

## **OBJET**

### Spécifications complètes MVP Toolbox24 (Web + App + Admin intégré)

---

## **CONTEXTE GLOBAL**

Bonjour Guillaume,

Je te transmets le document regroupant l'ensemble des visuels du MVP **Toolbox24** .

Ce document représente **l'intégralité du produit**, incluant :

- le parcours client
- les interfaces web
- la logique métier
- et **l'interface administrateur (intégrée dès le MVP)**

👉 Objectif :

Créer une plateforme complète de location d'outillage en libre-service via casiers connectés, **directement exploitable en production dès la V1.**

---

## **1. PARCOURS UTILISATEUR (COEUR DU PRODUIT)**

### **Workflow principal :**

1. Réservation d'un outil
2. Paiement + empreinte CB (caution)
3. Génération QR code + code PIN
4. Accès au container
5. Ouverture du casier
6. Utilisation avec minuterie
7. Retour avec contrôle obligatoire
8. Ajustement automatique
9. Libération caution

👉 Ce flux est **100% fonctionnel dès le MVP**



## 2. FRONT OFFICE (UTILISATEUR)



### Accueil

- présentation du concept
- CTA "Réserver"
- accès aux catégories



### Catalogue

- liste des outils
- filtres
- prix dynamiques



### Réservation

- choix outil
- durée
- calcul automatique



### Paielement

- Stripe
- gestion caution (empreinte CB)



### Confirmation

- QR code
  - code PIN
  - détails réservation
-



## Dashboard utilisateur

- locations en cours
  - historique
  - accès rapide
- 



## Compte utilisateur

- infos personnelles
  - moyens de paiement
  - notifications
  - favoris
  - factures
- 



## 3. MODULE CASIERS (CRITIQUE)

👉 À connecter côté backend / API :

- ouverture via QR code ou PIN
- attribution automatique d'un casier
- gestion disponibilité casiers

👉 Logique :

Réservation → Casier attribué → Code généré → Ouverture → Utilisation → Retour

---



## 4. CONTRÔLE MATÉRIEL (OBLIGATOIRE)

### ◆ Avant utilisation

- photo obligatoire via application
- état du matériel :
  - ● bon
  - ● moyen
  - ● mauvais

## ◆ Après utilisation

- même process

👉 Condition critique :

! Sans photos → restitution caution bloquée

---



## 5. MINUTEUR & LOCATION

- timer en temps réel
  - prolongation possible
  - recalcul automatique du tarif
- 



## 6. CAUTION & PAIEMENT

- empreinte CB au départ
  - ajustement automatique en fin de location :
    - temps réel
    - éventuels dommages
- 



## 7. BOUTIQUE CONSOMMABLES

👉 Deux cas :

### 1. Ajout à une location

- débit automatique via caution

### 2. Achat seul

- paiement classique
  - casier attribué avec code
-

## 8. ADMIN (INTÉGRÉ DÈS LE MVP

👉 Partie essentielle à développer immédiatement

### Modules admin :

#### Gestion des outils

- ajout / modification / suppression
  - statut (disponible / en maintenance)
- 

#### Gestion des casiers

- état des casiers
  - attribution manuelle si besoin
  - suivi ouverture / fermeture
- 

#### Gestion des réservations

- visualisation en temps réel
  - historique
  - statut (en cours / terminé / problème)
- 

#### Gestion utilisateurs

- comptes clients
  - historique
  - blocage si besoin
- 

#### Gestion paiements

- suivi des paiements
  - gestion cautions
  - remboursements
- 

#### Gestion incidents

- matériel endommagé
  - litiges
  - validation manuelle
- 

### Validation des photos

- contrôle humain possible
  - validation état matériel
- 

### Dashboard admin

- stats globales
  - CA
  - utilisation des outils
  - taux de rotation
- 

## 9. DESIGN / UI

- fond noir
  - accents rouges
  - interface moderne type SaaS
  - priorité à :
    - lisibilité
    - rapidité
    - efficacité
- 

## 10. OBJECTIF DU MVP

👉 Contrairement à un MVP classique :

! Ici l'objectif est une version **complète et exploitable immédiatement**

- ✓ front client
- ✓ logique métier
- ✓ paiement
- ✓ casiers
- ✓ contrôle matériel
- ✓ **admin complet intégré**

---

## 11. ARCHITECTURE ATTENDUE (SUGGÉRÉE)

👉 Stack recommandée :

- Front : Next.js / React
- Backend : Supabase ou Node.js
- Paiement : Stripe
- Temps réel : WebSocket / Supabase
- API casiers : module séparé

---

## 12. PHILOSOPHIE PRODUIT

Le système doit être :

- ✓ autonome
- ✓ sécurisé
- ✓ simple à utiliser
- ✓ robuste

👉 L'utilisateur ne doit jamais avoir besoin d'assistance pour louer

---

## CONCLUSION

Ce document représente :

- un produit complet
- une expérience utilisateur finalisée
- une base solide pour développement

👉 L'objectif est d'obtenir une V1 directement opérationnelle sur le terrain.

---