



Lycée Saint Vincent

## Rapport de stage BTS2 : Conseil départemental du Val d'Oise

### Table des figures :

Figure 1: Hôtel départemental .....	3
Figure 2: Screenshot de Mosaik .....	4
Figure 3: SignalementResource .....	7
Figure 4: Modèle .....	8
Figure 5: Cas d'utilisation .....	11
Figure 6: Accueil Signes Backpack.....	12
Figure 7: Panel d'administration Signalement backpack.....	13
Figure 8: Formulaire signalement backpack .....	13
Figure 10: Visible.....	16
Figure 11: Log manager .....	16
Figure 12: Visuel Formulaire de Signalement.....	17
Figure 13: Code classe formuaire de signalement.....	17
Figure 14: Column.....	18
Figure 15: AfterStateUpdated .....	19
Figure 16: Méthode build de class Mailer .....	19
Figure 17: Observer .....	20
Figure 18: Page Widgets .....	21
Figure 19: FiltreWidget .....	21
Figure 20: Accueil Signes final .....	22

Table des matières :

## Table des matières

Introduction :	3
Contexte :	3
Présentation de l'entreprise :	3
Environnement de l'entreprise :	4
Glossaire :	5
Partie A : Contexte de la mission :	6
Nom de la mission :	6
Période :	6
Stack Technique :	6
Process :	9
Contexte de la mission :	9
Partie B : Déroulement :	13
Semaine 1 : Découverte de l'entreprise et du projet :	13
Semaine 2 : Correction Curl Error 60 + TP Filament :	14
Semaine 3 : Poursuite du TP et Développement :	15
Semaine 4 : Commencement du projet :	15
Semaine 5 : Développement Signalement :	16
Semaine 6 : Retour Curl Error 60 et améliorations front-end :	17
Semaine 7 : Améliorations suite à la review projet :	18
Semaine 8 (Semaine Finale) : Instauration de l'envoi automatique de mails et corrections finales :	19
<b>C / Bilan de la mission :</b>	<b>22</b>
Points négatifs :	22
Points positifs :	23
Bilan du stage :	23
Apports de stage perspectives 2025-2026 :	23

## Introduction :

### Contexte :

Pour cette seconde année de BTS SIO SLAM nous devons effectuer un stage de 8 semaines afin de remplir le quota de 10 semaines de stage permettant de valider le BTS. La recherche n'était pas évidente puisque EC2E, mon entreprise de l'année dernière ne pouvait pas me reprendre cette année par manque de place. Néanmoins mes parents ont un ami qui a travaillé à la mairie et qui a lancé ses contacts. Finalement j'ai réussi à obtenir une place au conseil départemental du Val d'Oise à Cergy grâce à lui.

### Présentation de l'entreprise :

Le Conseil départemental du Val d'Oise se situe à Cergy.



Figure 1: Hôtel départemental

Il s'agit d'une organisation structurée par Directions et métiers, couvrant tous les aspects liés aux services et compétences du Département : domaines social, culturel, éducation, administratif, routes, transport...

J'effectue mon stage à la direction des systèmes d'informations et du numérique comptant dans la direction adjointe des Ressources

Ma maîtresse de stage est Faiza BOUFHKAD, directrice de la direction des systèmes d'informations et du numériques. Toutefois, je travaille avec le chef de projet M. FOIRET et son apprentie Mme LERAY car ils s'occupent du

développement. Je travaille également avec la responsable du pôle Mme MILITARU et le data analyse M. BARROIS.

## Environnement de l'entreprise :

Mme MILITARU m'a expliqué que les équipes marchent avec Microsoft teams et Outlook. Ils utilisent ces outils afin de communiquer, s'envoyer des documents importants... J'ai également découvert que sur teams on avait un statut similaire à discord mais automatique. Ils s'adaptent en fonction de notre calendrier sur lequel il y a les réunions mais aussi en fonction de nos actions.

On peut être disponible en vert si on n'a rien sur le calendrier, indisponible en rouge si on est en réunion noter sur le calendrier, absent en gris si on n'est pas connecté, inactif en jaune si on n'a fait aucune action sur le poste depuis un certain temps, interdit en rouge avec une bande blanche quand on est en partage d'écran. Les ordinateurs du département Le conseil départemental possède également un intranet à haute sécurité.

Il est équipé d'une interface appelée Mosaic regroupant les informations et actualités sur l'activité en interne mais aussi les applications que les employés utilisent pour accomplir leurs tâches. Le midi nous pouvons aller manger à la cantine du Crous pour les employés du département.

Normalement on paye avec son badge mais en tant que stagiaire je n'ai pas cette possibilité et je dois payer par carte bancaire. 3 fois par semaine je vais manger là-bas avec mes collègues. D'ailleurs l'heure du déjeuner est à midi mais le retour est assez flexible, en général de 13H à 14H.

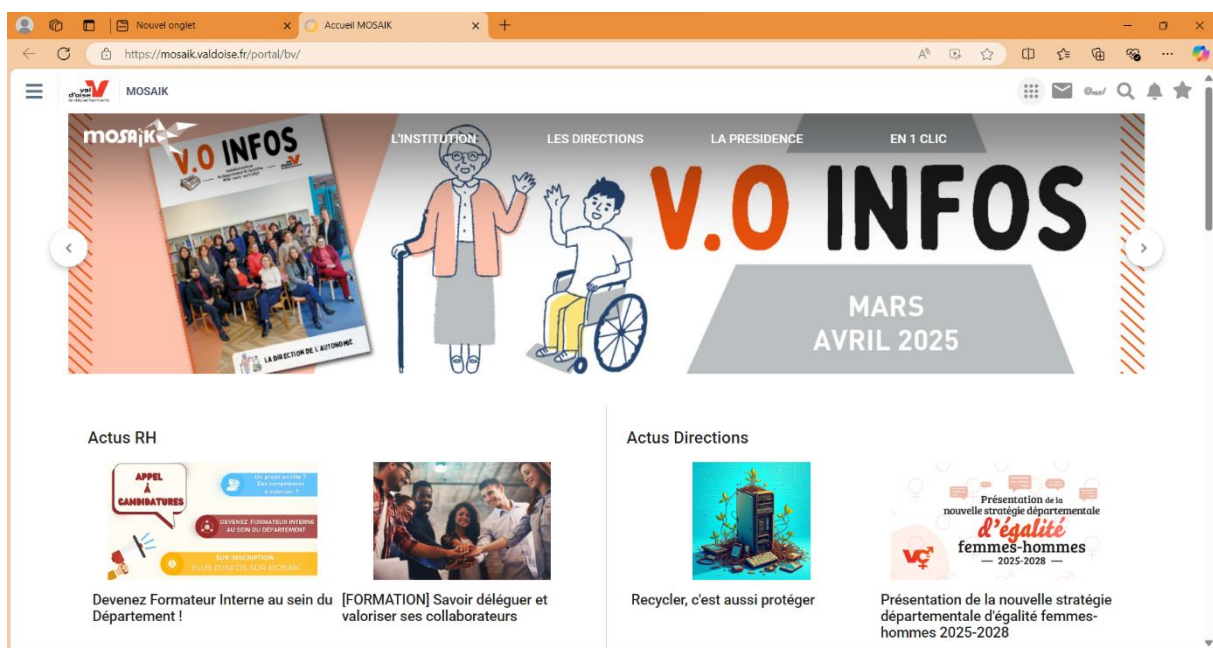


Figure 2: Screenshot de Mosaik

## Glossaire :

- **Backpack** : Autre générateur de CMS pour Laravel, permettant de créer des interfaces d'administration flexibles et personnalisables. Il offre des outils pour la gestion des entités et des ressources dans un projet web.
- **Event** : Un événement dans Laravel qui permet de gérer et d'écouter des actions spécifiques dans l'application. Les événements sont souvent utilisés pour déclencher des actions automatiques comme l'envoi d'un mail ou la mise à jour d'un modèle.
- **ESSMS : Établissements de Service Sociaux et Médico-Sociaux**. Ce terme désigne des établissements qui assurent des services dans le domaine social et médico-social.
- **Filament** : Framework open-source pour la création d'interfaces d'administration dans les projets PHP, particulièrement adapté pour des projets Laravel. Il sert de générateur de CMS et facilite la gestion des modèles, formulaires, tableaux, etc.
- **Framework** : Environnement de développement pour un projet web offrant une structure et une architecture définies, généralement en suivant le modèle **MVC** (Modèle-Vue-Contrôleur). Il facilite le développement rapide et l'organisation du code.
- **Générateur de CMS** : Package ou outil dans un framework permettant de créer automatiquement des panneaux d'administration et des fichiers où l'on définit des attributs d'entités, des colonnes, des champs, des filtres, des actions, etc.
- **Kernel** : La classe principale de Laravel qui gère les commandes, les tâches programmées, et l'exécution des processus dans l'application.
- **Mailable** : En Laravel, une **Mailable** est une classe qui permet de gérer l'envoi d'un email, en définissant son contenu, ses destinataires, et autres paramètres.
- **Migration** : Processus de transfert d'un projet web d'un environnement à un autre. Cela inclut la modification des données, de l'architecture ou des systèmes utilisés.
- **Observer** : Un design pattern dans Laravel permettant de surveiller et de réagir aux événements d'un modèle, comme la création, la mise à jour ou la suppression. Il est souvent utilisé pour automatiser des actions comme l'envoi de mails ou la mise à jour d'autres entités.
- **Schedule** : Méthode du **Kernel** dans Laravel permettant de planifier des tâches qui s'exécutent automatiquement à des intervalles de temps définis (par exemple, envoyer des emails tous les jours à 10h00).



- **Signalement** : Un signalement fait référence à une entrée ou un cas enregistré dans un système, souvent lié à un problème ou une demande nécessitant une action de suivi.
- **StateUpdated** : Une méthode ou fonction qui permet de vérifier et d'ajuster les valeurs des attributs ou champs d'une entité (par exemple, mettre à jour l'état d'un signalement après qu'un champ ait été modifié).
- **Trait** : Un ensemble de méthodes réutilisables qui peuvent être inclus dans des classes PHP. Les traits sont utilisés pour ajouter des fonctionnalités à plusieurs classes sans nécessiter d'héritage.
- **Widgets** : Petites interfaces interactives dans une application qui permettent d'afficher des informations de manière modulaire. Les widgets sont souvent utilisés pour afficher des tableaux de données, des formulaires, des graphiques, etc.

## Partie A : Contexte de la mission :

Nom de la mission : SIGNES 2.0

Période : 20 janvier 2025 – 7 mars 2025

Stack Technique : Laravel, filament

### Notes sur filament :

Vidéo : <https://youtu.be/JOPe7DvUq1Y>

Pour Filament, il faut l'installer via

Composer require filament/filament

Ensuite il faut le relier à une base de données via le .env et s'assurer d'avoir les tables de security bundle de symfony par exemple.

On peut ajouter des entités via la commande php artisan make :filament-user. Il faut ensuite ajouter les informations suivantes :

1. Nom de l'admin
2. Email de l'admin
3. Mot de passe de l'admin

Suite à cela on va sur l'url par défaut de notre site /admin/login afin d'accéder au dashboard.

Une fois sur le dashboard il faut créer des entités à administrer.

On fait php artisan make :filament-resource User

Après avoir fait cette commandes, le tableau apparaît vide, il faut remplir UserResource

```
class SignalementResource extends Resource
{
    public static function form(Form $form): Form
    {
        return $form
        ->schema([
            SignalementFields::getFields()->columns(1), //Appel de la classe des champs de formulaire
        ]);
    }

    public static function table(Table $table): Table
    {
        return $table
        ->columns(SignalementColumns::getColumns()) //Appel de la classe des colonnes
        ->headerActions([
        ])
        //->filters(FiltersSignalement::getFilters(), layout: FiltersLayout::AboveContent)
        ->actions([
            Tables\Actions\EditAction::make()->label('Modifier')
            ->visible(fn ($record ) => $record->etat != 'Fermé' ),
            Tables\Actions\DeleteAction::make()->label('Supprimer'),
        ])
        ->filters(SignalementFilters::getFilters(), layout: FiltersLayout::AboveContent) //Appel de la classe des filtres
        ->bulkActions([
            Tables\Actions\BulkActionGroup::make([ //Actions d'éléments sélectionnés
                Tables\Actions\DeleteBulkAction::make(),
                FilamentExportBulkAction::make('export') //Exportation
            ])
            ->label("Télécharger")
            ->fileName('signalements')
            ->defaultFormat('xlsx')
            ->defaultPageOrientation('landscape')
        ]),
        ->paginated([10,25,50,100,200,300])
        ->defaultSort('id','desc');
```

Figure 3: SignalementResource

Pour des entités plus classiques et pas géré automatiquement :

On fait php artisan make :model NomModel -m (-m sert à définir la migration en même temps) ;

On met les colonnes dans le fichier de migration

On met les colonnes dans le fillable du fichier Model



```
class ActionSignalement extends Model
{
    use HasRoles;
    use Encryptable;

    /*
    |-----
    | GLOBAL VARIABLES
    |-----
    */

    protected $guarded = ['id'];

    //Données cryptées
    protected $encryptable = [
        'question2',
        'reponse'
    ];

    /*
    |-----
    | RELATIONS
    |-----
    */

    public function signalement()
    {
        return $this->belongsTo(Signalement::class);
    }

    public function motif()
    {
        return $this->belongsTo(Option::class);
    }
}
```

Figure 4: Modèle

```

SignalmentColumns.php  SignalmentFields.php  DomicileSignalementCategorieNatureWidget.php M X
gnes > app > Filament > Widgets > DomicileSignalementCategorieNatureWidget.php > DomicileSignalementCategorieNatureWidget > getData
11 class DomicileSignalementCategorieNatureWidget extends ChartWidget
12
13 protected static ?string $pollingInterval = null ; //Empêche de recharger le widget pour le random color
14
15 protected function getData(): array
16 {
17     $rubs = Rubrique::all();
18
19     $annee = $this->filters['annee'] ?? null;
20     $directionId = $this->filters['direction'] ?? null;
21     $libelles = [] ;
22     foreach ($rubs as $ligne) {
23         $query = Signalement::join('secteurs', 'signalements.secteur_id', '=', 'secteurs.id')
24             ->join('rubriques', 'signalements.rub_nature1_id', '=', 'rubriques.id')
25             ->where('rubriques.id', $ligne->id)
26             ->where('secteurs.libelle', 'Domicile')
27             ->select('signalements.*')
28             ->get();
29
30         if (!empty($annee)) {
31             $query = $query->where('signalements.date_evenement', '>=', $annee, '-01-01')->where('signalements.date_evenement', '<', $annee, '-12-31');
32         }
33
34         if (!empty($directionId)) {
35             $query = $query->where('direction_id', $directionId);
36         }
37         $datas[] = $query->count(); //Ajoute les données count dans data
38         $colors[] = 'rgb(' . rand(0, 255) . ', ' . rand(0, 255) . ', ' . rand(0, 255) . ')'; //Random color pour les stats
39         $libelles[] = $ligne->libelle; //Les libellés des données
40     }
41     return [ //Retourne les tableaux
42         'datasets' => [
43             [
44                 'label' => 'Nb Signalement',
45                 'data' => $datas,
46                 'backgroundColor'=>$colors,
47                 'borderColor' => 'rgb(255, 255, 255)',
48             ]
49         ]
50     ];

```

**Process :** Bit bucket, github, VsCode, WAMP

### Contexte de la mission :

Le conseil départemental du Val d'Oise utilise une interface que leur intranet qui s'appelle Signes. C'est une interface de signalement d'incident dans les ESSMS (Etablissement de Service Sociaux et Médico-Sociaux ). Le principe est simple, quand un incident se produit dans un établissement, il faut qu'un membre du personnel face un rapport en remplissant le formulaire de signalement. Il note tous les détails, la description, les circonstances, les dispositions à prendre, l'impact médiatique, le suivi ... Par la suite le signalement est ouvert, d'autres membres du département peuvent créer des actions de signalement se traduisant par des questions posées dessus pour le suivi des choses, le signalement passe à l'état en cours. Si un signalement reste à l'état en cours 5 semaines après la date de l'incident, il passe à l'état relancé. Enfin un gestionnaire peut modifier le signalement pour le déclarer comme fermé, le signalement ne sera plus modifiable.

Pour créer cette application, le chef de projet a décidé d'utiliser un générateur de CMS appelé backpack. Problème aujourd'hui il s'avère que filament, un autre générateur de CMS est bien meilleur il offre la possibilité de faire des formulaires dynamiques d'adaptant aux réponses plus simple d'utilisation. De plus entre temps,

backpack est devenu payant. La mission est donc de reprendre le projet signes et de remplacer backpack par filament afin de créer SIGNES 2.0 .

Signes est hébergé sur un répertoire bitbucket (concurrent de github). Toutefois, M. FOIRET n'a pas pu me créer de branches donc j'ai utilisé un répertoire github.

Sur un plan technique :

### **Signes :**

Signes est une application web pour gérer les ESSMS (Établissements et Services Sociaux et

Médico-Sociaux ) et permet surtout de suivre es signalements liés à des incidents se produisant

dans ses établissements.

Nous avons un total de 12 entités :

- Utilisateurs
- Rôles
- Signalements
- Actions Signalements : Question sur un signalement
- Etablissements : ESSMS
- Catégories : Catégories d'établissements
- Secteurs : Secteurs d'établissements
- Options : Réponses pour les formulaires de signalements et d'action signalement
- Rubriques : Rubriques d'options
- Sections : Sections d'options
- Faq : Question d'aide en ligne
- CatFaq : Catégories de questions d'aides en ligne
- 

Voici le schéma relationnel de la base de données

Voici les cas d'utilisations :

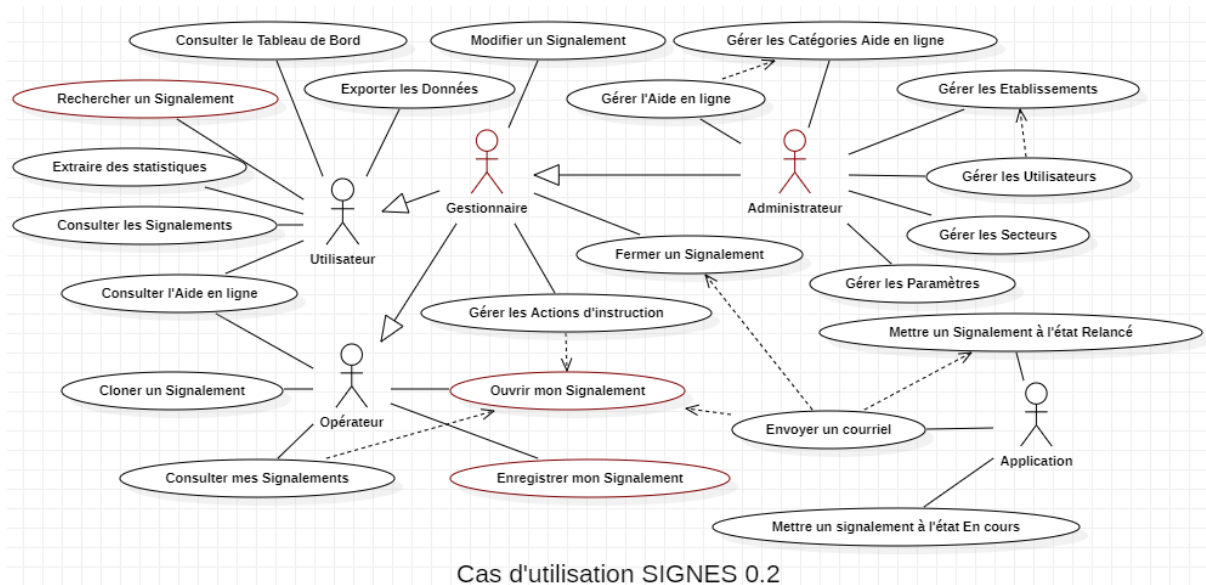


Figure 5: Cas d'utilisation

Au niveau des fonctionnalités de Signes nous avons :

- Administration de toutes les entités sauf communes, rubriques et sections
- Exportation des logs
- Exportation xlsx
- Widgets statistiques sur les signalements
- Authentification

Particularités :

Le formulaire de signalement doit être dynamique. Il doit s'adapter aux réponses de l'utilisateur :

- Quand on sélectionne un secteur, le select de etablissement doit nous proposer des établissements du secteur sélectionner
- Quand on sélectionne une catégorie de nature des faits, on doit avoir dans le select de nature des faits des réponses de cette catégorie.
- Quand on sélectionne autre, un champ Si Autre précise doit apparaître.

L'entité signalement possède deux champs dynamiques : complet et état :

- Le champ complet est lié à l'enregistrement de saisie. Si l'utilisateur enregistre sa saisie en laissant des champs requis null, alors complet est égal à false, sinon il est égal à true.
- Le champ etat regroupe les 5 états d'un signalement
  - o Ouvert : Le signalement est complet et créer
  - o En cours : une action de signalement a été créée sur un signalement
  - o Relancé : 5 semaines après la date de l'évènement, s'il est à l'état en cours, il passe à relancer
  - o Réceptionné : Un gestionnaire peut lire le signalement
  - o Fermé : Un gestionnaire a modifié le signalement et n'est plus modifiable

- La désactivation : Certaines entités tels que utilisateurs ou catégorie possède un champ actif plus tard remplacé par deleted\_at. Le but est de pouvoir les restaurer via une action dans le tableau des entités du panneau d'administration.

Visuel :

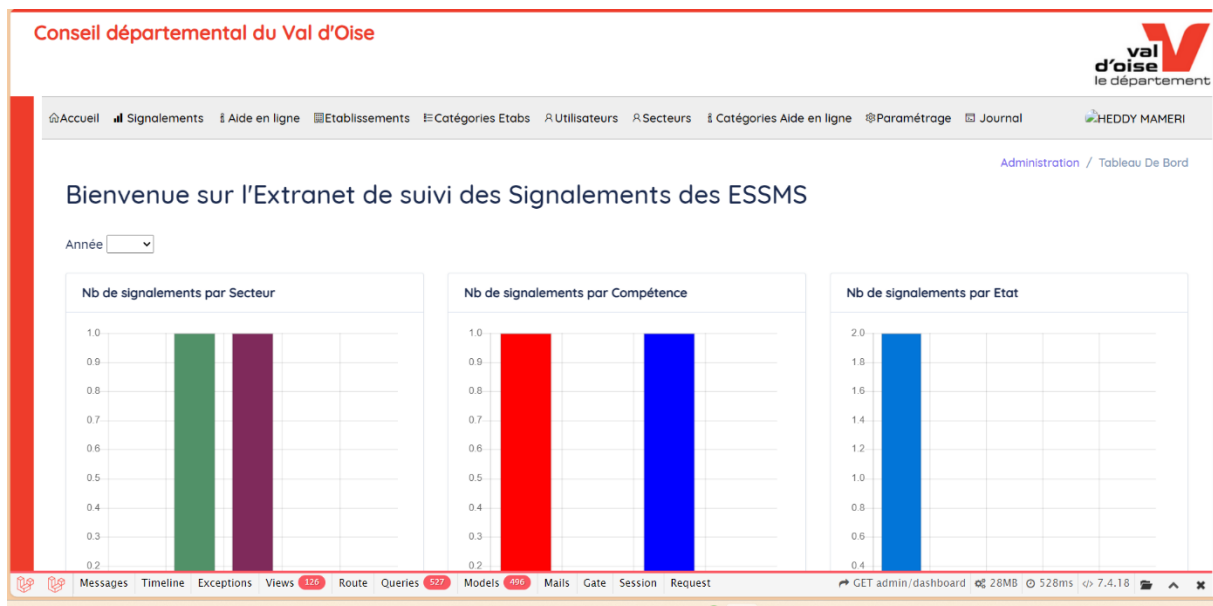


Figure 6: Accueil Signes Backpack

Conseil départemental du Val d'Oise

Administration / Signalements / Liste

## Signalements

Affichage des éléments 1 à 10 sur 30. [Réinitialiser la session](#)

[+ Ajouter un Signalement](#)

Recherche...

N°	Date évènement	Secteur	Etablissement	Etat	Complet	Actions
1	20 juin 2023	Personnes Handicapées	LE BON LOGIS	Réceptionné	Oui	<a href="#">Modifier</a> <a href="#">Supprimer</a>
2	17 nov. 2022	Domicile	A DOM	Fermé	Oui	<a href="#">Modifier</a> <a href="#">Supprimer</a>
3	27 déc. 2023	Personnes Handicapées	LE BON LOGIS	Relancé	Oui	<a href="#">Modifier</a> <a href="#">Supprimer</a>
4	22 janv. 2024	Personnes Handicapées	LE BON LOGIS	Fermé	Oui	<a href="#">Modifier</a> <a href="#">Supprimer</a>
5	21 nov. 2021	Personnes Agées	LOUIS GRASSI	Relancé	Oui	<a href="#">Modifier</a> <a href="#">Supprimer</a>
6	18 nov. 2022	Personnes Agées	CHABRAND THIBAUT	Relancé	Oui	<a href="#">Modifier</a> <a href="#">Supprimer</a>

Figure 7: Panel d'administration Signalement backpack

Administration / Signalement / Ajouter

## Signalement

Ajouter un Signalement.

Etablissement- Déclarant    Faits- Victimes    Conséquences- Mesures    Information- Dispositions    Suites- Répercussions    Suivi

### Nature des faits

Les exemples cités dans les catégories ci-après ne constituent pas une liste exhaustive, mais ont vocation à aider la structure à identifier les dysfonctionnements et les événements qui relèvent de l'article L. 331-8-1 du CASF

Catégorie Nature des Faits principale \*

Sélectionner une catégorie

Nature des Faits principale \*

Sélectionner une nature

Si Autre, précisez

Catégorie Nature des Faits secondaire

Sélectionner une catégorie

Nature des Faits secondaire

Sélectionner une nature

Si Autre, précisez

Figure 8: Formulaire signalement backpack

## Partie B : Déroulement

### Semaine 1 : Découverte de l'entreprise et du projet

#### Jour 1 : Accueil et découverte de l'environnement

Je récupère mon badge, découvre mon espace de travail et rencontre l'équipe. Mme Obry

me présente l'entreprise. Lors de la réunion avec M. FOIRET, je découvre les projets EvalPro et Cartotek. Mon objectif principal sera la migration de Cartotek, une application Laravel, de Backpack à Filament. Je vais chercher mon matériel dans le bâtiment G. L'entreprise privilégie Microsoft Teams et permet le télétravail. M. FOIRET et Mme MILITARU ne sont pas disponibles les mercredis et vendredis.

### **Jour 2 : Rencontre avec M. FOIRET et découverte du projet**

M. FOIRET m'explique comment sont gérés les projets : petits et moyens projets en interne, grands projets sous-traités. Il supervise une trentaine de projets. Je me connecte au dépôt Bitbucket de Cartotek et commence par étudier la documentation (MLD, fonctionnement). Cela me permet de mieux comprendre le projet avant de commencer la migration vers Filament.

### **Jours 3-5 : Premiers défis techniques et prise en main des outils**

L'installation des outils me pose problème :

- Visual Studio Code et Git Bash sont installés sans souci.
- WAMP, PHP et Composer nécessitent des droits administrateurs que je n'ai pas. J'essaie une version portable de WAMP (Uwamp), mais Composer ne fonctionne pas. Une demande d'accès administrateur est refusée. Je demande au service IT d'installer les logiciels nécessaires. En attendant, je me forme en autonomie : lecture de la documentation de Backpack et Filament, tutoriels YouTube, tests sur mon PC personnel, mais une erreur SSL/TLS bloque l'installation des dépendances. Je mets ce problème de côté et commence un projet Symfony.

### **Bilan de la première semaine**

Cette première semaine a été marquée par la découverte de l'entreprise et du projet Cartotek. Bien que j'aie rencontré des blocages techniques, j'ai progressé dans la compréhension du projet et des outils. Je suis prêt à commencer le développement dès que les problèmes d'installation seront résolus.

---

## **Semaine 2 : Correction Curl Error 60 + TP Filament**

### **Jour 1 à 4 : Correction de Curl Error 60**

Le problème de Curl Error 60 persiste. Je m'informe que la migration sera faite sur Signes, et non Cartotek. Je m'attaque à l'erreur SSL en tentant diverses solutions (réinstallation des certificats, génération de certificats auto-signés, etc.). Je découvre que certains certificats sont manquants ou expirés. Après plusieurs tentatives infructueuses, je réinstalle PHP et Composer via Scoop, ce qui résout le problème.

### **Réunions et événements**

Je participe à diverses réunions, dont la présentation de BASTION (solution numérique) et le bilan annuel des pôles. J'échange informellement lors de la galette des rois avec l'équipe et les alternants du pôle géodata.

### **Jour 5 : Mise en place du projet Filament-Characters**

Je crée un dépôt GitHub pour le projet Filament-Characters et initialise la base de données avec les tables Classe, Grade et Character. Je génère l'interface d'administration et configure les relations entre les entités. L'affichage et l'ajout fonctionnent immédiatement, et la suppression est activée. J'intègre un système de couleurs pour les grades, inspiré d'un projet précédent.



### **Bilan de la semaine**

La correction du problème SSL et la mise en place de Filament-Characters ont marqué des avancées importantes. Je suis maintenant plus à l'aise avec les outils et techniques utilisés.

---

## **Semaine 3 : Poursuite du TP et Développement**

### **Jour 1 : Ajout de filtres et personnalisation des grades**

Je me concentre sur l'ajout de filtres à Filament-Characters et personnalise l'affichage des grades avec des couleurs générées automatiquement. Après quelques essais, l'affichage fonctionne correctement.

### **Jour 2 : Mise en place de l'export et amélioration du projet**

J'ajoute une fonctionnalité d'export XLSX et règle les erreurs liées au téléchargement. M. Foiret me fournit un modèle d'application (QUALIAC) qui m'aide à organiser mon projet et intégrer l'export.

### **Jour 3 : Correction des erreurs d'export et installation des logiciels**

Je corrige les erreurs liées à l'export et continue l'installation des logiciels nécessaires. L'après-midi, je participe à une réunion Bastion et j'améliore le code de mon TP en appliquant le principe SOLID pour le rendre plus structuré.

### **Jour 4 : Réunion Bastion et amélioration du TP**

Je mets en place des filtres supplémentaires et améliore l'organisation du code en classes dédiées. Je veille à appliquer le principe SOLID, surtout la Responsabilité Unique.

### **Jour 5 : Ajout de statistiques et projet personnel**

Je mets en place des widgets statistiques dans Filament-Characters à l'aide de Chart.js. Je rencontre des problèmes de couleur mais réussis à obtenir un affichage fonctionnel. L'après-midi, je commence un projet personnel avec Power BI pour analyser des données financières.

### **Bilan de la semaine**

J'ai fait des progrès considérables dans le projet Filament-Characters, notamment avec l'ajout de filtres, l'exportation et les statistiques. Je suis plus autonome et maîtrisé mieux les outils utilisés.

---

## **Semaine 4 : Commencement du projet**

### **Jour 1 : Suppression de Backpack et mise à jour de PHP**

Je clone le dépôt Signes et mets à jour PHP de 7.4.27 à 8.4.3. J'élimine les dépendances Backpack et lance un composer update. Le projet affiche une erreur 404.

### **Jour 2 : Migration du projet et réunion avec JL**

Je commence la migration du code source de Qualiatic et adapte le projet à Signes. JL me présente les Business Objects et les ETL, ainsi que le rôle des ODS dans la gestion des données.

### **Jour 3 : Reprise de Filament et mise en place des entités**

Je redémarre le projet Filament et ajoute les modèles, migrations et seeders. La migration de la base de données Signes fonctionne, et j'intègre un jeu de données test.

#### Jour 4 : Avancées sur l'administration et gestion des rôles

Je crée des ressources et des filtres pour les entités et m'attaque à la gestion des rôles. Christophe m'aide à résoudre des problèmes liés à la table model\_has\_role.

#### Jour 5 : Débogage et finalisation du formulaire de signalement

Je corrige une erreur 403 liée à l'URL et finalise le formulaire de signalement avec 85 champs, divisés en plusieurs étapes via Filament/Wizard.

```
Forms\Components\TextInput::make('information_autre')
->label('Si Autre, précisez')
->visible(fn ($get) => $get('information') === 'Autre' ), //visible uniquement si Autre est sélectionné dans information
```

Figure 9: Visible

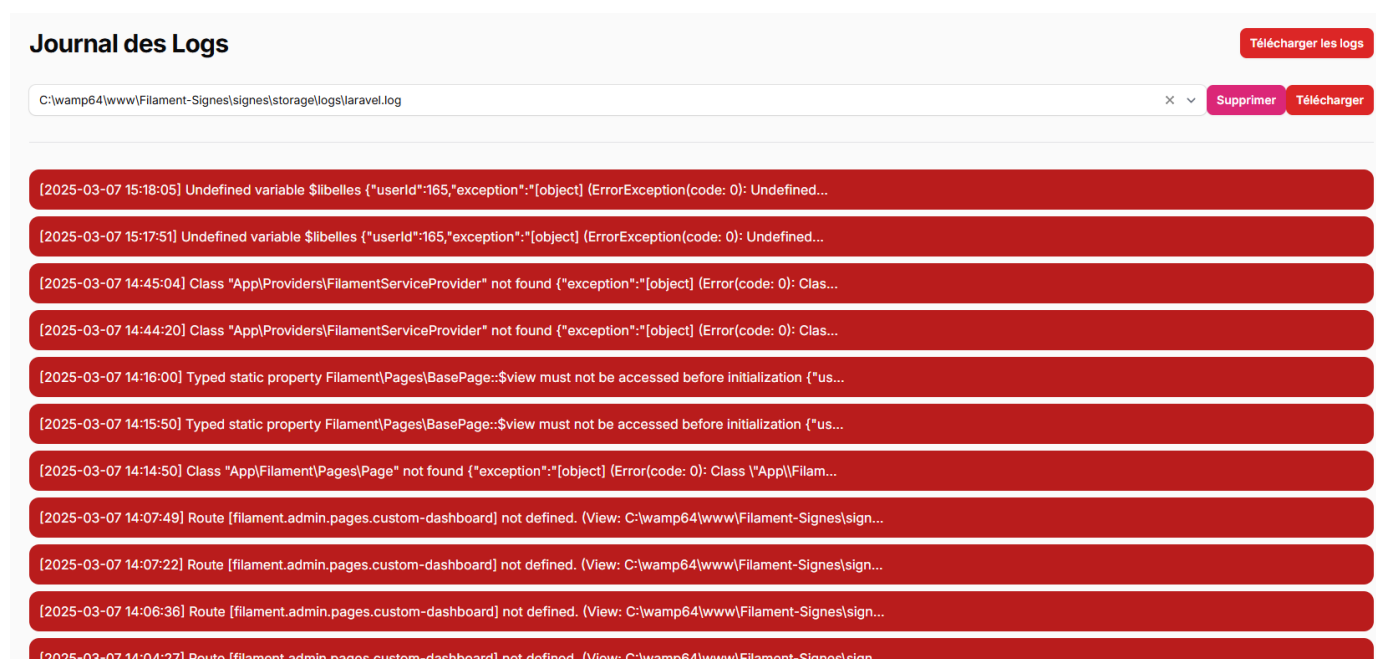


Figure 10: Log manager

## Semaine 5 : Développement Signalement

#### Jour 1 à 5 : Problèmes et ajustements sur le formulaire de signalement

Je résous des problèmes avec le formulaire de signalement, comme des erreurs de type sur certains champs. J'intègre aussi l'export dans Filament-Characters et je travaille sur la personnalisation de l'interface.

Figure 11: Visuel Formulaire de Signalement

```
class SignalementFields
{
    protected function getAutre($champ, $get): bool
    {
        return in_array('Autre', $get($champ, [])) ;
    }

    public static function getFields()
    {
        return Fieldset::make('SIGNALEMENT')->schema([ //Fieldset Crée un encadrage de formulaire
            Tabs::make('Signalement') //Tabs sert à avoir un formulaire en plusieurs onglets
            ->tabs([
                Tab::make('Etablissement-Déclarant') //Définit un onglet de formulaire
                ->schema([
                    Forms\Components\Grid::make(3)
                    ->schema([
                        Forms\Components\DateTimePicker::make('date')
                        ->label('Date et heure du signalement')
                        ->default((new \DateTime())->format('Y-m-d H:i:s'))
                        ->required(),

                        Forms\Components\DateTimePicker::make('date_evenement')
                        ->label('Date et heure de l\'événement')
                        ->default((new \DateTime())->format('Y-m-d H:i:s'))
                        ->helperText('Si cette date n\'est pas connue, merci de saisir la date et heure du signalement')
                        ->required(),
                    ])
                ])
            ])
        ])
    }
}
```

Figure 12: Code classe formulaire de signalement

## Semaine 6 : Retour Curl Error 60 et améliorations front-end

### Jour 1 à 3 : Problèmes de Curl Error 60 et améliorations sur les formulaires

Je continue de rencontrer des problèmes avec l'erreur Curl Error 60, mais j'avance sur la relation entre signalement et action\_signalement. Je mets en place une architecture ModelColumns pour toutes les entités administratives.

```

class SignalementColumns
{
    public static function getColumns(): array {
        return [
            //Tableau des colonnes du modèle
            Tables\Columns\TextColumn::make('id')
            ->label('N°')
            ->searchable() //Rend possible la recherche des données de la colonne dans la barre de recherche
            ->sortable() //Permet de trier par ordre croissant/décroissant
            ->wrap(), //Rend la colonne responsive
        ];
    }
}

```

Figure 13: Column

### Jour 4 à 5 : Finalisation et améliorations supplémentaires

Je corrige les labels et le design du formulaire de signalement, ajoute un "soft delete" et crée des widgets. Je termine la semaine en ajustant l'affichage des colonnes et en ajoutant des énumérations pour certains champs.

## Semaine 7 : Améliorations suite à la review projet

**Jour 1 :** Je classe les tâches par priorité, en commençant par la création de l'**ActionRelationManager**. Après quelques tests, je découvre que ses champs sont des copies de ceux de Ressource ou Fields/Columns, mais je ne comprends pas leur utilité. N'ayant pas de réponses de M. FOIRET (en congé), je passe aux relations managers de **Option**, **Faq** et **Etablissement**. Je remarque un comportement étrange dans le formulaire d'Etablissement : le champ readonly **Territoire** se remplit automatiquement sans que je comprenne l'origine des libellés. Ensuite, je consulte la documentation pour l'envoi de mails : prévu pour être géré par Backpack, mais je dois trouver son équivalent sous **Filament**. Je remplace également **ActiveBaseModel** par la classe **Model** de Filament, ce qui génère une erreur « The payload is invalid ». Après nettoyage du cache et de la session, je suspecte que le problème vient du décryptage des champs cryptés.

**Jour 2 :** Je me concentre sur la correction de l'erreur de payload. Après avoir ajusté les méthodes de décryptage, cela fonctionne. Je me penche sur l'enregistrement de saisie et trouve une solution Filament pour sauvegarder un brouillon, mais rencontre des erreurs de compatibilité. Je reviens donc à la solution de décryptage des champs. Je découvre dans le répertoire **Traits** que des traits liés au cryptage sont déjà présents. Une fois la méthode correcte appliquée, le décryptage fonctionne.

**Jour 3 :** Je corrige l'apparition des champs facultatifs dans le formulaire de signalement et ajoute des vérifications pour éviter l'erreur « payload is invalid ». Je travaille aussi sur l'enregistrement de saisie avec la méthode **SaveDraft** de Filament et ajuste les attributs **etat** et **complet**. Enfin, je crée une issue GitHub pour l'envoi de mails, fonctionnalité que je vais implémenter prochainement.

**Jour 4 :** Je termine les tests d'enregistrement et crée une nouvelle branche pour l'envoi des mails. Je découvre qu'il n'y a pas de méthode prête à l'emploi dans Filament, donc je choisis d'utiliser **Laravel** pour créer un Mailable et un Event pour l'envoi de mails. Après des tests infructueux, je réalise que le problème vient de la configuration du fichier **.env** pour l'intranet du département.

```

->afterStateUpdated(function (callable $set, callable $get) {
    $etablissementId = $get('etablissement_id'); //retourne l'id de l'établissement sélectionné
    if ($etablissementId) {
        $communeId = Etablissement::find($etablissementId)->commune_id; //Trouve l'id de la commune correspondante
        $set('commune_id', $communeId); //Incrémente la valeur communeId au champ caché commune
    }
});

```

Figure 14: AfterStateUpdated

```

public function build()
{
    $subject = $this->signalement->etablissement->nom." - Relance signalement N° ".$this->signalement->id." du ".date('d-m-Y H:i:s', strtotime($this->signalement->date));

    $expediteur = '';
    $destinataire = '';
    $destinataires_cc = array();

    $expediteur = 'test.valdoise@gmail.com' ;
    $destinataire = 'test.valdoise@gmail.com' ;
    $destinataires_cc = 'test.valdoise@gmail.com' ;
    return $this
        ->to($destinataire)
        ->cc($destinataires_cc)
        ->from($expediteur)
        ->subject($subject)
        ->markdown('emails.signalement_relance', ['signalement' => $this->signalement]);
}

```

Figure 15: Méthode build de class Mailer

**Jour 5 :** Je finalise les tests d'enregistrement et merge les branches. Je crée une nouvelle branche pour l'envoi de mails et effectue des ajustements, mais la configuration du **.env** pour l'intranet reste à finaliser la semaine prochaine.

#### Tâches pour la semaine prochaine :

- Configurer l'envoi de mails sur l'intranet du département.
- Ajouter les automatisations d'envoi de mails.
- Configurer l'API et vérifier les requêtes des widgets.
- Comparer le formulaire de signalement actuel avec celui du modèle en ligne.
- Vérifications générales de toutes les fonctionnalités.

## Semaine 8 (Semaine Finale) : Instauration de l'envoi automatique de mails et corrections finales

**Jour 1 :** Je crée un bouton pour envoyer des mails directement et mets en place un **observer** pour automatiser l'envoi de mails. Cela fonctionne pour **SignalementOuvert**.

```

class ActionSignalementObserver
{
    /**
     * Handle the ActionSignalement "created" event.
     */
    public function created(ActionSignalement $actionSignalement): void
    {
        //
        //dd($actionSignalement->signalement->etat = "En cours") ;
        $actionSignalement->signalement->etat = "En cours" ;
        $actionSignalement->signalement->save() ; //Sauvegarde la modif
        try {
            Mail::to('test.valdoise@gmail.com')->send(new ActionQuestion($actionSignalement)); //Appel le mailable
        } catch (\Exception $e) {
            $e->getMessage() ;
        }
    }
}

```

Figure 16: Observer

**Jour 2 :** Je travaille sur **ActionQuestion** et **SignalementFerme**, puis sur **SignalementRelance**, une fonctionnalité plus complexe qui se déclenche automatiquement après 5 semaines. J'utilise la classe **Kernel** et crée une méthode **schedule**. Je teste également des ajouts de colonnes **hidden** sur **User** et **Faq**.

#### Tâches restantes :

- Tester **SignalementRelance**.
- Utiliser l'observer pour vérifier les champs **required**.
- Vérifier les valeurs retournées par les widgets.
- Effectuer une vérification générale du système et corriger les anomalies.

**Jour 3 :** Je crée une branche **Correction** pour les modifications des widgets. Les tâches restantes incluent l'ajout de **HelperText**, la gestion des IDs automatiques dans **ActionSignalement**, l'ajustement de la largeur des éléments, l'implémentation de restrictions de rôle et l'application de filtres sur les widgets.

**Jour 4 :** Je continue avec les ajouts de **HelperText** et la gestion des IDs automatiques. Je travaille aussi sur les tables **signalement\_dispositionX** et applique un filtre sur l'envoi des mails du secteur **Enfance**.

**Jour 5 :** Je termine la semaine en me concentrant sur la correction visuelle des widgets et sur la page de filtrage des widgets. Je crée un **dashboard alternatif** pour améliorer l'affichage.

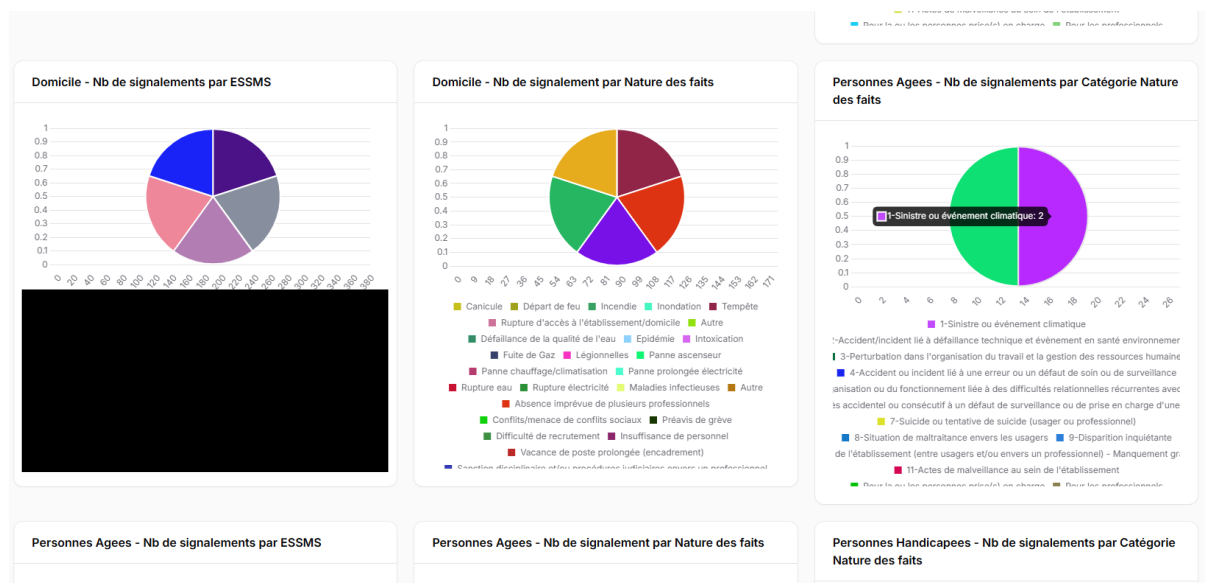


Figure 17: Page Widgets

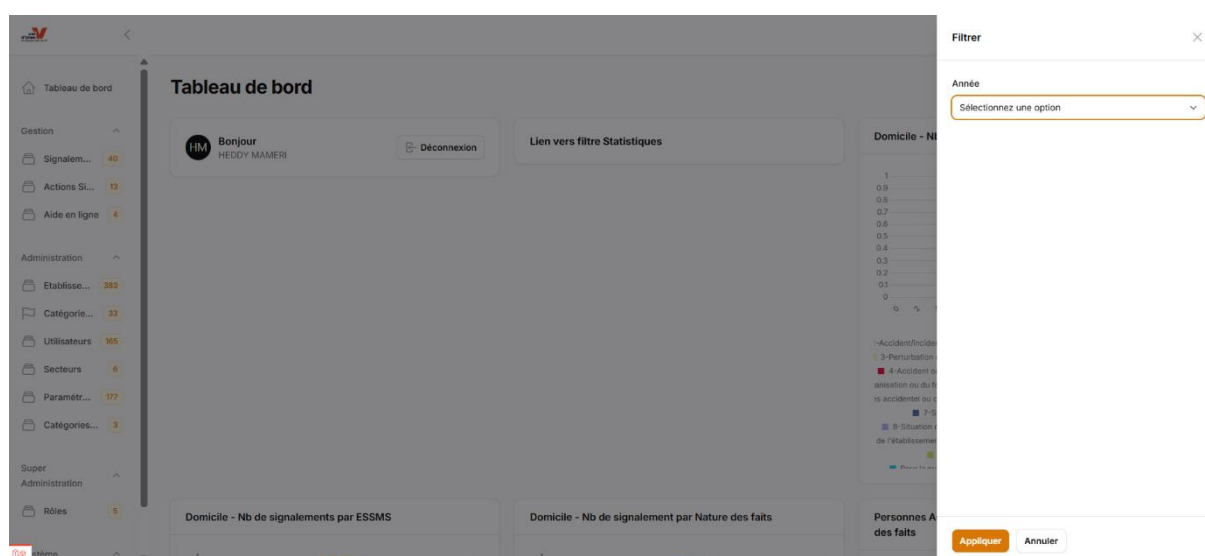


Figure 18: FiltreWidget



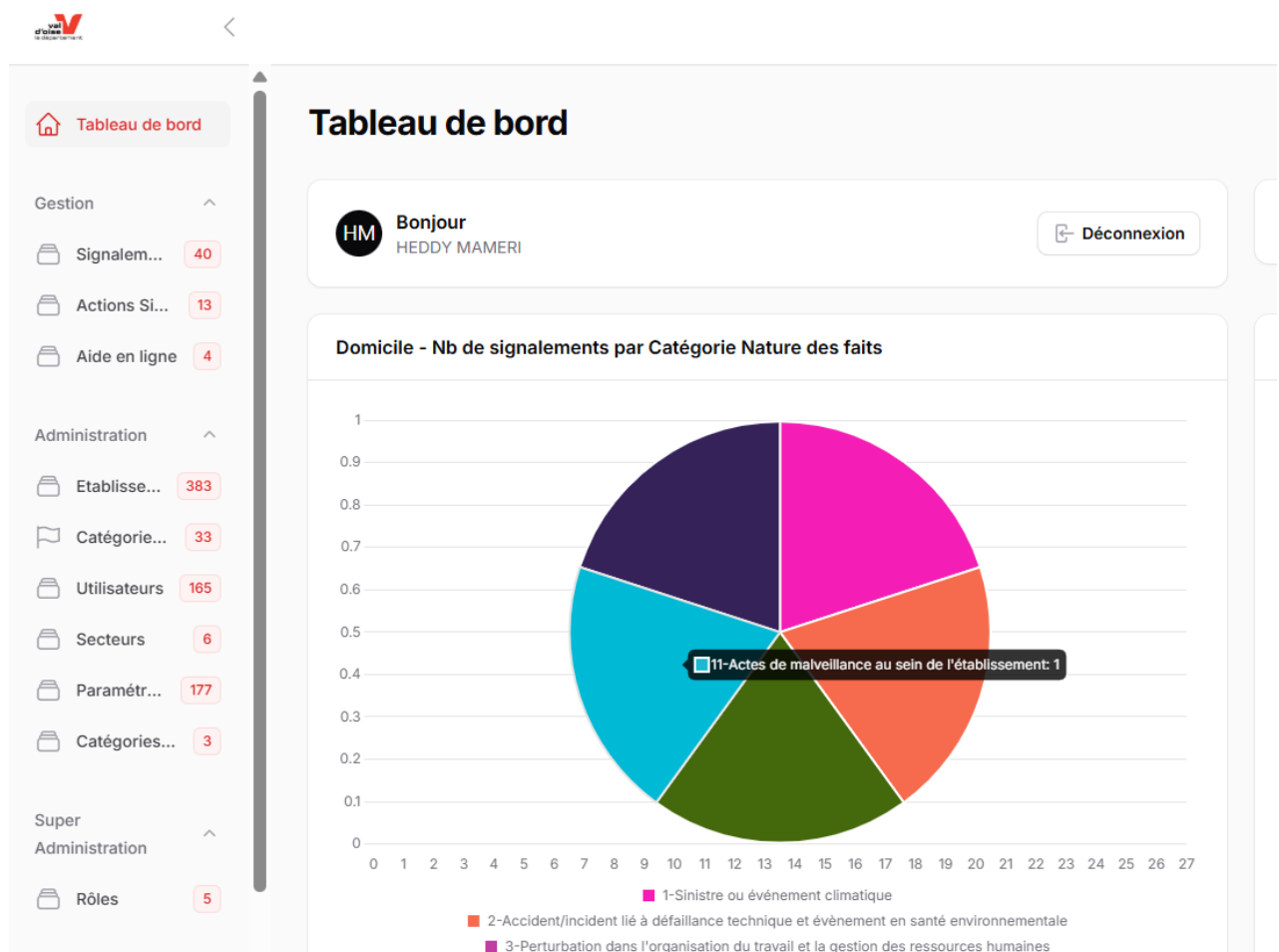


Figure 19: Accueil Signes final

## C / Bilan de la mission :

### Points négatifs :

- Je n'ai pas eu la totalité des 8 semaines pour réaliser le projet. :
  - o J'ai essayé de travailler sur le PC que le département m'a fourni mais il fallait installer certains outils dessus nécessitant des droits admins que je n'avais pas
  - o Je ne pouvais pas travailler sur mon poste personnel la deuxième semaine en raison d'une curl error 60, une erreur de certificat SSL TLS que je n'ai pas pu résoudre
- Le manque de temps m'a poussé à accélérer la cadence, je n'ai pas pris le temps d'étudier en profondeur la documentation du projet dans les détails. Certaines fonctionnalités sont incomplètes par manque de temps :
  - o Etat réceptionné
  - o Mise en place de l'api de l'intranet
  - o Installation du filtre de widget sur la page d'accueil

## Points positifs :

- La totalité des fonctionnalités a été traitée.
- L'ensemble du projet est fonctionnel
- L'objectif d'exploiter les avantages de filament pour améliorer l'interface est atteint via le formulaire de signalement
- Un enregistrement de saisie fonctionnel a été mis en place.

## Bilan du stage

### Points négatifs :

- Je me suis un peu trop cantonné dans ma bulle. Je n'ai pas assez interagi avec les autres au début du stage. Être sur un projet tout seul n'a pas aidé.

### Points positifs :

- J'ai participé à des activités avec mes collègues et me suis bien intégré à l'entreprise :
  - o Déjeuner à la cantine du Parc 3 fois par semaine
  - o Galettes des rois
  - o Réunion Bastion
  - o Tournoi Mario Kart
  - o Pauses café
  - o Petit déjeuner de départ que j'ai organisé en leur offrant les boissons et viennoiseries
- Je me suis montré autonome à la vue de l'emploi du temps très chargé de mes responsables
- Assiduité, à part une fois à cause d'embouteillages, je suis toujours arrivé en avance le matin et reparti le soir 10 ou 20 minutes après l'heure de fin.

## Apports de stage perspectives 2025-2026 :

Ce stage aura été une très riche expérience pour moi.

J'ai une meilleure idée de ce qu'est l'environnement de grande entreprise à haute échelle.

Lors de mon précédent stage à EC2E, c'était 4 semaines dans une entreprise d'une vingtaine de personnes, là c'est un organisme public d'envergure, avec 80 employés.

J'ai appris 2 éléments sur le plan technique :

Le générateur de CMS, cette extension de framework pour obtenir une interface d'administration rapidement est génial. Je pense l'utiliser pour un portfolio V3

Le métier de Data Analyst, je sais maintenant comment un data analyst travail, l'importance de ses productions, le niveau de rigueur que cela demande et les outils qu'ils utilisent à savoir Power BI et BO.