



PowerAMC de Sybase Rapport Modèle orienté objet

Modèle: tennis annalyse
Rapport: Rapport Analyse
Auteur: p2000087
Version:
Date: 17/12/2021
Résumé:

Table des matières

I	INTRODUCTION.....	3
I.1	HEBERGEMENT.....	3
I.1.1	Présentation.....	3
I.1.2	L'application.....	3
I.2	PLANNING DES MATCHS	4
I.2.1	Présentation.....	4
I.2.2	L'application.....	4
II	ANALYSE.....	5
II.1	DIAGRAMME DE PACKAGES.....	5
II.2	PACKAGE HEBERGEMENT.....	6
II.2.1	Diagramme DCU Hebergement.....	6
II.2.2	Diagramme DIT Réserver une chambre	6
II.2.3	Diagramme DIT Supprimer une réservation	7
II.2.4	Diagramme DIT Valider une réservation	8
II.2.5	Diagramme DSQ Réservation d'un hôtel par un membre VIP	9
II.2.6	Diagramme DCL Hebergement.....	10
II.3	PACKAGE PLANNING.....	11
II.3.1	Diagramme DCU Planning des matchs.....	11
II.3.2	Diagramme DIT Modifier l'attribution des courts	12
II.3.3	Diagramme DIT Modifier l'attribution de l'arbitre	13
II.3.4	Diagramme DIT Modifier l'attribution des ramasseurs.....	14
II.3.5	Diagramme DSQ Modifier l'attribution de l'arbitre	15
II.3.6	Diagramme DCL Planning des matchs.....	16
III	MAQUETTES	17
III.1	MAQUETTES DU PLANNING DES MATCHS	17
III.2	MAQUETTES DES HEBERGEMENTS	17
IV	CONCEPTION.....	18
IV.1	PACKAGE PLANNING DES MATCHS	18
IV.1.1	DCL Planning des matchs	18
IV.1.2	Scripts SQL.....	20
IV.1.2.1	Table court.....	20
IV.1.2.2	Table match	20
IV.1.2.3	Code JAVA	22
IV.1.2.4	Class joueur	22
IV.1.2.5	Class matchs	26
IV.2	PACKAGE HEBERGEMENT.....	31
IV.2.1	DCL Hebergement	31
V	ANNEXE.....	34

I Introduction

I.1 Hébergement

I.1.1 Présentation

L'application pour l'hébergement a pour but de faciliter, et de gérer la réservation des chambres d'hôtel pendant le tournoi. Chaque joueur (et son équipe), arbitre, membre du staff peuvent eux-mêmes réserver une chambre dans un établissement partenaire. Le responsable de logement du Tournoi doit valider chaque réservation et vérifier que les joueurs et les arbitres ne soient pas dans le même hôtel, il gère aussi tous les problèmes liés à l'hébergement. L'application permet aussi au gérant des hébergements de rentrer les informations et caractéristiques de l'hôtel et doit aussi mettre à jour le nombre de places disponibles chaque jour.



I.1.2 L'application

L'application aurait une page connexion qui renvoie à une interface différente selon la catégorie de personnes qui se connectent (arbitre, joueurs, gérants d'hôtel, responsable de l'hébergement du tournoi...).

Comme les joueurs et les arbitres ne seront pas dans les mêmes hôtels, on a choisi une solution qui permet à tous de voir le plus d'hôtels possible. Tout le monde peut voir les hôtels qui n'ont aucune réservation. Dès que quelqu'un réserve dans un de ces hôtels, **on regarde sa catégorie puis en fonction de ça on le rend plus visible aux autres catégories**. Par exemple, si la première réservation d'un hôtel est faite par un arbitre, cet hôtel ne sera alors visible que par les arbitres. A contrario, si la première réservation est faite par un joueur, alors l'hôtel ne sera visible que par des joueurs.

Quand c'est un joueur qui se connecte :

- a accès à tous les hôtels prenant que des joueurs (avec le nombre de places restantes)
- en cliquant sur l'hôtel choisi, ils peuvent réserver
- ils indiquent le nombre de personnes et les différentes contraintes qu'il a

Quand c'est un arbitre qui se connecte :

- a accès à tous les hôtels prenant les arbitres (avec le nombre de places restantes)
- en cliquant sur l'hôtel choisi, ils peuvent réserver
- ils indiquent les différentes contraintes qu'il a

Quand c'est un gérant de l'hôtel qui se connecte :

- a accès aux données des hôtels qu'il gère
- peut modifier/ajouter/supprimer des services offerts
- peut modifier/ajouter/supprimer des informations concernant l'hôtel
- peut mettre à jour le nombre de places disponibles



Quand c'est le responsable de l'hébergement du tournoi qui se connecte :

- a accès à tous les hôtels avec le nombre de places restantes
- peut ajouter un hôtel
- valider ou non la réservation d'un joueur/d'un arbitre

I.2 Planning des matchs

I.2.1 Présentation

Le planning des matchs est composé de deux parties.

La première a pour but d'aider les organisateurs à organiser les matchs. Une fonctionnalité doit permettre de proposer un 1er planning des matchs. Les organisateurs pourront ensuite modifier celui-ci. Un match sera défini aura comme paramètres:

- le type de match (simple ou double)
- les joueurs
- les arbitres
- les 2 équipes de ramasseurs
- le court
- l'horaire et la date
- l'arbitre



Sur ce planning, on retrouvera la nationalité et la catégorie des arbitres. Elles permettent au responsable de modifier, supprimer ou déplacer un match existant.

L'application doit mentionner le nom du gagnant pour chaque match avec le score qu'il y a eu. La deuxième partie concerne plus les joueurs. L'application doit leur permettre de réserver un court pour s'entraîner. Il suffit de saisir le nom du joueur, et de sélectionner un créneau de libre.

I.2.2 L'application

Tout d'abord l'application aura aussi un système de connexion **car les organisateurs et les joueurs ne doivent pas forcément voir la même chose.**

Il y aura en premier une page commune à tous qui comportera le planning des matchs ainsi que les matchs qui ont déjà eu lieu avec le nom des joueurs qui a gagné et le score.

Il y aura aussi un menu qui proposera différentes fonctionnalités en fonction de qui s'est connecté, nous détaillerons ce point juste après. Tout le monde aura accès dans le menu à un **onglet information sur les joueurs, les arbitres, les ramasseurs de balles avec leur nom, prénom, nationalité....**

Les joueurs auront accès à :

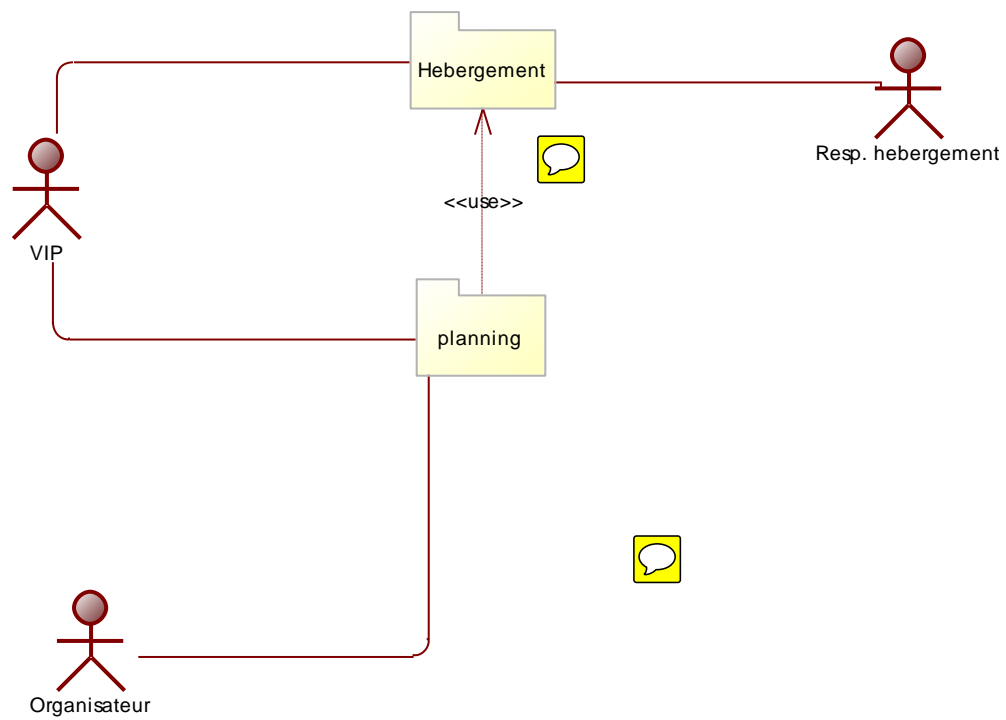
- une page pour réserver les cours d'entraînements à n'importe quelle heure tant qu'il est disponible

Les organisateurs auront accès à :

- la possibilité d'insérer un score
- la possibilité de modifier les informations du planning
- la possibilité de modifier/supprimer les infos générales

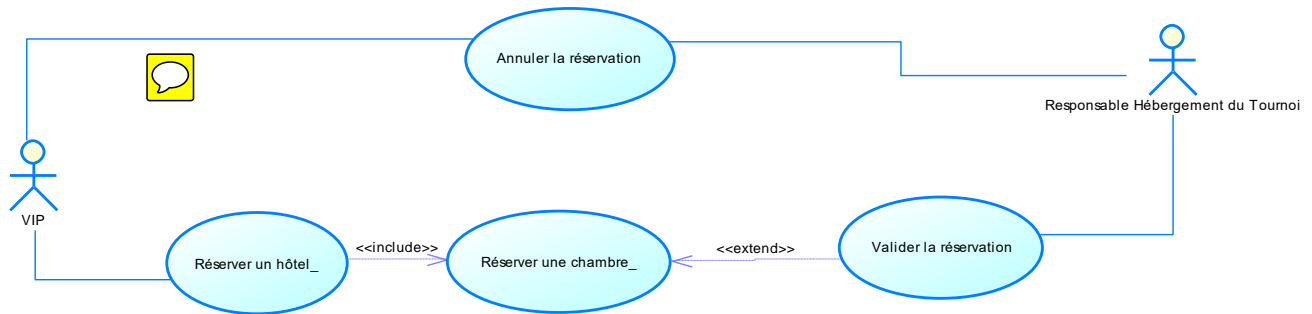
II Analyse

II.1 Diagramme de packages



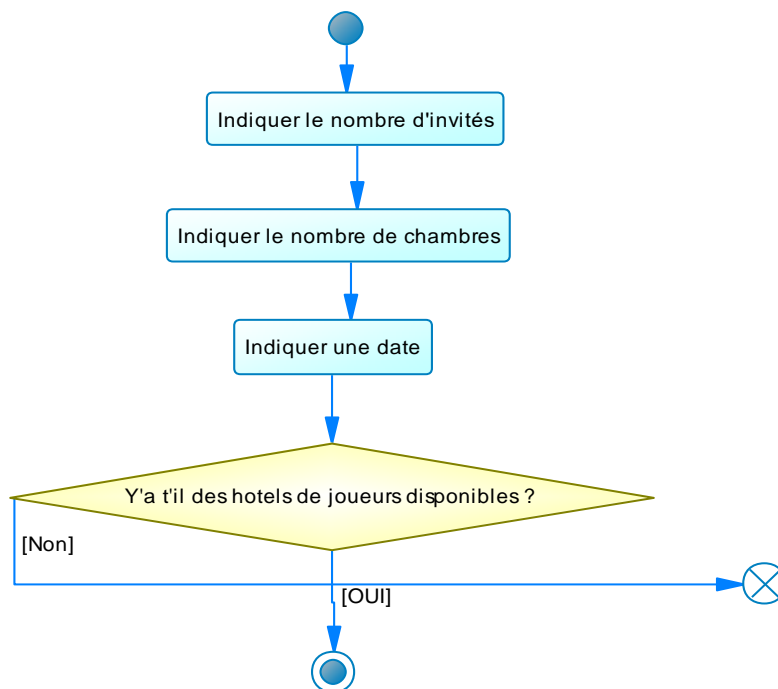
II.2 Package Hebergement

II.2.1 Diagramme DCU Hebergement



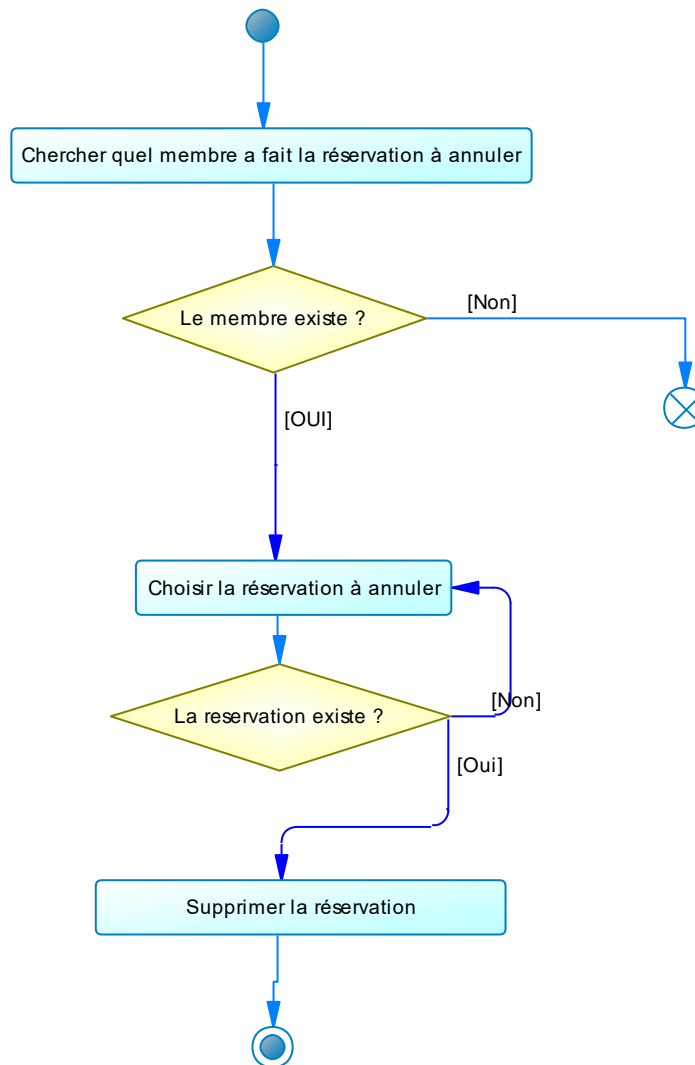
II.2.2 Diagramme DIT Réserver une chambre

Réserver une chambre



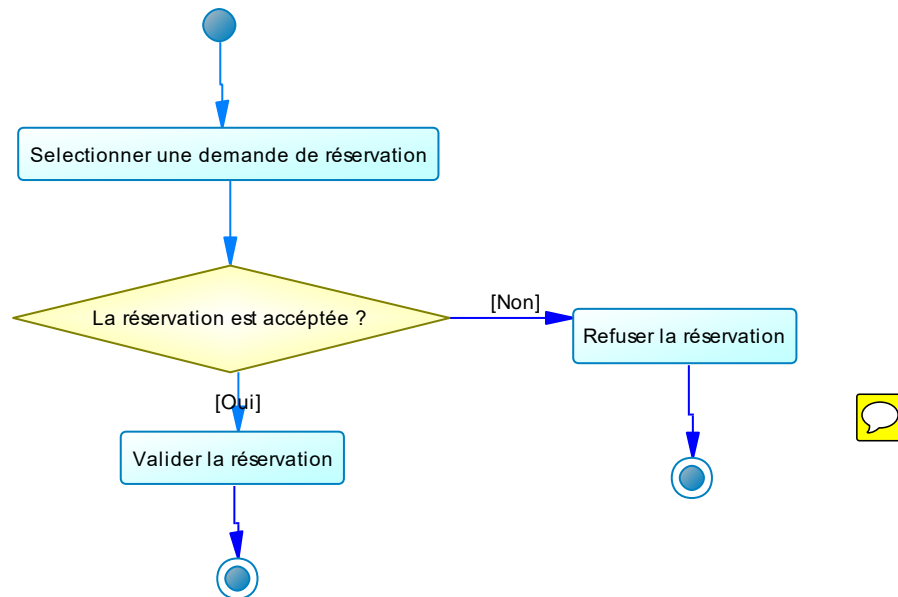
II.2.3 Diagramme DIT Supprimer une réservation

Supprimer une réservation

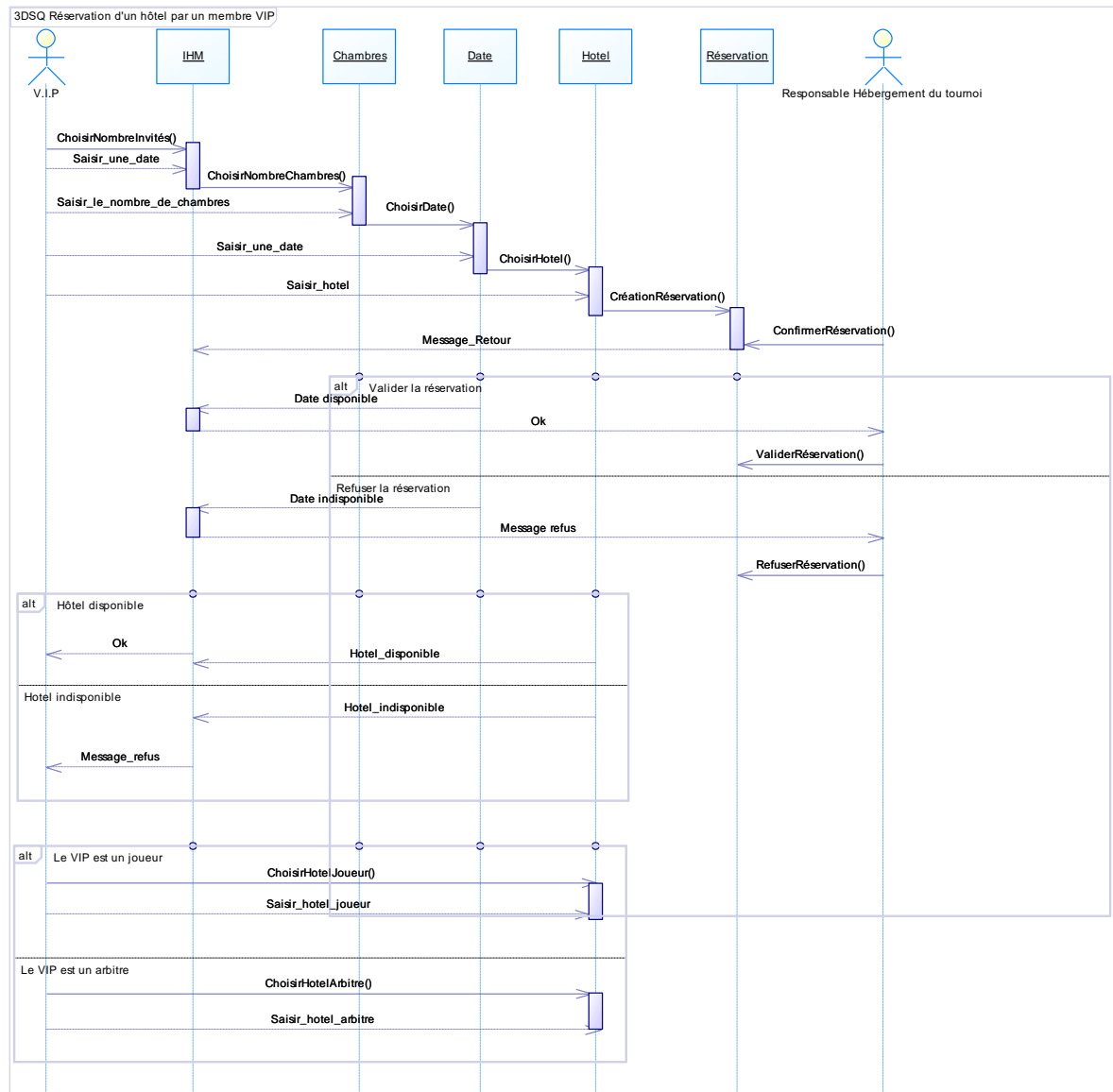


II.2.4 Diagramme DIT Valider une réservation

Valider une réservation

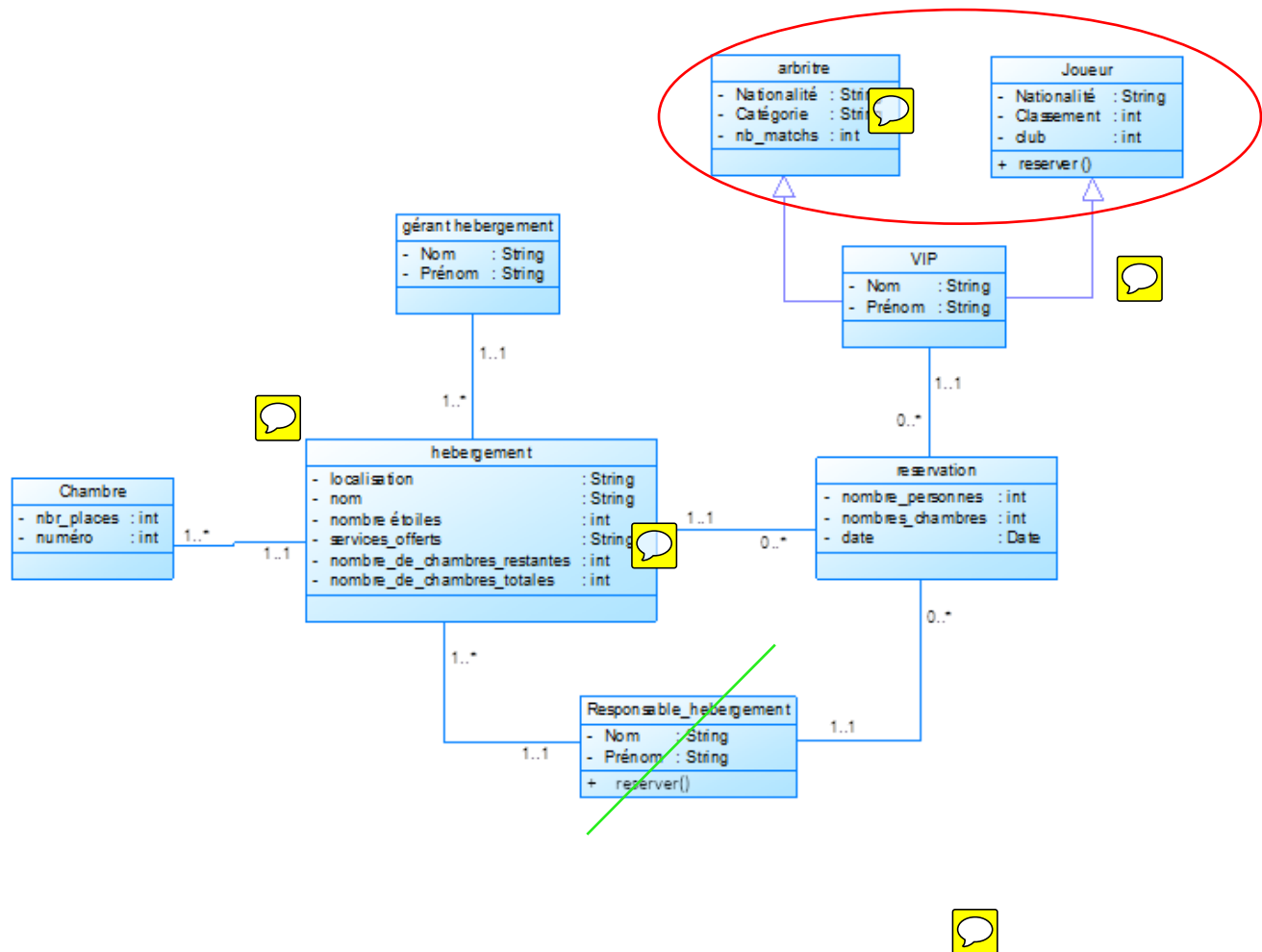


II.2.5 Diagramme DSQ Réserveation d'un hôtel par un membre VIP



II.2.6 Diagramme DCL Hébergement

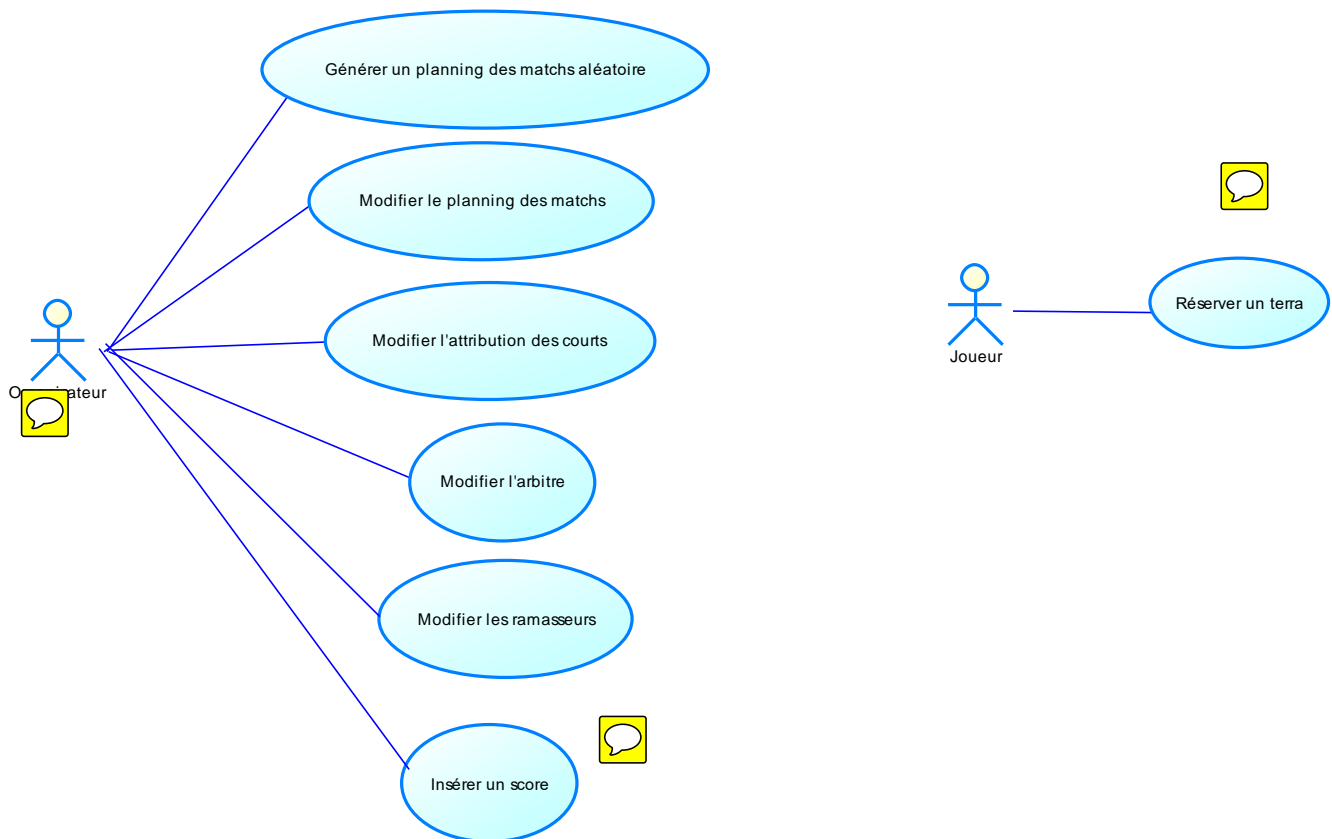
Dans ce diagramme les arbitres et les joueurs sont traités comme des VIP. On enregistre leur réservation, qui est transmise au responsable des hébergements. Celui-ci doit ensuite valider la réservation.



II.3 Package planning

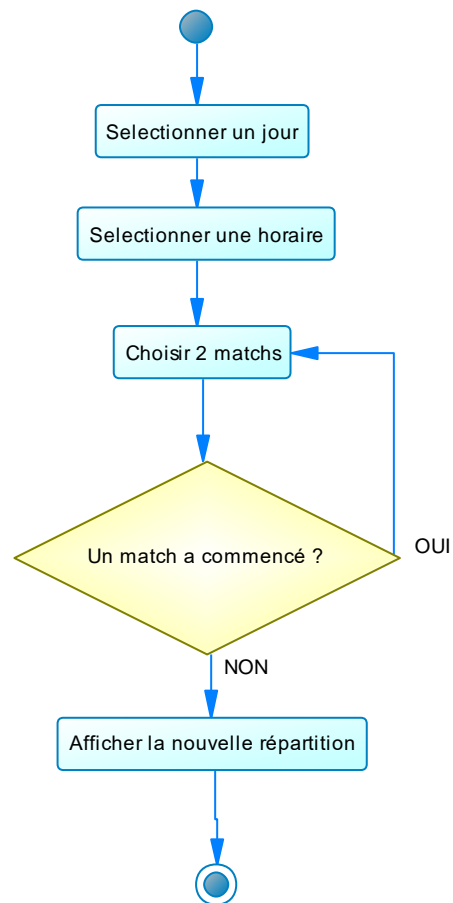
II.3.1 Diagramme DCU Planning des matchs

Diagramme de cas d'utilisation: Planning des matchs



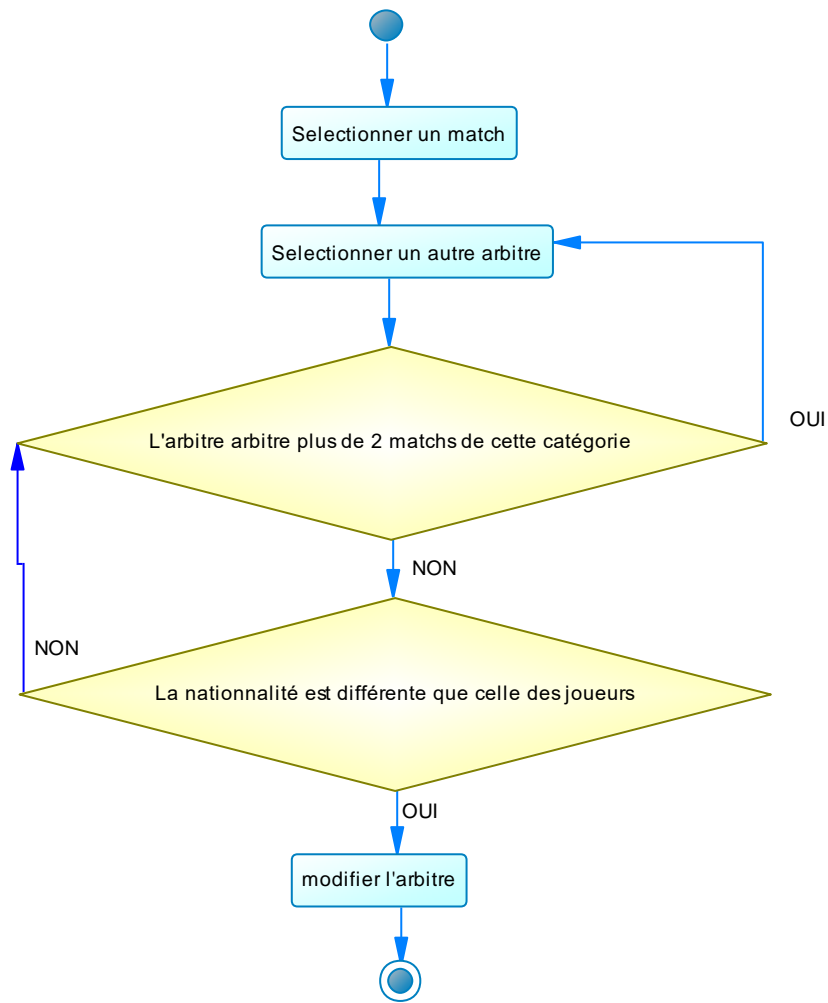
II.3.2 Diagramme DIT Modifier l'attribution des courts

Modifier l'attribution des courts



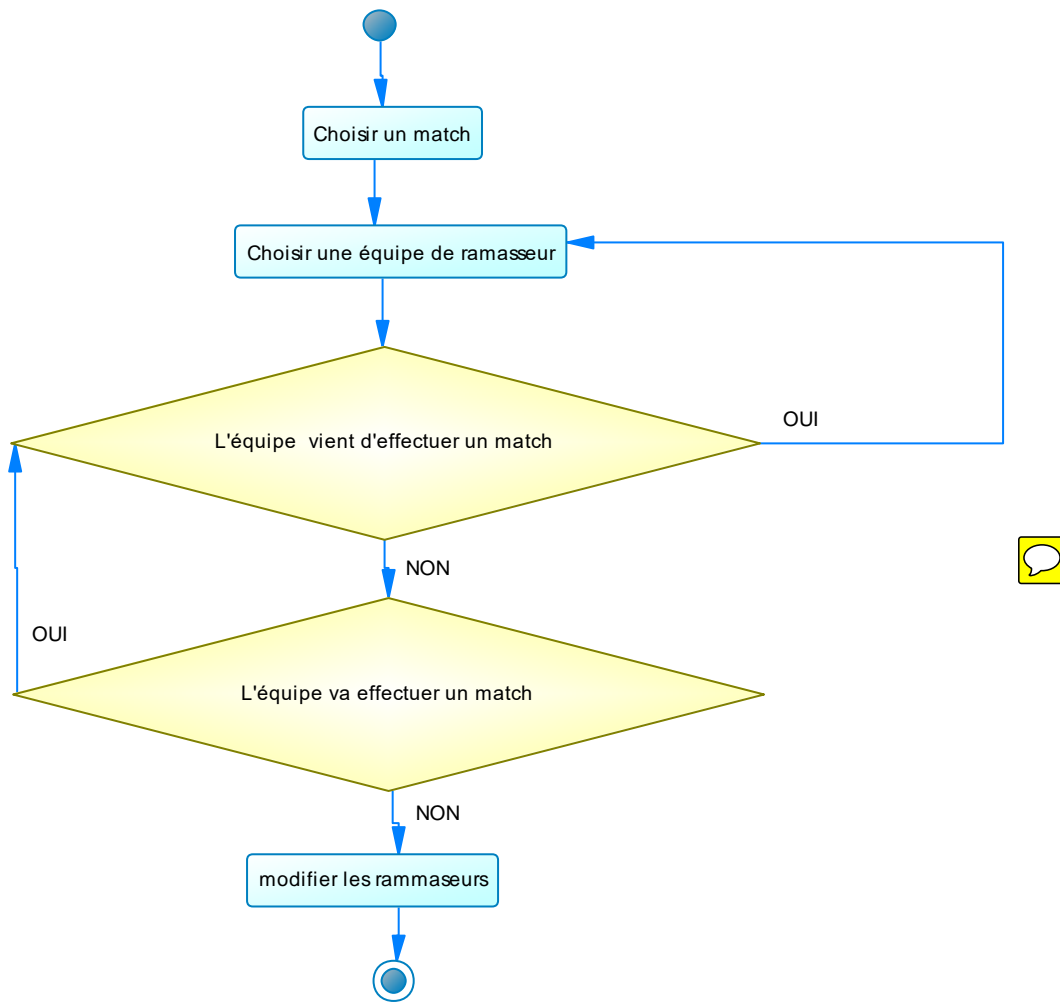
II.3.3 Diagramme DIT Modifier l'attribution de l'arbitre

Modifier l'attribution de l'arbitre

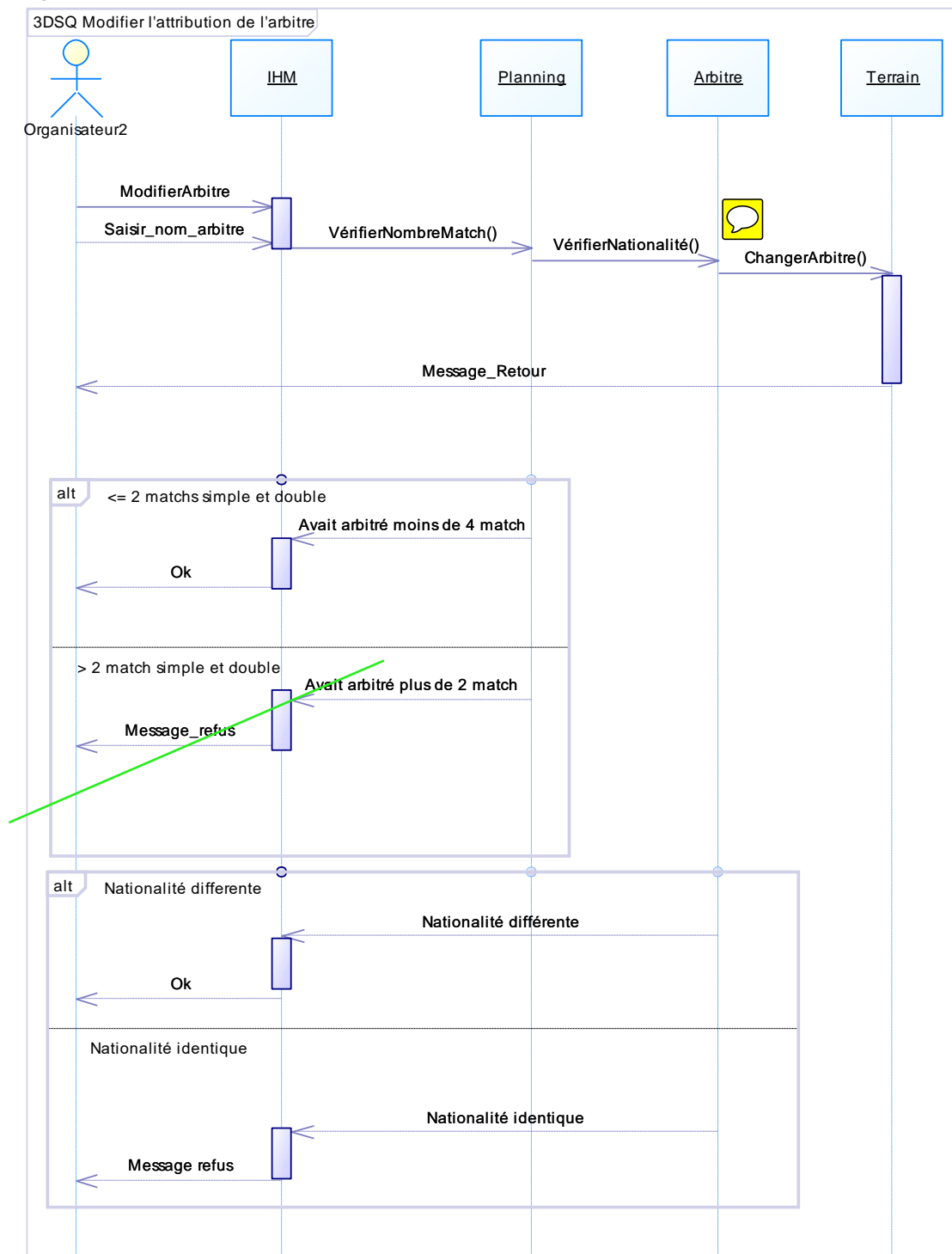


II.3.4 Diagramme DIT Modifier l'attribution des ramasseurs

Modifier l'attribution des ramasseurs

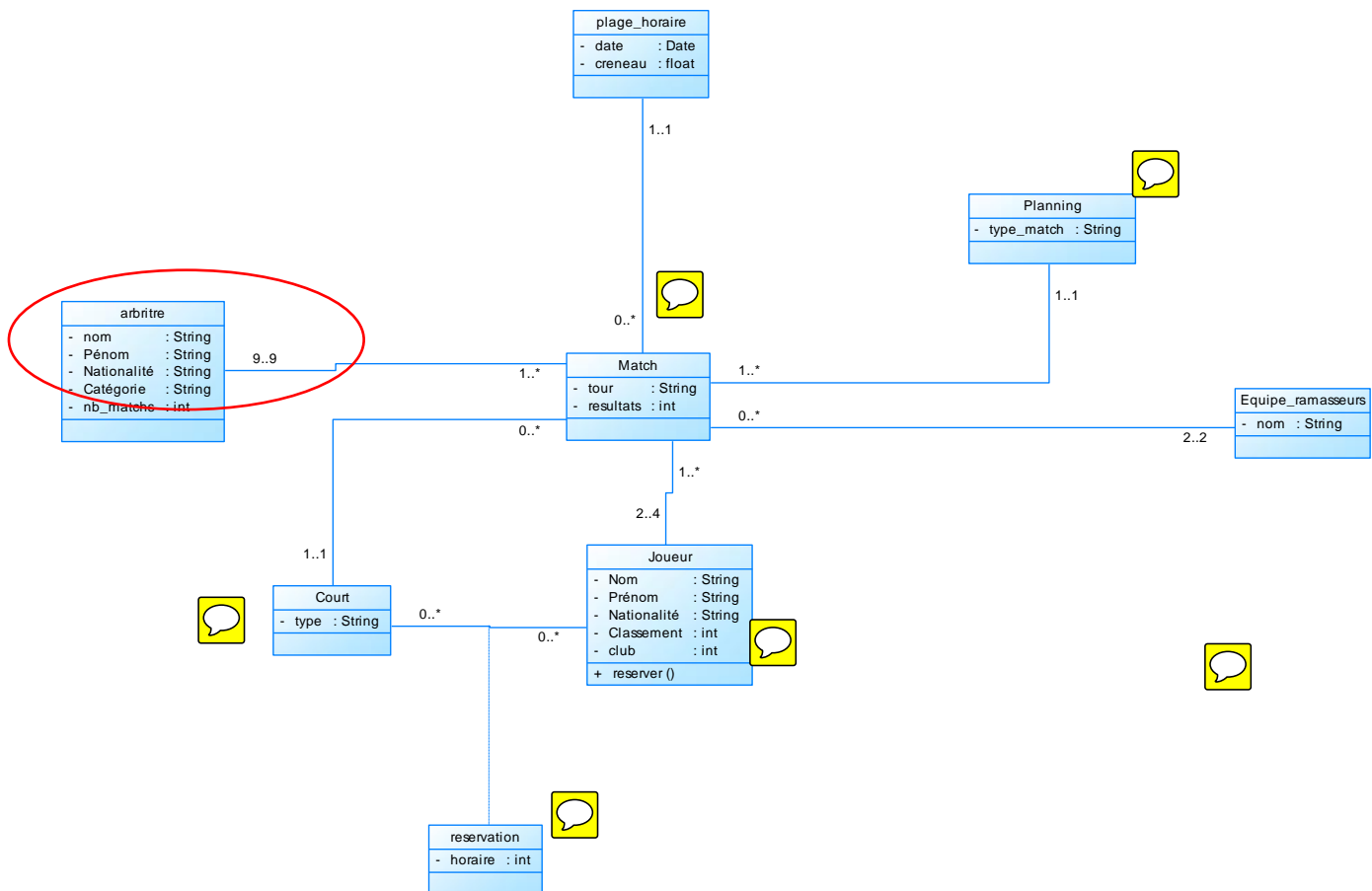


II.3.5 Diagramme DSQ Modifier l'attribution de l'arbitre



II.3.6 Diagramme DCL Planning des matchs

Dans ce diagramme on enregistre tous les matchs dans des plannings, suivant la catégorie des matchs. On assigne à chaque match, 9 arbitres, 2 équipes de ramasseurs, 1 court, 1 plage horaire, et 2 ou 4 joueurs suivant le type de match. De plus les joueurs ont la possibilité de réserver un court pour s'entraîner.



III Maquettes

Les maquettes ont été faites sur un logiciel qui ne nous permettait pas de les inclure dans le rapport. C'est pour cela qu'elles sont trouvables en annexe.

III.1 Maquettes du planning des matchs

Cf. Annexe n°1

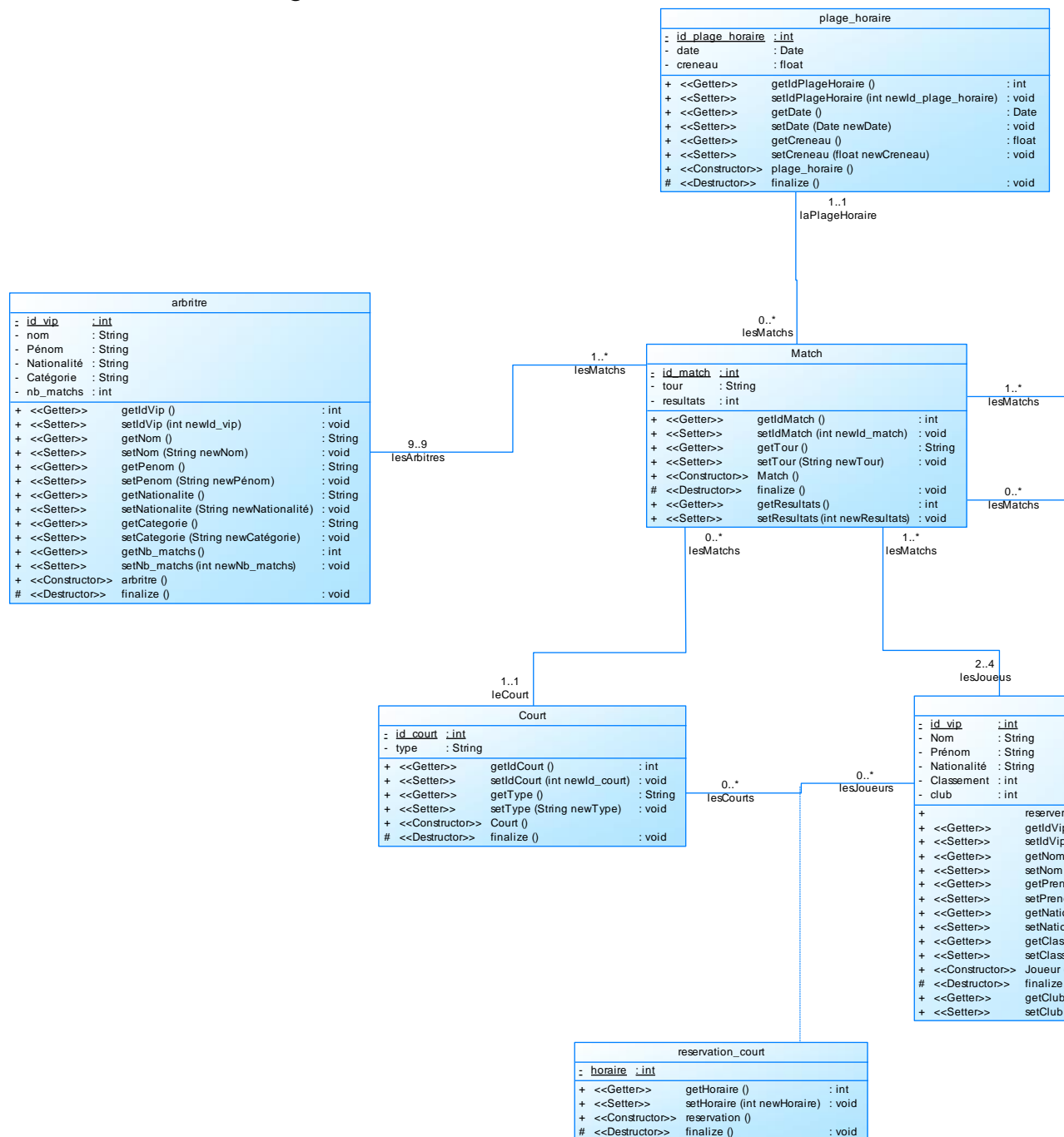
III.2 Maquettes des hébergements

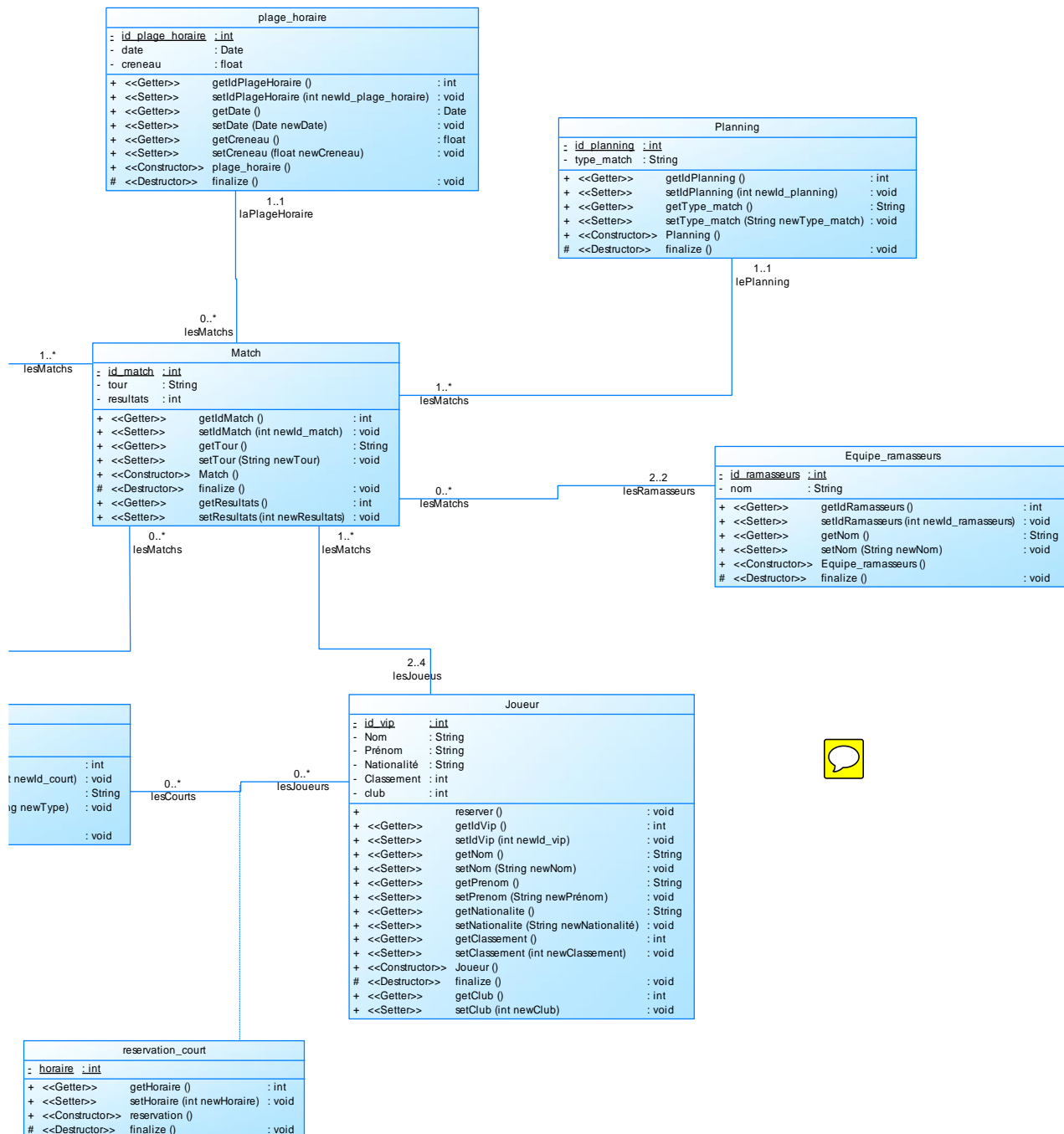
Cf. Annexe n°2

IV Conception

IV.1 Package Planning des matchs

IV.1.1 DCL Planning des matchs





IV.1.2 Scripts SQL

IV.1.2.1 Table court

```

/*=====*/
/* Table : court */
/*=====*/
create table court
(
  idcourt          int          not null,
  type             varchar(254),
  primary key (idcourt)
)
type = innodb;

```

IV.1.2.2 Table match

```

/*=====*/
/* Table : `match` */
/*=====*/
create table `match`
(
  idmatch          int          not null,
  idplanning       int          not null,
  idcourt          int          not null,
  idplagehoraire   int          not null,
  tour             varchar(254),
  resultats        int,
  primary key (idmatch)
)
type = innodb;

/*=====*/
/* Index : association_5_fk */
/*=====*/
create index association_5_fk on `match`
(
  idcourt
);

/*=====*/
/* Index : association_7_fk */
/*=====*/
create index association_7_fk on `match`
(

```

```
    idplagehoraire
);

/*=====*/
/* Index : association_10_fk */
/*=====*/
create index association_10_fk on `match`
(
    idplanning
);

alter table `match` add constraint fk_association_10 foreign key (idplanning)
references planning (idplanning) on delete restrict on update restrict;

alter table `match` add constraint fk_association_5 foreign key (idcourt)
references court (idcourt) on delete restrict on update restrict;

alter table `match` add constraint fk_association_7 foreign key (idplagehoraire)
references plage_horaire (idplagehoraire) on delete restrict on update restrict;
```



III.1.2 Code JAVA

IV.1.2.3 Class joueur

```

/*****
 * Module: Joueur.java
 * Author: p2000087
 * Purpose: Defines the Class Joueur
 *****/

```

```

package planning;
import hebergement.reservation;
import java.util.*;

```

```

/** @pdOid 1871b3ff-3afa-4961-a3cc-116a9e20242b */
public class Joueur {
    /** @pdOid 9a578b7c-b1bd-44c2-847f-c4090f1dd427 */
    private int idVip;
    /** @pdOid 2c1b1313-ad35-4b5e-98a6-56b101f9ca71 */
    private String Nom;
    /** @pdOid 6b1ef5e2-40a3-47c1-a439-a387a9ee860d */
    private String Prenom;
    /** @pdOid 83ef30ef-0509-4a31-b0f2-3b22fba9601f */
    private String Nationalite;
    /** @pdOid ee866539-22cc-41ee-96fb-14dee0695b5a */
    private int Classement;
    /** @pdOid eae04c76-ab23-4aa4-a8ff-e7e60e03e921 */
    private int club;

```



```

/** @pdOid 80ea3b87-73fd-4ff1-a511-9e6527e2baf0 */
protected void finalize() {
    // TODO: implement
}

```

```

/** @pdRoleInfo migr=no name=Match assc=Association_12 coll=java.util.Collection impl=java.util.HashSet
mult=1..* */
public java.util.Collection<Match> lesMatches;
public java.util.Collection Association_6;
/** @pdRoleInfo migr=no name=reservation assc=association7 mult=0..* side=A */
public reservation[] lesRéservations;

```

```

/** @pdOid b0f6b3b0-db59-470a-876f-b5e78dc48d92 */
public void reserver() {
    // TODO: implement
}

```

```

/** @pdOid 7512187f-0ec2-488f-b19e-833c5f176900 */
public int getIdVip() {
    return idVip;
}

```

```
/** @param newIdVip
 * @pdOid eeb86269-8ce3-4698-9b63-6da26af5a3d7 */
public void setIdVip(int newIdVip) {
    idVip = newIdVip;
}

/** @pdOid de9676f9-8395-44e0-85e8-1c36d08cab96 */
public String getNom() {
    return Nom;
}

/** @param newNom
 * @pdOid 7668af0f-4e4c-4640-82fc-844eb55c6ca4 */
public void setNom(String newNom) {
    Nom = newNom;
}

/** @pdOid db329f6f-0fd4-4b5d-aaac-dbb7f149d501 */
public String getPrenom() {
    return Prenom;
}

/** @param newPrenom
 * @pdOid 017abcea-1c3b-48cb-ab96-9d34d3a9f8c0 */
public void setPrenom(String newPrenom) {
    Prenom = newPrenom;
}

/** @pdOid 89f502c1-2ab7-4fd5-8ed7-e46ce3dcbf35 */
public String getNationalite() {
    return Nationalite;
}

/** @param newNationalite
 * @pdOid d2b2066b-f7dd-4d5f-a894-14532e3230b3 */
public void setNationalite(String newNationalite) {
    Nationalite = newNationalite;
}

/** @pdOid 5f093714-6c96-436a-9405-fd3c9e4b1f15 */
public int getClassement() {
    return Classement;
}

/** @param newClassement
 * @pdOid 6534279b-2f84-4ab5-889f-d7b010a8c2da */
public void setClassement(int newClassement) {
    Classement = newClassement;
}

/** @pdOid e1efa9c5-c52a-427e-9c1c-2ae355b360f2 */
```

```
public Joueur() {
    // TODO: implement
}

/** @pdOid eb5b2229-7777-41cd-9013-bde49f8a4e40 */
public int getClub() {
    return club;
}

/** @param newClub
 * @pdOid 9959383e-7748-41b9-b4b1-7245fea45ce8 */
public void setClub(int newClub) {
    club = newClub;
}

/** @pdGenerated default getter */
public java.util.Collection<Match> getLesMatches() {
    if (lesMatches == null)
        lesMatches = new java.util.HashSet<Match>();
    return lesMatches;
}

/** @pdGenerated default iterator getter */
public java.util.Iterator getIteratorLesMatches() {
    if (lesMatches == null)
        lesMatches = new java.util.HashSet<Match>();
    return lesMatches.iterator();
}

/** @pdGenerated default setter
 * @param newLesMatches */
public void setLesMatches(java.util.Collection<Match> newLesMatches) {
    removeAllLesMatches();
    for (java.util.Iterator iter = newLesMatches.iterator(); iter.hasNext();)
        addLesMatches((Match)iter.next());
}

/** @pdGenerated default add
 * @param newMatch */
public void addLesMatches(Match newMatch) {
    if (newMatch == null)
        return;
    if (this.lesMatches == null)
        this.lesMatches = new java.util.HashSet<Match>();
    if (!this.lesMatches.contains(newMatch))
    {
        this.lesMatches.add(newMatch);
        newMatch.addLesJoueurs(this);
    }
}
```



```
/** @pdGenerated default remove
 * @param oldMatch */
public void removeLesMatches(Match oldMatch) {
    if (oldMatch == null)
        return;
    if (this.lesMatches != null)
        if (this.lesMatches.contains(oldMatch))
        {
            this.lesMatches.remove(oldMatch);
            oldMatch.removeLesJoueurs(this);
        }
}

/** @pdGenerated default removeAll */
public void removeAllLesMatches() {
    if (lesMatches != null)
    {
        Match oldMatch;
        for (java.util.Iterator iter = getIteratorLesMatches(); iter.hasNext();)
        {
            oldMatch = (Match)iter.next();
            iter.remove();
            oldMatch.removeLesJoueurs(this);
        }
    }
}
}
```

IV.1.2.4 Class matches

```

/*****

```

```

* Module: Match.java

```

```

* Author: p2000087

```

```

* Purpose: Defines the Class Match

```

```

*****/

```

```

package planning;
import java.util.*;

```

```

/** @pdOid 81bb8259-82a5-4e37-8583-0c3a63e352b2 */

```

```

public class Match {

```

```

    /** @pdOid 6105b1fd-d110-4d07-b14a-ace28c0de14b */

```

```

    private int idMatch;

```

```

    /** @pdOid e82c360d-3d08-4ba3-aebf-2d1b8e6f1d22 */

```

```

    private String tour;

```

```

    /** @pdOid 0ab0fe72-c196-49db-a213-d05437e4b37c */

```

```

    private int resultats;

```

```

    /** @pdOid 4cc7d1cd-5a91-44ea-b607-97fa6695d9f2 */

```

```

    protected void finalize() {

```

```

        // TODO: implement

```

```

    }

```

```

    /** @pdRoleInfo migr=no name=Joueur assc=Association_12 coll=java.util.Collection impl=java.util.HashSet
    mult=2..4 side=A */

```

```

    public java.util.Collection<Joueur> lesJoueurs;

```

```

    /** @pdRoleInfo migr=no name=Equipe_ramasseurs assc=Association_9 coll=java.util.Collection

```

```

    impl=java.util.HashSet mult=2..2 side=A */

```

```

    public java.util.Collection<Equipe_ramasseurs> lesRamasseurs;

```

```

    /** @pdRoleInfo migr=no name=Planning assc=Association_10 coll=java.util.Collection

```

```

    impl=java.util.HashSet mult=1..1 side=A */

```

```

    public Planning lePlanning;

```

```

    /** @pdOid 2f5d1d99-3efe-458e-813b-bbf05d15d443 */

```

```

    public int getIdMatch() {

```

```

        return idMatch;

```

```

    }

```

```

    /** @param newIdMatch

```

```

    * @pdOid 08d970f2-7482-4272-a8ad-c30781ba5054 */

```

```

    public void setIdMatch(int newIdMatch) {

```

```

        idMatch = newIdMatch;

```

```

    }

```

```

    /** @pdOid fac77b8f-1181-48b3-b762-977029328d7d */

```

```

    public String getTour() {

```

```

        return tour;

```

```

    }

```



```
/** @param newTour
 * @pdOid 2899bc99-537e-450a-98c0-955fddb4f839 */
public void setTour(String newTour) {
    tour = newTour;
}

/** @pdOid 89379842-7e85-4a3f-aa4a-f18ff7d72546 */
public Match() {
    // TODO: implement
}

/** @pdOid 1d9d4dfe-f4a8-4bf6-a7b2-53d66dec1b5d */
public int getResultats() {
    return resultat;
}

/** @param newResultats
 * @pdOid a94aad20-cef7-43b1-9df6-5bf758b70f61 */
public void setResultats(int newResultats) {
    resultat = newResultats;
}

/** @pdGenerated default getter */
public java.util.Collection<Joueur> getLesJoueurs() {
    if (lesJoueurs == null)
        lesJoueurs = new java.util.HashSet<Joueur>();
    return lesJoueurs;
}

/** @pdGenerated default iterator getter */
public java.util.Iterator getIteratorLesJoueurs() {
    if (lesJoueurs == null)
        lesJoueurs = new java.util.HashSet<Joueur>();
    return lesJoueurs.iterator();
}

/** @pdGenerated default setter
 * @param newLesJoueurs */
public void setLesJoueurs(java.util.Collection<Joueur> newLesJoueurs) {
    removeAllLesJoueurs();
    for (java.util.Iterator iter = newLesJoueurs.iterator(); iter.hasNext();)
        addLesJoueurs((Joueur)iter.next());
}

/** @pdGenerated default add
 * @param newJoueur */
public void addLesJoueurs(Joueur newJoueur) {
    if (newJoueur == null)
        return;
    if (this.lesJoueurs == null)
```

```
        this.lesJoueurs = new java.util.HashSet<Joueur>();
    if (!this.lesJoueurs.contains(newJoueur))
    {
        this.lesJoueurs.add(newJoueur);
        newJoueur.addLesMatches(this);
    }
}

/** @pdGenerated default remove
 * @param oldJoueur */
public void removeLesJoueurs(Joueur oldJoueur) {
    if (oldJoueur == null)
        return;
    if (this.lesJoueurs != null)
        if (this.lesJoueurs.contains(oldJoueur))
        {
            this.lesJoueurs.remove(oldJoueur);
            oldJoueur.removeLesMatches(this);
        }
}

/** @pdGenerated default removeAll */
public void removeAllLesJoueurs() {
    if (lesJoueurs != null)
    {
        Joueur oldJoueur;
        for (java.util.Iterator iter = getIteratorLesJoueurs(); iter.hasNext(); )
        {
            oldJoueur = (Joueur)iter.next();
            iter.remove();
            oldJoueur.removeLesMatches(this);
        }
    }
}

/** @pdGenerated default getter */
public java.util.Collection<Equipe_ramasseurs> getLesRamasseurs() {
    if (lesRamasseurs == null)
        lesRamasseurs = new java.util.HashSet<Equipe_ramasseurs>();
    return lesRamasseurs;
}

/** @pdGenerated default iterator getter */
public java.util.Iterator getIteratorLesRamasseurs() {
    if (lesRamasseurs == null)
        lesRamasseurs = new java.util.HashSet<Equipe_ramasseurs>();
    return lesRamasseurs.iterator();
}

/** @pdGenerated default setter
 * @param newLesRamasseurs */
public void setLesRamasseurs(java.util.Collection<Equipe_ramasseurs> newLesRamasseurs) {
```

```
removeAllLesRamasseurs();
for (java.util.Iterator iter = newLesRamasseurs.iterator(); iter.hasNext();)
    addLesRamasseurs((Equipe_ramasseurs)iter.next());
}

/** @pdGenerated default add
 * @param newEquipe_ramasseurs */
public void addLesRamasseurs(Equipe_ramasseurs newEquipe_ramasseurs) {
    if (newEquipe_ramasseurs == null)
        return;
    if (this.lesRamasseurs == null)
        this.lesRamasseurs = new java.util.HashSet<Equipe_ramasseurs>();
    if (!this.lesRamasseurs.contains(newEquipe_ramasseurs))
    {
        this.lesRamasseurs.add(newEquipe_ramasseurs);
        newEquipe_ramasseurs.addLesMatches(this);
    }
}

/** @pdGenerated default remove
 * @param oldEquipe_ramasseurs */
public void removeLesRamasseurs(Equipe_ramasseurs oldEquipe_ramasseurs) {
    if (oldEquipe_ramasseurs == null)
        return;
    if (this.lesRamasseurs != null)
        if (this.lesRamasseurs.contains(oldEquipe_ramasseurs))
        {
            this.lesRamasseurs.remove(oldEquipe_ramasseurs);
            oldEquipe_ramasseurs.removeLesMatches(this);
        }
}

/** @pdGenerated default removeAll */
public void removeAllLesRamasseurs() {
    if (lesRamasseurs != null)
    {
        Equipe_ramasseurs oldEquipe_ramasseurs;
        for (java.util.Iterator iter = getIteratorLesRamasseurs(); iter.hasNext();)
        {
            oldEquipe_ramasseurs = (Equipe_ramasseurs)iter.next();
            iter.remove();
            oldEquipe_ramasseurs.removeLesMatches(this);
        }
    }
}

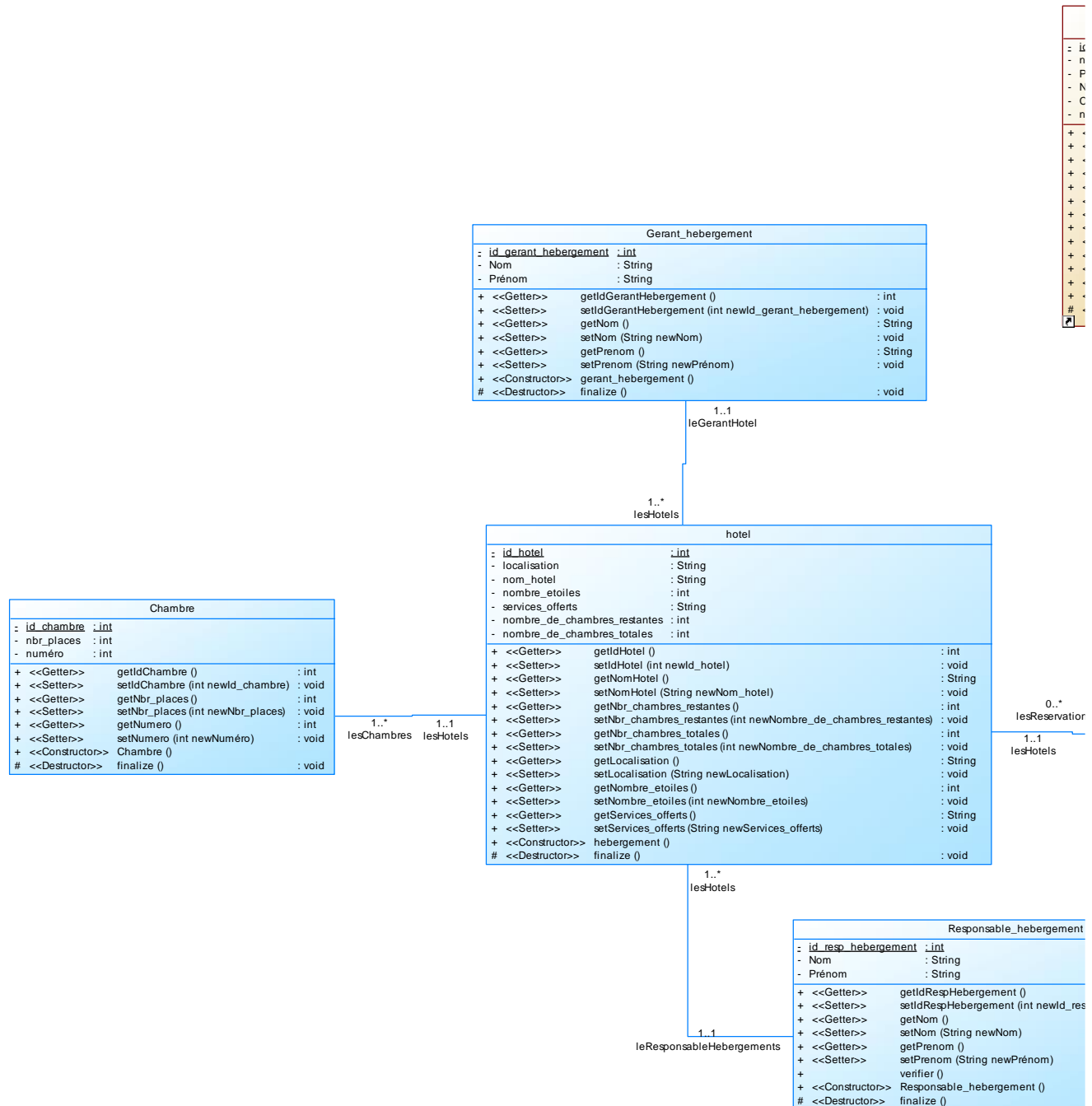
/** @pdGenerated default parent getter */
public Planning getLePlanning() {
    return lePlanning;
}

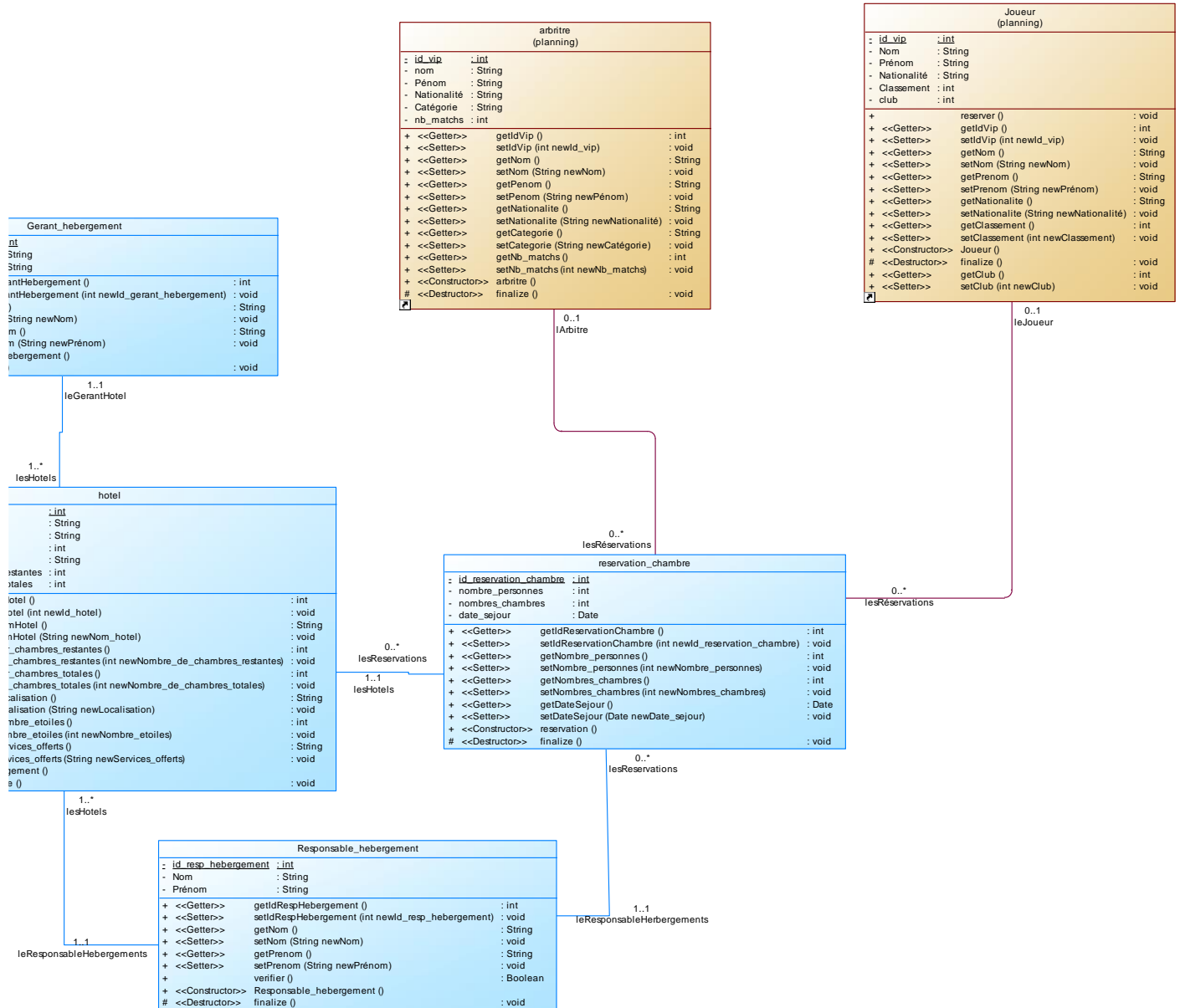
/** @pdGenerated default parent setter
```

```
* @param newPlanning */
public void setLePlanning(Planning newPlanning) {
    if (this.lePlanning == null || !this.lePlanning.equals(newPlanning))
    {
        if (this.lePlanning != null)
        {
            Planning oldPlanning = this.lePlanning;
            this.lePlanning = null;
            oldPlanning.removeLesMatches(this);
        }
        if (newPlanning != null)
        {
            this.lePlanning = newPlanning;
            this.lePlanning.addLesMatches(this);
        }
    }
}
```

IV.2 Package Hebergement

IV.2.1 DCL Hebergement





IV.2.2 Scripts SQL

IV.2.2.1 Table réservation

```

/*=====*/
/* Table : reservation */
/*=====*/
create table reservation
(
    idreservationchambre      int          not null,
    idvip                     int,
    idresphebergement        int          not null,
    arb_idvip                 int,
    idhotel                   int          not null,
    nombre_personnes          int,
    nombres_chambres          int,
    datedejour                datetime,
    primary key (idreservationchambre)
)
type = innodb;

/*=====*/
/* Index : association_3_fk */
/*=====*/
create index association_3_fk on reservation
(
    idhotel
);
/*=====*/
/* Index : association_4_fk */
/*=====*/
create index association_4_fk on reservation
(
    idresphebergement
);
/*=====*/
/* Index : association6_fk */
/*=====*/
create index association6_fk on reservation
(
    arb_idvip
);

```

```

/*=====*/
/* Index : association7_fk */
/*=====*/
create index association7_fk on reservation
(
    idvip
);

alter table reservation add constraint fk_association_3 foreign key (idhotel)
    references hebergement (idhotel) on delete restrict on update restrict;

alter table reservation add constraint fk_association_4 foreign key (idresphebergement)
    references responsable_hebergement (idresphebergement) on delete restrict on update restrict;

alter table reservation add constraint fk_association6 foreign key (arb_idvip)
    references arbitre (idvip) on delete restrict on update restrict;

alter table reservation add constraint fk_association7 foreign key (idvip)
    references joueur (idvip) on delete restrict on update restrict;

```



IV.2.2.2 Table chambre

```

/*=====*/
/* Table : chambre */
/*=====*/
create table chambre
(
    idchambre          int          not null,
    idhotel             int          not null,
    nbr_places          int,
    numero              int,
    primary key (idchambre)
)
type = innodb;

/*=====*/
/* Index : association_7_fk */
/*=====*/
create index association_7_fk on chambre
(
    idhotel
);

alter table chambre add constraint fk_association_7 foreign key (idhotel)
    references hebergement (idhotel) on delete restrict on update restrict;

```

IV.2.3 Code Java

IV.2.3.1 Class réservation

```

/*****
 * Module: reservation.java
 * Author: p2000087
 * Purpose: Defines the Class reservation
 *****/

package hebergement;
import planning.arbitre;
import planning.Joueur;
import java.util.*;

/** @pdOid 29365a28-7abd-45b4-ab9c-c09989c50655 */
public class reservation {
    /** @pdOid 8203fff0-8a4c-46de-84fd-918a42608d67 */
    private int idReservationChambre;
    /** @pdOid cae87b9e-a2db-4df7-ab09-3e313f40f0a6 */
    private int nombre_personnes;
    /** @pdOid 0fbc3498-4ffd-4508-b901-b5ab979f16bc */
    private int nombres_chambres;
    /** @pdOid 02c698f1-ab8d-4d65-bc3d-5984de51e7ad */
    private Date dateSejour;

    /** @pdOid d92055fd-4f90-436e-bc21-a69471e4f273 */
    protected void finalize() {
        // TODO: implement
    }

    /** @pdRoleInfo migr=no name=hebergement assc=Association_3 coll=java.util.Collection
    impl=java.util.HashSet mult=1..1 */
    public hebergement lesHotels;
    /** @pdRoleInfo migr=no name=arbitre assc=association6 mult=0..1 */
    public arbitre lArbitre;
    /** @pdRoleInfo migr=no name=Joueur assc=association7 mult=0..1 */
    public Joueur leJoueur;
    /** @pdRoleInfo migr=no name=Responsable_hebergement assc=Association_4 coll=java.util.Collection
    impl=java.util.HashSet mult=1..1 side=A */
    public Responsable_hebergement leResponsableHerbergements;

    /** @pdOid bbe4fbac-a219-437f-b7cb-8da10c659a9e */
    public int getIdReservationChambre() {
        return idReservationChambre;
    }

    /** @param newIdReservationChambre
     * @pdOid 0f6ed23b-653e-431a-887a-2b55905d16c5 */
    public void setIdReservationChambre(int newIdReservationChambre) {
        idReservationChambre = newIdReservationChambre;
    }

```



```
}

/** @pdOid 8110d2d2-e8e6-450f-a454-a176f557cda8 */
public int getNombre_personnes() {
    return nombre_personnes;
}

/** @param newNombre_personnes
 * @pdOid 3ab39333-4ea0-4989-9f36-bdfc8f55019b */
public void setNombre_personnes(int newNombre_personnes) {
    nombre_personnes = newNombre_personnes;
}

/** @pdOid 61518bf4-5f0c-411b-b2fb-787dd7f6816a */
public int getNombres_chambres() {
    return nombres_chambres;
}

/** @param newNombres_chambres
 * @pdOid 17d13c00-6895-4cd7-bdf1-48d66221514e */
public void setNombres_chambres(int newNombres_chambres) {
    nombres_chambres = newNombres_chambres;
}

/** @pdOid 5d7c83ab-65a3-418a-901b-2b614ed9776d */
public Date getDateSejour() {
    return dateSejour;
}

/** @param newDateSejour
 * @pdOid 4b05f16b-f8d9-498a-a983-1cdadf9c53d9 */
public void setDateSejour(Date newDateSejour) {
    dateSejour = newDateSejour;
}

/** @pdOid 9ab31b11-2012-4ead-b4fc-2b68461b800f */
public reservation() {
    // TODO: implement
}

/** @pdGenerated default parent getter */
public hebergement getLesHotels() {
    return lesHotels;
}

/** @pdGenerated default parent setter
 * @param newHebergement */
public void setLesHotels(hebergement newHebergement) {
    if (this.lesHotels == null || !this.lesHotels.equals(newHebergement))
    {

```

```
        if (this.lesHotels != null)
        {
            hebergement oldHebergement = this.lesHotels;
            this.lesHotels = null;
            oldHebergement.removeLesReservations(this);
        }
        if (newHebergement != null)
        {
            this.lesHotels = newHebergement;
            this.lesHotels.addLesReservations(this);
        }
    }
}

/** @pdGenerated default parent getter */
public Responsable_hebergement getLeResponsableHerbergements() {
    return leResponsableHerbergements;
}

/** @pdGenerated default parent setter
 * @param newResponsable_hebergement */
public void setLeResponsableHerbergements(Responsable_hebergement newResponsable_hebergement) {
    if (this.leResponsableHerbergements == null ||
!this.leResponsableHerbergements.equals(newResponsable_hebergement))
    {
        if (this.leResponsableHerbergements != null)
        {
            Responsable_hebergement oldResponsable_hebergement = this.leResponsableHerbergements;
            this.leResponsableHerbergements = null;
            oldResponsable_hebergement.removeLesReservations(this);
        }
        if (newResponsable_hebergement != null)
        {
            this.leResponsableHerbergements = newResponsable_hebergement;
            this.leResponsableHerbergements.addLesReservations(this);
        }
    }
}
}
```

IV.2.3.2 Class chambre

```

/*****

```

```

* Module: Chambre.java

```

```

* Author: p2000087

```

```

* Purpose: Defines the Class Chambre

```

```

*****/

```

```

package hebergement;

```

```

import java.util.*;

```

```

/** @pdOid 19d119be-7670-445b-8956-3200a75791a1 */

```

```

public class Chambre {

```

```

    /** @pdOid 7e367268-a1db-4998-affb-b94c23d1f1ef */

```

```

    private int idChambre;

```

```

    /** @pdOid bafa2796-83e8-4d90-8a6c-e661f3ed3ec2 */

```

```

    private int nbr_places;

```

```

    /** @pdOid 825c746d-5660-49f6-9992-5df9dada8e06 */

```

```

    private int numero;

```



```

    /** @pdOid d65a2504-e2c7-4a7d-a30e-f62120a506d5 */

```

```

    protected void finalize() {

```

```

        // TODO: implement

```

```

    }

```

```

    /** @pdRoleInfo migr=no name=hebergement assc=Association_7 coll=java.util.Collection
    impl=java.util.HashSet mult=1..1 side=A */

```

```

    public hebergement lesHotels;

```

```

    /** @pdOid 7adb9449-23a9-4d5e-9791-04b162ebba91 */

```

```

    public int getIdChambre() {

```

```

        return idChambre;

```

```

    }

```

```

    /** @param newIdChambre

```

```

    * @pdOid 564b7029-5cf2-4ec7-b95d-50f67d5ce572 */

```

```

    public void setIdChambre(int newIdChambre) {

```

```

        idChambre = newIdChambre;

```

```

    }

```

```

    /** @pdOid 86e67657-e8d3-4d84-b382-1647f844f2ef */

```

```

    public int getNbr_places() {

```

```

        return nbr_places;

```

```

    }

```

```

    /** @param newNbr_places

```

```

    * @pdOid db61ef57-3441-429b-852a-f4dfac9f8056 */

```

```

    public void setNbr_places(int newNbr_places) {

```

```

        nbr_places = newNbr_places;

```

```

    }

```

```
/** @pdOid 976473b6-f603-4c88-a832-798f50d99998 */
public int getNumero() {
    return numero;
}

/** @param newNumero
 * @pdOid cb449cdf-bfb4-43ea-a48e-9f13e3be1ef8 */
public void setNumero(int newNumero) {
    numero = newNumero;
}

/** @pdOid beed9775-18af-46fd-a4a3-b15ff886e8a2 */
public Chambre() {
    // TODO: implement
}

/** @pdGenerated default parent getter */
public hebergement getLesHotels() {
    return lesHotels;
}

/** @pdGenerated default parent setter
 * @param newHebergement */
public void setLesHotels(hebergement newHebergement) {
    if (this.lesHotels == null || !this.lesHotels.equals(newHebergement))
    {
        if (this.lesHotels != null)
        {
            hebergement oldHebergement = this.lesHotels;
            this.lesHotels = null;
            oldHebergement.removeLesChambres(this);
        }
        if (newHebergement != null)
        {
            this.lesHotels = newHebergement;
            this.lesHotels.addLesChambres(this);
        }
    }
}
}
```

IV Annexes

IV.1 Annexe n°1 : Maquettes du planning des matchs

planning match - Menu



OPEN PARC
AUVERGNE-RHÔNE-ALPES
LYON



MatchScoreConnexion


Présentation

Fondé en 1987, l'Open de Tennis de Lyon fait partie des tournois internationaux comptant pour le classement ATP 250. En 2017, il devient l'Open Parc Auvergne-Rhône-Alpes Lyon (ou Open Parc ARAL) et est organisé sur terre battue au Parc de la Tête d'Or, durant la semaine précédant Roland Garros. Le gagnant remporte un prix d'un peu plus de 500 000€ (tournois de Master 1000 : plus de 3M€).

L'Open comprend deux tournois : le Simple Messieurs et le Double Messieurs.

Plan



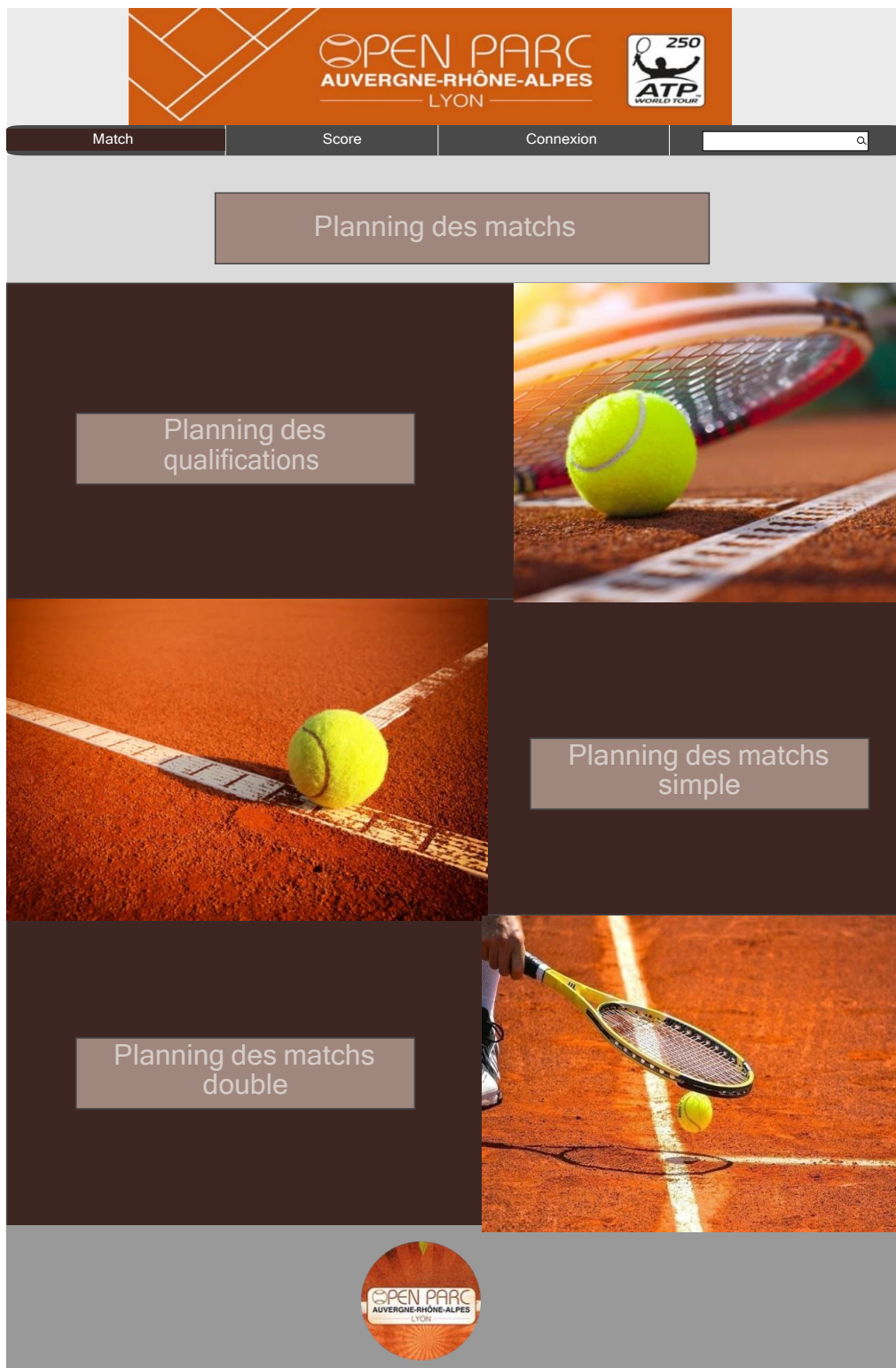


Cette page est la première page de l'application.

Tout le monde peut la voir.

Elle présente le tournoi ainsi que le plan des lieux.

planning match - Match



En cliquant sur l'onglet match on voit les différentes catégories de match.

Quand on clique sur une catégorie on est redirigé vers une autre page.



www.website.com

1

Mercredi 5 novembre

13h30/15h			
Terrain 5		Terrain 6	
LOMBARDO	Alex	CASARA	Quentin
WEYLAND	Floris	NITTLER	Jérémy
BELLONI	Vincent	NOIR	Zacharie
NICOLI	Paul	KOSOWSKI	Mattéo
NOESSER	Nathan	BUJOR	Darius
FEDELI	Sacha	FEDELI	Louis

15h/16h30			
Terrain 5		Terrain 6	
SCHWARTZ	Pauline	BOYER	Raphael
VATREY	Lena	VRANCKEN	Noah
SANFILIPPO	Chloe	PETITNICOLAS	Alex
MACHERET	Mathilde	SPECHT R.	Léandro
CHENILLOT	Juliette	LEPIT	Anthony
POETZ	Héloïse	CASARA	Bastien

16h30/18h00			
Terrain 5		Terrain 6	
WEILAND	Julien	BATCHELOR	Ben
GRZYWCZAK	Mathias	KIMOUCHE	Bilel
PETITNICOLAS	Tom	KABLITZ	Anthony
MANCUSO	Mateo	BILLON	Mike
HERMANN	Yohan	MUSQUIN	Clément
MEREL	Baptiste	BONDUE	Arthur
MANCINO	Lucas	DEBANDE	Mattéo

Mercredi 12 novembre

13h30/15h			
Terrain 5		Terrain 6	
PICKENHAGEN	Noah	BARGMANN	Claire
MUSQUIN	Maxime	MESLARD S.	Coline
DALI	Hedi	CHENILLOT	Léa
NGUYEN	Luong	JANTZEN	Gabrielle
SCHUMACHER	Maxime	ARIB F.	Léopoldine
MESNIL	Oscar	MANCINO	Lucas

15h/16h30			
Terrain 5		Terrain 6	
LENTCHAT	Lucas	BOURDOU	Romane
BALSAMO	Nicolas	SULTANO	Marie
DIEBOLD	Tom	HAYANI	Marina
SCHOEPP	Thomas	REA-MANIN	Nina M.
DALI	Elias	KAMINSKI	Clara
SCHAFF	Cesar	BRUNCHER	Anais
ESTEVE D.	Antoine		

16h30/18h00			
Terrain 5		Terrain 6	
NICOLAS-M.	Louis	MERAN	Margaux
DIEBOLD	Ben	MAZZERO	Alice
MARION	Tom	LECOMTE	Capucine
VILOHR	Arthur	PAWELSKI	Juliette
LAGRANGE	Timeo	ELLES	Lea
FAGOTTO	Jules	ARIB-F.	Adele
		DANIEL	Clara
		BAZARD	Emmy
		BATCHELOR	Chloe
		FAGOTTO	Elisa
		RICHARD	Melissa

On arrive sur cette page quand on clique sur une des catégories de match.
On y retrouve les plannings de jeux.
Ils sont accessibles par tout le monde.

planning match - Classement

www.website.com

OPEN PARC
AUVERGNE-RHÔNE-ALPES
LYON

250
ATP
WORLD TOUR

Match

Score

Connexion

Score

Numéro du match	Type de match	joueur 1	joueur 2	vainqueur
1	qualification	Djokovic	Nadal	Nadal
2	qualification	Federer	Murray	Murray
3	qualification	Zverev	Thiem	Thiem
4	qualification	Wawrinka	Ferrer	Wawrinka
5	qualification	Berdych	Nishikori	Nishikori
6	qualification	Del Potro	Tsonga	Del Potro
7	qualification	Dimitrov	Goffin	Dimitrov
8	qualification	Fish	Cilic	Cilic
9	qualification	Raonic	Almagro	Almagro
10	qualification	Youzhny	Isner	Isner
11	qualification	Simon	Bautista-Agut	Simon
12	qualification	Haas	Pouille	Haas
13	qualification	Schwartzman	Melzer	Melzer
14	qualification	Anderson	Gasquet	Anderson
15	qualification	Llodra	Verdasco	Llodra

OPEN PARC
AUVERGNE-RHÔNE-ALPES
LYON

En cliquant sur l’onglet Score on arrive sur une page avec le récapitulatif de tous les matchs joués et le vainqueur.



planning match - Connexion

OPEN PARC
AUVERGNE-RHÔNE-ALPES
LYON

250
ATP
WORLD TOUR

Match

Score

Connexion

Connexion

Identifiant :

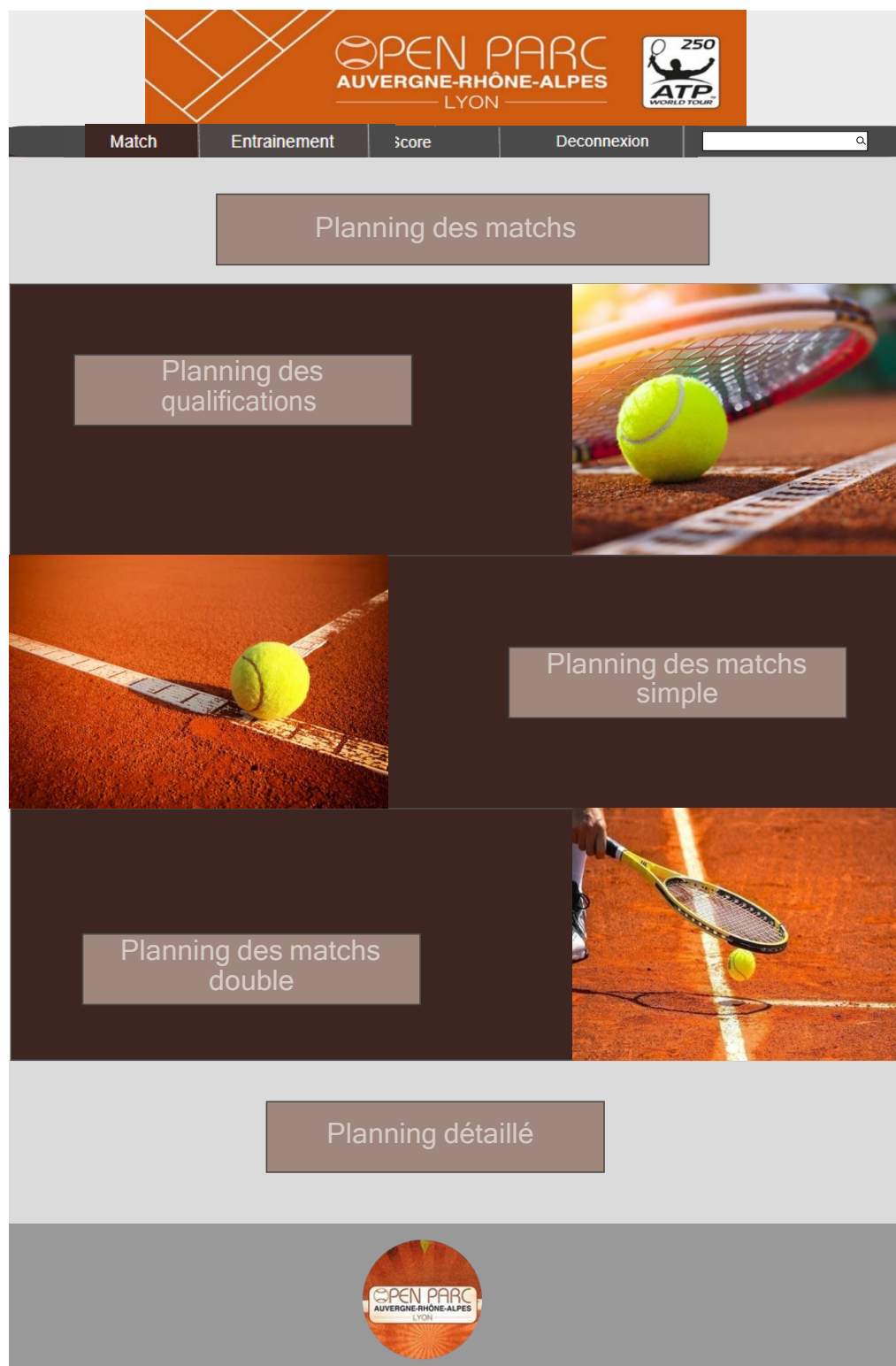
Mot de passe :

Valider

OPEN PARC
AUVERGNE-RHÔNE-ALPES
LYON

Le système de connexion sert aux joueurs et aux organisateurs d'avoir des fonctionnalités en plus.

planning match - Match joueur



Quand un joueur se connecte de nouvelles fonctionnalités apparaissent. Dans l'onglet match on retrouve le bouton planning détaillé.



planning match - Planning détaillé

OPEN PARC
AUVERGNE-RHÔNE-ALPES
LYON

250

ATP

WORLD TOUR

Match

Score

Saisir un score

Modification match

Déconnexion

Planning détaillé

numéro du match	type de match	joueur 1	joueur 2	Arbitres	Equipe de ramasseurs de balles
1	qualification	Djokovic	Nadal	Mohamed Lahyani (Suède), Kader Nouni (France), Carlos Bernardes (Brésil), Carlos Ramos (Portugal), Cédric Mourier (France), Jake Garner (USA),	3

Les autres planning arrivent...

revenir au planning

OPEN PARC
AUVERGNE-RHÔNE-ALPES
LYON

Une fois qu'on clique sur planning détaillé, les plannings détaillés des matchs avec les arbitres et l'équipe de ramasseurs de balles marqués.



planning match – Entraînement joueur

OPEN PARC
AUVERGNE-RHÔNE-ALPES
LYON

250
ATP
WORLD TOUR

Match

Entraînement

Score

Deconnexion

Réservation d'un court pour un entraînement

Court 1 :

☒ 11h

☐ 13h

☐ 17h

Court 2 :

☐ 11h

☒ 13h

☐ 17h

Court 3 :

☒ 11h

☐ 17h

Valider

Court réservé


Court 3 :

☒ 13h

Annuler

OPEN PARC
AUVERGNE-RHÔNE-ALPES
LYON

Un onglet entraînement apparaît aussi.

Il permet au joueur de réserver un court pour un entraînement quand il n'y a pas de match qui s'y joue. 

Le joueur peut voir les créneaux qu'il a réservés et les annulés.

Les horaires déjà réservés n'apparaissent plus dans les possibilités de réservation.

planning match - Saisir score

OPEN PARC
AUVERGNE-RHÔNE-ALPES
LYON

250
ATP
WORLD TOUR

Match

Score

Saisir un score

Modification match

Déconnexion

Saisir un score

Données

Type de match :


Joueur 1 :


joueur 2 :

Vainqueur :

[Modifier les données d'un match](#)

Valider





Quand un organisateur se connecte les onglets **saisirent** un score et modification match apparaissent.

Ici dans saisir un score l'organisateur saisit les données d'un match qui sera affiché ensuite dans score.

planning match - Modifier score

OPEN PARC
AUVERGNE-RHÔNE-ALPES
LYON

250
ATP
WORLD TOUR

Match

Score

Saisir un score

Modification match

Déconnexion

Saisir un score

Donnée

Numéro du match :

Type de match :

Joueur 1 :

joueur 2 :

Vainqueur :

[Saisir un score](#)

Valider

OPEN PARC
AUVERGNE-RHÔNE-ALPES
LYON

L'organisateur peut aussi modifier le score d'un match en cliquant sur le lien.
Il lui suffit juste d'indiquer le numéro du match.

planning match - Modification planning

OPEN PARC
AUVERGNE-RHÔNE-ALPES
LYON

250

ATP
WORLD TOUR

Match

Score

Saisir un score

Modification match

Déconnexion

Modifier un match

Données du match

Numéro du match

Equipe/joueur 1 :

Equipe/joueur 2 :

Arbitres :

Equipe de ramasseurs
de balles :

Valider


OPEN PARC
AUVERGNE-RHÔNE-ALPES
LYON

Il peut aussi modifier un match en rentrant le numéro du match et en indiquant les nouvelles données.
Les données seront automatiquement mises à jour.



IV.2 Maquettes des hébergement

www.website.com



Connexion

Recherche :

Connexion

Identifiant

Mot de passe

Valider

COMPANY LOGO

Duis sodales vitae ipsum non
facileis. Donec convallis diam in
arcu mollis pulvinar. Donec vitae
metus at enim interdum

ABOUT

SERVICES

INFO

FAQ

CONTACT

ICON

ICON

ICON

ICON

hébergement - Menue

Ceci est la page sur laquelle arrivent les joueurs quand ils sont connectés.

Ils peuvent voir tous les hôtels de libres et en cliquant sur le nom ça les amène à la page de réservation.

www.website.com

Menue Hotels

Mes réservations

Déconnexion

Recherche :

Hôtels disponible :

Hôtel Méditerranée

Hôtel Cordoba Center

Hôtel Fairmont

Hôtel Mama Shelter

Hôtel radisson blu

Hôtel MOB

COMPANY LOGO

Duis sodales vitae ipsum non facilisis. Donec convallis diam in arcu mollis pulvinar. Donec vitae metus at enim interdum.

ABOUT

SERVICES

INFO

FAQ

CONTACT

ICON

ICON

ICON




ICON

hébergement - Sélection hôtel

www.website.com

Menue Hôtels Mes réservations Déconnexion Recherche :

Hôtel Mama Shelter



Description:
Situé dans un quartier commerçant, cet hôtel branché se trouve à 5 minutes à pied de la gare de Lyon-Jean-Macé, à 2 km de la place Bellecour et à 3 km de la basilique Notre-Dame de Fourvière.

Réserver

Information supplémentaire :

Localisation : 13 Rue Domer, 69007 Lyon

Nombre de place par chambre : 2 ou 4

Nombre de chambres disponible : 20

Numéro de téléphone : 04 78 02 58 00

Wi-Fi gratuit

Petit déjeuner payant

Parking

Accès handicapés

Climatisation

Centre d'affaires

COMPANY LOGO

Duis sodales vitae ipsum non facilisis. Donec convallis diam in arcu mollis pulvinar. Donec vitae metus at enim interdum.

ABOUT SERVICES

INFO

FAQ

CONTACT

ICON

ICON


ICON

ICON

Les joueurs arrivent sur cette page quand ils ont cliqués sur un hôtel.

Il y a toutes les informations nécessaires sur l’hôtel

www.website.com



Menu Hotels

Mes réservations Déconnexion

Recherche :

Hôtel Mama Shelter

Informations :

Numéro de téléphone :

Nombre de personnes :

Nom :

Prénom :

Email :

Contraintes :

Type de chambre

☒ Simple

☐ Double

☐ XXL Mama Family

Valider

COMPANY LOGO

Duis sodales vitae ipsum non
facilisis. Donec convallis diam in
arcu mollis pulvinar. Donec vitae
metus at enim interdum.

ABOUT
SERVICES

INFO
FAQ
CONTACT

ICON

ICON

ICON

ICON

Page de réservation d'un hôtel


Ceci est la page sur laquelle arrivent les arbitres quand ils sont connectés.

Ils peuvent voir tous les hôtels de libres et ceux déjà réservés que par des arbitres et en cliquant sur le nom ça les amène à la page de réservation.


www.website.com

[Menue Hotels](#) [Mes réservations](#) [Déconnexion](#) Recherche :


Hôtels disponible :




Hôtel des remparts



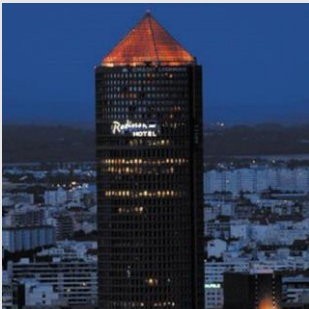
Hôtel Cordoba Center




Hôtel Fairmont



Hôtel Okko



Hôtel radisson blu



Hôtel MOB

COMPANY LOGO

Duis sodales vitae ipsum non facilisis. Donec convallis diam in arcu mollis pulvinar. Donec vitae metus at enim interdum.

ABOUT

SERVICES

INFO

FAQ

CONTACT

ICON

ICON

ICON

ICON

www.website.com

[Menue Hotels](#) [Mes réservations](#) [Déconnexion](#) [Recherche :](#)

Hôtel Okko

Description:
Situé en face du Rhône, cet hôtel ultra tendance se trouve à 7 minutes à pied de la place du Maréchal-Lyautey et à 13 minutes du musée des Beaux-Arts de Lyon.

Réserver

Information supplémentaire :

Localisation : 14 Bis Quai du Général Sarraill, 69006 Lyon

Nombre de place par chambre : 2

Nombre de chambres disponible : 60

Numéro de téléphone : 04 28 00 02 50

Wi-Fi gratuit

Petit déjeuner payant

Accès handicapés

Climatisation

Service de blanchisserie

Accepte les animaux

COMPANY LOGO

Duis sodales vitae ipsum non facilisis. Donec convallis diam in arcu mollis pulvinar. Donec vitae metus at enim interdum.

ABOUT

SERVICES

INFO

FAQ

CONTACT

ICON

ICON


ICON

ICON

Les arbitres arrivent sur cette page quand ils ont cliqués sur un hôtel.

On retrouve toutes les informations nécessaires sur l’hôtel

www.website.com



Menu Hotels Mes réservations Déconnexion

Recherche :

Hôtel Okko

Informations :

Numéro de téléphone :

Nombre de personnes :

Nom :

Prénom :

Email :

Contraintes :

Type de chambre

☒ Classique

☐ Premium

Valider

COMPANY LOGO

Duis sodales vitae ipsum non facilisis. Donec convallis diam in arcu mollis pulvinar. Donec vitae metus at enim interdum.

ABOUT SERVICES

INFO

FAQ

CONTACT

ICON

ICON


ICON

ICON

Ceci est la page de réservation d'un hôtel sélectionné par un arbitre.

hébergement - Mes réservations

www.website.com



Menue Hotels

Mes réservations

Déconnexion

Recherche :

Mes réservations

Hotels OKKO :

Numéro de téléphone : 06-XX-XX-XX-XX

Nombre de personnes : 5

Nom : Nadal

Prénom : Raphael

Email : raphael.nadal@gmail.com

Contraintes : Aucunes

Types de chambres : classiques

Nombre de personnes : 4

Nombre de chambres : 2

Modifier

COMPANY LOGO

Duis sodales vitae ipsum non
facilisis. Donec convallis diam in
arcu mollis pulvinar. Donec vitae
metus at enim interdum.

ABOUT

SERVICES

INFO

FAQ

CONTACT

ICON

ICON

ICON


ICON

En cliquant sur l’onglet mes réservations les joueurs et les arbitres peuvent voir l’hôtel qu’ils ont choisi ainsi que les informations qu’ils ont rentré



hébergement - Modification des données

www.website.com



Menu Hotels Mes réservations Déconnexion

Recherche :

Hôtel Okko

Informations :

Numéro de téléphone :

Nombre de personnes :

Nom :

Prénom :

Email :

Contraintes :

Type de chambre

☒ Classique

☐ Premium

Valider

COMPANY LOGO

Duis sodales vitae ipsum non facilisis. Donec convallis diam in arcu mollis pulvinar. Donec vitae metus at enim interdum.

ABOUT

SERVICES

INFO

FAQ

CONTACT

ICON

ICON

ICON

ICON

En cliquant sur modifier dans une réservation arrive sur cette page pour les modifier.


hébergement - menue hôtel gérant

Ceci est la page sur laquelle arrivent un gérant d’hôtel quand ils sont connectés.

Ils peuvent voir tous les hôtels qu’ils gèrent.

Il a aussi la possibilité devoir ou modifier les données de ses hôtels en cliquant sur un bouton.

www.website.com




Mes hôtels

Déconnexion

Recherche :

Mes hôtels :

Hôtel Okko :



Voir les données

Modifier les données

COMPANY LOGO

Duis sodales vitae ipsum non facilisis. Donec convallis diam in arcu mollis pulvinar. Donec vitae metus at enim interdum.

ABOUT

SERVICES

INFO

FAQ

CONTACT


ICON

ICON

ICON

ICON

www.website.com







Mes hôtels

Déconnexion

Recherche :

Hôtel Okko



Description:

Situé en face du Rhône, cet hôtel ultra tendance se trouve à 7 minutes à pied de la place du Maréchal-Lyautey et à 13 minutes du musée des Beaux-Arts de Lyon.


Information supplémentaire :


Localisation : 14 Bis Quai du Général Sarrail, 69006 Lyon


Nombre de place par chambre : 2


Nombre de chambres disponible : 60


Numéro de téléphone : 04 28 00 02 50


 Wi-Fi gratuit


 Petit déjeuner payant

 Accès handicapés

 Climatisation

 Service de blanchisserie

 Accepte les animaux



COMPANY LOGO

Duis sodales vitae ipsum non facilisis. Donec convallis diam in arcu mollis pulvinar. Donec vitae metus at enim interdum.

ABOUT

SERVICES

INFO

FAQ

CONTACT

ICON

ICON

ICON


ICON

En cliquant sur voir les données, cette page apparait avec la description complète de l’hôtel.

hébergement - Modification données hôtel gérant

En cliquant sur modifier données, le gérant arrive sur cette page où, ils peuvent saisir les données de l'hôtel.

www.website.com



Mes hôtelsDéconnexionRecherche :

Hôtel OkkO

Informations :

Nom de l'hôtel :

Description :

Localisation :

Nombre de place par chambre :

Nombre de place disponible :

Numéro de téléphone :

Type de chambre :

Valider

COMPANY LOGO

Duis sodales vitae ipsum non facilisis. Donec convallis diam in arcu mollis pulvinar. Donec vitae metus at enim interdum.

ABOUT

SERVICES

INFO

FAQ

CONTACT

ICON

ICON

ICON

ICON



hébergement – Menue respo tournoi

Quand le responsable de l'hébergement du tournoi se connecte ils arrivent sur cette page, ou il peut voir tous les hôtels qui sont proposés.

www.website.com

Les hôtels


Les réservations

Ajouter un hôtel


Déconnexion

Recherche :


Tous les hôtels :




Hôtel des remparts



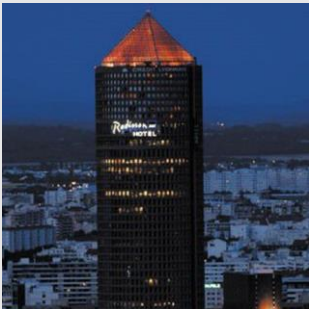
Hôtel Mama Shelter




Hôtel Fairmont



Hôtel Okko



Hôtel radisson blu



Hôtel MOB

COMPANY LOGO

Duis sodales vitae ipsum non facilisis. Donec convallis diam in arcu mollis pulvinar. Donec vitae metus at enim interdum.

ABOUT

SERVICES

INFO

FAQ

CONTACT

ICON

ICON

ICON


ICON

hébergement - Ajout hôtel respo

Cette page est faite pour l'ajout d'un hôtel.

Le responsable peut ajouter les hôtels et les gérants peuvent modifier les données.

www.website.com



[Les hôtels](#) [Les réservations](#) [Ajouter un hôtel](#) [Déconnexion](#) Recherche :

Informations :

Nom de l'hôtel :

Nom du gérant de l'hôtel :

Description :

Localisation :

Nombre de place par chambre :

Nombre de place disponible :

Numéro de téléphone :

Type de chambre :

Valider

COMPANY LOGO

Duis sodales vitae ipsum non facilisis. Donec convallis diam in arcu mollis pulvinar. Donec vitae metus at enim interdum.

ABOUT SERVICES

INFO

FAQ

CONTACT

ICON


ICON

ICON

ICON



www.website.com



Les hôtels

Les réservations

Ajouter un hôtel

Déconnexion

Recherche :

Les réservations

Hotels OKKO :

Numéro de téléphone : 06-XX-XX-XX-XX

Nombre de personnes : 5

Nom : Nadal

Prénom : Raphael

Email : raphael.nadal@gmail.com

Contraintes : Aucunes

Types de chambres : classiques

Nombre de personnes : 4

Nombre de chambres : 2

Valider

Refusez

Hotels Mama :

Numéro de téléphone : 06-XX-XX-XX-XX

Nombre de personnes : 1

Nom : Bacchi

Prénom : Pierre

Email : pierre.bacchi@gmail.com

COMPANY LOGO

Duis sodales vitae ipsum non
facilisis. Donec convallis diam in
arcu mollis pulvinar. Donec vitae
metus at enim interdum.

ABOUT

SERVICES

INFO

FAQ

CONTACT

ICON

ICON

ICON

En cliquant sur l’onglet « Les réservations » le gérant à accès a toutes les réservations faites. Il peut du coup les accepter ou les refuser.

