

Universität Leipzig

Fakultät für Mathematik und Informatik

Institut für Informatik

Betriebliche Informationssysteme

Softwaretechnik-Praktikum 2017

Recherchebericht

Projekt:	mok17
Betreuer:	Dr. Heiko Kern Dr. Fred Stefan
Tutor:	Franz Teichmann
Gruppe:	Tobias Zschiezschmann Fabian Götz Adrian Fietta Fabian Wenzel Simon Wilhelm Markus Kempe Nico Reichenbach Christian Becker
Eingereicht am:	19.12.2016

Inhaltsverzeichnis

Inhaltsverzeichnis	1
1 Begriffe	3
1.1 API.....	3
1.2 Framework.....	3
1.3 Datenbank(DB).....	3
1.4 Datenbank-Managementsystem (DBMS):	3
1.5 Datenbanksystem (DBS)	3
1.6 Informationssystem (IS)	3
1.7 Datenbanksprache.....	3
1.8 SQL.....	3
1.9 Webserver.....	3
1.10 Servlet.....	4
1.11 DB-Connection.....	4
1.12 DB-Schnittstelle	4
1.13 Datenbankdump.....	4
1.14 Secure Shell (SSH).....	4
1.15 Shell.....	4
1.16 Apache Tomcat.....	4
1.17 Native App.....	4
1.18 Ionic Framework	4
1.19 Apache Cordova	5
1.20 Android Studio	5
1.21 Vaadin.....	5
1.22 Kennzahlen	5
2 Konzepte.....	5
2.1 REST	5
2.2 Rich Internet Application	6
2.3 Client-Server-Modell.....	6
2.4 Drools	6

2.5 Anwendung der Datenbank	7
2.6 Datenbankinstallation	7
3 Aspekte	8
3.1 Ziel des Projekts	8
3.2 Softwarearchitektur	8
Quellen.....	

1 Begriffe

1.1 API

Eine Programmierschnittstelle, genauer Schnittstelle zur Anwendungsprogrammierung, häufig nur kurz API genannt (englisch application programming interface, wörtlich ‚Anwendungs-programmier-schnittstelle‘), ist ein Programmteil, der von einem Softwaresystem anderen Programmen zur Anbindung an das System zur Verfügung gestellt wird. Die Programmanbindung erfolgt auf Quelltext-Ebene.

1.2 Framework

Ein Framework ist eine semi-vollständige Applikation. Es stellt für Applikationen eine wiederverwendbare, gemeinsame Struktur zur Verfügung. Es wird in eine Applikation eingebaut, und derart erweitert, dass es ihren spezifischen Anforderungen entspricht. Sie stellen eine kohärente Struktur zur Verfügung, anstatt eine einfache Menge von Hilfsklassen. Alle Frameworks, welche in diesem Projekt verwendet werden, sind im Recherchebericht erwähnt.

1.3 Datenbank(DB)

Eine große Menge von Daten, die in einem Computer nach bestimmten Kriterien organisiert sind und komplexe Abfragen zulassen.

1.4 Datenbank-Managementsystem (DBMS):

Software-System zur Definition, Verwaltung, Verarbeitung und Auswertung der DB-Daten

1.5 Datenbanksystem (DBS)

Einheit von DB und DBMS

1.6 Informationssystem (IS)

System, welches zum Speichern, bearbeiten, analysieren und Abfragen von Informationen dient; es besteht aus dem Datenbanksystem (DBS), Anwendungssystemen und Benutzerschnittstellen.

1.7 Datenbanksprache

formale Sprachen, die speziell zur Kommunikation mit der Datenbank ausgelegt sind; ermöglicht Zugriff auf die DB

1.8 SQL

Datenbanksprache zur Definition von Datenstrukturen in Datenbanken sowie zum Bearbeiten (Einfügen, Verändern, Löschen) und Abfragen von darauf basierenden Datenbeständen

1.9 Webserver

Server, der Dokumente an Clients (z.B. Webbrowser) überträgt; als Webserver bezeichnet man den Computer mit Webserver-Software oder nur die Webserver-Software selbst.

1.10 Servlet

Java-Klassen die dynamisch auf Webanfragen von Clients reagieren. Mit den Servlets kommuniziert der Webcontainer, der zuständig für deren Management ist (erstellen, laden, und beenden der Servlets).

1.11 DB-Connection

Verbindung der Datenbank mit dem Anwender/Endbenutzer über ein Netzwerk

1.12 DB-Schnittstelle

Programmierschnittstelle, für den Zugriff und Datenaustausch mit einer DB, d. h. Ermöglichung der Kommunikation zwischen einer Softwareapplikation und einer DB; durch eine definierte Datenbankschnittstelle können Datensätze ausgelesen oder verändert werden, ohne die Verwaltungs- und Speicherungsstruktur der Datenbank zu kennen.

1.13 Datenbankdump

Anteilige oder ganze Auszüge aus einer Datenbank

1.14 Secure Shell (SSH)

Netzwerkprotokoll zum Herstellen einer verschlüsselten Netzwerkverbindung mit einem entfernten Gerät

1.15 Shell

Kommandozeileninterpreter auf Unix-basierten Systemen

1.16 Apache Tomcat

(Open-Source-) Webserver und Webcontainer, der die Spezifikation für Java Servlets und JavaServer Pages (JSP) implementiert; somit ist es möglich, in Java geschriebene Web-Anwendungen auf Servlet- beziehungsweise JSP-Basis auszuführen.

1.17 Native App

Apps die speziell auf die Zielplattform angepasst und zugeschnitten sind, d.h. direkte Entwicklung für Android oder iOS aber nicht beides gleichzeitig. Jede Zielplattform nutzt ein eigenes Software Development Kit (SDK). In unserem Fall Android Studio.

1.18 Ionic Framework

Open Source SDK für die Entwicklung von Hybrid Mobile Apps. Dieses Framework baut auf AngularJS und Apache Cordova auf und erlaubt es mit Web Technologien, wie CSS3, HTML5 und Sass, erstellte Apps über die nativen App Stores zu verteilen. Ionic bietet Features der Mobile Development SDKs und erlaubt dem Nutzer seine Apps für Android oder iOS anzupassen und diese durch Cordova zu verteilen.

1.19 Apache Cordova

Populäres Mobile Framework, das anstatt auf plattformsspezifische APIs zurückgreift CSS3, HTML5 und JavaScript nutzt. Daraus resultieren hybrid Apps die weder native Mobile Apps noch reine Web-Apps sind. Cordova nutzt CSS3 und HTML5 für die Darstellung und JavaScript für die Logik der Applikationen.

1.20 Android Studio

Ist die offizielle integrated Development Environment der Android Plattform. Hier binden wir Ionic und Cordova ein um dann auf Grundlage des Ionic Frameworks die App zu entwickeln.

1.21 Vaadin

Vaadin ist ein freies Webanwendungs-Framework für RIA's. Im Gegensatz zu JavaScript-Bibliotheken und auf Browser-Plugins basierenden Lösungen bietet es eine serverseitige Architektur, was bedeutet, dass der Großteil der Programmlogik auf dem Server läuft. Auf der Client-Seite baut Vaadin auf dem Ajax-Framework Google Web Toolkit auf und kann damit erweitert werden.

Eines der Hauptmerkmale von Vaadin ist, dass der Softwareentwickler die gesamte Anwendung in Java schreiben kann. Das Framework umfasst ereignisgesteuerte Programmierung sowie Steuerelemente, die das Programmiermodell näher an die Modelle üblicher GUI-Software bringt als bei traditioneller Webentwicklung mit HTML und JavaScript.

1.22 Kennzahlen

Zusammenfassung von quantitativen, d.h. in Zahlen ausdrückbaren Informationen für den innerbetrieblichen (betriebsindividuelle Kennzahlen) und zwischenbetrieblichen (Branchen-Kennzahlen) Vergleich.

2 Konzepte

2.1 REST

Representational State Transfer (abgekürzt REST, seltener auch ReST) bezeichnet ein Programmierparadigma für verteilte Systeme, insbesondere für Webservices. REST ist eine Abstraktion der Struktur und des Verhaltens des World Wide Web. REST hat das Ziel, einen Architekturstil zu schaffen, der die Anforderungen des modernen Web besser darstellt. Dabei unterscheidet sich REST vor allem in der Forderung nach einer einheitlichen Schnittstelle von anderen Architekturstilen.

2.2 Rich Internet Application

In der Regel versteht man unter diesem Begriff Internetanwendungen, die eine reiche Menge an Interaktionsmöglichkeiten mit ihrer Benutzeroberfläche bieten. Insbesondere RIAs, die in Webbrowsern laufen, ähneln eher dynamischen Desktopanwendungen als klassischen (statischen) Webseiten.

2.3 Client-Server-Modell

Das Client-Server-Modell beschreibt eine Möglichkeit, Aufgaben und Dienstleistungen innerhalb eines Netzwerkes zu verteilen. Die Aufgaben werden von Programmen erledigt, die in Clients und Server unterteilt werden. Der Client kann auf Wunsch einen Dienst vom Server anfordern (z. B. ein Betriebsmittel). Der Server, der sich auf demselben oder einem anderen Rechner im Netzwerk befindet, beantwortet die Anforderung (das heißt, er stellt im Beispiel das Betriebsmittel bereit); üblicherweise kann ein Server gleichzeitig für mehrere Clients arbeiten.

2.4 Drools

Was ist Drools?

Drools ist ein in Java geschriebenes **Business Rule Management System** (BRMS). Ein **BRMS** entkoppelt die logischen Regeln, die das von der Anwendung zu lösende Problem beschreiben, von der technischen Realisierung und Implementierung der Anwendung. Die Regeln werden deklarativ definiert, sodass sich Anwendungsentwickler*innen keine Gedanken über die Realisierung der Regelauswertung machen müssen.

Hier ein kurzer Einblick in die Syntax einer Regeldefinition:

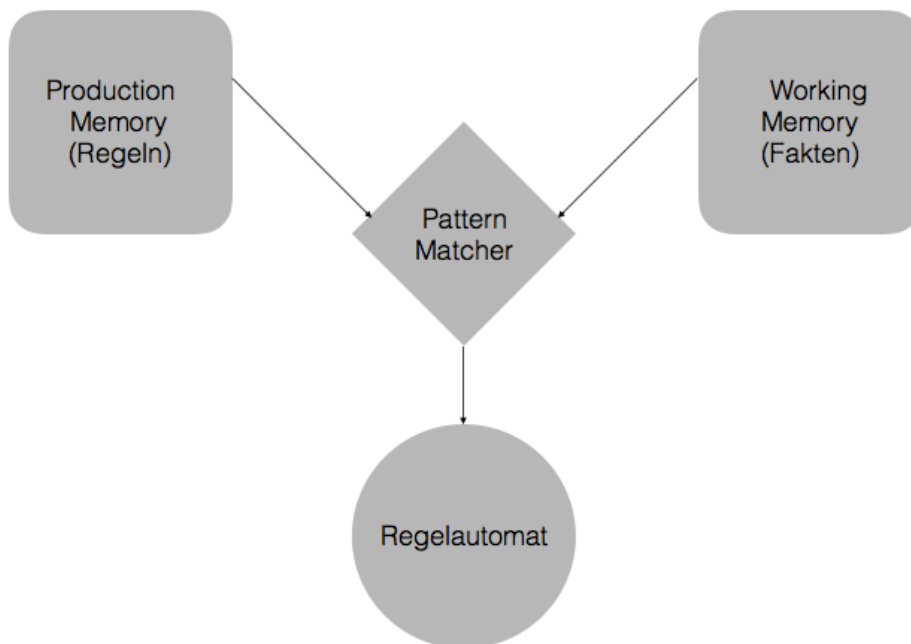
1. rule <rule_name>
2. <attribute> <value>
3. when
4. <condition>
5. then
6. <action>
7. end

Vorteile

Durch die Trennung der Logik des zu lösenden Problems und der Implementierung der Anwendung wird das Projekt leichter erweiterbar, da Regeländerungen ohne tiefgreifende Veränderungen im Programmcode durchgeführt werden können. Auch komplexere Probleme lassen sich einfacher durch die deklarative Definition der Regeln umsetzen. Dies führt dazu, dass Projektbeteiligte deren Arbeitsbereich nicht in der Anwendungsentwicklung liegt auch an der logischen Problemlösung arbeiten und somit ihre Kompetenzen direkter in die Anwendung einfließen lassen können.

Funktionsweise

Regeln werden in den 'Production Memory' geladen, auf den jederzeit zugegriffen werden kann. Fakten befinden sich im 'Working Memory' und werden durch den 'Pattern-Matcher' miteinander verglichen. Je nach Definition wertet dann der 'Regelautomat' die Ergebnisse aus und führt entsprechend der Regeldefinition Aktionen aus. [vgl. Grafik "DroolsFunktionsweise"]



2.5 Anwendung der Datenbank

Für das Logistik-Projekt werden alle Informationen über Pakete, Kunden, Kosten, Zeitpunkte, ... etc. in der DB abgespeichert. Die Datenbank liegt auf einem Server der Uni Leipzig. Diese DB besteht bereits und wir haben einen Datenbankdump zur Verfügung gestellt bekommen.

D.h., dass wir auf Änderungen in der DB reagieren, dort Daten abspeichern und lesen müssen. Daher muss eine ständige Verbindung zur DB gewährleistet sein.

2.6 Datenbankinstallation

Einrichtung der Datenbank, sodass auf sie zugegriffen werden kann; dies beinhaltet eigentliche Bereitstellung des Servers, auf dem die Datenbank liegt. In unserem Fall wurde das von der Uni-Leipzig (bzw. unseren Betreuern) übernommen. Außerdem sollte jedes Teammitglied auf die Datenbank über einen externen PC zugreifen können. Die Datenbank liegt auf einem Linux-Server und es wird über SSH eine Verbindung zum Server hergestellt. Dann wird auf der Linux-Shell des Servers eine MySQL-Shell aufgerufen. Auf dieser werden dann Datenbankbefehle abgesetzt. Im späteren Verlaufs des Projekts wird auf die Datenbank „automatisch“ über ein Programm zugegriffen, welches die Daten ausliest, ändert, etc..

3 Aspekte

3.1 Ziel des Projekts

Ziel des Projektes ist es eine Softwarelösung, genauer eine mobile App inkl. dazugehörigem Backend und Kontrolldashboard zu entwickeln, welche unterschiedliche Positionen im Speditionsunternehmen hinsichtlich ihrer Tätigkeiten in Zusammenhang mit bestimmten Geschäftsprozesse unterstützt. Vor allem Transport- und Versandprozesse sollen dabei erfolgskritisch anhand von Kennzahlen bewertet und entsprechend aufbereitet werden. Es sollen zum einen Disponenten einer Spedition in die Lage versetzt werden anhand der visuell aufbereiteten Kennzahlen entsprechend auf bestimmte Events innerhalb eines Transport und Lagerungsprozesses reagieren zu können. Zum anderen soll auch die Geschäftsleitung einen Überblick über die Gesamtlage der aktuellen Transporte bekommen.

3.2 Softwarearchitektur

Eine Softwarearchitektur beschreibt in der Informatik eine strukturierte oder hierarchische Anordnung der Systemkomponenten sowie ihrer Beziehungen. In diesem Projekt setzen eine mobile Anwendung(iOS und Android) und eine Webseite/Dashboard(vaadin) im Frontend auf ein Backend bestehend aus einer Datenbank(MySQL) und einer Business Rules Software(Drools) auf, welche auf einem Suse Enterprise Linux laufen. Das Webfrontend nutzt dabei einen Webserver(oder node.js?).

Quellen

<https://de.wikipedia.org/wiki/Programmierschnittstelle>

https://de.wikipedia.org/wiki/Representational_State_Transfer

<https://de.wikipedia.org/wiki/Framework>

<https://de.wikipedia.org/wiki/Informationssystem>

https://de.wikipedia.org/wiki/Apache_Tomcat

<https://de.wikipedia.org/wiki/SQL>

<https://de.wikipedia.org/wiki/Datenbankschnittstelle>

<https://de.wikipedia.org/wiki/Datenbanksprache>

<https://de.wikipedia.org/wiki/Datenbankdump>

https://de.wikipedia.org/wiki/Secure_Shell

<https://de.wikipedia.org/wiki/Servlet>

https://en.wikipedia.org/wiki/Web_container

<https://ionicframework.com/docs/>

<https://de.wikipedia.org/wiki/Framework>

https://en.wikipedia.org/wiki/Software_deployment

[https://en.wikipedia.org/wiki/Ionic_\(mobile_app_framework\)](https://en.wikipedia.org/wiki/Ionic_(mobile_app_framework))

<http://de.slideshare.net/onysame/regelbasierte-systeme-mit-jboss-drools-presentation>

<https://de.wikipedia.org/wiki/Gesch%C3%A4ftsregel-Managementsystem>

<https://de.wikipedia.org/wiki/Gesch%C3%A4ftsregel>