naast

MVP 1.0 / 11 SEPTEMBER 2026

The plan. The little moments. The first release.

Two homes. One continuous life for the child.

A consolidated product scope and implementation plan, based on MVP 4, the selected B3 direction and current competitor research.

> **Build baseline · scope update.**
> Dani agrees with the remaining scope. Quick, small clips are now planned; exact media limits, pricing and the cofounder's layout choice remain open.

01 / Assessment and development tradeoffs

02 / Launch scope and acceptance gates

03 / Delivery plan and historical estimate

04 / Native app and implementation sessions

Scope: phone app first · web companion · two parents · multiple children · photos, text and short clips
English pilot · no production implementation claimed · source Markdown in docs/

PART 1 / ASSESSMENT

MVP 4 assessment and development recommendation

11 September 2026 · NAAST · Updated after Dani’s agreement, short-video clarification and phone-app-first direction

Delivery update: the product is now explicitly phone-app first; web is a companion. The feature assessment below still applies. Native implementation, push and beta distribution are covered by document 13; the original web estimate is historical.

We are on track. MVP 4 preserves the right core and gives the commercial argument a proper place. The additions make sense when each has a bounded implementation. The main risk is the meaning hidden inside labels such as “light”, “simple”, “questions”, “short video” and “holidays”. This assessment resolves those labels into a buildable baseline in 10 — Final MVP scope, with estimates and sequencing in 11 — Implementation plan.

Reviewed inputs: all ten pages of `NAAST_MVP_4_Complete_Product_and_Functional_Design_EN (1).pdf`, the two original concept PDFs through the existing assessments, the September 10 memory decision, the current demo, and the founder messages. B3 / Side by side is explicitly selected. M1–M4 are memory explorations; M3 is now Dani’s preference, with the cofounder’s choice pending.

What the competitors actually establish

Official public pages checked on 11 September 2026. These are advertised capabilities, not hands-on verification of quality, security, usage or commercial success. No accounts were created.

Product	Relevant public evidence	Implication
Duonido	Explicitly advertises a beta, care scheduling, iCal export, handover notes, requests, a journal and a broad administrative offering. Official site	The broad feature list is already occupied. Stay disciplined about the quality and coherence of NAAST's experience.
TweeHuizen	Advertises care patterns, holiday proposals, activity packing lists and family roles. Its site links a released iOS app; the App Store shows a version history. Official site, App Store	Planning and packing are credible user expectations, not unique selling points. Do not assume every competitor is still a beta.
BeNice	Describes communication coaching/filtering, CheckPoint, solo use and a Photo Diary. Feature page	Photo memories and open questions already appear in the category. NAAST needs a clear reason to prefer its combination of features.

TweeHuizen's support material also describes calendar export and backup; this does not establish that it offers the same subscription-feed implementation proposed for NAAST. Support. BeNice's site offers app download/trial routes and mixes current features with “new or coming soon” material. Its precise maturity and feature availability were not independently established. BeNice home.

The opportunity is a calm, child-centred service where the practical plan, unresolved coordination and family memories fit together. That is a positioning and execution hypothesis. A memory timeline can contribute to retention even when it is not unique. Accumulated history should remain portable and useful to the family.

Assessment of the additions

Relative complexity below is incremental to a functioning, authorized family/calendar platform. It is not the total effort to build the feature from an empty repository. Numerical planning allowances and assumptions are in the implementation plan.

Addition	Development judgment	Launch boundary
Open actions on Today	Low for aggregating existing states; medium when adding explicit questions. High if detecting questions and answers automatically in free text.	Show requests requiring my response, questions addressed to me, and near-term packing. Parents create and answer questions explicitly. No AI extraction or general task manager.
Packing lists	Low–medium. Recurrence and checklist state are the important parts.	Reusable item definitions; separate check state for each activity or handover occurrence. Parents select relevant activities when preparing a handover.
Richer child profiles	Low–medium for practical fields and contacts. Sensitive health information changes the risk and review effort.	School/class, sport, practical notes and optional practical contacts. Initials by default, optional photo through the memory media service. No allergy/medication fields or clinical records.
Photo/text memories	Medium–high as a complete production feature. The timeline layout is the easy part.	One optional photo, text, occurrence date, child tags, category and milestone; private media delivery, reliable capture, retrieval and export.
Short video	Medium–high incremental impact with a managed service. Small output files control storage; processing, phone formats and playback still need validation.	Plan quick clips within F08, potentially available on a free tier. Test roughly 2–3 MB processed clips and a short duration; exact limits remain proposals.
iCal file export	Low–medium after calendar correctness. Useful for portability.	Include as a fallback, clearly labelled as a snapshot.
Read-only iCal subscription	Medium, commercially more useful for avoiding re-entry. Requires private tokens, stable event identity, updates/cancellations and client testing.	Include a bounded one-way feed. NAAST stays the place to make and verify changes. No two-way integration.
Holidays and exceptions	High-value core complexity, not cosmetic calendar decoration.	Explicit dated overrides, partial days, per-child schedules and conflict review. No automatic holiday allocation or maintained school-holiday database.

Microsoft documents the important distinction: importing a file does not keep it updated, while a subscription refreshes from its source and may take more than 24 hours to reflect changes. A feed therefore reduces re-entry but cannot promise immediate handover updates. Microsoft's import/subscription guidance.

Answers to MVP 4's development questions

The source PDF numbers these questions 11–20 on p. 10.

11. Memory Timeline light? Photo/text capture and bounded short clips, with dependable media operations, belong in the revised launch plan. A milestone and category are fields on the same entry. Video is a separate media capability, not a small extension of the photo picker. The existing demo is visual and behavioural reference; it implements no durable storage or permission enforcement.

12. Export or subscribe early? Yes. Build a standards-based read-only feed and a downloadable snapshot from the same agreed-calendar projection. Validate Apple, Google and Outlook behaviour before the pilot. Expanding a finite date range into individual events keeps external recurrence simpler; it does not eliminate the need for stable IDs or cancellation tests. The iCalendar specification defines event identity and recurrence properties. RFC 5545.

13. Shared packing model? Yes: use reusable item definitions and occurrence-specific checklist entries. “Swim goggles” can seed several lists, while Friday's checkmarks remain independent of next Wednesday's. A shared handover retains the labels and state it was shared with, even when a template later changes.

14. Robust recurrence? A versioned 7- or 14-day pattern per child, anchored to a date and local timezone, plus explicit approved interval overrides. An override replaces the affected care interval; conflicting overrides need resolution rather than silent priority. Preview the resulting dated plan before approval. Care patterns, activity recurrence and holiday reference labels are related but distinct concepts.

15. Solo separation? Enforce owner/workspace visibility on the server. Solo planning is a personal draft, not the absent parent's agreement. An invitation has an explicit sharing preview. The recipient's membership acceptance grants only the disclosed access; confirming a shared starting plan is a separate action. Unselected solo memories remain private.

16. Media limits? Use configurable pilot guardrails rather than announcing freemium tiers: one attachment per memory, 20 MB photo input, 60 MP decoded-image limit and 2 GB of stored media per family, including derivatives. These are proposed engineering limits, not previously agreed commercial terms. Preserve a sanitized master and generate display sizes. Measure actual sizes and retrieval traffic; short clips now have a separate provisional encoding/playback work package and share the family’s media allowance. See F08 for proposed source-versus-output limits; freemium pricing is not settled. See the explicit cost model in the implementation plan.

17. What belongs in history? Care-pattern and override versions; proposals/responses; activity and responsibility changes; shared handovers; profile updates; sharing, membership and export events. Keep memory authorship/edit metadata separately. Operational logs record identifiers and outcomes, not duplicate message bodies, photographs, calendar tokens or private notes. No court-proof or indefinite-retention claim.

18. Privacy and account removal? Export/access, correcting data, deleting own memories, revoking an invitation/feed, leaving a workspace and requesting account closure need executable workflows before real families join. Two connected parents have equal membership; one cannot silently evict the other or wipe the shared archive. Shared-record disputes and legal retention decisions require a documented support procedure and review. This is product policy design, not a legal determination of ownership or rights.

19. What can be reused? B3's visual tokens, navigation, labels and tested interaction patterns; practical layouts; memory capture/detail patterns and licensed assets for demo/testing. Rebuild the static string templates as maintainable components using one production component system. Missing designs include recurring-pattern editing, holidays/conflicts, sharing transition, open-question states, retry/error states, feed settings, archive export and account departure. The three additional UI kits are exploration references, not a requirement to ship four frameworks.

20. Time and cost? Use the implementation plan's module ranges as historical web/domain inputs, then re-estimate the native-first delivery after the scaffold and device spike. The estimate is for a real family service with permissions, shared state and recoverable operations, not another static demo. Separate engineering work, contingency, outside review and pilot observation. Small-clip work is included provisionally; paid checkout remains excluded. No supplier rate, budget or delivery date has been agreed.

Changes needed to make MVP 4 a fixed scope

- Replace "Launch if feasible / early beta" with explicit inclusion: all six bounded commercial additions above are part of the defined pilot product; short video is bounded and included in the revised plan; automatic highlights remain outside it.
- Keep Today as the practical home and Memories as a primary destination. A memory teaser provides continuity; the M1–M4 visual choice does not change the data or permission contract.
- Separate informational activities, practical questions and plan-changing proposals. Answering a question must never change care time.
- Treat account access, privacy operations and notification infrastructure as work that starts with the foundation. Final hardening verifies them; it should not discover them at the end.
- Describe the first release as an invite-only, real-data MVP beta. Paid public launch additionally needs pricing, billing, subscription expiry policy and operational capacity. Feature differentiation and a monetization implementation are separate decisions.
- Use memory-retention data carefully. Families who voluntarily add several memories may already be more engaged. Compare cohorts and qualitative reasons; do not present that correlation as proof of causation.

The resulting scope is wider than the first coordination-only recommendation, deliberately so. It is still bounded: shared planning, explicit coordination, practical child context and a useful memory collection, with tightly bounded video processing and without AI, financial administration or multiple adult role systems.

PART 2 / BUILD SCOPE

NAAST MVP 1.0 — Consolidated build scope

Version 1.0 · 11 September 2026 · Phone app first · Web companion · B3 / Side by side

Platform update: Dani clarified that the phone app is the primary product; web is a companion and testing surface. The earlier web-first/native-later delivery assumption is superseded. The working pilot assumption is iPhone first, Android-ready, pending platform coverage confirmation. See the native kickoff.

Status: Dani accepted the remaining scope on 11 September 2026 and asked to retain small, quick videos for a possible freemium allowance. Short clips are now planned within F08 rather than categorically deferred. Exact clip limits, pricing and provider fit remain to be validated; the cofounder’s final memory-layout choice is still pending. This document is the current scope reference, superseding the launch recommendations in documents 01, 02, 04 and 07 wherever they differ.

Source: the supplied ten-page MVP 4 document, the earlier concepts and agreed design work. Assessment and rationale · Implementation steps and estimates.

---

1. Product and release definition

Two homes. One continuous life for the child. NAAST helps parents know the plan, resolve practical coordination and preserve the moments of their child's life across households.

The MVP is an invite-only, production-backed beta for approximately 20 families, with 15–25 as the recruitment range. It must work with real accounts, durable shared records, reliable permissions and recoverable failures. The current public prototype is not that service.

Launch includes the functional scope below, with memories present when pilot families start. This tests practical usefulness and accumulated personal value together. Retention, differentiation and willingness to pay remain hypotheses.

The first pilot is free. One family-level entitlement can be administered for the pilot; payment collection, subscription tiers and freemium pricing are outside this build baseline. A paid public release requires a separate billing and commercial-readiness increment. This boundary must be included in any quote.

Platform: installable phone app first, with a secondary responsive web companion. Recommend React Native/Expo, using iPhone as the initial design/device reference and supporting Android through shared contracts and platform-specific testing. Whether Android joins the first pilot is still open. Native development builds, beta distribution and native push are now delivery workstreams. English is the pilot interface, as agreed for the prototype; organize copy for translation. Full offline editing remains outside launch. Public store release follows an installable beta and operational gates; the current HTML review is not the native app.

---

2. Fixed launch boundary

CORE and COMMERCIAL explain why an item exists. Both are included when marked Launch; COMMERCIAL does not mean optional work hidden in the estimate.

ID	Capability	Rationale	Release decision
F01	Accounts, family, children, two households, solo start and invitation/sharing transition	Core	Launch
F02	Home: children, memory capture/revisit, agreed plan and attention shortcuts	Core	Launch
F03	Care patterns, activities, holidays and dated exceptions	Core	Launch
F04	Plan-change requests, acceptance, decline, counter, withdrawal and history	Core	Launch
F05	One text conversation and explicit practical questions	Core + commercial	Launch
F06	Handover note, checklist and reusable activity packing lists	Core + commercial	Launch
F07	Practical child profile, activities and optional contacts	Commercial	Launch
F08	Photo/text/short-clip memories, child/category filters and milestones	Commercial	Launch
F09	Read-only calendar subscription and snapshot download	Commercial	Launch
F10	In-app/native push, email fallback, reminders and personal quiet hours	Core	Launch
F11	Necessary history, privacy operations, export, account departure and support	Core	Launch
F12	Instrumentation, error recovery, operational monitoring and pilot readiness	Core	Launch

Explicitly outside launch: long-form video, video editing, standalone audio capture, AI of any kind, automatic question detection, generated highlights, anniversary reminders, photo books, public memory links, extended-family roles, child accounts, finances/payments between parents, document vaults, two-way integrations, school-feed parsing, medical records/BSN, legal or professional portals, care-distribution scores and general task/project management.

Changes from MVP 4: video is bounded to quick, small clips; question tracking is explicit; holiday dates are manually entered; health-specific fields are excluded; the calendar feed is included with a defined contract; in-app/native push with email fallback are the launch notification channels. Dani agrees with the other scope resolutions. Clip inclusion is the revised direction; numerical media allowances remain configurable working proposals.

---

3. Design and navigation

Use **B3 / Side by side**: ink-blue navigation, yellow selected state, clear child context and practical hierarchy. Preserve accessible labels alongside colour.

Home centres the children and their memories: child profiles, a prominent Add a memory action and a recent accessible moment. The next agreed handover and incoming attention shortcut remain immediately reachable. Memories stays a primary destination; practical details continue below the first view. This hierarchy follows

Dani’s child/memory-centred direction and needs parent task testing.

The primary phone app uses five destinations: Home, Calendar, Memories, Messages and More. More exposes Requests (including the attention count), Children, Handover and Settings. Requests are also directly reachable from Home and linked messages. The secondary wide web view has six destinations. The revised phone preview separates review controls from app content and uses day/week agendas instead of a horizontally scrolling desktop calendar. Production notification counts and all failure states still need implementation.

M3 / Chapters is Dani’s current preference within B3; the cofounder’s choice is pending. Estimate one collection with deterministic month chapters, a milestone chapter, chronological entries and the shared add/detail flow. M1, M2 and M4 remain review alternatives. No custom album editor, automatic storytelling or generated covers are implied. These presentations use the same memory records; build one browsing experience. The phone refinements and research are recorded in the mobile design review and the child-centred intuitiveness review.

---

4. F01 — Family, access and solo start

One family workspace per account in this pilot; up to two connected adult parent accounts and multiple child profiles. Model memberships explicitly so later roles do not require replacing ownership checks. No extended-family permissions UI.

Account flows: registration, email verification, sign-in, recovery, sign-out/session revocation and account closure request. Invitations are bound to the intended recipient, single-use, revocable and expire after a configurable period; seven days is the proposed pilot default.

Solo users can create child profiles, a personal care-plan draft, activities, packing lists and memories. The other household is a schedule label, not an active account. Shared messages/questions cannot be delivered until a second parent joins. No email-to-chat bridge or pretend agreement by an absent parent.

Before inviting, show a preview of the specific information that will become visible. The invitation carries a versioned disclosure selection. If the selection changes, obtain a new explicit confirmation; do not silently expand access. Solo memories are private until individually or explicitly batch-selected for sharing. Solo private notes are never silently promoted.

Joining grants the disclosed access. It does not approve the initial schedule. Both parents review and explicitly confirm a shared starting plan; until then it is labelled a shared draft. Keep the underlying personal records and their provenance distinct from confirmed shared records.

Two connected parents have equal membership. Either may leave or request support; neither can unilaterally remove the other parent, take over their identity, or delete the family workspace. Pending invitations can be revoked by their sender. Access-dispute handling is a supported operational workflow before the pilot.

---

5. F02–F03 — Home and the agreed plan

Home leads with child context and memory capture/revisit, plus compact next-handover and attention shortcuts. Its practical section includes scheduled household per child; next handover with time/location/collector; relevant activities; incoming requests/questions requiring my response; packing due soon. Outgoing requests waiting on the other parent are visually separate. No open items produces a calm empty state, not pressure to create activity.

The screen supports children at different households, partial-day handovers, no activities, stale data and loading/error states. “With Sanne” describes the agreed schedule, never live tracking.

Calendar offers a week view, month overview and a selected-day agenda/detail. These share the same events and state; there is no separate resource-booking day-grid engine. Parents can view their children together or filter a child.

Care recurrence is a versioned 7- or 14-day template anchored to a start date, with named households and timed boundaries. Presets such as alternating weeks and 2-2-3 populate that template. No arbitrary recurrence-rule editor. A new pattern takes effect from a chosen future date without rewriting past history.

Holidays, Christmas, birthdays and school closures can be entered as named date ranges or activities. A reference label does not assign care. Actual departures from the regular care pattern are explicit dated overrides per child, with partial days supported. Automatic school-region calendars, equal-share calculations, fairness scores and recurring annual holiday-allocation rules are excluded.

The effective plan is the base pattern plus accepted interval replacements. Overlapping accepted overrides cannot be silently stacked; a new proposal must identify the conflicting interval and show the resulting plan for review. Review gaps, overlaps, collection responsibility and nearby handovers before acceptance. Unassigned time remains visibly unresolved.

Activities contain child/children, title, occurrence time or all-day date, location, optional note, drop-off/pick-up responsibility and packing items. Activity recurrence is weekly or fortnightly with an end date, plus single-occurrence edits/cancellations. “This and future” editing closes one series and starts a versioned replacement after a preview.

Parents may publish ordinary informational activities with attribution and notification; that does not change care allocation or commit the other parent to transport. A parent's own transport assignment can be recorded directly. Assigning the other parent, changing an obligation they have accepted, changing care time, or changing an agreed handover requires a request. Informational event edits never bypass those checks.

Store instants and the relevant IANA timezone, while defining repeating rules in local wall time. The family default is Europe/Amsterdam; each user can set a display timezone. All-day dates stay dates. DST transitions, cross-midnight care and interval boundaries are mandatory test cases.

---

6. F04 — Explicit requests and decisions

Launch request types: care interval/pattern change, holiday override, handover logistics change and other-parent activity transport assignment/change. These are structured practical requests. General legal, medical, school-choice and travel-consent decisions are outside this workflow.

Show the current agreed values, proposed values, affected children and dates, proposer, addressee and optional reason. Only the addressed connected parent can accept. The proposer can withdraw a pending request. A counterproposal closes that version as countered and creates a linked pending proposal in the opposite direction.

Statuses: pending, accepted, declined, countered, withdrawn and expired. For the pilot, a pending request expires when the earliest affected change would take effect (for a handover-time change, the earlier of the old and proposed times). Past changes cannot be accepted retroactively; expiry never implies approval. Editing a proposal means withdrawing/replacing its version rather than modifying what the recipient has already reviewed.

Acceptance checks the current affected planning version and applies the change plus history atomically. A conflicting intervening change yields a stale-request state and a fresh preview. Duplicate submissions are idempotent. Chat agreement, reading, silence and answering a practical question never count as acceptance.

After an accepted response, all NAAST views refetch the affected data. Other active clients refresh promptly through a bounded refresh mechanism; the pilot target is within 30 seconds while online, with immediate refresh on focus. Network failure is visible. An external calendar may update later.

---

7. F05 — Messages and explicit open questions

One family text thread with sender and sent time, older-message pagination, links to requests and clear pending/sent/failed states. “Sent” means accepted by the server, not read by another person. Retry cannot duplicate a message. Attachments, reactions, typing status, read receipts, full-history search and message editing are excluded. Privacy/support removal procedures still apply.

A parent can create an explicit practical question, optionally linked to a message, child, activity or handover. It has an addressee, text, optional requested reply date and one simple answer field. States are open, answered and closed/withdrawn. Answering records actor/time; the creator can reopen with an explanation or close it. It does not mutate the calendar.

Examples: “Is Finn's reading book at your house?” is a question. “Can you collect at 18:00 instead of 17:00?” belongs in a plan-change request. The UI provides a route to create a request without treating the question's answer as approval.

Home aggregates these records; it does not store a second task list. No automatic text analysis, auto-completion inferred from chat, escalation ladder, parent responsiveness score or general recurring tasks.

---

8. F06 — Handover and packing

A handover is tied to the agreed care transition and affected children. Show the agreed time/location/collector, a short practical note and a packing checklist. Draft and shared states are distinct; sharing records author/time/version. Amendments produce a visibly updated shared version.

Packing definitions can be reused between an activity and a handover. Check state belongs to a specific occurrence: packing a swimming bag today does not pack it next week. Include item label, optional child/source activity, completion and last-change metadata. No household inventory, permanent possession tracking or parent-performance scoring.

When preparing a handover, offer activities during the upcoming care interval and let the parent select which item lists to include. Preserve source references, handle duplicate items visibly and snapshot the selected item labels. Later template edits do not alter a handover already shared. An accepted schedule change prompts review of a prepared handover instead of silently moving its contents.

Receiving-parent acknowledgement, automatic summaries and automated cross-household packing inference are outside launch.

---

9. F07 — Child context

Name, optional birth year, initials/avatar, optional profile photo, school/class/location, sport or hobby, relevant recurring activities and a short practical note. All nonessential fields are optional. Link directly to the child's calendar, packing and memories.

Practical contacts have a name/organization, role, phone and optional email. Examples are a school office, coach, GP/dentist practice or trusted emergency contact. These contacts are information entries, not invited app members. Clinical notes, allergy/medication fields, medical attachments and BSN are outside this baseline;

adding them needs a separate product/privacy decision. Do not encourage sensitive information in free-text hints.

Both connected parents can update shared practical fields; updates have attribution, version checks and history. Solo private notes remain separate. Profile-photo changes use the same validated private image service as memories.

---

10. F08 — Memories at launch

One entry contains an optional title, one optional photo or short video, optional written story, occurrence date, one or more child tags, category, author and visibility. Require a written story or one media attachment; a title alone is insufficient. For an untitled entry display a short story excerpt or “A little moment”, without treating that fallback as an authored title. Categories: moment, milestone, birthday, school, sport, holiday and other. One attachment per entry is the first implementation boundary; mixed galleries, multi-photo posts and bulk import are later options.

The collection is chronological by occurrence date, both across children and per child, with month and category filtering, incremental loading, detail and own-entry editing. Creation/edit timestamps remain distinct from the occurrence date. Text-only memories are first-class. Milestones use the same record model. No AI tagging, public links, comments, reactions, autoplay, generated highlights or anniversary notifications.

In a connected workspace, new shared memories clearly name the audience before saving. Solo entries remain private until explicitly shared. Author changes and audience expansion cannot be hidden. Authors can edit or remove their own entries; another parent cannot edit them or use ordinary controls to delete them. A removal request concerning another person's shared record goes through the support procedure. Product authorship is attribution, not a legal determination of ownership.

Upload states: selected/preview, uploading, processing, saved and failed/retry. A record is not presented as saved while its media is unusable. Handle disconnects, duplicate retries, corrupt files, unsupported media, excessive size and a full allowance. Keep text recoverable in the active session if an upload fails.

Proposed pilot guardrails: JPEG, PNG, WebP and HEIC inputs; one photo; 20 MB source size; 60 MP decoded size; 2 GB per family across stored photos, processed clips and derivatives. These are configurable engineering allowances for the free pilot, not freemium or paid-plan promises. Validate HEIC handling with real iPhone files during the foundation spike; a failed spike is a scope issue to resolve before committing to a provider.

Preserve a high-quality sanitized master, correct orientation, remove unnecessary location/device metadata and create responsive previews/thumbnails. Explain any format conversion and quality change. Serve images through authenticated access or short-lived authorized links, with no public bucket. Check actual content and decoded dimensions, not just filename/type headers. OWASP's file-upload guidance supports layered validation and controlled storage. OWASP.

At 80% of the proposed allowance show a quiet warning; at the limit stop new media uploads while preserving text capture, reading, download and account-rights access. Quota checks include concurrent uploads and cleanup. No real family history is deleted to make room without an explicit policy and action.

Short clips are planned for MVP 1.0 and may be available on a future free tier. Dani suggested “a couple MB”; this is a product direction, not a finalized quota. Keep the existing free pilot separate from eventual freemium pricing.

Working target for the media spike: one clip of roughly 10–15 seconds, with a processed file around 2–3 MB and up to 720p. These numbers are proposals. Interpret the small file allowance as the processed clip, so parents need not manually compress camera originals; validate that interpretation and the quality with sample families. Use a separate, bounded source upload limit (50 MB as an initial engineering test value), duration/pixel/frame-rate limits and per-family processing limits. No unlimited original uploads or unlimited retry jobs. Freeze exact caps after actual iPhone/Android samples and cost measurements.

Use an established managed encoding service if suitable. Produce one broadly compatible playback file and poster image, preserve orientation and intended audio, strip unnecessary location/device metadata, and show a play control and duration. No autoplay, editing/trimming UI, live streaming, custom adaptive-streaming stack or promise to retain camera originals. A compressed download/export must be available; disclose the quality conversion and remove temporary sources after processing under a defined cleanup policy. Failed jobs must also expire their temporary sources.

Video uses the same audience, authorship, deletion and export rules as photos. Poster images and processing callbacks must respect family access; reserve quota before processing and reconcile it against actual stored bytes. A failed clip must leave the written memory recoverable. The launch gate includes real-phone upload, processing, sound/orientation, authorized playback, export and deletion. If the spike misses quality, cost or compatibility targets, explicitly revisit the clip boundary rather than silently dropping it.

File size alone does not determine playback compatibility or quality; codecs, source motion and encoding settings also matter. MDN’s codec guide.

---

11. F09 — Calendar outside NAAST

Include a revocable read-only subscription URL per parent, plus a date-range `.ics` download generated from the same calendar projection. Scope the feed to the parent's authorized workspace/children. Default window: 30 days back and 365 days ahead, displayed in settings. No incoming calendar import or two-way editing.

Export only agreed care/handover/transport commitments and published informational activities, clearly distinguished in event titles. Pending proposals, questions, messages, notes, contact details and memories are never exported. Default to minimal labels and no exact addresses. Preview what an external provider will receive; let the parent explicitly opt into descriptive names/locations.

Use stable event/occurrence identifiers and update timestamps. The bounded feed may expand recurrence into concrete instances. Test edited and cancelled instances, changed patterns, DST and rolling-window boundaries in Google Calendar, Apple Calendar and Outlook. An accepted change updates the NAAST feed immediately, but receiving clients control refresh timing. A downloaded file is a snapshot and does not keep updating. Microsoft guidance, Google subscription setup, Apple subscriptions.

The URL is a secret bearer credential: anyone with it can read the selected feed. Show that consequence before enabling it. Store token hashes, redact URLs from logs, support rotation/revocation and invalidate access when membership ends. Revocation stops future service access; it cannot retract files or entries already copied into another provider. Check NAAST for the latest handover when timing matters.

---

12. F10 — Notifications and unreliable connections

Launch channels are in-app notifications, native push on the supported phone platforms, and email fallback. Browser push and SMS remain later options. Ask for OS notification permission in context; denial must not prevent app use. Keep lock-screen payloads generic by default, revoke device tokens on sign-out and refetch

authorized data when a notification opens the app. All channels obey quiet hours, deduplication and stale-event cancellation. Test foreground/background/app-closed delivery and auth return links on physical phones.

Categories: new message, request/response, practical question/answer, shared planning change, upcoming handover/activity and optionally a new memory. Parents control their own preferences. Support a quiet period that crosses midnight. Event reminders use a small set of lead-time choices; queued obsolete reminders are cancelled after edits.

Default notification content is generic. Memory email is opt-in. A daily digest can be enabled; it replaces same-category routine email rather than doubling it. In-app state remains available during quiet hours; outbound notifications wait for the next allowed window. A reminder that becomes irrelevant while waiting is dropped. No emergency override controlled by the other parent.

Deduplicate delivery, retry failed jobs and expose in-app failure/stale states without claiming that email was read. Online access is required for committed mutations; the MVP has no offline conflict-merging queue. Preserve an active form's text on transient failure and revalidate before resubmitting. Explicitly sign out on shared devices; test app background/resume and sign-out explicitly.

---

13. F11–F12 — Trust, support and release quality

Authorization is enforced for every record, media fetch, feed and export. Tenant-isolation and two-parent/solo tests are launch gates. Use EU-region services where supported, encrypted transport/storage, least-privilege administration, monitored failures, rate limits, restore-tested backups and an executable incident/support process. Hosting location alone is not a privacy assurance.

Keep necessary planning/request history with actor, time, affected version and before/after. Record shared handover versions, practical profile changes and membership/sharing events. Memory edits use authorship metadata, not a parallel legal evidence archive. Operational logs omit content, access tokens and media. Final retention periods and shared-record rights handling require a documented review before admitting real families; this specification does not assert legal compliance.

Users can request an export of their authorized family records and obtain a usable ZIP containing permitted photo masters and structured dates/captions/child associations, plus calendar/messages/requests and practical records in readable formats. Exclude the other parent's private entries and all credentials. Exports are authenticated, expire after a proposed 24-hour download window and are invalidated on loss of access. A support-assisted job is acceptable for the small pilot if fulfillment is tracked and tested; polished books or legal reports are unnecessary.

Account departure explains what happens before confirmation. Immediately stop the departing account's sessions, feed and memberships; apply the reviewed deletion/retention rules to their private data and shared contributions. A departing parent cannot automatically erase the continuing parent's entire history. A remaining parent cannot gain access to undisclosed private records. Sole-member workspace closure includes its media, derivatives, outstanding invitations and backup-expiry handling.

Provide own-memory deletion, invitation revocation, contact correction, access/export requests, account closure and disputed-record support. Removal propagates to origin storage, derivatives and controlled caches; backup retention and external copies must be explained. There must be a named operator and a tested workflow before pilot admission, not just a link to a privacy page.

Accessibility target: keyboard-operable workflows, visible focus, labelled controls, readable contrast, colour-independent status and comfortable phone targets. Complete physical-device tests of native invitation/auth return links, photo/clip selection, keyboard/scrolling forms, background/resume, push permission and calendar

subscription setup. Include platform text scaling and screen-reader checks. An installable beta build is a launch prerequisite. Instrument minimal product events without sending message text, child names or images to analytics.

---

14. Release acceptance and scope control

Launch gates:

- Two independent parent accounts complete setup, disclose/accept sharing, confirm a baseline plan and see the same resulting care schedule.
- Solo/private records never appear to an unapproved person; permission, stale-request and duplicate-submit tests pass.
- A holiday override and handover-time counterproposal update the agreed calendar, Home, packing context and export correctly, including after reload.
- A practical question appears on the addressee's Home, can be answered and cannot modify care planning.
- Repeated activities have independent packing states; a shared handover retains the correct version.
- Photo/text/short-clip memories survive reload, appear only to authorized people, can be found by child/month, and support recovery, export and removal.
- Calendar subscription changes/cancellations and revocation are tested in the three named clients. Refresh limitations are explained.
- Quiet hours, stale/failed UI, restore, export and account-departure procedures pass. No known critical access, data-loss or agreement-integrity defect remains.

Pilot learning window: four to six weeks after staged onboarding. Recruit 15–25 families, initially admitting 3–5 before the rest. Evaluate task understanding, useful solo activity, invitation acceptance, completed request loops, repeated voluntary memory use and reasons for staying/returning to other tools. Small-cohort results guide iteration; they do not establish causal retention or product-market fit.

B3 and the remaining feature scope are agreed by Dani; phone app first supersedes the previous web-first delivery plan. Small clips are the revised direction; exact media caps and the final memory layout remain open. Scope additions must specify behaviour, dependencies, effort and budget impact; providers, policies, staffing and pricing require separate decisions.

NAAST MVP 1.0 — Steps to implementation

11 September 2026 · Native-first delivery update; prior web estimate retained as historical input

> **Current delivery:** phone app first, web companion. Use document 13 for the stack recommendation, initial installable milestone and AI-session briefs. The earlier web-only total in section 4 does not quote this revised delivery.

This plan implements the consolidated scope. The assessment explains the tradeoffs against MVP 4. The public demo remains a design reference; this document does not claim production work has begun.

1. Immediate next steps

Step	Proposed owner	Concrete output	Completion condition
Review the scope resolutions	Both founders	One annotated copy of document 10	Record Dani's agreement on the remaining scope. Retain short clips in the plan; validate duration, processed/source size and quota proposals separately from freemium pricing. The cofounder's layout choice remains open.
Close the B3 design gaps	Dani + design/product input	One coherent native app design, companion view and state checklist	Review the revised M3 chapter shelf against M1/M2/M4 and choose one memory browse layout. Validate the demo's six-destination desktop/five-destination phone navigation; design the recurrence/holiday editor, initial shared-plan confirmation, open questions, media errors, feed settings and departure/export.
Run short task sessions	Founders + 6–8 parents	Observations and specific corrections	Include solo initiators and invited/less-enthusiastic parents. Test current-vs-proposed time, holiday planning, sharing audience, packing and memory capture; ask visual preference after tasks.
Complete foundation spikes	Engineering	Written decisions with working technical examples	Prove per-family authorization, transactional acceptance, actual iPhone HEIC and short-video processing/playback, bounded job costs and calendar-client subscription behaviour. Choose one production UI stack and managed services.
Turn this baseline into an implementation backlog	Engineering + founders	Issues with scope IDs, acceptance criteria, dependencies and updated estimates	Replace allowances with task-level estimates and an agreed staffing/rate/start assumption. Begin the first shared-state vertical slice.

Scope/design choices that alter behaviour should be resolved before the corresponding module. Freemium pricing can remain open while the native scaffold and first persisted child/memory loop are built. Native device integration and beta distribution are part of this delivery, not a later web-conversion task. Privacy/access review begins with the foundation and must conclude before real-family admission.

---

2. Delivery sequence and evidence

A. Foundation and the first complete shared flow

Set up an isolated production application, staging environment, migrations, CI, configuration/secrets, account service, relational database, private object storage, email delivery, job processing and operational logging. Create the Expo/React Native app shell and a small native B3 component system. Keep the web companion on the same domain contracts. Set up session storage and verified app return links alongside auth, and produce an installable development build early. The existing preview server is not automatically the approved production data host.

Build account verification/recovery, family membership, children/households, invitations, disclosure preview and solo/private separation. First implement a text/photo memory linked to a child, with optional title, explicit audience, retrieval after restart and filtered browsing. Prove access and upload recovery. Then implement one dated care block and one change proposal before building the full calendar editor. Use the shared contract draft as the handoff boundary.

First demonstration: create a child, save a text-only and photo-only memory, restart and reopen each from that child's history. No unrelated-family or private-memory content leaks.

Next demonstration on two app sessions: parent A proposes a change; parent B accepts; both sessions and a fresh reload show the same value. A duplicate acceptance is harmless, a stale acceptance is rejected, and an unrelated family cannot read the record. Together these increments prove both the child/memory centre and safe shared coordination.

B. Reliable planning, then practical coordination

Add versioned 7/14-day care patterns, activities, interval overrides, holidays and all affected request types. Build the week/month/day-detail views over one effective-plan service. Test time boundaries and history before adding calendar polish.

Add one text thread, explicit questions and Home aggregation. Add handover drafts/shared versions, packing templates and independent occurrence checklists. Add practical profile fields/contacts using the established access/version model.

Demonstration: two children follow different care blocks over a holiday; one parent proposes Friday 18:00; a counteroffer is resolved; Home and the next handover update; the packing context is reviewed; a separate reading-book question is answered without changing the plan.

C. Complete memories and calendar portability

Extend the image pipeline and photo/text records established in the first slice with bounded clip processing and playback in the same module. Include filters, editing, deletion and archive jobs for both media types. Retention value depends on reliable saved history; do not postpone export/access design until after media is stored. Reuse the image service for an optional child avatar.

Build the minimal calendar projection, secret subscription tokens and snapshot download. Validate the actual target clients, especially changed/cancelled instances and subscription setup from phones.

Demonstration: an iPhone photo, a short phone-recorded clip and a text-only memory survive reload and are visible only to the intended parent; a corrupt or interrupted upload recovers; exported photo masters, processed clips and metadata are usable. Verify clip sound/orientation, expired processing jobs, quota reservations and deletion of both playable files and posters. A calendar change appears in NAAST immediately and eventually in subscribed clients; revoked tokens stop fetching.

D. Release verification and pilot operations

Native notification permission/token handling and delivery infrastructure are present from the foundation; complete categories, quiet-hour scheduling, digests and obsolete-reminder cancellation once domain events exist. Finish physical-device, keyboard, contrast, accessibility, tenant-isolation and unreliable-network checks.

Exercise restoring a backup, producing an authenticated export, deleting an own memory, revoking invitations/feed links and processing account departure. Finalize retention/access policies, vendor review, pilot information and a named support contact. Build minimal product metrics without recording family content.

Demonstration: execute every launch gate in scope section 14 with evidence, rather than relying on the prototype's browser tests. Do not recruit real families while privacy or data-integrity defects are still unresolved.

E. Staged pilot and iteration

Admit 3–5 families first. Verify account/sharing, schedule correctness, media recovery and support load; then expand toward approximately 20. Give the recruited cohort four to six weeks to use the combined product. Calendar subscription instructions, native notification permission and email fallback expectations belong in onboarding. Distribute the installable iOS beta through the chosen beta channel; test Android before admitting Android families.

Review misunderstandings and support incidents weekly. At the end, decide which problem deserves the next increment: usability, stronger memories, expanded clip allowances, browser push, Dutch translation or integration. Paid launch requires its own pricing/billing/expiry-policy work; do not start charging simply because the feature checklist is complete.

---

3. Engineering shape and important dependencies

The recommended candidate is Expo/React Native/TypeScript with managed PostgreSQL, Supabase Auth/storage and server-side domain transitions. Validate provider fit and device integration before committing to deployment. The HTML design kits supply visual references, not native runtime components. A single application/backend with a relational database, private media storage and background jobs is enough; microservices are unnecessary for this scope. Managed services can reduce infrastructure work, but permission design and data lifecycle remain our responsibility.

Domain	Key objects / invariant
Access	User, Family, Membership, Household, Invitation, SharingSelection; every query derives permissions from authenticated membership/ownership, never a client-supplied family ID alone.
Planning	CarePatternVersion, CareOverride, ActivitySeries, ActivityOccurrence, Responsibility; stable occurrence identity and a single effective-plan projection.
Decisions	RequestVersion, Response, affected planning version; atomic update plus history.

Domain	Key objects / invariant
Coordination	Message, PracticalQuestion, HandoverVersion, PackingTemplateItem, OccurrenceChecklistItem; answering and checking are separate from agreement.
Memories	Memory, MemoryChild, MediaAsset, UploadJob; owner/visibility, occurrence date, processing state and deletion lifecycle.
Delivery	NotificationPreference, domain event/outbox, delivery job; retries and obsolete-event cancellation.
Portability	CalendarFeedToken, ExportJob; scoped access, revocation, expiring output, no private cross-parent leakage.
Operations	AuditEvent, SupportRequest, deletion/retention job; minimal logs and accountable operations.

Write authorization tests before widening the endpoints. Use idempotency keys or equivalent constraints for acceptance, messaging and uploads. Persist domain changes and their delivery/outbox events consistently so a committed agreement is not lost because email fails.

Planning is on the critical path for requests, Home, handovers, reminders and external calendars. Access/media lifecycle is on the critical path for memories. A second developer can work on media after interfaces are agreed, but adding a developer does not halve all elapsed time.

4. Historical web-first planning allowances

Superseded delivery estimate: this table was prepared for a web-first product. Retain the domain/backend allowances as inputs; do not use its total as the native-app quote. Native navigation/forms, sessions/deep links, push, physical-device coverage and distribution need re-estimation after the first installable slice. Some web UI work is replaced, so do not count two full interfaces automatically.

Original assumptions: one experienced full-stack engineer working full time; prompt founder decisions; light design/QA support; managed authentication/storage/email; one web product and English pilot; bounded short clips through a managed service; no native apps, paid billing, two-way sync or legacy-data migration. These are judgment-based work allowances, not measured velocity, supplier promises or market rates. Each range includes its module's implementation and focused tests; the final verification row covers cross-module/system work.

Work package	Engineering days	Main uncertainty
Scope/design closure, spikes and backlog	4–6	Missing states and supplier fit
Foundation, accounts, access, solo/invite, practical profiles	10–15	Sharing transition and managed-auth integration
Care/calendar/activities/holidays	12–18	Interval correctness, recurrence and responsibilities
Requests and history	7–10	Concurrency, affected-version checks and counteroffers

Work package	Engineering days	Main uncertainty
Messages, explicit questions and Home aggregation	6–9	Recoverable sending and coherent state views
Handover and packing	3–5	Occurrence snapshots and changed schedules
Photo/text memories and media lifecycle	8–12	HEIC, processing, access, export and removal integration
Bounded short-clip processing and playback	8–15	Real-phone formats, quality at the proposed size, job limits, private playback/export
iCal feed and file export	3–5	Client behaviour, stable identities and revocation
Notification rules and delivery	3–5	Quiet-hour/digest correctness and job recovery
Integrated QA, rights/support operations, restore and pilot readiness	6–10	Device issues, policy-driven changes and integration defects
Total before reserve	**70–110**	Dependencies and scope decisions remain material

Add approximately 25% uncertainty reserve: **about 90–140 engineering days**, or **18–28 full-time engineering weeks**. This is an initial allowance for a real pilot service, not a calendar-date commitment. Founder response time, independent review, holidays and pilot observation can lengthen elapsed time. Re-estimate after the foundation spikes and the first shared-state milestone.

If the agreed engineering day rate is **R**, the pre-reserve engineering cost is **70R–110R**, and the planning envelope is approximately **90R–140R**. Apply the same multiplication to each module to quote it separately. Design, independent QA/privacy review, hosting, email/media charges, taxes and app-store work are not included in those engineering figures. No rate has been invented.

Commercial additions within those totals

These allowances are already contained in the module table; do not add them again. Some infrastructure and QA are shared, so these are useful quote lines rather than perfectly independent savings.

Bounded addition	Incremental allowance
Explicit questions and Home attention list	About 2–3 days within coordination
Activity packing-list reuse	About 1–2 days within handover/packing
Extra practical profile/contact fields	About 1–2 days within foundation/profiles
Photo/text memories	8–12 days as the memory module, with shared infrastructure dependencies
Bounded short clips	8–15 days, included in the revised table above; re-estimate after the media spike
One-way calendar subscription + snapshot	3–5 days as the calendar-portability module

The revised total includes the existing provisional **8–15 engineering-day** allowance for short clips through a managed media service. This is not a measured estimate for a 3 MB upload; re-estimate after the phone/media spike. Operating charges and provider review remain separate. A custom transcoding system or broader formats can exceed that allowance. AI question detection, an AI tone assistant and two-way calendar sync need separate discovery rather than a small fixed allowance.

---

5. Media capacity and recurring-cost model

The scope proposes pilot guardrails, not subscription tiers. Before real uploads, validate format/quality behaviour and measure sanitized master plus derivative sizes. Use a 2 GB per-family allowance only as a configurable operational starting point. Confirm that it is sufficient for the recruited cohort; preserve read/text/export access when uploads reach the limit.

Illustrative workload, not observed usage: 20 families × 10 photos/week × 6 weeks × 2.5 MB stored per entry = **3,000 MB, approximately 3 GB**, before backups and operational overhead. At that same assumed activity and size for 52 weeks, the cohort stores approximately **26 GB**. An assumed two additional stored copies would make the latter approximately 78 GB; actual backup/versioning policy and pricing must be measured.

Delivery cost depends on image sizes actually served, cache hit rate and revisits. For example, 12,000 thumbnail/preview views at 0.25 MB transferred each is about 3 GB of delivery, excluding master downloads and exports. Do not multiply full-resolution masters by every screen view if the application serves smaller derivatives.

Model monthly cost as: storage GB-month × provider rate, plus delivered GB × rate, processing/jobs, request counts, database/compute, transactional email, monitoring and support. Select rates only after choosing the providers; the current SSH preview hosting arrangement is not a production cost estimate.

For an illustrative 15-second clip, 1.4 Mbit/s video plus 96 kbit/s audio is about 2.8 MB before container overhead. This shows that the suggested size can be plausible; it does not establish acceptable quality for moving children or guarantee an output cap. Twenty families adding five 3 MB clips weekly for six weeks would store about 1.8 GB of playback files, excluding posters, temporary sources, backups and delivery. Measure encoding/retry costs and playback quality as well as storage. The duration, size and bitrate are test assumptions, not published freemium terms.

---

6. Pilot evidence and decision gates

Use 6–8 moderated prototype sessions first, then the 15–25-family real-data pilot. Suggested usability target: at least 80% correctly identify the agreed handover and intended sharing audience without help; any consistent misunderstanding triggers a design correction. These are proposed thresholds, not industry benchmarks. Access leaks, corrupted agreements and data loss are release blockers regardless of percentages.

Measure setup completion, useful solo activity, invitation acceptance within 7/30 days, completed request loops, packing usage, photo/text/clip creation in successive weeks, revisits to earlier memories and external-calendar activation. Record denominators at family level and distinguish solo from connected families. Treat opens and WAU/MAU as supporting observations; successful coordination can reduce the need to open the app frequently.

Interview families about the tasks NAAST replaced and why they would pay or leave. Families who add three memories may be more engaged already; compare similar cohorts, account for time in the product and avoid causal retention claims from small observational samples. Do not expose internal engagement metrics as parent comparisons in the app.

At the pilot review, decide whether to improve the existing flows, trial an excluded addition, or prepare paid launch. If capacity is lower than this estimate, use staged internal milestones or explicitly renegotiate the scope. Do not quietly admit real families before permissions, data recovery or export/departure procedures work.

---

7. What is ready now

The B3 visual direction, phone-first memory refinements (M3 currently preferred by Dani), competitive assessment, consolidated scope and implementation sequence are ready to review. The next concrete development milestone is an installable native shell, persisted child/memory loop and subsequent two-parent care-change loop in section A, following the concrete briefs in document 13. Existing demo tests verify the static examples only; they are not proof that the production service described here exists.

PART 4 / NATIVE KICKOFF

Phone app first — design and implementation kickoff

11 September 2026 · Supersedes the earlier web-first delivery assumption.

> **NAAST is a phone app.** Web is a companion for occasional access and a convenient review/testing surface. B3 remains the identity; M3 is Dani's leading memory layout, with the cofounder's choice pending. Small clips remain in the launch plan with provisional limits. Dani agrees with the remaining feature scope.

The working release sequence is iPhone first, with Android supported by the architecture. Whether both platforms enter the first pilot is still open. This assumption allows design/foundation work to proceed; it is not a recorded platform preference from Dani.

---

1. What changed in the visuals

The phone review is now the main interactive route. On a computer it shows an app-sized viewport, with review controls outside the app. On a phone it fills the available screen. A separate wide companion link remains available.

- Home leads with the children, Add a memory and a recent moment. Compact next-handover and request shortcuts remain visible. Child profiles link directly to their filtered memories.
- Calendar opens a seven-day selector and a selected-day agenda; Week shows grouped daily agendas. The phone no longer depends on dragging a desktop calendar sideways. Friday keeps 17:00 as agreed and routes the 18:00 proposal to Requests.
- Primary actions use large, rounded, yellow buttons with visible verbs. Secondary actions use a light surface and border. This hierarchy also applies to request responses and forms.
- Messages has a scrollable conversation and a composer that stays in view in the app preview. Main tabs remain accessible.
- All four memory layouts keep their distinct structure, now using the same app header, labelled Add memory action, family filters and sheets. The M3 shelf remains the leading candidate.
- Requests, packing, child profiles, setup and settings use the same readable fields, rounded groups and button sizes.

The BeNice screenshots informed grouped rows, prominent rounded actions and sheet hierarchy. NAAST keeps its own B3 colours, compositions and assets. The screenshots are a visual reference, not proof of usability; their expense and AI features are not imported into scope.

The HTML prototype communicates the intended native screens. It is not an installed app, and the current CSS kits do not become native components automatically. The app will use actual React Native views and platform APIs. No fake system status bar is presented as device behaviour.

2. Recommended implementation

React Native + Expo + TypeScript, using Expo Router. Start from a current supported template when implementation begins, commit its lockfile and use the Expo-compatible package versions. Use native stack/tab navigation, platform date/media pickers, native text inputs and accessibility APIs. Build a small B3 component set: Button, ScreenHeader, ChildFilter, GroupedList, RequestCard, MemoryCard and BottomSheet.

Expo documents a common project targeting iOS, Android and web, including platform-specific handling. That fits the phone-first product with a secondary web client; it does not guarantee every component or OS capability is identical across platforms. Expo tutorial.

Use a development build for native integration and physical-phone testing. Expo Go can help early visual iteration but is not the release artifact. Local iOS builds need Xcode; cloud builds are another route. No signing credentials, paid build service or app-store account has been configured for this repository. Expo development builds.

For the first backend, **managed PostgreSQL with Supabase Auth/storage is the recommended candidate**, subject to the existing access/media spike. Keep domain transitions in reviewed server-side functions/services. Database row-level policies enforce family access; request acceptance must atomically check membership, request state and affected care version, update the care block and write history. Do not let the native app directly patch agreement state. Supabase row-level security, database functions.

Use one backend and one authoritative effective-plan model. Share TypeScript domain contracts and design tokens across mobile and web; platform components can differ. The native application must not depend on browser DOM, CSS frameworks or localStorage. Use appropriate protected session storage and validate invitations/auth return links on the receiving device.

Native push is now a launch delivery workstream alongside in-app notifications and email fallback. Keep the existing quiet-hour and privacy rules. Permission denial must not block normal app use; lock-screen payloads default to generic text. Test foreground, background, app-closed and signed-out behaviour with physical devices. Expo notifications.

3. Start now with one complete slice

We have enough direction to begin the foundation. We do not need another full visual exploration before testing shared data. M1/M2/M4 alternatives can remain in the review gallery while the app uses the shared memory record contract and M3 as its temporary presentation.

First milestone: add a child, save a photo/text memory, close the app and find it again on that child's timeline. Then prove safe shared coordination with the two-parent care-change loop. This sequence tests the commercial centre early.

1. Review the bounded contract draft v0.1, encode shared schemas/fixtures and create a Git repository before parallel edits. Keep the static review in `figmaish_pages/`; proposed production folders are `apps/mobile/`, `packages/contracts/`, `packages/design-tokens/` and `supabase/`. They have not been scaffolded by this visual task.
2. Build the native Home, Calendar, Memories, Messages and More shell with B3 primitives and explicit loading/empty/error states. Use M3 provisionally; do not implement all four layouts in production.
3. Add authentication/recovery, scoped family membership and a child profile. Implement text-only and photo-only capture without a mandatory title, child filtering and reopen-after-restart. Start with solo/private records; implement the disclosure/invitation contract before connecting another parent.

4. Prove unrelated-family denial, private/shared visibility, upload recovery, duplicate-create handling and author-only editing/deletion. An uploaded object or local toast alone is not a successfully saved memory. Use fictional records until access isolation is proven.
5. Next implement one already-confirmed dated care block and a proposal from 17:00 to 18:00. The recipient explicitly accepts/declines; server acceptance atomically changes the plan and writes history. Both sessions/restart show the result. Duplicate/stale/expired/wrong-actor responses cannot corrupt it.
6. Produce an installable development build and demonstrate both loops on a physical iPhone, plus Android if included in the pilot. Browser checks are supplementary.

Definition of done: persistent child/memory and two-parent request flows, repeatable migrations/seeds, focused access/concurrency/recovery tests, documented setup and a development build tested on the chosen phone platform. Each loop is an integration checkpoint; neither alone is the full launch MVP.

After these slices, broaden recurrence/holiday planning, questions and packing, then complete bounded clips, memory lifecycle/export and calendar portability. Run real-phone photo/video format research early alongside the first slice. Export/access design precedes real family data; public launch follows all acceptance gates. The intuitiveness review includes the small parent task test to run alongside this work.

4. Contracts to settle before parallel work

Contract	Required boundary
Actor/context	Server derives authenticated actor and family membership; client-supplied family IDs never grant access.
Care block	Stable ID, child IDs, start/end instants, family timezone, agreed version. Display local dates; store unambiguous instants.
Proposal	Request ID, expected care version, proposed values, author, pending state, expiry, idempotency key.
Response	Request ID/version and explicit action. Acceptance is one transaction; conflict is a distinct result with the current state.
Client states	Loading, ready, submitting, accepted, declined, expired, conflict, offline/retry. A cached view cannot imply a write succeeded.
Media	A memory references a typed asset with processing state. Poster/playback/export permissions follow the memory, not a public URL.
Notifications	Server event/outbox after a committed transition; minimal push payload; refetch on open.

The first-slice contract draft now specifies proposed routes, records, errors, media states and acceptance fixtures. Review and encode it as the first implementation artifact. Do not have separate sessions invent different auth, status names or database models.

5. Using more AI sessions

Start with **one lead session for the scaffold and contracts**, then at most **two implementation sessions in parallel**. Keep an independent review session available after the first integration. More simultaneous builders before the interfaces exist will mostly create reconciliation work.

Session	Owns	Avoid editing	Completion evidence
Lead / integration	Scope decisions, shared contracts, root configuration, dependency versions, integration	Unreviewed live production data	Versioned contracts, running scaffold, an integrated slice
Backend	Migrations, access policies, memory/media lifecycle, request transitions, audit/outbox and focused tests	App navigation and visual components	Isolation, double-submit, stale-version and restart-persistence tests
Native app	Screen primitives, navigation, auth UI, child/memory capture and request flow against the agreed API	Database migrations or shared contracts without coordination	Typecheck, supported-device interaction evidence, real API errors handled
Review (after integration)	Read-only code review and focused regression checks	Product scope expansion	Reproducible findings, especially access, conflicts, device and keyboard behaviour

Use separate Git worktrees/branches per builder. One lead owns the shared contracts and lockfile; changes to them must be communicated before implementation continues. Merge one coherent slice and run integration checks before starting more features. Do not run multiple agents modifying the same checkout independently. No additional sessions were launched during this visual pass.

Ready-to-use session briefs:

Lead session

> Read docs/10-final-mvp-scope.md, docs/13-native-app-and-kickoff.md and docs/15-first-slice-contracts.md. Create the Expo/TypeScript native scaffold and versioned contracts for the child/memory slice and subsequent care-change loop. Preserve the static review. Use B3, with M3 as the provisional memory presentation. Document what runs locally and what requires a device/signing account. Do not provision paid services or publish an app. Hand off stable API contracts and folder ownership before starting parallel builders.

Backend session, after the lead has created the contracts

> Implement the first-slice family membership, child/memory persistence, private media and subsequent care block/request transition against the agreed contracts. Own supabase/ and backend tests. Accepting must validate the actor, pending state and care version and atomically write the plan and history. Prove unrelated-family denial, duplicate submission, stale acceptance and persistence. Do not change mobile UI or silently broaden scope. Return migration/setup steps and test evidence.

Native app session, after contracts and scaffold exist

> Implement the B3 app shell, child/memory capture/revisit and subsequent care-change flow in apps/mobile/ using the shared contracts/tokens. Use native controls and the phone review as reference. Show loading, pending, accepted, declined, conflict and retry states. Keep server state authoritative. Do not change database schemas or invent endpoint shapes. Verify on the available device/simulator and report any physical-device limitation explicitly.

These briefs do not authorize sending messages to the cofounder, purchasing services or public store submission.

---

6. Estimates and release gates

The previous 70–110 days before reserve / roughly 90–140 with reserve estimated the web-first implementation including small clips. **That is not a current quote for the native-first delivery.** Backend/domain allowances remain useful inputs, but native UI, session/link handling, push, device/media integration, signing/distribution and platform QA need a revised task breakdown. Avoid simply adding a complete second UI budget: some web work is being replaced, not duplicated.

Re-estimate after the scaffold/device spike and the child/memory and request loops. The first milestone is deliberately much smaller than the launch scope; it should produce an installable internal build before committing to a full release date. Staffing, rates, platform coverage and provider selection remain open.

Suggested gates: working development build → first persisted two-parent loop → internal feature-complete build → physical-device/access/recovery review → 3–5-family pilot → approximately 20 families → public launch preparation. Use TestFlight for an iOS beta when account/signing and the build are ready; it is a distribution step with its own setup and review conditions. Apple TestFlight.

The current workspace has the static design review and planning documents. There is no production backend or installed native build yet. This plan makes the next implementation session concrete without claiming those systems already exist.