**ELECTRIKA Gestion**

**Réalisé par :**

Marc Longmar

**Période de Stage :**

Du 15 mars 2025 au 1er juin 2025

**Entreprise d'accueil :**

ELECTRIKA

Sommaire

1. Introduction .......................................................... 1

2. Contexte du Projet ................................................... 2

2.1 Contraintes du projet ...................................... 2

2.2 Livrables attendus ........................................... 3

3. Environnement humain et technique ............... 4

3.1 Environnement humain ............................... 4

3.2 Environnement technique ........................... 5

3.3 Objectifs de qualité ........................................ 5

4. Réalisations Front-End ............................................. 6

4.1 Maquettes ...................................................... 6

4.2 Schéma de navigation ..................................... 7

4.3 Captures d'écran d'interfaces utilisateur ........ 8

4.4 Extraits de code statique ............................... 9

4.5 Extraits de code dynamique .......................... 10

4.6 Argumentation des choix techniques .......... 11

5. Réalisations Back-End .............................................. 12

5.1 Base de données ................................................ 12

5.2 Composants métier ........................................... 13

5.3 Composants d'accès aux données .................... 14

5.4 Éléments de sécurité ......................................... 15

5.5 Jeu d'essai et analyse ..................................... 16

5.6 Veille technologique ...................................... 17

6. Conclusion ............................................................ 18

7. Annexes ................................................................ 19

1. **Introduction**

Dans le cadre de ma formation de Développeur Web et Web Mobile à l’AFPA de Cannes, j’ai réalisé un projet personnel intitulé **ELECTRIKA Gestion**. Ce projet, bien qu’ayant été développé en parallèle d’un stage effectué dans une autre structure, s’inscrit pleinement dans les objectifs pédagogiques de la formation et m’a permis de mobiliser l’ensemble des compétences techniques acquises.

Fort de 20 années d’expérience dans le secteur événementiel, j’ai choisi de concevoir une application répondant à une problématique concrète que j’ai souvent rencontrée sur le terrain : la **gestion en temps réel du matériel électrique**. L’application ELECTRIKA Gestion permet de visualiser l’état, la disponibilité et la localisation du matériel de manière claire, fluide et accessible, afin que chaque technicien puisse remplir efficacement son rôle, même en l’absence de préparateur dédié.

Le projet a été **entrièrement conçu, développé et documenté par moi-même**. Il s’agit d’une **première version fonctionnelle**, pensée comme une base évolutive, qui sera soumise ultérieurement à l’entreprise pour un ajustement aux besoins spécifiques et une intégration sur le terrain.

Ce développement s’appuie sur les technologies suivantes : **Symfony**, **Twig**, **JavaScript**, **MySQL**, **Dompdf** pour l’édition de documents PDF, ainsi que les outils **VSCode**, **Composer**, **GitHub** et **Webpack Encore**. J’ai structuré le projet selon une logique inspirée de la **méthode agile**, en organisant mon travail en itérations, depuis la phase de conception jusqu’aux tests fonctionnels.

Ce dossier présente en détail les différentes étapes du projet, les choix techniques effectués, les extraits de code significatifs, ainsi qu’une réflexion sur les aspects de sécurité, d’accessibilité et d’évolution future.

**2. Contexte du projet**

**2.1 Contraintes du projet**

**Contraintes techniques**

Le projet a été développé en **Symfony**, un choix motivé à la fois par les exigences de ma formation — qui consacre une part importante de l’enseignement à ce framework — et par ses qualités techniques. Travaillant seul avec un délai restreint, Symfony m’a permis de bénéficier d’un cadre de développement robuste et bien documenté, facilitant ainsi la mise en place rapide d’une architecture professionnelle.

L’hébergement a été réalisé chez **OVH**, une solution que j’ai retenue pour son **bon rapport qualité/prix** et ses **performances suffisantes** pour accueillir une première version stable du projet. L’application est **déjà en ligne**, accessible dans un environnement test afin de permettre des retours utilisateurs et des évolutions futures.

**Contraintes fonctionnelles**

L’application devait répondre à des besoins très concrets :

* **Affichage de la disponibilité du matériel sur une période de 4 semaines**,
* **Transmission automatique de bons de préparation, de livraison et de retour**,
* **Envoi d’e-mails automatisés** aux responsables techniques à chaque mouvement de matériel (sortie ou retour).

Ces fonctionnalités devaient être disponibles dès la première version, ce qui a représenté une charge importante compte tenu du temps imparti.

**Contraintes organisationnelles**

Le projet a été **entrièrement réalisé en autonomie**. Aucune contrainte spécifique n’a été imposée en termes de méthodes de travail ou de conventions techniques. J’ai néanmoins suivi une organisation personnelle structurée pour respecter au mieux un **planning prévisionnel** :

* 1 semaine pour la **conception et le maquettage**,
* 2 semaines pour le **développement front-end**,
* 4 semaines pour le **développement back-end**,
* 3 semaines pour la **rédaction du dossier de stage** et du **dossier professionnel**.  
  Ce planning a été partiellement dépassé, mais il m’a permis de garder une vision claire de l’avancement global du projet.

**2.2 Livrables attendus**

À l’issue de cette première phase, les **livrables prévus** étaient les suivants :

* Une **application fonctionnelle en ligne** permettant le suivi du matériel,
* La **documentation technique** des principales fonctionnalités,
* Des **jeux d’essais**, notamment sur la fonctionnalité de suivi des disponibilités,
* Un **script de création de la base de données**,
* Des **modèles PDF** pour les documents générés automatiquement (préparation, livraison, retour),
* L’intégration d’un **système de notification par e-mail** opérationnel.

Ces livrables constituent la base d’une version que je prévois de faire évoluer et adapter en collaboration avec l’entreprise pour une mise en production définitive dans les mois à venir.

**3. Environnement humain et technique**

**3.1 Environnement humain**

Bien que ce projet ait été conçu et développé en autonomie, il a bénéficié de **nombreux échanges techniques** avec d'autres apprenants de ma formation ainsi qu’avec notre formateur. Ces échanges ont été précieux, notamment lors des phases de débogage ou pour valider certaines approches fonctionnelles. Ce soutien collectif a contribué à maintenir une dynamique de travail productive et motivante.

Les tests de l'application ont pour l’instant été **réalisés uniquement par moi-même**. Une **phase de test utilisateur** auprès des employés d’ELECTRIKA est envisagée d’ici **juillet 2025**, avec un déploiement plus large prévu durant le mois d’**août**, période plus calme pour l’entreprise.

Le développement a été principalement réalisé **depuis mon domicile**, dans un espace de travail dédié, selon des horaires réguliers. Quelques sessions de travail ont également eu lieu sur le site de l’entreprise ELECTRIKA afin de m’immerger dans les conditions réelles d’usage. Il convient de noter que **la conciliation entre les exigences du projet et les contraintes familiales**, notamment avec des enfants en bas âge le mercredi, a parfois nécessité une adaptation de mon organisation.

**3.2 Environnement technique**

Le projet a été développé à l’aide d’un environnement technique complet et structuré. Les **outils et technologies** utilisés incluent :

* **VSCode** comme éditeur de code principal,
* **Symfony CLI** pour la gestion du framework,
* **Node.js** et **NPM** pour la gestion des packages,
* **Composer** pour la gestion des dépendances PHP,
* **WampServer** pour le développement local (MySQL + PHP),
* **MailDev** pour la simulation des e-mails,
* **phpMyAdmin** pour la gestion visuelle de la base de données,
* **GitHub** pour la gestion de version et les sauvegardes,
* **Figma** pour la réalisation des maquettes.

Le développement a été réalisé sur **plusieurs machines**, notamment un poste personnel, un ordinateur portable (pour les déplacements), et l’ordinateur de l’AFPA. Cette flexibilité m’a permis de rester productif selon les conditions de travail disponibles.

Concernant les serveurs locaux, j’ai utilisé **WampServer** en complément de **Symfony Server**, ce qui m’a permis de tester le projet dans différents environnements.

**3.3 Objectifs de qualité**

Dès le départ, j’ai intégré une logique de **qualité globale** dans le développement de l’application. Les objectifs principaux étaient les suivants :

* **Responsive design** pour garantir l’utilisation sur ordinateur comme sur mobile,
* Respect des bonnes pratiques en matière de **sécurité** (CSRF, validation des données, protection des routes),
* Prise en compte de l’**accessibilité** et de la **lisibilité du code**,
* Une réflexion sur l’**éco-conception** (réduction des requêtes, optimisation des ressources).

L'ensemble de ces objectifs ont guidé mes choix tout au long du projet et continueront à structurer les futures évolutions de l’application.

**4. Réalisations Front-End**

**4.1 Présentation des maquettes**

Les maquettes de l’application ont été conçues avec **Figma**. Elles présentent les écrans principaux de l’interface utilisateur, dont :

* La **page d’accueil** intégrant une vue planning,
* L’interface d’**affichage des produits**,
* Les pages dédiées aux **événements** et aux **sites d’événements**,
* Une page **contact**,
* Ainsi que des vues d’**affichage individuel** pour chaque entité.

Une attention particulière a été portée à la clarté de la navigation et à la lisibilité des informations. Les versions mobiles sont prévues mais non finalisées à ce stade ; elles seront complétées en phase d’évolution.

**4.2 Schéma de navigation**

L’application suit une logique de navigation simple et fonctionnelle :

* Depuis la **page d’accueil** (avec planning intégré), l’utilisateur peut accéder aux **événements à venir**,
* Chaque événement regroupe plusieurs **documents de gestion** : bons de préparation, de livraison, de retour,
* Chaque entité (produit, utilisateur, événement, site...) est accessible via une **liste** puis une **fiche individuelle**,
* Un onglet spécifique permet d’accéder à la section **préparation** : les articles y sont validés avec un événement lié, puis suivent un cycle d’état : **new → prêt → livré → retour**.  
  À chaque changement d’état, un **document PDF est généré** et un **e-mail est envoyé automatiquement** au responsable technique.

Une future évolution envisagée avec le client est l’implémentation d’une **page d’accueil personnalisée** selon le type d’utilisateur (technicien ou profil commercial), mais cela reste à définir avec lui.

**4.3 Captures d’écran et responsive**

L’interface est développée avec le framework **Bootstrap**, garantissant un **affichage responsive** adapté aux ordinateurs et aux terminaux mobiles. Les captures d’écran des différentes pages web et versions mobiles seront intégrées en annexe.

**4.4 Extraits de code des interfaces statiques**

Le développement des vues HTML est basé sur **Twig** (intégré à Symfony).  
L’interface utilise Bootstrap avec des composants personnalisés. Les éléments d’**accessibilité** tels que le contraste des couleurs ont été gérés, et l’ajout des balises **ALT** est en cours de finalisation. Ce point sera revu en détail avant livraison finale.

**4.5 Extraits de code des interfaces dynamiques**

Le **JavaScript** a été utilisé pour améliorer l’interactivité. Notamment :

* **Filtrage dynamique des articles** dans les listes sans appel serveur,
* Mise à jour de l’état d’un article dans une préparation via des interactions immédiates.

Pour l’instant, aucune **requête AJAX** n’a été mise en place, mais l’idée d’ajouter :

* Une **mise à jour en temps réel du stock** (sans rechargement de page),
* Ou un **système de recherche intelligent** (autocomplete avec appel serveur),  
  pourrait offrir une **valeur ajoutée** notable et sera proposée lors des prochains échanges avec le client.

**4.6 Argumentation des choix techniques**

Le choix de **Twig** permet une séparation claire du front-end et des données serveur, tout en limitant les failles XSS grâce à l’échappement automatique des variables.  
L’utilisation de **Bootstrap** permet de gagner du temps tout en garantissant une cohérence visuelle.  
Le recours à JavaScript a été volontairement limité aux besoins essentiels, afin de garder l’application légère et simple à maintenir.

**5. Réalisations Back-End**

**5.1 Base de données**

La base de données a été conçue à partir d'un **MCD et d'un MPD**, comportant un total de **11 tables** (hors tables techniques comme doctrine\_migration\_versions et messenger\_messages). Le modèle conceptuel inclut les entités suivantes : utilisateurs, produits, événements, contacts, sites, détails d’événement, galeries photos, catégories, etc.

Toutes les tables possèdent une **clef primaire (PK)**, et certaines colonnes critiques (comme les adresses e-mail) sont **contraintes en UNIQUE**. Les relations entre les entités sont assurées par des **clefs étrangères (FK)**. La génération de la base est entièrement gérée via **Doctrine**, sans script SQL écrit manuellement.

**5.2 Composants métier**

Des **services Symfony** ont été développés pour deux besoins principaux :

* La **génération de fichiers PDF** (DOMPDF),
* L'**envoi automatisé d'e-mails** à chaque changement d'état d'un bon de préparation/livraison/retour.

La logique métier principale repose sur le **suivi des états du matériel** et la **gestion du stock** associé à chaque événement (new, prêt, livré, retour).

**5.3 Composants d'accès aux données**

Le projet utilise **Doctrine ORM** pour tous les accès aux données. Plusieurs méthodes personnalisées ont été développées dans les Repository, notamment :

* findByEventStatus()
* countOrdersByStatus()
* countStockByProduct()

**5.4 Éléments de sécurité**

La sécurisation de l’application comprend plusieurs éléments clés :

* Mise en place de **rôles utilisateur** (ROLE\_ADMIN, ROLE\_TECH),
* **Hashage des mots de passe** via le composant Security de Symfony,
* **Protection CSRF** pour les formulaires sensibles,
* Validation des données en entrée.

**5.5 Jeu d’essai**

Des jeux d’essai ont été mis en place pour tester :

* L’insertion de données (utilisateurs, produits, événements, contacts, etc.),
* Le calcul dynamique de la **disponibilité du stock** sur 6 semaines glissantes,
* Le **retour d’informations filtrées** et triées selon les statuts et disponibilités.

Les premières données ont été générées avec **Data Fixtures**, puis remplacées progressivement par des insertions manuelles afin de mieux contrôler l’accès selon les rôles.

**5.6 Veille technologique**

Aucune veille spécifique n’a été réalisée à ce jour sur les vulnérabilités de sécurité. Ce travail est prévu lors de la deuxième phase du projet pour renforcer la protection contre les attaques classiques (XSS, injections SQL, etc.).

**6. Conclusion**

Le projet **ELECTRIKA Gestion** m'a permis de mettre en application l'ensemble des compétences acquises tout au long de la formation, de la conception à la mise en ligne d'une application fonctionnelle. Ce travail concret m'a offert l'opportunité de créer une solution répondant à un besoin réel, en lien avec mes années d'expérience dans l'événementiel.

Ce projet m'a permis de renforcer mes compétences en :

* Conception de base de données relationnelle,
* Développement front-end responsive et accessible,
* Intégration de composants métiers en Symfony,
* Mise en place de fonctionnalités de sécurité,
* Génération de documents dynamiques et notifications automatiques.

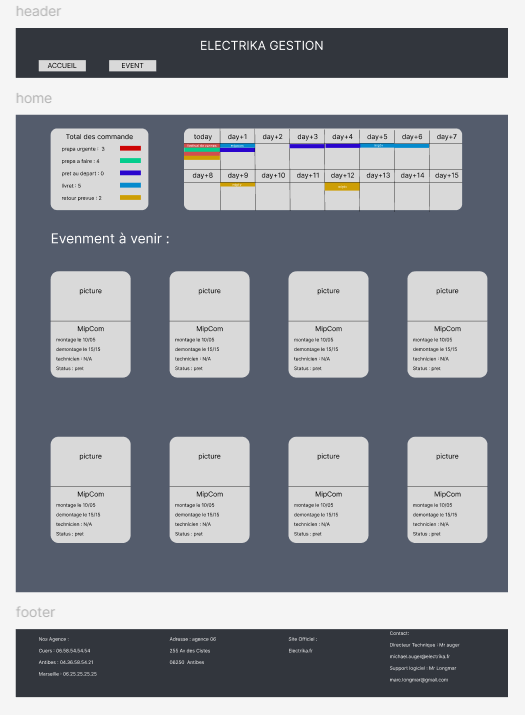
J'ai rencontré plusieurs difficultés, notamment dans la gestion du temps, la complexité des relations entre les entités, ou encore l'adaptation de l'interface aux attentes des futurs utilisateurs. Cependant, ces épreuves m'ont permis de progresser, d'apprendre à prioriser, à tester et à corriger avec rigueur.

Ce projet constitue une première version solide et évolutive qui pourra être affinée en lien avec l'entreprise, en intégrant de nouveaux besoins fonctionnels et des optimisations techniques lors d'une deuxième phase de développement.

**7. Annexes**

**1. Maquettes Figma (3-4 pages)**

**Page d’accueil avec planning**

****

Visualisation rapide des commande et évènement en cours et a venir avec accès rapide et intuitive .

**Entité product , évent , site, contact**

****

Visualisation des Entité avec barre de création et de recherche. Affichage d’option supplémentaire si le rôle est Admin ou tech

**Une image contenant texte, capture d’écran, conception

Le contenu généré par l’IA peut être incorrect.**

Elément non centrer pour permettre l’affichage de fichier liée ( photo pour les user , planning pour les article , évent pour les site etc..)

****

Liste des articles avec leur disponibilité sur 6 semaine avec suppression de colonne en vue mobile pour le mobile first .

**2. Captures d’écran de l’application**

* **Accueil réel du site**
* **Liste des produits**
* **Détail d’un produit**
* **Page de préparation (workflow bon : new → prêt → livré → retour)**
* **Téléchargement d’un PDF généré**
* **Notification e-mail**

**3. Extraits de code commentés**

**Front-end :**

* **Exemple de Twig avec rendu dynamique + sécurisation**
* **Utilisation de JavaScript (filtrage dynamique des articles)**

**Back-end :**

* **Méthode countStockByProduct() ou autre Repository utile**
* **Service de génération de PDF**
* **Service d’envoi de mail avec le changement d’état**

**4. Base de données et schémas**

* **Capture du MCD / MPD**
* **Export image de ton schéma actuel**
* **Capture d’un script généré par Doctrine si pertinent**

**5. Divers (1 page)**

* **Planning prévisionnel initial**
* **Tableau des rôles utilisateurs**
* **TODOs futurs ou évolutions prévues**