Project Architecture

Directory Structure

```
src/
├── app/
│   ├── account-management/       # Account operations
│   ├── admin-menu/               # Admin navigation
│   ├── finance-management/       # Financial operations
│   ├── home/                     # Dashboard/home page
│   ├── incoming-scan/            # Incoming package scanning
│   ├── indopaket/                # Regional package handling
│   ├── layouts/                  # Layout components (header, sidebar, breadcrumb)
│   ├── login/                    # Authentication
│   ├── manifest/                 # Manifest management
│   ├── manual-handling/          # Manual handling operations
│   ├── marketplace/              # Marketplace integration
│   ├── material-component/       # Material Design wrappers
│   ├── mitra-courier/            # Courier partner management
│   ├── modal/                    # Modal dialogs
│   ├── models/                   # Data models
│   ├── monitoring-booking/       # Booking monitoring
│   ├── monitoring-courier/       # Courier monitoring
│   ├── notification/             # Notification system
│   ├── outgoing-scan/            # Outgoing package scanning
│   ├── package-return/           # Package returns
│   ├── package-return-handling/  # Return processing
│   ├── pod-approval/             # Proof of Delivery approval
│   ├── profile/                  # User profile
│   ├── reports/                  # Standard reports
│   ├── reports-dc/               # Distribution center reports
│   ├── saldo/                    # Account balance
│   ├── services/                 # Business logic services
│   ├── shared/                   # Shared utilities and components
│   ├── tariff/                   # Pricing and tariff
```

```
│   ├── tracking/              # Package tracking
│   ├── trucking/              # Trucking operations
│   ├── user-manage/           # User management
│   ├── waybill/               # Waybill management
│   ├── api.service.ts         # Main API service
│   ├── auth.guard.ts          # Authentication guard
│   ├── role.guard.ts          # Role-based authorization guard
│   ├── app.module.ts          # Root module
│   └── app.routing.ts         # Routing configuration
├── assets/                    # Static assets (images, icons)
├── environments/              # Environment configurations
└── index.html                 # Main HTML file
```

Architecture Pattern

The application follows Angular's **module-based architecture**:

1. **Root Module** (`app.module.ts`) - Core module with essential dependencies
2. **Feature Modules** - Lazy-loaded modules for each feature area
3. **Shared Module** - Reusable components and utilities
4. **Services** - Business logic and API communication
5. **Guards** - Route protection and authorization
6. **Models** - TypeScript interfaces for data structures

Module Loading Strategy

- **Eager Loading**: Core components (layouts, login, profile)
- **Lazy Loading**: All feature modules for optimal performance
- Routes are protected by `AuthGuard` requiring authentication

Revision #1
Created 24 February 2026 07:09:59 by ondeliveloper
Updated 24 February 2026 07:10:57 by ondeliveloper