Software-Projekt: Schlussbericht

Praktikant

Remo Koch
Depotstrasse 2
5620 Bremgarten
Tel: +41 79 731 63 62
E-Mail: remo.koch@uzh.ch
Matrikel-Nr: 12-728-291

Arbeitgeber

Contovista AG (https://www.contovista.com/)
Rütistrasse 14
8952 Schlieren ZH

Betreuer

Administrativ
Gian Reto à Porta
CEO
E-Mail: gian@contovista.com

Technisch
Fabio Bernasconi
E-Mail: fabio@contovista.com

Nicolas Cepeda
E-Mail: nicolas@contovista.com

Zeitraum 02.02.2015 - 13.03.2015 (6 Wochen)

1 Aufgabenstellung

Gemäss dem mit der Firma vereinbarten Projektbeschrieb sollte das Projekt folgende Phasen beinhalten:

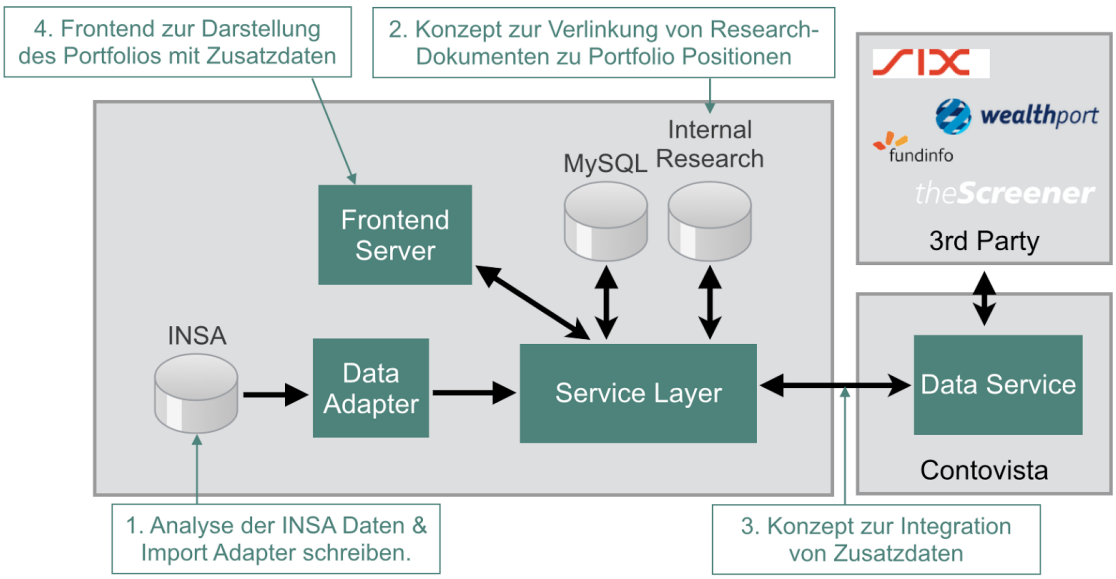

Abbildung 1: Übersicht der Aufgabenstellung

1. Portfolio Daten aus der PMS-Schnittstelle der Bank (*INSA* genannt) sollen analysiert werden, so dass sie in eine intern normalisierte Struktur gebracht werden können um dort weiter verarbeitet zu werden.
   Die interne Struktur (In der Grafik als *Service Layer* bezeichnet) baut auf der bestehenden Plattform auf, welche die Contovista AG bereits entwickelt hat.

2. Sind die Daten normalisiert, können sie mit einer bestehenden internen Datenbank (*Internal Research* in der Grafik) verknüpft werden. Diese Datenbank besteht aus Dokumenten, welche direkt mit den Portfolio-Daten korreliert werden können und die internen Daten somit anreichern. Dazu soll ein Konzept aufgestellt und implementiert werden, wie diese zusätzlichen Daten eingefügt werden können.

3. Damit dieses Projekt tatsächlich eine Neuerung gegenüber herkömmlicher Portfoliomanagements-Software bietet, müssen 3rd-party Datenprovider dazugezogen werden, welche zusätzliche Informationen zu Assets bereitstellen, wie zum Beispiel Neuigkeiten oder erwartete Kursänderungen. Dazu soll ein Konzept erstellt werden, wie diese Daten bei den entsprechenden Providern abgeholt werden sollen und danach in die Bank gelangen, da in diesem Fall der Kunden-Datenschutz ein grosses Thema ist. Dazu sollen 1-2 Providers integriert werden als Proof-Of-Concept. Diese Anbindung wird nach dem Projekt ein wichtiges Thema werden, da man sich hier gut von der Konkurrenz abgrenzen kann.

4. Als letzte und wichtigste Komponente soll ein Frontend entwickelt werden, welche die aggregierten Daten der vorherigen Projektschritte ansprechend anzeigen kann. Dieses Frontend wird als Zusatzmodul zur bestehenden PFM-Suite konzipiert. Hier ist es wichtig, dass diese Darstellungen nicht nur informativ sind für den Benutzer, sondern auch optisch sehr ansprechend sind. Dies ist natürlich auch ein wichtiges Verkaufsargument. Deshalb wird das Hauptaugenmerk auf diese Komponente gerichtet.

2 Arbeitsablauf und Vorgehensweise

Durch eine glückliche Fügung hat ein Kunde (Schwyzer Kantonalbank) am ersten Arbeitstag des Projekts Interesse an diesem Projekt gezeigt und sich bereit erklärt mit uns einen Proof-Of-Concept durchzuführen. Dies bedeutete, dass ich mit Daten arbeiten konnte, welche direkt von der Bank stammten und somit reale Portfolio-Situationen darstellten.

Dadurch, dass das Projekt nun von einem Kunden bezahlt war, mussten die Prioritäten neu gesetzt werden um diesem möglichst schnell etwas zu präsentieren. Dies ergab die im Folgenden beschriebenen Meilensteine.

2.1 MS1: UI Mockups

Wie oben erwähnt wurde das Projekt unerwarteterweise am ersten Arbeitstag durch einen Kunden unterstützt. Zuvor war geplant, einen Proof-Of-Concept zu erstellen mit den in anderen Meetings geäusserten Wünschen des Kunden und ihm diesen am Ende des Projekts zu präsentieren. Da sich diese Vorgehensweise jedoch erübrigte konzentrierte ich mich in der ersten Woche - neben dem Kennenlernen der Firma - hauptsächlich auf die konzeptuelle Interaktion mit der Applikation, sowie mit einem Mockup der dabei entstehenden Seiten. Dies

musste in der ersten Woche erledigt sein, da wir am Ende der Woche bereits einen ersten Kundentermin mit Projekt-Kickoff abgemacht hatten. Dabei wollten wir bereits einen ersten Draft des Aufbaus präsentieren.

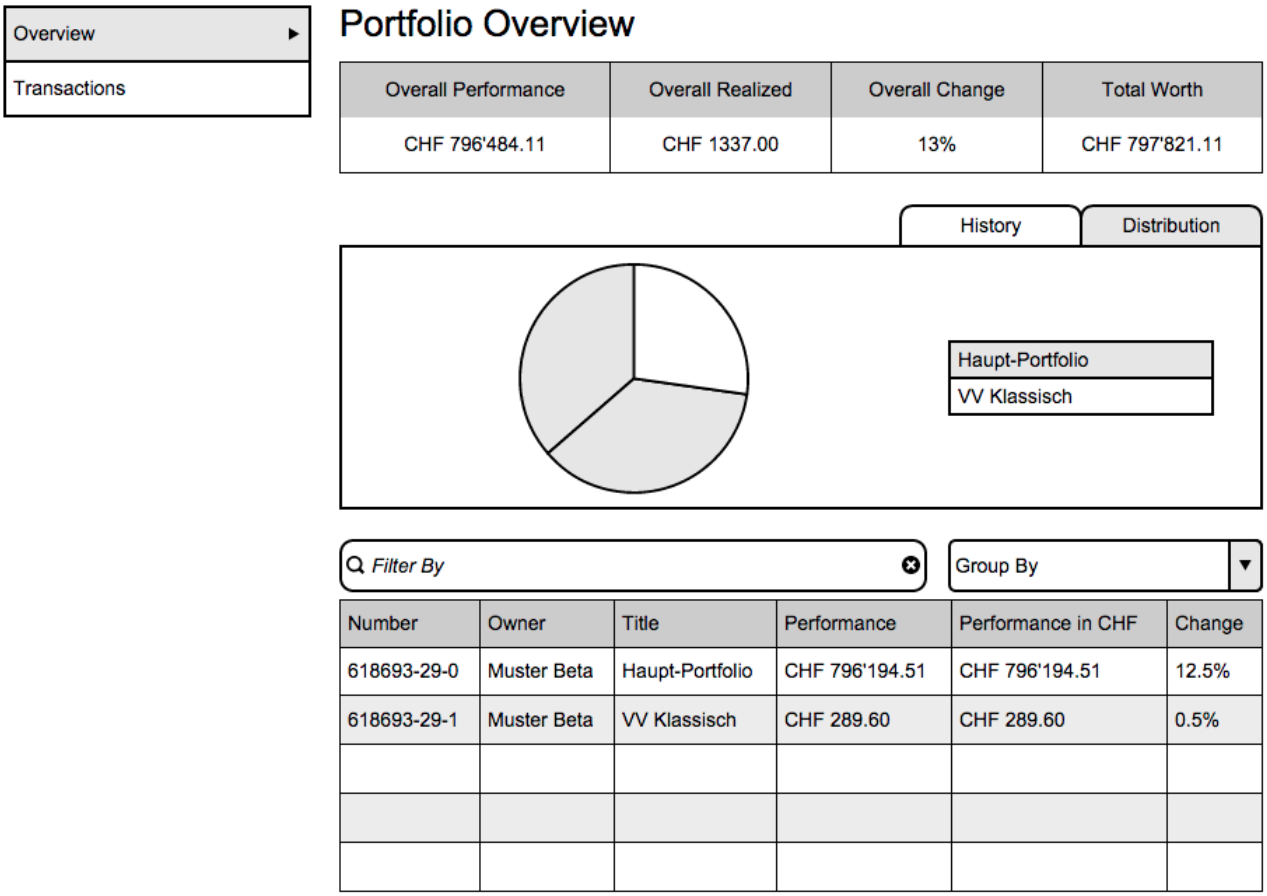

Abbildung 2: Mockup der Einstiegsseite (Portfolio Uebersicht) - In dieser Form wurde es auch dem Kunden präsentiert.

Der Kunde war von dem Konzept begeistert und hat uns den Startschuss für die Prototypenimplementation gegeben.

2.2 MS2: Analyse und Import der INSA-Schnittstelle und grundlegendes UI

Die nächsten vier Wochen war ich damit beschäftigt die INSA-Schnittstelle, welche von der Bank beziehungsweise dem Core Banking Provider (in unserem Fall der Finnova) bereitgestellt wird. Nach einem initialen Datenimport, werden neue Datensätze in regelmässigen Zeitabständen in Batches an den Import-Layer geliefert. Dieser extrahiert die benötigten Felder und importiert diese in unsere interne Datenbank.

Herauszufinden, welche die korrekten Felder sind war einer der herausfordernsten Tasks dieses Projekts, da die INSA-Schnittstelle einerseits extrem breit ist und andererseits die Daten aus der Bank aus Datenschutzgründen nur anonymisiert geliefert wurden. Durch das bereits bestehende Vorwissen meiner technischen Betreuer konnten wir schliesslich einen Konsens finden und die Queries für den Import schreiben.

Der nächste Task bestand darin, die im Mockup gezeichneten Seiten tatsächlich in HTML und JavaScript umzusetzen. Um das UI möglichst schnell auf eine präsentierbare Version zu bringen, habe ich AngularJS, ein Single-Page-Application-Framework von Google eingesetzt.

Dabei habe ich gleichzeitig auch noch die serverseitige Implementation, welche die Daten aus der Datenbank lädt, gewisse Aggregationen durchführt und zum Browser schickt, aufgebaut. Der Server ist in Clojure, ein Lisp Dialekt auf der JVM, geschrieben und exponiert ein RESTful API.

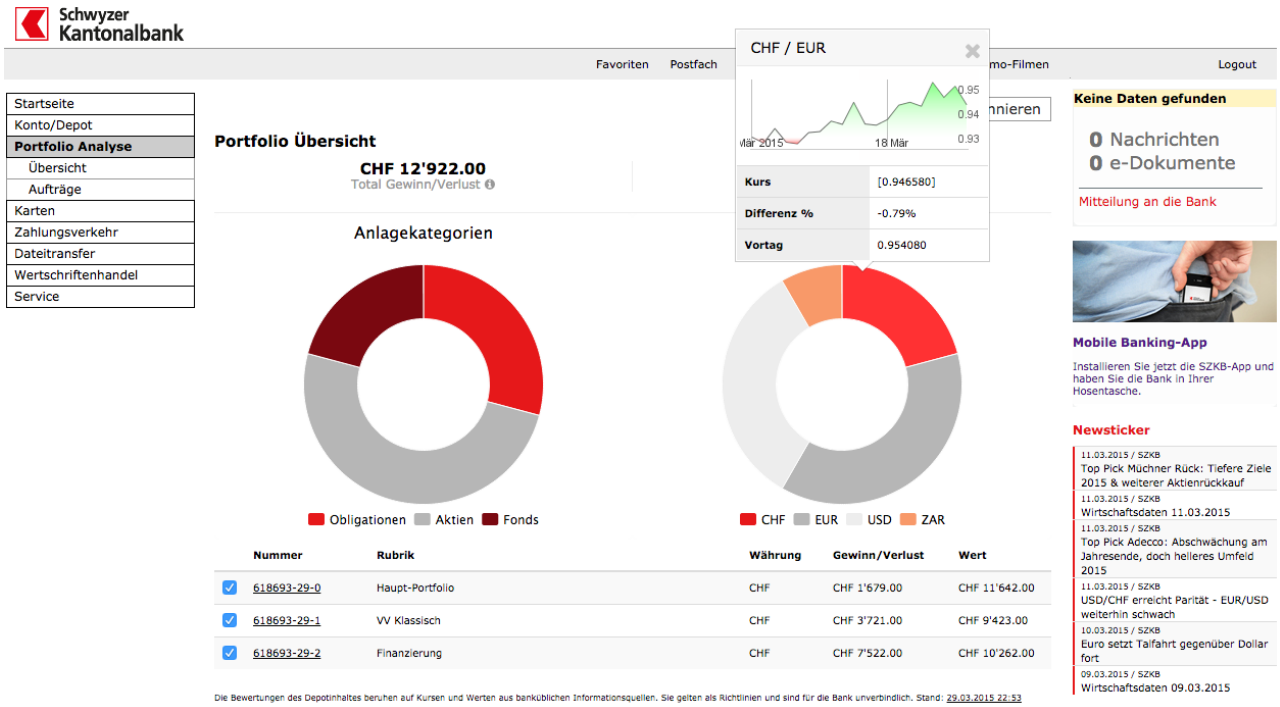

Abbildung 3: Die Portfolio Übersicht als Einstiegsseite, auf welcher die Portfolios des jeweiligen Endkunden aufgelistet werden.

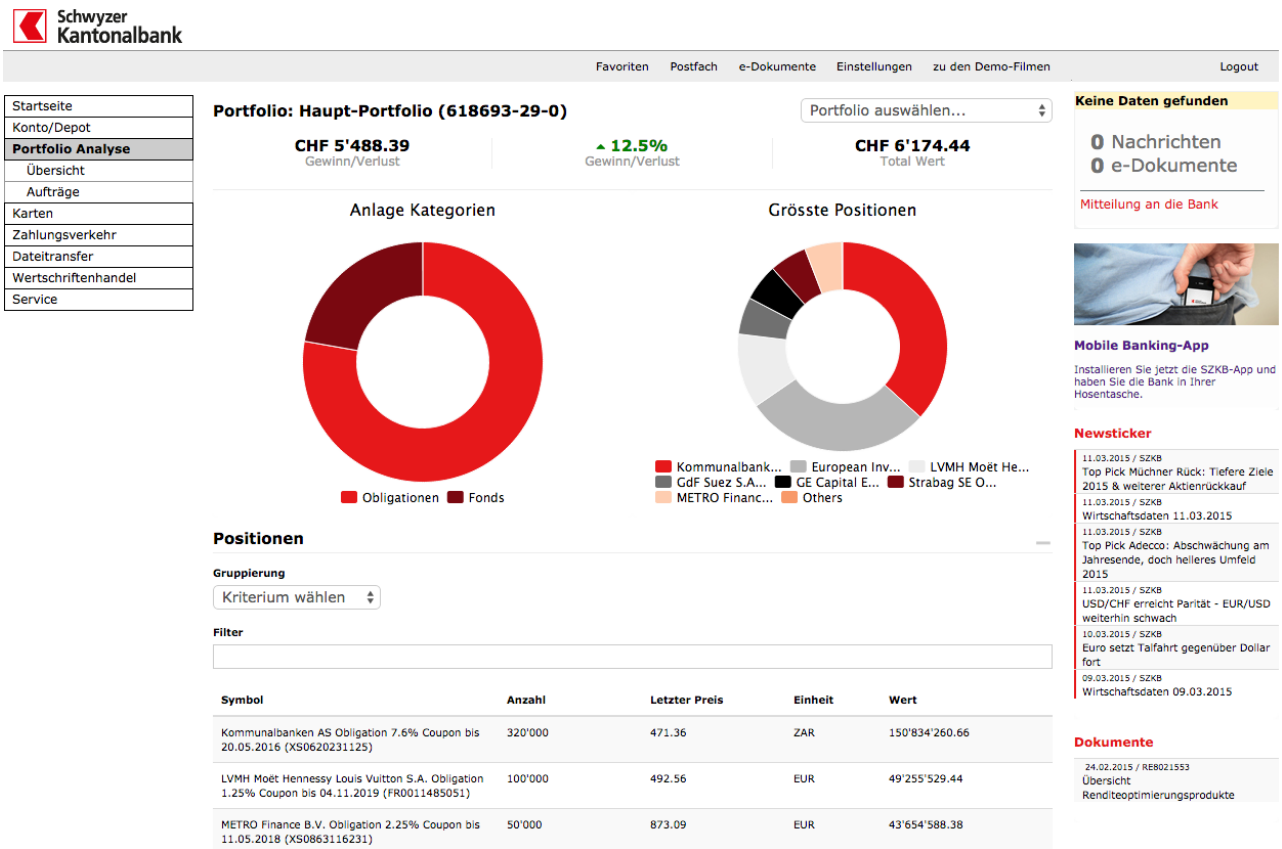

Abbildung 4: Die Detailansicht eines Portfolios

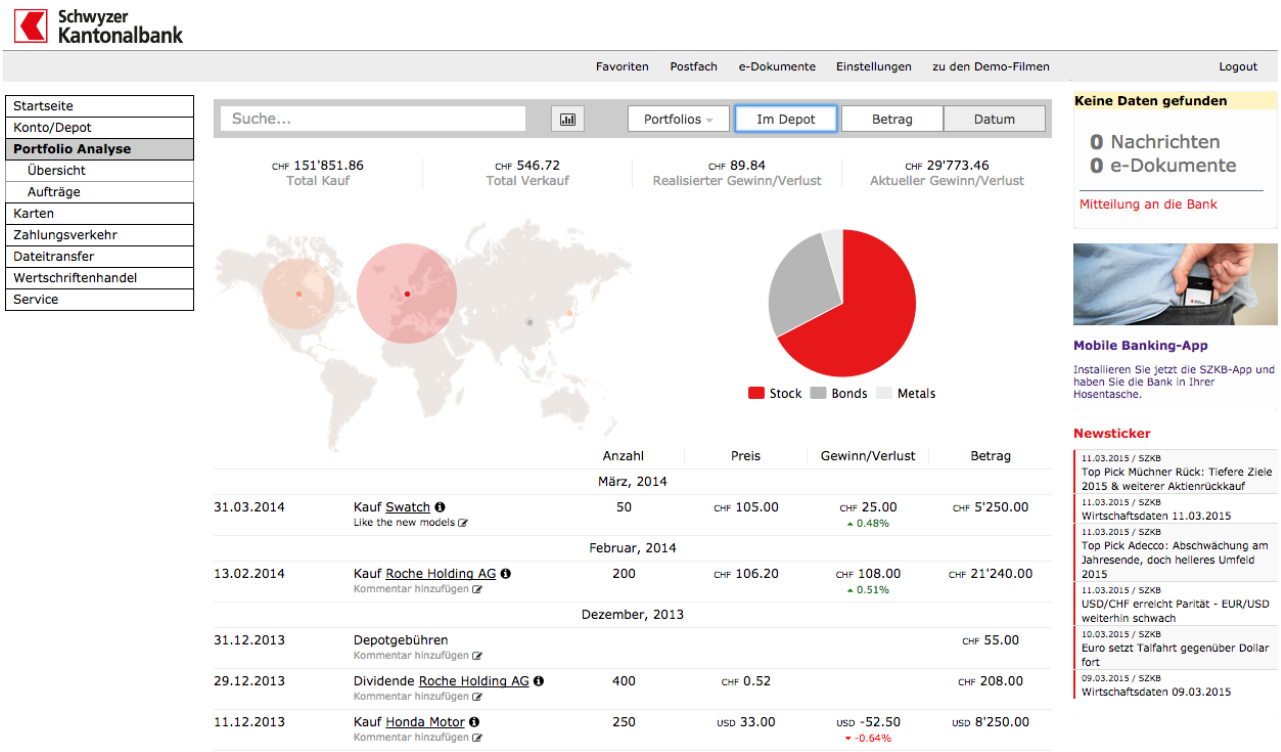

Abbildung 5: Übersicht aller Aufträge (Trades) auf einer Zeitlinie

Am Ende dieser Phase haben wir uns nochmals mit dem Kunden getroffen, und die aktuelle Lösung präsentiert. Dort haben wir die Anbindung der Dokumente, welcher aus der internen Marktforschung (entweder in PDF oder Textform geliefert) diskutiert, sowie die Anbindung einer externen Datenquelle. Dabei ist herausgekommen, dass die Bank bereits mit einem externen Provider kollaboriert, welcher Widgets bereitstellt um aktuelle Marktdaten in einer HTML-Seite einzubinden. So nahmen wir uns vor diese Datenquelle im nächsten Meilenstein zu verwenden.

2.3 MS3: Anbindung *Internal Resarch* und externer Datenquelle

Gerade zu Beginn der letzten Projektwoche haben wir uns mit dem zuvor erwähnten externen Daten-Provider getroffen und mit ihnen unsere Anforderungen diskutiert. Nach diesem Meeting erhielten wir einen Testzugang um diese Widgets in unsere Applikation einzubauen. Dies ging bis auf einige kleinere technische Schwierigkeiten, welche ich während dieser Woche gelöst habe, ziemlich gut.

Weiter habe ich in dieser Woche auch die interne Marktforschungs-"Datenbank" angebunden. Da wir keine Rechte hatten direkt auf diese Datenserver zuzugreifen, habe ich es so gelöst, dass wir in regelmässigen Zeitabständen eine CSV-Datei geliefert bekommen, welche auch über den Import-Layer in die interne Datenbank importiert werden. Die Datei enthält dann zu jedem Eintrag (ein Finanzinstrument) entweder einen Link zum PDF oder einen Textblock, welcher direkt in der Applikation eingebunden werden kann.

Diese letzte Woche war die intensivste Woche im ganzen Projekt, da ich am Montag in der Woche danach bereits die finale Präsentation geplant hatte und natürlich nicht mit unerledigten Tasks dort erscheinen wollte.

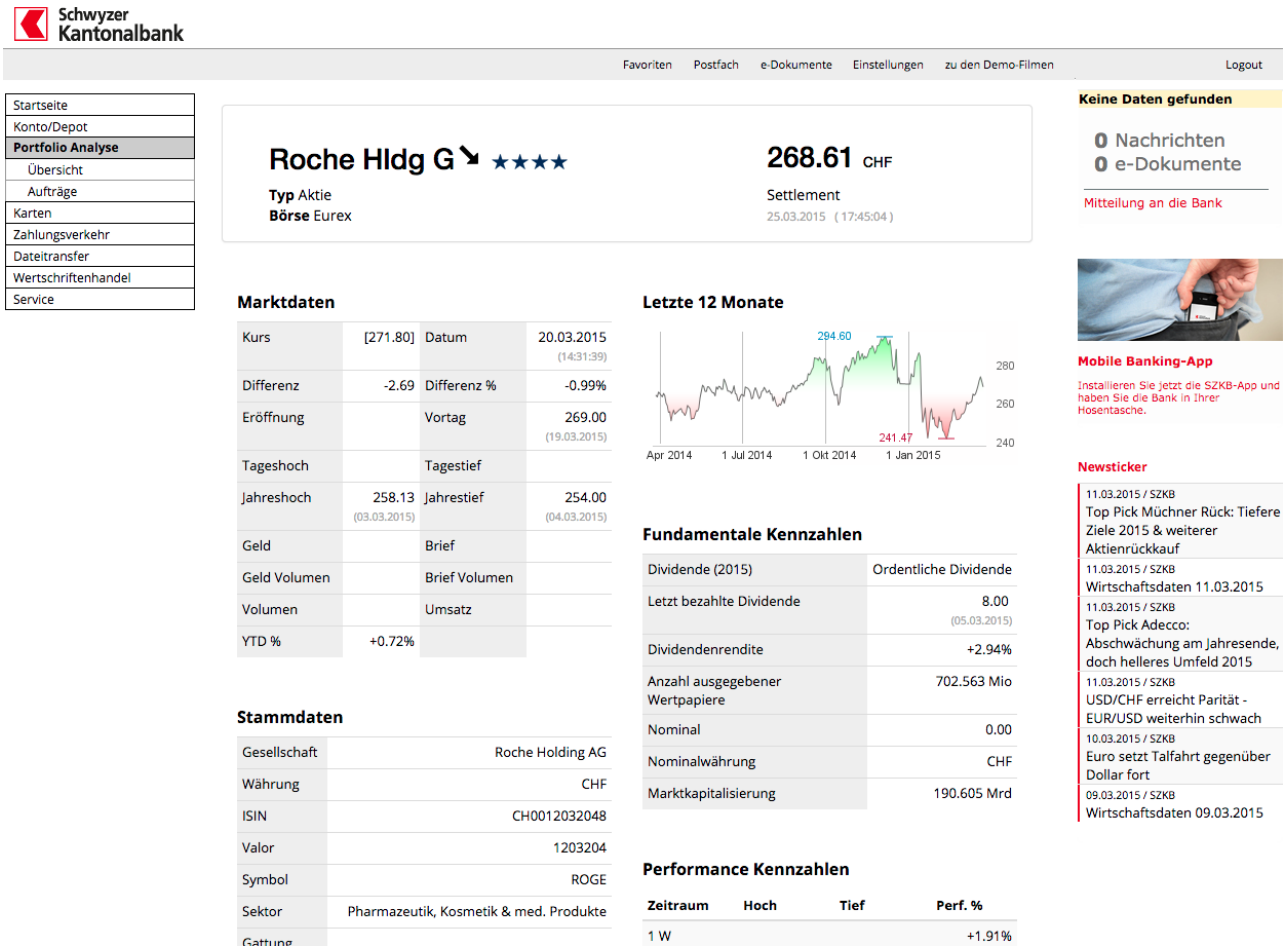

Abbildung 6: Detailansicht eines Titels. Diese Ansicht hat nun bereits die Widget-Ansicht des externen Daten-Providers integriert.

2.4 MS4: Updates und Bugfixes nach Schlusspräsentation

Nach der erfolgreichen Schlusspräsentation des Projekts hatte der Kunde doch noch einige Wünsche bevor wir diesen Proof-Of-Concept abschliessen konnten. Dies waren jedoch nur Kleinigkeiten wie zum Beispiel die Applikation auf Deutsch zu übersetzen (was sehr einfach war, weil die Mehrsprachigkeit bereits eingebaut war).

Diese Änderungen hatten dann noch circa zwei Tage gedauert. Die Applikation ist inzwischen auf einem unserer Servern deployed mit geschütztem Zugang des Kunden.

2.5 Ausblick

Unser Kunde in der Bank wird diesen Proof-Of-Concept nun intern vorstellen und sich Sponsoren suchen, welche eine tatsächliche Umsetzung unterstützen würden und auch prüfen ob die Bankkunden an einer solchen Lösung innerhalb ihres eBanking-Accounts interessiert wären.

Da die Implementation teilweise unter hohem Druck entstanden ist, ist sie weder auf Performance optimiert, noch ist der Code wirklich sauber. Dies müsste bei einer Umsetzung natürlich korrigiert werden.

3 Schlussreflexion

3.1 Beurteilung des Erreichten

Die ursprüngliche Zeitplanung wurde bereits am ersten Tag obsolet, da unser Kunde unerwarteterweise Interesse am Projekt hatte. Dies zwang uns, unsere Planung anzupassen auf eine Weise in der wir schnell auf geänderte Anforderungen und neue Ideen reagieren konnten. Die Planung wurde am Schluss nicht auf das komplette Projekt hinaus gemacht, sondern Ad-hoc immer wieder die nächsten Tage geplant. Dies ist ein eher agiles Vorgehen an ein solches Projekt und hat soweit eigentlich sehr gut funktioniert.

Im Gegensatz zu früheren Firmen ist mir die Agilität in der Contovista AG sehr stark aufgefallen. Da man als Startup noch keine festen Strukturen hat und einen Weg finden muss, sich zu etablieren, plant man die nächsten Schritte sehr kurzfristig und es kann schnell passieren dass die Arbeit vom Vortag am nächsten Tag bereits wieder rückgängig gemacht werden muss. Dies macht die tägliche Arbeit sehr spannend da man nie weiss was passiert, stellt jedoch auch hohe Anforderungen an die Mitarbeiter selbst.

3.2 Persönliche Erfahrung

Die Projektzeit war eine spannende Zeit für mich, da ich wiedermal in eine neue, etwas andere Firma reinschauen konnte. Obwohl ich in den letzten Jahren einige grössere Projekte durchgeführt habe, ist jedes Projekt wieder eine neue Herausforderung. Dies war auch hier der Fall. Man wird zwar mit jedem Projekt schlauer, und doch muss man immer wieder neue Ideen haben um die gewünschten Resultate zu erreichen. Das macht das Software-Engineering spannend.

Die grösste Herausforderung war die Programmiersprache Clojure, welche ich bis anhin noch nie verwendet hatte. Ich hatte mich zuvor zwar bereits mit den Grundlagen der funktionalen Paradigmen auseinandergesetzt, musste aber noch sehr viel dazulernen. Es ist aber auch eine Erfahrungssache. Ich bin mir die imperativen Programmiersprachen schon lange gewöhnt und habe deshalb normalerweise keine grossen Probleme eine neue Sprache dazuzulernen, da die Konzepte immer etwa ähnlich sind. Die funktionale Welt ist so grundlegend anders aufgebaut, dass ich mich zu Beginn gefühlt habe, wie als ich das erste Mal Java gesehen hatte.

Die Zusammenarbeit mit dem Team hat sehr gut funktioniert. Diese war auch nicht nur einseitig, da auch ich sie unterstützen konnte in den Bereichen, in denen ich mehr Erfahrung habe. So hatten wir einen gegenseitigen Austausch, von dem alle profitieren konnten.

Ich werde weiterhin mit der Contovista AG zusammenarbeiten.