

SocialShuffle

Candidat : **DOS SANTOS Samuel**, FIN2

Chef de Projet : **CHARMIER Grégory**

Expert 1 : **VENRIES Luc**

Expert 2 : **OBERTSON Bernard**

Durée : **88h**

Table des matières

1	SPÉCIFICATIONS	3
1.1	TITRE	3
1.2	INTRODUCTION	3
1.3	OBJECTIFS	3
1.4	POINTS TECHNIQUES	3
1.5	MATÉRIEL ET LOGICIELS À DISPOSITION	3
1.6	PRÉREQUIS	4
1.7	LIVRABLES	4
2	PLANIFICATION INITIALE	5
3	ANALYSE	6
3.1	MÉTHODOLOGIE DE TRAVAIL	6
3.2	ORGANISATION DES RÉSULTATS DU TRAVAIL	6
3.3	ENVIRONNEMENT	6
3.3.1	Laravel	6
3.3.2	MVC	6
3.3.3	L'ORM Eloquent	7
3.4	CONCEPTION ET ANALYSE	8
3.4.1	Modèle conceptuel de données	Erreur ! Signet non défini.
3.4.2	Modèle logique de Données	Erreur ! Signet non défini.
3.5	STRATÉGIE DE TESTS	11
3.6	RISQUES TECHNIQUES	11
4	RÉALISATION	12
4.1	DOSSIER DE RÉALISATION	12
4.2	MODIFICATIONS	12
5	TESTS	12
5.1	DOSSIER DES TESTS	12
6	CONCLUSION	12
6.1	BILAN DES FONCTIONNALITÉS DEMANDÉES	12
6.2	BILAN DE LA PLANIFICATION	12
6.3	BILAN PERSONNEL	12
7	DIVERS	12
7.1	JOURNAL DE TRAVAIL	12
7.2	BIBLIOGRAPHIE	12
7.3	WEBOGRAPHIE	13
8	ANNEXES	13

1 SPÉCIFICATIONS

1.1 Titre

Réalisation de l'application web SocialShuffle qui a pour but de mélanger des participants pour 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 dans le but de favoriser les interactions sociales.

Un des cas d'utilisation possible est lors du début d'une année scolaire, de proposer des activités de team building en faisant en sorte que tous les participants interagissent entre eux. Cette application permettra donc de retirer la difficulté de devoir créer les groupes manuellement.

L'enjeu principal est la mise en pratique du Framework Laravel dans un projet concret.

1.3 Objectifs

Les objectifs évalués pour ce projet sont les suivants :

- Les formulaires doivent afficher des erreurs tout en remplissant 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 :

- 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.

1.4 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

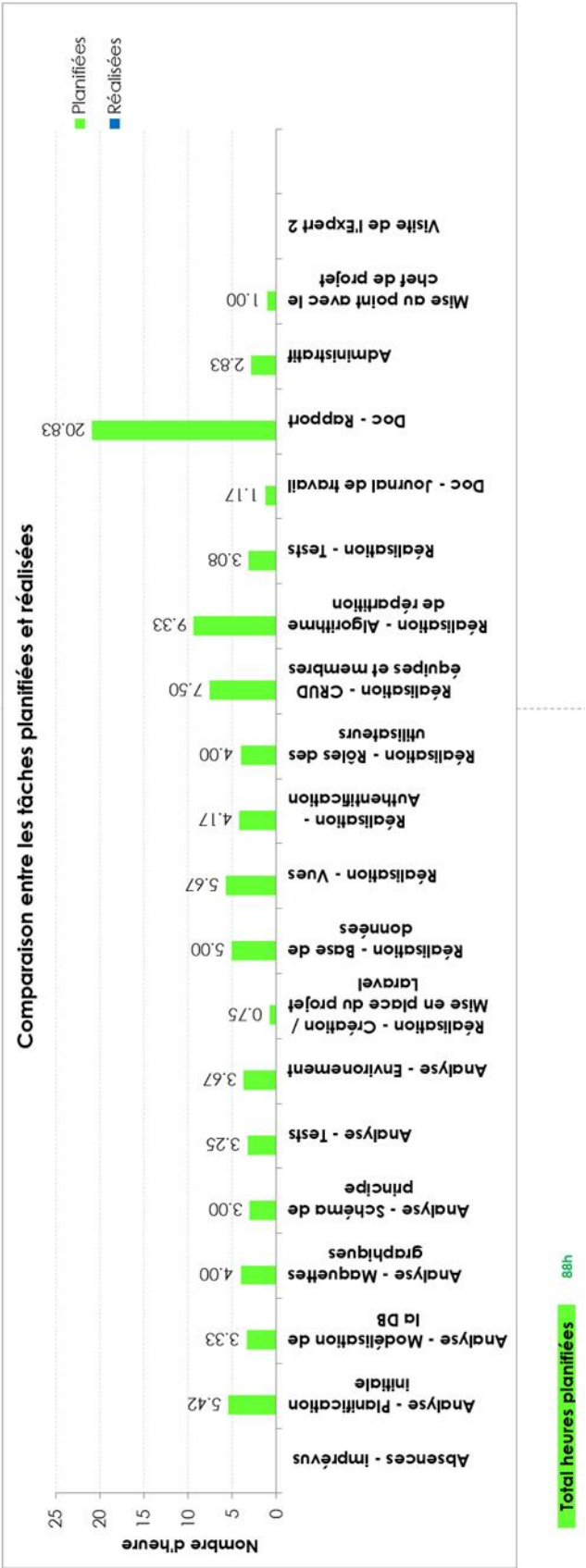
A compléter par ce qui est nécessaire pour le démarrage ...

1.6 Prérequis

A compléter par une description des compétences, des connaissances et de la formation minimum pour être à même de réaliser le projet ...

1.7 Livrables

2 PLANIFICATION INITIALE



- Ce paragraphe présente le planning d'origine (date de début, date de fin, vacances et congés, liste hiérarchique des tâches ou GANTT, jalons, durée totale)
- Toutes les mises à jour subies par le planning sont à reporter (avec date de mise à jour) et peuvent déboucher sur plusieurs versions de plannings.

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 jours, au moins un commit et push sont effectués afin d'enregistrer le travail
- Un répertoire est créé, dedans se trouve l'entièreté du projet. Il s'agit du repository.

3.3 Environnement

3.3.1 *Laravel*

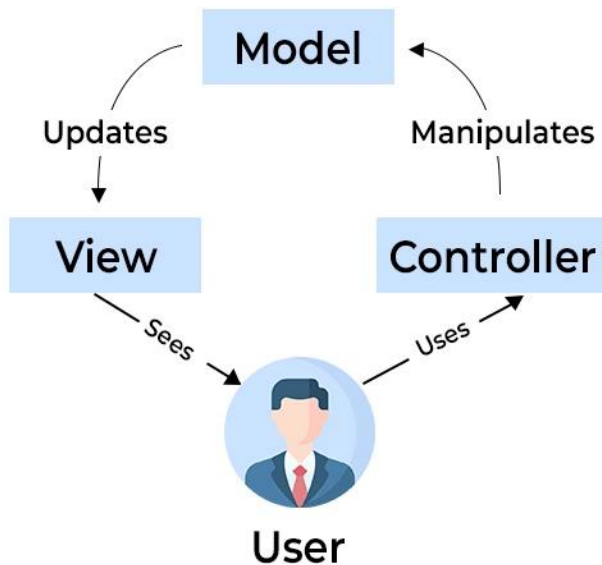
Laravel¹ est un Framework WEB basé sur le modèle [MVC](#). Il possède une syntaxe expressive qui permet de faire appel à fonctions de manière naturelle. L'interface en ligne de commandes (CLI) Artisan² permet de pouvoir interagir rapidement avec l'application (créer un contrôleur, effectuer des migrations vers la base de données ou même optimiser l'application pour le déploiement).

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 à l'application et le contrôleur permet de transporter les requêtes entre le modèle et la vue.

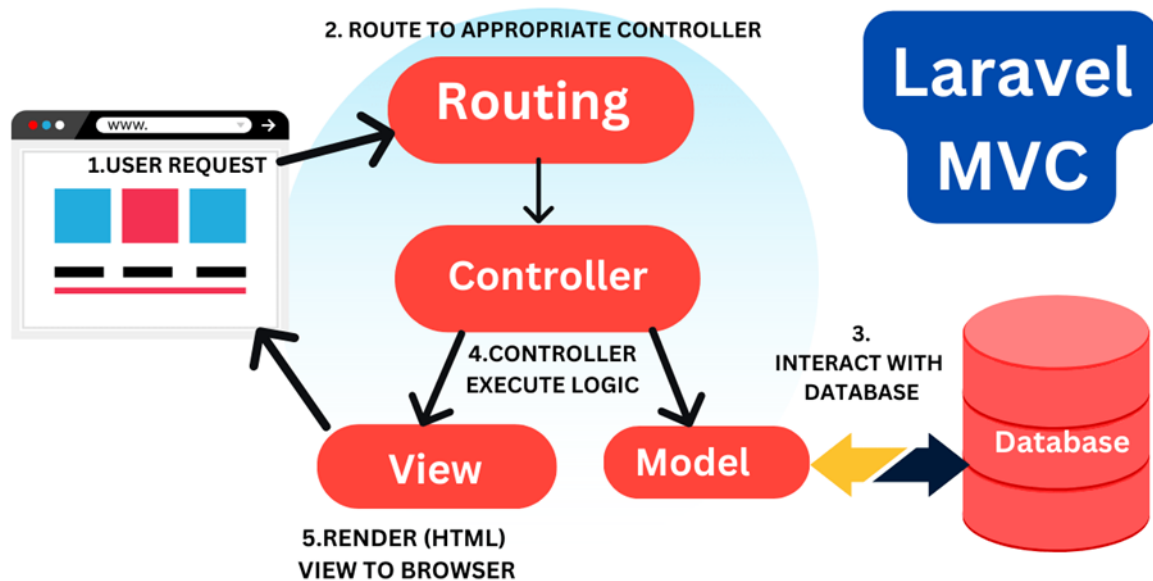
¹ Documentation officielle de Laravel <https://laravel.com/docs/11.x>

² Documentation sur Artisan : <https://laravel.com/docs/11.x/artisan#introduction>



1. Représentation schématique du modèle MVC

Laravel respecte ce principe fondamental en ajoutant le concept de routes. Celles-ci permettent à partir de l'Url de rediriger vers une fonction spécifique d'un contrôleur.



2. Représentation schématique du modèle MVC appliqué à Laravel.

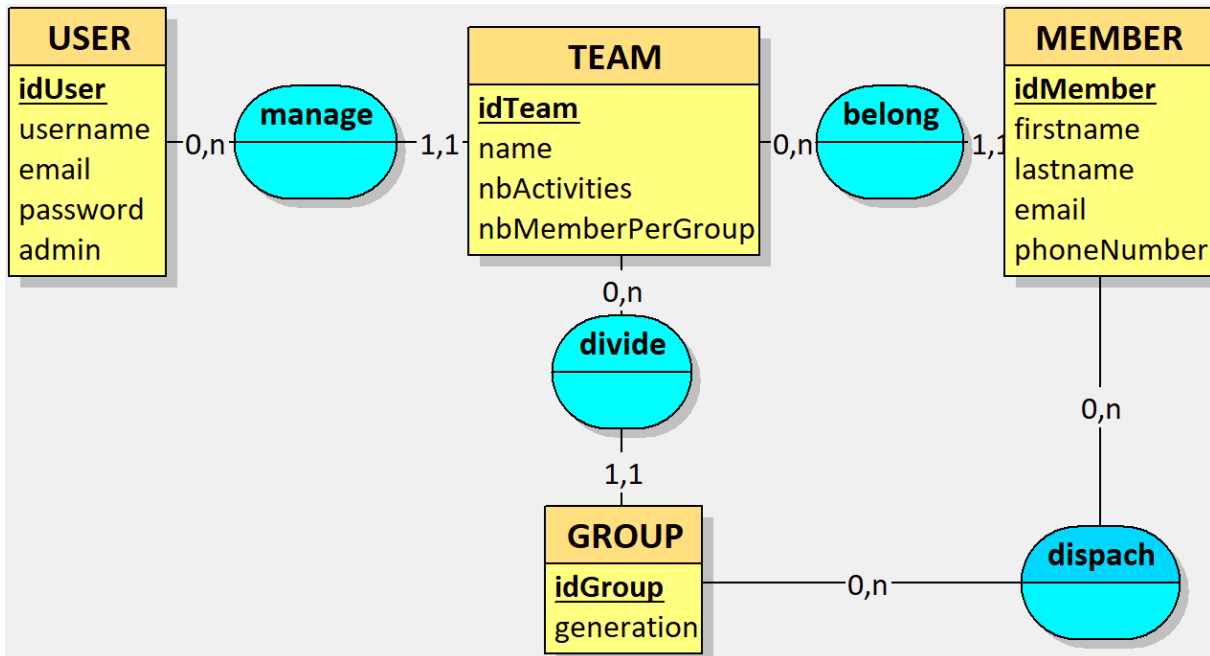
3.3.3 L'ORM Eloquent

Laravel met à disposition un ORM qui permet de décrire des requêtes à la base de données avec du code au lieu de taper des commandes directement des commandes 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.

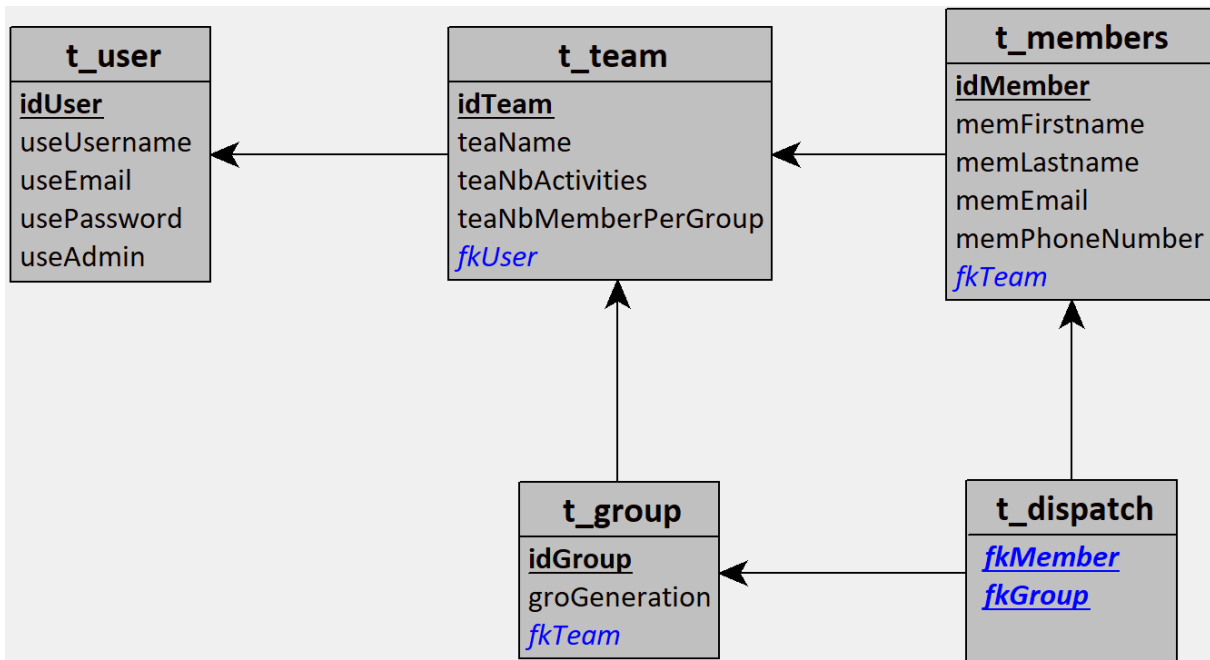
3.4 Conception et analyse

3.4.1 MCD / MLD

Le MCD et le MLD respectent ici les normes de codage de l'ETML. Il est cependant important de noter que les contraintes techniques imposées par Laravel empêchent l'utilisation de ces nommages. Ceci est la raison pour laquelle les noms diffèrent entre les modèles et le code.



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

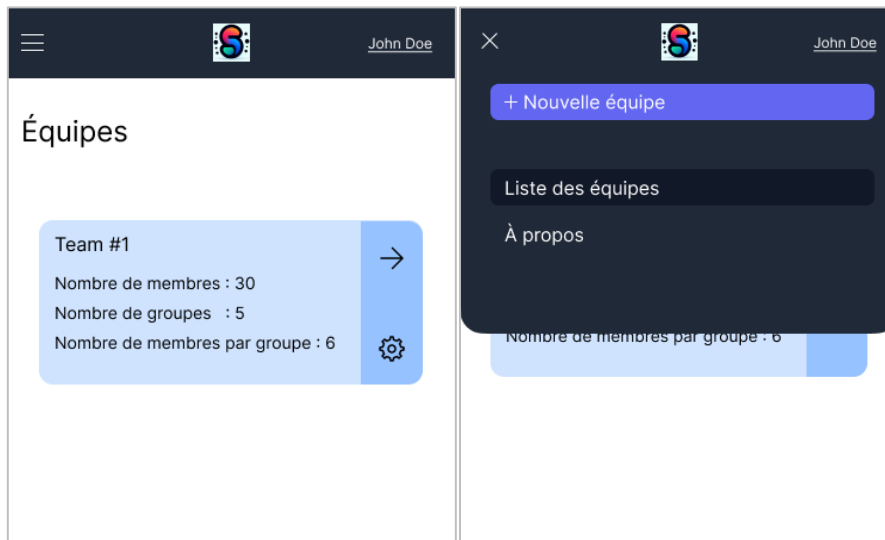


4. Modèle Logique de Données (MLD) pour l'application SocialShuffle

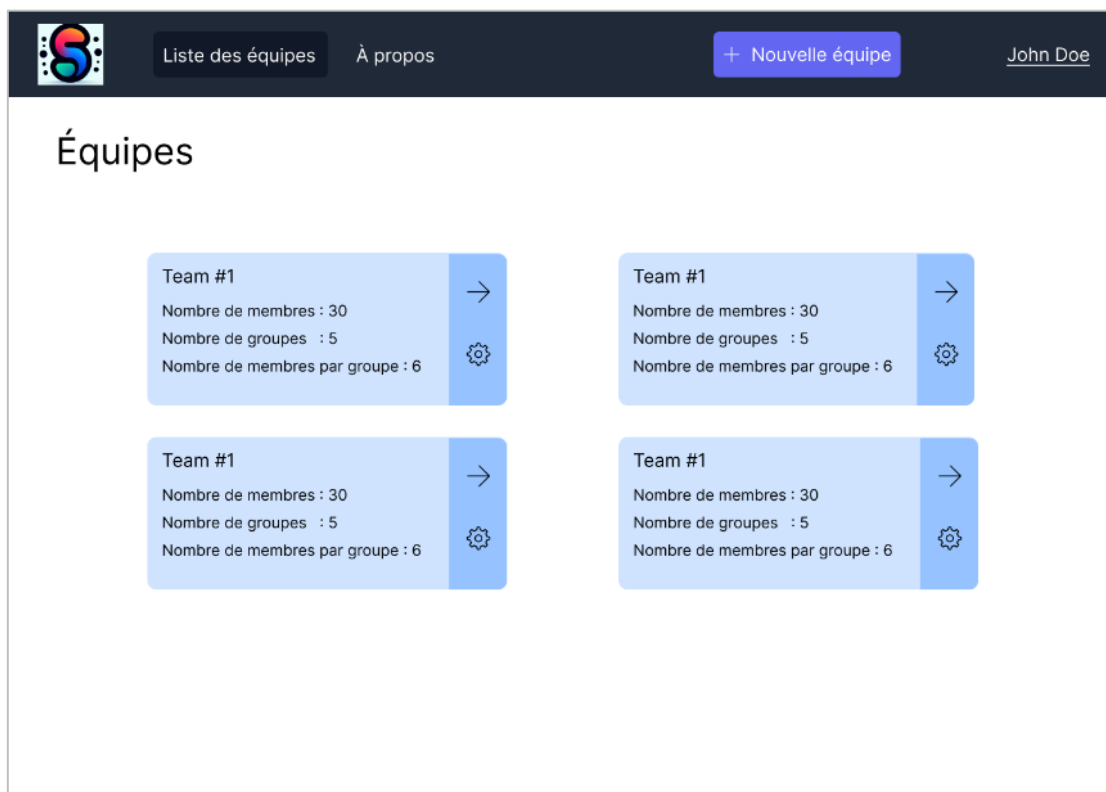
3.4.2 Maquettes graphiques

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 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 équipes par exemple.



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



6. 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.

Exemple de formulaire

A mobile application mockup for a social network. The top navigation bar is dark blue with a hamburger menu icon on the left, a logo in the center, and the user's name 'John Doe' on the right. The main content area is white and titled 'Ajouter les membres'. It contains a form with four input fields: 'Prénom' (containing 'John'), 'Nom' (containing 'Doe'), 'Email' (containing 'john.doe@example.ch'), and 'Numéro de téléphone' (containing '+41790000000'). A red error message is displayed below the email field: 'Email invalide. Veuillez respecter le format : exemple@exemple.com'. To the right of the 'Prénom' field is a blue button labeled 'Suivant →'. Below the form are two blue buttons: 'Ajouter un membre' and 'Importer depuis un CSV'. At the bottom, it says 'Nombre de membres : 25' and shows a list of members. The first member is 'John Doe' with email 'john.doe@example.ch', displayed in a light blue card.

7. Exemple de formulaire avec la maquette d'ajout des membres.

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.

Paramétrage de l'équipe

John Doe

Pramétrer l'équipe

Team #1

Modifier le nom

Nombre de membres : **30**

John Doe
john.doe@example.ch

John Doe
john.doe@example.ch

John Doe
john.doe@example.ch

Modifier les membres

Nombre d'activités : **5**

Nombre de membres par activité : **6**

Modifier les paramètres

8. Page de paramétrage d'une équipe

Cette maquette montre la page où sont affichés 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

3.6 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-qrcode³ »

4 RÉALISATION

4.1 Dossier de Réalisation

- Cette partie permet de reproduire ou reprendre le projet par un tiers.
- Pour chaque étape, il faut décrire sa mise en œuvre. Typiquement :
 - Versions des outils logiciels utilisés (OS, applications, pilotes, librairies, etc.)
 - Configurations spéciales des outils (Equipements, PC, machines, outillage, etc.)
 - Code source commenté des éléments logiciels développés.
 - Modèle physique d'une base de données.
 - Arborescences des documents produits.
- Il faut décrire le parcours de réalisation et justifier les choix.

4.2 Modifications

- Historique des modifications demandées (ou nécessaires) aux spécifications détaillées.
- Date, raison, description, etc.

5 TESTS

5.1 Dossier des tests

- On dresse le bilan des tests effectués (qui, quand, avec quelles données...) sous forme de procédure. Lorsque cela est possible, fournir un tableau des tests effectués avec les résultats obtenus et les actions à entreprendre en conséquence (et une estimation de leur durée).
- Si des tests prévus dans la stratégie n'ont pas pu être effectués :
 - raison, décisions, etc.
- Liste des bugs répertoriés avec la date de découverte et leur état :
 - Corrigé, date de correction, corrigé par, etc.

6 CONCLUSION

6.1 Bilan des fonctionnalités demandées

- Il s'agit de reprendre point par point les fonctionnalités décrites dans les spécifications de départ et de définir si elles sont atteintes ou pas, et pourquoi.
- Si ce n'est pas le cas, estimer en « % » ou en « temps supplémentaire » le travail qu'il reste à accomplir pour terminer le tout.

6.2 Bilan de la planification

- Distinguer et expliquer les tâches qui ont généré des retards ou de l'avance dans la gestion du projet. Indiquer les différences entre les planifications initiales et détaillées avec le journal de travail.

6.3 Bilan personnel

- Si c'était à refaire :
 - Qu'est-ce qu'il faudrait garder ? Les plus et les moins ?
 - Qu'est-ce qu'il faudrait gérer, réaliser ou traiter différemment ?
- Qu'est-ce que ce projet m'a appris ?
- Suite à donner, améliorations souhaitables, ...
- Remerciements, signature, etc.

7 DIVERS

7.1 Journal de travail

- Date, activité (description qui permet de reproduire le cheminement du projet), durée, liens et références sur des documents externes. Lorsqu'une activité de recherches a été entreprise, il convient d'énumérer ce qui a été trouvé, avec les références.

7.2 Bibliographie

- Références des livres, revues et publications utilisés durant le projet.

³ Tutoriel d'installation et utilisation de « simple-qrcode » : <https://www.akilischool.com/cours/laravel-generer-un-qr-code-avec-simple-qrcode>

7.3 Webographie

- Références des sites Internet consultés durant le projet.

8 ANNEXES

- Listing du code source (partiel ou, plus rarement complet)
- Guide(s) d'utilisation et/ou guide de l'administrateur
- Etat ou « dump » de la configuration des équipements (routeur, switch, robot, etc.).
- Extraits de catalogue, documentation de fabricant, etc.