**DrsMainApp — Engineering Roadmap (2025-10-26)**

Objective: Build a robust macOS-first (with iOS option) doctor-facing app that imports patient bundles, previews data, generates/share reports, and never retains PHI by default. Follow industry best practices and share code with PatientViewerApp via the PediaShared Swift Package.

**Guiding Principles**

• Security & privacy first: PHI is ephemeral; default to no local retention.

• Single source of truth: the patient bundle (SQLite + docs).

• Shared code where it matters: logging, bundle paths, DB helpers, rendering utilities.

• Crash-safe, testable, observable: unit tests + signposts + os.Logger.

• Small, incremental milestones with user-visible value.

**Workspace & Structure**

• PediWorkspace.xcworkspace contains: DrsMainApp, PatientViewerApp, PediaShared (SwiftPM).

• PediaShared targets iOS 17+ and macOS 14+; no platform-conditional APIs in core utilities.

• Local packages for DB and archive: SQLite.swift (or FMDB), ZIPFoundation; minimize external deps.

**Data & Storage**

• Bundle format: /<Patient Alias>/db.sqlite + /docs + manifest.json (as in PatientViewerApp).

• Open DB read-only; write only when exporting/annotating per explicit user action.

• No background sync; manual export of signed bundles to parents’ device.

• Keychain for API keys/config; never log PHI; redact paths in logs.

**Security & Compliance**

• Use secure-scoped bookmarks to access external bundles on macOS.

• Gate sharing behind NSSharingServicePicker; audit file types (PDF, TXT, CSV only).

• App Sandbox: File Access (User Selected) + Read-only default; temporary write during export.

• Privacy manifest + analytics off by default.

**Phased Delivery Plan**

Phase 0 — Workspace & Tooling (DONE / KEEP HEALTHY)

• Unified workspace; PediaShared package wired to DrsMainApp.

• Upgrade checklist + backup scripts committed; os.Logger baseline.

Phase 1 — Foundations (Weeks 1–2)

• App shell (SwiftUI, macOS target), AppState, simple navigation.

• PediaShared.SharedPaths (bundle locations, temp/persistent directories).

• PediaShared.Logging (AppLog categories, signposts).

• PediaShared.BundleIO (ZIP open, manifest read, DB path resolve).

• Read-only SQLite access layer (lightweight query helpers).

• Unit tests for SharedPaths/BundleIO/DB open.

Phase 2 — Import & Patient Browser (Weeks 3–4)

• macOS file importer (NSOpenPanel) → secure bookmark → mounted bundle cache.

• Patient list view: alias, DOB, sex; last updated; validation badges.

• Detail view: Visits list (well/sick toggle), documents tab (QuickLook preview).

• Share from preview (PDF + documents) using system share sheet.

Phase 3 — Reporting (Weeks 5–6)

• Well/Sick visit PDF generation (reuse renderers from PatientViewerApp; move common pieces to PediaShared as needed).

• Pluggable watermark/signature and clinic header.

• One-click export/print from detail view.

Phase 4 — AI Assistant (Optional, Weeks 7–8)

• Abstraction over providers (OpenAI, local) with background-safe queues.

• Prompts: visit summary, parent instructions, vaccine schedule cross-check.

• All outputs treated as draft; no auto-save without user action.

Phase 5 — Export to Parent Bundle (Weeks 9–10)

• Create sanitized bundle (subset of DB + docs) compatible with PatientViewerApp.

• Checksum/manifest; optional encryption-at-rest zip.

• Post-export purge of temp files; audit log entry.

Phase 6 — Quality, Hardening, Release (Weeks 11–12)

• Unit/UI tests; fixture bundles for regression.

• Performance pass with signposts; memory audits on large bundles.

• Crash reporting (symbolicated) and privacy manifest verification.

• macOS notarization pipeline; TestFlight (if iOS variant).

**Shared Code Candidates (PediaShared)**

• Logging (AppLog + categories).

• SharedPaths + FileScopes (temp, caches, exports).

• BundleIO (zip/manifest/db discovery + validation).

• DB schema helpers (typed queries for patients/visits/notes).

• Report rendering utilities (growth chart drawing primitives, date/age formatting).

• QuickLook/Share adapters (platform-conditional wrappers).

**Key Decisions & Risks**

• DB layer: keep FMDB/SQLite.swift vs move to GRDB — decide in Phase 1.

• Min OS versions (macOS 14, iOS 17) — impacts APIs and testers.

• AI provider API quotas/keys — gate behind feature flag.

• PHI handling: enforce transient storage; consider 'clinic hold copy' opt-in.

**Branching, CI/CD & Dev Practices**

• Git: main (release), develop (integration), feature/\* branches; PR reviews required.

• SwiftFormat + SwiftLint pre-commit; danger/swift for PR checks.

• GitHub Actions: build + unit tests + swiftformat/lint; export artifacts for TestFlight/notarization.

• Semantic versioning; CHANGELOG.md updated per release; UPGRADE\_CHECKLIST.md kept current.

**Definition of Done (per feature)**

• Unit tests and UI smoke tests pass locally and in CI.

• No new linter warnings; performance within targets; logs redact PHI.

• Docs updated (README, user guide, upgrade notes).

**Sprint 1 Backlog (Kickoff)**

1. Create AppState + skeleton views (Sidebar, PatientList, Details).

2. Implement PediaShared.SharedPaths and BundleIO with tests.

3. Wire import (NSOpenPanel) + bookmark storage; display patient metadata.

4. Integrate QuickLook preview; add share button; basic logging signposts.

**Sprint 2 Backlog**

1. Read-only DB helpers (patients, visits, notes) + unit tests.

2. Visits list (well/sick) and detail; preview PDF via existing generators.

3. Experimental: move shared PDF/growth helpers into PediaShared.