Jahresprojekt  
2012/2013

AQUILA

Lastenheft

Nagy, Pawlowsky & Sochovsky

Versionierung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Autor | Datum | Status | Kommentar |
| 0.1 | Pawlowsky,  Sochovsky | 02.09.12 | draft | Produkteinsatz, Produktfunktionen, Produktdaten |
| 0.2 | Pawlowsky, Sochovsky | 07.09.12 | draft |  |
| 0.3 | Nagy | 10.09.12 | final | Überarbeitung, Qualitätsmanagement |

Nach Änderungen annehmen / ablehnen Inhaltsverzeichnis aktualisieren

Inhalt

1. Präambel 1

2. Zielbestimmungen 2

3. Produkteinsatz 2

4. Produktfunktionen 3

4.1. Funktionen der Website 3

4.1.1 Controlling -Funktionen der Website 3

4.2. Funktionen der Software 5

5. Produktdaten 7

6. Zwingende Randbedingungen 7

6.1. Produktumgebung und Systemintegration 7

6.2. Schnittstellen 7

7. Vertragsgegenstand 8

7.1. Lieferumfang 8

7.2. Produktleistungen 8

7.3. Produktbezogene Leistungen 8

8. Qualitätsanforderungen 8

9. Glossar 9

# Präambel

Viele großen Institutionen handeln seit geraumer Zeit voll automatisch mit Hilfe von Algorithmen an der Börse und haben damit auch teilweise für große Erträge gesorgt. Für kleinere Unternehmen und ambitionierte semiprofessionelle Trader sind solche Lösungen schwierig, da oft die nötige Kontrolle und die Steuerungsmöglichkeiten nicht vorhanden sind.

Manche bekannte Ansätze zum Controlling eines derartigen Programms erweisen sich dabei als unpraktisch. Zum Beispiel kann eine solche Software, auf einem Server laufend, oft nur durch den Zugriff über eine Remote-Desktop-Verbindung in ihren Einstellungen verändert werden. Die Visualisierung der Performance und der Überblick über die Handlungsweisen des Algorithmus sind meist ebenfalls sehr mangelhaft vorhanden. Auch ein Einblick in die Details und die Aufzeichnungen des Online-Brokers ist kaum zu finden.

Bei genau diesen Punkten soll das vorliegende Projekt Abhilfe schaffen und damit die Steuerung erleichternund die Übersichtlichkeit erhöhen.

# Zielbestimmungen

Es soll eine Software entwickelt werden, die automatisiert Aktien kauft und verkauft und die Signale dazu von einem existierenden Algorithmus erhält. Dazu soll die Software zuerst Kursdaten einer Aktie von einem Datenprovider laden. Danach soll der implementierte Algorithmus eine Entscheidung treffen, ob das Instrument in einer Long- oder Shortposition gehandelt werden soll. Diese Entscheidung soll anschließend über einen Online Broker abgewickelt werden, wodurch die entsprechende Transaktion für das eigene Depot durchgeführt wird.

Um die Nachteile eines algorithmisch automatisierten Handels auszugleichen, sollen außerdem diverse Controlling-Funktionen über eine zu entwickelnde Webseite durchführbar sein. Durch eine gesicherte Accountverwaltung soll sich der Trader von überall aus an der Webseite anmelden können, um Kontrollfunktionen durchzuführen. Dazu gehört sowohl das Anzeigen der aktuellen Performance wie die Überwachung der Einzelteile des Algorithmus als auch das nachträgliche Vornehmen von Einstellungen an der laufenden Software.

Dazu müssen die Software und die Website über eine Schnittstelle mit einander verbunden werden, um einen Datenaustausch zur Laufzeit zu ermöglichen. Außerdem muss die Applikation zu diesem Zweck auch ausreichend zur Laufzeit parametrisiert werden können.

# Produkteinsatz

Das Produkt soll den Benutzer, der tagsüber (oder auch nachts) seinen üblichen Tätigkeiten nachgehen will, unterstützen, den Gewinn seines Aktienportfolio von einem „virtuellen Manager“ verbessern zu lassen.

Außerdem soll es die Software ermöglichen, dass der Benutzer auch unterwegs, z.B. von seinem Tablet aus, die gewünschten Einstellungen treffen und im manuellen Modus die Benachrichtigungen erhalten kann.

Mittels Charts soll dem Benutzer außerdem übersichtlich die Arbeitsweise des Algorithmus nahe gebracht werden, wodurch gegebenenfalls automatisierte Entscheidungen ausgesetzt werden können.

# Produktfunktionen

## Funktionen der Website

/LF10/  
*Charts darstellen*Charts zur Darstellung der Algorithmen werden auf der Website angezeigt. Dabei muss besonders auf die Schnittstelle zur bereits bestehenden Software geachtet werden. Diese kann zum Beispiel über Webservices oder Datenbanken implementiert werden.

/LF20/  
*Account-Management bereitstellen*Über einen primären Account kann sich ein Benutzer anmelden, um alle sämtliche Controlling-Funktionen der Website durchführen zu können.

*/*LF30*/  
News anzeigen*Wenn möglich können Nachrichten-Headlines, die gehandelte Aktien beeinflussen, auf der Website angezeigt werden. Ein Beispiel wäre der Bloomberg-Newsfeed, der auch vom Programm benutzt werden könnte.

## Controlling-Funktionen der Website

/LF40/  
*Website-Software-Schnittstelle implementieren*  
Um die Software von der Website aus kontrollieren zu können, wird eine Schnittstelle benötigt, diese kann zum Beispiel über Datenbanken, Streams oder Webservices realisiert werden.

/LF41/  
*Investitionskapital einstellen*  
Die Höhe des Kapitals, das zur Investition freigegeben ist, kann je nach Aktie eingestellt werden.

/LF42/  
*Handeln starten*Das Handeln des Programms soll mittels einer Usereingabe gestartet werden können.

/LF43/  
*Handeln stoppen*  
Das Handeln der Software soll mittels einer Usereingabe gestoppt werden können.

/LF44/  
*Entscheidungsmodus ändern*Der User muss zwischen einem automatischen und einem manuellen Modus auswählen können; beim automatischen Modus soll die Software automatisiert entscheiden und handeln; beim manuellen Modus wird der User mittels einer Benachrichtigung (siehe */LF160/*) zum Entscheiden aufgefordert.

/LF45/  
*Benachrichtigungen verschicken*  
Dem User wird zum Beispiel mittels SMS, E-Mail oder Push-Benachrichtigung der Handlungsvorschlag zugesandt.

/LF46/  
*Aktie auswählen*  
Der User kann aus einer prädefinierten Liste die gewünschten Aktien zum Handeln selektieren.

## Funktionen der Software

/LF110/  
*Input speichern*Die vom Datenprovider ausgesendeten Daten (Bars) sollen gespeichert werden, um später weiter verarbeitet werden zu können.

/LF120/   
*Bars weiterleiten*Der bisherige Kurs wird in Form von Bars an den Rechenkern übergeben.

/LF130/   
*Entscheidung berechnen*Der Rechenkern soll aufgrund des Entscheidungsalgorithmus berechnen, wie das Wertpapier am besten zu behandeln ist (buy, sell, hold).

/LF140/   
*Entscheidung ausgeben*Die berechnete Entscheidung wird vom Rechenkern in den Verarbeitungskern geleitet, wo sie in der Konsole ausgegeben und ausgeführt wird.

/LF150/   
*Wertpapier kaufen*Das System kauft über den Online-Broker-Account eine bestimmte Anzahl an Wertpapieren.

/LF160/   
*Wertpapier verkaufen*

Das System verkauft über den Online-Broker-Account eine bestimmte Anzahl an Wertpapieren.

/LF170/   
*Log erstellen*Das System legt beim Programmstart lokal ein Log-File an, in dem alle Entscheidungen, versehen mit Datum und Uhrzeit, während des Programmlaufs eingetragen werden.

/LF180/  
*Kursdaten beziehen*  
Die Software muss eine Schnittstelle zum Beziehen aktueller und historischer Bars besitzen. Diese kann zum Beispiel über e-Signal oder Interactive Brokers realisiert werden.

/LF190/  
*Datenbankverbindung bereitstellen*  
Das Programm muss eine Datenbankschnittstelle erhalten und die berechneten Daten für die Website erreichbar machen.

/LF200/ *Laufzeitparameter ändern*Um die Controlling-Funktionen zu ermöglichen, muss das Programm genauer parametrisierbar sein. Es sollen zum Beispiel die Einstellungen auch während der Laufzeit verändert werden können.

/LF210/  
*Multi-Threading ermöglichen*Die Software muss Multi-Threading unterstützen, um die Entscheidung über automatisches Kaufen und Verkaufen von verschiedenen Aktien gleichzeitig zu treffen.

# Produktdaten

/LD10/  
*Datenprovider Anmeldedaten*Es muss ein Account für einen Datenprovider mit Username und Passwort vorhanden sein. Dieser muss Bars anbieten. (Mögliche Provider wären e-Signal oder Interactive Brokers.)

/LD20/  
*Historische Wertpapierdaten*Zu Testzwecken und auch um das spätere Berechnen zu ermöglichen, werden historische Daten über einen größeren Zeitraum benötigt.

# Zwingende Randbedingungen

## Produktumgebung und Systemintegration

Die fertiggestellte Software soll in der Programmiersprache C# und/oder in F# implementiert werden, um die Integration in eine .NET-Systemumgebung zu ermöglichen. Dabei ist der Algorithmus allenfalls in F# umgesetzt. Die Kompatibilität mit anderen Betriebssystemen als Windows kann dabei vernachlässigt werden.

## Schnittstellen

Als interne Schnittstelle wäre eine Datenschnittstelle erwünscht, die es ermöglicht sowohl die berechneten Entscheidungen als auch alle darunterliegenden und z.B. zum Zeichnen von Charts oder Kontrollieren der Software benötigten Dateien zwischen der Software und der Website auszutauschen. Die kann z.B. über eine Datenbankschnittstelle, Streams oder SOAP realisiert werden.

# Vertragsgegenstand

## Lieferumfang

Der Lieferumfang umfasst die kompilierte Software sowie ein Abbild der Website mit einer umfassenden Dokumentation zur Inbetriebnahme der Produkte.

Die fertiggestellte Software soll in einem Lizenzmodell vertrieben werden, wobei die Eigentums- und Nutzungsrechte bei den Verfassern des Programmcodes bleiben.

## Produktleistungen

/LL10/  
*Maximale Useranzahl*Das Account-Management der Website muss es ermöglichen mindestens 10 User anzulegen.

## Produktbezogene Leistungen

Für die Installation einer lizensierten Version der Software inklusive Web-Controlling-Oberfläche soll eine Installationsanleitung angelegt werden. Weitere Leistungen wie Wartung und Betrieb sind nicht vorgesehen.

# Qualitätsanforderungen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Produktqualität | Sehr Gut | Gut | Normal | Irrelevant |
| Funktionalität |  | x |  |  |
| Zuverlässigkeit |  | x |  |  |
| Benutzbarkeit |  | x |  |  |
| Effizienz |  |  | x |  |
| Änderbarkeit |  | x |  |  |
| Übertragbarkeit |  | x |  |  |

# Glossar

**Algorithmus** Ein Algorithmus ist eine aus endlich vielen Schritten bestehende eindeutige Handlungsvorschrift zur Lösung eines Problems oder einer Klasse von Problemen.

**MA** s. Moving Average

**Market Exposure** Ausmass, in dem ein Portfolio den Schwankungen eines Marktes ausgesetzt ist

**Moving Average** MA;Gleitender Mittelwert/Durchschnitt: Tiefpassfilter, der dazu verwendet wird, um Kurse zu glätten und temporäre Schwankungen zu entfernen. Dazu wird für jeden weiteren Wert mit den x vorigen Werten ein Durchschnitt berechnet, wodurch der MA dem Kurs hinterherhinkt.

**Performance** Güte des Algorithmus in Bezug auf Gewinn, Volatilität, Risiko und andere Faktoren.

**Trade** Entweder ein Kauf (long) oder Verkauf (short), der auf eine Order folgt.

**Bar** Eine Kombination aus den vier Zahlen (in dieser Reihenfolge) für Open-, High-, Low- und Close-Wert des Preises eines Wertpapiers über einen bestimmten Zeitabschnitt.

**Open** Der Open-Wert beschreibt den ersten Wert eines bestimmten Zeitabschnitts (Bar) eines Wertpapiers.

**High** Der High-Wert beschreibt den größten Wert innerhalb eines bestimmten Zeitabschnitts (Bars) eines Wertpapiers.

**Low** Der Low-Wert beschreibt den kleinsten Wert innerhalb eines bestimmten Zeitabschnitts (Bars) eines Wertpapiers.

**Close** Der Close-Wert beschreibt den letzten Wert eines bestimmten Zeitabschnitts (Bars) eines Wertpapiers.