Jar

Comprehensive Product Requirements Document

Version 1.0.0 — Production Release Specification

Document Version:	2.0 (Comprehensive)
Target Release:	v1.0.0
Author:	Erich Benjamin Hubert
Last Updated:	January 2026
Repository:	github.com/Enrichment2/jar
Design System:	Material Design 3 (Material You)
Target Platform:	Android 8.0+ (API 26+)

This document defines every requirement for Jar v1.0.0. The app must look and feel indistinguishable from a native Google-made Android application.

Table of Contents

1. Executive Summary & Vision

Jar is a local-first, privacy-focused productivity application for Android. It combines notes, tasks, and reminders into a single, beautifully designed app that respects user privacy absolutely. All data stays on-device. No accounts. No cloud. No tracking.

1.1 Vision Statement

Create the most user-respecting productivity app on Android that proves privacy and excellent UX are not mutually exclusive. Jar will be indistinguishable in quality from first-party Google apps while maintaining complete user data sovereignty.

1.2 Success Definition

- Users cannot tell Jar apart from a Google-made app in a blind test
- Zero data ever leaves the device (verifiable via network inspection)
- App launches in under 500ms on mid-range devices
- 4.5+ star rating potential based on UX quality

1.3 Target Users

- Privacy-conscious Android users tired of apps harvesting their data
- Users who want a unified productivity solution (not 5 separate apps)
- Open-source advocates who want to inspect and trust their tools
- Users migrating from Google Keep/Tasks who want data ownership

2. Core Principles (Non-Negotiable)

These principles override all other requirements. Any feature that conflicts with these principles must be redesigned or rejected.

LOCAL-FIRST

- All data stored exclusively in app-private storage on device
- App must function 100% offline with zero degradation
- No network permissions requested in AndroidManifest.xml
- No analytics, telemetry, crash reporting, or any data transmission
- No Firebase, no Google Play Services dependencies for core functionality

PRIVACY BY DESIGN

- Zero knowledge architecture — app knows nothing about users
- No accounts, sign-ups, emails, or any personal identifiers
- No clipboard monitoring or background data access
- Exported data contains no device identifiers or metadata
- Secure deletion: overwrite data before unlinking from filesystem

OPEN SOURCE (AGPL-3.0)

- 100% of code publicly auditable on GitHub
- No obfuscated code in release builds (users can verify)
- No proprietary dependencies for core functionality
- Build process fully reproducible from source
- All third-party dependencies must be open-source compatible

USER DATA OWNERSHIP

- Export all data in open formats (JSON, Markdown, plain text)
- Import from common formats (JSON, plain text)
- No vendor lock-in — users can leave anytime with all their data
- Data remains readable even if app is uninstalled (via prior export)
- Clear documentation of data format for third-party tool compatibility

NATIVE ANDROID QUALITY

- Indistinguishable from Google's first-party apps
- Full Material Design 3 compliance with Material You theming
- Support all standard Android interactions and patterns
- Predictive back gesture support

- Edge-to-edge design with proper insets handling

3. UI/UX Requirements — Android Design Compliance

The app must adhere to official Android design guidelines. Reference: developer.android.com/design/ui/mobile

3.1 System Bars

[REQ-UI-001] Status bar must be transparent

- Content draws behind the status bar (edge-to-edge)
- Status bar icons automatically adapt to light/dark content beneath
- Use WindowInsets to prevent content from being obscured

```
SPEC: enableEdgeToEdge() called in onCreate()
```

[REQ-UI-002] Navigation bar must be transparent (gesture nav) or translucent (3-button)

- Gesture navigation: fully transparent, content draws behind
- 3-button navigation: translucent scrim, content draws behind
- Bottom navigation components must not overlap system navigation

```
SPEC: Window.setNavigationBarContrastEnforced(false) for bottom app bar scenarios
```

[REQ-UI-003] Display cutout handling

- Content must not be obscured by camera notches or punch-holes
- Use DisplayCutout APIs to detect and respect safe areas
- Test on devices with various cutout shapes

3.2 Material Design 3 Components

[REQ-UI-004] Use only Material 3 components

- TopAppBar (Large, Medium, or CenterAligned depending on context)
- FloatingActionButton for primary actions (standard or extended)
- NavigationBar for bottom navigation (3-5 destinations max)
- Cards for content containment (ElevatedCard or OutlinedCard)
- Chips for tags (FilterChip, InputChip, AssistChip as appropriate)
- TextField with proper label, supporting text, and error states
- Dialogs (AlertDialog) for confirmations and simple inputs
- BottomSheet for contextual actions and overflow content
- Snackbar for brief feedback messages (with optional action)
- ProgressIndicator (Linear or Circular) for loading states

```
SPEC: All components from com.google.android.material or Compose Material3
```

3.3 Material You Dynamic Color

[REQ-UI-005] Full dynamic color theming

- On Android 12+: derive color scheme from user's wallpaper
- On Android 11 and below: use default theme (harmonious fallback)

- All interactive elements use dynamic color tokens
- Surfaces, containers, and backgrounds use appropriate surface tokens
- Text uses onSurface, onPrimary, onSecondary tokens correctly

`SPEC: DynamicColors.applyToActivitiesIfAvailable(application)`

[REQ-UI-006] Light and dark theme support

- Follow system setting by default
- Allow user override in settings (System / Light / Dark)
- Smooth transition when theme changes (no flash)
- All screens tested in both themes

`SPEC: AppCompatDelegate.setDefaultNightMode() with MODE_NIGHT_FOLLOW_SYSTEM default`

3.4 Typography

[REQ-UI-007] Material 3 type scale

- Display: 57sp (Large), 45sp (Medium), 36sp (Small)
- Headline: 32sp (Large), 28sp (Medium), 24sp (Small)
- Title: 22sp (Large), 16sp (Medium), 14sp (Small)
- Body: 16sp (Large), 14sp (Medium), 12sp (Small)
- Label: 14sp (Large), 12sp (Medium), 11sp (Small)
- Use system font (Roboto) unless user has accessibility override

`SPEC: Use MaterialTheme.typography tokens in Compose`

3.5 Spacing and Layout

[REQ-UI-008] 8dp grid system

- All spacing must be multiples of 8dp (8, 16, 24, 32...)
- Exception: 4dp for tight optical adjustments
- Consistent padding: 16dp horizontal screen margins
- Card padding: 16dp internal padding
- List item height: minimum 48dp for single-line, 64dp+ for two-line

[REQ-UI-009] Touch targets

- Minimum touch target: 48x48dp
- Recommended touch target: 56x56dp for primary actions
- Adequate spacing between touch targets (minimum 8dp)

`SPEC: Modifier.minimumInteractiveComponentSize() in Compose`

3.6 Navigation Patterns

[REQ-UI-010] Bottom navigation bar

- Exactly 3 primary destinations: Notes, Tasks, Settings
- Icons: filled when selected, outlined when unselected
- Labels always visible (not hidden on unselected)
- Navigation state preserved when switching destinations

- Badge support for notification counts (optional)

[REQ-UI-011] Predictive back gesture

- Support Android 14+ predictive back animations
- Show preview of previous destination during back gesture
- Handle back press interception properly (unsaved changes)

```
SPEC: android:enableOnBackInvokedCallback="true" in manifest
```

[REQ-UI-012] Navigation transitions

- Container transform for opening items from list
- Shared element transitions where appropriate
- Fade through for navigation bar destination changes
- Standard motion duration: 300ms (Material standard)

3.7 Empty States

[REQ-UI-013] Thoughtful empty states for every list

- Centered vertically in available space
- Include: icon (optional), headline, supporting text, action (optional)
- Notes empty: "Your jar is empty" + "Tap + to capture your first thought"
- Tasks empty: "All clear!" + "Tap + to add something to do"
- Reminders empty: "No reminders set" + "Tap + to create a reminder"
- Search no results: "No results for '[query]'" + "Try different keywords"

3.8 Loading States

[REQ-UI-014] Loading feedback for all async operations

- Operations under 100ms: no loading indicator needed
- Operations 100ms-1s: show indeterminate progress indicator
- Operations over 1s: show progress with context ("Exporting notes...")
- Use skeleton screens for initial content load when appropriate
- Never block the entire UI — allow cancellation where possible

3.9 Error States

[REQ-UI-015] Graceful error handling

- Never show raw exceptions or stack traces to users
- Snackbar for recoverable errors with retry action
- Dialog for critical errors requiring user decision
- TextField error state for validation failures
- Error messages must be actionable (tell user what to do)

4. Feature Specifications — Notes Module

4.1 Notes List Screen

[REQ-NOTES-001] Notes list display

- Display notes in reverse chronological order (newest first) by default
- Each note card shows: title (bold, max 1 line), preview (max 2 lines), timestamp, tags
- Timestamp format: "Just now", "5 min ago", "Yesterday", "Jan 15", "Jan 15, 2025"
- Tags displayed as small FilterChips below preview (max 3 visible + "+N more")
- Pull-to-refresh gesture (even though local — for consistency)
- Smooth scrolling at 60fps minimum

[REQ-NOTES-002] Search functionality

- Search icon in TopAppBar expands to SearchBar
- Real-time filtering as user types (debounce 150ms)
- Search scope: title, body, and tag names
- Case-insensitive matching
- Highlight matching text in results (optional enhancement)
- Recent searches saved locally (last 10)
- Clear search with X button, keyboard dismiss on scroll

```
SPEC: Query latency < 50ms for 1000+ notes
```

[REQ-NOTES-003] Tag filtering

- Filter chip row below search bar (horizontal scroll)
- Tapping tag toggles filter (filled = active)
- Multiple tags can be selected (AND logic)
- "All" chip to clear filters
- Show count of filtered results

[REQ-NOTES-004] Sort options

- Access via overflow menu (3-dot icon)
- Options: Date modified (default), Date created, Alphabetical
- Persist sort preference

[REQ-NOTES-005] Note actions

- Tap: open note in editor
- Long press: enter selection mode, show contextual action bar
- Selection mode actions: delete, add tag, share, select all
- Swipe-to-delete with undo snackbar (5 second window)

4.2 Note Editor Screen

[REQ-NOTES-010] Editor layout

- Full-screen editor (not dialog)
- TopAppBar with back arrow, overflow menu
- Title field: single line, TextStyle.headlineSmall, hint "Title"

- Body field: multiline, fills remaining space, TextStyle.bodyLarge, hint "Start writing..."
- Keyboard appears automatically when creating new note (focus on title)
- Smooth keyboard animation (WindowInsets animation)

[REQ-NOTES-011] Auto-save behavior

- Save automatically on every pause (debounce 500ms)
- Save on back navigation
- Save on app backgrounding
- Never lose user data due to crash or kill
- Show subtle "Saved" indicator after save (fade out after 1s)
- Empty notes (no title, no body) are not saved

[REQ-NOTES-012] Tag management in editor

- Tags section at bottom of editor (collapsible)
- Existing tags shown as InputChips with X to remove
- "+ Add tag" chip opens tag picker bottom sheet
- Tag picker: shows all existing tags, text field to create new
- New tags created inline without leaving editor

[REQ-NOTES-013] Rich text editing (v1.0 scope)

- Bold text (Ctrl+B or toolbar button)
- Italic text (Ctrl+I or toolbar button)
- Bullet list
- Numbered list
- Headings (H1, H2)
- Formatting toolbar appears above keyboard when text selected
- Store as Markdown internally, render appropriately

[REQ-NOTES-014] Editor overflow menu actions

- Share (as plain text, Markdown, or PDF)
- Delete (with confirmation dialog)
- Copy all
- View info (created date, modified date, word count, character count)

4.3 Tags System

[REQ-NOTES-020] Tag data model

- Tag entity: id (auto), name (unique), color (nullable), createdAt
- Many-to-many relationship: NoteTagCrossRef table
- Tag name constraints: 1-30 characters, no special characters except - and _
- Maximum 20 tags per note

```
SPEC: Room database with proper foreign keys and indices
```

[REQ-NOTES-021] Tag colors

- 12 preset colors derived from Material color palette
- Colors work in both light and dark themes

- Default: no color (uses surface variant)
- Color picker shown as horizontal row in tag editor

[REQ-NOTES-022] Tag management screen

- Accessible from Settings > Manage Tags
- List all tags with note count for each
- Edit tag: rename, change color
- Delete tag: removes from all notes (with confirmation)
- Merge tags: combine two tags into one (advanced)

5. Feature Specifications — Tasks Module

5.1 Tasks List Screen

[REQ-TASK-001] Task list display

- Group by: Today, Tomorrow, This Week, Later, No Date, Completed
- Collapsible section headers with task count
- Each task shows: checkbox, title, due date/time (if set), priority indicator
- Overdue tasks: red text color, shown in "Overdue" section at top
- Completed tasks: strikethrough, moved to Completed section
- Completed section collapsed by default

[REQ-TASK-002] Task checkbox interaction

- Checkbox: 24dp, proper touch target (48dp)
- Completion animation: checkbox fills, text fades/strikes, item slides to Completed
- Undo available via snackbar for 5 seconds
- Haptic feedback on completion (subtle)

[REQ-TASK-003] Quick add

- FAB opens quick add bottom sheet (not full editor)
- Single text field for task title
- Optional chips: Today, Tomorrow, Next Week, set time
- Create with Enter key, sheet stays open for rapid entry
- "Done" button closes sheet

5.2 Task Editor Screen

[REQ-TASK-010] Task editor layout

- Full-screen editor
- Title field (required)
- Description field (optional, multiline)
- Due date picker (date only, or date + time)
- Priority selector (None, Low, Medium, High) as segmented button
- Repeat selector (None, Daily, Weekly, Monthly, Custom)
- Reminder toggle (if due date set, default reminder at due time)

[REQ-TASK-011] Due date picker

- Material 3 DatePicker dialog
- Quick options: Today, Tomorrow, Next Week, Pick Date
- Optional time picker (TimePicker dialog)
- Clear button to remove due date
- Display format: "Today", "Tomorrow", "Mon, Jan 15", "Mon, Jan 15 at 3:00 PM"

[REQ-TASK-012] Recurring tasks

- When recurring task completed: create next occurrence, mark current done

- Daily: same time next day
- Weekly: same day/time next week
- Monthly: same date next month (handle edge cases: 31st → 28th)
- Custom: every N days/weeks/months
- Show recurring icon on task list

[REQ-TASK-013] Priority levels

- High: red indicator, sorted first within section
- Medium: orange indicator
- Low: blue indicator
- None: no indicator (default)
- Priority colors work in both light and dark themes

6. Feature Specifications — Reminders Module

6.1 Reminders List Screen

[REQ-REM-001] Reminders list display

- Group by: Upcoming, Past
- Each reminder shows: title, date/time, repeat indicator
- Past reminders can be deleted or rescheduled
- Chronological order within each section

[REQ-REM-002] Creating reminders

- FAB opens reminder editor
- Required: title, date, time
- Optional: repeat schedule, link to note/task
- Quick options: In 1 hour, Tomorrow morning, Tomorrow evening

6.2 Notifications

[REQ-REM-010] Notification requirements

- Request POST_NOTIFICATIONS permission (Android 13+)
- Create notification channel: "Reminders" (user can configure)
- Notification content: title, time, snooze/dismiss actions
- Notification priority: HIGH (heads-up display)
- Sound and vibration based on channel settings
- Exact alarm permission for precise timing (SCHEDULE_EXACT_ALARM)

```
SPEC: Use AlarmManager with setExactAndAllowWhileIdle()
```

[REQ-REM-011] Snooze functionality

- Snooze options: 5 min, 15 min, 1 hour, 3 hours
- Snooze from notification action button
- Snooze from app when viewing reminder
- Show "Snoozed until X" in reminder list

7. Feature Specifications — Data Management

7.1 Export

[REQ-DATA-001] Export all data

- Export to JSON (machine-readable, for backup/restore)
- Export to ZIP (JSON + any future attachments)
- JSON schema versioned for forward compatibility
- Include: notes, tasks, reminders, tags, settings
- Filename: jar-backup-YYYY-MM-DD.json
- Use Android Storage Access Framework (SAF) for save location

[REQ-DATA-002] Export notes to Markdown

- Export all notes as individual .md files in ZIP
- Filename: note-title-slug.md
- Include YAML front matter: title, created, modified, tags
- Body as Markdown content

7.2 Import

[REQ-DATA-003] Import from JSON backup

- Validate JSON schema before import
- Options: Replace all data, or Merge (keep existing + add new)
- Show preview: N notes, N tasks, N reminders to import
- Handle schema version migration gracefully
- Show success/failure count after import

7.3 Data Management

[REQ-DATA-010] Clear all data

- Accessible from Settings > Data > Clear All Data
- Two-step confirmation: dialog + type "DELETE" to confirm
- Secure wipe: overwrite before delete
- After clear: return to fresh state (onboarding optional)

[REQ-DATA-011] Storage statistics

- Show in Settings: total notes, tasks, reminders, tags
- Show database size on disk
- Last backup date (if ever exported)

8. Feature Specifications — Settings

8.1 Settings Screen Layout

[REQ-SET-001] Settings organization

- Use Material 3 settings patterns (PreferenceScreen or manual)
- Group into sections with headers
- Section: Appearance (theme, color)
- Section: Notes (default sort, etc.)
- Section: Tasks (default list, etc.)
- Section: Notifications (reminder settings)
- Section: Data (export, import, clear)
- Section: About (version, licenses, links)

8.2 Appearance Settings

[REQ-SET-010] Theme setting

- Options: System default, Light, Dark
- Applied immediately on selection
- Persisted across app restarts

[REQ-SET-011] Dynamic color toggle

- On Android 12+: toggle to enable/disable Material You
- When disabled: use default app color scheme
- Hidden on Android 11 and below

8.3 About Screen

[REQ-SET-020] About screen content

- App icon and name
- Version name and version code
- "A local-first notes app. Your ideas in a jar."
- Links: GitHub repository, Privacy Policy, Licenses
- Open source licenses (OSS licenses screen)
- "Made with ♥ by Erich Benjamin Hubert"

9. Technical Architecture Requirements

9.1 Architecture Pattern

[REQ-ARCH-001] MVVM with Clean Architecture layers

- Presentation Layer: Activities/Fragments/Compose, ViewModels
- Domain Layer: Use cases (optional for v1.0, prepare structure)
- Data Layer: Repositories, DAOs, Entities
- Dependency direction: Presentation → Domain → Data

`SPEC: Single-activity architecture with Jetpack Navigation`

9.2 Database

[REQ-ARCH-010] Room database

- Database name: jar-database
- Entities: Note, Task, Reminder, Tag, NoteTagCrossRef, TaskTagCrossRef
- All queries as Flow (reactive)
- Proper indices on foreign keys and frequently queried columns
- Migration strategy: addMigrations() for schema changes
- fallbackToDestructiveMigration() NEVER in production

`SPEC: Export schema to JSON for version control (exportSchema = true)`

9.3 Dependency Injection

[REQ-ARCH-020] Hilt for DI

- @HiltAndroidApp on Application class
- @AndroidEntryPoint on Activities/Fragments
- @HiltViewModel on ViewModels
- DatabaseModule provides Room database and DAOs
- RepositoryModule provides repository implementations

9.4 Concurrency

[REQ-ARCH-030] Coroutines for async work

- ViewModelScope for UI-related async work
- Dispatchers.IO for database operations
- Dispatchers.Default for CPU-intensive work
- Never block the main thread
- Proper cancellation handling

10. Performance Requirements

[REQ-PERF-001] App startup

- Cold start to interactive: < 500ms on Pixel 4a class device
- Warm start to interactive: < 200ms
- Hot start: < 100ms
- Use AndroidX Startup library for init optimization
- Measure with Android Vitals / Macrobenchmark

```
SPEC: Use Baseline Profiles for improved startup
```

[REQ-PERF-002] Frame rate

- Maintain 60fps during all scrolling
- No frame drops during navigation transitions
- No jank during list item animations
- Measure with GPU profiling tools

[REQ-PERF-003] Database query performance

- All list queries: < 50ms for up to 10,000 items
- Search queries: < 100ms for full-text search
- Insert/update: < 10ms per item
- Use EXPLAIN QUERY PLAN to verify index usage

[REQ-PERF-004] Memory

- Peak memory usage: < 100MB in typical use
- No memory leaks (verify with LeakCanary in debug builds)
- Proper lifecycle handling (no retained references)

[REQ-PERF-005] APK size

- Release APK: < 10MB
- Use R8 full mode for shrinking
- Remove unused resources
- Use Android App Bundle for optimized delivery

[REQ-PERF-006] Battery

- Zero background battery usage when app is closed
- Reminders use AlarmManager, not persistent services
- No wake locks except for alarm delivery

11. Accessibility Requirements (WCAG 2.1 AA)

[REQ-A11Y-001] Screen reader support

- All interactive elements have contentDescription
- Meaningful descriptions (not "button" but "Add new note")
- Proper heading hierarchy for TalkBack navigation
- Live regions for dynamic content updates
- Test with TalkBack enabled

[REQ-A11Y-002] Visual accessibility

- Minimum contrast ratio: 4.5:1 for normal text, 3:1 for large text
- Support system font scaling up to 200%
- UI remains functional at largest font size
- Don't rely solely on color to convey information
- Support system bold text setting

[REQ-A11Y-003] Motor accessibility

- Minimum touch target: 48x48dp
- Support Switch Access navigation
- No time-limited interactions (except snackbar with adequate duration)
- Support keyboard navigation (for tablets with keyboards)

[REQ-A11Y-004] Motion and animation

- Respect "Remove animations" system setting
- No essential functionality depends on animation
- Avoid rapid flashing or strobing effects

12. Security Requirements

[REQ-SEC-001] Data storage

- All data in app-private internal storage
- Database not accessible to other apps
- No world-readable files
- SharedPreferences in MODE_PRIVATE

[REQ-SEC-002] Export security

- Exported files contain no device identifiers
- No android_id, IMEI, or other unique IDs
- User warned that exports contain personal data
- Optional: encrypted export with user-provided password (v1.1)

[REQ-SEC-003] Secure deletion

- When user deletes data: remove from database immediately
- Consider overwriting sensitive fields before delete
- Clear all data function: VACUUM database after delete

[REQ-SEC-004] No network

- AndroidManifest.xml: NO internet permission
- Verify with network traffic inspection (no outbound connections)
- No analytics SDKs (Firebase, etc.)
- No crash reporting SDKs (Crashlytics, Sentry, etc.)

13. Testing Requirements

[REQ-TEST-001] Unit tests

- ViewModels: 80%+ coverage
- Repositories: 80%+ coverage
- Use cases (if implemented): 90%+ coverage
- Utility functions: 100% coverage

[REQ-TEST-002] Integration tests

- Database: test all DAOs with in-memory database
- Migrations: test each migration path
- Repository + DAO integration

[REQ-TEST-003] UI tests

- Critical user flows with Espresso or Compose testing
- Create note flow
- Create task flow
- Search and filter flow
- Export/import flow

[REQ-TEST-004] Device testing

- Test on minimum API (26 / Android 8.0)
- Test on target API (35 / Android 15)
- Test on at least 3 screen sizes (phone, small phone, tablet)
- Test with different system themes and fonts

14. Release Checklist

All items must be checked before v1.0.0 release:

- ■ All REQ-* requirements in this document implemented
- ■ Zero crash bugs in Crashlytics-free testing (manual QA)
- ■ Zero data loss scenarios identified
- ■ All UI screens reviewed in both light and dark themes
- ■ All UI screens reviewed with TalkBack enabled
- ■ All UI screens reviewed at 200% font scale
- ■ Performance benchmarks met (Section 10)
- ■ Security audit passed (no network, proper storage)
- ■ Unit test coverage targets met
- ■ README.md complete with screenshots
- ■ CHANGELOG.md up to date
- ■ Privacy policy written and hosted
- ■ AGPL-3.0 LICENSE file in repository
- ■ Signed release APK builds successfully
- ■ APK size under 10MB
- ■ F-Droid metadata prepared (optional for v1.0)
- ■ Play Store listing prepared (optional for v1.0)
- ■ GitHub Release created with APK attached

15. Out of Scope for v1.0.0

These features are explicitly NOT included in v1.0.0:

Cloud sync: Conflicts with local-first principle. May consider encrypted P2P later.

AI features: No auto-tagging, summarization, or smart suggestions.

Voice notes: Audio recording deferred to v1.1.

Attachments: Images, files, etc. deferred to v1.1.

Widgets: Home screen widgets deferred to v1.1.

Budgeting module: Remains separate app (Based Budgeting).

Collaboration: Sharing with others conflicts with privacy model.

Calendar integration: System calendar sync deferred.

iOS version: Android-first. iOS considered after v1.0 stability.

Desktop version: Linux desktop planned for v2.0.

Wear OS: Smartwatch app is v2.0+ consideration.

16. Success Metrics

v1.0.0 will be considered successful when:

16.1 Quality Metrics

- Crash-free rate: 99.9%+ (measured via opt-in feedback)
- ANR rate: < 0.1%
- Zero data loss reports from users
- Cold startup p90: < 600ms

16.2 User Satisfaction

- User can complete blind comparison with Google Keep and not identify which is third-party
- Positive feedback from privacy-focused communities (r/privacy, r/fossdroid)
- F-Droid inclusion accepted (if submitted)

16.3 Adoption Metrics

- GitHub stars: 100+ in first month (stretch: 500+)
- Downloads from GitHub releases: 1,000+ in first 3 months
- At least 1 external contributor PR

— End of Document —

This document is version-controlled alongside the codebase.
Last generated: January 2026