**Conception et Développement d'une application web Unified Notification Service for Developers** :Une API avancée pour la génération et la notification sécurisée des codes d’accès

**WAJDI ZAYENE   
3eme année licence génie logiciel**

Année universitaire : 2024 / 2025

1. ***Présentation de l'entreprise d'accueil :***

Astrolab est une entreprise spécialisée dans le développement informatique. Depuis sa création en 2015 en Tunisie, elle accompagne ses clients en leur offrant **adaptabilité, productivité et réactivité** grâce à son expertise dans les domaines de la technologie, de l’industrie, du conseil et du marketing.

Avec une approche résolument **orientée client**, Astrolab s’engage à fournir un service professionnel et un accompagnement personnalisé à des clients du monde entier.

Nous sommes présents à **Paris, Doha, Sousse (Tunisie)** et continuons d’investir pour l’avenir en proposant des **solutions innovantes** qui respectent les principes de l’**agilité** et relèvent les défis de la **performance**.

****

Figure : Astrolab

1. ***Description :***

Les entreprises et les développeurs doivent gérer plusieurs canaux de notification (e-mail, SMS, push, in-app). Ce projet(Unified Notification Service for Developers ) vise à centraliser l'envoi des notifications via une **API unique**, simplifiant ainsi l'intégration et la gestion.

* 1. **Acteurs**
* **Administrateur** : Gère les utilisateurs et configure les canaux.
* **Développeurs** : Utilisent l’API pour envoyer des notifications.
* **Utilisateurs finaux** : Reçoivent les notifications

**2.2 Fonctionnalités Principales**

**Gestion des Notifications**

* Envoi de **notifications multi-canal** (Email, SMS, Push, In-App).
* Support des **modèles de messages** (Templates EJS pour Email).
* Support des **différents niveaux de priorité**.

**API RESTful**

* Endpoint unique pour envoyer une notification.
* Support de l’authentification **JWT / API Key**.
* Gestion des statuts et logs des notifications envoyées.

**Dashboard Web (React.js + Material UI)**

* Interface pour gérer les envois de notifications.
* Gestion des modèles d’e-mails avec un éditeur HTML.
* Statistiques et logs des notifications envoyées.

**Gestion des Canaux**

* **E-mail** : AWS SES / SMTP avec **EJS pour les modèles**.
* **SMS** : Intégration **Twilio**.
* **Push Web** : **Web-Push, Firebase Cloud Messaging (FCM)**.
* **In-App** : Gestion des notifications dans l’application web.

**Sécurité et Gestion des Droits**

* Authentification avec **JWT / API Key**.
* Gestion des permissions par rôle (RBAC).
* Chiffrement des données sensibles.

**Performance & Scalabilité**

* **File d’attente Redis** pour traiter les notifications en arrière-plan.
* **Docker + Nginx** pour le déploiement optimisé.

1. ***Les objectifs principaux :***

* Permettre aux développeurs d'envoyer des notifications via **une seule API**.
* Intégrer plusieurs canaux (**E-mail, SMS, Push Web, In-App**).
* Offrir une **interface web** pour la gestion et la configuration des notifications.
* Optimiser la **scalabilité et la sécurité**.

1. ***Technologies :***

**Backend : NestJS**

NestJS est un framework **backend** basé sur **Node.js** et conçu pour être modulaire et évolutif. Il utilise **TypeScript** et suit une architecture inspirée d’Angular, avec des **controllers, services et modules**. Il facilite l’intégration avec **PostgreSQL, Redis** et propose un support natif pour WebSockets et GraphQL

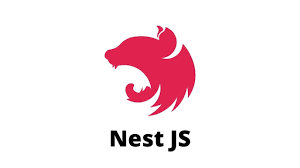
******

Figure  :nest Js

**Frontend : React.js & Material UI**

React.js est une **bibliothèque JavaScript** utilisée pour construire des interfaces utilisateur dynamiques et interactives. Il utilise un système de **composants réutilisables**, ce qui permet une meilleure organisation du code. Material UI fournit des **composants modernes et stylisés**, facilitant la conception du tableau de bord.

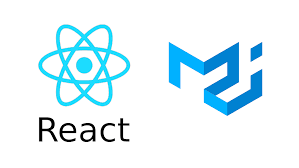


Figure React.js & Material UI

**Base de données & Queue : PostgreSQL & Redis**

* PostgreSQL est une base de données relationnelle **open-source** connue pour sa **fiabilité et scalabilité**. Il permet de gérer des **données complexes** et supporte les transactions ACID.
* Redis est un système de **cache et file d’attente** ultra-rapide, utilisé pour stocker temporairement des données et traiter les notifications en **arrière-plan**.



Figure :PostgreSQL & Redis

**Notifications : Email (AWS SES, SMTP, EJS), SMS (Twilio), Push (Web-Push, FCM)**

 AWS **SES & SMTP** permettent d’envoyer des e-mails en toute sécurité et à grande échelle, avec **EJS** pour personnaliser les modèles d’e-mails.

 Twilio est une API de messagerie qui simplifie l’envoi de **SMS** avec des options avancées (suivi, gestion des réponses).

 Web**-Push & Firebase Cloud Messaging (FCM)** permettent d’envoyer des **notifications push aux navigateurs et applications mobiles**.

****

Figure :twilo

Durée : 4mois