

# Cahier des charges – OnlyFeed

## 1. Présentation du projet

### Contexte

Dans un environnement numérique en constante évolution, les créateurs de contenu cherchent de plus en plus à valoriser leur savoir-faire tout en bâtissant des communautés engagées. Cependant, les plateformes actuelles ne répondent pas toujours aux besoins spécifiques des créateurs en matière de monétisation, d'interaction directe avec leurs abonnés, et de contrôle sur la diffusion de leurs contenus.

OnlyFeed s'inscrit dans cette dynamique en proposant une solution innovante, inspirée des réseaux sociaux premium, permettant aux créateurs de publier des contenus exclusifs (photos, vidéos, posts créatifs) accessibles via un abonnement payant.

### Objectifs

- Offrir une plateforme simple et intuitive permettant aux créateurs de contenu de publier, gérer et monétiser leurs créations de manière sécurisée
- Mettre à disposition des abonnés un espace exclusif où ils peuvent découvrir, soutenir et interagir avec leurs créateurs préférés
- Intégrer un système de paiement fiable et sécurisé (Stripe), garantissant une rémunération transparente pour les créateurs
- Développer une application web responsive et mobile native construite avec Flutter, utilisant un backend performant en Go et une infrastructure cloud scalable (Supabase, Amazon S3)
- Appliquer les bonnes pratiques de sécurité et de protection des données (RGPD)
- Fournir une expérience professionnelle de gestion de projet logiciel

### Positionnement

Réseau social premium orienté créateurs, avec un focus particulier sur le contenu visuel et créatif.

## 2. Étude de faisabilité

### 2.1 Analyse du marché

Le marché des plateformes de contenu premium est en forte croissance :

- OnlyFans : 170M d'utilisateurs, 1.5M de créateurs
- Patreon : 250k créateurs actifs, revenus de 1.2 milliard \$ versés
- Marché français du contenu numérique : 2.8 milliards €/an

### Opportunités identifiées :

- Manque de plateformes spécialisées dans le contenu créatif premium
- Demande croissante pour des interactions directes créateur-audience
- Besoin de solutions de monétisation alternatives pour les créateurs

## 2.2 Analyse des risques et contraintes

### Risques techniques :

- Sécurité des paiements (Impact: Fort, Probabilité: Moyenne)
- Scalabilité de l'infrastructure (Impact: Moyen, Probabilité: Moyenne)
- Conformité RGPD (Impact: Fort, Probabilité: Faible)

### Contraintes budgétaires :

- Budget développement : 0€ (projet étudiant)
- Coûts d'infrastructure : Google Cloud gratuit, Firebase gratuit
- Services tiers : Stripe (2.9% + 0.25€/transaction)

### Contraintes temporelles :

- Durée du projet : 4 mois
- Équipe : 3 développeurs
- Date de livraison : [Date de soutenance]

## 2.3 Veille technologique et justification des choix

### Système de paiement :

- **Stripe** vs PayPal vs Square
- **Choix : Stripe** - Meilleure intégration, commission compétitive (2.9%), documentation complète

### Authentification :

- **Supabase Auth** vs Auth0 vs Firebase Auth
- **Choix : Supabase Auth** - Intégration native avec notre BDD, gratuit jusqu'à 50k utilisateurs

### Stockage de fichiers :

- **Amazon S3** vs Google Cloud Storage vs Cloudinary
- **Choix : Amazon S3** - Fiabilité, coût abordable, intégration Supabase

### Frontend :

- **Flutter** vs React vs Vue.js
- **Choix : Flutter** - Cross-platform (web + mobile), performance native, single codebase

## **3. Spécifications techniques et fonctionnelles**

### **3.1 Acteurs du système**

#### **Créateurs de contenu :**

- Publier du contenu (gratuit/payant)
- Gérer les abonnements et prix
- Consulter les statistiques
- Interagir avec les abonnés via commentaires et messages privés

#### **Abonnés/Utilisateurs connectés :**

- S'abonner à des créateurs
- Consulter le contenu exclusif
- Publier du contenu gratuit
- Commenter et liker
- Envoyer des messages privés

#### **Administrateurs :**

- Modérer le contenu
- Gérer les utilisateurs
- Consulter les statistiques globales

### **3.2 Cas d'usage détaillés**

#### **UC1 : Inscription utilisateur**

- Acteur : Visiteur
- Précondition : Accès à l'application
- Scénario nominal :
  1. Saisie email/mot de passe
  2. Validation email
  3. Choix du type de compte (créateur/abonné)
  4. Création du profil

#### **UC2 : Publication de contenu**

- Acteur : Créateur
- Précondition : Compte créateur activé
- Scénario nominal :
  1. Sélection du type de contenu (image/vidéo)
  2. Upload fichiers/texte
  3. Paramétrage visibilité (gratuit/payant)
  4. Publication

### UC3 : Abonnement à un créateur

- Acteur : Abonné
- Précondition : Compte validé, moyen de paiement
- Scénario nominal :
  1. Sélection du créateur
  2. Choix de l'abonnement
  3. Paiement sécurisé Stripe
  4. Accès au contenu exclusif

### UC4 : Messagerie privée

- Acteur : Abonné/Créateur
- Précondition : Utilisateur authentifié
- Scénario nominal :
  1. Recherche d'utilisateur
  2. Création/Ouverture conversation
  3. Envoi de messages texte/médias
  4. Lecture et suppression de conversations

### UC5 : Système de likes

- Acteur : Utilisateur authentifié
- Précondition : Utilisateur connecté
- Scénario nominal :
  1. Consultation d'un post
  2. Toggle like/unlike
  3. Affichage du compteur mis à jour

## 3.3 Matrice des droits

Action	Visiteur	Utilisateur connecté	Créateur	Admin
Consulter contenu public	✓	✓	✓	✓
S'abonner	X	✓	✓	✓
Publier contenu gratuit	X	✓	✓	✓
Publier contenu payant	X	X	✓	✓
Liker du contenu	X	✓	✓	✓
Messagerie privée	X	✓	✓	✓
Modérer contenu	X	X	X	✓
Consulter statistiques	X	X	✓ (siennes)	✓ (toutes)

### 3.4 Workflow principal

Inscription → Validation email → Choix profil → Configuration compte

↓

[Créateur] → Création contenu (gratuit/payant) → Publication → Monétisation

↓

[Utilisateur connecté] → Publication contenu gratuit → Navigation → Abonnement → Consultation contenu → Interaction (likes/messages)

## 4. Planification du projet

### 4.1 Équipe et rôles

**Équipe de développement :**

- **Chef de projet** : Thibaud LEFOUR - Coordination, suivi global, architecture technique
- **Développeur Full Stack/DevOPS** : Arthur DELAPORTE - Go, API REST, base de données, déploiement, Flutter, interfaces utilisateur, responsive design
- **Développeur Full Stack/DevOPS** : Alexandre MEME - Go, API REST, base de données, déploiement, Flutter, interfaces utilisateur, responsive design

**Profils nécessaires identifiés :**

- Compétences Go/API REST
- Compétences Flutter/Dart (web + mobile)
- Compétences DevOps (CI/CD, déploiement)
- Connaissances sécurité web
- Compétences fullstack (frontend + backend)
- Gestion de projet Agile

### 4.2 Planning des réunions

**Réunions hebdomadaires :**

- **Lundi 14h** : Point d'avancement hebdomadaire
- **Mercredi 16h** : Revue technique et décisions
- **Vendredi 10h** : Préparation sprint suivant

**Réunions de suivi pédagogique :**

- **Semaine 2** : Validation architecture
- **Semaine 6** : Revue mi-parcours
- **Semaine 10** : Préparation soutenance

## 4.3 Estimation des coûts

### Coûts de développement :

- Chef de projet : 300€/jour x 80 jours = 24 000€
- Développeur Full Stack senior : 300€/jour x 80 jours = 24 000€
- Développeur Full Stack junior : 250€/jour x 80 jours = 20 000€
- **Total développement : 68 000€**

### Coûts d'infrastructure :

- Hébergement backend (Google Cloud) : 0€ (tier gratuit)
- Base de données Supabase : 0€ (tier gratuit)
- Stockage S3 : 0€ (tier gratuit AWS)
- Frontend hosting (Firebase) : 0€
- **Total infrastructure : 0€**

### Coûts de services :

- **Commission Stripe** :  $2,9\% + 0,25\text{€} = (5\text{€} \times 2,9\%) + 0,25\text{€} = 0,395\text{€}$
- Monitoring Google Cloud = 0€

**Budget total projet : 68 000€**

## 4.4 Gestion de projet

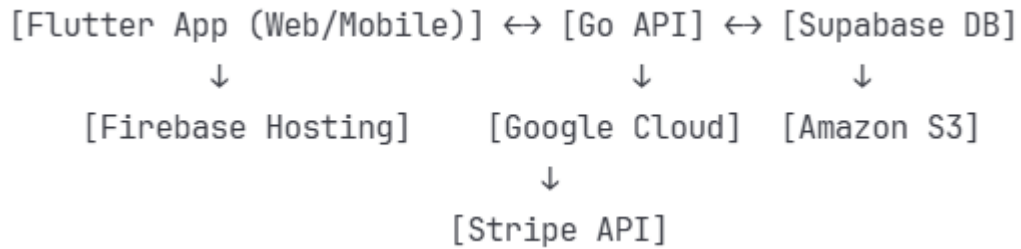
**Outil choisi : Trello**

**Méthode : Agile/Scrum**

- Sprints de 2 semaines
- Daily standups (async sur Discord)
- Sprint review/retrospective

## 5. Conception technique

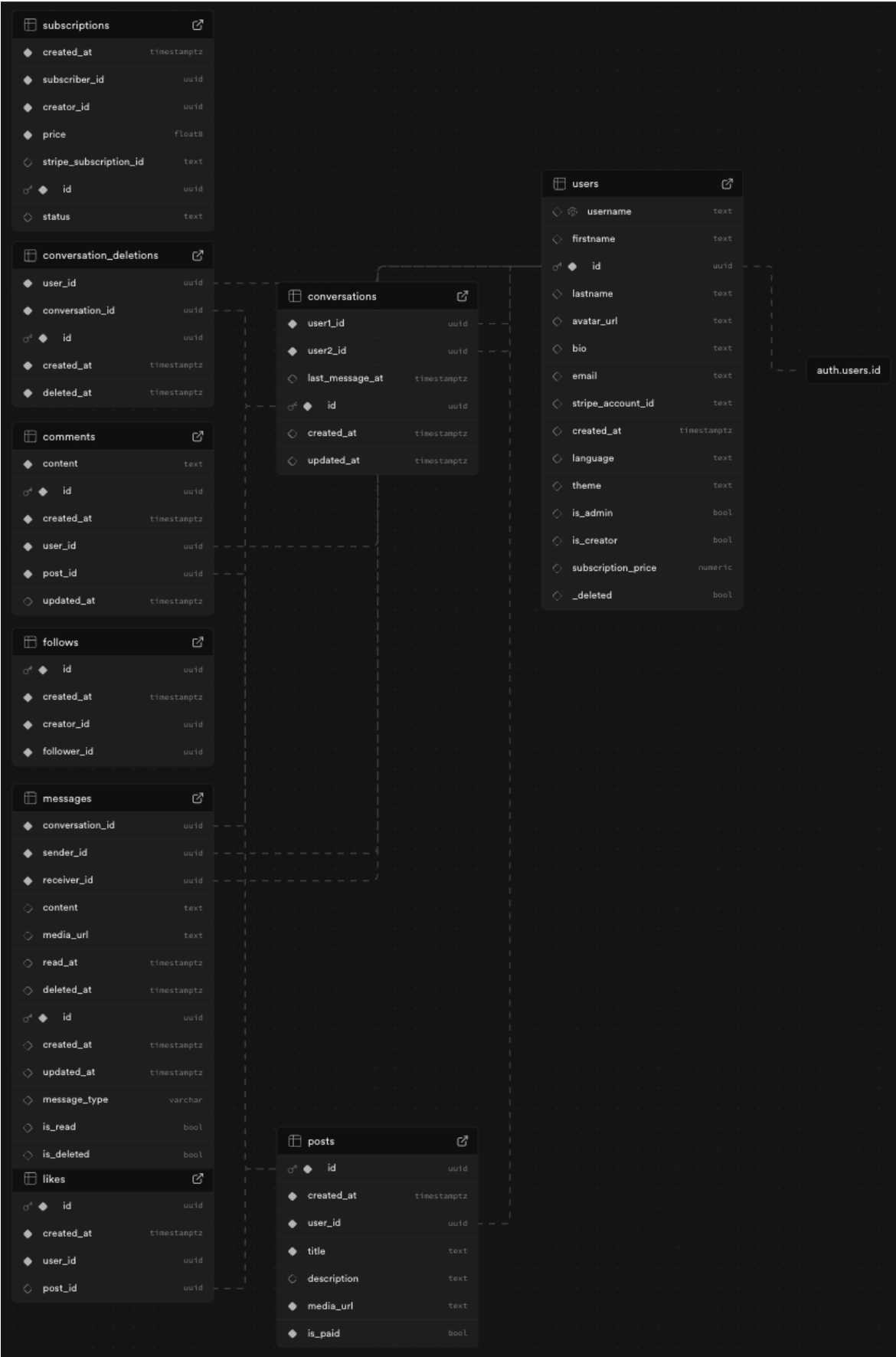
### 5.1 Architecture globale



Mockup :

<https://www.figma.com/design/qA5WMeM5wn68dYbWRN9lhu/Mockup-OnlyFeed?node-id=0-1&t=EYr039ualX7F4uNK-1>

BDD :





## 5.2 Schéma de base de données

### Tables principales :

- **users** : Informations utilisateurs (id, username, email, avatar\_url, bio, is\_creator, subscription\_price, stripe\_account\_id)
- **posts** : Publications (id, user\_id, title, description, media\_url, is\_paid, created\_at)
- **subscriptions** : Abonnements actifs (id, subscriber\_id, creator\_id, status, stripe\_subscription\_id, price)
- **likes** : System de likes (id, user\_id, post\_id, created\_at)
- **comments** : Commentaires (id, post\_id, user\_id, content, created\_at)
- **follows** : Système de suivi (id, follower\_id, creator\_id, created\_at)
- **conversations** : Conversations privées (id, user1\_id, user2\_id, last\_message\_at)
- **messages** : Messages privés (id, conversation\_id, sender\_id, receiver\_id, content, message\_type, media\_url, is\_read)

## 5.3 Stack technique détaillée

### Backend (Go) :

- Framework : Gin
- Base de données : PostgreSQL (Supabase)
- Authentification : Supabase Auth + JWT
- Storage : Amazon S3
- Paiements : Stripe
- Logs : JSON structurés

### Frontend (Flutter) :

- Framework : Flutter 3.x
- État : Provider
- Routing : Go Router
- HTTP : Dio
- Traductions : Easy Localization
- Themes : Material Design 3

### Infrastructure :

- Backend : Google Cloud Platform (tier gratuit)
- Frontend : Firebase Hosting
- Database : Supabase PostgreSQL
- Storage : Amazon S3
- Monitoring : Google Cloud

## 6. Développement et mise en production

### 6.1 Tests

Tests unitaires (Go) - 6 tests :

**TestIsFollowing :**

- **Test 1 : User\_is\_following** - Vérifie qu'un utilisateur qui suit un autre utilisateur est correctement détecté comme "suiveur" dans la base de données
- **Test 2 : User\_is\_not\_following** - Vérifie qu'un utilisateur qui ne suit pas un autre utilisateur retourne bien "false" quand aucun enregistrement n'est trouvé

**TestIsAdmin :**

- **Test 3 : User\_is\_admin** - Vérifie qu'un utilisateur ayant le statut administrateur dans la base de données est correctement identifié comme admin
- **Test 4 : User\_is\_not\_admin** - Vérifie qu'un utilisateur normal (non-admin) retourne bien "false" pour les vérifications de permissions administrateur

**TestIsSubscriberAndPrice :**

- **Test 5 : Active\_subscription\_exists** - Vérifie qu'un abonnement actif est correctement détecté et que le prix de l'abonnement est récupéré depuis la base de données
- **Test 6 : No\_subscription\_exists** - Vérifie que l'absence d'abonnement retourne bien "false" avec un prix null quand aucun abonnement n'est trouvé

**Couverture :** Ces tests valident les 3 fonctions critiques de gestion des utilisateurs (suivi, permissions admin, abonnements) avec mocking de base de données.

**Tests d'intégration (Flutter) :**

- Tests de navigation
- Tests de formulaires
- Tests d'interaction API

### 6.2 CI/CD

**Pipeline GitHub Actions :**

1. Tests automatisés (Go + Flutter)
2. Build et packaging
3. Déploiement automatique

4. Tests de smoke en production

## 6.3 Déploiement

**Backend Go** : Google Cloud Platform avec PostgreSQL **Frontend Flutter** :

- Web : Firebase Hosting
- Mobile : APK généré via GitHub Actions

## 7. Surveillance et monitoring

### 7.1 Logging

**Logs structurés (JSON)** :

- Événements utilisateur
- Erreurs d'API
- Transactions de paiement
- Performances

### 7.2 Monitoring

**Outils** :

- **Google Cloud Monitoring** : Performance du backend
- **Stripe Dashboard** : Suivi des paiements

**Alertes** :

- Taux d'erreur > 5%
- Temps de réponse > 2s
- Échec de paiement

## **8. Sécurité et conformité**

### **8.1 Sécurité**

#### **Authentification :**

- JWT tokens avec refresh via Supabase
- Hachage SHA-256 des mots de passe

#### **Données sensibles :**

- Chiffrement en transit (HTTPS)
- Variables d'environnement pour les secrets
- Validation stricte des inputs
- Upload de fichiers sécurisé vers S3

### **8.2 RGPD**

#### **Conformité :**

- Consentement cookies explicite
- Droit à l'oubli (suppression compte)
- Portabilité des données (export JSON)
- Politique de confidentialité

## **9. Fonctionnalités implémentées**

### **9.1 Fonctionnalités principales**

#### **Gestion des utilisateurs :**

- Inscription/Connexion avec Supabase Auth
- Profils utilisateurs avec avatar
- Système créateur/abonné
- Édition de profil

#### **Système de contenu :**

- Publication de posts avec médias (images/vidéos)
- Contenu gratuit/payant
- Grille de posts style Instagram
- Page de détail des posts
- Système de likes avec compteurs

#### **Système de paiement :**

- Intégration Stripe Connect pour créateurs
- Abonnements mensuels
- Gestion des webhooks Stripe
- Commission de 20% sur les abonnements

#### **Interactions sociales :**

- Système de follow/unfollow
- Messagerie privée complète
- Commentaires sur les posts
- Partage de posts via messages

#### **Interface utilisateur :**

- Design responsive web/mobile
- Thèmes clair/sombre
- Multilingue (FR/EN)
- Navigation intuitive

## **9.2 Fonctionnalités avancées**

#### **Administration :**

- Dashboard administrateur (placeholder)
- Gestion des permissions

#### **Technique :**

- API REST complète en Go
- Logs structurés JSON
- Déploiement automatisé
- Monitoring avec Sentry

## **10. Livrables**

#### **Repository GitHub :**

- Code source complet Backend (Go)
- Code source complet Frontend (Flutter)
- Documentation technique
- Tests automatisés
- Pipeline CI/CD

#### **Application en production :**

- Version web accessible : <https://onlyfeed-9f05c.web.app>
- APK Android téléchargeable
- Backend API déployé
- Base de données opérationnelle

**Documentation :**

- Cahier des charges (ce document)
- README détaillés pour installation
- Architecture technique documentée
- Présentation de maintenance

## **11. Critères de succès**

**Techniques :**

- Application fonctionnelle en production
- Couverture de tests > 70%
- Temps de réponse < 2s
- Aucune faille de sécurité critique

**Fonctionnels :**

- Inscription/connexion opérationnelle
- Publication de contenu fonctionnelle
- Système de paiement intégré
- Interface responsive web/mobile
- Messagerie privée complète
- Système de likes fonctionnel

**Pédagogiques :**

- Respect des attendus RNCP
- Travail d'équipe documenté
- Présentation professionnelle
- Livrables complets et de qualité