

Hype – Einfache Analyse von Aktien

Beschreibung der Webseite

**Abgabe für das Modul Entwicklung von betrieblichen Informationssystemen**

eingereicht im: Januar 2020

von (Matrickelnummer): Hagen Sostak (2728018)

Yorrick Reinhart (2706676)

Patrick Kurz (2727571)

Johannes Gutenberg-Universität Mainz

Fachbereich Rechts- und Wirtschaftswissenschaften

Lehrstuhl für Wirtschaftsinformatik und BWL

Telefon: +49 6131 39-22734, Fax +49 6131 39-22185

Internet: <http://wi.bwl.uni-mainz.de>

Inhaltsverzeichnis

[Abbildungsverzeichnis iii](#_Toc30873227)

[1 Einleitung 1](#_Toc30873228)

[2 Aufbau der Datenbank 1](#_Toc30873229)

[3 Funktionalitäten 3](#_Toc30873230)

[3.1 Accountmanagement 3](#_Toc30873231)

[3.2 Analyseseite 3](#_Toc30873232)

[3.2.1 Suchleiste 3](#_Toc30873233)

[3.2.2 Generieren der Finanzdaten 4](#_Toc30873234)

[3.2.3 Anzeigen von Graphen und bewertenden Ampeln 4](#_Toc30873235)

[3.3 Forum 5](#_Toc30873236)

[3.4 Chatbot 5](#_Toc30873237)

[3.5 Responsive Design 6](#_Toc30873238)

[4 Ablauf einer Aktienabfrage 6](#_Toc30873239)

[5 Probleme 7](#_Toc30873240)

[5.1 Datenquelle 7](#_Toc30873241)

[5.2 Infrastruktur 8](#_Toc30873242)

[6 Vorkenntnisse 9](#_Toc30873243)

[7 Ausblick 10](#_Toc30873244)

[Literaturverzeichnis ix](#_Toc30873245)

Abbildungsverzeichnis

[Abbildung 1: ERR-Modell Usermanagement und Forum. 2](#_Toc8819316)

[Abbildung 2: ERR-Modell Aktien. 2](#_Toc8819316)

# Einleitung

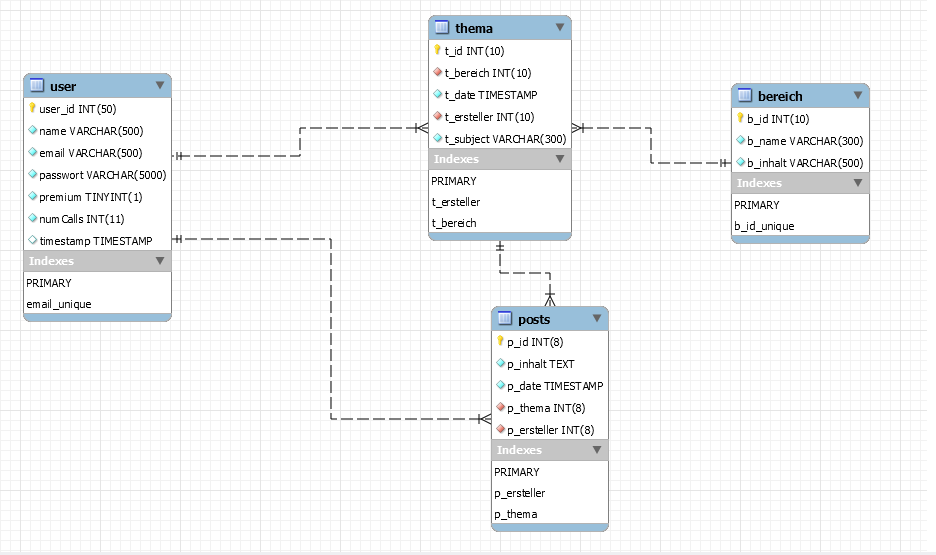
In Deutschland liegt die Aktionärsquote bei ca. 6%. Dies ist im internationalen Vergleich z.B. zu den USA (25%) oder Großbritannien (23%) deutlich geringer (Hasler, 2018, Abb.5). Die Gründe dafür sind vielseitig, zum einen gilt Deutschland als risikoaverse Nation, zum anderen sorgten die stabil funktionierenden Pensionssysteme dafür, dass Aktien nicht so relevant für die Altersvorsorge sind wie beispielsweise in den USA (Kaya, 2018, S.1). Die über die Jahre steigende Performance des Dax aber auch die Angst des Einbrechens der Rente überzeugt mittlerweile immer sich mit Aktien zu beschäftigen und zu Investieren.

Mit dem Blick auf eine stabile Rente verwenden wir eine dividendenbasierte Strategie, gelegentlich auch als Einkommensstrategie bezeichnet, mit dem Ziel Aktien zu identifizieren, die besonders stabil und risikoavers ihre Dividenden auszahlen (Schacht & Fackler; 2009). In unserem Tool HYPE nutzen wir die wichtigsten Kennzahlen und bilden diese besonders einsteigerfreundlich mittels „Ampelsystem“ ab. Dabei geben wir noch kleine Hinweise zu den verschiedenen Kennzahlen, sowie die Entscheidungsgrößen, warum die Ampel nun grün, gelb oder rot zeigt. Ziel ist es die Hemmungen der Deutschen gegenüber der Börse zu reduzieren, indem Kennzahlen vereinfacht und erläutert werden und ausschließlich risikoarm bewertete Aktien als „gut“ präsentiert werden. Die Zielgruppe sind Börsen-Neulinge, weshalb unsere Software im Zweifel eine Aktie eher als riskant einstuft, das Erkennen von Chancen wird dem Nutzer unter seiner eigenen Verantwortung überlassen.

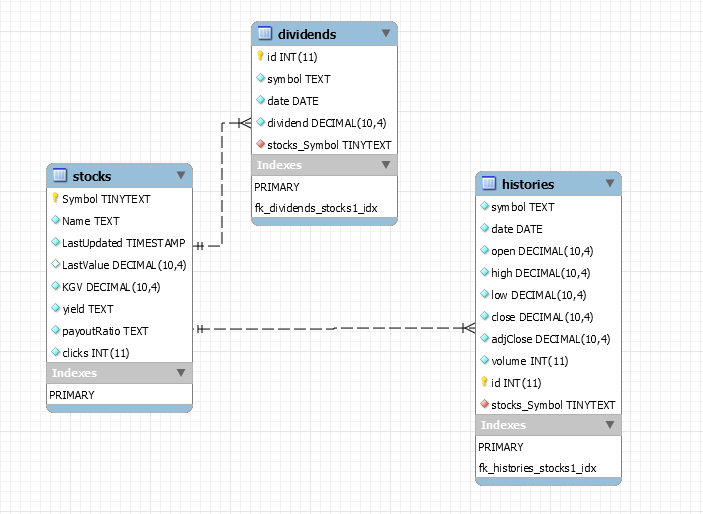
Die Webseite stellen wir als Freemium Modell bereit. Bei kostenloser Nutzung sind zwar alle derzeitigen Funktionalitäten gegeben, jedoch ist es nur möglich maximal 5 Aktien am Tag zu betrachten. Bei einem Upgrade auf die Premium Version, als Abo Modell, können beliebig viele Aktien betrachtet werden und es steht ein persönlicher Support zur Verfügung.

# Aufbau der Datenbank

Für unsere Webseite verwenden wir sieben Tabellen, wovon vier für Usermanagement und das Forum genutzt werden. Die restlichen drei Tabellen werden für das Speichern der aktienbezogenen Daten benötigt. Aus diesem Grund sind diese Tabellen nicht direkt mit dem user verknüpft. Ihre Beziehungen untereinander haben wir mithilfe eines ERR Modells dargestellt. (erstellt mit SQL Workbench). Siehe dazu Abbildung 1 und 2.



(Abb. 1 ERR-Modell Usermanagement und Forum)



(Abb. 2 ERR-Modell Aktien)

# Funktionalitäten

Im Folgenden gehen wir auf die wichtigsten Funktionen unserer Webseite und deren Implementierung genauer ein. Technisch uninteressante Funktionalitäten, wie beispielsweise die Registrierung werden verkürzt abgehandelt. Zur Übersichtlichkeit stellen wir genutzte Tabellen aus der Datenbank im Fließtext kursiv dar (z.B. user).

Die Zuständigkeiten der Funktionen nennen wir jeweils am Ende einer Funktionalität. An dieser Stelle möchten wir jedoch hervorheben, dass viel gemeinsam (über github) programmiert wurde und sich daher viele Funktionen nicht klar trennen lassen.

## Accountmanagement

Das Accountmanagement umfasst das Registrieren eines neuen Nutzers, sowie die Möglichkeit zur Anmeldung bestehender Nutzer. Außerdem können angemeldete Nutzer ihre Zugangsdaten ändern und die Premiumfunktion (ab-)bestellen. Des Weitern sind die wichtigen Seiten, wie die Analysen und das Forum, vom Zugriff durch nicht angemeldete Nutzer geschützt.

Mit Hilfe von HTML Formularen werden mittels POST, Daten an die Datenbank übermittelt und bei Gültigkeit in user eingetragen, bzw. bei der Anmeldung mit user abgeglichen. Der Schutz vor unbefugtem Zugriff auf oben genannte Seiten ist mittels Sessions und ggf. einer Weiterleitung umgesetzt.

Hauptzuständiger: Hagen

## Analyseseite

Innerhalb der Analyseseite sind mehrere Funktionalitäten zusammengefasst, die wir im Folgenden, für eine bessere Übersichtlichkeit, separat betrachten.

### Suchleiste

Die Suchleiste bildet den Ausgangspunkt der Aktienanalyse. In ihr kann man nach dem Namen einer Aktie suchen, wobei per Autovervollständigung passende und vollständige Namen angezeigt werden. Die Autovervollständigung funktioniert derzeit für alle Aktien des Dax und einige bekannte Unternehmen weltweit (z.B. Apple). Es kann jedoch für jede beliebige Aktie die Analyse gestartet werden, wenn das Symbol (nach Yahoo Finance Standard) eingetragen wird.

Für die technische Umsetzung haben wir das Framework JQueryUI genutzt, welches bereits eine „autocomplete“-Funktion mitliefert. Diese nutzt unser dictionary (stocks.js) in der wir „label“ (Name der Aktie), sowie „values“ (die Symbole) definieren.

Alternativ zur Suchleiste, kann man auch auf die Top-5 gesuchten Aktien zugreifen. Auf eine genauere Erläuterung verzichten wir an dieser Stelle, da eine einfache SQL-Abfrage an stocks gesendet wird, welche prüft, welche Aktie die meisten Aufrufe generiert hat.

Hauptzuständiger: Patrick

### Generieren der Finanzdaten

Um die Analyseseite für ein übergebenes Symbol zu laden, wird zunächst überprüft, ob die Aktie bereits vor weniger als 24h in stocks geladen wurde, da für unseren Zweck eine Aktualität von 1-2 Tage vollkommen ausreicht. Ist die Aktie bereits vorhanden, kann direkt das Frontend geladen werden, wenn nicht müssen die Daten erst von Yahoo Finance abgerufen werden.

Auf technischer Ebene ist diese Funktion in zwei Hauptfunktionen zu unterteilen, den Scraper und die cURL-Abfrage. Zunächst werden im Scraper mittels simple\_html\_dom-Framework zwei verschiedene Yahoo Finance Seiten gescraped. Dies bereitete uns einige Probleme, da Yahoo sehr dynamisch aufgebaut ist. So muss man einerseits gewisse Kennzahlen mittels html-Attribut finden und auslesen. Andererseits muss per Schleifendurchlauf nach bestimmten Strings gesucht werden, denen dann die für uns relevante Kennzahl folgt. Zusätzlich wird eine cURL-Abfrage benötigt, welche per POST-Request die csv-Dateien der Aktien- und Dividendenhistorie laden. An dieser Stelle ist eine cURL-Abfrage nötig, da Yahoo mit dem Einstellen ihrer API einen crumb (Cookie) setzt, welcher vorhanden sein muss, um csv-Dateien herunter zu laden. Den Crumb kann man jedoch mit Hilfe von cURL umgehen.

Hauptzuständiger: Yorrick, Patrick für den Scraper

### Anzeigen von Graphen und bewertenden Ampeln

Auf unserer Seite Analysen können alle Aktien, die bei Yahoo Finance gelistet sind, angezeigt werden. Dabei stellen wir den Aktienkursverlauf und die Dividendenauszahlungshöhe und -rate als Graphen dar. Unterhalb dieser zwei Graphen, sind diverse Kennzahlen in einer Tabelle aufgelistet, welche das Interpretieren der Aktie übernehmen. Dabei besteht eine Reihe immer aus dem Namen der Kennzahl mit einem Infobutton für weitere Informationen, den kritischen Werten (wann die Ampel grün, gelb oder rot ist), dem aktuellen Wert für die Aktie und der Ampel selbst.

Um die Daten für eine Aktie auf der Webseite anzuzeigen, nutzen wir neben html CanvasJS (für die Graphen) auch JavaScript / JQuery (für Allgemeines). Beim Aufrufen der Analyse Seite für ein gegebenes Symbol wird zuerst geprüft ob die Aktie bereits in der Datenbank ist, und falls ja, ob das letzte Herunterladen weniger als 24 Stunden her ist. Falls dies zutrifft, werden alle Daten aus der Datenbank geladen. Falls die Daten noch nicht in der Datenbank oder älter als 24 Stunden sind, werden sie von Yahoo herunter und in unsere Datenbank geladen.

Um die Graphen mit CanvasJS anzuzeigen, laden wir die gespeicherten Daten als Array aus der Datenbank und hinterlegen die x-Werte (Timestamp) und y-Werte (Aktienkurs bzw. Dividendenhöhe) als Datengrundlage. Mittels CanvasJS können diese Daten dann als Line-Chart bzw. als Balkendiagramm dargestellt werden. Für die Tabelle der Kennzahlen überprüfen wir zuerst, ob die Kennzahl der Aktie in einem bestimmten Wertebereich liegt (hardcoded für jede Kennzahl). Je nach dem in welchen Bereich die Kennzahl hineinfällt, setzten wir die Bildquelle unterschiedlich, um eine grüne, gelbe oder rote Ampel anzeigen zu können.

Hauptzuständiger: Patrick

## Forum

Das Forum ist einfach und intuitiv gehalten. Man kann Kategorien erstellen, in diesen Themen posten und zu verschiedenen Themen Beiträge verfassen. Dies ermöglicht einen einfachen Wissensaustausch der Community auf einer Plattform.

Auf die technische Umsetzung muss nicht weiter eingegangen werden, da alles mithilfe von html-Formularen, Tabellen sowie SQL-Statements umgesetzt ist. Besonderheiten gibt es hier keine.

Hauptzuständiger: Hagen

## Chatbot

Der Chatbot unserer Seite soll die Navigation vereinfachen und die generelle Userexperience verbessern. Es ist beabsichtigt, insbesondere weniger computeraffine Personen an die Hand zu nehmen und über die Webseite zu begleiten. Bisher hat dieser einige grundlegende Funktionen, wie das Anzeigen der meistgesuchten Aktien oder das Navigieren zu den Analysen innerhalb des Hauptfensters. Hier ist jedoch herauszustellen, dass dies nur eine initiale Beladung von Daten und Funktionen darstellt und in der Zukunft noch ausgebaut werden kann. Für eingeloggte und nicht-eingeloggte Nutzer unterscheiden sich die vom Chatbot angebotenen Funktionen.

Zur Umsetzung nutzen wir das auf JavaScript basierende Framework BotUI. Das Framework ist so aufgebaut, dass man eine BotUI Instanz startet, die mit einem div-Continer verknüpft ist. Die BotUI Instanz besitzt dann verschiedene Funktionen wie „message“, um Inhalte auszugeben oder „action“, um Nutzerabfragen zu gestalten. Grundsätzlich agiert unser (deterministischer) Chatbot auf Eingaben des Nutzers. So wird beispielsweise nach dem Namen des Nutzers per input-Feld gefragt und das Ergebnis im Folgenden Begrüßungssatz genutzt. Auf diese Art und Weise funktionieren auch die input-Buttons, wobei bei diesen das Ergebnis durch eine Reihe von if-Abfragen läuft, um zu determinieren was passieren muss. Eine interessante Funktion versteckt sich noch hinter dem Weiterleiten. Dadurch, dass wir beim Öffnen des Chatbots das aktuelle Fenster (parent) als Variable mitgeben, können wir zum Beispiel, wenn auf eine der Top5 Aktien gedrückt wurde, diese im Hauptfenster (und nicht innerhalb des Chatbots) öffnen.

Hauptzuständiger: Patrick

## Responsive Design

Unsere Webseite ist durchweg responsive, das bedeutet, dass sie auf Bildschirmen mit kleinerer Auflösung trotzdem noch gut angezeigt werden kann. Besonders im Zeitalter der hand-held-devices ist dies essenziell.

Auf technischer Seite geschieht die Umsetzung des responsive Designs mittels des weit verbreiteten Framework Bootstrap 4.0. Dieses Unterteilt das Layout einer Webseite in beliebig viele Reihen und 12 Spalten. Mit Hilfe von CSS-Klassenbezeichnungen greift man auf dieses Framework zu und bestimmt, ab welcher Bildschirmgröße eine Spalte in eine neue Reihe umgebrochen werden soll. Auch wenn es viele fertige Templates basierend auf Bootstrap gibt, haben wir uns für unser eigenes Layout entschieden.

Hauptzuständiger: Hagen

# Ablauf einer Aktienabfrage

Die zentrale Funktion unseres Webservices ist das Anzeigen von Aktieninformationen, die durch den Nutzer abgefragt werden. Dieses Kapitel beschreibt den möglichen Ablauf, sobald der Nutzer eine Anfrage stellt.

Der Nutzer gibt im Suchfeld das gesuchte Symbol ein und kann bei Bedarf einen Eintrag aus der Suchliste wählen, dessen Symbol dann in das Textfeld geladen wird. Jeder eingegebene Text, bzw. Symbol wird automatisch in Großbuchstaben umgewandelt, da das Symbol in der Stocks-Table als Primärschlüssel verwendet wird und so garantiert wird, dass dieselbe Aktie nicht wegen unterschiedlicher Groß- und Kleinschreibung mehrmals in der Stocks-Table, und somit in allen anderen darauf bezogenen Aktien eingetragen wird.

Das Symbol wird nun an die Funktion updateDB übergeben. An dieser Stelle öffnen sich mehrere Zweige im Programm.

Zuerst wird geprüft, ob das vom Nutzer eingegebene Symbol in der Datenbank ist oder nicht. Falls es noch nicht in der Datenbank ist wird der Scraper aufgerufen, der alle betreffenden Yahoo-Seiten aufruft und die gesuchten Werte von der Webseite herunterlädt.

Stellt der Scraper fest, dass er keinen Opening Value auf der Yahoo-Webseite finden kann, so schließen wir daraus, dass die Aktie nicht existiert (da jede handelbare Aktie schließlich einen Eröffnungswert haben sollte), brechen das Programm ab und liefern dem Nutzer eine Fehlermeldung aus.

Existiert die passende Seite bei Yahoo, so werden der Eröffnungswert, die erwartete Dividendenrendite, KGV und die Pay-Out-Ratio ausgelesen und mit einem Timestamp und einem Klickzähler gleich null in die Datenbank geladen. Der Klickzähler dient zur bestimmung der meist gesuchten Aktien. Danach werden mittels cURL die zwei CSVs der historischen Daten und der Dividenden heruntergeladen, in SQL-Befehle konvertiert, und in die Tables „Histories“ und „Dividends“ eingetragen.

Existiert das Symbol bereits in der Stocks-Table, so wird geprüft ob der Timestamp älter als 24 Stunden ist. Falls dies zutrifft startet der Scraper mit dem oben erklärten Verhalten erneut und die korrespondierenden Einträge mit den neuen Einträgen des Scrapers in der Datenbank geupdated. Die vorhandenen Historien- und Dividendeneinträge des Symbols werden erst gelöscht, dann werden die beiden CSVs heruntergeladen und die Werte in die beiden entsprechenden Tables eingetragen. Zum Schluss wird der Timestamp auf die aktuelle Uhrzeit aktualisiert.

Sollte der Timestamp jünger als 24 Stunden sein, so wird nur der letzte Opening Value gescraped.

Zum Schluss wird der Klickzähler der Aktie in der Datenbank inkrementiert, wodurch die Zweige des Programmes wieder zusammengeführt werden.

Nun lädt das Frontend die allgemeinen Daten und Datenpunkte der Historien und der Dividenden aus der Datenbank um diese in den Graphen und Labels anzuzeigen, startet die Berechnungen der für die Ampeln notwendigen Daten, prüft zwischen welchen Grenzwerten die Werte liegen, lädt die korrekt gefärbten Ampeln und zeigt die Seite dem Nutzer an.

# Probleme

## Datenquelle

Ein zentrales Problem war die Wahl der Datenquelle und unsere Abhängigkeit von dieser.

Da viele kommerzielle Datenquellen hinter hohen Bezahlschranken liegen, versuchten wir zuerst Alpha Vantage zu nutzen, da dieses einen kostenlosen Zugang anbietet. Allerdings sind bei diesem Zugang nur fünf Anfragen pro Minute möglich, wodurch unsere Webseite selbst bei moderater Auslastung Probleme bekommen würde. Außerdem würden das Entwickeln und Debuggen dadurch sehr gebremst. Um das zu umgehen nutzen wir den Scraper um unsere Daten von Yahoo auszulesen, was Aufgrund des dynamischen Seitenaufbaus nicht trivial war.

Sehr problematisch ist für uns die Abhängigkeit von Yahoo, da wir nicht selbst für Vollständigkeit und Korrektheit der dargestellten Daten garantieren können. So gab es Aktien, bei denen die von Yahoo angebotenen CSVs fehlerhaft waren, da sie wenige oder nur einen einzelnen Datenpunkt enthielten. Ebenso tritt gelegentlich das Phänomen auf, das Yahoo bei Aktien die eigentlich Dividenden zahlen, trotzdem keine Werte, sondern „N/A“ anzeigt. Diese fehlerhaften Daten werden dann ebenso fehlerhaft von uns dargestellt und führen zu einer schlechteren Nutzererfahrung.

## Infrastruktur

Ein wiederkehrendes Problem für uns war der limitierte Zugang zur Infrastruktur, auf der wir deployen sollten.

So konnten wir zu Beginn kein cURL nutzen, da dies auf den Servern der Universität nicht freigeschaltet war. Glücklicherweise wurde uns dies freigeschaltet, andernfalls wissen wir nicht, wie und wo wir die historischen Daten sonst ohne weitere Kosten hätten finden können und möglicherweise uns ein anderes Projekt überlegen müssen.

Ebenfalls bereitete uns die Datenbank Probleme. Wir entwickelten lokal und nutzten die in XAMPP enthaltene MySQL-Datenbank. Diese scheint mehr Features zu unterstützen als die in der Universität verwendete Version. Zum Beispiel wurden Default-Werte der Tables beim Import der Datenbankkonfiguration nicht übernommen. Dadurch funktionierten unsere IMPORT-Statements nicht mehr, da wir uns auf die Einstellungen der Tables verließen. Allerdings erhielten wir darüber keine Fehlermeldungen, unsere Statements wurden schlicht nicht ausgeführt, das restliche Programm lief aber weiter. Da wir den Fehler aber lokal nicht reproduzieren konnten, verging viel Zeit bis wir den Fehler an dieser Stelle bemerkten.

Die cURL-Befehle scheinen gelegentlich fehlzuschlagen, wenn auf dem Webserver aufgeführt, da aber einfach ein leeres Array zurückgegeben wird, läuft das Programm weiter, es werden nur keine Werte in die Histories- und/oder Dividends-Table eingetragen. Wir konnten dies lösen indem wir den Statuscode auslesen und solange dieser ungleich 200 ist, mit einer Pause von einer Sekunde, bis zu 11 Anfragen absenden bevor dem Nutzer im schlimmsten Fall eine Fehlermeldung ausgegeben wird.

Ein Fehler, den wir nur auf dem Webserver beobachten können und erst wenige Tage vor der Präsentation entdeckt wurde, da er nur erkennbar ist, wenn zu einer Aktie keine Daten vorhanden sind, tritt immer bei den gleichen Aktien auf, z.B.: bei „MSFT“. Dabei werden bei diesen Aktien Werte nicht in die Datenbank geladen, obwohl der cURL erfolgreich ist. Da wir normalerweise unsere Datenbank mit Inhalt vom lokalen System auf den Webserver luden, blieb dies unentdeckt.

Um sicherzugehen, dass während der Präsentation dieser Fehler nicht zu beobachten ist, haben wir am Abend vor der Präsentation alle Aktien die für unsere Suchfunktion hinterlegt sind auf unserer lokalen Version der Webseite heruntergeladen und manuell in die Datenbank des Webservers importiert, wodurch alle Informationen für mindestens die nächsten 24 Stunden, spätestens bis zum nächsten Aufruf nach Ablaufen des Timestamps, in der Datenbank des Webservers verbleiben und den Nutzern angezeigt werden.

# Vorkenntnisse

Die Vorkenntnisse innerhalb unserer Gruppe sind relativ homogen, wenn auch gewisse Unterschiede bestehen. Yorrick hat mit seinem Bachelor der Informatik das meiste theoretische Wissen in der Gruppe. PHP war kein Bestandteil seiner Module, jedoch konnte er sich durch das Wissen anderer Programmiersprachen schnell einarbeiten. Dadurch war er innerhalb der Gruppe schnell für das Backend zuständig. Hagen hat als theoretischen Hintergrund das wirtschaftswissenschaftliche Studium; neben den einführenden Modulen „Einführung in die EDV“ und „E-Business“ hat er auch bereits das Modul „Computational Intelligence“ und „Intelligent Information Systems“ gehört hat. Insbesondere durch sein in bereits absolvierten Uni-Projekten erworbene Wissen, ist Hagen der Allrounder der Gruppe, der meist, wenn auch nicht ausschließlich die Schnittstelle zwischen Front- und Backend bildet. Das Frontend wiederum, das meist aus JavaScript, JQuery und weiterer Frameworks bestand, wurde hauptsächlich von Patrick übernommen. Neben den gleichen Modulen die auch Hagen in der Universität gehört hat, arbeitet er als Werkstudent im Bereich des Webdevelopments. Dadurch, dass er dort oft mit JQuery, Bootstrap und Co arbeitet, hat es sich angeboten, dass er dies auch innerhalb des Projekts übernimmt.

# Ausblick

Die Grundfunktionalitäten unserer Webseite sind implementiert und funktionieren. Darauf aufbauend kann man besonders für Premiumnutzer den Funktionsumfang erweitern. Besonders das direkte Vergleichen von mehreren Aktien auf einer Seite bietet sich hierbei an. Des Weiteren kann man ein Musterdepot implementieren, damit z.B. junge Nutzer mit Aktien arbeiten können, ohne echtes Geld einsetzen zu müssen oder um Notifications per Mail zu erhalten, wenn eine beobachtete Aktie als risikoarm eingestuft wird. Außerdem kann man sich überlegen das Konzept des einfachen Darstellens der Dividendenstrategie auf andere Strategien zu übertragen. Diese sollten dann jedoch stark getrennt von der Dividenden Analyse stattfinden, um die Einfachheit und Übersichtlichkeit bei zu behalten. Abschließend könnte man Premiumnutzern die Möglichkeit bieten, sich die Grenzwerte der Ampeln selbst zu definieren.

Literaturverzeichnis

[Hasler 2018] Hasler, P. T.: *Reich werden an der Börse: 100 Weisheiten für Ihr Geld*. (Hrsg.):

*Börsenbuchverlag*, 2018

[Kaya 2018] Kaya, O.: *Finanzvermögen der Privathaushalte in Deutschland*. 2018

[Schacht & Fackler 2009] Schacht, U., & Fackler, M.: *Praxishandbuch*

*Unternehmensbewertung: Grundlagen, Methoden, Fallbeispiele*. (Hrsg.): *Springer-Verlag*. 2009