
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) |

| Component | Description |
|------------------------------|--|
| 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.