



## Parcel Manager – Enterprise Logistics Management System

*Cross-Platform Mobile Application / Flutter, Dart*

**Project Overview:** Designed and developed a comprehensive **parcel logistics management system** from the ground up, implementing a modular architecture with clean separation of concerns using the **GetX state management framework**.

### Architecture & Technical Implementation:

- **State Management:** Implemented reactive state management using **GetX** with dependency injection via custom bindings, supporting both local and global state with automatic lifecycle management
- **Database Layer:** Designed and implemented a robust **SQLite persistence layer** with:
  - Schema versioning and migration support (`_onUpgrade`)
  - Hierarchical data modeling for chart of accounts with parent-child relationships
  - Complex JOIN queries for reporting and transaction retrieval

- Automatic daily backup with JSON serialization and restore functionality
- **Business Logic:** Built comprehensive controllers for consignment lifecycle management including financial transaction processing with **double-entry bookkeeping** implementation
- **Platform Integration:** Developed native **Android platform channel integration** for dual-SIM SMS support with callback handling for delivery status tracking

### Key Technical Features:

| Feature                    | Implementation  |
|----------------------------|---|
| <b>PDF Generation</b>      | Custom PDF engine using pdf package with RTL Arabic text rendering, multi-page statements, vouchers, and reports with embedded fonts (Cairo, Amiri) |
| <b>SMS Service</b>         | Native Kotlin bridge via MethodChannel for dual-SIM support, background message handling, and delivery status callbacks                             |
| <b>Financial Engine</b>    | Double-entry bookkeeping with journal entries, receipt/payment vouchers, opening balances, and hierarchical chart of accounts                       |
| <b>Data Modeling</b>       | 8+ Dart models with factory constructors, JSON serialization, async hydration, and computed properties  |
| <b>Contact Integration</b> | Native contact picker integration with phone number extraction and validation   |
| <b>Image Processing</b>    | Image cropping and photo capture for delivery confirmation with ID verification   |

### Technical Stack:

Framework: Flutter 3.x / Dart 3.9

State: GetX (Reactive Controllers, Bindings, Routing)

Database: SQLite (sqflite) with 1,400+ lines of service code

PDF: pdf + printing packages with custom Arabic text reshaping

SMS: Platform Channels (Kotlin) + another\_telephony

UI: Responsive design with flutter\_screenutil, custom widgets

Localization: RTL support with bidi + arabic\_tools packages

#### **Code Metrics:**

- **138+ Dart files** across controllers, models, views, and services
- **1,384 lines** in database service layer alone
- **898 lines** in PDF report generation service
- **18 controllers** managing distinct business domains
- **Modular routing** with 15+ screens and navigation management

#### **Key Engineering Decisions:**

- Implemented **repository pattern** for data access abstraction
- Used **factory constructors** with async initialization for model hydration
- Created **reusable widget library** for consistent UI components
- Applied **SOLID principles** with single-responsibility controllers
- Designed **defensive coding** with comprehensive null safety