



## **YOU OWN ME MONEY**

App zur Verwaltung gemeinsamer Ausgaben

Ein Projekt im Rahmen der Lehrveranstaltung  
Spezielle Programmierung des Studiengangs  
Wirtschaftsinformatik der HTW Berlin

**Carolin Gellner**

Matrikel-Nr.: 539540

Version 1.0 | 30.07.2016

## Änderungsübersicht

| Version | Datum      | Beschreibung                  |
|---------|------------|-------------------------------|
| 0.1     | 28.07.2016 | Initialisierung des Dokuments |
| 0.2     | 29.07.2016 | In Bearbeitung                |
| 1.0     | 30.07.2016 | Fertigstellung                |

## Inhaltsverzeichnis

|  |           |
|--|-----------|
| <b>1. EINFÜHRUNG.....</b>  | <b>3</b>  |
| 1.1. AUFGABENSTELLUNG .....  | 3         |
| 1.2. IDEE ZUR UMSETZUNG DES PROJEKTS .....                                 | 3         |
| 1.3. ZIELGRUPPE .....  | 3         |
| <b>2. RANDBEDINGUNGEN .....</b>  | <b>3</b>  |
| <b>3. FUNKTIONEN .....</b>   | <b>4</b>  |
| <b>4. TECHNISCHER KONTEXT .....</b>  | <b>5</b>  |
| 4.1. PAKET-/KLASSENDIAGRAMM.....   | 5         |
| 4.2. BESCHREIBUNGEN DER PAKETE & KLASSEN.....                              | 5         |
| 4.2.1. START .....   | 5         |
| 4.2.2. DATABASE .....  | 6         |
| 4.2.3. OBJECTS .....   | 6         |
| 4.2.4. OVERVIEWPAYMENTS .....  | 6         |
| 4.2.5. SETTINGS .....  | 7         |
| 4.2.6. VIEWPAGER.....  | 7         |
| <b>5. ANWENDERDOKUMENTATION.....</b>                                       | <b>8</b>  |
| 5.1. START & MENÜ .....  | 8         |
| 5.2. ERFASSEN EINER AUSGABE.....   | 9         |
| 5.3. ERFASSEN EINER RÜCKZAHLUNG .....                                      | 12        |
| 5.4. ÜBERSICHT DER ZAHLUNGSDIFFERENZEN .....                               | 14        |
| 5.5. LISTE AUSGABEN UND RÜCKZAHLUNGEN & DETAILANSICHT (MASTER-DETAIL)..... | 15        |
| <b>6. DATENBANK .....</b>  | <b>20</b> |
| 6.1. AUFBAU DER DATENBANK .....  | 20        |
| 6.2. DATABASE-KLASSE .....   | 20        |

## 1. Einführung

### 1.1. Aufgabenstellung

Im Rahmen der Lehrveranstaltung Spezielle Programmierung des Studiengangs Wirtschaftsinformatik der Hochschule für Technik und Wirtschaft ist die Konzeptionierung und Programmierung einer Android-App vorgesehen.

### 1.2. Idee zur Umsetzung des Projekts

Die Idee zur Umsetzung des Projekts ist die Konzeptionierung und Programmierung einer App, die zur Organisation von gemeinsamen Ausgaben zwischen verschiedenen Personen dient. Die App bietet die Möglichkeit Ausgaben zu erfassen und ermittelt daraufhin die Differenzen zwischen den einzelnen Personen. Etwaige entstandene Schulden können dann durch die Erfassung einer Rückzahlung ausgeglichen werden. Die Anwendung bietet des Weiteren eine Übersicht pro angelegter Person, aus der sich erkennen lässt, wem die betreffende Person in welcher Höhe einer anderen Person Geld schuldet oder ob diese Person noch Geld (z.B. durch eine Rückzahlung) von einer anderen erwartet.

*Im Folgenden ist ein Anwendungsbeispiel für die Applikation beschrieben:*

*Die drei Freunde Anna, Maria und Julia fahren gemeinsam in den Urlaub. Dort tätigen sie u.a. gemeinsame Lebensmitteleinkäufe. Diese zahlt besonders häufig Julia, manchmal auch Maria, aber Anna bezahlt meistens nicht. Normalerweise würden sie am Ende des Urlaubs überlegen, wer was bezahlt hat und dann hin und her rechnen, wer welcher Person noch etwas zurückzahlen muss. Durch die App können Sie die getätigten Ausgaben direkt erfassen und haben sofort eine Übersicht über ihre Ausgaben und auch ein ewiges Hin- und Her-Rechnen fällt weg.*

### 1.3. Zielgruppe

Die App richtet sich an Personen, die sich oft Ausgaben mit anderen teilen und gerne einen genauen Überblick diesbezüglich haben möchten.

## 2. Randbedingungen

| Systemvorgaben des Endgeräts |  |
|------------------------------|--|
| Betriebssystem               | Android 5.0                                  |
| Programmiervorgaben          |  |
| Sprache                      | Java   |
| Software                     | Android Studio 2.1 inkl. Sdk, Java Version 8 |
| Datenbank                    | SQLite                                       |
| Gestaltung                   | mind. 1 „Master-Detail“-Layout               |
| Funktionen                   | Speichern und auslesen von Daten             |

### 3. Funktionen

Das folgende Anwendungsfalldiagramm stellt die Funktionen dar, welche dem Benutzer in der App zur Verfügung stehen. Die grün hinterlegten Funktionen, stellen die Hauptfunktionen dar, die weiß hinterlegten Nebenfunktionen.

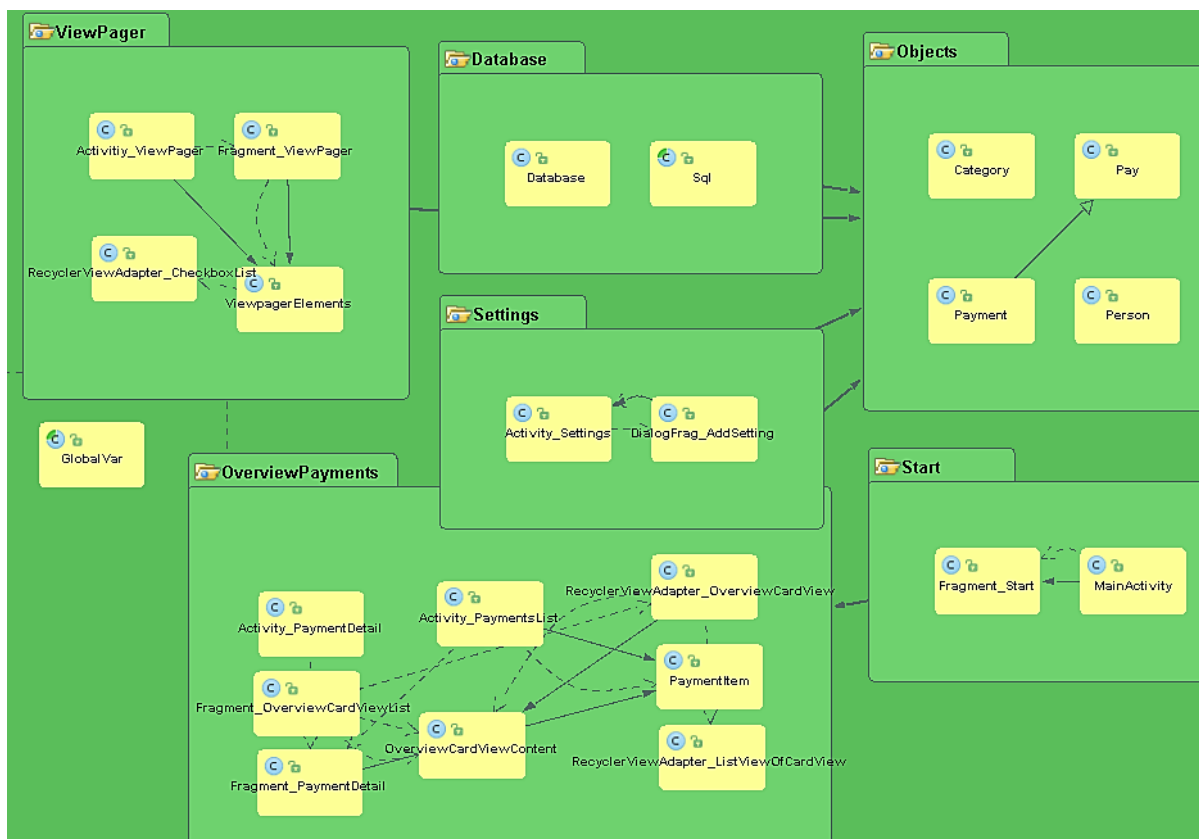


## 4. Technischer Kontext

In diesem Kapitel wird der Aufbau des Programms erläutert. Eine detaillierte [Quelcodedokumentation](#) im JavaDoc-Stil ist im Ordner der Anwendung hinterlegt unter dem folgenden Pfad: [YOMM/app/Dokumentation/index.html](#).

### 4.1. Paket-/Klassendiagramm

Die folgende Darstellung zeigt eine Übersicht der wichtigsten Pakete und Klassen des Programms. Die Verbindungsstriche zeigen die Beziehungen zwischen den Paketen und Klassen auf.



### 4.2. Beschreibungen der Pakete & Klassen

Die folgenden Bilder enthalten kurze Beschreibungen der einzelnen Klassen in den Paketen. Die Darstellungen wurden mithilfe der integrierten JavaDoc-Funktion des Android-Studios generiert.

#### 4.2.1. Start

Das Paket „Start“ beinhaltet alle Klassen, welche die Startseite der Applikation beinhalten. Dazu zählen die Haupt-Activity „MainActivity“, welche auch ein Menü beinhaltet, und das Fragment „Fragment\_Start“, welches die Startansicht beinhaltet.

## Package com.cgellner.yomm.Start

| Class Summary  |  |
|----------------|--|
| Class          | Description  |
| Fragment_Start | Die Klasse beinhaltet das Fragment, welches die Startseite der App darstellt.                |
| MainActivity   | Die Klasse stellt die Hauptactivity der App dar und beinhaltet die Startseite und ein Menue. |

### 4.2.2. Database

Das Paket „Database“ beinhaltet zwei Klassen, welche die Datenbank betreffen. Dabei repräsentiert die Klasse „Database“ die **SQLite-Datenbank** mit verschiedensten Funktionen zum Speichern und Auslesen der Daten. Die Klasse „Sql“ stellt lediglich eine Sammlung der Tabellen- und Spaltennamen der Datenbank dar sowie die Befehle zum Initialisieren der Datenbank (Create-Befehle).

Die Datenbank wird **innerhalb des Programms in der Klasse „GlobalVar“ gespeichert** und über diese Klasse auf die Datenbank jeweils zugegriffen.

## Package com.cgellner.yomm.Database

| Class Summary |   |
|---------------|---|
| Class         | Description   |
| Database      | Die Klasse beinhaltet die Datenbank und bietet diverse Methoden zum Ausführen von Abfragen an die Datenbank.      |
| Sql           | Die Klasse beinhaltet alle Tabellen- und Spaltennamen der Datenbank sowie Sql-Befehle zum Erstellen der Tabellen. |

### 4.2.3. Objects

Das Paket „Objects“ repräsentiert die verwendeten Objekte oder auch Datensätze im Programm. Die Klasse „Pay“ stellt eine Rückzahlung dar, die durch die Attribute der Klasse „Payment“ erweitert werden kann und dann eine Ausgabe darstellt.

## Package com.cgellner.yomm.Objects

| Class Summary |   |
|---------------|---|
| Class         | Description   |
| Category      | Die Klasse repräsentiert eine Kategorie.  |
| Pay           | Die Klasse Pay stellt die Oberklasse für ein Zahlungsobjekt dar, welche gleichzeitig auch eine Rückzahlung darstellen kann. |
| Payment       | Die Klasse stellt eine Zahlung in Form einer Ausgabe dar und enthält die Oberklasse Pay.                                    |
| Person        | Die Klasse repräsentiert eine Person.   |

### 4.2.4. OverviewPayments

Das Paket „OverviewPayments“ enthält alle Klassen, welche die Möglichkeit bieten, die erfassten Ausgaben und Rückzahlungen einzusehen sowie eine Übersicht der Zahlungsdifferenzen.

Die Funktion **„Schuldenübersicht einsehen“** wird durch die folgenden Klassen realisiert:

Fragment\_OverviewCardListView (wird in der MainActivity angezeigt)

RecyclerViewAdapter\_ListViewOfCardView (inkl. ViewHolder-Klasse)

RecyclerViewAdapter\_OverviewCardViews (inkl. ViewHolder-Klasse)

OverviewCardViewContent

Die Funktion „Ausgaben und Rückzahlungen einsehen“ wird in Form des sogenannten „Master-Detail-Layouts“ dargestellt, welches durch die folgenden Klassen realisiert wird:

Activity\_PaymentsList

Activity\_PaymentDetail

Fragment\_PaymentDetail

Im Rahmen der Detailansicht einer Zahlung wurde die Funktion „Alte Ausgaben und Rückzahlungen löschen“ umgesetzt.

Package com.cgellner.yomm.OverviewPayments

| Class Summary   |   |
|---|---|
| Class   | Description   |
| Activity_PaymentDetail  | Die Klasse repräsentiert eine Activity in welcher das Detail einer (Rueckzahlung) Repayment oder Ausgabe (Payment) im Rahmen des Master-Detail-Layouts angezeigt wird.  |
| Activity_PaymentsList   | Die Klasse repräsentiert eine Activity, welche Teil den Master-Teil des sogenannten Master-Details-Layouts beinhaltet und in diesem Fall eine Liste von Payment- und Repayment-Datensätzen beinhaltet.              |
| Fragment_OverviewCardViewList   | Die Klasse repräsentiert die Uebersichtsansicht, welche eine Liste mit Karten (CardViews) darstellt, die pro angelegter Person dargestellt werden und eine Auflistung der Zahlungsdifferenzen je Person beinhaltet. |
| Fragment_PaymentDetail  | Die Klasse repräsentiert die Detailansicht einer Ausgabe oder Rueckzahlung.   |
| OverviewCardViewContent   | Die Klasse repräsentiert ein Objekt, welches die Daten fuer eine CardView in der Uebersichtsansicht beinhaltet.   |
| PaymentItem   | Die Klasse repräsentiert ein Objekt, welches die Daten fuer die Items des Master-Detail-Layouts beinhaltet.   |
| RecyclerViewAdapter_ListViewOfCardView                                  | Die Klasse repräsentiert den RecyclerViewAdapter fuer die Liste mit den CardViews fuer die Uebersichtsliste.  |
| RecyclerViewAdapter_ListViewOfCardView.ViewHolder_Cardview_RecyclerView | Die Klasse repräsentiert eine CardView in der Uebersichtsliste.   |
| RecyclerViewAdapter_OverviewCardView                                    | Die Klasse repräsentiert den Adapter der Liste innerhalb einer CardView, welche alle Zahlungsdifferenzen pro Person beinhaltet.   |
| RecyclerViewAdapter_OverviewCardView.ViewHolder_OverviewCards           | Die Klasse repräsentiert ein Item in der RecyclerView-Liste innerhalb einer CardView.   |

#### 4.2.5. Settings

Das Paket „Settings“ enthält die Klassen zum Verwalten der Personen und Kategorien. Durch die beiden Klassen werden alle Funktionen zu den Personen und Kategorien realisiert. Diese beinhalten das „Erfassen, bearbeiten und löschen von Personen“ und das „Erfassen, bearbeiten und löschen von Kategorien“.

Package com.cgellner.yomm.Settings

| Class Summary         |   |
|-----------------------|---|
| Class                 | Description   |
| Activity_Settings     | Die Klasse stellt eine Activity dar, in welcher Elemente wie Personen oder Kategorien verwaltet werden koennen.             |
| DialogFrag_AddSetting | Die Klasse stellt eine Ansicht dar, in welcher Elemente wie Personen oder Kategorien angezeigt und verwaltet werden koennen |

#### 4.2.6. ViewPager

Das Paket „ViewPager“ enthält alle Klassen, die das Erfassen von Ausgaben und Rückzahlungen realisieren und somit die Funktionen „Ausgabe erfassen“ und „Rückzahlung erfassen“ bieten.

Package com.cgellner.yomm.ViewPager

| Class Summary  |  |
|--|--|
| Class  | Description  |
| Activitiy_ViewPager                                      | Die Klasse beinhaltet die Activity in welcher der Viewpager angezeigt wird, welcher die Formulare zum Erfassen einer Ausgabe oder Rueckzahlung beinhaltet. |
| Fragment_ViewPager                                       | Die Klasse stellt ein Fragment (eine Seite/Ansicht) innerhalb eines ViewPagers dar.  |
| RecyclerViewAdapter_CheckboxList                         | Die Klasse beinhaltet den Adapter der RecyclerView fuer eine Liste mit Checkboxes Diese Liste wird in Formular zur Erfassung einer Ausgab verwendet.       |
| RecyclerViewAdapter_CheckboxList.ViewHolder_CheckboxList | Die Klasse beinhaltet den ViewHolder der RecyclerView.   |
| ViewPagerElements  | Die Klasse beinhaltet die Elemente der ViewPager-Seiten, welche die Formulare zum Erfassen einer Ausgabe oder Rueckzahlung repraesentieren.                |



## 5. Anwenderdokumentation

Die folgenden Abbildungen zeigen die aktuellen Ansichten zu den implementierten Funktionen.

Achtung, wenn die App zum ersten Mal gestartet wird, müssen zunächst mindestens zwei Personen (siehe 5.6.) und mindestens eine Kategorie angelegt werden (siehe 5.7.), bevor eine Ausgabe erfasst werden kann.

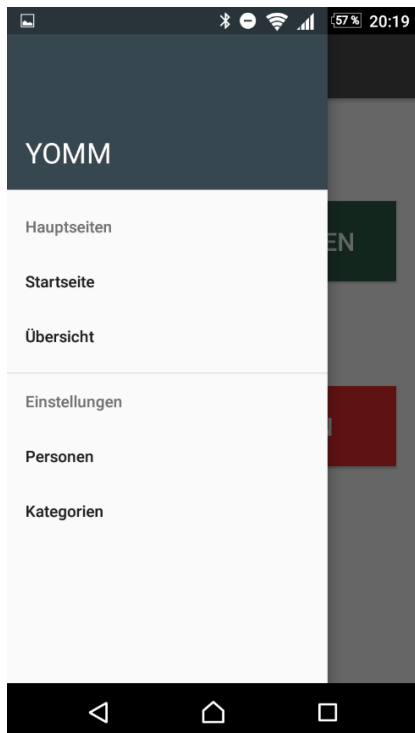
### 5.1. Start & Menü

Diese Sicht stellt den Startbildschirm der App dar. Derzeit sind dieser Sicht noch keine Informationen zu entnehmen



Die Seite stellt die **Startseite** der App dar und beinhaltet zwei Buttons, welche die Möglichkeit bieten, Formulare zum Erfassen von Ausgaben und Rückzahlungen zu öffnen.

Des Weiteren lässt sich über diese Ansicht ein Menü öffnen.



Das **Menü** bietet die folgenden Punkte:

#### **Startseite**

Startseite mit den Buttons zum Öffnen von Formularen zur Erfassung von Ausgaben und Rückzahlungen

#### **Übersicht**

Bietet eine Zusammenfassung der aktuellen Ausgabendifferenzen zwischen den jeweiligen Personen.

#### **Personen**

Hier können Personen verwaltet werden. Diese können angelegt, umbenannt oder gelöscht werden.

#### **Kategorien**

Hier können Kategorien verwaltet werden. Diese können angelegt, umbenannt oder gelöscht werden.

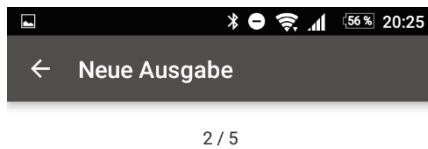
## 5.2. Erfassen einer Ausgabe

Diese folgenden Sichten stellen ein Formular dar, welches aus mehreren Teilen besteht und womit sich eine Ausgabe erfassen lässt.

Die erste Seite des Formulars dient zur **Eingabe des Gesamtbetrags**.

Zum Wechsel einer Seite muss lediglich nach rechts (oder links) mit dem Finger gewischt werden.

Zu diesem Zeitpunkt ist der Speichern-Button noch grau hinterlegt, da eine Speicherung noch gesperrt ist, da die Daten noch unvollständig sind.



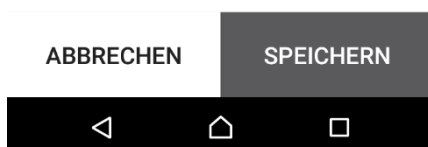
**Auf welche Personen wird  
die Ausgabe aufgeteilt?**

- ☐ Carolin
- ☐ Lisa



**Wer hat bezahlt?**

- ☐ Carolin
- ☐ Lisa



Die zweite Seite des Formulars dient der **Auswahl aller Personen auf die die Ausgabe aufgeteilt werden soll** (Debitoren).

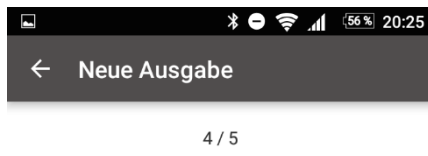
ACHTUNG:

Sollten hier keine Personen stehen, müssen noch welche angelegt werden.

Zu diesem Zeitpunkt ist der Speichern-Button auch noch grau hinterlegt, da die Daten immer noch unvollständig sind, um sie speichern zu können.

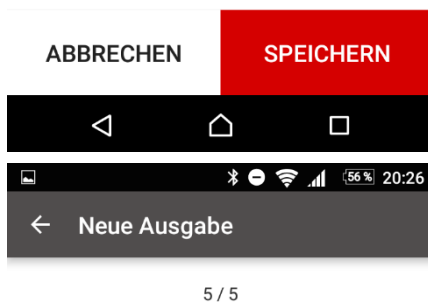
Die dritte Seite des Formulars dient zur **Auswahl der Person, welche die Ausgabe bezahlt hat** (Kreditor).

Zu diesem Zeitpunkt ist der Speichern-Button auch noch grau hinterlegt, da die Daten immer noch unvollständig sind, um sie speichern zu können.



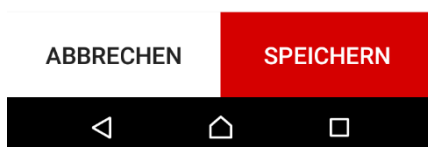
### Kategorie

- ☒ Lebensmittel  
☐ Kosmetik



### Bemerkungen

Einkauf bei Netto



Die vierte Seite des Formulars dient zur **Auswahl der Kategorie**, in welche die Ausgabe zuzuordnen ist.

#### ACHTUNG:

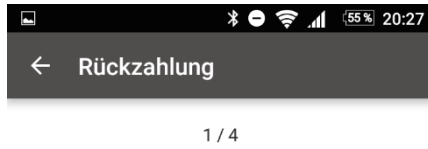
Sollten hier keine Kategorien stehen, müssen noch welche angelegt werden. Eine Speicherung ist ohne eine Kategorie nicht möglich.

Sobald auf dieser Seite eine Kategorie ausgewählt wurde (und auch alle Daten auf den Seiten davor gewählt wurden) leuchtet der Speicher-Button in roter Farbe auf und signalisiert somit, dass nun die Speicherung der Ausgabe möglich ist.

Die letzte Seite des Formulars dient zur **Erfassung einer Bemerkung** zu der Ausgabe. Diese Angabe ist optional und braucht nur bei Bedarf ausgefüllt werden.

### 5.3. Erfassen einer Rückzahlung

Diese folgenden Sichten stellen ein Formular dar, welches aus mehreren Teilen besteht und womit sich eine Rückzahlung erfassen lässt, wodurch die Begleichung von Schulden möglich ist.



**Folgende Personen haben Schulden. Bitte wähle eine Person:**

☒ Lisa

Die erste Seite des Formulars dient zur **Auswahl der Person, für die eine Rückzahlung erfasst werden soll.**

In dieser Ansicht werden nur Personen aufgelistet, die auch tatsächlich Schulden zu einer oder mehreren Personen haben. Sollte die Ansicht also leer sein sind die Zahlungen zwischen allen Personen ausgeglichen und niemand hat Schulden. (Oder es wurden einfach noch keine Personen angelegt)

Zu diesem Zeitpunkt ist der Speichern-Button auch noch grau hinterlegt, da die Daten immer noch unvollständig sind, um sie speichern zu können.

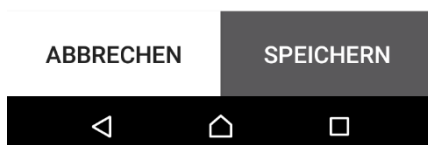


**Die gewählte Person hat zu folgenden Personen Schulden. Bitte wähle eine Person:**

☒ Carolin

Die zweite Seite des Formulars werden alle Personen aufgelistet, zu welchen die vorher gewählte Person von der ersten Seite aktuell Schulden hat. Dort soll dann eine **Person ausgewählt werden, zu welcher die betreffende Person ihre Schulden ausgleichen möchte.**

Zu diesem Zeitpunkt ist der Speichern-Button auch noch grau hinterlegt, da die Daten immer noch unvollständig sind, um sie speichern zu können.



← Rückzahlung

3 / 4

**Bemerkungen**

Online überwiesen

Die dritte Seite des Formulars dient zur **Erfassung einer Bemerkung** zu der Rückzahlung. Diese Angabe ist optional und braucht nur bei Bedarf ausgefüllt werden.

Sobald diese Seite angezeigt wird und vorher die Personen ausgewählt wurden erscheint der Speichern-Button in roter Farbe und signalisiert, dass nun die Speicherung der Rückzahlung möglich ist. Es empfiehlt sich allerdings noch die letzte Seite anzuschauen, da dort der Betrag angezeigt wird, welcher zurückgezahlt werden muss.

ABBRECHEN SPEICHERN

← Rückzahlung

4 / 4

**Gesamtbetrag**

Die letzte Seite des Formulars dient zum **Anzeigen des Betrags der Rückzahlung** und repräsentiert den Gesamtbetrag der Schulden der angegebenen ersten Person zur angegebenen zweiten Person.

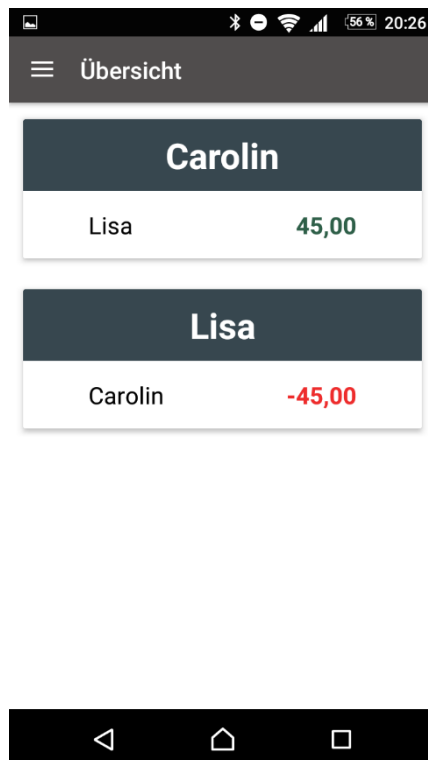
**45,00 Euro**

Bitte begleiche den Betrag und drücke auf Speichern.

ABBRECHEN SPEICHERN

## 5.4. Übersicht der Zahlungsdifferenzen

Diese folgende Sicht stellt ein Kernstück der App dar und beinhaltet eine Übersicht, welche den Benutzer darüber informiert, wie die jeweiligen Zahlungsdifferenzen der Personen zueinander sind.



Diese Ansicht zeigt die **Übersicht aller Zahlungsdifferenzen** auf. Pro angelegter Person wird vom Programm eine kleine Übersicht angelegt. Ein grüner Wert bedeutet, dass die Hauptperson der Übersicht noch Geld von der jeweiligen Person erhält und rot bedeutet, dass die Hauptperson der Übersicht noch Schulden bei der jeweiligen Person hat.

*In diesem Fall bekommt Carolin noch 45,00 Euro von Lisa und Lisa hat 45,00 Euro Schulden bei Carolin.*

Durch einen Klick auf den Namen einer Person gelangt man zur Liste in der alle Ausgaben und Rückzahlungen der betreffenden Person(en) angezeigt werden (siehe 5.5.)

## 5.5. Liste Ausgaben und Rückzahlungen & Detailansicht (Master-Detail)

Diese folgende Sicht stellt eine Liste dar, welche durch einen Klick auf eine Übersichtskarte geöffnet wird und alle Ausgaben (schwarz) und Rückzahlungen (grün) der betreffenden Person(en) auflistet.

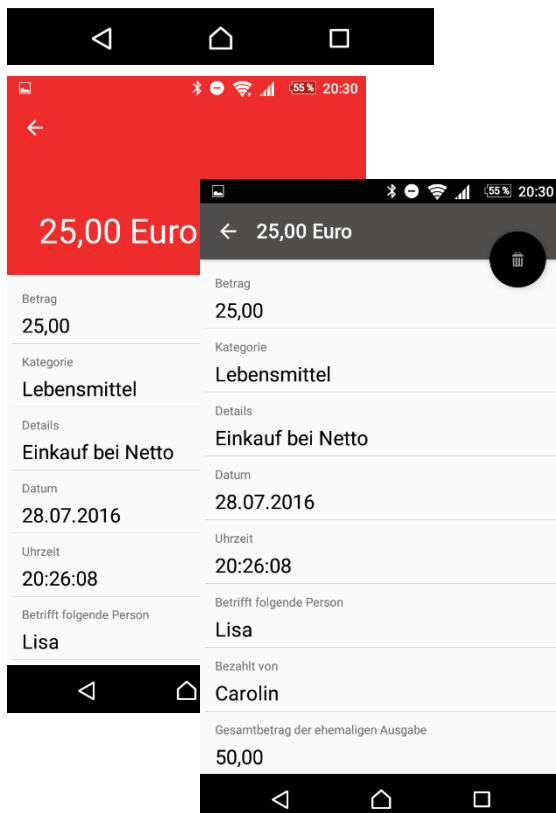


|        |       |
|--------|-------|
| 28.07. | 45,00 |
| 28.07. | 20,00 |
| 28.07. | 25,00 |

Die Ansicht zeigt **alle Ausgaben und Rückzahlung** an, welche zur betreffenden Person erfasst wurden. Steht in der Titelleiste nur ein Name, so werden alle Datensätze angezeigt, welche einen Bezug zu dieser einen Person haben. Werden zwei Namen oben angezeigt, so wie im Bild, werden alle Datensätze angezeigt, die genau diese beiden Personen zusammen betreffen.

Eine **Ausgabe** hat die Farbe **Schwarz** und eine **Rückzahlung** die Farbe **Grün**.

Des Weiteren lassen sich die Listenelemente nach ihrem Typ (Ausgabe oder Rückzahlung) filtern.

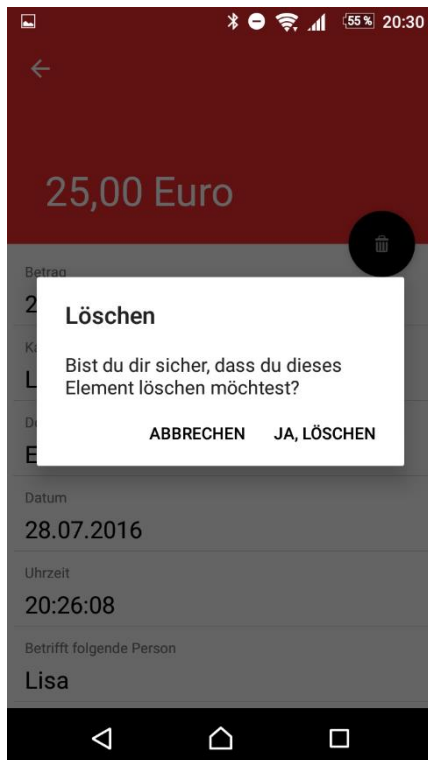


In der Detailansicht sind alle erfassten Dateien pro Ausgabe oder Rückzahlung erfasst.

Eine **Ausgabe** hat die Farbe **Rot** und eine **Rückzahlung** die Farbe **Grün**.

Über den schwarzen Button kann die jeweilige Ausgabe oder Rückzahlung gelöscht werden.





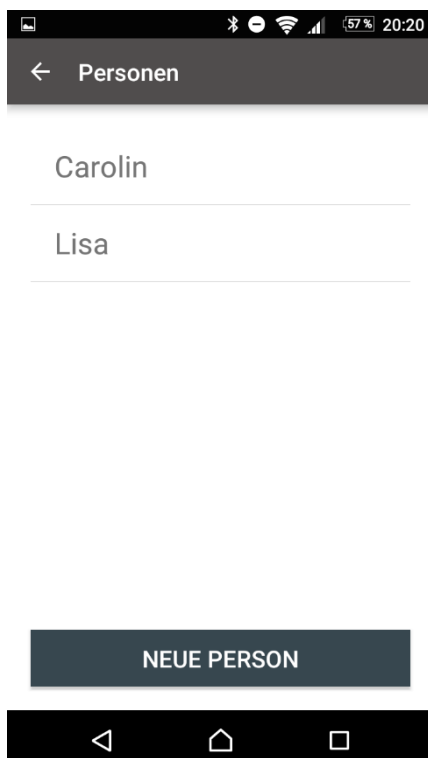
Die Detailansicht eines Elements bietet die Funktion die entsprechende **Ausgabe oder Rückzahlung zu löschen**. Dazu muss auf den schwarzen runden Button mit dem Papierkorb geklickt werden und die folgende Nachricht erscheint. Nach einem Klick auf „Ja, löschen“ wird das Element gelöscht.

Achtung:

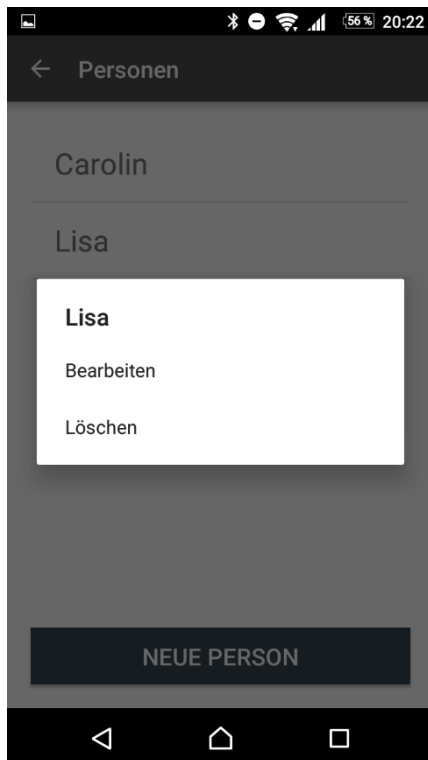
Um eine aktualisierte Übersicht der Zahlungsdifferenzen und der Liste mit allen Ausgaben und Rückzahlung zu haben, ist es erforderlich alle diese Ansichten erst zu schließen. Am besten durch einen Klick im Menü auf „Startseite“. Wenn Sie dann im Menü wieder auf „Übersicht“ gehen, ist die Ansicht aktualisiert und das gelöschte Element erscheint nirgends mehr.

## 5.6. Verwalten der Personen

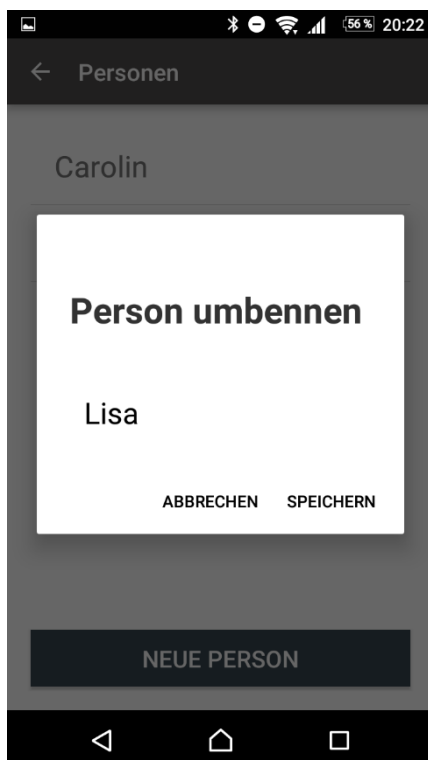
Diese folgende Sicht stellt die Verwaltungsmöglichkeit der Personen dar. Dort werden alle Personen aufgelistet. Es können neue Personen erfasst werden. Diese können dann bearbeitet oder auch gelöscht werden.



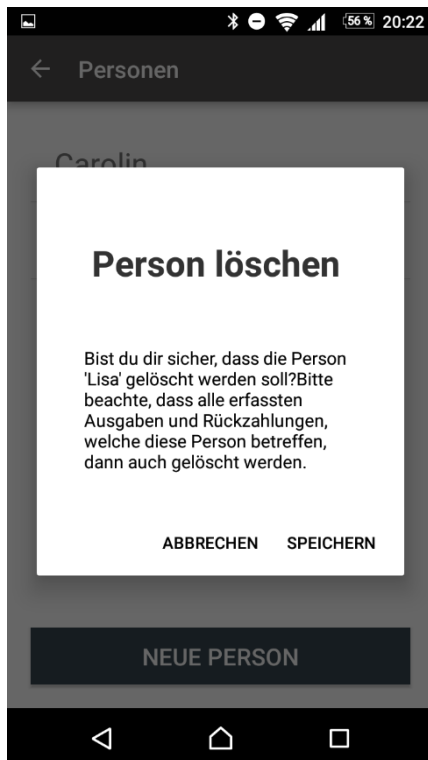
Die Seite stellt eine **Auflistung aller Personen** dar. Über den Button „Neue Person“ kann eine neue Person erfasst werden.



Durch langes Drücken auf eine Person erscheint ein kleines **Menü zum Bearbeiten und Löschen eines Elements**.



Wählt man „Bearbeiten“ so öffnet sich ein Fenster zum **Ändern des Namens der Person**.

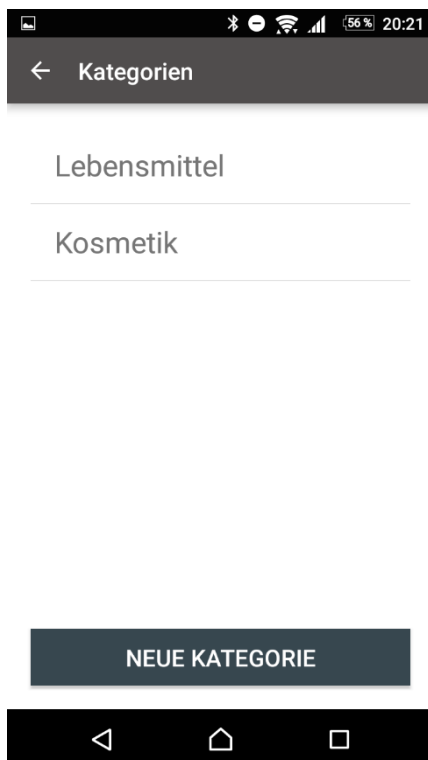


Wählt man **Löschen** so erfolgt zunächst eine Prüfung, ob die jeweilige **Person** keine offenen Zahlungsdifferenzen hat. Erst wenn alle Zahlungsdifferenzen der Person ausgeglichen sind und in der Übersichtskarte der Person alle Werte auf 0 Euro stehen, kann die Person gelöscht werden. In diesem Fall wird ein entsprechender Informationstext angezeigt.

Für den Fall, dass die Person gelöscht werden darf, erscheint der Text, der auf dem Bild zu sehen ist.

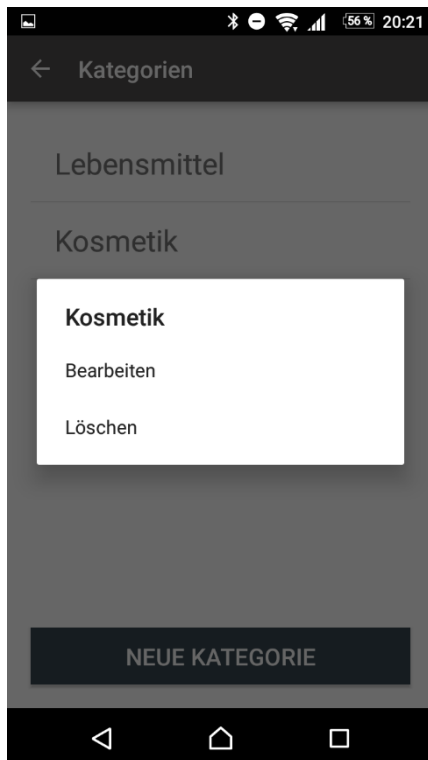
## 5.7. Verwalten der Kategorien

Diese folgende Sicht stellt die Verwaltungsmöglichkeit der Kategorien dar. Dort werden alle Kategorien aufgelistet. Es können neue Kategorien erfasst werden. Diese können dann bearbeitet oder auch gelöscht werden.

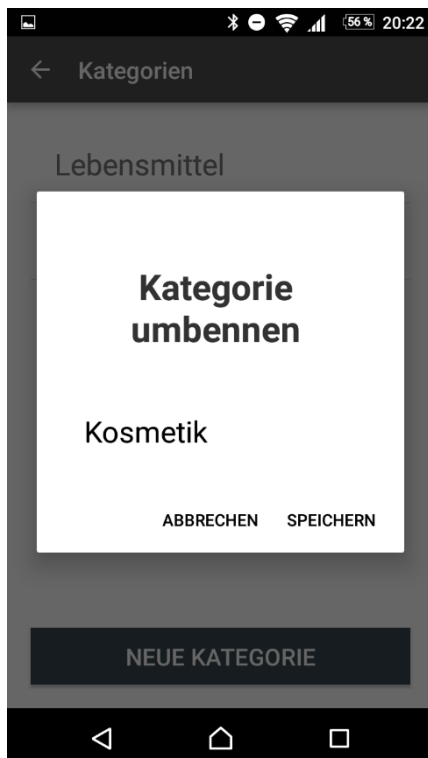


Die Seite stellt eine **Auflistung aller Kategorien** dar. Über den Button „Neue Kategorie“ kann eine neue Kategorie erfasst werden.

Es muss immer mindestens eine Kategorie erfasst sein, da ansonsten keine Ausgaben erfasst werden können.



Durch langes Drücken auf eine Kategorie erscheint ein kleines **Menü zum Bearbeiten und Löschen eines Elements**.



Wählt man „Bearbeiten“ so öffnet sich ein Fenster zum **Ändern des Namens der Kategorie**.

Wählt man **Löschen** so wird die Kategorie gelöscht, vorher wird wie bei einer Person ein kleiner Infotext ausgegeben.

## 6. Datenbank

Die App hat eine Datenbank als Bestandteil. Dafür wird die **SQLite-Datenbank** benutzt. Diese ermöglicht das Speichern sämtlicher erfasster Daten sowie das Auslesen zum Verarbeiten und Darstellen von Daten.

### 6.1. Aufbau der Datenbank

Die Datenbank besteht aus den folgenden vier Tabellen:

- Payments (Ausgaben)
- Repayments (Rückzahlungen)
- Persons (Personen)
- Categories (Kategorien)

| PAYMENTS     |                 |          |
|--------------|-----------------|----------|
| Spalte       | Sql-Bezeichnung | Datentyp |
| ID           | id              | integer  |
| Zeitstempel  | dateTime        | text     |
| Kreditor-Id  | creditorId      | integer  |
| Debitor-Id   | debtorId        | integer  |
| Betrag       | value           | decimal  |
| Kategorie-Id | categoryId      | integer  |
| Bemerkungen  | details         | text     |
| Gesamtbetrag | sum             | decimal  |

| REPAYMENTS  |                 |          |
|-------------|-----------------|----------|
| Spalte      | Sql-Bezeichnung | Datentyp |
| ID          | id              | integer  |
| Zeitstempel | dateTime        | text     |
| Betrag      | value           | decimal  |
| Kreditor-Id | creditorId      | integer  |
| Debitor-Id  | debtorId        | integer  |
| Bemerkungen | details         | text     |

| PERSONS     |                 |          |
|-------------|-----------------|----------|
| Spalte      | Sql-Bezeichnung | Datentyp |
| ID          | id              | integer  |
| Bezeichnung | name            | text     |

| CATEGORIES  |                 |          |
|-------------|-----------------|----------|
| Spalte      | Sql-Bezeichnung | Datentyp |
| ID          | id              | integer  |
| Bezeichnung | name            | text     |

### 6.2. Database-Klasse

Das Speichern und Auslesen von Daten aus der Datenbank wird mithilfe der Klasse „Database“ ermöglicht in der Anwendung ermöglicht. Die Klasse beinhaltet die von Android bereitgestellte Klasse **„SQLiteOpenHelper“**, welche den Zugriff auf eine SQLite-Datenbank ermöglicht. Die Klasse Database beinhaltet die Variable „Database“ vom Typ **„SQLiteDatabase“**. Diese Variable repräsentiert die Datenbank. Die Klasse Database stellt des Weiteren alle Methoden zum Auslesen und Speichern der Daten bereit.

Die Klasse „Sql“ alle Tabellen- und Spaltennamen sowie „Create Table“-Befehle zum initialisieren der Datenbank bereit.

Das Auslesen von Daten erfolgt in diesem Fall durch einen sogenannten „Cursor“, während das Speichern, Ändern und Löschen von Daten über einen einfachen Sql-Query und dem Befehl `>> Database.execSQL(sql) << erfolgt.`