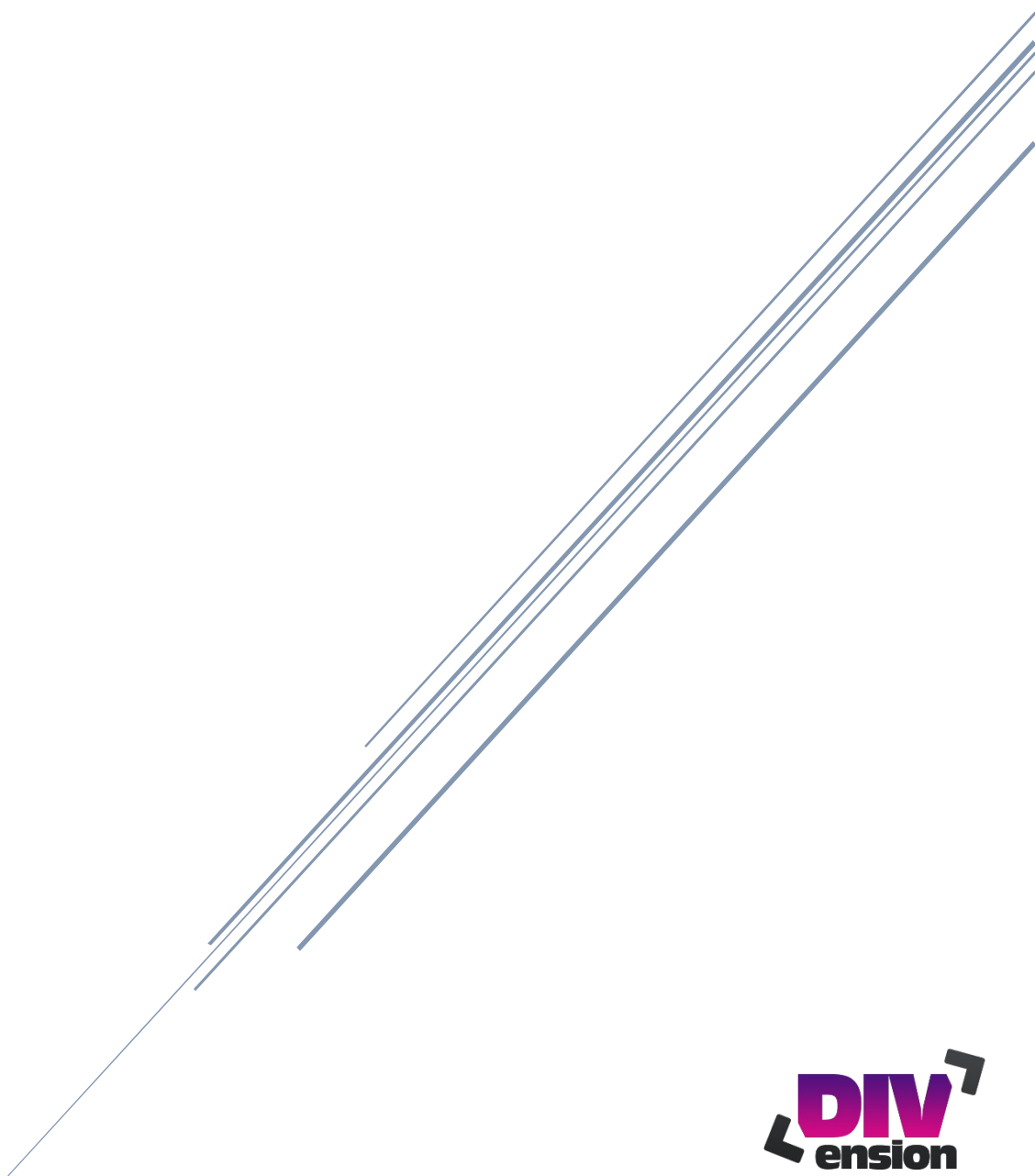


DOSSIER DE PROGRAMMATION

Open Tennis – Gestion des VIPs



Bartholomé GILI
Ethan BOURCEREAU

Table des matières

Éléments techniques	2
1. État d'avancement	3
1.1 Ce qui a été fait.....	3
1.2 Ce qui n'a pas été fait	3
1.3 Ce qui devrait être amélioré.....	3
2. Description de l'architecture.....	4
3. Retour à l'analyse	5
4. Si c'était à refaire.....	5
Éléments de gestion de projet	6
5. Planning et partage des tâches	7
5.1 Prévisionnel	7
5.1 Final	7
6. Système de gestion de versions	9
7. Système de gestion des versions.....	9
7.1 Bartholomé GILL.....	9
7.2 Ethan BOURCEREAU	10

Première partie

Éléments techniques

1. État d'avancement

1.1 Ce qui a été fait

- Back-end
 - Connexion à la base de données
 - Création des différentes entités et relations
 - Opérations CRUD
 - Système d'authentification sécurisé
 - Mise à disposition de routes API reflétant les opérations CRUD pour l'application client
 - Gestion des erreurs
- Front-end
 - Design du site
 - Connexion au back-end (API)
 - Listage des vips / suivis
 - Ajout des vips / suivis
 - Modification des vips / suivis
 - Suppression des vips / suivis
 - Tri et recherche dans les listes
 - Système d'authentification sécurisé avec une page de connexion
 - Accès restreint à certaines pages en fonction de si l'utilisateur est authentifié ou non

1.2 Ce qui n'a pas été fait

- Front-end
 - Gestion et affichage des erreurs
 - Tests unitaires
 - Tests d'intégration
 - Rafraîchissement automatique des jetons JWT (utilisés pour l'authentification)

1.3 Ce qui devrait être amélioré

- Back-end
 - Tests unitaires
 - Tests d'intégration

2. Description de l'architecture

Nous avons choisi une architecture client/serveur avec à la fois du Server Side Rendering (SSR) et du Client Side rendering.

Ainsi, notre application est concrètement divisée en 2 parties totalement indépendantes :

- Une application écrite en PHP avec le framework Symfony et la librairie API Platform, faisant office de serveur avec lequel on communique au travers d'une API.
- Une application écrite en Javascript avec le framework Next.js.

Voici l'architecture complète comprenant les 2 applications et leurs différentes couches :

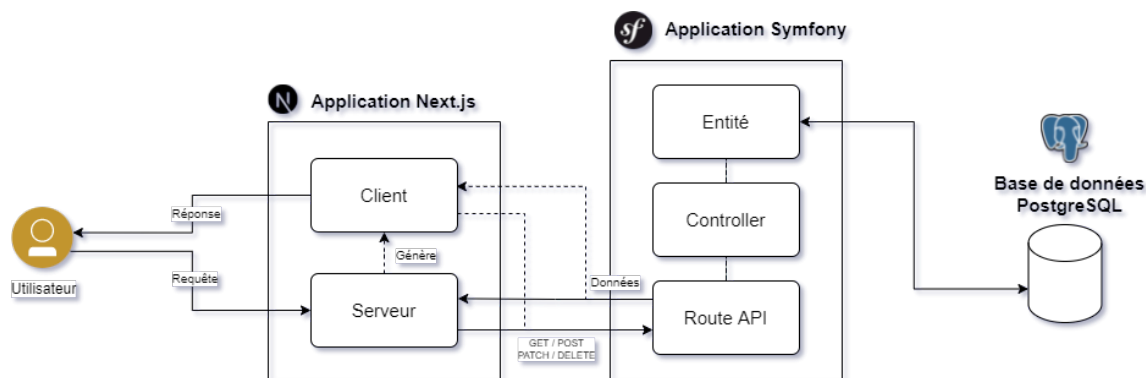


FIGURE 1 – DIAGRAMME D'ARCHITECTURE

Nous avons fait ces choix car, étant un groupe de 2, nous n'avons eu à réaliser qu'un seul module. Comme nous souhaitons tous les 2 continuer dans le développement web par la suite, cela nous a paru bien plus bénéfique de choisir le module web.

De plus, le fait de dissocier totalement le front-end du back-end nous a beaucoup plu. En effet, plus les différentes couches d'une architecture globale sont indépendantes et séparées des autres, plus la maintenance, la flexibilité, l'adaptabilité et la scalabilité s'en voient améliorées. Cela permet également de produire plusieurs applications front-end pour divers supports et environnements avec une couche métier et données centralisées (ex : Une application mobile pour Android).

Enfin, nous avons tous les 2 des bases avec le framework React ainsi que Symfony, mais voulions pousser un peu plus loin et apprendre de nouvelles technologies récentes et intéressantes. C'est ainsi que l'utilisation de Next.js nous a paru évident.

Next.js est un framework javascript qui permet de faire du rendu d'applications React côté serveur, qui n'est pas possible nativement avec ce dernier. Le développement d'une application en Next.js s'approche donc de celui d'une application en PHP, mais nous avons une application React délivrée à la fin, et que c'est fait de manière très optimisée et contrôlable. Par exemple, et c'est ce que nous avons décidé de faire, nous pouvons rendre une partie d'une page côté serveur, puis générer le reste côté client. C'est ce mode dit « hybride » qui nous a fortement intéressé.

De plus, l'aspect serveur de Next.js, que nous ne pensions pas du tout utiliser au début puisque nous avons un serveur séparé écrit en Symfony, s'est finalement avéré très utile pour le système d'authentification.

3. Retour à l'analyse

- Le diagramme de cas d'utilisation a été très utile pour se représenter rapidement toutes les différentes fonctionnalités de l'application et ne pas se perdre en route. Nous l'avons respecté à la lettre.
- Nous avons jugé utile de rajouter un diagramme non demandé dans l'énoncé initial, à savoir celui d'États-Transitions, qui, à notre sens, est primordial pour un site web puisqu'il permet de visualiser rapidement les différentes pages et connexions entre elles.
- Également, le diagramme de classe nous a permis de savoir rapidement tous les attributs de chaque entité, et donc de bien structurer l'information de manière uniforme entre le front-end et le back-end
- Nous n'avons pas bien mis à profit l'IHM, étant donné que nous avions des designs déjà en tête et que cela changeait régulièrement. Au final, c'est surtout la page principale du Dashboard, qui liste les VIPs et les suivis, qui a subi du changement par rapport à ce qui était initialement prévu.

Finalement, nous n'avons que très peu dévié de notre dossier de conception lors de l'implémentation, et avons compris à quel point cette partie est importante dans le développement d'une application.

4. Si c'était à refaire

Si nous repartions de zéro pour faire l'application :

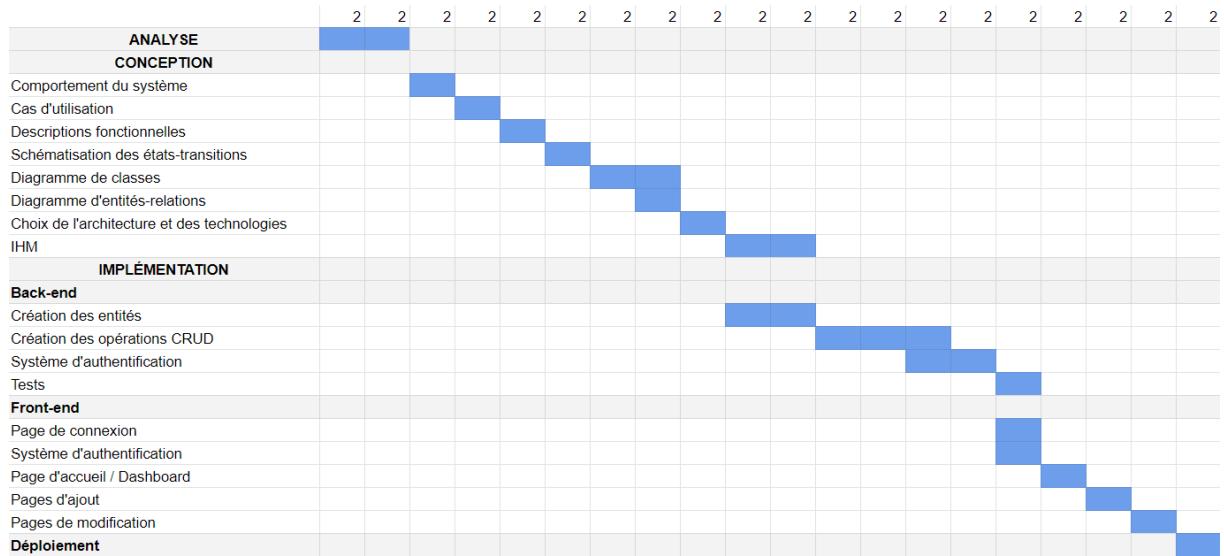
- Il faudrait que l'on réalise une IHM un peu plus précise, nous avons perdu beaucoup de temps lors de l'implémentation front-end à imaginer le design, même si ça nous a plus finalement.
- Même si nous avions établi un planning des tâches très précis, nous ne l'avons pas toujours respecté.

Deuxième partie

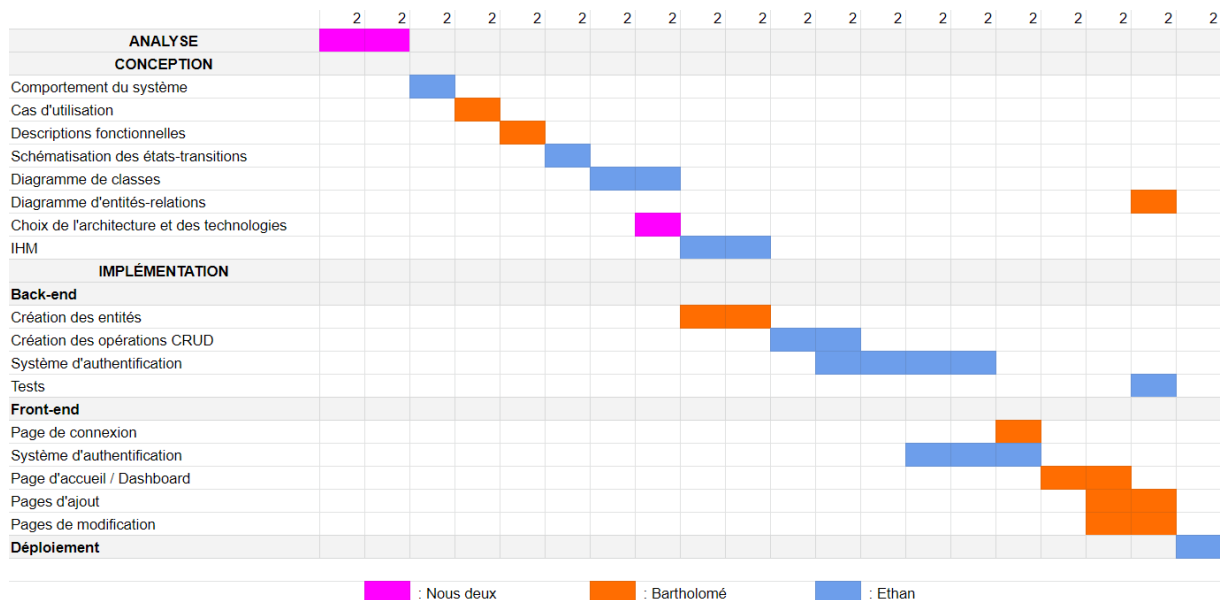
Éléments de gestion de projet

5. Planning et partage des tâches

5.1 Prévisionnel



5.1 Final



En commentaire, nous pouvons remarquer que la partie conception s'est à peu près déroulée comme prévu initialement.

En revanche, il y a eu pas mal de changements au niveau de l'implémentation.

En effet, nous avons surestimé la durée de création des opérations CRUD, qui a été grandement simplifiée par l'utilisation de la librairie API Platform qui est native à Symfony. Au contraire, pour le système d'authentification, nous avons voulu essayer la technologie JWT, ce qui s'est avéré assez complexe à mettre en place. Heureusement, et après y avoir passé un temps assez conséquent, nous avons un système d'authentification viable et très sécurisé. Cela nous a aussi appris le fonctionnement des JWT, ce qui sera très utile par la suite.

Pour gérer toutes les tâches et le planning, nous avons utilisé le logiciel Notion :

The screenshot displays a Notion project management board with a Kanban-style layout. At the top, there are four columns representing task status: 'Not started' (6 tasks), 'In progress' (2 tasks), 'Attente d'approbation' (2 tasks), and 'Completed' (8 tasks). Below these columns, tasks are organized by assignee.

- Bartholomé Gili (9 tasks):**
 - Not started:** Diagrammes de séquence + activité (Conception), Diagramme de GANTT (Conception), Diagramme base de données (Conception), + New.
 - In progress:** Descriptions fonctionnelles (Conception, 3 comments), + New.
 - Attente d'approbation:** + New.
 - Completed:** Système d'authentification (Back-end, Front-end), Endpoints API (Back-end), Diagramme de classe (Conception), Cas d'utilisation (Conception), Diagramme d'état-transition (Conception), + New.
- Ethan (9 tasks):**
 - Not started:** Dashboard / Suivis (Front-end), Modification Suivis (Front-end), Modification VIP (Front-end), + New.
 - In progress:** Page VIP (Front-end), + New.
 - Attente d'approbation:** Ajout VIP (Front-end), Ajout Suivis (Front-end), + New.
 - Completed:** Dashboard / VIPs (Front-end), IHM (Conception), Login (Front-end), + New.
- No Assign (0 tasks):**
 - Not started:** + New.
 - In progress:** + New.
 - Attente d'approbation:** + New.
 - Completed:** + New.

6. Système de gestion de versions

Pour la gestion des versions, nous avons utilisé Git de pair avec la forge de l'université Lyon 1 en tant que remote, afin que vous, professeurs, y ayez accès.

Nous avons réalisé une trentaine de commits, nommés au mieux selon une charte de nommage professionnelle.

Son utilisation, déjà bien acquise de nous 2, nous a permis de collaborer efficacement et sans conflit sur le projet. Cela nous a aussi permis de gagner en temps et de pouvoir travailler sans problème sur différentes machines sans s'encombrer d'une clé de stockage que l'on peut oublier facilement.

7. Système de gestion des versions

7.1 Bartholomé GILI

J'ai trouvé la partie conception très intéressante car je suis depuis toujours convaincu de sa grande utilité mais je n'ai jamais eu la détermination de la mettre en place sur mes projets, qu'ils soient scolaires ou personnels. Ainsi, réaliser de A à Z un dossier de conception m'a permis d'avoir une vue globale sur le projet, et m'a mis en confiance quant à ma capacité à en réaliser un.

Comme nous n'avions que le module de web à réaliser, j'ai jugé intéressant de faire une application complexe divisée en plusieurs sous-applications interconnectées. Ethan m'a tout de suite suivi dans l'idée et nous avons pu apprendre un nouveau framework, à savoir Next.js, de manière saine et ludique. Je me suis personnellement bien plus occupé du back-end, et ai particulièrement buté sur le système d'authentification. Ayant choisi une authentification par JWT (JSON Web Token), en croyant que cela serait plus simple, j'ai été très surpris de voir sa complexité d'intégration avec Next.js, qui a un statut très particulier puisqu'il s'agit à la d'un serveur et d'une application React client. J'ai testé quelques librairies à droite à gauche, mais ai finalement décidé de tout faire à la main puisque j'avais bien compris le fonctionnement et l'architecture à avoir avec des JWT. C'était aussi la première fois que je m'essayais à Symfony, contrairement à Ethan. Je lui ai donc demandé si je pouvais réaliser l'API afin de monter en compétences dessus, ce que je n'ai pas manqué de faire.

Pour conclure, ce projet a été assez bénéfique malgré son aspect très chronophage dans une période déjà bien chargée en projets dans les autres modules de l'IUT. Il m'a apporté beaucoup de compétences, principalement dans la conception d'une application et la sécurisation d'une API.

7.2 Ethan BOURCEREAU

Ce module m'a permis d'approfondir mes connaissances en développement web. En effet, avant celui-ci je n'avais jamais développé avec React ou bien même Next.js, étant un fanatique des nouvelles technos il me tardait de les apprendre, ce que m'a permis ce projet. D'autre part, cela m'a permis d'approfondir mes compétences de travail en équipe en matière d'organisation, de brainstorming, de création, etc... Le peu de difficultés rencontrées était sur l'ergonomie du projet, je ne savais pas vraiment comment faire un Dashboard avant ça, néanmoins après quelques recherches et quelques heures de développement j'ai réussi à obtenir un résultat concluant qui est désormais implémenté dans l'application. Je dirai pour conclure ce bilan, que l'expérience apportée par ce module me permettra une meilleure gestion de mes projets futurs, ainsi donc qu'une plus grande flexibilité face aux deadlines.