Product Requirements Document

Conversational Budgeting App — Powered by Sage

A personalized financial operating system — money coach, operating dashboard, and action engine in one.

TARGET USER

Mixed-income young professionals

VERSION

v1.0

DATE

March 2026

STATUS

ACTIVE DEVELOPMENT

1

Vision & Core Belief

> 🎯 We are not building another budget tracker. We are building a personalized financial operating system.

The Core Belief

Most leading budgeting apps (YNAB, Monarch, Copilot) are **productivity tools**, not **behavior-change systems**. They're great at tracking money, but they don't actually change how people make financial decisions.

We are building something different. A system that combines three elements:

- 🎯 **Money Coach** — Sage answers your specific financial questions in natural language
- 🎯 **Operating Dashboard** — One-glance clarity on your complete financial situation
- ⚡ **Action Engine** — Tells you the one thing you should do right now to improve your finances

This combination creates a system that doesn't just track money—it changes behavior.

2

The Team

Vibe-Coding Lead

Building the entire app using AI tools

Using Claude Code and MagicPatterns to ship product at 10x speed. Every pixel, every API call, every component built with AI assistance.

Accountant Cofounder

Financial logic & expert validation

Providing all financial formulas, rules, and framework expertise. The irreplaceable human who ensures our product is mathematically sound and financially responsible.

Product & GTM Lead

Product strategy & go-to-market

Leading product strategy, user research, and go-to-market execution. Owns roadmap prioritization and customer discovery.

3

Four Big Bets

These are the core bets that differentiate us from every other budgeting app.

1

Permission to Spend, Not Restriction

Users want clarity, not guilt. The question isn't "Why did I spend $87?" — it's "What can I safely spend today without screwing up next month?" Our daily safe-to-spend number is the hero feature and emotional core. It reframes budgeting from restriction to empowerment.

2

Mixed-Income First

No budgeting app serves people with salary + side hustles + freelance + dividends well. Every existing app assumes a single paycheck. Our operating margin metric consolidates all income streams into one clear number: "After all my income, what's left?"

3

Action Over Information

Instead of dashboards full of charts and numbers, we answer one question: "What should I do right now?" A single prioritized action (pay off this debt, save $50 more this month, cancel that subscription) makes us a behavior-change system, not a data visualization tool.

4

True Personalization

A student, a new grad, a freelancer, and a parent with dual income have wildly different financial realities. Our app adapts its categories, rules, frameworks, and entire UI to who you are. Not one-size-fits-all. Personalized from day one.

4

Feature Overview — 22 Features Across 6 Pillars

All features organized by pillar, color-coded for easy scanning.

Pillar A — Onboarding & Personalization

4 FEATURES

- ✓ **Framework Selection** — Choose budgeting philosophy (50/30/20, Envelope, Pay Yourself First)
- ✓ **Life-Stage Archetypes** — Identify as Student, New Grad, Freelancer, or Dual-Income Parent
- ✓ **Savings vs Lifestyle Sliders** — User controls priority trade-offs in real time
- ✓ **Guided Onboarding with WHY** — Every decision includes education, not just forms

Pillar B — Transaction Ingestion

1 FEATURE

- ✓ **CSV Upload & Auto-Categorization** — Upload bank exports, auto-categorize, learn from user corrections. Plaid added in Phase 2.

Pillar C — Core Budget Engine

7 FEATURES

- ✓ **AI-Generated Starter Budget** — Based on actual spending patterns
- ✓ **Cash Flow Calculation** — Month-by-month income vs expenses
- ✓ **Daily Safe-to-Spend** — (Income − Fixed − Savings) ÷ Days Remaining
- ✓ **True Operating Margin** — Consolidated view of all income streams
- ✓ **Fixed vs Variable Breakdown** — Understand your cost structure
- ✓ **Irregular Income Handling** — 3-month rolling average or conservative baseline
- ✓ **Multiple Income Stream Tracking** — Salary, side gigs, freelance, dividends in one view

Pillar D — Tracking & Monitoring

4 FEATURES

- ✓ **Overspend Alerts** — With encouraging framing, not guilt
- ✓ **Subscription Detection** — Auto-flag recurring charges, find unused subs
- ✓ **Debt Balance & Trend Tracking** — Monitor paydown progress
- ✓ **Progress vs Survival Status** — Am I winning or barely surviving?

Pillar E — Savings & Investment

2 FEATURES

- ✓ **Savings Allocation Recommendations** — With user override controls
- ✓ **Investable Surplus Guidance** — "You have $X available to invest monthly" (with disclaimers)

Pillar F — Engagement & Insights

4 FEATURES

- ✓ **Weekly Check-In Summaries** — Email & push notifications with trends
- ✓ **Behavioral Insights Engine** — "Your spending increased 20%", "You're on pace to save 15%"
- ✓ **Priority Action Engine** — Single CTA: "What should I do right now?"
- ✓ **Anonymous Benchmarking & Social Sharing** — "You spend less on dining than X% of users"

---

5 Priority Tiers

Features organized by priority to launch. **P0** is absolute must-have. **P1** is where our moat lives. **P2** and **P3** are future phases.

7 CORE FEATURES

CRITICAL

P0 — Must Have

- ✓ Transaction ingestion (CSV upload)
- ✓ AI-generated starter budget
- ✓ Framework selection & onboarding
- ✓ Cash flow calculation engine
- ✓ Daily safe-to-spend number
- ✓ Overspend detection & alerts
- ✓ Dashboard hero view

12 FEATURES

DIFFERENTIATION

P1 — Should Have (Our Moat)

- ✓ User personalization profiles
- ✓ Behavioral insights engine
- ✓ Subscription tracking
- ✓ Priority action engine
- ✓ Irregular income handling
- ✓ Multiple income stream tracking
- ✓ True operating margin metric
- ✓ Debt tracking & trends
- ✓ Savings recommendations
- ✓ Fixed vs Variable breakdown
- ✓ Progress vs Survival status
- ✓ Sage conversational AI bar

2 FEATURES

POLISH

P2 — Nice to Have

- ✓ Weekly check-in system (email + push)
- ✓ Investment allocation guidance

1 FEATURE

LATER

○ P3 — Future

- ✓ Anonymous benchmarking & social sharing

6

Key Feature Deep Dives

Detailed breakdown of 7 critical features. Click each to expand.

▶ **Transaction Ingestion & Auto-Categorization**

What it Does

Users upload bank CSV files or connect via Plaid (Phase 2). System parses transactions, auto-categorizes them using rule-based logic, and improves categorization accuracy with every user correction.

⚡ Why it Matters

This is the foundation of everything else. Without accurate transaction data, every metric (cash flow, safe-to-spend, savings rate) is wrong. It's also the #1 UX risk — if categorization is off, users lose trust in the entire system.

Key Requirements

- ✓ CSV upload supporting top 10 bank formats (Chase, BofA, Wells Fargo, Capital One, Amex, Schwab, etc.)
- ✓ Rule-based auto-categorization with 200+ seed rules
- ✓ Persistent user correction system that improves accuracy over time
- ✓ Duplicate detection & merging
- ✓ Date range filtering
- ✓ Plaid API integration (Phase 2)

Risk Level

HIGH — CSV formats vary wildly. Even top 5 banks have quirky date formats, encoding issues, and column variations. Categorization accuracy directly impacts user trust.

Level of Effort

3 weeks — Parser development, seed rules curation, UI for corrections, test coverage

Accountant Input Required

Critical. Needs validation on categorization rules. Some transactions are ambiguous (is "Amazon" a business expense or personal? Is "DoorDash" dining or groceries?). Accountant expertise prevents categorization errors.

▶ AI-Generated Starter Budget

What it Does

After uploading 2-3 months of transactions, the app generates a suggested budget based on actual spending patterns and the user's chosen budgeting framework. Users can accept with one click or edit each category.

⚡ Why it Matters

The "Wow Moment." Users download budgeting apps because they don't know how to budget, yet most apps require users to figure out their budget themselves (YNAB, Copilot). We do it for them. This is a huge adoption driver.

Key Requirements

- ✓ Budget generation algorithm (weighted averages per category, framework-aware allocation)
- ✓ Category allocation rules for each framework (50/30/20, Envelope, PYF)
- ✓ Editable budget UI with drag-to-adjust sliders
- ✓ One-click accept
- ✓ Educational tooltips explaining allocation logic

Risk Level

MEDIUM — Algorithm needs to balance realism with aspirational budgeting. Too lenient and users don't change behavior. Too strict and they reject it.

Level of Effort

2 weeks — Algorithm design, UI/UX, A/B testing different weighting approaches

Accountant Input Required

High. Needs financial expertise to set realistic allocation percentages and handle edge cases (e.g., what happens for users with very high savings rates or very tight margins?).

Framework Selection & Creator Partnerships

What it Does

During onboarding, users pick a budgeting philosophy. The app's behavior adapts: budget allocation percentages, category names, educational messaging. Frameworks include 50/30/20 Rule, Envelope Method, Pay Yourself First, plus creator partnerships (Ramit Sethi, Dave Ramsey, Barefoot Investor).

⚡ Why it Matters

Different people respond to different frameworks. A minimalist responds to Envelope. A growth-focused person responds to Pay Yourself First. Creator partnerships add credibility and community (users feel they're using Ramit's system, not some random startup).

Key Requirements

- ✓ 4-6 framework templates with unique allocation rules
- ✓ Framework switcher (users can change anytime)
- ✓ Creator partnership assets (logos, educational blurbs, branded category names)
- ✓ Framework-specific onboarding messaging

Risk Level

LOW — Straightforward UI/config. Main risk is creator partnerships requiring business deal negotiations.

Level of Effort

1.5 weeks — UI design, config system, partnership comms (separate from dev work)

Accountant Input Required

Medium. Needs validation on framework allocation percentages and category mapping.

⚡ Daily Safe-to-Spend (THE Hero Feature)

What it Does

A single number prominently displayed on the dashboard. Calculated as: (Total Available Income − Fixed Monthly Costs − Monthly Savings Targets) ÷ Days Remaining in Month = Safe Daily Spend. Users check this number before any discretionary purchase.

⚡ Why it Matters

This reframes budgeting entirely. Instead of "I overspent and failed," it becomes "I have $23 left today and I need to decide if this coffee is worth it." It answers the emotional core question: "What can I safely spend today without screwing up next month?"

Key Requirements

- ✓ Real-time calculation from current date
- ✓ Month-to-date spending integration

- ✓ Accounts for one-time payments (car insurance, annual subscriptions)
- ✓ Visual indicator (green = healthy, yellow = tight, red = overspent)
- ✓ Tooltip explaining the formula

Risk Level

LOW — Pure math. Biggest risk is edge cases (what if someone is in overspend territory? Show negative? Absolute value?). Accountant handles.

Level of Effort

1.5 weeks — Calculation logic, UI, edge case handling, messaging

Accountant Input Required

Critical. Needs validation on formula and edge cases. Should negative safe-to-spend numbers show as $0, negative, or a warning? This matters for user psychology.

▶ Personalization Profiles (Archetypes)

What it Does

During onboarding, a 5-8 question quiz identifies user's archetype: Student, New Grad, Freelancer, or Dual-Income Parent. Each archetype has preset category weights, savings targets, and UI emphasis. Example: Freelancers see income smoothing options; Students see lower savings expectations.

⚡ Why it Matters

This is Big Bet 4 in action. A student's financial reality is completely different from a dual-income parent's. Personalization means every user sees a tailored system, not a generic one.

Key Requirements

- ✓ 5-8 question archetype quiz
- ✓ Preset category weights per archetype
- ✓ Savings vs Lifestyle sliders (recalculate in real time)
- ✓ Ability to change archetype anytime
- ✓ Archetype-specific educational content & examples

Risk Level

LOW — Quiz logic is straightforward. Main work is UX design and archetype definition.

Level of Effort

2.5 weeks — Quiz design, slider interaction, config system, copy creation

Accountant Input Required

High. Needs expertise defining archetypes and setting realistic savings targets for each. A student's savings goal is different from a new grad's.

▶ Behavioral Insights & Priority Actions

What it Does

Rule-based insights engine generates 10-15 behavioral insights (spending up 20%+, on pace for savings goal, approaching budget limit, positive streak, unused subscription detected). Natural language coaching

generated via Claude API Sonnet. A single "Priority Action" card surfaces the one thing user should do right now.

⚡ Why it Matters

This IS Big Bet 3: Action Over Information. Most apps show you data. We tell you what to do. A single CTA saying "Cancel this unused subscription and save $15/month" creates behavior change. That's our moat.

Key Requirements

- ✓ 10-15 rule-based insights (spending trends, savings pace, limit proximity, subscription detection)
- ✓ Claude API Sonnet for natural language NLG of insights and coaching
- ✓ Priority action ranking algorithm (what's most impactful right now?)
- ✓ Insight history/feed so users see patterns over time
- ✓ Actionable CTAs (link to subscription cancellation, link to debt payoff calculator, etc.)

Risk Level

MEDIUM-HIGH — Insight quality directly impacts perceived value. Bad insights erode trust. Good insights drive engagement. Also, NLG quality matters (users judge us by coaching quality).

Level of Effort

3 weeks — Insight rule design, NLG prompts, priority ranking algorithm, A/B testing messaging, analytics instrumentation

Accountant Input Required

High. Needs expertise to define what qualifies as a "problem" (is 20% spending increase always bad? What about planned lifestyle changes?). Rules must be financially sound.

Irregular Income & Multiple Streams

What it Does

Big Bet 2 in action. Instead of assuming a single paycheck, system handles salary + side gigs + freelance + dividends. Income Smoothing: Users choose between 3-month rolling average (optimistic) or lowest-month-as-baseline (conservative). System auto-classifies income sources.

⚡ Why it Matters

YNAB, Monarch, Copilot all assume single stable income. For the 40%+ of people with side hustles, freelance, or investment income, those apps are broken. We serve this underserved market.

Key Requirements

- ✓ Income source classification (Salary, Side Gig, Freelance, Dividends, Other)
- ✓ Income smoothing settings (rolling avg vs conservative baseline)
- ✓ 3-month rolling average calculation
- ✓ Lowest-month detection for conservative forecasting
- ✓ Visual breakdown by income source
- ✓ Transfer detection (prevent transferring own money being classified as income)

Risk Level

MEDIUM — Transfer misclassification is the big risk. A user's transfer from savings to checking could be misread as income if categorization isn't perfect. Needs robust detection.

Level of Effort

2 weeks — Classification algorithm, smoothing logic, UI for source breakdown, transfer detection rules

Accountant Input Required

Critical. Income smoothing logic is financially non-trivial. When is conservative (lowest-month) better than rolling avg? How should tax withholding on freelance income factor in? Accountant expertise is essential.

7

What's Currently Built (MagicPatterns Design Status)

Current design prototype status in MagicPatterns. ▯ = Designed and ready for handoff. ▯ = Gap that needs design work.

▯ **Dashboard with safe-to-spend hero**

▯ **"Ask Sage" conversational AI bar**

▯ **Bills view with paycheck swimlanes**

- Debt tracking cards & metrics
- Cash flow chart visualization
- Category drilldown page
- Transaction row components
- Onboarding flow page
- Profile/settings page
- Plans tab with sub-views

- Insight cards & Sage branding
- Progress arc component
- Bottom tab navigation
- Month selector
- Sage mascot character
- Add transaction sheet
- Framework selection UI

- Creator partnership presets
- Personalization quiz flow
- Savings vs Lifestyle sliders
- CSV upload & bank selection
- Subscription tracking view
- Weekly check-in summary
- Operating margin display

- Fixed vs Variable breakdown
- Progress vs Survival status
- Investment guidance card
- Overspend alert designs
- Priority action CTA card
- Behavioral insights feed
- Anonymous benchmarking UI

Income smoothing settings

Plaid bank connection flow

16 of 49 screens designed and ready (35%)

8

Recommendations & Next Steps

Based on the gap analysis above, here are 7 specific recommendations for the next design sprint(s).

1. Design the Onboarding Decision Points

The onboarding page exists but lacks the critical decision flows.
Prioritize:

- Framework selection cards (visual + compelling copy for each)
- Personalization archetype quiz (5-8 questions with smart UI)
- Savings vs Lifestyle interactive sliders

Why: These decisions are the #1 differentiator during first-impression. Getting them right drives long-term retention.

2. Build the CSV Upload Flow

P0 blocker. This is the data pipeline powering everything.

- Bank selection UI (Chase, BofA, Wells Fargo, etc.)
- CSV upload component
- Column mapping preview & correction
- Categorization review before finalization

Why: Without this, we can't test the core product or onboard beta users.

3. Create the Priority Action Card

This is Big Bet 3 made visual. No UI exists yet.

- Prominent dashboard card (possibly above safe-to-spend)
- Single CTA with clear outcome (e.g., "Cancel unused Netflix, save $18/mo")
- Supporting context and action button

Why: This is our unique value proposition. It needs premium positioning and UX.

4. Design the Subscription Tracker

Currently subscriptions are buried in Bills. They need their own view.

- Separate Subscriptions tab or sub-view in Plans
- "Unused" subscription flagging
- Monthly savings potential calculation
- Quick cancel links

Why: Many users have $50-200/month in zombie subscriptions. This is a quick win.

5. Add Operating Margin & Fixed/Variable Views

These are key metrics from Big Bet 2 with no dedicated UI.

- Operating Margin card (similar to safe-to-spend prominence)
- Fixed vs Variable breakdown (pie chart or bar chart)
- Trend visualization over 3-6 months

Why: These metrics separate us from competitors and educate users on their cost structure.

6. Design Overspend Alert States

Our non-punishing alert system is a key differentiator. Design all states:

- Category approaching limit (warning state)
- Category limit reached (gentle reminder)
- Monthly overspend detected (supportive message, not guilt)
- Push notification templates

Why: Copy and design tone matter. These set the behavior-change tone for the entire product.

7. Plan the Weekly Check-In Experience

This drives retention and behavior change. Design full flow:

- Weekly summary screen (trends, wins, upcoming priorities)
- Email template with key metrics & CTAs
- Push notification teaser
- Week-over-week comparison visuals

Why: Weekly engagement is proven to drive retention. This needs premium design.

9

Technical Architecture

Layered architecture showing tech stack, APIs, and tools used to build this product.

Mobile App

React Native with Expo

iOS + Android from single codebase. Smooth animations with Reanimated 3.

Backend & Database

Supabase (PostgreSQL)

Auth, real-time updates, Row-Level Security, Edge Functions. Zero DevOps.

AI Intelligence

Claude API (Haiku + Sonnet)

Haiku for categorization. Sonnet for coaching NLG. Cost-optimized via batch API.

Push Notifications

Firebase Cloud Messaging

Cross-platform push, deep linking, in-app messaging.

Bank Connection (Phase 2)

Plaid API

Direct bank connections. Pay-as-you-go, free first 200 connections.

AI Dev Tools

Claude Code, MagicPatterns, v0

Full-stack AI development. Design via MagicPatterns, code via Claude Code.

Why This Stack?

- **Cost-Optimized:** No expensive ML infrastructure. Rule-based + Claude API falls back.
- **Speed to Market:** Supabase eliminates backend work. React Native eliminates iOS/Android duplication.
- **Scalability:** Supabase auto-scales. Claude API handles variable workload without infrastructure changes.
- **AI-Native:** Claude API is the entire AI strategy. Easy to upgrade models (Haiku → Sonnet → Opus) without rewriting code.

10

Costs Summary

Breakdown of all infrastructure, API, and tool costs.

Category	Service	Cost	Notes
DEVELOPMENT PHASE (0-16 weeks)			
Backend	Supabase	$25-50/mo	Pro plan during dev. Scales with usage.
AI Inference	Claude API	$80-150/mo	Haiku for categorization, Sonnet for coaching. ~5-10M input tokens/mo during beta testing.
Push Notifications	Firebase	$5-10/mo	Free tier covers initial usage.
Design & Dev Tools	MagicPatterns, Claude Code, Figma	$50-100/mo	Figma Pro, MagicPatterns, Claude API (already counted above).
Development Phase Monthly Total: $155-325/mo			
POST-LAUNCH (16+ weeks)			
Backend	Supabase	$25-50/mo	Pro/Team plan as user count grows. Add

Category	Service	Cost	Notes
			database replicas at scale.
AI Inference	Claude API	$25-75/mo	Scales with DAU. With 1K active users, expect 20-50M tokens/mo.
Push Notifications	Firebase	$0-10/mo	Free tier up to 100K messages/day, then pay-as-you-go.
Bank Connections	Plaid (Phase 2)	$0-25/mo	First 200 connections free. Then $0.25/connection. At 1K users, ~$200/mo.
Analytics	PostHog or Mixpanel	$0-10/mo	PostHog open-source (free) or Mixpanel free tier.
Post-Launch Monthly Total: $55-130/mo (excluding Plaid)			
ONE-TIME COSTS			
App Store Submissions	Apple + Google	$125	Apple Developer: $99/year. Google Play: $25 one-time.
Domain & SSL	Namecheap + Cloudflare	$15-30/year	Domain ~$12/year. SSL free via Cloudflare.
		$15-50	

Category	Service	Cost	Notes
Marketing Assets	Midjourney + Loom		Demo video, hero image. Already budgeted in dev tools.
One-Time Total: $155-205			

Cost Optimizations

- **CSV-first approach (Phase 1):** Skip Plaid for launch. Saves $200/mo initially.
- **Rule-based categorization:** No expensive ML training. Claude API fallback is cost-effective.
- **Supabase managed:** Zero DevOps overhead. Auto-scaling included.
- **AI-generated design:** No design agency budget. MagicPatterns + Claude Code is <$100/mo vs $10K+ agency.

11

Timeline — 16 Weeks to Launch

Visual roadmap of development phases from concept to public launch.

Week 0: Brand, Design, Research

1 week

Finalize visual brand identity (colors, typography, component library). User research interviews with 10-15 target users. Competitive analysis of YNAB, Monarch, Copilot. Define core metrics (DAU, retention, NPS).

⚙ Weeks 1-3: Foundation (Pillar B & C Baseline)

3 weeks

CSV parser & categorization engine. Budget calculation logic. Safe-to-spend algorithm. Database schema. Supabase setup with RLS. Basic API endpoints. Foundation for everything else.

Weeks 4-6: Core Experience (Pillar A, C, D)

3 weeks

Onboarding flows (framework selection, archetype quiz, sliders). Cash flow visualization. Safe-to-spend display. Overspend alerts. Transaction view. Dashboard. Bills & debt tracking. "Ask Sage" conversational AI bar.

Weeks 7-9: Intelligence (Pillar E, F)

3 weeks

Behavioral insights engine (10-15 rule-based insights). Priority action engine. Savings recommendations. Debt trends. Subscription detection. Weekly summary email system. Push notification system.

Weeks 10-12: Polish & Beta Prep

3 weeks

Performance optimization. Animations & transitions. QA & bug fixes. App Store & Google Play setup. Privacy policy & legal review. Plaid integration (optional, Phase 2). Accessibility review.

Weeks 13-14: Closed Beta

2 weeks

Invite 20-30 target users (mixed-income professionals). Gather feedback via interviews & surveys. Fix critical bugs. Iterate on core flows based on real usage. Measure retention & NPS.

Weeks 15-16: Public Launch

2 weeks

App Store & Google Play submissions (48-72 hour review). Product Hunt launch prep & posting. Reddit outreach. Influencer outreach (personal finance creators). Monitor analytics & fix issues. Celebrate! 🎉

Conversational Budgeting App — PRD v1.0
Last Updated: March 30, 2026
Status: Active Development

Built for cofounders, investors, and advisors.
Questions? Refer to the sidebar TOC for quick navigation.