Notelayer iOS - One-Page Summary

Repo evidence only | Generated 2026-02-11

What it is

- Notelayer is a native SwiftUI iOS app for managing notes and to-dos with categories, reminders, calendar export, and cloud sync.
- The root experience is a three-tab layout: Notes, To-Dos, and Insights.

Who it's for

- Primary persona: Not found in repo.
- Inferred from repo docs: iOS users who want a personal operations dashboard for planning, completing, and reviewing tasks over time.

What it does

- Captures and manages notes and tasks with title, categories, priority, due date, and task notes.
- Supports four task organization modes: List, Priority, Category, and Date.
- Schedules local reminders and supports notification actions to complete or open tasks.
- Exports tasks into Apple Calendar using EventKit event editing.
- Syncs notes, tasks, and categories to Firebase Firestore for authenticated users.
- Provides an Insights tab with trends, category stats, feature usage, time-of-day usage, and gap drilldowns.
- Processes iOS Share Extension items into tasks via App Group storage.

How it works

- UI components: NotelayerApp initializes AuthService and FirebaseBackendService, then presents RootTabsView with NotesView, TodosView, and InsightsView.
- Data layer: LocalStore is the app's observable local source of truth and persists notes/tasks/categories to App Group UserDefaults.
- Sync layer: FirebaseBackendService reacts to auth state, binds LocalStore to Firestore user collections, runs initial sync, and keeps live listeners active.
- Feature services: ReminderManager handles UserNotifications, CalendarExportManager handles EventKit export, and ThemeManager controls appearance tokens.
- Telemetry flow: AnalyticsService logs to Firebase Analytics and mirrors events into InsightsTelemetryStore; InsightsAggregator builds snapshots for InsightsView.
- Data flow: UI action -> LocalStore mutation -> local persist plus optional backend upsert and side effects; remote updates -> backend listeners -> LocalStore applyRemote methods -> SwiftUI refresh.

How to run

- Install Xcode on macOS.
- Open ios-swift/Notelayer/Notelayer.xcodeproj in Xcode.
- Select an iPhone simulator (or connected device).
- Press Cmd+R to build and run.
- For physical device runs, Apple Developer account details may be required for signing.

Evidence files

Sources: docs/Project_Readme.md, docs/Quick_Start.md, docs/Project_Changelog.md, docs/App_Store_Release_Notes.md, ios-swift/Notelayer/Notelayer/App/NotelayerApp.swift, ios-swift/Notelayer/Notelayer/Views/RootTabsView.swift, ios-swift/Notelayer/Notelayer/Views/TodosView.swift, ios-swift/Notelayer/Notelayer/Views/InsightsView.swift, ios-swift/Notelayer/Notelayer/Data/LocalStore.swift, ios-swift/Notelayer/Notelayer/Services/FirebaseBackendService.swift, ios-swift/Notelayer/Notelayer/Services/ReminderManager.swift, ios-swift/Notelayer/Notelayer/Services/CalendarExportManager.swift, ios-swift/Notelayer/Notelayer/Services/AnalyticsService.swift, ios-swift/Notelayer/Notelayer/Data/SharedItem.swift