

## Inhaltsverzeichnis

1	Beschreibung der Applikation.....	2
2	Datenbank.....	3
3	Ruby on Rails Server.....	5
4	Android Applikation.....	7

# 1 Beschreibung der Applikation

Die Applikation soll erlauben, mit Hilfe eines Web-Frontends basierend auf *Ruby on Rails*, mehrere Einkaufslisten anzulegen und diese zu Verwalten. Dabei können verschiedene Preise für die Artikel gespeichert werden sowie angegeben werden in welchem Markt diese erhältlich sind.

Das Anlegen der dazu nötigen Artikel, Adressen und Märkte soll vorerst ausschließlich über das Web Frontend geschehen.

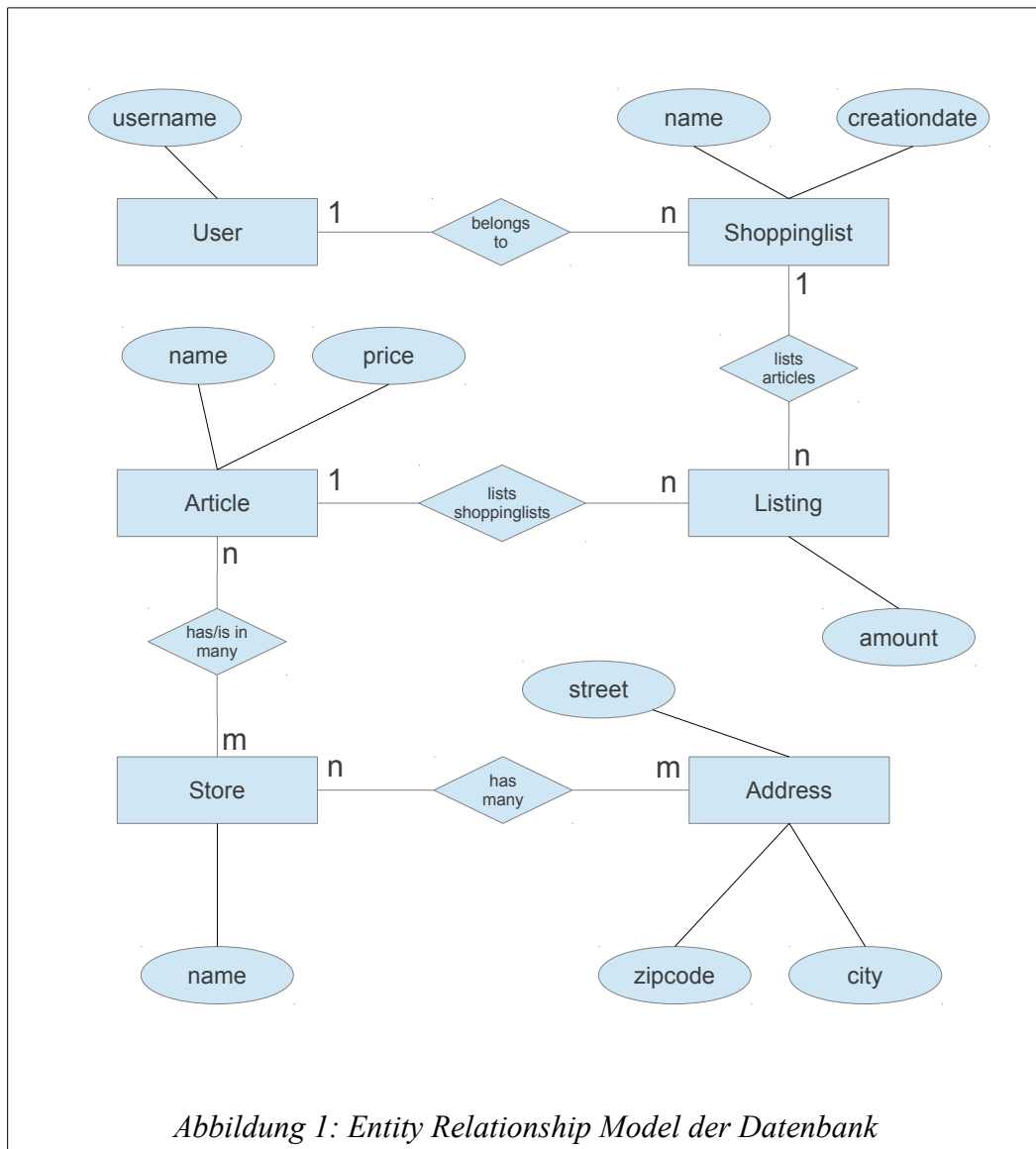
Die mit dem Webfrontend angelegten Einkaufslisten und die zugehörigen Daten können über eine *Android App* geladen werden. Dazu muss das *Android Smartphone* über *WLAN* mit dem *Ruby on Rails* Server verbunden sein.

Um die Daten synchronisieren zu können muss der Nutzer die lokale *IP-Adresse* des Servers wissen und in der *Android App* eintragen.

Nachdem die Daten auf das *Android Smartphone* übertragen wurden, können die Daten offline auf dem *Smartphone* genutzt werden.

Eine nähere Beschreibung folgt in den nächsten Kapiteln.

## 2 Datenbank



User können mehrere „*Shoppinglists*“ haben, diese werden gelöscht wenn der Benutzer gelöscht wird.

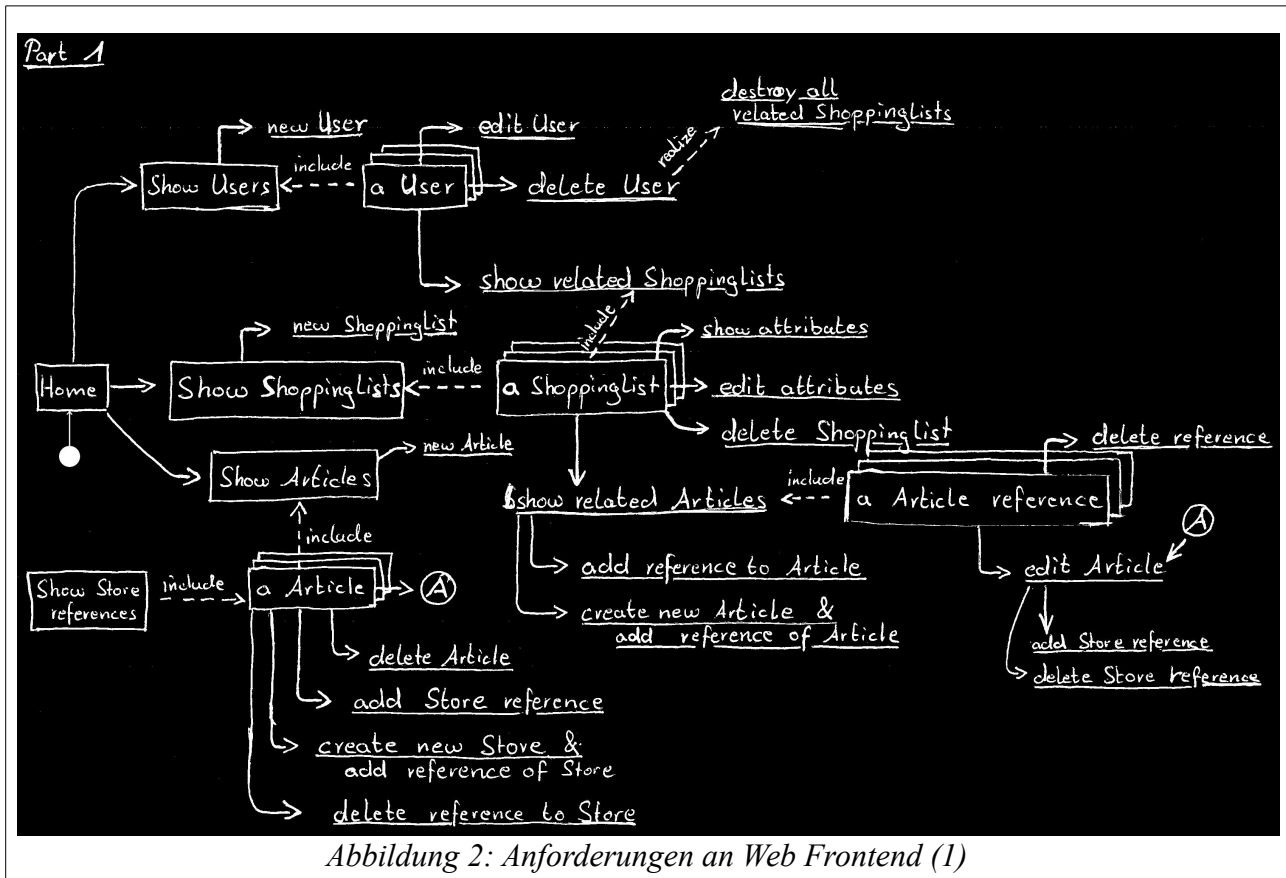
Da eine „*Shoppinglist*“ viele „*Artikel*“ und ein „*Artikel*“ in vielen „*Shoppinglists*“ aufgelistet werden kann, besteht zwischen diesen Entitäten eine *N:M* Beziehung. Jedoch wurde diese durch eine zusätzliche Tabelle aufgelöst, da in der „*Shoppinglist*“ jeder gelisteter Artikel eine Anzahl zugeordnet bekommen soll.

Die Beziehung zwischen „*Artikel*“ und „*Store*“ ist trivialerweise *N:M*, da ein bestimmter „*Artikel*“ in verschiedenen „*Stores*“ vorkommen kann und ein

*Store* mehr als einen *Artikel* zugeordnet hat.

Die *N:M* Beziehung zwischen *Store* und *Address* kommt zustande, da viele *Stores*, wie zum Beispiel eine Supermarktkette, oft mehrere Standorte hat in denen zudem das Artikelangebot gleicht ist. Zudem können an einer *Adresse* mehrere *Stores* sein, wie es in Einkaufszentren üblich ist.

### 3 Ruby on Rails Server

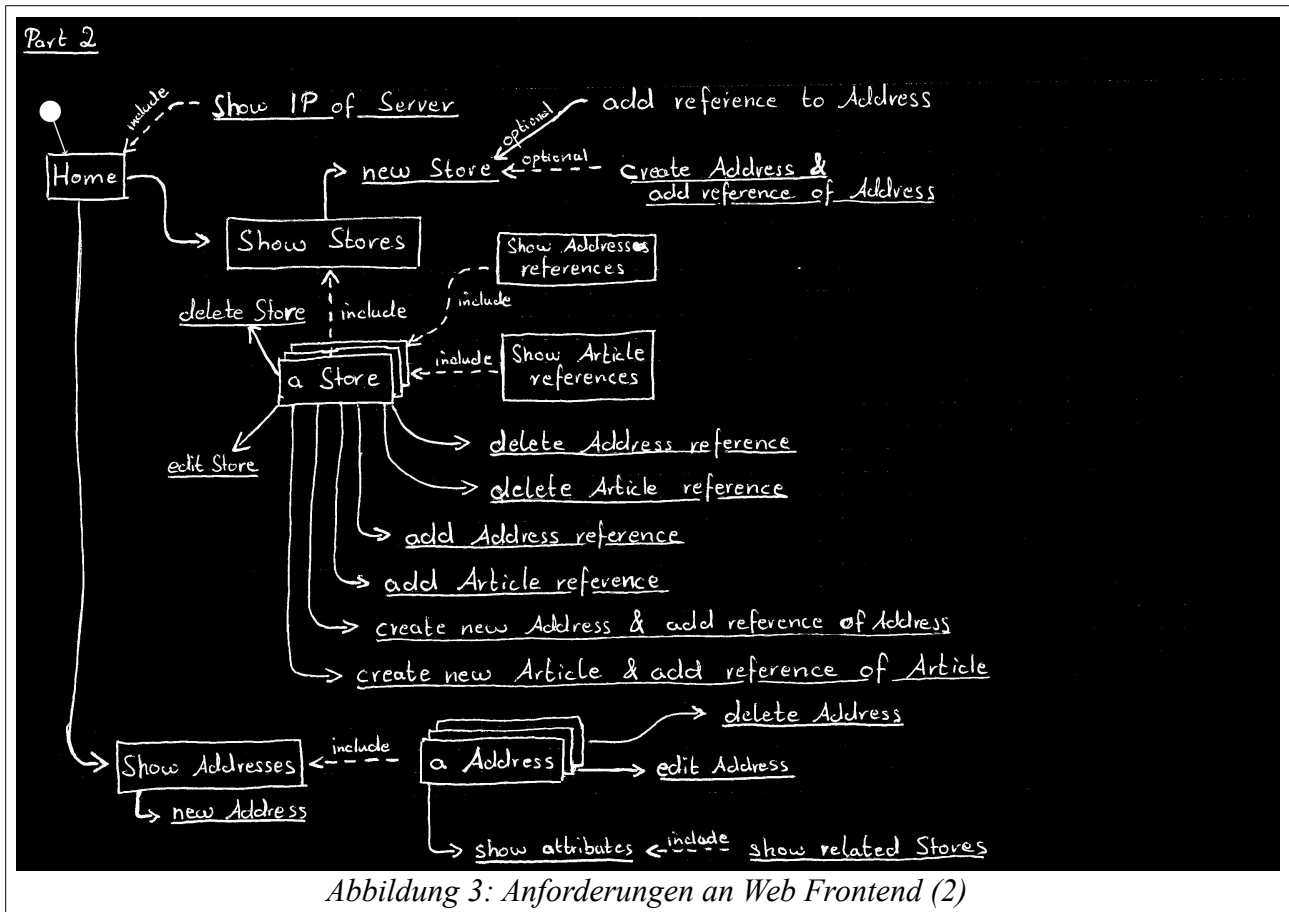


Wie im Kapitel „Beschreibung der Applikation“ beschrieben soll das Web Frontend eine umfangreiche Verwaltung der Datenbank erlauben. Somit soll es für jede Entität möglich sein diese zu Verwalten. In Abbildung 2:

Anforderungen an Web Frontend (1) und Abbildung 3: Anforderungen an Web Frontend (2) ist zu sehen dass von der Hauptseite aus, jede Entität (ohne *Listing*) über eine Seite erreicht werden kann, die alle enthaltenen Elemente anzeigt.

Zudem können verschiedene Aktionen (unterstrichen) für jede der aufgeführten Entitäten ausgeführt werden . Dabei soll nicht nur das Hinzufügen einzelner Elemente möglich sein sondern auch Beziehungen zwischen den Elementen der Entitäten erstellt werden, basierend auf Abbildung 1: Entity Relationship Model der Datenbank.

Die Entität *Listing* kann jedoch nur bearbeitet werden wenn eine konkrete *Shoppinglist* vom Nutzer ausgewählt wurde.



Somit kann immer nur das Mapping der Artikel mit der konkreten Shoppinglist verwaltet werden.

Die Hauptseite *Home* dient zum einen dazu um die IP Adresse des Servers im lokalen Netzwerk anzuzeigen und ist zudem die einzige Seite von der aus alle Views der Entitäten erreicht werden können.

Die angezeigte IP-Adresse ist nötig um die Android Applikation mit der Datenbank zu synchronisieren.

## 4 Android Applikation

In Kapitel 1 wurde bereits allgemein die Funktionsweise der Android Applikation beschrieben.

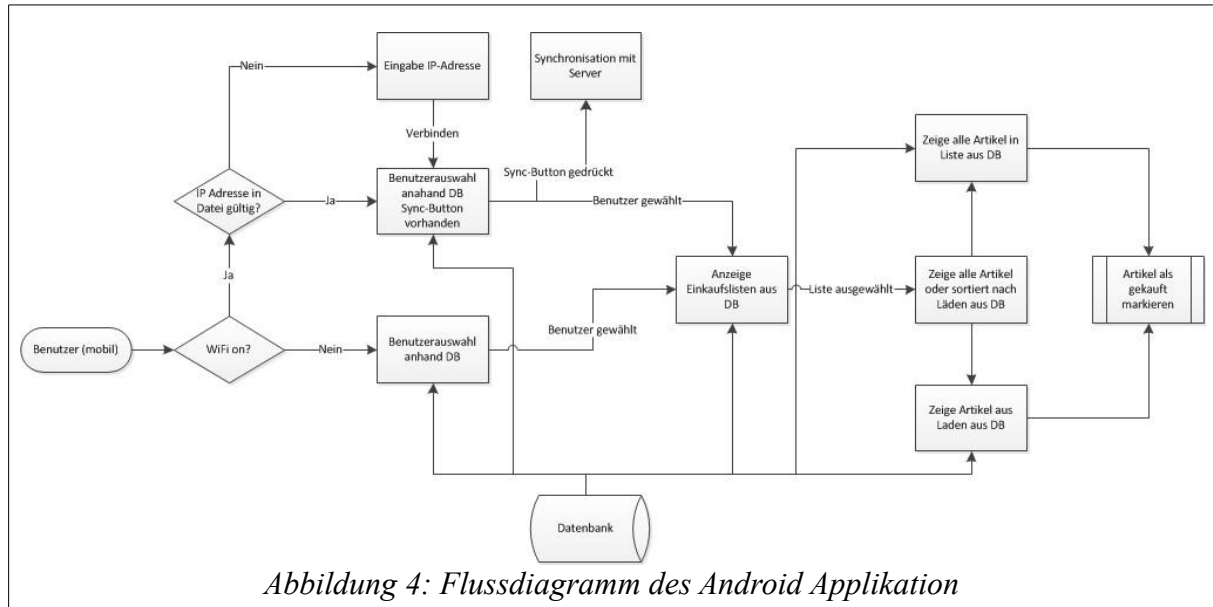
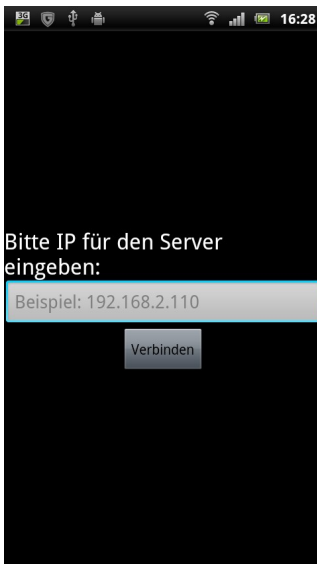


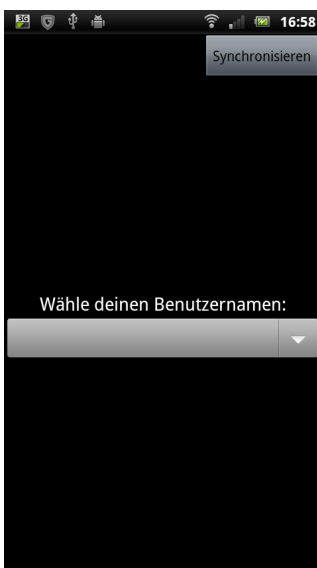
Abbildung 4: Flussdiagramm des Android Applikation

Das oben abgebildete Flussdiagramm zeigt die Benutzerführung in der Android App. Beim erstmaligen Start sollte Wifi (WLAN) an sein, damit der Benutzer eine Synchronisation mit dem Server durchführen kann. Um nicht jedes Mal bei einem Start mit Wifi die IP-Adresse erneut eingeben zu müssen, wird die IP-Adresse gespeichert und beim nächsten Start auf Gültigkeit und Erreichbarkeit geprüft.

Die folgende Abbildung zeigt den Startbildschirm mit aktiven Wifi.

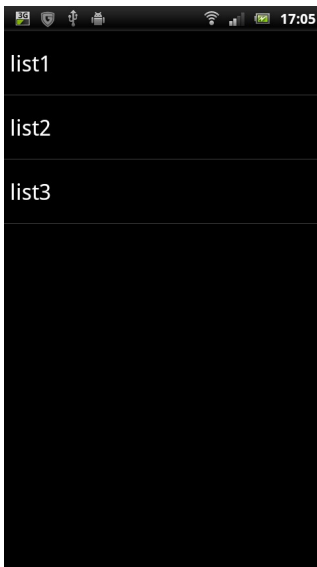


Nachdem die IP-Adresse eingegeben und eine Verbindung hergestellt wurde bzw. die hinterlegte IP erreichbar ist, kann der Benutzer entweder eine Synchronisation durchführen oder, falls Daten bereits vorhanden sind, einen Benutzer auswählen (Abb.).

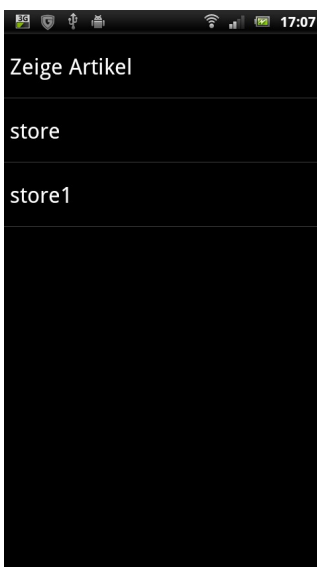


Ist der Benutzer offline, so wird der Synchronisieren-Knopf nicht angezeigt. Wurde ein Benutzer ausgewählt, werden die Einkaufslisten des gewählten Benutzers angezeigt.





Nachdem eine Liste ausgewählt wurde, hat der Benutzer die Möglichkeit entweder sich alle Artikel der Liste anzeigen zu lassen, oder die eines bestimmten Ladens.



Folgende Abbildung zeigt den Menüpunkt „Zeige Artikel“.



Möchte der Benutzer einen Artikel als „im Warenkorb“ markieren, so tippt er einfach auf diesen Artikel. Dadurch wird dem Artikel rein visuell ein „XX“ voran- und nachgestellt. Man hat auch die Möglichkeit, falls ein falscher Artikel gedrückt wurde, die Markierung aufzuheben.



Anmerkung:

Getestet wurde die Android Applikation mit einem Sony (Ericsson) Xperia Arc mit Android 2.3.4 (Kernel Version: 2.6.32.9-perf; Build-Nummer: 4.0.2.A.0.62).