GEO Blocks

B2B Sales & Order Management Platform - Flutter Cross-Platform Application

Executive Summary

GEO Blocks is an enterprise-grade Flutter application designed for B2B sales management, distributor coordination, and real-time order placement. The platform enables salesmen and distributors to manage product catalogs, view distributor information, place orders efficiently, and receive real-time notifications. Built with BLoC state management pattern, RESTful API integration, Firebase backend infrastructure, and comprehensive notification systems, the project demonstrates expertise in enterprise mobile development, scalable architecture, API design, and business-critical applications.

Project Metrics & Scope

Codebase Size: 28+ Dart files | 249KB+ code library

Core Modules: 5 main BLoC modules with clean architecture

Platform Support: iOS, Android, Web, Windows (multi-platform)

Current Version: 1.0.0 (Production-ready)

State Management: BLoC pattern with reactive streams

Backend Integration: Firebase + Custom REST APIs

Asset Library: 421KB+ images + 297KB+ custom fonts

Key Screens: Dashboard, Login, Place Order, Product List

Core Features & Functionality

- **User Authentication & Login**
    - Secure user login system with Firebase
    - Session management and token handling
    - User role management (salesman, distributor)
    - Password reset and account recovery
    - Persistent user authentication
    - Secure credential storage with SharedPreferences
- **Dashboard & Analytics**
    - Real-time sales dashboard with key metrics
    - Order summary and status tracking

- Sales performance analytics
- Distributor activity overview
- Revenue and inventory tracking
- Quick access to key functions

- **Distributor Management**

  - View comprehensive distributor list with details
  - Distributor information and contact details
  - Location-based distributor discovery
  - Distributor performance metrics
  - Direct communication with distributors
  - Order history by distributor

- **Product Catalog Management**

  - Browse complete product catalog
  - Product details and specifications
  - Pricing and inventory information
  - Product categorization and filtering
  - Real-time product availability
  - Search and filtering capabilities

- **Order Placement System**

  - Intuitive order creation interface
  - Select products and set quantities
  - Calculate total order value
  - Apply discounts and promotions
  - Multiple payment options
  - Order confirmation and tracking

- **Order Management**

  - View order history and status
  - Track order fulfillment progress
  - Order cancellation and modifications
  - Order-to-delivery timeline
  - Invoice generation
  - Order export and printing

- **Push Notifications & Alerts**

  - Firebase Cloud Messaging integration
  - Real-time order status updates
  - Distributor communication alerts
  - Promotion and discount notifications
  - System alerts and reminders
  - Local notification rendering with high priority

- **Real-Time Synchronization**

- Live inventory updates
- Real-time order status changes
- Instant distributor availability
- Push-based data refresh
- Automatic sync on app launch
- Offline cache with sync capability

Technical Stack & Architecture

Frontend Framework: Flutter (Dart 2.16.1+) - 100% code share across platforms

State Management: BLoC Pattern with Reactive Streams

Backend Services: Firebase (Authentication, Messaging, Cloud Infrastructure)

API Communication: HTTP (v0.13.4) - RESTful API integration

Reactive Programming: RxDart (v0.27.3) - Observable streams and operators

Local Storage: Shared Preferences (v2.0.15) - User data persistence

Notifications: Firebase Cloud Messaging + Flutter Local Notifications

Permission Management: Permission Handler (v11.3.1) - Runtime permissions

UI Utilities: Toastification (v2.3.0) - Toast notifications

Text Effects: Marquee (v2.3.0) + Text Marquee (v0.0.2) - Scrolling text

Icons: FlutterIcon (v2.0.0) - Icon library

Internationalization: Intl (v0.17.0) - Multi-language support

Icon Generation: Flutter Launcher Icons (v0.9.0) - App icon generation

Architecture & Design Patterns

- BLoC Pattern: Complete separation of business logic and presentation
- Repository Pattern: Abstraction layer for data sources (API and local storage)
- Observer Pattern: Reactive streams for real-time data updates
- Model-View-Presenter: Clear separation of concerns
- Dependency Injection: Service locator pattern with GetIt
- Factory Pattern: API response handling and model creation
- Singleton Pattern: Shared services and managers
- Stream Architecture: Event-driven updates with RxDart

Key Technical Accomplishments

- Designed and implemented clean BLoC architecture for scalability
- Integrated multiple REST APIs for product, order, and distributor data

- Implemented Firebase Cloud Messaging for real-time notifications
- Built reactive UI with RxDart streams for efficient updates
- Created responsive design supporting multiple screen sizes
- Implemented comprehensive error handling and user feedback
- Built local caching layer with offline capability
- Integrated permission handling for modern Android/iOS requirements
- Designed intuitive order placement workflow
- Implemented secure user authentication and session management
- Built efficient API client with request/response handling
- Created notification system with background message processing

BLoC State Management Modules

Login BLoC: User authentication, session management, credential validation

Distributor List BLoC: Fetch, cache, and manage distributor data

Product List BLoC: Product catalog management, filtering, searching

Place Order BLoC: Order creation, validation, submission, confirmation

Salesman BLoC: Salesman profile and permissions management

API Layer & Integration

- Login API: User authentication and token generation
- Distributor List API: Fetch all available distributors with details
- Product List API: Retrieve complete product catalog
- Place Order API: Submit orders with validation
- Error Handling: Graceful error management with user feedback
- Request/Response Interceptors: Centralized request handling
- Timeout Management: Configurable timeouts for API calls

Screen Architecture & User Flow

Login Screen: User authentication with email/password validation

Dashboard: Main home screen with analytics and quick access (85KB)

Product List Screen: Browse and search products with filtering

Place Order Screen: Comprehensive order creation interface (68KB)

Distributor List: View all distributors with contact information

Notifications: View and manage notification history

Firebase Integration & Backend Services

- Authentication: Secure user login and account management
- Cloud Messaging: Push notifications for order updates and alerts
- Real-Time Updates: Live data synchronization across devices
- Cloud Storage: Backup and archival of orders and documents
- Security Rules: Granular access control for user data
- Analytics: Track user behavior and app usage patterns

Data Models & Entities

User Model: User credentials, role, and authentication state

Product Model: Product details, pricing, and inventory information

Distributor Model: Distributor information and contact details

Order Model: Order items, quantities, total amount, and status

Order Item Model: Individual product in order with quantity and price

Development Practices & Tools

- Code Quality: Flutter linting with comprehensive analysis
- State Management: BLoC pattern for predictable state changes
- Error Handling: Comprehensive error handling throughout app
- UI Responsiveness: Adaptive layouts for all screen sizes
- Firebase Configuration: Multi-platform Firebase setup
- Notification System: Foreground and background message handling
- Performance: Optimized API calls and efficient state management
- Version Control: Git-based development workflow

Key Technologies & Skills Demonstrated

✓ Enterprise Mobile Development ✓ BLoC State Management

✓ RESTful API Integration ✓ Firebase Backend Services

✓ Reactive Programming (RxDart) ✓ Push Notifications

✓ User Authentication ✓ Cross-Platform Development

✓ Database Design ✓ UI/UX for Enterprise Apps

✓ Permission Management ✓ Real-Time Synchronization

Asset & Resource Management

- Image Assets: 421KB+ of high-quality graphics and UI elements
- Custom Fonts: 297KB+ including Ubuntu font family
- App Icons: Multi-resolution icons for Android and iOS
- Brand Assets: Logo and branding elements
- UI Graphics: Buttons, backgrounds, and decorative elements
- Responsive Images: Multi-resolution support for different devices

Multi-Platform Deployment

- Android: Full support with Material Design, optimized for Kotlin
- iOS: Complete support with Swift interop and native features
- Web: PWA support for browser-based access
- Windows: Desktop application for Windows systems

Project Structure & Organization

lib/src/view/ - 4 main screens (dashboard, login, placeorder, notifications)

lib/src/bloc/ - 5 BLoC modules for state management

lib/src/api/ - REST API integration layer

lib/src/model/ - Data models and entities

lib/src/constant/ - App constants and configurations

lib/utils/ - Utility functions and helpers

assets/images/ - UI graphics and brand assets (421KB)

assets/fonts/ - Custom font families (297KB)

android/ & ios/ - Platform-specific configurations

web/ & windows/ - Web and Windows support files

Notification System & Real-Time Updates

- Firebase Cloud Messaging: Server-to-device push notifications
- Foreground Notifications: User alerts while app is active
- Background Notifications: Message handling when app is closed
- Notification Permissions: Runtime permission handling
- Local Notifications: High-priority alerts with custom channels
- Rich Payloads: Detailed notification content with actions

Security & Best Practices

- Firebase Authentication: Secure token-based authentication
- Encrypted Credentials: Secure storage with SharedPreferences
- HTTPS Communication: Secure API communication
- Permission Management: User-controlled access
- Error Handling: No sensitive data in logs
- Session Management: Automatic logout on token expiry

Business Use Cases & Applications

- B2B Sales: Streamline distributor-to-customer sales processes
- Order Management: Efficient order placement and tracking
- Inventory Management: Real-time product availability
- Sales Analytics: Track sales performance and metrics
- Distributor Network: Manage multiple distribution channels
- Mobile-First Selling: Enable sales on-the-go

Future Enhancement Roadmap

- Advanced Analytics: Detailed sales reports and dashboards
- Mobile Payments: In-app payment processing
- Offline Mode: Full functionality without internet
- Multi-Currency Support: International business expansion
- Inventory Sync: Real-time warehouse synchronization
- Advanced Filtering: Complex search and filtering options
- Customer Portal: End-customer access and tracking
- API Versioning: Support for multiple API versions

Core Competencies Demonstrated

✓ Enterprise Mobile Development ✓ BLoC Architecture

✓ REST API Integration ✓ Firebase Services

✓ Reactive Programming ✓ Cross-Platform Development

✓ Database Design ✓ UI/UX for Business Apps

✓ Permission & Security ✓ Real-Time Updates

Technical Challenges Solved

- Complex State Management: BLoC pattern for clean architecture
- Real-Time Sync: Firebase integration for instant updates
- API Integration: Multiple endpoints with error handling
- Responsive Design: Supporting multiple screen sizes efficiently
- Background Processing: Firebase notification handling

- Data Persistence: Offline caching with sync capability