CPOA

Rapport de programmation

Une image contenant tennis, sport, sport athlétique, orange

Description générée automatiquement

Juan Gomez Sanchez

Anne-Sophie Beal

Lena Iglesis

G1S3

**Table des matières :**

1. **ÉLÉments techniques**
2. **Etat d’avancement**
3. **Description de l’architecture**
4. **Retour À l’analyse**
5. **SI C’ÉTAIT À REFAIRE**
6. **ÉLÉMENTS DE GESTION DE PROJET**
7. **PLANNING**
8. **PARTAGE DES TÂCHES**
9. **UTILISATION DE GIT**
10. **BILAN PERSONNELS**
11. ELEMENTS TECHNIQUES
12. État d’avancement

* Billetterie

Notre billetterie est fonctionnelle. On peut accéder à la billetterie grand public, la billetterie licenciés (accessible via un numéro de licence) et la billetterie promotion (accessible via le code promo de l’association).

Un administrateur peut également se connecter sur le site sur la page « Connexion ».

Nous avons également ajouté une page d’accueil, page « Joueurs » et page « Planning » sur notre site.

Sur la billetterie, nous pouvons choisir notre billet en fonction de la catégorie, le prix est affiché pour chaque billet, par date et catégorie.

* Planning des matchs

Nous avons réalisé l’IHM.

Il y a d’abord une page pour permettre à l’administrateur de se connecter. Ensuite l’utilisateur est envoyé sur un menu qui permet d’accéder à la page du planning des matchs du 1er tour.

Cette dernière interface affiche les 8 matchs ainsi que les joueurs, arbitres et ramasseurs concernés.

Cette partie est implémentée manuellement. Nous récupérons la liste des joueurs et les disposons aléatoirement dans les matchs. Ensuite nous ajoutons les arbitres et équipes de ramasseurs de balle adaptés en fonction des différentes contraintes (nombre de matchs d’un arbitre, nationalité de l’arbitre de chaise, etc.).

1. Description de l’architecture

* Billetterie

Couche Présentation :

Nous avons réalisé plusieurs pages. Une page d’accueil, 3 pages de billetterie (billetterie grand public, licenciés, promotion), une page avec la liste des joueurs, une page de connexion pour l’administrateur et une page de planning.

Toutes ces pages sont reliées entre elles.

Couche Métier :

Nous avons plusieurs classes :

* Association, avec le nom de l’association et son code promo
* Billet, qui contient toutes les infos d’un billet (numéro, type de place, date…)
* Client, qui contient les noms et prénoms du client
* Licencié, pour les numéros de licences

Cependant nous n’avons pas utilisé la classe Client puisque nous ne stockons pas les informations sur les acheteurs.

La classe Billet est également inutilisée car nous avons manqué de temps.

Couche Données :

Nous avons choisi de réaliser ces 4 classes. La classe association nous permet, lorsqu’un utilisateur souhaite accéder à la billetterie promotion, de vérifier le code promo entré par l’utilisateur.

La classe Licencié a la même utilité mais pour la billetterie des licenciés. Elle nous permet de vérifier si le numéro de licence entré par l’utilisateur existe.

Les classes clients et billets ne nous ont finalement pas servi par manque de temps mais également car nous avons jugé finalement que ce n’était pas très utile et compliquait les choses.

Les quelques problèmes rencontrés avec les tables étaient que nous avions au départ prévu des tables finalement inutiles, ou, dans certaines tables des attributs inutiles. Par exemple, nous avions dans la classe Licencié le nom et prénom du licencié.

* Planning des matchs

Couche Présentation :

L’application est composée de 3 interfaces :

* Connexion
* Menu
* Planning

L’interface Connexion permet à un administrateur de se connecter à l’application à l’aide d’un nom d’utilisateur et d’un mot de passe unique.

Une fois l’administrateur connecté, il est dirigé sur l’interface Menu. Celle-ci est composée d’un bouton qui renvoie ensuite sur le planning des matchs. Nous devions ajouter plusieurs boutons au départ mais cela n’a pas été fait par manque de temps.

L’interface Planning permet l’affichage du planning des matchs du 1er tour avec ses 8 matchs. Nous voyons pour chaque match les 2 joueurs concernés.

Couche Métier :

Nous avons plusieurs classes, les trois classes les plus importantes sont : Joueurs, arbitres, ramasseurs. Chacune d’elle contient des getter et setter.

Ces 3 classes permettent notamment de récupérer la liste des joueurs, arbitres et ramasseurs du tournoi pour les inclure dans le planning.

Il y a également une classe “EquipeRamasseurs” qui permet de faire des équipes de ramasseurs. Nous avons aussi les classes Réservation et Court mais nous ne les avons finalement pas utilisées puisque nous avons choisi de générer manuellement les matchs par manque de temps.

Couche Données :

Via la base de données, nous récupérons la liste des joueurs, des arbitres et des ramasseurs pour ensuite les placer dans le planning.

Nous avons dû réaliser quelques changements au cours de notre projet, notamment rajouter la nationalité des joueurs, la catégorie des arbitres. Certaines tables ont de même été supprimées, par exemple la table Planning que nous avons finalement jugé inutile.

La connexion à la base de données a également été compliquée au début, nous avions des problèmes pour nous connecter mais cela a été réglé.

1. Retour à l’analyse

La partie conception et analyse du projet nous a permis d’établir les diagrammes. Ceux-ci nous ont été utiles au début du développement pour ne pas oublier ce que nous devions faire.

Les diagrammes de classes nous ont énormément servis pour mettre en place la base de données, les tables et leurs attributs.

Pendant l’analyse, nous avions cependant prévus beaucoup trop de classes par rapport à ce qui était réellement nécessaire. En effet nous avons eu à supprimer, par la suite, quelques tables de la base de données qui étaient inutiles.

Nous avons effectué énormément de changements dans la table de données et la couche Métier, notamment au niveau des sous-classes, comme la classe “Personne”, reliée avec Joueur, ramasseur, arbitre. Nous nous sommes rendu compte que cela prendrait trop de temps et nous compliquerait énormément plus la tâche. À la place nous avons créé des variables qui déterminent la catégorie, au lieu de créer des tables à part.

Au niveau des diagrammes, nous nous sommes très peu appuyés sur les diagrammes de cas d’utilisation et diagrammes d’activités.

Les maquettes de la partie web ont été très utile pour réaliser la partie HTML/CSS de la billetterie puisque nous pouvions nous appuyer sur ce modèle, il y a eu très peu de changements.

1. Si c’était à refaire

Si les deux applications avaient été à refaire nous aurions ajouté plus de fonctionnalités en nous organisant mieux vu la charge de travail à fournir.

Ce qu’on aurait refait pour l’application WEB :

Il aurait fallu rendre tout le site fonctionnel pour le module Billetterie.

Ce qu’on aurait ajouté aux différentes pages du site :

-> Front

* Modification de la page d’Accueil :

Diaporama des actualités : les actualités récentes du site s’afficheraient en diaporama c’est à dire que l’actualité changerait au bout d’un certain temps et l’utilisateur pourrait choisir de les consulter rapidement grâce à des flèches directionnelles de gauche (pour voir l’actualité précédente) à droite (pour voir l’actualité suivante).

Également pour le fond de la page d’accueil il aurait fallu que l’effet d’une page tricolore soient mieux appliqué car les div ne séparent pas la page en trois couleurs différentes, les couleurs se découpent seulement en rectangles dans la page. Ce qui n’est pas le résultat escompté.

* Conception d’une page Panier :

Il aurait fallu permettre à l’utilisateur de pouvoir acheter plusieurs billets différents, valider son panier et être renvoyé à la page de redirection d’achat. Pour ce faire il aurait fallu utiliser du JSON ce qui aurait ajouté à la difficulté car le JSON est un langage que l’on ne connaît pas. Mais grâce à JSON et à PHP nous aurions pu alors faire un panier dynamique qui à la sélection d'un billet ajouterait celui-ci au panier avec la possibilité de supprimer un ou des billets ajoutés et tout ça en calculant le montant total du panier (avec ou sans code promotionnel).

-> Back

Nous aurions aimé rendre l’entièreté de notre site fonctionnel. Nous avons décidé de faire un menu à 4 onglets (il y en aurait eu plus si nous avions dû développer les modules gestion des VIP, gestion des hébergements, le système de Billetterie et la définition du planning des matchs) : Actualités, Billetterie, Boutique, Se connecter.

Nous avons choisi ces onglets en nous inspirant des sites de tennis existant*s*. Pour le module gestion Billetterie il nous semblait alors des plus pertinents à afficher, parmi ceux que nous avions vu, sur notre propre site tout en menant les onglets Actualités et Boutique à de réelles pages actualités et boutique.

De plus, au niveau du back-end de l’application nous aurions aussi aimé ajouter plusieurs associations. Notamment, une association entre les tournois et les billets et entre les joueurs et les tournois.

Association tournoi – billets :

L’utilisateur en sélectionnant un certain tournoi aurait pu acheter les billets disponibles pour le tournoi en question.

Association joueurs – tournois :

L’utilisateur en sélectionnant, un certain joueur, aurait pu voir à quel tournoi ce joueur participe et ainsi choisir le billet correspondant à un match en fonction d’un joueur.

Ensuite, nous aurions également pu mettre en place un système qui permettrait à l’utilisateur de choisir certains joueurs comme ses “joueurs favoris”. Ce qui nous aurait laissé la possibilité d’ajouter un système de notification au site web qui notifierait l’utilisateur de promotions sur la billetterie ou lorsqu’un de ses joueurs favoris participe à un tournoi.

Par la suite, lors de l’inscription, nous aurions pu laisser l’utilisateur renseigner son âge pour pouvoir automatiquement bénéficier de réductions s'il a moins de 12 ans.

Pour l’application Java :

Il aurait été plus souhaitable d’avoir une meilleure organisation, afin de ne pas se perdre dans tout le travail donné. Au lieu de basculer entre l’interface, la connexion à la base de données et les classes nécessaires au fonctionnement de l’application, cela aurait été mieux de commencer par les classes, puis la base de données et pour finir l’interface connectée à nos données. À cause de cette perte de temps il y a eu un retard dans le projet java.

Nous avons rencontré énormément de problèmes de problèmes pour la connexion entre l’application et la base de données.

Nous n’avons pas eu le temps de répondre à toutes les demandes du cahier de charges.

Voici ce que nous aurions pu faire faire si on avait eu plus de temps.

* Interface de connexion :

Pour l’interface connexion nous aurions pu rajouter un autre utilisateur qui n’aurait accès qu’à la lecture du planning match, et aussi Joueur afin qu’il puisse accéder à la partie cours. Ainsi ils pourront choisir la date

* Interface des menus :

Une fois connecté en tant qu’administrateur, nous aurions pu rajouter directement des fonctionnalités tel que “créer un nouveau tournoi”, “modifier le tournoi”, “changer les scores”.

* Interface du planning :

Nous aurions pu faire en sorte que le planning soit généré automatiquement. Aussi, nous aurions pu faire en sorte que l’interface soit plus ergonomique.

Nous aurions pu aussi implémenter la fonctionnalité permettant aux joueurs de réserver des créneaux pour leurs entraînements.

1. ELEMENTS DE GESTION DE PROJET
2. Planning

Voici le premier planning prévisionnel que nous avions réalisé :



Finalement notre planning a plutôt été celui-ci, il n’y a pas eu de gros changements à part la durée de quelques tâches et le fait que nous avons finalement terminé la gestion des plannings à plusieurs :



1. Partage des taches

Au sein du groupe nous avons décidé de tous travailler sur la partie conception :

* Anne-Sophie a réalisé les maquettes des deux modules
* Juan et Léna ont travaillé sur la conception des diagrammes

Par la suite, Anne-Sophie et Léna ont travaillé sur le module Web (Billetterie) et Juan sur la partie Java (Gestion des planning).

Sur la partie billetterie Anne-Sophie s’est surtout occupée du front et Léna du back end.

1. Utilisation de Git

Nous avons utilisé Git tout au long de la conception et du développement du projet. Cela nous a été très utile notamment lorsqu’on travaillait en distanciel, pour le partage des données.

Nous avons décidé de stocker le projet ainsi que les diagrammes et rapport sur le dépôt Git afin d’avoir accès à tous les documents et fichiers à chaque instant.

Nous avons fait sur le repository, 2 branches :

* Une branche « main », utilisée pour la billetterie
* Une branche « planning-match », utilisée pour la gestion du planning

1. Bilans personnels

* Anne-Sophie :

Le module et le projet m’ont apporté énormément en termes de communication, de travail d’équipe et de connaissances personnelles. Le volume horaire du cours étant relativement court pour le projet que nous avions à créer, le module était assez stressant, nous avons dû beaucoup communiquer et nous organiser pour parvenir à faire et à gérer le maximum de fonctionnalités pour nos applications WEB et Java sur ce petit laps de temps. En revanche la création de l’application WEB m’a permis d’apprendre de nombreuses fonctionnalités HTML et CSS.

J’ai malgré tout rencontré de nombreuses difficultés sur la fabrication du site WEB. Notamment les majeures difficultés suivantes :

* Imbrication des <div> et la gestion de ces dernières dans le CSS :

Je me perdais parfois dans mes <div> car j’effectuais une mauvaise indentation dans mes fichiers qui ne me permettait pas de pouvoir me repérer facilement dans mon code. J’ai privilégié alors par la même occasion de créer plus de classes “container” pour avoir un rendu propre du site et effectuer un responsive plus facile.

* Les balises < a > et les effets des hover associés :

Par pur apprentissage de ma part et pour un rendu un peu plus beau de l’application WEB je voulais des jeux d’opacités sur le site. Or, il était difficile de placer correctement la fonction “opacity” dans le CSS. Et ça d’autant plus dans la page billetterie où lorsque l’utilisateur veut cliquer sur un billet acheter est afficher à la place de la date.

* Effectuer le responsive :

Ce n’est pas réellement quelque chose de difficile mais je dirais que le responsive est très pénible à faire.

Mais ce qui a été le plus difficile, pour moi, était l’ergonomie du site. Car c’est quelque chose que je n’arrive pas personnellement à optimiser. J’ai donc consulté de nombreux site internet de tennis pour m’inspirer de leurs architectures lorsque je voyais que celles-ci me plaisaient bien et qui était donc la plus “ergonome” pour moi.

* Léna :

Ce travail m’a apporté de nouvelles connaissances. En effet j’ai pu utiliser GIT tout au long du projet, et gérer des conflits.

J’ai appris de nouvelles choses sur l’analyse d’un projet, en réalisant les plannings prévisionnels, les différents diagrammes nécessaires etc. J’ai également pu voir que ce qui était prévu n’est pas toujours ce qui est réalisé, parfois par manque de temps ou de connaissances ou à cause de difficultés rencontrées.

De plus j’ai pu comprendre de nouvelles fonctionnalités en PHP et Java puisque j’ai travaillé sur ces deux langages. En PHP j’ai pu réaliser les différentes classes manager pour accéder à la base de données.

En Java, j’ai appris à connecter l’application à une base de données MySQL, cela a d’abord était une difficulté puisque nous ne savions pas comment faire.

J’ai pu implémenter toute la base de données (création des tables, ajout des données…).

Enfin, la plus grosse difficulté a surtout été de finir le projet dans les temps, car il fallait prévoir combien de temps passer sur l’analyse puis sur le développement. Cette partie « Prévision » était assez compliqué, c’est pourquoi nous avons manqué de temps lors du développement.

* Juan :

Ce projet m’a permis de découvrir de nouvelles notions et de perfections celle que j’avais déjà découvert en théorie.

Tout de même, j'ai rencontré des obstacles dans la réalisation du projet, me bloquant même quelques jours :

La connexion à la base de données.