

Rapport d'Analyse – CPOA Généré avec PowerAMC

Modèle: CPOA_OpenParc

Rapport: Analyse – Projet Open Parc Lyon

Auteur: Antonin DEHU, Manoël NOHRA,
Noah BLANCHARD, Jolan THOMASSIN

Version: 1

Date: 17/12/2021

Résumé: Rapport d'analyse du projet d'application au service
de l'Open Parc de Lyon 2022.
(Diagrammes, Scénarios...)



Table des matières

I	INTRODUCTION.....	3
I.1	DESCRIPTION.....	3
II	DESCRIPTION COMPLÈTE DU MODÈLE.....	4
II.1	OBJETS DE NIVEAU MODÈLE	4
II.1.1	Liste des diagrammes.....	4
II.1.2	Diagramme 1CPOA_Tennis	4
II.2	PACKAGE BILLETTERIE	4
II.2.1	Liste des diagrammes.....	4
II.2.2	Diagramme 2aDCU_Billetterie	5
II.2.3	Diagramme 2bDIT_AnnulerPanier	5
II.2.4	Diagramme 2bDIT_Choisir les billets	6
II.2.5	Diagramme 2bDIT_ValiderPanier	7
II.2.6	Diagramme 2cDSQ_Choisir les billets	8
II.2.7	Diagramme 2dDC_Billetterie	9
II.2.8	Diagramme de Classe de Conception	10
II.3	PACKAGE PLANNING	11
II.3.1	Liste des diagrammes.....	11
II.3.2	Diagramme 2aDCU_Planning	12
II.3.3	Diagramme 2bDIT_ModifPlanning	13
II.3.4	Diagramme 2bDIT_ReserverCourt.....	14
II.3.5	Diagramme 2bDIT_SaisirScore.....	15
II.3.6	Diagramme 2cDSQ_GenererPlanningQualif.....	16
II.3.7	Diagramme 2cDSQ_GererUnMatch	17
II.3.8	Diagramme 2dDCL_Planning	18
II.3.9	Diagramme de Classe de Conception	19
III	GÉNÉRATION DE CODE AVEC POWERAMC.....	20
III.1	GÉNÉRATION DE CODE JAVA	20
III.1.1	Classe Billet.java	20
III.1.2	Classe Match.java	22
III.2	GÉNÉRATION DE TABLE SQL.....	25
III.2.1	Table Billet	25
III.2.2	Table Match	27
IV	MAQUETTE	29
IV.1	MAQUETTE DE L'APPLICATION PLANNING	29
IV.1.1	Page de Lancement.....	29
IV.1.2	Espace Administrateur.....	30
IV.1.3	Sélection d'un match et modification d'un match.....	31
IV.1.4	Génération et consultation des plannings.....	32
IV.1.5	Espace Joueur.....	33
IV.1.6	Réservation de court et consultation des réservations	34
IV.2	MAQUETTE DE L'APPLICATION BILLETTERIE.....	35
IV.2.1	Sélection du billet voulu.....	35
IV.2.2	Sélection de la catégorie du billet.....	36
IV.2.3	Récapitulatif du panier et bons de réductions.....	37
IV.2.4	Simulation de paiement par carte	38

I Introduction

I.1 Description

Ce rapport fait l'objet du résultat de l'analyse d'un projet d'application au service de l'événement de tennis Open Parc de Lyon. Cet événement regroupe deux tournois, double et simple, ainsi qu'une phase de qualifications pour le tournoi simple.

Il s'agit dans ce rapport de rendre compte de l'analyse de deux modules de l'application finale : **le Planning**, et **la Billetterie**.

- **Planning**

Le but de ce module est d'offrir une solution unifiée et automatique de generation du planning. Les responsables de l'événements pourront accéder à une interface ergonomique qui leur permettra de générer un planning en fonction de different critères. Ce planning sera bien évidemment à valider par les responsables qui pourront le retoucher et y apporter les modifications nécessaires.

Le module proposera aussi d'autres fonctionnalités comme la saisie des scores pour les matchs terminés, et la reservation d'un court libre par un joueur souhaitant s'entraîner.

- **Billetterie**

Le but de ce module est de mettre à disposition des potentiels visiteurs d'accéder aux billets rapidement sur une interface Web. Une liste des billets, de leur catégorie, de leur date de validité sera proposée aux clients qui pourront alors facilement sélectionner le billet leur convenant le mieux. Le paiement se fera sur le site via une interface de paiement sécurisée.

Le principe fondamentale de ce projet sera l'existence d'une base de données qui unifiera ces deux modules afin qu'ils communiquent entre eux. En effet, les plannings et matchs organisés dans le premier module seront visibles dans le second.

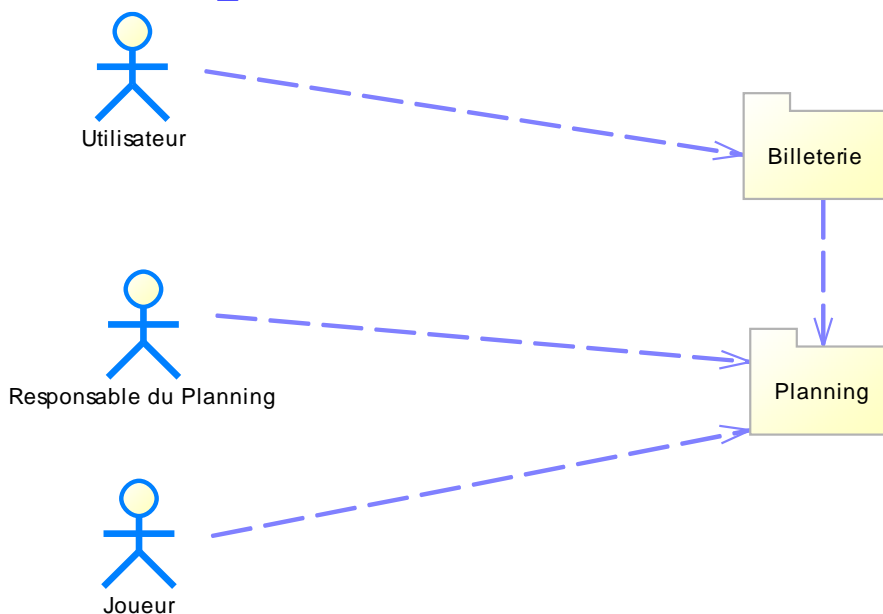
II Description complète du modèle

II.1 Objets de niveau modèle

II.1.1 Liste des diagrammes

Nom	Code
1CPOA_Tennis	1CPOA_Tennis

II.1.2 Diagramme 1CPOA_Tennis



II.2 Package Billeterie

Scénario général :

L'utilisateur va sur l'application web et choisi sont billet et l'ajoute à son panier pour ensuite le valider et le payer directement sur le site, il pourra ajouter un code promo ou/et son numéro de licence s'il veut obtenir une réduction.

L'utilisateur peut annuler tout son panier ou alors supprimer un billet non voulu ou plus voulu de son panier

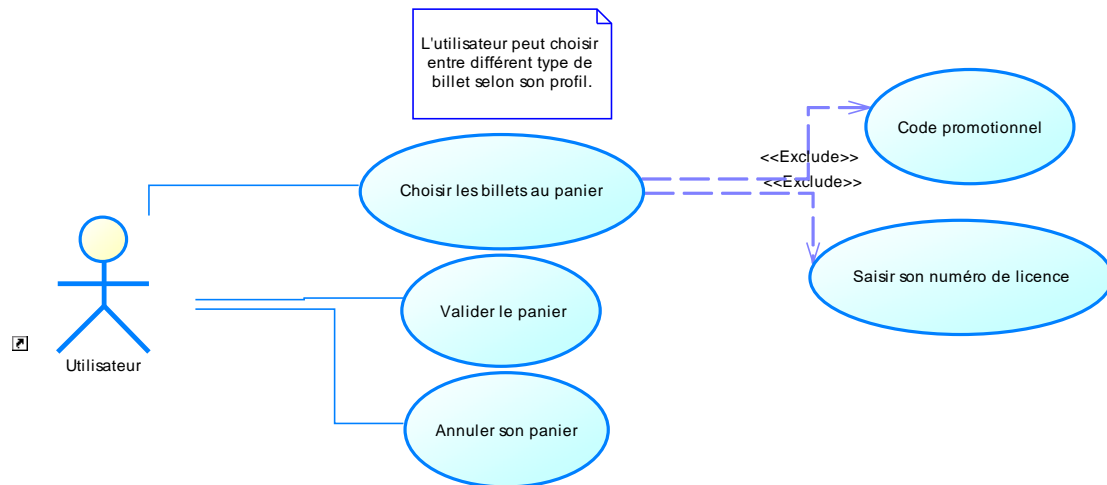
II.2.1 Liste des diagrammes

Nom	Code
2aDCU_Billeterie	2aDCU_Billeterie
2bDIT_AnnulerPanier	2bDIT_AnnulerPanier
2bDIT_Choisir les billets	2bDIT_Choisir les billets
2bDIT_ValiderPanier	2bDIT_ValiderPanier

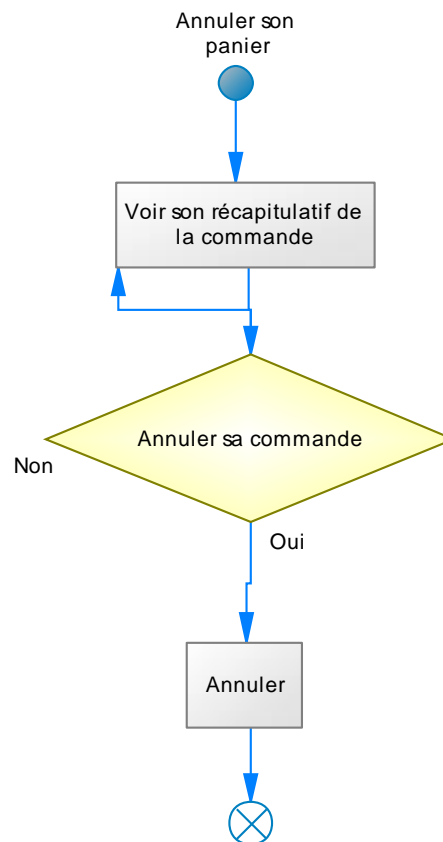
2cDSQ_Choisir les billets
2dDC_Billetterie

2cDSQ_Choisir_les_billets
2dDC_Billetterie

II.2.2 Diagramme 2aDCU_Billetterie

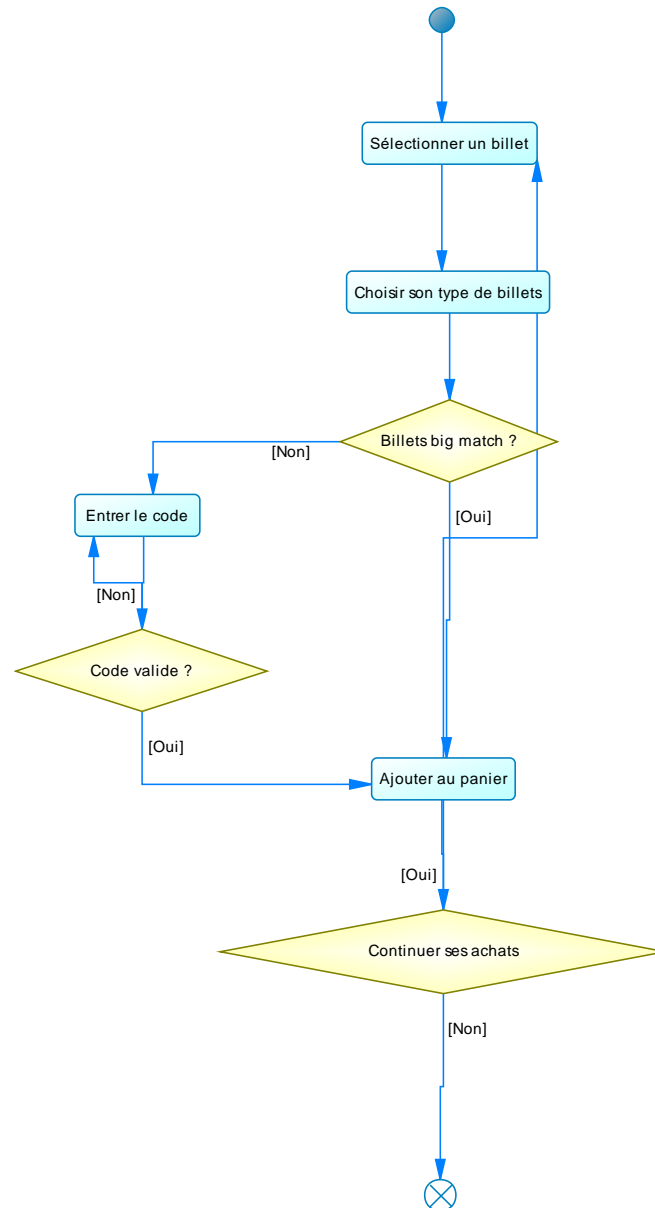


II.2.3 Diagramme 2bDIT_AnnulerPanier

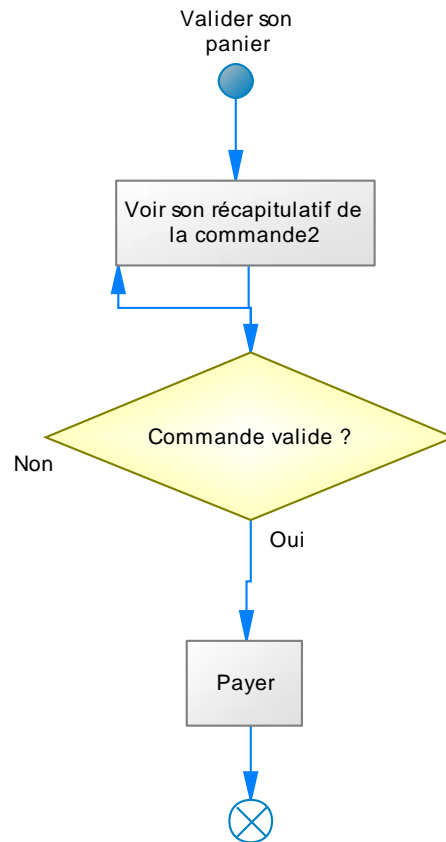


Scénario :

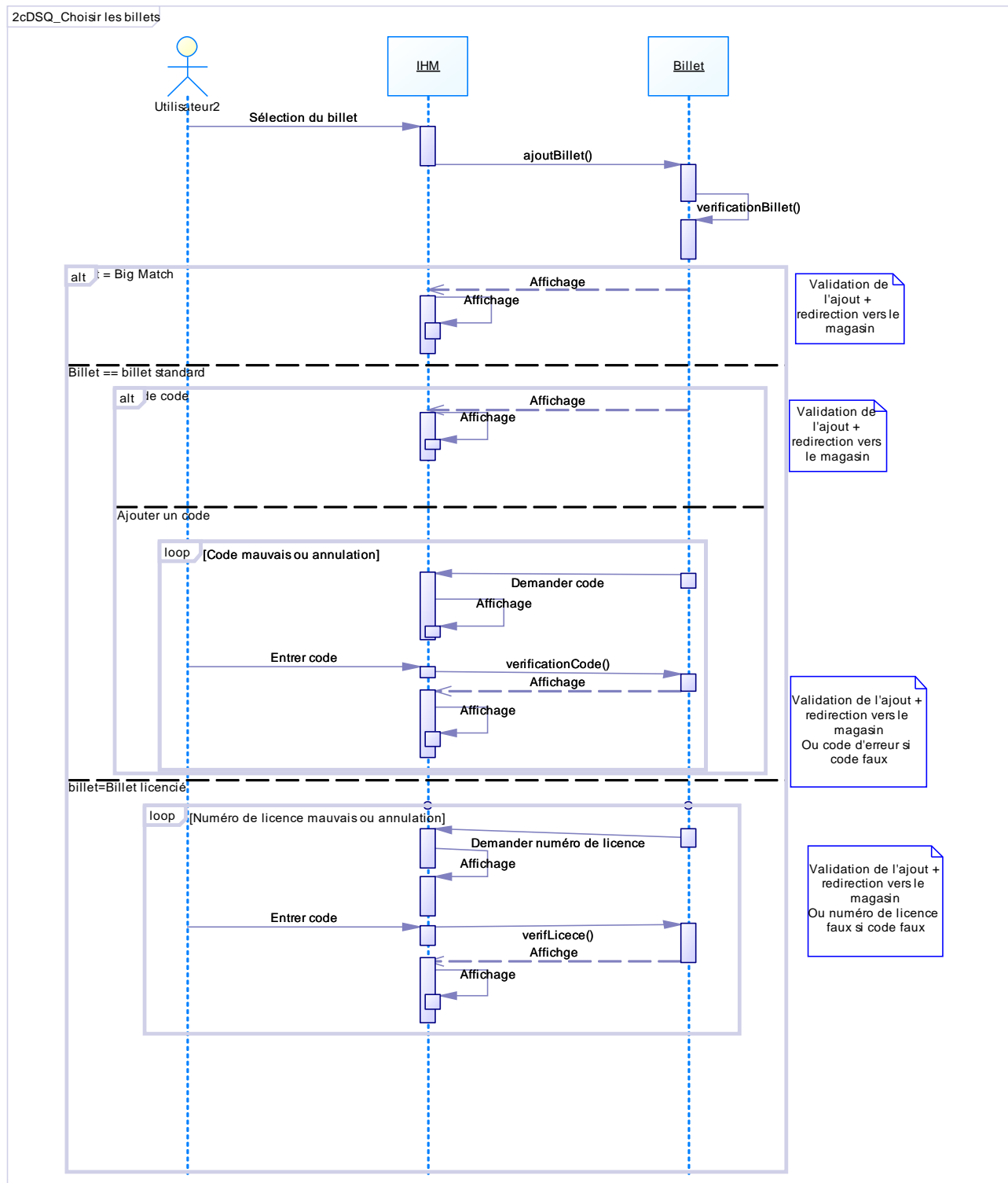
II.2.4 Diagramme 2bDIT_Choisir les billets



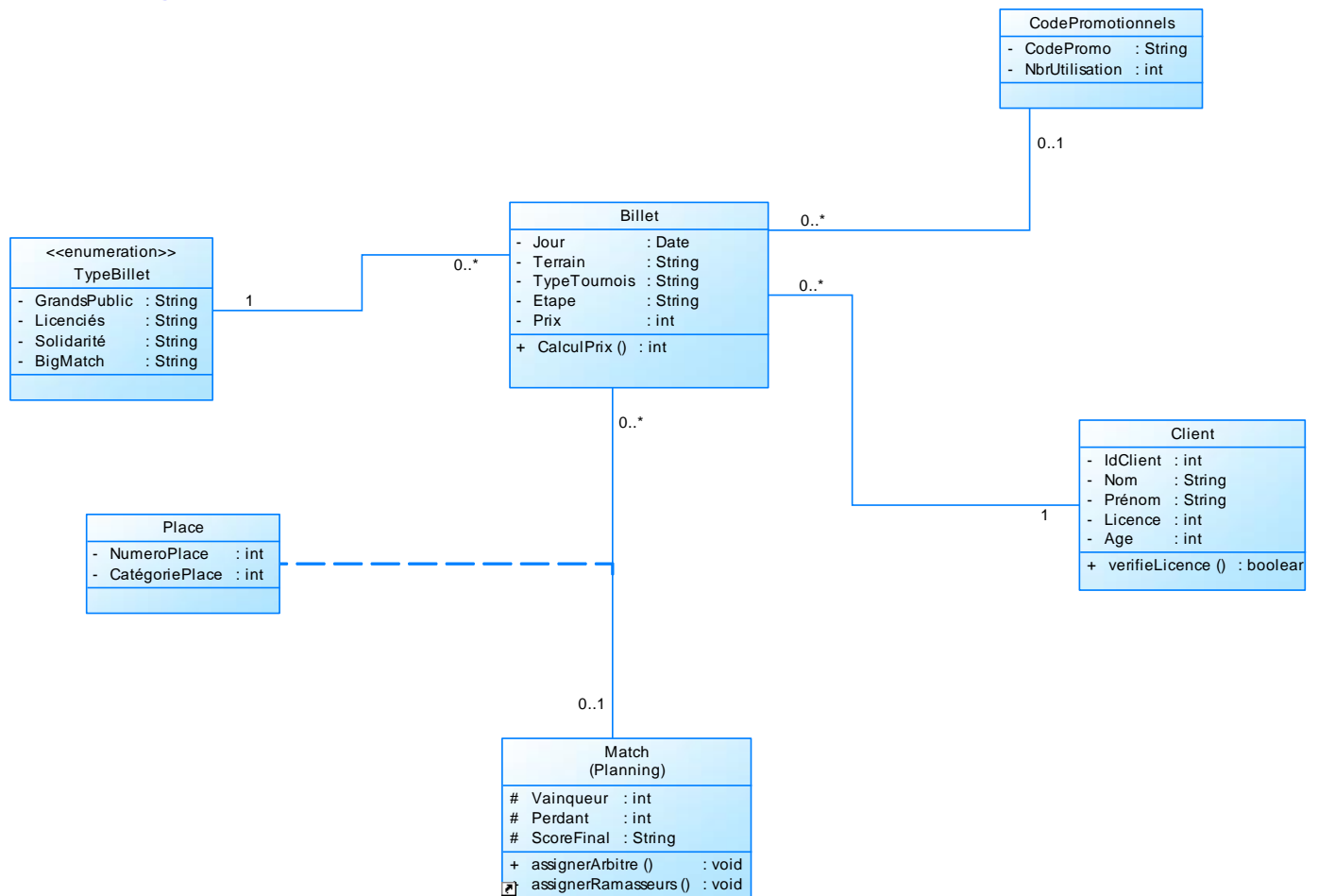
II.2.5 Diagramme 2bDIT_ValiderPanier



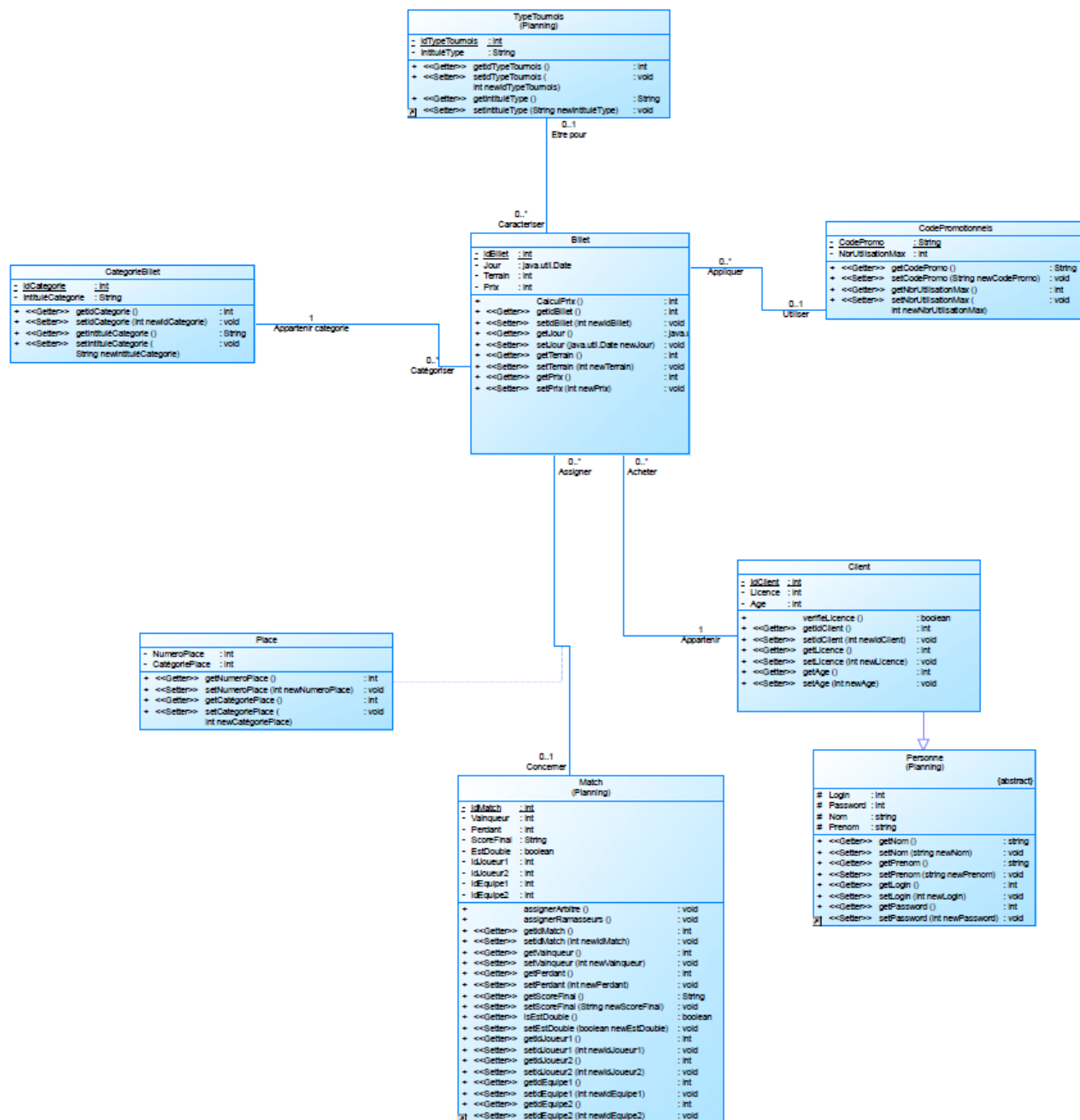
II.2.6 Diagramme 2cDSQ_Choisir les billets



II.2.7 Diagramme 2dDC_Billeterie



II.2.8 Diagramme de Classe de Conception



II.3 Package Planning

Scénario general :

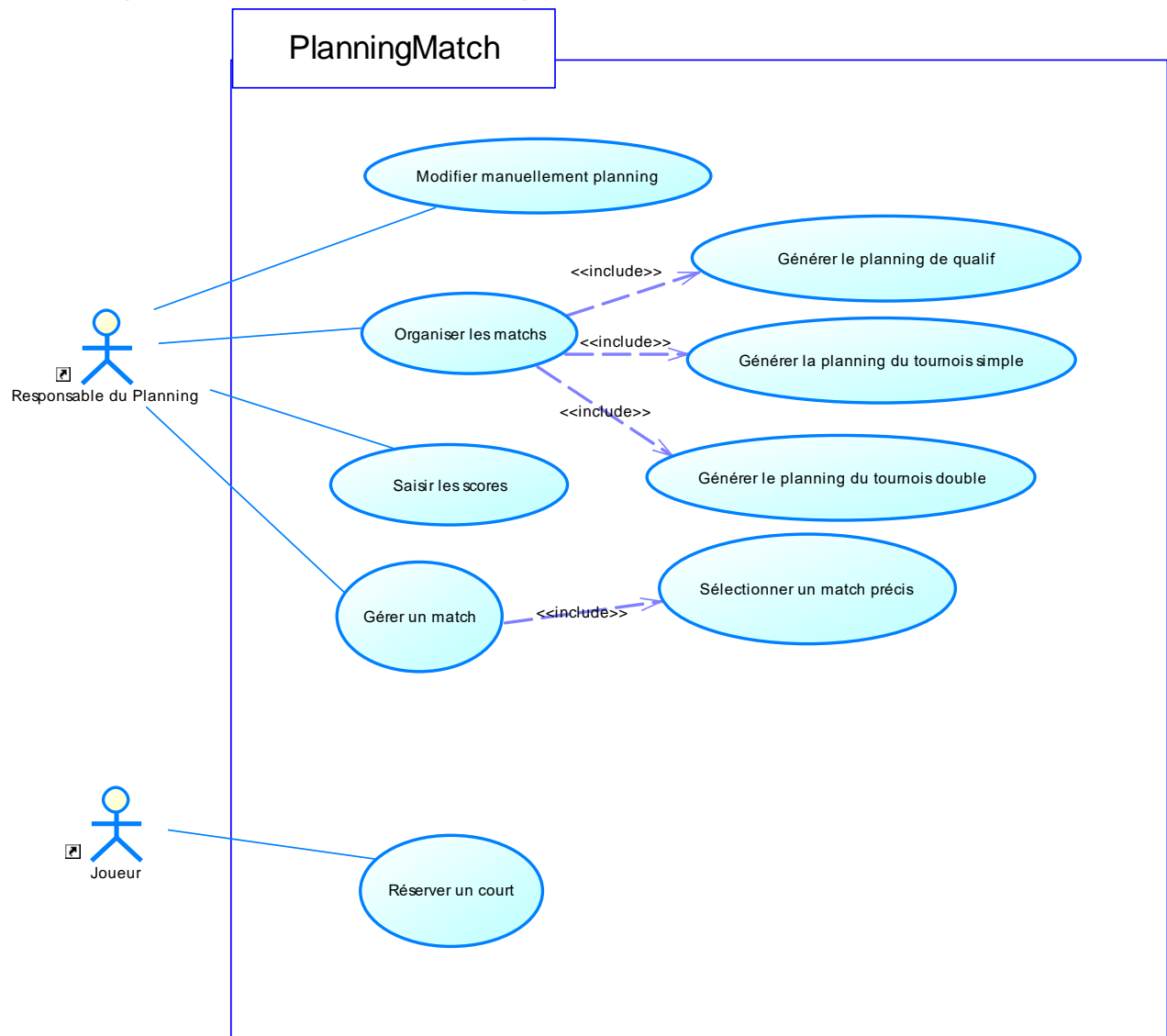
Le **responsable** de planning s'identifie sur l'application. Il peut modifier manuellement le planning, organiser les matchs, saisir les scores, ou alors gérer le match si nécessaire. L'application permet aussi une generation automatique des plannings.

Le **joueur** peut réserver un court pour s'entrainer ou annuler sa réservation après s'être identifié sur le site.

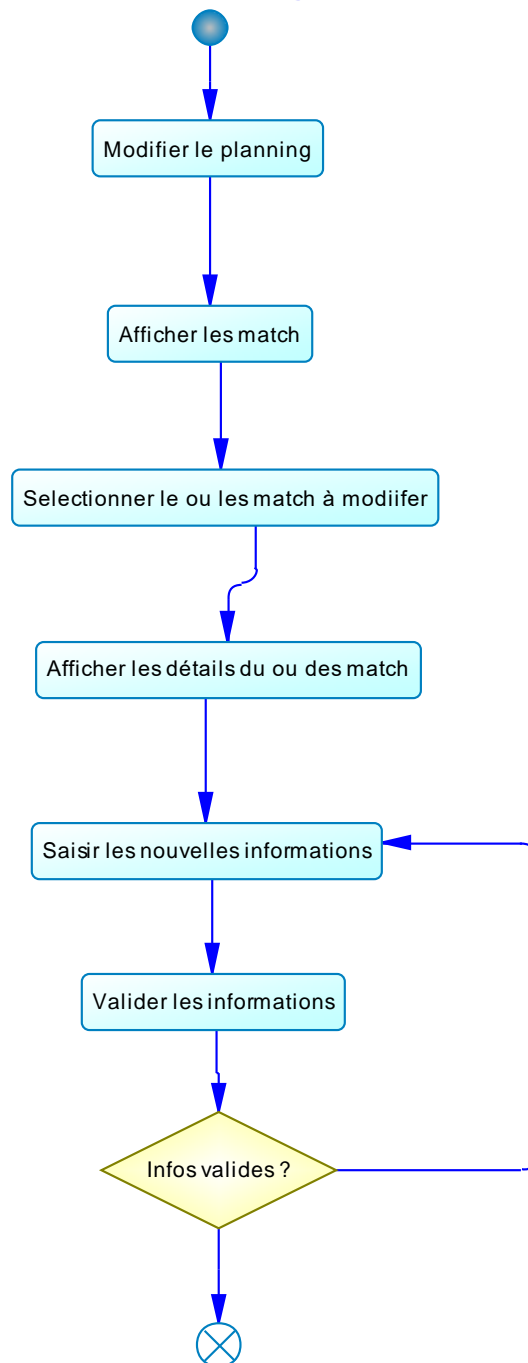
II.3.1 Liste des diagrammes

Nom	Code
2aDCU_Planning	2aDCU_Planning
2bDIT_ModifPlanning	2bDIT_ModifPlanning
2bDIT_ReserverCourt	2bDIT_ReserverCourt
2bDIT_SaisirScore	2bDIT_SaisirScore
2cDSQ_GenererPlanningQualif	2cDSQ_GenererPlanningQualif
2cDSQ_GererUnMatch	2cDSQ_GererUnMatch
2dDCL_Planning	2dDCL_Planning

II.3.2 Diagramme 2aDCU_Planning

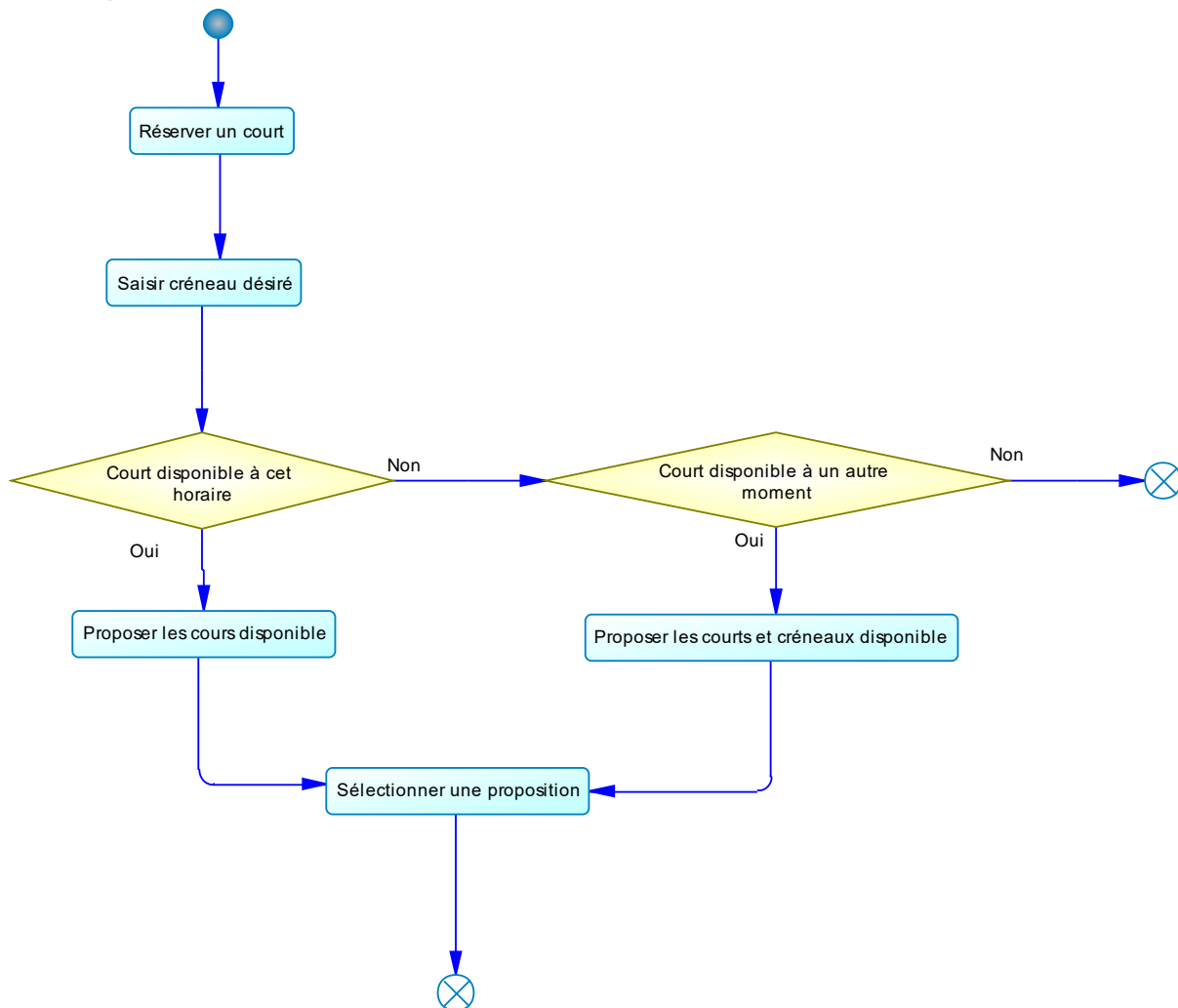


II.3.3 Diagramme 2bDIT_ModifPlanning



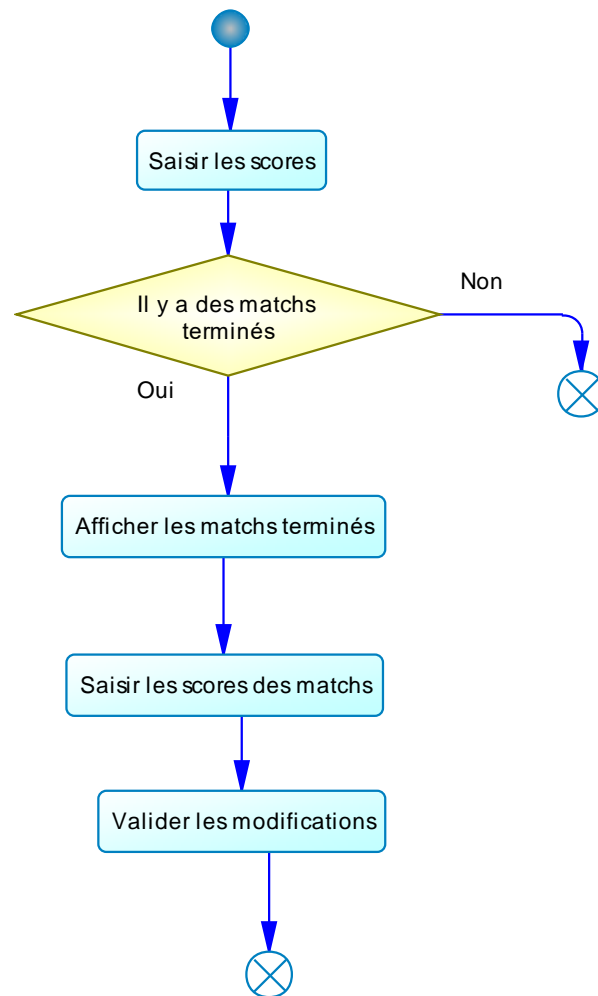
Scénario : le responsable du planning est dans le besoin de modifier l'un des plannings préalablement généré par l'application. Il peut alors se rendre sur l'application, afficher la liste de tous les matchs et en sélectionner un dont il souhaite modifier les informations. Par exemple, en cas de pluie, il pourrait être amené à devoir repousser un match.

II.3.4 Diagramme 2bDIT_ReserverCourt



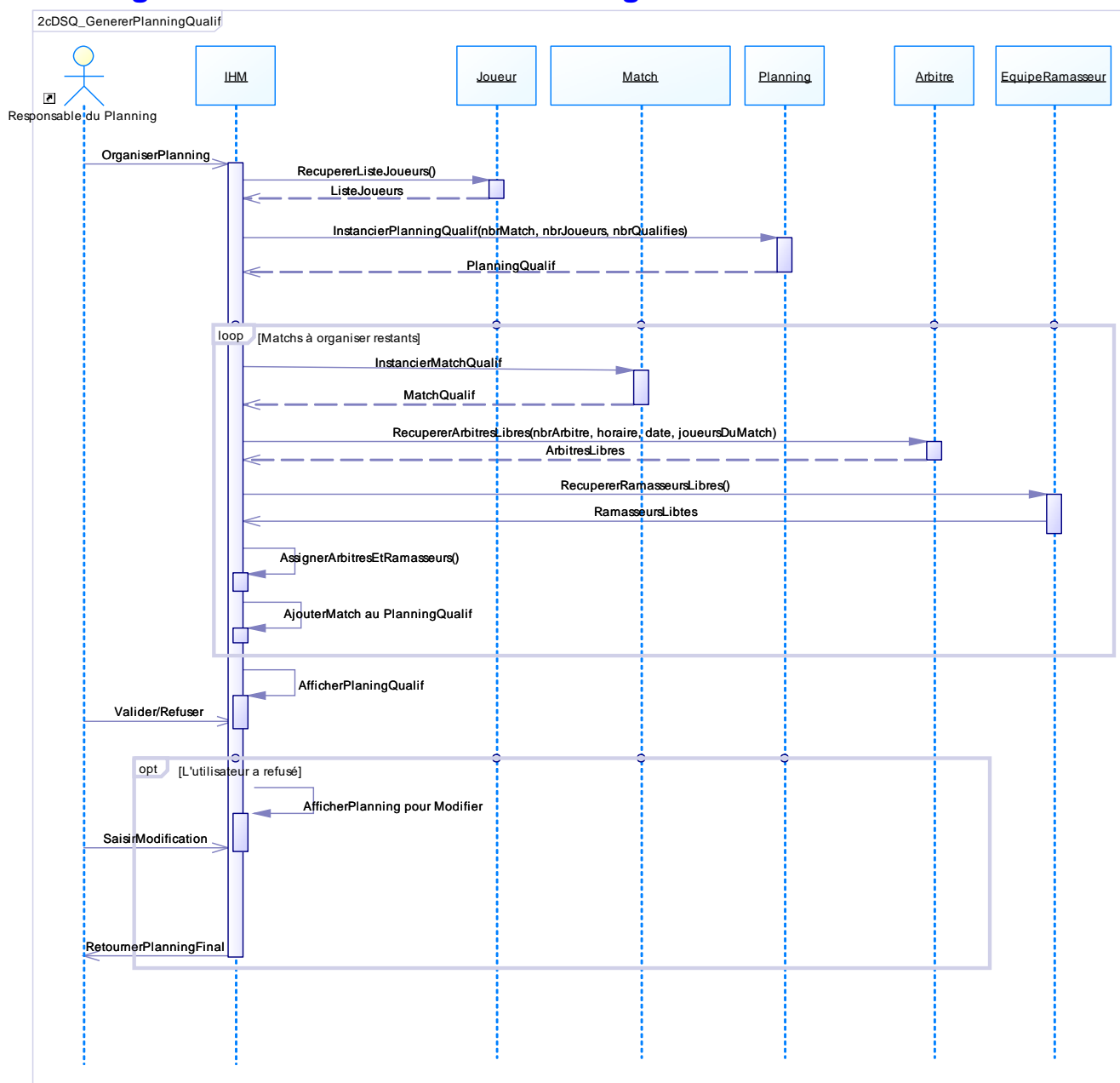
Scénario : les différents joueurs pourront tout au long de l'événement réserver l'un des courts pour s'y entraîner. L'application propose un espace réservé à ces joueurs pour que ceux-ci puissent réserver un court, consulter leurs réservations, les annuler... Ici, un joueur X se rend sur l'application, entre son identifiant unique et accède à son espace de reservation. La recherche de court disponible se fait par date et par créneau, et l'application renvoie tous les courts disponibles aux alentours de l'horaire demandé.

II.3.5 Diagramme 2bDIT_SaisirScore



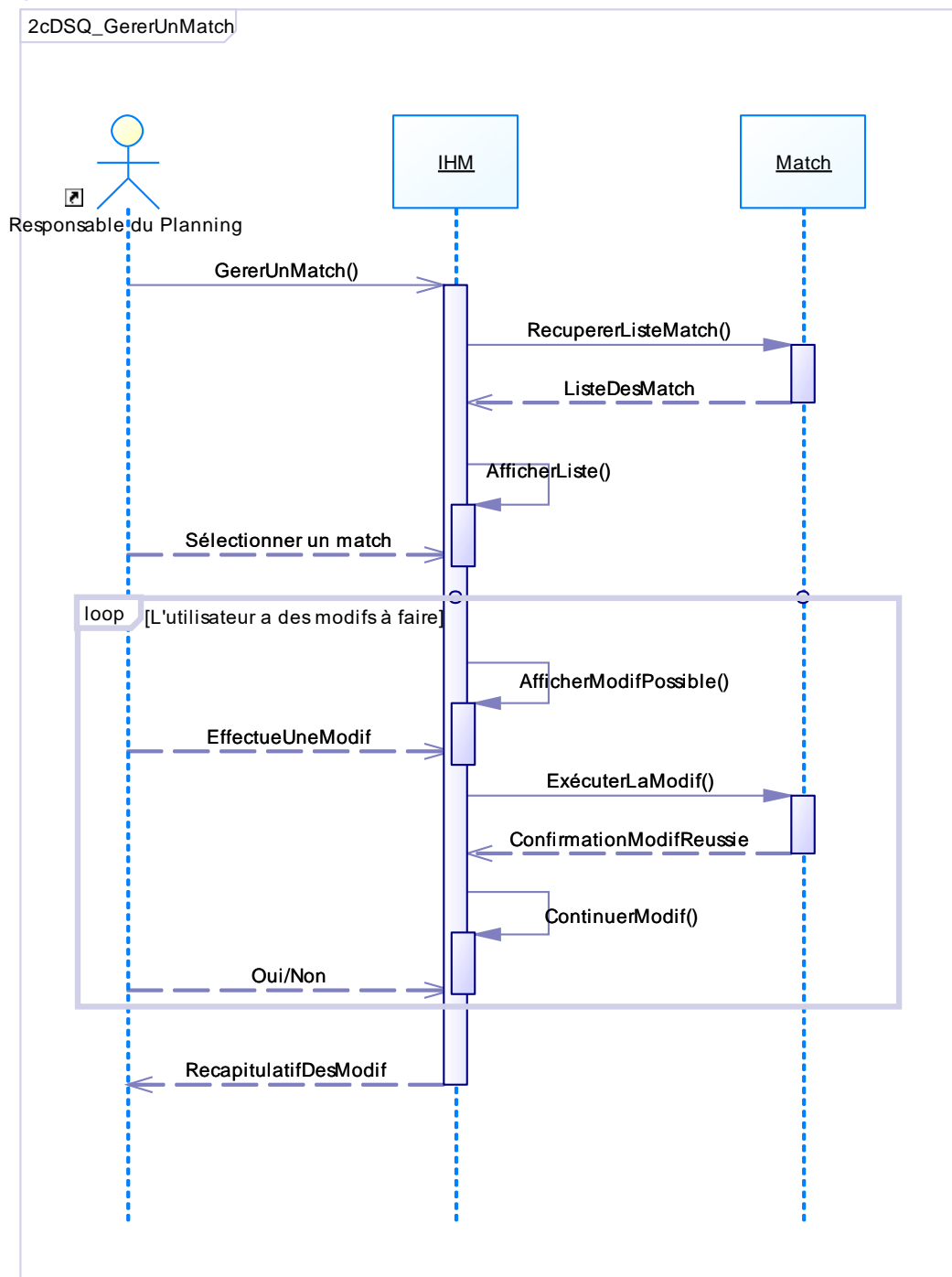
Scénario : l'application permet aussi la saisie des scores par le responsable. En effet, lorsqu'un match se termine, le responsable Y saisit le résultat du match. L'application détermine alors le vainqueur et le perdant.

II.3.6 Diagramme 2cDSQ_GenererPlanningQualif



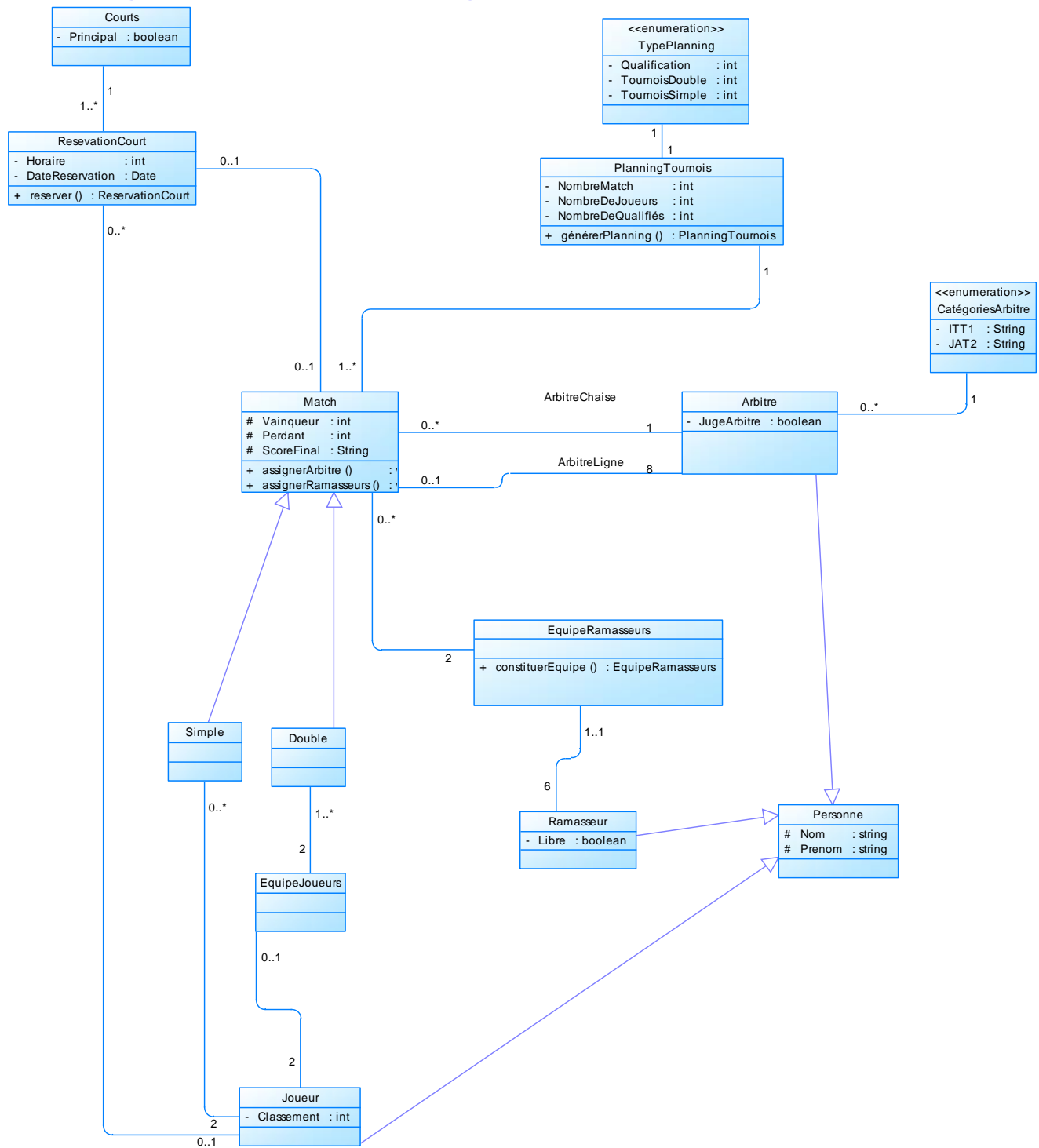
Scénario : en début d'événement, le responsable Y doit, à l'aide de l'application générer les plannings pour les trois tournois. L'application s'en occupe automatiquement. Cependant, certaines entrées peuvent ne pas plaire au responsable Y qui est dans la capacité de revoir le planning avant de le valider.

II.3.7 Diagramme 2cDSQ_GererUnMatch

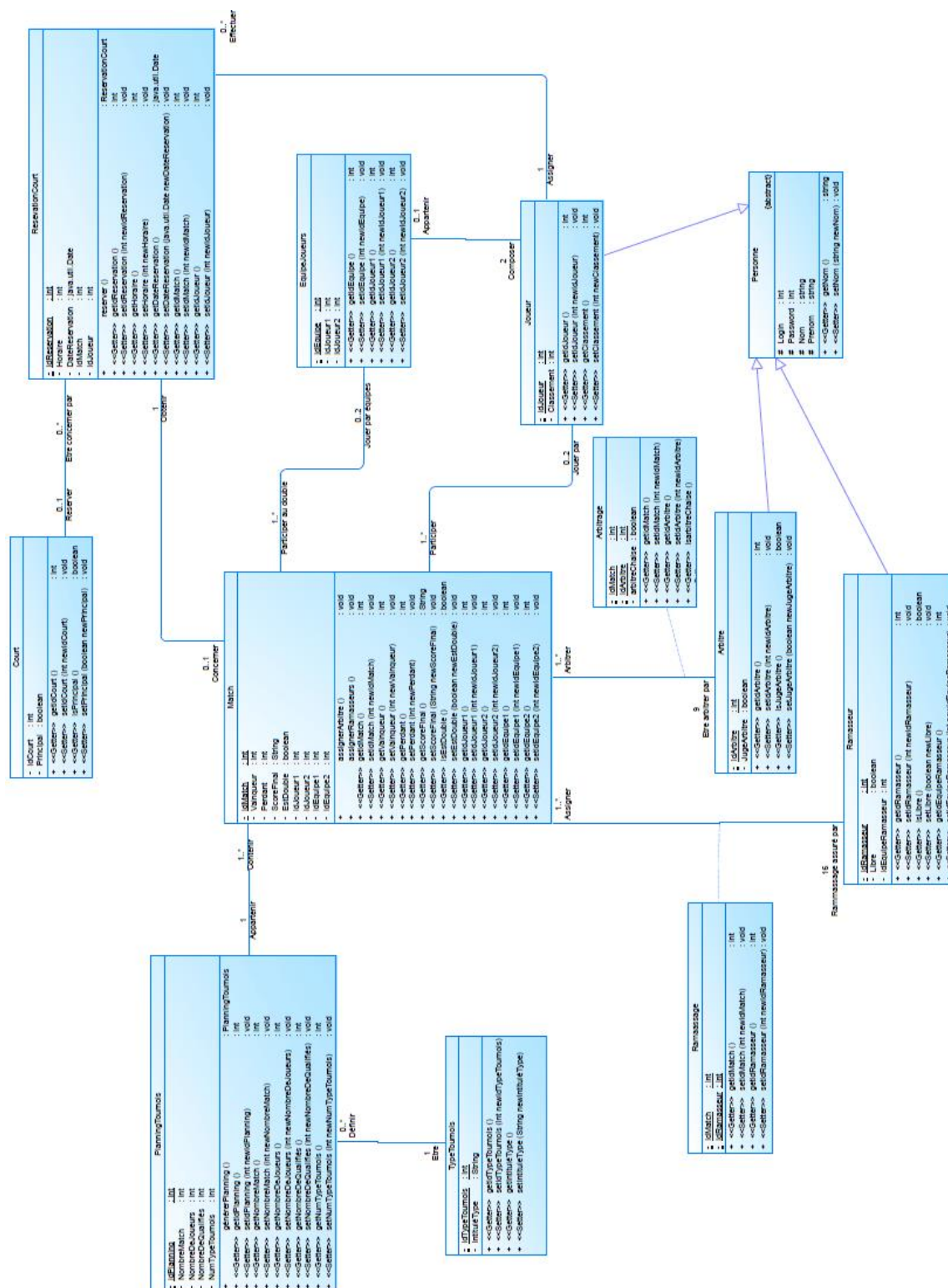


Scénario : le responsable Y peut modifier absolument toutes les infos d'un match dans les limites du possible (créneau, date, score, joueur...).

II.3.8 Diagramme 2dDCL_Planning



II.3.9 Diagramme de Classe de Conception



III Génération de code avec PowerAMC

III.1 Génération de Code JAVA

III.1.1 Classe Billet.java

```
1 /*****
2  * Module: Billet.java
3  * Author: matan
4  * Purpose: Defines the Class Billet
5  *****/
6
7 package Billeterie;
8
9 import Planning.TypeTournois;
10 import java.util.*;
11
12 /** @pdOid df83f82a-ba48-4fb3-8e4b-5cddf6d225b6 */
13 public class Billet {
14     /** @pdOid bc75a281-19e7-430f-be50-7e82e54cfc34 */
15     private int idBillet;
16     /** @pdOid dc2946bc-ed6b-4536-ae87-6d199d62b5dc */
17     private java.util.Date Jour;
18     /** @pdOid 3204e475-3710-48d3-87fc-357a45d89c5a */
19     private int Terrain;
20     /** @pdOid 87e408c5-54ce-4409-ac59-deaa9c615b46 */
21     private int Prix;
22
23     public Place Association_2;
24
25     /** @pdOid 03ef71d2-bc17-4496-95d4-23673539c9e7 */
26     public int CalculPrix() {
27         // TODO: implement
28         return 0;
29     }
30
31     /** @pdOid 5f75846a-9c98-4726-a22e-3e05125bc96f */
32     public int getidBillet() {
33         return idBillet;
34     }
35
36     /** @param newIdBillet
37      * @pdOid 322f02bd-f67c-4297-aa98-d5602d23d5d9 */
38     public void setidBillet(int newIdBillet) {
39         idBillet = newIdBillet;
40     }
41
42     /** @pdOid 8a46e739-89e4-418c-aeba-eddf48dd2b23 */
43     public java.util.Date getJour() {
44         return Jour;
45     }
```

```
46

47 /** @param newJour
48 * @pdOid fd664f0b-e6b4-40c0-9a36-434ab6d01194 */
49 public void setJour(java.util.Date newJour) {
50     Jour = newJour;
51 }
52
53 /** @pdOid 4db8e562-b378-4c77-9e55-bf2c35ac485b */
54 public int getTerrain() {
55     return Terrain;
56 }
57
58 /** @param newTerrain
59 * @pdOid e79b9212-1866-409f-b0fe-6d594589bef8 */
60 public void setTerrain(int newTerrain) {
61     Terrain = newTerrain;
62 }
63
64 /** @pdOid 4066d5f4-fa51-4ead-9849-ae4525a1b8c7 */
65 public int getPrix() {
66     return Prix;
67 }
68
69 /** @param newPrix
70 * @pdOid 0175d756-7b91-4a63-a810-726a6d97604d */
71 public void setPrix(int newPrix) {
72     Prix = newPrix;
73 }
74
75 }
```

III.1.2 Classe Match.java

```
1 /*****
2 * Module: Match.java
3 * Author: matan
4 * Purpose: Defines the Class Match
5 *****/
6
7 package Planning;
8
9 import Billeterie.Place;
10 import java.util.*;
11
12 /** @pdOid 5cce4fdd-b270-4838-85ad-e6e4d3de1a25 */
13 public class Match {
14 /** @pdOid 0828c425-a4de-4519-877d-a4c63216f179 */
15 private int idMatch;
16 /** @pdOid fe38a8b7-425d-4343-b519-88068e78b7ad */
17 private int Vainqueur;
18 /** @pdOid 3ca36478-d28a-4680-9cb2-87463e899953 */
19 private int Perdant;
20 /** @pdOid 73cf27a5-d3f2-41f3-a7bd-5bc3de140b01 */
21 private String ScoreFinal;
22 /** @pdOid dfd42681-b832-496f-9e27-e5e3848f7708 */
23 private boolean EstDouble;
24 /** @pdOid 1e07dc82-250d-4a0b-b4dd-9b2b3482e045 */
25 private int idJoueur1;
26 /** @pdOid 0cabec93-dc4d-4483-bd82-8dfb70d25c69 */
27 private int idJoueur2;
28 /** @pdOid 38011e83-5457-4e00-8c38-60908f35bbc2 */
29 private int idEquipe1;
30 /** @pdOid 3bae3d85-d5ab-487d-8e51-66c64ce88278 */
31 private int idEquipe2;
32
33 public java.util.Collection Association_2;
34
35 /** @pdOid 6b19ee88-fabc-43d4-bbc2-a32ecb2591c9 */
36 public void assignerArbitre() {
37 // TODO: implement
38 }
39
40 /** @pdOid f0fc5b9b-0d57-43f6-aef6-882844b18f9e */
41 public void assignerRamasseurs() {
42 // TODO: implement
43 }
44
45 /** @pdOid 85074820-7870-406b-958d-339f545810d3 */
46 public int getidMatch() {
47 return idMatch;
48 }
49
```

```
50 /** @param newIdMatch
51 * @pdOid 0e2a0125-4405-4c17-9b8c-cd0244eb2854 */
52 public void setIdMatch(int newIdMatch) {
53 idMatch = newIdMatch;
54 }
55
56 /** @pdOid 073a4c82-9d65-4505-9061-7fc94233f3de */
57 public int getVainqueur() {
58 return Vainqueur;
59 }
60
61 /** @param newVainqueur
62 * @pdOid d1f3ea98-4657-4ff8-8547-86a9e6fcea9b */
63 public void setVainqueur(int newVainqueur) {
64 Vainqueur = newVainqueur;
65 }
66
67 /** @pdOid 18d72ce2-e412-4f93-8b72-5ee2dfd78305 */
68 public int getPerdant() {
69 return Perdant;
70 }
71
72 /** @param newPerdant
73 * @pdOid b18ccc6d-9f91-425a-bcdc-150057e96d6f */
74 public void setPerdant(int newPerdant) {
75 Perdant = newPerdant;
76 }
77
78 /** @pdOid a128d51c-59d3-4209-b30f-f401516810a6 */
79 public String getScoreFinal() {
80 return ScoreFinal;
81 }
82
83 /** @param newScoreFinal
84 * @pdOid 6974c7c7-bedf-4487-b80b-ae9cde80ab00 */
85 public void setScoreFinal(String newScoreFinal) {
86 ScoreFinal = newScoreFinal;
87 }
88
89 /** @pdOid a9f45af1-3900-4627-9105-45fc98d69353 */
90 public boolean isEstDouble() {
91 return EstDouble;
92 }
93
94 /** @param newEstDouble
95 * @pdOid 44b4c4d7-2d81-465b-b60f-4522117cc38f */
96 public void setEstDouble(boolean newEstDouble) {
97 EstDouble = newEstDouble;
98 }
99
100 /** @pdOid 1bb680dc-2132-4d08-bcb7-b91e7571edca */
101 public int getidJoueur1() {
```

```
102 return idJoueur1;
103 }
104
105 /** @param newIdJoueur1
106 * @pdOid 89ddbc7c-661a-4255-ab14-64ddc907d7ca */
107 public void setidJoueur1(int newIdJoueur1) {
108 idJoueur1 = newIdJoueur1;
109 }
110
111 /** @pdOid 7c19ed93-d6e8-47cc-b730-dfe5d8bb12c9 */
112 public int getidJoueur2() {
113 return idJoueur2;
114 }
115
116 /** @param newIdJoueur2
117 * @pdOid 348f40d9-8e5f-46e2-a7f1-ab7181f9638a */
118 public void setidJoueur2(int newIdJoueur2) {
119 idJoueur2 = newIdJoueur2;
120 }
121
122 /** @pdOid 89353a34-674c-40a8-ab88-e50610fa09da */
123 public int getidEquipe1() {
124 return idEquipe1;
125 }
126
127 /** @param newIdEquipe1
128 * @pdOid 47866825-5c4c-41d1-ad9e-5adf38232123 */
129 public void setidEquipe1(int newIdEquipe1) {
130 idEquipe1 = newIdEquipe1;
131 }
132
133 /** @pdOid 5647bda8-c414-40e2-ae99-b4d243f64ed3 */
134 public int getidEquipe2() {
135 return idEquipe2;
136 }
137
138 /** @param newIdEquipe2
139 * @pdOid 3f95b81a-50a8-4b59-b6db-45cc08f13152 */
140 public void setidEquipe2(int newIdEquipe2) {
141 idEquipe2 = newIdEquipe2;
142 }
143
144 }
```


III.2 Génération de table SQL

III.2.1 Table Billet

```
1 /*=====*/
2 /* Nom de SGBD : MySQL 5.0 */
3 /* Date de création : 17/12/2021 23:10:03 */
4 /*=====*/
5
6
7 alter table Billet
8 drop foreign key FK_BILLET_ACHAT_CLIENT;
9
10 alter table Billet
11 drop foreign key FK_BILLET_ASSOCIATI_CODEPROM;
12
13 alter table Billet
14 drop foreign key FK_BILLET_ASSOCIATI_CATEGORI;
15
16 alter table Billet
17 drop foreign key FK_BILLET_ASSOCIATI_TYPETOURL;
18
19
20 alter table Billet
21 drop foreign key FK_BILLET_ACHAT_CLIENT;
22
23 alter table Billet
24 drop foreign key FK_BILLET_ASSOCIATI_CODEPROM;
25
26 alter table Billet
27 drop foreign key FK_BILLET_ASSOCIATI_CATEGORI;
28
29 alter table Billet
30 drop foreign key FK_BILLET_ASSOCIATI_TYPETOURL;
31
32 drop table if exists Billet;
33
34 /*=====*/
35 /* Table : Billet */
36 /*=====*/
37 create table Billet
38 (
39 idBillet int not null comment "",
40 idTypeTournois int comment "",
41 IdClient int not null comment "",
42 idCategorie int not null comment "",
43 CodePromo varchar(254) comment "",
44 Jour datetime comment "",
45 Terrain int comment "",
46 Prix int comment "",
47 primary key (idBillet)
48 );
```

```
49
50 alter table Billet add constraint FK_BILLET_ACHAT_CLIENT foreign key (IdClient)
51 references Client (IdClient) on delete restrict on update restrict;
52
53 alter table Billet add constraint FK_BILLET_ASSOCIATI_CODEPROM foreign key (CodePromo)
54 references CodePromotionnels (CodePromo) on delete restrict on update restrict;
55
56 alter table Billet add constraint FK_BILLET_ASSOCIATI_CATEGORI foreign key
57 (idCategorie)
58 references CategorieBillet (idCategorie) on delete restrict on update restrict;
59
60 alter table Billet add constraint FK_BILLET_ASSOCIATI_TYPETOIR foreign key
61 (idTypeTours)
62 references TypeTours (idTypeTours) on delete restrict on update restrict;
```

III.2.2 Table Match

```
1 /*=====*/
2 /* Nom de SGBD : MySQL 5.0 */
3 /* Date de création : 17/12/2021 11:10:32 */
4 /*=====*/
5
6
7 alter table `Match`
8 drop foreign key FK_MATCH_ASSOCIATI_RESEVATI;
9
10 alter table `Match`
11 drop foreign key FK_MATCH_ASSOCIATI_PLANNING;
12
13 alter table `Match`
14 drop foreign key FK_MATCH_ASSOCIATI_ARBITRE;
15
16 alter table `Match`
17 drop foreign key FK_MATCH_ASSOCIATI_RESEVATI;
18
19
20 alter table `Match`
21 drop foreign key FK_MATCH_ASSOCIATI_RESEVATI;
22
23 alter table `Match`
24 drop foreign key FK_MATCH_ASSOCIATI_ARBITRE;
25
26 alter table `Match`
27 drop foreign key FK_MATCH_ASSOCIATI_PLANNING;
28
29 alter table `Match`
30 drop foreign key FK_MATCH_ASSOCIATI_RESEVATI;
31
32 drop table if exists `Match`;
33
34 /*=====*/
35 /* Table : `Match` */
36 /*=====*/
37 create table `Match`
38 (
39 idMatch int not null comment "",
40 idPlanning int not null comment "",
41 idArbitre int not null comment "",
42 idReservation int comment "",
43 Res_idReservation int not null comment "",
44 idVainqueur int comment "",
45 idPerdant int comment "",
46 ScoreFinal varchar(254) comment "",
47 EstDouble bool comment "",
48 primary key (idMatch)
49 );
```

```
50
51 alter table `Match` add constraint FK_MATCH_ASSOCIATI_RESEVATI foreign key
(idReservation)
52 references ResevationCourt (idReservation) on delete restrict on update
restrict;
53
54 alter table `Match` add constraint FK_MATCH_ASSOCIATI_PLANNING foreign key
(idPlanning)
55 references PlanningTournois (idPlanning) on delete restrict on update restrict;
56
57 alter table `Match` add constraint FK_MATCH_ASSOCIATI_ARBITRE foreign key (idArbitre)
58 references Arbitre (idArbitre) on delete restrict on update restrict;
59
60 alter table `Match` add constraint FK_MATCH_ASSOCIATI_RESEVATI foreign key
(Res_idReservation)
61 references ResevationCourt (idReservation) on delete restrict on update
restrict;
62
63 alter table `Match` add constraint FK_MATCH_ASSOCIATION_JOUEUR1 foreign key
(idVainqueur)
64 references Joueur (idJoueur) on delete restrict on update restrict;
65
66 alter table `Match` add constraint FK_MATCH_ASSOCIATION_JOUEUR2 foreign key
(idVainqueur)
67 references Joueur (idJoueur) on delete restrict on update restrict;
68
69
```

IV Maquette

IV.1 Maquette de l'application Planning

IV.1.1 Page de Lancement

L'utilisateur a le choix entre accéder à l'espace Administrateur (pour gérer les plannings) ou l'espace joueur pour réserver un court dans le cadre d'un entraînement. Ce choix amènera à une page où l'utilisateur devra s'identifier avec ses codes uniques.



IV.1.2 Espace Administrateur

Cette page constitue le menu principal des utilisateurs administrateurs (responsables du planning). Ils pourront à partir de cette page accéder aux fonctionnalités de l'application qui leur sont destinées.



IV.1.3 Sélection d'un match et modification d'un match

Ces deux pages représentent les fonctionnalités de modification des différents matchs du planning. L'une sert à sélectionner le match à modifier, l'autre à saisir les Nouvelles infos de celui-ci.

The screenshot shows a web application window titled "SPEN PARC AUVERGNE-RHÔNE-ALPES LYON". The main heading is "Sélectionner un match à gérer". Below it is a table with 5 columns: Court, Time, Round, Qualification, and Date. The table contains 9 rows of match data. The third row, "Court B 10:00 8ème Qualification 15/05/2022", is highlighted in blue. Below the table are two buttons: "Retour" (red) and "Modifier le match" (blue).

Court	Time	Round	Qualification	Date
Court A	10:00	8ème	Qualification	15/05/2022
Court Principal	10:00	8ème	Qualification	15/05/2022
Court B	10:00	8ème	Qualification	15/05/2022
Court C	10:00	8ème	Qualification	15/05/2022
Court Principal	12:00	8ème	Qualification	15/05/2022
Court A	12:00	8ème	Qualification	15/05/2022
Court B	12:00	8ème	Qualification	15/05/2022
Court C	12:00	8ème	Qualification	15/05/2022

The screenshot shows a web application window titled "SPEN PARC AUVERGNE-RHÔNE-ALPES LYON". The main heading is "Modifier le match". Below it are three input fields: "Court :" with a dropdown menu showing "Court B", "Créneau :" with a dropdown menu showing "10:10", and "Date :" with a text field showing "15/05/2022" and a calendar icon. Below these fields is a horizontal line. Under the line are two checkboxes: "Simple" (checked) and "Double" (unchecked). Below the checkboxes are two input fields: "Opposant 1 :" with a dropdown menu showing "Nom", and "Opposant 2 :" with a dropdown menu showing "Nom". Below these fields is a horizontal line. Under the line are two input fields: "Score :" with a text field showing "3", and "Score :" with a text field showing "3". Below these fields is a horizontal line. Under the line are two buttons: "Annuler" (red) and "Valider" (blue).

IV.1.4 Génération et consultation des plannings

L'application permettra au responsable du planning de générer un planning par tournois, d'y apporter des éventuelles modifications et de le valider.

SPEN PARC ALVERGNE-RHÔNE-ALPES LYON

Planning des qualifications

Court A	10:00	8ème	J-1	vs	J-2	15/05/2022
Court Principal	10:00	8ème	J-3	vs	J-4	15/05/2022
Court B	10:00	8ème	J-5	vs	J-6	15/05/2022
Court C	10:00	8ème	J-7	vs	J-8	15/05/2022
Court Principal	12:00	8ème	J-9	vs	J-10	15/05/2022
Court A	12:00	8ème	J-11	vs	J-12	15/05/2022
Court B	12:00	8ème	J-13	vs	J-14	15/05/2022
Court C	12:00	8ème	J-15	vs	J-16	15/05/2022

Modifier le match

Planning des simples

Court A	10:00	16ème	J-1	vs	J-2	15/05/2022
Court Principal	10:00	16ème	J-3	vs	J-4	15/05/2022
Court B	10:00	16ème	J-5	vs	J-6	15/05/2022
Court C	10:00	16ème	J-7	vs	J-8	15/05/2022
Court Principal	12:00	16ème	J-9	vs	J-10	15/05/2022
Court A	12:00	16ème	J-11	vs	J-12	15/05/2022
Court B	12:00	16ème	J-13	vs	J-14	15/05/2022
Court C	12:00	16ème	J-15	vs	J-16	15/05/2022

Modifier le match

Planning des doubles

Court A	10:00	8ème	E-1	vs	E-2	15/05/2022
Court Principal	10:00	8ème	E-3	vs	E-4	15/05/2022
Court B	10:00	8ème	E-5	vs	E-6	15/05/2022
Court C	10:00	8ème	E-7	vs	E-8	15/05/2022
Court Principal	12:00	8ème	E-9	vs	E-10	15/05/2022
Court A	12:00	8ème	E-11	vs	E-12	15/05/2022
Court B	12:00	8ème	E-13	vs	E-14	15/05/2022
Court C	12:00	8ème	E-15	vs	E-16	15/05/2022

Modifier le match

Retour

Régénérer

Valider les planning

IV.1.5 Espace Joueur

Espace atteint lorsque l'utilisateur se connecte en tant que joueur afin de réserver un court ou bien de consulter ou annuler une précédente réservation.



IV.1.6 Réserveation de court et consultation des réservations

Ces deux pages sont accessibles aux joueurs afin de gérer leurs réservations.

OPEN PARC
AUVERGNE-RHÔNE-ALPES
LYON

Nouvelle réservation

Daté désirée :

Créneau désiré :

Court A	10:30
Court B	10:30
Court C	11:00

OPEN PARC
AUVERGNE-RHÔNE-ALPES
LYON

Mes réservations de courts

Court A	10:30	16/05/2022
Court A	8:00	17/05/2022
Court C	18:00	17/05/2022

IV.2 Maquette de l'application Billetterie

IV.2.1 Sélection du billet voulu

L'utilisateur (client) choisit le billet à la date qu'il souhaite.



IV.2.2 Sélection de la catégorie du billet

Si le client prend une place de match sur le court central, il a alors la possibilité de choisir une catégorie de place, amenant à un prix final plus ou moins élevé.

Dimanche 16 mai

Qualifications pour les matchs simple

A partir de 10h00 8 matchs

Le programme en générale

(Boxlist)

Choisir le match voulu :

Match 1 : J1 vs J2

Match 2 : J1 vs J2

Match 3 : J1 vs J2

Match 4 : J1 vs J2

Terrain

Court central

(Boxlist)


Choisir la catégorie apparait uniquement si court central choisi

Catégorie 1


Catégorie 2

Catégorie 2 (-12 ans)

Prix total : €

Panier 

IV.2.3 Récapitulatif du panier et bons de réductions

Panier 

Votre panier

Dimanche 16 mai - Court central - Catégorie 1-----30€


Numéro de licence

Code de réduction

Total-----30€

Payer

IV.2.4 Simulation de paiement par carte

Panier 

Titulaire de la carte

Numéro de la carte

Date d'expiration

Code de vérification de la carte

Payer