Stock Trading Agent

Design Document v4 · Yang Liu · Feb 2026

1. Overview

An AI-powered trading agent for US tech stocks, focused exclusively on earnings report events. When a major tech stock beats earnings expectations and surges, the agent monitors the surge and waits — when the surge slows down (dual validation), it shorts the stock expecting a pullback. After the pullback, it covers the short in 3 batches to close the position and take profit.

Strategy: Short selling after post-earnings surge slows. Borrow shares at high price, return them at lower price after pullback — profit is the difference.

Broker: Interactive Brokers (IBKR) via Client Portal API. Requires Margin Account with short selling permission. All credentials stored in environment variables, never hardcoded.

Monitored Stocks

TSLA AAPL NVDA META GOOGL MSFT AMZN AMD QCOM WDC CRM PANW

2. Full Agent Flow

All 10 steps orchestrated by LangGraph with conditional routing. Security layer wraps every operation. Flow loops back to Step 1 after each completed trade.

[STEP 1: Earnings Calendar Collection]

TOOL: yfinance — fetch earnings dates for all 12 stocks every day

FUNCTION: check if any stock has earnings within 7 days

WRITE: log earnings schedule to daily log file

No upcoming earnings → sleep 24h, repeat

Earnings detected within 1 week → trigger Step 2

↓

[STEP 2: Earnings Result Detection]

[LLM + Skills]

TOOL: requests (web search) — search '[stock] earnings results Q[N] [year]'

AI (Claude) Prompt: 'Given this news: [text]. Did [stock] beat EPS expectations? By what %? Reply in JSON: {beat: true/false, beat_pct: float}'

FUNCTION: parse JSON response → check beat_pct value

EPS beat < 5% → skip, back to Step 1

EPS beat > 10% → strong signal, continue to Step 3

Miss expectations → stop, wait for next stock

↓

[STEP 3: Surge Detection]

TOOL: yfinance — fetch real-time price every 5 minutes

TOOL: yfinance — fetch SPY and QQQ daily change

FUNCTION: if SPY or QQQ down > 2% → pause, do not trade today

FUNCTION: if stock price up > 8% from pre-earnings close → surging

Not surging → wait 5 min, repeat

Surging detected → continue to Step 4

↓

[STEP 4: Slowdown Detection — Dual Validation — Short Trigger]

TOOL: yfinance — fetch price + volume every 5 minutes

FUNCTION (Hard Rules) — check at least 2 of 3 must be true:

- 5-min price increase < 0.3%
- Volume down 40% vs prior 30-min average
- Price pulled back > 1.5% from today's high

AI (Claude) Prompt: 'Current price data: [prices]. Volume: [volumes]. Is the post-earnings surge slowing? Give confidence score 0-100 and reasoning.'

AI confidence must be > 70% to proceed

FUNCTION: price guard — current price must be $40+ above pre-earnings price

FUNCTION: calculate dynamic stop loss from 30-day historical volatility:

- TSLA-type (volatility > 3%): stop loss = 8% above short price
- Medium (2-3%): stop loss = 6% · Low e.g. AAPL (< 2%): stop loss = 5%

All conditions met → trigger short sell signal, continue to Step 5

↓

[STEP 5: ReAct Self-Verification (fixed 2 iterations)]

[LLM Only]

AI (Claude) Iteration 1 — Prompt Enhancement:

Prompt: 'You are about to short [stock] at $[price]. Pre-earnings: $[base]. Surge: [X]%. Volume drop: [Y]%. AI score: [Z]. Verify: is this a valid short signal? What are the risks? Output final recommendation.'

AI (Claude) Iteration 2 — Verification:

Prompt: 'Review this analysis: [iter1 output]. Is the recommendation correct? Are there missed risks? Confirm YES or explain correction.'

Fixed at 2 iterations — no infinite loop risk

Output confirmed → continue to Step 6

↓

[STEP 6: SMS Notification + MFA]

FUNCTION: calculate suggested batch sizes based on IBKR account balance

TOOL: Twilio — send SMS with full summary (stock, price, surge %, AI score, stop loss, batch size)

Example: 'NVDA $875 (pre-earnings $820, +$55). Short 30 shares. Stop loss $945. Confirm? YES/NO/CHANGE N'

TOOL: asyncio — wait for user SMS reply (async, non-blocking)

NO → cancel entire operation, log cancellation

YES or CHANGE [N] → update batch size if changed, proceed to Step 7

↓

[STEP 7: Short in 3 Batches]

For each batch (30% / 30% / 40%):

TOOL: Twilio — send SMS: 'Batch [N]: short [X] shares @ market price. Confirm YES?'

TOOL: asyncio — wait for YES reply

TOOL: ib_insync (IBKR API) — execute short sell order (borrow + sell shares)

FUNCTION: monitor stop loss after each batch — if triggered, send alert immediately

All 3 batches done → proceed to Step 8

↓

[STEP 8: Wait for Pullback]

[LLM + Skills]

TOOL: yfinance — monitor price every 5 minutes

FUNCTION: check stop loss — if price rises above stop loss price:

AI (Claude) Prompt: 'Stop loss triggered for [stock]. Current price $[X], short price $[Y]. Analyze: is this a temporary spike or trend reversal? Recommend: cover now or hold?'

TOOL: Twilio — send SMS with AI analysis + recommendation, user decides

FUNCTION: check pullback — threshold dynamically calculated based on stock's historical volatility (same engine as stop loss calculation)

Pullback confirmed → trigger Step 9

FUNCTION: check timeout — if day 7+ with no pullback:

TOOL: Twilio — send SMS: 'Day 7: no pullback detected. Recommend covering to limit risk.'

↓

[STEP 9: Cover Short in 3 Batches — Close Position, Take Profit]

FUNCTION: calculate profit estimate per batch

TOOL: Twilio — send SMS: 'Pullback detected. [Stock] now $[price]. Est. profit: $[X]. Cover?'

For each batch (30% / 30% / 40%):

TOOL: Twilio — send SMS: 'Cover batch [N]: buy [X] shares to close short. Confirm YES?'

TOOL: asyncio — wait for YES reply

TOOL: ib_insync (IBKR API) — execute buy-to-cover (buy back borrowed shares, return to broker)

All 3 batches covered → all positions closed, trade complete → proceed to Step 10

↓

[STEP 10: Memory + Reflection]

WRITE short-term memory: full trade log (every price check, SMS sent, order executed)

WRITE long-term MEMORY.md: trade record, stock profile (win rate, avg surge, avg pullback days), user preferences

AI (Claude) Prompt: 'Trade summary: [data]. Reflect: Was short timing correct? How many days did pullback take vs historical average for this stock? What to improve?'

TOOL: chromadb — store reflection as vector embedding for future retrieval

Memory retrieval: vector search + keyword matching, recent trades weighted higher

Memory expiry: 6 months → lower weight · 1 year → archived

Clear short-term memory → loop back to Step 1

↓

3. LangGraph Task Orchestration

LangGraph connects all 10 steps using conditional routing. Each step either proceeds forward, loops back to wait, or exits based on the result. The entire flow restarts from Step 1 after each completed trade cycle.

Routing logic:

- Step 1 → 2: only when earnings detected within 1 week
- Step 2 → 3: only when EPS beat > 10%; otherwise skip to next stock
- Step 3 → 4: only when stock is surging AND market is healthy (SPY/QQQ not down > 2%)
- Step 4 → 5: only when dual validation passes (hard rules + AI confidence > 70%)
- Step 5 → 6: after ReAct self-verification completes
- Step 6 → 7: only when user replies YES via SMS; NO cancels the entire trade
- Step 7: loops until all 3 short batches are confirmed and executed
- Step 8 → 9: when pullback detected, stop loss triggered, or day 7+ timeout
- Step 9: loops until all 3 cover batches are confirmed and executed
- Step 10 → Step 1: after memory and reflection written, loop back

4. Three Core Principles

1. Prompt Enhancement

Before every node executes, the agent enriches the intent — adding context, breaking down subtasks, considering edge cases — then executes based on true intent, not literal instruction.

2. Cross-session Memory (OpenClaw-style)

Two-tier memory: short-term for current session (full trade timeline), long-term MEMORY.md for trade history, stock profiles, and user preferences. Retrieval uses vector search + keyword matching, recent memories weighted higher. Records older than 6 months lose weight, older than 1 year are archived.

3. Multi-tool Coordination

Each tool does what it does best: Yahoo Finance for real-time price data, web search for earnings news, Claude AI for analysis and judgment, Twilio for SMS confirmation, IBKR for order execution. LangGraph coordinates all tools without manual handoffs.

5. Security Layer

Security is not a separate module — it wraps every single operation in the flow.

Physical Isolation: Docker Sandbox

- All code execution runs inside a Docker container
- No external network except Yahoo Finance, IBKR, Twilio, and Anthropic APIs
- Read-only filesystem except /workspace · Non-root user, all Linux capabilities dropped
- Container destroyed after each session

Guardrails: Per-operation AI Safety Check

- Before every operation, Claude checks: is this safe to execute?
- Dangerous patterns blocked immediately (e.g. short entire account balance)
- Credentials never passed to AI context — environment variables only

MFA: Human in the Loop

- No trade executes without user SMS confirmation
- Every single batch requires a fresh YES reply
- User can reply NO or CHANGE at any point to cancel or adjust

6. Risk Control

Dynamic Stop Loss

- Calculated per stock based on 30-day historical volatility
- High volatility e.g. TSLA (> 3%): stop loss = 8% above short price
- Medium volatility (2-3%): stop loss = 6%
- Low volatility e.g. AAPL (< 2%): stop loss = 5%

Daily Loss Limit

- If total daily loss exceeds user-defined limit → stop all operations for the day
- SMS alert: 'Daily loss limit reached. Agent paused until tomorrow.'

Market Environment Check

- Check SPY/QQQ daily change before every trade
- Market down > 2% → pause all operations

Timeout Protection

- Day 7+: no pullback → SMS recommending cover to limit risk
- User decides whether to cover or continue holding

7. Initial Setup

Before running the agent for the first time, configure credentials and preferences. All sensitive information is stored in a .env file — never in the code.

Required Configuration (.env file)

- IBKR_USERNAME=your_username # IBKR account username
- IBKR_PASSWORD=your_password # IBKR account password
- IBKR_ACCOUNT_ID=your_account_id # IBKR account ID
- TWILIO_ACCOUNT_SID=xxx # Twilio SMS account SID
- TWILIO_AUTH_TOKEN=xxx # Twilio auth token
- USER_PHONE=+1xxxxxxxxxx # Phone for SMS confirmation
- ANTHROPIC_API_KEY=xxx # Claude API key

User Preferences (config.yaml)

- max_short_position_per_stock: 10000 # Max $ value to short per stock
- price_guard_min_gain: 40 # Min $ gain above pre-earnings price before shorting
- max_days_to_wait_cover: 7 # Max days to wait for pullback
- daily_loss_limit: 2000 # Max daily loss before agent pauses