

**Trading Signal System: Finale Projektspezifikation**

Alle Ihre Anforderungen sind geklärt und das Projekt kann nach Ihren Präferenzen umgesetzt werden [[1]](#fn1)[[2]](#fn2)[[3]](#fn3). Basierend auf unserer umfassenden Diskussion habe ich ein detailliertes Spezifikationsdokument erstellt, das als Grundlage für die Implementierung dienen wird.

**Bestätigte Projektparameter**

**Technologie-Stack und Präferenzen**

PyTorch 2.7.1 wurde als primäres Machine Learning Framework bestätigt und bietet mit CUDA-Unterstützung optimale Performance für Ihre RTX 3090 Ti [[2]](#fn2)[[4]](#fn4). Visual Studio Code wird als Entwicklungsumgebung eingesetzt, ergänzt durch aktuelle Python-Module mit verifizierten Versionen [[1]](#fn1)[[5]](#fn5)[[6]](#fn6).

**Systemarchitektur**

Das verteilte System nutzt Ihren Raspberry Pi 5 für 24/7-Betrieb und den Windows PC für rechenintensive ML-Operationen während der Tageszeit [[3]](#fn3)[[7]](#fn7). Die unterbrechbare ML-Verarbeitung ermöglicht flexible Nutzung Ihres PCs für andere Aufgaben ohne Datenverlust [[8]](#fn8)[[9]](#fn9).

**Backup und Sicherheit**

Wöchentliche Backups zu Ihrem 2TB pCloud-Speicher sind ausreichend und entsprechen dem Prinzip "so einfach wie möglich" [[10]](#fn10)[[11]](#fn11). Erweiterte Sicherheitsfeatures werden in der ersten Ausbaustufe bewusst weggelassen, um die Komplexität zu reduzieren [[3]](#fn3)[[7]](#fn7).

**Benachrichtigungssystem**

Telegram wird als primäres Benachrichtigungssystem implementiert mit formatierter Signalübermittlung und Charts [[12]](#fn12)[[6]](#fn6). Das lokale Dashboard über Streamlit 1.45.1 bietet umfassende Performance-Visualisierung ohne Remote-Zugriff [[13]](#fn13)[[14]](#fn14)[[15]](#fn15).

**Kernkomponenten des Systems**

**Datenerfassung und -verarbeitung**

Das System nutzt yfinance 0.2.63 mit korrekter Handhabung der neuen Multi-Index-Struktur für kontinuierliche Marktdatenerfassung [[1]](#fn1)[[16]](#fn16). Die Yahoo Finance API bietet ausreichend Abdeckung für Trade Republic-kompatible Derivate mit über 380.000 handelbaren Instrumenten [[7]](#fn7)[[17]](#fn17).

**Machine Learning Pipeline**

PyTorch-basierte Sentiment-Analyse mit FinBERT für Finanznachrichten kombiniert mit technischen Indikatoren generiert hochqualitative Trading-Signale [[2]](#fn2)[[4]](#fn4). Das unterbrechbare Verarbeitungssystem ermöglicht flexible Nutzung Ihrer Hardware-Ressourcen [[8]](#fn8)[[9]](#fn9).

**Performance-Tracking und Visualisierung**

Ein umfassendes Dashboard visualisiert Signal-Performance, Erfolgsraten und Asset-Verteilung für kontinuierliche Systemoptimierung [[13]](#fn13)[[14]](#fn14)[[5]](#fn5).

![](data:application/octet-stream;base64,)

Beispiel: Performance-Tracking für Trading Signals

Das Performance-Tracking zeigt die kontinuierliche Verbesserung der Signal-Qualität über die Zeit, sowohl in Bezug auf Erfolgsrate als auch durchschnittliche Rendite [[13]](#fn13)[[14]](#fn14).

![](data:application/octet-stream;base64,)

Asset-Verteilung der Trading Signale mit Erfolgsraten

Die Asset-Verteilung bietet Einblicke in die Häufigkeit und Erfolgsrate von Signalen nach verschiedenen Anlageklassen, was für die Strategieoptimierung entscheidend ist [[3]](#fn3)[[7]](#fn7).

**Implementierungsroadmap**

**Phase 1: Grundsystem (Wochen 1-4)**

* Raspberry Pi 5 Setup mit Python 3.12/3.13 und aktuellen Modulversionen
* SQLite-Datenbank für persistente Speicherung
* Telegram-Bot-Integration für Signalbenachrichtigungen
* Basis-Datensammlung mit yfinance 0.2.63 [[1]](#fn1)[[16]](#fn16)[[12]](#fn12)[[6]](#fn6)

**Phase 2: ML-Pipeline (Wochen 5-8)**

* PyTorch-basierte Sentiment-Analyse mit FinBERT
* Technische Analyse-Engine mit bewährten Indikatoren
* Unterbrechbare ML-Verarbeitung auf Windows PC
* Performance-Tracking-System [[2]](#fn2)[[4]](#fn4)[[8]](#fn8)[[9]](#fn9)

**Phase 3: Dashboard und Produktivbetrieb (Wochen 9-12)**

* Streamlit-Dashboard für lokale Visualisierung
* Backtesting und Performance-Validierung
* 24/7-Deployment mit automatischer Überwachung
* pCloud-Backup-Integration [[13]](#fn13)[[14]](#fn14)[[15]](#fn15)[[10]](#fn10)[[11]](#fn11)

**Technische Spezifikationen**

**Hardware-Optimierung**

Ihre RTX 3090 Ti ermöglicht GPU-beschleunigtes Training von Transformer-Modellen wie FinBERT, während der Raspberry Pi 5 mit seinem ARM Cortex-A76 Prozessor kontinuierliche Datensammlung bewältigt [[2]](#fn2)[[4]](#fn4). Die empfohlene 256GB SSD für den Raspberry Pi bietet ausreichend Speicher für drei Jahre Betrieb [[8]](#fn8)[[18]](#fn18).

**Software-Kompatibilität**

Alle Module wurden auf Kompatibilität geprüft: yfinance 0.2.63, pandas 2.3.0, scikit-learn 1.7.0, PyTorch 2.7.1, Streamlit 1.45.1 und python-telegram-bot 22.1 [[1]](#fn1)[[2]](#fn2)[[13]](#fn13)[[14]](#fn14)[[12]](#fn12)[[5]](#fn5)[[6]](#fn6). Diese Versionen gewährleisten stabile Integration und optimale Performance.

**Nächste Schritte**

Das Projekt ist vollständig spezifiziert und kann sofort mit der Implementierung beginnen [[3]](#fn3)[[7]](#fn7). Das erstellte Spezifikationsdokument dient als zentrale Referenz für alle Entwicklungsschritte und kann bei Bedarf erweitert werden [[1]](#fn1).

Die modulare Architektur ermöglicht iterative Entwicklung mit kontinuierlichen Verbesserungen basierend auf realen Trading-Ergebnissen [[8]](#fn8)[[18]](#fn18)[[11]](#fn11). Durch die bewusste Fokussierung auf Einfachheit in der ersten Ausbaustufe kann das System schnell produktiv eingesetzt und später nach Ihren Erfahrungen erweitert werden [[3]](#fn3)[[7]](#fn7)[[10]](#fn10).

⁂

1. <https://www.youtube.com/watch?v=037_883wHGo>

1. <https://pytorch.org/get-started/previous-versions/>

1. <https://extraetf.com/de/etf-broker/trade-republic-etf-depot-test>

1. <https://www.reddit.com/r/StableDiffusion/comments/1k23rwv/quick_guide_for_fixinginstalling_python_pytorch/>

1. <https://docs.streamlit.io/develop/quick-reference/release-notes/2024>

1. <https://pypi.org/project/python-telegram-bot/>

1. <https://support.traderepublic.com/de-at/87-Welche-Derivate-kann-ich-bei-Trade-Republic-handeln>

1. <https://docs.python.org/3/library/sqlite3.html>

1. <https://www.youtube.com/watch?v=7KduSlMDgvg>

1. <https://github.com/tomgross/pcloud>

1. <https://blog.julik.nl/2025/01/maximum-speed-sqlite-inserts>

1. <https://www.youtube.com/watch?v=KYgpmPGDFBE>

1. <https://docs.streamlit.io/develop/quick-reference/release-notes/2025>

1. <https://docs.streamlit.io/develop/quick-reference/release-notes>

1. <https://docs.streamlit.io>

1. <https://python-yahoofinance.readthedocs.io/en/latest/api.html>

1. <https://mcpmarket.com/server/yfinance-trader>

1. <https://stackoverflow.com/questions/17536034/performance-improvements-for-python-very-large-sqlite-database>