BWL2 Praktikum

Alex Mantel, Daniel Hofmeister

12. November 2014

Inhaltsverzeichnis

1 Erstes Praktikum

1.1 Architekturübersicht

Wir brauchen als Komponenten eine Warenverwaltung, eine Kundenverwaltung, eine Rechnungsverwaltung und eine Businesslogic mit einer Web-GUI.

1.2 UML-Sequenzdiagramm

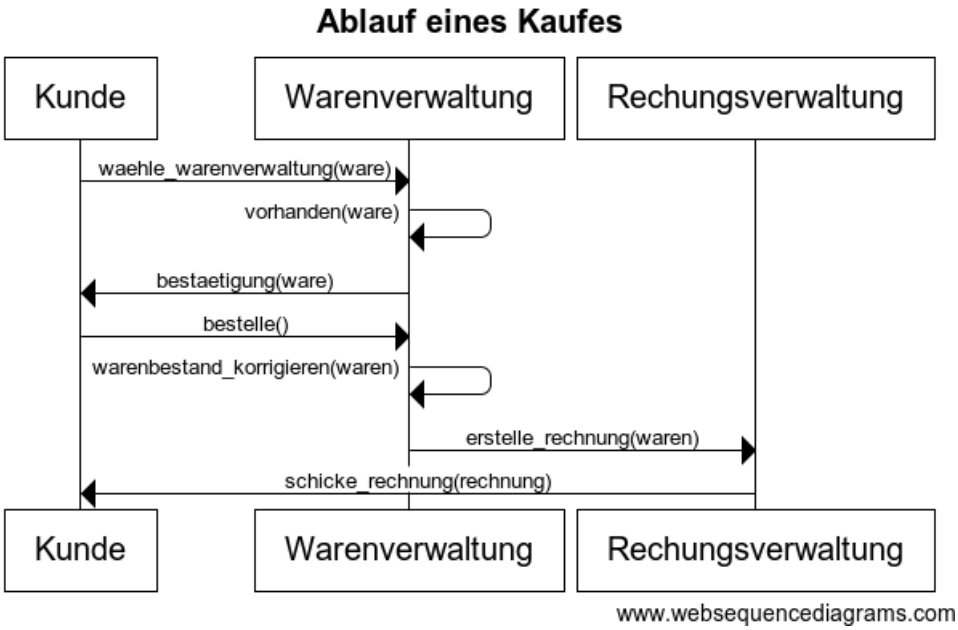


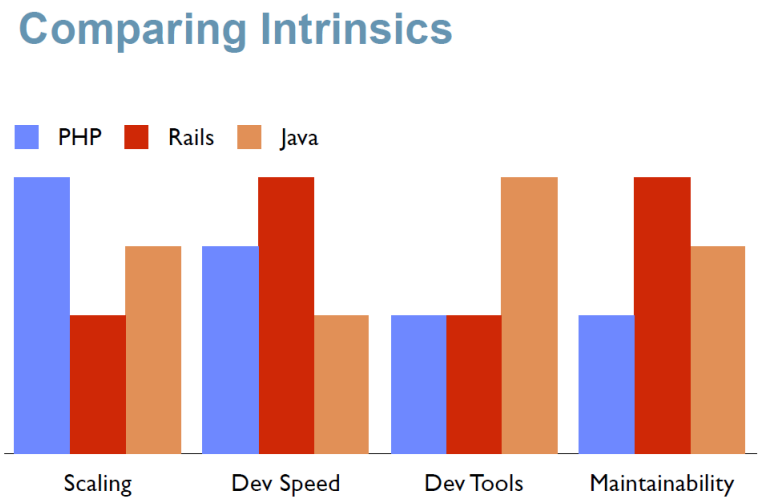
Abbildung 1: Ablauf eines Kaufes

Auf Abbildung 1 nicht gezeigt, das entfernen der ausgewählten Waren. Hier wird der Ablauf des Befüllen des Warenkorbs und der Ablauf der Bestellung gezeigt.

1.3 Begründung der gewählten Tenchnologien

Zur Debate stand, welche Programmiersprache bzw. welche Scriptsprache, welchen Webserver und welches Datenbankmanagementsystem wir für die Entwicklung des Webshops verwenden. Zur Option stellten wir uns hier aufgrund der Bekanntheit Ruby on Rails und PHP.

In Abbildung 2 zu sehen, ist ein Vergleich zwischen Java, Ruby on Rails und PHP. Wir werden aufgrund der Entwicklungsgeschwindigkeit, der Wartbarkeit und dem Grund, dass wir Ruby in Programmieren I verwendeten, Ruby on Rails verwenden. Offen bleibt nun, welches Datenbankmanagementsystem und welchen Webserver wir verwenden. Da Rails nativ einen Webserver bereitstellt, werden wir diesen verwenden. Die Anbindung an ein DMBS gestaltet Rails auch problemlos. Wir stellten uns SQLite und MySQL zur Option. Nach einigen Artikeln, welche diese Vergleichen fällt auf, dass MySQL eher für große Anwendun-



[Bra]

Abbildung 2: Vergleich zwischen Rails und PHP

gen geignet sind, welche auf Skalierbarkeit und Performanz Wert legen. SQLite hingegen soll sehr gut für Prototypen von Datenbanken, eine schnelle Entwicklung geeignet sein. Hierbei legt SQLite keinen Wert auf Nutzerverwaltung und Skalierbarkeit. Nachteile von MySQL ist, dass es eine höhere Komplexität in der Einrichtung aufweist. Beide verwenden offensichtlicher weise SQL. Letztendlich haben wir uns für MySQL entschieden, da wir den Umgang mit einem schwergewichtigen DBMS üeben möchten. In Ruby on Rails werden wir Gems verwenden, welche kleine Erweiterungen für das System sind. Die verwendeten Gems sind `paperclip` und `mysql`.

1.4 Design der Datenbank

Wir wurden gebeten eine Ware zu vertreiben, welche aus anderen Waren zusammengesetzt werden kann. Dieses Modell wird dadurch eine Rekursion enthalten, da wir die Bauteile der Produkte eventuell ebenfalls vertreiben würden. Interessant ist also die Ware mit ihrem Namen, einer Beschreibung, einem Bild der Ware und ihrer Zusammensetzung.

1.5 Dokumentation

1.5.1 Installation der Software Komponenten unter Arch Linux

Das Installieren der Softwarekomponenten hat sich unter Linux als äußerst einfach erwiesen. Unter der Distribution Arch Linux musste man zunächst MySQL, nodejs, ruby und die Gems von Ruby installieren. Die ersten drei waren mit dem Befehl `sudo pacman -S mysql nodejs ruby` abgehandelt. Für MySQL mussten wir zusätzlich `mysql_secure_installation` eingeben um das Passwort von root zu ändern. Nun muss man noch den Daemon mit `sudo systemctl start mysqld` starten. Wenn man möchte, dass der Daemon beim nächsten hochfahren des Rechners automatisch startet, gibt man zusätzlich `sudo systemctl enable mysqld` ein. Für die abschließende Installation der gems nutzten wir

`gem install mysql rails`. Um Ruby on Rails Bedienbarkeit zu verbessern haben wir zusätzlich die PATH Variable erweitert.

1.5.2 Installation der Software Komponenten unter Windows

1.5.3 Erstellen der Datenbank

Das Erstellen einer Datenbank erfolgt in Ruby on Rails implizit. Durch das Generieren von Models werden die Tabellen automatisch generiert. Hierfür verwenden wir den Generator von Rails. Mit `./rake db:create db:migrate` werden dann die erstellten Models in die Datenbank eingetragen.

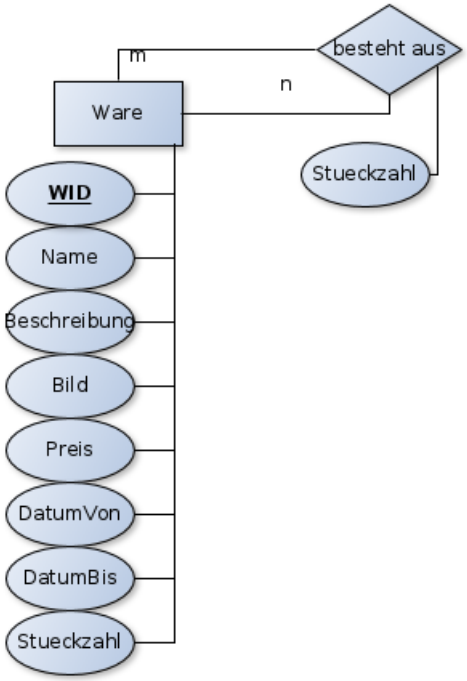


Abbildung 3: Zu sehen ist hier das Modell für die Ware und deren Bestandteile. Durch die die Kardinatlität m zu n können Waren sowohl aus mehreren, anderen Waren bestehen, als auch in welchen vorkommen.

2 Zweites Praktikum

3 Drittes Praktikum

4 Viertes Praktikum

Literatur

[Bra] Tim Bray. Issues of web frontends.