Solace Journal – Design Documentation

Problem Understanding

Journaling is widely recommended for mental clarity and emotional awareness, but many people struggle to actually do it consistently. The main blockers are friction at the start, feeling unsure what to write, and feeling talked at instead of listened to. Traditional journaling tools often feel either too bare or too prescriptive. AI tools can help, but many feel impersonal, overly clinical, or invasive.

The goal of Solace Journal is to make journaling feel like a **gentle conversation with yourself**. The companion should help users put thoughts somewhere safe, notice patterns over time, and feel understood, without pushing solutions or diagnoses. It is not meant to replace people or therapy. It is meant to always be available when someone wants to think out loud.

The target users include:

- People focused on mental wellness who want help recognizing emotional and behavioral patterns
- People new to journaling who need guidance without pressure
- Busy professionals who want a quick way to decompress and process their day

Core Product Principles

Several principles guided all design decisions:

- **Listener-first, not fix-it AI**
  Reflections focus on empathy, specificity, and noticing. Advice is optional and never forced.
- **Privacy by default**
  All insights and pattern analysis run locally. External AI is optional and explicitly opt-in.
- **Low friction, low judgment**
  Short, messy, or incomplete entries are valid. The system adapts to the user, not the other way around.
- **Pattern recognition over time**
  The value compounds as users write more, through themes, mood trends, and memory callbacks.

Feature Overview

Journaling

- Freeform entries grouped by local day
- Entries can be edited or deleted at any time
- Saving and reflecting are clearly separated to avoid confusion
- Deleting an entry updates insights and rebuilds memory correctly

Reflections

- Two modes:
    - **Private (local)**: heuristic-based, fully on-device
      Enhanced (LLM): optional, uses OpenAI for richer language understanding
- Reflections are short, contextual, and varied
- The system detects intent, tone, and topic so responses match what the user actually wrote
- Gibberish or ultra-short input is handled gracefully with gentle prompts
- Positive entries are treated as wins, not problems to solve

Guided Session (5-minute journal)

- Designed for “where do I start” moments
- Four modes: Unwind (work), Untangle (stress), Check-in (quick), Small win
- Presents 3 to 4 prompts one at a time
- Saves as a single journal entry and generates a reflection at the end
  Especially effective for new users and busy professionals

Insights Dashboard

- All insights are computed locally
- Includes:
    - Mood over time
    - Recurring themes (work pressure, relationships, self-care, loneliness, etc.)
    - What seems to help, inferred from past entries and memory
    - A weekly snapshot written in natural language
- Themes are inferred from context, not just repeated words
- Users can expand themes to see examples and explanations

Memory Layer

- A small local memory stores simple preferences and patterns such as coping behaviors
- Memory is rebuilt when entries are deleted
- Used subtly in reflections and insights, not on every entry

Privacy Controls

- Privacy mode is on by default
- Switching to enhanced mode requires confirmation and disclosure
- If no API key is available, enhanced mode is gracefully disabled
- Insights never make network calls, regardless of mode

Technical Design and Stack

Frontend

- React with TypeScript
- Vite for fast development
- Tailwind CSS for a clean, calm UI
- react-router-dom for routing

State and Storage

- Browser local storage for entries, reflections, and memory
- Helper utilities for ID generation and local-day grouping to avoid timezone issues

AI Integration

- OpenAI Responses API for enhanced reflections
- A stable reflection facade abstracts local vs enhanced logic
- Detection, generation, and post-processing are separated for maintainability

Because this is a frontend-only prototype, the API key lives in the client. This is not production-safe. In a real deployment, enhanced reflections would be routed through a lightweight backend proxy so the key is never shipped to the browser, following OpenAI best practices

Responsible AI and Trust

Several guardrails are built into both product behavior and language:

- Reflections avoid quoting user text directly to reduce a surveillance feeling
- Language is non-diagnostic and non-clinical
- The system does not assume all entries require emotional work
- Self-harm signals trigger a gentle, non-alarmist safety note with resources
  Users always remain in control of what is saved, remembered, or deleted

These choices are meant to build trust over time and avoid the failure mode where users feel they could have just typed into a generic chatbot. They intentionally prioritize engagement through low friction, insight through pattern recognition, and trust through local-first analysis.

Key Learnings

The biggest challenge was avoiding generic, template-like AI responses. Even with an LLM, early versions felt impersonal. The solution was to treat journaling as something that is not always about fixing or processing. Sometimes it is just noticing. Designing for that truth improved both local and enhanced reflections significantly.

Another key insight was that privacy is not just a technical feature, but a UX one. Explicit toggles, confirmations, and language choices matter as much as where the data flows.

Future Enhancements

There is significant room to expand while staying aligned with the core philosophy:

- More robust local NLP for theme and mood detection
- User-editable themes and the ability to correct insights
- Additional guided sessions for relationships, finances, and transitions
- Search and filters by theme, mood, or date range
- Optional local encryption for stored journals
- Export and import for device migration
- Performance optimizations for large journals
- A backend proxy for enhanced mode to support production security requirements
- A user feedback system for reflections and insights