LYCÉE PARC DE VILGÉNIS

BTS SIO 1B

2 juin 2025
Au
4 juillet 2025

Amin Chebbi

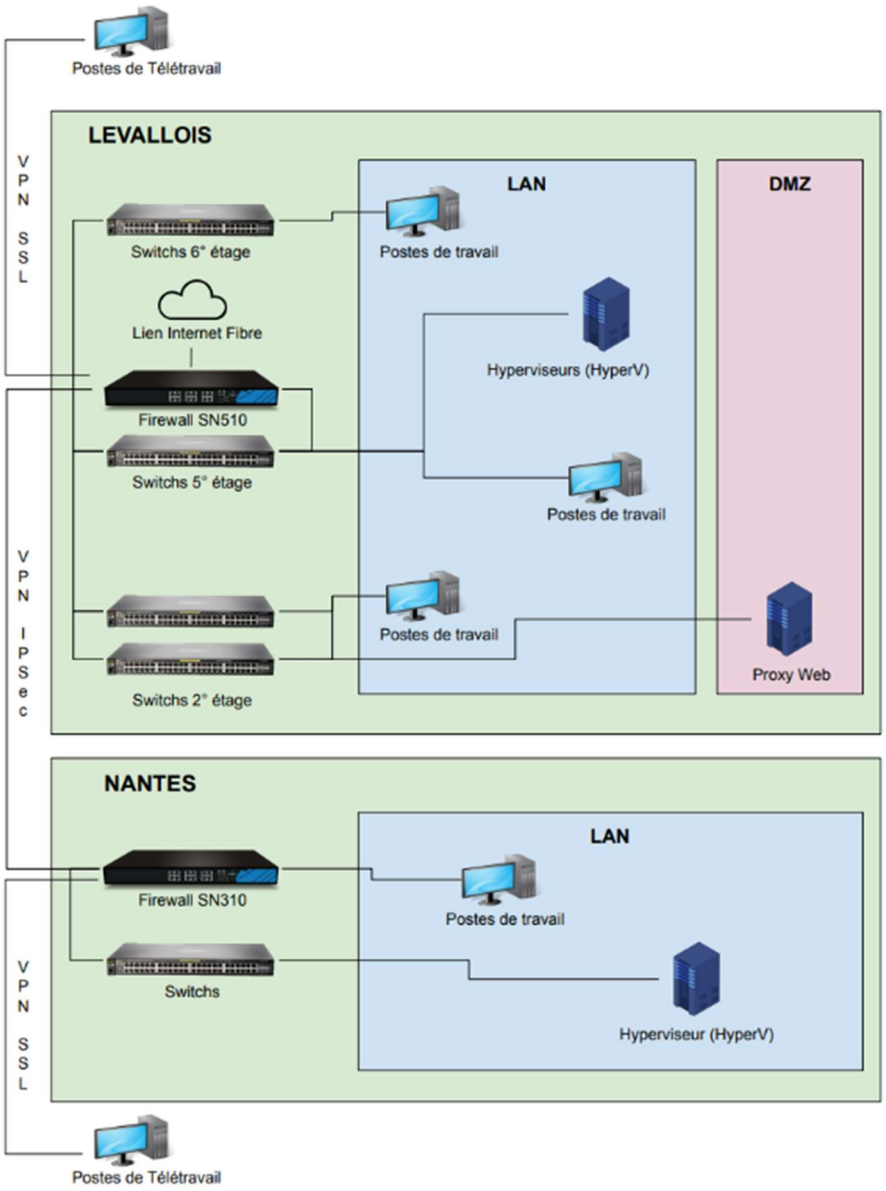
HEROIKS

Année Scolaire 2024-2025

SOMMAIRE

I. Présentation de l'organisation et des contextes professionnels

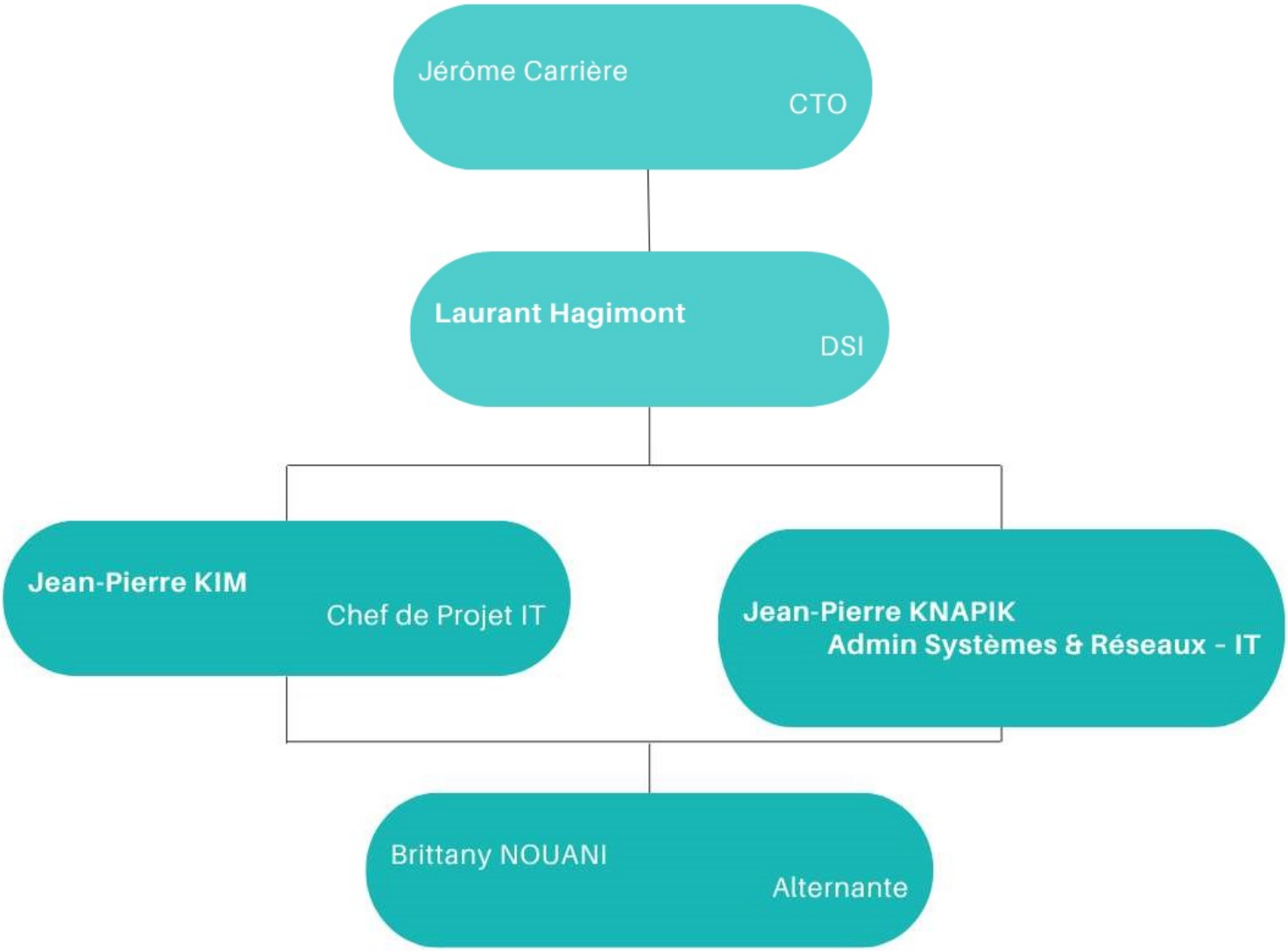
PRESENTATION DU SI



Présentation du Service

- Nom du Service : Service Informatique
- Activité du service : Infra Support et développement
- Nom et Fonction du tuteur : HAGIMONT Laurent – Directeur Informatique

II. Organigramme du service



III. Activité réalisée

LISTE DE TOUTES LES ACTIVITES FAITES EN ENTREPRISE

- Mise en place de Docker et Caddy pour l'hébergement sécurisé de sites web dans un serveur Ubuntu appartenant à l'entreprise.
- Création et gestion de plusieurs sous-domaines (ex : todo.odin.heroiks.com, paris.odin.heroiks.com)
- Développement d'un site web ToDo en PHP/MySQL avec fonctionnalités CRUD
- Développement d'un site web listant les évènements disponibles à Paris en PHP/MySQL avec automatisation des sauvegardes via les tâches CRON
- Conception d'une application mobile Flutter avec fonctionnalité CRUD pour me familiariser avec flutter et son langage dart.
- Mise en place et utilisation de Typesense pour la gestion rapide des données et recherche NoSQL
- ***Développement d'une application web permettant la gestion de note et de tag via des API REST sécurisées en PHP.***
- Montage CIFS pour centraliser des sauvegardes Windows sur Linux

Développement de l'application web JDB (Journal de Bord)

Dans le cadre de mon stage, j'ai été chargé de concevoir et développer une application web complète baptisée **JDB – Journal de Bord**. Cette plateforme collaborative a pour objectif de centraliser l'ensemble des notes techniques produites par l'équipe informatique, afin de suivre de manière structurée les actions, interventions et décisions techniques. Contrairement à une application personnelle, les notes saisies dans JDB sont **partagées entre tous les membres de l'équipe IT**, permettant ainsi un accès collectif, une meilleure traçabilité et un partage continu des connaissances.

Objectifs de l'application

L'application permet aux utilisateurs de :

- Consulter en temps réel toutes les notes techniques ;
- Ajouter de nouvelles notes avec un titre, un contenu, une date, et un tag ;
- Modifier ou supprimer une note existante ;
- Associer chaque note à un **tag coloré** pour faciliter la classification par thématique ou projet ;
- Filtrer dynamiquement les notes par tag ;
- Différencier les rôles des utilisateurs (standard ou administrateur), ces derniers étant les seuls à pouvoir gérer les tags.

Technologies utilisées

Le projet repose sur une architecture moderne et modulaire :

- **Flutter** : framework front-end permettant le développement d'interfaces réactives et multiplateformes ;
- **PHP** : pour la création d'une API REST sécurisée ;
- **Typesense** : moteur de recherche NoSQL utilisé comme base de données ultra-rapide pour stocker les notes, utilisateurs et tags.

Fonctionnalités mises en œuvre

Chaque utilisateur peut se connecter à l'application grâce à un identifiant (email) et un mot de passe, ce dernier étant stocké sous forme de hash. Une fois connecté, l'utilisateur accède à une page d'accueil où sont listées toutes les notes enregistrées par l'équipe.

Depuis cette page (homepage.dart), il est possible :

- D'ajouter une nouvelle note via un formulaire intégré ;
- De modifier une note existante avec ses champs préremplis ;
- De supprimer une note grâce à un bouton d'action, avec demande de confirmation.
- Rechercher une note via la barre de recherche ou le système de filtre via les tags.

Les notes sont affichées sous forme de cartes claires et filtrables. Chaque note peut être associée à un tag, défini par un nom et une couleur. Les tags sont gérés via une page spécifique (TagManagementPage.dart) accessible uniquement aux administrateurs, où il est possible d'ajouter, modifier ou supprimer un tag. Un sélecteur de couleur (color picker) permet d'attribuer une couleur intuitive à chaque catégorie.

Architecture du front-end

L'application Flutter a été conçue avec une architecture claire et modulaire :

- **main.dart** : fichier servant à démarrer l'application Flutter en lançant la fonction main() qui affiche la première interface à l'écran.
- **/pages/Connexion/login.dart** : formulaire de connexion avec validation des champs et interaction avec l'API ;
- **/pages/homepage.dart** : page principale, affichant les notes sous forme de liste dynamique, avec formulaires intégrés pour l'ajout, la modification et la suppression des notes ;
- **/pages/TagManagementPage.dart** : page réservée à la gestion des tags (nom et couleur), disponible uniquement pour les comptes administrateurs.

Le code a été organisé selon trois blocs logiques :

- **Modèles** : représentation des objets Note, Tag et Utilisateur ;
- **Interfaces utilisateur** : vues Flutter responsives et interactives ;
- **Services** : gestion centralisée des requêtes HTTP vers les API PHP.

Développement de l'API (PHP + Typesense)

Le backend est composé d'une API REST codée en PHP, structurée en quatre modules : auth, user, note, tag. Chaque module contient des fichiers dédiés aux actions principales : get, add, update, delete.

- **Note** :
    - Récupération de toutes les notes via l'API (partagées entre les membres de l'équipe) ;
    - Création d'une note avec titre, contenu, date, tag, identifiant de l'auteur et horodatage automatique ;
    - Mise à jour d'une note tout en conservant l'auteur d'origine ;
    - Suppression d'une note.
- **Auth** :
    - Vérification des identifiants (email + mot de passe) ;
    - Retour des données de l'utilisateur connecté (nom, prénom, rôle).
- **Tag** :
    - Liste de tous les tags disponibles avec tri par date de création ;
    - Ajout, édition et suppression de tags réservés aux administrateurs.

Toutes les données sont stockées dans **Typesense**, qui offre des performances élevées sur les recherches, tris et filtres dynamiques. Un fichier central assure la **connexion sécurisée** au cluster Typesense et génère des identifiants uniques pour chaque note, tag ou utilisateur.

Modélisation des données

Trois collections principales ont été créées dans Typesense :

- **note** : NoteID, Titre, Description, Date, TagID, UserID, *(pas de champ explicite pour la date de création, directement inclus dans le front-end)*
- **user** : UserID, Nom, Prenom, mail, mdp, rôle
- **tag** : TagID, Nom, Couleur

Cette structure permet un filtrage précis, une recherche rapide et une évolutivité facile en cas d'ajout de nouvelles fonctionnalités (archivage, commentaires, pièces jointes, etc.).

Stabilisation et amélioration continue

Durant la phase de développement, j'ai identifié et corrigé plusieurs dysfonctionnements :

- Rafraîchissement incomplet de la liste des notes après modification ;
- Problèmes d'affichage du nom de l'auteur ;
- Redondance ou confusion dans certaines fonctions.

Afin d'assurer la maintenabilité de l'application, j'ai refactorisé l'ensemble du code Flutter :

- Réorganisation des fichiers ;
- Clarification des responsabilités (vue, logique, modèle) ;
- Mise en place d'une convention de nommage claire (en prefixe ***Req*** pour les données entrantes, ***My*** pour les données locales, ***The*** pour les paramètres de fonction).

Ce projet a permis la création d'une application robuste, intuitive et collaborative, adaptée aux besoins concrets de l'équipe IT. Elle constitue désormais un outil de travail quotidien, centralisant efficacement toutes les interventions techniques de l'équipe.