System Design

Course: INF1002 Programming Fundamentals
Team ID: P1-8

Table of Contents

1. System/Module Breakdown 2

1.1 Big Picture 2

1.2 Core Modules 2

1.2.1 Data Acquisition and Preprocessing Module 2

1.2.2 Technical Analysis Engine 2

1.2.4 Web Interface and Visualization (Streamlit) 3

1.2.5. Application State Management (Streamlit) 3

1.2.6 Logging and Monitoring 3

1.2.7 External Integrations 3

2. Core Feature List 3

2.1 Primary Features (MVP) 3

2.2.1 Stock Data Import and Validation 3

2.2.2 Interactive Stock Analysis 4

2.2.3 Market Dashboard and Multi-indicator Comparison Tools 4

2.2 Advanced Features (Future Enhancement) 4

3. Dataset Description 4

3.1 Data Sources and Structure 4

3.2 Data Schema and Format 5

3.2.1 Raw Fields from Yahoo Finance 5

3.2.2 Optional Metadata 5

3.2.3 Calculated Fields (Post-Ingestion) 6

3.3 Data Quality Management 6

4. Updated Task Allocation 7

5. Flowchart 7

1. System/Module Breakdown

1.1 Big Picture

Stalking Stocks consists of three layers:

- Data Acquisition and Preprocessing (Clean historical stock data)
- Analysis Engine (Technical indicators, metrics)
- Web interface (interactive dashboard via Streamlit)

1.2 Core Modules

1.2.1 Data Acquisition and Preprocessing Module

Purpose: Retrieves historical stock data and prepares it for analysis

- Data Sources: yfinance API for up-to-date historical stock data
  A fixed list of 15 tickers across 5 sectors ensuring consistency
- Preprocessing Tasks:
  - Remove non-trading dates, weekends, and holidays
  - Handle missing values using forward-fill or interpolation
  - Standardize date formats and ensure chronological ordering
  - Detect and flag price outliers (>3 standard deviations)
  - Normalize data structures for consistent analysis
  - Data transformation → Data transformation reshapes stock data from long format (Date + Ticker + Price per row) into a matrix: rows = dates, columns = tickers, values = adjusted close prices.
  - This enables calculation of additional metrics for stock analysis

Output: Clean, analysis-ready datasets stored in pandas DataFrames

1.2.2 Technical Analysis Engine

Purpose: Implements core financial algorithms and performance metrics

- Technical Indicators:
  - Simple Moving Averages (SMA) using sliding-window algorithms
  - Daily returns and volatility calculations
  - Trend run detection (consecutive up/down movements)
- Performance Metrics:
  - Maximum profit calculations with multiple transaction scenarios
  - Risk-return ratios and volatility rankings
  - Sector performance comparisons
- Algorithm Optimization: Dictionary-based caching for repeated calculations

1.2.4 Web Interface and Visualization (Streamlit)

Purpose: Provides interactive user experience for stock analysis

- Dashboard Components:
    - Interactive price charts with technical indicators (Plotly)
    - Indicator comparison tools (compare simultaneously)
    - Sector performance overview
    - Market moments highlighting significant events
- User Controls:
    - Stock/sector selection dropdowns
    - Date range pickers
    - Technical indicator toggles
    - Comparison mode switches
- Responsive design for intuitive navigation and mobile use.

1.2.5. Application State Management (Streamlit)

Purpose: Manages user sessions and application configuration

- Session Management: Track user selections and preferences
- Configuration Settings: Environment variables for API keys and data sources
- Caching Strategy: Store frequently accessed data to reduce API calls

1.2.6 Logging and Monitoring

Purpose: Tracks application performance and user interactions

- Performance Metrics: Page load times, calculation speeds, API response times
- Error Tracking: Log technical errors and user-facing issues.
- Debug Information: Detailed logs for development and troubleshooting

1.2.7 External Integrations

- yfinance API: Historical and up to date stock data
- Streamlit Framework: Web application hosting and UI components
- Plotly Library: Interactive chart generation

2. Core Feature List

2.1 Primary Features (MVP)

2.2.1 Stock Data Import and Validation

- Automated data retrieval via yfinance API with error handling
- Covers 15 tickers across 5 sectors

2.2.2 Interactive Stock Analysis

- Price charts with customizable technical indicators
- Simple Moving Average overlays with adjustable periods
- Daily returns, volatility visualization, and trend detection

2.2.3 Market Dashboard and Multi-indicator Comparison Tools

- Side-by-side comparison of indicators
- Sector performance analysis across time periods
- Ranking top/worst performers identification

2.2 Advanced Features (Future Enhancement)

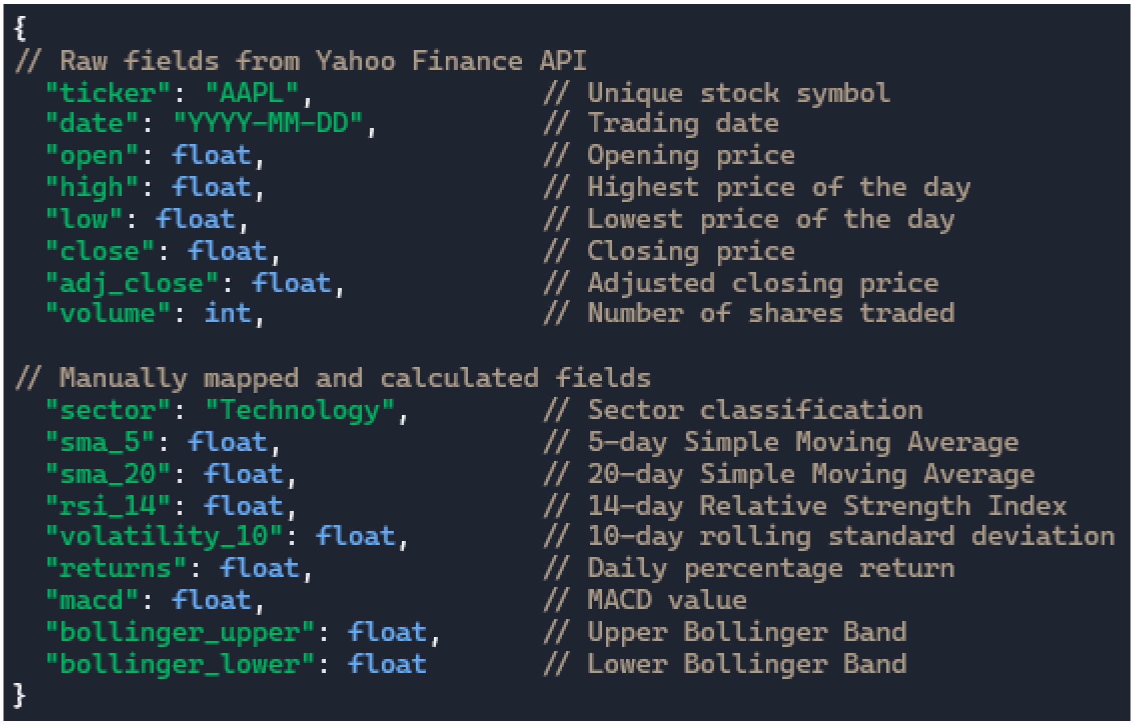
- Advanced technical indicators (RSI, MACD, Bollinger Bands)
- Export capabilities for charts and data

3. Dataset Description

3.1 Data Sources and Structure

- Primary Source: Yahoo Finance API, 3 years of daily data, refreshed daily (post-market close)
- Supported Securities: 15 tickers across 5 major sectors:
    - Technology
    - Finance
    - Healthcare
    - Consumer Goods
    - Energy

3.2 Data Schema and Format



```
{
// Raw fields from Yahoo Finance API
  "ticker": "AAPL",                // Unique stock symbol
  "date": "YYYY-MM-DD",            // Trading date
  "open": float,                   // Opening price
  "high": float,                   // Highest price of the day
  "low": float,                    // Lowest price of the day
  "close": float,                  // Closing price
  "adj_close": float,              // Adjusted closing price
  "volume": int,                   // Number of shares traded

// Manually mapped and calculated fields
  "sector": "Technology",          // Sector classification
  "sma_5": float,                  // 5-day Simple Moving Average
  "sma_20": float,                 // 20-day Simple Moving Average
  "rsi_14": float,                 // 14-day Relative Strength Index
  "volatility_10": float,          // 10-day rolling standard deviation
  "returns": float,                // Daily percentage return
  "macd": float,                   // MACD value
  "bollinger_upper": float,        // Upper Bollinger Band
  "bollinger_lower": float         // Lower Bollinger Band
}
```

3.2.1 Raw Fields from Yahoo Finance

- ticker: Stock symbol identifying the security (e.g., AAPL for Apple Inc.).
- date: Trading date in ISO format (YYYY-MM-DD), used as the primary time index.
- open: Price at market open.
- high: Highest price reached during the trading day.
- low: Lowest price reached during the trading day.
- close: Price at market close.
- volume: Total number of shares traded on that day.

The dataset includes **raw fields** from Yahoo Finance (prices, volume, dates), **metadata** (sector classification), and **calculated indicators** (SMA, RSI, MACD, Bollinger Bands, volatility, returns). This structure ensures clean, consistent, and analysis-ready data for our stock analysis system.

3.2.2 Optional Metadata

- sector: Industry classification for the ticker (e.g., Technology, Finance).
  Manually assigned via a predefined ticker-to-sector dictionary.
  Supports grouping and sector comparisons but not dynamically sourced from the API.

3.2.3 Calculated Fields (Post-Ingestion)

- sma_5 / sma_20: Simple Moving Averages over 5 and 20 days, used to smooth price trends.
- rsi_14: Relative Strength Index over a 14-day window, indicating momentum and potential overbought/oversold conditions.
- volatility_10: Rolling standard deviation of returns over 10 days, used to quantify price variability.
- returns: Daily percentage change in adjusted close price, used for performance and volatility analysis.
- macd: Moving Average Convergence Divergence, a momentum indicator based on the difference between short- and long-term EMAs.
- bollinger_upper / bollinger_lower: Upper and lower bounds of Bollinger Bands, which define a volatility envelope around the moving average

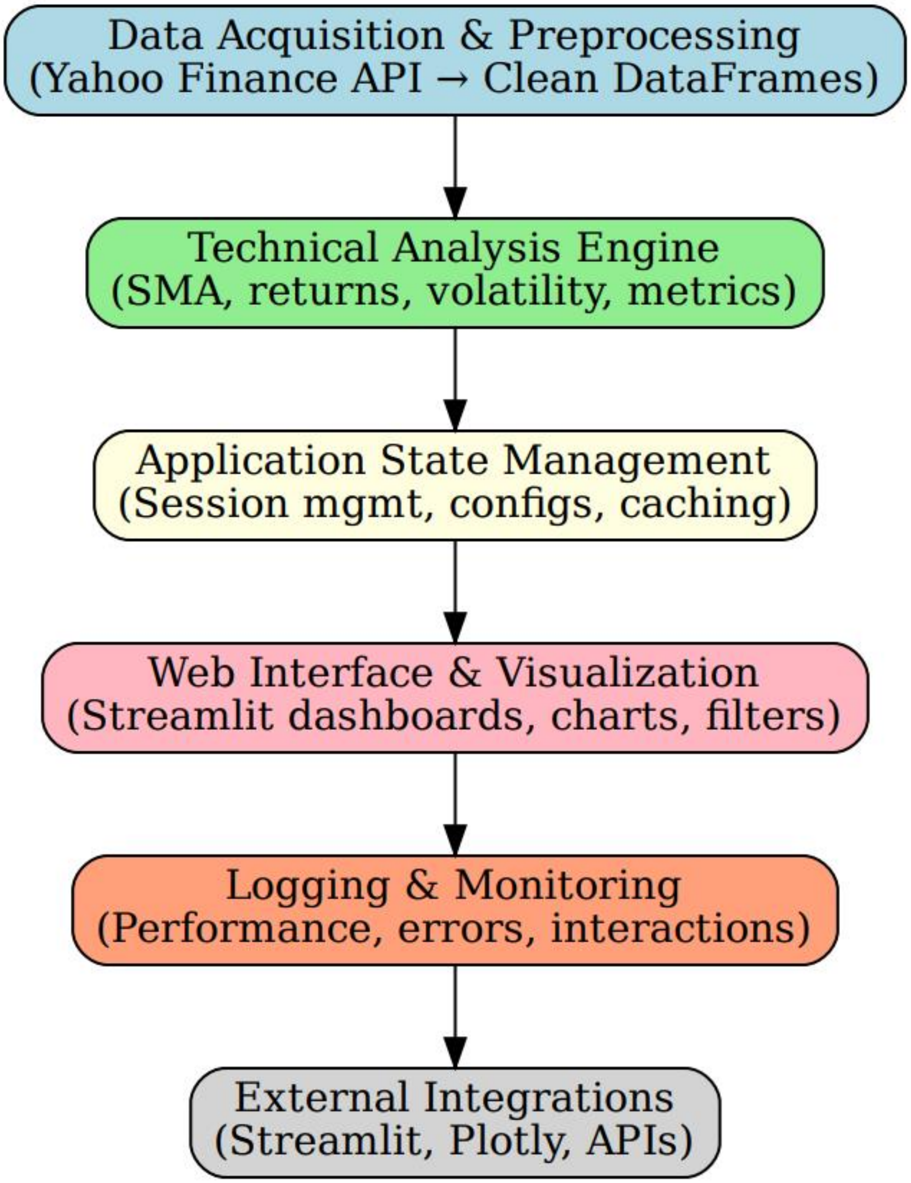
3.3 Data Quality Management

- Validation Rules:
    - Ensure price fields (open, high, low, close, adj_close) follow logical constraints (e.g., low ≤ close ≤ high)
    - Check for reasonable volume ranges to filter out erroneous spikes
    - Verify date continuity across tickers to prevent misalignment in time-series analysis
- Missing Data Handling:
    - Apply forward-fill for gaps shorter than 3 consecutive trading days
    - Exclude records with longer gaps to preserve analytical integrity
- Outlier Detection:
    - Use statistical methods (e.g., z-score thresholds, IQR filtering) to flag anomalous price movements
    - Flag anomalies for review or exclusion.

4. Updated Task Allocation

No changes to initial task allocation.

5. Flowchart


Data Acquisition & Preprocessing
(Yahoo Finance API → Clean DataFrames)
Technical Analysis Engine
(SMA, returns, volatility, metrics)
Application State Management
(Session mgmt, configs, caching)
Web Interface & Visualization
(Streamlit dashboards, charts, filters)
Logging & Monitoring
(Performance, errors, interactions)
External Integrations
(Streamlit, Plotly, APIs)