**Gestion d’équipe de Football**

CFPT

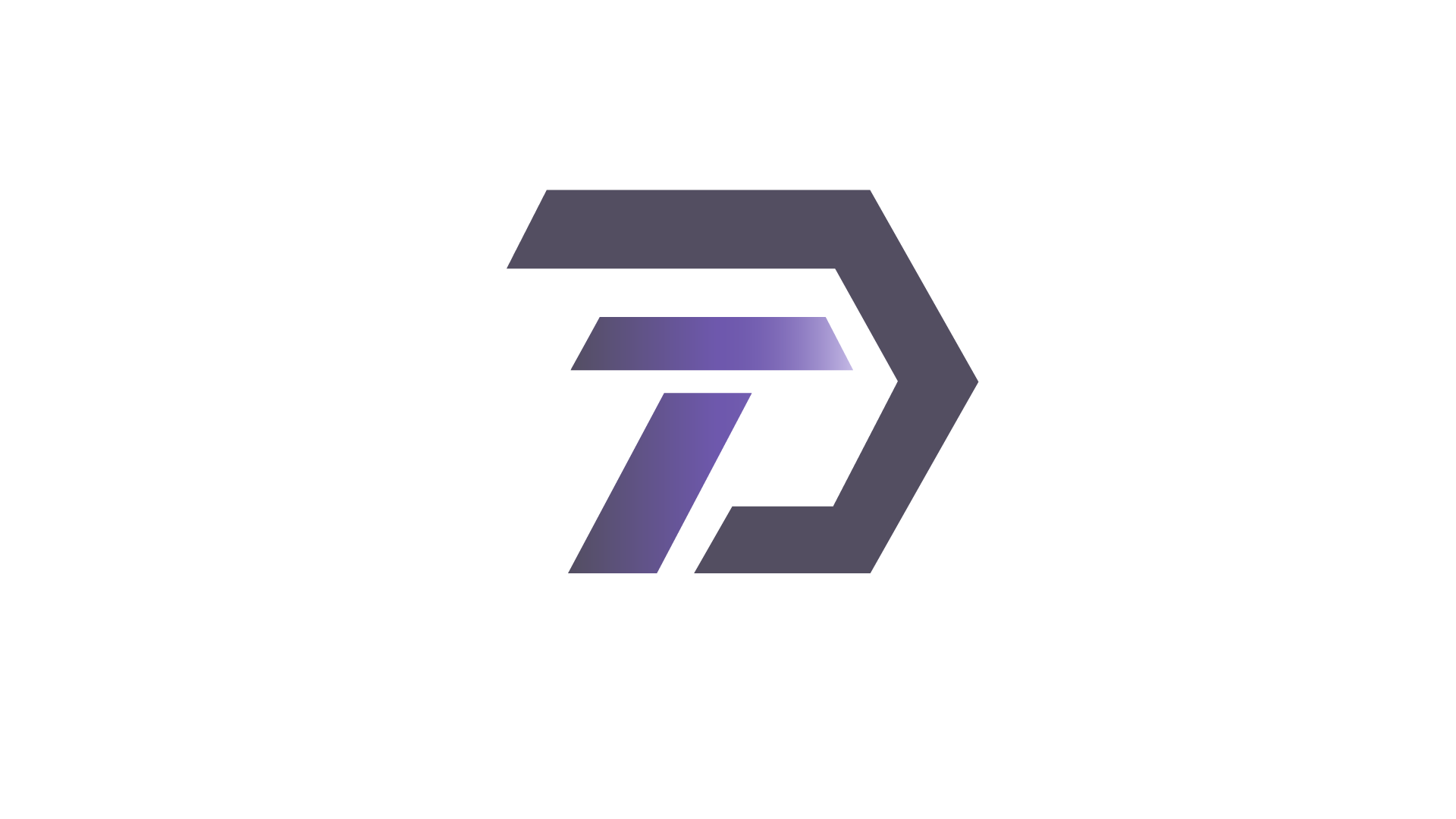
Maitre d’apprentissage : Mme. Terrier

Experts : Jean-Pierre Garnier & Thomas Tetart

CFPT

Maitre d’apprentissage : Mme. Terrier

Experts : Jean-Pierre Garnier & Thomas Tetart



Documentation technique

1.Table des matières

[2. Versions 3](#_Toc133796776)

[3. Introduction 3](#_Toc133796777)

[4. Résumé du cahier des charges 3](#_Toc133796778)

[4.1 Organisation 3](#_Toc133796779)

[4.2 But de l’application 4](#_Toc133796780)

[4.3 Fonctionnalité à réaliser 4](#_Toc133796781)

[4.4 Matériel et logiciels nécessaires 4](#_Toc133796782)

[4.5 Livrables 4](#_Toc133796783)

[5. Méthodologie 5](#_Toc133796784)

[1. S’informer 5](#_Toc133796785)

[2. Planifier 5](#_Toc133796786)

[3. Décider 5](#_Toc133796787)

[4. Réaliser 5](#_Toc133796788)

[5. Contrôler 5](#_Toc133796789)

[6. Evaluer 6](#_Toc133796790)

[6. Planification 6](#_Toc133796791)

[6.1 Backlog 6](#_Toc133796792)

[6.2 Planning prévisionnel 14](#_Toc133796793)

[6.3 Planning réel 14](#_Toc133796794)

[7. Analyse fonctionnelle 14](#_Toc133796795)

[7.1 Fonctionnalités 14](#_Toc133796796)

[7.2 Interfaces 14](#_Toc133796797)

[7.3 Analyse de l’existant 15](#_Toc133796798)

[8. Analyse organique 15](#_Toc133796799)

[8.1 Structure du projet 15](#_Toc133796800)

[8.2 Base de données 15](#_Toc133796801)

[9. Librairies et outils externes 16](#_Toc133796802)

[9.1 Git 16](#_Toc133796803)

[9.2 Visual Studio Code 16](#_Toc133796804)

[9.3 WSL 2 (Ubuntu) 16](#_Toc133796805)

[9.4 FullCalendar 17](#_Toc133796806)

[9.5 JQuery 17](#_Toc133796807)

[9.6 Composer 17](#_Toc133796808)

[9.7 Pecee Simple Router 18](#_Toc133796809)

[9.8 Autoload 18](#_Toc133796810)

[10. Plan de test et tests 19](#_Toc133796811)

[10.1 Périmètre des tests 19](#_Toc133796812)

[10.2 Equipement de test 19](#_Toc133796813)

[10.3 Scénarios de test 20](#_Toc133796814)

[10.4 Evolution des tests 22](#_Toc133796815)

[10.5 Echec des tests et raison 22](#_Toc133796816)

[11. Conclusion 23](#_Toc133796817)

[11.1 Difficultés rencontrées 23](#_Toc133796818)

[11.2 Planification 23](#_Toc133796819)

[11.3 Améliorations possibles 23](#_Toc133796820)

[11.4 Bilan personnel 23](#_Toc133796821)

[12. Annexes 24](#_Toc133796822)

[12.1 Bibliographie 24](#_Toc133796823)

# 2. Versions

|  |  |  |  |
| --- | --- | --- | --- |
| N° de version | Date | Auteur | Changements apportés |
| 1.0 | 21.03.2023 | Gabriel Martin | Version finale de la documentation du TPI finale du document pour le rendu du TPI |

# 3. Introduction

Voici la documentation technique du projet *Drafteam* réalisé dans le cadre du *Travail pratique individuel* (TPI) débutant le jeudi 27 avril 2023 et se terminant le mercredi 17 mai 2023. Ce travail a pour but de valider ma formation CFC effectuée au *Centre de Formation Professionnelle Technique* du Petit-Lancy à Genève.

Cette application web a pour but de gérer des équipes de football. Elle permet la création d’équipe, l’ajout de joueurs, la gestion d’entrainement et de match. Le tout peut être géré par un entraîneur qui a la possibilité de voir les absences et de mettre à jour lui-même le calendrier de l’équipe.

# 4. Résumé du cahier des charges

## 4.1 Organisation

T

Les informations qui vont suivre sont extraite du cahier des charges.

|  |  |  |
| --- | --- | --- |
| Élève |  | Maître d’apprentissage |
| Gabriel Martin | Anne Terrier |
| [gabriel-henri.mrtn@eduge.ch](mailto:gabriel-henri.mrtn@eduge.ch) |  | [anne.terrier@edu.ge.ch](mailto:anne.terrier@edu.ge.ch) |

|  |  |
| --- | --- |
| Experts |  |
| Jean-Pierre Garnier | Thomas Tetart |
| [jpg@gsinfo.ch](mailto:jpg@gsinfo.ch) | [thomas.tetart@laposte.net](mailto:thomas.tetart@laposte.net) |

## 4.2 But de l’application

Le but de ce projet est de proposer une application web dédiée aux clubs de football. Cette application de gestion d’équipes vous propose une interface simple permettant de créer et d’organiser vos équipes et vos entraînements.

Afin d’utiliser l’application, l’entraîneur doit tout d’abord se créer son compte en ligne. Cela étant fait, il se connecte et accède à différentes fonctionnalités :

• Création d’une équipe, ajout de joueurs ou de personnes faisant partie du staff grâce à leurs e-mails.

• Création des championnats auxquels participe l’équipe

• Création des événements (entraînement par exemple) et envoi d’invitations

Les joueurs reçoivent, de leur entraîneur, une invitation et un mot de passe par défaut. Une fois connectés à l’application, ils peuvent :

• Configurer leur profil

• Annoncer leurs présences ou leurs absences aux évènements auxquels ils ont été invités

Le site web est développé en HTML/CSS/PHP/JS

Les données sont stockées dans une base de données relationnelle MariaDB.

## 4.3 Fonctionnalité à réaliser

Visiteurs

Un visiteur accède à la page d’accueil du site. Il peut créer un compte (s’il est entraîneur et qu’il souhaite utiliser l’application pour gérer son équipe) ou se connecter.

Entraîneurs et administrateurs

Dans ce chapitre le terme entraîneur inclut également le terme administrateur.

Une fois son compte créé, l’entraîneur dispose des fonctionnalités administrateur. Il peut :

• Créer une équipe.

• Ajouter des sportifs à son équipe (joueurs ou personnels encadrants). Il doit renseigner au minimum, le nom, le prénom, la fonction (via une liste déroulante), l’email et un mot de passe par défaut. Les informations de compte seront communiquées au sportif afin qu’il puisse se connecter dans l’application.

• Créer des championnats.

• Ajouter des évènements. Pour chaque évènement, le nom, la date, l’heure de début, l’heure de fin ainsi que le type d’événement (liste déroulante) et le lieu (liste déroulante) doivent être renseignés. Les champs image et score sont facultatifs. Ils permettent d’ajouter des informations complémentaires.

• Inviter, pour chaque évènement créé, les membres de l’équipe. Il pourra choisir si l’invitation est envoyée à toutes les personnes de l’équipe, aux joueurs seulement ou aux membres encadrants seulement.

• Visualiser les confirmations de présences ou d’absences avec les commentaires liés.

• Ajouter une information sur le résultat de l’évènement en utilisant le champ score.

• Visualiser, modifier ou supprimer les équipes, sportifs, championnats et évènements liés à l’équipe dont il fait partie.

• Il est le seul à pouvoir modifier le rôle d’un sportif.

Joueurs

Le joueur reçoit les informations de connexion de son entraîneur. Lorsqu’il se connecte sur le site, il peut :

• Compléter ses informations personnelles ou les modifier. Il ne peut pas changer son rôle.

• Voir les informations de son équipe, des championnats auxquels il participe, les autres membres de son équipe.

• Voir les évènements futurs auxquels il est invité. Il peut confirmer sa présence ou dire qu’il sera absent. En cas d’absence, il devra justifier de la raison de son absence.

• Voir les évènements passés. Dans ce cas, il ne pourra plus indiquer sa présence mais verra le résultat qui aura été mis par l’entraîneur.

## 4.4 Matériel et logiciels nécessaires

• Un PC standard école, un écran

• Windows 10

• Visual Studio Code

• Suite Office

• Suite AMP à choix

• Outil de versioning (Type Git)

• Outil de création de maquette (outil proposé Evolus Pencil : <https://pencil.evolus.vn/>)

## 4.5 Livrables

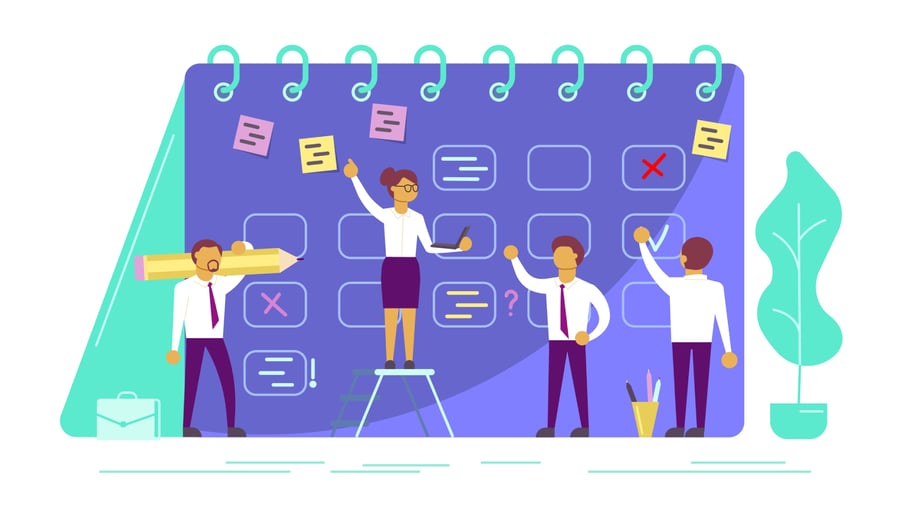
Le candidat doit livrer à son chef de projet et aux deux experts :

* Une planification initiale dès le premier jour
* Un exemplaire PDF de son mémoire avec annexe
  + Documentation technique
  + Manuel utilisateur
  + Les rapports de tests
  + Code source
* Le résumé du TPI
* Son journal de bord

Le candidat doit livrer à son chef de projet :

* Un accès aux sources du projet

# 5. Méthodologie



Pour le bon fonctionnement de ma gestion de projet, j’ai utilisé la méthode de planification en 6 étapes. Voici le schéma suivit étape par étape tout au long de mon TPI.

## 1. S’informer

La première étape consiste à prendre conscience des tâches à réaliser et à s'informer sur le problème ou le projet en question. Il s'agit de collecter systématiquement des informations pertinentes, d'identifier les parties prenantes et de comprendre les enjeux du projet.

## 2. Planifier

La deuxième étape consiste à planifier le projet. Cela implique de déterminer les objectifs à atteindre, d'établir des tâches, d’estimer le temps nécessaire pour les effectuer de déterminer les ressources nécessaires et d'établir un planning.

## 3. Décider

Une fois les informations récupérées, les tâches planifiées, il faut décider de la manière dont on va réaliser ces dernières. Pour cela, il faut classer les tâches par ordre d’importance afin de réaliser les tâches les plus importante en premier.

## 4. Réaliser

La quatrième étape consiste à la réalisations des tâches. Il faut donc suivre le planning effectuer dans l’étape « Planifier » ainsi que les décisions pour suivre les tâches dans le bon ordre.

## 5. Contrôler

Une fois la tâche terminée, il faut surveiller le progrès, que ce qui a été effectuer fonctionne correctement selon ce qui a été demander. Pour cela, il faut effectuer un protocole de test pour voir si la fonctionnalité ajouter est bien fonctionnel.

## 6. Evaluer

La dernière étape consiste à évaluer le projet. Cela implique de comparer les résultats obtenus aux objectifs établis, de mesurer l'efficacité des solutions mises en place et de déterminer les mesures à prendre pour améliorer les résultats si nécessaire.

# 6. Planification

## 6.1 Backlog

Pour le suivi de mon backlog, j'ai utilisé Trello afin de mettre à jour l'avancement et la difficulté de mes tâches.

|  |  |
| --- | --- |
| Nom | **S1 : Utilisation d’un dépôt Git** |
| Description (User story) | En tant que développer, je dois créer un dépôt git afin de pouvoir enregistrer, versionner et accéder à toutes les versions de mon site. |
| Critère d’acceptation | Dépôt Git crée en privé avec un droit d’accès pour moi, les experts et pour la maître d’apprentissage. |
| Priorité | Bloquant |

|  |  |
| --- | --- |
| Nom | **S2 : Création d’une base de données** |
| Description (User story) | En tant que développer, je dois créer ma base de données pour mon site en respectant l’UML fournit dans le cahier des charges. Je le créer donc manuellement dans DBeaver. |
| Critère d’acceptation | La base de données respecte l’UML de l’énoncer et permet à mon projet de fonctionner. |
| Priorité | Bloquant |

|  |  |
| --- | --- |
| Nom | **S3 : Implémentation de la Template** |
| Description (User story) | En tant que développer, je dois incorporer la Template dédier pour mon site. Je dois m’assurer que le design soit présentable et s’incorpore bien pour me focaliser sur le back-end |
| Critère d’acceptation | Le design respecte les consigne du cahier des charges et possède les élément requis comme une navigation, le contenue spécifique à la page etc… |
| Priorité | Bloquant |

|  |  |
| --- | --- |
| Nom | **S4 : Connexion à la base de données** |
| Description (User story) | En tant que développer, je dois avoir une connexion entre mon site et la base de données |
| Critère d’acceptation | Des données peuvent être insérer, récupérer, supprimer ou modifier dans la base de données grâce à mes fonction PHP |
| Priorité | Bloquant |

|  |  |
| --- | --- |
| Nom | **S5 : Implémentation du routage et de la structure du code** |
| Description (User story) | En tant que développer, je dois avoir un code propre, bien ordonné et facile d’utilisation. |
| Critère d’acceptation | La navigation entre les pages se fait correctement grâce Simple Router, les class sont bien relier grâce à autoload et la structure du code est en MVC. |
| Priorité | Bloquant |

|  |  |
| --- | --- |
| Nom | **S6 : Modification du design** |
| Description (User story) | En tant que développer, je dois avoir un design qui s’adapte au fonctionnalités et à l’ajout de page. |
| Critère d’acceptation | Chaque page ajoutée a un design en fonction de ce qu’elle fait. |
| Priorité | Critique |

|  |  |
| --- | --- |
| Nom | **S7 : Inscription pour l’entraîneur** |
| Description (User story) | En tant qu’entraîneur non identifié, je dois pouvoir me créer un compte entraîneur grâce à mon email, nom, prénom, mot de passe et mon poste dans l’équipe |
| Critère d’acceptation | Le test 1.1 passe. |
| Priorité | Bloquant |

|  |  |
| --- | --- |
| Nom | **S8 : Gestion de championnats** |
| Description (User story) | En tant qu’entraîneur identifié, je dois pouvoir créer, modifier et supprimer un championnats. |
| Critère d’acceptation | Les tests 2.1 et 2.2 passent. |
| Priorité | Bloquant |

|  |  |
| --- | --- |
| Nom | **S9 : Gestion d’équipe** |
| Description (User story) | En tant qu’entraîneur identifié, je dois pouvoir créer, modifier et supprimer l’équipe dont je suis l’entraîneur. |
| Critère d’acceptation | Les tests 3.1 et 3.2 passent. |
| Priorité | Bloquant |

|  |  |
| --- | --- |
| Nom | **S10 : Gestion de sportif** |
| Description (User story) | En tant qu’entraîneur identifié, je dois pouvoir créer des comptes pour les joueurs et staff de mon équipe. Je dois pouvoir aussi les modifier et supprimer. |
| Critère d’acceptation | Le test 4.1 passe. |
| Priorité | Bloquant |

|  |  |
| --- | --- |
| Nom | **S11 : Connexion** |
| Description (User story) | En tant qu’utilisateur non identifié et possédant un compte, je peux me connecter à mon compte en entrant mes informations de connexion. |
| Critère d’acceptation | Le test 5.1 passe. |
| Priorité | Bloquant |

|  |  |
| --- | --- |
| Nom | **S12 : Déconnexion** |
| Description (User story) | En tant que sportif identifié, je peux me déconnecter de mon compte. |
| Critère d’acceptation | Le test 6.1 passe. |
| Priorité | Critique |

|  |  |
| --- | --- |
| Nom | **S13 : Modification du profil** |
| Description (User story) | En tant que sportif identifié, je peux modifier mes informations sur ma page profil comme mon nom, prénom, mot de passe, photo, numéro de maillot etc. |
| Critère d’acceptation | Le test 7.1 passe |
| Priorité | Critique |

|  |  |
| --- | --- |
| Nom | **S14 : Gestions des événements** |
| Description (User story) | En tant qu’entraîneur, je dois pouvoir créer des événements, les modifier et les supprimer. |
| Critère d’acceptation | Les tests 8.1 et 8.2 passe. |
| Priorité | Bloquant |

|  |  |
| --- | --- |
| Nom | **S15 : Affichage des événements** |
| Description (User story) | En tant que sportif identifié, je dois pouvoir voir les événements de mon équipe sur le site. |
| Critère d’acceptation | Le test 9.1 passe. |
| Priorité | Critique |

|  |  |
| --- | --- |
| Nom | **S16 : Invitations aux événements** |
| Description (User story) | En tant qu’entraîneur, je dois pouvoir inviter mes joueurs et mon staff aux événements que j’ai créé. |
| Critère d’acceptation | Le test 10.1 passe. |
| Priorité | Bloquant |

|  |  |
| --- | --- |
| Nom | **S17 : Confirmations de présence** |
| Description (User story) | En tant que joueur et staff, je peux confirmer ma présence aux événements dans lesquelles l’entraîneur de mon équipe m’a invité. |
| Critère d’acceptation | Le test 11.1 passe. |
| Priorité | Critique |

|  |  |
| --- | --- |
| Nom | **S18 : Confirmation d’absence avec un commentaire** |
| Description (User story) | En tant que joueur et staff, je peux répondre absent à un événement en y ajoutant un commentaire. |
| Critère d’acceptation | Le test 12.1 passe. |
| Priorité | Critique |

|  |  |
| --- | --- |
| Nom | **S19 : Visualisation des présences et commentaire** |
| Description (User story) | En tant qu’entraîneur, je peux visualiser les présences de mon équipe aux événements que j’ai créé et les commentaires des absents. |
| Critère d’acceptation |  |
| Priorité | Critique |

|  |  |
| --- | --- |
| Nom | **S20 : Gestion des scores pour les matchs terminés** |
| Description (User story) | En tant qu’entraîneur, je peux ajouter un score au matchs qui sont terminé. |
| Critère d’acceptation |  |
| Priorité | Critique |

|  |  |
| --- | --- |
| Nom | **S21 : Modification du rôle d’un sportif** |
| Description (User story) | En tant qu’entraîneur, je peux modifier les rôles des membres de mon équipe. |
| Critère d’acceptation |  |
| Priorité | Critique |

|  |  |
| --- | --- |
| Nom | **S22 : Affichage des informations de l’équipe** |
| Description (User story) | En tant que sportif identifié, je peux voir les informations relatives à mon équipe |
| Critère d’acceptation |  |
| Priorité | Critique |

|  |  |
| --- | --- |
| Nom | **S23 : Affichage des membres de l’équipe** |
| Description (User story) | En tant que sportif identifié, je peux voir les membres de mon équipe. |
| Critère d’acceptation |  |
| Priorité | Critique |

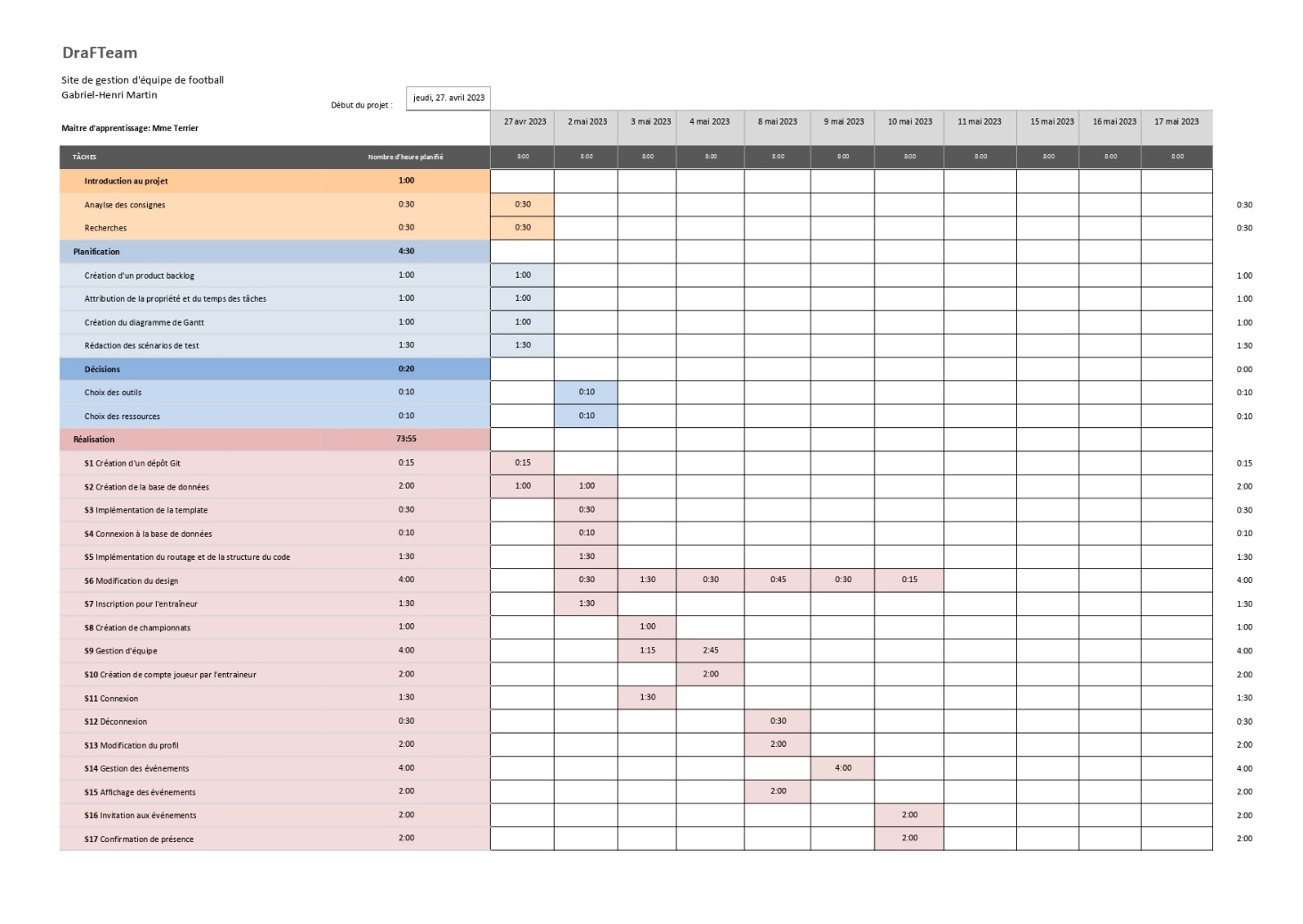
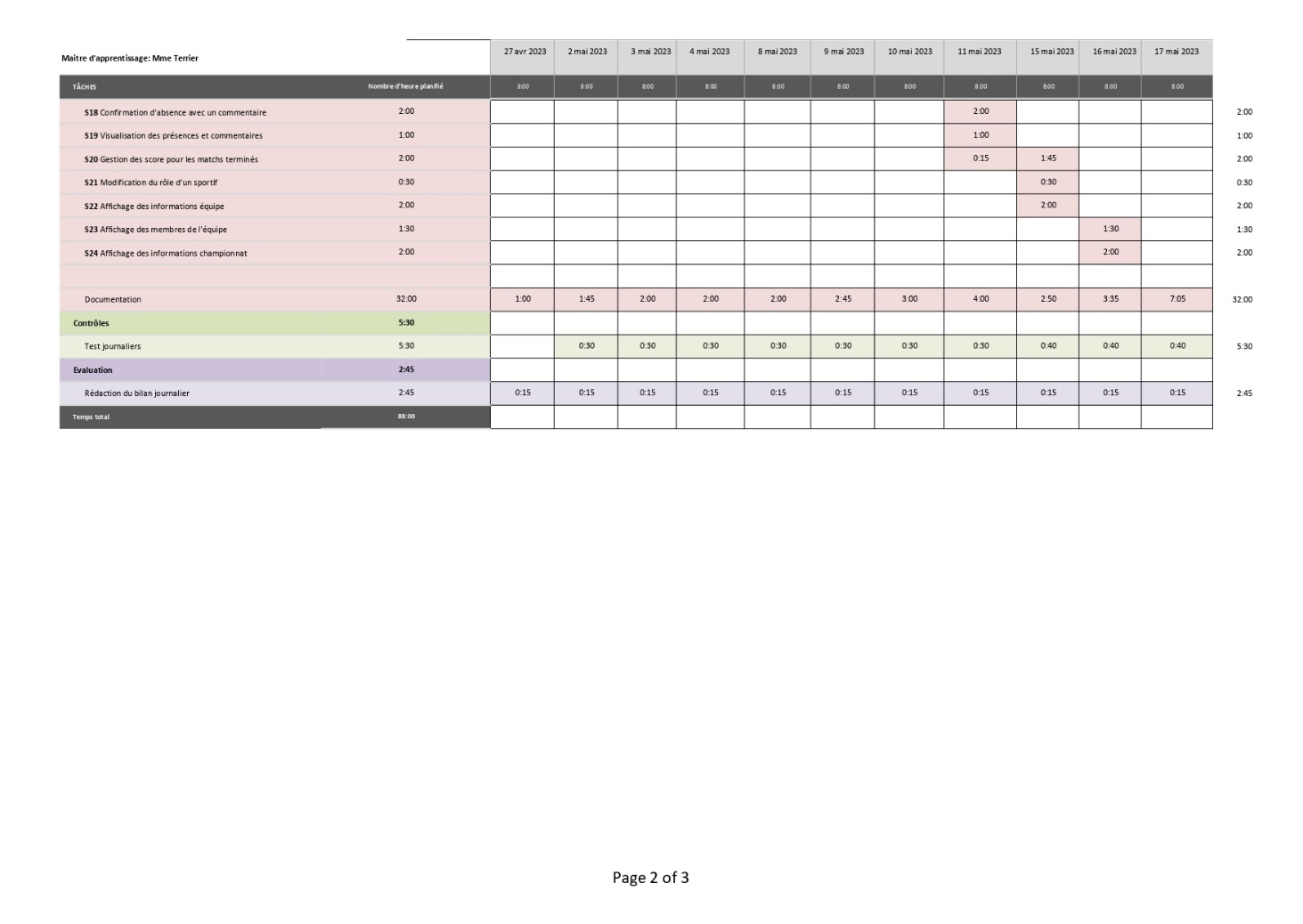
|  |  |
| --- | --- |
| Nom | **S23 : Affichage des membres de l’équipe** |
| Description (User story) | En tant que sportif identifié, je peux voir les informations relatives à mon championnat. |
| Critère d’acceptation |  |
| Priorité | Critique |

**Voici donc à quoi ressemblais mon Trello pendant la réalisation de mon TPI. J’ai pris comme exemple le deuxième jour, Mardi 2 mai :**

Une image contenant Site web

Description générée automatiquement

## 6.2 Planning prévisionnel



## 6.3 Planning réel

# 7. Analyse fonctionnelle

## 7.1 Fonctionnalités

## 7.2 Interfaces

## 7.3 Analyse de l’existant

Mon TPI implique la création d’un site web permettant la gestion d’équipe de football par un entraîneur pour ses joueurs.

Dans ce contexte, je souhaite explorer les avantages et les inconvénients de l'utilisation d'un site web plutôt que d'un outil dédié à la gestion d’un agenda comme l’agenda de google.

**Comparaisons avec l’agenda partagé de google**

Avec la liberté offerte par la programmation web, je trouve que l'agenda Google offre moins de possibilités et d'améliorations possibles que pour un site web. Dans mon cas, le PHP, l'HTML/CSS et le JS permettent de répondre à n'importe quelle demande, que ce soit avec des librairies externes ou du code effectué par moi-même.

En effet, avec l'agenda Google, nous pourrions utiliser quelque chose de déjà fait et nous n'aurions pas besoin de recréer la roue. La manipulation est simple et est intégrée à l'environnement Google, ce qui en fait un outil pratique et simple.

Cependant, avec l'agenda Google, nous sommes vite limités en termes de fonctionnalités. Dans le cas de mon site web, la bibliothèque JavaScript FullCalendar permet de créer un agenda de manière dynamique et de ne pas repartir de zéro. Ensuite, le PHP m'a permis d'enregistrer mes utilisateurs et événements dans une base de données et l'HTML/CSS m'a permis d'avoir un design unique et propre à mon site.

Pour finir, l'utilisation de mon site web plutôt que de l'agenda de Google favorise grandement la possibilité de fonctionnalités, la gestion d'équipe en tant que telle, les statistiques et l'interaction entre l'entraîneur et les joueurs pour les présences. De plus, les améliorations futures telles qu'un classement pour les championnats seraient une plus-value non négligeable.

# 8. Analyse organique

## 8.1 Structure du projet

## 8.2 Base de données

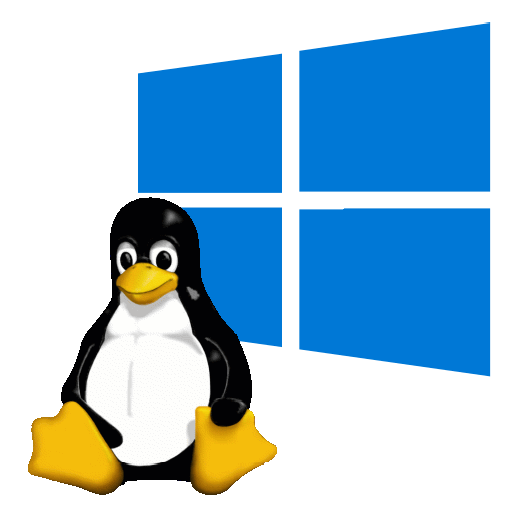
# 9. Librairies et outils externes

## Formation Git | Human Coders Formations9.1 Git

Git est un logiciel de gestion de versions distribué, développé par Linus Torvalds, le créateur du noyau Linux. Pour assurer le suivi des différentes versions de mon application, j'ai opté pour l'utilisation d'un dépôt distant sur la plateforme GitHub. Cela me permet de stocker mon code source en ligne et de suivre les changements apportés à chaque version de mon application.

## Microsoft Apps9.2 Visual Studio Code

Visual Studio Code est un environnement de développement intégré (IDE) polyvalent, simple à utiliser et doté d'une large gamme de fonctionnalités. Il permet de programmer dans divers langages de programmation et sa modularité est renforcée par son système d'extension. Personnellement, j'ai utilisé Visual Studio Code pour développer mon application et cet IDE m'a été extrêmement utile pendant la phase de développement.

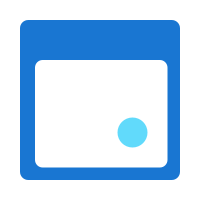


## 9.3 WSL 2 (Ubuntu)

WSL 2 est une fonctionnalité de Windows qui permet de créer un environnement Linux à l'intérieur de Windows, ce qui équivaut à avoir une machine virtuelle Linux intégrée dans la machine Windows. Cela permet d'exécuter des applications Linux sur Windows sans avoir besoin d'une machine virtuelle distincte. Mon application ainsi que ma base de données ont donc tourné sur un serveur en local.

## 

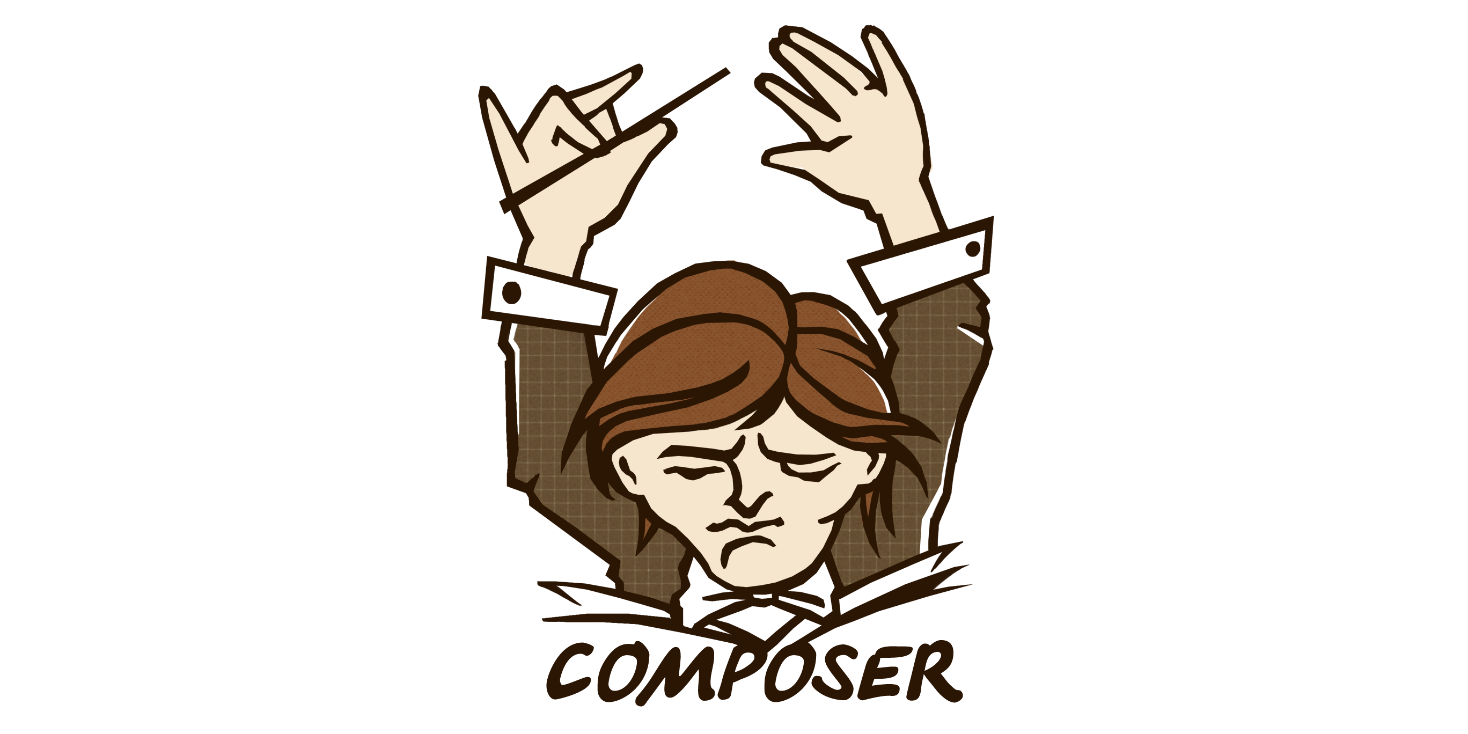
## 9.4 FullCalendar

FullCalendar est une bibliothèque JavaScript open-source permettant de créer des calendriers interactifs sur une page web. J'ai utilisé FullCalendar dans mon application de gestion d'équipe de football pour afficher les matchs et les événements importants dans un calendrier facile à naviguer. Grâce à FullCalendar, j'ai pu afficher les matchs, les entraînements et divers événements, puis permettre aux utilisateurs de cliquer dessus pour obtenir plus d'informations. L'utilisation de FullCalendar a grandement amélioré l'expérience utilisateur de mon application en offrant une vue d'ensemble de tous les événements à venir et en permettant aux utilisateurs de planifier en conséquence.

## jQuery · GitHub9.5 JQuery

JQuery est une bibliothèque JavaScript utile pour faciliter la manipulation du DOM et la gestion des événements. Dans mon code, l'utilisation de jQuery a simplifié la sélection des éléments HTML et la gestion des événements, ce qui a rendu le code plus court et plus facile à comprendre.

## 9.6 Composer

Composer est un gestionnaire de dépendances qui m'a permis de gérer efficacement les bibliothèques, les frameworks et les packages dont mon projet avait besoin. Grâce à Composer, j'ai pu automatiser l'installation et la mise à jour de ces dépendances en créant un fichier de configuration nommé "composer.json". En utilisant Packagist, un référentiel de packages, Composer a pu trouver et télécharger les packages requis.

L'utilisation de Composer a apporté de nombreux avantages à mon projet. En gérant les dépendances de manière centralisée, j'ai pu éviter les erreurs d'incompatibilité et m'assurer que toutes les bibliothèques et les packages étaient correctement installés. En outre, en permettant une mise à jour automatique des dépendances, Composer facilite la maintenance de mon code. Enfin, en automatisant la gestion des packages, Composer m'a fait gagner du temps et m'a permis de me concentrer sur le développement de mon application.

## 9.7 Pecee Simple Router

Pecee Simple Router est une bibliothèque PHP qui permet la gestion des routes dans une application web. Elle permet de définir des URL pour chaque page ou fonctionnalité de mon site et de rediriger les requêtes HTTP vers la bonne page en fonction de l'URL demandée.

Pecee Simple Router a rendu mon site plus intéressant de plusieurs façons. Tout d'abord, elle permet de créer des URL plus propres, ce qui améliore l'expérience utilisateur. Ensuite, elle facilite la maintenance et l'organisation de mon code en séparant clairement les actions à réaliser en fonction des routes demandées. Enfin, elle améliore les performances de mon site en réduisant la charge de traitement nécessaire pour déterminer la page ou la fonctionnalité à exécuter.

En somme, l'utilisation de Pecee Simple Router m’a aidé à améliorer la qualité globale de mon site web en améliorant l'expérience utilisateur, la maintenance et l'organisation de mon code, et les performances.

## 9.8 Autoload

Une image contenant texte, sombre, orange

Description générée automatiquementL'autoload est une fonctionnalité de PHP qui permet de charger automatiquement les classes nécessaires à l'exécution de mon code, sans que j'aie à les inclure manuellement dans chaque fichier. Cela permet de simplifier le processus de développement, en évitant les erreurs d'inclusion de fichiers et en facilitant la maintenance du code en utilisant des namespace et use.

Utiliser l'autoload m'a permis de travailler plus efficacement en évitant les tâches répétitives liées à l'inclusion manuelle de classes. Ensuite, elle a facilité la maintenance de mon code en permettant une organisation claire de mes classes dans des dossiers spécifiques. Enfin, elle a permis de réduire le temps de chargement de mes pages en évitant le chargement de classes inutiles.

# 10. Plan de test et tests

## 10.1 Périmètre des tests

J'ai effectué mes tests sur deux navigateurs différents : Google Chrome et Edge.

Pour m'assurer que toutes les fonctionnalités de mon projet fonctionnent correctement, j'ai utilisé deux types de comptes : un compte utilisateur en tant que joueur et un compte entraîneur. Chaque jour, j'ai exécuté une série de tests pour vérifier que l'ajout d'une nouvelle fonctionnalité n'affecterait pas le fonctionnement d'une autre.

## 10.2 Equipement de test

**Environnement**

Durant le développement du projet, j'ai effectué mes tests sur l'environnement de développement suivant : Google chrome version 112.0.5615.138 (Build officiel) (64 bits) sous Windows version 10.0.19045 Build 19045.

Concernant le serveur, j'ai utilisé Apache version 2.4.41 Ubuntu version 20.04 avec PHP version 8.1.2, ainsi qu'un serveur MariaDB pour ma base de données.

**Fichiers de test**

Pour les tests, j'ai utilisé un compte entraîneur et un compte utilisateur standard, et j'ai créé manuellement les évènements. Toutes ces informations sont stockées dans la base de données et sont accessibles via le script SQL disponible dans le dossier /SQL du projet.

## 10.3 Scénarios de test

Un scénario de tests pour évaluer les fonctionnalités principales de l'application est disponible ci-dessous. Ce scénario de tests vient en complément des tests unitaires, qui eux permettent de vérifier que chaque méthode fonctionne correctement. Le scénario de tests prévoit d'évaluer toutes les fonctionnalités disponibles dans le programme afin de s'assurer qu'elles fonctionnent toutes correctement.

|  |  |
| --- | --- |
| Nom | * 1. **Création d’un compte entraîneur** |
| User story | S7 : Inscription pour l’entraîneur |
| Situation | Lorsqu’un visiteur arrive sur le site et souhaite se créer un compte entraîneur, il peut s’inscrire. En appuyant sur le bouton s’inscrire, si le compte est déjà créé il y a une erreur qui s’affiche. Si des informations sont fausses, un message s’affiche en fonction de l’erreur effectuée. |
| Résultats obtenus | Si le compte est existant ou que des champs sont faussement remplis, un message d’erreur s’affiche. Sinon, il est redirigé sur la page d’accueil avec la possibilité de créer une équipe dans un championnat qui existe déjà ou alors de créer un championnat puis une équipe. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | * 1. **Création de championnats** |
| User story | S8 : Gestion de championnats |
| Situation | Lorsqu’un entraîneur est identifié, il peut choisir de créer un championnat pour pouvoir y placer son équipe. Il doit entrer le nom du championnat et la saison, si un de ces deux champs est mal renseigné, un message d’erreur s’affiche. |
| Résultats obtenus | Un message d’erreur s’affiche si les champs ne sont pas bien renseignés sinon le championnat est créé dans la base de données et l’entraîneur est redirigé sur la page création d’équipe. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **2.2 Modifications / suppression championnats** |
| User story | S8 : Gestion de championnats |
| Situation | Lorsqu’un entraîneur est identifié il peut supprimer et modifier les championnats qu’il a créés uniquement. Pour chaque action un message de confirmation s’affiche. |
| Résultats obtenus | Les messages s’affichent pour toutes les actions et les changements sont faits dans la BD. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **3.1 Création d’équipe** |
| User story | S9 : Gestion d’équipe |
| Situation | L’entraîneur crée une équipe en rentrant les champs demander. Des messages d’erreur s’affichent si des champs sont mal renseignés. |
| Résultats obtenus | Les messages d’erreurs s’affichent en cas d’erreur sinon il est redirigé vers la page pour créer les comptes des différents sportifs qu’il y aurait dans son équipe. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **3.2 Modifications / suppression d’équipe** |
| User story | S9 : Gestion d’équipe |
| Situation | L’entraîneur peut supprimer ou modifier une équipe. Des messages de confirmations s’affichent pour chaque action. |
| Résultats obtenus | Les messages s’affichent pour toutes les actions et les changements sont faits dans la BD. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **4.1 Création de compte joueur par l’entraîneur** |
| User story | S10 : Création de compte joueur par l’entraîneur |
| Situation | L’entraineur peut accéder à la page de création de compte de sportif. Il peut créer un compte pour un staff ou joueur en entrant son email et un mot de passe. |
| Résultats obtenus | Un message d’erreur s’affiche si le compte est déjà créé ou que les champs sont mal renseignés. Sinon, le compte est bien créé et ajouter à la base de données. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **5.1 Connexion** |
| User story | S11 : Connexion |
| Situation | Un sportif possédant un compte peut se connecter en entrant son email et mot de passe. Un message d’erreur s’affiche si l’email ou le mot de passe est faux. |
| Résultats obtenus | Un message d’erreur s’affiche si les champs sont mal renseignés. Sinon, le sportif est connecté et redirigé sur la page d’accueil. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **6.1 Déconnexion** |
| User story | S12 : Déconnexion |
| Situation | Un sportif connecté peut se déconnecter en appuyant sur le bouton « se déconnecter » dans la barre de navigation. Un message de confirmation s’affiche. |
| Résultats obtenus | Le message de confirmation s’affiche et s’il est validé, le sportif est déconnecté sinon il retourne sur la page d’accueil. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **7.1 Modifications du profil** |
| User story | S13 : Modification du profil |
| Situation | Un sportif connecté peut modifier les informations sur son profil en se rendant dans la page profil. Il peut modifier sa photo de profil, son numéro, son nom etc. En appuyant sur confirmer un message de confirmation s’affiche. |
| Résultats obtenus | Le message de confirmation s’affiche et s’il est validé, les modifications seront effectuées dans la BD et le sportif reste sur la page profil avec un message comme quoi tout est bien modifié. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **8.1 Création d’un événements** |
| User story | S14 : Gestion des événements |
| Situation | L’entraineur peut créer un événement avec un nom une date un lieux et une description. A la création de l’événement un message qui montre que l’événement a bien été créer s’affiche. |
| Résultats obtenus | Le message de confirmations s’affiche et l’événement est créer dans la BD. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **8.2 Modifications / suppression d’événement** |
| User story | S14 : Gestion des événements |
| Situation | Lorsqu’un entraîneur est identifié il peut supprimer et modifier les événements qu’il a créés uniquement. Pour chaque action un message de confirmation s’affiche. |
| Résultats obtenus | Les messages s’affichent pour toutes les actions et les changements sont faits dans la BD. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **9.1 Affichage des événements** |
| User story | S15 : Affichage des événements |
| Situation | Quand un sportif est identifié, il peut voir tous les événements dans la page agenda sur le calendrier et les 5 prochains événements sur la page d’accueil. |
| Résultats obtenus | Les événements sont bien affichés sur ces deux pages. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **10.1 Invitations au événements** |
| User story | S16 : Invitations aux événements |
| Situation | L’entraîneur peut inviter les joueurs et staff qu’il souhaite en les cochant dans une liste comportant tous les sportifs de l’équipe. Une fois l’invitation envoyée, les sportifs invités verront deux boutons sur l’événement pour choisir d’être présent ou absent. |
| Résultats obtenus | Les sportif inviter voient bien les deux boutons sur l’événement. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **11.1 Confirmations de présence** |
| User story | S17 : Confirmations de présence |
| Situation | Les sportifs invités à l’événement peuvent confirmer leur présence en appuyant sur le bouton « Présent ». Si le sportif veut changer de choix, il peut toujours changer sa présence en cliquant sur l’événement et le changer dans la page de l’événement en question. |
| Résultats obtenus | Si le bouton « Présent » est cliqué, le bouton devient vert et le bouton « Absent » disparaît. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **12.1 Confirmations d’absence avec un commentaire** |
| User story | S18 : Confirmations d’absence avec un commentaire |
| Situation | Les sportifs invités à l’événement peuvent confirmer leur absence en appuyant sur le bouton « Absent ». Si le sportif veut changer de choix, il peut toujours changer sa présence en cliquant sur l’événement et le changer dans la page de l’événement en question. |
| Résultats obtenus | Si le bouton « Absent » est cliqué, un textbox apparaît pour pouvoir laisser une explication à cette absence et après confirmation, le bouton absent devient rouge et le bouton « Présent » disparaît. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **13.1 Visualisation des présences et commentaire** |
| User story | S19 : Visualisation des présences et commentaire |
| Situation | L’entraîneur peut se rendre sur la page d’un événement et voir qui sera présent ou absent à l’événement et voir les excuses pour les absences |
| Résultats obtenus | Accès à la page de l’événement en cliquant sur l’événement en question et la liste des présences s’affiche. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **14.1 Gestion des scores pour les matchs terminés** |
| User story | S20 : Gestion des scores pour les matchs terminés |
| Situation | En se rendant sur la page du match quand il est terminé, l’entraîneur peut ajouter un score, le modifier ou l’enlever. |
| Résultats obtenus | Accès à la page de l’événement en cliquant sur l’événement en question et après avoir validé le résultat tout l’effectif de l’équipe pourra consulter le score. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **15.1 Modification du rôle d’un sportif** |
| User story | S21 : Modification du rôle d’un sportif |
| Situation | En se rendant sur la page gestion équipe, l’entraîneur peut changer les rôles des sportifs de l’équipe. Il doit confirmer les modifications pour qu’elle s’effectue. |
| Résultats obtenus | Accès à la page de gestion d’équipe que pour l’entraîneur. Un message de confirmation s’affiche quand il fait un changement de rôle puis et rediriger sur la page gestion d’équipe. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **16.1 Affichage des informations de l’équipe** |
| User story | S22 : Affichage des informations de l’équipe |
| Situation | En se rendant sur la page « Mon équipe », chaque sportif peut voir les informations relatives à leur équipe. |
| Résultats obtenus | Accès à la page « Mon équipe » pour chaque sportif identifié. Une fois sur la page, toutes les infos de l’équipe s’affichent. |
| Statut | KO |

|  |  |
| --- | --- |
| Nom | **17.1 Affichage des membres de l’équipe** |
| User story | S23 : Affichage des membres de l’équipe |
| Situation | En se rendant sur la page d’accueil, chaque sportif peut voir les membres de l’équipe dans un carrousel au milieu de la page. |
| Résultats obtenus | Le carrousel affiche tous les membres de l’équipe et défile correctement. |
| Statut | KO |

## 10.4 Evolution des tests

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Num test | J1  Jeu, 27.04 | J2  Mar 02.05 | J3  Mer 03.05 | J4  Jeu 04.05 | J5  Lun 08.05 | J6  Mar 09.05 | J7  Mer 10.05 | J8  Jeu 11.05 | J9  Lun 15.05 | J10  Mar 16.05 | J11  Mer 17.05 |
| **1.1** | ❌ |  |  |  |  |  |  |  |  | ✔️ |  |
| **2.1** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **2.2** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **3.1** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **3.2** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **4.1** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **5.1** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **6.1** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **7.1** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **8.1** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **8.2** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **9.1** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **10.1** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **11.1** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **12.1** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **13.1** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **14.1** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **15.1** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **16.1** | ❌ |  |  |  |  |  |  |  |  |  |  |
| **17.1** | ❌ |  |  |  |  |  |  |  |  |  |  |

## 10.5 Echec des tests et raison

# 11. Conclusion

## 11.1 Difficultés rencontrées

## 11.2 Planification

## 11.3 Améliorations possibles

## 11.4 Bilan personnel

# 12. Annexes

## 12.1 Bibliographie

**Code repris**

* Template du site repris et modifier de mon ancien projet : <https://github.com/gotham226/CloseLake.git>

Lui-même repris de cette Template :

[GitHub - dzarrorn/NFTwebsite\_html-sass](https://github.com/dzarrorn/NFTwebsite_html-sass)

* Peece Simple Router : <https://github.com/skipperbent/simple-php-router>

**Sites utilisés**

* Documentation PHP : <https://www.php.net/>
* Documentation FullCalendar : <https://fullcalendar.io/docs>