

Professional documentation package

Pedigree Technical Manual

Architecture, data model, hosting, and deployment notes

Roles

owner, admin, manager, worker

Stack

Flutter, Firebase, Riverpod, GoRouter

Hosting

Firebase Hosting and Cloud Functions

Generated for the Pedigree workspace

System architecture

The client is Flutter; Firebase supplies identity, data, and hosting.

The application boots in lib/main.dart, initializes Firebase, enables Firestore persistence, registers a background messaging handler, and then hands control to a Riverpod-powered MaterialApp.router. GoRouter reads authentication and membership state to redirect each user to the correct surface.

- Flutter and Riverpod manage the UI and state.
- GoRouter enforces auth and membership redirects.
- Cloud Firestore stores users, farms, memberships, and feature records.
- Cloud Functions provide server-side orchestration and the public stats endpoint.
- Firebase Hosting serves the landing page and the generated manuals.

Firebase services in use

Each service has a narrow responsibility so the system stays easy to reason about.

Service	Role in the system
Firebase Auth	Identity and sign-in
Firestore	Profiles, farms, memberships, and live records
Cloud Functions	Server-side workflows and public stats
Firebase Hosting	Landing page and manuals
Messaging	Background message handling and notifications
Analytics and Crashlytics	Telemetry and error reporting

Core paths

These paths are central to the current implementation.

Path	Use
users/{uid}	Profile and user settings
farms/{farmId}	Farm metadata and public form link
farms/{farmId}/members/{uid}	Role assignment and status
farms/{farmId}/{collection}/{docId}	Operational records
globalAds/{adId}	Cross-farm promotion content
invites/{inviteId}	Pending membership invites

Path	Use
formIngestQueue/{id}	Public capture submissions

Security model

Firestore rules are role-aware and farm-scoped.

- Signed-in users can read and write their own profile document.
- Farm membership is required before a user can read farm data.
- Owners and admins manage members.
- Managers can manage operations but not membership records.
- Workers can write tasks and interact with worker-safe collections.

Public landing page statistics

The landing page fetches a public Cloud Function so no private collection is exposed.

The `getSiteStats` function returns the current counts for users, customers, and farms. The landing page polls it periodically and updates the live tracker without direct Firestore access from the browser.

Deployment checklist

This is the minimum release path for the docs site and manuals.

- Run `scripts/generate_manuals.dart` to refresh the PDF manuals.
- Build or verify the landing page files in `site/`.
- Deploy hosting together with the public stats function.
- Use the Firebase emulator suite for local validation before release.

Operational notes

The app already includes analytics, Crashlytics, messaging, and sync-center hooks.

- Offline Firestore persistence is enabled at startup.
- The sync center surfaces pending writes and connectivity state.
- Crashlytics receives fatal Flutter errors and uncaught platform errors.
- Firebase Analytics logs the app open event on launch.
- Background messaging is initialized before the app starts.