

Abschlussprüfung Winter 2021

Fachinformatiker für Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

Implementierung und Bereitstellung einer Trainee Umgebung

Prüfungsbewerber:

Prüflingsnummer: 7063

Jason Prezioso

Rotdornweg 4

65760 Niederhöchstadt



Ausbildungsbetrieb:

Deutsche Börse AG

Abteilung AD Reference Data

Mergenthalerallee 61

65760 Eschborn

Inhaltsverzeichnis

Abbildungsverzeichnis	4
Tabellenverzeichnis	4
Abkürzungsregister	5
1. Einleitung	6
1.1 Abweichungen zum Projektantrag	6
1.2 Projektumfeld	7
1.3 Projektbeschreibung	7
1.4 Projektziel	7
1.5 Projektbegründung	8
1.6 Projektabgrenzung	8
2. Analysephase	8
2.1 Ist-Analyse	8
2.2 Lastenheft	8
3. Projektplanung	9
3.1 Projektphasen	9
3.2 Wirtschaftlichkeitsanalyse	9
3.2.1 Make or Buy	9
3.2.2 Projektkosten	10
3.2.3 Nicht-monetäre Vorteile	10
3.3 Ressourcenplanung	11
3.4 Entwicklungsprozess	11
4. Planungsphase	11
4.1 Zielplattform	11
4.2 Architekturdesign	11
4.3 Entwurf der Benutzeroberflächen	13
4.4 Geschäftslogik	14
4.5 Pflichtenheft	14
5. Implementierungsphase	14
5.1 Implementierung des Frontends	14
5.2 Implementierung des Backends	15

5.3	Implementierung der Geschäftslogik	16
6.	Fazit	17
6.1	Abnahme	17
6.2	Soll/Ist-Vergleich	17
6.3	Lessons Learned	18
6.4	Ausblick	18
	Eidesstattliche Erklärung	19
A	Anhang	20
A.1	Detaillierte Zeitplanung	20
A.2	Verwendete Ressourcen	21
A.3	Lastenheft	21
A.4	Pflichtenheft	22
A.5	Klassendiagramm	22
A.6	Frontend Mockup	23
A.7	Auszug aus dem Benutzerhandbuch	24-29
A.8	Projektantrag	30-33

Abbildungsverzeichnis

Abbildung 1: Klassendiagramme	22
Abbildung 2: Mockup - Index-Such-Seite	23
Abbildung 3: Mockup - Index-Einstellung-und-Preisdiagramm-Seite	23
Abbildung 4: Index-Such-Seite Startansicht	24
Abbildung 5: Index-Such-Seite Suchfunktion	25
Abbildung 6: Index-Such-Seite Zurücksetzen der Suchkriterien und Tabelle	25
Abbildung 7: Index-Such-Seite Funktion zum Auswählen eines Index	26
Abbildung 8: Index-Einstellung-und-Preisdiagramm-Seite Startansicht	27
Abbildung 9: Index-Einstellung-und-Preisdiagramm-Seite - Funktion zum Ändern der Einstellungen	27
Abbildung 10: Index-Einstellung-und-Preisdiagramm-Seite - Funktion zum Speichern neuer Einstellungen	28
Abbildung 11: Index-Einstellung-und-Preisdiagramm-Seite - Navigieren zurück zur Index-Suchseite	29

Tabellenverzeichnis

Tabelle 1: Grobe Zeitplanung	9
Tabelle 2: Kostenrechnung	10
Tabelle 3: Soll/Ist-Vergleich	17
Tabelle 4: Detaillierte Zeitplanung	20

Abkürzungsregister

HTML	Hypertext Markup Language
JSON	Javascript Object Notation
API	Application Programming interface
IPO	Initial Public Offering
ISIN	International Securities Identification Number
C++	C Object-Oriented Programming Language
XML	Extensible Markup Language
CSS	Cascading Style Sheet
GVIM	Console Graphical Editor
IDE	Integrated Development Environment
ZeroMQ	An Asynchronous Messaging Library

1. Einleitung

Bei der folgenden Projektdokumentation handelt es sich um eine Beschreibung des Autors über den Verlauf seiner IHK-Abschlussarbeit, die er während seiner Ausbildung zum Fachinformatiker, Fachrichtung Anwendungsentwicklung, durchgeführt hat. Der Autor absolvierte seine Ausbildung bei der Deutschen Börse AG, die ein weltweit tätiger Marktinfrastukturanbieter im Börsenbereich ist. Die Deutsche Börse AG bietet die folgenden Produkte und Dienstleistungen entlang der gesamten Wertschöpfungskette des Wertpapier- und Derivatehandels an:

- Pre-IPO und Listing
- Handel
- Clearing
- Abwicklung
- Verwahrung
- Sicherheiten- und Liquiditätsmanagement
- Marktdaten
- Indizes
- Technologie

1.1 Abweichungen zum Projektantrag

Bei der anfänglichen Planung des Projekts wurde davon ausgegangen, dass eine unbestimmte Anzahl von Nutzern ein vorprogrammiertes Installationsskript verwenden könnte, um ihre eigene Umgebung auf einer Cloud-Maschine zu erstellen. Diese Idee wurde später wegen der damit verbundenen hohen Zusatzkosten verworfen. Cloud-Ressourcen sind derzeit sehr teuer, und die Möglichkeit, jedem Nutzer eine eigene Cloud-Umgebung zur Verfügung zu stellen, war mit dem derzeitigen Budget nicht machbar. Stattdessen wurde beschlossen, die Trainee-Umgebung auf nur einer Cloud-Maschine laufen zu lassen, auf die alle Nutzer Zugriff haben würden. Alternativ könnte ein Benutzer die Umgebung auch lokal auf seinem Arbeitsgerät installieren und hosten.

1.2 Projektumfeld

Das Projekt wird im „Index Services Development“-Team der Deutschen Börse AG durchgeführt. Die Deutsche Börse AG wurde 1992 gegründet und beschäftigt über 9000 Mitarbeiter weltweit. Der Hauptsitz befindet sich in Eschborn, Deutschland. Das „Index Services Development“-Team ist mit der Wartung und Pflege der Indexberechnungssysteme der Deutschen Börse betraut und hauptsächlich für die Vorbeugung von möglichen Systemausfällen verantwortlich. Zu seinen Aufgaben gehören die regelmäßige Entwicklung, Durchführung und Installation von Systempatches und -updates, Bugfixes, die Durchführung von Systemtests und die Systemüberwachung. Wir arbeiten eng mit den Kunden zusammen und bieten einen 24/7-Support an.

1.3 Projektbeschreibung

Das System wird ein Dummy-Backend in C++ haben, das rudimentär das „Ticken“ eines Indexes, d. h. die Preise, die für einen bestimmten Index eingehen, simuliert. Intern wird das System so programmiert, dass es Berechnungen durchführt, um die Funktionalität der komplexeren Berechnungen zu simulieren, die in unserem eigentlichen System durchgeführt werden, ohne jedoch den Code mit übermäßig komplizierten Berechnungsklassen zu überladen. Das System wird so eingerichtet, dass mehrere dieser Prozesse parallel ablaufen, wobei der Unterschied nur in der jeweiligen Konfiguration besteht. Die vom System generierten Daten werden dann als Dummy-Indizes auf einer Weboberfläche in Form von Tabellen und Diagrammen visualisiert, in denen der Benutzer die vergangenen und aktuellen Daten eines bestimmten Index sehen kann

1.4 Projektziel

Ziel des Projekts ist es, den Index-Berechnungsteil des XIS-Systems zu replizieren. Der replizierte Berechnungsteil soll die Kernfunktionen des „Tickens“ von Indizes (Preisgenerierung von Indizes) zusammen mit einigen kleineren Konfigurationsoptionen nachahmen. Das Replikat wird als vereinfachte Version des Originals erstellt, um neuen Mitarbeitern und Trainees eine benutzerfreundliche Möglichkeit zur Einarbeitung zu bieten, und dient außerdem als Test für eine mögliche Alternative zu unserem aktuellen System.

1.5 Projektbegründung

Da das „Index Services Development“-Team regelmäßig neue Trainees einstellt, ist die Schaffung eines vereinfachten Systems insofern nützlich, weil es die Lernkurve für neue Trainees erheblich verkürzen kann, indem es einige Unklarheiten beseitigt und ihnen einen klareren Blick auf die Kernfunktionen des Systems vermittelt. Ein Nebenprodukt des Aufbaus eines solchen Systems wird auch die Erprobung einer Alternative zum gegenwärtig verwendeten aber veralteten System sein, und einen potenziellen künftigen Ersatzkandidaten darstellen.

1.6 Projektabgrenzung

Da der Projektumfang von der IHK Frankfurt auf maximal 70 Stunden festgelegt wurde, werden vorerst nur einige ausgewählte Prozesse aus dem Hauptsystem durch die Web-Applikation visualisiert. Diese Auswahl wird gemeinsam mit dem Auftraggeber, („Index Services Development“-Team der Deutschen Börse AG) erstellt.

2. Analysephase

2.1 Ist-Analyse

Das „Index Services Development“-Team verfügt über kein System für die Einarbeitung von neuen Mitarbeitern. Die einzige Möglichkeit, sich mit unserem Indexberechnungssystem vertraut zu machen, ist, die ganze Systemdokumentation zu lesen, oder das zeitaufwändige „Learning-By-Doing“.

2.2 Lastenheft

Nach Abschluss der Analysephase wurde gemeinsam mit der „Index Services Development“-Team ein Lastenheft erstellt. Es enthält alle vom Auftraggeber gewünschten Anforderungen an das Projekt. Die Spezifikationen sind im Anhang [A.3 Lastenheft](#) enthalten.

3. Projektplanung

3.1 Projektphasen

Dem Autor wurden 70 Stunden eingeräumt, in denen er die Umsetzung des Projekts durchführen konnte. Zu Beginn wurden verschiedene Projektphasen aufgelistet und definiert, und dann wurden die 70 Stunden bei Bedarf auf diese Phasen verteilt.

Eine grobe Zeitplanung ist unten in der [Tabelle 1: Grobe Zeitplanung](#) zu sehen. Die Tabelle besteht aus Hauptphasen, die weiter in kleinere Unterphasen unterteilt wurden. Eine detaillierte Ansicht dieser Phasen finden Sie im Anhang [A.1 Detaillierte Zeitplanung](#).

Phase	Dauer
Analyse	5 h
Planung	9 h
Implementierung	42 h
Abnahme	1 h
Dokumentation	13 h
Gesamt	70 h

Tabelle 1: Grobe Zeitplanung

3.2 Wirtschaftlichkeitsanalyse

Der derzeitige Prozess, einen neuen Trainee mit unserem derzeitigen Indexberechnungssystem vertraut zu machen, ist mühsam und zeitaufwendig. Die Realisierung des Projekts wird den Lernprozess vereinfachen, und als direktes Ergebnis eine Reduzierung der Zeit, die ein Trainee benötigt, um eigenständig und einsatzbereit zu werden, haben. Da Zeit und Geld sehr eng miteinander verbunden sind, kann man davon ausgehen, dass die Verkürzung der Einarbeitungszeit eines Trainees in unser System direkt zu einem wirtschaftlichen Vorteil für das Unternehmen führen wird.

3.2.1 Make or Buy

In Anbetracht der technologischen und fachlichen Anforderungen, die für die Durchführung dieses Projekts erforderlich sind, kann die Durchführbarkeit und Kosteneffizienz einer Auslagerung dieses Projekts nicht außer Acht gelassen werden. Da jedoch viele Aspekte

dieses Projekts von der Kenntnis der inneren Funktionsweise unserer Systeme abhängen, von denen viele Teile als vertraulich eingestuft sind, wird eine externe Lösung sehr viel schwieriger, so dass eine "interne" Lösung aus Sicherheitsgründen vorzuziehen ist.

3.2.2 Projektkosten

Nachstehend finden Sie die geschätzten Kosten des Projekts. Aufgrund des Datenschutzes können die genauen Kosten des Projekts nicht ermittelt werden, sondern nur anhand von Branchendurchschnittswerten berechnet werden. Der durchschnittliche Stundensatz eines Auszubildenden liegt bei ca. 15 EUR und der eines Senior-Entwicklers bei ca. 65 EUR. Zu Beratungszwecken wurde ein Kollege hinzugezogen, weshalb der Stundensatz eines Senior-Entwicklers auch für das Projekt relevant ist. Abgesehen von der jährlichen Gebühr in Höhe von 12.000 EUR für Microsoft Azure wurden bei der Erstellung des Systems kostenlose Open-Source-Software verwendet. Zu den Gesamtkosten kommen noch Hardwarekosten in Form eines Lenovo ThinkPad im Wert von 2.500 EUR hinzu. Der Büroarbeitsplatz und Gemeinkosten für z.B. Strom werden nicht berechnet. Die Kosten sind unten in der [Tabelle 2 Kostenrechnung](#) aufgelistet.

Vorgang	Mitarbeiter	Zeit	Kosten pro Stunden	Gesamt
Entwicklung	1 x Auszubildender	70 h	15,00 €/h	1.050,00 €
Fachgespräch	1 x Mitarbeiter	5 h	65,00 €/h	325,00 €
Hardware	1x Lenevo ThinkPad	70 h	0,40 €/h	28,04 €
Software	Microsoft Azure-Servicegebühr	70 h	5,77 €/h	50,48 €
			Projektkosten Gesamt	1.453,52 €

Tabelle 2 Kostenrechnung

3.2.3 Nicht-monetäre Vorteile

Die nicht-monetären Vorteile dieses Projekts sind u.a. die Steigerung des Selbstbewusstseins, der Lernbereitschaft und der Beibehaltungsquote der neuen Trainees. All diese Faktoren werden wiederum zu einer funktionelleren und effizienteren Arbeitsmoral des gesamten Teams führen.

3.3 Ressourcenplanung

Eine Liste aller Ressourcen, die für das Projekt verwendet wurden, finden Sie im Anhang [A.2 Verwendete Ressourcen](#). Darin enthalten sind alle Hardware-, Software- und Personalressourcen, die für die Durchführung des Projekts erforderlich waren. Um die Kosten des Projekts so gering wie möglich zu halten, wurde anstelle von kostenpflichtiger Lizenzsoftware, freie Lizenzsoftware (Open-Source) und Lizenzsoftware, die sich bereits im Besitz der Gruppe Deutsche Börse befand, eingesetzt.

3.4 Entwicklungsprozess

Vor dem Beginn der Umsetzung des Projekts wurden geeigneter Entwicklungsprozessen verglichen. Der gewählte Prozess sollte die Vorgehensweise beschreiben, in deren Stil die Realisierung des Projekts erfolgen sollte. Dem aktuellen Industriestandard folgend, wurde für das Projekt der Agile Entwicklungsprozess gewählt, der eine iterative Entwicklung vorsieht.

4. Planungsphase

4.1 Zielplattform

Das endgültige Projekt wird in der Cloud als Web-Applikation bereitgestellt. Nach Abschluss des Projekts wird die Web-Applikation auf einem einzigen Cloud-Rechner auf der Basis von Microsoft Azure betrieben, auf den alle Nutzer mit den entsprechenden Zugriffsrechten zugreifen können. Ursprünglich war geplant, für jeden einzelnen Nutzer eine eigene Cloud-Maschine bereitzustellen. Dieser Plan wurde jedoch aufgrund der hohen Kosten des Cloud-Computing gestrichen. Wenn ein Nutzer seine eigene private Schulungsumgebung wünscht, hat er auch die Möglichkeit, die Web-Applikation lokal auf seinem Arbeitsgerät zu installieren und auszuführen.

4.2 Architekturdesign

Das Projekt verwendet das Python-Webframework Flask, um Python-Code zu interpretieren und den Webserver zu betreiben. Alle im Projekt verwendeten Websites werden auf dem Flask-Webserver gehostet. Um eine Kommunikationslinie zwischen den Model- und View-Komponenten der Web-Applikation herzustellen, wird XMLHttpRequest

verwendet. XMLHttpRequest ist eine API in Form eines Objekts, dessen Methoden Daten zwischen einem Webbrowser und einem Webserver übertragen. In diesem Projekt wurde es zusammen mit der HTML-"POST"-Methode verwendet, um variable Daten an den Webserver zu senden.

Die Daten konnten anschließend in Python verarbeitet und bei Bedarf über die "render_template"-Methode von Flask in Form von vordefinierten Variablen an die HTML-Dateien zurückgeschickt werden. Die XMLHttpRequest-API ermöglichte es auch, anwendungsrelevante Daten, die mit jQuery erfasst wurden, an den Flask-Webserver zu senden, wo die Daten zur späteren Verwendung in JSON-Dateien geschrieben und gespeichert werden konnten.

Auf der C++-Seite wurde die Kommunikation zwischen den Prozessen durch den Einsatz eines Brokers erreicht, der die Pub/Sub-Funktionalität durch die Verwendung von bestimmten Eingangs-/Ausgangsports ermöglichte. Der Broker empfängt Daten von einem Publisher, der die Daten an die Importportnummer des Brokers sendet. Der Broker sendet die Daten dann über die Portnummer des Ausgangs an die Abonnenten zurück.

Die Kommunikation zwischen der C++-Seite und der Python-Seite erfolgt über die JSON-Dateien. Alle Indexpreisdaten, die auf der C++-Seite erfasst werden, werden in JSON-Dateien gespeichert. Die Daten werden dann von den Python-Skripten in das HTML/jQuery-Frontend eingespeist und dem Benutzer angezeigt werden. Die JSON-Dateien dienen auch als Kommunikationslinie von Python zu C++. Wenn der Benutzer beispielsweise das Tick-Intervall, die Startzeit und die Endzeit eines ausgewählten Index ändern möchte. Die neuen Einstellungen werden dann von jQuery an Python gesendet, wo die JSON-Datei dann mit dem Tick-Intervall, der Start- und Endzeit für die C++-Datei aktualisiert wird.

4.3 Entwurf der Benutzeroberflächen

Die Web-Applikation sollte so intuitiv und benutzerfreundlich wie möglich sein, daher wurde eine klare und einfache Benutzeroberfläche entwickelt. Zu Beginn wurden Mockups erstellt, um das Projekt zu visualisieren, bevor mit dem Schreiben des Codes begonnen wurde. Zu

diesem Zweck wurde die Software Balsamiq Mockups verwendet. Ein Mitglied des "Index Services Development"-Teams wurde in diesen Prozess einbezogen, um das fertige Mockup entweder zu genehmigen oder abzulehnen, um die Erwartungen und Ideen des Teams bestmöglich zu erfüllen. Für das Projekt wurden zwei verschiedene Ansichten erstellt. Eine Ansicht, in der der Benutzer alle im konfigurierbaren Backend vorhandenen Indizes filtern kann, und eine detailliertere Ansicht, in der der Benutzer detaillierte Informationen zu einem bestimmten Index sehen und sogar einige Konfigurationseinstellungen im laufenden Betrieb ändern kann. Die Mockups, die für die einzelnen Ansichten erstellt wurden, sind im Anhang [A.5 Frontend Mockup](#) zu finden.

4.4 Geschäftslogik

Die Web-Applikation hat zwei Ebenen. Die oberste Ebene der Web-Applikation ist die Webschnittstelle. Die Webschnittstelle besteht aus zwei Webseiten. Die erste Webseite ist die Indexsuche-Seite, die einen allgemeinen Überblick über alle Indizes bietet, die in einer Backend-JSON-Datei vorhanden sind. Diese Indizes werden in einer Tabellenansicht angezeigt, wobei für jeden Index die Indexfamilie, die Index-ISIN, der Indexname, die Börse, der Preis und die Währung angezeigt werden. Je nach den Bedürfnissen des Benutzers können Indizes aus dieser Datei entfernt oder hinzugefügt werden, jedoch nur vom Backend aus. Das Hinzufügen oder Entfernen von Indizes ist nicht über die Weboberfläche möglich. Auf der Indexsuche-Seite kann der Benutzer umfangreiche Filterungen nach allen sechs oben genannten Eigenschaften vornehmen. Dies gibt dem Benutzer die Möglichkeit, nach bestimmten Indizes zu filtern, die genau die angegebenen Bedingungen erfüllen. Der Benutzer kann dann auf einen der Indizes, die in den Ergebnissen erscheinen, doppelklicken und wird auf die Indexchartdiagramm-und-Einstellungen-Seite für diesen Index weitergeleitet.

Auf der Indexchartdiagramm-und-Einstellungen-Seite der Webschnittstelle erhält der Benutzer weitere Informationen über den ausgewählten Index, wie z. B. das Tick-Intervall (wie oft der Preis des Index angezeigt werden soll), die Startzeit des Index und die Endzeit des Index. Der Benutzer sieht auch ein sich in Echtzeit aktualisierendes Preisdiagramm, das alle Preisbewegungen zwischen dem Start- und Endzeitpunkt des Index anzeigt. Darüber hinaus können die Einstellungen für das Tick-Intervall, die Startzeit und die

Endzeit vom Benutzer spontan geändert werden. Die Änderungen werden dann auf das Preisdigramm angewendet und in einer Konfigurationsdatei im Backend gespeichert.

Der untere Ebene der Web-Applikation ist der Flask-Python-Server. In den Python-Skripten wird alles zusammengeführt, verarbeitet und dann an die oberste Ebene weitergegeben. Die Standardfunktionalität der Python-Skripte besteht darin, alle JSON-Daten, die im Backend gespeichert sind, abzurufen und sie an die HTML-Dateien im Frontend zu senden, wo sie dem Benutzer angezeigt werden können. Außerdem werden alle Benutzereingabedaten, die vom jQuery-Code des Frontends erfasst werden, über XmlHttpRequests an die Python-Skripte gesendet. Diese Daten werden dann verarbeitet und in den Backend-JSON-Dateien gespeichert, und alle Änderungen werden dynamisch wieder auf dem Frontend angezeigt. Der C++-Code ist ebenfalls mit dem Flask-Python-Server verbunden und leitet alle Indexpreisdaten an die Python-Skripte weiter, wo sie dann verarbeitet, gespeichert und bei Bedarf auf dem Frontend angezeigt werden.

4.5 Pflichtenheft

Es wurde ein Pflichtenheft für das Projekt erstellt. Es enthält die Umsetzung der Anforderungen, die in Abschnitt [2.2 Lastenheft](#) festgelegt wurden. Es dient dazu, zu überprüfen, ob die vereinbarten Anforderungen bis zum Projektende umgesetzt wurden. Das Pflichtenheft ist im Anhang [A.4 Pflichtenheft](#) abgebildet.

5. Implementierungsphase

5.1 Implementierung des Frontends

Die Implementierung begann mit der Erstellung der Dateien IndexSearch.html, indexChart.html und stylesheet.css, um dem Autor einen visuellen Überblick über das Projekt zu verschaffen, auf dessen Grundlage er mit dem Hinzufügen der erforderlichen Funktionen beginnen konnte. Die erforderliche Funktionalität auf dem Frontend wurde mit den Dateien indexSearch.js und indexChart.js erreicht, wobei JavaScript und jQuery verwendet wurden, um Ereignisse auszulösen, sobald der Benutzer mit einem

HTML-Element interagiert. Ein Beispiel dafür sind Event-Listener auf der indexSearch.html-Webseite, die darauf warten, dass ein Benutzer Daten in die Suchfelder eingibt und dann die Filterfunktion ausführt. Ein weiteres Beispiel für die indexChart.html-Webseite wäre die click()-Funktion von jQuery, die die Änderung der Konfigurationsdaten für den ausgewählten Index auslöst, wenn der Benutzer auf die Schaltfläche "Einstellungen aktualisieren" klickt. Die Funktion click() aktualisiert dynamisch das Ticken des Live-Preisdiagramms und interagiert dann mit dem Flask-Python-Server über XMLHttpRequests, um die Konfigurationsdatei im Backend zu aktualisieren.

5.2 Implementierung des Backends

Nach der anfänglichen Analyse wurde ein kleiner Test-Pub/Sub eingerichtet, der so umgestaltet wurde, dass er den Preis-Broker-IndexObject-Teil der Architektur repräsentiert. Danach wurde die Rückseite der Indexberechnung auf die gleiche Weise implementiert.

Der grundlegende Aufbau ist: PriceGenerators verbinden sich mit dem nach innen gerichteten Port des PriceBrokers. Die Indizes verbinden sich mit dem nach außen gerichteten Teil des PriceBrokers. Auf diese Weise kann eine m:n-Beziehung für PriceGenerator- und IndexObject-Objekte hergestellt werden, und es ist immer möglich, die Anzahl der Peers in diesem einfachen Netzwerk zu erhöhen oder zu verringern. Dasselbe geschieht für das IndexObject und Index-Objekte mit dem Index Broker. Ein Preis wird nun zunächst von einem PriceGenerator "erzeugt", der eine Nachricht mit einem neuen Preis an sein jeweiliges Topic schickt. Mehrere PriceGenerators können an das gleiche Topic pushen, ZeroMQ macht einen Datendurchsatz von vielen tausend solcher Nachrichten zur gleichen Zeit sehr gut möglich. Die Nachricht wird durch den Broker propagiert, der dann den neuen Preis an seine jeweiligen abonnierenden Indexobjekte im Backend pusht. Ein Indexobjekt führt dann einfache Berechnungen mit der eingehenden Nachricht durch, wie z. B. eine einfache Addition. Anschließend gibt es den neu berechneten Preis an den IndexBroker weiter, und zwar auf ähnliche Weise, wie der Preis an den PriceBroker weitergeleitet wurde.

Währenddessen schreiben sowohl IndexObjects als auch Indexe ihre Preise in bestimmten Intervallen im json-Format auf die Festplatte, ebenso wie einige Metadaten. Diese geschriebenen Ticks können dann an das Frontend gesendet werden.

Alle Objekte laufen als separate Prozesse mit ihrer jeweiligen main()-Funktion. Das Ziel ist es, mindestens 5-10 Prozesse gleichzeitig laufen zu lassen, um einen kleinen Teil der Umgebung des Indexsystems zu imitieren. Der Entwicklungsprozess begann mit einer eingehenden Analyse des ZeroMQ-Pub/Sub-Musters und der Frage, wie eine n:m-Beziehung ermöglicht werden kann, bei der die Mitglieder ihre Verbindung zueinander leicht aufbauen und wieder trennen können, nämlich mit dem Pub-Broker-Sub-Muster.

5.3 Implementierung der Geschäftslogik

Die Geschäftslogik wurde unter Verwendung der Atom IDE und GVIM implementiert, wobei eines der vier Kernprinzipien der agilen Arbeitsmethodik befolgt wurde. Das gewählte Prinzip war das Prinzip "Reagieren auf Veränderungen hat Vorrang vor dem Befolgen eines Plans". Diese Arbeitsweise ist effizienter und flexibler, da oft viel Zeit damit vergeudet wird, Probleme zu lösen, die auftreten können, wenn man an konkrete Implementierungspläne gebunden ist. Durch die Befolgung dieses Kernprinzips können stattdessen kreative und unkonventionelle Lösungen genutzt werden, um Probleme zu lösen und Innovationen zu fördern.

Die folgenden Dateien wurden erstellt:

- indexSearch.html
- indexChart.html
- stylesheet.css
- indexSearch.js
- indexChart.js
- webapp.py
- broker.cpp
- controller.cpp
- index.cpp

- index.hpp
- indexFamily.cpp
- indexFamily.hpp
- priceTicker.cpp
- priceTicker.hpp

6. Fazit

6.1 Abnahme

Nach Abschluss des Projekts präsentierte der Autor dem Auftraggeber das fertige Produkt. Alle Anforderungen und Spezifikationen wurden erfolgreich erfüllt, so dass das Ziel als erreicht galt. Das Projekt wurde zur vollen Zufriedenheit des Auftraggebers übergeben und umgesetzt.

6.2 Soll/Ist-Vergleich

Wie unten zu sehen ist, wich der geplante Zeitbedarf von den tatsächlichen Stunden ab, die für die Durchführung des Projekts erforderlich waren. Dies ist der [Tabelle 3 Soll/Ist-Vergleich](#) zu entnehmen

Phase	Soll	Ist	Differenz
Analyse	5 h	5 h	
Planung	9 h	10 h	+1 h
Implementierung	35 h	37 h	+2 h
Abnahme	4 h	3 h	-1 h
Dokumentation	17 h	15 h	-2 h
Gesamt	70 h	70 h	

Tabelle 3 Soll/Ist-Vergleich

Ursprünglich war geplant, jedem Nutzer eine eigene virtuelle Maschine in der Cloud zur Verfügung zu stellen, auf der er seine eigene Schulungsumgebung installieren und ausführen kann. Aufgrund der hohen Kosten, die mit Cloud Computing verbunden sind, wurde diese Idee verworfen. Dies ist u.a. im Abschnitt [1.1 Abweichungen zum Projektantrag](#) zu sehen. Aufgrund dieser unerwarteten Entwicklung wurde der Planungsphase des Projekts eine zusätzliche Stunde hinzugefügt. In der Abnahmephase

des Projekts konnte jedoch eine Stunde eingespart werden, da das Cloud Computing verkleinert wurde. Da Python auch eine neue Programmiersprache für den Autor war, traten mehr Probleme als erwartet auf. Dies verlängerte die Entwicklungsphase des Projekts um 2 Stunden, um die Anwendung richtig zu debuggen. Die Dokumentation des Projekts verlief reibungslos und wurde dank eines guten Zeitmanagements 2 Stunden früher als geplant fertiggestellt.

6.3 Lessons Learned

Durch die Durchführung dieses Projekts konnte der Autor wertvolle Erfahrungen in der Planung und Durchführung von Projekten sammeln. Es wurde deutlich, dass gute Kommunikationsfähigkeiten und die Zusammenarbeit zwischen dem Auftraggeber und dem Autor für den erfolgreichen Abschluss dieses oder jedes anderen Projekts von größter Bedeutung sind. Es wurde dem Autor auch klar, dass selbst eine abgerundete Theorie in der Praxis nicht immer gut umgesetzt werden kann, und dass eine gute Planung auch bedeutet, mit Fehlern zu rechnen, und einen Plan für eventuell auftretende Probleme zu haben. Darüber hinaus wurden vielfältige technische Fähigkeiten in Form eines besseren Verständnisses des derzeit eingesetzten Indexberechnungssystems, seiner Fehler und allgemeiner Kenntnisse in der Webentwicklung erworben. Die Entwicklung der Webanwendung stellte viele Herausforderungen dar, war aber letztendlich sowohl lohnend als auch befriedigend.

6.4 Ausblick

Da das Projekt auf einen Zeitraum von 70 Stunden beschränkt war, konnte leider keine vollständige, vereinfachte Version des Indexberechnungssystems gebaut werden. Die aufgewendete Zeit reichte jedoch aus, um einen gut funktionierenden Prototyp zu bauen, der als guter Ausgangspunkt für zukünftige Weiterentwicklungen dienen kann. Das Projekt hinterlässt dem Auftraggeber eine Grundlage, auf der er leicht aufbauen kann, indem er kleinere Module zum Web-Applikation hinzufügt, entweder durch derzeit beschäftigte Entwickler oder besser noch durch zukünftige Trainees. Diese zukünftigen Trainees könnten durch die Implementierung neuer Module wertvolle Erfahrungen in der Web-Entwicklung und im allgemeinen Wissen über das Indexberechnungssystem selbst sammeln.

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Arbeit selbstständig und nur unter Zuhilfenahme der ausgewiesenen Hilfsmittel angefertigt habe.

Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach anderen gedruckten oder im Internet verfügbaren Werken entnommen sind, habe ich durch genaue Quellenangaben kenntlich gemacht.

Eschborn, den 10.11.2021



Jason Prezioso

A Anhang

A.1 Detaillierte Zeitplanung

Analyse	5 h
Durchführen der Ist-Analyse	1 h
Definieren von abbildbaren Prozessketten	2 h
Kundenunterstützung bei Erstellung des Lastenhefts	2 h
Planung	10 h
Erstellung eines Projektplans	2 h
Erstellung eines Mockups des Frontends	2 h
Erstellung des Cloud-Layouts	3 h
Bestimmung des Backends	3 h
Implementierung	37 h
Implementierung des Backends	18 h
Implementierung des Clouds	5 h
Implementierung des Frontends	14 h
Abnahme	3 h
Transport und Integrationstest	1 h
Übergabe an den Auftraggeber	1 h
Kosten- und Zeitvergleich	1 h
Dokumentation	15 h
Erstellen der Projektdokumentation	15 h
Gesamt	70 h

Tabelle 4 Detaillierte Zeitplanung

A.2 Verwendete Ressourcen

Hardware:

- Lenovo Entwickler-Laptop

Software:

- Ubuntu v20.04
- Atom IDE
- Balsamiq Mockups
- Google Docs
- Flask Server v2.0.2
- Python v3.10
- C++ 20
- HTML5
- jQuery v3.5.1
- Bootstrap v5.1.1

A.3 Lastenheft

Folgendes Lastenheft beschreibt die Anforderungen, welche die Anwendung zu erfüllen hat.

Die Anwendung muss...

- Als Webanwendung auf einem Flask-Server implementiert werden.
- Visuelles Feedback von benutzergetriggerten Prozessen liefern
- Eine detaillierte Sicht auf die Daten einzelner Indizes bieten.
- Mit asynchronen Datenübertragungen arbeiten.
- Ein benutzerfreundliches Framework und Design-/Farbkonzept verwenden, um dem Benutzer eine einfache und intuitive Arbeitsplattform zu bieten.
- Verfügen über ein responsives Design.

A.4 Pflichtenheft

Folgendes Pflichtenheft beschreibt die Anforderungen, welche die Anwendung zu erfüllen hat.

Die Anwendung wird...

- Als Webanwendung laufen, die auf einem Flask-Server gehostet wird
- Dem Benutzer visuelles Feedback zu den vom Benutzer ausgelösten Prozessen geben
- Eine detaillierte Ansicht der einzelnen Indexdaten bieten
- Mit asynchroner Datenübertragung arbeiten
- Eine einfache und intuitive Arbeitsplattform für den Nutzer bieten
- Ein responsives Design haben

A.5 Klassendiagramm

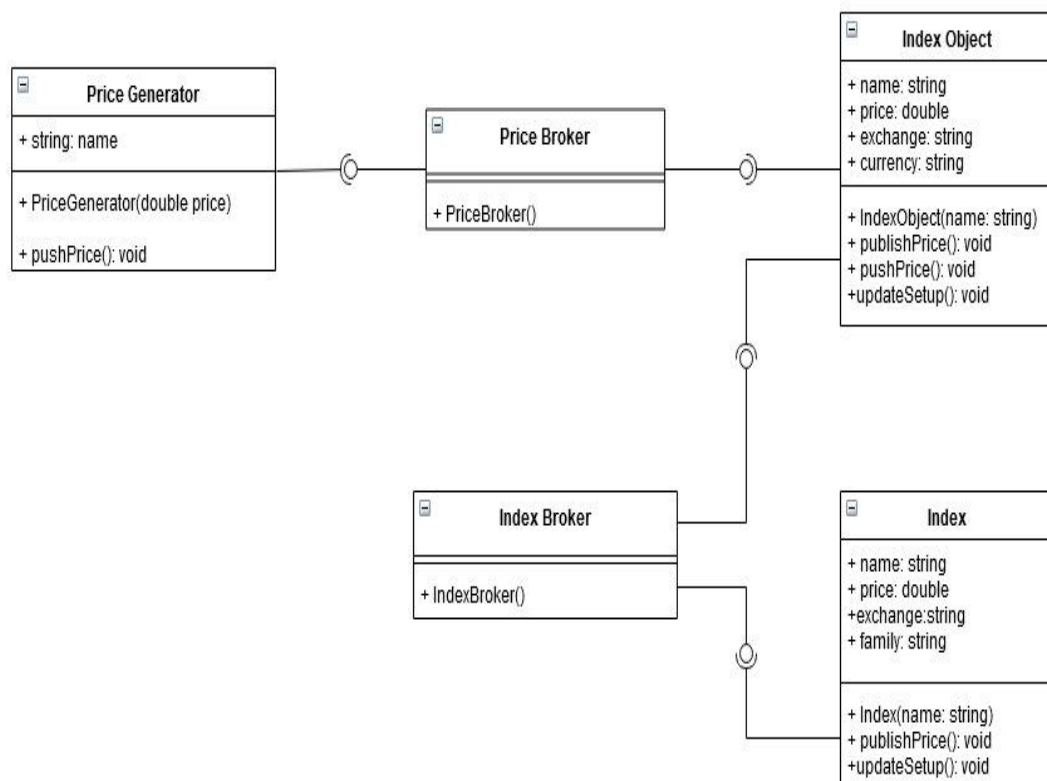


Abbildung 1 klassendiagramm

A.6 Frontend Mockup

A Web Page

https://

Search Criteria

Index Family Index ISIN Index Name

Exchange Currency

Results

Index Family	Index ISIN	Index Name	Exchange	Currency
FLEX	CH0109185402	ESTX 50 DVP Futures Index	ST2	EUR
FLEX	CH0109515863	ESTX 50 Vola.Short Term FutureIndex	ST2	EUR
FLEX	DE000A161DG1	DAX Daily Hedged Index (Price)	ETR	USD
FLEX	DE000A13PJE9	DAX Monthly Hedged Index (Price)	ETR	AUD
FLEX	DE000A3CK389	DAX Daily Hedged Index (Net Return)	ETR	CZK

Abbildung 2 Mockup - Indexsuchseite

A Web Page

https://

Settings

Index ISIN: ABC1234 Tick Interval: 60 Start Time End Time

Start Time: 09:15 End Time: 16:15 Tick Interval

Index Chart: Beispiel

Preis

Zeit

Abbildung 3 Mockup - Indexeinstellungen und Preisdigrammseite

A.7 Auszug aus dem Benutzerhandbuch

Der folgende Auszug aus dem Benutzerhandbuch beschreibt, wie Sie die Web-Applikation verwenden, um alle verfügbaren Indizes zu filtern. Sobald der gewünschte Index gefunden ist, wird gezeigt, wie man zur Seite mit den Einstellungen und dem Preisdigramm navigiert. Von dort aus wird beschrieben, wie die Einstellungen geändert werden können.

Indexsuche und Änderung der Einstellungen

Zu Beginn werden alle Indizes in der Index-Suchtable angezeigt.

Deutsche Börse AG

Index Search

Search Criteria

Index Family	Index ISIN	Index Name
<input type="text"/>	<input type="text"/>	<input type="text"/>
Exchange	Currency	<input type="button" value="Reset"/>
<input type="text"/>	<input type="text"/>	

Table Results

Show entries

Index Family	Index ISIN	Index Name	Exchange	Price	Currency
AGP	CH1234509876	IKEA Holdings	ST2	266.34	EUR
AGP	DE1948375381	Apple Client Holdings	ST2	0.45	USD
FLEX	CH2958275913	Madeup-name inc.	ETR	420.69	CAD
FLEX	DE9958477234	Kaese Kuche GmbH	ST2	13.37	EUR
IBOXX	CH2220593003	Cool Beans Ltd	ETR	801.08	CZK
UNIA	DE1234567890	Eliside inc.	ETR	52.08	AUD

Abbildung 4 Indexsuchseite - Startansicht

Um nach bestimmten Indizes zu filtern, kann der Benutzer die Suchkriterien in die Felder oberhalb der Tabelle eingeben. Indizes, deren Attribute nicht mit den eingegebenen Kriterien übereinstimmen, werden dann automatisch aus der Tabelle entfernt.

Index Search

Search Criteria

Index Family AGP	Index ISIN 	Index Name Holdings
Exchange ST2	Currency 	Reset

Table Results

Show 10 entries

Index Family	Index ISIN	Index Name	Exchange	Price	Currency
AGP	CH1234509876	IKEA Holdings	ST2	266.34	EUR
AGP	DE1948375381	Apple Client Holdings	ST2	0.45	USD

Showing 1 to 6 of 6 entries

Previous 1 Next

Abbildung 5 Indextsuchseite - Suchfunktion

Ein Zurücksetzen sowohl der Index-Suchtablette als auch aller eingegebenen Suchkriterien kann dann durch Klicken auf die Schaltfläche "Reset" erfolgen.

Index Search

Search Criteria

Index Family 	Index ISIN 	Index Name
Exchange 	Currency 	Reset

Table Results

Show 10 entries

Index Family	Index ISIN	Index Name	Exchange	Price	Currency
AGP	CH1234509876	IKEA Holdings	ST2	266.34	EUR
AGP	DE1948375381	Apple Client Holdings	ST2	0.45	USD
FLEX	CH2958275913	Madeup-name inc.	ETR	420.69	CAD
FLEX	DE9958477234	Kaese Küche GmbH	ST2	13.37	EUR
IBOXX	CH2220593003	Cool Beans Ltd	ETR	801.08	CZK
IBOXX	DE1234567890	Elmidea inc.	ETR	53.08	AUD

Abbildung 6 Indextsuchseite - Zurücksetzen der Suchkriterien und Tabelle

Sobald der gewünschte Index gefunden ist, kann der Benutzer einfach auf den in der Tabelle angezeigten Index klicken, um zu den entsprechenden Einstellungen und der Preisdiagramm zu gelangen.

Deutsche Börse AG

Index Search

Search Criteria

Index Family

Index ISIN

Index Name

Exchange

Currency

Reset

Table Results

Show entries

Index Family	Index ISIN	Index Name	Exchange	Price	Currency
AGP	CH12345095	IKEA Holdings	ST2	266.34	EUR
AGP	DE1948375381	Apple Client Holdings	ST2	0.45	USD
FLEX	CH2958275913	Madeup-name inc.	ETR	420.69	CAD
FLEX	DE9958477234	Kaese Küche GmbH	ST2	13.37	EUR
IBOXX	CH2220593003	Cool Beans Ltd	ETR	801.08	CZK

Abbildung 7 Indexsuchseite - Funktion zum Auswählen eines Index

Sobald der Benutzer auf die Seite mit den Einstellungen und dem Preisdiagramm weitergeleitet wird, können die aktuell angewandten Einstellungen links unter "Settings" eingesehen werden. Auf der rechten Seite unter "Settings" kann der Benutzer die Start- und Endzeit sowie das Tick-Intervall des Indexes aktualisieren.

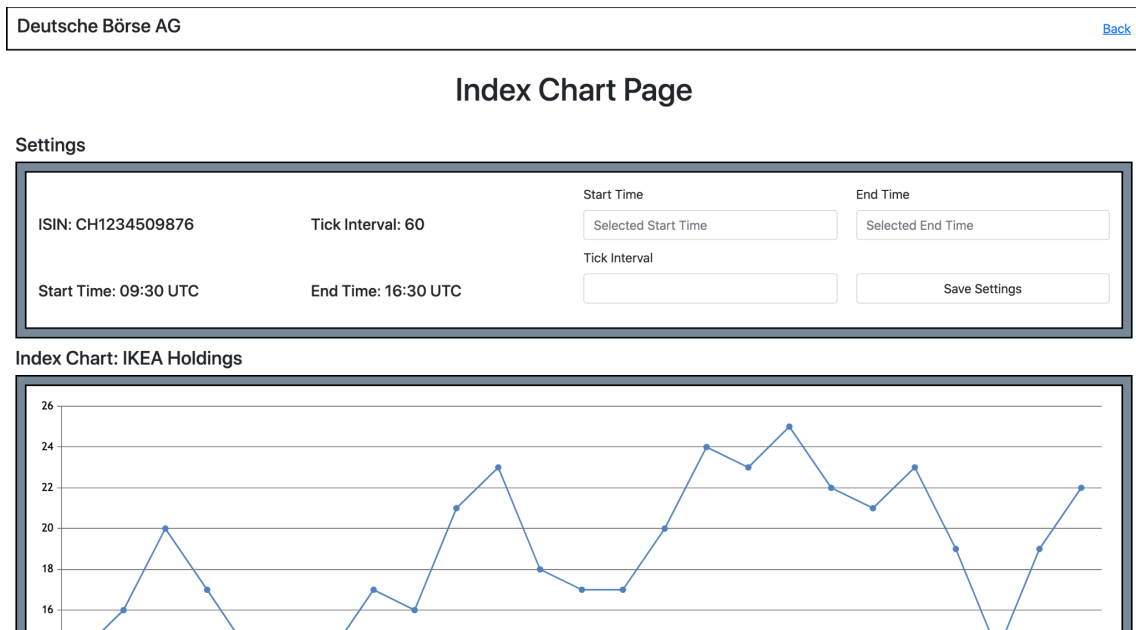


Abbildung 8 Indexeinstellungen und Preisdiagrammseite - Startansicht

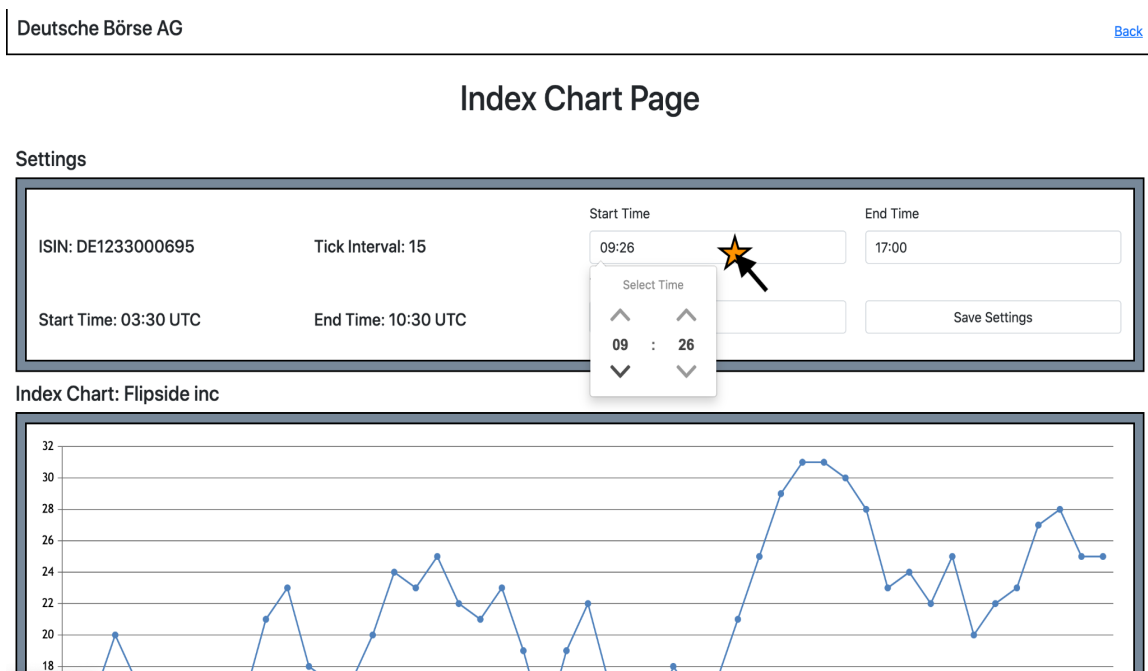


Abbildung 9 Indexeinstellungen und Preisdiagrammseite - Funktion zum Ändern der Einstellungen

Die neuen gewünschten Einstellungen können dann durch Klicken auf die Schaltfläche "Save Settings" auf den Index angewendet werden. Wenn die Einstellungen geändert werden, werden die Ergebnisse sofort im Preisdiagramm angezeigt und dargestellt.

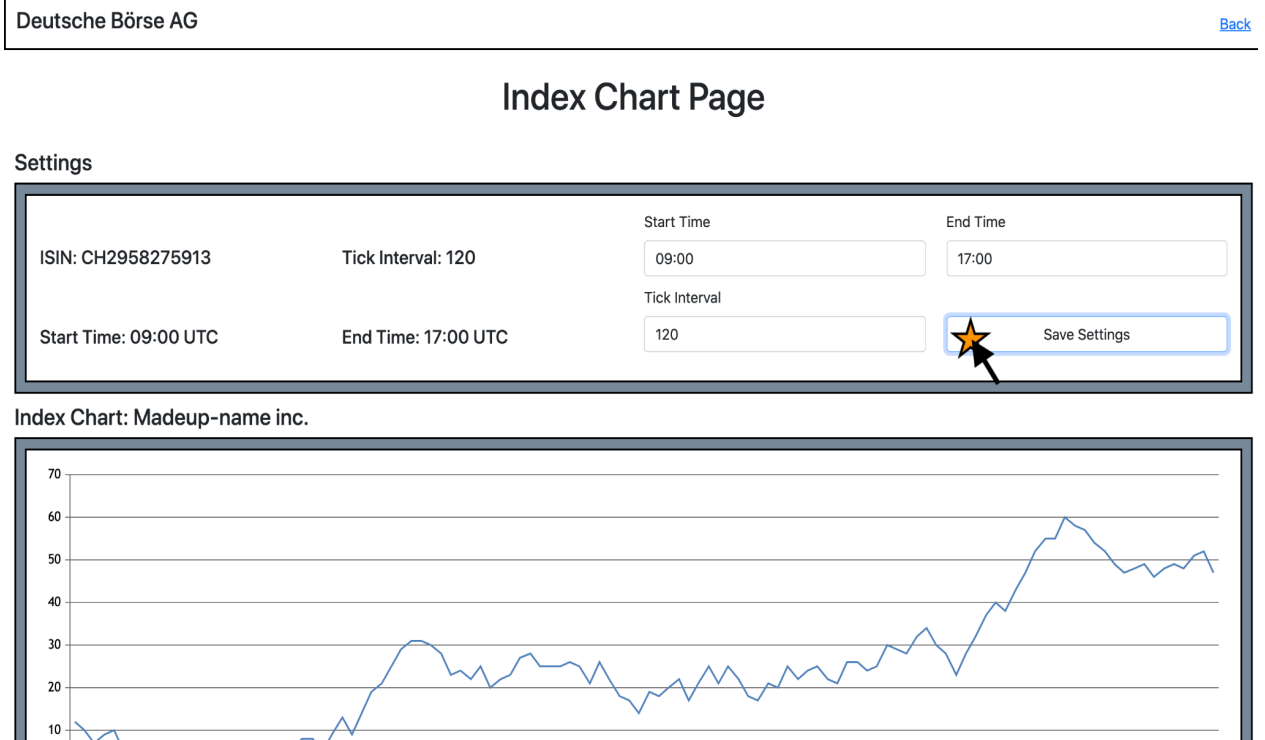


Abbildung 10 Indexeinstellungen und Preisdiagrammseite - Funktion zum Speichern neuer Einstellungen

Um zur Seite "Indexsuche" zurückzukehren, kann der Benutzer auf die Schaltfläche "Back" klicken, die sich oben rechts auf der Seite befindet.

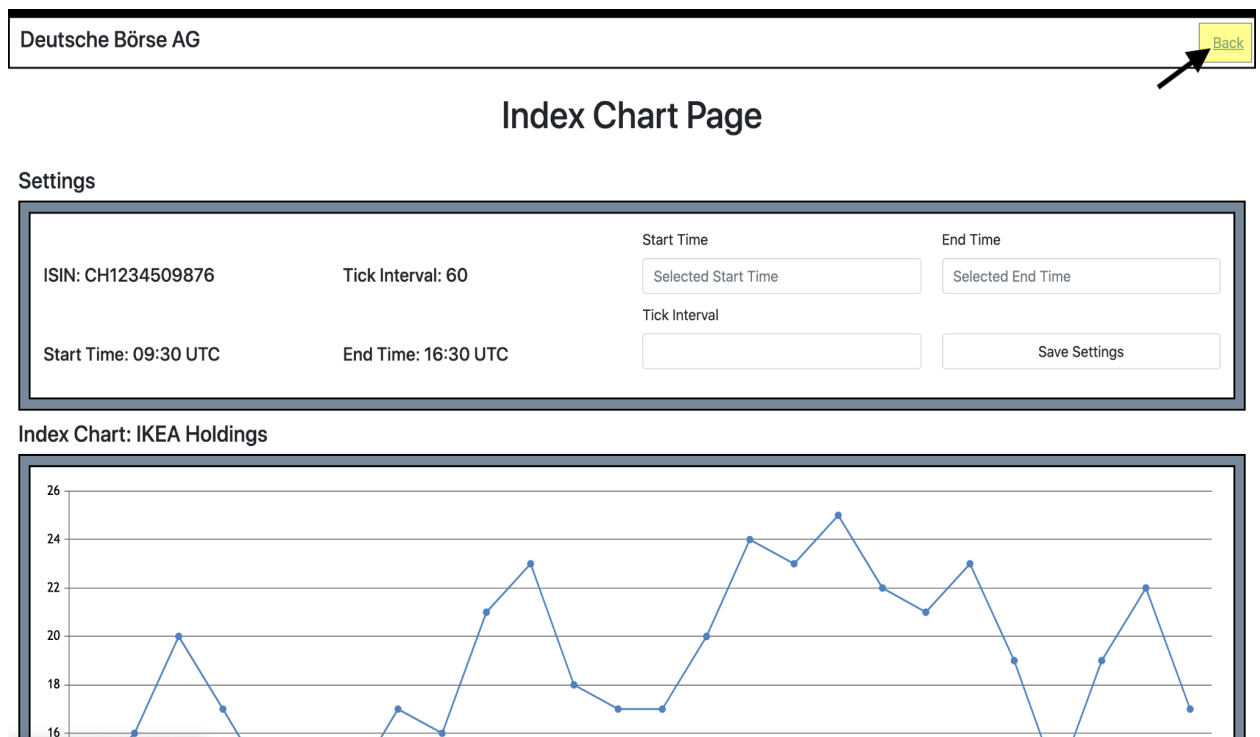


Abbildung 11 Indexeinstellungen und Preisdiagrammseite - Navigieren zurück zur Index-Suchseite



Winterprüfung 2021

Ausbildungsberuf

Fachinformatiker/-in Anwendungsentwicklung

Prüfungsbezirk

Frankfurt fiaf 23 (AP T2V1)

Herr Prezioso Jason

Identnummer: 1345851

E-Mail: jsonprezioso@protonmail.com, Telefon: +49 163 1747396

Ausbildungsbetrieb: Deutsche Börse AG

Projektbetreuer: Herr Jan Patrick Drehwald

E-Mail: jan.patrick.drehwald@deutsche-boerse.com, Telefon: +49 69211 14141

Thema der Projektarbeit

Implementierung und Bereitstellung einer Trainee Umgebung

1 Thema der Projektarbeit

Implementierung und Bereitstellung einer Trainee Umgebung

2 Geplanter Bearbeitungszeitraum

Beginn: 18.10.2021

Ende: 21.11.2021

3 Projektbeschreibung

Ziel des Projektes ist es, den Index-Berechnungsteil des XIS-Systems zu replizieren. Der replizierte Berechnungsteil soll die Kernfunktionen des „Tickens“ von Indizes (Preisgenerierung von Indizes) zusammen mit einigen kleineren Konfigurationsoptionen nachahmen. Das Replikat wird als vereinfachte Version des Originals erstellt, um neuen Mitarbeitern und Auszubildenden eine benutzerfreundliche Möglichkeit zur Einarbeitung zu bieten, und dient außerdem als Test für eine mögliche Alternative zu unserem aktuellen System.

Im derzeitigen System gibt es mehrere Maschinen, auf denen jeweils eine Vielzahl unterschiedlicher Prozesse zu verschiedenen Zwecken laufen, wobei jeder dieser Prozesse möglicherweise eine Vielzahl von Unterprozessen ausführt, die miteinander interagieren. Ein großer Teil des Systems wurde mit Blick auf alte und veraltete Programmierparadigmen geschrieben. Dies macht es für neue Mitarbeiter äußerst schwierig, auch nur einen Bruchteil des Systems zu verstehen. Um das System vollständig zu verstehen, müsste man mehrere verschiedene Kommunikationsprotokolle und Serialisierungsmethoden lesen, auswendig lernen und beherrschen, sowie das Innenleben vieler, möglicherweise völlig unterschiedlicher Ansätze zur Indexberechnung gemäß der Definition kennen und sich den umliegenden Systemen bewusst sein, die nicht unter der Aufsicht des „Index Services“-Teams stehen. Eine Überarbeitung des Systems, um es an modernere Standards anzupassen, ist von den derzeitigen Systembesitzern noch nicht genehmigt worden. Aus diesem Grund wurde ich beauftragt, ein reduziertes System zu bauen, das einige Kernfunktionen kapselt, um einen ersten Einblick in das komplette produktive System ohne die überwältigenden technischen Einzelheiten sowie historisch gewachsene Probleme und veraltete Entwicklungsideen zu geben. Das neue System wird mit Blick auf neue Mitarbeiter und Auszubildende entwickelt und minimiert den Arbeitsaufwand, der erforderlich ist, um sich mit den produktiven Systemen vertraut zu machen.

Da das „Index Services“-Team regelmäßig neue Auszubildende einstellt, ist die Schaffung eines vereinfachten Systems insofern nützlich, weil es die Lernkurve für neue Mitarbeiter erheblich verkürzen kann, indem es einige Unklarheiten beseitigt und ihnen einen klareren Blick auf die Kernfunktionen des Systems vermittelt. Ein Nebenprodukt des Aufbaus eines solchen Systems wird auch die Erprobung einer Alternative zum gegenwärtig verwendeten aber veralteten System sein, und einen potenziellen künftigen Ersatzkandidaten darstellen. Für mein Projekt werde ich ein Dummy-Backend in C++ implementieren, das rudimentär das „Ticken“ eines

Indexes, d. h. die Preise, die für einen bestimmten Index eingehen, simuliert. Intern wird das System so programmiert, dass es Berechnungen durchführt, um die Funktionalität der komplexeren Berechnungen zu simulieren, die in unserem eigentlichen System durchgeführt werden, ohne jedoch den Code mit übermäßig komplizierten Berechnungsklassen zu überladen. Das System wird so eingerichtet, dass mehrere dieser Prozesse parallel ablaufen, wobei der Unterschied nur in der jeweiligen Konfiguration besteht. Die vom System generierten Daten werden dann als Dummy-Indizes auf einer Weboberfläche in Form von Tabellen und Diagrammen visualisiert, in denen der Benutzer die vergangenen und aktuellen Daten eines bestimmten Index sehen kann. Zusätzlich kann der Benutzer die Backend-Parameter über die Plattform anpassen. Das Webinterface wird mit Python und HTML programmiert und auf Flask gehostet. Für das gesamte System wird auch ein Installationsskript implementiert, sodass weitere Systeme dieser Art einfach in der Cloud installiert oder gebootet werden können.

4 Projektumfeld

Das Projekt wird im „Index Services Development Team“ der Deutschen Börse AG durchgeführt. Die Deutsche Börse AG wurde 1992 gegründet und beschäftigt über 7000 Mitarbeiter weltweit. Der Hauptsitz befindet sich in Eschborn, Deutschland. Das „Index Services Development Team“ ist mit der Wartung und Pflege der Indexberechnungssysteme der Deutschen Börse betraut und hauptsächlich für die Vorbeugung von möglichen Systemausfällen verantwortlich. Zu seinen Aufgaben gehören die regelmäßige Entwicklung, Durchführung und Installation von Systempatches und -updates, Bugfixes, die Durchführung von Systemtests und die Systemüberwachung. Wir arbeiten eng mit den Kunden zusammen und bieten einen 24/7-Support an.

5 Projektphasen mit Zeitplanung

(14 Stunden):

- Ist-Analyse 1,0 Stunden
- Soll-Konzept 4,0 Stunden
- Recherche 6,0 Stunden
- Evaluierung 3,0 Stunden

Projektdurchführung(34,0 Stunden):

- Konfiguration des Systems 5,0 Stunden
- Installation 3,0
- Die Erstellung von Websites 18,0 Stunden
- Das Einrichten des Backends 8,0 Stunden

Testen(8,0 Stunden):

- Funktionstest 4,0 Stunden
- Fehleranalyse und ggf. Behebung 4,0 Stunden

Projektabschluss(14,0 Stunden):

- Übergabe 1,0 Stunden
- Erstellen der Projektdokumentation 13,0

Stunden Gesamtzeit: 70,0 Stunden

6 Dokumentation zur Projektarbeit

Lastenheft, Pflichtenheft, Kundendokumentation, Screenshots

7 Anlagen

keine

8 Präsentationsmittel

Laptop und Beamer

9 Hinweis!

Ich bestätige, dass der Projektantrag dem Ausbildungsbetrieb vorgelegt und vom Ausbildenden genehmigt wurde. Der Projektantrag enthält keine Betriebsgeheimnisse. Soweit diese für die Antragstellung notwendig sind, wurden nach Rücksprache mit dem Ausbildenden die entsprechenden Stellen unkenntlich gemacht.

Mit dem Absenden des Projektantrages bestätige ich weiterhin, dass der Antrag eigenständig von mir angefertigt wurde. Ferner sichere ich zu, dass im Projektantrag personenbezogene Daten (d. h. Daten über die eine Person identifizierbar oder bestimmbar ist) nur verwendet werden, wenn die betroffene Person hierin eingewilligt hat.

Bei meiner ersten Anmeldung im Online-Portal wurde ich darauf hingewiesen, dass meine Arbeit bei Täuschungshandlungen bzw. Ordnungsverstößen mit „null“ Punkten bewertet werden kann. Ich bin weiter darüber aufgeklärt worden, dass dies auch dann gilt, wenn festgestellt wird, dass meine Arbeit im Ganzen oder zu Teilen mit der eines anderen Prüfungsteilnehmers übereinstimmt. Es ist mir bewusst, dass Kontrollen durchgeführt werden.