Liora — Public Report

1. Introduction

Getting dressed is a high-frequency decision with real emotional stakes. Most people don’t see their wardrobe as a system—resulting in forgotten pieces, repeat looks, and impulse buys. At the same time, AI is finally good enough to “see” garments and “reason” in context, and Brazilian consumers already spend meaningful time in WhatsApp. Liora sits at that intersection: **style, convenience, and sustainability**—without asking users to download yet another app.

2. Overview of the Solution

2.1 What is Liora?

Liora is a **WhatsApp-native AI stylist**. Users send photos of real clothes; Liora recognizes attributes (type, color, pattern, vibe), builds a smart wardrobe, and generates **personalized outfit suggestions** for specific moments—workday, date night, rainstorm, big presentation.

Behind the scenes:

- **Mirror** (vision): An image understanding pipeline that isolates garments and produces compact, structured descriptors.
- **Stylist** (reasoning): A context-aware recommendation engine that composes outfits from the user's wardrobe, balancing cohesion, contrast, and occasion fit.

> We intentionally disclose *methods* (e.g., privacy filtering, semantic retrieval, constrained generation) but not the exact prompts, model blends, or scoring heuristics that drive our quality edge.

2.2 Deliverables and Development Process

We split the first build phase into two streams—**strategy & infrastructure** and **algorithm & integration**—progressing in short sprints to validate quickly:

- **Module 1 — Planning & Infrastructure**: Market research, persona & journey, key feature definition, architecture blueprint, database structure.
- **Module 2 — Algorithms & Integration**: WhatsApp workflow, functional MVP in a safe test chat, and an initial recommendation loop.
- **Artifacts delivered**: Market Report, Personas, Customer Journey, Key Feature Definition, Architecture, Database, Business Plan, MVP via webchat test.

3. Market Research

3.1 Market Size, Drivers, and Timing

Macro tailwinds

Ubiquitous smartphone + camera usage: frictionless wardrobe capture.
Messaging-as-a-platform (in Brazil, WhatsApp): zero-download distribution.
AI maturation: affordable vision + language for consumer UX.

Demand signals

Users want speed (decide faster), certainty (look appropriate), variety (avoid outfit fatigue), and conscience (buy less, use more). Retailers push personalization and conversion; consumers want post-purchase utility (what to wear with what I own).

3.2 Segmentation & Priority Users

Young Professionals (20–35): weekday utility, time-poor, fashion-aware.
Students & Early Career (18–25): confidence, norms, budget control.
Eco-minded Minimalists (25–40): maximize use, reduce waste.
Trend Seekers (18–30): curated novelty that actually pairs with the current closet.

3.3 Competitive Landscape (What Exists vs. What's Missing)

Wardrobe Apps (Acloset, Stylebook, etc.)

Strengths: feature depth, calendars, look books.
Gaps: app install friction; manual workflows; not native to chat; mixed AI quality. AI Style Advisors (StyleDNA, etc.)

Strengths: novelty, analysis.
Gaps: limited use of your actual wardrobe; shopping-first incentives. Generic GPT-style chat

Strengths: general reasoning.
Gaps: no personal closet graph; brittle fashion knowledge; no retrieval over your items.

3.4 Adoption Barriers & Our Responses

To address cold start, we guide new users with starter flows, “top five to add” prompts, and a simple, step-by-step capture. To build trust around photo privacy, we apply on-ingress redaction, clearly communicate retention controls, and enforce row-level security (RLS) by default. To reduce perceived effort, we enable single-message onboarding, deliver a fast “first win,” and provide default nudges based on weather or occasion.

3.5 Go-to-Market Hypotheses

For acquisition, we partner with creators to showcase **real closet** → **real outfits** moments and drive WhatsApp deep links from Reels/TikTok; activation centers on a single, clear promise: *“Send any shirt—get 3 looks in under 10 seconds.”* Retention is sustained through weekly **under-used item** challenges and context nudges tied to forecast or calendar moments. Revenue follows a freemium ladder—**Premium** (unlimited recommendations, weather features) and **Pro** (voice, insights)—plus selective curated commerce that appears only when a product complements ≥3 pieces a user already owns.

4. Personas

We treat personas as instrumented hypotheses—each includes triggers, objections, and success definitions that map directly to product and growth.

Laura (28) — Time-Pressed Professional

Jobs to be done: Pick a polished outfit in minutes for varied professional settings. Reduce morning indecision; avoid repeating the same “safe” look.

Triggers: early meetings, client days, travel weeks.

Objections: “Will this be fast enough?” “Do I have to do a lot of setup?”

Activation hooks: “Send a blazer photo → get 3 meeting-ready looks.”

Success metrics: D7 retention; “time-to-first-win” < 1 day; rec acceptance rate.

Feature fit: item-anchored recs; context prompts; minimal onboarding; look explanations (“why this works”).

Rafael (32) — Eco-Minded Minimalist

Jobs to be done: Use everything I own; buy only what truly complements.

Triggers: seasonal swaps, “shopping guilt,” capsule curation.

Objections: “Will it push shopping?” (we don’t by default)

Activation hooks: “Your 5 most under-used items this month—here’s how to wear them.”

Success metrics: unique items worn/month; reduction in impulse buys (self-report).

Feature fit: under-used surfacing; capsule tips; “complements 3+ items” gate before any shopping suggestion.

Camila (25) — Trend-Seeking Budget Keeper

Jobs to be done: Stay fresh and on budget; ensure new pieces work with my closet.

Triggers: events, seasonal trends, flash sales.

Objections: “I buy and then it sits unworn.”

Activation hooks: “That new skirt? 3 outfits using what you own.”

Success metrics: regret score drop; repeat use of “pair-before-purchase.”

Feature fit: link-based add, occasion looks, budget guardrails, shareable “fit cards.”

Pedro (19) — First-Principles Confidence

Jobs to be done: Dress appropriately; learn color/fit basics; reduce social anxiety.

Triggers: presentations, dates, networking.

Objections: “I don’t know the rules.”

Activation hooks: “Send what you’re thinking—get quick ‘swap this for that’ tips.”

Success metrics: acceptance rate; reduced changes before leaving.

Feature fit: inline color-pairing hints, casual

5. Customer Journey Map

To gain a deeper understanding of the user experience with Liora, we developed a Customer Journey Map that outlines each stage of interaction with our product—from initial contact to ongoing use. This tool was essential in identifying points of friction, opportunities to delight the user, and in guiding the prioritization of core MVP features.

The journey was divided into seven key stages, highlighting user behaviors, interaction channels, and the emotions experienced throughout the process.

Customer Journey Map

STAGE	Awareness	Consideration	Purchase	Onboarding	First Usage	Daily Use	Retention & Growth
USER ACTION	Visits the Vise landing page.	Reads about features & pricing.	Selects a plan, enters phone number, completes checkout.	Receives WhatsApp confirmation and first message from Vise.	Uploads photos of clothes, interacts with Vise via messages and voice notes.	Adds/deletes clothes to/from virtual wardrobe. Requests outfit recommendations.	Continues using Vise, explores more features. Shares with friends.
TOUCH POINTS	Website	Website	Website, Payment Gateway	WhatsApp	WhatsApp	WhatsApp	WhatsApp, Social Media
FEEL-INGS	Curious about the product. Wants an easy way to manage outfits.	Excited by the convenience of Vise. Wonders if it will work well.	Quick process. Happy with simplicity. Ready to start.	Feels welcomed and ready to try it out.	Feels engaged, enjoys the AI-driven conversation.	Finds it easy and convenient. Appreciates quick responses.	Feels loyal to Vise. Enjoys personalized fashion assistance.

This journey was visually represented in an interactive prototype developed in Figma, where we simulated the user flow of Liora within the WhatsApp environment. This visual representation allowed us to test the usability of the interaction, simulate real-world behavior, and refine key touchpoints in the conversational experience.

6. Key Feature Definition

After gaining a solid understanding of the market landscape and the profile of our target audience, we moved on to defining Liora's core functionalities using a structured, user-centered approach. The goal was to identify the essential capabilities the platform must deliver in order to create real value, while also taking into account the constraints of a viable MVP.

The methodology adopted for this stage was grounded in Design Thinking, specifically the Double Diamond framework. We began with a divergent phase, exploring a wide range of possibilities based on market trends, competitive analysis, and Liora's strategic vision. Using a collaborative Miro board, we grouped ideas, prioritized features, and translated insights into actionable functionality.

These features were then organized into two primary groups:

- **Essential**, representing the minimum viable core required for Liora to fulfill its value proposition;
- **Non-essential**, representing enhancements that improve the user experience but can be developed in later phases.

To structure these functionalities clearly and in a scalable way, we grouped them into four main epics, each containing related features and user stories:

Epic 1: Account Management

This epic encompasses the administrative and personalization functions related to the user's account. Among the essential features are:

- **Plan management and settings** (e.g., upgrading or downgrading a subscription, editing personal information, deleting an account);
- **Collection of demographic data** (such as location and age group) to personalize recommendations.

As future features, we propose:

- A **fashion-focused social space**, where users can exchange or sell clothing with others who share a similar style;
- **Body measurements and sizing preferences registration**, to further refine outfit curation and personalization.

Epic 2: Virtual Wardrobe

This is the visual and functional core of Liora. It is where users manage their digitized clothing items. The essential features within this epic include:

- Clothing registration via image, directly through WhatsApp;
- Removal of items from the digital wardrobe;
- Automatic attribute detection (such as type, color, and pattern) using computer vision.

As potential future functionalities, we propose:

- Clothing registration via shopping links, bridging online and physical wardrobes;
- Video-based item detection, to further simplify the registration process;
- Automatic background removal and the ability to create personalized moodboards.

Epic 3: Fashion Recommendation System

Personalized recommendations are the core intelligent differentiator of Liora. Among the essential features in this epic, we highlight:

- Outfit suggestions based on a specific item sent by the user;
- Context-aware recommendations, tailored to specific occasions (such as parties, work, or weather), and even to the user's mood.

In the platform's future development roadmap, we propose:

- A history of past suggestions, allowing users to revisit and reuse them;
- Weather-based reminders, offering recommendations aligned with forecast conditions;
- Smart shopping suggestions, which propose new items that consciously complement the user's existing wardrobe.

Epic 4: WhatsApp Interaction with Conversational AI

Finally, the interface layer is what makes Liora both accessible and unique: the interaction with the assistant happens entirely within WhatsApp. As an essential feature:

- Text-based chatbot with artificial intelligence, enabling the exchange of messages about clothing, context, and personalized recommendations.

As a future enhancement:

- Voice interaction, allowing users to send voice messages and receive suggestions in an even more natural and accessible way.

This feature definition served as the central guide for MVP prioritization, helping transform our vision into a concrete and actionable structure. With this foundation in place, we were able to move forward with the architectural design of the solution—already having a clear understanding of what to build, who to build it for, and why.

7. Key Feature Architecture

We designed for **speed, privacy, and iteration—**and we keep some implementation details proprietary to protect our edge.

7.1 Multi-Channel Architecture

- **WhatsApp (primary)**: An orchestrated webhook flow manages messages, media, and context.
- **Web companion (secondary)**: A clean interface for wardrobe browsing, history, and account controls.

7.2 Core Workflow

1. **Ingest**: Receive text/image → enrich with minimal context (intent, occasion hints).
2. **Understand**: Mirror isolates garments and outputs a compact descriptor set (category, palette, pattern, style cues). Face/PII elements are filtered upstream for privacy.
3. **Retrieve**: A semantic layer runs fast similarity over the user's wardrobe (lightweight embeddings + vector ops).
4. **Compose**: Stylist assembles outfits with constraint-aware generation (occasion, weather, user tone).
5. **Respond**: Return multiple looks, each with rationale and easy follow-ups ("more casual", "swap shoes", "rain-proof").

What we share, what we don't: We disclose the pipeline stages and quality principles (privacy filtering, semantic retrieval, constraint composition). We do **not** publish prompt templates, model ensembles, weighting schemes, or evaluation rubrics that materially enable replication.

7.3 Backend Tech Stack

- **Core principles:** managed services, clean contracts, small blast radius, pay-as-you-grow.
- **Practical setup:** a modern application backend, managed Postgres with vector search, object storage for images, and a thin chat orchestration layer.
- **Why this works now:** rapid delivery for MVP, clear scale path, vendor-portable patterns.

7.4 Database Architecture (high level)

- **Users & Preferences** (privacy-first fields; opt-in extended data)
- **Wardrobe Items** (image refs, canonical attributes, vector column)
- **Chats & Threads** (context continuity, auditability)
- **Recommendations** (inputs, outputs, acceptance signals)

We maintain strict row-level isolation and signed asset access. Attribute taxonomies are normalized to keep retrieval fast and portable.

7.5 AI Integration

- **Mirror**: Multi-step vision with privacy guards; outputs structured descriptors suitable for vectorization.
- **Stylist**: Context assembly + retrieval-augmented composition; blends rules with model-based reasoning to avoid fashion “hallucinations.”
- **Multi-modal**: Text + image today; **voice** as an opt-in layer with on-device redaction where supported.

7.6 Security & Compliance

- **Data protection:** encryption in transit/at rest, least-privilege keys, audit trails.
- **Privacy by design:** face/PII scrubbing on ingress; transparent retention controls; easy export/delete (LGPD-aligned).
- **Abuse prevention:** rate limiting, anomaly flags, safe-completion policies.

7.7 Early Performance (MVP)

- **Recommendation latency:** typically **7–10s** end-to-end under MVP load.
- **Reliability:** high availability with simple rollbacks.
- **Cost:** lean infra for early stage; scales predictably as cohorts grow.

8. Liora’s Database Structure

Our data design is deliberately simple to operate, fast to query, and strict on privacy. Below is a conceptual, public view of how we structure information—enough to show rigor and readiness without exposing replicable internals.

8.1 Objectives & Principles

Privacy first: minimize personal data; isolate user data at every layer; short-lived media links.
Performance under chat latency: optimize for sub-10s end-to-end recommendation rounds.
Clarity over cleverness: small set of well-defined entities; explicit relationships.
Portability: avoid vendor lock-in patterns; keep advanced features optional.
Observability: every critical action is auditable and attributable.

8.2 Conceptual Domains (What we store, not how)

1. Identity & Access

- Lightweight user profile and subscription state.
- Role/permission flags for rare operational needs.
- Consent snapshots tied to policy versions.

2. Wardrobe Graph (User’s Closet)

- Each garment has a node with stable, human-readable attributes (category, color family, style hints, etc.).
- Optional user categories (e.g., “Workwear”, “Weekend”).
- Saved outfits as “compositions” referencing multiple garments.

3. Conversations & Context

- Chat threads (WhatsApp or web) and messages (text, image, voice).
- Context markers (occasion, weather tag, mood signal) attached to a thread or a single turn.

4. Recommendations & Feedback

- Recommendation requests (inputs, timing).
- Proposed outfits (what was suggested and why).
- Lightweight quality signals (accepted, edited, skipped) for learning.

5. Observability & Audit

- Append-only events for product analytics (e.g., onboarding steps, wardrobe additions, feature usage).
- Audit trail for sensitive changes (who, what, when), retained per compliance policy.

6. Content Storage (Media)

- Images and attachments stored out of band (object storage) with signed, expiring access.
- No permanent public buckets for wardrobe media.

8.3 Data Flow at a Glance

- **Ingest**: User sends a photo or message. Minimal metadata and source channel are attached.
- **Sanitize**: On-ingress privacy guards (e.g., face/PII scrubbing) before any long-term storage.
- **Understand**: Vision pipeline extracts compact garment descriptors; text is normalized to a controlled vocabulary.
- **Index**: Descriptors are prepared for fast similarity lookups within the user's own closet.
- **Compose**: The recommendation engine combines context + wardrobe retrieval to produce outfits with rationales.
- **Deliver & Learn**: Suggestions are sent; user reactions (accept/edit/skip) feed future tuning.

8.4 Privacy, Security & Governance (Public Pattern)

- **Data minimization**: only what's needed to operate threads and wardrobes; optional email is decoupled from chat identity.
- **Strong isolation**: user-scoped access everywhere (logical isolation + policy checks).
- **Encrypted everywhere**: in transit and at rest; short-TTL, signed media URLs.
- **Transparency & control:** clear consent, easy export, and delete on request (LGPD aligned).
- **Abuse prevention:** rate limits, anomaly detection, and safe-completion policies.

8.5 Performance & Scale Approach

- **Hot paths**: optimized for per-user wardrobe reads and recent message history.
- **Similarity retrieval:** efficient, per-user vector lookups to find "what goes with what I own"—kept small and fast by scoping to one closet at a time.
- **Pagination & caching:** predictable pagination patterns and response caching for recent looks.
- **Graceful growth**: linear scaling of storage and retrieval as wardrobes expand.

8.6 Data Quality & Taxonomy

- **Canonicalization**: normalize colors, categories, and style tags to a curated vocabulary.
- **Confidence thresholds**: low-confidence items go to a lightweight review/repair queue.
- **Guardrails**: basic dress-code and occasion rules to avoid nonsensical pairings.
- **Proprietary mappings**: the exact harmonization rules and descriptors are intentionally private.

8.7 Lifecycle & Compliance

- **Retention**: operational data kept as long as the user benefits; raw artifacts trimmed aggressively; summaries retained longer for continuity.
- **Deletion**: soft-delete for recovery; hard-delete for user requests.
- **Backups**: encrypted daily snapshots and point-in-time recovery; periodic restore tests.

8.8 What We Intentionally Don't Publish

To protect Liora's defensibility, we do not disclose:

* Exact descriptor schema or embedding recipes.
* Ranking features, weights, or quality scoring heuristics.
* Prompt templates, model blends, or orchestration choreography.
* Internal evaluation datasets and rubrics.

9. Hypotheses Table

We run Liora like a lab—every feature ties to a testable outcome.

Persona	**Hypothesis**	**Metric**	**Evidence to Ship**
Laura	If we return **2–3 outfits** with rationale in <10s, she saves morning time and returns the next day.	D1 retention, reply time	+5–7pp D1 lift vs single suggestion
Rafael	If we surface **under-worn items** weekly, he uses more of his closet.	% unique items worn/mo	+20–30% unique wear count
Camila	If we gate purchases behind “works with 3+ items you own,” spend regret drops.	Self-reported regret, returns	−25% regret within 30 days
Pedro	If we add **color-pairing tips** inline, he accepts more suggestions.	Acceptance rate	+15% acceptance

10. Business Plan

10.1 Executive Summary

Mission: Help people dress better with what they already own—quickly, confidently, and sustainably. **Vision:** Be Brazil's most trusted conversational fashion companion. **Audience:** Urban 20–35 who value convenience and style; WhatsApp natives.

10.2 Market Analysis

- Growing global AI-fashion market; Brazil is under-served by wardrobe tech with high WhatsApp affinity.
- Strategic gap: **no mainstream WhatsApp-first wardrobe assistant** with serious AI quality.

10.3 Marketing Strategy

- **Channel-fit creative:** short demos in Reels/TikTok showing the "send shirt → get 3 outfits" moment.
- **Creators:** micro-influencers co-create looks using their *real* closets.
- **Lifecycle:** WhatsApp onboarding hints, weekly "capsule challenge," weather-aware nudges.

10.4 Operational Plan

- 24/7 assistant via orchestrated workflows and health checks.
- Continuous feedback loops: conversation marks ("helpful/nah"), quick polls, and opt-in interviews.
- Privacy and LGPD compliance from day one.

10.5 Financial Plan

- **Freemium** with three paid tiers (Premium / Pro / Enterprise) aligned to power-use features (unlimited recs, voice, shopping integration, insights).
- Lean infra in MVP, scaling steadily with user growth and B2B offsets.

10.6 Risks & Mitigations

- **Model drift / quality:** human-in-the-loop evals, offline test sets, and canary prompts.
- **Adoption:** obsession with first-week "time-to-first-win," low-effort onboarding.
- **Data security:** layered controls, external audits at scale milestones.

10.7 Success Metrics

- **Product:** D1/D7 retention, rec acceptance rate, wardrobe coverage.
- **Business:** MRR, CAC/LTV, paid conversion, churn.
- **Ops:** median latency, error rate, uptime.

10.8 Next Steps

- **0–3 months:** closed beta (≈100 users), iterate on onboarding and rec quality, instrument success metrics.
- **6–12 months:** public launch, voice beta, weather-aware routines, partner pilots.
- **24+ months:** B2B integrations, curated commerce where it *truly* complements the user's wardrobe.

11. Conclusion

What we proved. A WhatsApp-native assistant can deliver **useful, personal** outfit suggestions in **under 10 seconds** using a privacy-forward pipeline. Users understand it instantly, and early feedback shows clear perceived value: faster decisions, more confident looks, fewer impulse purchases.

What we're building next. Two threads in parallel: (1) **Sharper product**—voice, weather, better cold-start, and a memory that feels like a real stylist; (2) **Smart go-to-market**—creators, community challenges, and selectively curated shopping that respects the "use what you have" principle.

How we protect the edge. We'll continue to share our philosophy, guardrails, and high-level methods—but we'll keep the exact prompting, ranking, quality metrics, and workflow choreography internal. That's where a lot of Liora's magic—and defensibility—lives.