

ATHENA AFTERCARE

Developer Brief: Phase 1 MVP — Native iOS Build & App Store Launch

Prepared by: Lia Colabello, Founder | lia@athenaaftercare.com | Date: July 2026 | Confidential

1. Project Overview

Athena Aftercare, LLC educates breast cancer survivors and caregivers about breast cancer aftercare, with an immediate focus on Breast Cancer-Related Lymphedema (BCRL) — an incurable, chronic swelling of the arms, hands, chest, and trunk that can develop days to decades after breast cancer treatment. Our brand centers on the idea that cancer survivors are superheroes, expressed through a purple, gold, and star visual identity. Our website is athenaaftercare.com.

This engagement covers Phase 1 of a three-phase product roadmap: a general-wellness symptom-tracking mobile app that does not require FDA clearance. An interactive HTML prototype of the app already exists and defines the complete UI, screen flows, branding, and feature set. Your job is to rebuild that prototype as a production-quality native iOS app and ship it to the Apple App Store.

Important context: Phase 2 of our roadmap will introduce HIPAA-compliant backend and cloud infrastructure (accounts, sync, and care-team connectivity). Phase 3 will layer a computer-vision limb-measurement tool onto that infrastructure, positioned as a clinician-grade diagnostic aid and regulated as Software as a Medical Device (SaMD) requiring FDA clearance. Phase 1 — this engagement — is deliberately scoped to stay outside both HIPAA and FDA jurisdiction. This makes several constraints in this brief non-negotiable — they are regulatory strategy, not preference.

2. Scope of Work

1. Rebuild the existing HTML prototype as a native iOS app using the confirmed tech stack (Section 3). This is a rebuild, not a WebView wrapper — Apple guideline 4.2 (minimum functionality) rejects thin web wrappers, and we have flagged this risk already.
2. Implement all Phase 1 features as defined in the prototype: symptom tracking and logging, trend views, reminders, educational content, and onboarding. No new features; no Phase 2 or Phase 3 features (these are parked in a separate R&D spike note and must not appear in this build).
3. Implement the local-only data architecture described in Section 4.
4. Configure EAS Build, internal device testing, and TestFlight beta distribution.
5. Prepare and submit the app to the Apple App Store, including App Store Connect setup, App Privacy nutrition labels, screenshots, and metadata, and manage the review process through approval.
6. Deliver the complete source repository, build configuration, and a handoff document sufficient for a future developer to continue work.

Out of scope: Android/Google Play (may be a follow-on engagement), any backend or cloud infrastructure, user accounts/authentication, any diagnostic or measurement features, marketing assets.

3. Confirmed Tech Stack

The stack below is already decided and validated against our regulatory and product constraints. Substitutions require written approval from the founder before any work begins on the affected layer.

Layer	Technology
Framework	React Native with Expo (managed workflow), TypeScript
Local database	SQLite via Drizzle ORM (expo-sqlite driver) — on-device only
State management	Zustand
Build & distribution	EAS Build for device builds; TestFlight for beta; App Store Connect for release
Project guardrails	A CLAUDE.md file in the repo defines hard guardrails (no diagnostic behavior, no PHI leaving the device). These apply to all contributors, human or AI-assisted.

4. Data Architecture: Where Data Lives

All user data is stored exclusively on the user's device in a local SQLite database managed through Drizzle ORM. There is no backend, no cloud sync, no user accounts, and no server that receives health data. This is the single most important architectural constraint in the project.

4.1 Hard requirements

- **Local-only persistence** — All symptom logs, notes, reminders, and user-entered health information persist only in the on-device SQLite database.
- **Zero PHI egress** — No health data, symptom data, or personally identifiable information may be transmitted off the device — not to our servers (we have none), not to analytics providers, not to crash reporters. If crash reporting is used (e.g., Sentry), it must be configured to scrub all user content and collect technical diagnostics only, and this configuration must be documented and demonstrated.
- **No tracking SDKs** — No third-party SDKs that collect user data, advertising identifiers, or behavioral analytics. Every dependency added to the project must be justified in the handoff document.
- **Device backups** — The SQLite database will be included in standard iOS device backups (iCloud/local) by default. This is acceptable — it is Apple's infrastructure under the user's own Apple ID, not our infrastructure. Document this behavior clearly. Do not build custom cloud backup or export-to-server features.
- **User control** — Provide a user-facing 'delete all my data' function that drops the local database, and a local export (e.g., CSV or PDF shared via the iOS share sheet) so users can share logs with their own care team on their own initiative.

4.2 Why this matters

The local-only architecture is what keeps Phase 1 out of both FDA device territory and most data-privacy exposure. Because we never receive or store user health data, we are not acting as a covered entity or business associate under HIPAA, and our privacy policy can truthfully state that health data never leaves the device. Any deviation — however convenient for debugging or analytics — undermines the entire Phase 1 strategy. When in doubt, ask before adding anything that opens a network connection.

5. Regulatory & Content Guardrails

- **No diagnostic behavior** — The app is a general wellness product. All in-app copy must avoid diagnostic, treatment, or measurement claims. The app tracks and displays what the user enters; it never interprets symptoms, assesses severity, diagnoses, or recommends clinical action.
- **Language rules** — Prohibited language patterns include: 'detect,' 'diagnose,' 'assess your condition,' 'your lymphedema is worsening,' staging language, or any automated alerts implying clinical judgment. Acceptable: neutral trend displays, user-set reminders, and educational content with sources.
- **Disclaimers** — Educational content should encourage users to consult their care team (physicians, Certified Lymphedema Therapists) and must include a clear 'not medical advice' disclaimer surfaced during onboarding and in settings.
- **Apple 5.1.3** — Apple App Review guideline 5.1.3 (Health and Health Research) applies: health apps must not use health data for advertising, must safeguard it, and must not write false data. Our local-only architecture satisfies this, but App Store metadata and review notes must be written accordingly.
- **Phase 3 firewall** — Any feature idea that drifts toward measurement, imaging, or clinical assessment belongs to Phase 3 and must be declined for this build, even if requested informally mid-project.

6. Required Developer Skills

The ideal candidate is a mid-to-senior mobile developer who has personally shipped at least one React Native app to the App Store. Specifically:

- **React Native / Expo** — 2+ years with React Native and TypeScript; hands-on experience with the Expo managed workflow and EAS Build (not just bare React Native).
- **Local-first data** — Experience with local persistence on mobile — ideally Drizzle ORM or comparable (WatermelonDB, TypeORM, raw expo-sqlite), including schema migrations on-device.
- **State management** — Comfortable with Zustand or able to adopt it quickly (it is a small API; prior Redux/MobX experience transfers).
- **App Store submission** — Has taken an app through App Store Connect end to end: certificates and provisioning, TestFlight, App Privacy nutrition labels, screenshots, review notes, and responding to App Review rejections. Experience with a health/wellness app review is a strong plus.
- **UI implementation** — Able to implement a provided design faithfully — our purple/gold/star brand system and the prototype's screens are the spec. Pixel-level judgment matters; a designer is not part of this engagement.

- **Privacy mindset** — Understands why 'just add Firebase Analytics' is the wrong answer here. The willingness to respect the local-only guardrails is mandatory regardless of background.
- **HIPAA experience** — Prior experience building or maintaining a HIPAA-compliant product (covered entity or business associate side) is strongly preferred, even though this app's local-only architecture is deliberately designed to fall outside HIPAA's jurisdiction. That background means the developer already thinks in terms of PHI handling, access controls, audit trails, and data minimization by default — the same instincts this build needs — and it directly prepares them for Phase 2, where we will introduce HIPAA-compliant backend and cloud infrastructure, and Phase 3, where our computer-vision feature will require FDA clearance on top of that infrastructure.
- **Nice to have** — Nice to have: accessibility (VoiceOver, Dynamic Type), local notifications, CSV/PDF export via the iOS share sheet, unit testing with Jest.

7. What We Provide

- The interactive HTML prototype (complete UI, flows, and copy) — this is the functional spec.
- Brand assets: logo, color palette (purple/gold), star iconography, typography guidance.
- The CLAUDE.md guardrails file and the R&D spike note marking Phase 2 and Phase 3 boundaries.
- App Store metadata copy (description, keywords, subtitle) and the privacy policy hosted at athenaaftercare.com.
- An Apple Developer Program account (organization enrollment) with appropriate access granted to you.

8. Deliverables & Milestones

Milestone	Deliverable & acceptance criteria
M1 — Project setup (Week 1)	Repo initialized with Expo + TypeScript + Drizzle + Zustand; CLAUDE.md in place; EAS Build producing an installable dev build on the founder's device.
M2 — Core build (Weeks 2–4)	All prototype screens implemented natively; symptom logging writes/reads from local SQLite; schema migration path demonstrated; local notifications for reminders working.
M3 — Data controls & polish (Week 5)	Delete-all-data and local export functions complete; disclaimers and onboarding finalized; accessibility pass; no network calls verified (demonstrated via proxy/log inspection).
M4 — TestFlight beta (Week 6)	Build distributed via TestFlight to founder + up to 10 testers; feedback triaged; crash-free sessions across core flows.
M5 — App Store submission (Weeks 7–8)	App Privacy labels completed accurately (no data collected); metadata and screenshots uploaded; app submitted; review feedback addressed through approval and release.
M6 — Handoff	Source repo transferred; README covering build, signing, release process, dependency justifications, and known issues; 1-hour walkthrough call.

Timeline above is indicative; propose adjustments in your bid. Payment is milestone-based, with final payment tied to App Store approval and handoff, not merely submission.

9. Acceptance & Verification

7. The app runs on a physical iPhone via TestFlight and matches the prototype's screens and flows.
8. A network inspection session (e.g., Charles Proxy) demonstrates zero transmission of user-entered data under normal use.
9. App Privacy nutrition label in App Store Connect accurately reflects the no-collection architecture and is consistent with the privacy policy.
10. Deleting the app and reinstalling behaves as documented; delete-all-data works; export produces a readable file via the share sheet.
11. Codebase passes TypeScript strict checks and includes the dependency-justification list.

10. Terms & Working Arrangement

- Engagement type: fixed-scope contract with milestone payments (proposal should include total cost and per-milestone breakdown).
- IP assignment: all code, assets, and documentation are work-for-hire, assigned to Athena Aftercare; an NDA and IP assignment agreement will be signed before repo access.
- Communication: async-first (email/Slack) with one weekly check-in call; founder reviews each milestone within 3 business days.
- Post-launch: quote an optional 30-day bug-fix warranty and an hourly rate for follow-on work (an Android build of this same Phase 1 app is the likely near-term follow-on; Phase 2 backend work would be a separate future engagement).

To respond: send your proposal, portfolio (App Store links to apps you shipped), milestone pricing, and earliest start date to Lia Colabello at lia@athenaaftercare.com. Questions about scope are welcome and encouraged before bidding.