

Product Requirements Document (PRD)

Application de Gestion de Maintenance Automobile

Version: 1.0

Date: 13 Janvier 2026

Auteur: Anto

Statut: Draft pour développement

1. Vue d'ensemble du produit

1.1 Vision du produit

Application mobile iOS permettant aux propriétaires de véhicules de gérer efficacement l'entretien de leur voiture, d'optimiser sa durée de vie et de maîtriser leurs dépenses automobiles grâce à un suivi intelligent et personnalisé.

1.2 Objectifs business

- Lancer sur l'App Store une application gratuite avec modèle freemium
- Différenciation par rapport aux calendriers natifs via interface spécialisée et IA
- Monétisation via fonctionnalités premium (multi-véhicules, exports, analyses avancées)
- Création d'une base utilisateurs fidèles grâce à l'engagement émotionnel (photo du véhicule)

1.3 Public cible

Version 1.0 (MVP):

- Propriétaires individuels de véhicules
- Un seul véhicule par utilisateur (gratuit)
- Utilisateurs souhaitant optimiser la durée de vie de leur voiture

Versions futures:

- Gestionnaires de flottes (premium)
 - Propriétaires de plusieurs véhicules (premium)
-

2. Architecture du produit

2.1 Modèle de revenus

Version Gratuite (Free Tier)

- Gestion d'**un seul véhicule**
- Toutes les fonctionnalités de base
- Chatbot IA avec limite mensuelle (à définir, ex: 50 requêtes/mois)
- Stockage cloud inclus
- Notifications de base

Version Premium (Paid Tier)

Fonctionnalités débloquées:

- Support **multi-véhicules illimité**
- **Export de données** (PDF, CSV, Excel)
- Chatbot IA **quasi-illimité** (ex: 500 requêtes/mois)
- **Analyses et statistiques avancées**
 - Graphiques de coûts par catégorie
 - Prédictions de maintenance
 - Comparaison entre véhicules
 - Rapports détaillés
- Stockage cloud illimité
- Support prioritaire

Prix suggéré: 4,99€/mois ou 49,99€/an (à valider selon market research)

2.2 Stack technique

Frontend (Mobile)

- **Plateforme:** iOS (Swift/SwiftUI ou React Native)
- **Design system:** Thème sombre avec texte blanc
- **Stockage local:** Core Data / SQLite / Realm
- **OCR:** Vision Framework (Apple) ou Tesseract

Backend

- **Architecture:** Cloud synchronisé
- **Options à considérer:**
 - Firebase (Firestore + Auth + Storage + Cloud Functions)
 - Supabase (PostgreSQL + Auth + Storage + Edge Functions)
 - AWS Amplify
- **Storage:** Cloud storage pour photos et documents scannés

- **Backup:** Automatique et continu

IA & API

- **Chatbot:**
 - Option 1: OpenAI API (GPT-4)
 - Option 2: Perplexity API
 - À définir selon coûts et performances
- **Rate limiting:** Implémentation côté serveur pour gérer quotas gratuit/premium

Données véhicules

- **API constructeurs** (si disponible)
 - **Base de données publiques** maintenance recommandée
 - **Fallback:** Données manuelles enrichies progressivement
-

3. Spécifications fonctionnelles détaillées

3.1 PAGE 1: Dashboard / Accueil

3.1.1 Section principale: Véhicule

Affichage:

- Photo réelle du véhicule de l'utilisateur (upload par l'utilisateur)
- Informations clés superposées ou en dessous:
 - **Immatriculation**
 - **Kilométrage actuel**
 - **Type d'énergie** (Essence, Diesel, Électrique, Hybride)
 - **Année de mise en circulation**
 - **Motorisation** (ex: 1.5 dCi 110ch)

Interactions:

- Tap sur la photo → Accès à la page "Détails du véhicule"
- Bouton "✍️ Modifier" pour changer la photo ou les informations

3.1.2 Section: Budget investi

Affichage sous forme de cartes/widgets:

Card 1: Budget total

- Montant total investi depuis le début
- Icône: 

Card 2: Essance/Carburant

- Total dépensé en carburant
- Optionnel: Coût moyen au 100km
- Icône: 

Card 3: Entretien

- Total dépensé en maintenance/réparations
- Sous-catégories (révisions, pneus, freins, etc.)
- Icône: 

Card 4: Autres dépenses

- Assurance (si tracké)
- Stationnement
- Péages
- Amendes
- Icône: 

Période de référence:

- Par défaut: "Depuis l'achat"
- Options: Ce mois, 3 mois, 6 mois, 1 an, Total

3.1.3 Section: Prochaines échéances

- **Widget compact** affichant les 2-3 prochaines maintenances
- Indication de l'urgence (couleur/badge)
- Tap → Redirige vers page Calendrier

3.1.4 Navigation

Bottom Navigation Bar:

1.  **Accueil** (page actuelle)
2.  **Documents**
3.  **Assistant IA**
4.  **Calendrier**

5. Paramètres

3.2 PAGE 2: Détails du véhicule

3.2.1 Informations complètes

Section 1: Identité du véhicule

- Marque & Modèle
- Immatriculation
- VIN (numéro de série)
- Date de première immatriculation
- Date d'achat par l'utilisateur
- Prix d'achat
- Photo(s) du véhicule (galerie)

Section 2: Caractéristiques techniques

- Type de carrosserie
- Motorisation détaillée
- Puissance (chevaux/kW)
- Type d'énergie
- Boîte de vitesses (Manuelle/Auto)
- Nombre de places
- Couleur

Section 3: Kilométrage

- Kilométrage actuel (éditable)
- Kilométrage à l'achat
- Distance parcourue depuis l'achat
- Mise à jour manuelle ou via photo du compteur

Section 4: Historique de propriété

- Nombre de propriétaires précédents
- Historique d'entretien (si disponible avant achat)

Section 5: Documents associés

- Carte grise (photo)

- Certificat de conformité
- Carnet d'entretien
- Contrat d'assurance
- Lien vers page Documents

3.2.2 Actions rapides

- **Modifier les informations**
 - **Mettre à jour le kilométrage**
 - **Voir tous les documents**
 - **Voir les statistiques** (premium)
-

3.3 PAGE 3: Documents & Scanner

3.3.1 Vue liste des documents

Organisation par catégories:

- **Factures d'entretien** (triées par date décroissante)
- **Tickets de carburant**
- **Assurance**
- **Documents administratifs** (carte grise, CT, etc.)
- **Autres**

Chaque item affiche:

- Miniature du document scanné
- Type de document (badge coloré)
- Date
- Montant (si facture)
- Catégorie de maintenance (si applicable)
- Description courte

Filtres & Recherche:

- Barre de recherche par mot-clé
- Filtres: Date, Type, Montant, Catégorie
- Tri: Date, Montant, Type

3.3.2 Scanner de documents (OCR)

Déclenchement:

- Bouton "⊕ Scanner" flottant en bas à droite

Fonctionnement:

1. **Capture:** Appareil photo s'ouvre
2. **Détection automatique** des bords du document
3. **Capture de la photo**
4. **Traitemet OCR:**
 - Extraction automatique:
 - Date de la facture
 - Montant total
 - Nom du garage/fournisseur
 - Type d'intervention (huile, freins, pneus, etc.)
 - Kilométrage (si présent)
 - Intelligence: Reconnaissance des patterns de factures automobiles
5. **Validation & édition:**
 - Prévisualisation des données extraites
 - Possibilité de corriger manuellement
 - Ajout de notes personnelles
 - Catégorisation manuelle si OCR incertain
6. **Sauvegarde:**
 - Création automatique d'une entrée dans la base de données
 - Stockage du document original (photo)
 - Synchronisation cloud

Technologies:

- iOS: Vision Framework + VNDocumentCameraViewController
- Fallback: Tesseract OCR
- ML Model custom pour patterns factures auto (future itération)

3.3.3 Consultation d'un document

Tap sur un document → Vue détaillée:

- Image du document en plein écran (zoom possible)
- Toutes les données extraites

- Notes ajoutées
 - Actions:
 - Modifier
 - Supprimer
 - Partager
 - Exporter (premium)
-

3.4 PAGE 4: Assistant IA (Chatbot)

3.4.1 Interface conversationnelle

Design:

- Chat classique avec bulles de messages
- Photo de profil du véhicule visible en haut
- Indicateur "IA en train d'écrire..."

Contexte de l'IA: L'IA a accès à:

- Toutes les données du véhicule (modèle, année, km, motorisation)
- Historique complet de maintenance (dates, types, coûts)
- Documents scannés (factures, interventions)
- Kilométrage et évolution
- Budget dépensé par catégorie
- Prochaines échéances planifiées

3.4.2 Cas d'usage & exemples de questions

Informations sur l'historique:

- "Quand ai-je fait ma dernière vidange ?"
- "Combien ai-je dépensé en freins cette année ?"
- "Montre-moi toutes les interventions du garage Renault"
- "Quel est mon coût total d'entretien depuis 2024 ?"

Conseils personnalisés:

- "Quand dois-je faire ma prochaine révision ?"
- "Mon modèle a-t-il des problèmes courants à prévoir ?"
- "Quels sont les intervalles recommandés pour la courroie de distribution sur ma voiture ?"

Recherche dans documents:

- "Retrouve-moi la facture du changement de pneus en mars"
- "Quel garage a fait ma dernière révision ?"

Analyses:

- "Quel est mon poste de dépense le plus important ?"
- "Compare mes coûts d'entretien par rapport à la moyenne pour ce modèle"

Rappels & planification:

- "Rappelle-moi de faire la vidange dans 5000 km"
- "Quand expire mon contrôle technique ?"

3.4.3 Fonctionnalités IA avancées

Base de connaissances:

- Connexion à base de données de problèmes courants par modèle
- Intervalles de maintenance recommandés par constructeur
- Prix moyens du marché pour chaque type d'intervention

Suggestions proactives:

- Alertes basées sur le kilométrage
- Détection d'anomalies (ex: coût anormalement élevé)
- Recommandations de maintenance préventive

3.4.4 Limites & quotas

Version gratuite:

- 50 messages/mois (exemple)
- Badge "X messages restants ce mois"
- Prompt pour upgrade vers premium si quota atteint

Version premium:

- 500 messages/mois (quasi-illimité pour usage normal)
- Réponses prioritaires (si API avec tiers)

Gestion technique:

- Rate limiting côté serveur

- Tracking du nombre de requêtes par user
- Reset mensuel des compteurs

3.4.5 API IA

Options à évaluer:

Option 1: OpenAI GPT-4

- **Avantages:** Performance excellente, contexte large, fine-tuning possible
- **Coûts:** ~\$0.03 par requête (à calculer précisément)
- **Intégration:** API REST simple

Option 2: Perplexity API

- **Avantages:** Recherche web intégrée, sources citées
- **Coûts:** À vérifier (potentiellement moins cher)
- **Intégration:** API REST

Critères de décision:

- Coût par requête
- Qualité des réponses pour contexte automobile
- Latence
- Limites de contexte (taille des données véhicule)

Implémentation technique:

- Prompt system personnalisé avec contexte véhicule
- Vectorisation de l'historique de maintenance (RAG)
- Cache des réponses fréquentes pour réduire coûts

3.5 PAGE 5: Calendrier & Échéances

3.5.1 Vue calendrier

Affichage:

- Calendrier mensuel avec pastilles colorées selon urgence
- Vue liste chronologique en dessous

Pastilles de couleur:

-  **Rouge:** Urgent (échéance dépassée ou <7 jours)

- **Orange:** Bientôt (7-30 jours)
- **Vert:** Planifié (>30 jours)
- **Gris:** Optionnel/conseil

3.5.2 Types d'échéances

Catégories de maintenance trackées:

1. **Vidange & filtres**

- Vidange huile moteur
- Filtre à huile
- Filtre à air
- Filtre à carburant
- Filtre habitacle

2. **Freinage**

- Plaquettes de frein avant
- Plaquettes de frein arrière
- Disques de frein
- Liquide de frein

3. **Mécanique**

- Courroie de distribution
- Courroie d'accessoires
- Batterie
- Bougies d'allumage
- Embrayage

4. **Pneumatiques**

- Changement de pneus
- Permutation des pneus
- Géométrie/parallélisme
- Équilibrage

5. **Révisions**

- Révision constructeur
- Contrôle technique (obligatoire)
- Contre-visite CT

6. **Climatisation**

- Recharge climatisation

- Remplacement filtre habitacle

7. Personnalisé

- Maintenance libre définie par l'utilisateur

3.5.3 Création d'échéance

Bouton "⊕ Nouvelle échéance"

Formulaire:

1. **Type de maintenance** (sélection parmi catégories)

2. **Description** (optionnel)

3. **Type de rappel:**

-  Date fixe
-  Kilométrage
-  Date OU kilométrage (premier atteint)

4. **Date prévue** (si date fixe)

5. **Kilométrage prévu** (si basé sur km)

6. **Avance de notification:**

- Nombre de jours avant (ex: 7, 14, 30 jours)
- OU kilométrage avant (ex: 500, 1000 km)

7. **Récurrence** (optionnel):

- Tous les X mois
- Tous les X km
- Tous les X mois OU X km

8. **Coût estimé** (optionnel)

9. **Garage/lieu** (optionnel)

10. **Notes**

Suggestions intelligentes:

- Basées sur le modèle du véhicule
- Intervalles constructeur
- Historique des maintenances passées

3.5.4 Notifications & rappels

Système de notifications push:

Déclencheurs:

- X jours avant l'échéance (personnalisable)
- X km avant l'échéance (si basé sur kilométrage)
- Le jour J
- Après échéance dépassée (relances)

Contenu de la notification:

- "🔔 Révision des 60 000 km dans 7 jours"
- "⚠️ Contrôle technique expiré depuis 3 jours"
- "🛢️ Vidange recommandée dans 500 km"

Paramétrage:

- Activation/désactivation par catégorie
- Personnalisation des délais
- Heures autorisées pour notifications

Implémentation technique:

- iOS: Local Notifications (pour rappels hors ligne)
- Push Notifications (pour synchro multi-devices en premium)
- Background tasks pour vérifier kilométrage vs échéances

3.5.5 Complétion d'échéance

Quand une maintenance est effectuée:

Action: "✅ Marquer comme fait"

Formulaire rapide:

1. Date de réalisation (pré-remplie avec aujourd'hui)
2. Kilométrage de réalisation (pré-rempli avec km actuel)
3. Coût réel
4. Garage/lieu
5. Notes
6. 📄 **Attacher une facture** (scan ou sélection depuis Documents)

Actions automatiques:

- Mise à jour de l'historique de maintenance
- Ajout dans base de données pour chatbot
- Calcul de la prochaine échéance (si récurrente)

- Mise à jour des statistiques budgétaires
-

4. Spécifications techniques

4.1 Base de données locale (Device)

Schéma de données

Table: VEHICLES

```
id (UUID, PRIMARY KEY)
user_id (UUID, FOREIGN KEY)
brand (VARCHAR)
model (VARCHAR)
registration_plate (VARCHAR)
vin (VARCHAR)
year (INTEGER)
purchase_date (DATE)
purchase_price (DECIMAL)
purchase_mileage (INTEGER)
current_mileage (INTEGER)
fuel_type (ENUM: gasoline, diesel, electric, hybrid)
engine (VARCHAR)
transmission (ENUM: manual, automatic)
body_type (VARCHAR)
color (VARCHAR)
seats (INTEGER)
photo_url (VARCHAR)
created_at (TIMESTAMP)
updated_at (TIMESTAMP)
```

Table: MAINTENANCE_HISTORY

id (UUID, PRIMARY KEY)
vehicle_id (UUID, FOREIGN KEY)
category (ENUM: oil_change, brakes, filters, tires, mechanical, revision, ac, custom)
subcategory (VARCHAR)
date (DATE)
mileage (INTEGER)
cost (DECIMAL)
description (TEXT)
location (VARCHAR) -- Garage name
invoice_id (UUID, FOREIGN KEY, nullable)
notes (TEXT)
created_at (TIMESTAMP)
updated_at (TIMESTAMP)

Table: MAINTENANCE_SCHEDULE

id (UUID, PRIMARY KEY)
vehicle_id (UUID, FOREIGN KEY)
category (ENUM)
subcategory (VARCHAR)
description (TEXT)
reminder_type (ENUM: date, mileage, both)
due_date (DATE, nullable)
due_mileage (INTEGER, nullable)
notification_advance_days (INTEGER)
notification_advance_km (INTEGER)
recurrence_type (ENUM: none, monthly, yearly, km_based)
recurrence_value (INTEGER) -- Months, years, or km
estimated_cost (DECIMAL, nullable)
location (VARCHAR, nullable)
notes (TEXT)
status (ENUM: pending, completed, overdue)
completed_at (TIMESTAMP, nullable)
created_at (TIMESTAMP)
updated_at (TIMESTAMP)

Table: DOCUMENTS

id (UUID, PRIMARY KEY)
vehicle_id (UUID, FOREIGN KEY)
type (ENUM: invoice, fuel_receipt, insurance, administrative, other)
category (ENUM, nullable) -- Liens avec catégories maintenance
date (DATE)
amount (DECIMAL, nullable)
vendor (VARCHAR, nullable) -- Nom garage/fournisseur
description (TEXT)
file_path (VARCHAR) -- Local ou cloud URL
ocr_data (JSON) -- Données extraites par OCR
mileage (INTEGER, nullable)
notes (TEXT)
created_at (TIMESTAMP)
updated_at (TIMESTAMP)

Table: EXPENSES

id (UUID, PRIMARY KEY)
vehicle_id (UUID, FOREIGN KEY)
type (ENUM: fuel, maintenance, insurance, parking, tolls, fines, other)
date (DATE)
amount (DECIMAL)
mileage (INTEGER, nullable)
description (TEXT)
document_id (UUID, FOREIGN KEY, nullable)
created_at (TIMESTAMP)
updated_at (TIMESTAMP)

Table: AI_CHAT_HISTORY

id (UUID, PRIMARY KEY)
user_id (UUID, FOREIGN KEY)
vehicle_id (UUID, FOREIGN KEY)
message (TEXT)
response (TEXT)
tokens_used (INTEGER)
timestamp (TIMESTAMP)

Table: USER_SETTINGS

```
id (UUID, PRIMARY KEY)
user_id (UUID, FOREIGN KEY)
notification_enabled (BOOLEAN)
notification_time_start (TIME)
notification_time_end (TIME)
notification_categories (JSON) -- Array des catégories activées
mileage_unit (ENUM: km, miles)
currency (VARCHAR)
language (VARCHAR)
theme (ENUM: dark, light, auto)
created_at (TIMESTAMP)
updated_at (TIMESTAMP)
```

4.2 Backend Cloud (Firebase/Supabase)

Architecture recommandée: Firebase

Services utilisés:

1. Firebase Authentication

- Email/Password
- Apple Sign-In (requis pour App Store)
- Google Sign-In (optionnel)

2. Cloud Firestore

- Synchronisation temps réel entre devices
- Structure similaire aux tables locales
- Règles de sécurité strictes

3. Firebase Storage

- Stockage des photos véhicules
- Stockage des documents scannés
- Organisation: `/users/{userId}/vehicles/{vehicleId}/`

4. Cloud Functions

- Traitement OCR côté serveur (si lourd)
- Gestion des quotas API IA
- Envoi de notifications programmées
- Génération de rapports (premium)

5. Firebase Cloud Messaging (FCM)

- Notifications push cross-device

Synchronisation

Stratégie:

- Offline-first: App fonctionne 100% hors ligne
- Sync automatique quand connexion disponible
- Conflict resolution: Last-write-wins avec timestamp
- Queue de sync pour actions hors ligne

4.3 Sécurité & Privacy

Données sensibles:

- Chiffrement end-to-end pour documents
- Pas de stockage des données bancaires complètes
- Conformité RGPD (utilisateurs EU)
- Exportation complète des données (RGPD)
- Suppression définitive sur demande

Backup:

- Backup automatique quotidien vers cloud
- Retention: 30 jours
- Restauration possible via paramètres

4.4 Performance

Optimisations:

- Lazy loading des images documents
- Compression automatique des photos
- Pagination des listes longues (historique)
- Cache intelligent des données IA

Métriques cibles:

- Temps de lancement: < 2 secondes
 - Temps de réponse IA: < 5 secondes
 - Sync cloud: < 10 secondes
 - Poids app: < 50 MB
-

5. Expérience utilisateur (UX)

5.1 Onboarding (Première utilisation)

Étape 1: Bienvenue

- Écran splash avec logo
- Value proposition claire
- Bouton "Commencer"

Étape 2: Authentification

- Créer un compte (email + mot de passe)
- Ou Apple Sign-In
- Ou Google Sign-In

Étape 3: Ajout du premier véhicule

- "Parlez-nous de votre voiture"
- Formulaire simplifié:
 - Marque (sélection ou saisie)
 - Modèle
 - Année
 - Immatriculation
 - Kilométrage actuel
- Option: "Scanner la carte grise" pour auto-remplir

Étape 4: Photo du véhicule

- "Ajoutez une photo de votre voiture"
- Appareil photo ou galerie
- Skip possible

Étape 5: Notifications

- "Recevez des rappels pour l'entretien"
- Autorisation notifications iOS
- Explication de la valeur

Étape 6: Tour guidé (optionnel)

- Tooltips sur les fonctionnalités principales
- Skip possible

Étape 7: Dashboard

- Arrivée sur l'écran d'accueil
- Badge "Nouveau" sur fonctionnalités clés

5.2 Flux utilisateur principaux

Flux 1: Ajouter une facture d'entretien

1. Tap sur "Documents" dans bottom nav
2. Tap sur bouton "⊕ Scanner"
3. Appareil photo s'ouvre
4. Capture de la facture
5. OCR traite automatiquement
6. Écran de validation avec données extraites
7. Correction si nécessaire
8. "Sauvegarder"
9. Retour à liste Documents avec nouveau document visible
10. Badge de notification sur "Calendrier" si échéance créée automatiquement

Flux 2: Demander conseil à l'IA

1. Tap sur "Assistant IA" dans bottom nav
2. Saisie de la question (ex: "Quand faire ma vidange ?")
3. Envoi
4. IA traite (indicateur de chargement)
5. Réponse affichée avec contexte du véhicule
6. Possibilité de follow-up
7. Si quota atteint: Prompt "Passer à Premium"

Flux 3: Planifier une maintenance

1. Tap sur "Calendrier" dans bottom nav
2. Tap sur "⊕ Nouvelle échéance"
3. Sélection du type (ex: Vidange)
4. Remplissage du formulaire
 - IA suggère intervalle recommandé
5. Choix du rappel (ex: 7 jours avant)
6. "Créer"

7. Échéance visible dans calendrier avec pastille verte

Flux 4: Marquer maintenance comme effectuée

1. Notification iOS: "Révision dans 2 jours"
2. Tap sur notification → App ouvre Calendrier
3. Tap sur l'échéance concernée
4. Bouton " Marquer comme fait"
5. Formulaire rapide (date, coût, garage)
6. Option "Attacher facture" → Scanner
7. "Valider"
8. Échéance disparaît du calendrier
9. Ajout automatique dans historique
10. Prochaine échéance créée si récurrent

5.3 États de l'application

États vides (Empty states):

- Aucun véhicule: Illustration + "Ajoutez votre premier véhicule"
- Aucun document: "Scannez votre première facture"
- Aucune échéance: "Planifiez votre prochain entretien"
- Pas d'historique chat: "Posez votre première question"

États de chargement:

- Skeleton screens pour listes
- Spinners pour actions ponctuelles
- Progress bar pour OCR et IA

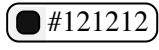
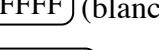
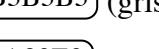
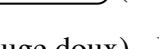
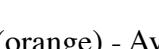
États d'erreur:

- Messages clairs et actionnables
 - Bouton "Réessayer"
 - Fallback graceful (ex: OCR échoue → saisie manuelle)
-

6. Design system & UI

6.1 Thème visuel

Palette de couleurs (Dark Theme):

- **Background principal:**  #121212 (noir profond)
- **Background secondaire:**  #1E1E1E (cartes, modals)
- **Texte principal:**  #FFFFFF (blanc)
- **Texte secondaire:**  #B3B3B3 (gris clair)
- **Accent principal:**  #4A90E2 (bleu clair) - Actions, liens
- **Accent secondaire:**  #50C878 (vert) - Succès, validations
- **Alerte:**  #FF6B6B (rouge doux) - Urgence, erreurs
- **Warning:**  #FFB347 (orange) - Avertissements
- **Borders:**  #2C2C2C (gris très foncé)

Typographie:

- **Titre H1:** SF Pro Display Bold, 28pt
- **Titre H2:** SF Pro Display Semibold, 22pt
- **Titre H3:** SF Pro Display Semibold, 18pt
- **Body:** SF Pro Text Regular, 16pt
- **Caption:** SF Pro Text Regular, 14pt
- **Small:** SF Pro Text Regular, 12pt

Iconographie:

- SF Symbols (natif iOS)
- Style: Rounded, poids Medium
- Taille: 20x20pt (small), 24x24pt (medium), 32x32pt (large)

6.2 Composants UI réutilisables

Cards:

- Radius: 12pt
- Shadow: 0 2px 8px rgba(0,0,0,0.3)
- Padding: 16pt

Boutons:

- **Primary:** Background bleu accent, texte blanc, radius 8pt

- **Secondary:** Border bleu accent, texte bleu, radius 8pt
- **Destructif:** Background rouge, texte blanc

Input fields:

- Background:  #1E1E1E
- Border: 1pt  #2C2C2C
- Border active: 2pt bleu accent
- Radius: 8pt
- Height: 48pt

Bottom Navigation:

- Height: 60pt
- 5 items centrés
- Icône + label
- Active state: Accent color + bold

Floating Action Button (FAB):

- Position: Bottom right, 16pt margin
 - Size: 56x56pt
 - Shadow: 0 4px 12px rgba(0,0,0,0.4)
 - Background: Accent color
-

7. Roadmap & Priorisation

7.1 Version 1.0 (MVP - Launch)

Fonctionnalités essentielles:

-  Authentification (email + Apple Sign-In)
-  Gestion d'un seul véhicule
-  Dashboard avec photo + infos clés + budget
-  Page détails véhicule complet
-  Scanner de documents avec OCR basique
-  Liste et consultation des documents
-  Calendrier de maintenance avec échéances
-  Notifications de rappel personnalisées
-  Chatbot IA (quota gratuit: 50 msg/mois)

- Synchronisation cloud basique
- Backup automatique
- Dark theme

Catégories de maintenance MVP:

- Vidange & filtres
- Freinage
- Révisions & contrôle technique
- Personnalisé

Hors scope V1.0:

- Multi-véhicules
- Export de données
- Statistiques avancées
- Mode clair (light theme)
- Partage avec mécanicien
- Intégration calendrier iOS natif

Timeline estimée: 3-4 mois

7.2 Version 1.1 (Post-Launch Improvements)

1-2 mois après V1.0:

- Bug fixes basés sur feedback utilisateurs
- Statistiques basiques (graphiques coûts par catégorie)
- Amélioration des notifications (options avancées)
- Personnalisation thème (light/dark/auto)
- Localisation (FR, EN initialement)
- Catégories de maintenance étendues (pneus, batterie, clim)

7.3 Version 2.0 (Premium Features)

3-6 mois après launch:

- **Lancement modèle Premium:**
 - Support multi-véhicules illimité
 - Export PDF/CSV/Excel
 - Chatbot IA quasi-illimité
 - Analyses & statistiques avancées

- Dashboard analytics avancé
- Comparaison entre véhicules (premium)
- Partage sécurisé avec mécanicien (premium)
- Intégration calendrier iOS natif
- Widgets iOS (home screen)

7.4 Version 3.0 (Scale & Fleet)

6-12 mois après launch:

- Mode "Gestionnaire de flotte"
 - Gestion multi-utilisateurs (entreprises)
 - Prédictions IA avancées (maintenance prédictive)
 - API publique pour intégrations tierces
 - Chat avec mécaniciens partenaires
 - Programme de fidélité / gamification
-

8. Métriques de succès (KPIs)

8.1 Acquisition

- **Téléchargements App Store:** Objectif 10k premiers 6 mois
- **Taux de conversion listing → install:** > 20%
- **Coût d'acquisition (CAC):** < 5€ (si ads)

8.2 Activation

- **Taux de complétion onboarding:** > 80%
- **Taux d'ajout premier véhicule:** > 90%
- **Temps moyen d'onboarding:** < 3 minutes

8.3 Engagement

- **DAU (Daily Active Users):** 30% des MAU
- **MAU (Monthly Active Users):** 50% des installs
- **Sessions par semaine:** > 3
- **Durée moyenne session:** 2-5 minutes
- **Taux de rétention J7:** > 40%
- **Taux de rétention J30:** > 25%

8.4 Monétisation

- **Taux de conversion Free → Premium:** > 5%
- **ARPU (Average Revenue Per User):** > 2€/mois
- **LTV (Lifetime Value):** > 50€
- **Taux de churn premium:** < 5%/mois

8.5 Qualité

- **Crash-free rate:** > 99.5%
- **Note App Store:** > 4.5/5
- **NPS (Net Promoter Score):** > 40
- **Temps de réponse IA:** < 5 secondes (P95)

8.6 Fonctionnalités

- **Documents scannés par user actif:** > 5/mois
 - **Questions posées à l'IA:** > 3/mois
 - **Échéances créées:** > 2/véhicule
 - **Taux de complétion échéances:** > 60%
-

9. Risques & Mitigations

9.1 Risques techniques

Risque 1: Coûts API IA incontrôlables

- **Impact:** Élevé
- **Probabilité:** Moyenne
- **Mitigation:**
 - Quotas stricts par tier
 - Cache des réponses fréquentes
 - Monitoring en temps réel des coûts
 - Circuit breaker si coûts explosent
 - Option de basculer entre providers

Risque 2: OCR peu fiable

- **Impact:** Moyen
- **Probabilité:** Élevée

- **Mitigation:**

- Toujours permettre édition manuelle
- Amélioration progressive avec ML custom
- Feedback loop: user corrige → améliore modèle
- Templates de factures pré-configurés pour garages populaires

Risque 3: Problèmes de synchronisation cloud

- **Impact:** Élevé

- **Probabilité:** Moyenne

- **Mitigation:**

- Offline-first architecture
- Queue de synchronisation robuste
- Conflict resolution clair
- Logs détaillés pour debugging

9.2 Risques business

Risque 4: Adoption faible du modèle premium

- **Impact:** Élevé

- **Probabilité:** Moyenne

- **Mitigation:**

- Tester pricing (A/B testing)
- Communiquer clairement la valeur
- Trial gratuit 14 jours
- Upsells contextuels dans l'app

Risque 5: Concurrence (Apps similaires)

- **Impact:** Moyen

- **Probabilité:** Élevée

- **Mitigation:**

- Différenciation par l'IA personnalisée
- UX supérieure (photo véhicule, design)
- Community building
- Itérations rapides basées sur feedback

Risque 6: Rejet App Store

- **Impact:** Critique
- **Probabilité:** Faible
- **Mitigation:**
 - Conformité stricte App Store Guidelines
 - Apple Sign-In implémenté
 - Privacy policy claire
 - Beta testing avec TestFlight avant soumission

9.3 Risques légaux & conformité

Risque 7: Non-conformité RGPD

- **Impact:** Critique
- **Probabilité:** Faible
- **Mitigation:**
 - Privacy by design
 - Consentement explicite
 - Export de données
 - Droit à l'oubli implémenté
 - DPO si nécessaire

Risque 8: Stockage de données sensibles (cartes de paiement)

- **Impact:** Critique
- **Probabilité:** Faible (hors scope)
- **Mitigation:**
 - Ne JAMAIS stocker de données de paiement complètes
 - Utiliser Apple Pay / Stripe pour paiements premium
 - PCI-DSS compliance si besoin

10. Annexes

10.1 Glossaire

- **OCR:** Optical Character Recognition - Reconnaissance optique de caractères
- **RAG:** Retrieval-Augmented Generation - Génération augmentée par récupération
- **MAU/DAU:** Monthly/Daily Active Users
- **LTV:** Lifetime Value - Valeur vie client

- **ARPU:** Average Revenue Per User
- **NPS:** Net Promoter Score
- **KPI:** Key Performance Indicator
- **MVP:** Minimum Viable Product
- **CAC:** Customer Acquisition Cost

10.2 Références techniques

APIs à considérer:

- OpenAI GPT-4: <https://platform.openai.com/docs/api-reference>
- Perplexity AI: <https://docs.perplexity.ai/>
- Firebase: <https://firebase.google.com/docs>
- Apple Vision Framework: <https://developer.apple.com/documentation/vision>

App Store Guidelines:

- <https://developer.apple.com/app-store/review/guidelines/>

RGPD:

- <https://gdpr.eu/>

10.3 Wireframes & Mockups

Référence:

- Fichiers Figma créés précédemment
- 5 écrans principaux designés avec thème sombre

10.4 Checklist pré-lancement

Développement:

- App iOS fonctionnelle (toutes features MVP)
- Tests unitaires (coverage > 70%)
- Tests UI automatisés
- Tests manuels complets sur devices réels
- Performance optimisée (metrics atteintes)
- Crash-free rate > 99%

Backend & Cloud:

- Firebase setup complet
- Authentification fonctionnelle

- Synchronisation testée
- Backup automatique configuré
- Rate limiting API IA implémenté
- Monitoring & alerting actifs

Légal & Conformité:

- Privacy Policy rédigée
- Terms of Service rédigés
- RGPD compliance vérifiée
- Apple Sign-In implémenté
- Permissions iOS justifiées (appareil photo, notifications)

App Store:

- Developer account Apple actif
- App Store listing rédigé (titre, description, keywords)
- Screenshots 6.5" et 5.5" préparés
- App icon 1024x1024 créé
- Video preview créée (optionnel mais recommandé)
- Métadonnées traduites (FR/EN minimum)

Marketing & Communication:

- Landing page produit créée
- Email de lancement préparé
- Posts réseaux sociaux préparés
- Support email configuré (support@...)
- FAQ rédigée

Analytique:

- Firebase Analytics configuré
 - Events trackés correctement
 - Funnels définis
 - Dashboard KPIs créé
-

11. Questions ouvertes & Décisions à prendre

11.1 Décisions techniques à valider

1. Choix final API IA: OpenAI vs Perplexity

- Besoin: Tester les deux et comparer coûts réels
- Deadline: Avant implémentation chatbot

2. Langage de dev: Swift natif vs React Native

- Pros Swift: Performance, accès APIs iOS natives
- Pros React Native: Potentiel cross-platform futur (Android)
- Recommandation: Swift pour V1.0 (iOS first)

3. Backend: Firebase vs Supabase

- Recommandation: Firebase (écosystème plus mature, docs, extensions)

11.2 Décisions business à valider

1. Prix Premium:

- Option 1: 4,99€/mois ou 49,99€/an
- Option 2: 2,99€/mois ou 29,99€/an (plus accessible)
- Besoin: Analyse concurrence + willingness to pay

2. Quota gratuit IA:

- Proposition: 50 messages/mois
- À ajuster selon coûts réels API

3. Stratégie marketing lancement:

- ASO (App Store Optimization) uniquement?
- Ads payantes (Facebook, Google)?
- Partenariats avec influenceurs auto?
- Budget: TBD

11.3 Features à investiguer

1. Intégration OBD-II (lecteur diagnostic):

- Permettrait lecture automatique kilométrage + codes erreurs
- Complexité élevée, à considérer pour V2.0+

2. Marketplace garages partenaires:

- Réservation directe dans l'app
- Commissions sur transactions
- À considérer pour monétisation V2.0+

3. Social features:

- Partage de conseils entre users
- Forum/community
- Comparaison anonymisée avec autres users même modèle

12. Conclusion & Next Steps

12.1 Récapitulatif

Ce PRD définit une application mobile iOS de gestion de maintenance automobile avec:

- Modèle freemium (gratuit 1 véhicule, premium multi-véhicules + features)
- 5 écrans principaux (Dashboard, Détails, Documents, IA, Calendrier)
- Scanner OCR intelligent pour factures
- Chatbot IA personnalisé connecté à l'historique véhicule
- Système de rappels intelligent basé sur dates et kilométrage
- Synchronisation cloud et backup automatique

12.2 Immediate Next Steps pour développement

Phase 1: Setup (Semaine 1)

1. Créer projet Xcode avec Swift/SwiftUI
2. Setup Firebase project (Auth, Firestore, Storage)
3. Configurer Git repository
4. Setup environnements (Dev, Staging, Prod)

Phase 2: Architecture & Modèles (Semaine 1-2)

1. Implémenter schéma de base de données locale (Core Data)
2. Créer models Swift pour toutes les entités
3. Setup architecture MVVM ou Clean Architecture
4. Implémenter synchronisation Firebase basique

Phase 3: Authentification (Semaine 2)

1. Écrans login/signup
2. Intégration Apple Sign-In
3. Gestion session utilisateur
4. Onboarding flow

Phase 4: Core Features (Semaines 3-8)

1. Dashboard (Semaine 3)

- Layout principal avec photo véhicule
- Cards budget
- Navigation

2. Détails véhicule (Semaine 4)

- Formulaire ajout/édition véhicule
- Affichage informations complètes

3. Documents & Scanner (Semaines 5-6)

- Implémentation OCR avec Vision Framework
- Liste documents
- Upload vers Firebase Storage

4. Calendrier (Semaine 7)

- Vue calendrier
- Création/édition échéances
- Système de notifications local

5. Chatbot IA (Semaine 8)

- Intégration API (OpenAI ou Perplexity)
- UI chat
- Rate limiting

Phase 5: Polish & Testing (Semaines 9-10)

1. UI/UX refinement
2. Bug fixing
3. Performance optimization
4. Testing devices réels
5. Beta testing (TestFlight)

Phase 6: Launch (Semaine 11-12)

1. App Store submission
2. Marketing materials
3. Launch!

Document préparé pour: Claude Code

Status: Ready for development kickoff

Version PRD: 1.0

Dernière mise à jour: 13 Janvier 2026

Appendice: Checklist développeur

Pour Claude Code - utiliser cette checklist pour valider l'implémentation:

Fonctionnalités obligatoires V1.0

- Auth: Email + Apple Sign-In
- Onboarding: 6 étapes guidées
- Dashboard: Photo véhicule + 4 cards budget + navigation
- Détails véhicule: Formulaire complet + affichage
- Scanner: Camera + OCR + extraction données factures
- Documents: Liste + filtres + consultation
- Calendrier: Vue mensuelle + création échéances + rappels
- Notifications: Push locales + personnalisables
- Chatbot: Interface chat + API IA + rate limiting
- Sync: Firebase Firestore + offline-first
- Backup: Automatique quotidien
- Dark theme: Appliqué sur toute l'app

Qualité & Performance

- Temps de lancement < 2s
- Réponse IA < 5s
- Crash-free rate > 99.5%
- Tests unitaires coverage > 70%
- App weight < 50MB
- Offline fonctionnel

App Store Requirements

- Privacy Policy implémentée
- Apple Sign-In obligatoire
- Permissions justifiées (Camera, Notifications)
- Guidelines compliance
- Screenshots + App Icon + Description prêts

Ce PRD est votre référence unique pour l'implémentation. Bonne chance! 🚀