RMinder – Offline Budget App Design Document

1. Overview

RMinder is a personal finance management tool that allows users to create budgets, record expenses, and track spending trends – **completely offline**. All data is stored locally on the device, with no internet access or personal data collection required.

2. Goals

- Privacy-first: No internet connection required.
- Simplicity: Minimal setup, intuitive user interface.
- Portability: Allow users to transfer data securely between devices using QR codes.
- Reliability: All features work offline.

3. Key Features

3.1 Budget Management

- Create, edit, and delete budget categories (e.g., Rent, Food, Transport).
- Set monthly budgets for each category.
- View remaining balance per category in real time.

3.2 Expense Tracking

- Add transactions with category, amount, date, and optional note.
- View transaction history grouped by category or date.
- Edit or delete transactions as needed.

3.3 Reports & Insights

- Monthly summary: Total spent vs. budget.
- Category breakdown pie chart.
- Simple trend visualization for spending over time.

3.4 Data Storage

- Use **SQLite** for local storage.
- All data stays on-device.
- No external servers or tracking.

3.5 Data Transfer (QR Code Method)

- **Export:** Serialize user data (budgets, transactions) to JSON → compress (Gzip) → optionally encrypt (AES) → generate QR codes.
- **Import:** Scan QR codes sequentially on the new device → reassemble data → decrypt/decompress → import into SQLite.
- Multi-QR support with progress indicator for large datasets.
- Option to set a passphrase for additional security.

3.6 Optional Fallback Transfer Method

- Export/import JSON file manually (via USB, SD card, or file sharing apps).
- Useful for very large datasets where QR codes might be too slow.

4. Technical Design

- **Framework:** Flutter (cross-platform support for Android & iOS)
- **Local Database:** SQLite (via sqflite package)
- **State Management:** Provider or Riverpod for predictable state handling
- **QR Code Generation:** qr_flutter
- **QR Code Scanning:** qr_code_scanner or mobile_scanner
- **Compression:** archive package (for Gzip)
- **Encryption (Optional):** encrypt package (AES)

5. Security

- No external servers used.
- All data stored locally, encrypted if user chooses.
- Optional passphrase required to import/export data.

6. Milestones & Timeline (1 Month Goal)

Week 1:

- Set up Flutter project structure
- Implement local storage and data models (SQLite)
- Build UI for adding/editing budgets & transactions

Week 2:

- Implement reporting views (summary, charts)
- Create QR export & import logic (basic JSON transfer)

Week 3:

- Add compression/encryption support
- Implement multi-QR scanning flow
- Polish UI and user experience

Week 4:

- Testing on multiple devices
- Bug fixing & UI refinements
- Prepare app for publishing (Google Play & local sideloading)

7. Future Enhancements

- Support for recurring transactions
- Multi-currency support
- Simple predictive insights based on spending patterns
- Optional cloud backup (for users who want online sync)

This design ensures **RMinder** remains private, offline, and fully functional, while giving users a convenient and secure way to migrate data between devices using QR codes.