

Cahier des charges

Projet Stock Merch

Thomas Siest

Sommaire

Présentation générale du projet.....	3
Objectifs du projet	3
Périmètre fonctionnel.....	3
Fonctionnalités principales (MVP)	3
Fonctionnalités secondaires (hors MVP – à implémenter plus tard).....	3
Contraintes techniques.....	4
Contraintes de développement	4
Méthodologie de travail	4
Planning très prévisionnel	4
Livrables attendus.....	5

Présentation générale du projet

Nom du projet : Stock Merch

Porté par : Thomas Siest – Étudiant M2 CYBER 2024-2025

Contexte : Projet de Master visant à développer une application de gestion de stocks de goodies (ou « merch ») destinée à une utilisation associative (notamment pour l'association United Glowstick) ainsi qu'à des collectionneurs individuels.

Objectifs du projet

Le projet vise à développer une application web mobile-first (et mobile-native dans un second temps) pour :

- Faciliter la gestion des stocks de goodies lors des événements associatifs ;
- Permettre aux collectionneurs de gérer leur collection personnelle de manière simple et intuitive.

Périmètre fonctionnel

Fonctionnalités principales (MVP)

Pour les associations :

- Gestion CRUD des stocks de produits
- Visualisation des quantités en temps réel
- Suivi des ventes et bénéfices par événement
- Accès multi-utilisateur avec gestion des permissions

Pour les utilisateurs individuels :

- Ajout d'articles personnels (nom, description, photo, prix d'achat, lieu de stockage)
- Statistiques personnelles (nombre total d'objets, valeur estimée, etc.)

Fonctionnalités secondaires (hors MVP – à implémenter plus tard)

- Mode enchères pour certains produits
- Historique d'achats
- Comparaison de collections entre utilisateurs
- Exportation ou partage de collection

Contraintes techniques

- Développement initial mobile-first en React (application web responsive)
- Intégration de Capacitor prévue en fin de projet pour publication mobile (iOS, Android, PWA)
- Backend serverless via Convex.dev
- Récupération et mise à jour des données via TanStack Query
- Stockage d'images sur un service tiers (Cloudinary ou Supabase Storage)
- Hébergement sur Vercel, versionnage via GitHub

Contraintes de développement

- Respect des principes SOLID
- Architecture modulaire
- Tests unitaires obligatoires pour chaque nouvelle fonctionnalité (avec Vitest)
- Maquettage uniquement des fonctionnalités essentielles (via Figma)
- Documentation continue (technique et fonctionnelle)

Méthodologie de travail

- Méthodologie agile : organisation par itérations courtes (Scrum ou Kanban)
- Gestion du projet via JIRA / GitHub Projects
- Revue de code systématique
- Démonstrations régulières à l'équipe de l'association pour validation fonctionnelle

Planning très prévisionnel

Semaine	Dates	Objectifs principaux
Semaine 1	24 mars → 30 mars	Finalisation du cahier des charges, revue des besoins prioritaires (MVP), préparation backlog + découpage des tâches dans JIRA
Semaine 2	31 mars → 6 avril	Maquettes Figma basiques, initialisation repo GitHub, configuration de l'environnement React + Vite + Convex + TanStack
Semaine 3	7 avril → 13 avril	Développement authentification, modèles backend, affichage goodies

Semaine 4	14 avril → 20 avril	CRUD goodies, stockage de photos, premiers tests unitaires
Semaine 5	21 avril → 27 avril	Quantité en temps réel, permissions utilisateurs
Semaine 6	28 avril → 4 mai	Ventes par événement, bénéfices, statistiques personnelles
Semaine 7	5 mai → 11 mai	Tests manuels et automatiques, responsive, retours des utilisateurs
Semaine 8	12 mai → 18 mai	Correctifs, intégration Capacitor si possible, documentation utilisateur
Semaine 9	19 mai → 25 mai	Documentation technique, préparation démo, cas d'usage
Semaine 10	26 mai → 31 mai	Révision finale du code, soutenance, livraison finale

Livrables attendus

- Application web responsive fonctionnelle
- Code source versionné sur GitHub
- Jeux de tests unitaires et rapport de couverture
- Présentation orale (soutenance)
- Cahier des charges (ce document)