**📄 Product Requirements Document (PRD)**

**Project Name: My Records**

**Platform: Android (Flutter, Dart, SQLite)**

**Version: 1.0**

**Default Mode: Dark Mode**

**1. Project Overview**

“My Records” is a personal folder-based record management mobile application where users can create folders and store categorized information.

All png are kept in D:\Anant\VSCodeProjects\My\_Records

Additionally, the existing project D:\Anant\VSCodeProjects\Flutter\_Health\_Records\health\_records\_app will be analyzed only for reference (no modification) to reuse any reusable widgets, core structure, theme logic, database handling, android version or design considerations. You can also refer the project information via memory mcp.

**2. Objectives**

✔ Allow users to create, view, edit, duplicate, and delete folders.  
✔ Show total folder count under the section “Folder Lists”.  
✔ Provide Dark/Light mode toggle and Help section.  
✔ Use local SQLite database for storing folder records.  
✔ Default theme should be Dark Mode.  
✔ Use the provided image file ("") as App Icon (to be replaced with actual filename when provided).

**3. Core Features & Requirements**

**3.1 Home Screen (Refer to 1.png layout),** All png’s are kept in D:\Anant\VSCodeProjects\My\_Records

| **Component** | **Description** |
| --- | --- |
| AppBar | Title aligned to left: **“Folder Lists”** |
| Folder Count | Below title, display “Total: X Folders” dynamically from database |
| Theme Toggle | Light/Dark mode icon (Right side of AppBar) |
| Help Icon | On top right side (next to theme toggle), opens basic instructions dialog/page |
| Folder List View | List of folders in card/list format, each with folder name & icons for edit, duplicate, delete |
| Floating Action Button (FAB) | Positioned at bottom-right (“+” icon) to add new folder |

**3.2 Create New Folder (Refer to 2.png)**

When user clicks “+” button:

* Show Pop-up Window / Dialog titled **“Create New Folder”**
* Fields Required:
  + Folder Name (TextField)
  + Description (optional)
  + Creation Date auto-filled (but hidden or visible as per 2.png)
* Buttons:
  + **Save** – inserts into SQLite
  + **Cancel / Close**

**3.3 Folder Actions (after creation)**

Each folder row/card should display the following actions:

| **Icon** | **Functionality** |
| --- | --- |
| ✏ Edit | Opens pop-up to update folder name/description |
| 📄 Duplicate | Creates a copy of the folder with same details and “(Copy)” appended |
| 🗑 Delete | Shows confirmation pop-up before deleting from database |

**3.4 Dark/Light Mode**

* Default mode = **Dark Mode**
* Toggle icon in the AppBar for switching theme
* Theme settings saved using SharedPreferences or SQLite table settings
* Inherit reusable theme logic from **health\_records\_app**: check for themes, icons, helper functions.

**3.5 Help Section**

* Triggered via **Help Icon**
* Display pop-up or full screen with:
  + Purpose of the app
  + Instructions on creating folders and using app features
  + Usage of duplicate/edit/delete options
  + Theme switching info

**4. Database Design (SQLite)**

**4.1 Table: folders**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| id | INTEGER PRIMARY KEY AUTOINCREMENT |  |
| folder\_name | TEXT NOT NULL |  |
| description | TEXT |  |
| created\_at | TEXT (DateTime ISO format) |  |

**5. Reference Project Analysis (No Code Changes Made)**

Path: D:\Anant\VSCodeProjects\Flutter\_Health\_Records\health\_records\_app

**Helpful reusable components identified:**  
✅ SQLite database helper structure  
✅ SharedPreferences theme management  
✅ Backup/restore logic – useful for future enhancements  
✅ UI theming & dark mode handling  
✅ Reusable widgets for dialogs and form validation

**Potential issues to avoid in My Records app:**

* Complex navigation & too many dependencies (keep structure modular)
* Avoid mixing business logic with UI widgets (use provider or simple MVC)
* Ensure folder duplication creates unique IDs and avoids primary key conflicts
* Optimize list rendering with ListView.builder to prevent UI lag

**6. App Icon**

* The provided icon image ("D:\Anant\VSCodeProjects\My\_Records\records.png") will be used as app icon
* To be processed into Android mipmap format using flutter\_launcher\_icons package

**7. Tech Stack**

| **Component** | **Technology** |
| --- | --- |
| Language | Flutter (Dart) |
| Database | SQLite using sqflite package |
| Theme | Light/Dark with Provider or ChangeNotifier |
| State Mgmt | Provider / GetX (simple approach preferred) |
| Local Settings | Shared Preferences |

**Incorporate the above requirements/fixes.**

* Use Sequential thinking and other MCP as required to achieve the objective.
* The UI should be stunning and easy to use.
* Improvise as required, Fill the missing requirements or gaps.
* Ensure this change/enhancement does not break any existing code or functionality.

**Check the Android build settings in the project**

**D:\Anant\VSCodeProjects\Flutter\_Health\_Records\health\_records\_app**

**and apply the same settings to the current project to generate a debug APK.**

**Finally install the apk to my mobile.**

**Incorporate the above requirements/fixes.**

* Use Sequential thinking and other MCP as required to achieve the objective.
* The UI should be stunning and easy to use.
* Improvise as required, Fill the missing requirements or gaps.
* Ensure this change/enhancement does not break any existing code or functionality.
* **The handle icon shown in 1.png should allow users to move items up and down. Check how this is implemented in the project at  
  D:\Anant\VSCodeProjects\Flutter\_Health\_Records\health\_records\_app  
  and apply the same functionality in the current project.**
* **Add fingerprint authentication to the app, similar to the implementation in the health\_records\_app project.**
* **When saving a folder name or optional description, automatically capitalize the first letter.**
* **The app name displayed below the app icon should be “My Records”, but it is currently showing as “my\_records”.**
* **Fix the “overflowed” error shown in 3.png in the requirements folder.**
* **Not working - Add fingerprint authentication to the app, similar to the implementation in the health\_records\_app project.**
* **On the Welcome Back screen (see 6.png in requirements folder), entering the PIN or using fingerprint authentication does not log in successfully.**
* **When I click on a folder item, it should open to Record screen which should have a + button on bottom right**
* **When + button is clicked it should open a popup screen which 2 fields Field Name, Field Value, the user will enter data in them and either cancel or create**
* **Add a + icon next to the Field Value input, allowing the user to add additional Field Value rows. The user should be able to add as many Field Values as needed.**
* **Also the date and time format on record should be like 17 Oct 2025, 4:40 PM**
* **The additional field(s) value added should show in same record not as different record.**
* **When displaying additional Field Values, they are currently shown with bullet points. Remove the dot (bullet point) before each value.**
* **Sort the records alphabetically by Field Name when displaying them.**
* Add a **Help screen** for the Record screen, similar to the one used for the Folder screen.
*  Add a **Search bar** on the Folder screen, below the folder count (refer 8.png using MCP).
*  Include a **filter dropdown** to search either in **All Folders** or within a **specific folder**.
  +  The dropdown list should display:
  + **“All Folders”** as the first and default option
  + Followed by all folder names, sorted alphabetically.
*  The search function should look for data **inside each folder’s records**, not by folder names.
*  If the same value is found in multiple records across different folders, show **all matching results along with their folder names**.
* The search result should show complete record of matching search phrase and folder name below it. No need to show search keyword, matched xxx in search record.
* When a search result is clicked, or when **“View Folder”** is selected, it should navigate to the respective folder and show that **specific record at its exact position** within the record screen for that folder.

Add a **Settings icon** on the top right of the Folder screen. When clicked, it should display two options: **Change PIN** and **Backup & Restore**.

* Implement both features similar to the **health\_records\_app** project located at D:\Anant\VSCodeProjects\Flutter\_Health\_Records\health\_records\_app, with the only difference being that the PIN here should be **6 digits**.
* The backup file should be saved in the **my\_records** folder under **Android/Downloads**, with the filename format:  
  my\_records28Oct25\_0955AM
* All other functionalities—such as retaining the **last three backups**, providing **multiple share options**, and other related features—should work exactly as implemented in the health app.
* Add a **Help screen** for both features (**Change PIN** and **Backup & Restore**), similar to the one in the Health app.
* Finally, **verify that all features, functionalities** and **screen layout** match those in the Health app, **except for the specific exceptions mentioned** earlier.
* In the **Change PIN** screen, the text box should automatically gain focus so the user can start typing immediately without needing to tap inside it.
* When the user selects **Backup and Restore**, the screen should load and function exactly as it does in the **Health app**. Refer to **1.png** and **2.png** in the requirements folder for visual reference (use **Desktop Commander MCP** or any other MCP to view the PNG files).  
  The corresponding implementation can be found in the file:  
  D:\Anant\VSCodeProjects\Flutter\_Health\_Records\health\_records\_app\lib\presentation\screens\backup\_restore\_screen.dart
* Also, **verify all functionalities** in the **Backup and Restore** screen for **buttons, Display values, dropdowns, and icons, etc.** — to ensure they are implemented exactly as in the **Health app, except for backup file name which should be** the filename format: my\_records28Oct25\_0955AM.
* Fix all errors listed in the **Problems tab**, including minor ones, and then **build and install the debug APK** on my mobile device to test all features.
* The backup should only be created if there are any folders, and its should create a backup file consisting of all folders and records within it and same will be used to restore at the later date.
* Check 4.png using **Desktop Commander MCP** or any other MCP to view the PNG files and fix the issue.
* When creating a **backup file**, first check if the folder **my\_records** exists in **Android/Downloads**. If it does not exist, **create the folder**, and then save the backup file inside it.
* Review how **folder permissions** are handled in the **Health app** to prevent errors such as:  
  Backup failed: Exception: Failed to create backup: PathAccessException: Cannot open file, path = '/storage/emulated/0/Android/Downloads/my\_records/my\_records28Oct25\_1100AM.json' (OS Error: Operation not permitted, errno = 1)