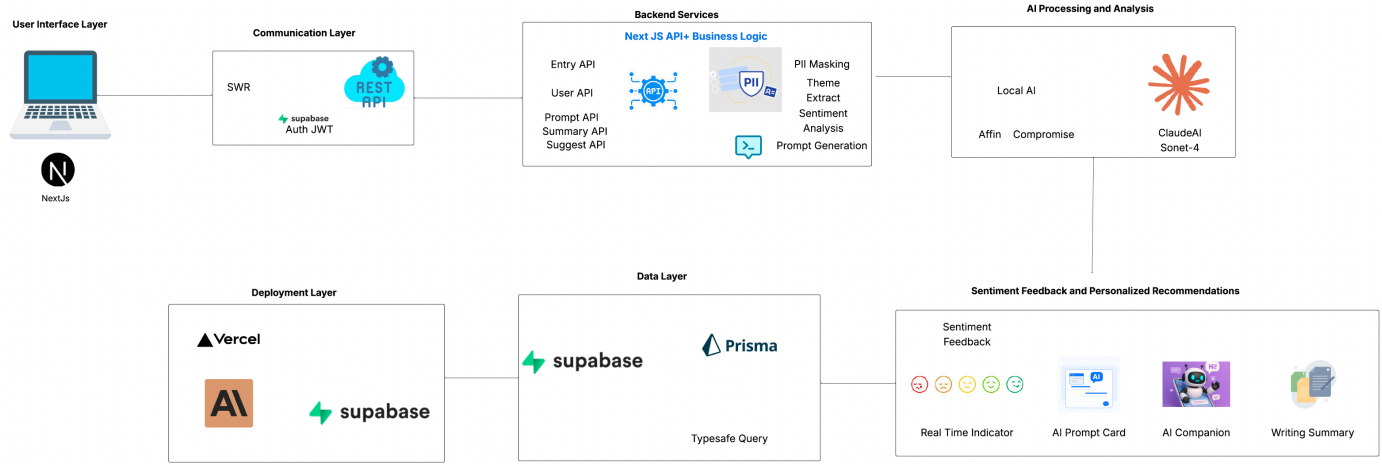
Architecture Diagram



Reflect AI - Design Choices

1. Privacy-First Architecture

Choice: PII masking before all external API calls, local processing for sentiment analysis

Rationale: Critical for user trust in a journaling app. Demonstrates security-first mindset for PANW judges.

2. Hybrid AI Processing

Choice: Local AFINN sentiment (instant) + Cloud Claude API (advanced features)

Rationale: Instant feedback without latency, reduces API costs, works offline, maintains privacy.

3. Next.js 14 App Router

Choice: Server Components by default, Client Components only when needed

Rationale: Smaller bundle size, better performance, direct database access, improved SEO.

4. TypeScript Throughout

Choice: 100% TypeScript with strict type checking

Rationale: Catch errors at compile time, better IDE support, safer refactoring, types as documentation.

5. SWR for Data Fetching

Choice: SWR library for client-side data fetching with caching

Rationale: Automatic caching, background revalidation, optimistic updates, reduces API calls.

6. Prisma ORM

Choice: Prisma instead of raw SQL queries

Rationale: Type-safe queries, excellent migrations, prevents SQL injection, better developer experience.

7. PostgreSQL via Supabase

Choice: PostgreSQL database hosted on Supabase

Rationale: Relational data fits journal entries, JSON support for flexible data, integrated auth, free tier.

8. REST API Architecture

Choice: REST API routes instead of GraphQL

Rationale: Simpler to implement, better HTTP caching, works well with TypeScript, less overhead.

9. Aggregated Data Only

Choice: Weekly summaries send only statistics, not full entry text

Rationale: Privacy protection, smaller payloads, lower costs, faster API calls.

10. Debouncing for AI Suggestions

Choice: 2-second delay + >100 characters before AI suggestions

Rationale: Reduces API calls, prevents overwhelming users, natural rate limiting, cost efficiency.

11. Caching Strategy

Choice: Cache prompts (24h) and summaries (weekly)

Rationale: Reduces API costs, faster responses, stays within rate limits, better UX.

12. Supabase Auth

Choice: Supabase Auth instead of custom authentication

Rationale: Battle-tested security, faster implementation, JWT tokens, integrated with database.

13. Zod Input Validation

Choice: Zod schemas for all API request validation

Rationale: Type inference, runtime validation, clear error messages, prevents malicious input.

14. Component-Based Architecture

Choice: Modular React components organized by feature

Rationale: Reusability, maintainability, easier testing, clear organization, better collaboration.

15. shadcn/ui Component Library

Choice: shadcn/ui built on Radix UI

Rationale: Excellent accessibility, easy Tailwind customization, copy-paste components, TypeScript support.

16. Separation of Concerns

Choice: Clear separation between API routes, business logic, and data layer

Rationale: Maintainability, testability, reusability, better organization, easier debugging.

17. Vercel Deployment

Choice: Deploy on Vercel platform

Rationale: Optimized for Next.js, global CDN, serverless functions, easy Git-based deployment.

18. Local Processing First

Choice: Sentiment analysis and theme extraction happen locally

Rationale: Instant feedback, no external calls, privacy, works offline, cost-effective.

19. Context-Aware Prompts

Choice: Prompts generated based on entry history, themes, and sentiment patterns

Rationale: Personalized experience, reduces blank page anxiety, creates continuity, improves engagement.

20. User Privacy Controls

Choice: Users can disable AI features, export data, delete account

Rationale: User trust, compliance with privacy regulations, transparency, user control.

Reflect AI Tech- Stack

1.Frontend Technologies

-> Core Framework

- **Next.js 14.2.0** (App Router)
  - Server Components by default
  - Route Handlers for API endpoints
  - Built-in optimizations (image, font, script)
  - File-based routing with route groups

-> UI Framework

- **React 18.3.0**
  - Functional components with hooks
  - Server Components for data fetching
  - Client Components for interactivity

-> Language

- **TypeScript 5.x**
  - Full type safety throughout
  - Strict mode enabled
  - Type inference and narrowing

-> Styling

- **Tailwind CSS 3.4.1**
  - Utility-first CSS framework
  - Mobile-first responsive design
  - Custom color palette (teal primary)
  - Dark mode support

-> UI Component Library

- **shadcn/ui (built on Radix UI)**
  - Accessible component primitives
  - Customizable with Tailwind
  - Components used:
    - Button, Card, Input, Textarea
    - Dialog, Dropdown Menu
    - Switch, Toast, Label

-> Icons

- Lucide React 0.400.0
  - Consistent icon set
  - Tree-shakeable
  - Used throughout UI

-> Data Visualization

- **Recharts 2.12.0**
  - Sentiment trend charts
  - Responsive line charts
  - Customizable styling

-> State Management & Data Fetching

- **SWR 2.2.5**
  - Client-side data fetching
  - Automatic revalidation

- Cache management
- Optimistic updates

-> Forms

- **React Hook Form 7.52.0**
  - Form state management
  - Validation integration
  - Performance optimized

-> Validation

- **Zod 3.23.0**
  - Schema validation
  - Type inference
  - API request validation
  - Error messages

2. Backend Technologies

-> API Framework

- **Next.js API Routes (Route Handlers)**
  - Server-side API endpoints
  - TypeScript support
  - Middleware integration
  - Edge runtime compatible

-> Database ORM

- **Prisma 5.20.0**
  - Type-safe database client
  - Migration management
  - Query builder
  - Relationship handling

-> Database

- PostgreSQL (via Supabase)
  - Relational database
  - JSON column support
  - Full-text search capability
  - Connection pooling

-> Authentication

- **Supabase Auth (@supabase/ssr 0.8.0)**
  - JWT-based authentication
  - Email/password auth
  - Session management
  - Server-side auth helpers

3.AI & NLP Technologies

-> Cloud AI Service

- Anthropic Claude API (@anthropic-ai/sdk 0.40.0)
  - Model: claude-sonnet-4-20250514
  - Prompt generation

  - Writing suggestions
  - Weekly summaries
  - PII masking before API calls

-> Local Sentiment Analysis

- sentiment 5.0.2

  - AFINN-165 word list
  - Real-time sentiment scoring
  - No external API calls
  - Score range: -5 to +5

-> Local NLP Processing

- compromise 14.10.0

  - Theme extraction
  - Part-of-speech tagging
  - Noun phrase extraction
  - Local text processing

-> PII Detection

- email-regex 5.0.0

  - Email address detection

- phone-regex 1.0.0

  - Phone number detection

- Custom regex patterns

  - SSN, credit card, address detection

4.Development Tools

-> Build Tools

- TypeScript Compiler

- Type checking
- ES2020 target
- Module resolution

-> Code Quality

- ESLint 8.x

- Next.js config
- TypeScript rules
- React hooks rules

-> Package Management

- npm

- Dependency management
- Script execution
- Post-install hooks

-> Environment Management

- **dotenv-cli 11.0.0**

- Environment variable loading
- Local development

5.Deployment & Infrastructure

-> Hosting

- Vercel

- Edge network
- Automatic deployments
- Environment variables
- Serverless functions

-> Database Hosting

- Supabase

- Managed PostgreSQL
- Connection pooling
- Real-time capabilities
- Free tier available

-> Version Control

- Git

- Source control
- Branch management
- Commit history

6.Utility Libraries

-> Utilities

- clsx 2.1.0

- Conditional class names

- Tailwind class merging

- tailwind-merge 2.3.0

- Merge Tailwind classes
- Conflict resolution

- class-variance-authority 0.7.0

- Component variant management
- Type-safe variants

-> Date Handling

- date-fns 4.1.0

- Date formatting
- Date calculations
- Timezone handling

7.Architecture Patterns

-> Design Patterns Used

1. Server Components First: Default to Server Components, use Client Components only when needed

2. API Route Pattern: RESTful API routes for data operations

3. Repository Pattern: Prisma abstracts database access

4. Service Layer: Business logic separated from API routes

5. Middleware Pattern: Authentication and route protection

-> Code Organization

```
src/
├── app/            # Next.js App Router pages and API routes
├── components/     # React components
├── lib/            # Utility functions and business logic
├── hooks/          # Custom React hooks
└── types/          # TypeScript type definitions
```

Performance Optimizations

1. Server Components: Reduce client bundle size
2. Code Splitting: Automatic with Next.js
3. Image Optimization: Next.js Image component
4. SWR Caching: Reduce API calls
5. Debouncing: Limit AI API requests
6. Database Indexing: Optimized queries
7. Connection Pooling: Efficient database connections

Security Features

1. PII Masking: Before external API calls
2. Input Validation: Zod schemas
3. SQL Injection Prevention: Prisma ORM
4. Authentication: Supabase JWT tokens
5. HTTPS: Encrypted traffic
6. Environment Variables: Secure secret management

Browser Support

- Modern browsers (Chrome, Firefox, Safari, Edge)
- Mobile responsive (iOS Safari, Chrome Mobile)
- Progressive Web App (PWA) ready

Development Environment

- Node.js: 18.x or higher
- npm: 9.x or higher
- TypeScript: 5.x
- PostgreSQL: Managed via Supabase

Key Dependencies Summary

-> Production Dependencies

```json
{
  "@anthropic-ai/sdk": "^0.40.0",
  "@prisma/client": "^5.20.0",
  "@radix-ui/react-*": "^1.0.5 - ^2.1.16",
  "@supabase/ssr": "^0.8.0",
  "@supabase/supabase-js": "^2.45.0",
  "class-variance-authority": "^0.7.0",
  "clsx": "^2.1.0",
```

```
  "compromise": "^14.10.0",
  "date-fns": "^4.1.0",
  "email-regex": "^5.0.0",
  "lucide-react": "^0.400.0",
  "next": "14.2.0",
  "next-themes": "^0.4.6",
  "phone-regex": "^1.0.0",
  "react": "^18.3.0",
  "react-dom": "^18.3.0",
  "react-hook-form": "^7.52.0",
  "recharts": "^2.12.0",
  "sentiment": "^5.0.2",
  "swr": "^2.2.5",
  "tailwind-merge": "^2.3.0",
  "tailwindcss-animate": "^1.0.7",
  "zod": "^3.23.0"
}
```

-> Development Dependencies

```json
{
  "@types/node": "^20",
  "@types/react": "^18",
  "@types/react-dom": "^18",
  "autoprefixer": "^10.4.19",
```

```
  "dotenv-cli": "^11.0.0",
  "eslint": "^8",
  "eslint-config-next": "14.2.0",
  "postcss": "^8.4.38",
  "prisma": "^5.20.0",
  "tailwindcss": "^3.4.1",
  "typescript": "^5"
}
```

Technology Choices Rationale

-> Why Next.js 14 App Router?

- Server Components reduce client bundle size
- Built-in API routes simplify backend
- Excellent developer experience
- Optimized for production

-> Why Prisma?

- Type-safe database access
- Excellent migration system
- Great developer experience
- Strong TypeScript support

-> Why Supabase?

- Managed PostgreSQL database
- Built-in authentication

- Free tier for hackathons

- Easy to set up

-> Why Anthropic Claude?

- Demonstrates PANW product knowledge

- High-quality AI responses

- Good prompt engineering support

- Privacy-conscious API

-> Why Local Sentiment Analysis?

- Instant feedback (no API delay)

- Privacy-first (no external calls)

- Cost-effective (no API costs)

- Reliable (no rate limits)

-> Why SWR?

- Automatic caching and revalidation

- Optimistic updates

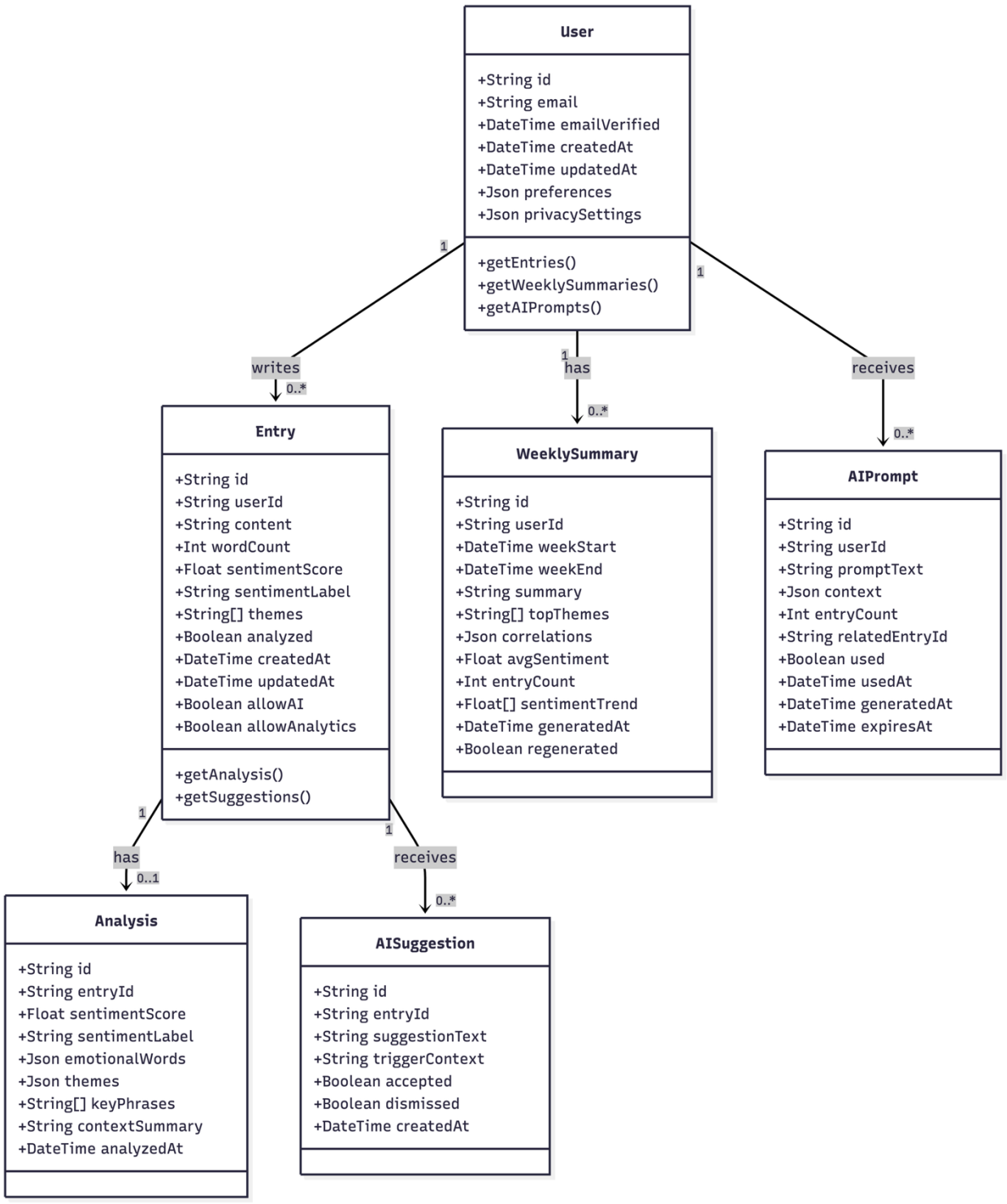
- Error handling built-in

- Small bundle size

UML Diagrams

1.Class Diagram

The class diagram shows the database models and their relationships in Reflect AI.



- User → Entry: One-to-many relationship. A user can write multiple entries.

- User → WeeklySummary: One-to-many relationship. A user can have multiple weekly summaries.

- User → AIPrompt: One-to-many relationship. A user can receive multiple AI prompts.

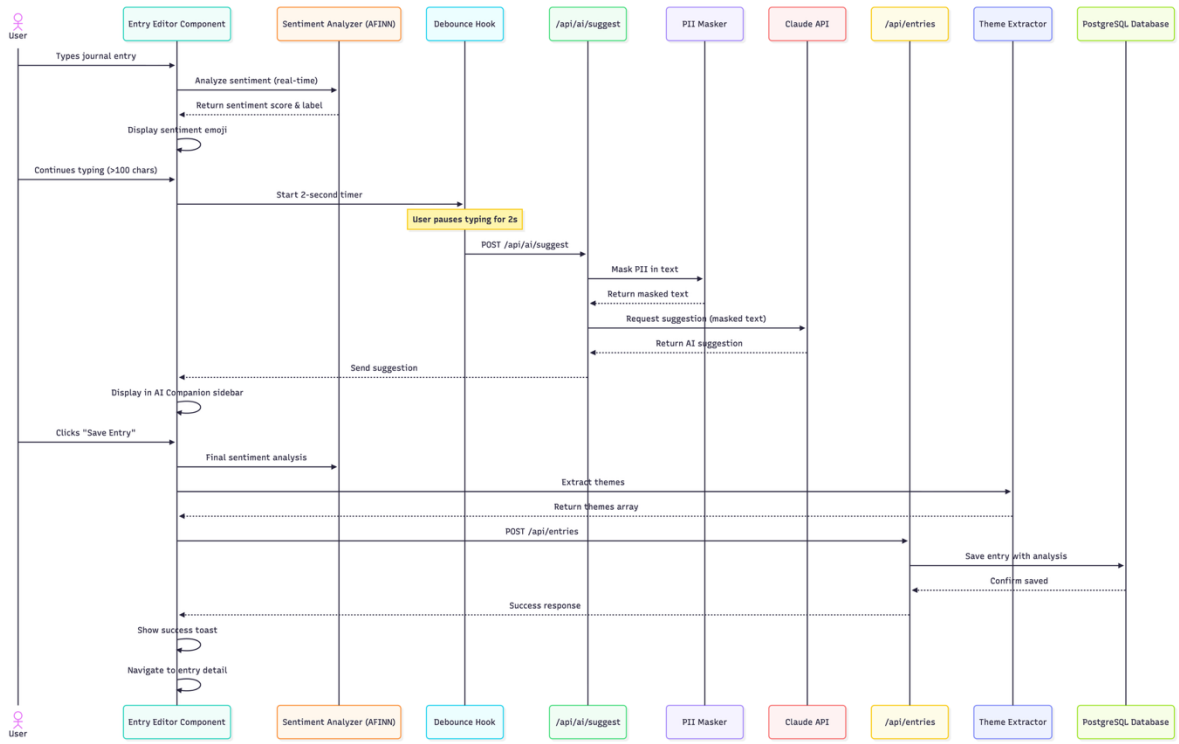
- Entry → Analysis: One-to-zero-or-one relationship. An entry can have an optional detailed analysis.

- Entry → AISuggestion: One-to-many relationship. An entry can receive multiple AI suggestions.

2. Sequence Diagrams

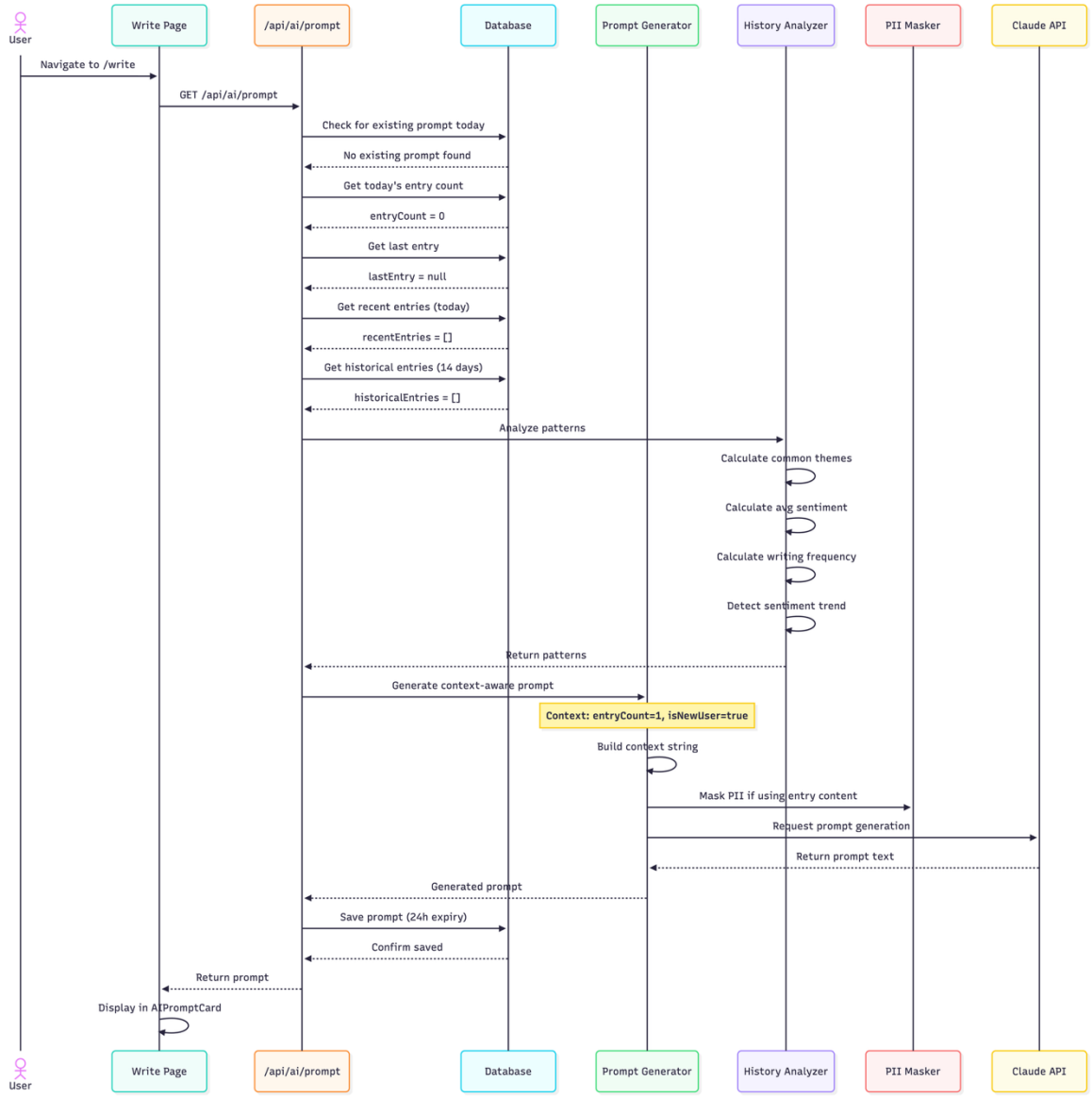
2.1. **Entry Creation with Real-Time Companion**

This sequence diagram shows the complete flow when a user creates a journal entry with AI companion suggestions.



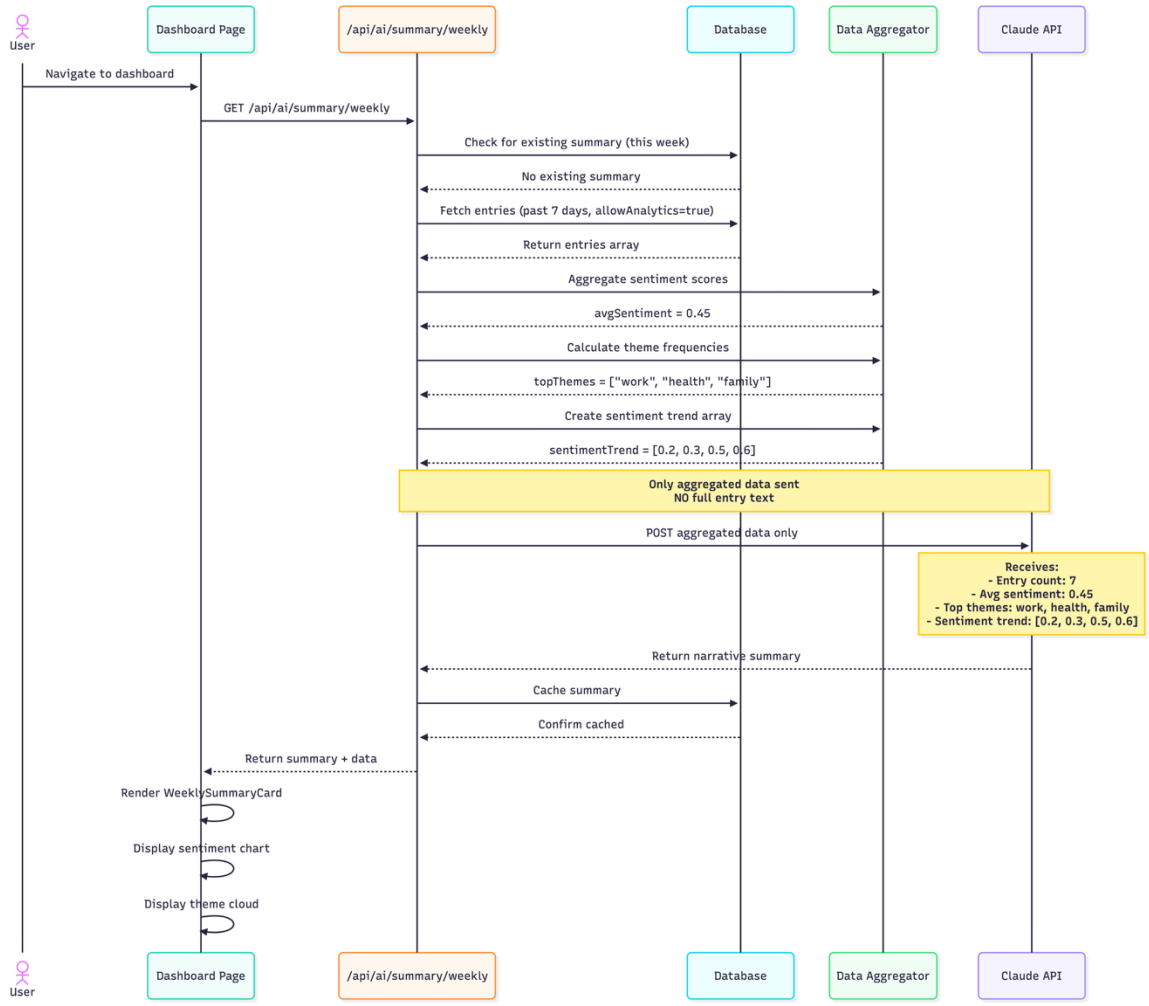
2.2 **Context-Aware Prompt Generation**

This sequence shows how the system generates context-aware prompts based on user history.



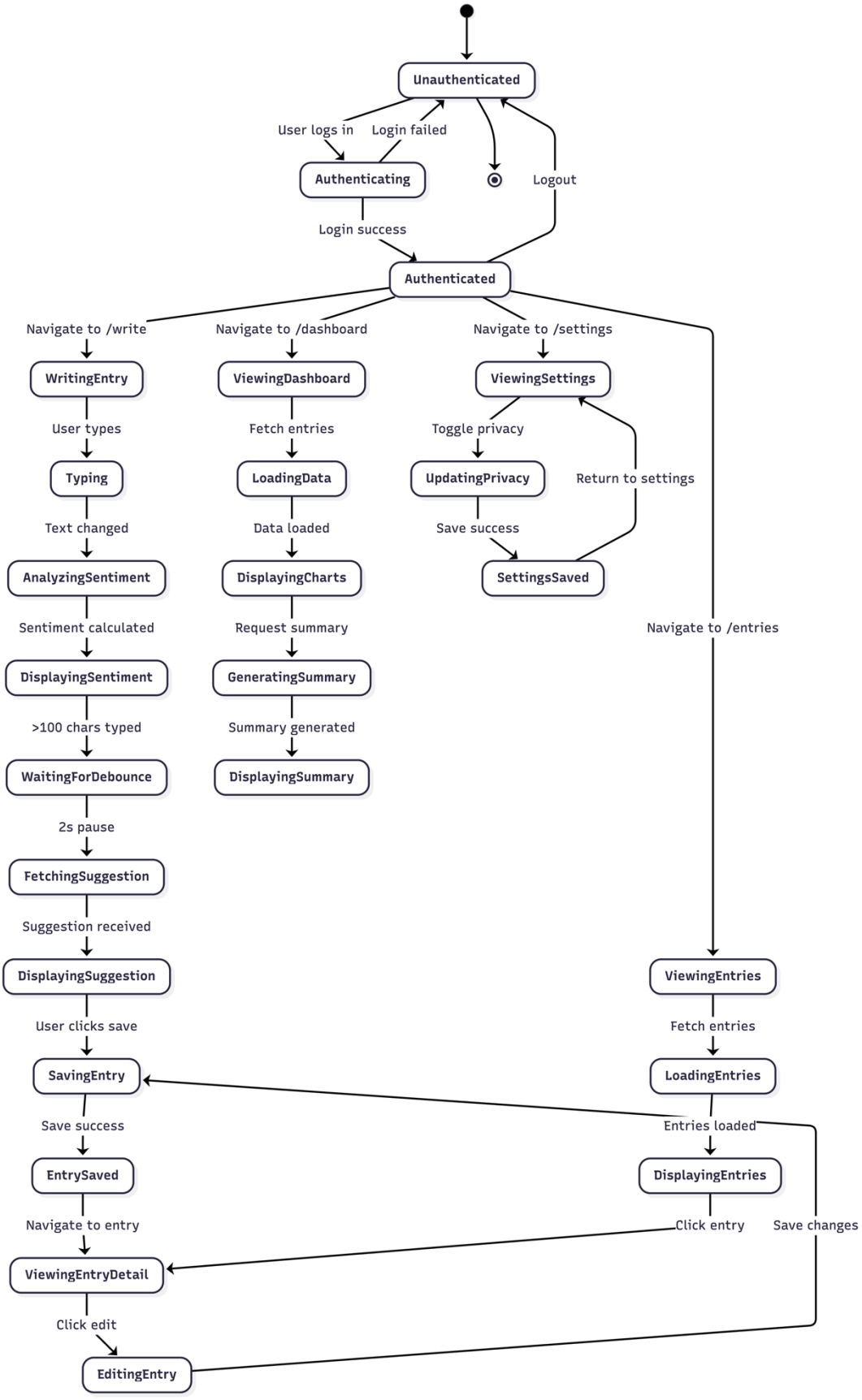
2.3 **Weekly Summary Generation**

This sequence shows how weekly summaries are generated using only aggregated data.



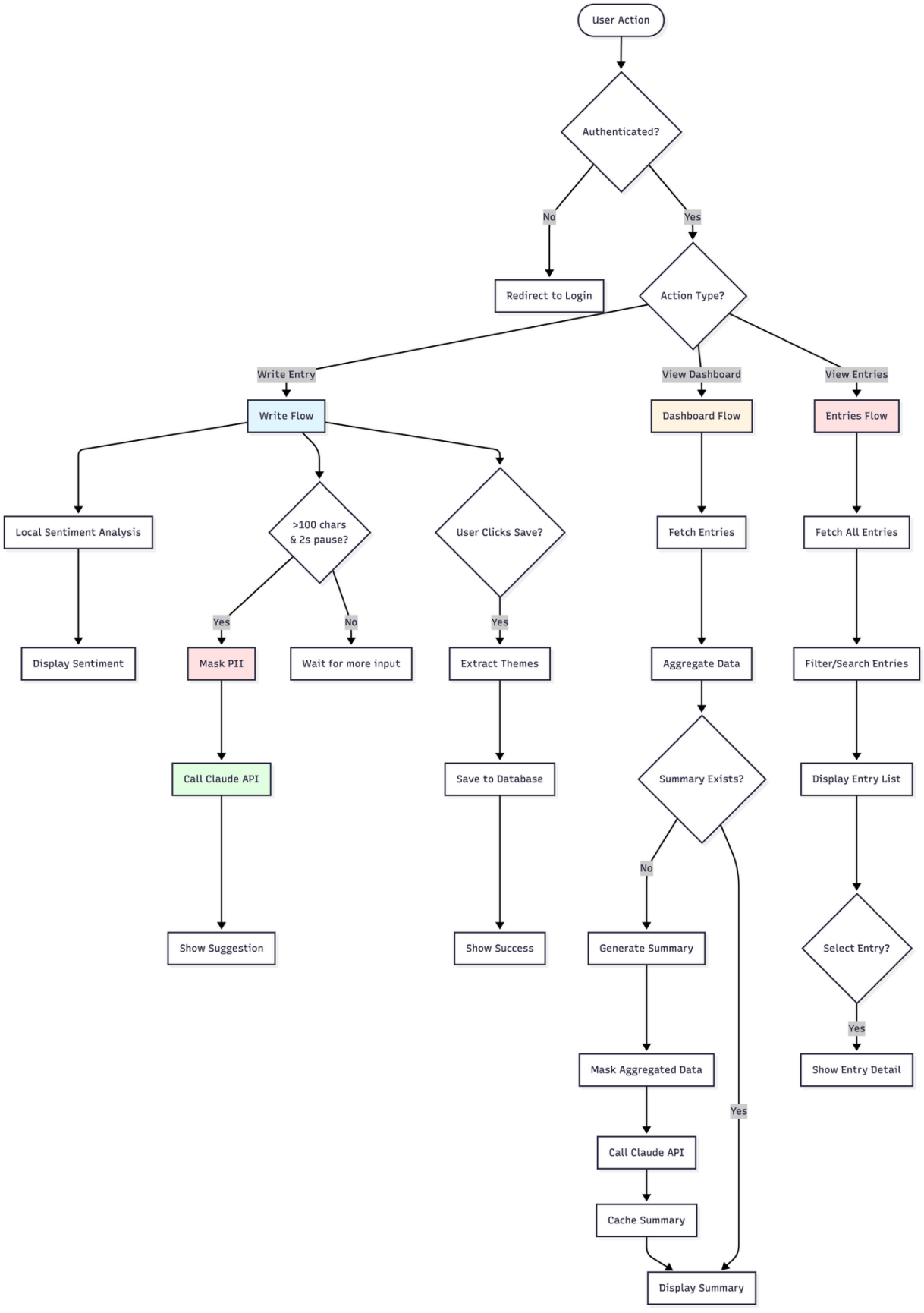
3. State Management Diagram

This diagram shows how state flows through the application.


Unauthenticated
User logs in
Login failed
Authenticating
Logout
Login success
Authenticated
Navigate to /write
Navigate to /dashboard
Navigate to /settings
WritingEntry
ViewingDashboard
ViewingSettings
User types
Fetch entries
Toggle privacy
Typing
LoadingData
UpdatingPrivacy
Return to settings
Text changed
Data loaded
Save success
AnalyzingSentiment
DisplayingCharts
SettingsSaved
Sentiment calculated
Request summary
Navigate to /entries
DisplayingSentiment
GeneratingSummary
>100 chars typed
Summary generated
WaitingForDebounce
DisplayingSummary
2s pause
FetchingSuggestion
Suggestion received
DisplayingSuggestion
ViewingEntries
User clicks save
Fetch entries
SavingEntry
LoadingEntries
Save success
Entries loaded
EntrySaved
DisplayingEntries
Navigate to entry
Click entry
Save changes
ViewingEntryDetail
Click edit
EditingEntry

4. **Data Flow Diagram**

This diagram shows how data flows through the system for different operations.


User Action
Authenticated?
No
Yes
Redirect to Login
Action Type?
Write Entry
View Dashboard
View Entries
Write Flow
Dashboard Flow
Entries Flow
Local Sentiment Analysis
>100 chars & 2s pause?
User Clicks Save?
Fetch Entries
Fetch All Entries
Yes
No
Yes
Display Sentiment
Mask PII
Wait for more input
Extract Themes
Aggregate Data
Filter/Search Entries
Call Claude API
Save to Database
Summary Exists?
Display Entry List
No
Show Suggestion
Show Success
Generate Summary
Select Entry?
Yes
Mask Aggregated Data
Show Entry Detail
Yes
Call Claude API
Cache Summary
Display Summary

Potential Future Enhancements

1. Sentiment-Based Activity Recommendations

- Music Recommendations: Suggest calming music when user is stressed, energizing music when mood is low

- Interactive Games: Mindfulness games, breathing exercises, puzzles to help manage stress

- Doctor/Professional Recommendations: Detect depression patterns and suggest mental health resources, therapists, or crisis hotlines when needed

- Wellness Activities: Meditation guides, gratitude exercises, and relaxation techniques based on sentiment

2. Mobile Applications

- Native iOS and Android apps with offline journaling support

- Push notifications for daily prompts and reminders

- Quick entry widgets for home screen

- Voice-to-text journaling capabilities

3. Voice & Multimedia Journaling

- Voice journaling with speech-to-text conversion

- Photo and video entries

- Audio sentiment analysis from voice tone

- Visual timeline with media attachments

4. Social Features (Privacy-First)

- Share prompts with community
- Share entries with trusted friends or family
- Anonymous support groups by theme
- Couples or family journaling spaces