SocialShuffle

Candidat : **DOS SANTOS Samuel**, FIN2

Chef de Projet : **CHARMIER Grégory**

Expert 1 : **Venries Luc**

Expert 2 : **Oberson Bernard**

Durée : **88h**

Table des matières

[1 Spécifications 3](#_Toc166761792)

[1.1 Titre 3](#_Toc166761793)

[1.2 Introduction 3](#_Toc166761794)

[1.3 Objectifs 3](#_Toc166761795)

[1.4 Points techniques 3](#_Toc166761796)

[1.5 Matériel et logiciels à disposition 3](#_Toc166761797)

[1.6 Prérequis 4](#_Toc166761798)

[1.7 Livrables 4](#_Toc166761799)

[2 Planification Initiale 5](#_Toc166761800)

[3 Analyse 6](#_Toc166761801)

[3.1 Méthodologie de travail 6](#_Toc166761802)

[3.2 Organisation des résultats du travail 6](#_Toc166761803)

[3.3 Environnement 6](#_Toc166761804)

[3.3.1 Laravel 6](#_Toc166761805)

[3.3.2 MVC 6](#_Toc166761806)

[3.3.3 L’ORM Eloquent 7](#_Toc166761807)

[3.4 Conception et analyse 8](#_Toc166761808)

[3.4.1 Modèle conceptuel de données **Erreur ! Signet non défini.**](#_Toc166761809)

[3.4.2 Modèle logique de Données **Erreur ! Signet non défini.**](#_Toc166761810)

[3.5 Stratégie de tests 11](#_Toc166761811)

[3.6 Risques techniques 11](#_Toc166761812)

[4 Réalisation 12](#_Toc166761813)

[4.1 Dossier de Réalisation 12](#_Toc166761814)

[4.2 Modifications 12](#_Toc166761815)

[5 Tests 12](#_Toc166761816)

[5.1 Dossier des tests 12](#_Toc166761817)

[6 Conclusion 12](#_Toc166761818)

[6.1 Bilan des fonctionnalités demandées 12](#_Toc166761819)

[6.2 Bilan de la planification 12](#_Toc166761820)

[6.3 Bilan personnel 12](#_Toc166761821)

[7 Divers 12](#_Toc166761822)

[7.1 Journal de travail 12](#_Toc166761823)

[7.2 Bibliographie 12](#_Toc166761824)

[7.3 Webographie 13](#_Toc166761825)

[8 Annexes 13](#_Toc166761826)

# Spécifications

## Titre

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

## 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.

## 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.

## 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.

## Matériel et logiciels à disposition

A compléter par ce qui est nécessaire pour le démarrage …

## Prérequis

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

## Livrables

# 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.

# Analyse

## 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.

## 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.

## Environnement

### Laravel

Laravel[[1]](#footnote-1) est un Framework WEB basé sur le modèle [MVC](#_MVC). Il possède une syntaxe expressive qui permet de faire appel à fonctions de manière naturelle. L’interface en ligne de commandes (CLI) Artisan[[2]](#footnote-2) 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).

### 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.

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement

1. Représentation schématique du modèle MCV

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.

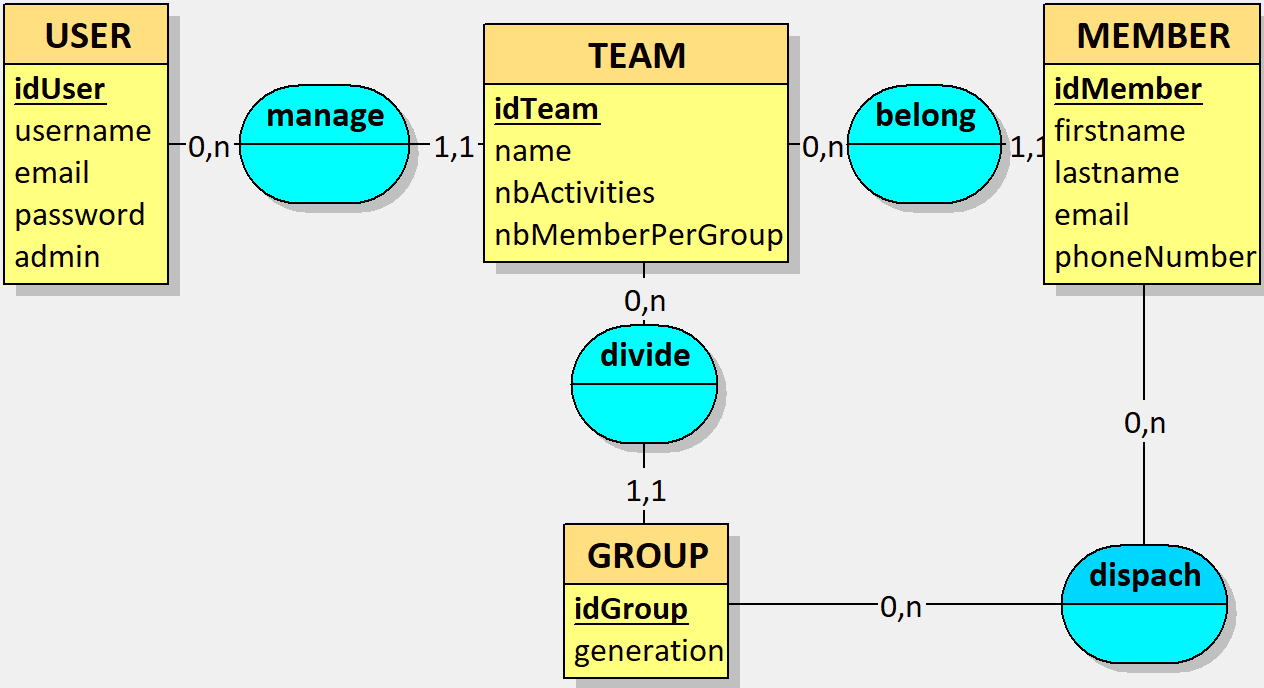
### L’ORM Eloquent

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.

## Conception et analyse

### 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

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

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

### 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.

Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Description générée automatiquementUne image contenant texte, capture d’écran

Description générée automatiquement

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

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

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

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

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

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

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.

## Stratégie de tests

## 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-qrcode[[3]](#footnote-3) »

# Réalisation

## 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.

## Modifications

Historique des modifications demandées (ou nécessaires) aux spécifications détaillées.

Date, raison, description, etc.

# Tests

## 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.

# Conclusion

## 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.

## 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 différence entre les planifications initiales et détaillées avec le journal de travail.

## 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 que ce projet m’a appris ?

Suite à donner, améliorations souhaitables, …

Remerciements, signature, etc.

# Divers

## 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.

## Bibliographie

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

## Webographie

Références des sites Internet consultés durant le projet.

# 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.

1. Documentation officielle de Laravel <https://laravel.com/docs/11.x> [↑](#footnote-ref-1)
2. Documentation sur Artisan : <https://laravel.com/docs/11.x/artisan#introduction> [↑](#footnote-ref-2)
3. Tutoriel d’installation et utilisation de « simple-qrcode » : <https://www.akilischool.com/cours/laravel-generer-un-qr-code-avec-simple-qrcode> [↑](#footnote-ref-3)