# Stock Watch Project

## 1. Introduction

### 1.1 Purpose

The purpose of this project is to create a tool (backend focused, simplify frontend can be added later – with gradio UI or anything similar) to allow user to keep track on stock prices fluctuation real-time, probing to right timing to buy stock or sell them.

### 1.2 Scope

Describe the software being specified, its benefits, objectives, and goals.

The software should allow user to add the stock that they want to add in the tracking list, then when each stock price is changing (up or down) by X%, alert user real time (via what – TBD).

Maybe – Integrating AI to read news article (source undefined just yet – TBD) then make reasonable reason behind each spike or dip of prices – probing reasonable buying or selling action. 🡪 AI-powered news summarization and sentiment tagging

* MVP: Pull in news headlines related to stock tickers.
* Next: Summarize headlines using OpenAI / Hugging Face / LLMs.
* Future: Sentiment analysis (positive/negative/neutral).

Goal is to allow user to work real time into making profit from stock buying.

### 1.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Term | Definition |
| API | Using predefined backend API for services |
| Gradio UI | Simplified way to create UI |
| etc. | … |

### 1.4 References

List any external documents or APIs used, such as financial data providers or UI libraries.

### 1.5 Overview

Brief description of what the rest of the document covers.

## 2. Overall Description

### 2.1 Product Perspective

Explain how this software fits in with other systems, if applicable.

### 2.2 Product Functions

List the major functions of the system, such as:  
- Real-time stock price monitoring  
- User-defined watchlists  
- Price alerts and notifications  
- Historical charting – No chart 🡪 Just average price over weeks and months, then min and max for the day!

- Integrate AI services to read news for the day (on the stock in watchlist only) 🡪 leading to reasonable buying or selling behavior

### 2.3 User Classes and Characteristics

Define the different types of users (e.g., casual investor, trader, admin) and their needs.

- Only the developer for now!

- If successful, push to casual investor too.

### 2.4 Operating Environment

What tech stack are you planning?

Frontend: gradio UI

Backend: unknown just yet 🡪 will define along the way

* Maybe fast API and flask (self-defined API)

Coding: Mainly python

### 2.5 Design and Implementation Constraints

Web browser only for now for testing purposes. -> will switch to react for responsive design if successful.

### 2.6 User Documentation

Only readme file is available.

### 2.7 Assumptions and Dependencies

Are there any assumptions (e.g., reliable API data) or third-party dependencies?

Reliable API data srouce from “Market Watch” app? For free? 🡪 leading reasonable analysis on stocks? Available for each stocks?

Market watch might need to be paid!!!

Free options are:

* [Alpha Vantage](https://www.alphavantage.co/)
* [IEX Cloud](https://iexcloud.io/)
* [Finnhub](https://finnhub.io/)
* [Yahoo Finance unofficial APIs (via yfinance)](https://pypi.org/project/yfinance/)

## 3. Specific Requirements

### 3.1 Functional Requirements

Detail the features and behaviors, broken down by priority or module. Example:  
- FR1: The system shall allow users to search and add a stock ticker to their watchlist.  
- FR2: The system shall fetch real-time stock data using [API Name].  
- FR3: The system shall notify users when a stock crosses a threshold price by X%.

- FR4: The system shall allow user to set the percentage of threshold to trigger notification.

- FR5: The system shall allow user to check the stats of each stock (min/max for the day, prices average for week or months range) when going into detail.

- FR6: TBD

### 3.2 Non-Functional Requirements

Examples:  
- NFR1: The system should load stock data updates within 5 second.  
- NFR2: The system should be available 99.9% of the time.  
- NFR3: The frontend must be responsive for desktop browser windows for now.

- NFR4: TBD

### 3.3 System Interfaces

List any external APIs, databases, or services the app will integrate with.

- Will add along the way.

### 3.4 User Interfaces

Unknown, planning to use Gradio for simplified UI just for demonstration and testing!

### 3.5 Error Handling

How will the system respond to bad API data, user typos, connection issues?

- Unknown

## 4. Future Enhancements

List any planned features not in scope for the MVP version. e.g.:  
- Portfolio tracking  
- Social sentiment integration  
- AI-powered stock suggestions

## 5. Appendices

Any diagrams, links to relevant tools or libraries, user flow sketches, etc.

Others note:

### 📈 Is This Project Worth Working On?

#### ✔ Value for Portfolio

* **High** if you complete the MVP (watchlist, threshold alerts, real-time price view)
* **Very High** if you add the news summarization + React frontend
* Recruiters love **end-to-end projects** that mix backend APIs + frontend + real-time processing

#### ✔ New Tech Stack Exposure

* Gradio (for quick prototyping)
* FastAPI/Flask (API design)
* yfinance or similar (external data fetching)
* LangChain or OpenAI (for AI add-on)
* React (if expanded)
* WebSocket (if you go for true real-time)

This builds well on your current Python/React stack.

### 📊 Difficulty Level

* **Basic–Intermediate** for backend + Gradio MVP
* **Intermediate–Advanced** for AI integration or a React port
* **Advanced** if you attempt real-time WebSocket support or full mobile app

### 🤔 Market Competitors

There are tons of stock tracking tools already:

* Yahoo Finance
* TradingView
* Robinhood
* Webull
* SeekingAlpha

✅ Your competitive **advantage**:

* Simplicity + alerting
* Developer-oriented UI (Gradio for fast demo)
* Integrating **LLM-powered news analysis** is unique

### 💡 Bonus Unique Ideas (Optional)

* Tag each stock as “volatile,” “stable,” “trending” based on news + price deltas
* Weekly insights via email (use background jobs like cron + SMTP)
* Backend testing + CI/CD with GitHub Actions

Comparing to existing tool?

| **Feature** | **Robinhood / Webull / Yahoo** | **Your Tool** |
| --- | --- | --- |
| **Real-time Stock Data** | ✅ Yes | ✅ Planned |
| **Custom Watchlists** | ✅ Yes | ✅ Planned |
| **Manual Buy/Sell Interface** | ✅ Yes | ❌ Not core focus |
| **AI-Powered Spike Reasoning** | ❌ No | ✅ Unique Feature |
| **Automated Alert System for % Change** | ❌ Limited / Clunky | ✅ Central Feature |
| **Self-Pilot Trading Logic** | ❌ No (humans only) | 🔄 Planned via AI |
| **Lightweight Developer Tool** | ❌ No | ✅ Yes, with Gradio |
| **Built-in Sentiment Summary from News** | ❌ No | ✅ Planned |
| **Emphasis on Alert-Driven Profits** | ❌ No | ✅ Yes |