

# Cahier des Charges - Plateforme StreamIO

## 1. Introduction

### 1.1 Contexte

StreamIO est une plateforme web visant à connecter trois types d'utilisateurs : les créateurs de contenu (vidéastes), les entreprises et les administrateurs. Elle permet aux créateurs de gérer leurs chaînes et produits, aux entreprises de rechercher des créateurs, et aux administrateurs de superviser la plateforme.

### 1.2 Objectifs

- Offrir une interface sécurisée et intuitive pour l'inscription, la connexion et la gestion des comptes.
- Permettre aux créateurs de gérer leurs chaînes (ajout, modification, suppression) et leurs produits.
- Faciliter la recherche de chaînes par les entreprises avec filtres par thème.
- Donner aux administrateurs des outils pour gérer les utilisateurs (blocage, suppression, promotion).
- Garantir la sécurité des données et des sessions.

### 1.3 Périmètre

- Développement d'une application web avec backend (API REST) et frontend.
- Gestion de trois types d'utilisateurs avec fonctionnalités spécifiques.
- Sécurité des données et protection contre les attaques courantes (CORS, injections).
- Gestion des fichiers (images uniquement, 10MB max).

## 2. Besoins fonctionnels

### 2.1 Authentification et gestion des comptes

- **Inscription :**
  - Créateurs : email (unique), mot de passe (8+ caractères, majuscule, minuscule, chiffre, caractère spécial), pseudo (unique), nom, prénom.
  - Entreprises : email (unique), mot de passe (mêmes critères), nom de l'entreprise, numéro SIRET, adresse.
  - Validation : détection d'email/pseudo déjà utilisé, conformité du mot de passe.
- **Connexion :**
  - Via email/mot de passe.

- Session de 4 heures avec token JWT.
- Redirection selon le type d'utilisateur (créateur, entreprise, administrateur).
- Messages d'erreur : identifiants incorrects, compte bloqué, session expirée.
- **Gestion du compte :**
  - Modification des informations personnelles.
  - Réinitialisation du mot de passe via email.
  - Suppression du compte.

## 2.2 Fonctionnalités par type d'utilisateur

- **Créateurs de contenu :**
  - **Gestion des chaînes :**
    - Ajout d'une chaîne : email, nom, thèmes (max. 3), URL, nombre d'abonnés.
    - Modification/suppression des chaînes.
    - Consultation de la liste des chaînes.
  - **Gestion des produits :**
    - Ajout de produits avec image (10MB max, formats image uniquement).
    - Consultation/suppression des produits.
- **Entreprises :**
  - Recherche de chaînes avec filtres par thème.
  - Visualisation des informations des chaînes (nom, thèmes, URL, abonnés).
- **Administrateurs :**
  - Consultation de tous les utilisateurs.
  - Blocage d'un utilisateur avec motif.
  - Suppression d'un utilisateur.
  - Promotion d'un utilisateur en administrateur.

## 3. Besoins non fonctionnels

### 3.1 Sécurité

- Hachage des mots de passe avec bcrypt.
- Tokens JWT avec expiration (4h) et cookies sécurisés.
- Protection CORS contre les requêtes non autorisées.
- Validation des fichiers uploadés (images, 10MB max).

### 3.2 Performance

- Temps de réponse API : < 1 seconde pour 95% des requêtes (hypothèse).
- Support d'au moins 1000 utilisateurs simultanés (hypothèse).

### 3.3 Compatibilité

- Navigateurs : Chrome, Firefox, Safari (dernières versions).
- Responsive design pour desktop et mobile (hypothèse).

### 3.4 Disponibilité

- Disponibilité de la plateforme : 99,9% (hypothèse).
- Maintenance planifiée avec notifications aux utilisateurs.

## 4. Spécifications techniques

### 4.1 Architecture

- Backend : API REST avec routes définies (voir documentation).
- Frontend : interface web (technologie à définir, ex. : React, Vue.js).
- Base de données : relationnelle (ex. : MySQL, PostgreSQL) pour stocker utilisateurs, chaînes, produits.

### 4.2 Technologies

- **Backend :**
  - Langage : Node.js (hypothèse basée sur l'usage de JWT et bcrypt).
  - Sécurité : bcrypt pour mots de passe, JWT pour sessions, CORS.
- **Frontend :**
  - Framework : React ou Vue.js (à confirmer).
  - Gestion des uploads : validation côté client et serveur.
- **Base de données :**
  - SGBD relationnel avec tables pour utilisateurs, chaînes, produits.
- **Hébergement :**
  - Cloud (ex. : AWS, Azure) pour scalabilité (hypothèse).

### 4.3 Routes API

- **Utilisateurs :**
  - `POST /api/ajoutuser` : inscription.
  - `POST /api/session/connection` : connexion.
  - `GET /api/session/logout` : déconnexion.
  - `POST /api/resetPassword` : réinitialisation mot de passe.
- **Chaînes :**
  - `POST /api/addchannel` : ajout.
  - `POST /api/getchannels` : liste.
  - `POST /api/getchannel` : modification.
  - `POST /api/deletechannel` : suppression.
- **Produits :**
  - `POST /api/getProducts` : liste.
  - `POST /api/addProduct` : ajout.
  - `GET /api/deleteProduct/:id` : suppression.
- **Administration :**
  - `GET /api/getUsers` : liste utilisateurs.

- `POST /api/blockUser/:id` : blocage.
- `DELETE /api/deleteUser/:id` : suppression.

## 5. Contraintes et hypothèses

### 5.1 Contraintes

- Sécurité : conformité aux critères de mots de passe et protection des sessions.
- Fichiers : images uniquement, 10MB max.
- API : respect des routes et messages d'erreur définis.

### 5.2 Hypothèses

- Pas de contraintes de budget ou de délai précis (à clarifier avec le client).
- Nombre d'utilisateurs cible : 10 000 (à valider).
- Environnements : développement, test, production (à configurer).

## 6. Livrables

- Code source (backend et frontend).
- Documentation API (ex. : OpenAPI/Swagger).
- Documentation utilisateur (guide pour créateurs, entreprises, administrateurs).
- Tests unitaires et d'intégration.
- Environnement de déploiement configuré.

## 7. Planning (hypothétique)

- **Mois 1** : Analyse, conception, maquettes UI/UX.
- **Mois 2-3** : Développement backend et frontend.
- **Mois 4** : Tests, corrections, déploiement.
- **Mois 5** : Formation utilisateurs, maintenance initiale.
- Durée totale estimée : 5 mois.

## 8. Organisation et responsabilités

- **Client** : validation des maquettes, tests, fourniture des retours.
- **\*\*Équipe's** : développement, tests, déploiement.
- **Points de contact** : chef de projet (à désigner).
- **Processus de validation** : revues hebdomadaires avec le client.

## 9. Budget

- À définir avec le client en fonction des ressources et de l'hébergement.

