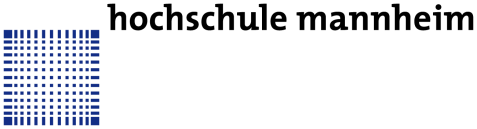
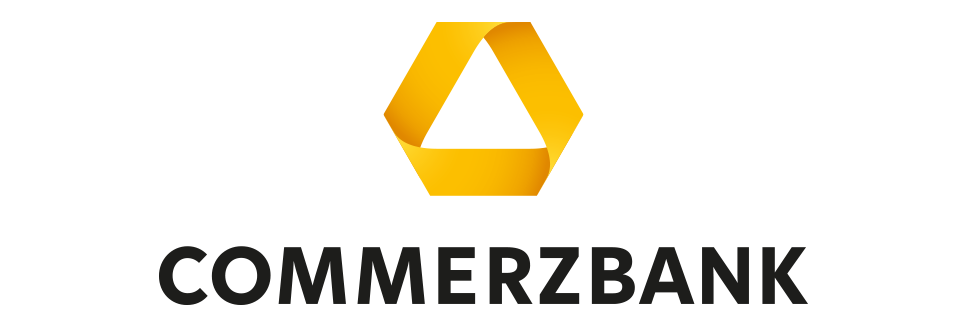
Bericht zum Praktischen Studiensemester

|  |  |
| --- | --- |
| **Zeitraum:** | **01. April 2016 – 30. August 2016** |
| **Studiengang:** | **Informatik – Bachelor of Science** |
| **Betreuer-Firma:** | **Adil El Madbouh** |
| **Betreuer-HS** | **Prof. Dr. Astrid Schmücker-Schend** |
| **Student:** | **Thorsten Müller – 1312756 – SS2016** |

****



|  |  |
| --- | --- |
| **Firma:** | **Commerzbank AG** |
| **Programm:** | **Neue-Wertpapier-Plattform** |
| **Projekt:** | **Retail-Order-Processing** |

**Sichtvermerk der Firma:**

Inhaltsverzeichnis

[1 Commerzbank – Die Bank an ihrer Seite 6](#_Toc459467172)

[2 Tätigkeitsbeschreibungen 7](#_Toc459467173)

[2.1 Willkommen im Praktikum 7](#_Toc459467174)

[2.2 KIT – Vorbereitung 7](#_Toc459467175)

[2.3 FIX-Nachrichten und Testszenarien erstellen 8](#_Toc459467176)

[2.4 Lösungen, Anforderungen Verwaltungen und ein Orientierungsgespräch mit dem Betreuer 8](#_Toc459467177)

[2.5 Traceability Matrix Pattern erstellen 9](#_Toc459467178)

[2.6 Traceability Matrix Makro schreiben 9](#_Toc459467179)

[2.7 Überprüfung der Traceability Matrix 10](#_Toc459467180)

[2.8 11](#_Toc459467181)

[2.9 Fertigstellen des Schnittstellenmakros 11](#_Toc459467182)

[2.10 Traceability Matrix Pflege und den QS Prozess starten 12](#_Toc459467183)

[2.11 Pflichtschulungen absolvieren, sowie anfangen eine Webanwendung zu entwickeln 13](#_Toc459467184)

[2.12 Die Webanwendung weiter entwickeln 13](#_Toc459467185)

[2.13 Lauffähige Webanwendung 14](#_Toc459467186)

[2.14 . Machbarkeitsstudie Tracability Matrix 14](#_Toc459467187)

[2.15 Filterfunktion fertigsstellen sowie die Webanwendung erweitern 15](#_Toc459467188)

[2.16 Fertigstellung der Erweiterung Exceution Report für CSI 16](#_Toc459467189)

[2.17 Lerne das Maven, Spring, React und das Redux Framework kennen 17](#_Toc459467190)

[2.18 Meeting mit dem Webentwickler der Webanwendung 17](#_Toc459467191)

[2.19 Die Eigene Webanwendung in einer .jar Datei 18](#_Toc459467192)

[2.20 Mit Redux ein Formular bauen 18](#_Toc459467193)

[2.21 Week 21 22.08-24.08 – Last Day 19](#_Toc459467194)

[2.22 Week 22 25.08-30.08 – Holiday 19](#_Toc459467195)

[3 Project Descriptions 19](#_Toc459467196)

[3.1 New Security Paper Platform 19](#_Toc459467197)

[3.1.1 Preface 19](#_Toc459467198)

[3.1.2 Abstract 19](#_Toc459467199)

[3.1.3 Principles 19](#_Toc459467200)

[3.1.3.1 Stock exchange process 19](#_Toc459467201)

[3.1.3.2 Smart Order Routing 20](#_Toc459467202)

[3.1.4 Objective 20](#_Toc459467203)

[3.1.5 Including Projects 20](#_Toc459467204)

[3.1.6 Retail Order Processing 20](#_Toc459467205)

[3.1.6.1 Abstract 20](#_Toc459467206)

[3.1.6.2 Objectives 21](#_Toc459467207)

[3.1.6.3 Principles 21](#_Toc459467208)

[3.1.6.4 Including systems 21](#_Toc459467209)

[3.1.6.5 Architecture 22](#_Toc459467210)

[3.2 Traceability Matrix 23](#_Toc459467211)

[3.2.1 Preface 23](#_Toc459467212)

[3.2.2 Excel Pattern 23](#_Toc459467213)

[3.2.3 Excel Macro 23](#_Toc459467214)

[3.2.4 Conclusion 23](#_Toc459467215)

[3.3 Interface Makro 24](#_Toc459467216)

[3.3.1 Preface 24](#_Toc459467217)

[3.3.2 Requirements 24](#_Toc459467218)

[3.3.3 Conclusion 24](#_Toc459467219)

[3.4 Web application 24](#_Toc459467220)

[3.4.1 Principles 24](#_Toc459467221)

[3.4.2 Requirements 25](#_Toc459467222)

[3.4.3 First stage 25](#_Toc459467223)

[3.4.4 Technologies 26](#_Toc459467224)

[3.4.4.1 Maven 26](#_Toc459467225)

[3.4.4.2 Spring Boot 26](#_Toc459467226)

[3.4.4.3 React 26](#_Toc459467227)

[3.4.4.4 Redux 27](#_Toc459467228)

[3.4.4.5 JavaScript ES2016 27](#_Toc459467229)

[3.5 Summary 27](#_Toc459467230)

[3.6 Perspective 27](#_Toc459467231)

[3.7 References 27](#_Toc459467232)

[4 Glossary 29](#_Toc459467233)

# Commerzbank – Die Bank an ihrer Seite

Die Commerzbank AG hat sich seit ihrer Gründung in 1870 dauern vergrößert. Seit der Gründung übernahm sie mehrere Banken vereinigte sich mit anderen, unteranderem mit der Dresdner Bank. Heute ist die Commerzbank eine der führenden Banken in Deutschland, sie hat ca. 1050 Filialen in Deutschland. Des Weiteren hat sie Tochterbanken und Filialen in über 50 verschiedenen Ländern. Mit über 30% von Deutschlands Außenhandel ist die Commerzbank eine der stärksten deutschen Banken. Die Bank hat mit der Comdirect und mit der mBank zwei Tochterbanken, welche in Deutschland und in Polen arbeiten. Die mBank wurde als eine der innovativsten online Banken ausgezeichnet (Commerzbank - Überblick, kein Datum). Die Bank agiert in verschiedenen Bereichen. Darunter Zählen der Business- sowie der Private- Bereich als auch der Börsenhandel. Das Logo hat sich über die Jahre mehrmals verändert. Das aktuelle Logo steht mit den gelben Farben für Fairness und Kompetenz (Commerzbank - Markengeschichte, kein Datum).

Die Kunden der Commerzbank werden in drei verschiedene Bereiche unterteilt. Es gibt jeweils ein Bereich für Private Kunden, für Business Kunden und ein Bereich für Firmenkunden. Die Commerzbank bietet viele verschieden Arten, sein Geld anzulegen, darunter gehören: ein normales Konto, online-Banking, ein Aktien Depot, verschiedene Kreditarten, eine Lebensversicherung (Commerzbank - Privatkunden, kein Datum).

Intern ist die Bank in verschiedene Bereiche eingeteilt. Mein Praktikum ist in dem Bereich GS-IT (Group-Service Information Technology). Dieser Bereich beinhaltet weitere kleinere Abteilungen. Die zwei größten Bereiche im GS-IT sind der ‚Commercial Banking‘, sowie der Bereich ‚Investment Banking‘. Der Bereich des ‚Commercial Banking‘ beinhaltet die Kontoführung, Beratungen für Kunden, etc.. Der Sektor ‚Investment Banking‘ beinhaltet verschieden Aktienangebote für die Kunden, z.B. Geldanlagen an Börsen, mit verschieden hohen Risiken bei der Anlage. In dem Sektor ‚Investment Banking‘ ist letztes Jahr ein neues Programm gestartet. Dieses Programm „Neue Wertpapier Plattform“ beinhaltet verschiedene Projekte und soll die Commerzbank in dem Aktienbereich komplett erneuern. Im Rahmen meines Praktikums bin ich im Projekt „Retail Order Processing“, welches ein Teil aus dem Programm „Neue Wertpapier Plattform“ ist

# Tätigkeitsbeschreibungen

## Willkommen im Praktikum

01.04 & 04.04 – 08.04

Mein erster Tag war sehr interessant. Zuerst traf ich meinen Betreuer, der mich erstmal den anderen Kollegen im Projekt vorgestellt hat. Nach der Vorstellungsrunde wurde mir kurz mein zukünftiger Arbeitsplatz gezeigt. Da ich keinen Commerzbank-Ausweise hatte, welchen ich brauchte um in die Gebäude zu kommen, war meine erste Aufgabe zur Betriebsausweisstelle zu gehen und diesen abzuholen. Zusätzlich hatte ich auch noch keinen Benutzer im Commerzbanksystem, dies bedeutete ich konnte mich weder an einem Computer anmelden, oder einen E-Mail Account haben. Während ich im anderen Gebäude war hatten Kollegen bereits einen neuen Benutzer beantragt, welcher auch sofort Freigeschaltet wurde. Mit meinem freigeschalteten Computerzugriff nutzte ich die restliche Zeit des erstens Tages um mich in das Projekt einzulesen und auch um Software, einen E-Mail Account und eine Telefonnummer zu beantragen, diese können erst mit einem aktiven Benutzer im Commerzbanksystem beantragt werden.

Ich begann mein Praktikum kurz bevor der erste Komponenten-Integration-Test (KIT) startete. Dies war eine gute Möglichkeit um mich mit der Materie von dem Projekt auseinanderzusetzen und mehr Informationen über das Projekt zu bekommen. Meine erste Aufgabe für den KIT war, dass ich FIX-Nachrichten erstelle, welche für den ersten KIT benutzt werden können. Damit ich FIX-Nachrichten erstellen könnte, brauchte ich zum einen mehr Fachwissen über die FIX-Nachrichten zum anderen benötigte ich ein Tool, welches bestimmte Teile der FIX-Nachricht berechnet. Damit ich Wissen über FIX-Nachrichten bekam, recherchierte ich im Internet, startete die Leistungsbeschreibung zu lesen und sprach vermehrt mit Kollegen. Über die Gespräche mit Kollegen hatte ich erfahren, dass es einige Punkte gibt auf die ich bei der Erstellung der FIX-Nachrichten achten musste. Die Gespräche waren außerdem eine gute Möglichkeit die Kollegen besser kennenzulernen und herauszufinden welches System im Projekt wie weit in der Entwicklung ist.

## KIT – Vorbereitung

11.04 – 15.04

In meiner zweiten Woche las ich weiter die Leistungsbeschreibung, damit ich mit dem Projekt weiter vertraut wurde. Nachdem ich mit dem Projekt mehr vertraut wurde und weil eine für den KIT zuständige Kollegin krank wurde, war ich mehr im KIT integriert als zu Beginn vorgesehen. Im Rahmen des KIT-Supports ich sollte verschiedene Test-Szenarien erstellen. Nach der Absprache mit Testern, wusste ich, dass es drei Testarten gibt, welche alle im KIT enthalten sein müssen. Es gibt den

* Verbindungstest Ein Test ob ein System mit dem nächsten System kommunizieren kann (wird auch Hello World Test genannt)
* Streckentest Ein Streckentest beinhaltet beim Test verschiedene Systeme, die gesendete Nachricht muss im Original ankommen
* End 2 End Test Beinhalt einen kompletten Workflow mit Beispielnachrichten

Die Testszenarien zu erstellen ging schneller als ich dachte, allerdings musste ich auch noch die dazugehörigen FIX-Nachrichten erstellen. In einem Gespräch mit einem Kollegen ich bekam ich langsam ein Gefühl, wie groß und wie kompliziert das Projekt wirklich ist. Es gibt den Auftragsmanager (eines von fünf Systemen), welches auf eine erhalten FIX-Nachricht mit einer von drei FIX-Nachrichten antwortet. Da ich immer noch die FIX-Nachrichten für den KIT erstellen musste, sprach ich einen Kollegen aus dem Testteam an. Er hatte mit jruby ein Tool geschrieben, welches FIX-Nachrichten aus einer Excel-Tabelle erstellt. Er half mir sein Tool zu installieren und warnte mich vorab, dass er noch Änderungen machen müsste. Die Änderungen kamen von dem Wechsel vom FIX4.0 ins aktuellere FIX5.0 Format.

## FIX-Nachrichten und Testszenarien erstellen

18.04 – 22.04

In dieser Woche bekam ich verschiedene Online-Schulungen, welche ich absolvieren musste. Diese Schulungen muss jeder Mitarbeiter alle zwei Jahre wiederholen und bestehen. Die Schulungen sind dafür man ein Gefühl mit ob Mitarbeiter geheime Daten ausspionieren möchte, andere gehen darum wie man sich am Arbeitsplatz richtig ernährt, richtig sitzt und fit bleibt. Der zweiwöchige KIT startete in dieser Woche und hatte alle zwei Tage eine Telefonkonferenz mit allen beteiligten Firmen im Projekt. Für den KIT war es natürlich auch von Nöten, dass wir alle Testfälle dokumentieren. In der Commerzbank wird dafür der HP Quality Center benutzt. Dies ist eine von HP speziell für Testzwecke entwickelte Software. Mit dem HPQC kann jeder Tester in einem Projekt Testfälle anlegen und verwalten, jeder Tester hat eine spezifische Testrolle im HPQC. Je nach Testrolle kann man Grafiken erstellen lassen, Status von Testfällen verändern, etc.. . Den Aufbau von den Testrollen erfuhr, da ich mich mit einem Tutorial über den HPQC vertraut machte. Der Testaufbau im HPQC ist in Testszenarien, Testfälle und Testschritt unterschieden, damit man am einfachsten sehen kann an welchem Punkt ein Testszenario fehlschlägt. Als mir ein Kollege die Benutzung seines Tools mittels jruby erklärte war ich in der Lage die FIX-Nachrichten zu erstellen. Allerdings musste ich diese erst überarbeiten, da meine Testszenarien nicht zu dem Prinzip vom HPQC gepasst haben und die Testfälle aber im HPQC dokumentiert werden sollten. Damit ich meine Testszenarien richtig designe sprach ich das Testteam an. Diese erklärten mir, dass ein Testszenario mehrere Testfälle haben und jeder Testfall hat ca. zwischen drei und acht Testschritten. Mit diesem Wissen hatte ich meine Testszenarien neu erstellt und zur Kontrolle noch einmal von einem Tester kontrollieren lassen, aber zum Glück hatte er nichts zu beanstanden.

## Lösungen, Anforderungen Verwaltungen und ein Orientierungsgespräch mit dem Betreuer

25.04 – 29.04

Weil es keine wirklich klare Übersicht gab, welche aussagt was für Anforderungen und Lösungen gibt es und welche Lösungen welche Anforderungen abdecken. Um dies darzustellen gibt es in der Commerzbank den Enterprise Architekt (kurz: EA). Nachdem ich in den letzten Wochen bereits erste Lösungen für das Projekt eingetragen und gemappt (die Lösungen mit der Anforderung verbunden) hatte kam in dieser Woche ein großer Blog an Anforderungen, welcher noch nicht im EA vorhanden war. Nachdem ich am meine ersten Anforderungen von einer Liste übertragen hatte musste ich dieses mal die Lösungen, mit der Beschreibung aus der Leistungsbeschreibung herauslesen. Mit dieser Aufgabe bekam ich einen noch besseren Einblick in das Projekt und von wem (welche Firma / welches System) welche Anforderungen löst. Auf der anderen Seite war dies nur eine copy-paste Aufgabe. Da die Aufgabe für das Projekt sehr wichtig war, beendete ich diese noch aber schon suchte zwischendrin schon das Gespräch mit meinem Betreuer. Mein Problem mit der Arbeit war, obwohl diese wichtig war, war der Schwierigkeitsgrad so gering, dass die Arbeit jeder ausführen konnte. Gerade da ich diese Arbeit auf lange Sicht nicht aushalten würde habe ich meinen Betreuer auf das Problem angesprochen. In dem Gespräch erzählten wir uns auch was wir für Erwartungen haben und auf was wir achten sollten. Das Ergebnis aus dem Gespräch war, dass wir beide nach einem zweiten Bereich, welcher mir mehr liegt, schauen, aber mein Hauptaugenmerk blieb erstmal die Verwaltung von den Anforderungen und den Lösungen.

## Traceability Matrix Pattern erstellen

02.05 – 06.05

Nach meinem Gespräch mit meinem Betreuer, war meine Aufgabe diesmal ein Excel Sheet zu erstellen, womit man schnell und auf einen Blick sehen kann welche Anforderung von welcher Lösung umgesetzt wird, welche Lösungen bereits entwickelt sind und welche Lösungen bereits geliefert sind oder sich bei der Lieferung verzögert haben. Mein Betreuer wusste noch ein anderes Tool, dass diese Aufgabe übernehmen könnte, aber für dieses hatte ich noch keinen Zugriff und zudem dauerte es lange bis man sich in das Tool eingearbeitet hat. Da das Projekt die Traceability Matrix schnell benötigte benutzten wir zum Anfang Excel. Da ich noch nie anwendungsvolle Aufgaben in Excel erledigt hatte, musste ich viel im Internet recherchieren und ausprobieren um das Sheet mit allen Funktionalitäten, wie Gruppieren, Filtern, Titelzeile einfrieren oder Auswahlfunktionen für Zellen zu erstellen. Nachdem ich einige grundlegende Funktionalitäten konnte richtete ich mich dem Design des Excel Sheets. Nachdem von einem Kollegen erfuhr, wie der Export aus dem Enterprise Architect ausschaut, konnte ich das Excel Sheet besser gestalten. Der Export sollte in dem Excel Sheet vorhanden sein, aber dabei nicht die Übersicht beeinträchtigen. Gegen Ende der Woche probierte ich weiter das Excel Sheet weiter zu vereinfachen, aber die Grundlage war bereits gut geworden. Am Freitag besprach ich die Tabelle mit einem Betreuer und einem Kollegen. Nach ein zwei kleinen Strukturänderungen war das Pattern für die eigentliche Aufgabe fertiggestellt.

## Traceability Matrix Makro schreiben

09.05 – 13.05

Nachdem ich im EA alle, bis zu diesem Zeitpunkt, Anforderungen sowie Lösungen eingetragen war es meine Aufgabe auch diese zu Pflegen. Direkt am Anfang der Woche bekam ich die Aufgabe mehrere Lösungen zu ändern. Da die Anforderungen in Gruppen geordnet waren, dachte ich es wäre praktisch die Lösungen auch zu Gruppieren, in welche die zu den Anforderungen passen. Nach ein paar Gesprächen mit Kollegen sah ich davon ab, da ich bei jeder Änderung die Nummerierung für die Lösungen anpassen und eventuell die Lösung neu verschieben müsste, dadurch wüsste keiner mehr im Projekt was welche Lösung war und müsste sich beim jeden Öffnen der Tabelle wieder neu orientieren, deswegen lies ich die Idee. In unserem Projekt gab es verschiedene Stufen, diese waren wiederum in Pakete unterteilt. Für das Paket 2, das folgende Paket, fehlten noch viele Lösungen in dem Excel Sheet und in dem EA, diese hatte ich am Anfang der Woche hinzugefügt. Zusätzlich hielt ich eine Telefonkonferenz, mit Kollegen, um den anderen Firmen das neue Excel Sheet zu erklären. In der Konferenz erklärte ich, mit Hilfe von Kollegen, wie die Traceability Matrix aufgebaut, welche Funktionalitäten sie bietet und an welchen Stellen Änderungen gemacht werden dürfen. In der Konferenz hörte ich viel positive Kritik aber auch, dass mehrere Zuordnungen sowie Liefertermine nicht stimmen und angepasst werden müssen. Weil es eine Firma gab die zum Teil die Funktionalitäten der anderen Firma nur testen müssen, dies aber trotzdem als Lösung im System dargestellt werden soll, musste ich mir eine Verbindung zwischen diesen Lösungen überlegen. Damit es leicht erkenntlich ist, entschied ich mich zweimal die gleiche Lösung zu erstellen. Allerdings war eine Lösung von der einen und die andere von der anderen umgesetzt und dadurch zu unterscheiden. Dies machte ich anschließend noch für eine ganze Menge mehr Lösungen zwischen diesen beiden Firmen. Danach richtete ich mich meiner eigentlichen Aufgabe, diese war es zu ermöglichen den Export aus dem EA mittels Mikro zu integrieren. Der Export aus dem EA war eine große Tabelle welche pro Lösung (in den Zeilen) ein „X“ für jede umzusetzende Anforderung (in den Spalten). Meine Aufgabe war die Verbindungen zwischen Lösung und Anforderung zu suchen und in einer Spalte anzuzeigen. Mein Betreuer gab mir noch ein paar Tipps um das Excel Sheet weiter zu vereinfachen, zugleich wollte er auch ein zwei kleine Schönheitsveränderungen. Um alle Anforderung von einer Lösung anzuzeigen suchte ich zuerst in den Excel Funktionen, allerdings gibt es dort nur eine Funktion, welche nach einer Verbindung zwischen Lösung und Anforderung aufhört. Da es keine Funktion gab, entschloss ich mir die Makrosprache VBA (Visual Basic for Application) anzuschauen. Um das Makro zu schreiben suchte ich zuerst im Internet Beispiele um mich mit der Sprache vertraut zu machen. Nachdem ich mit dieser speziellen Sprache etwas vertraut worden war, entwickelte ich schritt für schritt verschiedene Makros. Am Ende der Woche führte ich alle einzelnen Makros erfolgreich zusammen. Das Makro konnte die nötigen Anforderungen kopieren, Lösungsnamen kürzen (im Export aus dem EA sind die Ordnernamen noch mitenthalten) und die Tabelle noch etwas stylen. Als ich diese Aufgabe gelöst hatte wurde die erste Version der Traceability Matrix an alle im Parteien versendet. Kurz danach bekam ich die Information, dass es sehr wichtig ist, dass die Lösungen ein Tracing bekommen. Damit die Nachverfolgung gewährleistet wird musste ich mir ein Konzept ausdenken, welches die Nachverfolgung sinnvoll darstellt. Des Weiteren sollte es noch möglich sein die Lösungen nach Bereichen zu sortieren zu können.

## Überprüfung der Traceability Matrix

17.05 – 20.05

In dieser Woche musste ich leider feststellen, dass das Excel Sheet für Dropdownliste mit den möglichen Eingabewerten gelöscht wurde. Der Fehler war wohl anscheinend, dass eine Excel Datei im „xlsm“ Format nicht im alten Excel bearbeitet werden soll, da so Konvertierungsfehler entstehen. Mir blieb nichts anderes übrig als die verlorenen Inhalte wieder neu zu entwickelten. Die Matrix war danach wieder gut, allerdings gab es neben den Lieferterminen auch Nachlieferungen, welche noch nicht sichtbar waren. Also entschlossen wir uns die Matrix erneut zu erweitern. Zum einem ermöglichte ich Nachlieferungen im Liefertermin, des anderen fügten wir Releasenotes hinzu. Dieses beinhalteten den genauen Lieferzeitpunkt, die Liefernummer, eine Referenz zur Anforderung und eine Referenz in das Lösungsbuch (hier sind alle Lösungen beschrieben) damit die Tester alle gelieferten Lösungen testen können. Bei einer Firma mussten ein Kollege und ich leider die Releasenotes durchgehen und Ergänzen. Die Lieferbeschreibungen waren leider so schlecht, dass wir zum Teil nicht erkennen könnten welche von Ihren Lösungen in der Beschreibung gemeint war. Nachdem wir Firma bezüglich der Fehlenden Verbindung kontaktiert hatten wurde festgestellt, dass sogar mehrere Lösungen in der Traceability Matrix aber auch in den Releasenotes falsch Verbunden wurden. Dies verbesserten nach gründlicher Absprache. Da ich mich Verantwortlich für die Matrix fühlte, entschied ich mich einen „Quality Process“ durchzuführen. Ich ging einfach jede Verbindung zwischen Lösung und Anforderung noch einmal durch und hinterfragte anschließend alles was nicht zu 100% gestimmt hatte. Dies brachte mich dazu, dass ich noch mehr Verständnis von den Lösungen und von den Anforderungen hatte, aber auch gab es tatsächlich noch ein paar Fehler. Alles was mir aufgefallen war besprach ich mit einem Kollegen, die Fehler verbesserten wir in der Traceability Matrix, sowie auch im EA.

## 

23.05 – 27.05

Am Anfang der Woche hatte ich ein Meeting zwei Kollegen. Einer von ihnen war aus dem Testteam und wollte, dass wir das Testteam in das Excel Sheet integrieren. Sein belangen war lediglich, dass die Testfälle im HPQC und der Traceability Matrix schneller sichtbar werden. Hierfür reichte es ihm eine Spalte mit dem genauen Bereich der Lösung hinzuzufügen. Ich nahm mich seinem Anliegen an und überlegte mir einen Algorithmus, der jeweils den genauesten Bereich einer Lösung in einer Spalte darstellt. Der Algorithmus war recht einfach, in einem Bereich von vier Spalten, schaue ob die nächste Spalte einen Wert hat (solange du im Bereich bleibst), falls ja gehe in die nächste Spalte und wiederhole die Abfrage, falls nicht kopiere diesen Wert in die für die Tester vorgesehene Bereiche. Da alle Lösungen, welche in den EA eingefügt habe den gleichen Präfix haben wie die Anforderung war meine Aufgabe dieses Präfix zu ändern. Auch wenn es leider keine „Search and Replace“ Funktion im EA, gab es immerhin eine Funktion wo ich mir alle Einträge in einem Ordner untereinander anzeigen lassen und bearbeiten kann. Dies war eine Aufgabe, die nicht viel Spaß machte aber ich konnte sie gut verstehen. Es ist in der Tat verwirrend, wenn man das gleiche Präfix für Lösungen und Anforderungen hat. Nach der Änderung hatte ich die Präfixes „SQL“ für Solution (Lösung) und „WPP“ für Wertpapierprojekt (Anforderung) erfolgreich im Projekt integriert. In Mitte der Woche war ich in einem Meeting, wo meine nächste Aufgabe besprochen wurde. Die Aufgabe war diesmal zwei große Excel Dateien mittels Makro zu kleineren Dateien zu extrahieren. Das Problem war, dass es für die Schnittstelle aus und zu dem Hauptsystem zwei Excel Sheets gab, welche allerdings durch die Größe eher etwas unübersichtlich sind. Deswegen sollte ich mittels Makro für jede Schnittstelle, insgesamt fünf Stück, die nötigen Informationen in ein eigenes Excel Sheet kopieren. Mein Problem dabei einmal herauszufinden welcher Eintrag für welche Schnittstelle wichtig ist aber auch welche zusätzlichen Sheets, gekennzeichnet mit unterschiedlichen Einträgen in der Haupttabelle, kopiert werden müssen. Mit dem ersten Problem half mir ein Kollege, ich musste lediglich per Makro eine Spalte auslesen.

## Fertigstellen des Schnittstellenmakros

30.05 – 03.06

Nachdem ich mit Makro etwas haderte entschied einen neuen Ansatz. Ich wollte ursprünglich alle nötigen Einträge aus den Hauptdateien rauskopieren aber mein Makro hatte Probleme bei manchen Einträgen zu erkennen ob diese in die neue Tabelle sollen oder nicht. Da ich meiner Ansicht nach an diesem Problem zu lange ausprobierte, probierte ich den umgekehrten Weg. Ich kopierte mir zuerst die Tabelle und löschte alle Einträge, welche nicht in dieses spezielle Schnittstellendokument gehörten. Aus dem Versuch wurde Praxis, ich war mit diesem neu gestalteten Makro sehr schnell fertig geworden. Allerdings hielt die Freude nicht lange. Der Share Point, ist wie das Moodle von der HS Mannheim bloß Projektbezogen, bietet Online-Bearbeitung an, die Datei wird solange für jeden gesperrt (man kann sich nur eine Kopie anzeigen lassen) bis der User die Bearbeitung abgeschlossen hat. Wir bemerkten, dass die Zugriffskontrolle für die beiden Masterdateien, für die Schnittstellen, nicht mehr funktionierte. Das war bei diesen Dateien Problematisch, da diese Dateien fast den ganzen Tag über in Bearbeitung von jemand anderen waren. Der einzige Grund konnte sein, dass sich das Format von „xls“ auf „xlsm“ geändert hatte, ansonsten gab es keine Erklärung für dieses Problem. Wir lösten das Problem indem ich ein eigenes Verzeichnis im Share Point anlegte und dort immer wieder die Masterdateien zum Erstellen der einzelnen Schnittstellendokumente kopierte. Das Problem löste ich für mich, indem ich einfach eigenes Directory erstellte und die beiden Masterdateien in das neu erstellte Directory kopierte und dort arbeite. Dies hatte als Konsequenz, dass ich zukünftig die Dateien neu kopieren musste und die Struktur der Masterdateien überprüfen müsste. Bevor ich das neue Makro überprüfen lassen konnte, löste ich noch ein paar Bugs in dem Makro. Zusätzlich baute ich ein, dass das Makro unnötige Zellen löscht, damit der Fokus auf dem wesentlichen bleibt, und dass das Makro nach links oben scrollt. Ich zeigte einem Kollegen, der für die Schnittstellen zuständig ist, die Makros und er war voll und ganz zufrieden. Während ich am entwickelten von dem Makro fragte ich bei meinem Betreuer nach einem Zwischenfeedback, welches ich bei einem gemeinsamen Mittagsessen bekam. Später fragte ich die Kollegen von der Accenture GmbH um Hilfe bei der Installation von WebMethods. WebMethods ist eine Software speziell für Schnittstellen. Mit dieser Software ist es möglich Warteschlangen zwischen System zu entwickelten, die Nachrichten zu öffnen und zu bearbeiten vor der Weiterleitung zum Zielsystem oder um WebServices zu erstellen. Da mich das Thema in der Schnittstelle interessiert hatte war mein Hintergedanke, dass ich möglicherweise im Schnittstellenbereich das Projekt weiter unterstützen kann.

## Traceability Matrix Pflege und den QS Prozess starten

06.06 – 10.06

In dieser Wocher hatte die Traceability Matrix einiges an Arbeit für mich. Es gab eine Menge von Lösungen, die im EA sowie in der Traceability Matrix hinzugefügt werden mussten. Ab der letzten Version nur mit der Schwierigkeit, dass ich auf die Struktur genau aufpassen musste. Die Struktur der in der Excel Datei musste genauso sein wie im EA, da ansonsten das Makro zwar funktioniert aber die Attribute von den Lösungen in der Zuordnung nicht mehr stimmen. So hätte es passieren können, dass Lösungen einen späteren Lieferzeitpunkt bekommen als nötig wäre, dass manche Lösungen schon geliefert sind, etc.. Eine Partei hatte noch keine Lösungen in der Matrix, daher kam der größere Blog an Lösungen zum Hinzufügen. Dabei war leider die Datei, aus welcher ich die Lösungen herauslesen musste, in einer unpraktischen Struktur. Dadurch kam es dazu, dass die Aufgabe länger gedauert hat und dass ich beim Überprüfen ein Fehler festgestellt hatte und daher die Aufgabe direkt nochmal machen musste. Ich hatte in den Wochen bis hierher schon ein bisschen über den internen Qualitätssicherungsprozess (QS) gehört. In dieser Woche war es für mich Zeit, dass ich mich näher mit diesem Prozess auseinandersetzte. Der QS Prozess muss von jedem projektspezifischen Dokument durchlaufen werden, dabei sind immer mehrere Prüfer anzugeben. Nachdem ein Dokument im QS Portal eingestellt wurde, gibt es eine gewisse Zeitspanne in der die Prüfer Zeit haben Kommentare an den Autor zu senden. Danach hat der Autor nochmal Zeit auf die Kommentare einzugehen und diese gegebenfalls mit den Prüfern zu klären. Nachdem alle Kommentare beseitigt wurden gibt es nocheinmal eine Zeitspanne in der die Prüfer das Dokument nochmals auf Fehler überprüfen, bevor es abgenommen ist. Die einzelnen Schnittstellendokumente, welche ich mittels Makro erstellte, mussten in den QS Prozess. Damit ich diesen einmal kennenlerne und weil ich die Dateien erstellt hatte, war es auch meine Aufgabe den QS Prozess zu starten. Am Ende der Woche hatte ich noch einmal ein ausführliches Gespräch mit meinem Betreuer. In dem Gespräch gab ich ihm auch Feedback, wie ich das Praktikum bis jetzt wahrnahm. In dem Gespräch erfuhr ich, dass er bereits nach einer Aufgabe für mich schaut aber noch nichts passendes gefunden hatte aber auch bekam ich die wichtige Information, dass ich nicht auf eine Aufgabe von ihm warten soll, sondern auch schon selbstständig überlegen und schauen wie ich das Projekt weiter unterstützen könnte.

## Pflichtschulungen absolvieren, sowie anfangen eine Webanwendung zu entwickeln

13.06 – 17.06

Am Anfang der Woche hatte ich wieder verschiedene Pflichtschulungen zu absolvieren. Diesmal waren die Thema Geldwäsche, Sanktionen, sowie Sanktionen von der OFAC (Office of Foreign Assets Control). Da ich bislang als einziger die Traceability Matrix bedienen konnte entschloss ich mich ein ausführliches ‚HowTo‘ zu schreiben. Damit meine Kollegen die Matrix richtig benutzen können, musste sie natürlich auch wissen wann und wo man eventuell Anpassungen im Makro macht. Nachdem ich damit fertig war erfuhr ich von einer größeren Aufgabe, die mir viel Spaß bereiten würde. In einem Meeting erfuhr ich, dass Pdv einen REST WebService entwickelt hatte. Dieser Web Service ist an Decide angeschlossen wodurch es möglich ist den ersten Teil eines realistischen Trades zu Simulieren. Mit dem WebService konnte man eine neue Order anlegen, diese modifizieren oder stornieren, allerdings funktioniert die modifizieren und die stornieren Funktion nur solange die Order noch nicht an eine Testbörse weitergeleitet wurde. Meine Aufgabe war es jetzt eine Webanwendung zu schreiben, mit der es möglich ist Orders zu erstellen, den Status einer Order zu sehen, eine Order zu modifizieren oder zu stornieren. Die ersten Zeilen code musste ich noch im Windows Explorer schreiben, da das Projekt nicht für die Entwicklung vorgesehen ist. Glücklicherweise hatten ein zwei Kollegen einen portablen Notepad++, den ich über Umwege auch benutzen konnte. Nachdem ich eine erste, ganz einfache, Version von der web app entwickelt hatte testete ich diese und bekam beim Zugriff auf den REST WebService von Decide eine CORS (Cross-Origin Resource Sharing) error Meldung. Nach längerer Diskussion mit den Kollegen von Accenture war die Entscheidung, dass sie meine Webanwendung auf ihrem hosten. Damit war dieses Problem gelößt, allerdings bekam ich keinen Serverzugriff und musste das Verzeichnis, in dem sich die Webanwendung befand, zu einer zip Datei komprimieren und einem Kollegen schicken damit er die Datei hochladen kann.

## Die Webanwendung weiter entwickeln

20.06 – 24.06

Am Anfang der Woche ist mir ein Bug in dem von mir benutzten XMLHttpRequest aufgefallen. Der Fehler war, nachdem ich die empfangenen Daten verarbeitet hatte wurde die Website neu geladen, wodurch die Empfangenen Daten verloren waren. Zusätzlich wollte mein Betreuer die Webanwendung in dieser Woche vorstellen. Weil die Kollegen von der Accenture #GmbH# Angestellte haben, welche Webentwickler sind (in anderen Projekten) entwickelten sie auch eine Webanwendung, die präsentierbar war. Damit wir eine Webanwendung präsentieren konnten, half ich den Kollegen bei der Fehlersuche, zusätzlich konnte ich ihnen oft direkt sagen wo der Fehler lag. Das Wissen um Fehler in den FIX-Nachrichten zu erkennen kam durch den ersten KIT, indem ich mich intensiv mit den FIX-Nachrichten auseinandersetzte und Beispielorders erstellen musste. Während die Kollegen mit dem ausbessern der Fehler beschäftigt, widmete ich mich wieder meiner Webanwendung. Ich wollte diese auch noch fertigstellen, leider konnte ich nicht den Fehler in Bezug mit dem XMLHttpRequest finden. Damit ich mich nicht noch länger an dem Fehler aufhielt entschloss ich mich die Serverantwort erst einmal in einer lokalen Variable zu speichern. Dadurch konnte ich erstmal die restlichen Funktionalitäten entwickeln und mich später wieder diesem Fehler widmen.

## Lauffähige Webanwendung

27.06 – 01.07

Weil Ich den Fehler aus der letzten Woche nicht gelöst und schon sehr lange an dem Fehler nachdachte, schrieb ich die Datei neu. In der Hoffnung, dass ich irgendwo einen typo oder ein semantischer Fehler übersehen hatte. Dies war leider nicht so. Im Gegensatz ich entdeckte dabei neue Fehler in meinem Coding. Mit der neu geschriebenen Datei hatte ich nun das Problem, dass die FIX-Nachricht nach dem Ausfüllen des Formulars zwar angezeigt wird aber nur für etwa eine halbe Sekunde und dann wieder verschwindet. Damit hatte ich meine Anforderung, dass jeder User die Nachricht nach dem Ausfüllen des Formulares noch editieren kann verfehlt. Die Fehlersuche hielt noch etwas an, mit den verschiedesten Fehlern. Mir gelang den Fehler erstmal zu lösen, indem ich den html form tag durch einen div tag ersetze. Damit war es mir endlich gelungen, meine Webanwendung auch Fertigzustellen. Bevor ich diese aber Präsentierte entschied ich mich meine Webanwendung noch zu designen. Während ich mit der Fehlersuche beschäftigt war erfuhr ich, dass der nächste Schritt im Projekt, damit auch von der Webanwendung, ein Execution Report für CSI war. Also war ich Glücklich, dass die Website funktionierte, aber ich konnte direkt die nächste Erweiterung anfangen zu entwickeln. Bevor ich dies tat, entschloss ich mich mein Coding aufzuräumen. Durch die Fehlersuche hatte ich sehr viel auskommentiertes Coding in der JavaScript Datei. Nachdem ich erfolgreich entfernte entschloss ich mich zusätzlich mein Coding nach Bereichen, die einzelnen Teile der Webanwendung, einzuteilen. Am Ende der Woche erhielt ich noch eine neue Aufgabe, welche mit dem Execution Report zusammenhing. Die Business Architekten wussten schon wie ein Großteil der FIX-Nachricht für CSI aussehen wird. Da kam meine Aufgabe, die Webanwendung um ein Formular für einen Execution Report zu erweitern. Gleichzeitig sollte ich erstmals ein, zwei Beispielnachrichten erstellen und diese die beteiligten Parteien senden, damit die die Nachricht auf Richtigkeit prüfen können.

## . Machbarkeitsstudie Tracability Matrix

04.07 – 08.07

Am Anfang der Woche machte ich eine Änderung an der Startseite, von der Webanwendung. Ich benutze eine HTML Tabelle, welche funktionierte aber keine Möglichkeit für ein Styling anbot. Ich fand relativ schnell heraus, dass ich mittels einer ungeordneten Liste (<ul>) zwar viel Aufwand hatte, ich diese aber dafür schön gestalten konnte. Nachdem ich die Liste erfolgreich eingebunden hatte, sah die Webanwendung sehr schön aus. Leider brauchten die Kollegen von der Accenture ihre Arbeitsmaschine wieder, sodass ich auf meinen eigenen Laptop weiterentwickelten musste. Meine Alternative war nochmals den Windows Editor zu nehmen, aber dies war für mich keine Alternative. In der Zwischenzeit hörte ich oft, dass die von mir erstellte Traceability Matrix gut ist, aber niemand so richtig wusste welche die aktuelle Version ist. Durch das letzte Feedback von meinem Betreuer entschied ich mich selber eine Art Machbarkeitsstudie eine Webbasierte Traceability Matrix zu testen. Ich stellte bereits beim Überlegen, über die Funktionalitäten, fest, dass dies eine schwere Aufgabe darstellen würde. Ich machte mir eine Liste mit den Funktionalitäten und probierte diese jeweils zuerst einmal einzeln zu implementieren und später erst zusammenzufügen. Bei dieser Webanwendung sollte der Benutzer in der Lage sein, Zeilen sowie Spalten hinzuzufügen nach verschiedenen Einträgen zu suchen, Nach Einträgen zu sortieren, auswählen welche Spalten er sehen möchte, eine "Excel" Suchfunktion für jede einzelne Spalte (die Einträge sollen gefiltert werden können) und jeder User muss in der Lage sein, verschiedene Zellen inhaltlich zu ändern, solange er die Berechtigung oder ein Passwort hat. Ich startete mit der Benutzerverwaltung, welche auch Anhand von dem SessionStorage sehr gut funktionierte. Ich entschied für den Session Storage, da nach jedem neuen Laden der Webanwendung, die Session Storage geleert wird. Als nächstes schrieb ich ein neues Makro, welche mir den Inhalt der Traceability Matrix zeilenweise in ein JSON-Array schrieb. Selbst das Auslesen und die dynamische Generierung erfolgte ganz gut. Eine dynamische Generierung des Contents musste sein, da der User sonst nicht in der Lage wäre, neue Spalten oder Zeilen hinzuzufügen. Gegen Ende der Woche startete ich noch die Filterfunktion für die einzelnen Spalten. Ich designte diese Funktion so, dass jeder Eintrag zeilenweise herausgelesen wird und falls der Eintrag noch nicht vorhanden ist wird in einem Fenster, auf dem der Fokus liegt, eine Checkbox mit dem Eintrag dahinter erstellt. Zusätzlich gab ich noch die Auswahl 'Alles anzeigen' damit der Benutzer den Filter wieder aufheben kann.

## Filterfunktion fertigsstellen sowie die Webanwendung erweitern

11.07 – 15.07

In der lezten Woche gelang es mir die Filter Funktion für eine Spalte zu implementieren. Da aber jeder meistens mindestens zwei Filter gesetzt hat, im Excel der Traceability Matrix, entschloss ich mich auch meine Filter Funktion für mehrere Spalten auszuweiten. Es war relativ leicht herauszufinden welche Einträge mit einem aktiven Filter gefiltert werden können, aber die Schwierigkeit war diese Werte danach auch anzeigen zu können. Dass die richtigen Zeilen mit zwei aktiven Filtern angezeigt wurden war wirklich schwer, ich hing fast einen ganzen zusätzlichen Tag um an dem Konzept dieser Funktionalität nachzudenken. Aber ich schaffte es trotzdem noch, ich hatte es übersehen die ID der einzelnen Zellen zu erweitern. Die IDs waren bislang dreistellig was mir nicht gereicht hatte, erste nachdem diese auf fünf Stellen erweiterte (fünf Stellen für Zeilennummer und Zellennummer) funktionierte die Funktion. Die Filter Funktionalität verteilte sich auf verschiedene Funktionen welche insgesamt über 170 Zeilen Coding beinhalteten, ich war sehr froh und stolz diese Funktion erfolgreich umzusetzen. Am nächsten Tag gab mir mein Betreuer die Aufgabe meine Webanwendung um einen Execution Report zu erweitern und um Beispiel Execution Reports an CSI zu senden. Nachdem ich die offizielle Angabe hatte, mich um den Execution Report für CSI zu kümmern entschied ich mich diese Webversion der Traceability Matrix erstmal ruhen zu lassen. Damit ich erstmal einen Überblick über die Execution Reports bekam, sprach mich mit Business Architekten, die gaben mir alle nötigen Information und Beispiele für verschiedene Teile der kompletten Nachricht. Eine einfache Beispielnachricht konnte ich recht schnell erstellen, diese versendete ich an die einzelnen Firmen im Projekt um Feedback für die erstellte Nachricht zu erhalten. Nachdem ich die Nachricht versendete hatte, startete ich die Entwicklung, der Erweiterung für meine ursprüngliche Webanwendung. Dabei war die Schwierigkeit, dass es Wiederholgruppen gab. Die Wiederholgruppen hatten eine feste Anzahl an Feldern, welche aber in der Menge der Vorkommnisse auswählbar waren. Zusätzlich musste ich die FIX-Tag Nummern irgendwie den Formularwerten zuordnen. Dies löste ich indem ich für jedes Eingabefeld ein zusätzliches Eingabefeld, welches nicht angezeigt wird, mit der Nummer des jeweiligen FIX-Tags erstelle und diese am Ende ausliest. Dies hatte den Vorteil, dass ich das Formular Feld für Feld auslesen konnte und die korrekte FIX-Nachricht hatte. Auch merkte ich, dass es zwei Arten von Execution Reports gab und sich dabei um ein paar Felder unterschieden. Damit ich keine Falschen Felder in der Nachricht versendete, wurden die nicht enthaltenen Felder nicht angezeigt und beim Erstellen der FIX-Nachricht erweiterte ich die ursprüngliche Funktion um eine Abfrage ob das Feld angezeigt wird. Falls das Feld nicht angezeigt wird, werden die Werte auch nicht in die FIX-Nachricht übernommen.

## Fertigstellung der Erweiterung Exceution Report für CSI

18.07 – 22.07

Am Anfang dieser Woche stellte ich die Erweiterung für den Execution Report fertig. Wie bei den anderen Formularen in der Webanwendung sollte es möglich sein, die FIX-Nachricht nach dem erstellen noch editieren zu können, bevor die Nachricht gesendet wird. Dadurch ist die Erweiterung gut für den Testeinsatz geeignet. Ich sprach mit den Kollegen von Accenture ab, dass die einen Webservice entwickeln, welcher die gesendete FIX-Nachricht direkt in die Warteschlange für CSI legt. Somit konnte gewährleistet werden, dass der, über die Webanwendung gesendete, Execution Report für CSI auch direkt in das CSI System geleitet wird. Zusätzlich beantragten die Kollegen noch einen Account für das CSI Team. Damit das CSI Team in der Lage ist, selbständig die Webanwendung zu nutzen. Ein neuer User war notwendig, da die Webanwendung, aus Sicherheitsgründen, nicht für jeden erreichbar ist. Schließlich gaben wir den Kollegen von CSI die Zugriffsinformationen und zeigten meinem Betreuer die aktuelle Webanwendung, beide waren sehr zufrieden. Mein Betreuer und die Kollegen von Accenture kamen in der Zwischenzeit auf die Idee, dass ich bei Webanwendung zusammenführen sollte. Ich fand die Idee gut, da ich nur basic Javascript für die Webanwendung genommen hatte und dachte, dass es professioneller sein sollte. Zu diesem Zeitpunkt wusste ich lediglich, dass die andere Webanwendung mittels Maven und dem Spring Framework zu einer .jar Datei gebaut wurde. Ich erfuhr kurz danach, dass die Webanwendung in eine Server- und eine User Interface (UI)- Anwendung aufgeteilt war. Die Serveranwendung bestand aus dem Spring Framework welches, mittels Maven, eine .jar Datei baut und die UI Anwendung basierte auf den Javascript Frameworks React und Redux. Das einzigste Framework, welches ich bis dahin schon mal gehört hatte war das Spring Framework. Dadurch gestaltete sich der Rest der Woche, dass ich mich erstmal in die einzelnen Frameworks einließ.

## Lerne das Maven, Spring, React und das Redux Framework kennen

25.07 – 29.07

Mir gefiel die neue Aufgabe, beide Webanwendungen zusammenzuführen, auch wenn ich schnell merkte, dass es sehr schwer werden würde. Am Anfang der Woche Probierte ich eine einfache ‚hello Spring & Maven‘ Webanwendung von einem Tutorial selbst zu entwickeln. Mein Problem war direkt am Anfang, dass mir die nötigen Libraries zum Bauen der .jar Datei fehlten. Nachdem ich mich eine Weile rumschlug, die fehlenden Libraries zu finden und in meine Entwicklungsumgebung einzubinden, geling mir die erste kleine .jar Datei. Direkt danach widmete ich mich dem React Framework und probierte meine ersten einfachen Webseiten mit React zu schreiben. Mein größtes Problem war die eigene Syntax, die das Framework mitbrachte. Der ganze Aufbau von den Inhalten der Webseiten wird in verschiedenen Modulen gebaut. Die React Dateien sind im JSX Format, dies ist eine Mischung aus Javascript, HTML Elementen und XML Elementen, dazu kommt das die JSX Dateien mit der neuen Javascript Version ES6 verwendet wird, welche mir noch nicht vertraut war. Obwohl es eine ganze Menge Stolpersteine beim Lernen von React gab, war es relativ einfach umso mehr man das Prinzip von React verstand. Als es mir gelang ein paar einfache Webseiten mit React zu gestalten wechselte ich wieder sofort zu Redux. Redux baut auf React auf und bietet ein sehr gutes Konzept, welches umfassende Möglichkeiten für die User Interaktionen auf Webseiten bietet. Allerdings muss man, wie bei React, sich erstmal viel einlesen und das Prinzip hinter Redux verstehen, bevor man anfangen kann eigenen Webanwendungen damit zu entwickeln. Als ich das Prinzip von Redux und von seinem Arbeitsablauf verstanden hatte, entwickelte ich meine eigene kleine Webanwendung mittels Redux. Dies gelang mir mit dem ‚React-Redux-Starter-Kit‘. Dieses kann man bei Github downloaden und es beinhaltet bereits einen vollständig konfigurierten NodeJS Server, welche mit dem Express Framework entwickelt wurde. Der Vorteil ist, dass man einfach seine eigene Webanwendung in die bestehende einbauen kann und mittels ‚hot reloading‘ sieht man auf der Webanwendung jede Veränderung, im Quellcode, sofort. Der Nachteil ist allerdings, dass dieses Starter Kit von jedem Tutorial abweicht und dabei selbst kein eigenes anbietet. Daher muss man direkt am Anfang schon eine gute Wissensgrundlage haben, wenn man anfange will mit dem ‚React-Redux-Starter-Kit‘ zu entwickeln.

## Meeting mit dem Webentwickler der Webanwendung

01.08 – 05.08

Mein Betreuer informierte mich über die nächste Erweiterung für die Webanwendung. Diesmal sollte es möglich sein eine Preisanfrage an die Börse zu senden und darauf mit einer Kauf Nachricht oder Ablehnen Nachricht zu antworten. Weil die Kollegen von DECIDE allerdings diesen Request nur per Hand ausführen konnten, reichte es erstmal eine einfache Website in die aktuelle Webanwendung zu integrieren. Dies erfolgte allerdings später, weil ich die beiden Webanwendungen zuerst zusammenführen wollte. Da ich das Redux Framework noch nicht ganz verstanden hatte, entschied ich mich am Anfang der Woche mir weitere Tutorials zu Redux anzuschauen. Diese halfen mir aber nur in der Theorie, da die Webanwendung von den Kollegen auch mit dem ‚React-Redux-Starter-Kit‘ entwickelte wurde. Mit der Zeit kam ich mehr und mehr hinter das System von Redux, leider Verstand ich aber die Struktur der Webanwendung noch nicht ganz. Um diesem Problem entgegenzuwirken wurde der Entwickler von der Webanwendung eingeladen, er arbeitet auch für die Accenture und war in einem anderen Projekt in Frankfurt, so dass er Zeit hatte bei uns vorbeizukommen. Wir hatten ein ausführliches Meeting, wo er auch die Grundlagen wiederholte. Bei diesem Meeting habe ich die Struktur von seiner Webanwendung noch mehr verstanden, auch wurde mir klar, dass ich im Bereich React und Redux bereits gute Grundlagen habe. In dem Meeting probierte ich alle möglichen Informationen mitzunehmen, da ich über zwei Stunden mit Informationen überflutet wurde, war dies keine leichte Aufgabe. In den nächsten Tagen stellte sich heraus, dass das Meeting sehr viel Wert war, ich kam dazu meine erste eigene Redux Seite in das ‚React-Redux-Starter-Kit‘ zu integrieren und konnte darauf aufbauen. Zudem erfahr ich in dem Meeting von ‚SCSS‘. Dies ist eine Erweiterung für herkömmliche ‚css‘ Dateien, welche ich mir auch anschauen wollte, wenn ich noch Zeit hatte.

## Die Eigene Webanwendung in einer .jar Datei

08.08 – 12.08

Damit ich ein noch besseres Verständnis von Redux ehrhalte, entschloss ich mich eine Todolist erweiterung in das bestehende Starter-Kit zu integrieren. Die Todoliste sollte neue Aufgaben hinzufügen, diese als erledigt markieren und die anzuzeigenden Aufgaben (Alle anzeigen, erledigt anzeigen oder noch nicht bearbeitet Anzeigen). Dafür brauchte ich mehr Zeit als ich ursprünglich dachte, dass lag daran, dass jetzt die Logik etwas komplexer geworden war. Aber als ich die Ergänzung erfolgreich eingebaut hatte, war ich Zufrieden und hatte das Gefühl, dass ich relativ leicht jetzt die Webanwendung erweitern könnte. Ich wollte aber bevor ich die Webanwendung erweitere, eine .jar Datei mit meinem eigenen React, Redux Coding ausprobieren zu bauen. Dabei hatte ich das Problem, dass das Springframework die falschen Dateien benutzte um die .jar Datei zu bauen. Nach einem E-Mail Austausch mit dem Entwickler der Webanwendung wusste ich wo mein Fehler lag. Ich musste zuerst das ‚React-Redux-Starter-Kit‘ benutzen um Dateien generieren zu lassen, welche die gesamte Anwendungslogik abdeckten. Diese musste ich zuerst in das Verzeichnis von der Spring Anwendung kopieren um daraus die .jar Datei mit dem richtigen Inhalt bauen lassen zu können. Bei der Konversation mit dem Entwickler fand ich auch heraus, dass er Webpack benutzte um die Gesamtgröße der Webanwendung gering zu halten. Ich las mich in Webpack ein bisschen ein und bemerkte schnell den Sinn dahinter, sobald man eine größere Webanwendung hat. Ich probierte aus Webpack an kleinen einfachen Beispielen aus, was gut funktionierte. Ich hatte allerdings Probleme das ‚React-Redux-Starter-Kit‘ mit Webpack entwickeln. Zum Glück bemerkte ich, dass das Webpack schon in das Starter-Kit mit, vom Webentwickler der Webanwendung, eingebaut wurde.

## Mit Redux ein Formular bauen

15.08 – 19.08

Am Anfang der Woche probierte ich meine einfachen Beispielanwendungen in die aktuelle Webanwendung zu integrieren. Da dies sehr gut klappte ging ich sofort über die Webanwendung um den Execution Report für CSI zu erweitern. Die Webanwendung von Accenture hatte diesen noch nicht implementiert, weil meine Webanwendung benutzt wurde als diese Fertig war. Um die Erweiterung entwickeln zu können musste ich erstmal ein anderes, bestehendes Formular verstehen, welches die gleiche Logik hatte. Es wurde neben dem globalen Speicher ein lokaler Speicher benutzt, worin die Formulardaten gespeichert und an Funktionen zur Weiterverarbeitung übergeben wurden. Allerdings war hier das Problem, dass der lokale Speicher bei der Initialisierung eine feste Größe hatte. So konnte ich nicht für die Wiederholgruppen oder für die Felder (welche nur in einem von beiden Reports vorkommen) die Logik implementieren. Also suchte ich im Internet ob es nicht eine andere Möglichkeit gab. Schnell wurde ich auf Redux-Form aufmerksam. Dies war eine Erweiterung für Redux, die speziell für Formulare geeignet ist. Mit dieser war es möglich Formulare zu validieren, um Felder zu erweitern oder Felder zu löschen, sowie auch ein Formular über mehrere Seiten anzuzeigen. Leider gab dafür noch kein Tutorial, sodass ich mein Wissen aus den hinzugefügten Beispielen und der Dokumentation dieser Erweiterung nehmen musste. Dies machte es mir sehr schwer das Formular für den Execution Report zu entwickeln. Leider schaffte ich es nicht in dieser Woche das Formular fertigzustellen und nahm das Problem so mit in die nächste Woche.

## Week 21 22.08-24.08 – Last Day

22.08 – 24.08

-redux-form benutzt

-csi execution report logik

## Week 22 25.08-30.08 – Holiday

25.08 – 30.08

# Projektbericht

Weil ich in einem Langzeitprojekt tätig war und meine Tätigkeiten den Zweck hatten, das Projekt zu unterstützen, werde ich auf das Neue Wertpapier Plattform Projekt eingehen und danach meine Projekte beschreiben.

## Neue Wertpapier Plattform

### Abstract

Die Commerzbank AG hat für den Börsenhandel die Plattform COWIAS, welche in manchen Teilen schon über 25 Jahre alt ist. Aufgrund der Machbarkeitsstudie „Due Diligence WP-Plattform“ in 2014, entschied sich die Commerzbank den Börsenhandel bei sich zu behalten und nicht über eine andere Firma abzuwickeln. Aus dieser Entscheidung entstand das Projekt „Neue Wertpapier Plattform“. Das Ziel dieses Projektes ist es, langfristig weniger Risiken und Wartungskosten beim bestehenden System zu haben, sowie langfristig im Börsenhandel Konkurrenzfähig zu bleiben. In dem Projekt wird die Plattform INSPIre (Innovative Securities Processing) entwickelt. Jede FIX- Nachricht, für eine Order an der Börse, wird durch die INSPIre Plattform an den zugehörigen Bereich weitergeleitet. In dieser Plattform wird ein „Integration Layer“ entwickelt, welcher Zuständig ist, dass die FIX-Nachrichten zwischen den Systemem ankommen.

### Grundlagen

Anschließend werden die Grundlagen zum Börsenhandel mit Bezug auf das Programm beschrieben, da das Projekt auf den Börsenhandel abzielt.

#### Aktienhandelprozess

Der Vorgang eines Aktientausches ist in dem Neue Wertpapier Plattform Programm in drei Bereiche aufgeteilt. Es gibt den Auslöser (‚trigger‘), z.B. jemand plaziert eine Order an der Börse, dass er 3 Aktien kaufen möchte. Als nächste kommt der Prozess (‚processing‘), in diesem Teilprozess finden sich ein Käufer und ein Verkäufer. Zum Schluss kommt die Abrechnung (‚ledger booking‘), hier wird der Aktienhandel durchgeführt und die Abrechnung (das Depot wird von in der Höhe des Kaufbetrages belastet oder bekomm den Betrag der Verkaufshöhe zugeschrieben) wird vollzogen. Der Auslöser verwaltet den Bereich von der Ordereingabe bis zur Weiterleitung an die entsprechende Börse. Dabei wird die Order vom System noch vor der Weiterleitung evaluiert. Dies hat den Grund, dass jede Börse für eine Order eine Gebühr nimmt. Damit diese Gebühr nicht mehrmals bezahlt werden muss wird nur eine valide Order vom System angenommen. Es gibt die Möglichkeit, dass die Order vom System nicht weitergeleitet wird, aber in den ‚Approval-Pool‘ geleitet wird. Dort lagern sich alle Orders welche eine kleine manuelle Änderung von einem Angestellten benötigen, bevor die Order an die Börse weitergeleitet wird. Wenn die Order an der Börse platziert ist, sind wir im Bereich (‚Processing‘). Sobald eine Order an der Börse plaziert ist, wird das Depot vom Benutzer gesperrt (im Falle eines Verkaufs) oder belastet (im Falle eines Kaufes). Gleichzeitig wird an der Börse nach einem Kontrahenten gesucht. Die Order kann dabei von der Börse in verschiedene Orders unterteilt werden. Falls die Börse keinen Kontrahenten für eine Order findet, wird die Order nach einer bestimmten Zeit wieder gelöscht und das Depotmanagement regelt, dass die Sperrung aufgehoben wird oder dass der Kaufbetrag wieder gutgeschrieben wird. Wenn allerdings ein Kontrahent für eine Order gefunden wird, wird der Abwicklungsprozess eingeleitet. In diesem Prozess werden alle Daten zum Trade an ein System weitergeleitet. Dieses System wechselt u.a. die Besitzer der Aktie(n). Nach dem Abwicklungsprozess kommt noch das (‚Ledger Booking‘). In diesem letzten Schritt wird das Depot, sowie das Kundenkonto von beiden Kontrahenten aktualisiert.

#### Smart Order Routing

Smart Order Routing wird im Rahmen des Prozesses entwickelt. Es bietet die Möglichkeit, dass der Benutzer eine Order absenden kann ohne dabei eine Börse anzugeben. Das System wird dann mit verschiedenen Algorithmen die Börse ermitteln, wo die größten Tradingmöglichkeiten gesehen werden. Nachdem die entsprechende Börse ermittelt wurde, wird die Order dorthin geleitet.

### Ziele

Der Hauptfokus von dem Programm liegt auf der Sicherung, sowie der Stärkung um in Zukunft im Bereich Aktienhandel konkurrenzfähig zu bleiben. Damit das Ziel erreicht wird sind folgende zusätzlichen Ziele definiert.

* Änderung effizient umsetzen zu können
* Eine verbesserte Performance zu ermöglichen
* Die Kosten im Business Sektor zu senken
* Die Risiken im Business Sektor zu senken

Auch ein Ziel ist, dass in Zukunft alle Abrechnungsprozesse von den verschiedenen Bereichen über eine Plattform mit modernen Technologien umgesetzt wird.

### Beteiligte Projekte

Anfang Juni (6. Juni 2016) waren in dem Neue Wertpapier Plattform Programm vier Projekte enthalten. Die Projekte waren

* Retail Order Processing
* CSI (Central Settlement Interface)
* Fund Processing (
* SecFinance BO

Bis das Programm abgeschlossen ist, wird sich diese Liste noch mehrmals ändern, da nicht jedes Projekt bis zum Ende aktiv sein wird und es werden auch andere Projekte hinzukommen.

### Retail Order Processing

#### Abstract

Im Rahmen des Neue Wertpapier Programm ist das Projekt Retail Order Processing aktiv. Dieses Projekt beinhaltet fünf verschiedene System (Comdirect, Integration Layer, Decide, Mainframe und CSI). Das Hauptaugenmerk von dem Projekt ist eine Verbindung zwischen den Systemen zu versichern. Durch das System Decide eine Anbindung an die verschiedenen Börsen gewährleistet und mit dem ‚Central Settlement Interface‘ wird die Abrechnung nach einem erfolgreichen Trade gewährleistet. Dadurch kann man vereinfacht sagen, dass unser Projekt für den Börsenhandel zuständig ist. Decide ist das Order-Management-System. Dies ist dafür da um die Orders, welche an die Börse gesendet werden sollen, zu evaluieren bevor diese an die Börse weitergeleitet werden um keine unnötigen Kosten entstehen zu lassen. Decide prüft einerseits, ob die Order von der entsprechenden Börse angenommen wird, aber auch ob die Börse geöffnet hat oder geschlossen ist. Wenn die Börse geschlossen ist, wird Decide die Order puffern und sobald die Order geöffnet hat wird diese an die Börse weitergeleitet. Zusätzlich wird von Decide das Smart-Order-Routing Funktion entwickelt. Diese Funktion ermöglicht dem Benutzer keine Börse auszuwählen, sondern Decide die Börse mit dem größten Kauf oder Verkaufschancen zu wählen. Der aktuelle Plan von diesem Projekt sieht zwei Stufen vor. Diese Stufen sind in verschiedene Pakete unterteilt, wobei am Ende einer Stufe ein Go-Live vorgesehen wird. Die erste Stufe solle Ende 2016 fertig entwickelt sein, wie lange das Projekt noch läuft ich aber noch nicht abzusehen.

#### Ziele

Das Ziel von größter Bedeutung ist die Integration von dem neuen Order-Management-System (Decide). Daneben gibt es aber noch weitere Ziele

* Bereitstellen einer Standartsoftware
* Die Standartsoftware enthält alle möglichen Validierungen für eine Order und hat damit das gleiche oder ein besseres Evaluationssystem als die Börsen
* Die Erfolgreiche Integration von der Standartsoftware in das bestehende System

Das Projekt zielt auch darauf ab, dass alle Systeme zumindest ein wenig im Börsenhandel mitarbeiten. Aktuell erledigt der Mainframe noch fast die ganze Arbeit in dem Wertpapieraustauschprozess. Lediglich das Routing einer Order an eine Börse erfolgt durch ein anderes System, ansonsten ist der Mainframe überall beteiligt. In der Zielarchitektur wird Decide den ersten Schritt im Prozess, eine Order an eine Börse weiterleiten übernehmen. CSI, angenommen ein Trade findet statt, führt daraufhin den Inhaberwechsel durch. Der Mainframe kommt zum Schluss und übernimmt zum Schluss die Abrechnung und das Konto- / Depotmanagement. Dadurch sind die Systeme im Börsenhandel enthanteln, aber jedes System hat seine eigene Aufgabe. Mit dieser Aufgabe pro System ist es viel einfacher die Wartungskosten niedrig zu halten und das System bei Bedarf zu erweitern.

#### Grundlagen

Die Projektplanung unterteilt sich in mehrere Stufen, wobei die erste Stufe Ende dieses Jahres vollendet sein soll. Zum Abschluss der ersten Stufe ist ein Go-Live geplant. In diesem Go-Live ist geplant, dass die Comdirekt in das neue System vollständig integriert ist, Decide (das Order-Management-System) erweitert werden, Decide stellt die Smart-Order-Routing zur Verfügung, Decide wird in die neu entwickelte Plattform integriert und CSI soll für den Retail Bereich erweitert werden.

#### Beteiligte Systeme

Das Projekt Retail Order Processing beinhaltet fünf verschiedene Systeme. Jedes System kommuniziert mit dem anderen System über den Integration Layer. Lediglich die Comdirect hat eine direkte Verbindung zu Decide. Der Integration Layer bietet zur Kommunikation UM (Universal Messaging), MQ (IBM WebSpehre MQ) und MO & COMI3 (COMI3 ist eine von der Commerzbank, für die Kommunikation zwischen System, Art von MQ. Jede Nachricht wird unabhängig vom System im FIX5.0 Format (ein Format speziell für den Austausch von Nachrichten mit Aktieneinzelheiten). Sollte ein System FIX5.0 nicht oder nur zum Teil lesen können, bietet der Integration Layer Konvertierungen für das System an.

##### Integration Layer

Der Integration Layer wird von der Accenture GmbH in Kronberg entwickelt. Die Aufgabe des Layers ist es die Kommunikation zwischen den Systemen zu gewährleisten. Damit gewährleistet werden kann, dass die Nachrichten am Zielsystem ankommen benutzen alle System bei der Kommunikation mit dem Integration Layer die ‚Message-Orientated-Middleware‘

##### DECIDE

Decide wird von der Firma ‘pdv Financial Software GmbH’ in Hamburg geliefert. Dieses System wird die Kommunikation mit den Börsen übernehmen. Es überprüft ob eine Order syntaktisch, sowie semantisch korrekt ist, gibt eine Benachrichtigung ob die Börse geöffnet hat oder geschlossen ist. Decide liefert auch das Smart-Order-Routing System.

##### Comdirect

Das Tochterunternehmen Comdirekt benutzt für ihren Börsenhandel das Commerzbanksystem. Die Comdirekt muss auch einen Teil ihres Systems neu entwickeln um die FIX-Nachrichten senden zu können.

##### Mainframe

Der Mainframe ist ein System in der Commerzbank, welches in Rahmen dieses Projektes erneuert wird. Es ist dafür zuständig, dass die Account- sowie die Depotabrechnung zwischen den Kontrahenten klappt.

##### Central Settlement Interface (CSI)

Das ‘Central Settlement Interfaced’ ist benötigt wenn ein Trade zustande kommt. In diesem Fall führt CSI die Inhaberwechsel auf den Aktien aus und übergibt die Kontrahenten Informationen an den Mainframe.

#### Architektur

Der Integration Layer ist der dreh- und Angelpunkt in diesem Projekt. Er entkoppelt aller System voneineander, mit ausnahme von Decide und der Comdirekt. Die Anwendung basiert dabei auf dem WebMethods Integration Server, welcher von der Accenture eingesetzt wird. Die WebMethods Anwendung ist eine etablierte Platform um heterogene Anwendung in bestehende System zu integrieren.

## Traceability Matrix

Mein erstes größeres Projekt, war die Entwicklung eines Excel Tools. Mit diesem Tool sollten die Benutzer in der Lage sein, schnell den aktuellen Status des Projektes herauszufinden. Der aktuelle Status bezog sich in meinem ersten Projekt hauptsächlich auf den Planungsstand und den Lieferungsstatus der einzelnen Lösungen. Der erste Teil war in Excel eine ansehnliche Vorlage zu entwickeln. Im zweiten Teilt entwickelte ich ein Makro mit VBA, welches die Tabelle ordnete und leserlicher machte.

### Abstarct

Zum Start des Projektes waren alle Anforderungen des Retail Order Processing Projektes im EA (Enterprise Architect) gespeichert. Das Programm ist dafür ausgelegt Anforderungen und Lösungen in Projekten zu Verwalten. Leider ist es auch sehr unübersichtlich. Durch diese Unübersichtlichkeit kam meine Aufgabe, eine übersichtliche Tabelle zu erstellen. Bis ich den ersten Teil des Projektes fertiggestellt hatte, gab es für jedes System eine andere Tabelle mit verschiedenen Versionen in Meetings, wodurch eine große Verwirrung entstand wie der aktuelle Status ist. Die Traceability Matrix sollte so aufgebaut sein, dass jeder im Projekt ohne Mühe sehen kann welche Lösungen, welche Anforderungen umsetzen, wie die Ursprungsplanung und die aktuelle Planung ist, sowie auch wie der Lieferstatus ist. Des Weiteren war meine Aufgabe ein Export vom EA, welcher im ‚.csv‘ Format gespeichert wird zu benutzen.

### Erster Part – Excel Pattern

Für diesen Teil des Projektes war meine Aufgabe eine geeignete Excel Tabelle zu entwerfen.

Die Anforderung an das Pattern waren:

* Eine strukturierte und verständliche Excel Tabelle
* Eine geeignete Gruppierung der Spalten in der Tabelle
* Eine Auswahl an verschiedenen Werten für verschieden Zellbereiche entwickeln
* Einen Bereich für die Nachverfolgung des Lieferstandes

### Zweiter Part – Excel Makro

Der zweite Teil war, ein Makro zu entwickeln welchen den ‘.csv’ Export aus dem EA integrieren konnte und diesen dabei für die User lesbar machen.

Dieses Mal waren die Anforderungen:

* Einen Algorithmus der die Namen der Lösungen an einer exakten Stelle abschneidet (diese Stelle kommt in jedem Lösungsnamen vor aber oftmals variiert diese Stelle)
* Per Algorithmus alle Anforderungen, welche von einer Lösung umgesetzt werden, raussuchen und einmal mit und einmal ohne Anforderungsbeschreibung in eine Zelle kopieren
* Die Tabelle nach der Bearbeitung verschönern

### Zusammenfassung

Der Zweck dieses Projektes war, dass ich einen Einblick in das Verwalten von großen Projekten mit verschiedenen Systemen bekomme. Meiner Meinung nach war dieses Projekt sehr gut um eine Idee zu bekommen wie man solch ein Großprojekt managed. Ich erhielte von vielen Kollegen ein positives Feedback zu der Traceability Matrix. Die Manager benutzen nach wie vor diese Excel Tabelle um den aktuellen Projekt- und Lieferstatus zu sehen. Ich glaube das Projekt war ein sehr wichtiges. Durch dieses Projekt, haben die Mitarbeiter eine einfache Übersicht über den Status des Projektes und ich habe einen Eindruck bekommen, was es bedeutet solch ein großes Projekt zu verwalten.

## Interface Makro

My second larger project was to develop a macro, which creates for every system interface his own document. The excel files should describe how the interface look when the development is finished, the current status was not important.

### Preface

The system DECIDE has five different interfaces between the systems, stored in two huge excel tables. My task was to create a macro which is able to create a file, based on the two main tables, which shows exactly one interface.

### Requirements

* Find out and copy all necessary sheets for the specific interface in the excel file
* Filter the parts, which shouldn’t be shown (i.e. development status)

### Conclusion

The project was useful, due to the two main files didn’t offer a good overview. The separated files offer the detailed view for every interfaces. They were used for the quality process, in this process were decided if the specific interface is alright designed or need some more work.

## Web application

My biggest project during the internship was to develop a web application. This project had two stages. The first was to developed a web application which can create, for test purpose, fix messages, send them to a web service, offered by Decide, and display the answer of Decide. It should be simulated how the flow in the future will look like. The second stage where to use modern web development technologies like React and Redux and merge my application with an application developed by accenture.

### Principles

The colleagues from Pdv created a service, that simulates Decide. The service is able to accept and to interpret fix messages, as well as to send the specific answer to a fix message. In the web application, we used the following fix messages to send to Decide

* CreateOrder (creates a new order)
* ModifyOrder (modify an existing order)
* CancelOrder (cancel an existing order)
* OrderStatus (receive the status of an order)
* OrderMassStatus (receive the status of all order from one account)
* Quotes (an offer from the stock market, to buy or sell a stock paper for a specific price)

Also the system CSI was included in the web application. For CSI the application should be able to send the following messages

* New Execution Report (when a trade was successful, CSI receive the data of the trade)
* Execution Modify Report (a change or a cancel of a successful trade)

All messages were send to the IL, which forwards the message to the specific system.

### Requirements

The requirements were for both web applications the same. The second stage was only to merge both applications. Due to those facts here are all requirements.

* The web application must have a branding
* The web application must be able to show all order of one account that Decide has in his system
* The information of a shown order must give the user a short overview about the order
* The application must be able to offer for every order which is shown a button that can be used to modifying, canceling or show details of one specific order
* The application must be able to show the details of a specific order
* The application must ensure that every fix message is in the fix format and can be proceed
* The application must provide a form to create and send a CreateOrder
* The application must be able to create and send a CancelOrder fix messages
* The application must be able to create and send a ModifyOrder
* The application must be able to serve a log-in function
* The application must provide a form to create and send a NewExecutionReport
* The application must provide a form to create and send a ModifyExecutionReport

### First stage

The first stage of the application was a simple web application which contained several html files, a CSS file and a JavaScript file to create and send the fix message and also to make the html content a little bit dynamic. Also the colleagues from Accenture developed parallel also a web application with modern technologies. Because my supervisor wanted to present the web application very fast firstly the application from Accenture was used. The first version, at the presentation of my supervisor, included a simple form to create and send a CreateOrder fix message, a shown list of all orders in Decide from your account, a button at the shown list to show the order details, a button to send an CancelOrder message for the order at the shown list, a button to create and send a ModfiyOrder and the log-in function. At the end of the first stage my application had the same offer of functions as the demo from Accenture but moreover I was able to implement the extension for creating and sending the NewExecutionReport and the ModifyExecutionReport.

### Technologies

#### Maven

The idea of maven was to simplify the build process of a web application, additional they wanted to set a new standard in the web application server file building. Maven allows easily to use plug-in and add-ons. They created a POM (project object model), which is stored in an xml file. In the POM there are all dependencies and add-ons written down for the completed project, this makes Maven very easy to handle the dependencies of a project (Maven - Idea, 2016).

#### Spring Boot

SpringBoot is a framework from the Spring project. In the time there were a lot of frameworks build by spring, i.e. Spring Cloud, Spring Mobile, Spring for Android or Spring Web Services (Spring Documents, 2016). All Spring products are created in Java, it is definitely a reason why Spring is so popular. Spring Boot allows to create a Java server within five or six lines of coding, the user just have to execute maven or gradle to build an executable jar file. Another huge advantage is that Spring applications don’t need to deploy a war file (Web Application Archive). The created jar file can be run either on a local machine or on a webserver. The framework offers an integrated Tomcat, Jetty or Undertow Java Servlet application (Spring Boot, 2016), which covers the server part in the web application.

#### React

React is a very young JavaScript framework, which should simplify the developing process of a user interface. The framework was created by Facebook and can be pleased that it is getting very popular (React , 2016). The most important part is that React uses components, but there is no standard which describes what is a component and what is not a component. With this fact every UI developer is free to create and group the component, which brings a huge advantage. Every developer need to be aware of what he wants to develop, if you would start without a clear idea about what do you want to build you would mess up the homepage. The components in React allows you to store are state (which are attributes of an object) and also to call a function of another component and give the called component some props (like parameters in a function call). With this principle the coding gets very structured. In example to create a list with a few entries, you should have at least need two components. The first component creates the html list and calls the second component which creates and returns the html list elements. Additional you would need a third component which stores your elements for the list, but you can use those two components (which creates the list) every time again in your web application. This is another reason why React is becoming very popular. Another reason to use React, is the reusability of the components. The example with the list brings it to the point. You create several components but using them over and over again. React uses it owns syntax, which is called JSX. This makes the start very hard, because you have JavaScript, XML and HTML elements in one file. After a short period of time the advantage of the JSX syntax is getting clear. The html page has one element. This element is the entry point for React, from there it renders the complete webpage. Every time some data changed in the webpage, React will automatically render it again without that the user sees a flickering (React Advantage, 2016). This is a huge advantage against other framework, which already tried to simplify the UI development but had always a flickering when the webpage was rendered again.

#### Redux

Add-on for React

#### JavaScript ES2016

* -Virtuelles DOM
* -Flux Architecture
* -Webpack

#### Webpack

## Summary

* EA pflege
* CIT support
* Traceability Matrix
* Macro for Interfaces
* Web application

One point that was new for me. In this project always the technical data and the content data are clearly separated, also they are in different documents. So for interface descriptions or for the architecture documents there are at least two documents, one for the technical information and one for the content information.

## Perspective

## References

(kein Datum). Abgerufen am 07. Juni 2016 von Commerzbank - Privatkunden: http://www.commerzbank.de/portal/de/privatkunden/startseite.html

(kein Datum). Abgerufen am 07. Juni 2016 von Commerbzank - Geschaeftskunden: http://www.commerzbank.de/portal/de/geschaeftskunden/startseite.html

(kein Datum). Abgerufen am 24. April 2016 von Commerzbank - Überblick: https://www.commerzbank.de/de/hauptnavigation/konzern/commerzbank\_im\_\_berblick/commerzbank\_ueberblick.html

(kein Datum). Abgerufen am 07. Juni 2016 von Commerzbank - Markengeschichte: http://www.commerzbank.de/de/hauptnavigation/konzern/die\_marke\_commerzbank/markengeschichte/Markengeschichted.html

Bereich, L. S.–E. (kein Datum).

*Maven - Idea*. (07. 08 2016). Von https://maven.apache.org/what-is-maven.html abgerufen

*React* . (07. 08 2016). Von https://facebook.github.io/react/ abgerufen

*React Advantage*. (07. 08 2016). Von https://facebook.github.io/react/docs/why-react.html abgerufen

*Spring Boot*. (07. 08 2016). Von http://projects.spring.io/spring-boot/ abgerufen

*Spring Documents*. (07. 08 2016). Von https://spring.io/projects abgerufen

* Dokument: Titel, Namen von Autor/Verantwortlicher, Themen, Datum, Version
* Vertraulich oder nicht

# Glossary

|  |  |
| --- | --- |
| expression | Explanation |
| NWPP | “Neue Wertpapier Platform Program” - The current Program “New Security Paper Platform Program” |
| ROP | “Retail Order Processing” – One project in scope of the NWPP |
| CD | Comdirect |
| COWIAS Mainframe host | The main system in the current workflow. |
| IL | Integration Layer |
| Decide | Decide is the new Order-Management-System and will be developed by pdv Financial Software GmbH |
| CSI | System – Central Settlement Interface |
| SOR | Smart-Order-Routing |
| EA | Enterprise Architect An software used in the Commerzbank to manage requirements for a project. |
| HPQC | HP Qualitiy Center An software used in the Commerzbank to create, manage and execute your tests. The software can also be used to create report diagrams. |
| CIT | Component-Integration-Test |
| OFAC | Office of Foreign Assets Control |
| QS | Quality Security |
| REST | Representational State Transfer  It relies on a stateless, client server, cacheable communication protocoll |