

SocialShuffle



Candidat : **DOS SANTOS Samuel**, FIN2 2024

Chef de Projet : **CHARMIER Grégory**

Expert 1 : **VENRIES Luc**

Expert 2 : **OBERSON Bernard**

Durée : **88h**, du 13.05.2024 au 03.06.2024

Table des matières

1 SPÉCIFICATIONS	5
1.1 TITRE	5
1.2 INTRODUCTION.....	5
1.3 OBJECTIFS	5
1.4 POINTS TECHNIQUES	6
1.5 MATÉRIEL ET LOGICIELS À DISPOSITION	6
1.6 PRÉREQUIS	6
1.7 LIVRABLES.....	6
2 PLANIFICATION INITIALE	7
3 ANALYSE	8
3.1 MÉTHODOLOGIE DE TRAVAIL.....	8
3.2 ORGANISATION DES RÉSULTATS DU TRAVAIL.....	8
3.3 ENVIRONNEMENT	8
3.3.1 <i>Laravel</i>	8
3.3.2 <i>MVC</i>	9
3.3.3 <i>L'ORM Eloquent</i>	10
3.4 CONCEPTION ET ANALYSE	10
3.4.1 <i>MCD / MLD</i>	10
3.4.2 <i>Diagramme de flux</i>	11
3.4.3 <i>Maquettes graphiques</i>	12
3.5 STRATÉGIE DE TESTS	16
3.5.1 <i>Tests de la base de données avec les Factories</i>	16
3.5.2 <i>Laravel Dusk</i>	17
3.5.3 <i>Tests automatisés avec GitHub Action</i>	17
3.5.4 <i>Scenarios des tests End-To-End</i>	18
3.6 RISQUES TECHNIQUES	18
4 RÉALISATION	19
4.1 MISE EN PLACE DE LARAVEL 11	19
4.1.1 <i>Le fichier .env</i>	19
4.1.2 <i>Tester le fonctionnement de l'installation</i>	20
4.1.3 <i>Installation de Tailwind CSS</i>	21
4.2 MIGRATIONS	22
4.2.1 <i>Table t_user (users)</i>	22
4.2.2 <i>Table t_team (teams)</i>	23
4.2.3 <i>Table t_member (members)</i>	23
4.2.4 <i>Table t_group (groups)</i>	24
4.2.5 <i>Table t_dispatch (group_member)</i>	24
4.3 MISE EN PLACE DES CONTRÔLEURS	24
4.3.1 <i>Contrôleurs de ressources</i>	25
4.4 MISE EN PLACE DES ROUTES	26
4.4.1 <i>La route « / » (Racine)</i>	28
4.4.2 <i>Ressources</i>	28

4.4.3	<i>Authentification</i>	28
4.5	MISE EN PLACE DES VUES	29
4.5.1	<i>Layout</i>	29
4.5.2	<i>Composant réutilisable pour l'affichage des membres</i>	29
4.5.3	<i>Utilisation du même formulaire pour la création et la modification</i>	30
4.5.4	<i>Affichage des groupes</i>	31
4.5.5	<i>Difficultés rencontrées</i>	32
4.6	FORMULAIRES DE CRÉATION D'UNE ÉQUIPE AVEC LES MEMBRES	32
4.7	AUTHENTIFICATION	34
4.7.1	<i>Formulaire d'authentification</i>	34
4.7.2	<i>Traitement de l'authentification</i>	35
4.7.3	<i>Déconnexion</i>	36
4.8	DROITS DES UTILISATEURS	36
4.8.1	<i>Création d'une Policy</i>	36
4.8.2	<i>Utilisation des Policies dans les vues</i>	37
4.8.3	<i>Utilisation des Policies dans les contrôleurs</i>	38
4.9	ALGORITHME DE RÉPARTITION DES MEMBRES	38
4.9.1	<i>Mise en place</i>	38
4.9.2	<i>Pistes d'améliorations</i>	41
4.10	MISE EN PLACE DU QR CODE	41
4.10.1	<i>Installation</i>	41
4.10.2	<i>Intégration de la génération de QR codes</i>	41
4.10.3	<i>Remarque</i>	42
4.11	IMPORTATION DEPUIS UN FICHIER CSV	42
4.11.1	<i>Mise en place</i>	42
4.11.2	<i>Téléchargement d'un fichier d'exemple</i>	43
4.11.3	<i>Difficulté rencontrée</i>	43
4.12	PROCÉDURE UTILISÉE POUR LE DÉPLOIEMENT	44
4.12.1	<i>Contraintes</i>	44
4.12.2	<i>Procédure</i>	44
4.12.3	<i>Exécuter des commandes Artisan depuis une URL</i>	46
4.13	TROIS MESURES DE SÉCURITÉ	47
4.13.1	<i>Token CSRF</i>	47
4.13.2	<i>Validation des formulaires</i>	47
4.13.3	<i>Hachage des mots de passe</i>	49
4.14	TROIS MESURES POUR LE « RESPONSIVE DESIGN »	49
4.14.1	<i>Affichage des cartes en fonction de la largeur de l'écran</i>	49
4.14.2	<i>Barre de navigation</i>	50
4.14.3	<i>Boutons de gestion des membres</i>	51
4.15	MISE EN PLACE DES TESTS LARAVEL DUSK DANS GITHUB ACTIONS	52
4.15.1	<i>Installation de Laravel Dusk</i>	52
4.15.2	<i>Mise en place de GitHub Action</i>	53
4.15.3	<i>Création d'un test Laravel Dusk</i>	53
4.15.4	<i>Risque de sécurité</i>	56
4.15.5	<i>Difficultés rencontrées</i>	56
5	TESTS	57
5.1	DOSSIER DES TESTS	57
6	CONCLUSION	58

6.1	BILAN DES FONCTIONNALITÉS DEMANDÉES	58
6.1.1	<i>Génération des groupes</i>	58
6.2	BILAN DE LA PLANIFICATION	58
6.3	BILAN PERSONNEL	60
GLOSSAIRE	61
7 ANNEXES	63
7.1	RÉSUMÉ DU RAPPORT DE TPI.....	63
7.1.1	<i>Situation de départ</i>	63
7.1.2	<i>Mise en œuvre</i>	63
7.1.3	<i>Résultats</i>	63
7.2	ACCÈS AU REPOSITORY GITHUB.....	64
7.3	BIBLIOGRAPHIE	64
7.4	TABLE DES ILLUSTRATIONS.....	65
7.5	PLANNING.....	66
7.7	JOURNAL DE TRAVAIL.....	75

1 SPÉCIFICATIONS

1.1 Titre

Réalisation de l'application web SocialShuffle ayant pour but de mélanger des participants afin de favoriser les interactions sociales.

1.2 Introduction

Ce projet a pour but de réaliser une application Web ayant pour but de mélanger des membres d'une équipe dans des groupes hétérogènes afin de favoriser les interactions sociales.

Un des cas d'utilisation possible est lors du début d'une année scolaire, où les élèves ne se connaissent pas encore entre eux, de proposer des activités de team building en faisant en sorte que tous les participants puissent se rencontrer et interagir.

Cette application permettra donc de retirer la contrainte de devoir créer des groupes manuellement.

L'enjeu principal est la mise en pratique du Framework Laravel^{*1} dans un projet concret.

1.3 Objectifs

Les objectifs évalués pour ce projet sont les suivants :

- Les formulaires doivent afficher des erreurs tout en remplaçant à nouveau les formulaires lors de ces occurrences. Cela implique une validation minutieuse des champs.
- Mise en place de mesures de sécurité
- Un manuel d'installation sous la forme d'un fichier README.md sur le repository GitHub.
- Mettre en place et expliquer l'algorithme permettant la répartition des membres dans les différents groupes.
- Le site doit être responsive*.
- Justification des choix faits dans le MCD / MLD / MPD

En plus de ces objectifs, des éléments supplémentaires devront être intégrés à l'application si le temps le permet :

- Importation des membres depuis un fichier CSV en plus de l'entrée manuelle sur le site.
- Affichage d'un code QR permettant de rediriger vers la page montrant les différents groupes.

¹ Les termes marqués du symbole * sont détaillés dans le glossaire

1.4 Points techniques

L'application devra respecter les contraintes techniques suivantes :

- Utilisation du Framework **Laravel**.
- Une base de données **MySQL** reliée à l'application.
- L'interface de l'application devra être responsive.

1.5 Matériel et logiciels à disposition

- Un PC de l'ETML
- Visual Studio avec des extensions facilitant le développement en Laravel
 - PHP
 - PHP Intelephense
 - PHP Namespace Resolver
 - Laravel Extra Intellisense
 - Laravel Blade formatter
 - Beautify Blade
- PHP 8.3 pour l'environnement de développement local
- Composer
- uWamp pour le serveur de base de données de développement
- node.js

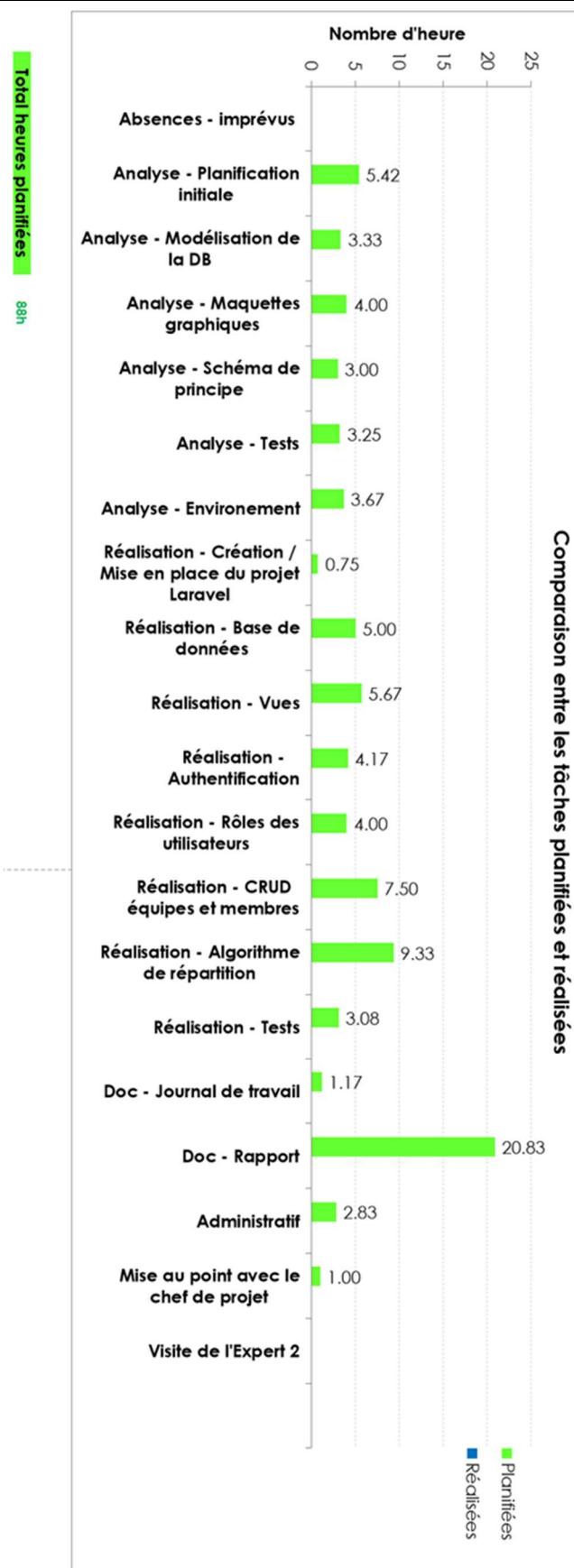
1.6 Prérequis

- Connaître les notions de la Programmation Orienté Objet
- Savoir coder en PHP
- Connaitre les principes de développement avec le Framework Laravel
 - Les routes
 - MVC avec Laravel
 - CLI Artisan
 - Migrations
 - Moteur de templating Blade

1.7 Livrables

- Planification initiale
- Rapport du projet
- Journal de travail
- Le code de l'application fourni en annexe

2 PLANIFICATION INITIALE



3 ANALYSE

3.1 Méthodologie de travail

La méthode des **six pas** sera utilisée pour la réalisation de ce projet. Celle-ci s'inscrit parfaitement dans le cadre du TPI où une limite de temps est imposée.

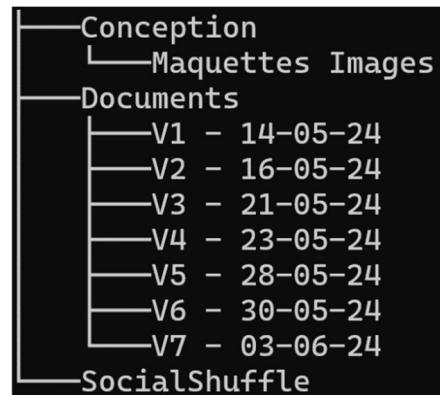
3.2 Organisation des résultats du travail

Afin d'assurer la bonne organisation du travail tout au long du projet, la procédure suivante est appliquée :

Chaque mise à jour aux experts (mardi et jeudi soir) correspond à une nouvelle version des documents.

- Les versions sont citées au début du nom du document
- Chaque jour, au moins un commit et push sont effectués afin d'enregistrer le travail effectué
- Un répertoire est créé, dedans se trouve l'entièreté du projet. Il s'agit du repository.

Voici donc la structure du repository liés au projet :



1. Organisation du projet

3.3 Environnement

3.3.1 Laravel

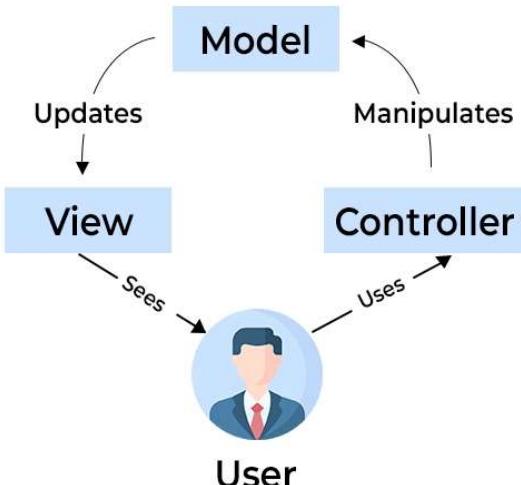
Laravel² est un Framework WEB basé sur le modèle MVC*. Il possède une syntaxe expressive qui permet de faire appel à des fonctions de manière naturelle. L'interface en ligne de commandes (CLI), Artisan^{*3} permet de pouvoir interagir rapidement avec l'application (créer un contrôleur, effectuer des migrations vers la base de données ou même optimiser l'application pour le déploiement).

² Documentation officielle de Laravel <https://laravel.com/docs/11.x>

³ Documentation sur Artisan : <https://laravel.com/docs/11.x/artisan#introduction>

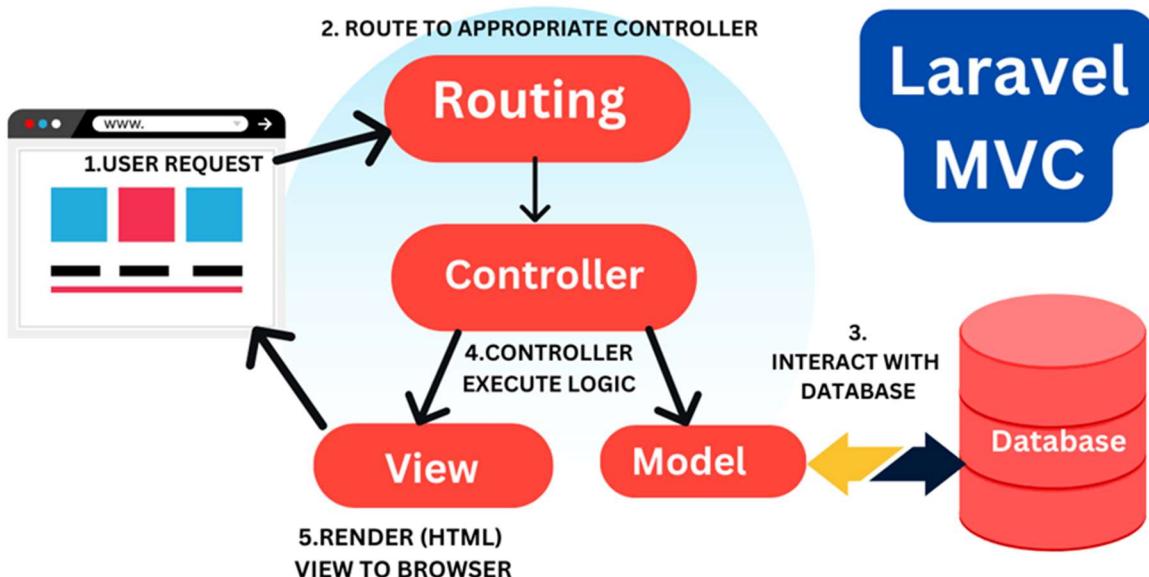
3.3.2 MVC

Le modèle MVC (Modèle*, Vue*, Contrôleur*) vise à diviser la charge de travail d'une application. Le modèle a pour fonction d'interagir avec la base de données en créant ou en récupérant des enregistrements, la vue quant à elle sert à proposer un affichage pour les application et le contrôleur permet de transporter les requêtes entre le modèle et la vue.



2. Représentation schématique du modèle MVC

Laravel respecte ce principe fondamental en ajoutant le concept de routes*. Celles-ci permettent à partir de l'Url et le type de requête effectuée, de rediriger vers une fonction spécifique d'un contrôleur, ou plus globalement, d'exécuter directement du code.



3. Représentation schématique du modèle MVC appliquée à Laravel.

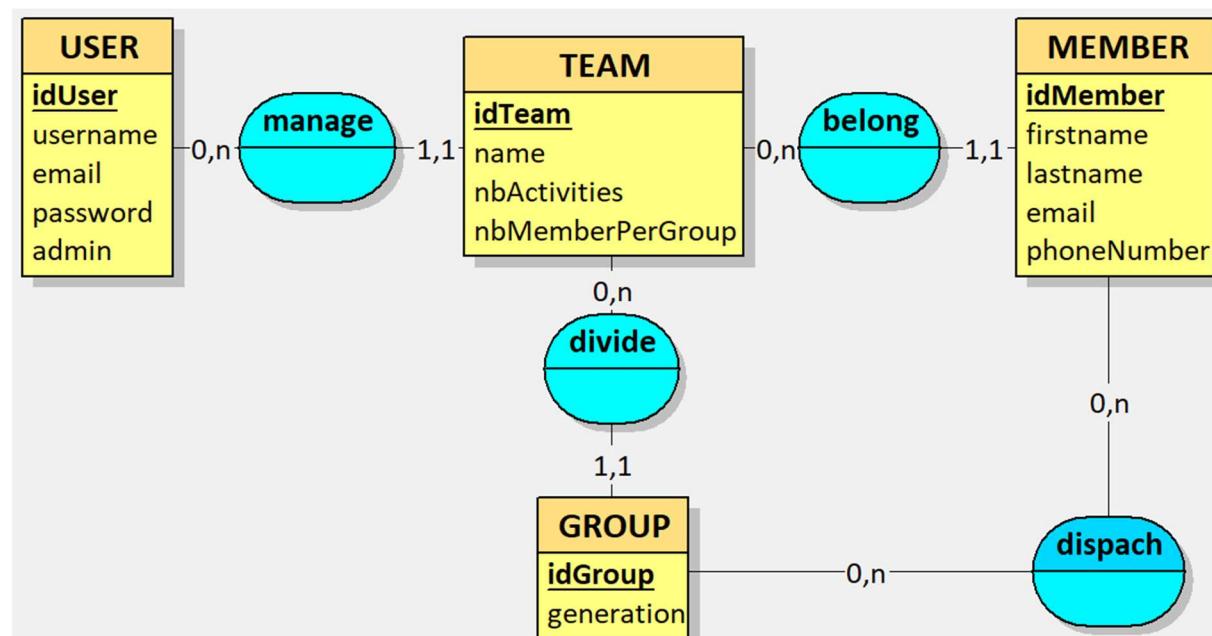
3.3.3 L'ORM Eloquent

Laravel, met à disposition un ORM* qui permet de décrire des requêtes à la base de données avec du code PHP au lieu de taper directement des requêtes SQL. L'utilisation de cet outil permet de rendre les interactions avec la base de données plus naturelles, ce qui est censé rendre le développement plus rapide et agréable.

3.4 Conception et analyse

3.4.1 MCD / MLD

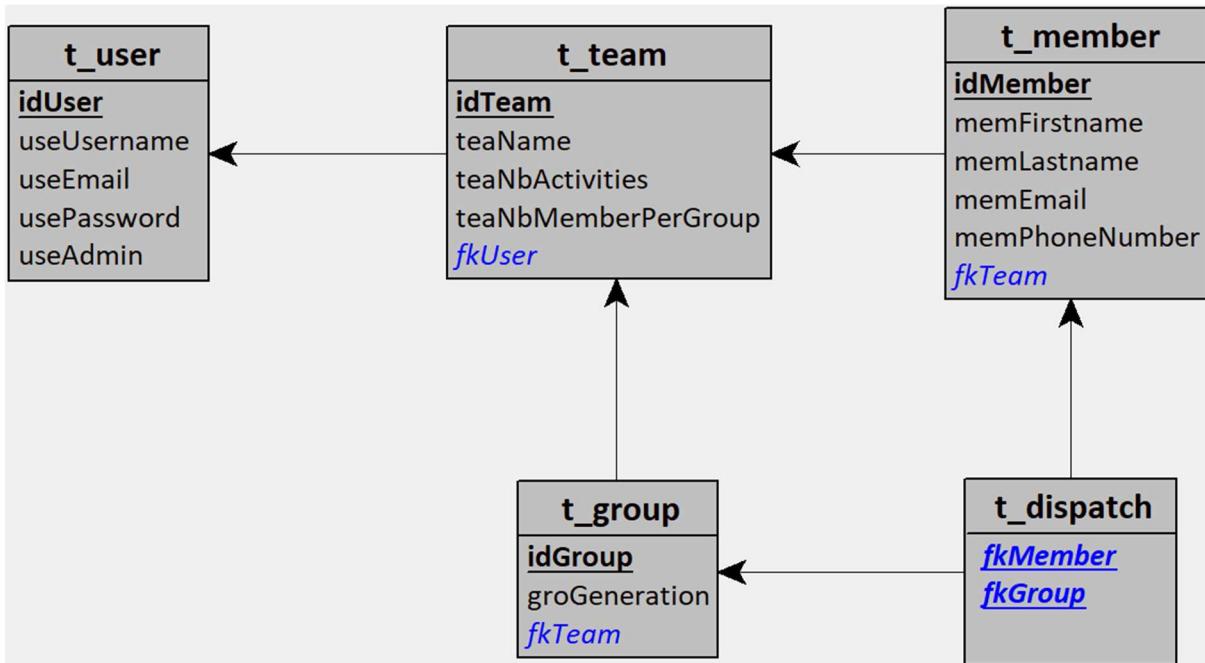
Le MCD et le MLD respectent ici les normes de codage de l'ETML. Afin de respecter les bonnes pratiques de développement en Laravel, ce seront les normes de Laravel qui seront utilisées durant la réalisation. Ces normes permettent d'automatiser⁴ certains éléments de l'application. Ceci est la raison pour laquelle les noms diffèrent entre les modèles et le code.



4. Modèle Conceptuel de Données (MCD) pour l'application SocialShuffle

⁴ Exemple d'automatisation avec les normes de Laravel :

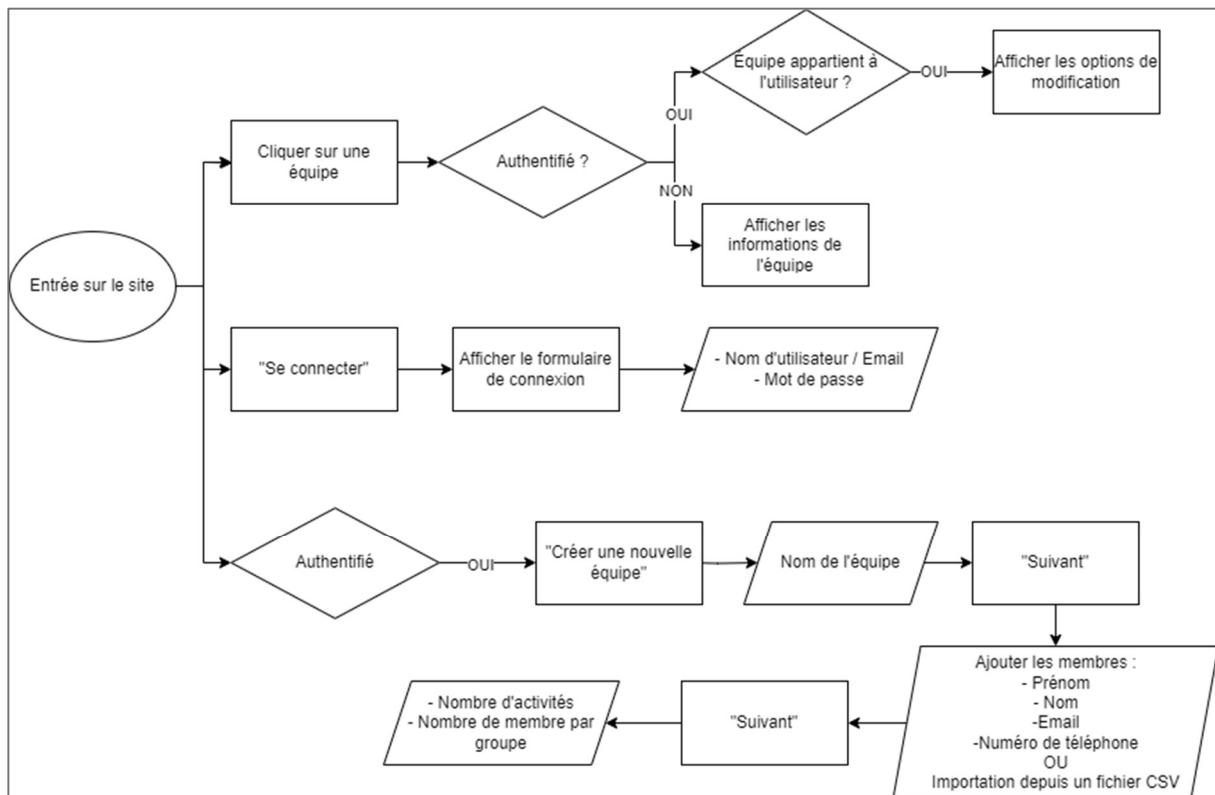
<https://laravel.com/docs/11.x/migrations#:~:text=Since%20this%20syntax,rewritten%20like%20so%3A>



5. Modèle Logique de Données (MLD) pour l'application SocialShuffle

3.4.2 Diagramme de flux

Ici est démontré une vue générale des interactions possibles avec le site, il y est notamment montré les différentes interactions nécessaires pour créer une équipe.

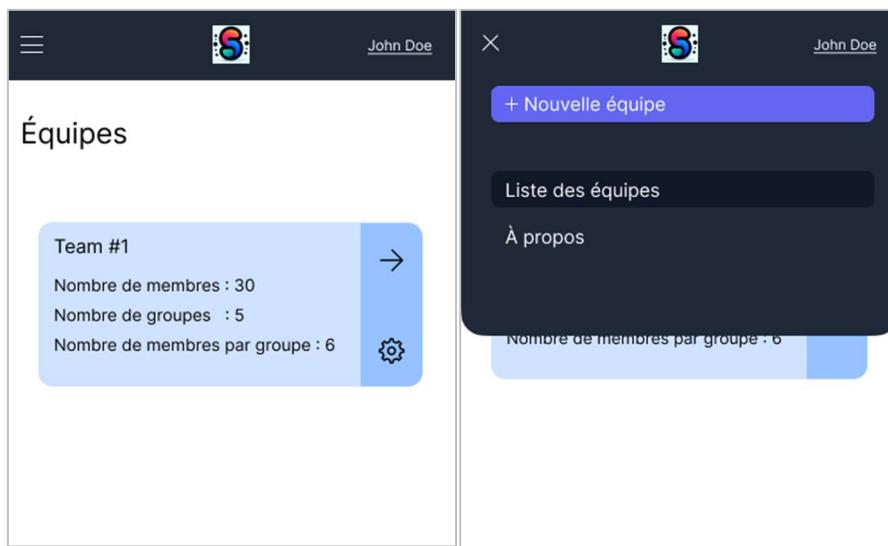


6. Représentation des interactions principales avec le site.

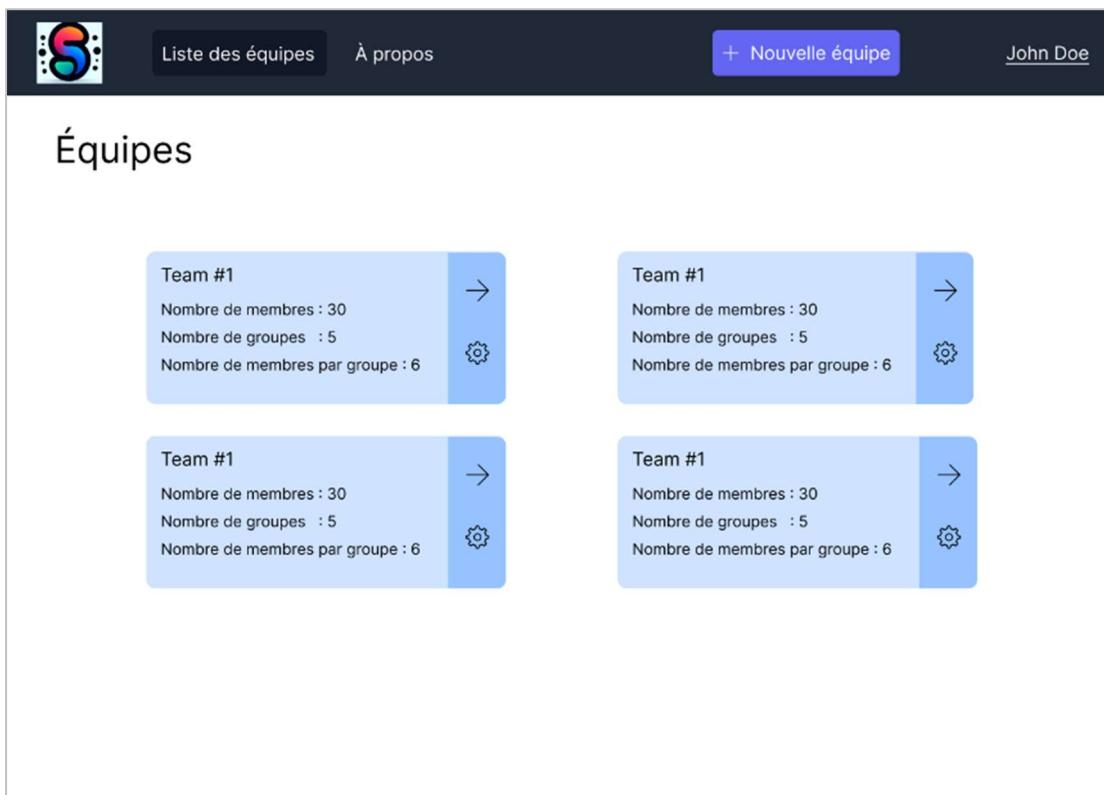
3.4.3 Maquettes graphiques

3.4.3.1 Page principale

Les maquettes graphiques qui suivront, permettront de donner une perspective de ce à quoi le site devrait ressembler. Le but est notamment de définir les principales différences entre la version mobile et la version desktop (responsive design). Elles permettent également d'avoir un aperçu des messages d'erreurs ainsi que les messages d'avertissements. Un des aspects majeurs est l'affichage des informations sous forme de cartes. Cela permet de distinguer intuitivement chaque équipe par exemple.



7. Maquette de la page principale en version mobile. Avec le menu ouvert et fermé.



8. Maquette de la page principale en version desktop

Dans la version mobile, contrairement à la version desktop, pour pouvoir accéder à la navigation, il faut ouvrir un menu. Cela évite que trop d'éléments se trouvent en même temps sur l'écran ce qui réduirait l'expérience utilisateur.

3.4.3.2 Exemple de formulaire

The wireframe shows a mobile application interface for adding team members. At the top, there is a header bar with a menu icon, a logo, and the text "John Doe". Below the header, the title "Ajouter les membres" is displayed. The form consists of several input fields: "Prénom" (First Name) with value "John", "Nom" (Last Name) with value "Doe", and "Email" with value "john.doe@examplech". An error message "Email invalide. Veuillez respecter le format : exemple@example.com" is shown below the email field. The "Email" field is highlighted in red. There is also a field for "Numéro de téléphone" (Phone Number) with value "+41790000000". Below the form are two buttons: "Ajouter un membre" (Add member) and "Importer depuis un CSV" (Import from CSV). A summary section at the bottom shows "Nombre de membres : 25" and a card for "John Doe" with the email "john.doe@example.ch".

9. Exemple de formulaire avec la maquette d'ajout des membres en version mobile.

Cette maquette montre un exemple de formulaire. Celui-ci correspond à l'ajout des membres lors de la création d'une équipe. On peut y voir une erreur dans le champ « email » où l'utilisateur fictif a oublié un point, ce qui a déclenché l'affichage d'un message d'erreur.

Une fois les membres ajoutés, ils apparaissent sous forme de cartes en dessous du formulaire.

Voici également l'équivalent de ce formulaire dans la version desktop :

Ajouter un membre

Prénom
John

Nom
Doe

Email
john.doe@examplech
Email invalide. Veuillez respecter le format : exemple@example.com

Numéro de téléphone
+41792101885

Suivant →

Importer depuis un CSV

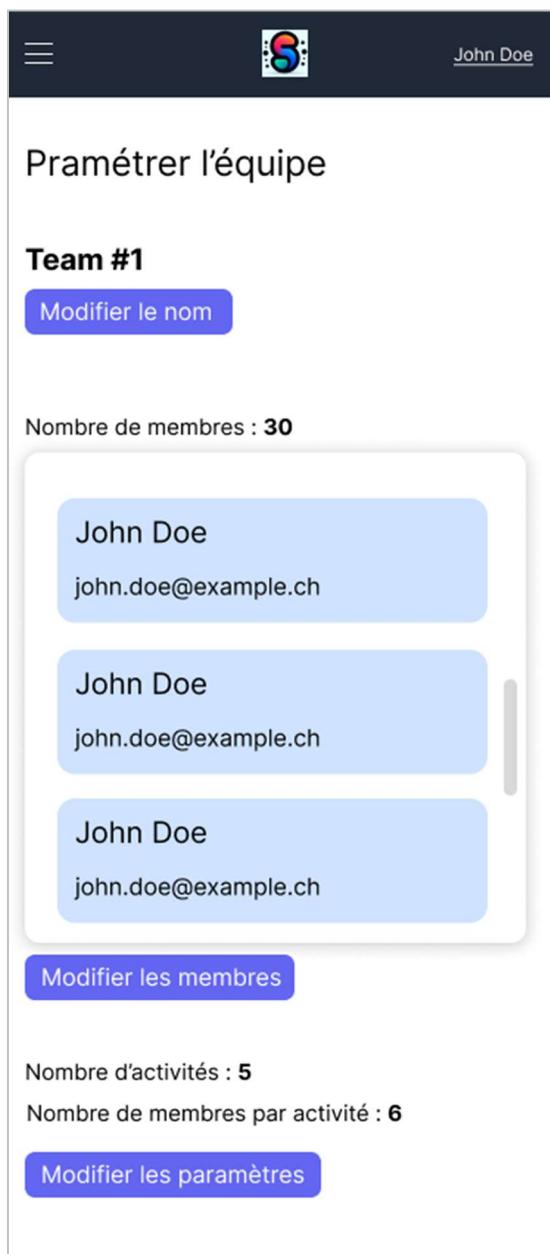
Ajouter un membre

Nombre de membres : 25

John Doe john.doe@examplech	John Doe john.doe@examplech
John Doe john.doe@examplech	John Doe john.doe@examplech

10. Exemple de formulaire avec la maquette d'ajout des membres en version mobile.

3.4.3.3 Paramétrage de l'équipe



11. Page de paramétrage d'une équipe

Cette maquette montre la page où sont affiché toutes les options de paramétrage qu'il est possible de faire une fois qu'une équipe a été créée. Chaque bouton sera ensuite censé rediriger vers un formulaire de modification.

3.5 Stratégie de tests

3.5.1 Tests de la base de données avec les Factories

Les factories⁵ sont un outil qui permet de tester une base de données et les modèles associés en définissant un type de données fictives dans des tables et colonnes données. Cela permet habituellement de populer facilement et rapidement une base

⁵ Factories: <https://laravel.com/docs/11.x/database-testing>

de données afin de vérifier que l'application fonctionne correctement avec ces dernières. Les factories sont utilisées en parallèle des seeders^{*} qui eux exécutent la création des données et l'enregistrement dans la base de données.

3.5.2 Laravel Dusk

Laravel Dusk^{*6} est un outil qui s'intègre à Laravel et qui s'inscrit dans la catégorie des tests de navigateurs. Cet outil permet notamment de simuler des interactions qu'effectuerai un utilisateur avec le site.

Ces types de tests sont très intéressants car ils sont effectués en interagissant directement avec l'interface du site, comme un utilisateur. Cela permet donc par exemple de tester le bon fonctionnement des différents formulaires. Il sera donc possible de tester la création des équipes avec leurs membres.

Une fois Laravel Dusk installé et les différents scénarios de test configurés, ils peuvent être lancés depuis le CLI Artisan.

```
$ php artisan dusk
```

Dusk doit être en mesure d'accéder à un navigateur Web, c'est pour cela que le pilote Chrome (ChromeDriver^{*}) doit également être installé.

3.5.3 Tests automatisés avec GitHub Action

Github Actions^{*78} est un outil d'intégration continue avec lequel il est possible d'automatiser des tests lors de chaque push. Cela permettra lors de chaque modification de s'assurer que l'application fonctionne toujours comme prévu. Si une erreur apparaît pendant un test, une notification sera affichée sur GitHub. Dans le cadre de ce projet, il sera utilisé pour automatiser les tests Dusk.

⁶ Laravel Dusk : <https://laravel.com/docs/11.x/dusk>

⁷ GitHub Actions : <https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions>

⁸Laravel Dusk sur Github Actions : <https://laravel.com/docs/11.x/dusk#running-tests-on-github-actions>

3.5.4 Scenarios des tests End-To-End

La planification des scénarios de tests End-to-End* permet de définir tous les aspects fonctionnels de l'application. Ainsi si tous les tests sont validés, on pourra considérer que l'application est fonctionnelle et donc utilisable.

Nom du test	Description	Critère de validation
LoginTest	Test de l'authentification. L'utilisateur entre ses authentifiants dans les champs requis.	L'utilisateur est connecté.
WrongLoginTest	Test de l'authentification avec des informations erronées.	L'utilisateur n'est pas authentifié, il reste sur la page de connexion.
CreateTeamTest	Test de la création d'une équipe complète avec ses membres et ses groupes.	L'équipe est créée sans erreurs. Tous les membres sont présents dans leurs groupes respectifs et selon les générations.
CreateWrongMemberTest	Test de la création d'une équipe en insérant des données erronées.	Des messages d'erreur sont retournés avec les données erronées.
DeleteTeamTest	Tester la suppression d'une équipe complète, qui contient des membres et des groupes.	L'équipe est supprimée sans erreurs retournées.

3.6 Risques techniques

Plusieurs éléments techniques de ce projet risquent de présenter une difficulté plus marquée. L'élément le plus évident est la mise en place de l'algorithme permettant de distribuer les membres dans leurs différents groupes.

Un autre point technique est celui de l'implémentation d'un code QR qui redirige sur une page spécifique du site. Pour ce point particulier, des librairies devraient déjà exister pour Laravel. Une librairie qui pourrait être utilisée est « simple-qrcode⁹ ».

⁹ Tutoriel d'installation et utilisation de « simple-qrcode » : <https://www.akilischool.com/cours/laravel-generer-un-qr-code-avec-simple-qrcode>

4 RÉALISATION

4.1 Mise en place de Laravel 11

L'installation de Laravel¹⁰ se fait à l'aide de l'outil en ligne de commande Composer* qui permet de créer le projet et d'installer toutes les dépendances nécessaires au fonctionnement du Framework.

La création du projet se fait donc avec la commande suivante :

```
$ composer create-project Laravel/laravel SocialShuffle
```

4.1.1 Le fichier .env

Le fichier .env contient les valeurs de configuration de l'application. Ces valeurs définissent notamment la connexion à la base de données ou bien encore le statut de l'application, c'est-à-dire si le site est lancé localement dans un environnement de développement ou sur un serveur de production.

Par défaut, ce fichier se trouve dans la liste des fichiers ignorés par Git car il contient des informations de configuration sensibles comme le mot de passe de connexion à la base de données. Ces informations ne doivent donc en aucun cas être rendues publiques.

¹⁰ Installation de Laravel : <https://laravel.com/docs/11.x>

Afin d'adapter ces valeurs d'environnement à ce projet, il faut modifier certains champs comme le nom de l'application, sa langue et entrer les informations de connexion à la base de données.

The screenshot shows a terminal window with a dark background. It displays the contents of a .env file. Several environment variables are highlighted with red boxes: APP_NAME, APP_ENV, APP_KEY, APP_DEBUG, APP_TIMEZONE, APP_URL, APP_LOCALE, APP_FALLBACK_LOCALE, APP_FAKE_LOCALE, APP_MAINTENANCE_DRIVER, APP_MAINTENANCE_STORE, BCRYPT_ROUNDS, LOG_CHANNEL, LOG_STACK, LOG_DEPRECATED_CHANNEL, LOG_LEVEL, DB_CONNECTION, DB_HOST, DB_PORT, DB_DATABASE, DB_USERNAME, and DB_PASSWORD. These highlighted variables represent the configuration values that have been modified for the development environment.

```
1 APP_NAME=SocialShuffle
2 APP_ENV=local
3 APP_KEY=base64:vnuMnfZnTYIJxuLPxJvf3xzaUaxyY3Ydd/xRoENT/Fk=
4 APP_DEBUG=true
5 APP_TIMEZONE=UTC
6 APP_URL=http://localhost
7
8 APP_LOCALE=fr
9 APP_FALLBACK_LOCALE=en
10 APP_FAKE_LOCALE=en_US
11
12 APP_MAINTENANCE_DRIVER=file
13 APP_MAINTENANCE_STORE=database
14
15 BCRYPT_ROUNDS=12
16
17 LOG_CHANNEL=stack
18 LOG_STACK=single
19 LOG_DEPRECATED_CHANNEL=null
20 LOG_LEVEL=debug
21
22 DB_CONNECTION=mysql
23 DB_HOST=127.0.0.1
24 DB_PORT=3306
25 DB_DATABASE=SocialShuffle
26 DB_USERNAME=root
27 DB_PASSWORD=root
```

12. Configuration initiale du fichier .env pour un environnement de développement. (En rouge, les valeurs qui ont été modifiées).

4.1.2 Tester le fonctionnement de l'installation

Après avoir lancé le serveur de base de données locale, il faut effectuer une première migration afin de créer les tables natives du Framework dans la base de données. Pour cela, il faut lancer la commande suivante :

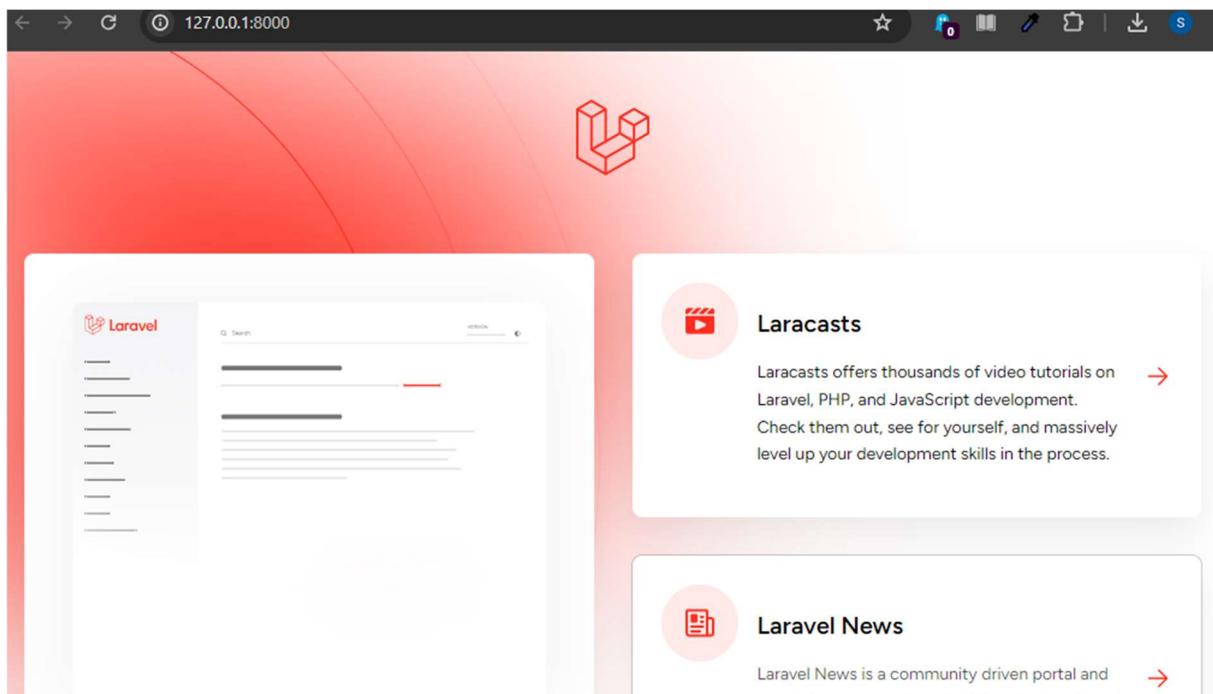
```
$ php artisan migrate
```

Il nous sera ensuite demandé si l'on souhaite créer la base de données (si celle-ci n'existe pas encore).

Une fois cela fait, on peut tester que Laravel ait été correctement installé en lançant le serveur de développement fourni avec Artisan, l'interface en ligne de commande fourni avec Laravel. Pour lancer le serveur, il faut lancer la commande suivante :

```
$ php artisan serve
```

On peut ensuite accéder au site depuis le navigateur et constater que Laravel fonctionne :



13. Page par défaut de laravel.

4.1.3 Installation de Tailwind CSS

L'installation de Tailwind^{*11} se fait en installant les dépendances avec npm :

```
$ npm install -D tailwindcss postcss autoprefixer
```

Il faut ensuite générer les fichiers « tailwind.config.js » et « postcss.config.js » :

```
$ npx tailwindcss init -p
```

Une fois que le fichier « tailwind.config.js » a été créé, il faut ajouter les chemins pour tous les fichiers Template. Dans ce cas-ci, les fichiers « .js » et « .vue » sont laissés afin de prévoir le cas où l'application sera maintenue et mise à jour dans le futur avec ces technologies.

```
export default {  
    content: [  
        "./resources/**/*.blade.php",  
        "./resources/**/*.js",  
        "./resources/**/*.vue",  
    ],
```

Et pour terminer, il faut ajouter les directives suivantes dans le fichier app.css qui est présent par défaut à la création d'un projet Laravel.

Ces directives permettent d'importer les éléments CSS de Tailwind :

¹¹ Installation de Tailwind : <https://tailwindcss.com/docs/guides/laravel#vite>

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

4.2 Migrations

Les tableaux qui suivent représentent les tables qui ont été créées après avoir lancé les migrations*. Afin de respecter les bonnes pratiques, ce sont les conventions de nommage de Laravel qui sont utilisées¹².

4.2.1 Table *t_user* (*users*)¹³

Colonne	Type	Longueur	NULL	Autre	Valeur par défaut
id	BIGINT	-	Non	UNSIGNED AUTO_INCREMENT	-
username	String	-	Non	-	-
email	String	320	Non	-	-
password	String	-	Non	-	-
admin	Bool	1	Oui	-	NULL
created_at	DATE	-	Oui	-	NULL
updated_at	DATE	-	Oui	-	NULL

Cette table contient toutes les informations d'authentification des utilisateurs. La colonne **email** est limitée à 320 caractères afin de respecter la norme RFC 3696¹⁴.

¹² Conventions de nommage de Laravel : <https://webdevetc.com/blog/laravel-naming-conventions/>

¹³ Le nom de table entre parenthèses représente le nommage appliqué dans l'application qui respecte les noms de Laravel.

¹⁴ Norme RFC 3696 : <https://www.rfc-editor.org/rfc/rfc3696#page-7>

4.2.2 Table *t_team* (teams)

Colonne	Type	Longueur	NULL	Autre	Valeur défaut
id	BIGINT	-	Non	UNSIGNED AUTO_INCREMENT	-
name	String	-	Non	-	-
nb_activities	INTEGER	-	Oui	-	NULL
group_size	INTEGER	-	Oui	-	NULL
user_id	BIGINT	-	Non	UNSIGNED AUTO_INCREMENT CASCADE_ON_DELETE	-
created_at	DATE	-	Oui	-	NULL
updated_at	DATE	-	Oui	-	NULL

Cette table contient les données relatives aux équipes. La colonne **nbActivities** contient le nombre d'activités pour une équipe. La colonne **group_size** contient le nombre de membres qu'il devrait y avoir dans une équipe. Dans la modélisation, cette colonne était appelée **nbMemberPerGroup** mais le nom a été changé afin de le rendre plus simple et explicite. Ces deux informations permettent de générer les groupes dans lesquels se trouveront les membres.

4.2.3 Table *t_member* (members)

Colonne	Type	Longueur	NULL	Autre	Valeur par défaut
id	BIGINT	-	Non	UNSIGNED AUTO_INCREMENT	-
firstname	String	-	Non	-	-
lastname	String	-	Non	-	-
email	String	320	Non	-	-
phone_number	String	-	Non	-	-
team_id	BIGINT	-	Non	UNSIGNED AUTO_INCREMENT CASCADE_ON_DELETE	-
created_at	DATE	-	Oui	-	NULL
updated_at	DATE	-	Oui	-	NULL

Cette table contient les données relatives aux membres d'une équipe. La colonne **team_id** correspond à la clé étrangère qui lie le membre à une équipe.

4.2.4 Table *t_group* (*groups*)

Colonne	Type	Longueur	NULL	Autre	Valeur par défaut
id	BIGINT	-	Non	UNSIGNED AUTO_INCREMENT	-
generation	INTEGER	-	Non	-	-
team_id	BIGINT	-	Non	UNSIGNED AUTO_INCREMENT CASCADE_ON_DELETE	-
created_at	DATE	-	Oui	-	NULL
updated_at	DATE	-	Oui	-	NULL

Cette table correspond aux groupes auquel appartiendront les membres d'une équipe. La colonne **generation** permet de spécifier la génération/activité à laquelle appartient le groupe.

4.2.5 Table *t_dispatch* (*group_member*)

Colonne	Type	Longueur	NULL	Autre	Valeur par défaut
member_id	BIGINT	-	Non	UNSIGNED AUTO_INCREMENT CASCADE_ON_DELETE	-
group_id	BIGINT	-	Non	UNSIGNED AUTO_INCREMENT CASCADE_ON_DELETE	-

Cette table est la table de pivot* qui lie entre elles les tables **groups** et **members** dans leur relation « many-to-many ».

4.3 Mise en place des contrôleurs

Les contrôleurs responsables de la gestion des membres et des équipes sont des contrôleurs de ressources. Une ressource est un modèle sur lequel on peut notamment effectuer les opérations CRUD.

Afin de respecter les contraintes d'appartenance (par exemple un ensemble de membre appartient à une équipe), les ensembles des ressources liées à un modèle

« parent » seront gérés dans ce dernier. Par exemple, la création des groupes est gérée depuis le contrôleur des équipes.

4.3.1 Contrôleurs de ressources

Les contrôleurs de ressources offrent une structure de code prédéfinie une fois créés. La création d'un contrôleur de ressource se fait avec la commande suivante :

```
$ php artisan make:controller TeamController --resource
```

Une fois exécuté, un nouveau fichier est créé sous **app/http/Controllers**. Ce fichier contient les fonctions présentes par défaut qui permettent de réaliser les opérations CRUD sur la ressource. Les fonctions par défaut sont les suivantes :

- **index()** : Affiche un listing de la ressource.
- **create()** : Affiche le formulaire pour créer une nouvelle ressource.
- **store()** : Stock la nouvelle ressource dans la base de données.
- **show()** : Affiche une ressource spécifique.
- **edit()** : Affiche le formulaire pour modifier une ressource spécifique.
- **update()** : Modifie une ressource spécifique dans la base de données.
- **destroy()** : Supprime une ressource spécifique de la base de données.

Laravel met à disposition des méthodes pour créer des enregistrements en base de données. La création et la modification se fera par l'assignement de masse qui permet de créer ou modifier plusieurs attributs d'un modèle avec une seul ligne de code.

Cette méthode présente un risque de sécurité car il est possible, si il n'y avait pas de protections, d'insérer des attributs à modifier dans une requête HTML. Pour contrer cela, il faut définir les attributs qui sont autorisés pour l'assignement de masse¹⁵ dans le modèle correspondant en créant une propriété nommée **\$fillable** :

```
protected $fillable = [  
    'name',  
    'nbActivities',  
    'nbMemberPerGroup',  
    'user_id',  
];
```

¹⁵ Assignement de masse : <https://laravel.com/docs/11.x/eloquent#mass-assignment>

Avec cela, il est désormais possible de créer modèles avec plusieurs attributs en une ligne de code. Voici un exemple avec la création d'une équipe :

```
public function store(TeamRequest $request)
{
    $validatedName = $request->validated();
    $team = Team::create($validatedName);
    return ...;
}
```

4.4 Mise en place des routes

Laravel met à disposition un système de routes permettant d'exécuter des instructions en fonction d'une URL spécifique et d'un type de requêtes. Les routes sont définies dans le répertoire « routes » dans le fichier web.php. Laravel offre la possibilité de nommer les routes afin de les catégoriser et de les grouper. Les routes de ressources n'ont pas besoin d'être nommées car cela est fait automatiquement.

Le lien entre les routes et les contrôleurs des ressources se fait en spécifiant la classe correspondant à la route. Pour ce qui est des autres contrôleurs, il faut également préciser la classe du contrôleur, mais également le nom de la méthode à exécuter.

Voici la version finale des routes qui ont été mises en place pour le fonctionnement de l'application :

```
// root
Route::get('/', function () {
    return redirect()->route('team.index');
});

// Team algorithm (Additional routes to a resource controller must be defined before calling the resource route)

// CSV import
Route::post('import-csv/{team}', [TeamController::class, 'importCSV'])
    ->name('team.importCSV');

// Create groups
Route::get('groupForm/{team}', [TeamController::class, 'groupForm'])
    ->name('team.groupForm');

Route::post('createGroups/{team}', [TeamController::class,
    'generateGroups'])
    ->name('team.createGroups');

// Show a specific activity
Route::get('showActivity/{team}/{generation}', [TeamController::class,
    'showActivity'])
    ->name('team.showActivity');

// Download CSV example
Route::get('csv-example',[TeamController::class, 'csvDownload'])
    ->name('team.csvDownload');

// Delete a specific group
Route::delete('group/destroy/{group}',[GroupController::class,
    'deleteGroup'])
    ->name('group.destroy');

// resources
Route::resource('team', TeamController::class);
Route::resource('team.members', MemberController::class);

// Authentication
Route::get('login', [AuthController::class, 'login'])
    ->name('auth.login');

Route::post('login', [AuthController::class, 'applyLogin'])
    ->name('auth.applyLogin');
```

```
Route::get('logout', [AuthController::class, 'logout'])
    ->name('auth.logout');

// About page
Route::get('about', function(){
    return view('about');
})->name('about');
```

4.4.1 La route « / » (Racine)

Cette route a pour seul fonction de rediriger la requête vers la page d'index correspondant aux équipes. C'est là où sera affichée la liste de toutes les équipes ou les équipes appartenant à l'utilisateur authentifié.

4.4.2 Ressources

Des routes de type « resource » ont été mises en place. Les ressources en Laravel permettent de créer automatiquement les routes nécessaires à la mise en place des opérations CRUD sur un modèle Eloquent. Cela permet d'éviter d'avoir à créer un nombre important de routes qui se répètent pour chaque modèle. Les routes de ressources s'appliquent pour les **équipes** et les **membres**.

Des routes ont été ajoutées pour le contrôleur des équipes car des fonctions supplémentaires ont été rajoutées qui ne s'incluent pas dans le cadre des opérations CRUD de la ressource.

La table des **membres** s'appelle **team.member** en raison de la contrainte qui lie un membre à son équipe. On parle de ressource imbriquée.¹⁶

4.4.3 Authentification

La partie authentification contient trois routes : la première pour appeler la méthode qui affiche le formulaire de connexion, la deuxième qui redirige la requête vers la méthode de validation une fois le formulaire validé et la troisième qui redirige sur la méthode qui permet de déconnecter l'utilisateur en regénérant une nouvelle session.

¹⁶ Ressources imbriquées : <https://laravel.com/docs/11.x/controllers#restful-nested-resources>

4.5 Mise en place des vues

4.5.1 Layout

Une bonne pratique mentionnée également par Piotr Jura¹⁷, formateur sur la plateforme Udemy, est l'utilisation d'un fichier Blade contenant la structure HTML de base avec des éléments visuels redondants, comme une barre de navigation ou un footer. Afin de pouvoir ajouter du contenu à ce fichier, il faut définir les sections dans lesquelles il est possible d'ajouter du code. Cela est possible avec la directive Blade `@yield('nom_de_la_section')`.

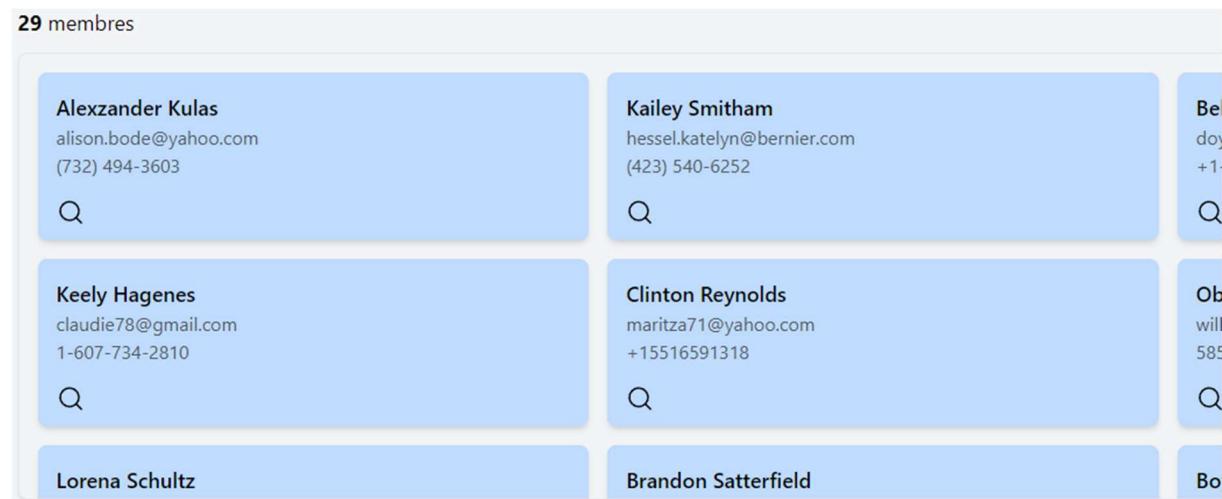
Ce layout peut ensuite être appelé depuis n'importe quel fichier à l'aide de la directive `@extends('chemin_du_fichier')`. Puis, pour ajouter du contenu aux sections définies dans le layout, il faudra utiliser la directive `@section('nom_de_la_section')`.

L'utilisation de cette méthode de création des vues rend les fichiers légers, lisibles et peu redondants, ce qui facilitera la maintenance de l'application dans le futur.

Par exemple, si un élément de la barre de navigation venait à devoir changer, il suffirait d'effectuer la modification sur un seul fichier.

4.5.2 Composant réutilisable pour l'affichage des membres

L'affichage des membres lors de l'ajout des membres dans l'équipe et sur la page où sont affichés les détails de l'équipe se fait sous la forme d'une collection de cartes que voici :



14. Affichage de la collection des membres.

Ce composant devant être présent à deux endroits, il a été décidé de déplacer le code associé à l'affichage de la collection des membres dans un seul fichier appelé

¹⁷ Formation Udemy sur Laravel 10, Piotr Jura : <https://www.udemy.com/course/laravel-beginner-fundamentals/>

membersComponent.blade.php qui se situe dans le répertoire **layouts** et qui peut désormais être appelé depuis n'importe où grâce à la directive Blade suivante :

```
@include('layouts.membersComponent')
```

Si l'affichage de la liste des membres venait à devoir être changée, seul le fichier de ce composant devra être modifié.

4.5.3 Utilisation du même formulaire pour la création et la modification

L'application propose trois formulaires pour la gestion des équipes. Pour optimiser le code, les fonctionnalités de création et de modification ont été fusionnées en un seul formulaire par action.

Le mécanisme qui détermine si les fonctionnalités de création ou de modification doivent être affichées repose sur la présence ou l'absence de données existantes.

Ce mécanisme est réparti sur trois principaux éléments du formulaire :

1. L'action du formulaire, c'est-à-dire la route à emprunter une fois le formulaire soumis.

```
<form action="{{ isset($team) ? route('team.update', ['team' => $team]) : route('team.store') }}" method="POST">
```

Dans cet exemple, selon si la variable **\$team** existe ou non, ce sera la route liée à la modification ou à un nouvel enregistrement dans la base de données qui sera utilisée.

2. La méthode de la requête à utiliser lorsque le formulaire soumis (POST, PUT).

```
@isset($team)
    @method('PUT')
@endisset
```

La méthode par défaut du formulaire est POST mais avec Laravel il est possible d'utiliser d'autres types de méthodes. Dans cet exemple, si la variable **\$team** existe, ce sera alors la méthode PUT qui sera utilisée au moment de la soumission du formulaire.

3. L'affichage ou non des données déjà existantes.

```
<input type="text" ... value="{{ $team->name ?? old('name') }}" class="...">
```

L'existence de la propriété `$team->name` est vérifiée. Si elle existe, on utilise la fonction `old()` qui permet de récupérer automatiquement l'ancienne valeur présente dans la requête précédente ayant été automatiquement enregistrée dans la session¹⁸.

4.5.4 Affichage des groupes

Lorsque l'utilisateur accède à la page d'une équipe, il peut voir une liste de liens représentant chaque activité :

Détails de l'équipe

cumque
Créé le 31-05-2024

[Activité 1](#)

[Activité 2](#)

[Activité 3](#)

[Activité 4](#)

[Activité 5](#)

15. Affichage de la liste des activités

Une fois un de ces liens cliqué, les groupes appartenant à l'activité sont affichés avec un code QR qui redirige vers la même page :

¹⁸ Récupération d'anciennes valeurs : <https://laravel.com/docs/11.x/requests#retrieving-old-input>

[<-Retour](#)



Activité 1

1	2
- Mertz Retta - Jacobs Claud - Kulas Braulio - Rutherford Meredith	- Spinka Jerome - Schulist Mary - Harvey Elias - Kessler Lilly

16. Affichage des groupes participant à une activité.

La répartition des cartes contenant les membres d'un groupe se fait dans une grille dont le nombre de colonnes change en fonction de la largeur de l'écran afin de respecter les contraintes de responsivité de l'application.

4.5.5 Difficultés rencontrées

L'affichage des groupes a présenté une difficulté particulière en raison du fait que l'affichage se fait dynamiquement en fonction de plusieurs éléments que sont les membres, les groupes et la génération à laquelle appartient un groupe. Cela a entraîné la mise en place relativement complexe de l'utilisation de d'itérations et de conditions avec les directives Blade.

Il a fallu notamment suivre la valeur actuelle de la génération pour afficher les bons groupes ce qui permet de dynamiquement fermer des **div** en fonction de la valeur en cours de la génération.

Lorsque les données des groupes sont envoyées à la vue, elles sont triées en fonction de leur numéro de génération.

4.6 Formulaires de création d'une équipe avec les membres

La création des équipes avec les membres et les groupes a été séparée en trois formulaires :

1. Entrée du nom de l'équipe qui permet d'enregistrer une nouvelle équipe dans la base de données
2. Ajout des membres.
3. Entrée des informations concernant la répartition des membres dans les différents groupes.

La raison de ce choix vient du fait qu'un groupe ne peut pas être créé sans membres et les membres ne peuvent être créés sans équipe existante. De plus, ce choix permet une meilleure modularité car l'utilisateur peut choisir de créer une équipe sans terminer la configuration.

Voici les formulaires sous différents formats par lesquels passera l'utilisateur s'il souhaite créer une équipe complète :



Créer une nouvelle équipe

Équipe 1

Ce champ est obligatoire

Suivant

19

17. Formulaire d'ajout du nom lors de la création d'une équipe en version mobile.

¹⁹ Ici les messages d'erreurs ont été volontairement affiché à des fins de démonstration.

Ajouter les membres

Prénom

Nom

Email

Numéro de téléphone

+ Ajouter un membre Suivant →

[Télécharger un exemple de fichier CSV](#)

Importer depuis un fichier CSV

Sélectionner un fichier CSV

Soumettre le fichier

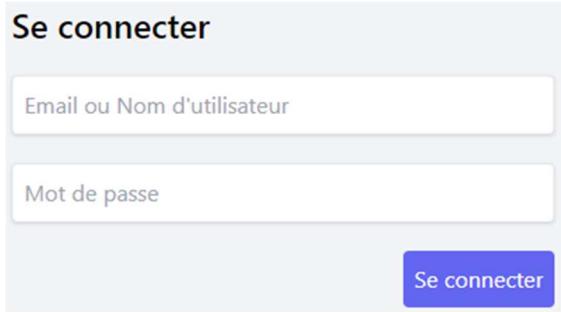
3 membres

Josh Doe
18. Démonstration du formulaire d'ajout des membres.

4.7 Authentification

4.7.1 Formulaire d'authentification

Le formulaire d'authentification est très simple. Il comporte un champ pour l'entrée du nom d'utilisateur ou de son email et un second champ qui est destiné à l'entrée du mot de passe.



19. Formulaire d'authentification

4.7.2 Traitement de l'authentification

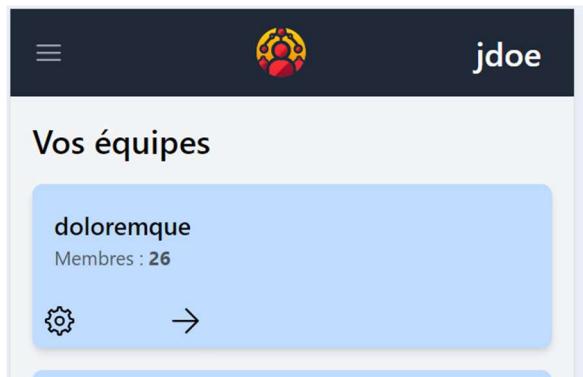
L'authentification des utilisateurs est gérée depuis le contrôleur **AuthController** et doit pouvoir accepter à la fois les noms d'utilisateurs et les emails. Pour cela, il faut trouver dans la base de données une correspondance entre ce que l'utilisateur a entré et la base de données.

```
$user = User::where('email', $login)
    ->orWhere('username', $login)->first();
```

Une fois l'utilisateur trouvé, il faut tenter de l'authentifier avec l'email puis le nom d'utilisateur séparément ce qui donne cela :

```
// Attempt login with the email, then with the username
if(Auth::attempt(['email' => $user->email,
                  'password' => $request->input('password')]) ||
   Auth::attempt(['username' => $user->username,
                  'password' => $request->input('password')]))
```

Une fois l'utilisateur connecté, son nom d'utilisateur est affiché en haut à droite de son écran :



20. Affichage du nom d'utilisateur.

4.7.3 Déconnexion

Pour se déconnecter, l'utilisateur doit cliquer sur son nom d'utilisateur. De là s'affichera un menu ou l'option « Se déconnecter » apparaîtra :



21. Affichage du bouton "Se déconnecter".

4.8 Droits des utilisateurs

Laravel possède plusieurs outils permettant la mise en place de règles permettant d'autoriser des actions²⁰. Ces autorisations permettent de définir si un utilisateur authentifié a le droit d'effectuer une action, comme modifier un modèle Eloquent spécifique. Une des manières de mettre en place ces autorisations est l'utilisation des Policies*.

4.8.1 Création d'une Policy

Afin de créer une Policy²¹ sur un modèle en particulier, voici la commande Artisan à utiliser (ici il s'agit d'une Policy s'appliquant aux équipes) :

```
$ php artisan make:policy TeamPolicy --model=Team
```

Après cela, un nouveau fichier sera créé sous **app/Policies** contenant déjà des fonctions prédéfinies. Ces fonctions servent à retourner un booléen après avoir vérifié si l'utilisateur spécifié a le droit d'effectuer une action sur la ressource.

²⁰ Autorisations avec Laravel : <https://laravel.com/docs/11.x/authorization#creating-policies>

²¹ Crée des Policies : <https://laravel.com/docs/11.x/authorization#creating-policies>

Pour les équipes, il faut vérifier les droits sur la modification et la suppression. Voici comment cela a été mis en place :

```
/**  
 * Determine whether the user can update the model.  
 */  
public function update(User $user, Team $team): bool  
{  
    return $user->id == $team->user_id || $user->admin;  
}  
  
/**  
 * Determine whether the user can delete the model.  
 */  
public function delete(User $user, Team $team): bool  
{  
    return $this->update($user, $team);  
}
```

Dans le cas de **delete**, on fait appel à la fonction **update** car la règle de vérification est identique.

En ce qui concerne les autres fonctions qui ont été créées automatiquement, elles ont été retirées car elles n'ont pas d'utilité dans l'état actuel de l'application.

4.8.2 Utilisation des Policies dans les vues

Afin de déterminer s'il faut afficher certains éléments ou non il est possible d'accéder aux autorisations avec la directive Blade **can()**.

Voici comment cela a été mis en place pour déterminer l'affichage du bouton permettant de supprimer un utilisateur :

```
@can('delete', $team)  
    <form action="{{ route('team.destroy', ['team' => $team]) }}"  
method="POST" class="...">  
        @csrf  
        @method('DELETE')  
        <label>  
            <input type="submit" class=" hidden">  
        </label>  
        <svg ...>  
    </form>  
@endcan
```

4.8.3 Utilisation des Policies dans les contrôleurs

Dans le cas où un utilisateur non-autorisé parvenait à envoyer une requête afin de modifier ou supprimer une ressource, une vérification supplémentaire a été ajoutée dans les fonctions permettant la modification et la suppression. Voici une démonstration avec le cas de la modification²² :

```
public function update(TeamRequest $request, Team $team)
{
    if($request->user()->cannot('update', $team)){
        return abort(403);
    }
    ...
}
```

Le fonctionnement ressemble à celui utilisé dans la directive Blade, il faut récupérer l'utilisateur qui a émis la requête et vérifier s'il ne possède pas le droit de réaliser une certaine action. Dans le cas où il ne possède pas le droit, il faut renvoyer une erreur 403 qui correspond à une interdiction d'accès.

4.9 Algorithme de répartition des membres

Dans l'état actuel, l'algorithme n'est pas entièrement fonctionnel mais permet malgré tout de générer des groupes selon les valeurs demandées.

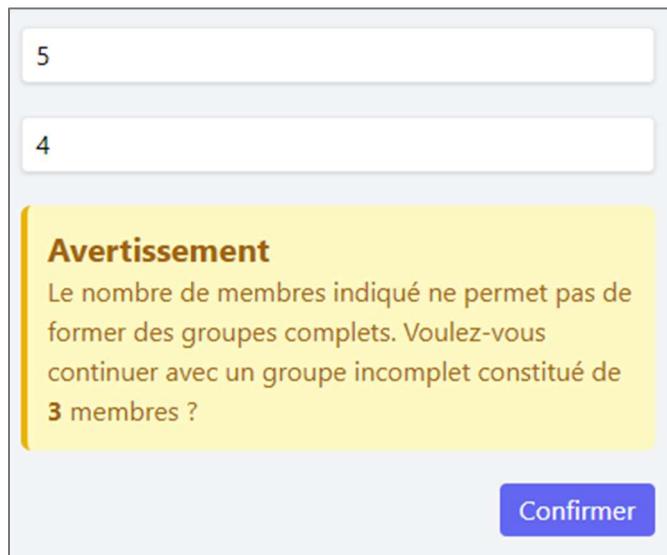
4.9.1 Mise en place

L'algorithme actuel permet de répartir aléatoirement les membres dans des groupes, qui sont eux même répartis en différentes générations (activités). Il n'y a donc pas de vérification qui permette de s'assurer que deux membres n'ont pas déjà été dans le même groupe.

Le fonctionnement de l'algorithme repose sur deux valeurs : le nombre d'activités et la taille idéale des groupes, c'est-à-dire le nombre de membres pour un groupe voulu par l'utilisateur.

Si les valeurs entrées par l'utilisateur entraînent la création d'un groupe incomplet, un message d'avertissement est affiché et l'utilisateur doit confirmer son choix :

²² Provient de la documentation officielle : <https://laravel.com/docs/11.x/authorization#via-the-user-model>



22. Avertissement lors de la génération des groupes.

Voici le code responsable de cette vérification :

```
if($totalMembers % $membersPerGroup != 0 && !$request->has('confirm')){  
    return redirect()->back()->withInput($request->input())  
        ->with(  
            [  
                'confirm' => true,  
                'notCompleteGroupeSize' => $totalMembers %  
$membersPerGroup,  
            ]);  
}
```

Le principe est de vérifier qu'il ne reste aucun membre supplémentaire après la division du nombre total de membres par le nombre de membres voulu pour un groupe (**totalMembers / membersPerGroup**).

Ensuite, le code va itérer autant de fois que le nombre d'activités voulues par l'utilisateur. À chaque itération, un tableau de groupes correspondant à une génération est créé et les membres sont mélangés dans un ordre aléatoire.

À la suite de cela, les membres seront distribués dans leurs groupes :

```
// Iterate through each generation
for($i = 0; $i < $totalGenerations; $i++){
{
    $group = []; // Contains one group at a time
    $generationGroups = []; // Contains the groups of one generation

    // Shuffle the members in a random order
    shuffle($membersId);

    // Cycle through each member
    foreach($membersId as $member){

        // Add the member to the group
        $group[] = $member;

        // Check if the number of stored members isn't higher than the
defined group size
        if(count($group) >= $membersPerGroup){

            // Add the created group to the generation
            $generationGroups[] = $group;

            // Reset the array
            $group = [];
        }
    }
}
```

La création d'un groupe incomplet est également prise en compte :

```
// If the data doesn't allow to store the inputed number
// of members for the last groups, then store the incomplete
//group in the generation.
if($totalMembers % $membersPerGroup != 0){
    $generationGroups[] = $group;
}
```

Puis, la génération est stockée dans un tableau:

```
// Store the generation
$previousAssignments[] = $generationGroups;
```

4.9.2 Pistes d'améliorations

Afin de rendre l'algorithme entièrement fonctionnel, il faudrait implémenter un moyen permettant de vérifier que des membres n'ont pas été précédemment dans le même groupe, et si tel est le cas, réassigner de la manière la plus efficace les membres dans de nouveaux groupes.

Un autre moyen d'améliorer le code serait de le refactoriser afin de séparer les différentes parties de l'algorithme dans plusieurs fonctions ce qui améliorera la maintenabilité du code.

4.10 Mise en place du QR Code

Afin de pouvoir facilement créer un accès à une équipe, un QR Code est généré pour chaque équipe. La génération du QR Code se fait à l'aide du projet « simple-qrcode »²³.

4.10.1 Installation

Cet outil s'intègre directement à Laravel avec la commande composer suivante

```
$ composer require simplesoftwareio/simple-qrcode "~4"
```

4.10.2 Intégration de la génération de QR codes

La génération du QR code se déroule dans la méthode du contrôleur responsable de l'affichage d'une équipe spécifique (**show**).

Voici la ligne de code responsable de cette fonctionnalité :

```
$qrCode = QrCode::size(100)  
    ->generate(route('team.show', ['team' => $team]));
```

Le QR code est généré selon la route qui redirige vers une équipe spécifique. Le QR Code est créé sous forme de SVG qui est stocké dans la variable **\$qrCode** puis envoyé à la vue correspondant à l'équipe.

Une fois dans la vue, l'image est simplement appelée depuis la variable qui la contient :

```
{{ $qrcode }}
```

²³ Projet simple-qrcode : <https://github.com/SimpleSoftwareIO/simple-qrcode>

Voici le résultat une fois la page chargée :



23. Affichage du QR code sur la page d'une équipe

4.10.3 Remarque

La mise en place du QR Code a été faite en suivant un tutoriel²⁴.

Dans la globalité, la mise en place de cette fonctionnalité était plus simple et rapide qu'attendu. Aucune difficulté n'a été rencontrée.

4.11 Importation depuis un fichier CSV

Afin de faciliter l'importation des membres, il a fallu ajouter une fonctionnalité permettant d'importer les membres depuis un fichier CSV. La mise en place de cette fonctionnalité a été inspirée d'un article écrit par Kanishka Perera²⁵

4.11.1 Mise en place

La création des membres depuis un fichier CSV est identique à l'autre méthode consistant à entrer manuellement les données à la seule différence que les données proviennent d'un fichier sur lequel il faut extraire les informations présentes sur une ligne, les vérifier et réitérer sur les suivantes.

²⁴ Tutoriel pour la mise en place d'un QR Code : <https://www.akilischool.com/cours/laravel-generer-un-qr-code-avec-simple-qrcode>

²⁵ Opérations avec les fichiers CSV : <https://medium.com/@kanishkanwanperera/building-a-csv-import-export-feature-in-laravel-10-step-by-step-guide-with-no-external-packages-6b73295aa0c1>

L'utilisateur peut importer son fichier depuis la page de création des membres :

Télécharger un exemple de fichier CSV
Importer depuis un fichier CSV
Sélectionner un fichier CSV
Soumettre le fichier

24. Formulaire d'importation d'un fichier CSV.

4.11.2 Téléchargement d'un fichier d'exemple

Un lien de téléchargement est présent sur le formulaire qui permet de télécharger un fichier CSV servant d'exemple. Afin de mettre en place ce système de téléchargement, plusieurs étapes ont été nécessaires.

Dans un premier lieu il a fallu exécuter la commande suivante :

```
$ php artisan storage:link
```

Cette commande permet de créer un lien symbolique depuis le répertoire **public/storage** vers **storage/app/public**.²⁶

Une fois cela fait, un fichier d'exemple a été créé :

```
firstname,lastname,email,phone_number  
John,Doe,jdoe@example.com,9999999999
```

Celui-ci a été placé dans le répertoire **storage/app/public**.

Pour terminer, la méthode permettant de déclencher le téléchargement a été créée dans le contrôleur des équipes :

```
public function csvDownload(){  
    return Storage::disk('public')->download('CSV-Exemple.csv');  
}
```

4.11.3 Difficulté rencontrée

Lors de l'importation du fichier CSV, une erreur apparaissait en raison du fait que le fichier CSV était créé avec Excel et que ce dernier utilise le point-virgule (;) comme séparateur au lieu de la virgule (,). Le fichier échouait donc la vérification du type de fichier après l'importation. Une fois le caractère de séparation corrigé, l'importation s'est faite sans erreurs.

²⁶ Crédit à la documentation officielle de Laravel pour l'explication de cette étape : <https://laravel.com/docs/11.x/filesystem#the-public-disk>

4.12 Procédure utilisée pour le déploiement

4.12.1 Constraintes

N'ayant pas d'accès SSH au serveur, il va être impossible d'exécuter des commandes artisan liées à l'optimisation, notamment la mise en cache de certains éléments comme les routes.

Il existe cependant une solution qui consiste à mettre en place des routes permettant d'exécuter des commandes Artisan en accédant à une certaine URL. Ces routes ne devront donc pas rester indéfiniment car elles présentent un risque de sécurité évident, elles seront présentes sur la version de production uniquement durant le temps du déploiement et à des fins de tests.

4.12.2 Procédure

Afin de ne pas endommager la version de développement du projet, une copie du répertoire de l'application est effectuée afin de réaliser les manipulations nécessaires à la mise en production.

4.12.2.1 Modification du fichier d'environnement (.env)

L'affichage ou non des erreurs et les informations de connexion à la base de données sont définies par les valeurs définies dans les variables présentes dans le fichier .env.

Afin de pourvoir mettre l'application en production, il faut donc modifier plusieurs valeurs, en voici la liste :

- **APP_ENV=production**
- **APP_DEBUG=false** (ou true temporairement afin de pouvoir facilement débugger en cas de besoin)
- **APP_URL=[URL réel du site]**
- **DB_CONNECTION=[Type de connexion pour la base de données, pour ce projet, ce sera MySQL]**
- **DB_HOST=[Adresse IP du serveur de base de données]**
- **DB_PORT=[Port permettant d'accéder à la base de données, généralement 3306]**
- **DB_DATABASE=[nom de la base de données]**
- **DB_USERNAME & DB_PASSWORD=[valeurs d'authentification à la base de données].**

4.12.2.2 Suppression des dépendances liées au développement

Afin d'éviter d'importer des dépendances inutiles en production, il faut veiller à les supprimer correctement. Pour cela, il faut lancer la commande suivante :

```
$ composer install --no-dev
```

Cette commande permet d'installer uniquement les dépendances nécessaires à la production²⁷. On ne retrouvera donc pas les outils comme les Factories.

4.12.2.3 Compiler le CSS

Afin de compiler le CSS utilisé avec Tailwind dans un seul fichier, il faut utiliser la commande suivante :

```
$ npm run build
```

Cela permet de créer dans le dossier **public**, un répertoire nommé **build** dans lequel se trouve un fichier .css et .js.

4.12.2.4 Préparation de la base de données

Maintenant que le fichier d'environnement est configuré avec les informations de connexion à la base de données de production, il est possible d'exécuter une migration sur la base de données avec artisan depuis l'environnement local. Il suffit pour cela d'exécuter la commande suivante :

```
$ php artisan migrate
```

4.12.2.5 Éléments à vérifier avant de transférer les fichiers sur le serveur.

Il est important de vérifier deux points importants concernant la configuration du serveur afin de s'assurer du bon fonctionnement de l'application :

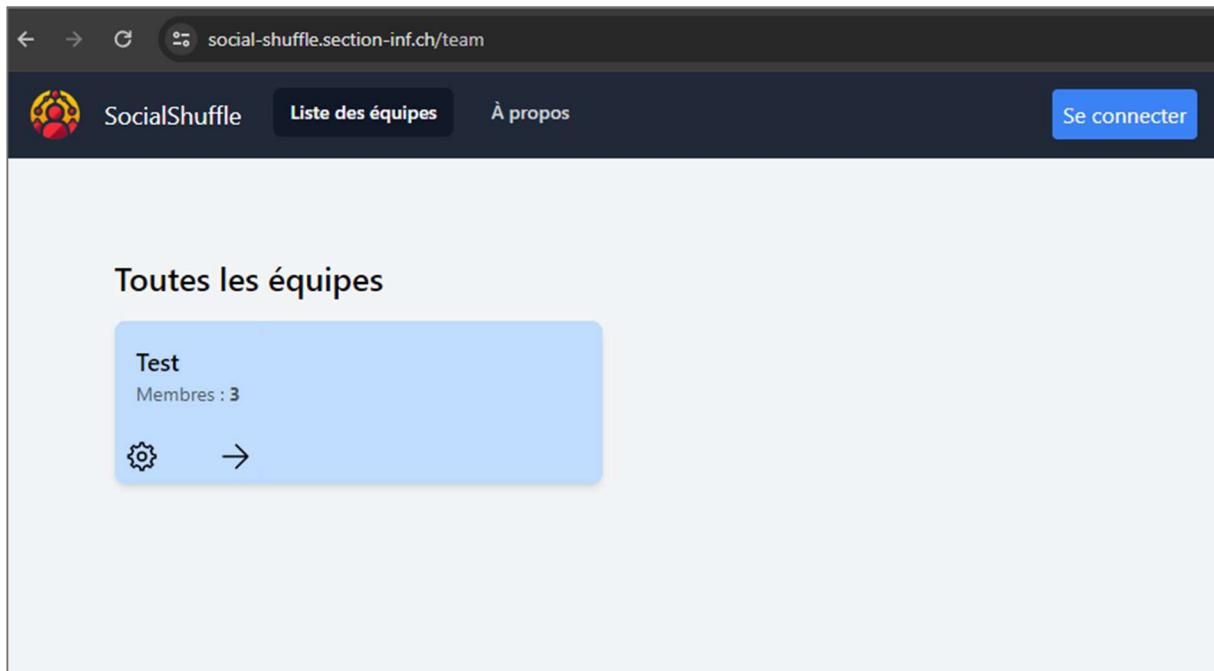
1. La bonne version de PHP est installée (pour la version 11 de Laravel, il faut au minimum la version 8.2 de PHP).
2. Le serveur effectue une redirection sur le dossier **public** de Laravel car le fichier **index.php** du site s'y trouve. C'est le point d'entrée de l'application.

²⁷ Mise en production d'un projet Laravel sans accès SSH :

<https://laracasts.com/discuss/channels/laravel/installing-laravel-on-server-without-ssh>

4.12.2.6 Transfer des fichiers

Avec un accès FTP au serveur il va être possible de transférer le projet une fois prêt pour la production. Pour cela il suffit d'effectuer un copier-coller de tous les fichiers de l'application à la racine du site. Il faut faire attention au temp de transfert qui peut parfois être long. Pour cette application, le transfert a pris environ 20 minutes. Si le transfert s'est fait sans erreurs, le site devrait être accessible :



25. Démonstration du site une fois déployé

4.12.3 Exécuter des commandes Artisan depuis une URL

Comme mentionné précédemment, il est possible d'exécuter des commandes Artisan en accédant à une URL spécifique. Voici comment cela a été mis en place pour cette application :

```
Route::get('cache', function(){
    Artisan::call('optimize:clear');
    Artisan::call('optimize');
});

Route::get('routes', function(){
    Artisan::call('route:clear');
    Artisan::call('route:cache');
});
```

Pour des raisons de sécurité évidentes, ces routes devront être supprimées une fois qu'elles ne sont plus utilisées afin d'éviter que n'importe quel utilisateur puisse manipuler l'application.

4.13 Trois mesures de sécurité

4.13.1 Token CSRF

Les attaques CSRF permettent l'exécution de requêtes au travers d'utilisateurs enregistrés ayant souvent des droits élevés sur les sites. Pour contrer cela, Laravel met à disposition une manière simple d'implémenter la protection CSRF. Pour chaque session d'utilisateurs, un jeton (token) CSRF est automatiquement généré. Ce jeton permet de s'assurer qu'il s'agit de l'utilisateur authentifié est bien la personne à l'origine de la requête.

La mise en place de ce jeton se fait à chaque formulaire HTML de type POST, PUT, PATCH ou DELETE. Il est ensuite nécessaire d'ajouter la directive Blade `@csrf` au tout début du formulaire sous peine d'avoir une erreur. Le Framework oblige donc les développeurs à protéger les formulaires de leurs applications contre ce type d'attaques.

```
<form action="{{ ... }}" method="POST">
    @csrf
    ...
</form>
```

4.13.2 Validation des formulaires

La validation des formulaires se fait par la validation de règles définies pour une requête particulière. Une fois le formulaire soumis, les entrées de l'utilisateur seront vérifiées pour s'assurer qu'elles respectent les règles définies.

Voici un exemple avec les règles liées à la création des membres :

```
class MemberRequest extends FormRequest
{
    ...
    public function rules(): array
    {
        return [
            'firstname' => 'required|string',
            'lastname'  => 'required|string',
            'email'     => 'required|email',
            'phoneNumber' => 'required|min:9|numeric',
        ];
    }
}
```

Le déclenchement de cette vérification se fait dans la méthode du contrôleur correspondant à la validation.

Dans ce cas il s'agit de la création et la modification d'un membre. La validation est appelée de la manière suivante :

```
/*
 * Store a newly created resource in storage.
 */
public function store(MemberRequest $request, Team $team)
{
    // Validate the request containing the new member data
    $data = $request->validated();
    ...
    return ...;
}
```

Si l'utilisateur a entré une information erronée, cela déclenchera une erreur. Selon la règle qui a été enfreinte, le message d'erreur sera différent. Ces messages se trouvent dans le répertoire **/lang** qui est accessible après avoir entré la commande suivante :

```
$ php artisan lang:publish28
```

Il est ensuite possible de définir plusieurs langues pour le site en copiant le dossier en anglais par défaut et en traduisant les messages. La langue active est celle mentionnée dans le fichier .env.

```
/lang
  /en
    messages.php
  /fr
    messages.php
```

Pour afficher les messages d'erreur dans les vues, il faut ajouter la directive Blade **@error** qui est automatiquement déclenchée quand une erreur est retournée à la vue. Il est ensuite possible d'afficher le message d'erreur avec la variable **\$message**. En ce qui concerne la récupération de la valeur précédente, elle se fait avec l'utilisation de la fonction **old()** après avoir vérifié que la ressource a été envoyée à la vue:

```
<input type="text" name="name" id="name" placeholder="Nom de l'équipe"
       value="{{ $team->name ?? old('name') }}>
@error('name')
  <p class="text-red-500">{{ $message }}</p>
@enderror
```

²⁸ Mise en place de plusieurs langues dans un projet Laravel :
<https://laravel.com/docs/11.x/localization>

4.13.3 Hachage des mots de passe

Bien que l'application ne permet pas aux utilisateurs de créer leurs comptes, des comptes par défaut ont été créés grâce aux Factories. Afin de garantir la sécurité de ces comptes, les mots de passe ont été hachés.

Laravel met à disposition la classe **Hash** permettant de gérer tout ce qui est lié au hachage. Cette classe utilise la librairie Bcrypt bien qu'il soit possible de la configurer afin d'utiliser d'autres drivers tels que **argon** ou **argon2id**.²⁹

La mise en place du hachage lors de la création des comptes a été faite de la manière suivante :

```
'password' => Hash::make('exempleDeMotDePasse'),
```

4.14 Trois mesures pour le « Responsive design »

Afin de répondre aux exigences de responsivité de l'application, le Framework Tailwind a été utilisé pour la mise en page de l'application. Ce Framework permet de mettre en place un design responsive très rapidement avec peu de code.

4.14.1 Affichage des cartes en fonction de la largeur de l'écran

Un bon exemple pour démontrer la mise en place d'un design responsive avec Tailwind est la manière dont les cartes contenant les équipes enregistrées sont disposées dans une grille dont le nombre de colonnes change en fonction de la largeur de la page :

```
<div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-4">
```

La classe par défaut **grid-cols-1** est destinée aux écrans mobiles, elle permet d'afficher une seule carte par ligne. Les deux classes restantes sont destinées aux écrans plus larges.

²⁹ Hachage avec Laravel : <https://laravel.com/docs/11.x/hashing>

Voici la mise en pratique de ce principe sur la version Desktop et mobile :

The screenshot shows a desktop browser window with the URL 127.0.0.1:8000/team. The page title is "SocialShuffle". The navigation bar includes links for "Liste des équipes", "À propos", and "+ Nouvelle équipe". A user profile "jdoe" is visible on the right. The main content area is titled "Vos équipes" and lists five teams: "doloremque" (26 members), "et" (37 members), "possimus" (37 members), "minima" (39 members), and "molestiae" (32 members). Each team card features a gear icon and a right-pointing arrow.

26. Affichage de l'écran principal en version Desktop.

The screenshot shows a mobile device screen with a resolution of 412 x 915 at 125% zoom, set to "No throttling". The top navigation bar includes a menu icon, the "SocialShuffle" logo, and the user profile "jdoe". Below the navigation bar, the "Vos équipes" section displays the same five teams as the desktop version: "doloremque" (26 members), "et" (37 members), "possimus" (37 members), "minima" (39 members), and "molestiae" (32 members). Each team card includes a gear icon and a right-pointing arrow.

27. Version mobile de l'écran principal

4.14.2 Barre de navigation

En raison de la largeur réduite de l'écran d'un smartphone, il était difficile de placer tous les éléments de la barre de navigation sur l'écran.

Pour remédier à cela, un Template de barre de navigation responsive fourni sur une plateforme de Tailwind³⁰ a été utilisé. Ce Template permet d'afficher une barre de navigation au-dessus de l'écran pour la version Desktop et dans un menu séparé dans la version mobile.

Voici une comparaison entre les deux versions de la barre de navigation :



28. Barre de navigation Desktop



29. Barre de navigation mobile

4.14.3 Boutons de gestion des membres

Sur la page permettant d'afficher les groupes et les membres, deux boutons permettent respectivement de Générer des groupes et d'ajouter des membres supplémentaires à l'équipe.

Sur un écran large, ces boutons sont disposés sur la même ligne et sur les écrans mobiles, ils sont placés l'un au-dessus de l'autre :



30. Boutons de gestion des membres en version Desktop.

³⁰ Template de barre de navigation : <https://tailwindui.com/components/application-ui/navigation/navbars>

Gestion des membres

Générer les groupes

+Ajouter des membres

31. Boutons de gestion des membres au format mobile.

4.15 Mise en place des tests Laravel Dusk dans GitHub Actions

4.15.1 Installation de Laravel Dusk³¹

L'installation de Laravel Dusk se fait uniquement dans un environnement de développement.

La commande suivante permet d'installer Laravel Dusk dans le projet.

```
$ composer require laravel/dusk --dev
```

Laravel Dusk étant un outil de test End-To-End, il doit pouvoir accéder à un navigateur Web. Pour cela, il faut installer le pilote Chrome (Chrome Driver) avec la commande suivante :

```
$ php artisan dusk:install
```

Désormais, un nouveau répertoire nommé **Browser** a été créé et contient un exemple de test simple qui permet de visiter la racine du site et vérifier qu'un certain texte est présent sur la page.

Voici le code du test légèrement modifié :

```
public function testBasicExample(): void
{
    $this->browse(function (Browser $browser) {
        $browser->visit('/')
            ->assertSee('SocialShuffle');
    });
}
```

Ce test est lancé afin de s'assurer de la bonne installation de Laravel Dusk :

```
$ php artisan dusk
```

On peut ensuite constater sur le terminal que le test s'est exécuté sans problèmes :

³¹ Installation de Laravel Dusk : <https://laravel.com/docs/11.x/dusk#installation>

Avant de lancer le test, il faut s'assurer que l'adresse du site (APP_URL) dans le fichier d'environnement, soit sous forme de IP (<http://127.0.0.1:8000>), sinon une erreur sera retournée lors de l'exécution du test.

Une fois la commande exécutée, on peut constater sur le terminal que le test s'est exécuté sans problèmes :

```
PASS Tests\Browser\ExampleTest
✓ basic example
```

4.15.2 Mise en place de GitHub Action

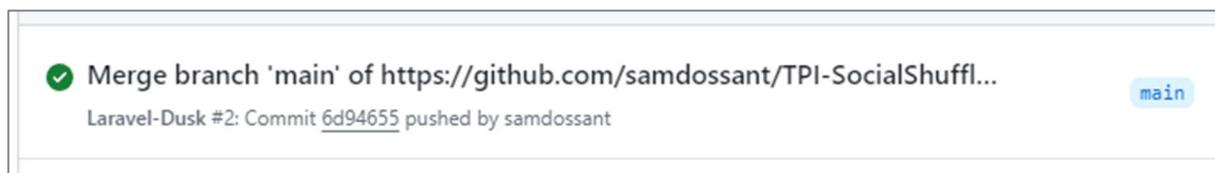
La configuration d'un Workflow GitHub Actions se fait dans sur fichier **.yml** contenant la procédure à suivre pour la mise en place d'un environnement de test virtuel. Laravel fournit un exemple de ce fichier qui peut servir de base pour la configuration de tests³².

Dans le cas de SocialShuffle, les tests seront exécutés à chaque push et se feront dans un Container Ubuntu sur laquelle sera installé un nouvel environnement propre pour l'exécution de l'application.

Le fichier **.yml** utilisé pour ce projet provient d'un autre projet réalisé avant le TPI. Il a fallu cependant adapter certaines valeurs pour qu'elles correspondent avec le projet actuel.

Pour mettre en place le workflow sur GitHub, il faut se rendre dans l'onglet **Actions** depuis lequel on peut créer le fichier **.yml**.

Une fois le fichier ajouté au projet, on peut essayer la mise en place des tests en faisant un push sur le repository, ce qui déclenchera le test configuré précédemment.



32. Test automatisé réussi sur GitHub Actions.

4.15.3 Crédation d'un test Laravel Dusk

Dans cette partie, est détaillée la mise en place du test **CreateTeamTest** qui permet de tester la création d'une équipe au complet.

Pour créer le fichier contenant le test, il faut exécuter la commande suivante :

```
$ php artisan dusk:make testCreateTeam
```

³² Exemple de fichier **.yml** : <https://laravel.com/docs/11.x/dusk#running-tests-on-github-actions>

Une fois terminé, un nouveau fichier est créé dans le dossier **Browser**.

Voici le code en détails comment ce test a été mis en place :

```
class CreateTeamTest extends DuskTestCase
{
    use DatabaseMigrations;
    ...
```

L'utilisation de **use DatabaseMigrations** permet de lancer une migration à chaque nouveau test. Cela permet de s'assurer que chaque test est effectué avec des données propres.

```
public function testCreateTeam(): void
{
    // Arrange
    $user = User::factory()->create([
        'username' => 'jDoe',
        'email' => 'johndoe@gmail.com',
        'password' => bcrypt('0000'),
        'admin' => false,
    ]);
}
```

Avant de lancer les différentes étapes du test, il faut créer un nouvel utilisateur. Afin de simuler au mieux le comportement d'un utilisateur lambda, celui-ci ne privilie pas des droits admins.

```
// Act
$this->browse(function (Browser $browser) use ($user) {
    $browser->visit('/')

    // Login
    ->clickLink('Se connecter')
    ->type('email', $user->username)
    ->type('password', '0000')
    ->click('@login-button')
```

Dans cette partie, Dusk va accéder au navigateur à l'aide du pilote Chrome discuté précédemment. Une fois le site chargé, le lien contenant le texte « Se connecter » sera activé, ce qui va permettre d'être redirigé sur le formulaire de connexion. Sur ce formulaire, les informations de connexion seront entrées et le bouton de connexion sera enclenché.

Ce bouton est marqué l'attribut **dusk** sur la balise HTML responsable de la soumission du formulaire comme montré ci-après :

```
<input type="submit" value="Se connecter" dusk="login-button">
```

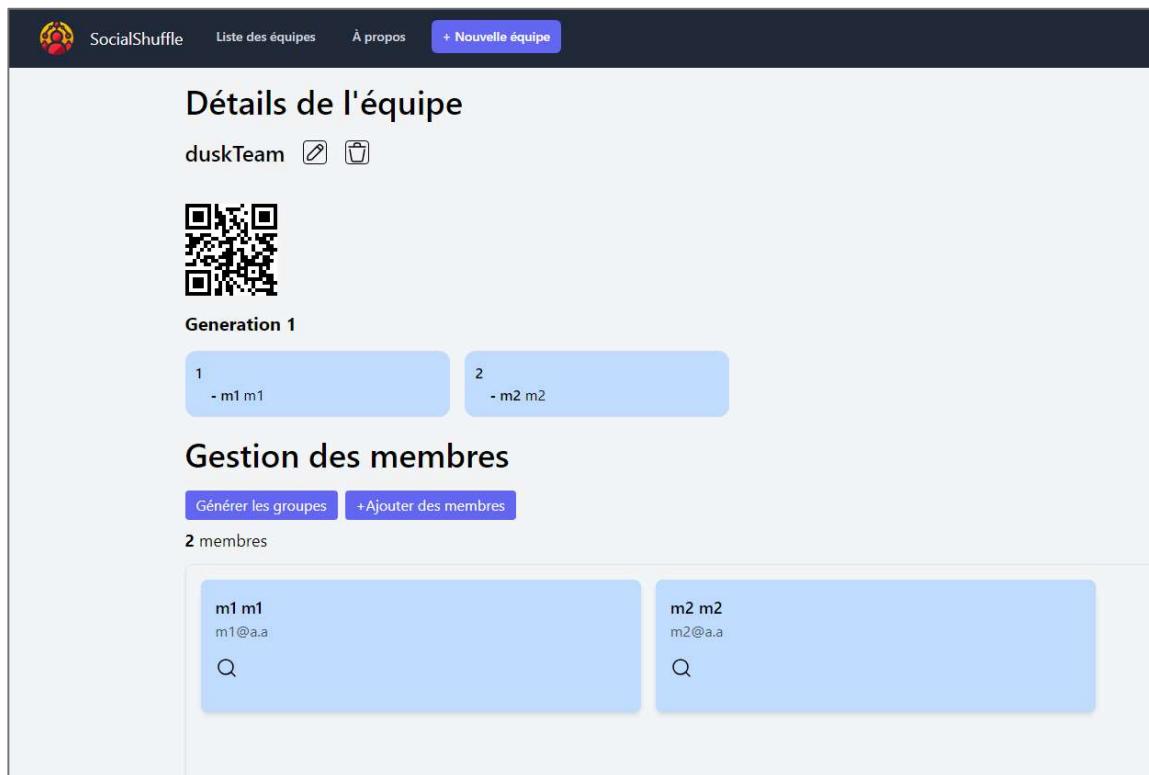
Cet attribut permet de cibler précisément lors des interactions simulées par les tests.

Le reste des étapes de ce test ayant un fonctionnement similaire à ce qui vient d'être vu, elles ne seront pas discutées ici.

```
->screenshot('finish')  
  
// Assert  
->assertPathIs('/team/1');
```

Une fois que toutes les étapes du test ont été réalisées, il faut faire une assertion afin de s'assurer que le résultat du test est celui qui est attendu. Dans le cas de ce test, on s'assure que le lien est celui qui est attendu avec l'**id** correcte de l'équipe, ce qui démontre également l'importance de réaliser une nouvelle migration de la base de données avant chaque nouveau test, car sinon celui-ci serait invalidé.

De plus ici, une capture d'écran est réalisée afin d'obtenir un visuel de la page que voici :



33. Capture d'écran provenant d'un test Dusk

La mise en place du reste des tests étant très similaire et répétitif par rapport à ce qui vient d'être démontré, ChatGPT-4 a été utilisé dans un but de gain de temps.

4.15.4 Risque de sécurité

L'utilisation de l'attribut Dusk dans les fichiers Blade peut présenter des risques, surtout si l'on ne veut pas exposer des informations concernant les tests internes au public.

Une manière potentielle de résoudre ce problème qui n'a pas été mis en place dans ce projet est l'utilisation d'une condition dans le fichier Blade qui vérifier l'état de l'environnement de l'application (donc s'il est en local ou en production).

4.15.5 Difficultés rencontrées

Lors de la mise en place des tests, il est crucial de placer le préfixe « test » dans le nom des tests. Sinon, ils ne seront pas détectés et exécutés par Laravel Dusk. Une erreur avait été commise lors de la mise en place du test présenté précédemment en nommant le test « createTeamTest », ce qui a entraîné des erreurs lors de l'exécution.

Après avoir diagnostiqué le problème, la solution a été trouvée en comparant le nom avec celui du test créé par défaut lors de l'installation de Dusk qui commence par « test ». Une fois que le nom a été corrigé, le test a fonctionné correctement.

5 TESTS

5.1 Dossier des tests

Pour rappel, les tests ont été mis en place avec Dusk, qui permet d'imiter le comportement d'un utilisateur sur l'application. Voici donc les résultats des tests planifiés précédemment :

Nom du test	Description	Validé ?
LoginTest	Test de l'authentification. L'utilisateur entre ses authentifiant dans les champs requis.	Oui
WrongLoginTest	Test de l'authentification avec des informations erronées.	Oui
CreateTeamTest	Test de la création d'une équipe complète avec ses membres et ses groupes.	Oui
CreateWrongMemberTest	Test de la création d'une équipe en insérant des données erronées.	Oui
DeleteTeamTest	Tester la suppression d'une équipe complète, qui contient des membres et des groupes.	Oui.

6 CONCLUSION

6.1 Bilan des fonctionnalités demandées

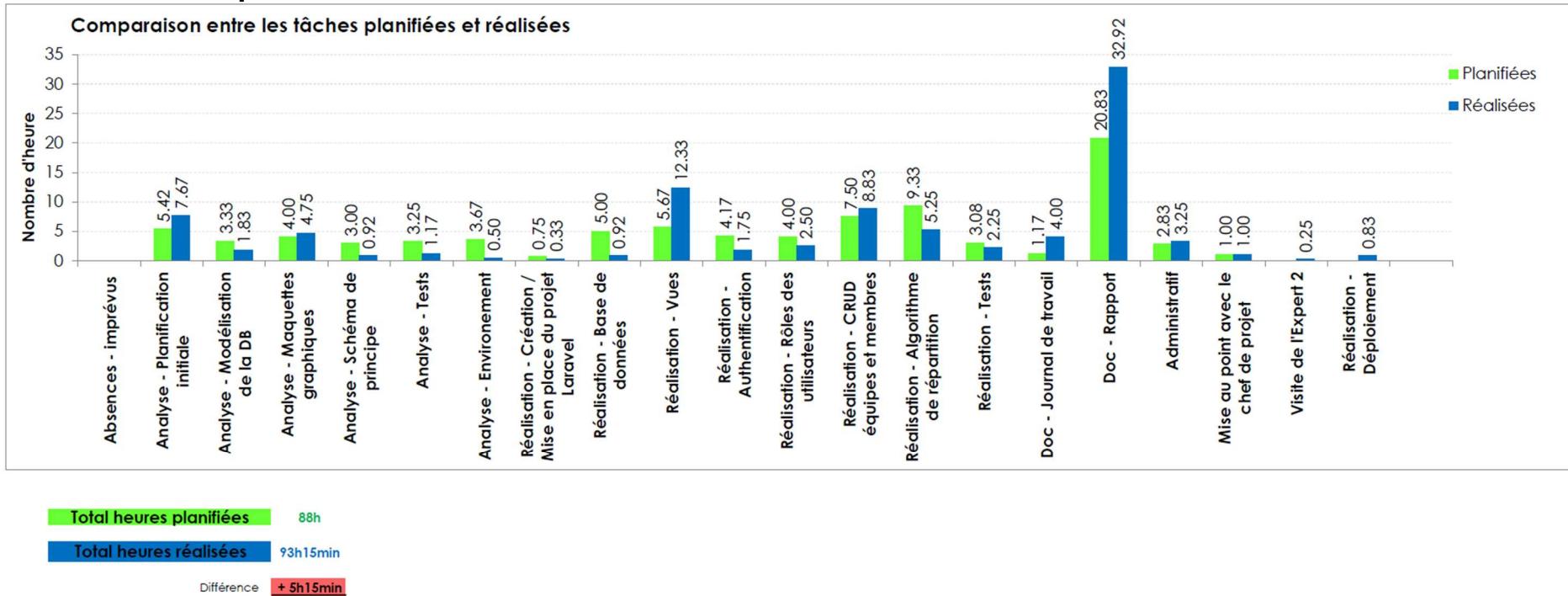
Dans l'ensemble, la plupart des fonctionnalités demandées ont été intégrées avec succès. Voici la liste des éléments entièrement fonctionnels :

1. Les formulaires respectent les bonnes pratiques, ils affichent des messages d'erreurs pertinents, les champs sont remplis automatiquement, sauf pour le champ de mot de passe dans le formulaire de connexion, et ce pour des raisons de sécurité. Chaque champ est également validé avant que les données soient traitées.
2. Le site respecte le design responsif.
3. Les opérations CRUD sur les équipes et les membres ont bien été intégrées avec l'importation des membres avec un fichier CSV.
4. Un rôle d'admin a été intégré à l'aide des Policies. Les utilisateurs admin ont tous les droits sur les ressources.

6.1.1 Génération des groupes

L'algorithme responsable de répartir les membres dans différents groupes n'est pas en mesure de vérifier si des membres ont déjà été dans le même groupe. Les groupes sont malgré tout générés selon les valeurs entrées par l'utilisateur et un message d'avertissement est affiché quand un groupe incomplet est créé.

6.2 Bilan de la planification



Plusieurs tâches contiennent des différences majeures dans leur temps de réalisation par rapport à ce qui avait été prévu dans la planification. Beaucoup de tâches ont été surestimées, comme la réalisation du schéma de principe et l'analyse de l'environnement.

Pour ce qui est de la réalisation, beaucoup de tâches contiennent de grands écarts, comme la mise en place de la base de données qui contient une différence d'environ quatre heures avec ce qui avait été planifié. Ce qui a été la cause d'un tel écart, était probablement dû à un manque de recul sur la manière d'approcher la mise en place de la tâche.

D'autres points de la réalisation ont compte à eux été sous-estimés, comme la mise en place des vues qui a pris le double du temps estimé. La raison pour cela est qu'il a fallu faire plusieurs itérations d'une même vue au fur et à mesure que des nouvelles fonctionnalités étaient ajoutées, ce qui n'avait pas été pris en compte lors de la planification.

Pour la réalisation du projet, du temps supplémentaire a été utilisé en dehors du temps alloué au projet.

6.3 Bilan personnel

Si ce projet était à refaire, une planification plus réaliste serait mise en place afin d'éviter les écarts drastiques observés. La rédaction du rapport ayant également pris beaucoup plus de temps que prévu, une autre approche aurait été utilisée, comme la documentation des fonctionnalités en parallèle de leur implémentation. Cette approche pourrait permettre un gain de temps sur la rédaction qui aurait pu être utilisée sur l'implémentation d'un algorithme fonctionnel en ce qui concerne la génération des groupes.

Ce projet a malgré tout donné la possibilité d'appliquer des connaissances dans un projet concret, dont la mise en place pourra servir d'inspiration pour des futurs projets du même type.

GLOSSAIRE

A

Artisan	
Interface en ligne de commande fourni avec Laravel permettant d'interagir avec différents aspects de l'application.	8
Assignment de masse	
Fait de modifier plusieurs attributs d'un modèle au lieu d'opérer sur chaque attribut.	25

C

ChromeDriver	
Pilote permettant à Laravel Dusk d'interagir avec un navigateur afin de performer des tests.	17
Composer	
Gestionnaire de dépendances écrit en PHP.	19
Contrôleur	
S'inscrit dans l'architecture MVC. Permet de faire le lien entre le modèle et la vue. S'occupe aussi de la plupart des opérations de logique.	9

D

Dusk	
Outil de tests fonctionnant avec Laravel qui permet de simuler des interactions utilisateurs.	17

F

factories	
Outil de Laravel permettant de décrire des données fictives à insérer dans les champs d'une base de données.	16

G

Github Actions	
Outil d'intégration continue (CI/CD) de GitHub.	17

L

Laravel	
Framework PHP basé sur une architecture MVC.	6

M

Migrations	
Outil intégré à Laravel permettant la gestion des bases de données avec du code PHP.	22
Modèle	
S'inscrit dans l'architecture MVC. Il s'agit de la partie du code qui interagit avec la base de données	9
MVC	
Architecture Model - View - Controller	8

O

ORM

(Object Relational Mapping) Interface permettant d'exécuter des requêtes sur la base de données avec du code logique. 10

P

Policies

Outil de Laravel permettant d'organiser des règles d'autorisations sur des modèles Eloquent spécifiques. 37

R

Responsive Design

Adaptation automatique de l'interface selon le type d'appareils utilisés (Smartphone, Desktop). 6

Routes

Permet d'exécuter des instructions en fonction d'une URL et d'un type de requête. 9

S

Seeders

Outil de Laravel permettant de définir comment générer des données fictives, comme le nombre d'occurrences d'un modèle. 17

T

Table de pivot

Table dans une base de données dont l'identifiant est la concaténation de deux clés étrangères. Ce genre de tables permet la mise en pratique d'une relation n-n (many-to-many). 24

Tailwind

Framework CSS. 21

Tests End-to-End

Méthodologie de test permettant de tester l'entièreté d'une fonctionnalité dans des conditions réelles. 18

V

Vue

S'inscrit dans l'architecture MVC. Il s'agit de la partie contenant généralement une interface graphique. 9

7 ANNEXES

7.1 Résumé du rapport de TPI

7.1.1 Situation de départ

Ce projet a pour but de réaliser une application web nommée SocialShuffle qui permet de mélanger des participants afin de favoriser les interactions sociales.

L'application devra respecter les bonnes pratiques concernant les formulaires, intégrer des mesures de sécurité, utiliser le design responsive et respecter les normes de l'ETML en ce qui concerne le nommage dans la modélisation de la base de données.

7.1.2 Mise en œuvre

Pour ce projet, le framework Laravel est utilisé, ce qui permet de bénéficier de sa structure MCV. La première étape a consisté à analyser les contraintes demandées par le cahier des charges, notamment la modélisation de la base de données et le design général des interfaces.

Une fois l'analyse terminée, les différentes parties de l'application (contrôleurs, modèles et vues) ont été intégrées. Chaque fonctionnalité a été implémentée en respectant les bonnes pratiques de développement, comme le respect des normes de codage de Laravel ou bien encore en prenant en compte la réutilisabilité du code. Des tests End-To-End, couplés à l'intégration continue sur GitHub Actions, ont été mis en place afin de garantir le bon fonctionnement des fonctionnalités.

7.1.3 Résultats

Pour terminer, l'application est désormais en mesure de gérer l'authentification des utilisateurs avec la possibilité d'avoir des utilisateurs avec des droits administrateurs. Il est en outre possible de créer des équipes, ajouter des membres manuellement ou avec un fichier CSV et générer aléatoirement des groupes selon un nombre d'activités donné, ce qui ne correspond pas exactement aux prescriptions du cahier des charges.

Des tests End-To-End ont aussi été mis en place et automatisés avec GitHub Actions.

En plus de ces fonctionnalités techniques, ce rapport de projet est également fourni.

7.2 Accès au repository GitHub

Lien : <https://github.com/samdoossant/TPI-SocialShuffle-Dos-Santos>

Le code source de l'application se trouve dans le répertoire « SocialShuffle ».

7.3 Bibliographie

- Ahadi, Wilo. "Laravel : Générer un QR code avec simple-qrcode." *Akili School*, 19 Nov. 2020, <https://www.akilischool.com/cours/laravel-generer-un-qr-code-avec-simple-qrcode>.
- binary1110. "I Want to Refresh Form Submit Page but Do Not Resubmit It Again in Laravel." *Stack Overflow*, 8 Oct. 2018, <https://stackoverflow.com/q/52696964>.
- *Correcteur Orthographe | Correction Grammaire | SCRIBENS*. <https://www.scribens.fr/>.
- *HTML Forms*. <https://www.w3.org/MarkUp/html3/forms.html#:~:text=There%20can%20be%20several%20forms%20in%20a%20single%20document%2C%20but%20the%20FORM%20element%20can%27t%20be%20nested>. Accessed 3 June 2024.
- "Https://Laracasts.Com/Discuss/Channels/Laravel/Installing-Laravel-on-Server-without-Ssh." *Laracasts*, <https://laracasts.com/discuss/channels/laravel/installing-laravel-on-server-without-ssh>.
- *Install Tailwind CSS with Laravel - Tailwind CSS*. <https://tailwindcss.com/docs/guides/laravel>.
- Klensin, John C. *Application Techniques for Checking and Transformation of Names*. Request for Comments, RFC 3696, Internet Engineering Task Force, Feb. 2004. IETF, <https://datatracker.ietf.org/doc/rfc3696/>.
- *Application Techniques for Checking and Transformation of Names*. Request for Comments, RFC 3696, Internet Engineering Task Force, Feb. 2004. IETF, <https://datatracker.ietf.org/doc/rfc3696/>.
- *Laravel - The PHP Framework For Web Artisans*. <https://laravel.com/>. Accessed
- *Laravel Naming Conventions - Web Dev Etc - My Software Development Blog*. <https://webdevetc.com/blog/laravel-naming-conventions/>.
- *Navbars - Official Tailwind CSS UI Components*. <https://www.tailwindui.com/>.
- Perera, Kanishka. "Building a CSV Import/Export Feature in Laravel 10: Step-by-Step Guide with No External Packages." *Medium*, 24 Feb. 2024, <https://medium.com/@kanishkanuwanperera/building-a-csv-import-export-feature-in-laravel-10-step-by-step-guide-with-no-external-packages-6b73295aa0c1>.
- *QR Code Generator for Laravel*. <https://github.com/SimpleSoftwareIO/simple-qrcode>.
- "Understanding GitHub Actions." *GitHub Docs*, <https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions>.

7.4 Table des illustrations

1. Organisation du projet	8
2. Représentation schématique du modèle MVC	9
3. Représentation schématique du modèle MVC appliquée à Laravel.	9
4. Modèle Conceptuel de Données (MCD) pour l'application SocialShuffle	10
5. Modèle Logique de Données (MLD) pour l'application SocialShuffle	11
6. Représentation des interactions principales avec le site.	11
7. Maquette de la page principale en version mobile. Avec le menu ouvert et fermé.	12
8. Maquette de la page principale en version desktop.....	13
9. Exemple de formulaire avec la maquette d'ajout des membres en version mobile.	14
10. Exemple de formulaire avec la maquette d'ajout des membres en version mobile.	15
11. Page de paramétrage d'une équipe	16
12. Configuration initiale du fichier .env pour un environnement de développement. (En rouge, les valeurs qui ont été modifiées).	20
13. Page par défaut de laravel.....	21
14. Affichage de la collection des membres.	29
15. Affichage de la liste des activités	31
16. Affichage des groupes participant à une activité.	32
17. Formulaire d'ajout du nom lors de la création d'une équipe en version mobile... ..	33
18. Démonstration du formulaire d'ajout des membres.	34
19. Formulaire d'authentification.....	35
20. Affichage du nom d'utilisateur.....	35
21. Affichage du bouton "Se déconnecter".....	36
22. Avertissement lors de la génération des groupes.....	39
23. Affichage du QR code sur la page d'une équipe.....	42
24. Formulaire d'importation d'un fichier CSV.....	43
25. Démonstration du site une fois déployé.....	46
26. Affichage de l'écran principal en version Desktop.	50
27. Version mobile de l'écran principal	50
28. Barre de navigation Desktop	51
29. Barre de navigation mobile	51
30. Boutons de gestion des membres en version Desktop.....	51
31. Boutons de gestion des membres au format mobile.....	52
32. Test automatisé réussi sur GitHub Actions.....	53
33. Capture d'écran provenant d'un test Dusk.....	55

7.5 Planning

Séquence 27			Date: lundi, 3 juin 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	28	2h20min	Impression si e n'est pas encore fait, dernière vérification et envoi de la version papier et électronique.	
Total tranche	28	2h20min		

Séquence 26			Date: lundi, 3 juin 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	36	3h	Finaliser, créer le document PDF annexe	
Administratif	12	1h	Impression de la version papier (1h pour prendre en compte le potentiel bouchon à l'imprimante)	
Total tranche	48	4h		

Séquence 25			Date: vendredi, 31 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Tests	37	3h5min		
Doc - Journal de travail	1	0h5min		
Total tranche	38	3h10min		

Séquence 24			Date: vendredi, 31 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Rôles des utilisateurs	48	4h		
Total tranche	48	4h		

Séquence 23			Date: jeudi, 30 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Authentification	7	0h35min	Gestion de la connexion	
Doc - Rapport	28	2h20min		
Administratif	2	0h10min	Envoyer du rapport et du journal de travail mis à jour.	
Doc - Journal de travail	1	0h5min		
Total tranche	38	3h10min		

Séquence 22			Date: jeudi, 30 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Algorithme de répartition	5	0h25min	Implémentation de l'algorithme dans le but de répartir les membres de l'équipe dans les groupes	
Réalisation - Authentification	43	3h35min	Gestion de la connexion	
Total tranche	48	4h		

Séquence 21			Date: mercredi, 29 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Algorithme de répartition	27	2h15min	Implémentation de l'algorythme dans le but de répartir les membres de l'équipe dans les groupes	
Doc - Journal de travail	1	0h5min		
Total tranche	28	2h20min		

Séquence 20			Date: mardi, 28 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	25	2h5min		
Administratif	2	0h10min	Envoi du rapport et du journal de travail mis à jour.	
Doc - Journal de travail	1	0h5min		
Total tranche	28	2h20min		

Séquence 19			Date: mardi, 28 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Mise au point avec le chef de projet	4	0h20min		
Réalisation - Algorithme de répartition	44	3h40min	Implémentation de l'algorythme dans le but de répartir les membres de l'équipe dans les groupes	
Total tranche	48	4h		

Séquence 18			Date: lundi, 27 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	27	2h15min		
Doc - Journal de travail	1	0h5min		
Total tranche	28	2h20min		

Séquence 17			Date: lundi, 27 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - CRUD équipes et membres	48	4h	Contrôleur permettant de réaliser les opérations CRUD sur les membres	
Total tranche	48	4h		

Séquence 16			Date: vendredi, 24 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - CRUD équipes et membres	30	2h30min	Contrôleur permettant l'ajout / modification / lecture / suppression des équipes	
Doc - Rapport	7	0h35min		
Doc - Journal de travail	1	0h5min		
Total tranche	38	3h10min		

Séquence 15			Date: vendredi, 24 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Algorithme de répartition	36	3h	Rechercher et tester des algorithmes permettant la mise en place de la fonctionnalité voulue.	
Réalisation - CRUD équipes et membres	12	1h	Contrôleur permettant l'ajout / modification / lecture / suppression des équipes	
Total tranche	48	4h		

Séquence 14			Date: jeudi, 23 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Vues	12	1h	Formulaire de création des membres	
Réalisation - Vues	8	0h40min	Page d'affichage de l'équipe	
Doc - Rapport	15	1h15min		
Administratif	2	0h10min	Envoi du rapport et du journal de travail mis à jour.	
Doc - Journal de travail	1	0h5min		
Total tranche	38	3h10min		

Séquence 13			Date: jeudi, 23 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Vues	12	1h	Layout	
Réalisation - Vues	12	1h	Formulaire de connexion	
Réalisation - Vues	12	1h	Page principale où s'affichent les équipes	
Réalisation - Vues	12	1h	Formulaire de création d'équipes	
Total tranche	48	4h		

Séquence 12			Date: mercredi, 22 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Base de données	25	2h5min		
Doc - Rapport	2	0h10min		
Doc - Journal de travail	1	0h5min		
Total tranche	28	2h20min		

Séquence 11			Date: mardi, 21 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	25	2h5min		
Administratif	2	0h10min	Envoyer du rapport et du journal de travail mis à jour.	
Doc - Journal de travail	1	0h5min		
Total tranche	28	2h20min		

Séquence 10			Date: mardi, 21 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Mise au point avec le chef de projet	4	0h20min		
Réalisation - Crédit / Mise en place du projet Laravel	9	0h45min		
Réalisation - Base de données	35	2h55min	Définir les migrations et les lancer afin de créer la base de données	
Total tranche	48	4h		

Séquence 9			Date: vendredi, 17 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Analyse - Tests	27	2h15min	Décrire la stratégie de tests à employer	
Doc - Rapport	10	0h50min		
Doc - Journal de travail	1	0h5min		
Total tranche	38	3h10min		

Séquence 8			Date: vendredi, 17 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Analyse - Schéma de principe	36	3h	Réalisation d'un schéma démontrant les actions possibles pour l'utilisateur.	
Analyse - Tests	12	1h		
Total tranche	48	4h		

Séquence 7			Date: jeudi, 16 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	35	2h55min		
Administratif	2	0h10min	Envoi du rapport et du journal de travail mis à jour.	
Doc - Journal de travail	1	0h5min		
Total tranche	38	3h10min		

Séquence			6	Date: jeudi, 16 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Analyse - Maquettes graphiques	48	4h			
Total tranche	48	4h			

Séquence			5	Date: mercredi, 15 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Analyse - Modélisation de la DB	27	2h15min			
Doc - Journal de travail	1	0h5min			
Total tranche	28	2h20min			

Séquence			4	Date: mardi, 14 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	12	1h	Création du document et mise en page initiale		
Analyse - Modélisation de la DB	13	1h5min			
Administratif	2	0h10min	Envoi de la version mise à jour du rapport et du journal de travail		
Doc - Journal de travail	1	0h5min			
Total tranche	28	2h20min			

Séquence	3			Date: mardi, 14 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Mise au point avec le chef de projet	4	0h20min			
Analyse - Environement	44	3h40min	Analyse de l'environnement du proejt (Public cible, méthodologie de projet, Laravel, MVC)		
Total tranche	48	4h			

Séquence	2			Date: lundi, 13 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Analyse - Planification initiale	25	2h5min			
Administratif	2	0h10min	Envoi de la planif initiale aux experts et au chef de projet		
Doc - Journal de travail	1	0h5min			
Total tranche	28	2h20min			

Séquence	1			Date: lundi, 13 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Administratif	8	0h40min			
Analyse - Planification initiale	40	3h20min			
Total tranche	48	4h			

7.7 Journal de travail

Séquence 28			Date: lundi, 3 juin 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	17	1h25min	Terminer la relecture et correction du rapport.	
Administratif	11	0h55min	Le reste du temps est consacré l'importation du journal de travail dans le rapport, l'envoi de la version finale en pdf et l'impression du rapport.	
Total tranche	28	2h20min		

Séquence 27			Date: lundi, 3 juin 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	2	0h10min	Création d'une bibliographie à l'aide de ZoteroBib	
Administratif	4	0h20min	Réalisation de l'auto-évaluation du projet dans son état actuel afin de s'assurer qu'aucun point critique n'a été évalué.	
Doc - Rapport	4	0h20min	Rédaction de la comparaison entre la planification et la réalisation. Vérification de la mise en page.	
Réalisation - Vues	2	0h10min	Après réflexion, la page "À propos est retirée"	
Réalisation - Vues	15	1h15min	Déplacer les l'affichage des Activités dans une page séparée, comme demandé dans le cahier des charges.	
Réalisation - CRUD équipes et membres	2	0h10min	Ajout d'une fonctionnalité de suppression	
Réalisation - Déploiement	3	0h15min	Copier le répertoire contenant l'application afin d'effectuer les manipulations de préparation pour effectuer un nouveau déploiement de l'application mise à jour.	
Doc - Rapport	14	1h10min	Relecture du rapport et correction des erreurs ou ajout de précisions. Non terminé	
Doc - Journal de travail	2	0h10min		
Total tranche	48	4h		

Séquence 26			Date: dimanche, 2 juin 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Tests	5	0h25min	Création d'un test Dusk Explication de la création du test.	
Doc - Rapport	4	0h20min		
Réalisation - Tests	7	0h35min	Création du restant des tests avec l'aide de Chat-GPT4. S'agissant d'une tâche répétitive, il était plus rapide d'agir ainsi.	
Doc - Rapport	9	0h45min	Expliquer la mise en place de la création de Codes QR et l'importation des membres depuis un fichier CSV.	
Doc - Rapport	7	0h35min		
Réalisation - Algorithme de répartition	24	2h	Tentative de corriger l'algorithme sans succès.	
Total tranche	56	4h40min		

Séquence 25			Date: vendredi, 31 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - CRUD équipes et membres	14	1h10min	Ajout de l'importation des membres depuis un fichier CSV et intégration de l'affichage d'un QR code	- Mise en place de la création de codes QR : https://www.akilischool.com/cours/laravel-generer-un-qr-code-avec-simple-qrcode - Importation d'un fichier CSV : https://medium.com/@kanishkanuwanperera/building-a-csv-import-export-feature-in-laravel-10-step-by-step-guide-with-no-external-packages-6b73295aa0c1
Doc - Rapport	23	1h55min	Terminer l'explication sur la mise en place de GitHub Action.	
Doc - Journal de travail	1	0h5min		
Total tranche	38	3h10min		

Séquence 24			Date: vendredi, 31 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	19	1h35min	Ajout de la planification dans les annexes et ajout d'une mesure sur responsive design.	
Réalisation - CRUD équipes et membres	10	0h50min	Ajout de la génération de QR code sur la page d'affichage d'une équipe. Installation d'une librairie existante pour Laravel permettant de facilement générer des QR Codes.	- Génération de QR codes : https://github.com/SimpleSoftwareIO/simple-qrcode
Réalisation - Vues	6	0h30min	Afin de rendre l'affichage des groupes plus simple, il faut déplacer les groupes d'une Activité dans une page séparée et uniquement montrer les activités sous forme de cartes dans la page de détails de l'équipe.	
Doc - Rapport	11	0h55min		
Doc - Journal de travail	2	0h10min		
Total tranche	48	4h		

Séquence 23			Date: jeudi, 30 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	16	1h20min	Terminer la rédaction sur les droits des utilisateurs.	
Réalisation - Tests	15	1h15min	Installation de Laravel Dusk et mise en place de GitHub Actions. Aucun test n'est encore créé à part le test présent par défaut.	
Doc - Rapport	4	0h20min	Explication de la mise en place des tests dans le rapport. Non terminé.	
Administratif	2	0h10min	Exporter les documents en PDF et envoi aux experts.	
Doc - Journal de travail	1	0h5min		
Total tranche	38	3h10min		

Séquence 22			Date: jeudi, 30 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Journal de travail	2	0h10min	Description du travail supplémentaire fait la veille.	
Doc - Rapport	36	3h	Rédaction du rapport pour la mise en place des vues et la procédure de déploiement.	
Visite de l'Expert 2	3	0h15min	Visite du deuxième expert, discussion sur le déroulement du projet. Ce qui est attendu pour la présentation.	
Doc - Rapport	6	0h30min	Début de la rédaction de la partie des droits pour les utilisateurs, non terminé.	
Doc - Journal de travail	1	0h5min		
Total tranche	48	4h		

Séquence 21			Date: mercredi, 29 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Déploiement	7	0h35min	Réaliser un déploiement initial de l'application sur le serveur de production. Copie du projet dans un nouveau répertoire, compilation du CSS avec <code>npm run build</code> et modification du dossier vendors contenant les dépendances avec <code>composer install --no-dev</code> . Au même moment, je réalise que le déploiement n'a pas été planifié, il s'agit d'une erreur de ma part.	https://laracasts.com/discuss/channels/laravel/installing-laravel-on-server-without-ssh
Doc - Rapport	4	0h20min	Régler certains éléments de mise en page pendant le transfert de fichier sur le serveur.	
Réalisation - Vues	16	1h20min	Continuer la réalisation de la vue responsable de l'affichage des groupes. Difficulté rencontrée : les équipes ayant un nombre souvent différent de groupes, il a fallu dynamiquement répartir les cartes contenant les membres de l'équipe en fonction de la génération à laquelle ils appartiennent.	
Doc - Journal de travail	1	0h5min		
			Le travail qui suit a été réalisé en dehors du temps alloué au projet. Il s'agit principalement de petites erreurs non critiques, qui mises bout-à-bout, prennent du temps à corriger.	
Réalisation - Vues	6	0h30min	Correction de plusieurs erreurs d'affichage dans l'application.	
Réalisation - Algorithme de répartition	4	0h20min	Ajout d'une vérification d'erreurs afin que l'utilisateur soit averti lorsque lorsqu'il n'est pas possible de créer une équipe de la taille souhaitée.	
Doc - Rapport	7	0h35min	Avancer dans la rédaction du rapport.	
Total tranche	45	3h45min		

Séquence 20			Date: mardi, 28 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	4	0h20min		
Réalisation - CRUD équipes et membres	6	0h30min	Ajout de commentaires supplémentaires dans le code.	
Doc - Rapport	9	0h45min	Continuer la rédaction du rapport. Modification de certains éléments de mise en page.	
Réalisation - Rôles des utilisateurs	6	0h30min	Transférer la gestion des rôles des utilisateurs dans les policies qui sont plus appropriées pour cela.	
Administratif	2	0h10min	Export des documents au format PDF et envoi aux experts et au chef de projet.	
Doc - Journal de travail	1	0h5min		
Total tranche	28	2h20min		

Séquence 19			Date: mardi, 28 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	9	0h45min	Continuer la rédaction des différentes parties de la réalisation du projet	
Mise au point avec le chef de projet	2	0h10min	Quelques petites erreurs sont présentes dans le rapport. Priorisation de certains éléments du projet, notamment le déploiement initial qui peut prendre du temps.	
Doc - Rapport	36	3h	Continuer la rédaction du rapport (Mise en place des contrôleur, Formulaire de création des équipes, Authentification, Mesures de sécurité).	
Doc - Journal de travail	1	0h5min		
Total tranche	48	4h		

Séquence 18			Date: lundi, 27 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Vues	23	1h55min	Continuer l'affichage des groupes. Non terminé, il faut réussire à séparer les groupes selon l'activité (génération) à laquelle ils appartiennent.	
Doc - Rapport	4	0h20min	Rédaction du rapport. Ajout de copies d'écran pour la partie démontrant la responsivité de l'application sous la forme mobile et Desktop.	
Doc - Journal de travail	1	0h5min		
Total tranche	28	2h20min		

Séquence 17			Date: lundi, 27 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Algorithme de répartition	35	2h55min	Implémentation de l'algorithme de répartition. L'algorithme réalisé jusqu'à présent est une version préliminaire, il n'est pas en mesure de détecter si deux membres ont déjà été dans la même équipe.	
Réalisation - Vues	12	1h	Réadapter la vue permettant d'afficher les membres dans leurs groupes. Pour l'instant seul une liste de membres est affichée. Pas encore terminé.	
Doc - Journal de travail	1	0h5min		
Total tranche	48	4h		

Séquence 16			Date: vendredi, 24 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	3	0h15min	Préparation préalable de certains chapitres à rédiger. Correction d'une erreur qui empêche de modifier les membres d'une équipe.	
Réalisation - CRUD équipes et membres	15	1h15min	Après plusieurs essais, le problème vient du fait que le bouton de suppression fait partie d'un formulaire qui se trouve à l'intérieur du formulaire de modification ce qui crée un conflit dans la méthode à utiliser pour la requête. De plus, d'après les standards HTML, les formulaires ne peuvent pas être imbriqués.	- Standard HTML interdisant l'imbrication des formulaires : https://www.w3.org/MarkUp/html3/forms.html#:~:text=There%20can%20be%20several%20forms%20in%20a%20single%20document%2C%20but%20the%20FORM%20element%20can%27t%20be%20nested.
Doc - Journal de travail	17	1h25min	Rédaction du rapport pour la partie de la réalisation.	
Doc - Journal de travail	3	0h15min	Oubli de la rédaction du journal de travail le matin.	
Total tranche	38	3h10min		
Séquence 15			Date: vendredi, 24 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Authentification	21	1h45min	Mise en place du système d'authentification. L'authentification sur SocialShuffle doit être possible en entrant le nom d'utilisateur ou l'adresse Email. La mise en place de ce système est inspirée d'une réponse du forum Laracast. Il est important de préciser que cette réponse est décrite comme ayant été générée par une intelligence artificielle. Il n'est pas demandé dans le cahier des charges de mettre en place la création de comptes. Cela sera ajouté si le temps le permet.	- Solution pour l'authentification avec Email et nom d'utilisateur : https://laracasts.com/discuss/channels/laravel/login-with-username-or-email-in-laravel
Réalisation - Rôles des utilisateurs	24	2h	Mise en place des rôles des utilisateurs. Dans les vues, si l'utilisateur est admin, il a accès à toutes les fonctionnalités. Pour les utilisateurs authentifiés sans les droits admin, ils peuvent uniquement créer et gérer leurs propres équipes. Les utilisateurs non-authentifiés quant à eux ne peuvent que visualiser les équipes.	
Réalisation - Vues	3	0h15min	Modification du layout de du site afin que les boutons de la page active soient mis en évidence.	
Total tranche	48	4h		

Séquence 14			Date: jeudi, 23 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	6	0h30min	Ajout des termes spécifiques dans l'index Continuer la rédaction du rapport. - Ajout d'une maquette en version Desktop pour mieux visualiser les différences avec la version mobile. - Rédaction d'une courte partie expliquant les contrôleurs de ressources. - Explication de la liaison entre les routes et les contrôleurs. - Diverses corrections et ajouts dans des parties déjà rédigées.	
Réalisation - Base de données	24	2h	Modification des contraintes de clé étrangères avec une syntaxe plus simple.	https://laravel.com/docs/11.x/migrations#:~:text=Since%20this%20syntax%20is%20rather%20verbose%2C%20Laravel%20provides%20additi
Doc - Journal de travail	4	0h20min		
Administratif	2	0h10min	Export au format PDF de la quatrième version des documents et envoi.	
Total tranche	38	3h10min		
Séquence 13			Date: jeudi, 23 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - CRUD équipes et membres	14	1h10min	L'erreur a été corrigée. Elle venait du fait que à chaque soumission du formulaire, la vue était directement appellée sans passer par la route. Pour corriger cela, j'ai mis une redirection vers la route qui retourne la vue.	https://stackoverflow.com/questions/52696964/i-want-to-refresh-form-submit-page-but-do-not-resubmit-it-again-in-laravel
Réalisation - CRUD équipes et membres	32	2h40min	Fonctionnalités de modification et de suppression des équipes et des membres. Lorsqu'une équipe est supprimée, une erreur apparaît concernant les contraintes de clés étrangères avec les membres. Pour régler cela, il faut ajouter "onDelete('cascade') dans la définition de la contrainte dans la migration. Cela permet de supprimer tous les membres lorsque l'équipe est supprimée.	
Réalisation - Base de données	1	0h5min	Ajout de la possibilité de supprimer une équipe avec ses membres malgré la contrainte de clé étrangère <code>[onDelete('cascade')]</code> .	
Doc - Journal de travail	1	0h5min		
Total tranche	48	4h		

Séquence 12			Date: mercredi, 22 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Vues	14	1h10min	Création du formulaire d'ajout des membres dans les équipes.	
Réalisation - CRUD équipes et membres	13	1h5min	Gérer la création des équipes et des membres et stockage dans la base de données. Quelques erreurs sont encore présentes, notamment lorsqu'on rafraîchit le formulaire de création des membres, le précédent membre est créé à nouveau ce qui crée des dupliques.	
Doc - Journal de travail	1	0h5min		
Total tranche	28	2h20min		

Séquence 11			Date: mardi, 21 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Vues	10	0h50min	Continuer la réalisation des vues : Intégration du formulaire de création de l'équipe (non terminé)	
Doc - Rapport	15	1h15min	Rédaction de la mise en place des routes et correction de certains éléments discuté le matin avec le chef de projet.	
Doc - Journal de travail	1	0h5min		
Administratif	2	0h10min	Exporter les documents au PDF et envoyer comme convenu.	
Total tranche	28	2h20min		

Séquence 10			Date: mardi, 21 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Vues	3	0h15min	Commencer la réalisation des vues en commençant par la page principale où seront affichés les équipes. Je possède un petit projet qui a été réalisé durant la période de pré-tpi. Certains éléments d'affichage de ce projet peuvent être repris et adaptés à ce projet-ci.	
Mise au point avec le chef de projet	6	0h30min	- Précision sur les objectifs : L'importation du CSV et le QR code sont à faire dans un deuxième temps - Orientation du graphique de la planification initiale - Précisions sur le fonctionnement de l'attribution du nombre de membres pour chaque groupe	
Réalisation - Vues	38	3h10min	Continuer la réalisation des vues	
Doc - Journal de travail	1	0h5min		
Total tranche	48	4h		

Séquence 9			Date: vendredi, 17 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Réalisation - Base de données	6	0h30min	Création des fichiers de migrations et lancer les migrations	
Analyse - Modélisation de la DB	1	0h5min	Changement du nom de la table des membres qui ne respecte pas les normes de l'ETML dans le MLD	
Doc - Rapport	29	2h25min	Déscription des tables créées dans la base de données. Ce sont les conventions de nommage de Laravel qui sont utilisées dans l'application.	
			Bilan : J'ai pu commencer la réalisation du projet plus tôt que ce qui était prévu en raison de la sur-estimation de la durée de réalisation du schéma de principe et de la description de la stratégie de tests.	
Doc - Journal de travail	2	0h10min		
Total tranche	38	3h10min		

Séquence 8			Date: vendredi, 17 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Analyse - Schéma de principe	11	0h55min	Réaliser un diagramme représentant les interactions possibles de l'utilisateur avec l'application.	
Administratif	1	0h5min	Ecrire à M.Oberson (Expert 2) afin de demander la date de passage pour le renouvellement.	
Doc - Rapport	2	0h10min	Intégrer le diagramme dans le rapport.	
			Décrire la stratégie de tests ainsi que les différents outils qui seront utilisés pour la mise en place des tests.	- Factories : https://laravel.com/docs/11.x/database-testing
Analyse - Tests	14	1h10min	Explication des factories, de laravel Dusk et de l'automatisation des tests avec Github Actions.	- Laravel Dusk : https://laravel.com/docs/11.x/dusk - Github Actions : https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions - Github Actions avec Laravel Dusk : https://laravel.com/docs/11.x/dusk#running-tests-on-github-actions
Réalisation - Crédit / Mise en place du projet Laravel	4	0h20min	Mise en place du projet Laravel avec Tailwind.	- Installer Laravel : https://laravel.com/docs/11.x#creating-a-laravel-project - Installer Tailwind dans un projet Laravel : https://tailwindcss.com/docs/guides/laravel
Doc - Rapport	6	0h30min	Explication de la procédure d'installation	
Doc - Journal de travail	1	0h5min		
Total tranche	39	3h15min		

Séquence 7			Date: jeudi, 16 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Analyse - Maquettes graphiques	12	1h	Continuer les maquettes graphiques.	
Doc - Rapport	23	1h55min	- Intégrer les maquettes dans la documentation et justification des choix faits.	
Administratif	2	0h10min	Préparer les documents PDF et les envoyer	
Doc - Journal de travail	1	0h5min		
Total tranche	38	3h10min		

Séquence 6			Date: jeudi, 16 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Analyse - Modélisation de la DB	3	0h15min	Changer les noms dans la modélisation afin de respecter les conventions de codage de l'ETML. Dans l'application cependant, ce seront les conventions de nommage de Laravel qui seront utilisés en raison des contraintes techniques et des bonnes pratiques.	
Doc - Rapport	6	0h30min	- Changement du modèle de rapport. Copie de ce qui a déjà été tapé dans le nouveau document du rapport. - Intégration de la modélisation dans le rapport.	
Analyse - Maquettes graphiques	38	3h10min	Continuer la conception des maquettes graphiques. Non terminé. Il manque encore quelques maquettes montrant l'affichage des générations et afficher certaines fonctionnalités manquantes.	
Doc - Journal de travail	1	0h5min		
Total tranche	48	4h		

Séquence 5			Date: mercredi, 15 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	19	1h35min	Continuer la rédaction du rapport. Cela me décale de ce qui avait été planifié initialement car en attente du retour sur la modélisation de la DB. Durant la rédaction du texte, la grammaire et l'orthographe sont contrôlées avec scribens.fr.	https://www.scribens.fr/
Analyse - Maquettes graphiques	7	0h35min	La rédaction des parties introducives étant assez avancées, je commence à réaliser un premier jet des maquettes graphiques avec l'outil en ligne "Figma".	
Analyse - Modélisation de la DB	1	0h5min	Retour sur la modélisation reçue. Il faut changer les noms des tables et des colonnes du MCD / MLD afin de respecter les conventions de nommage de l'ETML.	
Doc - Journal de travail	1	0h5min	La rédaction des parties introducitives a bien avancé. Certains chapitres devront encore être revus car ils présentent trop peu d'informations / contexte.	
Total tranche	28	2h20min		

Séquence 4			Date: mardi, 14 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Analyse - Modélisation de la DB	10	0h50min	Continuer et terminer la première version de la modélisation. Envoyer la première version à M.Charmier pour validation.	
Doc - Rapport	9	0h45min	Ajouter la numérotation aux pages. Insertion du graphique représentant la planification -> Difficulté rencontrée avec la lisibilité donc l'image a été tournée de 90°. Continuer la rédaction des parties d'analyse dans le rapport.	
Analyse - Environnement	6	0h30min	Commencer la rédaction de la description de l'environnement en expliquant le Framework Laravel.	https://laravel.com/docs/11.x/installation
Analyse - Modélisation de la DB			Le retour sur la modélisation devrait me parvenir demain (mercredi 15.05).	
Administratif	3	0h15min	Exporter tous les documents en PDF et envoyer aux experts et au chef de projet.	
Total tranche	28	2h20min		
Séquence 3			Date: mardi, 14 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Administratif	1	0h5min	Après avoir pris connaissance des remarques des experts concernant la planification, je crée un nouvel export de l'entièreté du document.	
Analyse - Planification initiale	4	0h20min	Des erreurs dans la planification concernant des horaires mal planifiés sont constatés à certaines séquences.	
Mise au point avec le chef de projet	4	0h20min	Je fais part des erreurs qui ont été remarquées. Il faut les corriger. Une fois cela fait, il faudra commencer la rédaction du rapport et commencer la modélisation de la base de données.	
Analyse - Planification initiale	22	1h50min	Continuer la correction des erreurs dans la planification. Cela change légèrement la durée de certaines tâches.	
Doc - Rapport	8	0h40min	Récupération du template Word du rapport. Rédaction d'une courte introduction et modification de la structure (Ajout et suppression de certains chapitres présents par défaut).	
Analyse - Modélisation de la DB	7	0h35min	Commencer la modélisation de la base de données avec le logiciel Looping qui permet de passer facilement du MCD au MLD. Le but est de terminer avant 15h afin d'obtenir un retour du chef de projet et insérer la modélisation dans le rapport.	Norme RFC 3696 pour la longueur maximale des emails
Doc - Journal de travail	2	0h10min		
Total tranche	48	4h		

Séquence 2			Date: lundi, 13 mai 2024	Après-Midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Analyse - Planification initiale	26	2h10min	Terminer la planification initiale.	
Administratif	2	0h10min	Export du bilan graphique contenant la planification et envoi aux experts et au chef de projet.	
Total tranche	28	2h20min		

Séquence 1			Date: lundi, 13 mai 2024	Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Administratif	8	0h40min	Visite du premier expert. Les conditions de réalisation ont été définies. Le cahier des charges a été transmis.	
Analyse - Planification initiale	40	3h20min	Définition des tâches à réaliser durant le Projet, en définir la durée et planifier le moment de leurs exécutions	
Total tranche	48	4h		