

|  |  |  |
| --- | --- | --- |
| *Le futur se dessine aujourd’hui* | | |
| Cahier  Des Charges  Technique | | |
| Version V01  Direction Technique  Buildingmap | | |
| Résultat de recherche d'images pour "icon orange mail" | contact@futurmap.com | Date d’édition :  jeudi 9 octobre 2025 | |
| Résultat de recherche d'images pour "icon orange website" | www.futurmap.com |

# **Information Générales**

**Titre du projet : ConfianceFmap**

**Demandeur(s) : Département Qualité**

**Date de la demande :**

**Date limite souhaitée :**

**Version du document : 1.0**

**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

# Objectifs Techniques

## 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.

## 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.

## Périmètre fonctionnel :

**1) Gestion des livraisons**

* **Création et organisation** : possibilité de créer une livraison attachée à un projet/client.
* **Upload multi-format** : prise en charge minimale de PDF, Excel, Word, images, modèles 3D, ZIP, etc.
* **Organisation par projet** : structuration claire par projet et par client externe (Guest).
* **Versioning** : gestion basique des versions de livrables (ex. v1, v2…).
* **Accusé de réception** : génération automatique d’un accusé horodaté (fichier PDF) lié à la livraison.
* **Notifications** : envoi d’email automatique aux clients externes via lien, lorsqu’une nouvelle livraison est disponible.
* **Téléchargement sécurisé** : accès aux fichiers via lien unique et temporaire.

**2) Mesure de la satisfaction client**

* **NPS** : envoi automatique d’un questionnaire simple (score 0–10 + commentaires) lié à une livraison.
* **CSAT** : sondage transactionnel configurable après livraison ou résolution d’une NCE.
* **Stockage et analyse** : stockage structuré des réponses pour analyse ultérieure .

**3) Gestion des non-conformités externes (NCE)**

* **Déclaration simple** : formulaire accessible via lien, pré-rempli si possible avec référence livraison/version.
* **Lien automatique** : liaison d’une NCE à une livraison et version spécifique.
* **Pièces justificatives** : possibilité d’ajouter fichiers justificatifs, commentaires ou captures d’écran.
* **Classification** : catégorisation (ex. précision, cohérence, oubli,…) + gravité (mineure, majeure, critique).
* **Workflow minimal** : suivi de statut simple (Ouverte, En investigation, En traitement, Clôturée).
* **Historique filtrable** : visualisation par client, projet et catégorie NCE.

**4) Reporting et tableau de bord**

* **Indicateurs clés** : nombre de livraisons, taux de NCE, score moyen CSAT/NPS.
* **Filtres essentiels** : par période, projet, client, catégorie de NCE.
* **Export minimal** : possibilité d’exporter au format Excel ou PDF des données et rapports simples.
* **Alertes basiques** : configuration d’alertes simples (ex. : NCE > seuil, CSAT < seuil, chute NPS).

# Spécifications fonctionnelles

## Cas d’usage principaux

* En tant que client (Guest) : accéder via lien sécurisé aux livraisons, télécharger des fichiers, répondre à un sondage CSAT/NPS, créer une NCE.
* En tant qu’utilisateur interne (User) : créer/modifier/supprimer livraisons, gérer NCE, consulter tableau de bord qualité.
* En tant qu’administrateur : gérer utilisateurs, permissions, configuration générale.

## Flux utilisateurs

* Diagramme de flux utilisateurs → à définir (annexes)

## Gestion des rôles et permissions

* Admin : CRUD complet sur toutes entités.
* Qualité : CRUD livraisons, NCE, sondages.
* Producer: CRUD livraisons, upload fichiers.
* Guest : lecture livraison, téléchargement, création NCE.

# ARCHITECTURE TECHNIQUE

* **Frontend** : Next.js (React, TypeScript, TailwindCSS)
* **Backend** : FastAPI (Python 3.12)
* B**ase de données** : PostgreSQL
* **Stockage fichiers** : MinIO (compatible S3)
* **Conteneurisation** : Docker / Docker Compose
* **Déploiement** : Local
* **Authentification** : JWT
* **Reverse Proxy** : Nginx

# 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).**
* Limites techniques **: taille max fichier, nombre versions.**

# Base de données

**User :**  
• id (UUID, PK)  
• email (VARCHAR, UNIQUE)  
• hashed\_password (TEXT)  
• role (ENUM: admin, quality, producer)  
• created\_at (TIMESTAMP)  
→ note : comptes internes sécurisés, identification et permissions.

**Guest :**  
• id (UUID, PK)  
• email (VARCHAR, UNIQUE)  
• created\_at (TIMESTAMP)  
→ note: clients externes accessibles via lien, suivi simple.

**Project :**  
• id (UUID, PK)  
• name (VARCHAR)  
• client\_guest\_id (UUID, FK → Guest.id) NULLABLE  
→ note : organisation des livraisons par projet/client.

**Delivery :**  
• id (UUID, PK)  
• project\_id (UUID, FK → Project.id)  
• title (VARCHAR)  
• version (VARCHAR)  
• issuer\_id (UUID, FK → User.id)  
• created\_at (TIMESTAMP)  
→ note : entité centrale pour livraison/version/accusé.

**File :**   
• id (UUID, PK)  
• storage\_key (VARCHAR) — clé MinIO/S3  
• filename (VARCHAR)  
• delivery\_id (UUID, FK → Delivery.id) NULLABLE  
• nce\_id (UUID, FK → NCE.id) NULLABLE  
• created\_at (TIMESTAMP)  
→ note: centralise tous fichiers (livrable, justificatif, accusé) sans BLOB.

**DeliveryReceipt :**   
• id (UUID, PK)  
• delivery\_id (UUID, FK → Delivery.id)  
• file\_id (UUID, FK → File.id)  
• generated\_at (TIMESTAMP)  
→ note : accusé comme entité métier distincte liée à un fichier.

**EmailToken :**   
• id (UUID, PK)  
• token (VARCHAR UNIQUE)  
• email (VARCHAR)  
• purpose (ENUM: view\_delivery, download\_file, respond\_survey, create\_nce)  
• resource\_id (UUID NULLABLE)  
• expires\_at (TIMESTAMP)  
→ note : gestion des liens sécurisés.

**Survey :**  
• id (UUID, PK)  
• delivery\_id (UUID, FK → Delivery.id) NULLABLE  
• type (ENUM: NPS, CSAT)  
• score (SMALLINT NULLABLE)  
• respondent\_email (VARCHAR NULLABLE)  
• responded\_at (TIMESTAMP NULLABLE)  
→ note : collecte des sondages liés aux livraisons.

**NCE :**  
• id (UUID, PK)  
• delivery\_id (UUID, FK → Delivery.id) NULLABLE  
• title (VARCHAR)  
• description (TEXT)  
• category (VARCHAR)  
• severity (ENUM: minor, major, critical)  
• status (ENUM: open, in\_progress, resolved, closed)  
• reporter\_email (VARCHAR NULLABLE)  
• created\_at (TIMESTAMP)  
→ note : déclaration et suivi minimal des non-conformités, avec catégorisation.

**Notification** :  
• id (UUID, PK)  
• recipient\_id (UUID, FK → User.id) NULLABLE  
• guest\_email (VARCHAR NULLABLE)  
• type (VARCHAR)  
• title (VARCHAR)  
• message (TEXT)  
• is\_read (BOOLEAN)  
• created\_at (TIMESTAMP)  
→ note : notifications internes ou externes, permettant d’alerter l’utilisateur ou invité, avec suivi du statut lu/non lu et tri chronologique.

**AuditLog :**  
• id (UUID, PK)  
• entity (VARCHAR) — table/ressource concernée  
• entity\_id (UUID NULLABLE)  
• action (VARCHAR)  
• actor\_email (VARCHAR NULLABLE)  
• details (JSONB NULLABLE)  
• created\_at (TIMESTAMP)  
→ note : traçabilité et conformité.

# SÉCURITÉ

* HTTPS.
* Authentification JWT.
* Gestion fine des rôles et permissions.
* Politique mots de passe (hachage sécurisé, renouvellement).

# DÉPLOIEMENT ET INFRASTRUCTURE

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

# TESTS ET QUALITÉ

* Stratégie tests complète : unitaires, intégration, E2E, performance.
* Couverture ≥ 80%.
* Critères d’acceptation définis.

# LIVRABLES TECHNIQUES

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

# Annexes

* Diagrammes UML : cas d’usage, séquence.
* Schéma d’architecture.
* Charte graphique.
* Exemples payload JSON pour API.

## Exemples payload JSON --- Livraison :

