Construction Expense & Material Tracker App

# 1. Overview

This is a mobile application designed to mimic the simplicity of Excel but improve the organization of daily construction expense data. It allows easy entry of project expenses and automatically groups and summarizes information by project.  
  
The goal is to have one sheet where all entries are added, and the app automatically:  
- Detects new projects  
- Groups and summarizes entries by project  
- Shows total quantities (e.g., Bricks: 50,000)  
- Displays financial totals (credit, debit, balance)

# 2. Core Features

## 2.1 Add Entry Screen

Fields:  
- Project Name (text)  
- Detail (text, e.g., "Bricks", "Cement")  
- Quantity (number)  
- Credit (number)  
- Debit (number)  
- “Save” button adds the entry to the main data store

## 2.2 Main Sheet (Entry List)

Displays all entries added (like rows in Excel), sorted by most recent first.   
Fields shown: Project, Detail, Quantity, Credit, Debit, Date

## 2.3 Dashboard (Project Cards)

Each unique project has a card showing:  
- Project Name  
- Total Credit  
- Total Debit  
- Balance = Credit - Debit  
- Total Quantities by Material Type (e.g., Bricks: 50,000, Cement: 300 bags)  
Cards are created automatically when a new project is detected.

## 2.4 Project Detail View

Tap a card to view all entries for that project.  
Shows: Table/List of all related entries  
Optional (future): filter by date range or material.

# 3. Data Model

Entry Object Example:  
{  
 "id": 1,  
 "project": "Sunrise Apartments",  
 "detail": "Bricks",  
 "quantity": 5000,  
 "credit": 0,  
 "debit": 20000,  
 "date": "2025-06-14"  
}

# 4. Calculations

Per project:  
- Total Credit: sum of all credit values  
- Total Debit: sum of all debit values  
- Balance: Credit - Debit  
- Quantity Per Material: group entries by detail and sum quantity

# 5. Tech Stack

Framework: React Native with Expo  
Language: JavaScript  
Storage: AsyncStorage (v1), upgrade to SQLite/MMKV later  
UI Components: FlatList, Modal, TextInput, Card  
Navigation: React Navigation  
Future Upgrade: TypeScript

# 6. Folder Structure

/App.js  
/src  
 /screens  
 - DashboardScreen.js  
 - AddEntryScreen.js  
 - ProjectDetailScreen.js  
 /components  
 - ProjectCard.js  
 - EntryRow.js  
 /storage  
 - storage.js  
 /utils  
 - dataHelpers.js

# 7. User Flow

1. User opens app → Sees project cards dashboard  
2. Taps + button → Adds an expense entry with project, detail, quantity, credit, and debit  
3. If project is new → A new card appears on dashboard  
4. Taps a card → Views entries for that project  
5. Data is saved locally (offline-first)  
6. Totals and material usage automatically update on the dashboard

# 8. Goals & Future Scope

Current Scope:  
- Fully offline-capable app  
- Clean UI  
- Easy entry + instant grouping  
- Works like a personal ledger  
  
Future Features:  
- Export data to Excel/PDF  
- Filter by date or material  
- Search functionality  
- Sync with cloud (Firebase or Supabase)  
- User authentication (if multi-user in future)

# 9. Stakeholders

Father: End user (construction manager)  
You: Developer, designer, project owner  
Recruiters (indirect): Will review this as portfolio project

# 10. Notes

- Data is stored in one unified sheet (just like Excel)  
- No need to manually create project groups  
- Entire app works offline  
- Focus is on speed, simplicity, and real-world utility