

EpiTrello — Cahier des charges

Stack : React + TypeScript | Next.js (App Router) | shadcn/ui | Supabase

Périmètre : MVP collaboratif Kanban en 4 mois

Date : 24/10/2025 • Version : 1.0

1) Contexte & objectifs

EpiTrello est une application de gestion de projets inspirée de la méthode Kanban. L'objectif est de permettre à des équipes d'organiser leur travail sous forme de tableaux (boards) regroupant des listes (colonnes) composées de cartes (tâches), avec un suivi visuel et en temps réel de l'avancement.

Objectifs principaux :

- Offrir une expérience fluide et collaborative (mise à jour en temps réel, commentaires, mentions, pièces jointes).
- Proposer un Kanban flexible (listes configurables, étiquettes, priorités, due dates, checklists).
- Fournir des vues adaptées (Board / Calendar / Table) et des filtres efficaces.

2) Périmètre fonctionnel

2.1) MVP (Must Have)

- Comptes & authentification : inscription, connexion, mot de passe oublié, profils utilisateurs.
- Espaces & tableaux : créer / renommer / archiver / supprimer un tableau ; visibilité (privé, équipe, public-lien).
- Listes : ajouter / renommer / réordonner / supprimer.
- Cartes : CRUD complet ; drag & drop entre listes ; champs (titre, description, étiquettes, date d'échéance, membres assignés, pièces jointes, checklists, commentaires) ; historique par carte.
- Collaboration : invitations par lien/email, rôles simples (Owner, Membre, Lecteur) au niveau du tableau.
- Recherche & filtres : par titre, étiquette, membre, échéance, état.
- Notifications : in- app (mentions, assignations).
- Internationalisation : FR/EN (détection + switch manuel).

3) Rôles & droits

- Owner du tableau : gestion complète du tableau, réglages, membres, suppression.
- Membre : créer/éditer cartes et listes, commenter, joindre des fichiers.
- Lecteur : lecture seule, filtres personnels, pas d'édition.

4) Parcours utilisateurs clés

- Onboarding : inscription → création d'un premier tableau via template → invitation d'un collègue.
- Gestion des cartes : créer carte, ajouter description + checklist + étiquette + assignation → déplacer au fil du workflow → marquer terminer.
- Collaboration : mentionner un membre, commenter, joindre un fichier.
Organisation : filtrer par étiquette/membre/échéance, sauvegarder un filtre, réordonner listes.

6) Contraintes techniques

- Architecture : Next.js (App Router) + React 18 (TypeScript), shadcn/ui (Radix), Supabase (Postgres, Auth, Storage, Realtime).
- Routing : /boards/[id], /boards/[id]/settings, /auth/* ; layouts imbriqués.
- Accessibilité : shadcn/ui.
- Qualité : TS strict, ESLint + Prettier.
- CI/CD : GitHub Actions (lint, test, build), CloudFare.
- Sécurité : Auth Supabase (email/OAuth), cookies HttpOnly (Next middleware), zod validation, logs d'audit.
- Stockage : Supabase Storage (bucket attachments), drizzle.

9) Règles métier & UX

- Drag & drop.
- Échéances : badges (overdue/soon/today).
- Labels : palette configurable par board, multi- sélection.
- Historique : ActivityLog.
- Invitations : lien signé valable 7 jours + invitation email.
- Archivage : soft- delete.

10) Critères d'acceptation (exemples)

- Création d'un board : depuis “+ Nouveau”, le board apparaît dans la liste et est accessible en < 2 s.

- Déplacement d'une carte : la position est persistée immédiatement et conservée après refresh.
- Assignation : ajouter un membre affiche son avatar et envoie une notification in-app.
- Filtre par label : n'affiche que les cartes taggées ; désactivation → vue complète.
- Commentaire : publication visible instantanément pour tous les membres connectés au board.

12) Sécurité & conformité

- TLS en transit ; stockage chiffré côté fournisseur.
- RBAC.
- Journal d'audit exportable (CSV/JSON).
- RGPD : export/suppression données compte ; mentions légales & politique de confidentialité.

13) Monitoring & analytics

- Logs applicatifs (info/warn/error), corrélation par requête.
- Métriques : temps de réponse, erreurs 5xx, connexions realtime, taille moyenne des boards.

14) Planning indicatif (4 mois ~ 16 semaines)

S1–S2 – Foundations

- Setup repo, CI/CD (GitHub Actions), shadcn/ui (theme).
- Auth Supabase, modèles DB.
- Pages : /auth, /boards (listing), création board.

S3–S4 – Kanban de base

- CRUD Lists & Cards, dnd-kit.

S5–S6 – Collaboration

- Membres & rôles, commentaires temps réel, labels, assignations, due dates, filtres de base.

S7–S8 – Fichiers & Recherche

- Upload Storage, aperçu images/PDF, recherche Filtrées, centre de notifications in- app.

S9–S10 – Vues & i18n

- Vue Calendrier (échéances), FR/EN, accessibilité.

S11–S12 – Realtime avancé & Perf

- Optimistic UI + rollback, métriques et monitoring.

S13–S14 – Durcissement & Tests

- Tests et Review.

S15–S16 – Stabilisation & Livraison

- Corrections, doc utilisateur & préparation prod.

Jalons : MVP démo fin S6 • Feature- complete fin S12 • Release candidate fin S16

15) Livrables

- Code source (front/back) + README d'installation.
- Schéma de données & spécifications d'accès.
- Guide utilisateur (PDF/MD) +tutoriels courts.

16) Gestion de projet

- Méthode : Kanban.
- Backlog : GitHub Issues taggées.
- Conventions : Conventional Commits ; PR review (2 approbations pour security).
- Docs : /docs (MDX).

17) Inventaire UI (shadcn/ui)

Primitives : Button, Input, Textarea, Dialog/Drawer, DropdownMenu, Tabs, Badge, Tooltip, Toast, Avatar, Skeleton.

Composants EpiTrello : BoardHeader, BoardSidebar, ListColumn, CardItem, CardModal, LabelPicker, MemberPicker, FilterBar, CalendarView, TableView, NotificationCenter.

Conclusion

Ce cahier des charges détaille les exigences pour le développement d'EpiTrello, une application de gestion de projets Kanban. Il couvre le contexte, les objectifs, le périmètre fonctionnel, les rôles et droits, les parcours utilisateurs clés et les exigences non fonctionnelles. Ce document servira de guide pour l'équipe de développement tout au long du projet.