



CodeCamp: Context Awareness, Entwicklung einer Android-App

Adrian Kunz, Jan Bingemann, Johann Feser, Michael Prasil, Sven Starcke

Kassel, den 11. März 2019

Inhaltsverzeichnis

1	Einleitung	3
1.1	Ziele	3
1.2	Funktionsweise	3
1.3	Überblick	3
1.4	Inhalt	3
2	Anwendung	3
2.1	Start	3
2.2	Navigation	3
2.3	Nutzer-bearbeiten-Screen	5
2.4	Shop-Screen	5
2.4.1	Artikel zum Einkaufswagen hinzufügen	6
2.4.2	Artikel aus dem Einkaufswagen entfernen	6
2.4.3	Verhalten des Einkaufswagen-Buttons	6
2.4.4	Artikel kaufen	6
2.4.5	Barcode von Artikeln scannen	6
2.5	Shop-Screen (Zusätzliche Administrator Funktionalitäten)	7
2.5.1	Artikel bearbeiten	7
2.5.2	Artikel hinzufügen	7
2.5.3	Barcode von Artikeln scannen	8
2.6	Einkaufs-Historie-Screen	8
2.7	Nutzer-Screen (Administrator Funktionalität)	9
2.8	Statistiken-Screen	9
2.9	Einstellungen-Screen	10
2.10	Aktualisieren von Server-Daten	10
2.11	Unterstützte Sprachen	11
2.12	Shortcut-Funktion zum Scannen von Artikeln	11
3	Architektur	11
3.1	Datenmodell	11
3.2	Frontend-Architektur	11
3.2.1	Aufbau der MainActivity	11
3.2.2	Fragmente	12
3.2.3	Toolbar	12
3.2.4	Navigation-Drawer	13
3.2.5	Ausnahmen	13
3.2.6	Kommunikation mit dem Backend-Layer	13
3.2.7	RecyclerViews	14
3.3	Backend-Architektur	14
3.3.1	Base-Layer	14
3.3.2	Abstraction-Layer	14
3.3.3	Access-Layer	14
3.3.4	Tests	14
4	Anhang	15
4.1	Bibliotheken	15

1 Einleitung

1.1 Ziele

Ziel des Projektes war es, im Rahmen der Veranstaltung „Code Camp 1: Context Awareness“ eine Android-basierte Anwendung zur Verwaltung und Aufzeichnung der Einkäufe, welche in der ComTec-Küche getätigt werden, zu erstellen.

1.2 Funktionsweise

Auf Basis einer Datenbank können Gegenstände angelegt, gekauft und gelöscht werden. Dazu benötigte Nutzer werden ebenfalls in der Datenbank gespeichert. Der Zugriff auf diese erfolgt über eine REST-API, welche alle notwendigen Operationen bereitstellt.

1.3 Überblick

Nutzer erhalten bei der Registrierung ein Token, welches für den Zugriff über die REST-API benötigt wird. Dieses wird in der Anwendung zwischengespeichert, sodass ein Nutzer mit gültigem Token automatisch angemeldet wird, wenn die Anwendung startet. Der Fokus der Anwendung liegt auf dem Kauf von Gegenständen. Dies ist auf zwei Arten möglich:

1. regulär in der App: Über eine Liste der verfügbaren Gegenstände können solche per Klick zum Warenkorb hinzugefügt werden.
2. über einen eingebetteten Barcode-Scanner können Gegenstände, deren Barcode bereits in der Datenbank registriert ist dem Warenkorb hinzugefügt werden.

Es ist dem Nutzer zudem möglich eine Statistik zur Kaufhäufigkeit der registrierten Gegenstände einzusehen. Zwecks Verwaltung ist es als Administrator ausgezeichneten Nutzern möglich neue Gegenstände wahlweise mit automatisch generierter ID oder dem Barcode-Scanner anzulegen.

1.4 Inhalt

In Abschnitt 2 werden die von der App bereitgestellten Features vorgestellt. Abschnitt 3 erklärt die Anwendungsstruktur aus technischer Sicht, jeweils für Datenmodell (Abschnitt 3.1), die Oberflächen-Architektur (Abschnitt 3.2) und die zugrunde liegende Grundstruktur inklusive Netzwerkverkehr (Abschnitt 3.3).

2 Anwendung

2.1 Start

Beim Öffnen der Anwendung wird geprüft, ob sich in den Shared-Preferences des Android-Geräts bereits ein gültiges User-Token und eine passende User-Id befindet. Ist dies der Fall wird der Benutzer anhand dieser eingeloggt. Andernfalls öffnet sich zunächst der Login-Screen.

Es wird ein neuer Nutzer auf Basis der in der Oberfläche eingegebenen Anmeldedaten (Name, E-Mail, Admin-Flag, siehe Abbildung 1) erstellt. Das User-Token wird für die weitere Verwendung in den Shared-Preferences zwischengespeichert. Der Login-Screen schließt sich wieder.

Sowohl im Fall des automatischen Logins, als auch nach der Anmeldung im Login-Screen, öffnet sich dann der Shop-Screen.

2.2 Navigation

Mittels des Kontext-Menu-Button in der oberen, linken Bildschirmecke des Shop-Screens oder durch Wischen nach Rechts erscheint ein Navigation Drawer.

Im oberen Bereich des Navigation Drawers befinden sich Informationen zum Benutzer (Der Nutzername, die Email-Adresse und das Guthaben). Falls der Nutzer ein Administrator ist, wird dies hinter dem Namen angezeigt. Über den Stift-Button auf der rechten Seite, wird der Nutzer-bearbeiten-Screen geöffnet. Dieses ermöglicht das Bearbeiten des eigenen Benutzer-Profiles.

Darunter zeigen sich viele Buttons, mit denen zu allen Screens der App navigiert werden kann:

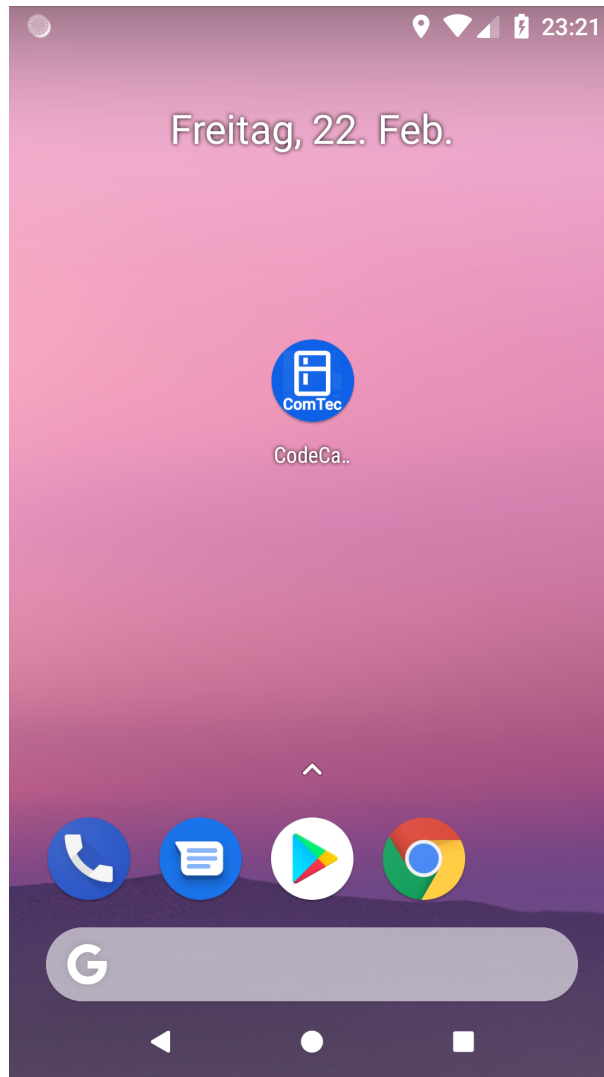


Abbildung 1: login-screen

- Shop-Button: Der Shop-Screen wird geöffnet. Dies ist der zentrale Screen der App, da hier Artikel gekauft werden können.
- Einkaufs Historie-Button: Der Einkaufs-Historie-Screen öffnet sich, in welchem alle vergangenen Einkäufe nach Datum und Zeit sortiert aufgelistet werden (siehe Abbildung ??).
- Nutzer-Button (Administrator): Dieser Button ist nur verfügbar, wenn der Benutzer ein Administrator ist. Klickt er auf diesen Button, öffnet sich das Nutzer-Fragment, das die Liste aller Nutzer zeigt.
- Statistiken-Button: Wird dieser Button geklickt, öffnet sich der Statistiken-Screen, welches Statistiken zum Verhalten des Benutzers liefert.
- Einstellungen-Button: Über diesen Button kann der Einstellungen-Screen geöffnet werden. In diesem kann ein Dunkel-Modus aktiviert werden.

Der unterste Button, der mit dem Label "Logout-Button" versehen ist, hat nichts mit der Navigation zwischen Screens zu tun. Bei einem Klick auf diesen Eintrag werden die in den Shared-Preferences gespeicherten Anmeldedaten gelöscht. Der Benutzer wird zum Login-Screen weitergeleitet, um sich einen neuen Account zu erstellen. Auf Grund des Designs des REST-Servers ist eine erneute Anmeldung nicht möglich.

2.3 Nutzer-bearbeiten-Screen

Im Nutzer-bearbeiten-Screen kann der Benutzer sein eigenes Profil bearbeiten. Er kann die folgenden Daten verändern:

- Den Nutzernamen
- Die Email-Adresse
- Das eigene Guthaben, falls er ein Administrator ist

Tippt der Benutzer auf den Speicher-Button, wird das Nutzer-Profil gespeichert und es öffnet sich der Nutzer-Screen.

2.4 Shop-Screen

Nach erfolgreicher Anmeldung erfolgt eine Weiterleitung auf den Shop-Screen. Dieser zeigt eine Liste der kaufbaren Gegenstände (siehe Abbildung 2).

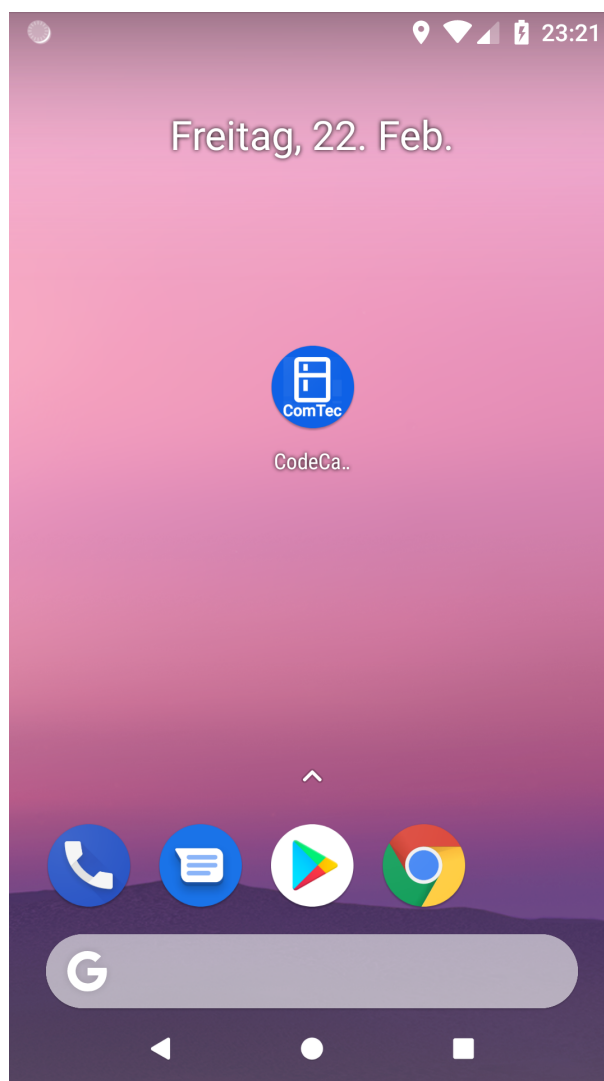


Abbildung 2: Haupt-Screen

Ein Eintrag in dieser Liste enthält folgende Komponenten:

- Den Namen des ausgewählten Gegenstands
- Den Preis für einen Gegenstände des ausgewählten Typs

- Die ausgewählte Anzahl der Gegenstände des ausgewählten Typs, welche man kaufen möchte. Sie erhöht sich mit jedem Klick auf den Eintrag. Wischt man hingegen von rechts nach links über den Eintrag, so wird die ausgewählte Anzahl auf 0 gesetzt. Die Anzahl wird nur angezeigt, wenn sie größer als 0 ist.
- Der Gesamtpreis für den Gegenstandstyp. Hierbei handelt es sich um den Kumulierten Preis (ausgewählte Anzahl · Preis) des Gegenstands. Dieser wird nur angezeigt, wenn mehr als ein Gegenstand des ausgewählten Typs verlangt wird.
- verfügbare Anzahl: Zeigt an, wie viele Gegenstände des ausgewählten Typs gekauft werden können. Sie kann aufgrund des Design der REST-API negativ werden.

Die Liste ist nach der Art der Gegenstände sortiert (Beispiel: Wasser, Saft, ...). Als Überschrift ist jeder Gruppierung von Artikeln derselben Art ein Listeneintrag vorangestellt, der den Namen der Art trägt. Tippt der Benutzer auf eine Überschrift, faltet sich die Kategorie zusammen. Bei erneutem Tippen werden die Artikel wieder angezeigt.

Auf dieselbe Art funktionieren alle Listen der App mit Kategorien.

2.4.1 Artikel zum Einkaufswagen hinzufügen

Tippt der Benutzer auf einen Listeneintrag, z.B. Zitronenlimonade, wird ein Exemplar in den Warenkorb gelegt. Der Benutzer kann einen Listeneintrag so oft hinzufügen, wie Exemplare verfügbar sind. Die Anzahl, der von einem Typ zum Einkaufswagen hinzugefügten Exemplare wird auf dem Listeneintrag abgebildet. Sind zwei oder mehr Exemplare eines Artikels im Warenkorb, wird der Gesamtpreis der ausgewählten Exemplare des Artikels auf dem Listeneintrag abgebildet.

2.4.2 Artikel aus dem Einkaufswagen entfernen

Artikel können aus dem Einkaufswagen entfernt werden, indem der entsprechende Listeneintrag nach links gewischt wird. Dabei zeigt sich unter dem Listeneintrag ein Mülleimer-Symbol auf dunklem Grund, welches den Vorgang verdeutlicht. Es werden dann alle Exemplare des Artikels aus dem Warenkorb entfernt.

2.4.3 Verhalten des Einkaufswagen-Buttons

Der Einkaufswagen-Button schwebt über der Artikel-Liste. Durch das Klicken auf diesen kann man den Einkaufswagen in einem separaten Screen betrachten und den Kauf tätigen.

Der Button zeigt sich erst, sobald sich mindestens ein Element im Warenkorb befindet. Werden alle Elemente aus dem Warenkorb entfernt, verschwindet der Button wieder.

Befindet sich mehr als ein Element im Warenkorb, verschwindet der Einkaufswagen-Button temporär, sobald der Benutzer nach unten scrollt, damit untere Listenelemente nicht vom Button verdeckt werden. Scrollt der Benutzer nach oben, erscheint der Einkaufswagen-Button wieder.

2.4.4 Artikel kaufen

Wenn alle gewünschten Gegenstände ausgewählt sind, wird durch einen Klick auf den Einkaufswagen-Button der Warenkorb-Screen geöffnet. Hierbei handelt es sich um eine Darstellung nur der ausgewählten Gegenstände mit der bereits bekannten Eintragsstruktur. Auch hier können durch einen Klick auf die Einträge Gegenstände hinzugefügt und durch Wischen über einen Eintrag alle Gegenstände dieses Typs entfernt werden. Durch einen Klick Löschen-Button kann der komplette Einkauf gelöscht werden. Durch einen Klick auf den Submit-Button (Haken) erscheint ein Modal-Fenster (siehe Abbildung ??) mit einer Abfrage, ob der Kauf durchgeführt werden soll. Hier wird nochmals der Gesamtpreis des Einkaufs angezeigt. Beim Klick auf „yes“ wird der Kauf getätigt und man wird auf den Shop-Screen weitergeleitet. Durch ein Pop-Up am unteren Bildschirmrand wird gezeigt, dass der Kauf erfolgreich war (siehe Abbildung ??).

2.4.5 Barcode von Artikeln scannen

Durch einen Klick auf das Kamera-Icon in der oberen rechten Ecke des Shop-Screens, das in Abbildung 2 zu sehen ist, öffnet sich die Kamera des Gerätes mit einem eingebauten Barcode-Scanner, wie in Abbildung 3. Hiermit kann der Benutzer Barcodes einscannen und die Artikel bequemer erwerben.

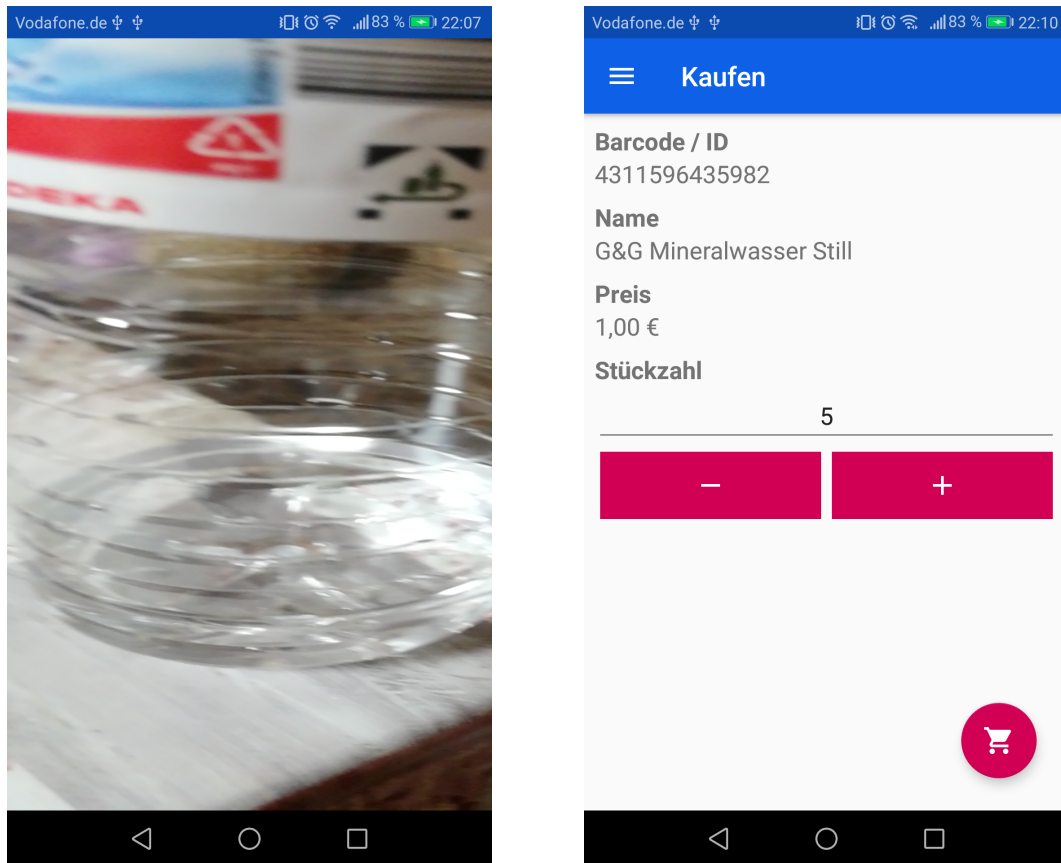


Abbildung 3: Barcode Scanner

2.5 Shop-Screen (Zusätzliche Administrator Funktionalitäten)

Ist der Benutzer ein Administrator, ist in der Toolbar ein Stift-Symbol zu sehen. Mit diesem lässt sich der Bearbeitungs-Modus aktivieren. Dann verschwindet in der Toolbar das Artikel-Scannen-Symbol und wird ersetzt durch ein Plus-Symbol. Außerdem färbt sich die Toolbar, um zu signalisieren, dass sich der Benutzer im Bearbeitungs-Modus befindet. Tippt der Benutzer erneut auf das Stift-Symbol, wird der Bearbeitungs-Modus deaktiviert. Die Toolbar wird wieder blau gefärbt und das Plus-Symbol wird wieder durch das Artikel-Scannen-Symbol ersetzt.

2.5.1 Artikel bearbeiten

Befindet sich der Benutzer im Bearbeitungs-Modus und tippt auf einen Listeneintrag so öffnet sich, wie es in Abbildung 4 rechts zu sehen ist, der Artikel-Bearbeiten-Screen. Der Benutzer hat nun die Möglichkeit die folgenden Eigenschaften des Artikels zu ändern:

- Der Name des zu erstellenden Gegenstands
- Der Preis des zu erstellenden Gegenstands
- Die Menge, in der der Gegenstand verfügbar ist
- Die Art des Gegenstands. Hier wird aus einem Drop-Down-Menu eine Art ausgewählt.

Der Barcode selbst kann nicht bearbeitet werden. Tippt der Benutzer auf den Bearbeiten-Button, wird er zum Shop-Screen im Bearbeitungs-Modus weitergeleitet, der den bearbeiteten Artikel enthält.

2.5.2 Artikel hinzufügen

Ist der aktuelle Benutzer als Administrator eingeloggt, kann dieser mit dem + Knopf im oberen Banner des Shop-Screens, wie in Abbildung 4 links demonstriert, neue Artikel hinzufügen. Dazu müssen für jeden neuen Artikel die folgenden Daten eingegeben werden:

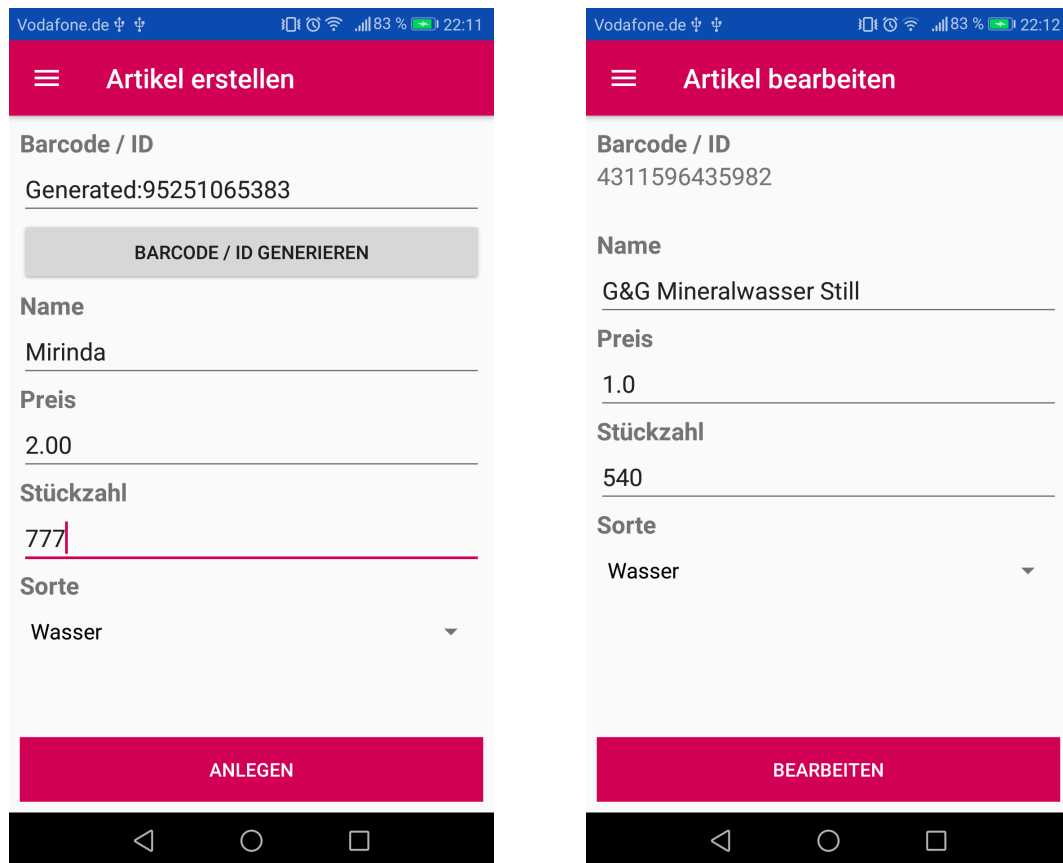


Abbildung 4: Anlegen eines neuen Artikels / Bestehenden Artikel bearbeiten

- Der Barcode des Artikels. (Anstatt selbst einen Barcode zu wählen, kann durch Tippen des Barcode/ID-Generieren-Buttons ein zufälliger Barcode generiert werden.)
- Der Name des neuen Artikels.
- Der Preis des neuen Artikels.
- Die Menge, in der dieser Artikel auf Vorrat ist.
- Die Warengruppe des Artikels. Hier wird aus einem Drop-Down-Menu eine Art ausgewählt.

Wird auf den Anlegen-Button getippt so wird der Artikel erstellt. Es öffnet sich wieder der Shop-Screen im Bearbeitungs-Modus, dessen Artikel-Liste den neu erstellten Artikel enthält.

2.5.3 Barcode von Artikeln scannen

Wird ein Barcode von einem Administrator gescannt so wird die App nach schauen ob es zu diesem Barcode bereits einen Artikel gibt oder nicht. Ist dieser Artikel bereits im Sortiment so kann der Administrator diesen wie aus Abbildung 3 gewohnt einfach erwerben. Ist dieser Artikel jedoch noch nicht im Sortiment wird dem Administrator, wie in Abbildung 5 geschildert, nun die Möglichkeit angeboten diesen Artikel dem Sortiment hinzuzufügen.

2.6 Einkaufs-Historie-Screen

Im Einkaufs-Historie-Screen sieht der Benutzer alle seiner getätigten Transaktionen in einer Liste.

Ein Listeneintrag stellt einen oder mehrere Artikel einer Sorte dar, die in der Transaktion gekauft wurden. Er enthält folgende Komponenten:

- Den Namen des Artikels
- Den Barcode des Artikels

Abbildung 5: Barcode Scanner - Einfach neue Artikel anlegen

- Die Anzahl, die man von diesem Artikeltyp gekauft hat.
- Der Gesamtpreis, der für diesen Artikeltyp anfällt. Hierbei handelt es sich um den Kumulierten Preis (ausgewählte Anzahl · Preis) des Artikels. Dieser wird nur angezeigt, wenn mehr als ein Artikel des Typs gekauft wurde.

Die Listeeinträge einer Transaktion sind nach dem Datum der Transaktion gruppiert.

2.7 Nutzer-Screen (Administrator Funktionalität)

Der Nutzer-Screen ist nur im Navigation-Drawer zu sehen, falls der Benutzer ein Administrator ist.

Im Nutzer-Screen hat der Benutzer einen Überblick über alle Nutzer, die beim Server angemeldet sind. Sie sind nach den Anfangsbuchstaben ihrer Benutzernamen gruppiert und alphabetisch geordnet.

Tippt der Benutzer auf einen List-Eintrag, öffnet sich der Nutzer-bearbeiten-Screen, der in Abschnitt 2.3 erläutert wurde. Statt des eigenen Benutzer-Accounts, kann der Account des ausgewählten Benutzers bearbeitet werden.

2.8 Statistiken-Screen

Der Statistiken-Screen ist über die Navigation erreichbar. Dieser stellt den Administratoren als auch den Benutzern der App unterschiedliche Informationen grafisch dar. Ein Administrator kann dort einsehen welche Artikel wie oft gekauft wurden und somit einen Überblick über die verkauften Stückzahlen für

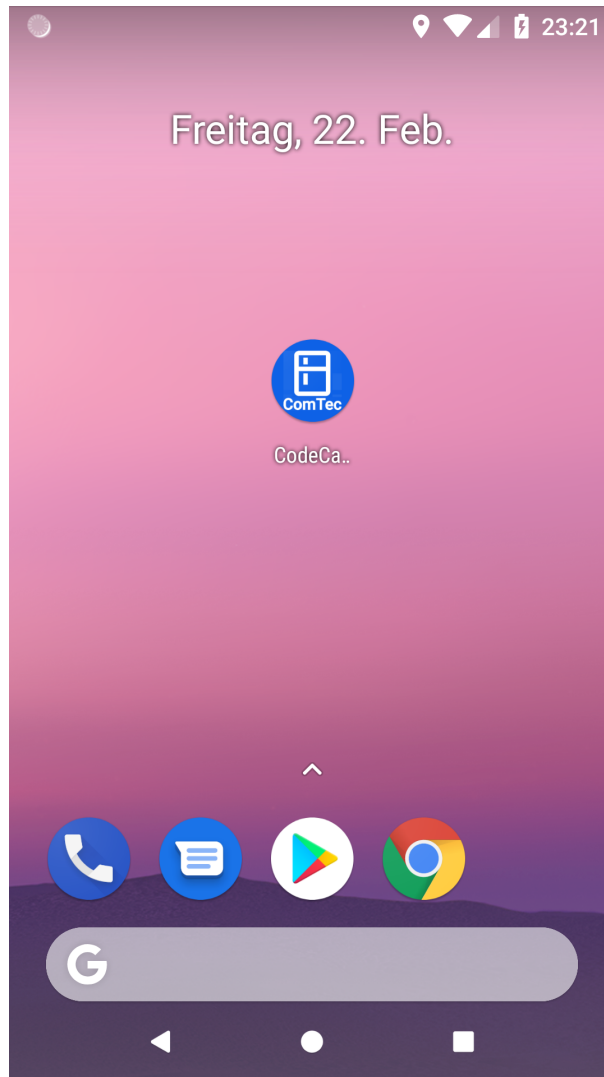


Abbildung 6: Liste getätigter Einkäufe

jeden Artikel erhalten. Des weiterem wird der Betrag von allen verkauften Artikeln, der Betrag von den selbst eingekauften Artikeln, die Anzahl aller verkauften Artikel und die Anzahl der selbst eingekauften Artikel dargestellt. Die Ansicht eines Administrators ist in Abbildung 7 zu sehen. Ein Benutzer erhält lediglich einen Überblick über seine eignen Einkäufe. Dabei kann der Benutzer sehen wie viele Exemplare eines Artikels dieser gekauft hat. Des weiterem wird dem Benutzer dargestellt wie viel Geld dieser bereits ausgegeben hat und wie viele Artikel dieser insgesamt schon gekauft hat. Die Ansicht eines Benutzers ist ebenfalls in Abbildung 7 dargestellt. Natürlich kann ein Administrator, wie der Benutzer, auch seine eignen Einkäufe einsehen. In den Statistiken werden alle Beträge in Euro angezeigt.

2.9 Einstellungen-Screen

Im Einstellungen-Screen hat der Benutzer die Möglichkeit, durch das Tippen eines Switch-Buttons den Dunkel-Modus der App zu aktivieren. Die App erscheint dann in einem dunkleren Design und ist dadurch schonender für die Augen. Bei erneutem Tippen des Switch-Buttons wird die App wieder in helleren Farben angezeigt. Ob der Dunkel-Modus aktiviert oder deaktiviert ist, wird in den Shared-Preferences gespeichert.

2.10 Aktualisieren von Server-Daten

Alle Screens, die Daten vom Server in einer Liste darstellen, können aktualisiert werden. Dazu gehören der Shop-Screen, der Nutzer-Screen, sowie der Einkaufs-Historie-Screen. Eine Aktualisierung erfolgt, indem

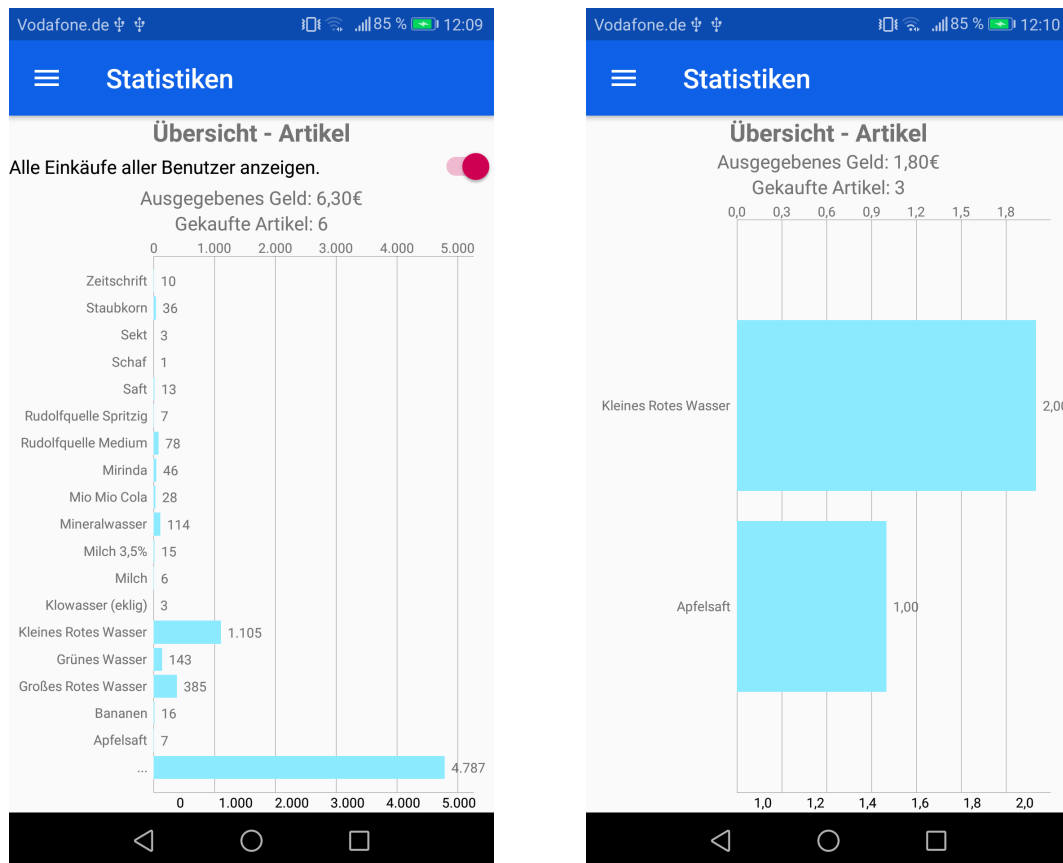


Abbildung 7: Statistiken - Ansicht des Administrators und Ansicht des Benutzers

am oberen Ende der Liste ein Zieh-Bewegung nach Unten durchgeführt wird. Dabei wird ein Lade-Balken sichtbar, der den Fortschritt der Aktualisierung anzeigt.

2.11 Unterstützte Sprachen

Von der App wird derzeit Englisch und Deutsch unterstützt. Die Wahl der Sprache hängt hierbei von der Systemsprache des Smartphones ab.

2.12 Shortcut-Funktion zum Scannen von Artikeln

3 Architektur

3.1 Datenmodell

Das Datenmodell umfasst drei Klassen: **Item**, **User**, **Purchase**. Die Klassen sind passend zur bereitgestellten Datenbank implementiert und haben die in Abbildung 8 gezeigte Struktur.

3.2 Frontend-Architektur

Die App nutzt eine Single Activity Architektur und läuft damit hauptsächlich in einer einzigen Activity, der **MainActivity**. Durch diese Architektur ergeben sich eine Vielzahl von Vorteilen.

3.2.1 Aufbau der MainActivity

Der Aufbau der **MainActivity** wird in Abbildung 9 dargestellt. Er besteht aus einem Frame-Layout und einer Toolbar. Zudem wird durch eine Wisch-Bewegung nach Rechts ein Navigation-Drawer angezeigt.

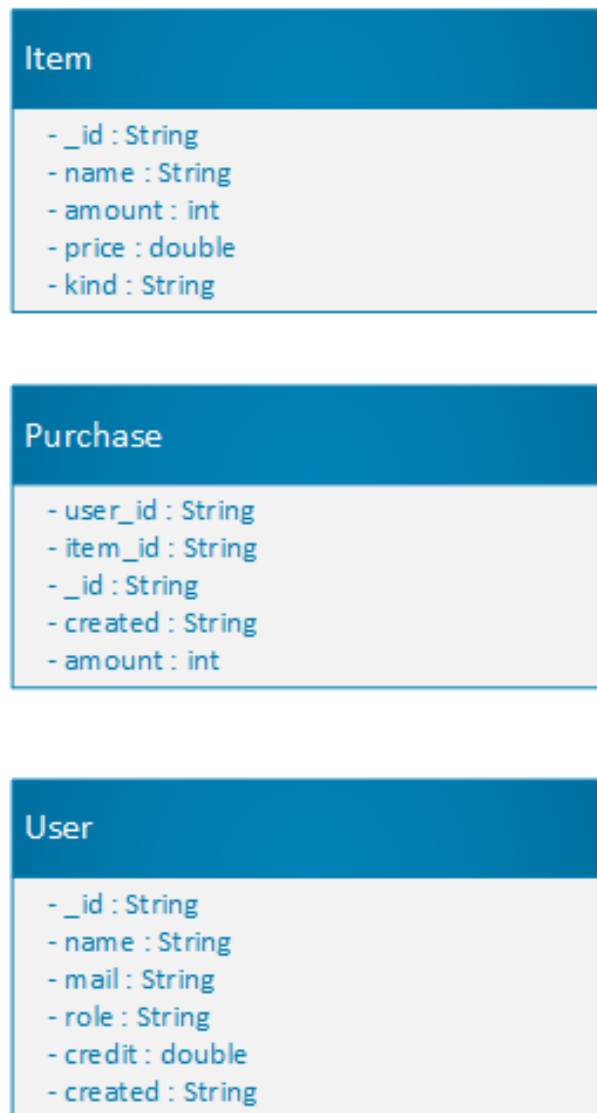


Abbildung 8: Datenmodell

3.2.2 Fragmente

Screens wurden durch Fragmente realisiert. Dabei handelt es sich um Container, die alle UI-Komponenten des jeweiligen Screens enthalten. Für jeden Screen, der in Abschnitt 2 beschrieben wurde, existiert ein solches. Befindet sich der Benutzer zum Beispiel im Shop-Screen, wird das Shop-Fragment im Frame-Layout angezeigt. Wechselt der Benutzer zu einem anderen Screen, wird das Fragment im Frame-Layout durch das Fragment des neuen Screens ersetzt. Dies erfolgt durch Zugriff auf den **FragmentManager** der **MainActivity**.

3.2.3 Toolbar

Im Gegensatz zu den Fragmenten, wird die Toolbar bei Screen-Wechseln nicht ausgetauscht, sondern kann vom neu angezeigten Fragment manipuliert werden.

Die Voraussetzung dafür ist, dass alle Fragments von **KitchenFragment** erben. Dieses stellt eine Methode bereit, die aufgerufen wird, sobald das Fragment in der **MainActivity** angezeigt wird. Beispielsweise können hier spezielle Symbole des Fragments in der Toolbar angezeigt werden.

Hier zeigt sich ein Vorteil der Single-Activity-Architecture. Wäre jeder Screen eine eigene Activity, müsste jedes Layout eine eigene Toolbar beinhalten. Die daraus resultierende Redundanz kann hier vermieden werden, da nur eine Toolbar im Layout der **MainActivity** benötigt wird.

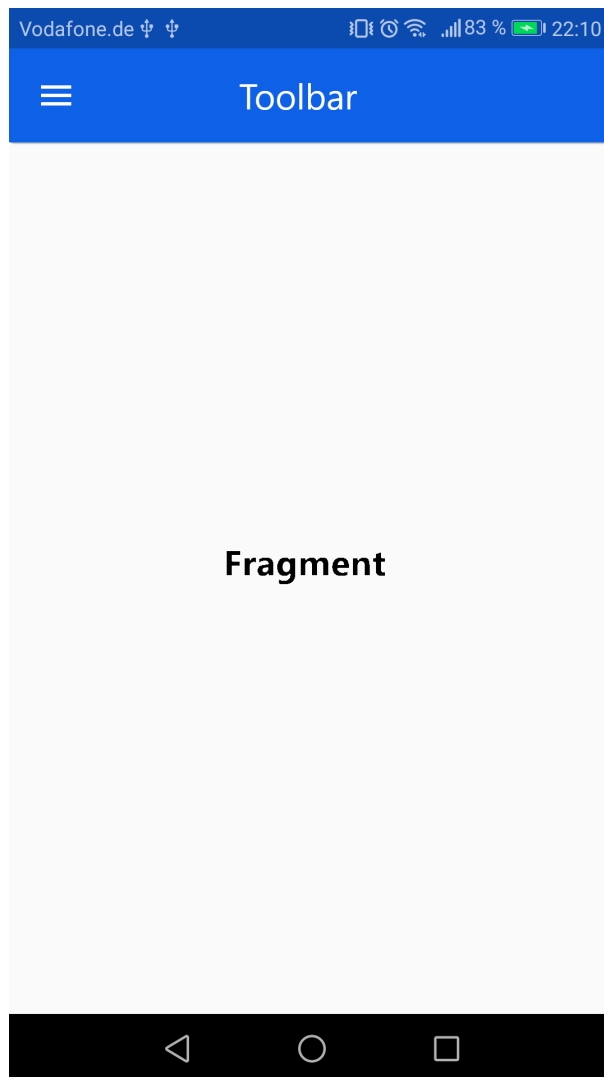


Abbildung 9: Aufbau der MainActivity

3.2.4 Navigation-Drawer

Durch die Single-Activity-Architektur wird das Benutzen eines Navigation-Drawers erst möglich, da einzig im Layout der MainActivity ein Navigation-Drawer eingebunden werden muss. Dies führt dazu, dass das Tippen eines Elements im Navigation-Drawer zentral verarbeitet werden kann. Dadurch kann beispielsweise das Zurückgleiten des Navigation-Drawers nach einem Tipp-Event, sehr einfach erreicht werden.

Im Gegensatz dazu müsste bei der Verwendung vieler Activities, jede Activity ein Navigation-Drawer in deren Layout definieren. Dies wäre nicht nur redundant, sondern würde auch die Verarbeitung von Tipp-Events und die korrekte Anzeige des Navigation-Drawers weitaus komplexer machen, da mit anderen Activities kommuniziert werden müsste.

3.2.5 Ausnahmen

Weil es sich angeboten hat und man sich idealerweise nur einmal anmeldet, ist der Login Screen eine eigene Activity. Auch der BarcodeScanner ist eine eigene Activity, da die BarcodeScanner Library dies verlangt.

3.2.6 Kommunikation mit dem Backend-Layer

Die Architektur ermöglicht sehr einfache Kommunikation mit dem Backend-Layer. Die MainActivity enthält eine Instanz des `KitchenManagers`, über den alle UI Elemente, einschließlich der Fragmente, mit

dem Backend kommunizieren können. Der genaue Aufbau des **KitchenManagers** wird in Abschnitt 3.3.3 beschrieben.

3.2.7 RecyclerViews

3.3 Backend-Architektur

Die Backend-Architektur besteht aus drei Ebenen (siehe Abbildung 10).

3.3.1 Base-Layer

Die unterste Ebene bildet die Klasse **OkHttpClientConnection**, welche über das Interface **HttpClientConnection** implementiert ist. Die Klasse ist verantwortlich für den primitiven Netzwerkverkehr per HTTP mit beliebigen Inhalten. Es werden die HTTP-Methoden **GET**, **POST**, **PUT** und **DELETE** implementiert. Bei fehlerhaften Anfragen (HTTP-Code $\neq 2xx$), sowie bei fehlerhaften URLs, wird eine Exception geworfen.

3.3.2 Abstraction-Layer

Auf der zweiten Ebene befinden sich die Klassen **KitchenConnection**, **LocalDataStore** und **JsonTranslator**.

- **KitchenConnection** ist verantwortlich für die spezialisierte Kommunikation mit der bereitgestellten REST-API. Die Klasse enthält Attribute für die Basis-URL des REST-Servers, sowie für Schlüssel zur Erzeugung von Nutzern. Alle Funktionen der REST-API werden hier implementiert, indem jede Methode der Klasse einen Pfad der REST-API abdeckt. Die Funktionsnamen entsprechen der Spezifikation der REST-API.
- **LocalDataStore** ist verantwortlich für das Zwischenspeichern der User, Items und Purchases. Dies ermöglicht schnelleren Zugriff auf die jeweiligen Objekte.
- **JsonTranslator** ist verantwortlich für die bidirektionale Umwandlung zwischen Objekten des Datenmodells und JSON-Strings. Die verwendete Bibliothek für die Umwandlung ist Google's GSON.

3.3.3 Access-Layer

Die Dritte Ebene beinhaltet nur die Klasse **KitchenManager**. Sie ist der Einstiegspunkt des Backend und stellt so alle Operationen zur Verfügung, welche die Oberfläche benötigt. **KitchenManager** bildet eine vollständige Abstraktion der REST-API, über die Klasse **KitchenConnection**, welche bereits die Adressen und benötigten Header der API-Aufrufe abstrahierte.

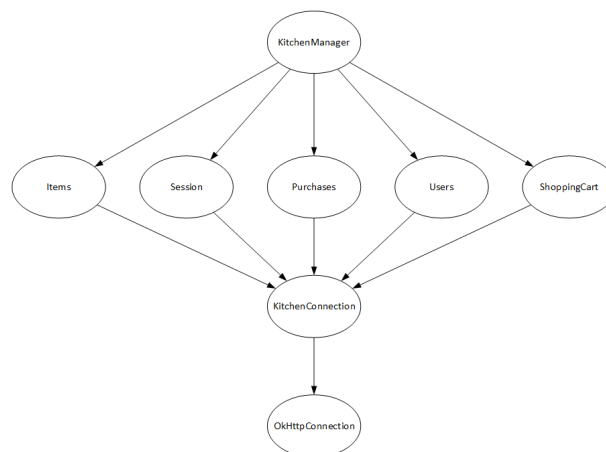


Abbildung 10: Backend Architektur

3.3.4 Tests

Alle Backend-Klassen sind durch Unit-, Integrations- und Akzeptanz-Tests ausreichend getestet.

4 Anhang

4.1 Bibliotheken