AgroTrace Application Detailed Pages Report

Generated: 2025-12-08 19:11:04.854798

Table of Contents

- SafeNetworkImage
- LocalStore
- Exporter
- API Client & Config
- Routes Summary
- Cross‑Platform & Offline Notes
- Future Wiring

Report

AgroTrace Application — Detailed Pages Report

Overview
- AgroTrace is a cross‑platform Flutter application for farm management, drone operations, and administrative oversight. It supports offline usage (local storage, asset fallbacks), cross‑platform exports, and simple on‑device ML scaffolding.
- Navigation is role‑driven: users select a role and are routed to context‑specific dashboards. Persistent UI settings (theme, role) are stored via `shared_preferences`.

Authentication
Login (`auth/login.dart`)
- Purpose: Authenticate users and provide entry points to account creation and recovery.
- UI & Inputs: Email field; password field with visibility toggle. Primary action `Login`.
- Behavior: On press, navigates to role selection (`/role`). Provides links to `Forgot Password` (`/forgot`) and `Sign Up` (`/signup`).
- Data/State: No backend call currently; prepared for wiring to API. Theme follows app setting.

Sign Up (`auth/signup.dart`)
- Purpose: Register a new account.
- UI & Inputs: Name, email, password, confirm password.
- Validation: Confirms passwords match; shows success/failure via snackbar.
- Behavior: On success, returns to Login. Ready to wire to `POST /auth/register` or similar.

Forgot Password (`auth/forgot_password.dart`)
- Purpose: Initiate password reset via email.
- UI & Inputs: Email field.
- Behavior: On submit, shows confirmation and returns to Login. Prepared to wire to `POST /auth/forgot`.

Role Selection (`auth/role_selection.dart`)
- Purpose: Choose app role (Farmer, Drone Operator, Admin/Agronomist).
- UI & Inputs: Three selectable role cards.
- Behavior: Persists `selectedRole` using `shared_preferences` and routes to `/farmer`, `/drone`, or `/admin`.

Farmer Workflow
Farmer Dashboard (`lib/main.dart`, route `/farmer`)
- Purpose: Primary workspace for farmers with five tabs: Home, Parcels, Sensors, Alerts, Reports.
- Navigation: Bottom navigation with fixed tabs.

Home (`lib/main.dart:207`)
- Purpose: Snapshot of farm status and quick actions.
- Sections:
  - Header with avatar leading to Profile (`/farmer/profile`) and bell icon to Alerts (`/alerts`).
  - Map preview card indicating farm area.
  - Quick Statistics: Total Area, Parcels, Active count.
  - Recent Alerts preview listing top alerts.
  - Today’s Irrigation Plan: progress, items, and button to view recommendations.
- Behavior: Taps navigate to system screens; data is illustrative and ready to be backed by real data.

Parcels (`lib/main.dart:498`)
- Purpose: List fields/parcels with key metadata.
- UI: Cards with parcel name, area, crop, and status color.
- Behavior: Tap opens `ParcelDetail` via `/parcel` passing arguments (name, area, crop, health, irrigation, tags).

Parcel Detail (`lib/main.dart:1590`)
- Purpose: Deep view for a single parcel.
- Sections:
  - Map panel for spatial context.
  - Identity cards: name, crop, area, and tags (chips colored by context).
  - Health and Irrigation panels with `PercentGauge` visualizations.
  - Disease History with sparkline trend.
  - Metadata cards: last scanned, nutrient levels, recommended actions.
  - Action button: Request Drone Scan.
- Behavior: Designed for both viewing and initiating interventions.

Sensors (`lib/main.dart:1718`)
- Purpose: Monitor environmental and soil conditions.
- UI: Dark themed dashboard with range chips (24H/7D/30D), stat tiles (temperature, humidity, soil moisture), and sparkline charts.
- Behavior: Range selection updates visual emphasis; ready to read real sensor feeds.

Reports (Farmer) (`lib/main.dart:740`)
- Purpose: Placeholder for farmer‑focused reports; intended to list and preview reports.
- Behavior: Currently a stub; Admin export provides the functional export.

Alerts (`alerts/alerts.dart`)
- Purpose: Centralized list of actionable notifications.
- UI: Tabbed view (All, Critical, Info). Each alert shows severity color, title, subtitle, and action.
- Behavior: Links to related flows: Disease Detection (`/disease`) and Irrigation Recommendation (`/irrigation`).

Irrigation Recommendation (`irrigation/recommendation.dart`)
- Purpose: Present recommended irrigation amounts and scheduling tips.
- UI: Cards for recommended application and schedule steps.
- Behavior: Guidance only; integrate with task execution when backend is ready.

Disease Detection (`disease/detection.dart`)
- Purpose: Summarize detections and run on-device ML.
- UI: List of detections (name, severity, confidence) and bounding box preview.
- ML: `DiseaseClassifier` stub invoked; shows dialog with label/score. Platform-specific implementations planned via conditional exports; web currently returns `unsupported-web`.

Drone Operator Workflow
Drone Dashboard (`lib/main.dart`, route `/drone`)
- Purpose: Operations cockpit with Live, Control, Missions, Profile tabs.
- Navigation: Bottom navigation with contextual pages.

Live Drone Stream Viewer (`lib/main.dart:796`)
- Purpose: Visualize live feed and quick actions.
- UI: Stream header with status indicator; player placeholder; info tiles.
- Actions: Capture via `/camera`; annotate stream; switch view.

Flight Control Panel (`lib/main.dart:847`)
- Purpose: Monitor flight status and perform actions during scanning.
- UI: Tiles for Battery, Signal, Altitude, Status; live map placeholder; scan progress bar.
- Actions:
  - Pause/Resume scan.
  - Upload Images: Opens native file picker, accepts multiple images, uploads to backend.
- Upload Flow:
  - `file_picker` selects images; builds `FormData` with repeated `files` entries (bytes or file path).
  - `ApiClient.instance.post('/uploads/images', data: form)` performs the upload via `dio`.
  - Snackbars provide outcome feedback; cross-platform support (mobile/desktop/web).

Mission Summary (`lib/main.dart:894`)
- Purpose: Consolidated mission review and reporting.
- UI: Hero image; mission card (name/date/status); quick stats; detection breakdown with legends; image gallery; `Generate Full Report` button.
- Behavior: Report generation leverages Admin export module for cross-platform PDF/CSV.

Drone Operator Profile (`lib/main.dart:1051`)
- Purpose: Operator identity, preferences, and session control.
- UI: Profile info, preferences placeholder, logout.
- Behavior: Logout returns to Login (`/login`).

Live Annotations (`lib/main.dart:1473`)
- Purpose: Annotate imagery in real time.
- UI: Overlay bounding boxes; info tiles (Threat Level, Affected Area, Detections); control row (pause, capture, save); preview thumbnail.
- Behavior: Designed for future linking to storage and analytics services.

Administration
Admin Dashboard (`lib/main.dart:1069`, route `/admin`)
- Purpose: Entry point for administrative modules.
- UI: Grid cards linking to Users, Farms, Devices, Health, AI Models, Task Management.
- Navigation: Uses named routes and context mapping.

Manage Users (`lib/main.dart:1124`)
- Purpose: CRUD foundation for users.
- UI: List tiles with avatar initials, email, role chip; floating add.
- Behavior: Edit/add placeholders; prepared to wire to backend endpoints.

Manage Farms & Parcels (`lib/main.dart:1150`)
- Purpose: Catalog of farms and parcel summaries.
- UI: Card per farm with cover image, location, summary metrics, parcel rows with status and edit.
- Behavior: Search field and edit actions ready for data binding.

Manage Sensors & Drones (`lib/main.dart:1190`)
- Purpose: Device fleet management.
- UI: Segmented buttons (Drones/Sensors), search, device cards showing status badge, signal, battery.
- Behavior: FAB/Add and map tab placeholders; intended to integrate with device registry APIs.

Manage AI Models (`lib/main.dart:1330`)
- Purpose: Oversight of ML models lifecycle.
- UI: Filters; model cards with status, version, accuracy, trained date, action (Deploy/Update/View/Restore), progress bars.
- Behavior: Actions designed for backend hooks; current UI demonstrates intent.

System Health Monitor (`lib/main.dart:1424`)
- Purpose: Platform observability.
- UI: Range chips; uptime/latency/error tiles; sparkline charts for trends.
- Behavior: Refresh control; intended to read system metrics from APIs.

Export Reports (Admin) (`lib/main.dart:1523`)
- Purpose: Generate and export operational reports in PDF or CSV.
- UI: Format selector (PDF/CSV), selection cards (date, type, scope), toggle for charts.
- Cross‑Platform Export:
  - Uses `pdf` to generate PDF in memory; CSV string for tabular output.
  - Calls `saveBytes(filename, bytes, mimeType)` from `core/exporter.dart`.
  - On mobile/desktop (`exporter_io.dart`): writes to application documents folder via `path_provider`/`dart:io`; snackbar shows full path.
  - On web (`exporter_web.dart`): creates a Blob and triggers browser download; snackbar indicates `download:<filename>`.
- Behavior: User sees success path or download prompt; designed for offline or online use.

Imaging
Device Camera View (`lib/main.dart:1875`, route `/camera`)
- Purpose: Capture imagery from device camera.
- UI: Camera preview or unavailable message; capture FAB; gallery icon in app bar.
- Behavior: Initializes camera; captures image; saves bytes to Hive (`LocalStore.savePhoto`); snackbar confirms.
- Error Handling: `mounted` checks; try/catch around camera operations.

Photo Gallery (`lib/main.dart:1925`)
- Purpose: View locally saved photos.
- UI: Grid of thumbnails; fetches IDs from `LocalStore.photoIds()` and bytes via `LocalStore.photoBytes(id)`.
- Behavior: Fully offline gallery using Hive.

Settings
Farmer Profile Settings (`farmer/profile_settings.dart`)
- Purpose: Toggle app theme with persistence.
- UI: Switch for dark/light theme.
- Behavior: Saves `isDark` to `shared_preferences`; `MaterialApp` reacts via `themeMode`.

Utilities & Infrastructure
SafeNetworkImage (`lib/main.dart:455`)
- Purpose: Resilient image loader with optional asset fallback for offline.
- Behavior: Shows progress indicator/loading; shows broken image icon on error.

LocalStore (`lib/core/local_store.dart`)
- Purpose: Hive setup and boxes (`settings`, `parcels`, `alerts`, `photos`).
- Behavior: `init()` bootstraps Hive; photo helpers to save and retrieve images.

Exporter (`lib/core/exporter.dart`, `exporter_io.dart`, `exporter_web.dart`)
- Purpose: Abstract file saving across platforms.
- Behavior: IO writes to sandbox path; web triggers download via Blob.

API Client & Config (`lib/core/api_client.dart`, `lib/core/config.dart`, `lib/core/backend.dart`)
- Purpose: HTTP client with dynamic base URL and Authorization header wiring.
- Behavior: Uses `dio`; listeners update base URL and headers when config changes.

Routes Summary
- `/login`, `/signup`, `/forgot`, `/role`
- `/farmer`, `/drone`, `/admin`
- `/alerts`, `/irrigation`, `/disease`, `/camera`, `/farmer/profile`
- `/admin/users`, `/admin/farms`, `/admin/devices`, `/admin/health`, `/admin/ai`, `/admin/tasks`
- `/parcel` (with arguments)

Cross‑Platform & Offline Notes
- Offline: Asset fallbacks for images; local Hive storage for settings and photos.
- Web: Export uses browser downloads; TFLite FFI not supported—stub returns `unsupported-web`.
- Mobile/Desktop: File saving to documents directory; camera capture supported.

Future Wiring
- Authentication endpoints (`/auth/register`, `/auth/login`, `/auth/forgot`) to be connected.
- Reports to include real data and charts; export to support sharing.
- Device registry, mission execution, and observability to connect to backend.