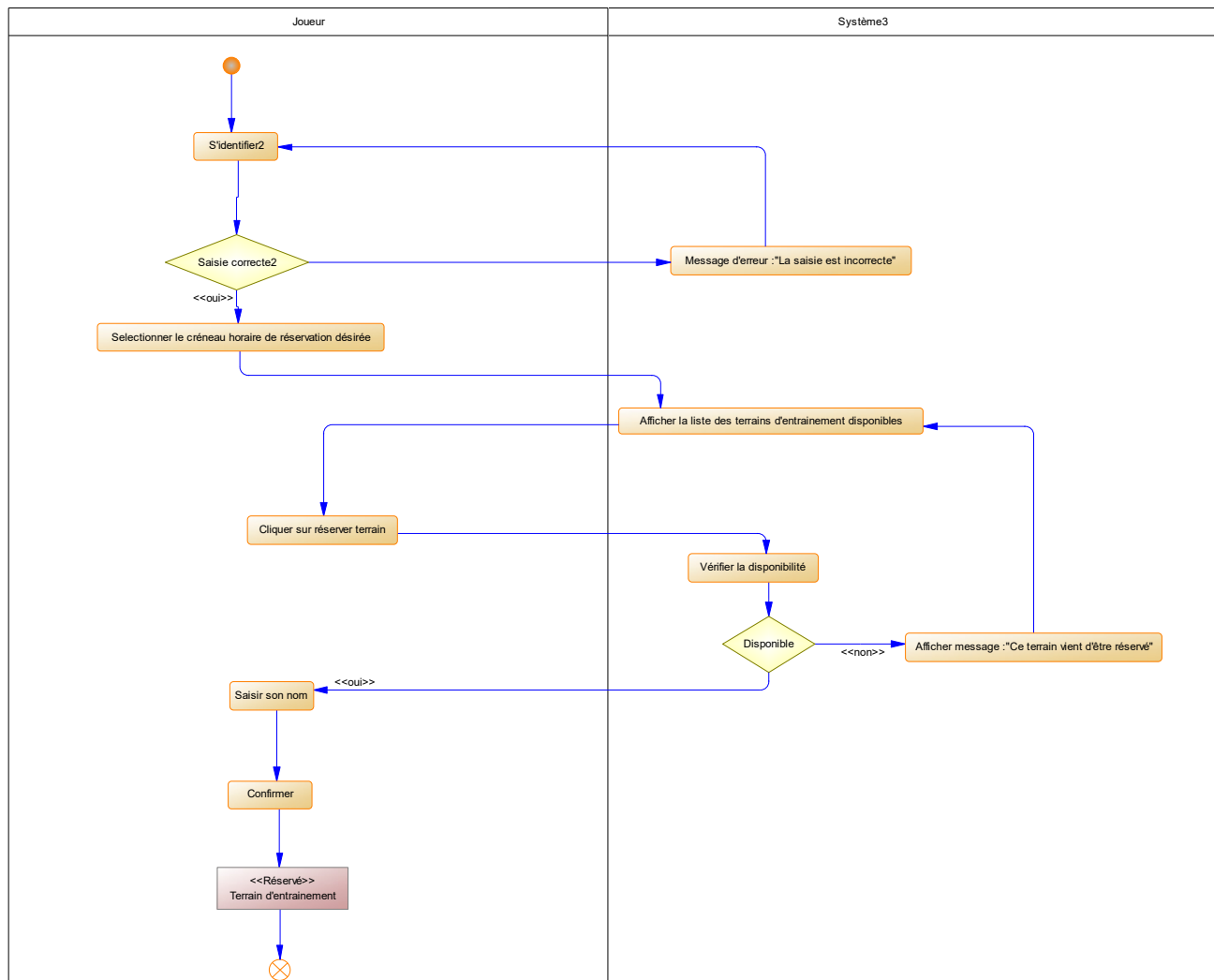
Une fois connecté, le comité génère le planning des matchs à travers un bouton qui le générera automatiquement. Puis le Pack PM affichera au comité le planning généré.

Le Comité pourra alors accéder aux détails de chaque match qui lui seront affichées, ainsi que demander une modification des détails du match.

Une fois la demande accordée par le pack PM.

Il saisit une nouvelle date de match.

e. Diagramme d'activité Réserveation du terrain d'entrainement



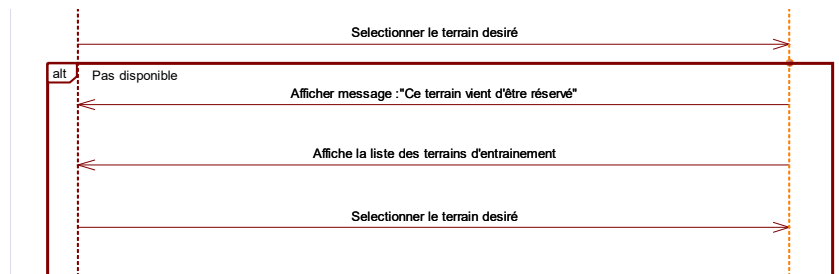
Le joueur s'identifie. Il saisit ses identifiants, si ceux-ci sont erronés, un message d'erreur lui est envoyé.

Lorsque l'identification est réussie, il sélectionne la date et le créneau horaire qu'il désire, les terrains disponibles pour ce créneau s'affichent automatiquement, puis il choisit un terrain. Il vérifie sa disponibilité (il se peut que ce terrain ait été réservé entre temps), si le terrain n'est plus disponible, alors un message d'erreur est affiché, sinon il lui sera demandé de saisir son nom et de confirmer. Le terrain sera alors réservé.

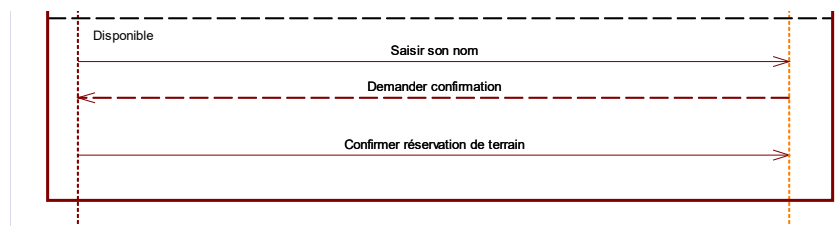
f. Diagramme de séquence système Réservation du terrain d'entrainement



Le **Joueur** génère la liste des créneaux. Le **Pack PM** la lui renvoie.
 Il sélectionne son créneau, puis génère la liste des terrains.
 Le Pack lui affiche la liste des terrains.



Le Joueur sélectionne son terrain.
 Si le terrain a été réservé entre temps, un message d'erreur s'affiche, et le pack PM lui renvoie la liste des terrains disponibles. (Il s'agit d'un cas certes mineur, mais qui peut éventuellement arriver et qui serait problématique)



Une fois le terrain disponible, il saisit son nom, et confirme sa réservation.

La classe **Match** est composée des attributs :

- *noMatch* qui sert de clé primaire
- *noCourt* qui est une clé étrangère et qui permet de connaître le court utilisé
- *phaseMatch* qui est un String qui indique la phase du tournoi du match (ex : 1/8 de finale)
- *finale* de type boolean qui s'il s'agit d'une finale
- *listePlacesRestantes* qui est décrémenté de un à chaque place réservée.
- *typeMatch* de type boolean qui indique si le match est Simple ou Double
- *résultat* qui indique le résultat du match

La classe **Personne** de laquelle héritent les classes **Joueur**, **Arbitre**, et **Ramasseur de balles**.

La classe **Personne** est dotée des attributs :

- *idPersonne* qui sert de clé primaire
- *nom* qui correspond au nom
- *prénom* qui correspond au prénom
- *âge* qui correspond à l'âge
- *nationalité* qui correspond à la nationalité de la Personne. Elle sera utile notamment pour les joueurs et arbitres afin de respecter les conditions fixées pour la génération des matchs.

La classe **Joueur** hérite de la classe **Personne**, à laquelle on rajoute l'attribut :

- *typeEntrée* (entrée directe, Wild Card, inviation, special Exempts)

La classe **Arbitre** hérite de la classe **Personne**, à laquelle on ajoute les attributs :

- *catégorie*
- *typeArbitre*

La classe **Arbitre** possède trois classes filles : **Arbitre de ligne**, **Arbitre de chaise** et **Arbitre de filet**.

La classe **Arbitre de chaise** est dotée des attributs :

- *nbMatchSimpleJugé* qui correspond au nombre de matchs simples jugés par l'arbitre pendant le tournoi, ce nombre ne pouvant pas dépasser 2 en raison des contraintes évoqués.
- *nbMatchDoubleJugé* qui correspond au nombre de matchs doubles jugés par l'arbitre pendant le tournoi, ce nombre ne pouvant pas dépasser 2 en raison des mêmes contraintes de génération de match.

La classe **Ramasseur de balles** hérite de la classe **Personne**, à laquelle on ajoute l'attribut :

- *équipe* car il y'a différentes équipes de ramasseurs de balles

Une classe **Créneaux** avec pour attributs :

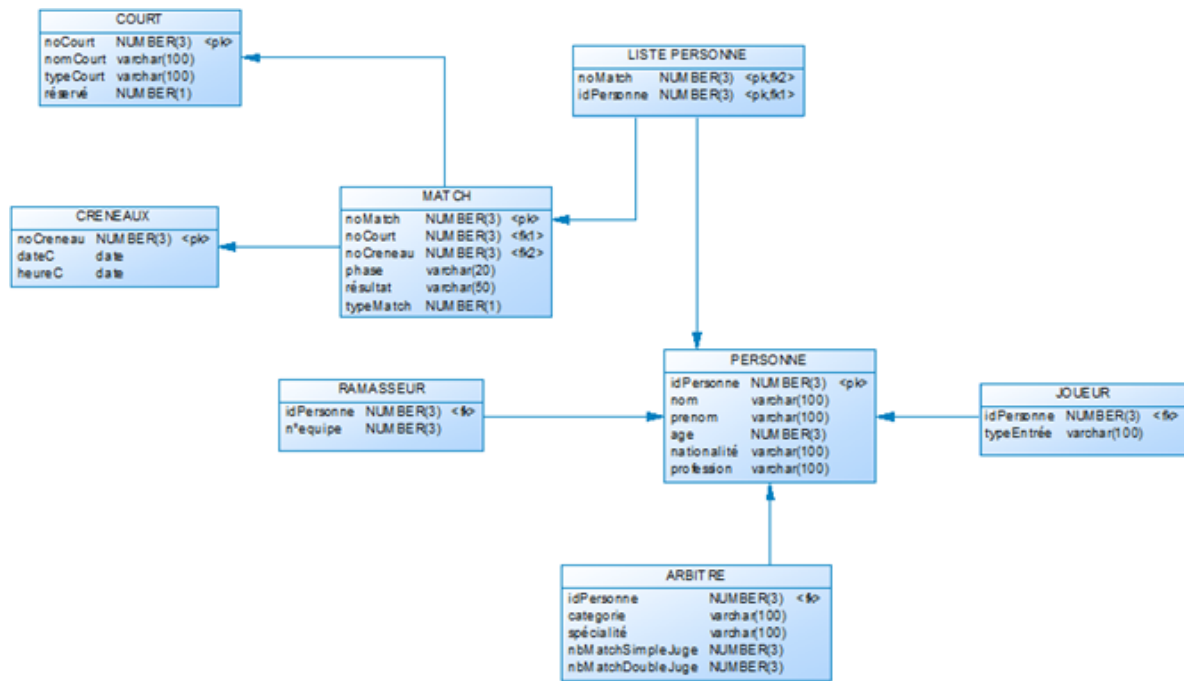
- *date* qui correspond à la date associée à un match
- *heure* qui correspond à l'heure associée à un match

Une classe **Court** avec comme attributs :

- *noCourt* qui sert de clé primaire
- *nomCourt* qui correspond au nom du court

Les méthodes de toutes les classes sont des méthodes Get et Set qui permettent de récupérer et de fixer les valeurs des attributs.

h. Schéma de la base de données (Modèle physique de données)



La génération de notre modèle physique de données permet de faire apparaître **les clés primaires et étrangères** de chaque table de notre future base de données.

Les types des attributs sont ceux d'Oracle :

- NUMBER : pour les attributs de de type entiere ou double.
- VARCHAR : pour ceux de type chaînes de caractères.
- DATE : pour les attributs de type date.

i. Squelettes de classes Java

Exemple de la classe Court :

```
/* *****  
 * Module: Court.java  
 * Author: 11404604  
 * Purpose: Defines the Class Court  
 * ***** */  
  
package Planning_des_matches;  
  
import java.util.*;  
  
/** @pdOid 49827366-51ce-44ef-9104-1308800b240c */  
public class Court {  
    /** @pdOid cd768e74-1997-459d-b85b-16b5e6161017 */  
    private int noCourt;  
    /** @pdOid 135900ff-ab76-48e7-9c98-ecd8b5385eab */  
    private String nomCourt;  
  
    /** @pdOid 74b300cb-4a8f-4ecd-b739-f5f652886968 */  
    public int getNumCourt() {  
        // TODO: implement  
        return 0;  
    }  
  
    /** @pdOid 1243c5ff-0928-4b35-94da-9c9194ca0d2a */  
    public void getNomCourt() {  
        // TODO: implement  
    }  
  
    /** @pdOid 27feb7c2-12f7-4d69-93e8-56d38e2d12e9 */  
    public int setNumCourt() {  
        // TODO: implement  
        return 0;  
    }  
  
    /** @pdOid a4215461-3b20-4cf9-a230-ce085153ce62 */  
    public void setNomCourt() {  
        // TODO: implement  
    }  
}
```

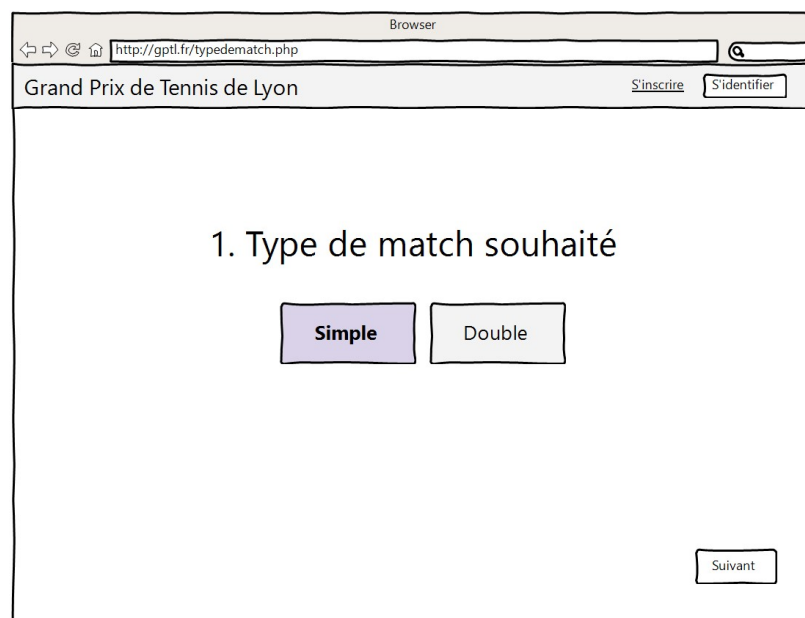
Exemple de la classe Joueur :

```
/* *****  
 * Module: Joueur.java  
 * Author: User  
 * Purpose: Defines the Class Joueur  
 * ***** */  
  
package Planning_des_matches;  
  
import java.util.*;  
  
/** @pdOid af1782c7-11e4-4745-8156-5ebe2542852b */  
public class Joueur extends Personne {  
    /** @pdOid 78bc1fe9-ed07-4ca9-a1d4-9fe85c934d72 */  
    private String entree;  
  
    /** @pdOid 430a3636-8e22-456b-8cda-abd6291675b4 */  
    public String getTypeEntree() {  
        // TODO: implement  
        return null;  
    }  
  
    /** @pdOid 4755c364-3ef7-458f-b91b-f237eb950c00 */  
    public String setTypeEntree() {  
        // TODO: implement  
        return null;  
    }  
}
```


j. Interface Homme-Machine Système de Billetterie (web)



La page d'accueil du site, avec un message de bienvenue et un bouton qui nous redirige afin de Réserver des places pour les matchs.



Cette interface permet de choisir le type de match que l'on souhaite réserver.

The screenshot shows a web browser window with the URL `http://gptl.fr/phasedematch.php`. The page title is "Grand Prix de Tennis de Lyon". There are links for "S'inscrire" and "S'identifier" in the top right. The main heading is "2. Phase de match". Below it, there are five buttons representing the match phases: "1/16 de finale" (highlighted in purple), "1/8 de finale", "1/4 de finale", "1/2 finale", and "Finale". At the bottom right, there are "Précédent" and "Suivant" buttons.

Cette interface permet de choisir la phase de match pour laquelle on veut réserver un billet

The screenshot shows a web browser window with the URL `http://gptl.fr/phasedematch.php`. The page title is "Grand Prix de Tennis de Lyon". There are links for "S'inscrire" and "S'identifier" in the top right. The main heading is "3. Sélectionnez votre match". Below it, the text "1/16 de finale - Simple" is displayed. The matches are listed by date: "Samedi 30" and "Dimanche 31". Each match entry includes a time slot, player names, a course name, and a "Sélectionner" button. The match on Sunday at 8h is highlighted in purple.

Date	Time	Joueur 1	Joueur 2	Cours	Action
Samedi 30	12h	Joueur 1 (FR)	Joueur 2 (EN)	Cours d'entraînement n°1	Sélectionner
	15h	Joueur 3 (DE)	Joueur 4 (ES)	Cours d'entraînement n°2	Sélectionner
Dimanche 31	8h	Joueur 5 (POR)	Joueur 6 (EST)	Cours Phillipe Chatrier	Sélectionner

Cette interface est celle du choix du match. La liste des matchs est affichée, et on sélectionne le match voulu.

Browser

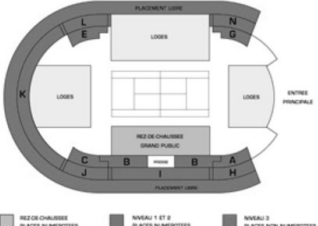
http://gptl.fr/choisirplace.php

Grand Prix de Tennis de Lyon

S'inscrire S'identifier

4. Sélectionnez votre place

Dimanche 31 - 8h - Joueur 5 / Joueur 6



☒ A (15 places disponibles) 15€
☐ B (26 places disponibles) 12€
☐ C (13 places disponibles) 12€
☐ D (19 places disponibles) 15€
☐ E (5 places disponibles) 19€
☐ F (0 places disponibles) 15€
☐ G (7 places disponibles) 12€
☐ H (23 places disponibles) 12€
☐ I (12 places disponibles) 15€
☐ J (3 places disponibles) 15€
☐ K (0 places disponibles) 15€
☐ L (23 places disponibles) 19€

Précédent Suivant

Cette interface permet de choisir l'emplacement sur le court, avec les différences de prix et les disponibilités.

Browser

http://gptl.fr/identification.php

Pour poursuivre vous devez vous identifier

Adresse mail

Mot de passe

☐ Se souvenir de moi

Valider

L'authentification nécessite l'adresse mail et le mot de passe, avec une option qui propose de se souvenir de l'utilisateur pour qu'il n'ait pas à retaper les mots de passe.

Browser

http://gptl.fr/choixbillet.php

Grand Prix de Tennis de Lyon

S'inscrire Se déconnecter

5. Choisir billet

Joueur 5 / Joueur 6
Dimanche 31 - 8h
Cours Philippe Chatrier - Tribune A

☒ Billet normal 15€

☐ Billet promotionnel 10€ - Entrez votre code promotionnel :

☐ Billet licencié Tennis 12€ - Entrez votre numéro licencié :

☐ Billet "Journée de la Solidarité" 12€ - Entrez votre code JdIS :

Précédent Confirmer

Cette interface permet de choisir son billet. On peut sélectionner le type de billet, et éventuellement entrer les codes promotionnels ou le numéro de licence, puis confirmer, ou revenir en arrière.

Browser

http://gptl.fr/paiement.php

Grand Prix de Tennis de Lyon

S'inscrire Se déconnecter

6. Paiement

Informations billet

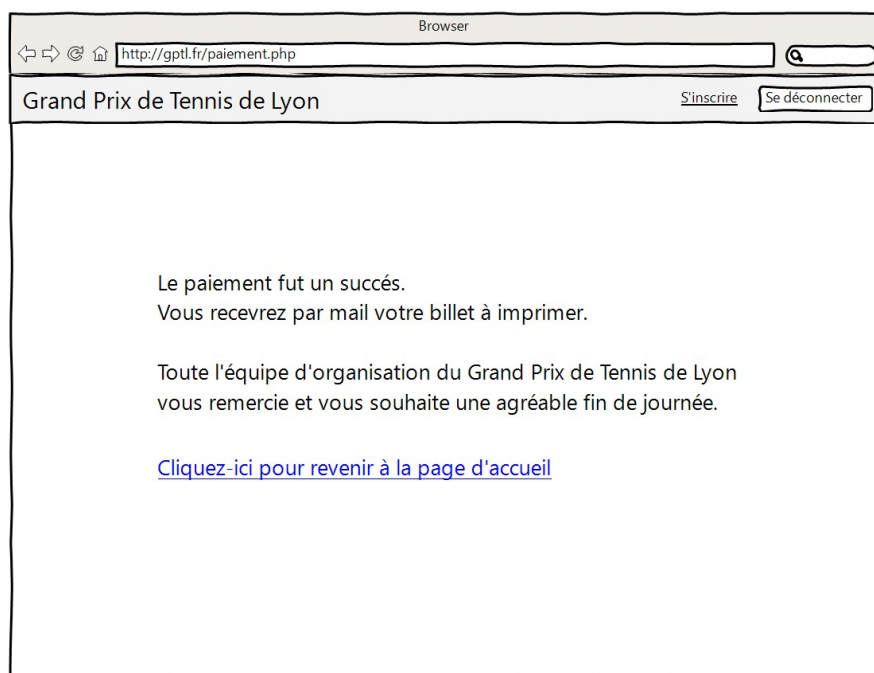
Joueur 5 / Joueur 6
Dimanche 31 - 8h
Cours Philippe Chatrier - Tribune A
Billet grand public

Prix total à payer : 15€

La gestion du paiement n'est pas à implémenter

Précédent Confirmer

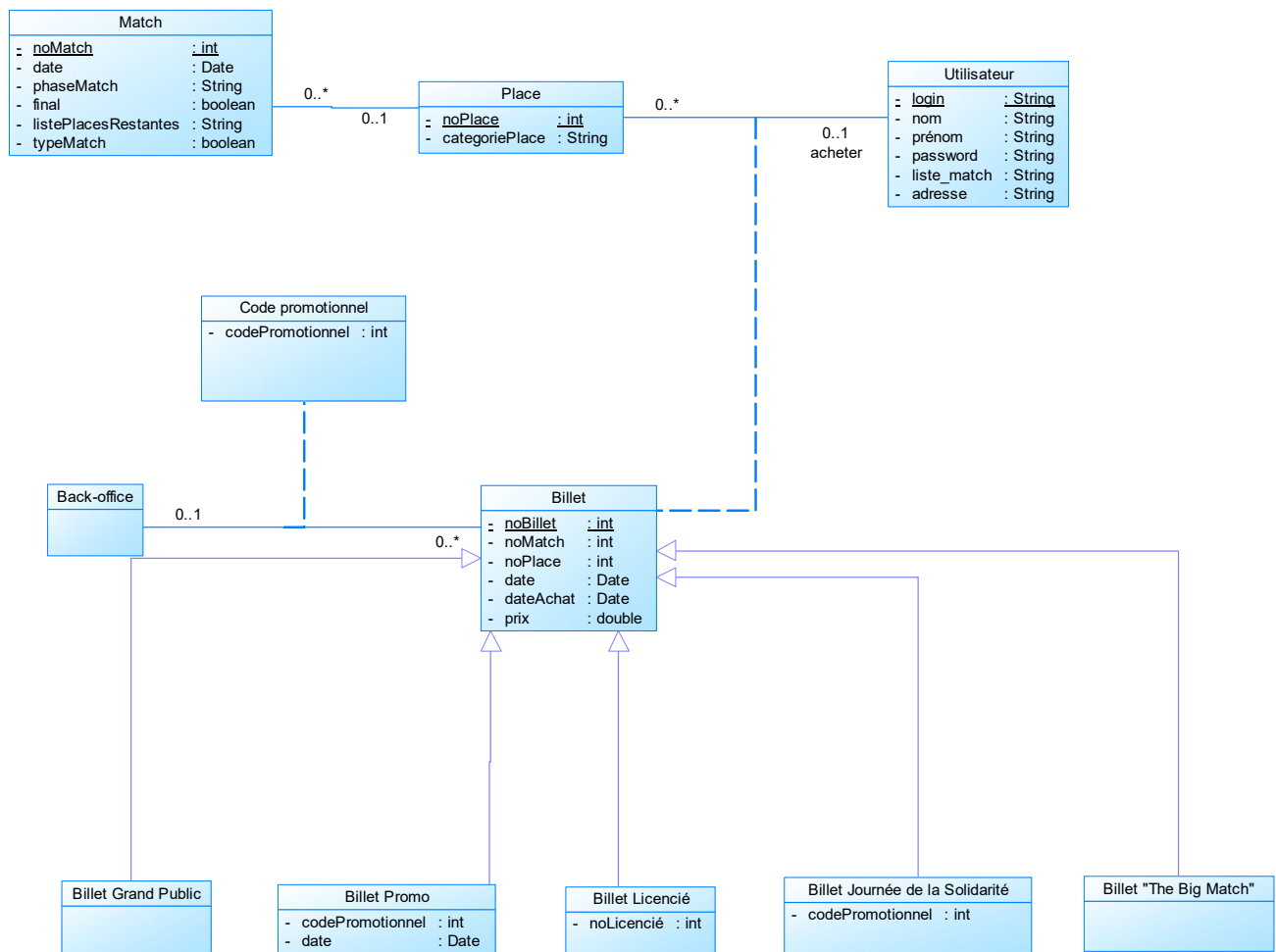
Cette interface est un récapitulatif de la facture qui redirige vers la page de paiement (qui n'est pas à implémenter)



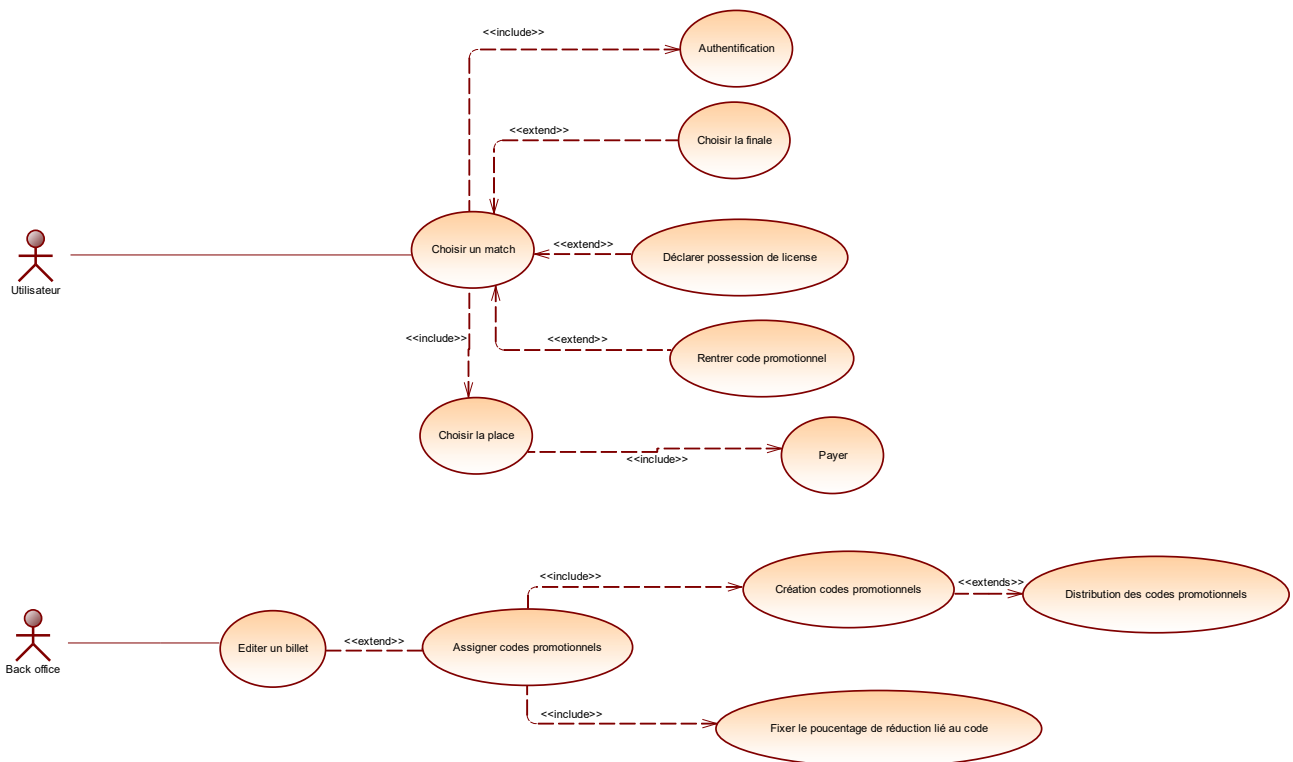
Enfin, l'interface indique le succès du paiement.

3. Package Système de billetterie

a. Diagramme de classe d'analyse



b. Diagramme de cas d'utilisation



L'**Utilisateur** choisit un match qu'il souhaite réserver.

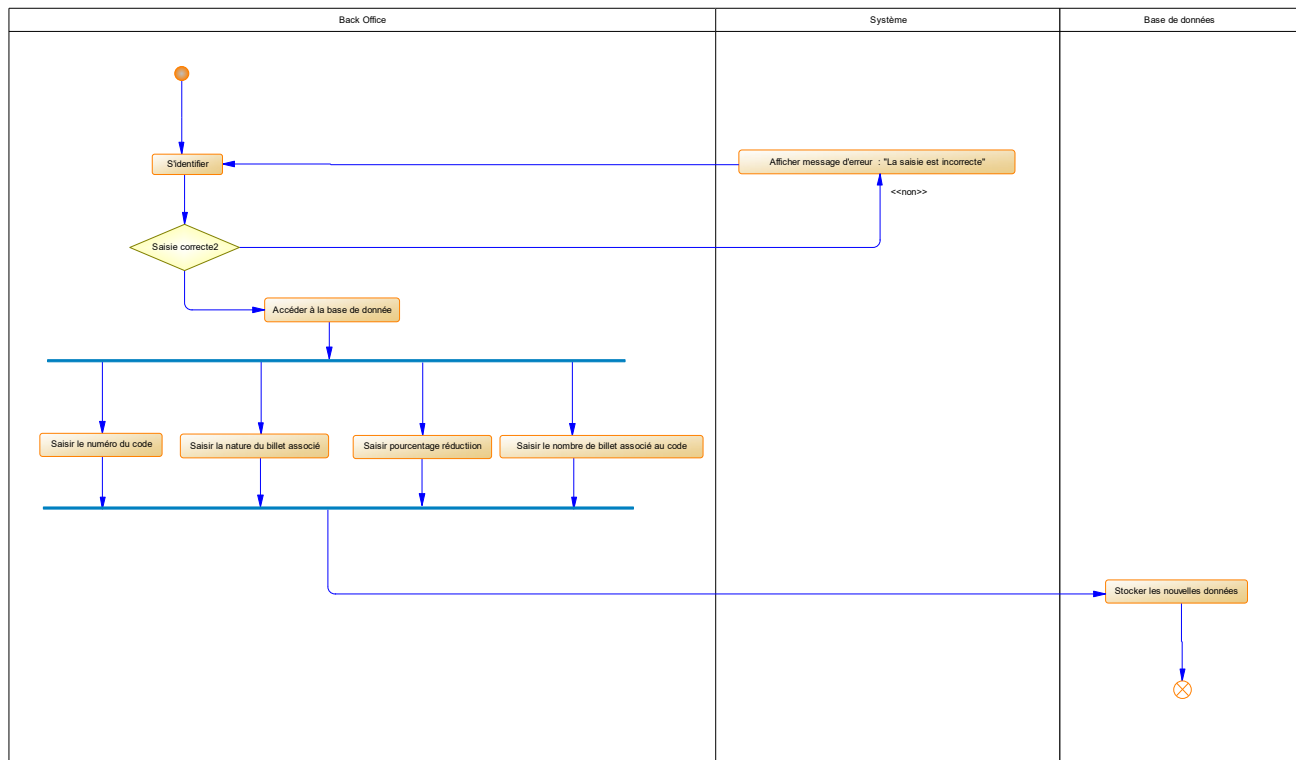
Ce choix inclut le fait de choisir sa place (c'est à dire son emplacement), et cette même action inclut le fait de Payer (le prix dépendant de l'emplacement choisi)

Choisir un match inclus aussi le fait d'identifier ses coordonnées.

Sont traitées ensuite les possibilités de déclarer une possession de licence pour les licenciés de tennis, ou de rentrer le un code promotionnel pour ceux qui en sont dotés.

De l'autre côté, le **Back Office** édite les billets, et assigne éventuellement les codes promotionnels, et les pourcentages de réduction associés aux billets spéciaux.

c. Diagramme d'activité Création des codes promotionnels



Le Back Office s'identifie, et saisit ses identifiants.

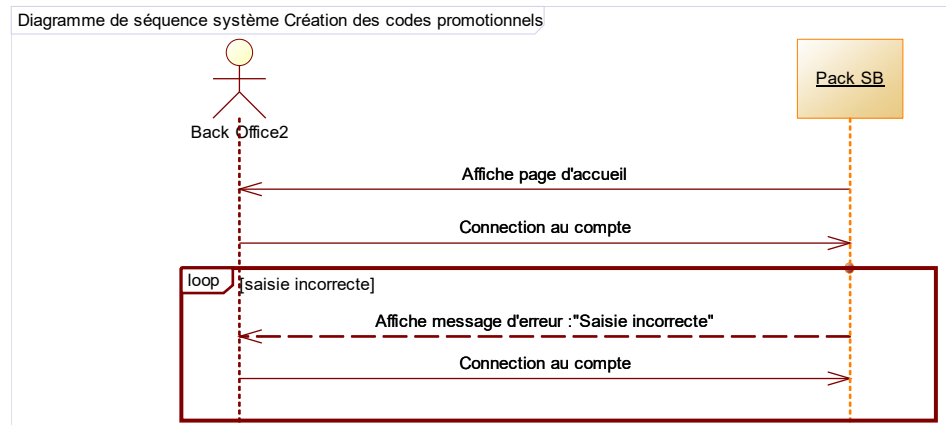
Si la saisie est incorrecte, alors on affiche le message d'erreur.

Lorsque la saisie est correcte, on accède à la base de données.

Il saisit alors simultanément le numéro de code, la nature du billet, le pourcentage de réduction, et le nombre de billets associés à la promotion.

Une fois ces données entrées, elles sont stockées dans la base de données.

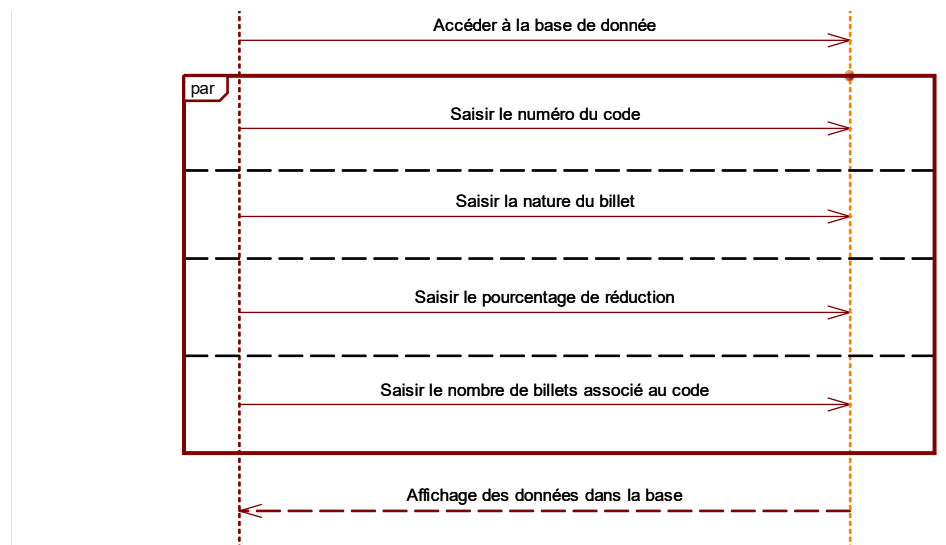
d. Diagramme de séquence système Création des codes promotionnels



La page d'accueil est affichée au Back Office.

Celui-ci se connecte à son compte.

Si les identifiants sont incorrects, alors on boucle jusqu'à ce qu'ils le soient.

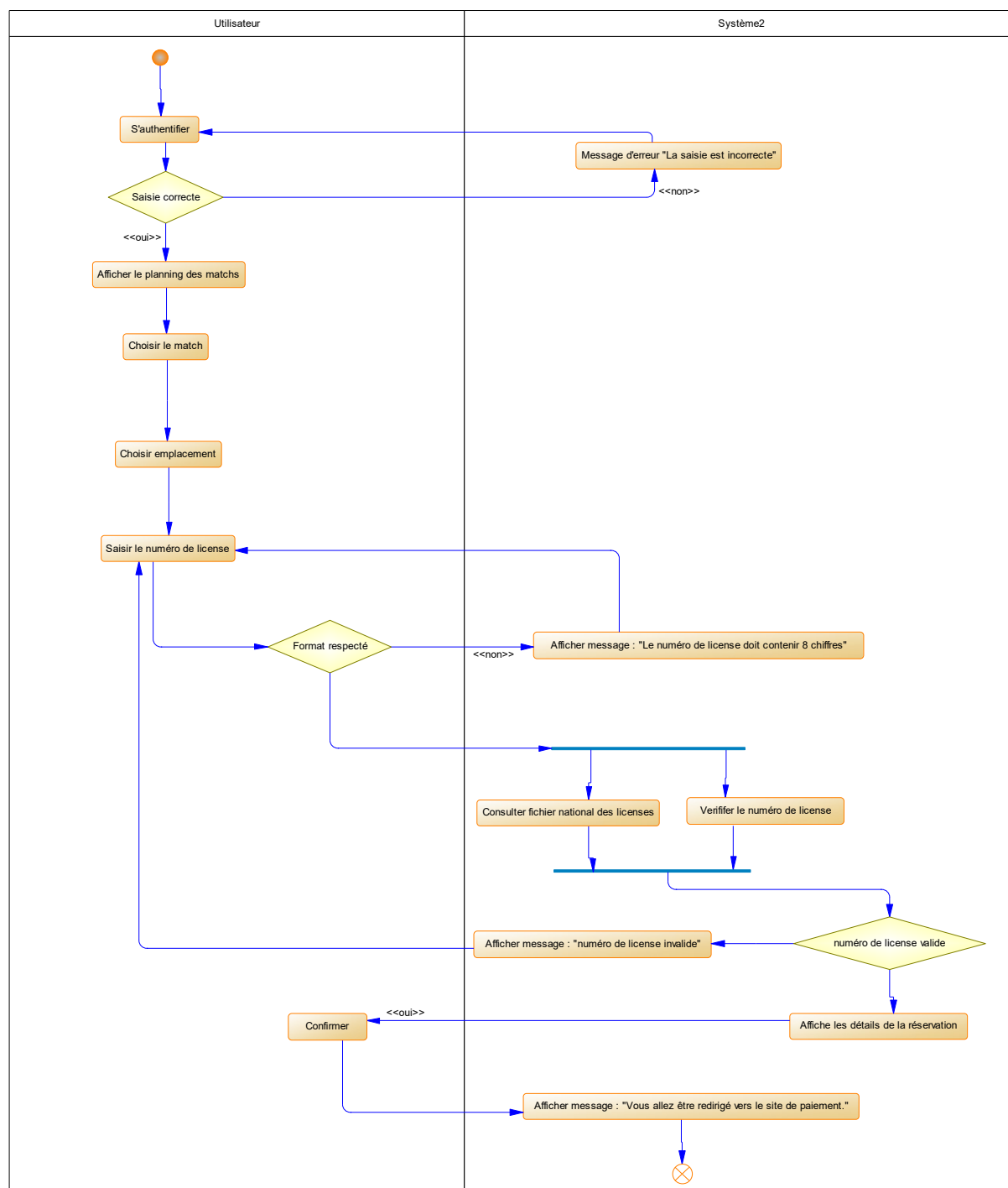


Le Back Office accède alors à la base de données.

Ensuite, il saisit le numéro de code, la nature du billet, le pourcentage de réduction, et le nombre de de billets associés à la promotion.

Puis le Pack SB affiche les données stockées dans la base

e. Diagramme d'activité Déclarer possession



L'utilisateur s'authentifie. Si l'authentification est incorrecte, un message d'erreur apparaît, et ceci tant que l'authentification n'est pas correct.

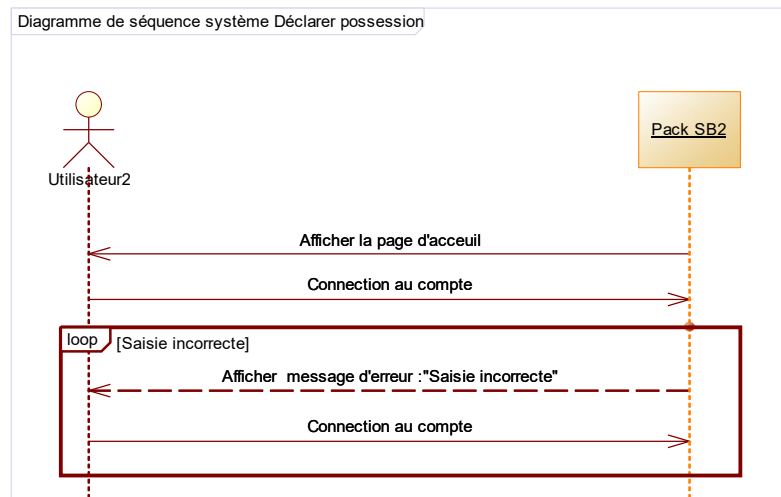
Une fois l'authentification effectuée, l'utilisateur affiche le planning des matchs, choisit le match, choisit l'emplacement souhaité, et saisit son numéro de licence.

Si le numéro n'est pas correctement entré, un message d'erreur est affiché, sinon, le fichier national des licences, et vérifie le numéro entré.

Si le numéro n'est pas valide, un message d'erreur est affiché, et l'utilisateur est invité à entrer son numéro de licence une nouvelle fois.

Si le numéro est correctement entré, les détails de la réservation sont affichées, l'utilisateur confirme, et est redirigé vers la page de paiement.

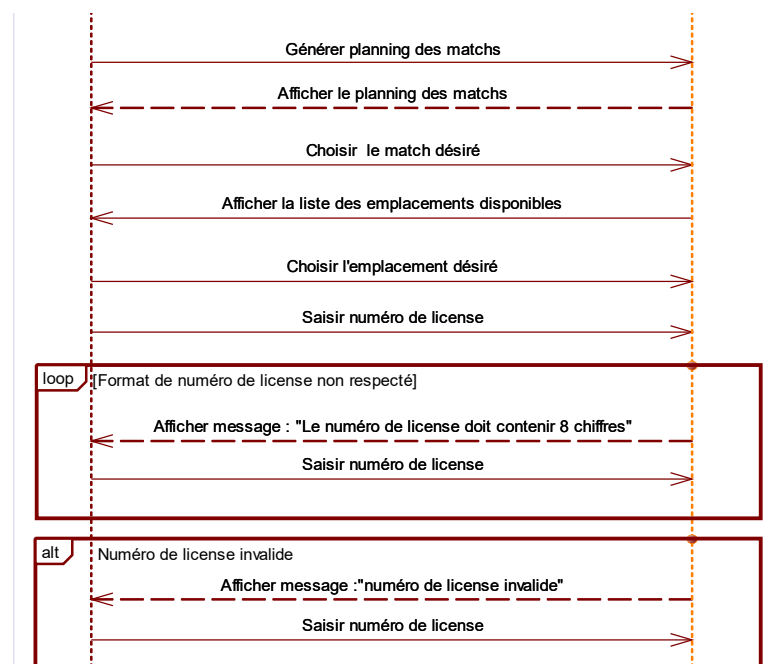
f. Diagramme de séquence système Déclarer possession



La page d'accueil est affichée à l'utilisateur.

Celui-ci se connecte à son compte.

Tant que la saisie est incorrecte, alors un message d'erreur est renvoyé.

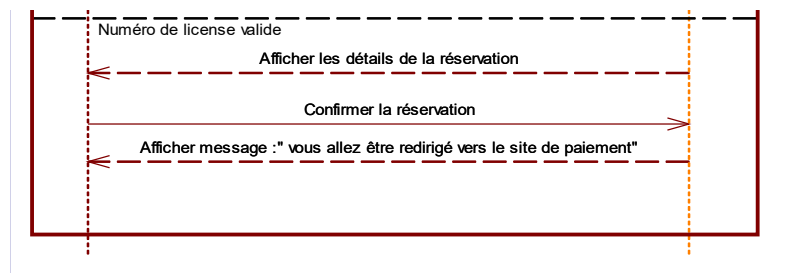


Une fois la saisie validée, l'utilisateur demandera d'afficher le planning des matchs, et le Pack SB le lui affichera.

Ensuite, il choisit le match désiré, et le Pack SB lui affiche la liste des emplacements disponibles.

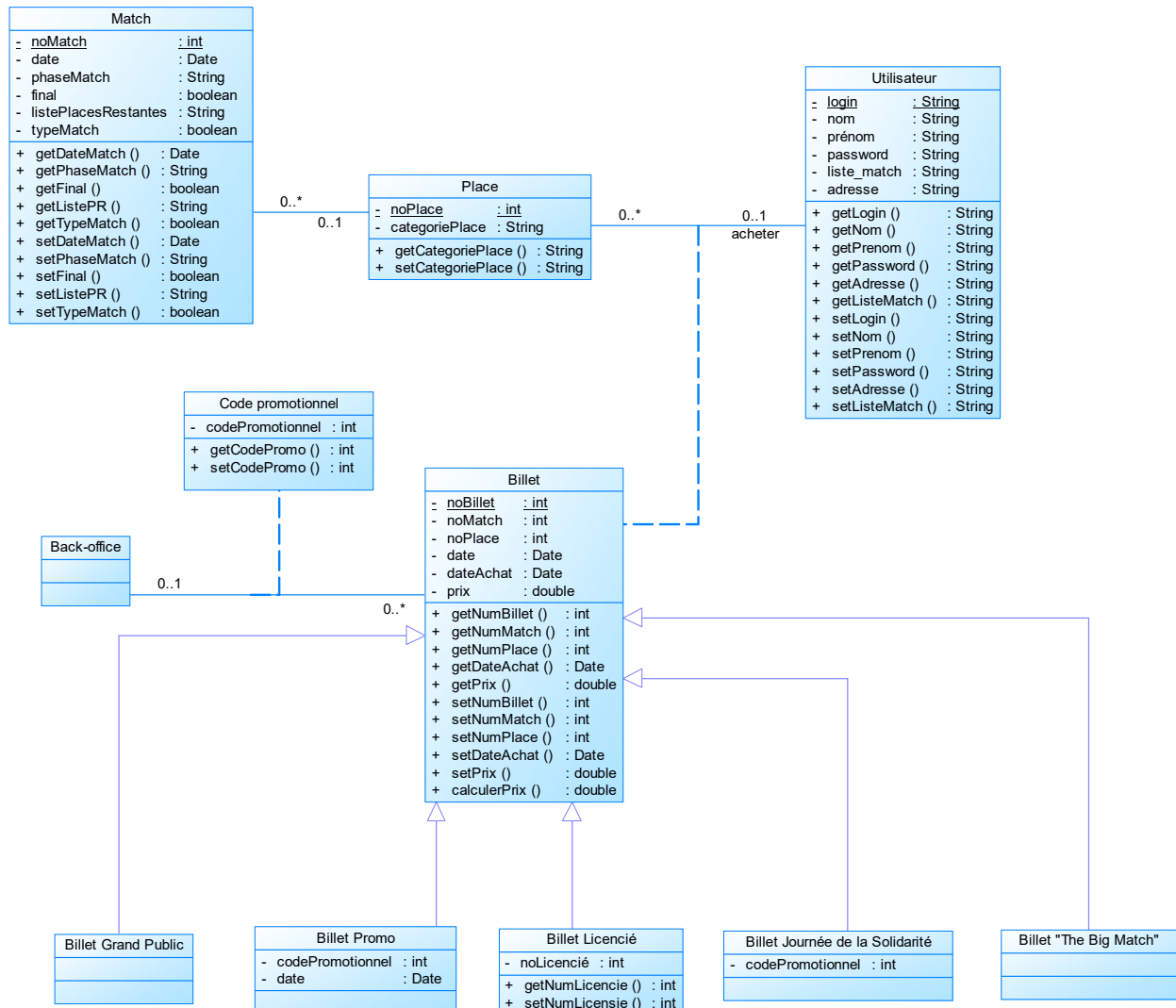
Il choisit l'emplacement désiré, saisit son numéro de licence.

Tant que le numéro de licence est incorrect, alors l'action ne sera pas validée.



Si le numéro de licence est correct, alors le Pack SB affiche le détail de la réservation, l'utilisateur confirme la réservation, et est redirigé vers le site de paiement.

g. Diagramme de classe



La classe **Billet** possède 5 classes filles : **Billet Grand Public**, **Billet Promo**, **Billet Licencié**, **Billet Journée de la Solidarité** et **Billet "The Big Match"**.

La classe **Billet** est dotée des attributs :

- *noBillet* qui sert de clé primaire
- *noMatch* qui est une clé étrangère, qui permet de connaître le match associé au billet
- *noPlace* qui est une clé étrangère qui permet de connaître le numéro de place associé au billet
- *date* qui correspond à la date du match
- *dateAchat* qui correspond à la date d'achat du billet
- *prix* qui correspond au prix d'achat du billet

La classe **Billet Grand Public** est le billet « de base ».

La classe **Billet Promo** a pour attribut :

- *codePromotionnel* qui correspond au code promotionnel associé au billet, et que le client doit entrer dans le formulaire pour avoir une réduction.

La classe **Billet Licencié** a pour attribut supplémentaire :

- *noLicencié* qui correspond au numéro de licence d'une personne licenciée dans un club de tennis, qu'elle doit entrer dans le formulaire pour obtenir une réduction.

La classe **Billet Journée de la Solidarité** avec pour attribut supplémentaire :

- *codePromotionnel* qui correspond au code promotionnel associé au billet, donné par l'association choisit, et que le client doit entrer dans le formulaire pour obtenir une réduction.

Et la classe **Billet “The Big Match”** qui correspond au billet d'une finale.

La classe **Utilisateur** avec les attributs :

- *login* qui sert de clé primaire et que l'utilisateur créé lors de son inscription
- *nom* qui correspond au nom de l'utilisateur
- *prénom* qui correspond au prénom de l'utilisateur
- *password* qui correspond au mot de passe du compte et que l'utilisateur choisit lors de son inscription
- *liste_match* qui correspond à la liste des matchs auxquels l'utilisateur va assister
- *adresse* qui correspond à l'adresse mail de l'utilisateur

La classe **Place** qui est dotée des attributs :

- *noPlace* qui sert de clé primaire
- *categoriePlace* qui permet de préciser la tribune/zone de tribune où se situe la place. Cet attribut est utile puisqu'il fait varier le prix du billet

La classe **Match** est composée des attributs :

- *noMatch* qui sert de clé primaire
- *date* qui correspond à la date du match
- *noCourt* qui est une clé étrangère et qui permet de connaître le court utilisé
- *phaseMatch* qui est un String qui indique la phase du tournoi du match (ex : 1/8 de finale)
- *finale* de type boolean qui s'il s'agit d'une finale
- *listePlacesRestantes* qui est décrémenté de un à chaque place réservée.
- *typeMatch* de type boolean qui indique si le match est Simple ou Double

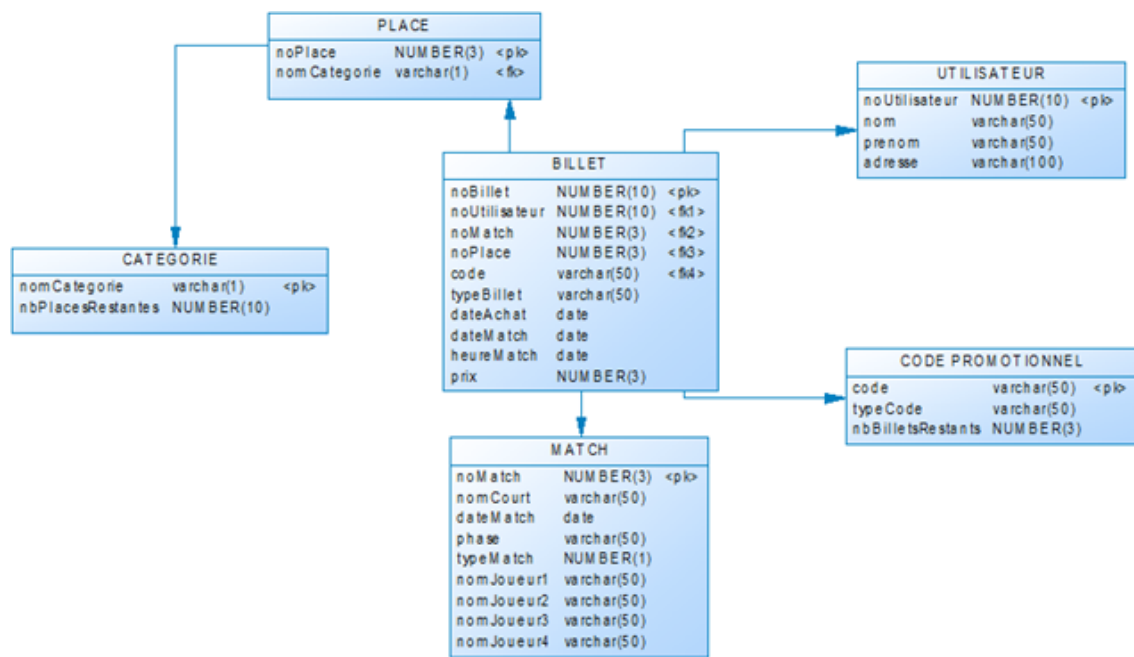
La classe **Code promotionnel** est dotée de l'attribut :

- *codePromotionnel* qui correspond au code promotionnel fixé par le Back-Office.

La classe **Back-Office**, liée à la classe **Billet**, produisent un code promotionnel.

Les méthodes des classes Match, Place, Utilisateur, Code Promotionnel, et Billet Licencié sont des méthodes Get et Set des attributs de ces classes. La méthode calculerPrix() permet de calculer le prix du billet en fonction : de la catégorie de place, du match en question et des éventuelles réductions.

h. Schéma de la base de données (Modèle physique de données)



i. Script de génération de la base de donnée

Création de la table BILLET :

```
/*=====*/
/* Table: BILLET */
/*=====*/
create table BILLET
(
  NOBILLET      NUMBER(10)      not null,
  NOUTILISATEUR NUMBER(10)      null,
  NOMATCH       NUMBER(3)       null,
  NOPLACE       NUMBER(3)       null,
  CODE          varchar(50)      not null,
  TYPEBILLET    varchar(50)      not null,
  DATECHAT      date            null,
  DATEMATCH     date            null,
  HEUREMATCH    date            null,
  PRIX          NUMBER(3)       null,
  constraint PK_BILLET primary key clustered (NOBILLET)
);
```

Création de la table CATEGORIE :

```
/*=====*/
/* Table: CATEGORIE */
/*=====*/
create table CATEGORIE
(
  NOMCATEGORIE  varchar(1)      not null,
  NBPLACESRESTANTES NUMBER(10)  null,
  constraint PK_CATEGORIE primary key clustered (NOMCATEGORIE)
);
```

j. Squelette de classes Java

Exemple de la classe Place :

```

/*****
 * Module: Place.java
 * Author: 1104604
 * Purpose: Defines the Class Place
 *****/

package Systeme_de_billetterie;

import java.util.*;

/** @pdOid dd41c0e5-ff05-448a-a12b-2e6fff4bc5d4 */
public class Place {
    /** @pdOid b30ff536-225c-49c6-8bc5-0c4409bb453e */
    private int noPlace;
    /** @pdOid 6d400f19-6865-4807-a0e3-e061b5a892ef */
    private String categoriePlace;

    /** @pdOid 80f49757-97f9-4dd7-af72-9057d0a791d7 */
    public String getCategoriePlace() {
        // TODO: implement
        return null;
    }

    /** @pdOid 4a5c4970-7a95-43d7-94e8-cd901a3a9f0f */
    public String setCategoriePlace() {
        // TODO: implement
        return null;
    }
}

```

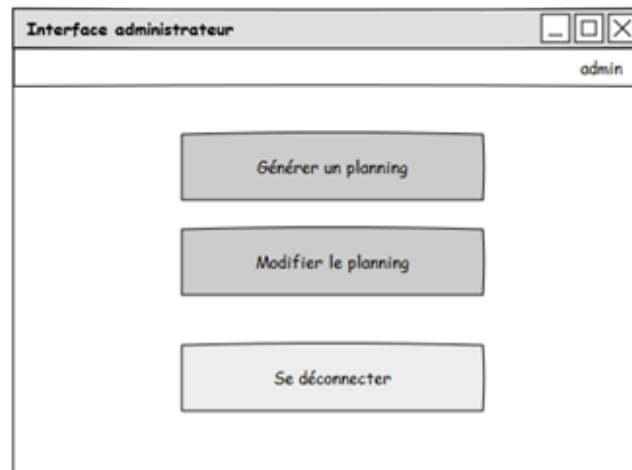
k. Interface Homme-Machine Planning des matchs

Côté administrateur :



A screenshot of a window titled "Authentification". It contains two input fields: "Nom d'utilisateur :" with the text "login" and "Mot de passe :" with masked characters "*****". Below the fields is a button labeled "Connexion".

C'est la page d'authentification dans laquelle l'administrateur entre son nom d'utilisateur et son mot de passe pour pouvoir se connecter.



A screenshot of a window titled "Interface administrateur". The window has a header bar with the text "admin" on the right. The main area contains three buttons stacked vertically: "Générer un planning", "Modifier le planning", and "Se déconnecter".

L'administrateur accède à un menu où il a le choix entre 3 boutons. Il peut soit générer un planning, soit le modifier soit se déconnecter. Lorsqu'il appuie sur « Modifier planning » il accède à une nouvelle fenêtre affichant la liste des matchs à partir du planning généré.

Gestion des matchs (Admin-mode)

Planning GPTL

Mettre à jour le tournoi
Revenir à l'accueil

Dimanche 31

Joueur 1	8h	<table border="1"><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	Détails
Joueur 2																			
Joueur 1	10h	<table border="1"><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	Détails
Joueur 2																			
Joueur 1	12h	<table border="1"><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	Détails
Joueur 2																			
Joueur 1	15h	<table border="1"><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	Détails
Joueur 2																			
Joueur 1	15h	<table border="1"><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	Détails
Joueur 2																			

Là il peut soit, remplir les résultats d'un match fini, en cliquant dans chacune des cases quadrillées. Puis il peut cliquer sur le bouton « Mettre à jour le tournoi » pour que l'application puisse éventuellement à partir des nouveaux résultats générer les prochains matchs du tournoi. Il peut également cliquer sur le bouton « Détails » d'un match pour faire apparaître une fenêtre contenant des informations spécifiques au match.

Détails match Joueur1 vs Joueur2 - Date (admin-mode)

Terrain : Nom terrain Date du match :

Joueur 1 : Nom Prénom Joueur 1

Joueur 2 : Nom Prénom Joueur 2

Arbitre de chaise : Nom Prénom Arbitre de chaise

Juge de ligne :

Juge de Ligne 1
Juge de Ligne 2
Juge de Ligne 3
Juge de Ligne 4

 Modifier

Ramasseurs de balles :

Ramasseur de balle 1
Ramasseur de balle 2
Ramasseur de balle 3
Ramasseur de balle 4

 Retour

Détails match Joueur1-Joueur2 vs Joueur3-Joueur4 - Date (admin-mode)

Terrain : Nom terrain Date du match :

Binôme 1 : Nom Prénom Joueur 1 - Joueur 2

Binôme 2 : Nom Prénom Joueur 3 - Joueur 4

Arbitre de chaise : Nom Prénom Arbitre de chaise

Juge de ligne :

Juge de Ligne 1
Juge de Ligne 2
Juge de Ligne 3
Juge de Ligne 4

 Modifier

Ramasseurs de balles :

Ramasseur de balle 1
Ramasseur de balle 2
Ramasseur de balle 3
Ramasseur de balle 4

 Retour

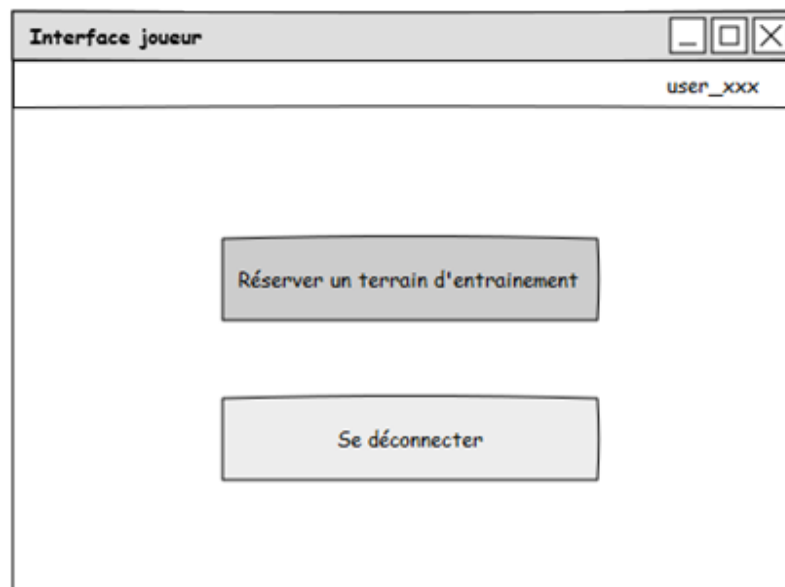
L'administrateur peut également modifier les détails du match en cliquant sur le bouton « Modifier ».

Côté joueur :



A screenshot of a window titled "Authentification". It contains two input fields: "Nom d'utilisateur :" with the text "login" and "Mot de passe :" with a masked password "*****". Below the fields is a button labeled "Connexion".

C'est la fenêtre d'authentification dans laquelle le joueur entre son nom d'utilisateur et son mot de passe pour pouvoir se connecter.



A screenshot of a window titled "Interface joueur". The title bar also shows "user_XXX". The main area contains two buttons: "Réserver un terrain d'entraînement" and "Se déconnecter".

Le joueur accède à un menu où il a le choix entre 2 boutons. Il peut soit réserver un terrain d'entraînement, soit se déconnecter. Lorsqu'il appuie sur « Réserver un terrain d'entraînement » il accède à une nouvelle fenêtre.

Réservation terrain d'entraînement - Grand Prix de Tennis de Lyon

Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
25	26	27	28	29	30	31
1	2	3	4	5	6	7

8h
11h
15h
18h
21h

Terrains disponibles :

Cours d'entraînement n°1

Cours d'entraînement n°3

Cours d'entraînement n°5

! Êtes vous sûr de vouloir réserver le Cours d'entraînement n°1 le Mardi 2 à 11h ?

Le joueur peut réserver un terrain d'entraînement en cliquant sur un des jours de la semaine, puis sur un créneau d'horaires. Et enfin, en cliquant sur le bouton « Réserver » des éventuels cours libres qui sont apparues dans le bas de la fenêtre. Une boîte de dialogue apparaît pour demander au joueur de confirmer sa réservation.

