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](#_Toc308526316)

[1.1 Titre 3](#_Toc308526317)

[1.2 Description 3](#_Toc308526318)

[1.3 Matériel et logiciels à disposition 3](#_Toc308526319)

[1.4 Prérequis 3](#_Toc308526320)

[1.5 Cahier des charges 3](#_Toc308526321)

[1.5.1 Objectifs et portée du projet (objectifs SMART) 3](#_Toc308526322)

[1.5.2 Caractéristiques des utilisateurs et impacts 3](#_Toc308526323)

[1.5.3 Fonctionnalités requises (du point de vue de l’utilisateur) 3](#_Toc308526324)

[1.5.4 Contraintes 3](#_Toc308526325)

[1.5.5 Travail à réaliser par l'apprenti 4](#_Toc308526326)

[1.5.6 Si le temps le permet … 4](#_Toc308526327)

[1.5.7 Méthodes de validation des solutions 4](#_Toc308526328)

[1.6 Les points suivants seront évalués 4](#_Toc308526329)

[1.7 Validation et conditions de réussite 4](#_Toc308526330)

[2 Planification Initiale 4](#_Toc308526331)

[3 Analyse 4](#_Toc308526332)

[3.1 Opportunités 4](#_Toc308526333)

[3.2 Document d’analyse et conception 4](#_Toc308526334)

[3.3 Conception des tests 5](#_Toc308526335)

[3.4 Planification détaillée 5](#_Toc308526336)

[4 Réalisation 5](#_Toc308526337)

[4.1 Dossier de Réalisation 5](#_Toc308526338)

[4.2 Modifications 5](#_Toc308526339)

[5 Tests 5](#_Toc308526340)

[5.1 Dossier des tests 5](#_Toc308526341)

[6 Conclusion 5](#_Toc308526342)

[6.1 Bilan des fonctionnalités demandées 5](#_Toc308526343)

[6.2 Bilan de la planification 5](#_Toc308526344)

[6.3 Bilan personnel 5](#_Toc308526345)

[7 Divers 6](#_Toc308526346)

[7.1 Journal de travail 6](#_Toc308526347)

[7.2 Bibliographie 6](#_Toc308526348)

[7.3 Webographie 6](#_Toc308526349)

[8 Annexes 6](#_Toc308526350)

# Spécifications

## Titre

A compléter,par un titre cours et pertinent. Cela peut être une reprise ou compléter le titre de la première page …

## 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.

## 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)