

Pflichtenheft

AQUILA

Tradingsoftware mit Controlling-Website

Ausgeführt in Zuge des Projektmanagement-Unterrichts im 5. Jahrgang
Ausbildungszweig Systemtechnik/Medientechnik

unter der Leitung von
Prof. Mag. Hans Brabenetz
Abteilung für Informationstechnologie

eingereicht am Technologischen Gewerbemuseum Wien
Höhere Technische Lehr- und Versuchsanstalt
Wexstrasse 19-23, A-1200 Wien

von
Peer Nagy 5CHIT
Gabriel Pawlowsky, 5BHITS
Josef Sochovsky, 5BHITS

Wien, im Oktober 2012

Version	Autor	QS	Datum	Status	Kommentar
0.1	Nagy	Pawlowsky	25.10.2012	draft	Ziel- bestimmungen
0.2	Pawlowsky	Sochovsky	31.10.2012	draft	Produkt- funktionen, Produktdaten, Entwicklungs- umgebung
0.3	Nagy	Pawlowsky	31.10.2012	draft	Ziel- bestimmungen, Produkteinsatz
0.4	Nagy	Sochovsky	01.11.2012	draft	Produkt- leistungen
0.5	Sochovsky	Nagy	02.11.2012	draft	Produkt- umgebung, Globale Test- fälle Qualitäts- bestimmungen
0.6	Nagy	Sochovsky	04.11.2012	draft	Benutzer- schnittstelle
1.0	Pawlowsky	Nagy	21.11.2012	final	Letzte Fehler ausgebessert

Inhaltsverzeichnis

Abbildungsverzeichnis	vi
1 Zielbestimmung	1
1.1 Musskriterien	1
1.2 Wunschkriterien	1
1.3 Abgrenzungskriterien	2
2 Produkteinsatz	3
2.1 Anwendungsbereiche und Zielgruppen	3
2.2 Betriebsbedingungen	3
3 Produktumgebung	5
3.1 Software	5
3.2 Hardware	6
3.3 Orgware	6
4 Produktfunktionen	7
4.1 Funktionen der Website	7
4.1.1 Controlling-Funktionen der Website	8
4.1.2 Programmabläufe	11
4.2 Funktionen der Website	11
4.2.1 Programmabläufe	15
4.3 Aktivitätsdiagramm	16
5 Produktdaten	17
6 Produktleistungen	19

7 Benutzerschnittstelle	21
8 Qualitätsbestimmung	25
9 Globale Testfälle	29
10 Entwicklungsumgebung	31
10.1 Software	31
10.2 Hardware	32
10.3 Orgware	32
11 Projektplanung	35
11.1 Projektstrukturplan	35
11.2 Balkenplan	36
11.3 Meilensteinplan	36
Glossar	37

Abbildungsverzeichnis

4.1	Ein möglicher Standardablauf von Noctua	16
7.1	Website-Grundgerüst	22
11.1	Projektstrukturplan	35
11.2	Balkenplan	36

KAPITEL 1

Zielbestimmung

1.1 Musskriterien

Das Produkt ermöglicht das automatisierte Handeln von Aktien über einen bestehenden Online-Broker-Account. Kursdaten und eventuell andere relevante Daten, werden über einen Datenanbieter abgerufen und an das Algorithmus-Modul zur Verarbeitung weitergereicht.

Die Steuerung und das Controlling der Software kann manuell über eine Web-schnittstelle vorgenommen werden. Über diese können die zu handelnden Aktien selektiert und das einzusetzende Kapital festgelegt werden. Zusätzlich soll dem Benutzer der Kursverlauf als Chart, die aktuelle Entscheidung des Algorithmus und falls praktikabel die Stärke der Entscheidung angezeigt werden. Das Risiko ist je Aktie regelbar. Die Implementierung der Risikosteuerung kann durch eine der folgenden Methoden erfolgen:

- Regelung der Höhe des Investitionskapitals
- Festsetzung der Höhe der Cut-Losses-Schwelle

Der Zugang zur Website ist gesichert und durch eine Benutzerverwaltung, die gegebenenfalls mehrere Accounts zulässt, geregelt.

1.2 Wunschkriterien

Die Webschnittstelle stellt neben den reinen Kursdaten zusätzlich noch relevante Moving Averages (MAs) und eine Art der Support- und Resistance-Level zur

Bestimmung von überkauften und überverkauften Bereichen dar, damit Entscheidungen des Algorithmus nachvollzogen werden können. Aus dem selben Grund wird sowohl die tatsächlich realisierte Performance über eine Zeitperiode als auch der noch nicht realisierte Gewinn oder Verlust des aktuellen Trades (vorzüglich inkl. Transaktionsgebühren) angezeigt.

Um die Entscheidungen besser nachvollziehen zu können oder auch um potentielle Fehlentscheidungen manuell korrigieren zu können, werden auf der Webseite aktuell relevante News-Headlines angezeigt.

1.3 Abgrenzungskriterien

Die bezogenen Kursdaten müssen, bedingt durch hohe Kosten, während der Entwicklungszeit nicht in Echtzeit bezogen werden. Allerdings muss darauf geachtet werden, dass die Umstellung für den tatsächlichen Einsatz problemlos vorgenommen werden kann.

Standardmäßig wird aufgrund der Datenkosten nur ein eingeschränkter Teil von Aktien unterstützt. Zusätzliche Aktien können gegen Entgelt nachträglich hinzugefügt werden.

Falls für das Versenden von Benachrichtigungen ein Server oder ein Gateway notwendig ist, ist dieses vom Kunden der Software bereitzustellen. Dieses Service ist nicht Teil des Produktes.

Die Zielgruppe des Produktes handelt mit relativ kleinen Positionen, weshalb Kaufentscheidungen weder in mehrere Orders geteilt (Order-Splitting) noch auf Slippage (Änderungen im Aktienpreis durch eigene Order) Rücksicht genommen wird.

2.1 Anwendungsbereiche und Zielgruppen

Das Produkt implementiert ein Handelssystem für Aktien. Daher ist für einen vernünftigen Umgang mit der Software ein Mindestwissen über Aktienhandel und Börsengeschäfte vorauszusetzen. Vorteilhaft wäre ebenso ein Verständnis von technischer Analyse und häufig genutzten Indikatoren, um darauf basierende Handelssysteme zu durchschauen. Programmier- b.z.w. vertiefende Computerkenntnisse müssen hingegen nicht vorhanden sein.

Als Zielgruppe sind insbesondere kleine und mittlere Unternehmen (KMU) anvisiert, in denen sich bereits Personen mit finanzwirtschaftlichen Angelegenheiten befassen. Hinzu kommen private Einzelpersonen, die über das nötige Kapital für kurzfristigen Aktienhandel verfügen und mit wenig Aufwand ein Komplettsystem dazu anwenden möchten.

2.2 Betriebsbedingungen

Die Software soll es prinzipiell ermöglichen unbeaufsichtigt und selbstständig zu arbeiten, wobei, da es sich um einen substanziellen Kapitalaufwand handeln kann, es insbesondere in der Anfangszeit ratsam ist, die Abläufe der Software zu überwachen.

Sowohl die Handelssoftware selbst als auch die Webseite laufen auf einem Server. Beide Komponenten müssen die Aufteilung auf und somit die Kommunikation zwischen mehreren Servern nicht unbedingt unterstützen, jedoch wäre dies vorteilhaft, da sowohl Zugriffsrechte separat geregelt werden könnten, als auch die Handelssoftware die vollen Kapazitäten des Servers nutzen kann.

Software und Website sollen einen Betrieb rund um die Uhr ermöglichen, ob ein solcher im Einsatz tatsächlich realisiert wird, liegt am Kunden.

3.1 Software

Um das Ausführen der Software zu gewährleisten, ist es nötig einen Webserver anbieten zu können. Weil das Teilprodukt, die Website, hauptsächlich mit ASP.NET geschrieben ist, muss einer der folgenden Webserver lauffähig installiert sein:

- **IIS** (Internet Information Services)
- **Apache-Webserver** (mit „mod_aspdotnet“ und „mod_mono“)
- **XSP-Webserver**
- **Cassini-Webserver**

Im Falle, dass ein Linuxserver für die Website benutzt werden soll, wird ein Apache oder der XSP-Webserver empfohlen, damit es zu keinen schwerwiegenden Problemen kommt.

Es wird angeboten, Website und Software auf unterschiedlichen Computern zu installieren und dort zu benutzen. Für das reibungslose Integrieren der Software wird benötigt:

- ein Datenbankserver zum Interagieren mit der Website (Postgresql)
- ein installierter und funktionierender **Interactive Brokers (IB)-Client**
- ein Client von e-Signal
- die aktuellste Version des **.NET-Frameworks**

3.2 Hardware

Im Falle **eines** Servers benötigt der Rechner auf jeden Fall 4 GB Arbeitsspeicher, zirka 100GB freien Speicher und zumindest einen Dual-Core Prozessor.

Allerdings liegen diese Werte bereits unter dem heutigen Standard für normale Server. Wenn man die mindeste Konfiguration wählt, wird empfohlen, keine weiteren Servertätigkeiten über dieses Gerät zu vollziehen. Es ist außerdem eine Internetanbindung erforderlich, sowie ausreichend viel Speicherplatz auf der/den Festplatte/n.

3.3 Orgware

Der Server, auf dem die Software und die Website verwendet werden, muss mit dem Internet verbunden sein, damit er sich mit IB und e-Signal verbinden kann. Dazu wird zuzüglich ein Account dieser beiden Anbieter benötigt.

KAPITEL 4

Produktfunktionen

4.1 Funktionen der Website

/F010/ **Charts darstellen**

Beschreibung	Charts zur Darstellung der Algorithmen werden auf der Website angezeigt. Dabei muss besonders auf die Schnittstelle zur Software geachtet werden. Dies wird über eine Sicherung der darzustellenden Daten in einer Datenbank realisiert.
Akteure	System
Teilsystem	Website
Ziel	Darstellung von Charts mit den Algorithmusnberechnungen auf der Website.
Vorbedingungen	/F040/ Website-Software-Schnittstelle implementieren
Nachbedingungen	-

/F020/ **Account-Management bereitstellen**

Beschreibung	Über zumindest einen primären Account kann sich ein Benutzer anmelden, um alle Controlling-Funktionen der Webseite durchführen zu können. Außerdem kann der Benutzer über diesen Account ein Portfolio aus Aktien auswählen, die er beobachtet. Zu diesem Portfolio kann sich der Benutzer dann alle Charts und Entscheidungen, die Aquila erzeugt, anzeigen lassen.
Akteure	System
Teilsystem	Website
Ziel	Benutzer sollen Accounts erstellen können.
Vorbedingungen	-
Nachbedingungen	-

/F030/ **News anzeigen**

Beschreibung	Es werden Nachrichten-Headlines, die gehandelte Aktien beeinflussen könnten, auf der Webseite angezeigt. Dafür wird ein frei zugänglicher RSS-Feed verwendet.
Akteure	System
Teilsystem	Website
Ziel	Anzeigen von einschlägigen News auf der Website.
Vorbedingungen	-
Nachbedingungen	-

4.1.1 **Controlling-Funktionen der Website**/F040/ **Website-Software-Schnittstelle implementieren**

Beschreibung	Um die Software von der Website aus kontrollieren zu können, wird eine Schnittstelle benötigt. Diese wird über Windows Communication Foundation (WCF) realisiert und soll alle Einstellungen, die auf der Website getätigt werden, einfach in die Software integrieren.
Akteure	System
Teilsystem	Software, Website
Ziel	Software und Website sollen kommunizieren können.
Vorbedingungen	-
Nachbedingungen	-

/F041/ Investitionskapital einstellen

Beschreibung	Die Höhe des Kapitals, das zur Investition freigegeben ist, oder die Anzahl kann je nach Aktie von der Website in der Software eingestellt werden.
Akteure	Benutzer
Teilsystem	Website
Ziel	Einstellung des Investitionskapitals der Software.
Vorbedingungen	/F040/ Website-Software-Schnittstelle implementieren, /F200/ Laufzeitparameter ändern
Nachbedingungen	-

/F042/ Handeln starten

Beschreibung	Das Handeln des Programms soll mittels einer Usereingabe auf der Website gestartet werden können.
Akteure	Benutzer
Teilsystem	Website
Ziel	Starten des Handels von der Website in der Software.
Vorbedingungen	/F040/ Website-Software-Schnittstelle implementieren, /F200/ Laufzeitparameter ändern
Nachbedingungen	-

/F043/ Handeln stoppen

Beschreibung	Das Handeln der Software soll mittels einer Usereingabe auf der Website gestoppt werden können.
Akteure	Benutzer
Teilsystem	Website
Ziel	Stoppen des Handels von der Website in der Software.
Vorbedingungen	/F040/ Website-Software-Schnittstelle implementieren, /F200/ Laufzeitparameter ändern
Nachbedingungen	-

/F044/ Entscheidungsmodus ändern

Beschreibung	Der User muss von der Website aus zwischen einem automatischen und einem manuellen Modus wählen können. Beim automatischen Modus soll die Software automatisiert entscheiden und handeln. Beim manuellen Modus wird der User mittels einer Benachrichtigung (siehe /F160/) zum Entscheiden aufgefordert.
Akteure	Benutzer
Teilsystem	Website
Ziel	Man soll von der Website aus in der Software zwischen einem manuellen und einem automatischen Handelsmodus wählen können.
Vorbedingungen	/F040/ Website-Software-Schnittstelle implementieren, /F200/ Laufzeitparameter ändern
Nachbedingungen	-

/F045/ Benachrichtigungen verschicken

Beschreibung	Dem User wird zum Beispiel mittels SMS jeder Handelsvorschlag zugesandt. Er muss dafür einen SMS-Server oder ein SMS-Gateway auf der Website spezifizieren, über die die SMS versendet werden, und auch alle Kosten, die durch den Versand der SMS anfallen, selbst übernehmen.
Akteure	System
Teilsystem	Website
Ziel	Der User wird über Handelsvorschläge informiert.
Vorbedingungen	/F040/ Website-Software-Schnittstelle implementieren
Nachbedingungen	-

/F046/ Aktie auswählen

Beschreibung	Der User kann von der Website aus einer zuvor definierten Liste die gewünschten Aktien zum Handeln mit der Software selektieren.
Akteure	Benutzer
Teilsystem	Website
Ziel	Möglichkeit von der Website aus, die Aktien, mit denen die Software handelt auszuwählen.
Vorbedingungen	/F040/ Website-Software-Schnittstelle implementieren, /F200/ Laufzeitparameter ändern
Nachbedingungen	-

4.1.2 Programmabläufe

Ausführung von Controllingoperationen (Erfolg)

Schritt	Akteur	Beschreibung
1	Benutzer	Meldet sich auf der Website an
2	System	Loggt den Benutzer ein
3	Benutzer	Wählt die Unterseite „Controlling“
4	Benutzer	Wählt alle Einstellungen aus, die er verändern möchte
5	Benutzer	Drückt den „Speichern“-Button
6	System	Baut eine WCF-Verbindung zur Software auf
7	System	Die getätigten Einstellungen werden übermittelt
8	System	Die Software arbeitet jetzt mit den neuen Einstellungen

Anzeigen von Charts und Entscheidungen (Erfolg)

Schritt	Akteur	Beschreibung
1	Benutzer	Meldet sich auf der Website an
2	System	Loggt den Benutzer ein
3	Benutzer	Wählt die Unterseite „Charts“
4	System	Fragt die aktuellen Daten zur Ermittlung der Charts aus der Datenbank ab
5	System	Berechnet die Charts
6	System	Zeigt die Charts und die aktuellen Entscheidungen an

4.2 Funktionen der Website

/F110/ Input speichern

Beschreibung	Die vom Datenprovider empfangenen Bars sollen entsprechend gespeichert werden, um später weiter verarbeitet werden zu können.
Akteure	System
Teilsystem	Software
Ziel	Speichern der historischen und aktuellen Aktiendaten vom Datenprovider.
Vorbedingungen	/F180/ Kursdaten beziehen
Nachbedingungen	/F120/ Bars weiterleiten

/F120/ Bars weiterleiten

Beschreibung	Der bisherige Kurs wird in Form von Bars an den Rechenkern übergeben, damit dieser mit Hilfe des Algorithmus Entscheidungen treffen und Signale generieren kann.
Akteure	System
Teilsystem	Software
Ziel	Transfer der Daten aus „/F110/ Input speichern“ in den Rechenkern
Vorbedingungen	/F110/ Input speichern
Nachbedingungen	/F130/ Entscheidung berechnen

/F130/ Entscheidung berechnen

Beschreibung	Der Rechenkern soll aufgrund des Entscheidungsalgorithmus berechnen, wie das Wertpapier am besten zu behandeln ist (buy, sell, hold). Dafür wird ein bereits bestehender Algorithmus verwendet, bei dem sicher gestellt werden kann, dass er einwandfrei und korrekt funktioniert.
Akteure	System
Teilsystem	Software
Ziel	Kaufs- oder Verkaufsentscheidung berechnen.
Vorbedingungen	/F120/ Bars weiterleiten
Nachbedingungen	/F140/ Entscheidung ausgeben

/F140/ Entscheidung ausgeben

Beschreibung	Die berechnete Entscheidung wird aus dem Rechenkern in den Verarbeitungskern geleitet, wo sie einerseits ausgegeben und andererseits ausgeführt werden.
Akteure	System
Teilsystem	Software
Ziel	Anzeige und Durchführung der Entscheidung.
Vorbedingungen	/F130/ Entscheidung berechnen
Nachbedingungen	/F150/ Wertpapier kaufen oder /F160/ Wertpapier verkaufen

/F150/ Wertpapier kaufen

Beschreibung	Das System kauft über den Online-Broker-Account eine bestimmte Anzahl an Wertpapieren.
Akteure	System
Teilsystem	Software
Ziel	Kaufen eine Wertpapiers.
Vorbedingungen	/F140/ Entscheidung ausgehen
Nachbedingungen	-

/F160/ Wertpapier verkaufen

Beschreibung	Das System verkauft über den Online-Broker-Account eine bestimmte Anzahl an Wertpapieren.
Akteure	System
Teilsystem	Software
Ziel	Verkauf eines Wertpapiers.
Vorbedingungen	/F140/ Entscheidung ausgehen
Nachbedingungen	-

/F170/ Log erstellen

Beschreibung	Das System legt beim Programmstart lokal ein Log-File an, in dem alle Entscheidungen versehen mit Datum und Uhrzeit während des Programmlaufs eingetragen werden. Besondere Fehler oder Probleme, die während des Programmlaufs auftreten, werden ebenfalls im Log-File vermerkt.
Akteure	System
Teilsystem	Software
Ziel	Erstellung eines Log-Files.
Vorbedingungen	/F150/ Wertpapier kaufen oder /F160/ Wertpapier verkaufen
Nachbedingungen	-

/F180/ Kursdaten beziehen

Beschreibung	Die Software muss eine Schnittstelle zum Beziehen von sowohl aktuellen als auch historischen Bars besitzen. Diese wird zum Beziehen eines großen Spektrums an Daten über eSignal realisiert.
Akteure	System
Teilsystem	Software
Ziel	Beziehen der Daten von einem der genannten Datenprovider.
Vorbedingungen	-
Nachbedingungen	/F110/ Input speichern

/F190/ Datenbankverbindung bereitstellen

Beschreibung	Das Programm muss eine Datenbankschnittstelle erhalten und die berechneten Daten für die Website erreichbar machen. Es müssen damit alle Entscheidungen, Signale und diagrammrelevante Daten für die Website zugänglich gemacht werden, um es ihr zu ermöglichen, alle Signale und Diagramme optimal darzustellen.
Akteure	System
Teilsystem	Software
Ziel	Datenbankschnittstelle zur Sicherung von Daten implementieren.
Vorbedingungen	-
Nachbedingungen	-

/F200/ Laufzeitparameter ändern

Beschreibung	Um die Controlling-Funktionen zu ermöglichen, muss das Programm genauer parametrisierbar sein. Das heißt, es sollen alle Controlling-Einstellungen (/F041/ - /F046/) auch während der Laufzeit verändert werden können.
Akteure	System
Teilsystem	Software
Ziel	Ermöglichung der Änderung von Software-Parametern zur Laufzeit.
Vorbedingungen	-
Nachbedingungen	-

/F210/ **Multi-Threading ermöglichen**

Beschreibung	Die Software muss Multi-Threading unterstützen, um die Entscheidung über automatisches Kaufen und Verkaufen von mehreren verschiedenen Aktien gleichzeitig treffen zu können.
Akteure	System
Teilsystem	Software
Ziel	Mehrere Aktien sollen gleichzeitig beachtet werden.
Vorbedingungen	-
Nachbedingungen	-

4.2.1 Programmabläufe**Starten der Software (Erfolg)**

Schritt	Akteur	Beschreibung
1	Benutzer	Startet die Software
2	System	Die Software bezieht die aktuellen Aktienkursdaten vom Online-Datenprovider.
3	System	Konvertiert die Daten in Bars und leitet diese an den Rechenkern weiter.
4	System	Der aktuelle Algorithmus wird gestartet und es wird eine Entscheidung berechnet.
5	System	Die Ergebnisse der Berechnungen und alle dafür benötigten Daten werden in der Datenbank abgespeichert.
6	System	Die berechnete Entscheidung wird durch Übermittlung an einen Online-Broker in die Tat umgesetzt und Aktien werden gekauft oder verkauft.
7	System	Die Software wartet nun auf etwaige Anfragen von der Website über WCF und beginnt anderenfalls nach jeder Periode (gewählte Barlänge) neu.

4.3 Aktivitätsdiagramm

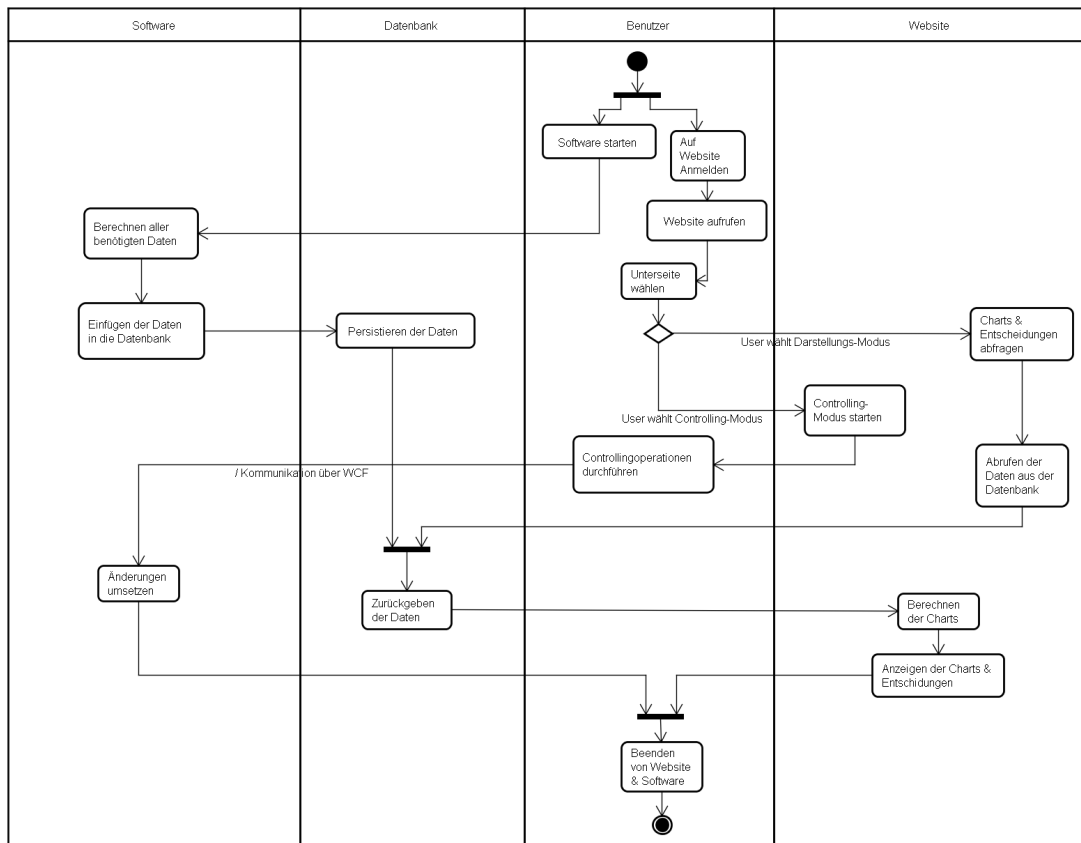


Abbildung 4.1: Ein möglicher Standardablauf von Noctua

KAPITEL 5

Produktdaten

/D010/ **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 InteractiveBrokers). Von ihm werden sowohl die aktuellen, als auch die historischen Aktienkursdaten bezogen.

/D020/ **Online-Broker Anmeldedaten**

Es muss ein Account für einen Online-Broker mit Username und Passwort vorhanden sein. In Frage kommt hierbei eigentlich nur ein InteractiveBrokers-Account, da alle anderen auf diesem Gebiet für Aquila zu teuer oder zu wenig umfangreich sind.

/D030/ **Historische Wertpapierdaten**

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

KAPITEL 6

Produktleistungen

/L010/ **Maximale Useranzahl**

Das Account-Management der Website muss es ermöglichen, mindestens 10 User anzulegen.

/L020/ **Minmale Aktienzahl**

Es müssen mindestens 5 handelbare Aktien für die Webseite konfiguriert werden.

/L030/ **Paralleler Handel**

Sowohl die Software als auch die Website sollen mindestens 10 Aktien gleichzeitig verwalten und handeln können.

KAPITEL 7

Benutzerschnittstelle

Die Benutzeroberfläche von Aquila besteht aus einer Website zur Steuerung und Information des Benutzers, die im Folgenden spezifiziert wird.¹

Die Website informiert über aktuelle Kennzahlen und Daten zu verwalteten Aktien, zeigt Performanceinformationen des verwalteten Portfolios an und stellt Charts dar, mit deren Hilfe historische Entwicklungen interpretiert werden können und der Algorithmus aus menschlicher Sicht verständlicher und übersichtlicher wird. Außerdem besitzt die Website eine zentrale Steuerfunktion, d.h. Einstellungen, Veränderungen und Parameter können über sie an die Software weitergegeben werden.

Bestandteile der Website sind eine Seite zur Übersicht über verwaltete Aktien, auf der diese sowohl bearbeitet als auch neue hinzugefügt werden können. Eine Seite gibt Informationen über die Performance des Portfolios bzw. auch über den Algorithmus. Im Einstellungsbereich sind Einstellungen zu Trading-relevanten Themen, wie beispielsweise Voreinstellungen für die Cut-Loss-Schwelle, sowie Accounteinstellungen vorzunehmen. Abbildung 7.1 stellt ein Grundgerüst der Aktien-Seite dar. Über die Suchfunktion können neue Aktien gefunden und hinzugefügt werden. Die Tabelle zeigt die wichtigsten Daten zu den verwalteten Aktien an. Durch Klick auf eines der Symbole gelangt der Benutzer zur Informationsseite dieser Aktie, wo Charts und jeweilige Performanceinformationen präsentiert werden.

¹Die Bezeichnungen und das aktuelle Design der beschriebenen Bereiche sind vorläufiger Natur und können sich im Verlauf des Projektes noch ändern.

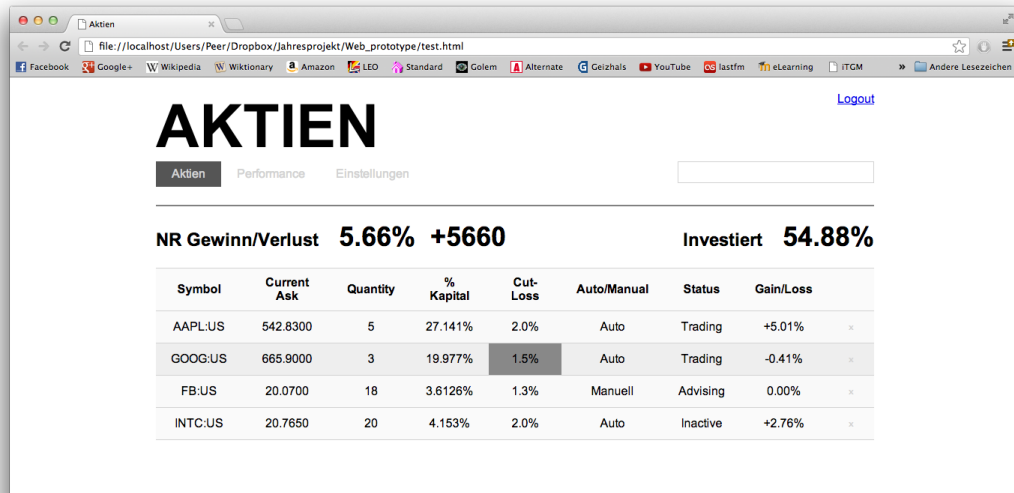


Abbildung 7.1: Website-Grundgerüst

Die Seitenstruktur weist folgende Struktur auf:

- Aktien
 - Symbolinformation
- Performance
- Einstellungen
 - Trading
 - Account

/B10/ Grafisches Design

Das Design soll modern und schlicht sein, wodurch eine flotte Bedienung erreicht wird und Informationen schnell aufgenommen werden können. Durch einfache Farbkontraste werden auf simple Art und Weise Hinweise über den Status der Aktie oder mögliche Benutzerinteraktionen angezeigt.

/B20/ Wechsel der betrachteten Aktie

Im Informationsbereich kann der Benutzer aus den Aktien, die er aktuell handelt, eine auswählen, für die anschließend alle Informationen und Charts angezeigt werden.

/B30/ Anzeige der Performance

Die Performance des Algorithmus wird in Zahlen für die betrachtete Aktie über

eine bestimmte Zeitperiode angezeigt.

/B40/ **Charts**

Zur Information des Benutzers sollen die Preisentwicklung der betrachteten Aktie, die MAs und sowohl Einstiegs- als auch Ausstiegspunkte des Algorithmus als Chart angezeigt werden. Dies soll kurz- und langfristig möglich sein. Außerdem soll das aktuelle Ergebnis des Algorithmus angezeigt werden.

KAPITEL 8

Qualitätsbestimmung

Produktqualitäten	Sehr Gut	Gut	Normal	Nicht relevant
Funktionalität	X			
Angemessenheit	X			
Richtigkeit	X			
Interoperabilität	X			
Ordnungsmäßigkeit	X			
Sicherheit		X		
Zuverlässigkeit		X		
Reife			X	
Fehlertoleranz		X		
Wiederherstellbarkeit	X			
Benutzbarkeit		X		
Verständlichkeit		X		
Erlernbarkeit			X	
Bedienbarkeit	X			
Effizienz	X			
Zeitverhalten		X		
Verbrauchsverhalten	X			
Änderbarkeit		X		
Analysierbarkeit	X			
Modifizierbarkeit		X		
Stabilität	X			
Prüfbarkeit	X			
Übertragbarkeit			X	
Anpassbarkeit		X		
Installierbarkeit			X	
Konformität			X	
Austauschbarkeit			X	

Funktionalität

Es ist notwendig, dass die Funktionalität des Produkts sehr zuverlässig ist. Bei einem Versagen kann es schnell zu einem sehr großen Kapitalverlust kommen. Dies ist natürlich der Worst-Case. Allerdings möchte man mit diesem Produkt eine allzeitbereite Plattform anbieten, der man vertrauen kann und auf keinen Fall ein Produkt, das schnell vom Konsumenten abgetan wird, weil es nicht zuverlässig arbeitet.

Es ist wichtig, dass die Daten, die gespeichert/ausgegeben/mitgeloggt werden, immer der Richtigkeit entsprechen und nicht falsch sind. Diese Daten unterliegen einer Ordnung, die immer die gleiche sein und keine willkürlichen Änderungen beinhalten soll.

Natürlich ist die Sicherheit auch ein wichtiges Thema, denn wenn darauf zu wenig geachtet wird, kann unerlaubtes Personal oder Außenstehende zu viel Einsicht in Firmenangelegenheiten erhalten; Dann kann auch passieren, dass unerlaubte Änderungen am Verhalten der Software vorgenommen werden.

Zuverlässigkeit

Das entwickelte Produkt soll zuverlässig arbeiten und keine Abstürze verzeichnen. Es hat einen wohl überlegten Grundgedanken und wandelt diesen um. Die Entscheidungen, die es trifft, basieren immer nur auf den Einstellungen des Benutzers und werden nicht willkürlich oder ohne logische Begründung getroffen. Die Fehlertoleranz soll es ermöglichen, bei einer fehlerhaften Datenübertragung kein Programmversagen zu verursachen.

Falls es doch zu einem Absturz kommt, kann mittels der Logs und den gespeicherten Transaktionen in der Datenbank das Produkt vollständig wiederhergestellt werden.

Benutzbarkeit

Die Benutzbarkeit ist ein wichtiges Thema, welches auf keinen Fall außer Acht gelassen werden darf. Die Absicht des Trading ist komplex genug, deswegen soll sich der Benutzer nicht noch zusätzlich mit einem komplexen und überladenen Produkt überfordert fühlen. Es soll leicht verständlich sein und man soll kein neues Wissen erlernen müssen, um damit umzugehen. Die Bedienung soll schnell und effizient von der Hand gehen und den Benutzer nicht allzu viel Zeit kosten.

Effizienz

Es ist wichtig, dass das Handeln mit Wertpapieren schnell vonstatten geht und den Computer nicht übermäßig belastet. Bei zu großer Rechenzeit kann es passieren, dass der berechnete Wert nicht mehr für das gegenwärtige Zeitfenster passt.

Änderbarkeit

Bei diesem Produkt ist es sehr wichtig, die Ergebnisse analysieren zu können. An dem Erzeugnis selbst soll man nicht mehr viel ändern.

Übertragbarkeit

Das Produkt wird vom Projektteam auf eine Kundenumgebung portiert, aber danach ist es für das Projekt nicht relevant, weitere Leistungen diesbezüglich anzubieten.

KAPITEL 9

Globale Testfälle

/T010/ **Angabe eines falschen Accounts**

Wenn ein Benutzer auf der Website versucht, sich mit einem nicht vorhandenen Benutzernamen oder einem falschen Passwort anzumelden, soll es ihm nicht möglich sein, weitere Einstellungen oder Veränderungen am System vorzunehmen.

/T020/ **Börse hat geschlossen**

Wenn der Fall eintritt, dass die Börse geschlossen ist, soll weiterhin die Möglichkeit bestehen, auf der Website Einstellungen und Veränderungen vorzunehmen; wenn diese die Software betreffen, wird diese, die Änderungen übernehmen und bei Ablaufstart anwenden.

/T040/ **Aktie entfernen**

Eine Aktie, mit der gehandelt wird, soll über die Website aus dem Handelsumfang der Software einfach und schnell entfernt werden können. Es ist wichtig, dass diese Funktion sehr schnell abläuft, damit diese Aktie nicht länger gehandelt wird.

/T040/ **Empfangen der Einstellungen von der Website**

Wenn die Website Befehle erhält, die Einstellungen der Software zu ändern, sollen diese Änderungen an die Software übertragen und dann übernommen werden. Dies soll im Logfile ersichtlich sein.

/T050/ **Aktie hinzufügen**

Es soll auf der Website möglich sein, eine Aktie hinzuzufügen mit der gehandelt wird. Diese Einstellung soll über die Verbindung mit der Software direkt übergeben werden und sofort in Kraft treten. Diese Interaktion ist essentiell, weil man nicht lange Wartezeiten in Kauf nehmen will, bevor eine neue Aktie hinzugefügt

werden kann.

/T060/ Daten empfangen, Log bearbeiten und Entscheidung speichern

Nach dem Start der Software sollen Daten empfangen werden, welche dann in den Algorithmus eingespeist werden, um mit ihnen zu rechnen. Das Ergebnis dieses Prozesses ist eine Entscheidung. Diese kann man im Log, in der Datenbank oder auch auf der Website nachvollziehen.

/T070/ Charts anzeigen

Auf der Website soll es möglich sein, Charts zu den letzten Berechnungen zu sehen. Zusätzlich kann man auf der gleichen Seite, auf der sich die Charts befinden, auch die letzte Entscheidung des Algorithmus betrachten. Dadurch erkennt man die Konnektivität zwischen Website und Software.

10.1 Software

- **Microsoft Office Suite**

Die Office-Produkte von Microsoft haben in diesem Projekt eine hohe Bedeutung. Bei nahezu jeder Art von Dokumentation werden die Programme Word bzw. Excel dieser Produktreihe verwendet. Sie bieten einen hohen Komfort im Umgang mit Dokumenten und sind bei den Projektmitgliedern bereits über Jahre hinweg zur Gewohnheit geworden.

- **Astah**

Astah findet vor allem in der Planungsphase des Projekts seinen Nutzen. Denn die UML-Diagramme, die in der Planung zur Programmierung erstellt werden müssen, können damit sehr einfach und schön dargestellt werden.

- **Microsoft Visio**

Microsoft Visio, das zusätzlich zur standardmäßigen Ausstattung der Office-Suite installiert werden kann, findet bei diesem Projekt auch seine Verwendung. Es wird hauptsächlich verwendet, um die ER-Diagramme für die Konstruktion der Datenbank zu zeichnen, die die Erstellung der Datenbank deutlich vereinfachen.

- **Microsoft Visual Studio**

Microsoft Visual Studio ist eigentlich die meistbenutzte Software bei diesem Projekt. Es dient als Entwicklungsumgebung, die alles verwaltet, was im Laufe des Projekts an C#- und auch F#-Code anfällt. Außerdem wird die Dokumentation des Codes in Microsoft Visual Studio vorgenommen.

- **GIT-Tool**

Um auf den verwendeten GIT-Server zugreifen zu können, ist ein GIT-Tool nötig. Da sich Aquila ausschließlich auf der Windows-Ebene bewegt und ein starkes aber dennoch einfaches GIT-Tool benötigt, fiel die Entscheidung auf SmartGIT. Dieses Tool ist für die kommerzielle Nutzung leider kostenpflichtig. Da dieses Projekt allerdings keinen direkt kommerziellen Nutzen haben soll, kann es ganz einfach kostenlos benutzt werden. Falls es allerdings dennoch notwendig sein sollte, unter einem anderen Betriebssystem zu arbeiten, so gibt es auf jeder anderen Plattform auch reichlich andere Alternativen.

10.2 Hardware

- **Global verfügbarer Server (GIT)**

Ein Server wird benötigt, um darauf GIT laufen zu lassen. Dieser sollte natürlich global verfügbar sein. Über Passwort und Usernamen können sich dann die Projektmitglieder mit diesem verbinden und die neuste Version des Quellcodes beziehen bzw. das Repository am GIT-Server aktualisieren. Dadurch wird ein Arbeiten im Team vereinfacht und optimiert, indem die Mitglieder immer den aktuellen Quellcode zur Verfügung haben. Die Authentifizierung am Server gewährleistet natürlich den Schutz vor unbefugtem Zugriff.

- **Computer für die Mitglieder**

Für den Fortschritt des Projekts und die Entwicklung des daraus entstehenden Systems ist für jedes Mitglied ein Computer bzw. Laptop notwendig. Diese Geräte müssen natürlich einen flüssigen Umgang mit der im Projekt verwendeten Software gewährleisten.

- **Weiterer Server**

Da sich manche User aus Sicherheitsgründen wünschen, den Application-Server und den Webserver, die bei der Verwendung des Produkts Aquila benötigt werden, auf unterschiedliche Rechner aufzuteilen, benötigen wir zu Testzwecken einen zusätzlichen Server. Dieser kann allerdings auch auf einem anderen PC aufgesetzt werden, also werden hierfür zum Testen keine Kosten anfallen.

10.3 Orgware

- **UML (Unified Modeling Language)**

Durch die Verwendung der standardisierten UML für Diagramme (und ähn-

liches) wird die Kommunikation im Team optimiert und es werden Unklarheiten beseitigt.

- **.Net Framework Library**

Bei etwaigen Problemen mit bestimmten Strukturen von C#/F# wird die offizielle Library verwendet.

Diese findet man unter <http://msdn.microsoft.com/en-us/library/>.

- **Richtlinien zur Codierung**

Um den Code zu vereinheitlichen und die Einlesezeit zu minimieren, werden gemeinsame Richtlinien für die Codierung geschaffen. Diese bestimmen, wie Variablen oder Methoden zu benennen sind und definieren die Struktur des Codes.

KAPITEL 11

Projektplanung

11.1 Projektstrukturplan

	PSP-Code	Vorgangsname	Dauer	Anfang	Fertig stellen	Vorgänger
1	1	▢ Aquila	106 Tage	Mi 14.11.12	Mi 10.04.13	
2	1.1	▢ Projektmanagement	106 Tage	Mi 14.11.12	Mi 10.04.13	
3	1.1.1	Projektstart durchführen	1 Tag	Mi 14.11.12	Mi 14.11.12	
4	1.1.2	Projektkoordination	99 Tage	Do 15.11.12	Di 02.04.13	3
5	1.1.3	Projektcontrolling durchführen	99 Tage	Do 15.11.12	Di 02.04.13	3
6	1.1.4	Projektmarketing durchführen	99 Tage	Do 15.11.12	Di 02.04.13	3
7	1.1.5	Projektabschluss durchführen	6 Tage	Mi 03.04.13	Mi 10.04.13	6
8	1.2	▢ Software	92 Tage	Do 15.11.12	Fr 22.03.13	
9	1.2.1	Struktur implementieren	20 Tage	Do 15.11.12	Mi 12.12.12	
10	1.2.2	Datenanbindung implementieren	20 Tage	Do 13.12.12	Mi 09.01.13	9
11	1.2.3	Broker implementieren	14 Tage	Do 10.01.13	Di 29.01.13	10
12	1.2.4	Controller implementieren	21 Tage	Mi 30.01.13	Mi 27.02.13	11
13	1.3	▢ Website	92 Tage	Do 15.11.12	Fr 22.03.13	
14	1.3.1	Informationsdesign durchführen	10 Tage	Do 15.11.12	Mi 28.11.12	3
15	1.3.2	Graphikdesign durchführen	7 Tage	Do 29.11.12	Fr 07.12.12	14
16	1.3.3	Datenbank einrichten	6 Tage	Do 29.11.12	Do 06.12.12	14
17	1.3.4	Website implementieren	37 Tage	Di 08.01.13	Mi 27.02.13	20
18	1.4	▢ Schnittstellen	70 Tage	Do 13.12.12	Mi 20.03.13	9
19	1.4.1	Datenbank-Software-Schnittstelle implementieren	11 Tage	Do 06.12.12	Do 20.12.12	16
20	1.4.2	Datenbank-Website-Schnittstelle implementieren	22 Tage	Fr 07.12.12	Mo 07.01.13	16
21	1.4.3	Website-Software-Schnittstelle implementieren	11 Tage	Do 28.02.13	Do 14.03.13	12
22	1.5	▢ Testing & Abschluss	18 Tage	Fr 15.03.13	Mi 10.04.13	
23	1.5.1	User-Acceptance-Test durchführen	5 Tage	Fr 15.03.13	Do 21.03.13	21
24	1.5.2	Securitytesting durchführen	5 Tage	Fr 15.03.13	Do 21.03.13	21
25	1.5.3	Unit-Testing durchführen	7 Tage	Fr 22.03.13	So 31.03.13	24
26	1.5.4	User-Testing durchführen	7 Tage	Fr 22.03.13	So 31.03.13	24
27	1.5.5	Fehlerbehebung durchführen	7 Tage	Mo 01.04.13	Di 09.04.13	26
28	1.5.6	Abnahme durchführen	0 Tage	Mi 10.04.13	Mi 10.04.13	27

Abbildung 11.1: Projektstrukturplan

11.2 Balkenplan

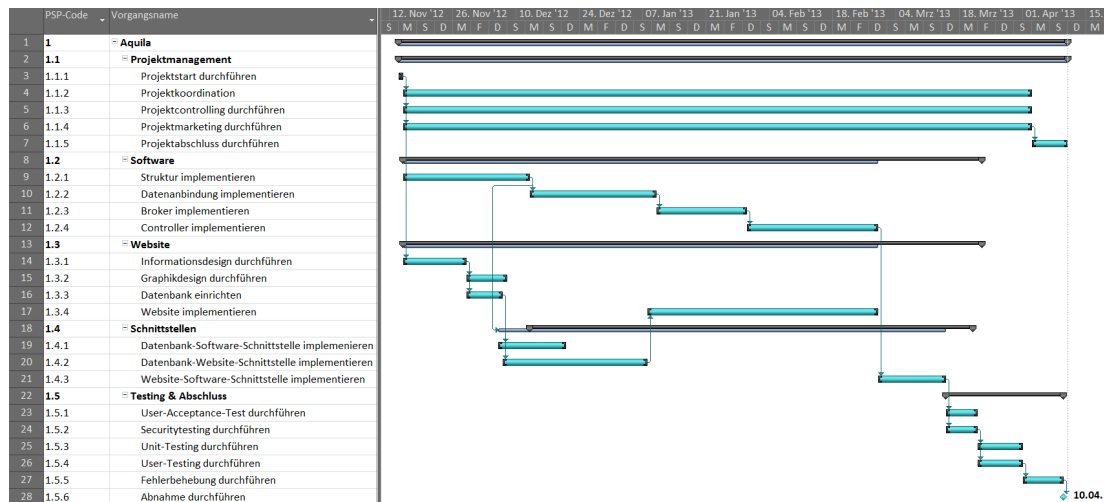


Abbildung 11.2: Balkenplan

11.3 Meilensteinplan

Meilenstein	Deliverable	Datum
Projektstart		14.11.2012
Datenbank eingerichtet	Script zum Erstellen der Datenbank + dazugehörige Dokumentation	06.12.2012
Externe Schnittstellen implementiert	Schnittstellen zur Kommunikation mit eSignal und InteractiveBrokers + dazugehörige Dokumentation	29.01.2013
Software & Website fertiggestellt	Tradingsoftware mit simplem Algorithmus und Websiteoberfläche und Charting + dazugehörige Dokumentation	27.02.2013
Produkt fertiggestellt	Version des Produkts (inklusive aller internen Schnittstellen), die noch nicht getestet wurde + dazugehörige Dokumentation	14.03.2013
Testing abgeschlossen	Testberichte und etwaige Verbesserungen am Produkt	09.04.2013
Projektabschluss	Ausgefülltes Abnahmeprotokoll	10.04.2013

Glossar

IB Interactive Brokers. 5, 6

MA Moving Average. 1, 23

WCF Windows Communication Foundation. 8

