

Cahier de charges technique — Application Domus (version mise à jour)

1. Objectif

Créer une application mobile **offline** pour gérer des maisons contenant des chambres occupées par des locataires, avec un suivi précis et manuel des paiements.

2. Entités principales

2.1 Maison

- Une maison peut avoir plusieurs chambres.
 - Champs à enregistrer :
 - `id`
 - `name` (Nom de la maison)
 - `address` (Adresse de la maison)
 - `created_at` (Date de création automatique)
-


2.2 Chambre

- Une chambre appartient à une seule maison.
- Une chambre est créée uniquement **lorsqu'un locataire est ajouté** (pas de chambres vides).
- Champs à enregistrer :
 - `id`
 - `house_id` (FK → maison)

- `name` (Nom de la chambre)
 - `type` (Type libre : studio, chambre simple, etc.)
-

2.3 Locataire

- Un locataire appartient à une maison et occupe une chambre.
- Champs à enregistrer :
 - `id`
 - `house_id` (FK → maison)
 - `room_id` (FK → chambre, unique par locataire)
 - `first_name`
 - `last_name`
 - `phone`
 - `email` (optionnel)
 - `entry_date` (date d'entrée)
 - `payment_frequency` (mensuelle, trimestrielle, semestrielle, annuelle)
 - `rent_amount` (montant du loyer du locataire)

 **Règle** : Lorsqu'un locataire est ajouté → il est automatiquement considéré comme “à jour”.

2.4 Paiement

- Les paiements sont **créés manuellement** via un formulaire.
- Champs à enregistrer :

- `id`
- `tenant_id` (FK → locataire)
- `month` (mois payé, ex. "2025-02")
- `amount` (montant payé)
- `paid_at` (date réelle de paiement)

⚠ **Ajout** : Le champ `month` permet de spécifier pour quel mois le paiement est effectué.

3. Fonctionnalités principales

3.1 Gestion des maisons

- Créer une maison avec ses caractéristiques (nom, adresse).
 - Voir la liste des maisons.
 - Cliquer sur une maison → voir les détails :
 - Infos générales.
 - Liste des locataires liés.
-

3.2 Gestion des locataires

- Ajouter un locataire à une maison (et créer sa chambre en même temps).
- Détails d'un locataire (modal) :
 - Infos personnelles.
 - Infos logement.
 - Infos paiement.
- Actions possibles :

- Modifier
 - Supprimer
 - Voir ses paiements et son statut (à jour / non à jour).
-

3.3 Gestion des paiements

- Création manuelle via un formulaire :
 - Sélectionner un locataire.
 - Auto-remplissage maison + chambre.
 - Montant payé.
 - Mois payé (ex. "Mars 2025").
 - Suivi des paiements :
 - **Paiements effectués** → liste avec détails (locataire, mois, montant).
 - **Paiements en retard** → générés automatiquement :
 - Si un mois attendu (selon fréquence) n'a pas de paiement enregistré → il apparaît en retard.
 - Le statut du locataire passe en "**non à jour**" tant qu'il existe au moins un mois impayé.
 - Lorsqu'un paiement est créé pour un mois impayé → ce mois disparaît des retards.
-

3.4 Dashboard

- Nombre de maisons.
- Nombre total de locataires.
- Nombre de paiements en retard.

- Liste des maisons avec accès rapide.
 - Boutons rapides : ajouter une maison / ajouter un locataire.
-

3.5 Recherche

- Page locataires avec barre de recherche (nom, prénom).
 - Badge de statut affiché (✓ à jour / ✗ non à jour).
-

3.6 Paramètres

- Bouton “**Nettoyer toutes les données**” (reset complet).
-

3.7 Graphiques (optionnel)

- Répartition des locataires par maison.
 - État des paiements (effectués vs en retard).
 - Historique de paiements par mois.
-

4. Base de données SQLite (mise à jour)

```
CREATE TABLE houses (  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  name TEXT NOT NULL,  
  address TEXT NOT NULL,  
  created_at TEXT DEFAULT (datetime('now'))  
);
```

```
CREATE TABLE rooms (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    house_id INTEGER NOT NULL,  
    name TEXT NOT NULL,  
    type TEXT,  
    FOREIGN KEY (house_id) REFERENCES houses(id) ON DELETE CASCADE  
);
```

```
CREATE TABLE tenants (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    house_id INTEGER NOT NULL,  
    room_id INTEGER NOT NULL UNIQUE,  
    first_name TEXT NOT NULL,  
    last_name TEXT NOT NULL,  
    phone TEXT NOT NULL,  
    email TEXT,  
    entry_date TEXT NOT NULL,  
    payment_frequency TEXT NOT NULL,  
    rent_amount REAL NOT NULL,  
    FOREIGN KEY (house_id) REFERENCES houses(id) ON DELETE CASCADE,  
    FOREIGN KEY (room_id) REFERENCES rooms(id) ON DELETE CASCADE  
);
```

```
CREATE TABLE payments (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    tenant_id INTEGER NOT NULL,
```

```
month TEXT NOT NULL, -- format YYYY-MM

amount REAL NOT NULL,

paid_at TEXT NOT NULL DEFAULT (datetime('now')),

FOREIGN KEY (tenant_id) REFERENCES tenants(id) ON DELETE CASCADE

);
```

5. Architecture de l'application

src/

```
|— components/    # Composants réutilisables (cards, modals, boutons)
|— screens/       # Pages principales (Dashboard, Maisons, Locataires, Paiements)
|— db/           # Gestion SQLite (init, migrations, CRUD)
|— hooks/        # Hooks custom (useHouses, useTenants, usePayments)
|— context/      # Context global (gestion état et données)
|— utils/        # Fonctions utilitaires (dates, formatage)
└— navigation/   # Navigation (Stack + Tabs)
```

6. Stack technique

- **React Native** (Expo ou bare RN)
- **SQLite** : `expo-sqlite` ou `react-native-sqlite-storage`
- **Navigation** : `@react-navigation/native` (Stack + Tabs)
- **État global** : Context API ou Zustand
- **Dates** : `date-fns`

- **Formulaires** : `react-hook-form`
 - **Charts (optionnel)** : `react-native-chart-kit` ou `victory-native`
 - **UI / Icônes** : `react-native-vector-icons`, `react-native-paper`
-

7. Logique des paiements

7.1 Enregistrement initial

- Lorsqu'un locataire est ajouté → son statut =  **à jour**.

7.2 Détection automatique des retards

- Chaque locataire a une fréquence (mensuelle, etc.).
- Exemple :
 - Locataire avec fréquence **mensuelle** → doit payer chaque mois.
 - Si au mois M aucun paiement n'a été créé → le mois M est marqué comme **impayé** et ajouté à la liste des retards.

7.3 Création d'un paiement

- Paiement ajouté manuellement :
 - `locataire = Jean`
 - `montant = 50.000`
 - `mois payé = Février 2025`
- Résultat :
 - Paiement ajouté dans la liste validée.
 - Février disparaît de la liste des retards.

- Si d'autres mois sont encore impayés → statut reste **X** non à jour.