Spezifikation

StockService

Version 1.0

**Gruppe 1**

|  |  |  |
| --- | --- | --- |
| Angermeier Rainer | 1155727 | angermeier.rainer@gmail.com |
| Atzenhofer Thomas | 1155249 | thomas.atzenhofer@gmx.net |
| Dansachmüller Olivia | 1155584 | olivia.d@gmx.at |
| Eckmaier Alexander | 1157492 | alexander.eckmaier@gmail.com |

Kursleiter

Sametinger Johannes

Plösch Reinhold

Weinreich Rainer

Stritzinger Alois

Kurs

259.028 – Service Engineering KV

Datum

**Das Aktuelle Datum der Abgabe!**

Inhalt

[1 Einleitung 3](#_Toc403165320)

[1.1 Zweck und Ziel 3](#_Toc403165321)

[1.2 Name, Einsatzbereich, Abgrenzung 3](#_Toc403165322)

[2 Funktionale Anforderungen 5](#_Toc403165323)

[2.1 Registrierung 5](#_Toc403165324)

[2.2 Login 5](#_Toc403165325)

[2.3 Aktienübersicht 6](#_Toc403165326)

[2.4 Aktienhistorie 6](#_Toc403165327)

[2.5 Aktienkauf 6](#_Toc403165328)

[2.6 Aktienverkauf 6](#_Toc403165329)

[2.7 Passwortänderung 6](#_Toc403165330)

[2.8 Logout 6](#_Toc403165331)

[3 Services 7](#_Toc403165332)

[3.1 Registrierung/Passwortänderung 7](#_Toc403165333)

[3.2 Login 7](#_Toc403165334)

[3.3 Aktienübersicht 8](#_Toc403165335)

[3.4 Aktienhistorie 8](#_Toc403165336)

[3.5 Aktienkauf 8](#_Toc403165337)

[3.6 Aktienverkauf 9](#_Toc403165338)

[3.7 Logout 9](#_Toc403165339)

[4 Sonstiges 10](#_Toc403165340)

# Einleitung

Diese Einleitung soll einen detaillierten Überblick über die Softwareanforderungsspezifikation (SAS) verschaffen. Sie beinhaltet verschiedene Aspekte wie den Zweck und das Ziel der SAS und soll dazu beitragen, verschiedene Begriffe zu beschreiben und etwaige Unklarheiten auszuräumen. Des Weiteren werden in der Einleitung auch der Name des Softwareprodukts und dessen Einsatzbereich festgelegt sowie Abgrenzungen zu anderen Produkten erläutert.

## Zweck und Ziel

Zweck und Ziel dieser SAS ist es dem Auftragnehmer beziehungsweise den Softwareentwicklern in schriftlicher Form die Anforderungen an das Softwareprodukt gut strukturiert, prägnant, verständlich, eindeutig, nachvollziehbar und vollständig darzulegen.

## Name, Einsatzbereich, Abgrenzung

Der zu realisierende Service mit Web-Client wird im Folgenden als StockService bezeichnet und als Server-Client-Applikation realisiert. Aufgabe des Servers ist es, Aktienkurse von zehn an der Börse notierten Firmen zu sammeln, zu speichern und für den Client verfügbar zu machen. Dem User soll es folglich möglich sein, verschiedene Informationen über diese Aktien am Client abzurufen und auf einer Website zu betrachten. Dabei wird zwischen einem Standard-, das heißt nicht eingeloggten, und einem eingeloggten User unterschieden. Ersterem stehen folgende Möglichkeiten zur Verfügung: Er kann sich die Liste mit allen zehn verfügbaren Aktien und den dazugehörigen Basisdaten ansehen und je nach Belieben zusätzliche Details zu einer von diesen einblenden lassen. Diese zusätzlichen Details bestehen aus Informationen über die letzte Zeit bis zum aktuellen Datum sowie über Höchst- bzw. Tiefstände und sonstige Veränderungen. Des Weiteren hat der Standard-User auch noch die Möglichkeit sich zu registrieren und damit die Fähigkeit zu erwerben, zu einem eingeloggten User zu werden. Diesem stehen wiederum zusätzliche Möglichkeiten offen: Er kann sein User-Portfolio, dessen Daten vom Server in einer Datenbank gespeichert und fortlaufend aktualisiert werden, verwalten, das heißt neue bzw. zusätzliche Aktien kaufen, bereits gekaufte Aktien wieder verkaufen und sein Passwort ändern. Zusätzlich kann sich ein eingeloggter User auch ausloggen und damit wieder zu einem Standard-User werden.

Um Aktien zu kaufen benötigt der User Geld. Daraus folgt, dass er nicht beliebig viele Aktien, sofern verfügbar, kaufen kann. Auch kann er sein Konto nicht überziehen. Die Kosten für einen Aktienkauf setzen sich aus dem jeweiligen Tagespreis multipliziert mit der gekauften Stückanzahl zusammen. Das Verkaufen von Aktien führt zu einem Geldzufluss, der sich ebenfalls aus dem jeweiligen Tagespreis multipliziert mit der verkauften Stückanzahl ergibt.

# Funktionale Anforderungen

Im folgenden Abschnitt werden sämtliche funktionale Anforderungen der Server-Client-Applikation übersichtlich dargestellt und im Detail erläutert. Diese decken das gesamte Spektrum an Möglichkeiten für User ab. Insgesamt gibt es acht Hauptanforderungen.

## Registrierung

Die Registrierung steht allen Usern zur Verfügung. Sie kann von der Website aus aufgerufen werden, wobei die dafür benötigten Daten, d.h. Benutzername/Username und Passwort, bereits zuvor in entsprechende Felder eingetragen werden müssen. Hierbei ist es wichtig, dass der gewählte Benutzername noch nicht existiert. Sowohl dieser als auch das Passwort müssen gültig sein. Nach dem erfolgreichen Abschluss der Registrierung ist es nun möglich, sich einzuloggen.

## Login

Der Login steht nur registrierten Benutzern zur Verfügung. Er funktioniert analog zur Registrierung. Ein existierender Username muss mit dem dazugehörigen Passwort korrekt eingegeben werden, um zum entsprechenden Account zu gelangen. Nach erfolgreicher Anmeldung können weiterhin die Liste mit den angebotenen Aktien sowie deren Details eingesehen werden. Zusätzlich bietet sich nun jedoch auch die Option, Aktien zu kaufen. Des Weiteren steht dem Benutzer sein persönliches Portfolio zur Verfügung, in dem sich seine derzeitigen Aktienbestände mit dem jeweiligen Gesamtwert und der im Besitz befindlichen Stückzahl wiederfinden. Von diesen können auch einige oder alle zum jeweiligen Tagespreis verkauft werden. Sämtliche Transaktionen, also alle Käufe und Verkäufe spiegeln sich in der Transaktionstabelle wider. Dort sind das Kurzzeichen, der vollständige Namen, die Anzahl der gekauften/verkauften Aktien, der Kauf-/Verkaufspreis und der Typ (Kauf oder Verkauf) ersichtlich. Zusätzlich stehen dem User auch noch die Möglichkeit der Passwortänderung und des Ausloggens zur Verfügung.

## Aktienübersicht

Die Aktienübersicht steht allen Usern zur Verfügung. Sie besteht aus einer Liste aller verfügbaren Aktien, in diesem Fall zehn. Zu jeder Aktie sind die Abkürzung, der vollständige Name, der aktuelle Preis und die Kursänderung ersichtlich. Diese Informationen werden ohne weiteres Zutun des Benutzers neben der entsprechenden Aktie dargestellt. Die Aktienübersicht wird durch die Aktienhistorie erweitert.

## Aktienhistorie

Die Aktienhistorie steht allen Usern zur Verfügung. Sie wird aufgerufen, indem man über eine bestimmte Aktie fährt. Dies führt dazu, dass seitlich davon ein neues Fenster eingeblendet wird, welches Informationen über die Anzahl der gesamten bzw. derzeit verfügbaren Aktien, das Tageshoch, das Tagestief und die Kursänderung des letzten Monats (vom aktuellen Datum an gerechnet) in Form einer Grafik preisgibt.

## Aktienkauf

[Der Benutzer kann auswählen, wie viele Aktien er kaufen möchte. Dies wird einerseits durch die verfügbare Stückzahl sowie andererseits durch das derzeitige Guthaben beschränkt.]

## Aktienverkauf

## Passwortänderung

## Logout

# Services

Folgend werden alle angebotenen Services erklärt und beschrieben. Insgesamt stehen 7 verschiedene Services zur Verfügung, die gemäß dem REST-Prinzip aufgebaut sind.

## Registrierung/Passwortänderung

Dieser Service bildet die funktionale Anforderung der Registrierung (2.1) und der Passwortänderung (2.7) ab. Er wird mithilfe der HTTP-Methode PUT realisiert.

Der Call hierfür lautet:

* /users/{username}?pw=pw?sessionID=s

Die Responses sehen wie folgt aus:

* success: Objekt: sessionKey, [Transaktion{symbol, name, amount, data, price, type}, [PortfolioStock{symbol, name, amount, balance}]
* error: Objekt: code=6000, message=Generic Error Message
* error: Objekt: code=7000, message=UN already exists
* error: Objekt: code=7001, message=Password not valid
* error: Objekt: code=7001, message=Password not valid
* error: Objekt: code=9000, message=No Session Available

## Login

Dieser Service bildet die funktionale Anforderung des Logins (2.2) ab. Er wird mithilfe der HTTP-Methode POST realisiert.

Der Call hierfür lautet:

* /users/{username}/login?pw=pw

Die Responses sehen wie folgt aus:

* success: Objekt: sessionKey, [Transaktion{symbol, name, amount, data, price, type}, [PortfolioStock{symbol, name, amount, balance}]
* error: Objekt: code=6000, message=Generic Error Message
* error: Objekt: code=7002, message=Wrong Credentials

## Aktienübersicht

Dieser Service bildet die funktionale Anforderung der Aktienübersicht (2.3) ab. Er wird mithilfe der HTTP-Methode GET realisiert.

Der Call hierfür lautet:

* /finance/stocks

Die Responses sehen wie folgt aus:

* success: Liste mit Stock Objekten: symbol, name, volume, lastTradedpriceOnly, daysHigh, daysLow, change
* error: Objekt: code=6000, message=Generic Error Message

## Aktienhistorie

Dieser Service bildet die funktionale Anforderung des Logins (2.4) ab. Er wird mithilfe der HTTP-Methode GET realisiert.

Der Call hierfür lautet:

* /finance/stocks/{stocksymbol}/history

Die Responses sehen wie folgt aus:

* success: Liste mit StockHistory Objekten: symbol, date, close
* error: Objekt: code=6000, message=Generic Error Message

## Aktienkauf

Dieser Service bildet die funktionale Anforderung des Logins (2.5) ab. Er wird mithilfe der HTTP-Methode PUT realisiert.

Der Call hierfür lautet:

* /finance/stocks/{stocksymbol}?amount=20?sessionID=s

Die Responses sehen wie folgt aus:

* success: Objekt: [Transaktion{symbol, name, amount, data, price, type}, [PortfolioStock{symbol, name, amount, balance}]
* error: Objekt: code=6000, message=Generic Error Message
* error: Objekt: code=8000, message=Not enough money
* error: Objekt: code=8001, message=Not enough stocks available
* error: Objekt: code=9000, message=No Session Available

## Aktienverkauf

Dieser Service bildet die funktionale Anforderung des Logins (2.6) ab. Er wird mithilfe der HTTP-Methode DELETE realisiert.

Der Call hierfür lautet:

* /finance/stocks/{stocksymbol}?amount=20?sessionID=s

Die Responses sehen wie folgt aus:

* success: Objekt: [Transaktion{symbol, name, amount, data, price, type}, [PortfolioStock{symbol, name, amount, balance}]
* error: Objekt: code=6000, message=Generic Error Message
* error: Objekt: code=8001, message=Not enough stocks available
* error: Objekt: code=9000, message=No Session Available

## Logout

Dieser Service bildet die funktionale Anforderung des Logins (2.8) ab. Er wird mithilfe der HTTP-Methode POST realisiert.

Der Call hierfür lautet:

* /users/logout?sessionID=s

Die Responses sehen wie folgt aus:

* success: Kein Objekt
* error: Objekt: code=6000, message=Generic Error Message
* error: Objekt: code=9000, message=No Session Available

# Sonstiges