

---

# Entwicklerhandbuch

---

## Natural Beverages Webshop

<i>Author:</i>	<i>Author A</i>
<i>Matrikelnummer:</i>	<i>42</i>
<i>Fach:</i>	<i>RDB</i>
<i>Dozent:</i>	<i>Dozent A</i>

### **Zusammenfassung**

Ein Entwicklertagebuch zum RDB Projekt

## **Inhaltsverzeichnis**

<b>1</b>	<b>Einleitung - Konzept</b>	<b>3</b>
<b>2</b>	<b>Anleitung und Benutzung</b>	<b>3</b>
<b>3</b>	<b>Webseiten Design</b>	<b>4</b>
3.1	Design . . . . .	4
3.2	Wege . . . . .	4
<b>4</b>	<b>Programmtechnisches Design</b>	<b>5</b>
4.1	Datenbank . . . . .	5
4.2	Dateistruktur . . . . .	5
4.3	Authentication und User Management . . . . .	6
4.4	Admin Tool . . . . .	7
<b>5</b>	<b>Reflektion und Fazit</b>	<b>7</b>
<b>6</b>	<b>User Stories</b>	<b>7</b>

## **Abbildungsverzeichnis**

## 1 Einleitung - Konzept

Natural Beverages ist ein Webshop der sich auf "natürliche" Getränke spezialisiert. Also Getränke aus Früchten, Getreide und anderen Rohstoffen.

Bio, FairTrade, lokale Ernährung sind aktuelle Stichworte und gerade zu Zeiten von Energy Drinks und Coca Cola hat sich eine Gruppe gefunden, die auf "natürliche" Produkte besteht. In diesem Sinne stellt "Natural Beverages" die Möglichkeit da auch lokale Sorten und Getränke zu kaufen. Hauptkategorien stellen Bier, Wein, Tees und Säfte.

Die Zielgruppe liegt als vor allem bei Jugendlichen, Mitte bis Ende 20, die sich sehr gut mit mobiler Technologie auskennen.

## 2 Anleitung und Benutzung

- Entweder "admin/databaseScript.sql" in mysql mit dem Befehl "source" ausführen
- oder Servlets in "public\_html/WEB-INF/classes/" compilieren und mit "/localhost/admin/NBCreate.html" ausführen.
- Ab hier lässt sich die Website (localhost:8080/) benutzen.
- Oben rechts auf den Menüschalter, auf "Sign Up" und dann Daten eingeben.
- links neben dem Menü lassen sich Email und Passwort zum Registrieren eingeben.
- Nun können die Daten geändert oder unter Menü-¿Products Produkte zum Warenkorb hinzugefügt werden.
- Der Warenkorb wird ebenfalls im Menü-¿Cart geöffnet und kann hier bestätigt/ zurückgesetzt/ angeschaut werden.
- nach erfolgreicher Bestellung lässt sich diese unter Menü-¿Settings einsehen.
- Mit dem Ausloggen mittels des Schalters links neben den Einstellungen (oder via Menü) ist das Programm komplett.

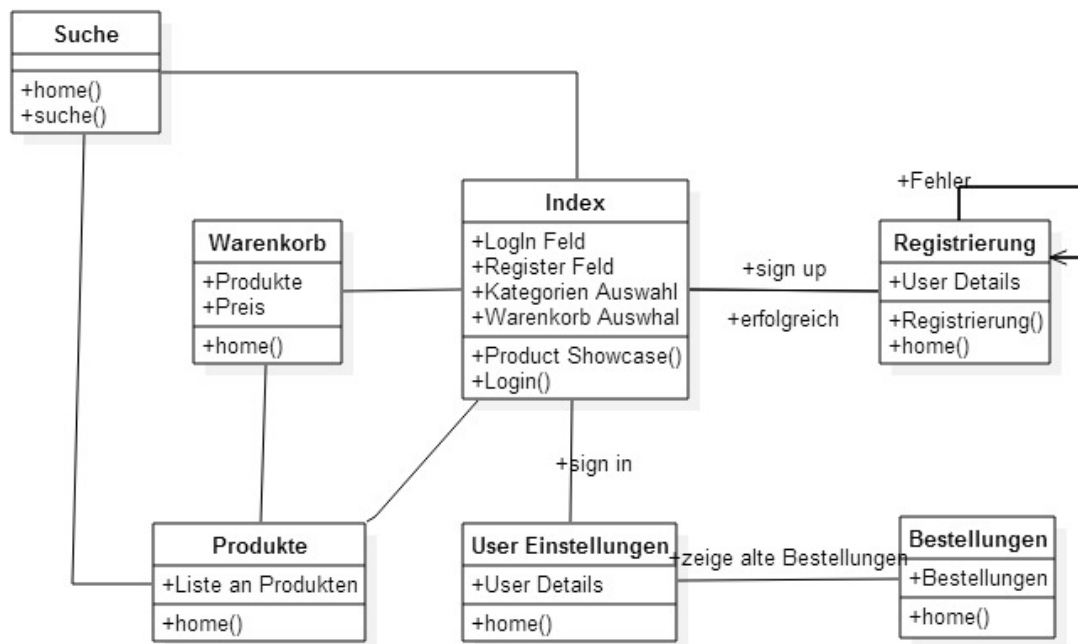
## 3 Webseiten Design

### 3.1 Design

Zum Darstellen der Elemente wird das CSS Framework Materialize (<http://materializecss.com/about.html>) genutzt. Begründung dafür liegt in der Zielgruppe. Es ist davon auszugehen, dass sich die Zielgruppe im Apple- oder im Googleumfeld bewegt. Android, iOS sind die Stichworte dieser Generation. Materialize holt das Webdesign in das aktuelle Aussehen von Google Software (Wie zum Beispiel auch das aktuelle Android 5).

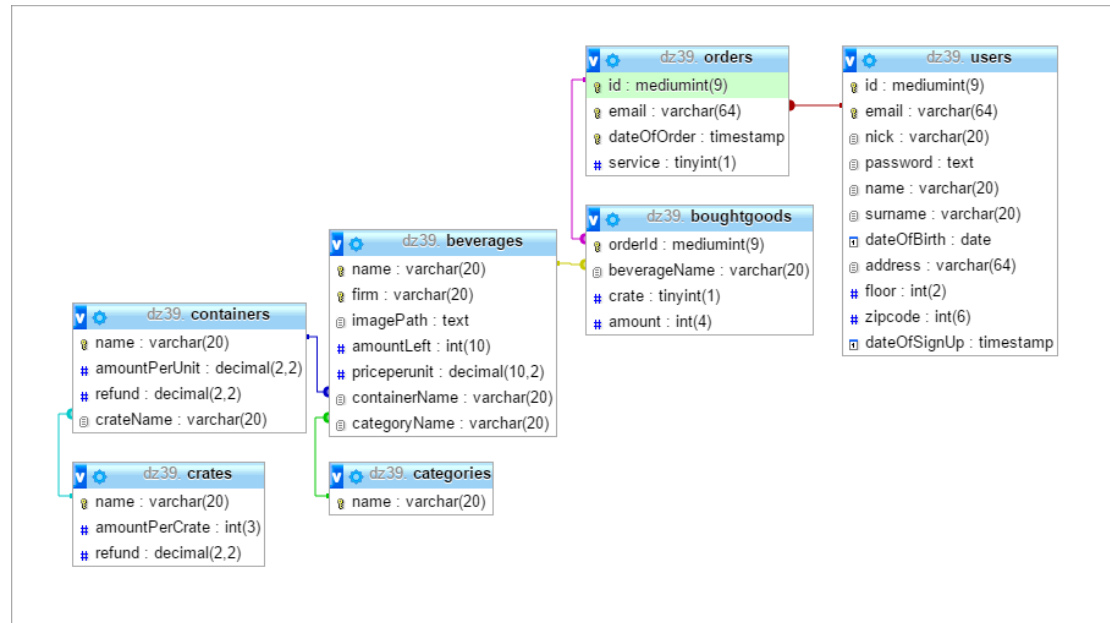
Gleichzeitig braucht das Design eine Note rustikal. Natur als Thema mit Technologie. Also grüne, braune Akzente. Ein dunkler Hintergrund hebt die hellen, freien Inhalte hervor.

### 3.2 Wege



## 4 Programmtechnisches Design

### 4.1 Datenbank



Bei der Datenbank wurde auf den Verzicht von Id's geachtet. Id's machten letztendlich nur bei den Orders, sowie den Users Sinn da, ein User am Tag mehrmals das gleiche bestellen kann und Date für einen Primary Key etwas unhandlich ist. Außerdem könnte sich die Email eines Users ändern.

Im Grunde teilt sich die Datenbank in drei Teile auf. Orders und Boughtgoods vereinigen dabei alle anderen: eine Bestellung eines Users enthält gekaufte Produkte mit Anzahl und Preis welche wiederum auf das Produkt zugreifen. Produkt, Category, Crate und Container beschreiben damit das Produkt insgesamt mit allen Anteilen.

Es lässt sich darüber streiten ob ein solche Datenbankkonstrukt in der reellen Welt einsatzbereit ist. Sobald eine Kategorie umbenannt werden soll entsteht ein riesiger Schaden, da das nicht einfach so geschehen kann, ohne das Informationen verloren gehen können. Mit einer id als Primary Key wäre dies jedoch ohne weiteres möglich.

### 4.2 Dateistruktur

- doc - enthält die Dokumentation (also Hauptsächlich dieses File, Graphen, ...).
- webapps/jsp - das eigentliche Programm, welches so auch im Praxi zu finden ist.

- admin - die Adminverwaltungsservlets zur Stammdatenverwaltung (siehe Admin Tool).
- css - die Stylesheets für die Website
- font - weitere Fonts für die Website (gehören zu materialize).
- images - Die Bilder der Website images/products für die Produkte.
- modules - enthält die jsp Seiten(teile)
- scripts - Javascriptscripte zum Darstellen von dynamischen Inhalten.
- WEB-INF/classes - enthält sowohl die servlet Klassen, als auch die Datenbankfiles. Außerdem wurden hier nochmal die \*.jars aus /WEB-INF/lib/ reinpackt.
- WEB-INF/lib - enthält eine json Lib (Die sollte bei einer aktuellen(!) Javaversion bereits vorhanden sein)

### 4.3 Authentication und User Management

Ein Login geschieht über einen Abruf und Abgleich der Datenbank. Das eingegebene Passwort wird in md5 gehasht und anschließend mit dem Datenbankeintrag (ebenfalls md5) abgeglichen. Vermutlich würde man in der realen Welt eine bessere Hashingmethode verwenden. Beim erfolgreichen Einloggen wird der Session eine Variable(id) gesetzt.

Ausloggen ist einfach nur das Entfernen dieser Variable.

Die ursprüngliche Idee war einen Filter(siehe /WEB-INF/NaturalBeverages/SessionFilter.java) zu benutzen. Sofern die SessionVariable "nick" (die beim einloggen gesetzt wurde) nicht null beträgt, wird der Client auf die angeforderte Seite weiter- oder ansonsten auf eine Fehlerseite geführt. Problem daran ist, dass auf dem Produktionsserver (praxi) Annotations nicht funktionieren und die web.xml auch nicht anfassbar war, da diese nur bei einem Serverneustart neu eingelesen wurde.

Somit dient eine einfache Abfrage über jeder entsprechenden Seite zur Kontrolle der Authentikation.

## 4.4 Admin Tool

Das "Admin Tool" ist ein auf Servlets basierendes Webtool mit dem sich Produkt Einträge aus der Datenbank hinzufügen und löschen lassen. Die Admintools werden durch den Zugriff auf /admin erreicht.

## 5 Reflektion und Fazit

Die Seite hat noch einige Baustellen. Zu den Produkten ließe sich eine allgemeine Übersicht, eine Suche hinzufügen. Die Adminverwaltung hätte besser ausfallen können (Listen auswählen, statt Text suchen). Es fehlt an gutem Design.

Außerdem ist das System der Anbindung nicht ganz durchdacht: Product.java dient nur als Halter für ein Produkt, zum Speichern des Warenkorbs wird dieses nochmal neu erstellt. Sinnvoller wäre es gewesen mit einer jsp Seite eine Klasse Products.java auszugeben, die die Produkte abgespeichert hat.

Die Arbeit mit den Servlets ist recht umständlich und selbst JSP Seiten kommen nicht an die Bequemlichkeit von Sinatra/Ruby On Rails Applications heran. Da durchgängig Probleme mit dem VPN, sowie Praxi(jsp Seiten wurden ständig gecached), gestaltete sich die Arbeit teilweise als Problematisch. Auf dem eigenem Netzwerk war die Arbeit aber durchaus sehr interessant.

## 6 User Stories

- As a client I want to browse the baverages of the shop
- As an older person I want the delivery service to carry my items into my apartment
- As a client I want to get a bill at the end of the order with the given address - I also want to see: - glass and crate refunds - delivery-into-apartment price per litre
- As a client I want to be able to find the beverages I am looking for
- As a client I want to be able to view a help page

- I want to have the comfort of a user account to store previous orders