



CAHIER DES CHARGES TECHNIQUE

Version V01

Direction Technique

Buildingmap





contact@futurmap.com



www.futurmap.com

Date d'édition :
jeudi 9 octobre 2025

1 INFORMATION GENERALES

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

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** : 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).

3 SPECIFICATIONS FONCTIONNELLES

3.1 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.

3.2 FLUX UTILISATEURS

- Diagramme de flux utilisateurs → à définir (annexes)

3.3 GESTION DES ROLES ET PERMISSIONS

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

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
- **Reverse Proxy** : Nginx

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

6 BASE DE DONNEES

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.

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.

7 SÉCURITÉ

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

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.
- Schéma d'architecture.
- Charte graphique.
- Exemples payload JSON pour API.

11.1 EXEMPLES PAYLOAD JSON --- LIVRAISON :

```
1 {
2   "id": "d2f1c8a4-8b2b-4f4f-b0d7-123456789abc",
3   "project": {
4     "id": "a1b2c3d4-e5f6-7890-abcd-1234567890ef",
5     "name": "Projet Construction Bâtiment A",
6     "client_guest": { ...
10   }
11 },
12 "title": "Livraison plans structurels",
13 "version": "v2",
14 "issuer": { ...
18 },
19 "created_at": "2025-10-09T11:30:00Z",
20 "files": [ ...
33 ],
34 "delivery_receipt": {
35   "id": "receipt-uuid-9876",
36   "file": { ...
41 },
42   "generated_at": "2025-10-09T11:36:00Z"
43 },
44 "nces": [
45   {
46     "id": "nce001-uuid-4321",
47     "title": "Erreur dans plancher étage 2",
48     "description": "La cote indiquée dans la feuille 4 est incorrecte.",
49     "category": "précision",
50     "severity": "major",
51     "status": "open",
52     "reporter_email": "client.exemple@domain.com",
53     "created_at": "2025-10-09T12:00:00Z"
54   }
55 ],
56 "surveys": [
57   {
58     "id": "survey001-uuid-7890",
59     "type": "NPS",
60     "score": 9,
61     "respondent_email": "client.exemple@domain.com",
62     "responded_at": "2025-10-09T12:05:00Z"
63   }
64 ]
```