|  |
| --- |
| Fiame Desk |



Christopher Pardo

SI-MI4a

TPI - 2022

Table des matières

[1 Analyse préliminaire 4](#_Toc104885515)

[*1.1* Introduction 4](#_Toc104885516)

[*1.2* Objectifs 4](#_Toc104885517)

[*1.3* Cahier des charges 5](#_Toc104885518)

[*1.4* Planification initiale 9](#_Toc104885519)

[2 Analyse / Conception 10](#_Toc104885520)

[*2.1* Concept 10](#_Toc104885521)

[2.1.1 Modèle Conceptuel des Données 10](#_Toc104885522)

[2.1.2 Modèle Logique des Données 11](#_Toc104885523)

[2.1.3 Maquettes 12](#_Toc104885524)

[*2.2* Stratégie de test 14](#_Toc104885525)

[*2.3* *Risques techniques* 14](#_Toc104885526)

[*2.4* Planification 15](#_Toc104885527)

[*2.5* Dossier de conception 15](#_Toc104885528)

[2.5.1 Matériel 15](#_Toc104885529)

[2.5.2 Logiciels 15](#_Toc104885530)

[2.5.3 Outils de développement 16](#_Toc104885531)

[3 Réalisation 16](#_Toc104885532)

[*3.1* Dossier de réalisation 16](#_Toc104885533)

[*3.2* Description des tests effectués 17](#_Toc104885534)

[3.2.1 Se connecter 17](#_Toc104885535)

[3.2.2 Modifier son profile 18](#_Toc104885536)

[3.2.3 Nommer un administrateur 20](#_Toc104885537)

[3.2.4 Ajouter un nouveau membre 21](#_Toc104885538)

[3.2.5 Consulter le calendrier 22](#_Toc104885539)

[3.2.6 Créer, modifier, supprimer un événement 23](#_Toc104885540)

[3.3 Erreurs restantes 24](#_Toc104885541)

[*3.4* Liste des documents fournis 24](#_Toc104885542)

[4 Conclusions 25](#_Toc104885543)

[4.1 Objectifs atteints 25](#_Toc104885544)

[4.2 Objectifs non atteints 25](#_Toc104885545)

[4.3 Difficultés particulières 25](#_Toc104885546)

[4.4 Suites possibles 25](#_Toc104885547)

[5 Annexes 26](#_Toc104885548)

[*5.1* Résumé du rapport du TPI / version succincte de la documentation 26](#_Toc104885549)

[5.1.1 Situation de départ 26](#_Toc104885550)

[5.1.2 Mise en œuvre 26](#_Toc104885551)

[5.1.3 Résultats 26](#_Toc104885552)

[*5.2* Sources – Bibliographie 27](#_Toc104885553)

[5.3 Journal de travail 28](#_Toc104885554)

[*5.4* Manuel d'Installation 31](#_Toc104885555)

[5.4.1 Prérequis 31](#_Toc104885556)

[5.4.2 Installation 31](#_Toc104885557)

[5.4.3 Alternative 31](#_Toc104885558)

[*5.5* Manuel d'Utilisation 32](#_Toc104885559)

[5.5.1 Utilisateurs de test 32](#_Toc104885560)

[5.5.2 Routes des pages 32](#_Toc104885561)

[6 Table des images 32](#_Toc104885562)

[7 Glossaire 33](#_Toc104885563)

# Analyse préliminaire

## Introduction

Ce projet s’inscrit dans le cadre du CFC d’informaticien comme projet final (TPI).

Il est encadré par un chef de projet (Xavier Carrel) et deux experts (Ernesto MONTEMAYOR et Benjamin WOLF).

Ce projet m’a été proposé car un autre candidat réalise une application mobile nomé Fiame. On m’a ensuite proposé de réaliser une API ainsi qu’une application web pour fonctionner en parelle de son projet.

Ce projet a pour but de créer une application web pour l’application mobile Fiame.

Cette application web permettra de gérer les rassemblements d’un groupe de personnes.

Je suis partit sur un site web en PHP mais avec la particularité d’utiliser le Framework Laravel. Cela me permettra d’approfondir plusieurs de mes fonctionnalités et d’avoir une base de développement.

En ce qui concerne le projet en lui-même, je ne pars de rien et rien n’a été réaliser avant.

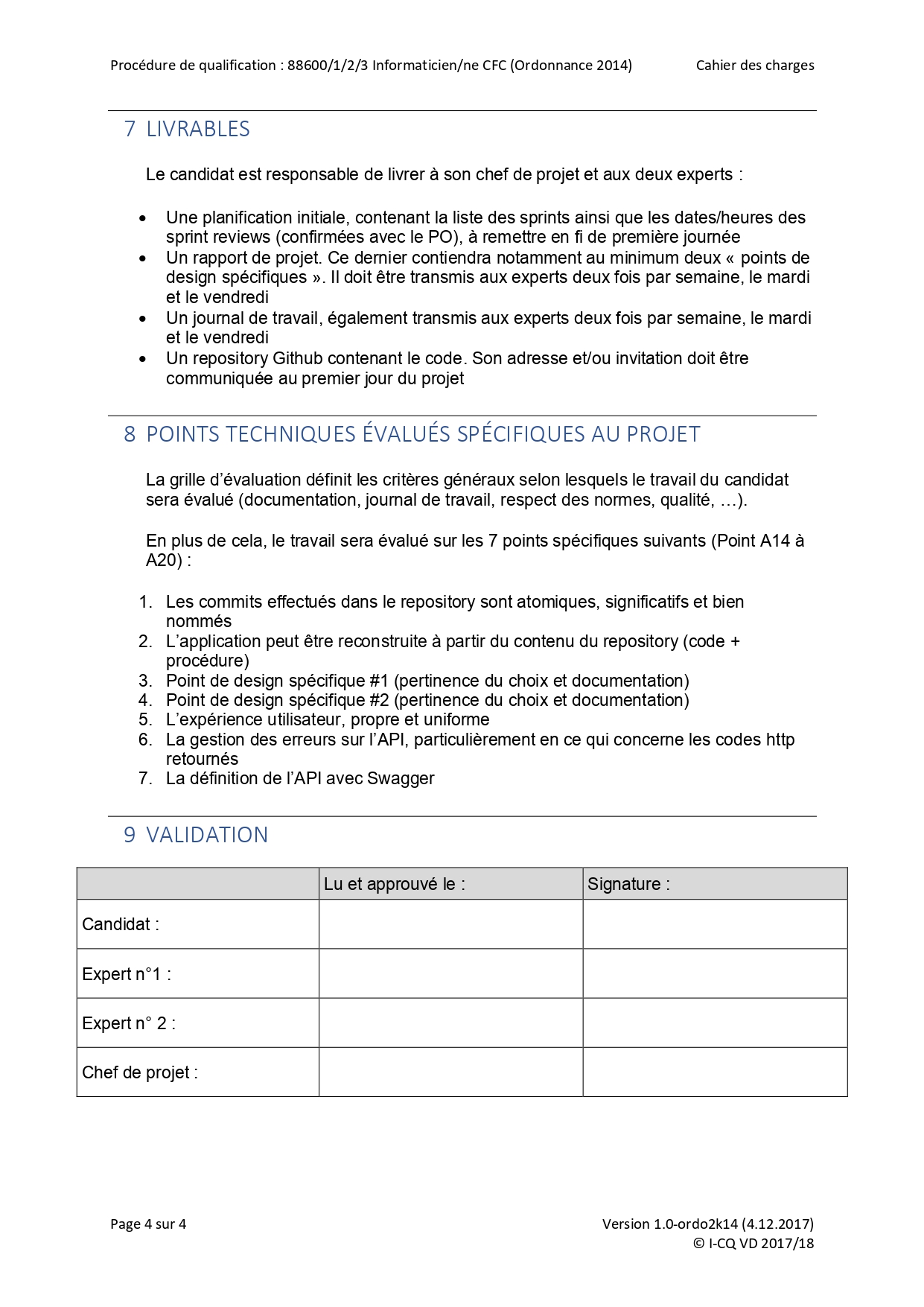
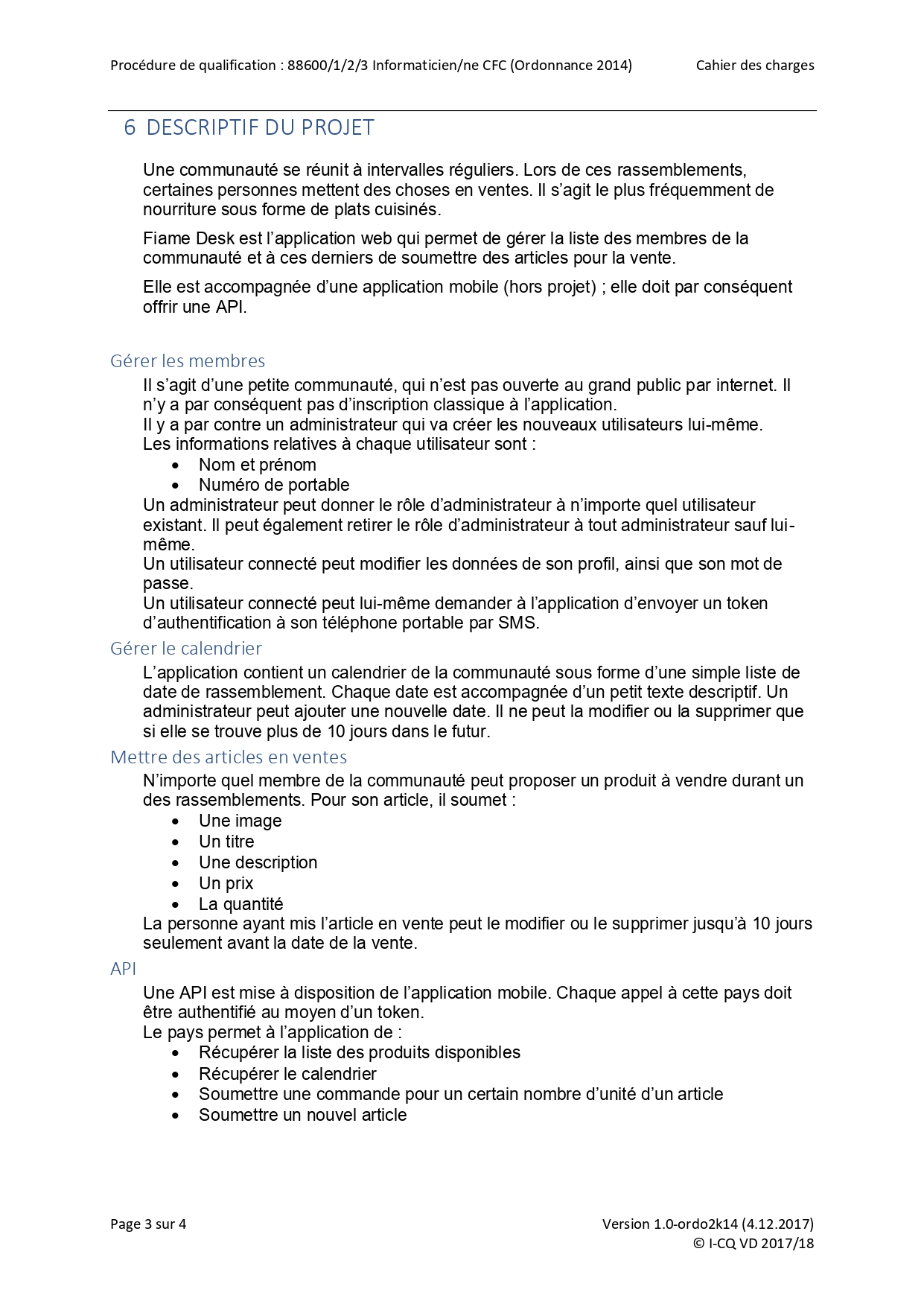
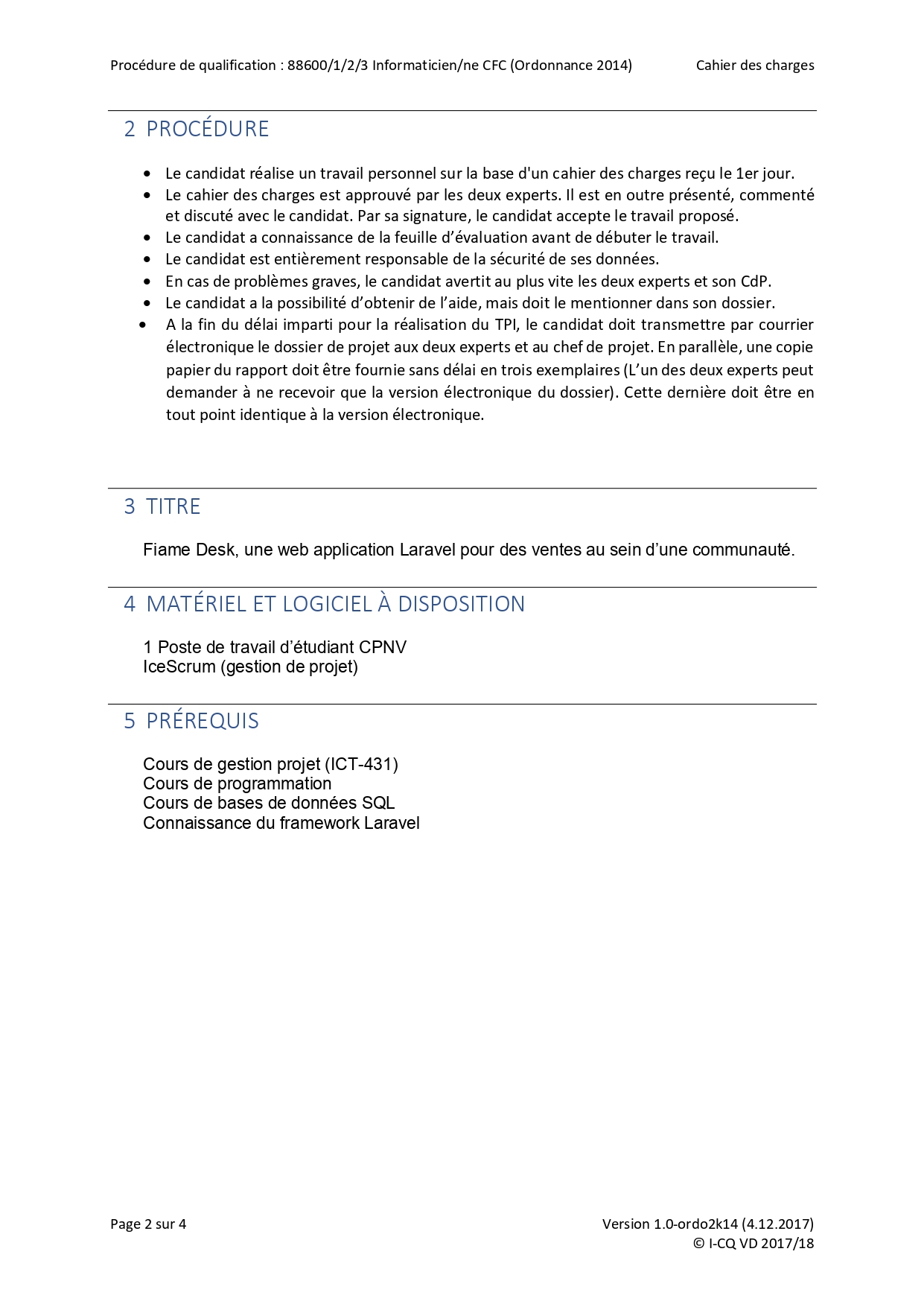
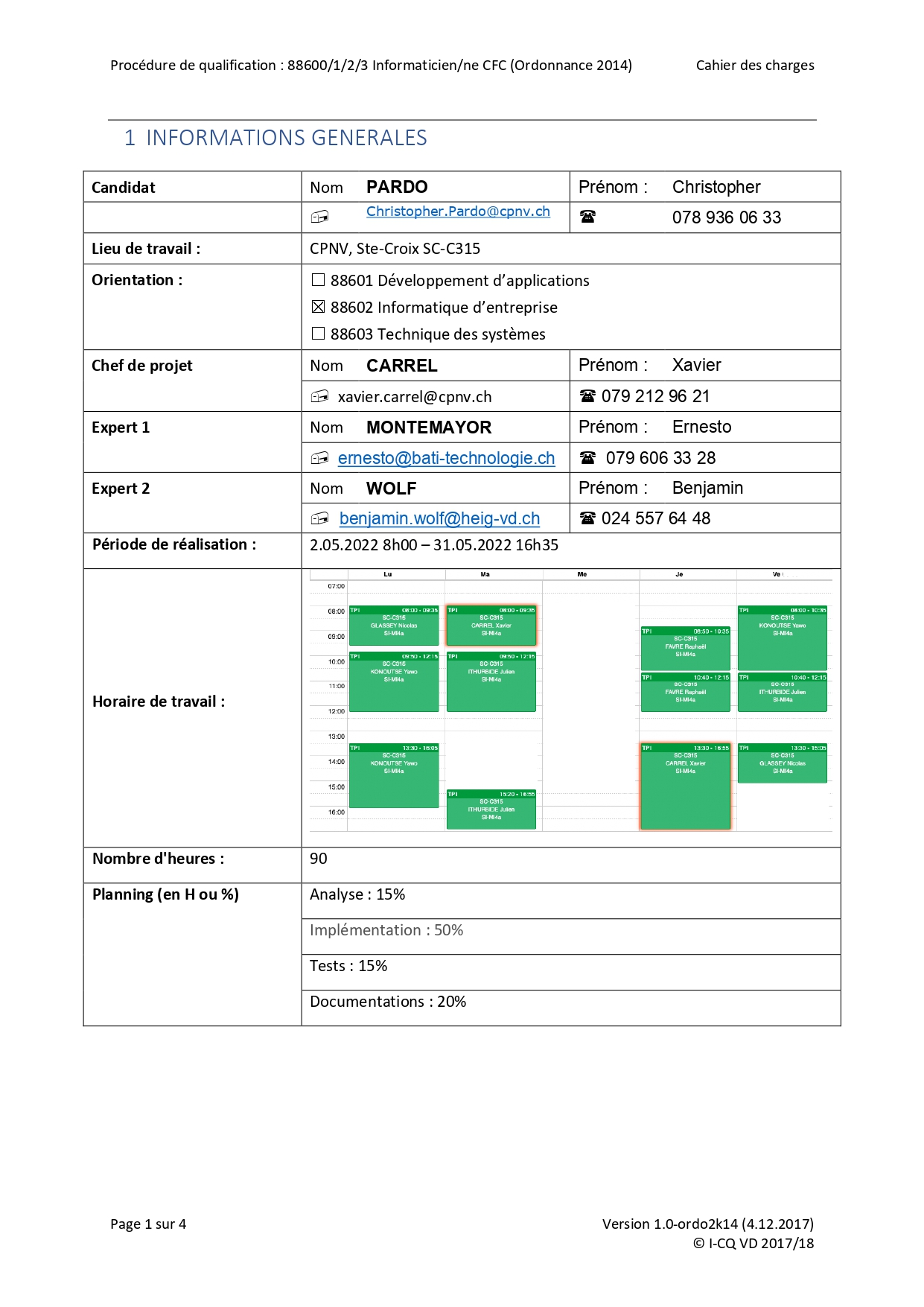
## Objectifs

Une communauté se réunit à intervalles réguliers. Lors de ces rassemblements, certaines personnes mettent des choses en ventes. Il s’agit le plus fréquemment de nourriture sous forme de plats cuisinés. Fiame Desk est l’application web qui permet de gérer la liste des membres de la communauté et à ces derniers de soumettre des articles pour la vente. Elle est accompagnée d’une application mobile (hors projet) ; elle doit par conséquent offrir une API.

Créer une application web en Laravel (un Framework PHP) contenant :

* Une gestion de membre
* Un système de login
* Un système de mise en vente de produit
* Un calendrier de rassemblement
* Un système de mise en vente d’objet par évènement
* Un système de modification de profils
* Un système d’administrateur
* Une API dont le but est de communiquer avec une application mobile

## Cahier des charges



## Planification initiale

Le projet est réalisé avec la méthode agile Scrum. Cette méthode nous permet de découper notre temps en différant section appelé Sprint. Durant les Sprint, nous réalisons certaines tâches définies appelées stories. Si les tâches ne sont pas terminées, elles sont renvoyées au sprint suivant. A chaque fin de sprint, nous avons une sprint review avec le chef de projet. Il valide si le travail correspond aux stories.

* Sprint 1 : 02/05/2022 - 06/05/2022
  + Démarrage du projet
  + Mise en place de l'environnement
  + Planification
  + Création du MCD
* Sprint 2 : 07/05/2022 - 17/05/2022
  + Création du site
  + Création de la base de donnée
  + Gestion des membres et des rôles
  + API des membres
* Sprint 3 : 18/05/2022 - 25/05/2022
  + Gestion des produits
  + Gestion des évènements
  + Validation des jetons
  + API des produits
* Sprint 4 : 26/05/2022 - 31/05/2022
  + Finalisation de la documentation
  + Testing

# Analyse / Conception

## Concept

### Modèle Conceptuel des Données

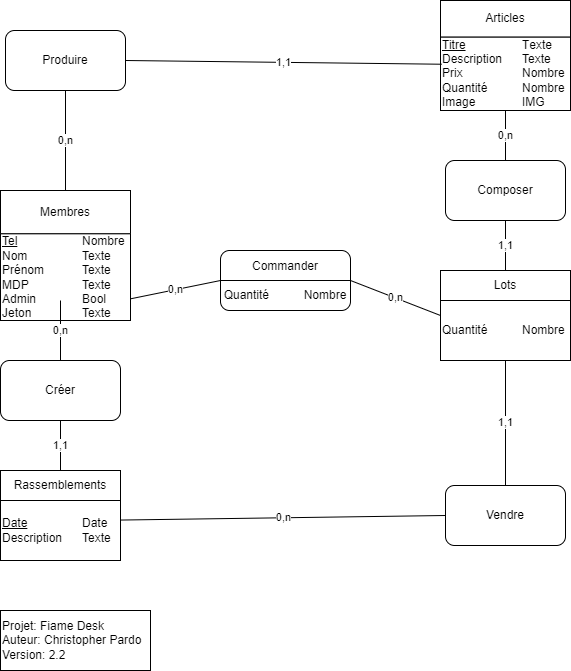


Figure 1 Modèle Conceptuel de donnée

Ce MCD se découpe en 4 tables et 5 jointures.

* Une table qui contiendra les membres
* Une table pour les rassemblements
* Une table qui va contenir les articles
* Une table qui va contenir les lots

Les articles ne sont pas directement vendus par les membres. Ils définissent une quantité qu’ils souhaitent mettre en vente et c’est ça un lot.

Les membres quant à eux peuvent :

* Produire les articles
* Vendre des lots lors de rassemblements
* Créer les rassemblements
* Commander des lots

### Modèle Logique des Données

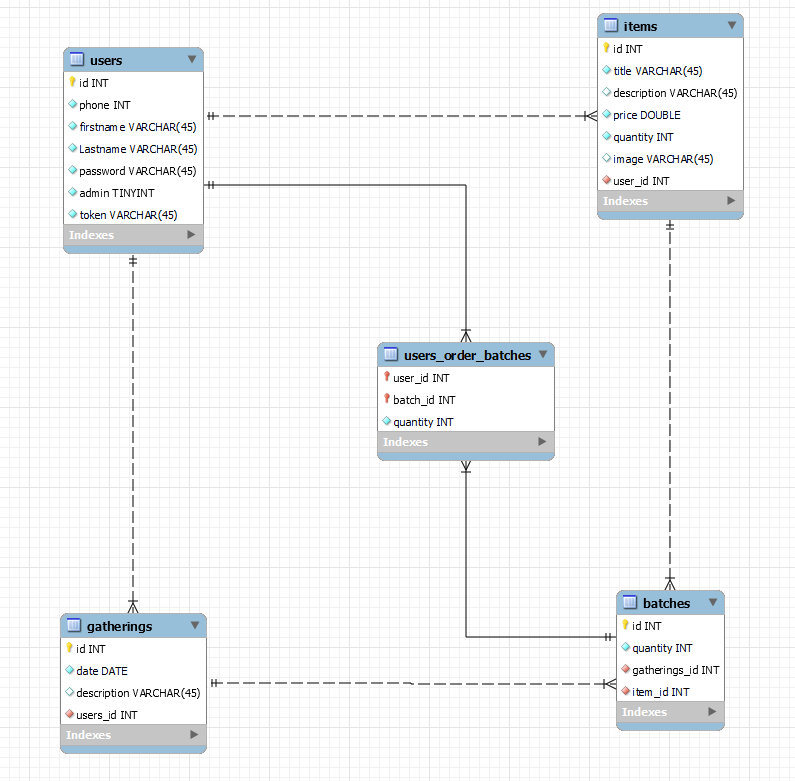


Figure 2 Modèle logique de donnée

Le MLD reprend les tables du MCD.

* Users : Membres
* Gatherings : Rassemblement
* Batches : Lots
* Items : Articles

Les points pleins signifient qu’ils sont facultatifs.

### Maquettes

Chaque maquette a été réalisé en accord avec les tests d’acceptation. Elles ont toutes été validée par le chef de projet.

Leur but est d’avoir un design simple et facile à prendre en main pour tout type de personne.

#### Page membres

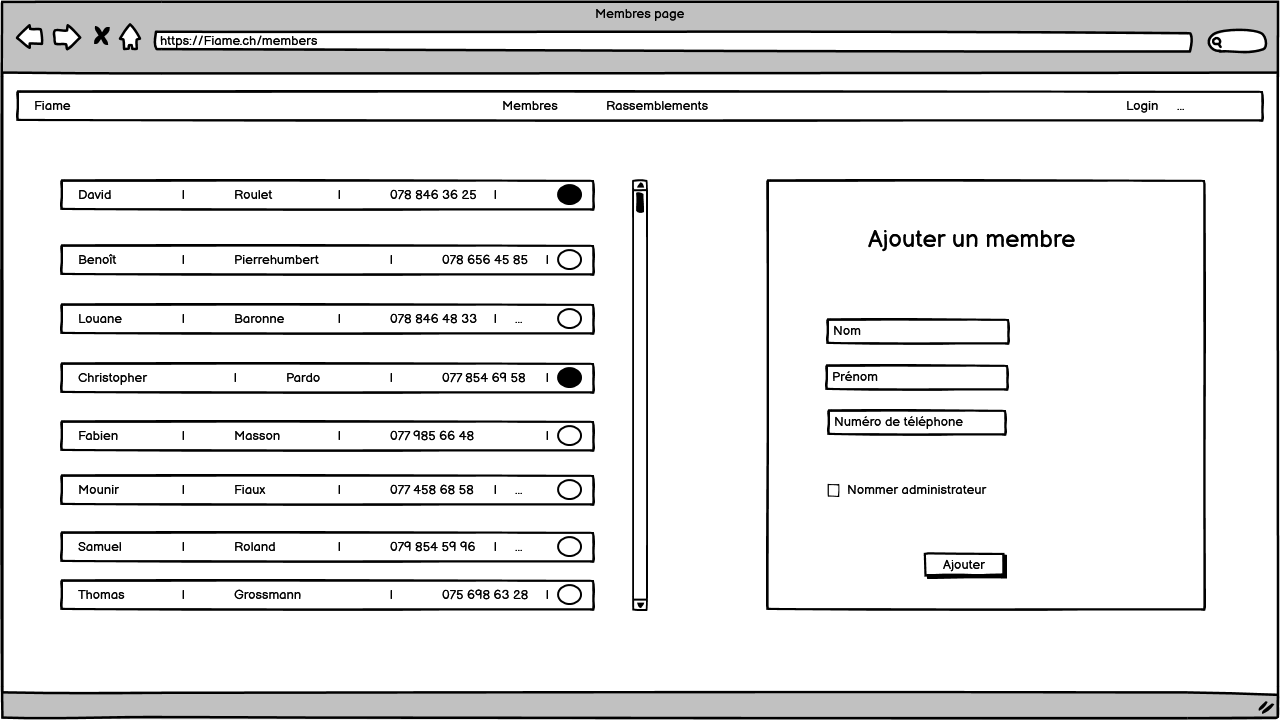


Figure 3 Maquette - Page de membres

#### Page de profile

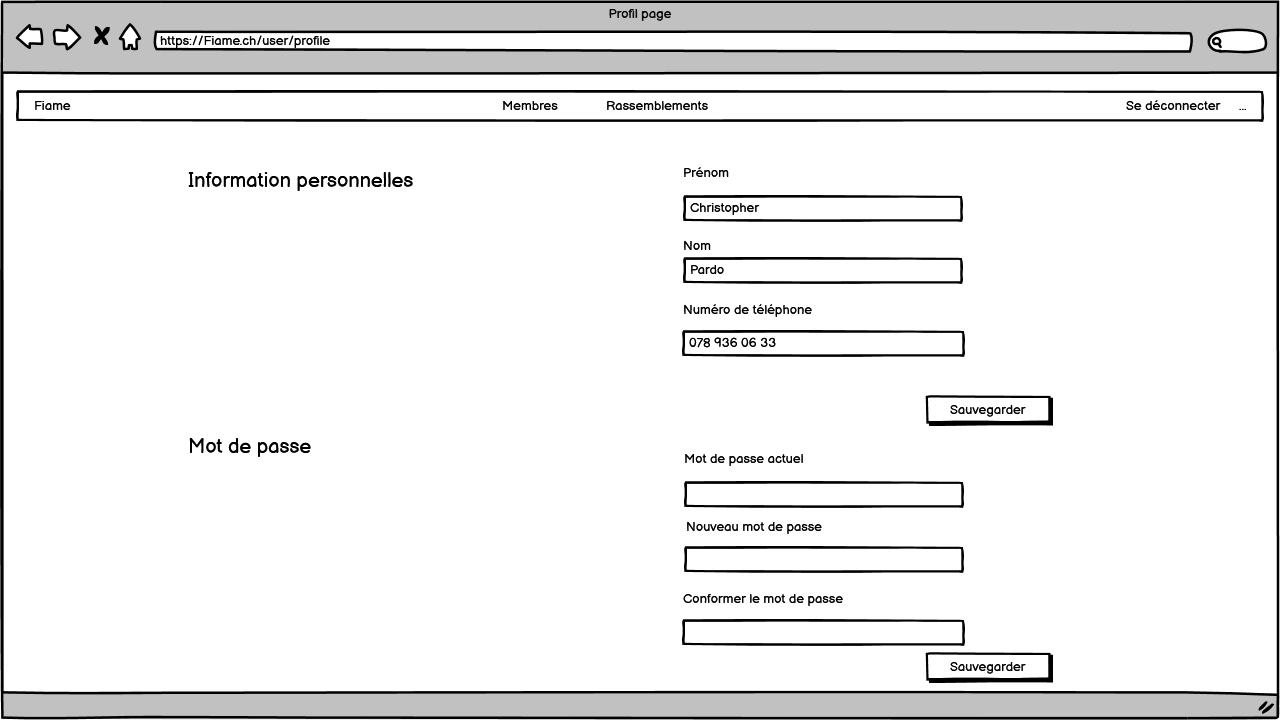


Figure 4 Maquette - Page de profile

#### Page d’évènements

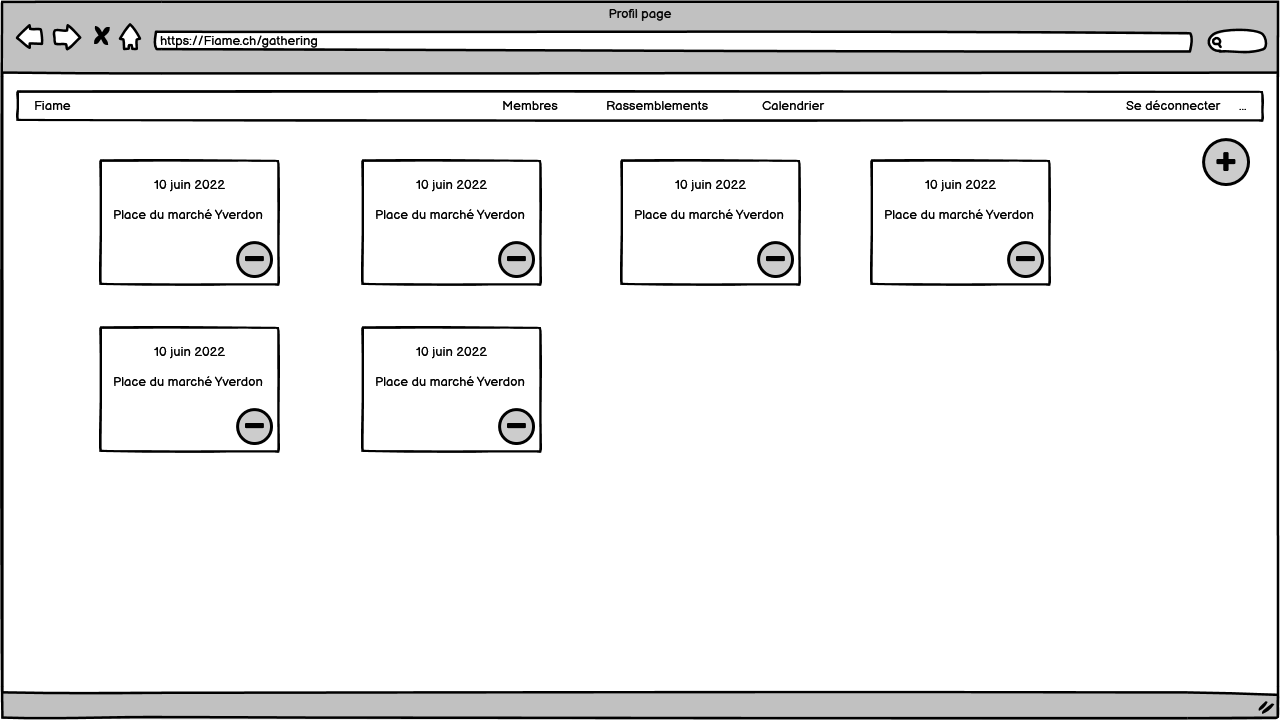


Figure 5 Maquette - Page d'évènement

## Stratégie de test

Au début de chaque sprint, des stories sont ajouter avec des tests d’acceptation très précis. Le chef de projet valide où non ces tests. *(Voir point 3.2)*

Les tests sont réalisés par mon camarades David Roulet avant la publication sur une instance en ligne.

Ils sont ensuite réalisés par Xavier Carrel (chef de projet) pendant la sprint review sur l’instance SwissCenter.

Les erreurs sont ensuite corrigées dans le sprint suivant.

## *Risques techniques*

Les risques techniques sont tout d’abord, le peu de connaissance sur le Framework. Effectivement je n’ai travaillé qu’une fois sur un projet en 2020 avec. Une grande partie du travail va donc être constitué de documentation sur ce Framework.

Un autre risque est que nous n’avons pas eu de cours au CPNV sur le Framework, ni sur les Framework en général. Donc tout le projet sera en autodidacte.

## Planification

* Sprint 1 : 02/05/2022 - 06/05/2022
  + Démarrage du projet
  + Mise en place de l'environnement
  + Planification
  + Création du MCD
  + Création du MLD
* Sprint 2 : 07/05/2022 - 17/05/2022
  + Création du site
  + Création de la base de données
  + Gestion des membres et des rôles
  + Gestion de son profile
* Sprint 3 : 18/05/2022 - 25/05/2022
  + Correction des problèmes de la gestion de son profile
  + Correction des problèmes de la gestion des membres
  + Gestion des évènements
  + Mise en ligne de la version web
* Sprint 4 : 26/05/2022 - 31/05/2022
  + Finalisation de la documentation
  + Testing

## Dossier de conception

### Matériel

Ordinateur : optiplex 7050

Processeur : Intel core I7-6700

Ram : 16GO

SSD : 477Go

Système d’exploitation : Windows 10 éducation

### Logiciels

* Navigateur de teste : Google chrome
* Outil de version : GitHub
* Editeur de code : Virtual Studio Code
* Client SQL : HeidiSQL
* Outil de MCD : Draw.io
* Outil de MLD : MySQL Workbench
* Logiciel de maquette : Balsamiq Studios
* Hébergement web : SwissCenter
* Gestion de projet : Icescrum
* Outil de traitement de texte : Microsoft Word
* Générateur de journal de travail : IceScrub (<https://github.com/XCarrel/IceScrub>)

### Outils de développement

* Langage : PHP 8.1.6
* Framework PHP : Laravel 9.11
* Framework CSS : Tailwindcss
* Framework JavaScript : Alpine JS
* Kit de développement : Jetstream (Livewire)
* Base de donnée : MySQL 8.0.29
* Gestion Git : GitFlow

# Réalisation

## Dossier de réalisation

Le projet prend la forme d’un MVC (Modèle-Vue-Contrôleur).

Cela signifie que ces trois parties sont séparée et individuelles

Le model vient gérer les données.

La vue est là pour tout ce qui est affiché à l’utilisateur

Quant au contrôleur, il vient effectuer les actions « logiques ». Les algorithmes et la sécurité sont géré par lui.

Ils sont respectivement dans les dossiers suivants :

* Vues : *resources/views*
* Contrôleurs : *app/Http/Controllers*
* Modèles : *app/Models*

Deux autres aspects techniques sont également :

* Les middlewares : Ils servent à autoriser centaines action à centaines personnes : *app/Http/Middleware*
* Les Migrations : Ils servent à créer une base de données sans utiliser de SQL : *database/migrations*

## Description des tests effectués

### Se connecter

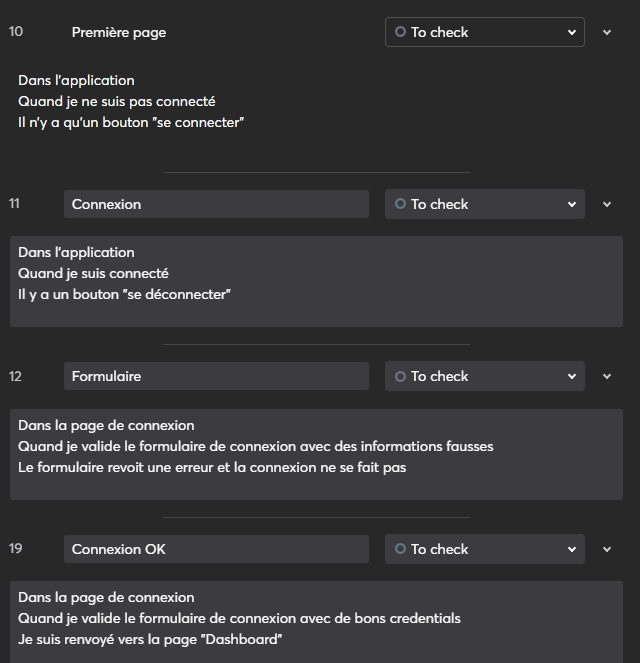


Figure 6 Testes - Se Connecter

### Modifier son profile

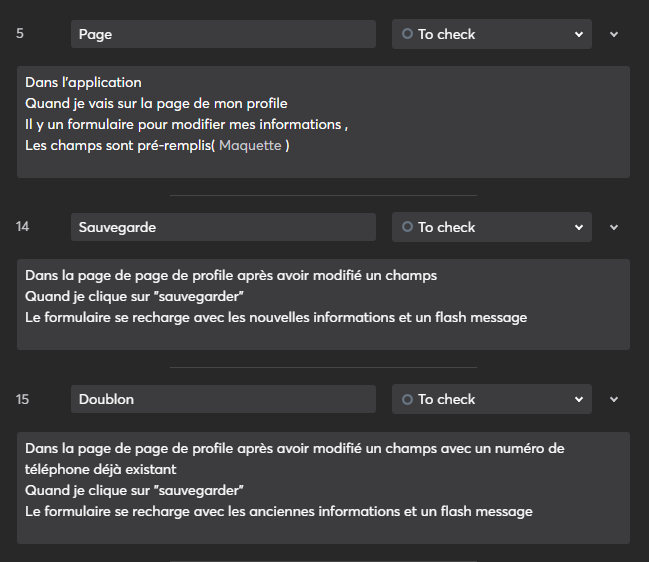


Figure 7 Testes - Modifier son profile (1)

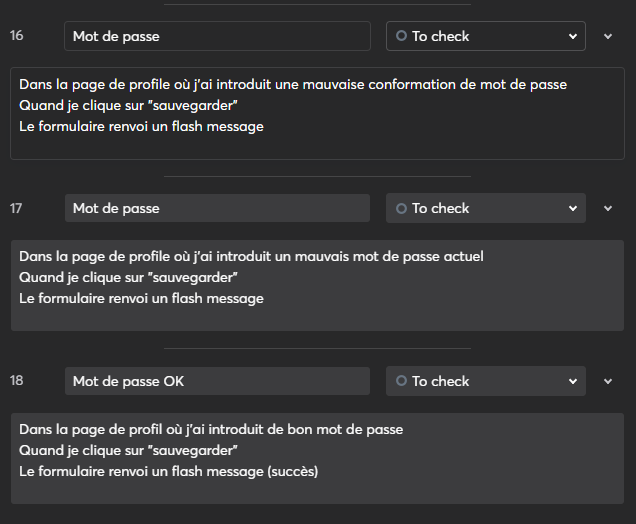


Figure 8 Testes - Modifier son profile (2)

### Nommer un administrateur

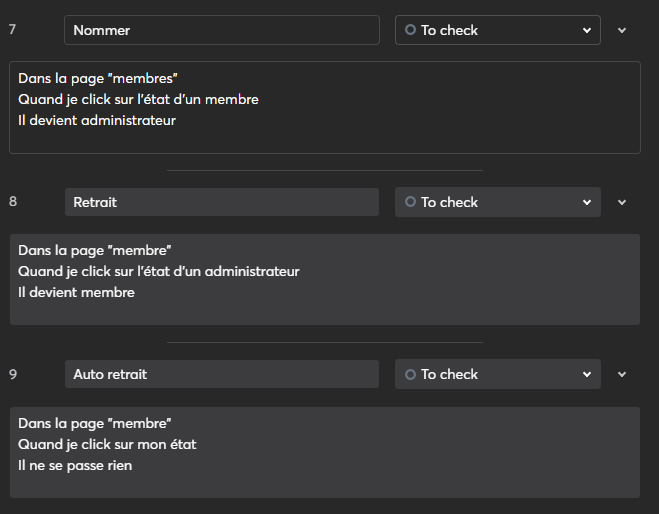


Figure 9 Testes - Nommer un administrateur

### Ajouter un nouveau membre

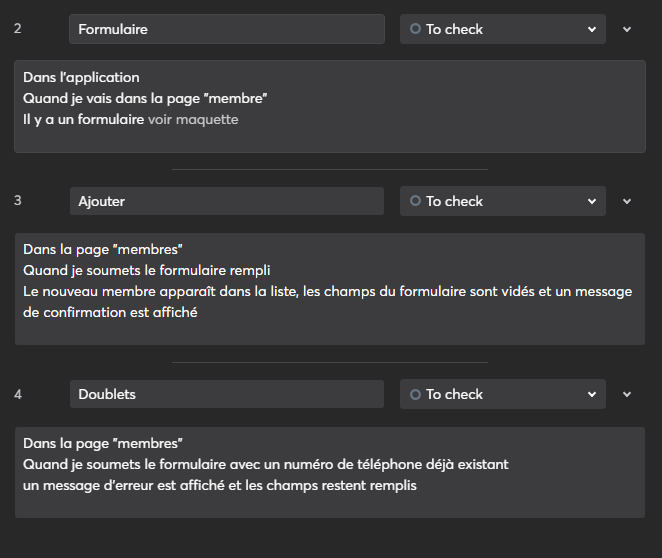


Figure 10 Testes - Ajouter un nouveau membre

### Consulter le calendrier

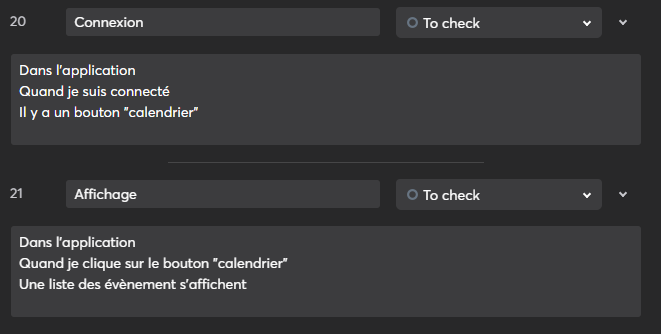


Figure 11 Testes - Consulter le calendrier

### Créer, modifier, supprimer un événement

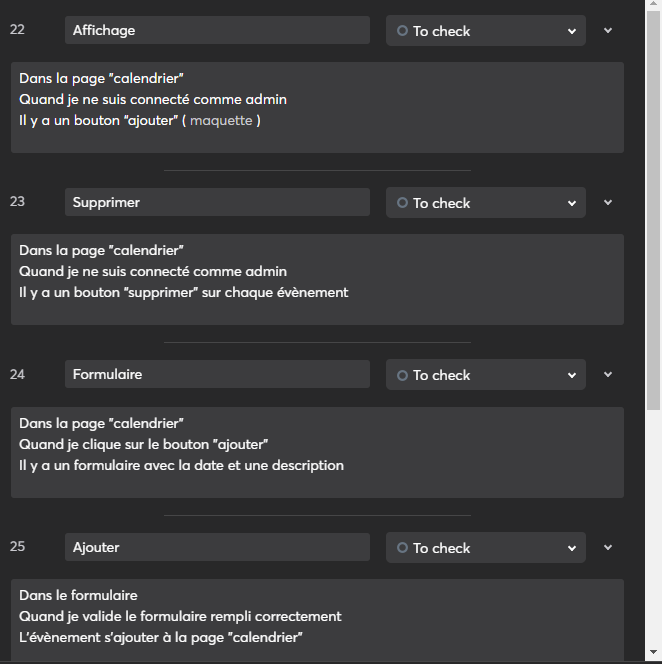


Figure 12 Testes - Gestion d'évènement (1)

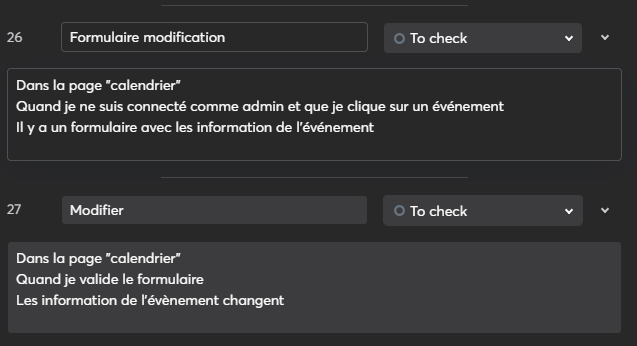


Figure 13 Testes - Gestion d'évènement (2)

## Erreurs restantes

* Il n’y a pas de message à l’ajout d’un membre
* Les numéros de téléphone ne prennent pas les espaces.
* Les noms ne prennent pas les caractères spéciaux.

## Liste des documents fournis

* Rapport de projet
* Repository git (https://github.com/ChristopherPardo/Fiame-Desk)
* Dossier du produit l’application

# Conclusions

## Objectifs atteints

* Une gestion de membre
* Un système de login
* Un calendrier de rassemblement
* Un système de modification de profils
* Un système d’administrateur

## Objectifs non atteints

* Un système de mise en vente de produit
* Une API dont le but est de communiquer avec une application mobile
* Un system de mise en vente d’objet par évènement

## Difficultés particulières

Une grande difficulté de ce projet a été la grande quantité de travail. Il y a eu beaucoup de tâche à réaliser dans le cahier des charges ainsi que dans icescrum.

Au fur et à mesure de l’avancement nous avons placé les priorités avec le chef de projet. Cela a été facilement fait, car comme nous avons utilisé la méthode Scrum, nous avons facilement pu nous concentrer sur certaines tâches et en délaisser d’autres.

## Suites possibles

Une suite possible est la création de l’API pour fonctionner avec l’application mobile développée en parallèle par un camarade pour son projet de TPI.

Une autre est de mettre en place le système de vente voulu à la base dans le projet.

# Annexes

## Résumé du rapport du TPI / version succincte de la documentation

### Situation de départ

Toutes les informations relatives au projet se trouve dans cette documentation.

Ce projet a pour but de créer une application web en Laravel (un Framework PHP).

Cette application doit convenir à une à une commuté cherchant à gérer leur membre pour des rassemblements et de vendre des produits.

Le projet s’inscrit dans la formation ICT pour l’obtention du CFC d’informaticien (TPI).

### Mise en œuvre

Il a d’abord fallu réfléchir la mise en place de l’application.

Les premières tâches ont été défini par le chef de projet, j’ai ensuite créé les tests d’acceptation de ces tâches. Ils ont été validés ou non par le chef de projet.

Les premiers documents techniques ont été réalisé en même temps (MCD et MLD).

Le début de la programmation à ensuite entamée.

En premier lui la mise en place d’une application vierge en laravel. J’ai pu alors commencer à créer ma base de données ainsi que mes premières données de test.

Une des premières étapes a été la gestion des membres avec différentes pages ainsi que la modification de son profile.

J’ai ensuite décidé de publier l’application du une instance SwissCenter pour éviter les problèmes lors des sprint review et avoir toujours le même système.

La prochaine étape a été la gestion des évènements.

### Résultats

Une application web a été rendue.

Cette application est disponible sur le web ainsi que publiquement sur mon repository GitHub. Elle est fonctionnelle et est exécutable.

Elle contient :

* Une gestion de membre
* Un système de login
* Un calendrier de rassemblement
* Un système de modification de profils
* Un système d’administrateur

## Sources – Bibliographie

Laravel : <https://laravel.com/docs/9.x>

Jetstream : <https://jetstream.laravel.com/2.x/introduction.html>

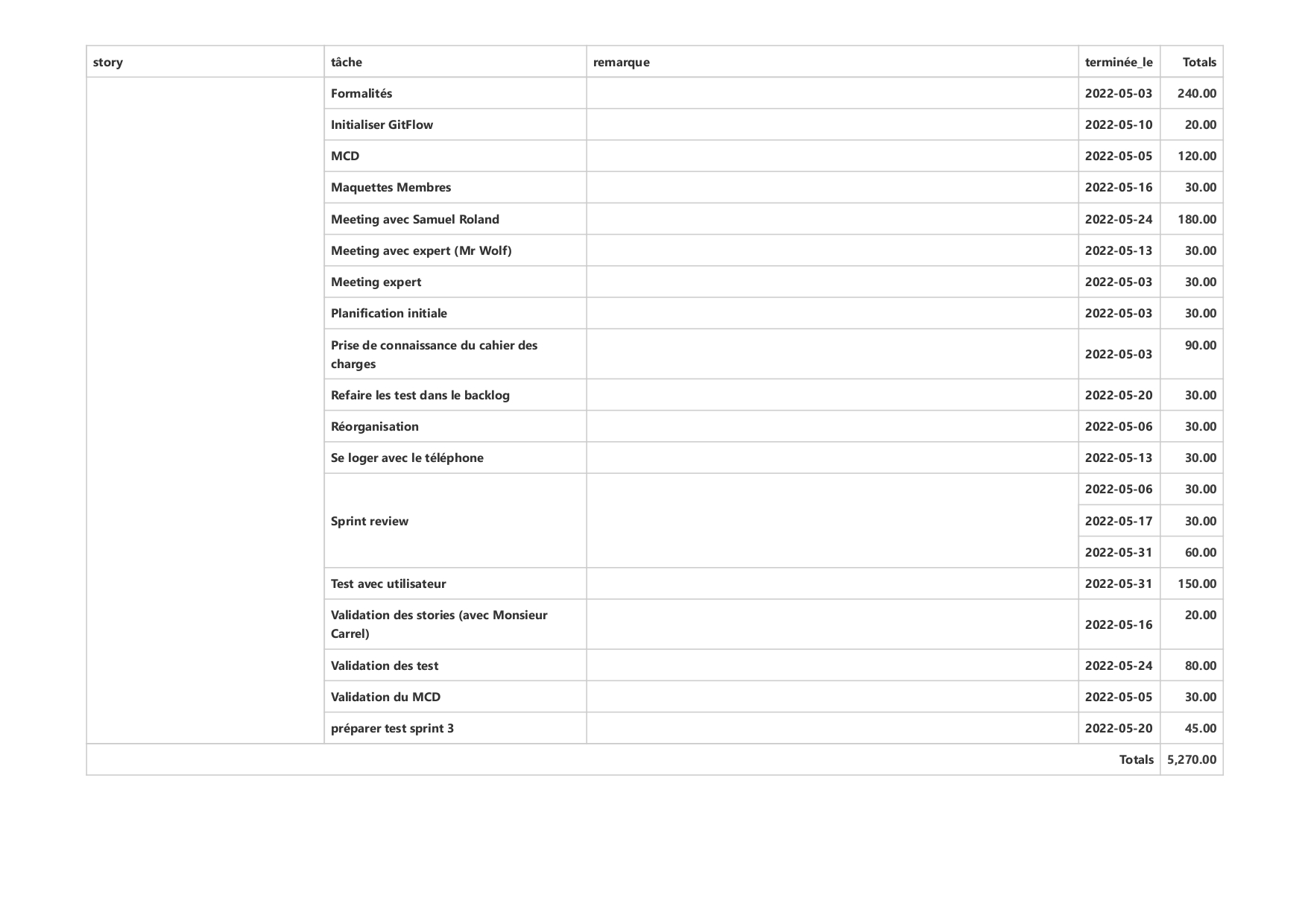
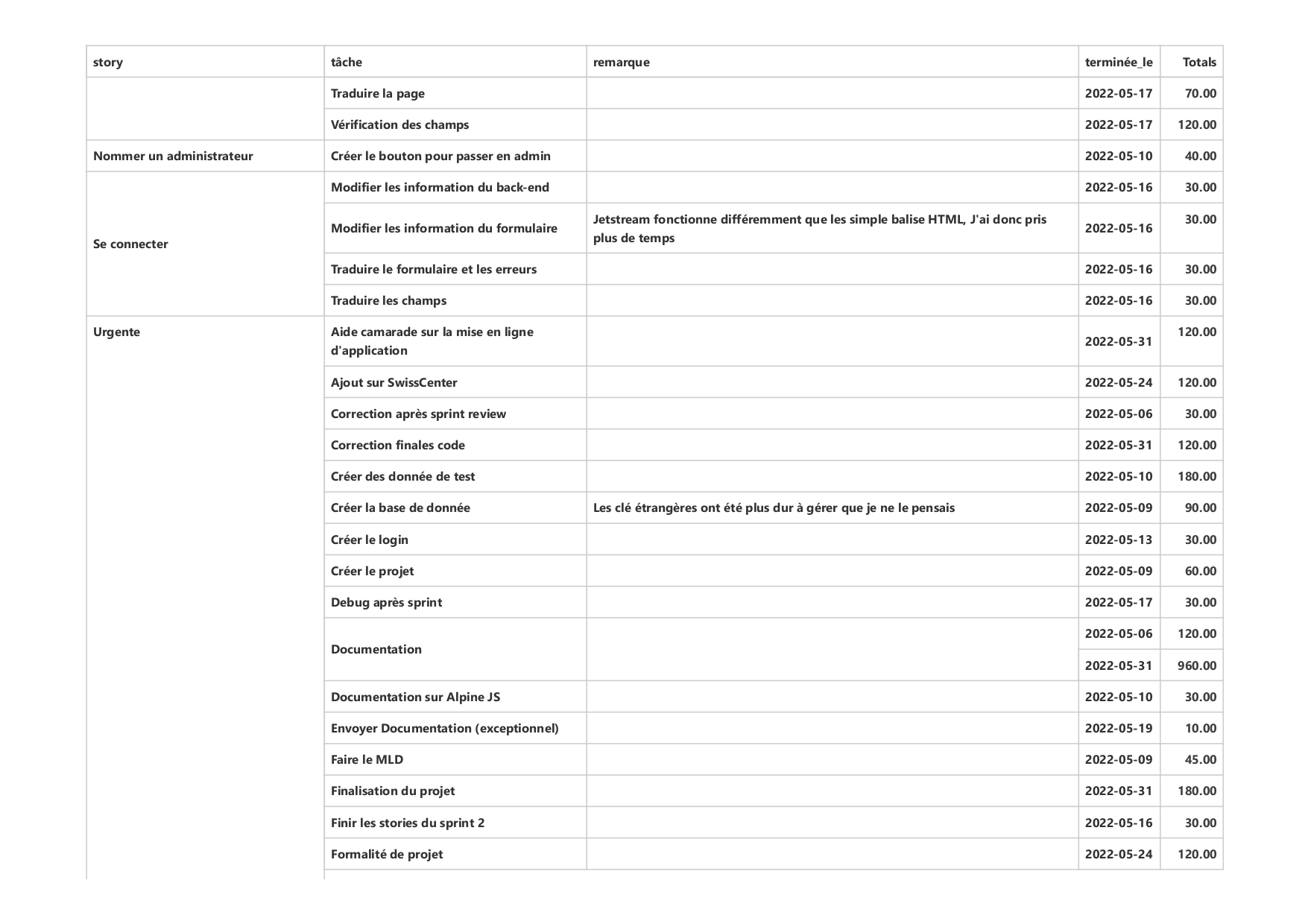
Alpine js : <https://alpinejs.dev>

Tailwindcss : <https://tailwindcss.com/docs/installation>

Mon camarade Samuel Roland m’a également aidé sur quelques bugs que j’ai pu rencontrer durant ce projet.

## 

## Journal de travail



## Manuel d'Installation

### Prérequis

* PHP 8.1.6
* MySQL 8.0.29
* Client Git
* Composer

### Installation

1. Créer une base de données MySQL
2. Ouvrir un terminal
3. *git clone* [*https://github.com/ChristopherPardo/Fiame-Desk.git*](https://github.com/ChristopherPardo/Fiame-Desk.git)
4. *cd Fiame-Desk\fiame-desk*
5. composer install
6. *npm install*
7. Modifier env.exemple en .env
8. Rentrer les informations de sa base de données

Example:

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=Fiame

DB\_USERNAME=root

DB\_PASSWORD=Pa$$w0rd

1. *php artisan key:generate*
2. *php artisan migrate:fresh –seed*
3. *npm run watch*
4. *php artisan serve*
5. Ouvrir sur navigateur *localhost:8000*

### Alternative

Se rentre sur la version publiée sur le web : <http://fiame.mycpnv.ch>

## Manuel d'Utilisation

### Utilisateurs de test

Admin :

Nom : Christopher Pardo

Tél : 0798765432

Mot de passe : Pa$$w0rd

Utilisateur :

Nom : David Roulet

Tél : 0723456789

Mot de passe : Pa$$w0rd

### Routes des pages

#### Non connecté :

* Page d’arrivée : */*
* Page de connexion : */login*

#### Connecté comme utilisateur

* Page principal : */dashboard*
* Page de rassemblement : */events*
* Page de gestion de profile : */user/profile*
* Déconnection : */logout*

#### Connecté comme admin

* Page de gestion de membre : */members*

# Table des images

[Figure 1 Modèle Conceptuel de donnée 10](#_Toc104884838)

[Figure 2 Modèle logique de donnée 11](#_Toc104884839)

[Figure 3 Maquette - Page de membres 12](#_Toc104884840)

[Figure 4 Maquette - Page de profile 13](#_Toc104884841)

[Figure 5 Maquette - Page d'évènement 14](#_Toc104884842)

[Figure 6 Testes - Se Connecter 17](#_Toc104884843)

[Figure 7 Testes - Modifier son profile (1) 18](#_Toc104884844)

[Figure 8 Testes - Modifier son profile (2) 19](#_Toc104884845)

[Figure 9 Testes - Nommer un administrateur 20](#_Toc104884846)

[Figure 10 Testes - Ajouter un nouveau membre 21](#_Toc104884847)

[Figure 11 Testes - Consulter le calendrier 22](#_Toc104884848)

[Figure 12 Testes - Gestion d'évènement (1) 23](#_Toc104884849)

[Figure 13 Testes - Gestion d'évènement (2) 24](#_Toc104884850)

# Glossaire

* API : Service permettant de récupérer des données
* Git : Logiciel de gestion de versions décentralisé
* Icescrum : Site de gestion de projet Scrum
* Instance SwissCenter : Serveur permettant de publier des sites sur le web
* Méthode agile : Méthode de travail permettant une grande communication entre le client et les équipes
* Repository GitHub : Serveur Git permettant de d’avoir son projet en ligne
* Scrum : Type de méthode agile fonctionnant avec des sprints
* Sprint review : Meeting de validation du travail effectué sur un certain temps
* Sprint : Temps de travail (plusieurs jours) dans lequel est attribué certaines tâches
* Stories : Fonctionnalité réduit en tâches
* Table (MCD) : Objet de base de donnée