

Technical Overview & Flow Documentation

Tiknet Partner App

1. Introduction

The **Tiknet Partner App** is a comprehensive mobile application designed for Internet Service Provider (ISP) partners to manage their business operations efficiently. Built with Flutter, it provides tools for user management, internet plan configuration, router fleet monitoring, and financial tracking.

2. Technology Stack

- **Framework:** Flutter (Dart)
- **State Management:** Provider pattern (ChangeNotifier)
- **Networking:** Dio for HTTP requests
- **Local Storage:** `shared_preferences` and `flutter_secure_storage` for token management
- **UI/UX:** Material Design 3 (M3) with Custom Theming
- **Localization:** `easy_localization` (English & French support)
- **Charts:** `fl_chart` for data visualization

3. Application Architecture

The application follows a clean, layered architecture to ensure scalability and maintainability.

3.1. Layered Structure

1. **Presentation Layer (Screens & Widgets):** Handles UI rendering and user interactions.
2. **State Management Layer (Providers):**
 - `AuthProvider` : Manages authentication state (login, logout, user session).
 - `UserProvider` : Handles customer data, roles, and subscriptions.
 - `BillingProvider` : Manages wallet, transactions, and payout methods.
 - `NetworkProvider` : Controls router configuration, hotspots, and sessions.
 - `ThemeProvider` : Manages app theming (Light/Dark modes).
3. **Repository Layer:** Abstracts data sources and handles business logic transformations.

- Examples: `AuthRepository`, `PartnerRepository`, `WalletRepository`, `RouterRepository`.

4. Data Layer (Services):

- `ApiClient` : Centralized HTTP client with interceptors for auth headers and error handling.
- `TokenStorage` : Securely persists access and refresh tokens.

4. Core Workflows

4.1. Authentication Flow

- **Splash Screen:** Checks for existing authentication tokens.
- **Onboarding:** New users are guided through an introduction flow (tracked via `shared_preferences`).
- **Login:** Users authenticate with email/password.
 - *Mechanism:* `AuthRepository` sends credentials -> Receive OAuth2 tokens -> Store in `TokenStorage` .
- **Forgot Password:** OTP-based password reset flow (`ResetPasswordScreen` , `VerifyPasswordResetOtpScreen`).
- **Auto-Logout:** Illegal 401 responses trigger a global logout via `ApiClient` interceptors.

4.2. Dashboard & Navigation

- **Home Screen:** Acts as the main shell with a bottom navigation bar (mobile) or navigation rail (tablet).
- **Dashboard:** Displays real-time KPIs (Total Users, Active Sessions, Revenue, Active Routers).

4.3. User Management

- **List View:** Searchable and filterable list of all customers (`UsersScreen`).
- **User Details:** Comprehensive view of a specific user including internet plans, assigned devices, and payment history.
- **Profile Management:** Create and edit user profiles with specific roles and permissions.

4.4. Internet Plan Management

- **Plan Creation:** Partners can define plans with specific constraints:
 - Download/Upload Speed (Mbps)
 - Data Limits (GB/MB)
 - Validity Periods (Days/Hours)
 - Concurrent Session Limits
- **Assignment:** Plans can be assigned significantly to users or routers.

4.5. Network & Router Management

- **Router Registration:** QR code scanning or manual entry to add new Mikrotik/compatible routers (`RouterRegistrationScreen`).
- **Monitoring:** Real-time health checks (`RouterHealthScreen`) showing CPU, Memory, and Uptime.
- **Configuration:** Remote configuration of router settings like idle timeouts and rate limits.

4.6. Billing & Finance

- **Wallet:** Overview of current balance and total earnings (`WalletOverviewScreen`).
- **Payouts:** Partners can request payouts to mobile money or bank accounts (`PayoutRequestScreen`).
- **History:** Detailed transaction logs for audits and reconciliation (`TransactionHistoryScreen`).

4.7. Settings & Configuration

- **App Settings:** Language toggles (EN/FR), Theme selection, Notification preferences.
- **Business Settings:** Default plan configurations, worker profile setups.

5. Security Measures

- **Token Storage:** Secure storage mechanism for sensitive auth tokens.
- **HTTPS:** All API communication is encrypted.
- **Role-Based Access Control (RBAC):** App features are gated based on the logged-in partner's permissions.

6. Deployment

- **Versioning:** Semantic versioning tracked in `pubspec.yaml` (currently `1.0.0+1`).
- **Platforms:** Optimized for Android and iOS, with Web support capabilities.