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](#_Toc403333475)

[1.1 Zweck und Ziel 3](#_Toc403333476)

[1.2 Name, Einsatzbereich, Abgrenzung 3](#_Toc403333477)

[2 Technologieauswahl 4](#_Toc403333478)

[2.1 Server 4](#_Toc403333479)

[2.2 Client 4](#_Toc403333480)

[2.3 Datenformate 5](#_Toc403333481)

[2.4 Protokolle 5](#_Toc403333482)

[3 Funktionale Anforderungen 6](#_Toc403333483)

[3.1 Aktienübersicht 6](#_Toc403333484)

[3.2 Aktienhistorie 6](#_Toc403333485)

[3.3 Registrierung 6](#_Toc403333486)

[3.4 Login 7](#_Toc403333487)

[3.5 Aktienkauf 7](#_Toc403333488)

[3.6 Aktienverkauf 7](#_Toc403333489)

[3.7 Transaktionsübersicht (Portfolio) 8](#_Toc403333490)

[3.8 Passwortänderung 8](#_Toc403333491)

[3.9 Logout 8](#_Toc403333492)

[4 Services 9](#_Toc403333493)

[4.1 Aktienübersicht 9](#_Toc403333494)

[4.2 Aktienhistorie 9](#_Toc403333495)

[4.3 Registrierung 10](#_Toc403333496)

[4.4 Passwortänderung 10](#_Toc403333497)

[4.5 Transaktionsübersicht (Portfolio) 11](#_Toc403333498)

[4.6 Transaktion hinzufügen (Aktienkauf, Aktienverkauf) 11](#_Toc403333499)

[5 Datenbank 12](#_Toc403333500)

[6 Sonstiges 13](#_Toc403333501)

# 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.

# Technologieauswahl

Dieser Teil der Arbeit beschäftigt sich mit der von uns ausgewählten Technologien, Frameworks, Datenformate und Protokolle. Es wird jeweils die verwendete Technologie genannt und kurz vorgestellt.

## Server

Der Server basiert auf dem Spring Framework. Das Springframework ist ein open-source Framework für Java-Plattformen, das die Entwicklung von Java Enterprise Applikationen erleichtern und unterstützen soll. Die großen Vorteile dieses Frameworks liegen nicht nur in der Möglichkeit der Aspektorientierten Programmierung und des Dependency Injection, sondern auch vor allem im erleichterten Entwicklung von RESTful Applikationen, was ausschlaggebend für die Auswahl dieses Framrworks war. Das Springframework ermöglicht durch Annotationen ein effizientes und dynamisches Serververhalten abzubilden.

## Client

Am Client bedienten wir uns ebenfalls eines open-source Frameworks, dem von Google entwickelten AngularJS Frameworks. Auf Javascript basierend, ermöglicht AngularJS die Erstellung von HTML und JavaScript Webanwendungen nach einem MVC Muster. AngularJS ermöglicht nicht nur wie z.B. in JQuery die Manipulation von DOM Knoten in einem Dokument, sondern die komplette Erstellung eines DOM Baumes zur Laufzeit. Dies geschieht durch ein clientseitiges Model, das die gesamte Logik enthält. Dieses Model wird in einem Controller definiert und diese Controller zu einem Modul zusammengefasst, welches später in die HTML Datei eingebunden wird. Außerdem ermöglicht AngularJS die Erstellung von Direktiven. Direktive sind sozusagen selbsterstellte HTML Tags, die es ermöglichen Funktionalität bzw. Code auszulagern und somit die Applikation übersichtlicher zu gestalten. Ein wesentlicher Grund für die Auswahl dieser Technologie war aber die Unterstützung des Frameworks bei REST Schnittstellen und der als Standard definierte Umgang mit dem JSON Format.

## Datenformate

Als Standard für den Datenaustausch zwischen dem Server und dem Client haben wir das JSON Format gewählt. Weiter kommuniziert unser Server mit dem YAHOO-Finance Server, der aktuelle und vergangene Kurswerte ausgewählter Aktien ebenfalls im JSON Format an den Server übermittelt. Abgefragt werden die Daten mit einer YAHOO-spezifischen Query Language, der Yahoo-Query-Language (YQL).

## Protokolle

Grundlegend für die Auswahl der Frameworks und Datenformate war die Auswahl der Protokolle. Die Auswahl der Protokolle fiel auf das HTTP bzw. das REST Protokoll, wo wir unter Verwendung der GET, POST und DELETE Request-Methoden die Kommunikation zwischen Server und Client sicherstellen.

# 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 neun Hauptanforderungen.

## Aktienübersicht

Die Aktienübersicht steht allen Usern zur Verfügung. Sie besteht aus einer Liste aller verfügbaren Aktien, in diesem Fall zehn. Zu jeder Aktie sind der vollständige Name 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 eine bestimmte Aktie auswählt. Dies führt dazu, dass seitlich davon ein neues Fenster eingeblendet wird, welches Informationen über das Aktienkürzel, den vollständigen Namen, den aktuellen Preis, 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 eines Graphen preisgibt.

## 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 aller Transaktionen 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. Zusätzlich stehen dem User auch noch die Möglichkeit der Passwortänderung und des Ausloggens zur Verfügung.

## Aktienkauf

Der Aktienkauf steht nur angemeldeten Benutzern zur Verfügung. Es wird eine Liste aller angebotenen Aktien sowie deren Details angezeigt. Zusätzlich bietet sich nun jedoch auch die Option, Aktien zu kaufen. Dafür wird, wie bereits erwähnt, ein Guthaben benötigt. Der Benutzer kann auswählen, wie viele Aktien er kaufen möchte. Dies wird einerseits durch die verfügbare Stückzahl sowie andererseits durch das derzeitige Guthaben beschränkt. Im Portfolio des Benutzers finden sich seine derzeitigen Aktienbestände mit dem jeweiligen Gesamtwert und der im Besitz befindlichen Stückzahl wieder.

## Aktienverkauf

Der Aktienverkauf steht nur angemeldeten Benutzern zur Verfügung. Es wird eine Liste der vom Benutzer erworbenen Aktien angezeigt. Von dieser können hier wieder 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.

## Transaktionsübersicht (Portfolio)

Die Transaktionsübersicht steht nur angemeldeten Benutzern zur Verfügung. Hier bekommt der User einen Überblick über alle abgeschlossenen Transaktionen, d.h. über seine Aktienkäufe und Aktienverkäufe. Jede Transaktion beinhaltet das Kürzel, den vollständigen Namen, die Anzahl der gekauften/verkauften Aktien, den Kauf-/Verkaufspreis und der Typ (Kauf oder Verkauf). Die Transaktionen werden nach Aktienname/Kürzel gruppiert und erhalten die gesamte aktuelle Aktienanzahl pro Aktie als Attribut.

## Passwortänderung

Die Passwortänderung steht nur angemeldeten Benutzern zur Verfügung. Hier hat der User die Möglichkeit eine Passwortänderung durchzuführen, die mit dem Speichern-Button abgeschlossen wird. Eine Änderung des Benutzernamens ist jedoch nicht möglich.

## Logout

Der Logout steht nur angemeldeten Benutzern zur Verfügung. Um in seine persönlichen Daten wieder einsehen zu können, muss sich ein Benutzer nach erfolgreichem Ausloggen wieder neu einloggen.

# Services

Folgend werden alle angebotenen Services erklärt und beschrieben. Insgesamt stehen 6 verschiedene Services zur Verfügung, die gemäß dem REST-Prinzip aufgebaut sind.

## Aktienübersicht

Dieser Service bildet die funktionale Anforderung der Aktienübersicht (2.1) ab. Er wird mithilfe der HTTP-Methode GET realisiert und liefert eine Liste der verfügbaren Aktien inklusive des Kürzels, des vollständigen Namens, der Aktienanzahl, des aktuellen Preises, des aktuellen Höchst- bzw. Tiefstands und der Kursänderung.

Der Call hierfür lautet:

* /finance/stocks

Die Responses sehen wie folgt aus:

* success: Liste mit Stock Objekten: symbol, name, volume, lastTradedpriceOnly, daysHigh, daysLow, change

## Aktienhistorie

Dieser Service bildet die funktionale Anforderung des Logins (2.2) ab. Er wird mithilfe der HTTP-Methode GET realisiert und liefert einen Graphen mit der betreffenden Aktienentwicklung des letzten Monats zurück.

Der Call hierfür lautet:

* /finance/stocks/{stocksymbol}/history

Die Responses sehen wie folgt aus:

* success: Liste mit StockHistory Objekten: symbol, date, close

## Registrierung

Dieser Service bildet die funktionale Anforderung der Registrierung (2.3) ab. Er wird mithilfe der HTTP-Methode POST realisiert und legt den neuen User in der Datenbank an.

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}]

## Passwortänderung

Dieser Service bildet die funktionale Anforderung der Passwortänderung (2.8) ab. Er wird mithilfe der HTTP-Methode PUT realisiert und ersetzt das vorhandene Passwort durch das neue. Des Weiteren befindet sich dieser Service unter /secured und muss daher beim Call einen HTTP-Basic Authentication Header mitschicken.

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}]

## Transaktionsübersicht (Portfolio)

Dieser Service bildet die funktionale Anforderung der Transaktionsübersicht (2.7) ab. Er wird mithilfe der HTTP-Methode GET realisiert und liefert ein Portfolio aller Transaktionen gruppiert nach Aktie. Des Weiteren befindet sich dieser Service unter /secured und muss daher beim Call einen HTTP-Basic Authentication Header mitschicken.

Der Call hierfür lautet:

* /secured/users/{username}/transactions

Die Responses sehen wie folgt aus:

* success: Objekt: [Transaktion{symbol, name, amount, date, price, type}, [PortfolioStock{symbol, name, amount, balance}]

## Transaktion hinzufügen (Aktienkauf, Aktienverkauf)

Dieser Service bildet die funktionale Anforderung des Aktienkaufs (2.5) und des Aktienverkaufs (2.6) ab. Er wird mithilfe der HTTP-Methode PUT realisiert und fügt Transaktionen zur Transaktionsübersicht hinzu. Des Weiteren befindet sich dieser Service unter /secured und muss daher beim Call einen HTTP-Basic Authentication Header mitschicken.

Der Call hierfür lautet:

* /secured/finance/transactions?stocksymbol=iwie&amount=20&isSell=true/false

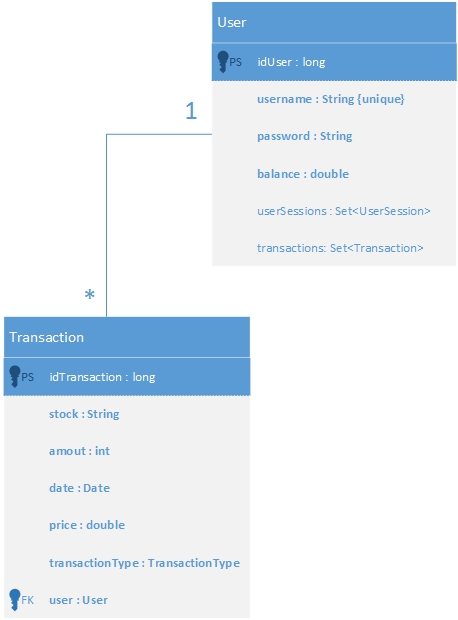
Die Responses sehen wie folgt aus:

* success: Objekt: [Transaktion{symbol, name, amount, date, price, type}, [PortfolioStock{symbol, name, amount, balance}]

Der Login (2.4) und der Logout (2.9) stellen keine Services dar, sondern warden am Client durch Cookies verwaltet.

# Datenbank

Im Anschluss wird das Datenbank-Schema dargestellt. Es umfasst drei Tabellen und damit den User, die Transaktion und die User Session.



# Sonstiges