

Stock Insights AI: System Architecture & Workflow Report

1. Executive Summary

Stock Insights AI is a specialized multi-agent system designed to democratize institutional-grade financial research. Unlike standard chatbots that hallucinate financial data, this system anchors its analysis in real-time data retrieved from **Yahoo Finance** and **Alpha Vantage**. It employs a "Crew" of specialized AI agents that debate, analyze, and synthesize data before presenting a final investment recommendation.

2. Agent Architecture

The system operates in two distinct modes, each utilizing a specific configuration of agents.

Mode A: Single Ticker Deep Dive (The "Investment Committee")

Objective: Analyze a specific stock (e.g., AAPL) over a specific date range to generate a Buy/Hold/Sell rating.

|

| **Agent Role | Responsibility | Tools Used | | Senior Sentiment Analyst** | Analyzes news headlines and market buzz to determine the psychological state of the market regarding the asset. | `fetch_news_sentiment` (Alpha Vantage) | | **Fundamental Analyst** | Evaluates the company's financial health, valuation (P/E), and stability. Acts as the "Value Investor". | `fetch_fundamental_data` (Yahoo Finance) | | **Technical Analyst** | Analyzes price action, trends, RSI, and Moving Averages. Acts as the "Chartist". | `calculate_technicals` (Yahoo Finance) | | **Portfolio Manager** | The decision maker. Reads reports from the three analysts above, resolves conflicts (e.g., "Good fundamentals but bad chart"), and writes the final report. | *Delegation & Synthesis* |

Workflow Flow:

1. User inputs Ticker + Date Range.
2. **Parallel Execution:** Sentiment, Fundamental, and Technical agents gather data simultaneously (simulated via sequential steps for rate limit safety).
3. **Synthesis:** The Portfolio Manager aggregates findings.
4. **Output:** A Markdown report with a Confidence Score and Verdict.

Mode B: Market Trend Scanner (The "News Desk")

Objective: Identify the top moving stocks of the day and explain the narrative behind the move.

| **Agent Role** | **Responsibility** | **Tools Used** | | **Market Historian** | Acts as a researcher. Takes the list of winning stocks and searches for the specific catalyst (Earnings, FDA approval, Macro news). | `search_internet` (DuckDuckGo / Custom) | | **Strategic Trend Analyst** | Looks for patterns across the winners (e.g., "Semiconductors are rallying"). Assigns a short-term trading signal. | *Analysis Logic* |

Workflow Flow:

1. **Python Layer:** The app calls Alpha Vantage API to get the "Top Gainers" list (math-based, zero hallucination).
2. **Agent Layer:** The list is passed to the Crew.
3. **Output:** A strategic summary explaining *why* these specific stocks are up.

3. Technology Stack & Tools

A. Large Language Model (LLM)

- **Model:** `gemini-2.5-flash`
- **Provider:** Google (via `litellm` integration).
- **Configuration:** * Temperature: `0.2` (Low creativity, high factual accuracy).

- **Optimization:** A single LLM instance is shared across agents to minimize memory overhead.

B. Data Tools (`tools.py`)

1. `fetch_market_movers` : Connects to Alpha Vantage `TOP_GAINERS_LOSERS` endpoint.
2. `fetch_news_sentiment` : Connects to Alpha Vantage `NEWS_SENTIMENT` endpoint. Returns a float score (-1.0 to +1.0) and headlines.
3. `fetch_fundamental_data` : Uses `yfinance.Ticker.info` to get Market Cap, P/E, Beta.
4. `calculate_technicals` : Uses `pandas` and `ta` (Technical Analysis library) to calculate RSI (Relative Strength Index) and MACD from raw price history.
5. `calculate_risk_metrics` : Calculates Annualized Volatility and Maximum Drawdown based on historical returns.

C. User Interface

- **Framework:** Streamlit.
- **Design System:** Custom CSS injection for a "Dark Neon" aesthetic using Glassmorphism effects.
- **Interactive Charts:** `plotly.graph_objects` for rendering professional Candlestick charts.
- **State Management:** `st.session_state` ensures analysis persists even when the user interacts with other UI elements (like downloading reports).

4. Operational Constraints & Handling

- **Rate Limiting:** The system uses `max_rpm=5` (Requests Per Minute) in the CrewAI configuration. This is crucial to strictly adhere to the Google Gemini Free Tier limits and prevent `429 RESOURCE_EXHAUSTED` errors.
- **Error Handling:** The UI includes `try-except` blocks for all API calls, providing user-friendly fallback messages if an external API (Yahoo/Alpha Vantage) is down.

5. Future Roadmap

- **Portfolio Integration:** Allow users to input their current holdings for personalized advice.
- **Alerting System:** Email notifications when a tracked stock hits a specific "Buy" signal.
- **Database:** Store analysis history in a local SQLite or Postgres database for trend tracking over time.