# ANCHOR APP

## Comprehensive Handoff Document

## Technical Specification & Development Guide

| | |
|---|---|
| **Version:** | 1.0 |
| **Date:** | January 5, 2026 |
| **Purpose:** | Complete technical specification for development with Claude Code |
| **Stack:** | React Native + Node.js + PostgreSQL + AI |
| **Timeline:** | 10-12 weeks MVP to full product |

# TABLE OF CONTENTS

# 1. EXECUTIVE SUMMARY

## 1.1 What is Anchor?

Anchor is a premium 'Intentional Living' mobile app that transforms written intentions into powerful visual symbols (sigils) using traditional chaos magick methodology combined with modern AI art generation. Users create personal 'anchors' - symbols charged with focused intention that serve as daily focus tools for manifestation and goal achievement.

## 1.2 Core Value Proposition

- **Mainstream positioning:** 'Visual Goal Setting' and 'Personal Anchoring' (not occult-branded)
- **Target market:** Athletes, entrepreneurs, faith-driven individuals, wellness enthusiasts
- **Differentiation:** AI-enhanced mystical art generation, multi-sensory activation (visual + vocal mantras), ritual-based practice

## 1.3 Business Model

**Freemium:**

- Free: 2 anchors, 3 AI styles, basic features
- Pro: $4.99/month - unlimited anchors, 12+ AI styles, Manual Forge, advanced features
- Merch: Print-on-demand products (hoodies, keychains, prints) with user's symbols

## 1.4 Key Metrics for Success

- User retention (Day 7, Day 30)
- Activation frequency (daily active practice)
- Conversion rate (Free → Pro)
- Merch attachment rate
- NPS score

# 2. PRODUCT VISION

## 2.1 User Journey

**New User** → Onboarding (5 slides) → Create First Anchor → Daily Practice → Track Progress → Expand Practice

**Create First Anchor:**

- Enter intention: 'Close the deal'
- See distillation: CLOSE THE DEAL → CLOSTHD
- Traditional sigil generation (3 variations)
- Choose favorite
- AI enhancement (intelligent symbol selection)
- Create mantra: 'Clo-Seth-Da'
- Charge with ritual (Quick or Deep)

**Daily Practice:**

- Morning activation ritual
- Mantra chanting
- Visualize goal

## 2.2 Core Philosophy

**'Trojan Horse for Sigil Magick'** - Teach traditional practice through modern UX, respect intellectual property and methodology, make powerful tools accessible without mystical baggage, and prioritize user agency and creative control.

# 3. TECHNICAL ARCHITECTURE

## 3.1 Tech Stack

### *Frontend (Mobile App)*

- Framework: React Native
- Language: TypeScript
- UI Components: React Native Paper + Custom
- Navigation: React Navigation 6.x
- State Management: Zustand
- Drawing Canvas: @shopify/react-native-skia
- Storage: SQLite + AsyncStorage
- Authentication: Firebase Auth
- Payments: RevenueCat

### *Backend (API Server)*

- Runtime: Node.js 20.x
- Framework: Express.js
- Language: TypeScript
- Database: PostgreSQL 15+
- Cache: Redis 7+
- ORM: Prisma
- File Storage: Cloudflare R2
- Hosting: Railway or Render

### *AI Engine (Separate Service)*

- Framework: FastAPI (Python)
- AI Models: Stable Diffusion XL (via Replicate API)
- NLP: Compromise.js for intention analysis
- TTS: Google Text-to-Speech for mantras
- Vector Graphics: Potrace (SVG generation)

## 3.2 System Architecture

The system follows a three-tier architecture: Mobile App (React Native) → Backend API (Node.js/Express) → AI Engine (Python/FastAPI). The mobile app handles local operations like letter distillation and traditional sigil generation, while the backend coordinates AI enhancement and manages persistent storage.

# 4. DATABASE SCHEMA

## 4.1 PostgreSQL Tables

### *Core Tables:*

- **users** - User accounts, subscription status, streaks
- **anchors** - Main anchor objects with sigils, mantras, stats
- **activations** - Tracks every activation event
- **charges** - Tracks charging rituals
- **burned_anchors** - Archive of deleted anchors

### *Social & Discovery:*

- **shared_anchors** - Publicly shared anchors for Discover feed

### *E-commerce:*

- **orders** - Merch orders with Printful integration

### *Settings & Sync:*

- **user_settings** - Notification preferences, defaults
- **sync_queue** - Offline mode pending actions

## 4.2 Key Relationships

- users → anchors (one-to-many)
- anchors → activations (one-to-many)
- anchors → charges (one-to-many)
- users → orders (one-to-many)

## 4.3 Redis Cache Structure

- user:{user_id}:anchors - List of anchor IDs
- user:{user_id}:stats - Cached stats JSON
- anchor:{anchor_id}:details - Full anchor object
- discover:trending:{category} - Trending anchors

# 5. CORE ALGORITHMS

## 5.1 Letter Distillation Algorithm

Implements the Austin Osman Spare method for sigil creation:

1. Remove spaces from intention text
2. Remove all vowels (a, e, i, o, u)
3. Remove duplicate consonants (keep first occurrence)
4. Return uppercase letters array

**Example:** 'Close the deal' → CLOSETHEDEAL → CLSTHD → C L O S T H D (final letters)

```
File: utils/sigil/distillation.ts
```

## 5.2 Traditional Sigil Generator

Merges distilled letters into geometric patterns using vector path analysis:

1. Convert each letter to vector paths (SVG)
2. Analyze natural connection points between letters
3. Merge letters by overlapping shared strokes
4. Generate 3 stylistic variations (dense, balanced, minimal)
5. Output as SVG

**Connection Types:** shared_vertical, shared_horizontal, nested, overlapping

```
Files: utils/sigil/traditional-generator.ts, utils/sigil/letter-vectors.ts
```

## 5.3 Intention Analysis Algorithm

Uses NLP to extract themes and intelligently select mystical symbols:

1. Extract keywords using Compromise.js (nouns, verbs, adjectives)
2. Map keywords to archetypal themes (wealth, success, love, clarity, etc.)
3. Select appropriate symbols from database based on themes
4. Determine aesthetic approach (grimoire, minimal, cosmic, etc.)
5. Generate human-readable explanation of choices

**Symbol Categories:** Planetary seals (Jupiter, Mercury, etc.), Elder Futhark runes, Elemental symbols, Sacred geometry, Lunar phases

```
File: backend/services/IntentionAnalyzer.ts
```

## 5.4 Mantra Generator

Creates pronounceable mantras from distilled letters:

- **Syllabic:** Groups letters into 2-letter syllables (C-LO-ST-HD)
- **Rhythmic:** Groups into 3-letter chunks with pauses (CLO / STH / D)
- **Letter-by-letter:** Individual letter pronunciation (CEE-ELL-OH-ESS...)

User can record their own voice or use AI-generated audio via Google Text-to-Speech.

# 6. FEATURE SPECIFICATIONS

## 6.1 Authentication Flow

- Firebase Auth with email and Google sign-in
- 5-slide onboarding for new users
- Skip option available
- Auto-create user profile on first sign-in

## 6.2 Anchor Creation Flow

**Complete 11-step flow:**

1. IntentionInput - User enters intention text
2. DistillationAnimation - Show letter removal process
3. SigilVariationPicker - Choose from 3 traditional variations
4. EnhancementChoice - AI Decide / Traditional / Manual Forge
5. AIAnalysis - Show selected symbols and rationale
6. AIGenerating - Processing screen (30-60 seconds)
7. VariationPicker - Choose from 4 AI-enhanced versions
8. MantraCreation - Generate 3 mantra styles
9. ChargingChoice - Quick (30s) vs Deep (5 phases)
10. ChargingRitual - Guided ritual with haptics
11. AnchorComplete - Success, save to vault

## 6.3 Charging Rituals

**Quick Charge (30 seconds):**

- Display symbol full-screen
- Countdown timer with haptic pulses every 5 seconds
- Simple focus on intention

**Deep Charge (5 phases, ~5 minutes):**

- Phase 1: Breathwork (30s) - Breathing guide
- Phase 2: Mantra (60s) - Audio playback, chanting
- Phase 3: Visualization (90s) - Text prompts
- Phase 4: Transfer (30s) - Touch interaction
- Phase 5: Seal (90s) - Hold with haptic feedback

## 6.4 Vault Organization

**Categories:** All, Career, Health, Wealth, Relationships, Custom
**Sorting:** Recent, Most Activated, Least Activated, Oldest, Name, Status

**Filters:** All, Charged, Uncharged, Archived (Pro only)
**Views:** Grid (2 columns) or List
**Search:** Full-text search across intention text

## 6.5 Activation Rituals

- Simple activation: View symbol for 10 seconds with haptics
- Mantra activation: Chant mantra 7 times
- Deep activation: Full visualization ritual
- All activations tracked for stats and streaks

## 6.6 Burning Ritual

- Double confirmation required
- User must type 'RELEASE' to confirm
- Fire animation during deletion
- Anchor archived to burned_anchors table
- Cannot be undone

# 7. UI/UX DESIGN SYSTEM

## 7.1 Color Palette ('Zen Architect' Theme)

| Color Name | Hex Code | Usage |
|---|---|---|
| Charcoal | #1A1A1D | Primary background |
| Navy | #0F1419 | Secondary background |
| Gold | #D4AF37 | Primary accent, CTAs |
| Bone | #F5F5DC | Primary text |
| Deep Purple | #3E2C5B | Accent |
| Bronze | #CD7F32 | Secondary accent |
| Silver | #C0C0C0 | Secondary text |

## 7.2 Typography

- **Headings:** Cinzel (elegant serif) - Regular, SemiBold, Bold
- **Body:** Inter (clean sans-serif) - Regular, SemiBold
- **Code:** Roboto Mono - Regular

**Size Scale:**

- H1: 32pt / H2: 24pt / H3: 20pt / H4: 18pt
- Body: 16pt (body1), 14pt (body2)
- Caption: 12pt / Button: 16pt

## 7.3 Spacing System

- XS: 4px / SM: 8px / MD: 16px / LG: 24px
- XL: 32px / XXL: 48px / XXXL: 64px

## 7.4 Component Library

**Key Components:**

- Button (4 variants: primary, secondary, outline, ghost)
- AnchorCard - Display anchor in vault
- TextInput - Custom styled input fields
- Modal - Full-screen and bottom sheet variants
- PhaseTimer - For ritual countdown displays
- CategoryTabs - Horizontal scrolling tabs

## 7.5 Animation Principles

- Duration: 200-300ms for UI, 500-800ms for transitions
- Easing: ease-in-out for most, ease-out for entrances
- Haptic feedback on all important interactions
- Loading states for all async operations

# 8. API ENDPOINTS

## 8.1 Authentication

```
POST /api/auth/register - Create new account

POST /api/auth/login - Sign in

POST /api/auth/refresh - Refresh token

GET /api/auth/me - Get current user
```

## 8.2 Anchors

```
POST /api/anchors - Create new anchor

GET /api/anchors - List user's anchors

GET /api/anchors/:id - Get anchor details

PUT /api/anchors/:id - Update anchor

DELETE /api/anchors/:id - Delete anchor

POST /api/anchors/:id/analyze - Analyze intention

POST /api/anchors/:id/enhance - Generate AI versions

PUT /api/anchors/:id/select-version - Select final

POST /api/anchors/:id/charge - Mark as charged

POST /api/anchors/:id/activate - Log activation

POST /api/anchors/:id/burn - Burn ritual
```

## 8.3 Discover Feed

```
GET /api/discover/trending - Trending intentions

GET /api/discover/featured - Featured collection

GET /api/discover/style/:style - Filter by style

POST /api/discover/:id/save - Save to inspiration
```

## 8.4 Shop (Merch)

```
GET /api/shop/products - List products

POST /api/shop/mockup - Generate mockup
```

```
POST /api/shop/orders - Create order

GET /api/shop/orders - List user orders

GET /api/shop/orders/:id/tracking - Tracking info
```

## 8.5 Subscriptions

```
GET /api/subscription/status - Current status

POST /api/subscription/create - Subscribe

POST /api/subscription/cancel - Cancel

POST /api/subscription/restore - Restore purchase
```

## 8.6 User & Stats

```
GET /api/user/profile - User profile

PUT /api/user/profile - Update profile

GET /api/user/stats - Stats dashboard

GET /api/user/settings - Get settings

PUT /api/user/settings - Update settings
```

# 9. THIRD-PARTY INTEGRATIONS

## 9.1 Stable Diffusion (via Replicate API)

Primary AI engine for generating enhanced sigil artwork. Uses Stable Diffusion XL with ControlNet for structure preservation.

- **Model:** stability-ai/sdxl
- **Cost:** ~$0.01 per image (4 variations = $0.04 per anchor)
- **Generation time:** 15-30 seconds per image
- **Quality:** 4K resolution output

## 9.2 Firebase Services

**Authentication:** Email/password and Google sign-in
**Cloud Messaging:** Push notifications for iOS and Android

- Daily activation reminders
- Streak protection alerts
- Order tracking updates

## 9.3 RevenueCat (Subscriptions)

Handles all subscription logic, receipt validation, and cross-platform purchases.

**Products:**

- anchor_pro_monthly - $4.99/month
- anchor_pro_annual - $49.99/year (save 17%)

- Free tier: 2 anchors, 3 AI styles
- Pro tier: Unlimited anchors, 12+ styles, Manual Forge

## 9.4 Printful (Print-on-Demand Merch)

Fulfillment partner for custom merchandise with user's anchor symbols.

**Products:**

- Hoodies ($45-55)
- T-shirts ($25-30)
- Keychains ($12-15)
- Canvas prints ($40-80)
- Phone cases ($20-25)

- Mockup generation API for product previews
- Order creation and tracking webhooks
- Automatic shipping calculations

## 9.5 Cloudflare R2 (File Storage)

S3-compatible object storage for all generated images and assets.

- Anchor symbol images (SVG and raster)
- AI-enhanced versions
- Mantra audio files
- User voice recordings
- Product mockups

# 10. DEVELOPMENT PHASES

## Phase 0: Project Setup & Architecture (Week 1)

- Initialize React Native and backend projects
- Set up database schema and migrations
- Configure Firebase Auth and RevenueCat
- Set up development environment and CI/CD

## Phase 1: MVP Core Features (Weeks 2-4)

**Goal:** Users can create, charge, and activate basic anchors

- Authentication and onboarding
- Letter distillation algorithm
- Traditional sigil generator (3 variations)
- Basic vault with grid view
- Quick charge ritual (30 seconds)
- Simple activation with haptics

## Phase 2: AI Enhancement (Weeks 5-6)

**Goal:** Intelligent AI-powered symbol generation

- Intention analysis system (NLP)
- Symbol selection database
- Stable Diffusion integration via Replicate
- AI enhancement UI flow
- 4-variation picker

## Phase 3: Advanced Features (Weeks 7-9)

**Goal:** Full feature set for power users

- Mantra generation and TTS
- Deep charge ritual (5 phases)
- Manual Forge (drawing canvas)
- Burning ritual
- Vault organization (categories, search, filters)
- Discover feed (trending, featured)

## Phase 4: Monetization & Polish (Weeks 10-12)

**Goal:** Production-ready app with monetization

- RevenueCat subscription integration

- Printful merch shop
- Push notifications system
- Offline mode capabilities
- Stats dashboard
- Testing, optimization, and polish

**Total Timeline:** 10-12 weeks from start to production launch

# 11. CODE STANDARDS

## 11.1 TypeScript Guidelines

- Always use interfaces for component props
- Use type for unions and intersections
- Explicit return types for all functions
- Use optional chaining (?.) and nullish coalescing (??)
- Strict mode enabled in tsconfig.json

## 11.2 File Naming Conventions

- PascalCase for components: Button.tsx, AnchorCard.tsx
- camelCase for utilities: distillation.ts, api-client.ts
- kebab-case for assets: icon-gold-star.png
- SCREAMING_SNAKE_CASE for constants: API_BASE_URL

## 11.3 Project Structure

```
src/screens/ - Full-screen views

src/components/ - Reusable UI components

src/navigation/ - React Navigation setup

src/services/ - Business logic, API calls

src/utils/ - Pure functions, helpers

src/hooks/ - Custom React hooks

src/theme/ - Colors, typography, spacing

src/types/ - TypeScript types
```

## 11.4 Git Workflow

**Commit message format:**

- feat: Add letter distillation algorithm
- fix: Resolve charging ritual timer bug
- refactor: Improve sigil generator performance
- docs: Add API endpoint documentation
- test: Add unit tests for distillation
- style: Format code with Prettier

**Branch naming:**

- feature/letter-distillation
- bugfix/timer-not-stopping
- hotfix/production-crash

## 11.5 Code Quality

- ESLint for linting (Airbnb style guide)
- Prettier for formatting
- Husky for pre-commit hooks
- Minimum 80% test coverage for critical paths

# 12. TESTING STRATEGY

## 12.1 Unit Tests (Jest)

**Test all core algorithms:**

- Letter distillation
- Traditional sigil generation
- Mantra generation
- Intention analysis

**Ex**

```
describe('distillIntention', () => {

it('should remove vowels', () => {...})

it('should remove duplicates', () => {...})

it('should return uppercase', () => {...})

})
```

## 12.2 Integration Tests

**Test complete flows:**

- User sign up → onboarding → first anchor creation
- Create anchor → charge → activate → view stats
- Upgrade to Pro → access premium features
- Create anchor → order merch → track shipment

## 12.3 E2E Tests (Detox)

**Critical user paths:**

- Complete anchor creation flow (all 11 steps)
- Subscription purchase and activation
- Merch order placement
- Offline mode sync

## 12.4 Manual Testing Checklist

- All animations smooth (60fps)
- Haptic feedback working correctly
- Audio playback (mantras) without glitches
- Images load and cache properly
- Network error handling (offline scenarios)
- Push notifications delivered

## 12.5 Performance Testing

- App launch time < 2 seconds
- Screen transitions < 300ms
- List scrolling at 60fps with 100+ anchors
- Memory usage < 150MB on mid-range devices
- Battery impact < 5% per hour of active use

# 13. DEPLOYMENT

## 13.1 Environment Setup

**Ba**

```
DATABASE_URL=postgresql://...

REDIS_URL=redis://...

JWT_SECRET=...

REPLICATE_API_TOKEN=...

PRINTFUL_API_KEY=...

CLOUDFLARE_R2_ACCESS_KEY=...
```

**Fr**

```
FIREBASE_API_KEY=...

REVENUECAT_IOS_KEY=...

REVENUECAT_ANDROID_KEY=...

API_BASE_URL=https://api.anchorapp.com
```

## 13.2 Hosting Recommendations

**Backend:** Railway or Render
- Automatic deployments from GitHub
- Built-in PostgreSQL and Redis
- Environment variable management
- Cost: ~$20-30/month for MVP

**Database:** Managed PostgreSQL
- Daily automated backups
- High availability setup
- Connection pooling

**File Storage:** Cloudflare R2
- S3-compatible API
- No egress fees
- Global CDN included
- Cost: ~$5-10/month for 100GB

## 13.3 CI/CD Pipeline (GitHub Actions)

**Automated on every push to main:**

1. Run linting and type checking
2. Run unit and integration tests
3. Build backend and frontend
4. Deploy backend to Railway
5. Build mobile app for TestFlight/internal testing
6. Notify team via Slack webhook

## 13.4 App Store Submission

**iOS (App Store):**

- Use Fastlane for automated builds
- TestFlight beta testing before public release
- Review time: 1-3 days typically

**Android (Google Play):**

- Use Fastlane for automated builds
- Internal testing track first
- Review time: 1-3 days typically

# 14. QUICK START GUIDE FOR CLAUDE CODE

## 14.1 Initialize Project

```
# Create React Native app

npx react-native init Anchor --template react-native-template-typescript
```

```
# Install core dependencies

npm install @react-navigation/native zustand

npm install @shopify/react-native-skia

npm install react-native-sqlite-storage

npm install @react-native-firebase/app @react-native-firebase/auth
```

## 14.2 Create Backend

```
mkdir backend && cd backend

npm init -y

npm install express typescript prisma

npx prisma init
```

## 14.3 First Feature: Letter Distillation

**Claude Code Task:**
Create the letter distillation algorithm in TypeScript at src/utils/sigil/distillation.ts following the Austin Osman Spare method: remove spaces, remove vowels (a,e,i,o,u), remove duplicate letters. Include unit tests using Jest.

## 14.4 Development Workflow

**Each Claude Code session should be ONE focused task:**

- ■ Good: 'Create IntentionInputScreen with validation'
- ■ Bad: 'Build the entire anchor creation system'

**For each task, provide:**

1. Task description (from this document)
2. Relevant design specs (copy specific sections)
3. File structure (where it should live)
4. Dependencies (what it connects to)

## 14.5 Build Order

Follow this sequence within each phase:

1. Data layer (models, storage)
2. Business logic (algorithms, services)
3. API (if backend needed)
4. UI components (reusable parts)
5. Screens (assembled components)
6. Navigation (wire screens together)
7. Testing (verify it works)

# 15. CONTACT & SUPPORT

## 15.1 Using This Document

This handoff document is designed to be referenced section-by-section as you build. Each algorithm has detailed implementation notes, and all UI mockups are described in the original conversation history.

## 15.2 Design Decisions

- Color palette and typography: Section 7
- Database schema: Section 4
- API structure: Section 8
- Follow 'Zen Architect' aesthetic throughout

## 15.3 Technical Resources

**Stable Diffusion:**

- Use Replicate API (easiest): https://replicate.com
- Or self-host (requires GPU)
- Cost: ~$0.01 per image

**Printful:**

- Requires business account setup
- API documentation: https://developers.printful.com

**RevenueCat:**

- Free tier available for development
- Documentation: https://docs.revenuecat.com

## 15.4 What's Included

- ■ Complete product vision
- ■ Full technical architecture
- ■ Database schema with all tables
- ■ Core algorithms with pseudocode
- ■ Feature specifications
- ■ UI/UX design system
- ■ API endpoint definitions
- ■ Third-party integration guides
- ■ Development phases (10-12 weeks)
- ■ Code standards and testing strategy

# Good luck building Anchor! ⚓