

Cahier des charges - Application mobile de planification hebdomadaire

Vision du projet

Application **mobile native** (iOS et Android) de gestion de tâches hebdomadaires avec système de rappels intelligents. L'objectif est de créer un rituel de planification en fin de semaine et d'accompagner l'utilisateur avec des notifications proactives qui l'aident à rester organisé sans le submerger.

Proposition de valeur unique : Contrairement aux gestionnaires de tâches classiques, cette application se concentre sur le cycle naturel hebdomadaire et crée un rituel de planification qui devient une habitude.

Plateforme et Stack technique

Plateforme

Application mobile uniquement - iOS et Android

- Pas de version web
- Focus 100% sur l'expérience mobile native
- Optimisée pour l'usage quotidien en mobilité

Stack technique

Frontend Mobile

- **Flutter (Dart)**
 - Performance native excellente
 - UI magnifique et fluide
 - Hot reload pour développement rapide
 - Single codebase pour iOS et Android

Gestion d'état

- **Riverpod** (recommandé) ou Provider
- Architecture propre et testable

Navigation

- **Go Router** - Navigation déclarative

Backend

- **Supabase** (Backend-as-a-Service)
 - PostgreSQL (base de données relationnelle)
 - Supabase Auth (authentification)
 - Row Level Security (RLS)
 - Realtime subscriptions
 - Edge Functions (Deno/TypeScript)
 - Storage (si besoin futur)

Base de données locale (hors ligne)

- **Isar** (recommandé) - Ultra rapide, typé
- OU **Hive** - Simple et efficace
- Synchronisation automatique avec Supabase

Notifications

- **Firebase Cloud Messaging (FCM)** pour iOS et Android
- **flutter_local_notifications** pour notifications locales
- Déclenchement via Supabase Edge Functions

Packages Flutter essentiels

```
yaml
```

dependencies:

flutter:

sdk: flutter

Backend & Auth

supabase_flutter: ^2.0.0

State management

flutter_riverpod: ^2.4.0

Navigation

go_router: ^13.0.0

Local database

isar: ^3.1.0

isar_flutter_libs: ^3.1.0

Notifications

firebase_messaging: ^14.7.0

flutter_local_notifications: ^16.3.0

UI & Utils

intl: ^0.19.0

shared_preferences: ^2.2.0

cached_network_image: ^3.3.0

flutter_svg: ^2.0.0

animations: ^2.0.0

Utilities

dio: ^5.4.0

freezed_annotation: ^2.4.0

json_annotation: ^4.8.0

Public cible

- **Profil principal :** Professionnels actifs (25-45 ans) qui jonglent entre vie pro/perso
- **Profils secondaires :**
 - Étudiants avec des deadlines multiples
 - Parents qui organisent les activités familiales
 - Freelances qui gèrent plusieurs projets

FONCTIONNALITÉS ESSENTIELLES (MVP - Version 1.0)

1. SYSTÈME DE PLANIFICATION HEBDOMADAIRE

1.1 Rituel de planification

- **Déclencheur :** Notification locale programmée
 - Jour : au choix (dimanche par défaut)
 - Heure : personnalisable (20h00 par défaut)
 - Message : "C'est l'heure de planifier votre semaine ! 
- **Interface de saisie rapide :**
 - Bottom sheet ou page dédiée
 - Titre : "Planifiez votre semaine"
 - Zone de texte pour saisir rapidement
 - Bouton FAB "+" pour ajouter
 - Ajout en série (validation = nouvelle tâche)
 - Gestes intuitifs (swipe, drag & drop)
 - Animation fluide d'ajout
- **Flexibilité :**
 - Planifier pour semaine actuelle ou suivante
 - Accès à tout moment via onglet ou FAB
 - Possibilité de reporter la planification

1.2 Création/Édition de tâche

Écran de création - Bottom Sheet ou Modal

Champs obligatoires :

-  Titre (3-100 caractères) avec validation en temps réel

Champs optionnels :

-  Date (DatePicker Material/Cupertino)
-  Heure (TimePicker)

- Description (250 caractères max, TextArea expandable)
- Catégorie (Chips sélectionnables avec couleurs)
- Priorité (Haute/Moyenne/Basse - Chips ou Toggle)
- Durée estimée (Dropdown: 15min, 30min, 1h, 2h, 4h+)
- Récurrence (Switch + configuration)

Actions :

- Sauvegarder (avec optimistic update)
- Annuler
- Supprimer (avec confirmation bottom sheet)
- Dupliquer
- Partager (version future)

UX/UI :

- Validation en temps réel
 - Auto-save brouillon (local)
 - Feedback visuel immédiat
 - Animations de transition Material
-

2. SYSTÈME DE RAPPELS INTELLIGENTS

2.1 Rappels par défaut

- **Délai** : 2 jours avant (configurable globalement)
- **Options globales** dans paramètres :
 - 1 jour avant
 - 2 jours avant (défaut)
 - 3 jours avant
 - Le matin même
 - Désactivé

2.2 Rappels personnalisés par tâche

- Override du paramètre global
- Choix de l'heure (8h, 12h, 18h, 20h, personnalisé)
- Rappels multiples possibles
- Rappel récurrent quotidien jusqu'à complétion

2.3 Types de notifications (via FCM + local)

Notification simple

dart

```
"📌 Rappel : [Titre tâche]
Dans 2 jours - [Date]"
```

Notification urgente (haute priorité)

dart

```
"🔴 URGENT : [Titre tâche]
À faire demain !"
```

Résumé quotidien (matin 8h)

dart

```
"☀️ Bonjour !
Vous avez 3 tâches cette semaine"
```

Résumé de fin de semaine

dart

```
"🎉 Semaine terminée !
8/10 tâches complétées - Bien joué !"
```

2.4 Gestion des rappels depuis notification

Actions de notification (Android/iOS) :

- Marquer comme fait

- Snooze (1h, 3h, demain)
- Voir détails (ouvre l'app)
- Ignorer

Gestion dans l'app :

- Liste des rappels programmés
 - Édition/suppression
 - Historique des rappels envoyés
-

3. ORGANISATION ET CATÉGORIES

3.1 Catégories pré définies (pré-remplies)

dart

```
List<Category> defaultCategories = [
    Category(name: 'Travail', color: 0xFF2196F3, icon: 'work'),
    Category(name: 'Personnel', color: 0xFF9C27B0, icon: 'person'),
    Category(name: 'Santé', color: 0xFFFF5722, icon: 'fitness'),
    Category(name: 'Famille', color: 0xFF4CAF50, icon: 'family'),
    Category(name: 'Études', color: 0xFFFF9800, icon: 'school'),
    Category(name: 'Courses', color: 0xFF795548, icon: 'shopping'),
    Category(name: 'Finance', color: 0xFF009688, icon: 'money'),
    Category(name: 'Loisirs', color: 0FFE91E63, icon: 'sports'),
];
```

Fonctionnalités :

- Création de catégories personnalisées (max 12 en gratuit)
- Picker de couleur Material
- Bibliothèque d'icônes (Material Icons)
- Réorganisation par drag & drop
- Archivage (pas suppression si tâches associées)

3.2 Filtrage et recherche

- Filtre par catégorie (Chips horizontales)
- Filtre par statut (Tabs ou SegmentedButton)

- Filtre par priorité
 - Recherche par titre (SearchDelegate)
 - Recherche vocale (speech_to_text)
-

4. VUES ET NAVIGATION

4.1 Structure de navigation (Bottom Navigation Bar)



4.2 Vue Semaine (Écran principal)

AppBar :

- Titre : "Ma semaine" + numéro semaine
- Actions : Recherche, Filtres, Menu
- Sélecteur de semaine (swipe gauche/droite)

Body - Liste de tâches :

- **Card par tâche** avec :
 - Checkbox à gauche (statut)
 - Pastille couleur catégorie
 - Titre en gras
 - Date/heure si définie
 - Badge priorité (si haute)
 - Icône récurrence (si applicable)
 - Durée estimée
 - Description tronquée (expandable)
- **Tri par** (menu dropdown) :
 - Date (chronologique) - défaut

- Date (chronologique) - default
- Priorité
- Catégorie
- Alphabétique
- **Affichage :**
 - Compact (petites cards)
 - Détailé (grandes cards avec description)
 - Toggle dans AppBar

Actions sur card :

- Tap : Ouvre détails
- Swipe droite : Marquer terminé ✓
- Swipe gauche : Supprimer 🗑
- Long press : Menu contextuel

États visuels :

- Tâche en retard : Card rouge pâle
- Tâche aujourd'hui : Card accentuée
- Tâche terminée : Strikethrough + opacité 50%

4.3 Vue Calendrier hebdomadaire

Affichage :

- 7 colonnes (Lun-Dim)
- Tâches affichées par jour (pastilles couleur)
- Indicateur charge de travail (barre sous chaque jour)
- Tap sur jour : Filtre les tâches de ce jour

Interactions :

- Drag & drop entre jours
- Pinch to zoom (vue mois future)
- Swipe pour changer de semaine

4.4 Vue Aujourd'hui (Optionnel MVP)

Focus tâches du jour :

- Liste filtrée automatiquement
- Ordre suggéré (priorité + durée)
- Mode "Focus" : Une tâche à la fois, plein écran
- Timer Pomodoro intégré (version future)

4.5 FAB (Floating Action Button)

Position : Centre du Bottom Nav Bar, surélevé **Action** : Ouvre bottom sheet création rapide **Animation** : Hero transition vers écran détail

5. RÉCURRENCE ET TÂCHES RÉPÉTITIVES

5.1 Types de récurrence

Fréquences :

- Quotidien (tous les jours)
- Hebdomadaire (chaque semaine, même jour)
- Bi-hebdomadaire (toutes les 2 semaines)
- Mensuel (même date chaque mois)
- Personnalisé (ex: Lun+Mer+Ven)

Configuration UI :

- Bottom sheet avec options
- Date de fin (optionnelle)
- Aperçu des prochaines occurrences

5.2 Gestion des récurrences

Lors de modification :

- Dialog : "Modifier uniquement cette occurrence OU toutes ?"
- Copie locale si modification unique
- Update template si toutes

Templates suggérés :

- "Sport - Tous les mardis et jeudis"
- "Courses - Tous les samedis"
- "Réunion hebdo - Tous les lundis 10h"

Badge visuel :

- Icône  sur la card
 - Couleur spéciale
-

6. GAMIFICATION ET MOTIVATION

6.1 Système de streaks 🔥

Affichage :

- Widget en haut de la vue semaine
- "🔥 12 semaines de suite !"
- Animation flamme

Paliers :

- 4 semaines : Badge Bronze
- 8 semaines : Badge Argent
- 12 semaines : Badge Or
- 26 semaines : Badge Platine
- 52 semaines : Badge Diamant

6.2 Statistiques de compléction

Card statistiques (vue semaine) :

Cette semaine : 75% (6/8 tâches)

Moyenne 4 semaines : 82%

Meilleure semaine : 100% 

Graphique (écran statistiques) :

- Line chart (4 dernières semaines)
- Bar chart (par catégorie)
- Pie chart (répartition temps)

6.3 Insights hebdomadaires

Messages intelligents (cards dans stats) :

- "👉 Vous complétez 85% de vos tâches !"
- "💼 Votre catégorie la plus active : Travail"
- "☀️ Vous êtes plus productif en début de semaine"
- "⚡ Tâches courtes = +20% complétion"

6.4 Récompenses visuelles

Animations :

- Confettis quand 100% semaine
- Lottie animations pour badges
- Vibration haptique sur succès
- Celebrations Material (hero animations)

Écran dédié "Succès" :

- Liste des badges obtenus
- Progression vers prochains badges
- Partage sur réseaux sociaux (futur)

7. SUGGESTIONS INTELLIGENTES (IA légère)

7.1 Détection de surcharge

Algorithme simple :

dart

```
if (totalHeuresEstimees > 20) {
    showDialog(
        title: "Semaine chargée détectée",
        content: "Vous avez planifié 24h de tâches. Voulez-vous en reporter ?",
    );
}
```

```

actions: [
    "Voir suggestions",
    "Garder tel quel"
]
);
}

```

Suggestions :

- Tâches basse priorité listées
- Bouton "Reporter à la semaine prochaine"
- Réorganisation automatique proposée

7.2 Équilibre vie pro/perso

Analyse hebdomadaire :

```

dart

if (tauxCategorieTravail > 80%) {
    showBottomSheet(
        "⚠️ 90% de vos tâches sont pro.  

        Pensez à ajouter des activités personnelles !"
    );
}

```

Suggestions catégories manquantes :

- "Aucune activité Santé cette semaine"
- "Pas de temps famille prévu"

7.3 Patterns et habitudes

Analyse locale :

```

dart

// Détection tâches souvent reportées
if (tache.reportCount > 3) {
    suggest("Cette tâche est souvent reportée.  

        La diviser en sous-tâches ?");
}

// Détection meilleurs moments

```

```
if (completionRateMorning > completionRateAfternoon) {  
    suggest("Vous complétez mieux les tâches le matin.  
    Planifier les tâches importantes en début de journée ?");  
}
```

7.4 Auto-complétion intelligente

Lors de la saisie :

- Suggestions basées sur historique
- "Vous aviez 'Dentiste' le mois dernier"
- Auto-remplissage catégorie/durée/priorité
- Recherche floue (fuzzy search)

8. PARAMÈTRES ET PERSONNALISATION

8.1 Écran Paramètres (ListView)

Notifications

- Toggle général
- Heures "Ne pas déranger" (TimeRangePicker)
- Fréquence rappels (Dropdown)
- Son et vibration (Toggle + SoundPicker)
- Mode silencieux weekend (Switch)

Planification 17

- Jour de planification (Dropdown)
- Heure de planification (TimePicker)
- Délai rappel par défaut (Slider 1-7 jours)
- Début de semaine (Lundi/Dimanche)

Apparence

- Thème (Radio: Clair/Sombre/Auto)
- Couleur accent (ColorPicker)
- Taille police (Slider)

- Animations (Toggle)

Langue & Région

- Langue (Dropdown: FR, EN)
- Format date (DD/MM/YYYY vs MM/DD/YYYY)
- Fuseau horaire

Compte

- Modifier profil
- Changer mot de passe
- Email de récupération
- Se déconnecter

Données

- Sauvegarde auto (Toggle)
- Export CSV/JSON
- Vider le cache local
- Supprimer compte (avec confirmation)

À propos

- Version app
- Conditions d'utilisation
- Politique de confidentialité
- Nous contacter
- Donner un avis (Store)

FONCTIONNALITÉS AVANCEES (Version 2.0+)

9. COLLABORATION ET PARTAGE

9.1 Tâches partagées

- Assigner à un membre famille/équipe

- Notifications de compléTION
- Commentaires sur tâches

9.2 Listes partagées

- "Courses" partagée
- "Projet X" en équipe
- Permissions (lecture/édition)

9.3 Vue famille

- Dashboard unifié
 - Calendrier familial
 - Synchronisation temps réel
-

10. INTÉGRATIONS MOBILES

10.1 Widgets natifs

iOS (WidgetKit)

- Widget petit : Prochaine tâche
- Widget moyen : Liste du jour
- Widget grand : Vue semaine

Android (App Widgets)

- Widget 2x2 : Tâches du jour
- Widget 4x2 : Vue semaine
- Widget 4x4 : Calendrier

10.2 Apple Watch / Wear OS

- Complications
- Glances des tâches du jour
- Quick actions (marquer terminé)
- Notifications synchronisées

10.3 Assistant

10.3 Assistants vocaux

iOS Shortcuts / Siri

"Hey Siri, ajoute 'acheter du pain' à ma semaine"

"Hey Siri, quelles sont mes tâches aujourd'hui ?"

Android App Actions / Google Assistant

"Ok Google, montre mes tâches"

"Ok Google, marque 'sport' comme fait"

10.4 Partage natif

- Share Sheet iOS/Android
 - Partager tâche/liste via WhatsApp, Email, etc.
 - Deep linking pour ouvrir tâche spécifique
-

11. FONCTIONNALITÉS PREMIUM (Monétisation)

11.1 Version gratuite (limites)

- 3 catégories personnalisées max
- 30 tâches actives simultanément
- Statistiques 4 dernières semaines
- 1 liste partagée
- Publicité légère (bannière en bas)

11.2 Version Premium (4,99€/mois ou 39,99€/an)

- Catégories illimitées
- Tâches illimitées
- Historique complet
- Listes partagées illimitées
- Thèmes premium + icônes
- Export avancé (PDF)

- ✨ Widgets premium
- ✨ Support prioritaire
- ✨ Suggestions IA avancées
- ✨ Backup cloud étendu
- ✨ Sans publicité

Implémentation : In-App Purchase (RevenueCat ou Purchases package)

11.3 Version Famille (9,99€/mois)

- Tout Premium
- 6 membres max
- Dashboard familial
- Statistiques consolidées

■ ARCHITECTURE BASE DE DONNÉES

Structure PostgreSQL (Supabase)

Table: users

sql

```
CREATE TABLE users (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    email TEXT UNIQUE NOT NULL,
    nom TEXT,
    avatar_url TEXT,
    preferences JSONB DEFAULT '{
        "jour_planification": "dimanche",
        "heure_planification": "20:00",
        "delai_rappel_defaut": 2,
        "theme": "auto",
        "langue": "fr",
        "debut_semaine": "lundi"
    }'::jsonb,
    premium BOOLEAN DEFAULT false,
    premium_expiry TIMESTAMP WITH TIME ZONE,
    fcm_token TEXT, -- Pour notifications push
    created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
```

```
updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);
```

-- Trigger pour updated_at

```
CREATE TRIGGER update_users_updated_at
BEFORE UPDATE ON users
FOR EACH ROW
EXECUTE FUNCTION update_updated_at_column();
```

Table: categories

sql

```
CREATE TABLE categories (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
    nom TEXT NOT NULL,
    couleur INTEGER NOT NULL, -- Color code (0xFF..),
    icone TEXT NOT NULL,
    ordre INTEGER DEFAULT 0,
    predefined BOOLEAN DEFAULT false,
    archived BOOLEAN DEFAULT false,
    created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);
```

```
CREATE INDEX idx_categories_user ON categories(user_id) WHERE NOT archived;
```

Table: taches

sql

```
CREATE TABLE taches (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
    titre TEXT NOT NULL CHECK (char_length(titre) BETWEEN 3 AND 100),
    description TEXT CHECK (char_length(description) <= 250),
    date DATE,
    heure TIME,
    priorite TEXT CHECK (priorite IN ('haute', 'moyenne', 'basse')) DEFAULT 'moyenne',
    categorie_id UUID REFERENCES categories(id) ON DELETE SET NULL,
    duree_estimee INTEGER, -- en minutes
    statut TEXT CHECK (statut IN ('a_faire', 'termine', 'reporte', 'archive')) DEFAULT 'a_faire',
    -- Récurrence
    recurrence_active BOOLEAN DEFAULT false,
    recurrence_type TEXT CHECK (recurrence_type IN ('quotidien', 'hebdomadaire', 'bi_hebdomadaire', 'mensuel', 'personnalisé')),
    recurrence_config JSONB, -- { "jours": ["lundi", "mercredi"], "fin": "2025-12-31" }
    recurrence_parent_id UUID REFERENCES taches(id) ON DELETE CASCADE,
    -- Métadonnées
```

```

created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
date_completion TIMESTAMP WITH TIME ZONE,
report_count INTEGER DEFAULT 0
);

-- Index pour performance
CREATE INDEX idx_taches_user_date ON taches(user_id, date) WHERE statut != 'archive';
CREATE INDEX idx_taches_statut ON taches(statut);
CREATE INDEX idx_taches_recurrence ON taches(recurrence_parent_id) WHERE recurrence_parent_id IS NOT NULL;

-- Trigger updated_at
CREATE TRIGGER update_taches_updated_at
BEFORE UPDATE ON taches
FOR EACH ROW
EXECUTE FUNCTION update_updated_at_column();

```

Table: rappels

```

sql

CREATE TABLE rappels (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    tache_id UUID NOT NULL REFERENCES taches(id) ON DELETE CASCADE,
    type TEXT CHECK (type IN ('avant', 'jour_meme', 'recurrent')) DEFAULT 'avant',
    delai_jours INTEGER,
    heure TIME,
    date_planifiee TIMESTAMP WITH TIME ZONE,
    envoye BOOLEAN DEFAULT false,
    date_envoi TIMESTAMP WITH TIME ZONE,
    created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);

CREATE INDEX idx_rappels_tache ON rappels(tache_id);
CREATE INDEX idx_rappels_pending ON rappels(date_planifiee, envoye) WHERE NOT envoye;

```

Table: statistiques_hebdomadaires

```

sql

```

```

CREATE TABLE statistiques_hebdomadaires (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
    annee INTEGER NOT NULL,
    numero_semaine INTEGER NOT NULL,
    date_debut DATE NOT NULL,
    date_fin DATE NOT NULL,

    taches_totales INTEGER DEFAULT 0,
    taches_completees INTEGER DEFAULT 0,
    taux_completion DECIMAL(5,2) GENERATED ALWAYS AS (
        CASE
            WHEN taches_totales > 0
            THEN (taches_completees::DECIMAL / taches_totales * 100)
            ELSE 0
        END
    ) STORED,
    temps_total_estime INTEGER, -- en minutes
    temps_par_categorie JSONB, -- { "travail": 480, "personnel": 120, ... }

    created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    UNIQUE(user_id, annee, numero_semaine)
);

CREATE INDEX idx_stats_user_semaine ON statistiques_hebdomadaires(user_id, annee, numero_semaine);

```

sql

```
CREATE TABLE badges (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
    type TEXT NOT NULL, -- 'streak', 'completion', 'tasks_count'
    niveau TEXT NOT NULL, -- 'bronze', 'argent', 'or', 'platine', 'diamant'
    obtenu_le TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    UNIQUE(user_id, type, niveau)
);

CREATE INDEX idx_badges_user ON badges(user_id);
```

Row Level Security (RLS)

sql

```
-- Activer RLS sur toutes les tables
ALTER TABLE users ENABLE ROW LEVEL SECURITY;
ALTER TABLE categories ENABLE ROW LEVEL SECURITY;
ALTER TABLE taches ENABLE ROW LEVEL SECURITY;
ALTER TABLE rappels ENABLE ROW LEVEL SECURITY;
ALTER TABLE statistiques_hebdomadiaries ENABLE ROW LEVEL SECURITY;
ALTER TABLE badges ENABLE ROW LEVEL SECURITY;

-- Politiques pour 'users'
CREATE POLICY "Users can view own profile"
    ON users FOR SELECT
        USING (auth.uid() = id);
```