Projet livres.staka.fr

Documentation technique et fonctionnelle

Réalisé par : *Christophe Mostefaoui*
Rôle : *Développeur web freelance*
Client : *M. Charles Tate (Staka)*
Date de rédaction : *19 juin 2025*
Version : *1.0*

Introduction

Ce document constitue la documentation complète du projet **livres.staka.fr**, réalisée à la suite de la validation du cahier des charges établi avec M. Charles Tate.

Rédigé le 19 juin 2025, il présente de manière structurée l'ensemble des éléments fonctionnels, techniques, réglementaires et organisationnels nécessaires à la réalisation du site.

L'objectif de cette documentation est double :

1. **Assurer une parfaite compréhension commune** entre le client et le développeur sur les fonctionnalités attendues, les limites, les livrables et les modalités de travail.
2. **Servir de base de référence** pour le développement, la recette et la maintenance du projet.

Chaque section a été pensée pour anticiper les besoins futurs, garantir la sécurité et la conformité du site, et permettre des évolutions faciles dans une logique de continuité technique.

Sommaire

1. Contexte du projet
2. Objectifs
3. Détail des prestations prévues
4. Informations techniques
5. Estimation du planning
6. Budget
7. Évolutions envisagées
8. Ce qui n'est pas inclus
9. Livrables remis par le prestataire
10. Conditions de collaboration
11. Mention spéciale hébergement

Spécifications fonctionnelles

12. Besoins fonctionnels – Admin
13. Besoins fonctionnels – Client
14. Besoins non fonctionnels

Modélisation et logique

15. User stories (Client / Admin)
16. Base de données (Diagrammes UML MCD + MLD + script SQL)
17. Architecture technique (front, back, DB, Stripe, WebSocket)
18. Table des routes API principales
19. Matrice des droits et rôles

Parcours utilisateur

20. Scénarios d'usage (inscription, projet, paiement, support, RGPD, etc.)

Sécurité & RGPD

21. Logs et supervision
22. Checklist RGPD
23. Politique de sécurité

Qualité et déploiement

24. Tests & assurance qualité
25. Environnements de développement, préprod, production
26. Déploiement et hébergement

Cahier des charges – Projet livres.staka.fr

1. Contexte du projet

La société Staka représentée par M. Charles Tate, souhaite déployer un site web dédié à la correction et à l'accompagnement de travaux de recherche, accessible depuis le sous-domaine livres.staka.fr. Ce site devra respecter la charte graphique fournie et intégrer des fonctionnalités avancées, dont un système de paiement en ligne via Stripe et proposer un espace client conforme RGPD.

2. Objectifs

• Reproduire fidèlement le site vitrine actuellement visible à l'adresse https://stakalivres.netlify.app (fourni par le client, version à jour fournie sous forme d'archive .zip)
• Intégrer la charte graphique fournie via Adobe XD
• Déployer le site sur le sous-domaine livres.staka.fr (DNS géré par le client)
• Garantir un site sécurisé (certificat SSL, bonnes pratiques serveur)
• Créer un back-office simple (système administrateur) permettant la modification autonome des contenus clés
• Intégrer un module de paiement Stripe pour la vente de prestations en ligne
• Développer un espace client (création de compte, suivi commandes, RGPD)

3. Détail des prestations prévues

- **Rédaction du cahier des charges :** élaboration d'un document structuré récapitulant les besoins fonctionnels et techniques du projet, validé avant démarrage.
- Configuration du sous-domaine livres.staka.fr avec certificat SSL
- Intégration du front-end (HTML/CSS/JS) fourni par le client en respectant la maquette
- Création d'un espace d'administration sécurisé permettant la modification des textes, titres, visuels (images), tarifs ou services si applicable.
  Le back-office permettra la modification des textes, titres, visuels (images) et éventuellement les prix ou services si applicable.
- **Développement d'un espace client RGPD**
  - Création de compte (email, mot de passe sécurisé, consentement)
  - Authentification et gestion du profil
  - Historique des commandes, accès aux reçus
  - Export et suppression des données personnelles (droit d'accès et d'effacement)

    - Journalisation des consentements
    - Sécurité conforme (hash bcrypt, HTTPS, logs)

- Intégration d'un module de paiement en ligne Stripe sécurisé, connexion au compte Stripe du client (clé à fournir)
- Adaptation responsive du site (mobile, tablette, desktop)
- Tests de validation, corrections mineures, mise en ligne
- **Synthèse détaillée finale du projet**
  Document remis à la fin de la mission, décrivant les livrables fournis, l'architecture technique et les accès associés.

4. Informations techniques

• Nom de domaine principal : staka.fr (géré par le client)
• Sous-domaine cible : livres.staka.fr
• Hébergement (à souscrire par le client, coût estimé entre 5 et 10€/mois) : à déterminer en fonction des choix techniques (Netlify, Vercel, OVH, etc.)
• Langages et outils : HTML/CSS/JS (fourni), intégration back-end personnalisé (Node JS + Mysql), module Stripe ; Admin : interface custom sécurisée ; client : authentification sécurisée, dashboard, RGPD

5. Estimation du Planning

• Démarrage des travaux : à partir du 16 juin 2025
• Livraison estimée : environ 4 semaines après réception des éléments et validation du devis

6. Budget

• Un devis a été établi séparément pour cette prestation, à hauteur de 1000 € TTC (TVA non applicable, art. 293B du CGI)

7. Évolutions envisagées

Le projet livres.staka.fr est le premier d'une série de sites satellites, avec les suivants déjà identifiés : coach.staka.fr, dba.staka.fr, et dec.staka.fr. L'objectif est de créer une base réutilisable pour assurer une continuité graphique et technique sur l'ensemble des déclinaisons futures.

8. Ce qui n’est pas inclus

– Le référencement naturel avancé (SEO technique ou sémantique)
– La maintenance continue après livraison (possibilité de devis séparé)

Je m’engage à vous assurer une maintenance gratuite de 14 jours en cas de disfonctionnement ou d’éventuels problèmes rencontrés sur le travail fourni, à compter de la date de livraison.

– La configuration du compte Stripe (à fournir par le client)

– Fourniture des images, textes finaux, et accès légaux (CGV, mentions)

9. Livrables remis par le prestataire

Les livrables finaux comprendront :

– Le site web fonctionnel et déployé sur livres.staka.fr
– Un accès administrateur sécurisé – Un accès client conforme RGPD
– Une synthèse technique du projet (PDF)
– Le code source (archive ou dépôt Git)
– Un support de 14 jours après mise en ligne (corrections mineures incluses)

10. Conditions de collaboration

Le projet sera réalisé en tant que prestation freelance. Toute modification significative du périmètre fera l’objet d’un avenant ou devis complémentaire.

11. Mention spéciale hébergement

Comme précisé dans le devis, les frais d’hébergement sont à la charge du client et non compris dans la présente prestation.

Les besoins fonctionnels

Admin

L'interface administrateur permet de gérer de façon centralisée tous les projets, fichiers, factures, paiements, profils clients, demandes de support et contenus publics du site.
Toutes les actions sont conformes aux exigences RGPD, et chaque opération impactant un client (suppression, désactivation, notification) est parfaitement synchronisée entre l'espace client et l'administration.
L'admin peut mettre à jour les pages d'information, tarifs, et ressources.
Un historique complet des commandes, paiements et interactions est disponible pour le suivi et le support.

Authentification

- Connexion sécurisée à l'espace administrateur (identifiant/mot de passe)

Gestion des commandes/projets

- Accès à la liste des projets soumis par les clients (avec recherche/filtre)
- Consultation des détails d'un projet :
    - Informations client
    - Titre, type, nombre de pages, pack choisi
    - Statut du projet (en attente, en correction, terminé)
    - Date de début, date prévue/eff effective de livraison
- Téléchargement du manuscrit original envoyé par le client
- Possibilité de voir l'**historique des statuts**
- Dépôt du ou des fichiers corrigés (Word, PDF, ZIP…) dans l'espace projet correspondant
- Mise à jour du statut du projet ("en correction", "correction terminée", etc.)
- Pas de gestion multi-correcteur (champ texte suffisant)

Gestion des fichiers

- Visualisation, téléchargement et gestion de tous les fichiers rattachés aux projets clients (originaux, corrections, annexes)
- Synchronisation totale des suppressions (client = admin = suppression totale)

Facturation & Paiement

- Consultation des factures associées à chaque projet/client
- Vérification des paiements (état payé ou en attente, selon retour Stripe)
- Téléchargement des factures générées automatiquement après le paiement
- Possibilité de voir l'historique des paiements par client/projet (pour audit/réclamation si besoin)

Gestion des contenus (pages publiques du site)

- Modification des contenus informatifs accessibles côté client :
  - Pages d'information
  - Tarifs, packs, options affichées
- Mise à jour des textes ou ressources (PDF, liens), pouvoir uploader/remplacer des ressources (PDF, images) côté admin

Gestion des comptes clients

- Consultation des profils clients inscrits (liste, recherche)
- Suppression définitive d'un compte client et de ses données associées (conformité RGPD)
- Export des données client sur demande
- Consultation de l'historique de commandes et de paiements d'un client

Support client

- Accès à la messagerie/support pour consulter les échanges avec les clients (si le système de chat/support passe par la même base)
- Réponse directe aux messages/support envoyés par les clients

Notifications

- Notification interne (ou par email) à chaque :
  - Nouvelle commande déposée
  - Paiement reçu
  - Message support reçu
- Gestion de l'historique des notifications pour l'admin

Sécurité & RGPD

- Suppression globale des données d'un client sur demande
- Export des données d'un client en cas de demande RGPD

Notes

- Aucune gestion de correcteurs multiples ou d'équipe n'est explicitement demandée, donc :
  - Attribution d'un correcteur à un projet : simple champ à renseigner, pas de gestion utilisateur/correcteur complexe.
- Pas de paramétrage dynamique de packs/options ou de gestion avancée des tarifs dans l'admin, juste édition du contenu affiché côté client.
- Toutes les opérations d'upload et de suppression de fichiers sont synchronisées entre l'espace client et admin (suppression client = suppression réelle).

Client

Authentification

- Création de compte avec prénom, nom, téléphone email et mot de passe
- Connexion à l'espace client
- Mot de passe oublié

Tableau de bord

- Vue synthétique :
  - Projets actifs
  - Projets terminés
  - Messages non lus
  - Note de satisfaction
- Liste des projets en cours (nom, type, nb pages, statut, progression, date de livraison, correcteur)
- Accès rapide :
  - Nouveau projet
  - Contacter l'équipe
- Accès aux notifications depuis le tableau de bord ("cloche" visible sur toutes les pages)

Gestion de projets

- Création d'un projet :
  - Titre
  - Type de manuscrit
  - Nombre de pages
  - Choix d'un pack (Correction seule, Pack intégral, Pack KDP)
  - Prix affiché selon le pack choisi
  - Description du projet
  - Upload d'un ou plusieurs fichiers manuscrits (.doc, .docx, .pdf, max 10 Mo)
- Suivi de la progression (barre de progression, statut)
- Visualisation des détails (dates, pack choisi, correcteur assigné)
- Téléchargement des fichiers corrigés une fois la correction terminée
- Noter un projet terminé (notation sur 5)
- Historique des projets passés et accès à tous les fichiers liés à un projet, pas seulement les corrections finales

Messagerie

- Messagerie interne :
  - Conversations avec le correcteur et le support
  - Historique accessible en permanence
  - Affichage de notifications de nouveaux messages
  - Messages instantanés (pièces jointes possibles dans la messagerie)
  - Les conversations sont liées à un projet OU globales, selon le contexte

Gestion des fichiers

- Accès à la liste des fichiers liés à ses projets (manuscrits originaux, corrections, couvertures…)
- Téléchargement et aperçu des fichiers disponibles
- Possibilité d'uploader de nouveaux fichiers (plusieurs) via le formulaire de création de projet
- Ajout de fichiers possible tout au long de la vie du projet

Facturation & Paiement

- Facture générée après paiement (téléchargeable en PDF)
- Paiement en ligne via Stripe (paiement unique, pas d'abonnement)
- Historique des factures et paiements
- Ajout/suppression de carte bancaire comme moyen de paiement
- Récap annuel :
    - Projets complétés
    - Pages corrigées
    - Total dépensé
    - Économies réalisées (statut VIP si applicable)

Support & FAQ

- FAQ statique consultable à tout moment
- Système de chat en direct avec l'équipe de support
- Envoi d'une demande de support via un formulaire
  (sujet, message, pièce jointe possible dans le support uniquement)
- Coordonnées du support visibles (email, téléphone, horaires)
- Accès à des ressources utiles (guides, tutoriels, FAQ PDF…)

Notifications

- Notification par email à chaque :
    - Changement d'état d'un projet
    - Nouveau message reçu
    - Mise à disposition d'un fichier corrigé
- Notification PUSH et SMS activables dans les paramètres
- Gestion des préférences de notifications (types et canaux) dans les paramètres
- Historique des notifications accessible à tout moment via l'icône dédiée

Gestion des données & RGPD

- Suppression de toutes les données et fichiers du compte client en un clic
  (Suppression côté client = suppression totale des données côté admin et base de données)
- Désactivation temporaire possible (données conservées, accès bloqué)
- Téléchargement/export des données personnelles en un clic
- Consentement explicite pour l'analyse d'usage anonymisée (statistique)
- Paramétrage de la visibilité du profil utilisateur (public/privé)

Les besoins non fonctionnels

Sécurité

- Connexion client et admin sécurisée (HTTPS, SSL obligatoire)
- Mots de passe stockés de façon sécurisée (hashés, jamais en clair)
- Accès aux espaces (client/admin) protégé par authentification obligatoire
- Séparation stricte des droits client/admin (un client ne peut accéder à aucune donnée admin ni d'un autre client)
- Accès aux fichiers uniquement pour le propriétaire ou l'admin
- Toutes les données sensibles (fichiers, emails, factures) stockées sur le serveur, jamais exposées publiquement
- Protection contre les attaques courantes (OWASP) :
  "Protection contre les attaques XSS, CSRF, injection SQL, brute-force sur les formulaires d'authentification."
- Gestion des tentatives de connexion :
  "Blocage temporaire après plusieurs tentatives de connexion échouées."
- Expiration de session :
  "Déconnexion automatique après 15 minutes d'inactivité."
- Logs d'accès et alertes sécurité (utile pour audit ou détection d'accès anormal).

RGPD & Confidentialité

- Conformité RGPD : droit à l'oubli (suppression totale), suppression irréversible des fichiers et données sur demande
- Conservation des données limitée à la durée strictement nécessaire à la prestation
- Consentement explicite à la création de compte
- Journalisation des actions sensibles (création/suppression de compte, paiement)
- Gestion des cookies :
  "Consentement explicite à l'utilisation de cookies non nécessaires."
- Procédure d'export des données personnelles :
  "Export des données sur demande utilisateur dans un format lisible (PDF, ZIP, JSON)."

Performance

- Temps de chargement inférieur à 2s pour toutes les pages principales sur connexion fibre/ADSL
- Upload et téléchargement de fichiers (jusqu'à 10 Mo) en moins de 30 s
- Mise en cache des contenus publics (FAQ, pages informatives) pour réduire la charge serveur

Disponibilité & Sauvegarde

- Disponibilité du service : 99 % sur l'année (hors maintenance annoncée)
- Sauvegarde quotidienne de la base de données et des fichiers clients
- Procédure de restauration documentée

Interopérabilité

- Formats de fichiers supportés à l'upload : .doc, .docx, .pdf uniquement
- Intégration avec Stripe en mode paiement unique
- Les notifications email doivent être envoyées via un service réputé (type Sendgrid, Mailgun, etc.)

Ergonomie & Accessibilité

- Responsive : interface utilisable sur ordinateur, tablette, mobile
- Respect des standards d'accessibilité web (labels, contrastes, navigation clavier)

Support & maintenance

- Logs d'erreurs serveur consultables pour l'admin/technicien
- Mise à jour du service sans interruption visible pour les clients (rolling update)

Les User stories

User story Client

Authentification

- En tant que visiteur, je peux créer un compte avec mon email et un mot de passe afin d'accéder à l'espace client.
- En tant que client, je peux me connecter avec mon email et mon mot de passe afin de gérer mes projets.
- En tant que client, je peux réinitialiser mon mot de passe si je l'ai oublié.

Gestion de projets

- En tant que client, je peux créer un nouveau projet de correction en renseignant toutes les informations demandées et en envoyant un fichier manuscrit.
- En tant que client, je peux suivre l'avancement de mes projets sur un tableau de bord (statut, progression, dates).
- En tant que client, je peux joindre plusieurs fichiers à la création d'un projet.
- En tant que client, je peux ajouter de nouveaux fichiers à un projet existant à tout moment.
- En tant que client, j'ai accès à l'historique de tous mes projets et à tous les fichiers associés.
- En tant que client, je peux télécharger les fichiers corrigés dès qu'ils sont disponibles.
- En tant que client, je peux attribuer une note/évaluation à un projet une fois terminé.

Messagerie

- En tant que client, je peux envoyer et recevoir des messages instantanés avec pièces jointes avec mon correcteur ou le support via l'interface dédiée.
- En tant que client, je peux retrouver l'historique de mes conversations à tout moment.

Gestion des fichiers

- En tant que client, je peux accéder à tous les fichiers liés à mes projets (manuscrits, corrections, couvertures) dans un espace dédié.
- En tant que client, je peux télécharger chaque fichier depuis cet espace.

Facturation & Paiement

- En tant que client, je peux payer ma commande en ligne de manière sécurisée par carte bancaire (Stripe).
- En tant que client, je peux télécharger la facture correspondante.
- En tant que client, je peux consulter l'historique de toutes mes factures et paiements.
- En tant que client, je peux enregistrer et supprimer mes moyens de paiement (carte bancaire).

- En tant que client, je peux consulter un récapitulatif annuel de mes projets, pages corrigées, dépenses et économies (statut VIP).

Support & FAQ

- En tant que client, je peux consulter la FAQ et les ressources utiles depuis mon espace.
- En tant que client, je peux contacter l'équipe de support via un formulaire ou un chat en direct.

Notifications

- En tant que client, je reçois un email de notification à chaque :
  - Changement d'état d'un projet,
  - Nouveau message reçu,
  - Fichier corrigé disponible.
- En tant que client, je peux choisir mes canaux de notification (email, push, SMS) et le type de notifications à recevoir.
  En tant que client, j'accède à l'historique de toutes mes notifications via l'interface.

Gestion des données personnelles

- En tant que client, je peux supprimer définitivement mon compte et toutes mes données personnelles en un clic.
- En tant que client, je peux désactiver temporairement mon compte (mes données restent conservées, mais l'accès est bloqué).
- En tant que client, je peux télécharger/exporter toutes mes données personnelles en un clic.
- En tant que client, je peux gérer la visibilité de mon profil (public/privé).
- En tant que client, je peux donner ou retirer mon consentement à l'analyse anonyme de mes usages (statistiques).

User story Admin

- En tant qu'admin, je peux me connecter à l'espace d'administration avec un identifiant et un mot de passe.
- En tant qu'admin, je peux accéder à la liste de tous les projets soumis par les clients, filtrer et rechercher un projet.
- En tant qu'admin, je peux consulter les détails d'un projet (informations client, statut, dates, pack choisi).
- En tant qu'admin, je peux télécharger le fichier manuscrit envoyé par le client.
- En tant qu'admin, je peux uploader le(s) fichier(s) corrigé(s) dans l'espace projet.
- En tant qu'admin, je peux changer le statut d'un projet (en correction, terminé…).
- En tant qu'admin, je peux saisir ou modifier le nom du correcteur assigné à chaque projet.
- En tant qu'admin, je peux accéder à tous les fichiers liés à tous les projets.
- En tant qu'admin, je peux consulter les factures et paiements de chaque projet/client.
- En tant qu'admin, je peux modifier les contenus publics du site : pages informatives, FAQ, tarifs/packs.
- En tant qu'admin, je peux consulter la liste des clients, leurs profils, leurs commandes, et supprimer un compte et toutes ses données (conformité RGPD).
- En tant qu'admin, je peux exporter toutes les données liées à un client en cas de demande RGPD.
- En tant qu'admin, je peux répondre aux messages/support reçus depuis la messagerie interne.
- En tant qu'admin, je reçois une notification (interne ou email) à chaque nouvelle commande, nouveau paiement, ou message support reçu.
- En tant qu'admin, je peux consulter l'historique des notifications reçues dans l'interface d'administration.

La base de données

La base de données a été conçue pour répondre précisément à l'ensemble des besoins fonctionnels du projet Staka Livres, en tenant compte des usages côté client et côté administrateur. Elle garantit à la fois sécurité, simplicité de gestion, évolutivité, et conformité RGPD.

Présentation des principales entités

1. Utilisateurs (User)

Cette table stocke toutes les informations relatives aux utilisateurs (clients et administrateurs) :

- **Informations personnelles** : prénom, nom, email, téléphone, mot de passe (crypté).
- **Rôle** : distingue un client d'un admin (sécurité).
- **Préférences** : gestion des notifications (email, push, SMS), consentement pour l'analyse anonymisée, visibilité du profil, etc.
- **Statut** : actif, désactivé, supprimé (conformité RGPD).

2. Projets (Project)

Chaque projet de correction correspond à une commande unique du client :

- **Informations sur le manuscrit** : titre, type, nombre de pages, description, pack choisi, statut (en attente, en cours, terminé…).
- **Dates importantes** : création, début, livraison prévue/effective.
- **Historique et notation** : correcteur assigné, note de satisfaction.
- **Lien direct avec le client (user_id)** et tous les fichiers du projet.

3. Fichiers (File)

Tous les fichiers (manuscrits originaux, corrections, annexes, pièces jointes) sont gérés ici :

- **Métadonnées** : nom, type, taille, lien sécurisé, date d'upload.
- **Lien avec le projet et l'uploader** (user_id, project_id).
- **Possibilité de rattacher un fichier à un message (via Message_attachment)**.

4. Factures (Invoice)

Chaque projet génère une facture unique après paiement :

- **Informations** : montant, statut (payé/en attente), lien PDF, date d'émission et de paiement.
- **Lien avec le projet et le client**.

5. Moyens de paiement (PaymentMethod)

Gestion des cartes bancaires Stripe associées à un compte client :

- **Données** : type, marque, 4 derniers chiffres, expiration, statut (actif/inactif).
- **Possibilité d'avoir plusieurs moyens par utilisateur** (ex : carte pro et perso).

6. Messagerie (Message, Message_attachment)

Permet la communication entre le client et l'admin/support/correcteur :

- **Message** : contenu texte, horodatage, statut lu/non lu, rattachement à un projet ou une demande support.
- **Message_attachment** : gestion des pièces jointes par message (un ou plusieurs fichiers).
- **Historique complet et centralisé.**

7. Support & tickets (SupportRequest)

Gestion des demandes d'aide/support :

- **Sujet, type (problème technique, facturation, projet, etc.), statut, date de clôture**.
- **Liée à la messagerie pour le suivi des échanges.**

8. Notifications

Gestion centralisée de toutes les notifications reçues par l'utilisateur :

- **Type** (nouveau message, changement de statut, fichier disponible…), canal (email, push, SMS), date, lu/non lu.
- **Lien avec les événements importants de l'espace client**.

9. Pages informatives (Page)

Contenus publics éditables côté admin :

- **Titre, contenu HTML/texte, ordre d'affichage, statut actif/inactif**.
- **Permet de modifier facilement les pages d'informations du site.**

Relations et logique d'ensemble

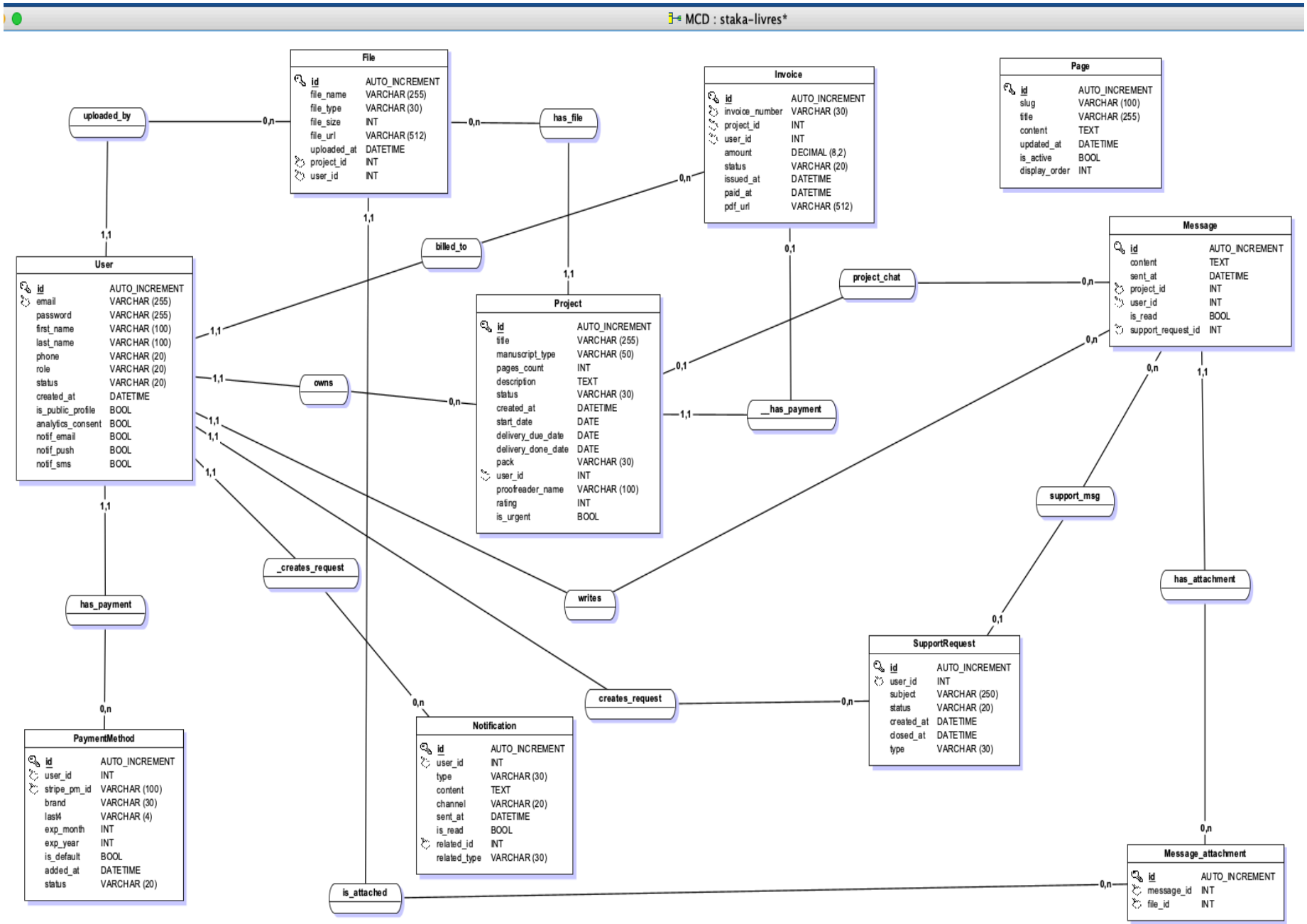
- **Chaque utilisateur** peut gérer plusieurs projets, moyens de paiement, fichiers, messages, notifications, et tickets support.
- **Chaque projet** est lié à un client, un ou plusieurs fichiers, une facture, des messages, et des notifications.
- **Chaque message** peut comporter des pièces jointes, et être lié à un projet ou une demande support.
- **La suppression d'un compte entraîne la suppression de toutes les données liées** (projets, fichiers, messages…), pour garantir la conformité RGPD.

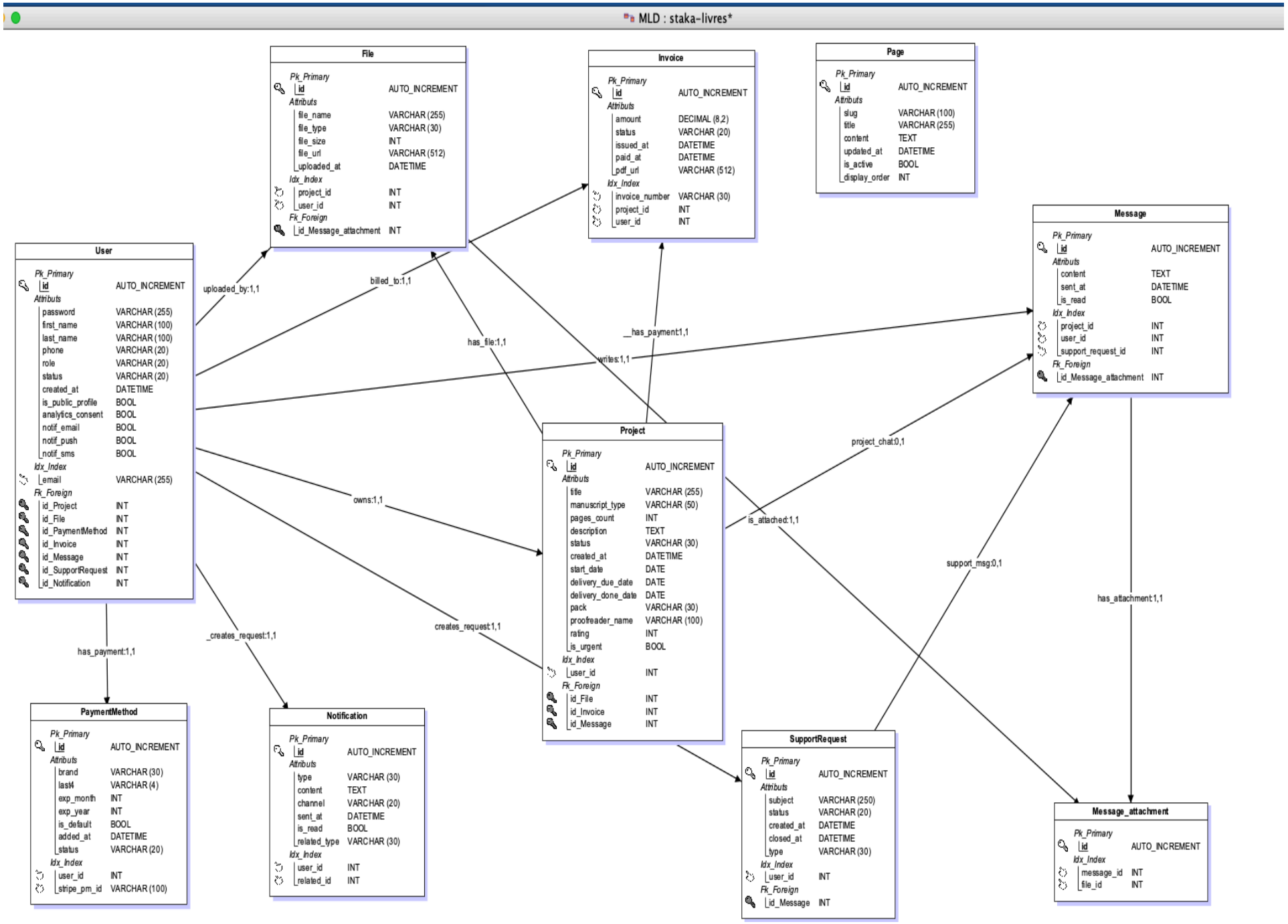
Avantages de la structure

- **Clarté** : chaque information est stockée une seule fois, pas de doublon.
- **Traçabilité** : toutes les actions (création, modification, suppression) sont historisées et rattachées à un utilisateur.
- **Sécurité** : aucune donnée critique n'est exposée publiquement, gestion stricte des droits d'accès.
- **Évolutivité** : la structure permet facilement d'ajouter de nouvelles fonctionnalités ou types de contenus.

Conclusion

Ce modèle relationnel permet une gestion fluide, sécurisée et conforme aux exigences du projet.
Il assure aussi une maintenance simplifiée et une adaptation possible pour des évolutions futures.




MLD : staka-livres*
File
Pk_Primary
id AUTO_INCREMENT
Attributs
file_name VARCHAR (255)
file_type VARCHAR (30)
file_size INT
file_url VARCHAR (512)
uploaded_at DATETIME
Idx_Index
project_id INT
user_id INT
Fk_Foreign
id_Message_attachment INT
Invoice
Pk_Primary
id AUTO_INCREMENT
Attributs
amount DECIMAL (8,2)
status VARCHAR (20)
issued_at DATETIME
paid_at DATETIME
pdf_url VARCHAR (512)
Idx_Index
invoice_number VARCHAR (30)
project_id INT
user_id INT
Page
Pk_Primary
id AUTO_INCREMENT
Attributs
slug VARCHAR (100)
title VARCHAR (255)
content TEXT
updated_at DATETIME
is_active BOOL
display_order INT
Message
Pk_Primary
id AUTO_INCREMENT
Attributs
content TEXT
sent_at DATETIME
is_read BOOL
Idx_Index
project_id INT
user_id INT
support_request_id INT
Fk_Foreign
id_Message_attachment INT
User
Pk_Primary
id AUTO_INCREMENT
Attributs
password VARCHAR (255)
first_name VARCHAR (100)
last_name VARCHAR (100)
phone VARCHAR (20)
role VARCHAR (20)
status VARCHAR (20)
created_at DATETIME
is_public_profile BOOL
analytics_consent BOOL
notif_email BOOL
notif_push BOOL
notif_sms BOOL
Idx_Index
email VARCHAR (255)
Fk_Foreign
id_Project INT
id_File INT
id_PaymentMethod INT
id_Invoice INT
id_Message INT
id_SupportRequest INT
id_Notification INT
Project
Pk_Primary
id AUTO_INCREMENT
Attributs
title VARCHAR (255)
manuscript_type VARCHAR (50)
pages_count INT
description TEXT
status VARCHAR (30)
created_at DATETIME
start_date DATE
delivery_due_date DATE
delivery_done_date DATE
pack VARCHAR (30)
proofreader_name VARCHAR (100)
rating INT
is_urgent BOOL
Idx_Index
user_id INT
Fk_Foreign
id_File INT
id_Invoice INT
id_Message INT
PaymentMethod
Pk_Primary
id AUTO_INCREMENT
Attributs
brand VARCHAR (30)
last4 VARCHAR (4)
exp_month INT
exp_year INT
is_default BOOL
added_at DATETIME
status VARCHAR (20)
Idx_Index
user_id INT
stripe_pm_id VARCHAR (100)
Notification
Pk_Primary
id AUTO_INCREMENT
Attributs
type VARCHAR (30)
content TEXT
channel VARCHAR (20)
sent_at DATETIME
is_read BOOL
related_type VARCHAR (30)
Idx_Index
user_id INT
related_id INT
SupportRequest
Pk_Primary
id AUTO_INCREMENT
Attributs
subject VARCHAR (250)
status VARCHAR (20)
created_at DATETIME
closed_at DATETIME
type VARCHAR (30)
Idx_Index
user_id INT
Fk_Foreign
id_Message INT
Message_attachment
Pk_Primary
id AUTO_INCREMENT
Idx_Index
message_id INT
file_id INT
uploaded_by:1,1
billed_to:1,1
has_file:1,1
__has_payment:1,1
writes:1,1
owns:1,1
has_payment:1,1
_creates_request:1,1
creates_request:1,1
project_chat:0,1
is_attached:1,1
support_msg:0,1
has_attachment:1,1

MLD (Modèle Logique de Données)

Ce schéma illustre l'organisation complète de la base de données du projet de correction de manuscrits.
Chaque **table** (rectangle) représente un type d'information principale ou une fonctionnalité du site : utilisateur, projet, fichiers, factures, messages, demandes de support, notifications, moyens de paiement, pages de contenu, etc.

Les **liens** entre les tables (traits) indiquent les relations entre ces informations (ex : un utilisateur possède plusieurs projets, un projet contient plusieurs fichiers…).

Ce modèle garantit une gestion claire, sécurisée et évolutive de toutes les données de la plateforme :

- Suivi détaillé des projets et des échanges,
- Gestion rigoureuse des fichiers et factures,
- Sécurité et traçabilité des actions utilisateurs,
- Respect des besoins RGPD et gestion simplifiée des suppressions de comptes et données.

Chaque relation et chaque champ a été conçu pour correspondre exactement aux besoins définis ensemble, sans ajouter de fonctionnalités inutiles ou non prévues.

Légende des schémas

- **Table (rectangle)** :
  Représente une famille d'informations (par exemple : *User* = compte utilisateur, *Project* = projet client, *File* = fichier, etc.).
- **Clé primaire** (pk / *id*) :
  Identifiant unique de chaque ligne dans une table.
- **Clé étrangère** (fk / *user_id*, *project_id*...) :
  Permet de lier une table à une autre (ex : à quel utilisateur appartient un projet).
- **Cardinalités** (0,n / 1,1...) :
  - **1,1** : chaque élément doit être lié à un et un seul élément d'une autre table (ex : une facture est toujours liée à un projet).
  - **0,n** : un élément peut être lié à aucun ou à plusieurs éléments d'une autre table (ex : un utilisateur peut avoir plusieurs projets, ou aucun).
  - **0,1** : un élément peut être lié à aucun ou un seul autre élément.
- **Trait entre tables** :
  Représente une relation logique (ex : un message peut contenir plusieurs fichiers joints).

Lecture guidée

- **Un utilisateur** (*User*) :
  - Peut créer plusieurs projets (*Project*), avoir plusieurs moyens de paiement (*PaymentMethod*), recevoir des notifications (*Notification*), envoyer/recevoir des messages (*Message*), et faire des demandes de support (*SupportRequest*).
- **Un projet** (*Project*) :
  - Est associé à un utilisateur, contient plusieurs fichiers (*File*), reçoit des messages (*Message*), et génère des factures (*Invoice*).
- **Un fichier** (*File*) :
  - Est lié à un projet et à l'utilisateur qui l'a uploadé. Peut aussi être joint à un message.
- **Un message** (*Message*) :
  - Est lié à un projet ou une demande de support, et peut avoir des fichiers joints (*Message_attachment*).
- **Une notification** (*Notification*) :
  - Est associée à un utilisateur, et peut pointer vers un projet, une facture, un message, etc.
- **Une facture** (*Invoice*) :
  - Est liée à un projet et à l'utilisateur concerné.
- **Une demande de support** (*SupportRequest*) :
  - Est liée à un utilisateur et regroupe les échanges associés.
- **Un moyen de paiement** (*PaymentMethod*) :
  - Est propre à chaque utilisateur.
- **Une page** (*Page*) :
  - Représente un contenu éditorial ou informatif géré par l'admin.
- **Les relations d'attachement** (*Message_attachment*) :
  - Permettent à chaque message d'avoir plusieurs fichiers joints.

Ce schéma constitue la base technique du projet et garantit la cohérence, la sécurité et la clarté de toutes les données manipulées par la plateforme.

Les diagrammes de cas d'utilisation

Ces diagrammes de cas d'utilisation mettent en évidence les **grandes fonctionnalités** de la plateforme Staka Livres et le **rôle de chaque acteur** (client, administrateur).

Côté client

Le client dispose d'un espace personnel sécurisé lui permettant de :

- S'inscrire et se connecter à la plateforme,
- Créer et gérer ses projets de correction (avec dépôt de fichiers, choix du pack…),
- Suivre l'avancement de ses projets et télécharger les fichiers corrigés,
- Communiquer par messagerie avec le support ou les correcteurs,
- Payer ses commandes en ligne et télécharger ses factures,
- Gérer ses moyens de paiement enregistrés,
- Consulter la FAQ, les pages informatives, et l'historique de ses actions,
- Contacter le support ou soumettre une demande d'aide,
- Gérer ses notifications (email, push, SMS),
- Supprimer ou désactiver complètement son compte (respect RGPD).

Côté administrateur

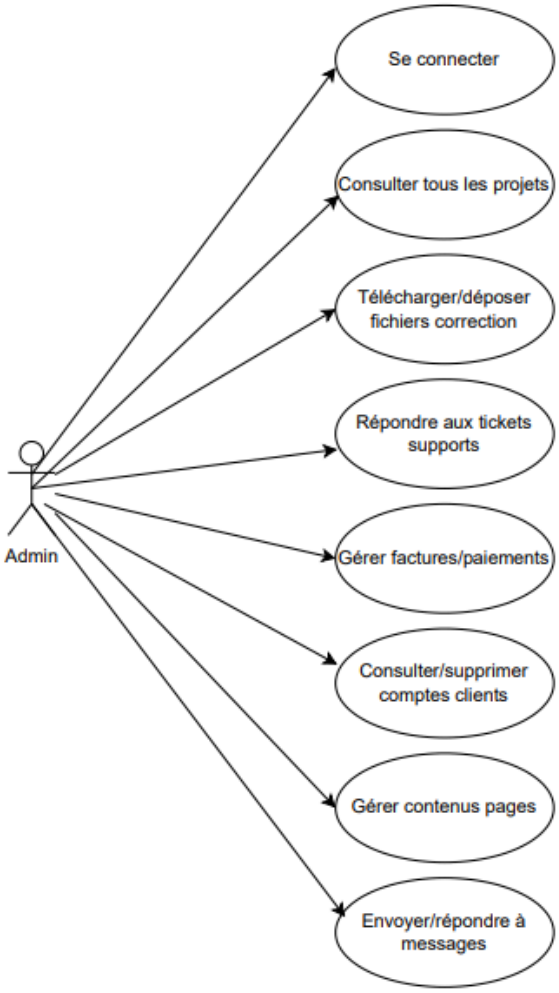


Schéma de cas d'utilisation — Staka-Livres

L’administrateur accède à un espace dédié lui permettant de :

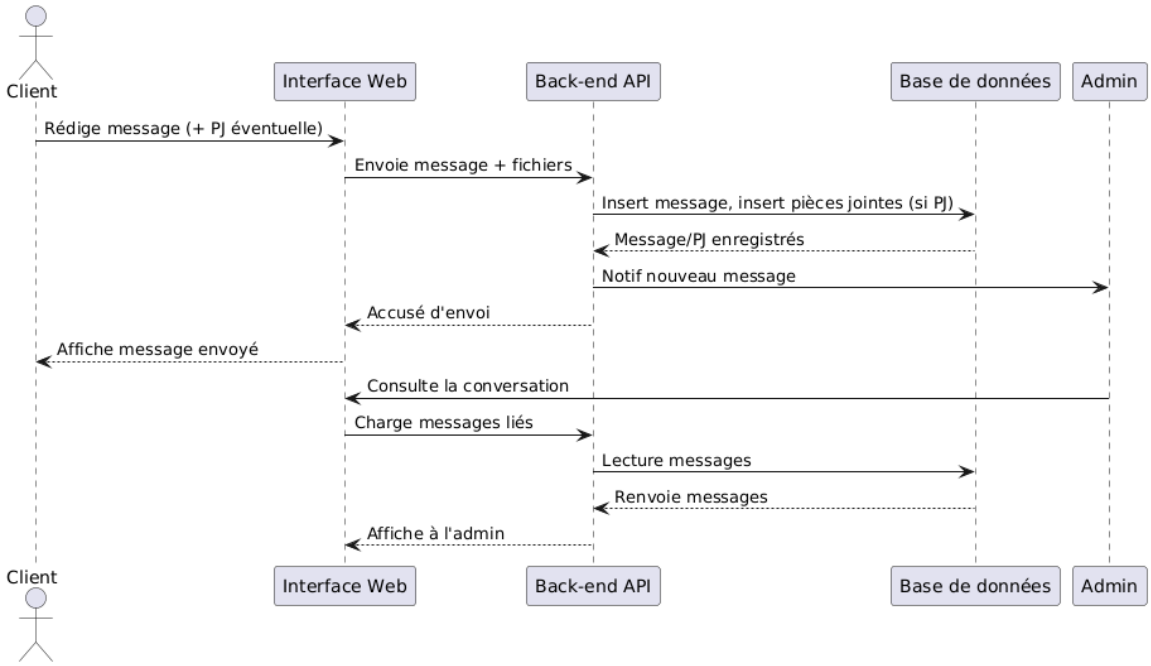
- Se connecter de façon sécurisée,
- Consulter l’ensemble des projets soumis par les clients,
- Gérer et déposer les fichiers corrigés sur chaque projet,
- Répondre à la messagerie et aux tickets de support client,
- Gérer les contenus éditoriaux du site (pages informatives, FAQ…),
- Consulter ou supprimer les comptes clients si besoin (RGPD),
- Gérer les factures, les paiements et le suivi de la facturation,
- Recevoir des notifications à chaque événement important.

En résumé

Ces schémas illustrent toutes les actions principales possibles sur la plateforme, pour chaque type d’utilisateur. Il met en avant la sécurité, la traçabilité, et la complétude fonctionnelle attendue du projet.

Les diagrammes de séquences

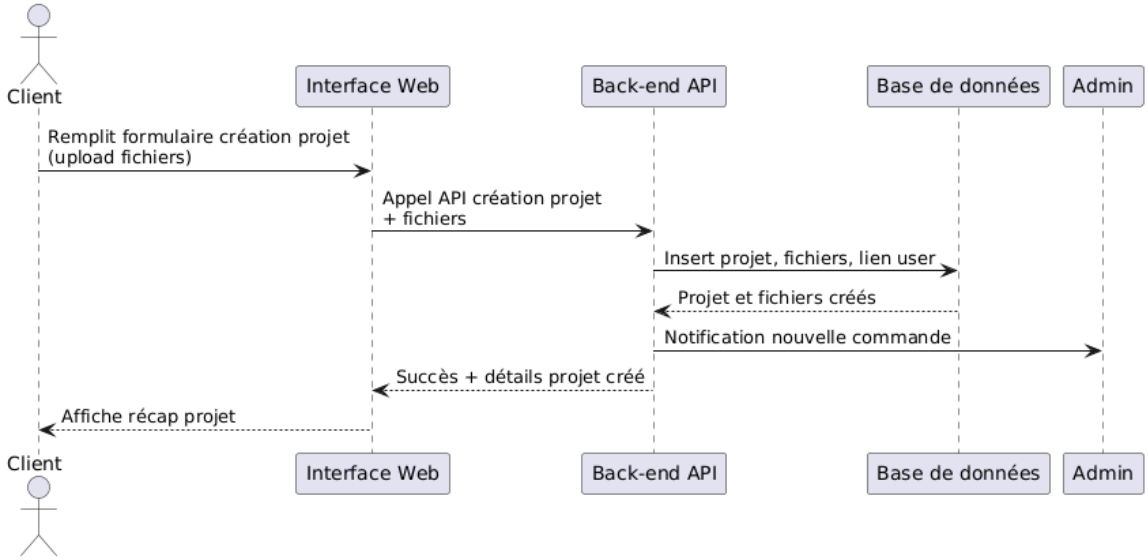
Voici quelques scénarios de séquence essentiels à produire pour ce projet, suivie d'une description textuelle étape par étape pour chaque séquence :



Envoi et réception d'un message (messagerie instantanée)

Participants : Client → Interface Web → Back-end → BDD → (Admin/correcteur → Notif)

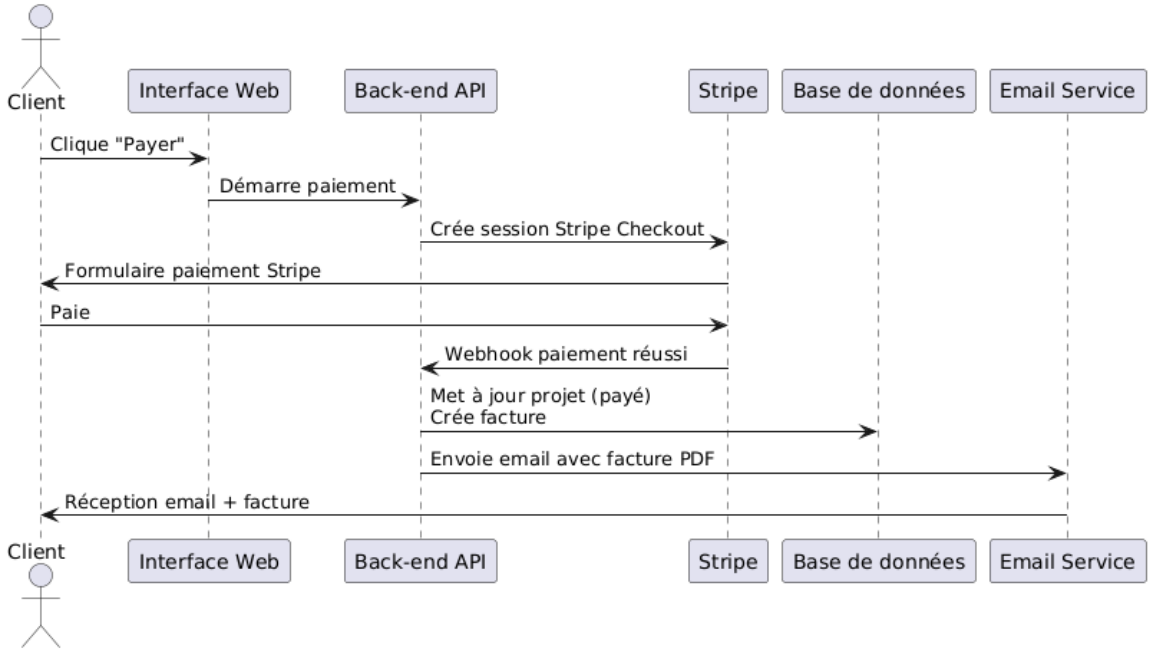
1. Le client écrit un message (option : joint une pièce jointe).
2. Le front envoie le message au back.
3. Le back crée le message (et le lien vers la pièce jointe, s'il y en a une).
4. Le message est stocké en BDD et notifie le destinataire (admin/correcteur).
5. L'autre partie (admin) consulte/répond dans l'interface admin.
6. Même flux inverse pour la réponse.



Création d'un projet client

Participants : Client → Interface Web → Back-end → BDD → Admin (notif)

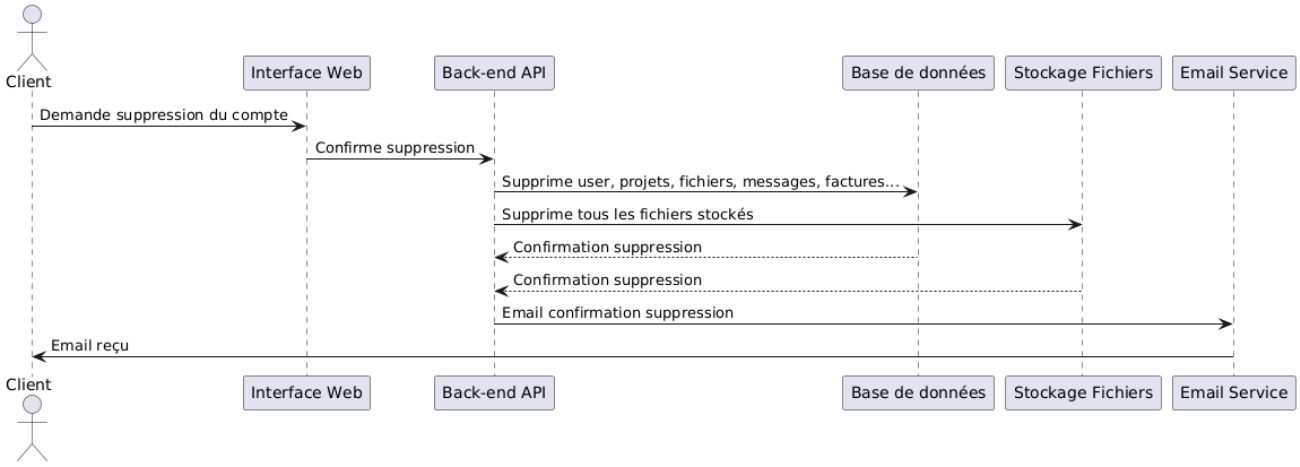
1. Le client clique sur "Nouveau projet", saisit les infos, upload les fichiers, valide.
2. Le front envoie la demande de création au back.
3. Le back valide les données, crée le projet en BDD.
4. Les fichiers sont uploadés et liés au projet.
5. Un email/notif est envoyé à l'admin.
6. Retour au client avec le récap de la création du projet.



Paiement d'un projet et génération de facture

Participants : Client → Front → Back → Stripe → BDD → Email

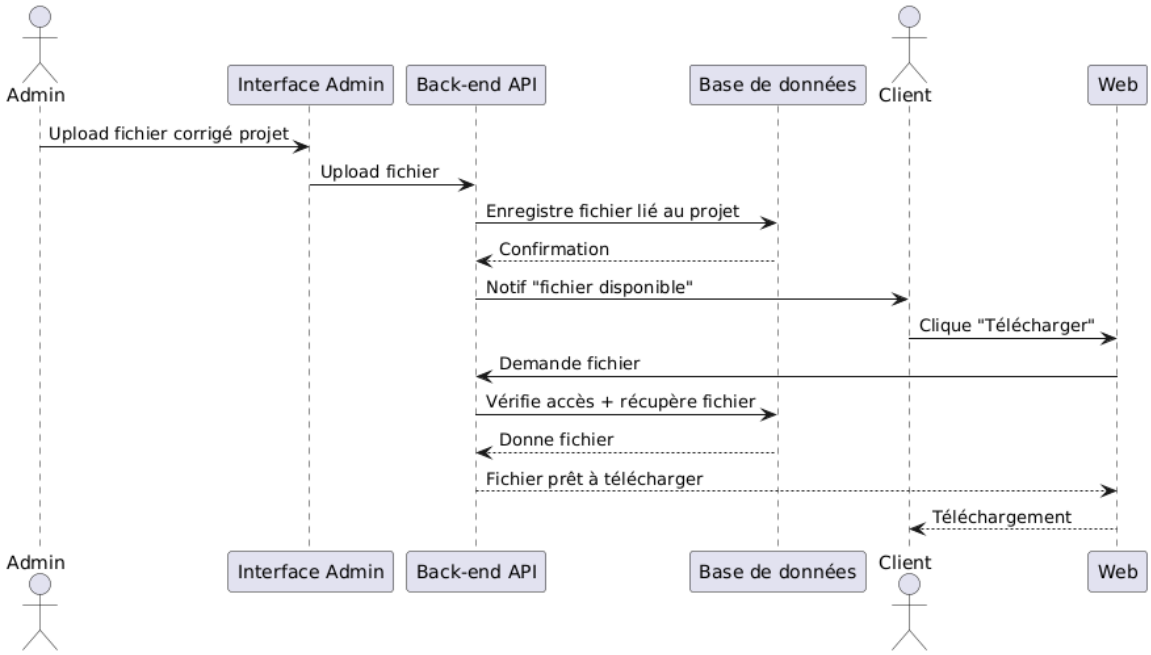
1. Le client clique sur "Payer" sur un projet.
2. Le back initie Stripe Checkout.
3. Le client paie sur Stripe.
4. Stripe notifie le back-end (webhook).
5. Le back valide le paiement, met à jour le projet + crée la facture PDF.
6. Un email est envoyé au client avec la facture.



Suppression complète du compte client (RGPD)

Participants : Client → Front → Back → BDD/Stockage

1. Le client clique sur “Supprimer mon compte”.
2. Le front demande une confirmation.
3. Le back supprime tous les fichiers, messages, projets, factures, données persos liées à ce user.
4. Le client reçoit une confirmation par email.



Téléchargement d’un fichier corrigé

Participants : Client → Front → Back → BDD → Stockage

1. L’admin/correcteur upload un fichier corrigé sur le projet (dans l’admin).
2. Le client reçoit une notif “fichier disponible”.
3. Le client clique sur “Télécharger”.
4. Le back contrôle les droits et donne le fichier.

Le script SQL de la création de la base de données

```
-- Table: User
CREATE TABLE User(
id INT AUTO_INCREMENT PRIMARY KEY,
email VARCHAR(255) NOT NULL UNIQUE,
password VARCHAR(255) NOT NULL,
first_name VARCHAR(100) NOT NULL,
last_name VARCHAR(100) NOT NULL,
phone VARCHAR(20),
role VARCHAR(20) NOT NULL,
status VARCHAR(20) NOT NULL,
created_at DATETIME NOT NULL,
is_public_profile BOOL DEFAULT 0,
analytics_consent BOOL DEFAULT 0,
notif_email BOOL DEFAULT 1,
notif_push BOOL DEFAULT 0,
notif_sms BOOL DEFAULT 0
);

-- Table: Project
CREATE TABLE Project(
id INT AUTO_INCREMENT PRIMARY KEY,
title VARCHAR(255) NOT NULL,
manuscript_type VARCHAR(50) NOT NULL,
pages_count INT NOT NULL,
description TEXT,
status VARCHAR(30) NOT NULL,
created_at DATETIME NOT NULL,
start_date DATE,
delivery_due_date DATE,
delivery_done_date DATE,
pack VARCHAR(30) NOT NULL,
proofreader_name VARCHAR(100),
rating INT,
is_urgent BOOL DEFAULT 0,
user_id INT NOT NULL,
FOREIGN KEY (user_id) REFERENCES User(id)
);

-- Table: File
CREATE TABLE File(
id INT AUTO_INCREMENT PRIMARY KEY,
file_name VARCHAR(255) NOT NULL,
file_type VARCHAR(30) NOT NULL,
file_size INT NOT NULL,
file_url VARCHAR(512) NOT NULL,
uploaded_at DATETIME NOT NULL,
project_id INT NOT NULL,
user_id INT NOT NULL,
FOREIGN KEY (project_id) REFERENCES Project(id),
FOREIGN KEY (user_id) REFERENCES User(id),
INDEX (project_id),
INDEX (user_id)
);
```

```
-- Table: Invoice
CREATE TABLE Invoice(
id INT AUTO_INCREMENT PRIMARY KEY,
invoice_number VARCHAR(30) NOT NULL,
project_id INT NOT NULL,
user_id INT NOT NULL,
amount DECIMAL(8,2) NOT NULL,
status VARCHAR(20) NOT NULL,
issued_at DATETIME NOT NULL,
paid_at DATETIME,
pdf_url VARCHAR(512),
FOREIGN KEY (project_id) REFERENCES Project(id),
FOREIGN KEY (user_id) REFERENCES User(id),
UNIQUE (invoice_number)
);

-- Table: PaymentMethod
CREATE TABLE PaymentMethod(
id INT AUTO_INCREMENT PRIMARY KEY,
user_id INT NOT NULL,
stripe_pm_id VARCHAR(100) NOT NULL,
brand VARCHAR(30) NOT NULL,
last4 VARCHAR(4) NOT NULL,
exp_month INT NOT NULL,
exp_year INT NOT NULL,
is_default BOOL DEFAULT 0,
added_at DATETIME NOT NULL,
status VARCHAR(20),
FOREIGN KEY (user_id) REFERENCES User(id),
UNIQUE (stripe_pm_id),
INDEX (user_id)
);

-- Table: Message
CREATE TABLE Message(
id INT AUTO_INCREMENT PRIMARY KEY,
content TEXT NOT NULL,
sent_at DATETIME NOT NULL,
is_read BOOL DEFAULT 0,
project_id INT, -- NULL si message global/support
user_id INT NOT NULL,
support_request_id INT, -- NULL si message classique
FOREIGN KEY (project_id) REFERENCES Project(id),
FOREIGN KEY (user_id) REFERENCES User(id),
FOREIGN KEY (support_request_id) REFERENCES SupportRequest(id),
INDEX (project_id),
INDEX (user_id),
INDEX (support_request_id)
);

-- Table: Message_attachment (liaison N:N message <-> fichier)
CREATE TABLE Message_attachment(
id INT AUTO_INCREMENT PRIMARY KEY,
message_id INT NOT NULL,
file_id INT NOT NULL,
FOREIGN KEY (message_id) REFERENCES Message(id) ON DELETE CASCADE,
FOREIGN KEY (file_id) REFERENCES File(id) ON DELETE CASCADE,
INDEX (message_id),
```

```
INDEX (file_id)
);

-- Table: SupportRequest
CREATE TABLE SupportRequest(
id INT AUTO_INCREMENT PRIMARY KEY,
user_id INT NOT NULL,
subject VARCHAR(250) NOT NULL,
status VARCHAR(20) NOT NULL,
created_at DATETIME NOT NULL,
closed_at DATETIME,
type VARCHAR(30),
FOREIGN KEY (user_id) REFERENCES User(id),
INDEX (user_id)
);

-- Table: Notification
CREATE TABLE Notification(
id INT AUTO_INCREMENT PRIMARY KEY,
user_id INT NOT NULL,
type VARCHAR(30) NOT NULL,
content TEXT NOT NULL,
channel VARCHAR(20) NOT NULL,
sent_at DATETIME NOT NULL,
is_read BOOL DEFAULT 0,
related_id INT,
related_type VARCHAR(30),
FOREIGN KEY (user_id) REFERENCES User(id),
INDEX (user_id),
INDEX (related_id)
);

-- Table: Page
CREATE TABLE Page(
id INT AUTO_INCREMENT PRIMARY KEY,
slug VARCHAR(100) NOT NULL,
title VARCHAR(255) NOT NULL,
content TEXT NOT NULL,
updated_at DATETIME,
is_active BOOL DEFAULT 1,
display_order INT
);
```

Architecture technique

Le projet repose sur une architecture moderne, robuste et sécurisée :

- **Backend** : Node.js + Express.js, base MySQL via ORM (Prisma)
- **Websocket temps réel** : gestion de la messagerie instantanée et des notifications live via socket .io, pour une expérience utilisateur fluide et réactive (chat et alertes en direct)
- **Stockage sécurisé des fichiers** : local (serveur sécurisé) ou cloud (AWS S3 ou équivalent), avec contrôle d'accès
- **Paiement en ligne** : intégration Stripe (mode paiement unique)
- **Notifications multicanal**: envoi automatique d'e-mails, SMS (via service tiers) et notifications push en temps réel grâce au websocket
- **API RESTful structurée et sécurisée** : authentification JWT/session, gestion fine des rôles (client/admin), validation centralisée des entrées
- **Organisation du code** : découpage par modules : modèles, contrôleurs, routes, middlewares, services externes, sockets, etc.
- **Déploiement** : serveur dédié (ou cloud managé), HTTPS obligatoire (certificat SSL), backups automatiques, procédures documentées de restauration

Cette architecture garantit :

- **Une sécurité élevée** : gestion des accès, chiffrement des données sensibles, conformité RGPD, bonnes pratiques Node.js, séparation stricte des espaces client/admin, stockage protégé
- **Une haute évolutivité** : ORM pour la base de données (migrations, versionning), code modulaire, support natif de nouvelles fonctionnalités ou besoins métiers
- **Une expérience utilisateur moderne** : messagerie et notifications réellement instantanées (websockets), interfaces rapides et réactives
- **Une maintenance facilitée** : code testable, logique claire, logs centralisés, documentation technique complète
- **Une intégration rapide des évolutions** : ajout de modules, d'API ou de services sans remise en cause de l'existant

En résumé, cette architecture allie sécurité, performance, évolutivité et expérience utilisateur haut de gamme grâce à l'utilisation combinée d'une API REST sécurisée et des websockets pour le temps réel.

Table des routes API principales

Ressource	Méthode	URL	Auth requise	Description	Réponse attendue
User	POST	/api/register	Non	Créer un nouveau compte utilisateur	201, user (token)
	POST	/api/login	Non	Connexion, retour token/session	200, token/user
	GET	/api/user/me	Oui (Client/Admin)	Infos profil connecté	200, user
	PUT	/api/user/me	Oui	Modifier ses infos	200, user
	DELETE	/api/user/me	Oui	Suppression complète compte (RGPD)	204
	GET	/api/user/export	Oui	Export données personnelles	200, fichier/JSON
Project	POST	/api/projects	Oui	Créer un projet	201, project
	GET	/api/projects	Oui	Lister mes projets	200, [projects]
	GET	/api/projects/:id	Oui	Détail d'un projet	200, project
	PUT	/api/projects/:id	Oui	Modifier projet (si autorisé)	200, project
	DELETE	/api/projects/:id	Oui	Supprimer projet (si autorisé)	204
File	POST	/api/projects/:id/files	Oui	Upload de fichier sur projet	201, file
	GET	/api/projects/:id/files	Oui	Liste des fichiers d'un projet	200, [files]
	GET	/api/files/:id/download	Oui	Télécharger un fichier	200, file (download)
Message	POST	/api/projects/:id/messages	Oui	Envoyer message projet	201, message
	POST	/api/support/:id/messages	Oui	Envoyer message support	201, message
	GET	/api/projects/:id/messages	Oui	Liste messages projet	200, [messages]
	GET	/api/support/:id/messages	Oui	Liste messages support	200, [messages]
Support	POST	/api/support	Oui	Créer un ticket support	201, support_request
	GET	/api/support	Oui	Lister mes tickets support	200, [support_requests]
	GET	/api/support/:id	Oui	Détail d'un ticket support	200, support_request
Invoice	GET	/api/invoices	Oui	Liste factures utilisateur	200, [invoices]
	GET	/api/invoices/:id	Oui	Télécharger facture PDF	200, file
Paiement	POST	/api/projects/:id/pay	Oui	Initier paiement Stripe	200, url/stripe_session

PaymentMethod	GET	/api/payment-methods	Oui	Liste moyens de paiement	200, [payment_methods]
	POST	/api/payment-methods	Oui	Ajouter une carte	201, payment_method
	DELETE	/api/payment-methods/:id	Oui	Supprimer une carte	204
Notification	GET	/api/notifications	Oui	Lister notifications	200, [notifications]
	PUT	/api/notifications/:id/read	Oui	Marquer notif comme lue	200, notification
Page/FAQ	GET	/api/pages/:slug	Non	Charger une page info/FAQ	200, page
Admin	GET	/api/admin/projects	Oui (Admin)	Lister tous les projets clients	200, [projects]
	GET	/api/admin/users	Oui (Admin)	Lister tous les clients	200, [users]
	PUT	/api/admin/users/:id	Oui (Admin)	Modifier/Supprimer compte client	200, user
	GET	/api/admin/invoices	Oui (Admin)	Toutes les factures	200, [invoices]
	PUT	/api/admin/pages/:slug	Oui (Admin)	Modifier page/FAQ/tarifs	200, page

Remarque :

- Auth requise : Oui = token JWT ou session nécessaire
- Les routes /admin/ sont accessibles uniquement pour les admins.
- À adapter selon l'implémentation finale (ex : gestion des fichiers de correction, ou APIs supplémentaires).

Matrice des droits et rôles

Fonctionnalité / Ressource	Client	Admin
S'inscrire / Se connecter	✓	✓
Créer un projet	✓	✗
Gérer ses projets	✓	✗
Consulter tous les projets	✗	✓
Télécharger ses fichiers	✓	✗
Télécharger/déposer corrections	✗	✓
Envoyer/recevoir messages (chat)	✓	✓
Recevoir notifications	✓	✓
Gérer contenus (pages/FAQ)	✗	✓
Payer un projet	✓	✗
Gérer factures et paiements	✓	✓
Consulter toutes les factures	✗	✓
Consulter/supprimer comptes clients	✗	✓
Gérer moyens de paiement	✓	✗
Consulter FAQ/pages	✓	✓
Contacter le support	✓	✓
Répondre au support/tickets	✗	✓
Supprimer son propre compte	✓	✓
Supprimer n'importe quel compte	✗	✓
Noter un projet	✓	✗
Modifier les tarifs/packs	✗	✓
Gérer statuts projets	✗	✓
Voir/modifier préférences notif	✓	✓

Explication de la matrice des droits et rôles

Cette matrice synthétise l'ensemble des fonctionnalités du site et indique clairement qui (client ou administrateur) peut accéder à quelle ressource ou action sur la plateforme Staka-Livres.

- **Le rôle "Client"** correspond à tout utilisateur inscrit classique. Il dispose de l'accès complet à la gestion de ses projets, fichiers, paiements, messagerie, support, notifications, et à toutes les actions liées à son propre espace personnel (création/suppression de compte, gestion des moyens de paiement, téléchargement, notation de projets, etc.).

- **Le rôle "Admin"** dispose d'un accès centralisé à toutes les données : consultation et gestion de l'ensemble des projets, fichiers, comptes utilisateurs, contenus éditoriaux (pages, tarifs), factures et support. L'admin gère la supervision globale, la sécurité et la conformité RGPD du service.

Cette séparation stricte des droits :

- Garantit la sécurité et la confidentialité des données : aucun client ne peut accéder à une donnée ou une action réservée à l'administration.
- Permet à chaque rôle d'avoir une interface et des fonctionnalités adaptées à ses besoins réels, sans "boutons" ou pages inutiles.
- Facilite la maintenance et le respect des exigences légales (accès aux données, suppression, gestion des contenus).

Ce tableau sert de référence pour toute la phase de développement et d'audit sécurité, afin de vérifier que chaque fonctionnalité est bien protégée côté API et interface utilisateur.

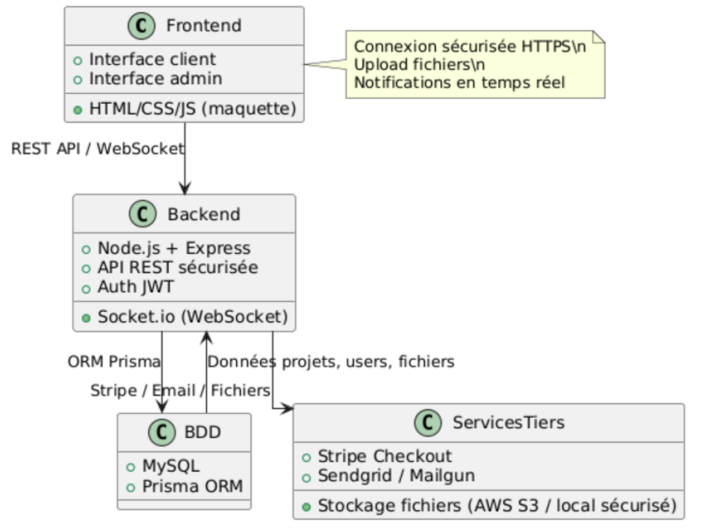


Schéma d'architecture technique – interactions entre front, back, base de données, et services externes (paiement, email, stockage)

Scénarios d'usage utilisateur

Scénario 1 : Parcours classique de correction d'un manuscrit

1. **Inscription / Connexion**
    - Je me rends sur la page d'accueil et clique sur "Créer un compte".
    - Je renseigne mon prénom, nom, email, téléphone et un mot de passe.
    - Je reçois un email de confirmation et accède à mon espace personnel.
    - Si j'ai déjà un compte, je me connecte directement.

2. **Création d'un projet**
    - Depuis mon tableau de bord, je clique sur "Nouveau projet".
    - Je remplis le formulaire (titre, type de manuscrit, nombre de pages, description…).
    - Je choisis un pack de correction et je vois le tarif correspondant.
    - J'upload un ou plusieurs fichiers (manuscrit, annexes…).
    - Je valide la création du projet.
    - Je reçois une confirmation et mon projet apparaît dans la liste.

3. **Paiement**
    - Je clique sur "Payer" pour régler ma commande.
    - Je suis redirigé vers la page de paiement Stripe.
    - J'entre mes coordonnées bancaires et valide le paiement.
    - Je reçois une facture PDF par email et la retrouve dans mon espace.

4. **Suivi du projet**
    - Je visualise l'état d'avancement de mon projet (barre de progression, dates, statut).
    - Je peux envoyer des messages à l'équipe/correcteur via la messagerie interne.
    - Je reçois des notifications (email, push) à chaque changement de statut ou nouveau message.

5. **Téléchargement du fichier corrigé**
    - Quand le projet est terminé, je reçois une notification "fichier disponible".
    - Je télécharge le(s) fichier(s) corrigé(s) depuis mon espace projet.

6. **Notation et clôture**
    - Je peux attribuer une note au projet réalisé et laisser un commentaire si je le souhaite.
    - Mon projet passe dans l'historique, accessible à tout moment.

Scénario 2 : Gestion de mes fichiers et données personnelles (RGPD)

1. **Accès à mes fichiers**
    - Je peux retrouver, prévisualiser ou télécharger tous les fichiers liés à mes projets dans mon espace.

2. **Gestion de mes informations**
    - Je mets à jour mes informations personnelles et mes préférences de notifications dans mon profil.

3. **Export ou suppression de mes données**
    - Je demande l'export de toutes mes données personnelles en un clic.
    - Je peux supprimer mon compte (et toutes mes données associées) définitivement si je le souhaite.
    - Je reçois une confirmation par email.

Scénario 3 : Utilisation du support et de la FAQ

1. **Recherche d'aide**
    - Je consulte la FAQ ou les pages d'aide si j'ai une question sur le fonctionnement du site.

2. **Demande de support**
    - Je contacte l'équipe via le chat ou un formulaire dédié.
    - Je joins une pièce jointe si besoin (ex : capture d'écran d'un bug).
    - Je reçois une notification dès qu'une réponse m'est apportée.

Scénario 4 : Gestion de mes moyens de paiement et historique

1. **Ajout/modification d'une carte**
    - J'ajoute une carte bancaire dans la section "Moyens de paiement".
    - Je peux la supprimer ou la remplacer à tout moment.

2. **Consultation de mon historique**
    - Je retrouve le récap annuel : nombre de projets complétés, pages corrigées, total dépensé, économies réalisées.
    - Je consulte et télécharge l'ensemble de mes factures passées.

Remarque

Ces scénarios d'usage couvrent tout le parcours utilisateur, de l'inscription à la gestion complète des projets, fichiers, paiements, support et données personnelles.
Ils servent de guide pour s'assurer qu'aucune fonctionnalité-clé n'a été oubliée dans la conception et la réalisation du projet.

Logs et supervision

1. Typologie des logs générés

- **Logs d'accès**
  - Toutes les connexions à l'espace client et admin sont tracées : date, IP, user agent, succès ou échec.
  - Permet de détecter les accès non autorisés ou les tentatives suspectes.

- **Logs d'erreur**
  - Toute erreur applicative (bug, exception, problème base de données, échec d'envoi d'email, dépassement de quota fichier, etc.) est enregistrée avec le contexte : timestamp, user concerné, endpoint/API, stack trace.
  - Ces logs servent à la correction rapide des bugs et à la supervision de la stabilité du service.

- **Logs d'activité critique**
  - Actions sensibles : suppression/export de données, création/suppression de projets, paiements, modifications du profil ou des permissions.
  - Utiles pour l'audit sécurité et la conformité RGPD.

- **Logs de mails et notifications**
  - Chaque envoi d'email ou de notification (Stripe, création projet, fichier dispo, etc.) génère un log : type, destinataire, statut (succès/échec), éventuelle erreur renvoyée par le service tiers.

- **Logs d'intégration externe**
  - Erreurs ou délais de Stripe, Sendgrid/Mailgun, API tiers (paiement, mail, SMS…) sont logués pour analyse.

2. Où sont stockés les logs ?

- **Fichiers de logs sur le serveur**
  - Accès, erreurs, activités : fichiers structurés (logs/access.log, logs/error.log, etc.)
  - Rotation automatique des fichiers (logs archivés après X jours pour éviter la saturation du disque).

3. Accès et gestion des logs

- **Accès réservé**
  - Seuls l'administrateur technique ou le prestataire de maintenance peuvent accéder aux fichiers de logs serveur.
  - Les logs sensibles ne sont jamais transmis à des tiers ou stockés dans des lieux non sécurisés.

En résumé : la solution génère et supervise des logs complets pour garantir la traçabilité, la sécurité, la maintenance rapide et la conformité du service, tout en respectant la confidentialité des utilisateurs.

Checklist RGPD

1. Consentement et transparence

- **Consentement explicite**
  Lors de la création d'un compte, l'utilisateur doit donner son accord explicite pour la collecte et l'utilisation de ses données personnelles.
  Une case à cocher spécifique est intégrée au formulaire d'inscription, avec lien vers la politique de confidentialité.
- **Droit à l'information**
  L'utilisateur peut à tout moment consulter la politique de confidentialité qui détaille les finalités, les destinataires et la durée de conservation de ses données.

2. Accès, portabilité et export des données

- **Droit d'accès**
  Chaque utilisateur peut accéder à toutes les données le concernant (profil, projets, fichiers, messages, paiements…) directement depuis son espace personnel.
- **Portabilité/export**
  Un bouton "Exporter mes données" permet à l'utilisateur de télécharger l'ensemble de ses informations sous forme d'un fichier lisible (JSON ou archive ZIP), incluant :
  - Informations de profil
  - Historique des projets et fichiers
  - Messages, factures, tickets de support
  - Préférences de notification

3. Rectification et mise à jour

- **Droit de modification**
  L'utilisateur peut mettre à jour ses informations personnelles et ses préférences (notifications, consentement analytics, etc.) à tout moment via son profil.

4. Suppression et droit à l'oubli

- **Suppression totale en un clic**
  L'utilisateur peut supprimer l'ensemble de son compte et toutes les données associées (projets, fichiers, messages, factures, moyens de paiement, etc.) en un seul clic via l'interface dédiée.
- **Suppression irréversible**
  La suppression est immédiate et irréversible : toutes les données sont supprimées de la base principale ET marquées pour suppression dans la prochaine sauvegarde.

Les fichiers liés sont effacés des stockages actifs et supprimés lors du prochain cycle de rotation des backups (max. 30 jours).

- **Notification de suppression**
  Un email de confirmation est envoyé à l'utilisateur après suppression.

5. Limitation et durée de conservation

- **Durée de conservation**
  Les données sont conservées uniquement le temps nécessaire à la réalisation de la prestation et au respect des obligations légales (comptabilité, facturation).
    - Données inactives (aucun accès depuis 36 mois) sont automatiquement anonymisées ou supprimées.

6. Sécurité et confidentialité

- **Protection des données**
  Toutes les données sont stockées sur des serveurs sécurisés, protégés par chiffrement, accès restreints et sauvegardes régulières.
- **Accès limité**
  Seuls les administrateurs autorisés et l'utilisateur concerné peuvent accéder à ses données personnelles.

7. Traçabilité et journalisation

- **Historique des actions sensibles**
  Les actions critiques (suppression, export, paiement, modification de profil) sont journalisées de manière anonymisée pour audit et sécurité.

8. Réponse aux demandes RGPD

- **Délai de réponse**
  Toute demande RGPD (export, modification, suppression, information) est traitée dans un délai maximal de 30 jours.

Cette stratégie RGPD assure à tous les utilisateurs une maîtrise totale de leurs données, un respect des obligations légales, et une totale transparence sur le traitement de leurs informations personnelles.

Sécurité

1. Sécurité des accès et des données

- **Chiffrement des mots de passe**
  Tous les mots de passe utilisateurs sont hashés et salés avec un algorithme de pointe (bcrypt) avant stockage en base de données. Aucune information sensible n'est jamais conservée en clair.
- **Connexion sécurisée (HTTPS/SSL)**
  Toutes les communications entre le client, l'admin et le serveur transitent exclusivement via des connexions sécurisées (HTTPS), protégeant ainsi les échanges contre les interceptions et attaques de type "man-in-the-middle".
- **Authentification et gestion des droits**
  Accès à l'espace client ou admin strictement protégé par authentification ; gestion des rôles et permissions contrôlée côté serveur et base de données.
- **Séparation stricte des espaces**
  Un client n'a jamais accès à des données ou fonctionnalités d'un autre client ou de l'admin. Les API et la BDD appliquent cette séparation à tous les niveaux.
- **Sécurité des fichiers**
  Les fichiers uploadés sont stockés dans des dossiers protégés et inaccessibles directement depuis Internet. Seuls les utilisateurs autorisés peuvent les télécharger via des liens temporaires ou sécurisés.
- **Protection contre les attaques courantes**
  Utilisation de middlewares de sécurité Node.js (Helmet, CORS, rate limiting, etc.) pour se prémunir des injections, XSS, CSRF et brute-force.
- **Notifications sensibles**
  Les notifications contenant des données sensibles ne sont jamais envoyées par email non sécurisé, et aucun lien direct vers un fichier ou une information critique n'est transmis dans les emails.

2. Sauvegardes et restauration

- **Sauvegarde quotidienne automatisée**
  - **Base de données** : une sauvegarde complète (dump SQL) de la base de données est générée chaque nuit et stockée sur un espace sécurisé distinct du serveur principal.
  - **Fichiers utilisateurs** : tous les fichiers uploadés (manuscrits, corrections, pièces jointes) sont sauvegardés quotidiennement sur un espace cloud sécurisé ou un stockage externe chiffré.
- **Durée de conservation des sauvegardes**
  - Les sauvegardes sont conservées pendant **30 jours**, permettant la restauration de toutes les données à n'importe quel moment sur cette période.
- **Procédure de restauration**
  - En cas d'incident (perte de données, corruption, attaque), la restauration des données est possible en moins de 2 heures ouvrées.

- Une procédure documentée permet de restaurer à la fois la base de données et les fichiers utilisateurs, avec vérification d'intégrité.

3. Disponibilité et conformité

- **Taux de disponibilité cible**
  - Le service vise un taux de disponibilité d'au moins 99 % sur l'année (hors plages de maintenance programmée).
- **Conformité RGPD**
  - Les sauvegardes intègrent les suppressions RGPD : si un utilisateur demande la suppression de ses données, elles seront également supprimées des sauvegardes lors de la rétention maximale (30 jours).
  - Droit d'export, d'accès et de suppression totale respecté à tout moment.

En résumé : la sécurité des données, leur confidentialité et leur pérennité sont assurées par une politique stricte d'accès, de sauvegarde et de restauration, adaptée aux exigences du projet et aux normes en vigueur.

Tests et qualité

1. Approche générale

La qualité logicielle et la robustesse du projet sont garanties par :

- L'adoption de bonnes pratiques de développement : code revu, normé (lint, prettier), structuré en modules testables.
- La mise en place de **tests automatisés** à différents niveaux : unitaires, d'intégration, et manuels sur les parcours critiques.

2. Tests unitaires

- **Définition** : Ils valident chaque composant "isolé" : fonctions métiers, contrôleurs, middlewares, helpers…
- **Outils** : Jest ou équivalent pour Node.js.
- **Exemples couverts** :
  - Fonctions de validation (emails, fichiers, calculs)
  - Hash et vérification de mot de passe
  - Génération de notifications
  - Vérification des droits d'accès

3. Tests d'intégration

- **Définition** : Ils vérifient le bon fonctionnement des routes API "de bout en bout", incluant le dialogue avec la base, la gestion des rôles, l'authentification, la persistance, etc.
- **Outils** : Supertest (Express), Postman (collections automatiques), tests manuels complémentaires.
- **Parcours couverts** :
    - Authentification (login, register, protection JWT/session)
    - Création/modification/suppression d'un projet (CRUD complet)
    - Upload et téléchargement de fichiers (vérification taille, type, accès)
    - Envoi et réception de messages/messagerie instantanée
    - Paiement Stripe (simulation, webhook, génération de facture)
    - Gestion et suppression RGPD (export/suppression de compte)
    - Support/ticket (création, échanges, fermeture)
    - Notifications (création, marquage comme lue, envoi email)

4. Tests manuels

- **Navigation sur les interfaces fournies** : création de compte, gestion de projet, paiement réel (sandbox Stripe), upload fichiers, suppression de compte.
- **Scénarios limites** (fichiers trop lourds, données incomplètes, tentative d'accès non autorisé, fausse carte bancaire…)
- **Accessibilité** : tests sur mobile, tablettes, navigation clavier, contraste.

5. Revue de code et linting

- **Lint automatique** (ESLint, Prettier)
- **Convention de nommage et structure de dossiers respectées**
- **Pull requests et revue régulière, même en solo (auto-audit/checklist)**

6. Procédure de correction

- Toute anomalie découverte en test est notée, priorisée et corrigée avant livraison.
- Une phase de recette avec le client peut être prévue pour valider les parcours réels.

En synthèse

L'ensemble des routes critiques (authentification, paiement, upload/suppression de fichiers, suppression/export de données) est systématiquement couvert par des tests automatisés et manuels, garantissant la fiabilité, la sécurité et la conformité du service avant chaque mise en production.

Environnement d'intégration et de déploiement

1. Organisation des environnements

- **Environnement de développement**
  Chaque fonctionnalité est développée localement ou sur une branche dédiée du dépôt Git.
  Un fichier .env spécifique au développement contient les variables sensibles (DB, Stripe, mail, etc.) non versionnées.
- **Environnement de pré-production**
  Permet de valider les fonctionnalités sur un serveur intermédiaire (identique à la prod) avant la mise en ligne finale.
  Peut servir aux tests du client ou à la recette "grandeur nature".
- **Environnement de production**
  Application déployée sur un serveur sécurisé, domaine officiel (livres.staka.fr), base de données isolée, backup et supervision active.

2. Gestion des variables d'environnement et secrets

- **Fichier** .env
  Contient :
  - Chaînes de connexion MySQL
  - Clés Stripe (test/production)
  - Secrets JWT
  - Clés API pour emails (Sendgrid, Mailgun, etc.)
  - URLs de frontend (CORS)
- **Bonne pratique** : Jamais versionner le .env dans Git ; utiliser un exemple .env.example.

3. Commandes de migration et gestion de la base

- **ORM (Prisma)**
  - npx prisma migrate deploy pour appliquer les évolutions du schéma.
  - Migrations lancées **automatiquement** ou manuellement lors de chaque déploiement.
- **Sauvegardes** : Dump automatique avant chaque migration majeure ou déploiement.

4. Procédure de déploiement (Node.js)

1. **Build** le projet
2. **Install** des dépendances (npm install --production)
3. **Mise à jour du** .env **de prod** (variables adaptées)
4. **Migration de la base** (cf. étape précédente)
5. **Redémarrage du serveur** via PM2, systemd, ou process manager équivalent
6. **Vérification du service** (logs, tests "smoke" sur les endpoints)

5. Hébergement / serveur

- **Plan d'hébergement conseillé**
  - VPS sécurisé (ex : OVH, Scaleway, Infomaniak…), ou PaaS moderne (Render, Railway, DigitalOcean…)
  - **Configuration serveur** :
    - OS Linux (Ubuntu, Debian)
    - Node.js (version LTS), Nginx (reverse proxy), Certificat SSL Let's Encrypt
    - Base MySQL isolée, backup automatique quotidien
  - **Surveillance** : Monit, logs centralisés, alertes en cas d'erreur critique

6. Mise en ligne initiale

- **Réservation/pointage DNS** vers le serveur cible
- **Installation du SSL** (HTTPS forcé)
- **Déploiement de l'application**
- **Tests post-deploiement** sur les principales fonctionnalités (auth, projet, paiement, upload, suppression)
- **Documentation complète** des étapes pour réinstaller si besoin

7. Gestion des mises à jour

- **Rolling update** (mise à jour sans coupure)
- **Backup systématique** avant chaque upgrade majeur

En résumé : le projet est structuré pour un passage du développement à la production sécurisé, traçable et rapide, avec respect des bonnes pratiques DevOps et une documentation claire de chaque étape pour faciliter la maintenance ou la reprise future.