

Projet Volscore

Pour le module 306 avec Xavier Carrel

Auteurs : Sarah Dongmo, Yago Iglesias et Omar Egal Ahmed

Table des matières

1	Analyse préliminaire	3
1.1	Introduction.....	3
1.2	Objectifs	4
2.3	Gestion de projet	5
2	Analyse / Conception	7
2.1	Domaine.....	7
2.2	Concepts	8
2.3	Analyse fonctionnelle	9
2.4	Stratégie de test	11
3	Réalisation.....	12
3.1	Points de design spécifiques	12
3.2	Déroulement.....	12
3.3	Mise en place de l'environnement de travail.....	13
3.4	Description des tests effectués	14
3.5	Bilan	15
3.5.1	Erreurs restantes	15
3.5.2	Stories.....	16
3.6	Recours à l'intelligence artificielle	16
4	Conclusions.....	17
5	Annexes.....	18
5.1	Journal de travail.....	18

1 Analyse préliminaire

1.1 Introduction

Le projet Volscore a pour ambition de faciliter et d'optimiser le travail des marqueurs de la Fédération Suisse de Volleyball lors des rencontres sportives. Volscore est une application web ergonomique et intuitive, conçue pour permettre aux marqueurs de saisir rapidement et efficacement toutes les données essentielles du match. Cela inclut notamment le suivi des points marqués, l'enregistrement des pénalités, la gestion des temps morts et d'autres événements clés. Grâce à son interface simplifiée, Volscore contribue à réduire les erreurs, à gagner du temps et à assurer une meilleure fluidité dans le déroulement des matchs.

Le projet a été réalisé dans le cadre du module 306 de l'ETML. Le temps consacré au développement de Volscore a été d'environ deux heures par semaine, réparties sur 8 séquences. Ce travail s'est déroulé sur plusieurs semaines et a chevauché les vacances de fin d'année, permettant une progression continue du projet malgré cette période.

Volscore représente ainsi une opportunité concrète de nous initier aux principes des méthodes de travail agiles, souvent utilisées dans le monde professionnel. Cette approche nous a permis de découvrir une manière structurée et collaborative de gérer un projet, en favorisant l'adaptabilité, la répartition efficace des tâches et les itérations régulières pour atteindre les objectifs fixés.

Le projet Volscore était déjà en cours de conception au moment de notre intervention, utilisant le framework PHP Symfony ainsi qu'une architecture basée sur le pattern MVC (Modèle-Vue-Contrôleur). Le livrable initial, fourni pour amorcer les améliorations, consistait en un répertoire GitHub contenant les éléments suivants :

- Une base de données SQL servant à stocker les informations nécessaires au fonctionnement du projet.
- Du code PHP permettant de gérer dynamiquement les différentes pages et interactions de l'application.
- Des fichiers HTML et CSS définissant la structure et le style visuel des pages.
- Une documentation générée via Doxygen, facilitant la compréhension et la prise en main du projet existant.

Ces ressources ont constitué une base solide pour analyser, comprendre et améliorer le projet tout en respectant l'architecture initialement mise en place.

1.2 Objectifs

Ce projet a pour but principal de faciliter et d'optimiser le travail des marqueurs lors des matchs de volleyball grâce à l'application Volscore. Pour atteindre cet objectif global, plusieurs sous-objectifs spécifiques ont été définis :

1. Amélioration de l'existant :

- Analyser et comprendre l'architecture actuelle du projet.
- Optimiser et améliorer les fonctionnalités déjà existantes, tout en respectant le framework Symfony et le pattern MVC.

2. Développement de nouvelles fonctionnalités :

- Ajouter des outils permettant une saisie plus rapide et intuitive des données pendant un match.
- Intégrer une gestion précise des points, des pénalités, des temps morts et des statistiques globales.
- Améliorer l'interface utilisateur (UI) pour la rendre plus ergonomique et accessible.

2. Fiabilité et performance

- Assurer la stabilité de l'application en réduisant les bugs et en améliorant les performances générales.
- Garantir une bonne gestion des bases de données pour le stockage et la récupération des informations.

4. Documentation et prise en main

- Rédiger une documentation technique claire et complète pour faciliter la prise en main du projet par d'autres développeurs.

Assurer une bonne compréhension des différentes fonctionnalités via des commentaires et des explications dans le code source.

En répondant à ces objectifs, le projet Volscore vise à offrir une solution robuste, performante et intuitive pour les marqueurs, tout en respectant les standards techniques du développement web moderne.

2.3 Gestion de projet

Dans le cadre de ce projet, la gestion a suivi une approche agile en s'appuyant sur la méthode Scrum. Cette méthodologie, connue pour sa souplesse et son adaptation rapide aux changements, a permis une organisation structurée tout en favorisant une collaboration active entre les membres de l'équipe. Cependant, certaines particularités liées au contexte ont influencé la mise en œuvre classique de Scrum.

2.3.1 Méthode Scrum Adaptée :

Le projet s'est articulé autour des principes fondamentaux de Scrum :

- **Backlog produit** : La liste des besoins et des fonctionnalités à développer a été répertoriée dans un backlog produit, permettant de centraliser les attentes du projet.
- **Sprints** : Le travail était organisé en cycles courts, appelés sprints, avec des objectifs clairs pour chaque période.
- **User stories** : Les besoins étaient exprimés sous forme de user stories, facilitant la compréhension et la planification des tâches.

Cependant, dans le cadre de ce projet, l'absence d'un Product Owner dédié a constitué une différence notable par rapport au cadre traditionnel de Scrum. En effet, la validation des user stories n'était pas directement assurée par un client ou un Product Owner, mais par le professeur responsable du projet et, dans certains cas, par les autres groupes travaillant sur le même projet. Cette dynamique collaborative a permis de pallier l'absence d'un interlocuteur unique tout en enrichissant la pertinence des livrables.

2.3.2 Outils Utilisés

Pour structurer et suivre le projet, plusieurs outils spécifiques ont été employés :

1. **IceScrum** : Cet outil a joué un rôle central dans la gestion globale du projet. Il a permis de :

- Gérer les backlogs (produit et sprint).
- Suivre l'état des user stories, leur priorisation et leur progression.
- Visualiser les sprints et coordonner les tâches attribuées aux membres de l'équipe.

2. **IceTools** : Pour compléter IceScrum, IceTools a été utilisé pour :

- Tenir un journal de travail détaillé, permettant de tracer les activités réalisées.
- Documenter les étapes de réalisation des user stories, offrant une transparence sur l'avancement et les ajustements effectués.

2.3.3 Particularités et Adaptations

Le contexte académique a entraîné certaines adaptations :

- **Validation collaborative** : L'implication du professeur et des autres groupes dans la validation des user stories a ajouté une dimension collaborative enrichissante, tout en permettant une rétroaction rapide et constructive.
- **Flexibilité des rôles** : Les rôles traditionnels de Scrum ont été répartis de manière adaptée aux compétences et disponibilités des étudiants.

En conclusion, la méthode Scrum adaptée, combinée à l'utilisation des outils IceScrum et IceTools, a permis une gestion efficace et structurée du projet. Les particularités du contexte, bien que différentes d'un environnement professionnel standard, ont constitué une opportunité d'apprentissage précieuse et ont favorisé une dynamique de travail collaborative et itérative.

2 Analyse / Conception

2.1 Domaine

Le domaine du projet est le volleyball. L'application est conçue pour faciliter le marquage des matchs de volleyball et dans un futur recueillir des statistiques. Elle sera utile pour les clubs de volleyball, les entraîneurs, les joueurs, les arbitres et les organisateurs de tournois.

L'application peut également intéresser les amateurs de volleyball et les spectateurs qui souhaitent suivre les statistiques. Ce projet peut être très intéressant pour la Fédération Suisse de volleyball, offrant un outil de gestion des matchs et des performances à un niveau professionnel.

Voici une liste des utilisateurs de l'application et leurs rôles :

Entraîneurs

- **Rôle** : Suivre les performances des joueurs, et analyser les statistiques pour améliorer les stratégies de jeu.

Joueurs

- **Rôle** : Consulter leurs propres performances et les statistiques de l'équipe afin de s'améliorer et d'ajuster leurs entraînements en conséquence.

Marqueur

- **Rôle** : Utiliser l'application pour marquer les points en temps réel, enregistrer les infractions, les cartons, et les temps morts.

Organisateurs de tournois

- **Rôle** : Gérer les matchs, les horaires, les résultats des tournois, et générer des rapports de statistiques.

Administrateurs du club

- **Rôle** : Superviser l'utilisation de l'application au sein du club, gérer les accès et s'assurer que toutes les données sont correctement enregistrées et disponibles.

Supporters et spectateurs

- **Rôle** : Utiliser l'application pour suivre les matchs en temps réel et consulter les statistiques.

Fédération Suisse de Volleyball

- **Rôle** : Utiliser l'application pour centraliser les données des matchs à un niveau national, suivre les performances des équipes et des joueurs, et organiser des compétitions officielles.

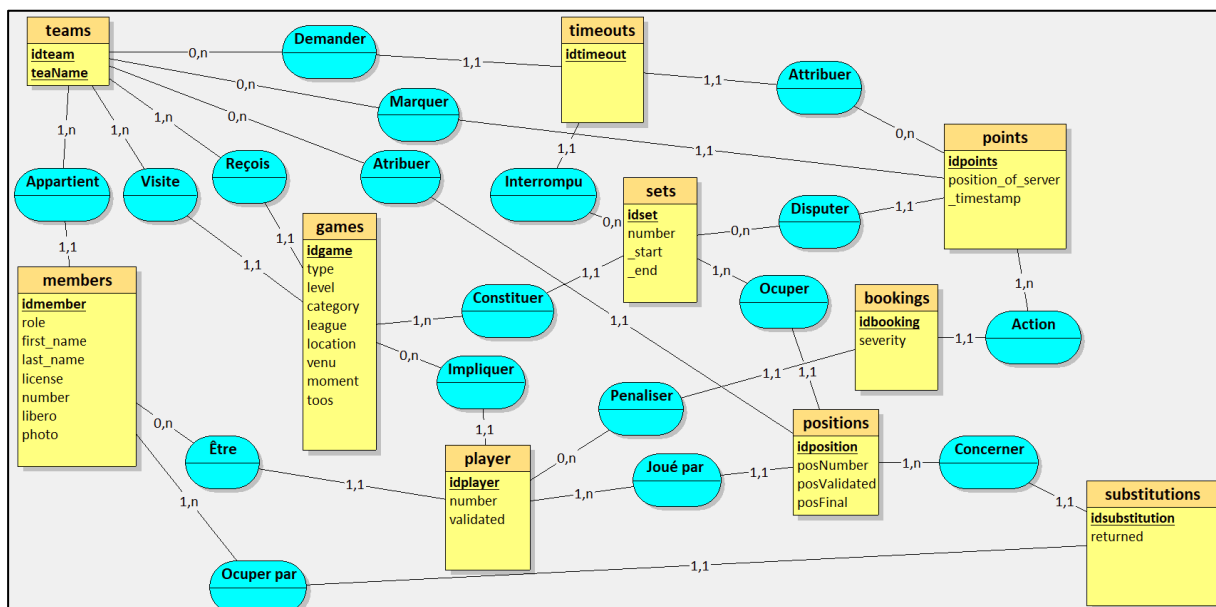
Ce projet est développé principalement pour la Suisse mais il est envisageable d'une expansion à l'internationale si l'application rencontre du succès.

2.2 Concepts

L'architecture de l'application est basée sur le modèle MVC (Modèle, Vue, Contrôleur). La partie frontend est développée en PHP afin de communiquer avec la base de données, hébergée localement via Docker, qui contient toutes les informations relatives aux matchs.

Le backend est également développé en PHP et gère la logique métier ainsi que les requêtes des utilisateurs. L'application fonctionne en environnement local, ce qui permet une gestion simplifiée et flexible de la base de données et des services associés.

Voici une vue sur le schéma MCD de la base de données :



2.3 Analyse fonctionnelle

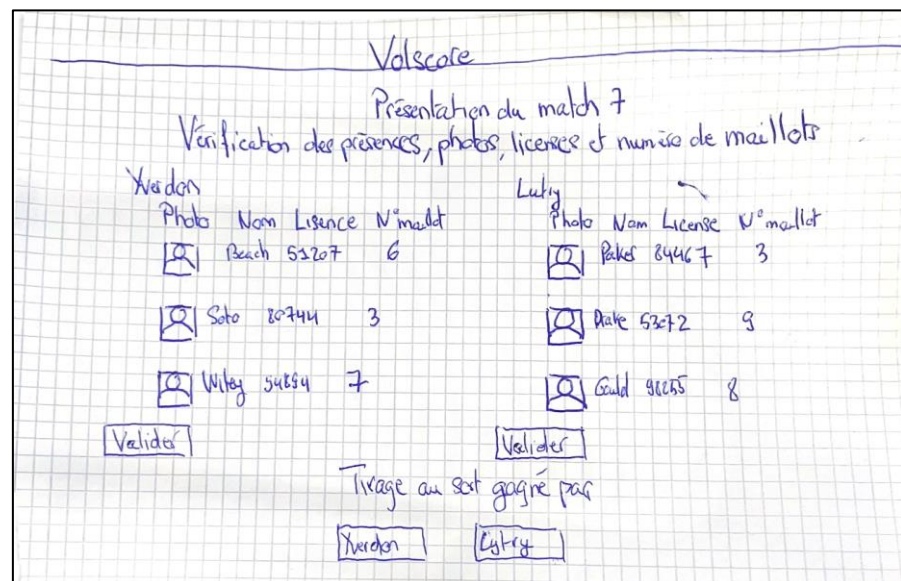
Contrôler la présence des joueurs

En tant que marqueur Je veux vérifier la photo des joueurs Pour réaliser un contrôle d'identité

Tests d'acceptance:

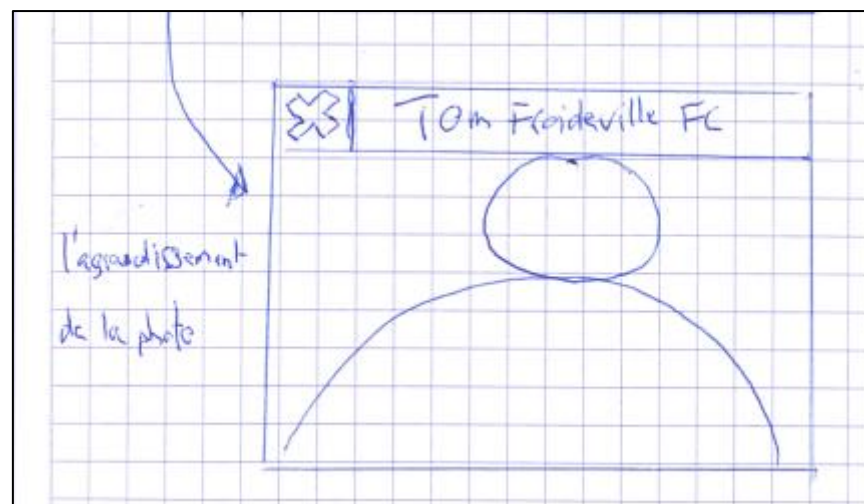
Photo du joueur

Sur la page de préparation du match, l'utilisateur voit une photo du joueur à coter de son prénom (maquette 1)



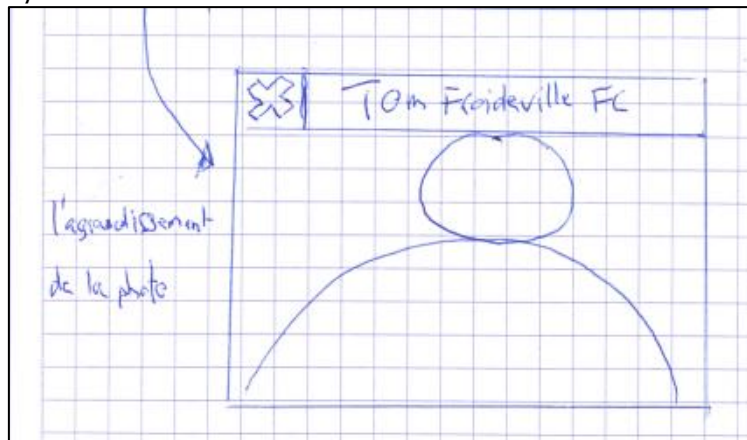
Agrandir la photo

En cliquant sur la photo du joueur elle s'agrandit. Elle s'affiche au milieu de l'écran (maquette 2)



Revenir à la liste

En cliquant sur la croix la photo en grand se ferme (Maquette 2)



Avoir la date des matchs

En tant que marqueur j'aimerais avoir la visualisation des dates des matchs Afin des mieux organiser mon marquage

créer une colonne date

Tests d'acceptance:

Après la colonne Numéro une colonne Date et heure est affichée (maquette 1)

VolScore

Matches

Date et heure	Numéro	Recevant	Visiteur	Score	Action
20 jan 2023 à 20:00	1	Froideville	Eculens	3-2	consulter
20 jan 2023 à 20:45	2	Yverden	Lutry		marquer modifier

2.4 Stratégie de test

Comme stratégie de test à réaliser nous avons décidé de cloner le répertoire depuis GitHub afin de mettre en place l'environnement dans des postes différents à ceux de développement. Après le clonage, l'application serait lancer afin de vérifier que tout démarre correctement.

La sprint review est réalisé dans un environnement neutre. Il n'est pas effectué sur le poste d'un des développeurs de l'équipe afin d'assurer que le projet est compatible dans un environnement différent.

Voici les types de test que nous aimerons réaliser.

- Test de compilation
- Test d'exécution
- Test de compatibilité
- Test de fonctionnalités
- Test de performance

Ces tests sont tous exhaustifs car le bon déroulement de ceci sont tous évidents à constater et à répertorier.

Il n'y a pas de données de test à prévoir pour nos types de tests.

3 Réalisation

3.1 Points de design spécifiques

3.1.1. Page « prepareGame.php »

Pour notre site, nous avons ajouté une colonne image dans la base de données, dans la table players. Cette colonne sert à stocker le chemin des images des joueurs. Celle-ci sont contenues dans un dossier nommé « images » et elles sont appelées via la base de données avec une variable qui est « \$player->photo ». Nous ouvrons l'image en cliquant dessus grâce du CSS et JS. Grâce à ces fonctions, les images s'ouvrent sur la page active en mode pop-up.

```
<?php foreach ($receivingRoster as $player) : ?>

<!-- Ajouter les photos -->
<tr>
    <td><a href="javascript:void(0);" class="open-popup" data-image="../images/<?=$player->photo?>">
        </a>
    </td>
    <td><?=$player->last_name ?></td>
    <td><?=$player->license ?></td>
    <td><?=$player->number ?></td>
</tr>

<?php endforeach; ?>
```

La pop-up permet d'agrandir l'image et de la fermer avec un bouton.

```
<!-- Pop-up pour afficher l'image agrandie -->
<div id="popup" class="popup">
    <span class="close-btn" id="close-btn">&times;</span>
    <img id="popup-image" class="popup-image" src="" alt="Image agrandie">
</div>
```

N.B. : Cette page étant le seul point de design ayant été appliqué, nous n'avons pas d'autres points à aborder.





















3.2 Déroulement

Dans l'ensemble, nous avons rencontré des difficultés pour comprendre le code et son fonctionnement. Lors de la phase de développement, nous nous sommes donc finalement contentés de l'essentiel malgré quelques doutes sur la manière de modifier le code déjà existant.

Pour la présentation agrandie de l'image des joueurs, nous avons envisagé différentes manières de voir en détail l'image de chaque joueur. D'abord en survolant l'image en question, puis nous avons intégré la possibilité d'un clic.

3.3 Mise en place de l'environnement de travail

- Pour bénéficier d'un serveur en local, nous utilisons Docker, qui met également à notre service une base de données.

<input type="checkbox"/>	<div> c105 2 containers</div>	-	Running (2/2)	-	 Open					
<input type="checkbox"/>	<div> pma a5c17ee6b96b </div>	phpmyadmin	Running	8081	7 hours ago					
<input type="checkbox"/>	<div> db 5af56454793f </div>	mysql	Running	6033	7 hours ago					

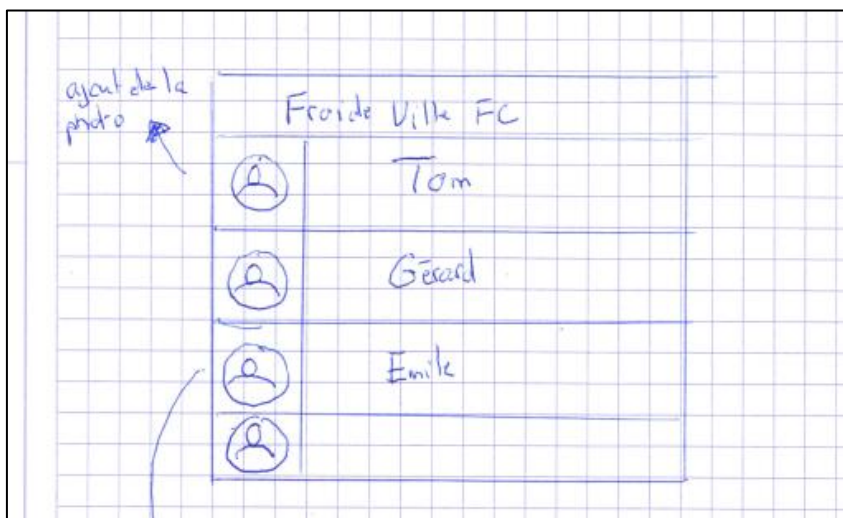
- Nous utilisons Visual Studio Code.
- Nous ouvrons notre répertoire « Volscore » et accédons au sous-dossier :
➤ ...\\PhP\\lamp\\www
- Depuis cet emplacement nous vérifions qu'il y ait bien un fichier ".credentials.php" dans le dossier "model " comprenant la configuration ci-dessous :

```
<?php
$hostname = "localhost:6033";
// $portNumber = "6033";
$databse = "volscore";
$username = "root";
$password = "root";
?>
```

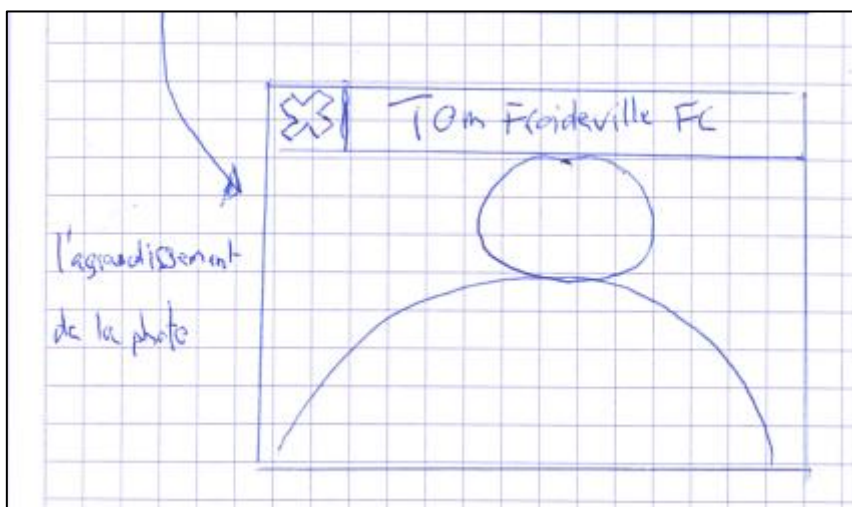
- Nous continuons en tapant la commande : « php -S localhost:8080 » depuis "...\\PhP\\lamp\\www", sur le terminal de VSCode.
Grâce à cette commande, nous accédons au lien « localhost :8080 », depuis le navigateur de notre choix, qui correspond à la page d'accueil de notre site.
- Avant de pouvoir utiliser le site, nous faisons une mise à jour en rajoutant dans l'URL : « / ?action=unittests ».
Après quelques secondes, toutes les pages du site sont maintenant à notre disposition pour observer le résultat de notre travail à mesure que nous opérons des changements dans notre code.

3.4 Description des tests effectués

1. Sur la page de préparation du match, l'utilisateur voit une photo du joueur à coter de son prénom (maquette 1)



2. En cliquant sur la photo du joueur elle s'agrandit. Elle s'affiche au milieu de l'écran (maquette 2)



3. Sur la page des matchs, en plus des infos sur le numéro de match, les équipes en duel, les scores et le type d'action actuellement possible sur ces données de matchs, l'utilisateur peut maintenant voir en premier la date et l'heure du match.

VolScore

Matches

Date et heure	Numéro	Recevant	Visiteur	Score	Action
20 jan 2023 à 20:00	1	Froidville	Eublens	3-1	<input type="button" value="consulter"/>
20 jan 2023 à 20:45	2	Yverden	Lutry		<input type="button" value="marquer"/> <input type="button" value="modifier"/>

3.5 Bilan

3.5.1 Erreurs restantes

Il n'y en a pas, nous avons réussi à faire toutes les tâches que nous avons jugées pertinentes et faisables dans le temps imparti.

3.5.2 Stories

Ce qu'on pensait faire :



Ce qu'on a fait :



Nous avons dû abandonner l'idée d'améliorer l'interface utilisateur après un premier essai, car cette user story s'avérait chronophage ; elle nécessitait une quantité de travail disproportionnel au temps que nous avions à disposition.

3.6 Recours à l'intelligence artificielle

L'IA a été uniquement utilisé pour ajuster la mise en forme de la page de contrôle des présences des joueurs. Son utilisation s'est avérée utile et nous a fait gagner du temps. Pour la deuxième user story, l'utilisation de l'IA ne s'est pas avérée nécessaire, car cette tâche requérait des connaissances déjà acquises en plus d'une compréhension du code.

4 Conclusions

Comme ce rapport le démontre, le travail effectué a été assez rudimentaire. La recherche d'idées pour répondre à d'éventuels besoins utilisateur s'est avérée laborieuse. Nous nous sommes donc contentés du peu d'idées réalistes qui s'offraient à nous. Il a également fallu du temps pour la cohésion de groupe.

Si on devait donner le projet à une autre équipe, il faudrait qu'elle ne se focalise pas sur des détails du code, qu'elle prenne son temps pour comprendre le code et surtout le fonctionnement du site dans son ensemble. Une équipe plus grande serait avantageuse, 5 à 6 personnes ; et qu'elle soit axée sur la communication pour éviter de perdre du temps. Un délai plus long, 1 mois en plus, et plus d'heure de travail dans la semaine aiderait à une profonde amélioration du site. L'application pourrait dans le meilleur des cas se développer à plus grande échelle en intéressant les marchés internationaux.

5 Annexes

5.1 Journal de travail

10/01/2025 12:11

Journal de travail voScore

					effectuée_par	Omar	Sarah	Yago	Durée [min]
story	terminée_le	tâche	temps_estimé	remarque					
Avoir la date des matchs	17 déc. 2024	Formatage et affichage de la date	5	Problème pour formater les dates de façon dynamique				10	0h10
Avoir un meilleure l'UI	03 déc. 2024	Mettre de le footer en bas de page	15			20			0h20
		Rendre symétrique les liens Matches et Equipes	20			15			0h15
Contrôler la présence des joueurs	03 déc. 2024	Inserer les photos dans la base de donnée	3	j'ai aider mon camarade Omar à lancer le projet , ensuite j'ai essayer de comprendre comment faire pour avoir les images dans la base de donnée.				60	1h00
	17 déc. 2024	Afficher les photos de l'autre équipe	1					1	0h01
		Création de style pour agrandir les images et style pour l'overlay	25	problème d'intégration du code fourni par chat gpt				80	1h20
		En-tête de tableau pour chaque colonne	5	Mauvaise balise utilise au début				10	0h10
		Insertion des images dans la base de données	60	Comprendre le code source, problème de chemin introduit dans la db				180	3h00
		Reconstitution de la maquette	15			20			0h20
		Redimensionner les images	5			2			0h02
Urgente	17 déc. 2024	Amélioration de la maquette du champs "date et heure"	15			15			0h15
		Réalisation de l'introduction de la documentation				60			1h00
	07 janv. 2025	Creation MCD	60					120	2h00
		Réalisation du point 2 de l'analyse préliminaire: Objectifs				30			0h30
		Régénérer les user stories avec IceTools	5					5	0h05
	?	Documentation (partie 3)					30		0h30
Total						2h42	0h30	7h46	10h58