

Rapport D’analyse et de Programmation

CPOA

Hana BEN HAMIDA | Marion MIRABEL | Bastien MORONVAL

[Introduction 2](#_Toc30591246)

[Description 2](#_Toc30591247)

[Fiche du modèle Open Tennis 2](#_Toc30591248)

[Diagramme 3](#_Toc30591249)

[Sous-projet VIP 3](#_Toc30591250)

[Modèle physique de données 6](#_Toc30591251)

[Modèle Gestion VIP 6](#_Toc30591252)

[Gestion du code 6](#_Toc30591253)

[Partie IHM 7](#_Toc30591254)

[Storyboard 7](#_Toc30591255)

[Volet Technique 8](#_Toc30591256)

[Etat d’avancement et points d’amélioration 8](#_Toc30591257)

[Description de l’architecture 8](#_Toc30591258)

[Retour à l’analyse 9](#_Toc30591259)

[Méthode de test 9](#_Toc30591260)

[Point GIT 10](#_Toc30591261)

[Planning 11](#_Toc30591262)

[Planning prévisionnel 11](#_Toc30591263)

[Répartition des tâches prévisionnelles 11](#_Toc30591264)

[Planning réel des tâches 12](#_Toc30591265)

[Bilan personnel : Hana 13](#_Toc30591266)

[Bilan personnel : Bastien 14](#_Toc30591267)

[Bilan personnel : Marion 14](#_Toc30591268)

# Introduction

## Description

L’application que nous devons réaliser se fera en Web et en Java.

Elle concerne l’Open de Tennis de Lyon et permettre de simplifier la gestion des VIP et le planning des matchs.

Pour la gestion des VIP qui se fera par une application web, nous auront besoin de mettre en place une base de données qui répertoriera les fiches des différents VIP avec leurs attributs (photo, nom, prénom, âge, nationalité, classement ATP, nombre de tournois grands chelem, prise en charge (totale, transport seulement, repas+ hébergement, hébergement seulement …), un compagnon attitré. Il faudra ne pouvoir consulter le site que par connexion, ou sinon il pourra s’inscrire et puis se connecter, et si celle-ci se fait que cela débloque les fonctionnalités de l’application.

Une fois connecter, l’utilisateur pourra créer de nouvelles fiches VIP, les modifier ou simplement les consulter, à partir d’une liste présente sur la page detailVIP.

# Fiche du modèle Open Tennis

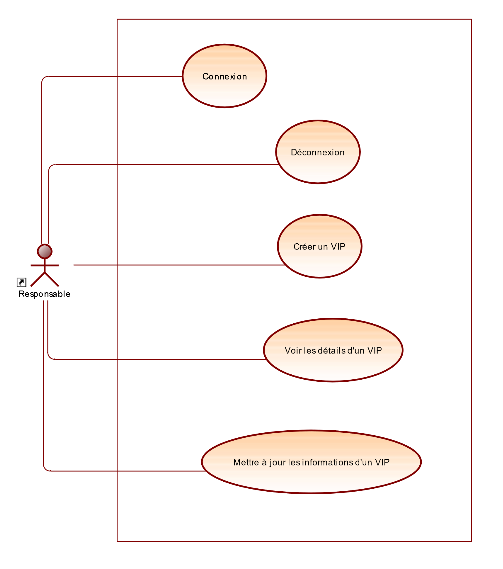
|  |  |
| --- | --- |
| Nom  Code  Langage objet  Commentaire  Auteur  Version | Open Tennis de Lyon  Tennis-Projet  Java, PHP  Modèle de l’application Java qui gère les plannings et de l’application Web qui gère les VIP  Marion MIRABEL, Bastien MORONVAL, Hana BEN HAMIDA |

# Diagramme

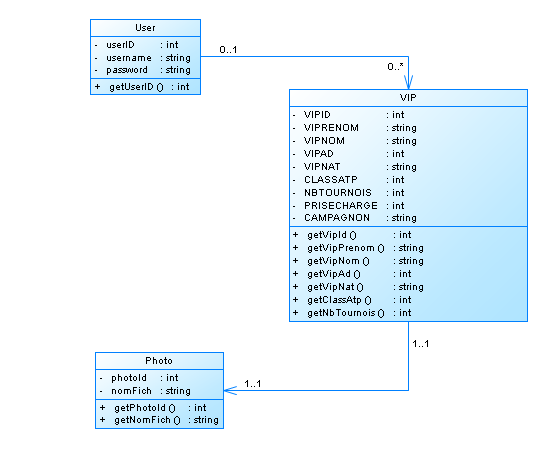
Nous avons décidé de découper le projet en deux sous-projet : un pour la partie gestion de VIP et un pour la partie application java.

## Sous-projet VIP

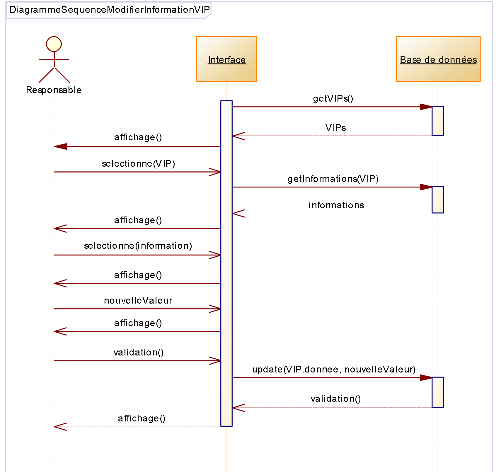
### Diagramme de Cas d’Utilisation



### Diagramme de Classe

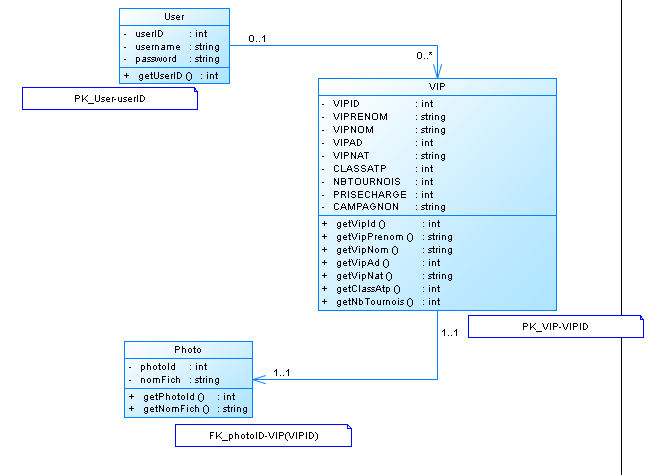


### Diagramme de Séquence (Modifier Informations VIP)



# Modèle physique de données

## Modèle Gestion VIP

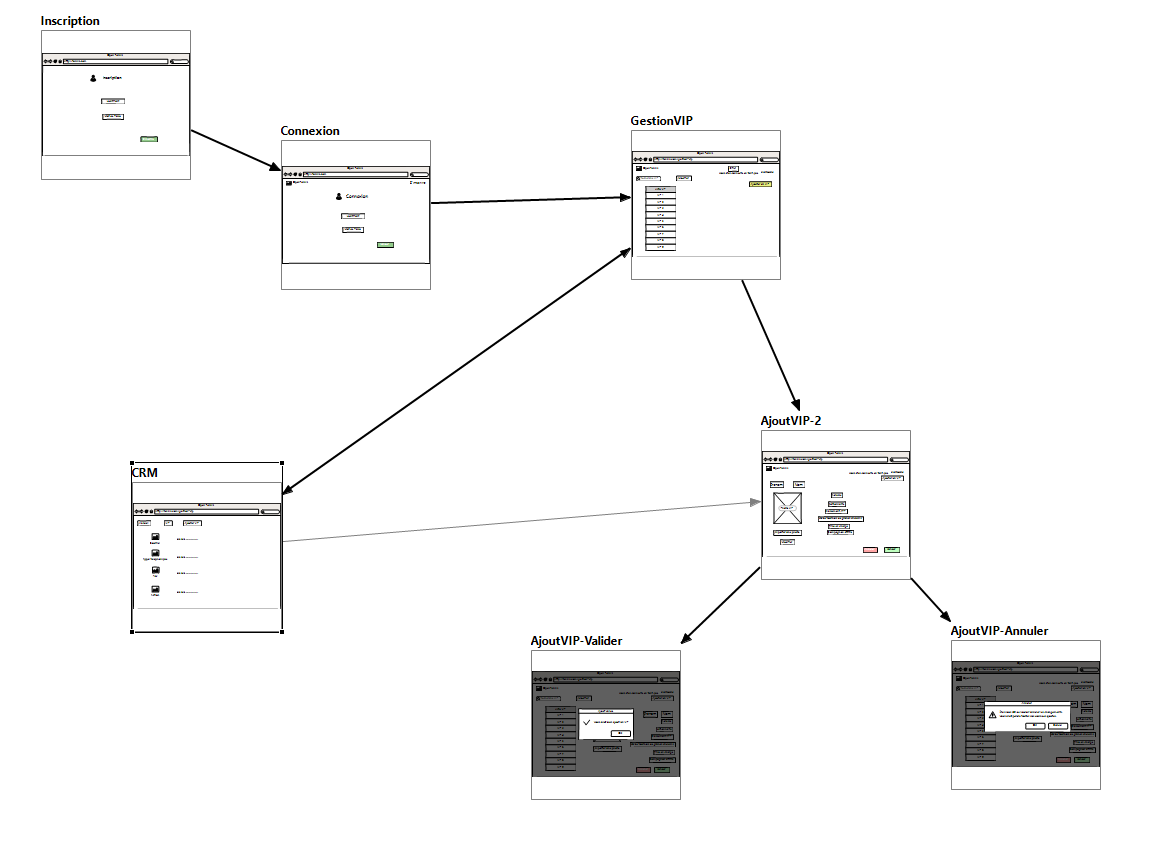


## Gestion du code

Pour la création des tables, nous n’avons pas utiliser de script afin de les créer automatiquement, on les a donc créées à la main sur phpMyAdmin.

# Partie IHM

## Storyboard



# Volet Technique

## Etat d’avancement et points d’amélioration

Notre site est terminé. Il permet de visualiser clairement les VIP, de les consulter en détail, de les supprimer, de les modifier et d'en créer de nouveaux. Lorsqu’on se connecte en mode administrateur, on a un menu supplémentaire qui s’affiche, le menu CRM pour consulter les échanges et actions faites par chaque VIP, et en tant qu’administrateur on peut rajouter des actions. Il reste cependant un bug sur les photos des VIP : avec un serveur web locale PHP 7 (que cela soit sur Linux ou Windows) la photo est bien ajoutée au dossier photo mais sur le serveur de l’IUT (PHP 5) la photo ne s’ajoute pas. Pourtant elle est bien enregistrée dans la base de données. Nous pensons qu’il s’agit d’un problème de version PHP. Il resterait l’ergonomie du site à améliorer.

L'application Java est fonctionnelle. Elle permet de visualiser le planning des matchs prévus ainsi que de réserver un cours pour un entrainement si besoin est. Quand on la lance une première page apparait avec le planning. On peut ensuite se connecter afin de modifier les scores, ajouter des matchs ou réserver un cours pour s’entrainer. Elle a été développer en suivant le modèle MVC. Une base de données a été créée sur PhpMyAdmin, elle comporte 8 tables. Nous avons dû rentrer tous les différents joueurs ainsi que leur nationalité. Notre application fonctionne, il manque l’ajout de matchs double.

## Description de l’architecture

### PHP

Nous utilisons une architecture MVC. En PHP nous avons donc les entités pour les classes métier, les contrôleurs, les modèles pour les accès à la base de données et les vues. Les contrôleurs des pages sont appelés par le fichier index.php à la racine du site, si le contrôleur existe il va vérifier que la vue existe avant d’effectuer les traitements puis il appelle la vue correspondante. Nous utilisons un model DAO qui possède une connexion à la base de données. Les modèles propres à chaque entité sont des extensions du modèle DAO. Chaque modèle métier implémente les méthodes de l’interface correspondante. Ainsi nous n’avons qu’à ajouter les fonctionnalités dont nous avons besoin dans les interfaces puis nous les implémentons. Les attributs informations enregistrées en base sont récupérées par les modèles (PHP) qui les stockent dans des objets qu’ils retournent. À chaque modification de ces objets, nous enregistrons les modifications en base pour sauver le travail de l’utilisateur.

### Java

L’application java a aussi été faite à partir du modèle MVC. Dans le package metierDAO, nous avons toutes les fonctions qui permettent la liaison aux bases de données. Ces données sont récupérées et sont utilisées grâce aux classes métier qui sont composées des mêmes éléments que nos bases de données. Nous avons ensuite les vues qui sont appelées pour l’affichage. Notre application est composée de trois vues importantes : Main (vue quand on lance l’application) ensuite on a une page de connexion pour utiliser toutes les fonctionnalités il faut se connecter avec l’identifiant suivant J1 et en mot de passe J1mdp. La vue qui vient après est Modification car une fois connecté on peut modifier le planning.

## Retour à l’analyse

### PHP

Nous avons eu des difficultés à réaliser l’analyse bien que nous y ayons passé cinq séances. Au final les maquettes nous ont beaucoup aidé mais il y avait des incohérences dans nos diagrammes de classes et nos diagrammes de séquence / cas d’utilisation. Lors du développement nous nous sommes rendu qu’il manquait certains éléments dans la base de données. Pour l’IHM nous nous sommes basés sur les maquettes plutôt que sur les diagrammes de cas d’utilisation, au final nous avons bien fait car nos diagrammes avaient été réalisés à partir de ces maquettes sauf qu’ils étaient incomplets. Les diagrammes de séquence nous ont aidé bien qu’ils ne soient pas très précis.

### Java

Nous avons eu un peu de mal à bien cerner le sujet au début, en cause, les nombreuses incohérences, mais après quelques séances et de nombreux tests, nous avons pu déterminer ce que nous devions faire et les pièges que nous devions éviter. C’est principalement au cours du développement que le projet s’est éclairci, nous nous sommes rendus compte que nous n’avions pas défini toutes les fonctionnalités nécessaires, mais ces dernières étaient assez rapides à rajouter. En ce qui concerne la base de données, elle n’était pas clairement définie dès le début mais a évolué tout au long du projet, pour s’adapter à chaque fois aux nouvelles fonctionnalités que nous rajoutions.

## Méthode de test

### PHP

Notre méthode de test consistait à tester toutes les fonctionnalités que nous avions implémenté. Cette méthode était très efficace pour ce projet car le nombre de fonctionnalités était très réduit il était très rapide pour nous de voir si cela fonctionnait. Nous utilisions le mode débug en PHP sur les modèles pour afficher plus de messages d’erreur (model = new model(true) pour activer le débug). Et nous regardions les valeurs dans la base de données depuis phpmyadmin pour vérifier que les traitements avaient bien été effectués. De même pour le Java, nous avons beaucoup utilisé les messages d’erreur de la console pour régler les problèmes à la source.

### Java

Nous avons testé notre code de deux façons, manuellement pour ce qui est du logiciel en lui-même et automatiquement pour ce qui est de l’accès à la base de données.

Nous avons essayé différentes interactions qu’aurait pu avoir un utilisateur avec le logiciel, qu’elles soient prévues par l’application (ajouter un match, se connecter) ou qu’elles soient sources d’erreurs : « si je mets deux fois le même jour pour un match », « si je mets deux matchs au même moment » …

Pour ce qui est des tests automatiques, nous nous somme servis de Junit 5, qui permet de créer des méthodes de test et de comparer les résultats obtenus avec les résultats attendus. Ce sont les méthodes d’accès à la base de données qui ont été testé de cette façon. Cela permettait de s’assurer que le code java ne modifiait pas la base de données de façon imprévue et ne créait pas d’erreurs.

## Point GIT

GIT a été très pratique tout au long du projet. Il nous a servi à mettre en commun nos fichiers lors du travail d'analyse et lors du développement il nous a permis d'avoir un suivi de l'avancement du projet et de pouvoir revenir sur le code plusieurs fois à cause de différents problèmes, de pouvoir récupérer les modifications faites par les membres de l’équipes. Il y aura eu un peu plus de 70 commit au total.

# Planning

## Planning prévisionnel

Hana

|  |  |
| --- | --- |
| Date | Travail prévu |
| Avant le 14 décembre | Analyse et rédaction du rapport d’analyse |
| Du 14 au 20 décembre | Création de la base de données  Création des classes PHP  Création des classes Java  Remplissage de la base de données |
| Pendant les vacances | Ajouter la connexion à la base de données en PHP  Ajouter la connexion à la base de données en Java  Ajouter les modèles d’accès aux données  Avancer sur les fonctions en Java |
| De la reprise au 10 janvier | Implémenter les fonctionnalités en PHP Avancer sur les traitements en Java |
| Du 10 janvier au 16 janvier | Gérer l’affichage et l’ergonomie du PHP Implémenter l’IHM Java |
| Du 16 au 21 janvier | Préparer la présentation |

## Répartition des tâches prévisionnelles

|  |  |
| --- | --- |
| Développeur(euse) | Tâches prévues |
| Hana | * Gestion de la base de données et adaptation des anciens projets pour avoir une structure MVC en PHP * Ajout des classes métier, modèles DAO / connexion * Ajout des contrôleurs et des vues en PHP * Gestion de l'aspect graphique du site web * Ajout des fenêtres et composants de l'IHM en PHP |

## Planning réel des tâches

### PHP

|  |  |
| --- | --- |
| Date | Travail Réalisé |
| Pendant les vacances | * Ajout des données dans la base de données * Réalisation de la connexion avec DAO en PHP |
| Séance du 08/01/2020 | * Ajout des classes en PHP * Début de l'ajout des modèles d'accès aux données en PHP * Début de l'interface graphique en PHP * Création contrôleur Connexion, Accueil en PHP * Création vues Connexion et Accueil en PHP * Modification et création des vues Accueil (table VIP), VIP (liste des caractéristiques, boutons Modifier et Mupprimer), AjouterVIP (inputs et bouton Ajouter), ModifierVIP (inputs et bouton modifier), Header (menu) et Footer en PHP * Modification et création des contrôleurs Accueil et VIP en PHP * Création et modification contrôleurs Header, Footer et ModifierVIP en PHP * Modification vue ModifierVIP en PHP |
| Séance du 15/01/2020 | * Correctifs sur les fonctions du model VIP en PHP * Modification du modèle VIPDAO en PHP * Modification contrôleurs AjouterVIP, SupprimerVIP et Header en PHP   Modification de Bootstrap en PHP Ajout de la fonctionnalité pour modifier/ajouter la photo d’un VIP en PHP |
| Séance du 22/01/2020 | PHP terminé :  Corrections des modèles VIPDAO et ImageDAO  Affichage / Ajout / Modification / Suppression VIP |

# Bilan personnel : Hana

De tous les projets que nous avons réalisé à l’IUT, celui-ci est de loin celui qui aura été le plus enrichissant. Depuis le S1 nous avons souvent travaillé en groupe et avons réalisé divers projets en Web, Java, IHM, Java/mySQL, UML et PHP mais nous n’avons jamais vraiment eu l’occasion de combiner ces technologies.

Lors du développement j’ai pu d’un côté renforcer mes compétences mais aussi comprendre beaucoup mieux pourquoi nous avons vu ces technologies, notamment la SQL. En effet, lors des TP nous avions juste des questions sur des tables données dans l’énoncé mais là nous avions un cas réel. En TP je ne comprenais pas pourquoi nous devions savoir utiliser telle ou telle commande et quel était l’intérêt de nous poser des questions complexes sur des conditions avec des jointures de tables... À présent j’ai compris que cela avait une réelle utilité.

Nous avons réparti les tâches, je savais que je voulais plus m’impliquer en web/php plutôt qu’en java. J’ai donc été affecté à réalisation l’implémentation de l’application web, Marion et Bastien se sont donc chargés de l’application java.

On a eu un peu de mal à se lancer au début, le sujet n’était pas très clair, mais on a su faire le point et rapidement après quelques explications.

Après avoir fait un affichage brouillon des fonctions principales en pour l’application web (les boutons, le tableau de VIP, etc), j’ai voulu gérer le CSS, mais ce n’était pas une bonne idée. Je n’étais pas familière avec Bootstrap et c’est pourtant le Framework qui était implémenté, j’ai dû me renseigner sur comment bien l’utiliser et j’ai perdu du temps au début en voulant aller trop vite. Je me suis résignée à travailler le CSS et j’ai implémenter les fonctions pour ajouter, modifier et supprimer un VIP. Après m’être assurée qu’elles fonctionnaient, j’ai repris là où je m’étais arrêté en CSS pour avoir un rendu qui me convenait mieux. Je ne suis pas 100% satisfaite du résultat, mais je suis contente d’avoir compris la manière d’utiliser Bootstrap et ça m’a vraiment motivé à continuer ce projet jusqu’au dernier moment. Je me suis ensuite mise au défi de permettre d’ajouter une photo sur les VIP, chose que j’ai réussie et dont je suis très satisfaite.

Finalement, en PHP, je considère que j’ai réussi à implémenter tout ce qui était demandé, l’application reste encore à développer mais je suis satisfaite du résultat. Faire ce projet de CPOA m’a beaucoup apporté : il m’a réconforté quant à mes capacités et m’a permis de découvrir le PHP sous un autre angle, beaucoup plus ludique que ce à quoi je m’attendais. J’avais été déçue de la plupart des matières informatique à l’IUT, mais travailler un projet en groupe tel que celui-ci m’a permis de regagner cette curiosité que j’avais à l’arrivée à l’IUT.

# Bilan personnel : Bastien

Ce projet est le plus conséquent que nous ayons eu à faire à l’IUT, il combinait différentes technologies tel que le Java, le PHP, le SQL…

La phase d’analyse ne m’a pas apporté grand-chose, car sortant d’un BAC S-SI, j’avais déjà eu l’occasion de faire des analyses sur des projets plus conséquents.

Concernant le développement, j’ai pu m’améliorer en java, principalement sur la partie accès à la base de données. Cette partie m’a aussi permis de comprendre que j’aimais bien le développement mais que je ne peux pas faire ça durant toute une journée, ce qui me conforte dans mon idée de réorientation après mon DUT.

# Bilan personnel : Marion

Pendant ce semestre, la matière CPOA était la seule matière avec un projet que nous avons eu. C’est pour moi, le seul moment où nous avons pu faire de l’informatique. J’aime beaucoup le Java c’est mon langage préféré mais je ne suis pas convaincu du sujet qui n’est pour moi pas assez détaillé. Nous avions tout le temps des questions sur certains points qui n’étaient pas abordés. Néanmoins ce travail de groupe était intéressant et nous a permis de coder et d’utiliser nos connaissances.