



CAHIER DES CHARGES TECHNIQUE

Version V01

Direction Technique

Buildingmap





contact@futurmap.com



www.futurmap.com

Date d'édition :
mardi 14 octobre 2025

1 INFORMATION GENERALES

Titre du projet : ConfianceFmap

Demandeur(s) : Département Qualité

Date de la demande :

Date limite souhaitée :

Version du document : 1.1

Glossaire :

- **NCE** : Non-Conformité Externe
- **NPS** : Net Promoter Score
- **CSAT** : Customer Satisfaction
- **JWT** : JSON Web Token
- **MinIO** : Stockage objet compatible S3
- **CRUD** : create, read, update,delete.
- **BLOB** :Binary Large Object.
- **UUID** : Universal Unique Identifier
- **PK** : Primary Key

2 OBJECTIFS TECHNIQUES

2.1 OBJET :

L'objectif principal est de développer une plateforme web responsive permettant la centralisation et la traçabilité des livraisons, la gestion des non-conformités externes et la mesure de la satisfaction client.

2.2 OBJECTIFS SECONDAIRES :

- Assurer la traçabilité des livraisons et des accusés.
- Automatiser les sondages NPS et CSAT.
- Gérer les non-conformités externes liées aux livraisons.
- Offrir un tableau de bord qualité interactif.
- Garantir la sécurité et la conformité ISO9001 des données.

2.3 PERIMETRE FONCTIONNEL :

1. Gestion des livraisons :

- Création et organisation : chaque livraison attachée à un projet.
- Upload multi-format : PDF, Excel, Word, images, modèles 3D, ZIP.
- Notifications email : livraison envoyée aux guests via email.
- Accusé de réception : génération automatique de fichier PDF horodaté.
- Versioning minimal : possibilité de versionner les livrables si nécessaire.

2. Collecte des NCE et sondages

- **Méthode** : réponse email automatisée via EmailService (parsing des réponses et pièces jointes).
- Guests ne doivent pas accéder directement à l'application.
- NCE : déclaration simple via réponse email, pièces justificatives attachées.
- Survey (NPS/CSAT) : réponse par score simple ou clic dans email, traitement automatisé côté serveur.
- Historique complet : liens entre NCE, Survey et livraisons.

3. Reporting et tableau de bord

- Indicateurs clés : nombre de livraisons, taux de NCE, score moyen CSAT/NPS.
- Filtres par période, projet, client, catégorie de NCE.
- Export Excel/PDF.

3 SPECIFICATIONS FONCTIONNELLES

3.1 CAS D'USAGE PRINCIPAUX

- **Guest(client)** : reçoit email, peut répondre NCE/sondage par email, joindre fichiers justificatifs.
- **User interne(producer)** : crée/modifie/supprime livraisons, gère NCE, suit reporting.
- **Administrateur** : gestion des utilisateurs et permissions, configuration globale.

3.2 FLUX UTILISATEURS

- Flux centralisé : **User → Notification email → Guest → EmailService → Survey/NCE**

3.3 GESTION DES ROLES ET PERMISSIONS

- Admin : CRUD complet sur toutes entités.
- Qualité : CRUD livraisons, NCE, sondages.
- Producer: CRUD livraisons, upload fichiers.
- Guest : Réception email, réponse NCE/Survey via email.

4 ARCHITECTURE TECHNIQUE

- **Frontend** : Next.js (React, TypeScript, TailwindCSS)
- **Backend** : FastAPI (Python 3.12)
- **Base de données** : PostgreSQL
- **Stockage fichiers** : MinIO (compatible S3)
- **Conteneurisation** : Docker / Docker Compose
- **Déploiement** : Local
- **Authentification** : JWT
- **Microsoft Graph API** : **EmailService local** pour envoi et traitement des réponses guest

5 SPÉCIFICATIONS TECHNIQUES

- API contract : **Swagger / OpenAPI** avec endpoints détaillés.
- Convention de code : **PEP8 (Python), ESLint (TypeScript)**.
- Versioning Git : **branches main/dev, commits conventionnels**.
- CI/CD : **GitHub Actions** pour tests et déploiement.
- Tests : **Pytest, Jest, Cypress (E2E)**.

6 BASE DE DONNEES

- **User** : id, email, role, created_at
- **Guest** : id, email, name, created_at
- **Project** : id, name, owner_id
- **Delivery** : id, project_id, title, description, issuer_id, created_at, status, published_at
- **File** : id, storage_key, filename, delivery_id, nce_id, created_at
- **Receipt** : id, delivery_id, file_id, generated_by, generated_at
- **Notification** : id, delivery_id, to_email, channel, status, sent_at
- **Survey** : id, delivery_id (nullable), type (NPS/CSAT), score, comment, respondent_email, responded_at
- **NCE** : id, delivery_id, title, description, category, severity, status, created_at, attachments
- **AuditLog** : id, user_id, action_type, entity_type, entity_id, timestamp

7 SÉCURITÉ

- HTTPS.
- Authentification JWT.
- Gestion fine des rôles et permissions.
- Politique de mot de passe gérée par **Azure Active Directory** (Microsoft Graph avec OAuth2)

8 DÉPLOIEMENT ET INFRASTRUCTURE

- Architecture réseau documentée.
- Environnements : dev / staging / production.
- Orchestration Docker Compose.

9 TESTS ET QUALITÉ

- Stratégie tests complète : unitaires, intégration, E2E, performance.
- Couverture $\geq 80\%$.
- Critères d'acceptation définis.

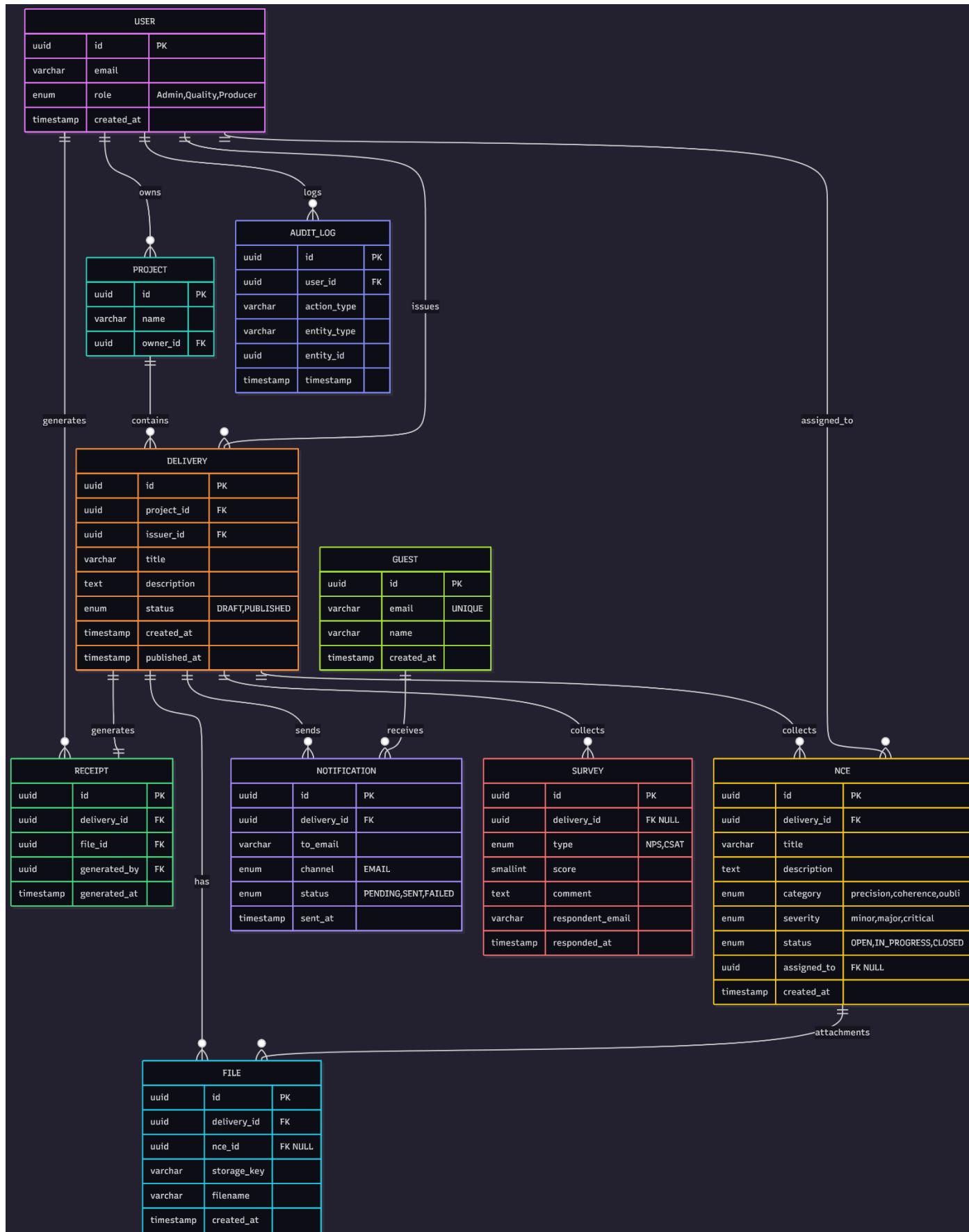
10 LIVRABLES TECHNIQUES

- Code source complet.
- Documentation technique et API.
- Scripts Docker et configuration.
- Diagrammes architecture et BDD.
- Manuel d'installation.

11 ANNEXES

- Diagrammes UML : cas d'usage, séquence.
- BDD: représentation ERD (Entity-Relationship Diagram)
- Schéma d'architecture.
- Charte graphique.
- Exemples payload JSON pour API.

11.1 EXEMPLE DE VISUALISATION ERD DE BDD



11.2 EXEMPLE DE DIAGRAMME DE CLASSE :

