

WI-Praktikum  
Online Börseninformationssystem

# Pflichtenheft

**Team:**

Gäckler, Marco (Teamleiter)  
Abolhassani, Ruhollah  
Lauer, Christian  
Mbongo, Max  
Saissi, Taoufik

## Inhalt

<b>1. Ausgangssituation .....</b>	<b>3</b>
<b>2. Zielsetzung.....</b>	<b>3</b>
<b>3. Systemeinsatz und Systemumgebung .....</b>	<b>3</b>
<b>4. Benutzerschnittstellen .....</b>	<b>4</b>
<b>5. Funktionale Anforderungen.....</b>	<b>5</b>
<b>6. Nichtfunktionale Anforderungen.....</b>	<b>5</b>
<b>7. Dokumentationsanforderungen .....</b>	<b>6</b>
<b>8. Abnahmekriterien .....</b>	<b>6</b>
<b>9. Glossar .....</b>	<b>7</b>
<b>Anhang: (Auszug aus der Datenstruktur/Klassenmodell) .....</b>	<b>8</b>

## 1. Ausgangssituation:

Im Rahmen des WI-Schwerpunktpraktikums an der FH Giessen-Friedberg soll ein Börseninformationssystem erstellt werden. Vorgegeben ist die Entwicklung eines Online-Börseninformationssystems und -buchungssystems in Form einer verteilten Anwendung auf Basis von ASP.NET und .NET-WebseVICES zu realisieren. Die Plattform auf der diese Web Applikation später laufen soll ist ein Microsoft Windows 2000 Server, auf dem IIS bereits als Webserver eingerichtet ist. Als SQL Datenbank wird die Microsoft SQL Server 2000 Desktop Engine (MSDE) verwendet werden. Die Anwendung sollte komplett in Microsofts .NET Umgebung realisiert werden.

## 2. Zielsetzung:

Dem Benutzer soll eine Online - Anwendung zur Verfügung gestellt werden, die es ihm ermöglicht aktuelle Börsenkurse abzufragen, Aktien zu kaufen und zu verkaufen, eine Übersicht seines Depots anzuzeigen sowie bei Bedarf seine Zugangsdaten zu ändern. Der Zugriff auf das Online-Depot soll nur nach einem erfolgreichen Login gestattet sein, während Kursabfragen von Aktien frei von jedem ohne jeglichen Login gestattet werden sollen.

## 3. Systemeinsatz und Systemumgebung:

Die Systemumgebung besteht aus einem MS Windows 2000 Server, auf dem IIS als Webserver, SQL Server 2000 Desktop Engine (MSDE) als SQL Server und das MS.NET Framework als Laufzeitumgebung installiert ist.

Die Webservices sollen auf Basis von ASP.NET und .NET-WebseVICES realisiert werden.

Das Börseninformationssystem besteht im wesentlichen aus folgenden Komponenten (*Abb. 1*):

- Einer Datenbank für die Onlinekunden-/Depot-Verwaltung (*Server DB Depot*)
- Ein *Börsensimulator* zum Testen der Applikation, der die Kommunikation mit der Börse simuliert. Er dient als Lieferant der aktuellen Notierungen und hält auch historische Daten vor. Selbstverständlich muß er auch die Kauf-/Verkaufsaufträge bearbeiten.
- Einer Web-Applikation (*Web Server*) die als ASP.NET Anwendung realisiert werden soll. Sie soll das User-Interface zur Verfügung stellen. Angesprochen werden diese von Web-Clients.
- Eine *MiddleWare* in der die Geschäftslogik des Börseninformationssystems realisiert werden soll. Diese MiddleWare steuert den Austausch zwischen allen Systemmodulen und verarbeitet die Aktionen/Anfragen der einzelnen Systemmodule.

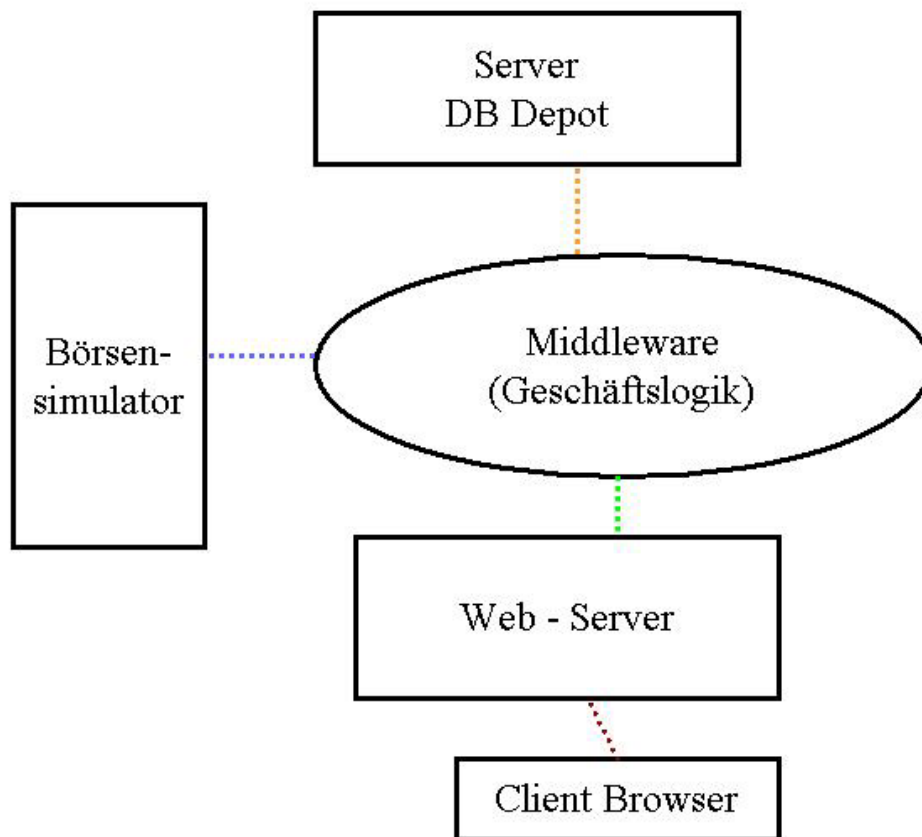


Abb. 1

Jedes der einzelnen Systemmodule läuft auf einem eigenen Rechner/Server. Aufgrund technischer Gegebenheiten im WI-Labor werden allerdings alle vier Module auf einem System installiert werden.

#### 4. Benutzerschnittstellen:

Dem Benutzer, der mit seinem Browser auf unser Dienst zugreift, steht ein Web - Service zur Verfügung. Ohne Login wird der Benutzer aber lediglich Kursinformationen abfragen können. Um ihm den vollen Funktionsumfang anbieten zu können wird ihm ein Login Formular angeboten, mit dessen Hilfe er sich nach einer auf herkömmlichem Wege erfolgten Registrierung einloggen kann. Die Registrierung kann aufgrund juristischer Gegebenheiten nicht online erfolgen. Weitere Funktionalitäten der Benutzerschnittstelle entnehmen Sie bitte dem Abschnitt Middleware (Geschäftslogik) im nächsten Abschnitt. Nachdem Login steht dem Benutzer die volle Funktionalität unseres Dienstes zur Verfügung.

## 5. Funktionale Anforderungen:

Die Funktionalitäten der Module im Einzelnen:

Server Depot DB:

Dieses Modul besteht aus einer Datenbank (MSDE). Diese Datenbank enthält alle Kundendaten (Kundenstammdaten, Depotdaten, Orderbuch, Girokonto).

Börsensimulator:

Der Börsensimulator nimmt Kauf/Verkaufsaufträge entgegen, und liefert Chart bzw. Kursdaten und benachrichtigt nach erfolgreichem/ erfolglosem Kauf/Verkauf. Solche Änderungen sind transaktionsgesichert durchzuführen. Darüberhinaus enthält er eine eigene Datenbank mit (historischen)Kursdaten, evtl. auch eigene Aktienbestände oder Handelsinformationen.



Web Server Applikation:

Diese Applikation soll dynamische HTML Seiten generieren und dem Benutzer die Funktionalitäten des Börseninformationssystem zur Verfügung stellen. Die einzelnen Funktionalitäten werden aber nicht in diesem Modul realisiert. Es erzeugt lediglich ein Benutzerinterface, welches die Aktionen des Benutzers an die Middleware zur Verarbeitung weiterleitet. Der Endanwender nutzt unseren Dienst durch Zugriff mit seinem Browser.

Middelware:

In diesem Modul soll die gesamte Geschäftslogik des Börseninformationssystems realisiert werden. Die Webapplikation soll auf die Funktionalitäten dieses Moduls mittelbar zugreifen können. Zu den Funktionen zählen:

- Einloggen
- Ausloggen
- Login Daten ändern
- Informationen zu den Aktien erfragen(aktueller Kurswert / Chart)
- Kaufauftrag abgeben (durch TAN abgesichert)
- Verkaufsauftrag abgeben (durch TAN abgesichert)
- (Giro-) Konto-Stand abfragen
- Auf (Giro-) Konto einzahlen
- Von (Giro-) Konto abheben
- Depotübersicht anfordern
- Orderbuch anzeigen



## 6. Nichtfunktionale Anforderungen:

Zum Entwickeln und Testen werden PC's mit MS Visual Studio.NET und Internetzugang sowie ein Testserver benötigt. Dazu nutzen wir das Equipement im WI-Labor. Eine eigene Lizenz von Visual Studio .NET pro Teammitglied ist wünschenswert.

## 7. Dokumentationsanforderungen

Zum Abschluß des Projektes wird eine Bedienungsanleitung der Benutzerschnittstelle vorgelegt.

Für die Dokumentation des Quelltextes werden schon bei der Kodierung aussagekräftige Funktionsnamen und Variablenbezeichner gewählt um zusätzliche Programmkommentare dadurch soweit wie möglich zu erübrigen. Bei den Beschreibungen der Funktionen (Funktionsköpfe) soll die Möglichkeit von C# zur automatischen Dokumentationsgenerierung aus Kommentaren heraus genutzt werden.

Für jede öffentliche Methode wird kurz (in Stichworten) ihre Tätigkeit erklärt, wenn nicht selbsterklärend dann auch entspr. die Parameter und "return values".

## 8. Abnahmekriterien

Das Produkt muss den im Pflichtenheft festgelegten Anforderungen an

- Funktion
  - Leistungsfähigkeit
- entsprechen.

Das Produkt muss in der festgelegten Umgebung lauffähig sein.

Das Produkt muss in der vorgeschriebenen Zeit fertiggestellt werden.

## 9. Glossar

Applet	Clientseitig ausgeführtes Java-Programm
Client	Benutzer , der auf den Server zugreift
HTML	Hypertext Markup Language
IIS	Internet Information Server
Link	Verbindung zu einer andern HTML-Seite
Server	Rechner, der die Daten für eine Internetpräsentation bereitstellt
DB	Datenbank
WKN	Wertpapier Kenn-Nummer
MS	Microsoft
SQL	Structured Query Language (eine Datenbankabfragesprache)
MSDE	SQL Server 2000 Desktop Engine
User	Synonym für Benutzer

## Anhang: (Auszug aus der Datenstruktur/Klassenmodell)

