

Studienprojekt im Rahmen der Vorlesung

Mobile Computing



CostPatrol

**Konzeption und Realisierung einer Android-basierten Appli-
kation zur Erfassung und Verwaltung von Ausgaben**

Dozent: Prof. Dr. Markus U. Mock

Autor: Jörn Hornickel

Matrikelnummer: 1035701

Landshut, den 14.05.2014

Inhalt

1. Einführung.....	1
1.1. Beschreibung und Vision	1
1.2. Zielgruppe	1
1.3. Vermarktungsstrategie	2
2. Planung und Konzeption.....	3
2.1. Funktionale Anforderungen.....	3
2.2. Oberflächenentwürfe und Navigation	12
2.3. Releaseschnitt und -planung	15
3. Entwurf und Implementierung	17
3.1. Activities.....	18
3.2. Domänenobjekte.....	19
3.3. Datenzugriffsschicht	20
4. Test	22
4.1. Stufe 1 - Junit-Tests	22
4.2. Stufe 2 - Smoke-Test.....	23
4.3. Stufe 3 - Manuelle Tests	26

Abbildungsverzeichnis

Abbildung 1 - Anwendungsfalldiagramm der mobilen Applikation „CostPatrol“	3
Abbildung 2 - Ausschnitt des Oberflächenentwurfs der Startansicht.....	4
Abbildung 3 - Ausschnitt des Oberflächenentwurfs zum Anlegen von Ausgaben	4
Abbildung 4 - Kontextabhängiger Modus der ActionBar beim Löschen von Ausgaben	6
Abbildung 5 - Ausschnitt der Auflistung von Ausgabenserien.....	6
Abbildung 6 - Ausschnitt der Ansicht zum Anlegen neuer Aufgaben Serien.....	7
Abbildung 7 - Ausschnitt des Oberflächenentwurfs der Ausgabenstatistik.....	10
Abbildung 8 - Ausschnitt des Dialogentwurfs der Kategoriedetails	10
Abbildung 9 - Ausschnitt des Entwurfs der Kategorieanzeige	11
Abbildung 10 - Entwurf des Dialogs zum Ändern von Kategorien	11
Abbildung 11 - Oberflächenentwürfe und Navigation	14
Abbildung 12 - Releaseplanung.....	16
Abbildung 13 - Struktureller Aufbau der Applikation.....	17
Abbildung 14 - Vereinfachtes Klassendiagramm der ListActivities.....	18
Abbildung 15 - Vereinfachtes Klassendiagramm der Activities zum Neuanlegen	19
Abbildung 16 - Klassendiagramm des Domänenmodells	20
Abbildung 17 - Klassendiagramm des Domänenmodells	21

Tabellenverzeichnis

Tabelle 1 - Beispiele zur Ermittlung des ersten Buchungstermins.....	8
Tabelle 2 - Beispiele zur Ermittlung des nächsten Buchungstermins	9
Tabelle 3 - Releasezuordnung der Funktionen	16
Tabelle 4 - Manuelle Schritte des Smoke-Tests	25
Tabelle 5 - Testfälle zur Funktion „Kategorie anlegen“	27

1. Einführung

1.1. Beschreibung und Vision

Ist am Monatsende wieder weniger Geld auf dem Konto als erwartet und ist unklar wofür es ausgegeben wurde? Oder fehlt der Überblick, in welchen Bereichen leicht Einsparungen vorgenommen werden könnten?

Die mobile Applikation „CostPatrol“ ermöglicht dem Benutzer diese und weitere Fragen zu beantworten. Mit der App lassen sich tägliche Ausgaben schnell und einfach erfassen, wodurch eine umfassende Übersicht der persönlichen Kostensituation entsteht. Regelmäßige Ausgaben, wie beispielsweise Mietzahlungen, lassen sich zudem als Ausgabenserien hinterlegen. Jeder Ausgabe wird bei der Speicherung eine frei konfigurierbare Kategorie zugeordnet. Dadurch ist eine spätere Gruppierung möglich, welche die Kontrolle und die Analyse des Ausgabenverhaltens erleichtern. Die Erhebung, Beobachtung und Analyse der Ausgaben wird zudem durch ein intuitives und konsistentes Bedienungskonzept unterstützt, das sich konsequent an den Richtlinien und Best Practices der Android-Plattform orientiert. Dies führt zu einer Vereinfachung der Bedienung und erleichtert die Nutzung der App im Alltag.

Bei kontinuierlicher Nutzung kann die App nicht nur dabei helfen die Kostentransparenz und das Kostenbewusstsein zu steigern, sondern auch konkrete Einsparpotentiale aufzudecken. Hiermit bildet die mobile Applikation die Grundlage, um beispielsweise monatliche Kosten zu senken. Neben diesem Einsatzszenario, bei dem die mobile Applikation typischerweise dauerhaft genutzt wird und alle anfallenden Kosten festgehalten sowie analysiert werden, unterstützt die App noch weitere Einsatzszenarien. Sowohl eine temporäre Nutzung der App, beispielsweise auf Reisen oder während einer Shoppingtour, als auch eine dauerhafte Nutzung mit dem Fokus auf bestimmte einzelne Ausgabenkategorien können sich als lohnenswert erweisen.

Ergänzend zu den bisher genannten Einsatzszenarien, ist die Synchronisation der mobil erfassten Daten mit einer serverbasierten Applikation angedacht. Der Benutzer soll dabei seine erfassten Daten über eine Weboberfläche aufrufen können. Dies ermöglicht eine erweiterte Analyse der Daten und deren Weiterverarbeitung auf dem heimischen Desktop-PC.

1.2. Zielgruppe

Die Zielgruppe von „CostPatrol“ zeichnet sich durch ein Bestreben oder einen Zwang zu einer sparsamen Lebensweise aus. Dies kann entweder soziografisch bedingt durch ein niedriges Haushaltseinkommen oder basierend auf psychografischen Eigenschaften hinsichtlich der persönlichen Einstellung begründet sein. Die Altersuntergrenze der Zielgruppe liegt bei etwa 20 Jahren und ergibt sich aus der Gründung eines ersten eigenen Haushalts. Ein typischer Benutzer dieser Altersuntergrenze befindet sich in einer Berufsausbildung, einem Studium, einem freiwilligen Dienst oder bereits im Berufseinstieg. Die Altersobergrenze der Hauptziel-

gruppe liegt bei 40 Jahren und setzt sich typischerweise aus Benutzern mit einem hohen Bestreben zur Sparsamkeit zusammen.

Unabhängig von persönlichen Eigenschaften liegen besonders die Benutzer im Fokus, die ihre Ausgabenituation über einen längeren Zeitraum und möglichst umfassend protokollieren und überwachen möchten. Aufgrund der hohen Nutzungsintensität werden in diesem Zielgruppensegment die höchsten Einnahmen durch In-App-Werbung erwartet.

Eine weitere Abgrenzung der Zielgruppe ergibt sich aus den bestehenden technischen Systemvoraussetzungen. Der potentielle Benutzer benötigt ein Smartphone, der vorgegebenen Android-Plattform, das mindestens mit der Version 3.0 (Honeycomb) ausgestattet sein muss. Rund 80 Prozent der aktuell genutzten Android-Geräte erfüllen dieses Kriterium¹. Aufgrund von hohen veranschlagten Aufwänden zur Realisierung der Abwärtskompatibilität werden ältere Versionen vorerst nicht unterstützt.

1.3. Vermarktungsstrategie

Eine Analyse der Marktsituation hat ergeben, dass in den verschiedenen Distributionsplattformen, wie beispielsweise „Google Play“, bereits einige Apps mit ähnlichem Funktionsumfang angeboten werden. Die Begutachtung von sieben der im Seitenranking zuerst angezeigten Apps, kristallisierte einige Aspekte heraus, in denen sich die geplante App „CostPatrol“ differenzieren kann. Ein Bereich, in dem ein Alleinstellungsmerkmal manifestiert werden kann, liegt in der Gestaltung von ansprechenden Oberflächen und einer intuitiven Bedienung. Hierzu sollten sich diese an den Richtlinien und Best Practices der Android-Plattform orientieren. Ein weiteres Merkmal zur Abgrenzung wird in der Entwicklung der ergänzenden Webanwendung zur Beobachtung und Kontrolle der Ausgaben erwartet. Aufgrund diverser Erweiterungsmöglichkeiten liegt in der Web-Anwendungskomponente aller Voraussicht nach das größte Potential zur Alleinstellung und Kundenbindung.

Die Vermarktung der App ist sowohl über die Distributionsplattform von Google („Google Play“) als auch über die von Amazon („Amazon App-Shop“) geplant. Die beschriebenen Eigenschaften der Zielgruppe lassen auf keine Zahlungsbereitschaft bei einer kostenpflichtigen Platzierung der App schließen. Daher soll „CostPatrol“ auf beiden Plattformen kostenlos angeboten werden.

Da die App im Rahmen des Studienprojektes konzipiert und entwickelt wird, wird keine Deckung für die entstehenden Aufwände angestrebt. Nach erfolgreichem Rollout soll allerdings ein Teil der Entwicklungskosten durch In-App-Werbung finanziert werden. Vorstellbar wäre hier bspw. die Einbindung von Google AdMob oder vergleichbaren Werbenetzwerken. Die genaue Auswahl soll vorgenommen werden, nachdem erste Downloadzahlen einen Trend anzeigen und eine Prognose erlauben. Auf diese Weise kann das geeignete Werbenetzwerk ermittelt werden.

¹ <http://developer.android.com/about/dashboards/index.html#Platform>

2. Planung und Konzeption

2.1. Funktionale Anforderungen

Das in der folgenden Abbildung 1 dargestellte Anwendungsfalldiagramm liefert eine erste Übersicht über die Funktionen von „CostPatrol“. Die Anwendungsfälle „Ausgaben verwalten“, „Ausgabenserien verwalten“ und „Kategorien verwalten“ stellen jeweils eine Zusammenfassung der Funktionen zur Anzeige, zum Anlegen, zum Ändern und zum Löschen dar.

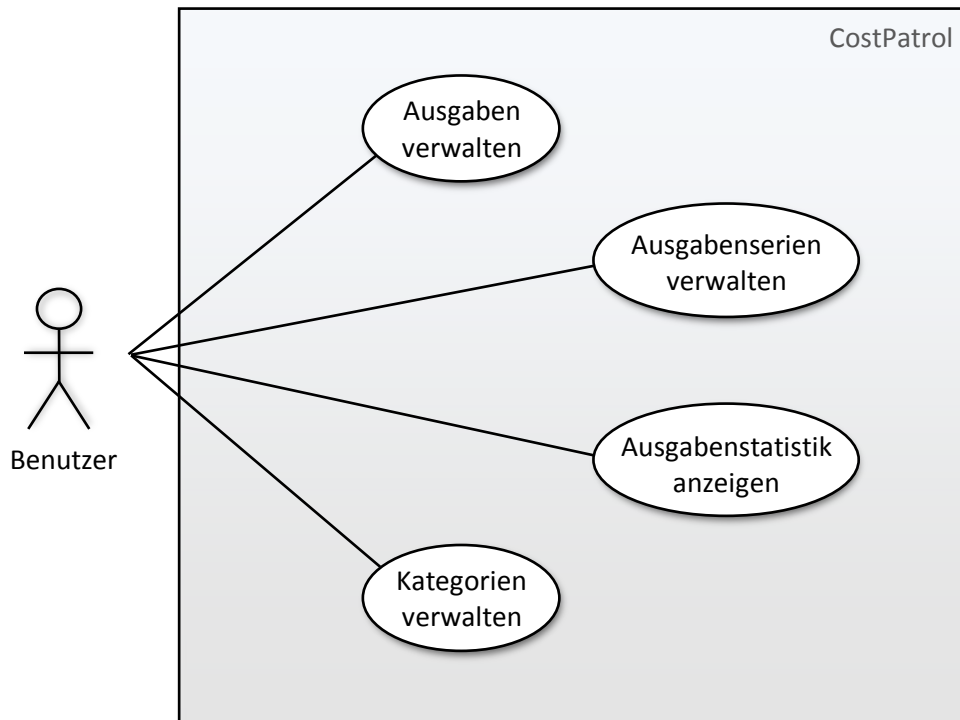


Abbildung 1 - Anwendungsfalldiagramm der mobilen Applikation „CostPatrol“

In den folgenden Unterkapiteln werden nun die Funktionen der mobilen Applikation einzeln angeführt und detailliert beschrieben. Nachfolgend werden übergreifende Anforderungen für die Darstellung in Listenform definiert. Zur Unterstützung des Verständnisses werden Ausschnitte von Oberflächenentwürfen verwendet, die im Kapitel 2.2 vollständig dargestellt werden.

2.1.1. Ausgaben anzeigen

Nach dem Anwendungsstart werden dem Benutzer alle erfassten Ausgaben in einer Liste angezeigt. Zur Sicherstellung der Übersicht, soll wie in Abbildung 2 dargestellt, eine Liste mit aufklappbaren Elementen verwendet werden. Deren Gruppen geben eine Kalenderwoche mit Jahreszahl an und beinhalten die Ausgaben, die sich anhand ihres Datums der Kalenderwoche zuordnen lassen.



Abbildung 2 - Ausschnitt des Oberflächenentwurfs der Startansicht

Ein Listeneintrag zu einer Ausgabe besteht aus den folgenden Informationen:

- Datum im Format "tt.mm.", an dem die Ausgabe getätigt wurde
- Bezeichnung der zugeordneten Kategorie
- Beschreibung der Ausgabe
- Preis mit Tausender- und Dezimaltrennzeichen, zwei Nachkommastellen und einem Euro-Zeichen

Die Sortierung der Gruppenelemente erfolgt nach der Kalenderwoche (inklusive Berücksichtigung des Jahres) absteigend. Die Listeneinträge der Ausgaben werden absteigend nach dem angezeigten Datum ("tt.mm."), an dem die Ausgaben getätigt wurden und danach nach dem Zeitstempel der Erfassung sortiert.

2.1.2. Ausgabe anlegen

Über das „Add“-Icon gelangt der Benutzer von der Auflistung der bereits erfassten Ausgaben zur Oberfläche zum Anlegen von neuen Ausgaben. Zur Eingabe der notwendigen Informationen sind die in Abbildung 3 enthaltenen Komponenten vorgesehen, die nachfolgend beschrieben werden.



Abbildung 3 - Ausschnitt des Oberflächenentwurfs zum Anlegen von Ausgaben

- Ein Feld zur Auswahl des Datums, an dem die Ausgabe getätigt wurde im Format "tt.mm.jjjj". Standardmäßig wird dieses Feld mit dem aktuellen Tagesdatum vorbelegt. Zur Änderung des Datumswertes öffnet sich bei einem Klick auf das Feld ein DatePicker.
- Eine Auswahlliste der Bezeichnungen der aktiven Kategorien, die der Ausgabe zugeordnet werden sollen. Die Kategorien werden hierbei alphabetisch absteigend anhand der Bezeichnung sortiert. Die oberste Kategorie wird zudem als Vorauswahl verwendet.
- Ein Textfeld mit Vorschlagsfunktion (auch auto-complete genannt) zur Kurzbeschreibung der Ausgabe. Bei Betreten des Feldes soll die eingeblendete Tastatur für den ersten Buchstaben auf Großbuchstaben eingestellt sein, danach auf Kleinbuchstaben. Ab der Eingabe des ersten Buchstabens einer Kurzbeschreibung sollen Vorschläge erscheinen, die die bisherige Eingabe der Kurzbeschreibung vervollständigen. Grundlage der Vorschläge sind die Kurzbeschreibungen aller bisher erfassten Ausgaben aus den letzten sechs Wochen. Es werden keine doppelten Werte angezeigt. Die Vorschlagsfunktion soll zudem dafür sorgen, dass bei Eingabe von Worten mit den Vokalen 'a','u','o' auch Worte mit entsprechenden Umlauten vorgeschlagen werden. Beispielsweise sollen bei der Eingabe von „Mu“ auch Vorschläge wie „Münchner Tiergarten“ erscheinen, sofern eine Ausgabe mit dieser Kurzbeschreibung in den letzten sechs Wochen erfasst wurde. Dadurch kann bei der wiederholten Eingabe von Beschreibungen mit Umlauten, die umständliche Auswahl des jeweiligen Umlauts entfallen und die Erfassung beschleunigt werden.
- Ein Textfeld zur Eingabe des Preises. Bei Betreten soll die eingeblendete Tastatur ausschließlich Ziffern und ein Dezimaltrennzeichen zur Eingabe anbieten. Fehleingaben in Form eines doppelten Dezimaltrennzeichens oder Einfügen von Buchstaben über die Zwischenablage sollen unterdrückt werden. Außerdem soll die Eingabe von mehr als zwei Nachkommastellen verhindert werden.

Über ein Icon der ActionBar hat der Benutzer die Möglichkeit zur Speicherung der Ausgabe. Neben den vorgenommenen Eingaben muss auch der aktuelle Zeitstempel persistiert werden, um in der Auflistung eine erwartungskonforme Sortierung realisieren zu können. Die Speicherung der Ausgabe soll allerdings nur durchgeführt werden, wenn die Eingabedaten erfolgreich validiert wurden. Andernfalls wird der Benutzer auf die ungültigen Daten hingewiesen. Zur erfolgreichen Validierung müssen folgende Aspekte erfüllt sein:

- Es wurde eine Kategorie ausgewählt. Aufgrund der Vorselektion ist dies immer erfüllt, wenn mindestens eine aktive Kategorie vorhanden ist.
- Das Feld für die Kurzbeschreibung enthält mindestens drei und maximal 30 Zeichen.
- Das Preisfeld enthält einen Geldbetrag der mindestens 0,01 € entspricht.

Neben der Speicherung hat der Benutzer die Möglichkeit die Ansicht zum Anlegen jederzeit zu verlassen und eine eventuell vorgenommene Eingabe damit zu verwerfen. Erreicht werden kann dies entweder über einen Klick auf das Symbol links neben dem ActionBar-Icon (sogenannte Up-Navigation) oder über die Zurück-Taste des Smartphones.

2.1.3. Ausgabe ändern

Über einen Klick auf einen Listeneintrag einer erfassten Ausgabe gelangt der Benutzer zur Oberfläche, in der die Daten der Ausgabe angezeigt und geändert werden können. Die Ober-

fläche und das Verhalten der Anwendung entspricht dabei, der zuvor beschriebenen Funktion „Ausgabe anlegen“.

2.1.4. Ausgaben löschen

Der Benutzer soll die Möglichkeit haben erfasste Ausgaben in der Startansicht zu löschen. Durch einen „langen Klick“ auf eine Ausgabe soll die Anzeige der ActionBar, wie in Abbildung 4 dargestellt, in einen kontextabhängigen Modus wechseln. In diesem Modus können weite-

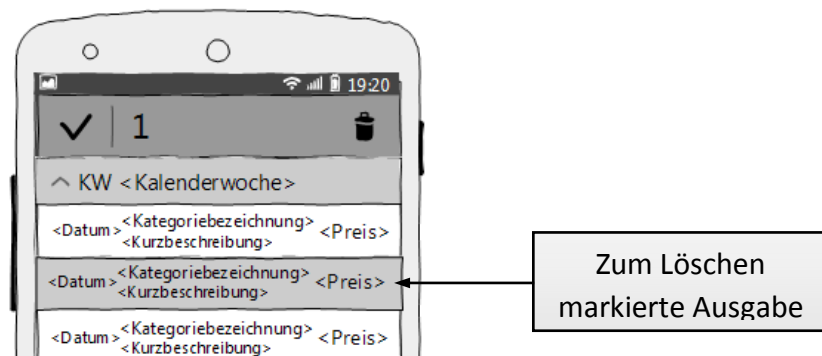


Abbildung 4 - Kontextabhängiger Modus der ActionBar beim Löschen von Ausgaben

re zu löschende Ausgaben mit einem „normalen“ Klick selektiert werden oder die Markierung von bereits selektierten Ausgaben entfernt werden. Im linken Bereich der ActionBar soll die Anzahl der zum Löschen markierten Ausgaben erscheinen. Bei Änderungen der Markierung aktualisiert sich die Anzahl entsprechend. Im rechten Bereich soll ein Icon platziert werden, über das per Klick der Löschvorgang, der zu diesem Zeitpunkt markierten Ausgaben ausgelöst wird.

2.1.5. Ausgabenserien anzeigen

Über die Aktion „Ausgabenserien verwalten“ der Startansicht gelangt der Benutzer zur Anzeige von Ausgabenserien. Eine Ausgabenserie ist eine Ausgabe, die periodisch anfällt und in einem konstanten Intervall automatisch gebucht wird. Die Auflistung der Ausgabenserien enthält, wie die folgende Abbildung zeigt, aufklappbare Listenelemente zur Gruppierung von monatlichen und vierteljährlichen Ausgaben. Ein Listeneintrag zu einer Ausgabenserie be-



Abbildung 5 - Ausschnitt der Auflistung von Ausgabenserien

steht aus folgenden Informationen:

- Datum der nächsten automatischen Buchung im Format "tt.mm.jjjj"
- Bezeichnung der zugeordneten Kategorie
- Kurzbeschreibung der Ausgabe
- Preis mit Tausender- und Dezimaltrennzeichen, zwei Nachkommastellen und einem Euro-Zeichen

Die Sortierung der Gruppenelemente erfolgt nach der Größe des Intervalls aufsteigend, so dass die Gruppe „Monatlich“ vor der Gruppe „Vierteljährlich“ erscheint. Die Listeneinträge der Ausgabenserien werden aufsteigend nach dem angezeigten Datum, der nächsten Buchung und danach nach dem Zeitstempel der Erfassung sortiert. Somit sieht der Benutzer die nächsten Buchungen auf den ersten Blick.

2.1.6. Ausgabenserien anlegen

Über das „Add“-Icon gelangt der Benutzer von der Anzeige der Ausgabenserie zum Anlegen einer neuen Ausgabenserie. Zur Eingabe der notwendigen Informationen sind die in Abbildung 6 enthaltenen Komponenten vorgesehen, die nachfolgend beschrieben werden.



Abbildung 6 - Ausschnitt der Ansicht zum Anlegen neuer Aufgabenserien

- Zwei Radiobuttons zur Auswahl des Intervalls, wobei „Monatlich“ vorgelegt ist.
- Ein Feld zur Auswahl des Datums, ab dem die Buchung der Ausgabenserie beginnen soll. Das Datum wird im Format "tt.mm.jjjj" angegeben. Standardmäßig wird dieses Feld mit dem aktuellen Tagedatum vorgelegt. Zur Änderung des Datumswertes öffnet sich bei einem Klick auf das Feld ein DatePicker.
- Ein Textfeld zur Eingabe des Tages im Monat, an dem die Buchung durchgeführt werden soll. Bei Betreten soll die eingeblendete Tastatur ausschließlich Ziffern zur Eingabe anbieten. Fehleingaben in Form von Werten außerhalb des Bereichs von 1 bis 31 werden nicht zugelassen.
- Eine Auswahlliste der Bezeichnungen der aktiven Kategorien.*
- Ein Textfeld für die Kurzbeschreibung.*
- Ein Textfeld zur Eingabe des Preises.*

* Die Eigenschaften dieser Komponenten entsprechen der Beschreibung im Rahmen der Funktion „Ausgabe anlegen“ und werden hier deshalb nicht nochmal beschrieben.

Über ein Icon der ActionBar hat der Benutzer die Möglichkeit die Speicherung der Ausgabeserie vorzunehmen. Die Speicherung der Ausgabeserien soll allerdings nur durchgeführt werden, wenn die Eingabedaten erfolgreich validiert wurden. Zur erfolgreichen Validierung müssen die im Rahmen der Funktion „Ausgabe anlegen“ definierten Aspekte erfüllt sein. Ist dies nicht der Fall, wird der Benutzer auf die ungültigen Daten hingewiesen.

Neben den Eingaben des Benutzers muss der aktuelle Zeitstempel für die Sortierung und das zu berechnende Datum der ersten Buchung persistiert werden. Das Datum der ersten Buchung ist der vom Benutzer eingegebene Beginn der Ausgabenserie, wenn dessen Tag, dem eingegebenen Tag des Monats entspricht (Beispiel 1). Andernfalls ist das Datum der nächste Tag vom Beginn ausgehend, der dem eingegebenen Tag des Monats gleicht (Beispiel 2-3). Eine Ausnahme besteht, wenn der Beginn der Ausgabenserie in einem Monat liegt, in dem die Anzahl der Tage kleiner ist, als der eingegebene Tag des Monats. In diesem Fall ist das Datum der ersten Buchung immer der Monatsletzte des Monats, in dem der Beginn liegt (siehe Beispiel 4). Die in der Tabelle 1 angeführten Beispiele verdeutlichen die Berechnung der ersten Buchung.

Beispiel	Beginn der Ausgabenserie	Buchungstag im Monats	Erste Buchung
1 - Normalfall	08.05.2014	8	08.05.2014
2 - Normalfall	08.05.2014	15	15.05.2014
3 - Normalfall	08.05.2014	1	01.06.2014
4 - Sonderfall	08.02.2014	31	28.02.2014

Tabelle 1 - Beispiele zur Ermittlung des ersten Buchungstermins

Nach der Speicherung der Ausgabenserie muss die Funktion „Automatische Buchung von Ausgabenserien“ ausgeführt werden.

Neben der Speicherung hat der Benutzer die Möglichkeit die Ansicht zum Anlegen jederzeit zu verlassen und eine eventuell vorgenommene Eingabe damit zu verwerfen. Erreicht werden kann dies entweder über einen Klick auf das Symbol links neben dem ActionBar-Icon (sogenannte Up-Navigation) oder über die Zurück-Taste des Smartphones.

2.1.7. Ausgabeserie ändern

Über einen Klick auf einen Listeneintrag einer erfassten Ausgabeserie gelangt der Benutzer zur Oberfläche, in der die Daten der Ausgabe angezeigt und geändert werden können. Die Oberfläche und das Verhalten der Anwendung entspricht dabei, der zuvor beschriebenen Funktion „Ausgabeserie anlegen“. Eine Besonderheit ergibt sich allerdings für das Beginn-Datum der Ausgabenserie. Beim Öffnen der Ansicht zum Ändern wird hier immer das tagesaktuelle Datum hinterlegt und nicht das ursprünglich angegeben Beginn-Datum geladen. Änderungen des Datums über den DatePicker lassen neben dem tagesaktuellen Wert nur Datumswerte der Zukunft zu. Auf diese Weise ist sichergestellt, dass Änderungen von Ausgabenserien nur die Buchung zukünftiger Ausgaben beeinflussen. Wird das Intervall oder der

Tag des Monats einer Ausgabenserie verändert und die Änderung gespeichert, muss das Datum für die nächste Buchung neu berechnet werden.

2.1.8. Ausgabenserien löschen

Der Benutzer soll ausgehend von der Auflistung der Ausgabenserien die Möglichkeit haben erfasste Ausgabenserien zu löschen. Hierbei verhält sich die mobile Applikation wie bereits im Rahmen der Funktion „Ausgaben löschen“ beschrieben.

2.1.9. Automatische Buchung von Ausgabenserien

Die automatische Buchung von Ausgabenserien ist für den Benutzer transparent und soll sowohl beim ersten Anwendungsstart pro Tag als auch nach dem Anlegen neuer Ausgabenserien geprüft und gegebenenfalls vorgenommen werden. Die automatische Buchung ist solange durchzuführen, bis kein Datum der nächsten Buchung einer Ausgabenserie mehr dem Datum des aktuellen Datums entspricht oder in der Vergangenheit liegt.

Falls eine Ausgabe einer Serie zu buchen ist, wird im ersten Schritt eine neue Ausgabe erstellt und befüllt. Hierzu werden die Kategorie, die Kurzbeschreibung und der Preis der Ausgabenserie übernommen. Als Datum der Ausgabe wird der Termin der nächsten Buchung der Ausgabenserie verwendet. Im zweiten Schritt muss ein neuer Termin für die nächste Buchung ermittelt und gespeichert werden. Beide Schritte müssen im Rahmen einer Datenbanktransaktion erfolgen, um Fehler im Datenbestand zu vermeiden.

Das Datum der nächsten Buchung entspricht dem, der letzten (=aktuellen) Buchung, zuzüglich des Intervalls der Serie (Beispiel 1-3). Hierbei gilt die Ausnahme für Monate, in denen die Anzahl der Tage kleiner ist, als der eingegebene Buchungstag im Monat. In diesem Fall ist das Datum der ersten Buchung immer der Monatsletzte des Monats, in dem der Beginn liegt (Beispiel 4). Traf diese Ausnahmeregelung bereits auf den letzten Buchungstermin zu, muss neben der Addition des Intervalls, der vom Benutzer bestimmte Buchungstag im Monat berücksichtigt werden (Beispiel 5). Die folgende Tabelle 2 enthält Beispiele dieser Ermittlung und deren Sonderfälle.

Beispiel	Buchungstag im Monats	Intervall	Letzte Buchung	Nächste Buchung
1 - Normalfall	8	Monatlich	08.05.2014	08.06.2014
2 - Normalfall	15	Vierteljährlich	08.05.2014	08.08.2014
3 - Normalfall	28	Monatlich	28.02.2014	28.03.2014
4 - Sonderfall	31	Monatlich	31.03.2014	30.04.2014
5 - Sonderfall	31	Vierteljährlich	28.02.2014	31.05.2014

Tabelle 2 - Beispiele zur Ermittlung des nächsten Buchungstermins

2.1.10. Ausgabenstatistik anzeigen

Über ein entsprechendes ActionBar-Icon der Startansicht gelangt der Benutzer zu der Ausgabenstatistik (siehe Abbildung 7). In Form von aufklappbaren Listenelementen werden die erfassten Ausgaben zunächst monatsweise und danach auf Basis der Kategorien gruppiert. Die aufklappbaren Einträge des Monats sollen nach der Monats- und Jahresangabe absteigend sortiert werden. Ein Monatseintrag besteht aus der Angabe des Monats (in Textform)

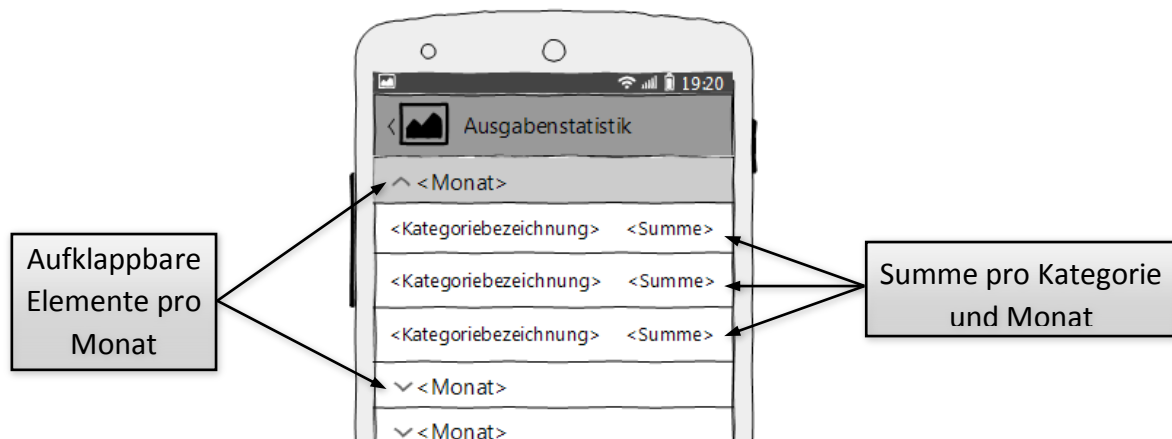


Abbildung 7 - Ausschnitt des Oberflächenentwurfs der Ausgabenstatistik

mit der Jahreszahl und aus der Summe der Ausgaben in diesen Monat. Die Summe wird mit Tausender- und Dezimaltrennzeichen, zwei Nachkommastellen und dem Euro-Zeichen angezeigt.

Wird ein Monatelement aufgeklappt, erscheinen Listeneinträge mit Kategoriebezeichnungen und der Summe der Ausgaben für diese Kategorien. Das Format der Summe entspricht dem zuvor beschriebenen Format der Monatssumme. Die Listeneinträge werden nach der Summe der Kategorie absteigend sortiert. Beim Öffnen dieser Oberfläche ist die Gruppe des am weitesten in der Zukunft liegenden Monats aufgeklappt und stellt die zugehörigen Kategorien dar. Die Elemente anderer Monate sind geschlossen.

Klickt der Benutzer auf einen Listeneintrag einer Kategorie, soll der in Abbildung 8 ausschnittsweise skizzierte Dialog erscheinen. Der Titel des Dialogs gibt die Bezeichnung der Ka-

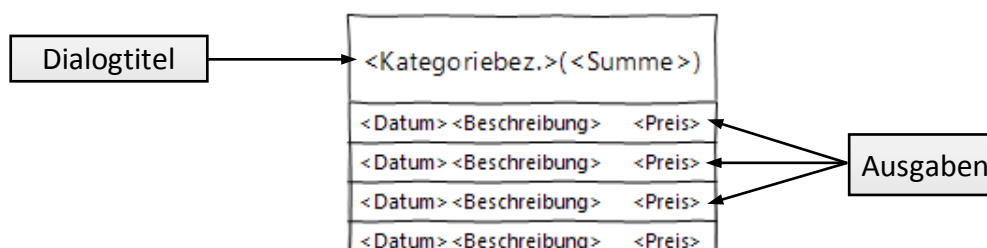


Abbildung 8 - Ausschnitt des Dialogentwurfs der Kategoriedetails

tegorie und die Summe für diese Kategorie in dem betreffenden Monat wieder. Die Listeneinträge beinhalten das Datum im Format "tt.mm", den Kurzbeschreibungstext und den jeweiligen Preis der Ausgabe. Die Listeneinträge werden aufsteigend nach dem angezeigten Datum ("tt.mm.") und anschließend nach dem Zeitstempel der Erfassung sortiert. Dadurch erhält der Benutzer einen chronologischen Überblick der getätigten Ausgaben der ausgewählten Kategorie. Der Dialog kann durch einen Button geschlossen werden, der unterhalb der Listeneinträge zu platzieren ist.

2.1.11. Kategorien anzeigen

Über die Aktion „Kategorien verwalten“ der Startansicht gelangt der Benutzer zur Oberfläche in der alle Kategorien aufgelistet sind (siehe Abbildung 9). Ein Listeneintrag besteht aus der

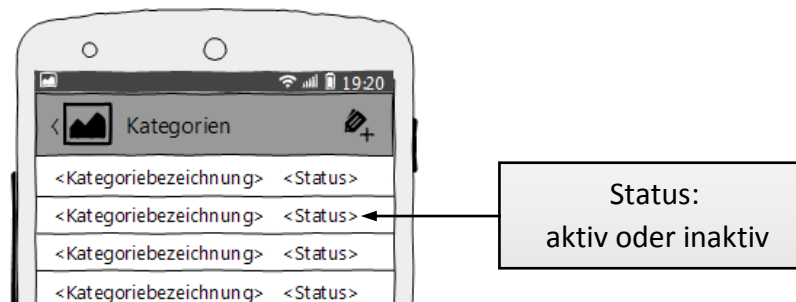


Abbildung 9 - Ausschnitt des Entwurfs der Kategorieanzeige

Bezeichnung und dem Status (aktiv/inaktiv) einer Kategorie. Die Sortierung erfolgt absteigend nach der Bezeichnung der Kategorien. Die mobile Applikation enthält im Auslieferungszustand bereits eine Reihe von angelegten Kategorien, da sonst keine Erfassung von Ausgaben möglich wäre.

2.1.12. Kategorie anlegen

Von der Oberfläche zur Anzeige der Kategorien hat der Benutzer über das „Add“-Icon der ActionBar die Möglichkeit zum Anlegen einer neuen Kategorie. Das Anlegen der Kategorie erfolgt in einem Dialog, in dem ein Textfeld zur Eingabe der Bezeichnung der Kategorie vorhanden ist. Bei Betreten des Feldes soll die eingblendete Tastatur für den ersten Buchstaben auf Großbuchstaben eingestellt sein, danach auf Kleinbuchstaben.

Der Dialog soll dem Benutzer einen Button zum Abbruch und Verlassen der Eingabe sowie einen Button zum Speichern der Kategorie bieten. Beim Speichern wird deren Status immer auf aktiv gesetzt. Die Speicherung der Eingaben erfolgt allerdings nur nach erfolgreicher Validierung. Andernfalls wird der Benutzer auf die ungültigen Daten hingewiesen. Die Eingabe des Textfelds für die Bezeichnung muss aus mindestens drei und maximal aus 20 Zeichen bestehen.

2.1.13. Kategorie ändern

Über einen Klick auf einen Listeneintrag einer angelegten Kategorie öffnet sich ein Dialog zum Ändern einer Kategorie, der in der folgenden Abbildung 10 dargestellt ist. Der Dialog

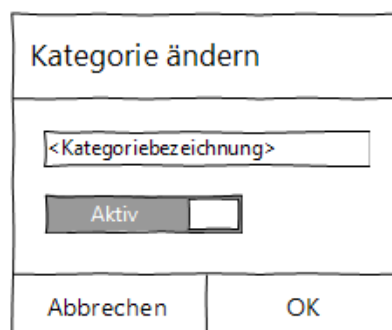


Abbildung 10 - Entwurf des Dialogs zum Ändern von Kategorien

enthält zusätzlich zum Anlegen von Kategorien eine Komponente, über die der Status der Kategorie (aktiv/inaktiv) geändert werden kann. Die Validierung und die angebotenen Buttons entsprechen der Funktion „Kategorie anlegen“.

2.1.14. Kategorien löschen

In der Anzeige der angelegten Kategorien soll der Benutzer die Möglichkeit zur Löschung von Kategorien haben. Die Bedienung zur Markierung der zu löschenden Einträge und zum Auslösen des Löschvorgangs verhält sich hierbei identisch mit den Funktionen „Ausgaben löschen“ und „Ausgabenserien löschen“. Bei der Löschung der Kategorie besteht allerdings die Einschränkung, dass nur die Kategorien gelöscht werden können, die in keiner referenziellen Abhängigkeit zu den Datensätzen von Ausgaben oder Ausgabenserien stehen. Sollten Kategorien von Datensätzen referenziert werden, können diese nicht gelöscht werden und werden stattdessen inaktiv gesetzt. Ein entsprechender Dialog informiert den Benutzer darüber und führt hierbei alle Bezeichnungen der Kategorien an, für die dies gilt.

2.1.15. Übergreifende Anforderungen für Listendarstellungen

Als Grundlage für ein konsistentes Verhalten aller Listendarstellungen von „CostPatrol“, müssen die folgenden Aspekte erfüllt werden.

- Während des Ladevorgangs der anzuzeigenden Daten soll die subjektiv wahrgenommene Wartezeit des Benutzers durch Einblendung einer sogenannten ProgressBar verkürzt werden. Da die genaue Ladedauer aufgrund diverser Einflussfaktoren nicht bestimmbar ist, wird die ProgressBar in Form eines animierten Kreises benutzt, der dem Benutzer auf die Ausführung eines Vorgangs hinweist. Gemäß der Empfehlung der Android Developer-Webseite² wird die ProgressBar ohne zusätzlichen Text verwendet.
- Der Ladevorgang aller Listen findet außerhalb des UI-Threads statt.
- Sind keine anzuzeigenden Daten vorhanden, erscheint statt der Liste ein entsprechender Hinweis.
- Wird die Ausrichtung des mobilen Gerätes vom Hochformat („Portrait“) zum Querformat („Landscape“) oder umgekehrt geändert, sind die relevanten Zustände der Liste wiederherzustellen. Als relevant gilt die Selektierung von Zeilen und ergänzend bei Listen mit aufklappbaren Elementen, der Zustand der bestimmt, ob ein Eintrag auf- oder zugeklappt erscheint.
- Beim Öffnen einer Listenansicht, die aufklappbare Elemente enthält, ist jeweils der erste Eintrag aufgeklappt und alle weiteren zugeklappt.
- Eine Gruppe eines aufklappbaren Listenelements ist nur anzuzeigen, wenn sie Untereinträge beinhaltet.

2.2. Oberflächenentwürfe und Navigation

Abbildung 11 zeigt die Entwürfe der geplanten Oberflächen und deren Navigierbarkeit. Aus Gründen der Übersichtlichkeit wurde weder die Up-Navigation (über das Symbol links neben dem ActionBar-Icon) noch die Navigation über die Zurück-Taste des Android-Gerätes explizit visualisiert. Außerdem wurde auf die Darstellung des kontextabhängigen Modus der ActionBar beim Löschen, eventuellen Folgedialogen und der Validierung in den Entwürfen verzich-

² <http://developer.android.com/design/building-blocks/progress.html>

tet. Die blau dargestellten Pfade stellen in der Abbildung die Vorwärtsnavigation dar. Die roten Pfade zeigen dagegen, welche Möglichkeiten (neben der Up-Navigation und der Zurück-Taste) existieren, um zur vorher angezeigten Ansicht zurückzukehren.

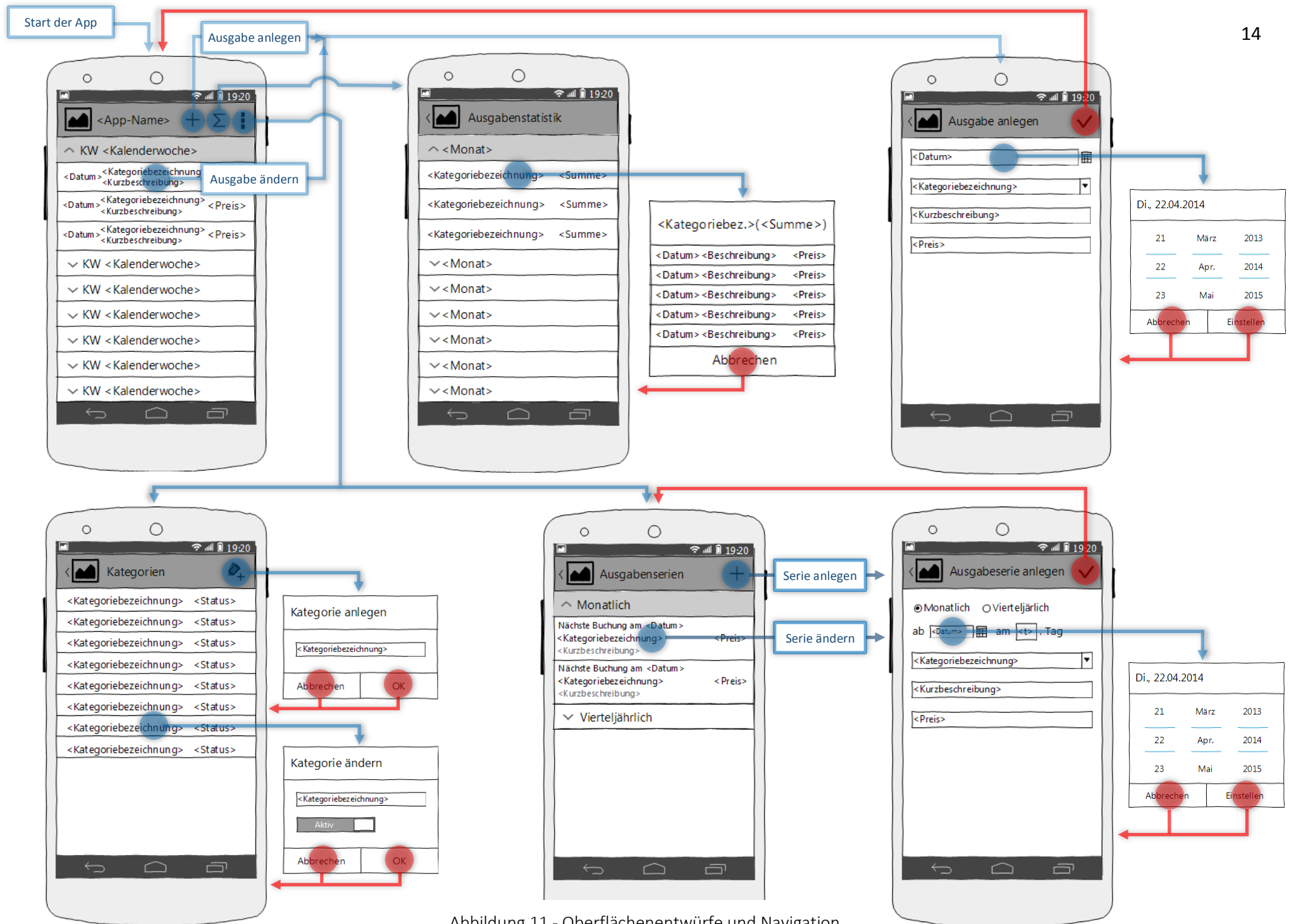


Abbildung 11 - Oberflächenentwürfe und Navigation

2.3. Releaseschnitt und -planung

Die folgende Tabelle zeigt die Zuordnung der zuvor beschriebenen Funktionen (fett markiert) und die dafür notwendigen Tätigkeiten zu den geplanten Releases. Die Entwicklung von JUnit-Tests ist in der Tabelle nicht explizit aufgeführt, da sie parallel zur Entwicklung vorgenommen wurde.

Release	Funktion mit den wesentlichen Tätigkeiten
0.1	<ul style="list-style-type: none"> • Einrichtung der Entwicklungsumgebung <ul style="list-style-type: none"> ○ Integration der Versionsverwaltung ○ Konfiguration des Code-Formatters • Anlegen der Projektstruktur • Bereitstellung einer ausführbaren Version mit rudimentärer Start-Activity
0.2	<ul style="list-style-type: none"> • Datenbankanbindung • Erstellung der Datenbanktabellen für Ausgaben- und Kategorien • Entwicklung der SQL-Statements für Ausgaben
0.3	<ul style="list-style-type: none"> • Ausgaben anlegen und Ausgaben ändern <ul style="list-style-type: none"> ○ Layout-Definition ○ Navigation zur Activity ○ Konfiguration der View-Komponenten (z. B. Spinner, Auto-Complete- oder Preis-Feld. Kategoriedaten werden aus Array gelesen) ○ Befüllung der View-Komponenten ○ Validierung der Eingaben ○ Speicher-Funktion
0.4	<ul style="list-style-type: none"> • Ausgaben anzeigen (in Liste mit aufklappbaren Elementen) <ul style="list-style-type: none"> ○ Entwicklung von SQL-Statements zur Ermittlung von Gruppen und deren Einträge • Ausgaben löschen
0.5	<ul style="list-style-type: none"> • Ausgabenstatistik anzeigen <ul style="list-style-type: none"> ○ Layout-Definition ○ Navigation zur Ausgabenstatistik ○ Aufklappbaren Liste inklusive SQL-Statements zur Ermittlung von Gruppen und deren Einträge • Detail-Statistik-Dialog für Ausgaben pro Monat und Kategorie
1.0	<ul style="list-style-type: none"> • Kategorien anzeigen, Kategorie anlegen, Kategorie ändern, Kategorie löschen <ul style="list-style-type: none"> ○ Anpassung der Tabellenstruktur der Datenbank ○ Befüllung der Auswahlliste bei Erfassung von Ausgaben ○ Erstellung der Activities zum Anzeigen, Anlegen und Ändern von Kategorien ○ Validierung der Eingaben ○ Entwicklung von Mechanismus falls Löschen nicht möglich ist
1.1	<ul style="list-style-type: none"> • Bugfixing • Refactoring für eventuell auftretende Schwachstellen

1.2	<ul style="list-style-type: none"> • Ausgabenserien anzeigen, Ausgabenserie anlegen, Ausgabenserie ändern, Ausgabenserie löschen <ul style="list-style-type: none"> ○ Erweiterung der Tabellenstruktur der Datenbank ○ Erstellung der Activities zum Anzeigen, Anlegen und Ändern ○ Validierung der Eingaben • Automatische Buchung von Ausgabenserien
1.3	<ul style="list-style-type: none"> • Bugfixing
1.4	<ul style="list-style-type: none"> • Optionale Funktionen <ul style="list-style-type: none"> ○ Serversynchronisation der Daten und deren Darstellung ○ Vergabe eines Budgets für Kategorien

Tabelle 3 - Releasezuordnung der Funktionen

Die Realisierung der zuvor angeführten Releases muss aufgrund des vorgegebenen Zeitrahmens innerhalb von sechs Wochen erfolgen. Abbildung 12 stellt die zeitliche Planung der Releases dar.

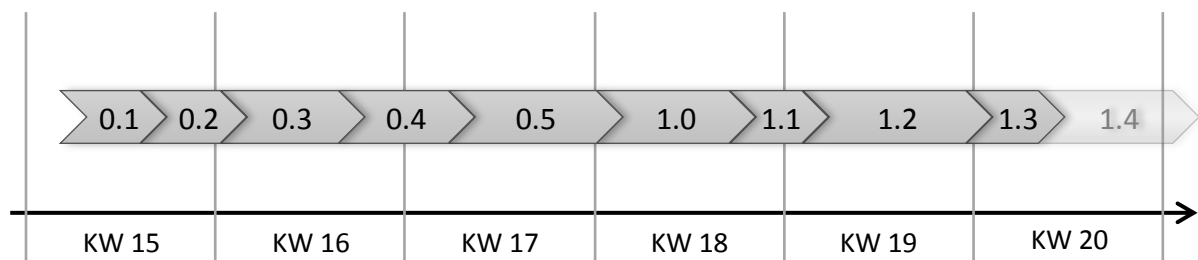


Abbildung 12 - Releaseplanung

3. Entwurf und Implementierung

Nach Abschluss der Anforderungsanalyse und Entwurf der Oberflächen wurde der strukturelle Aufbau der mobilen Applikation erörtert. Wie in der folgenden Abbildung 13 zu sehen ist, wurde die Applikation in zwei grundlegende Schichten unterteilt. Die Präsentations- und

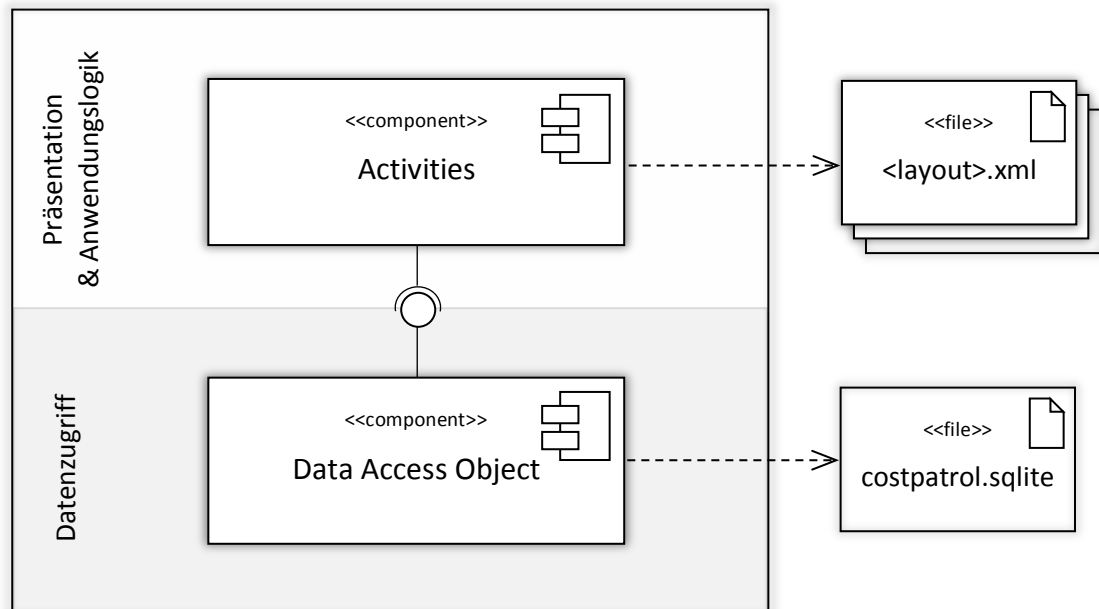


Abbildung 13 - Struktureller Aufbau der Applikation

Anwendungslogikschicht soll sich dabei um die Erzeugung der Oberflächenkomponenten kümmern, die Benutzeroberfläche darstellen, die Benutzeraktionen entgegennehmen und die jeweilige Funktion ausführen. Bei der Realisierung dieser Schicht wurde, entsprechend der empfohlenen Vorgehensweise der Android Developer-Webseite, auf eine konsequente Trennung von Ressourcen und Programmlogik geachtet.³ Die Definition und Konfiguration der Oberflächenkomponenten, deren Layout sowie die Definition von Zeichenketten, Farben und Größenangaben wurden daher durch den deklarativen Ansatz über XML-Dateien vorgenommen. Neben den XML-Dateien wird diese Schicht hauptsächlich durch Activity-Klassen realisiert, auf die im folgenden Unterkapitel genauer eingegangen wird.

Zur Erledigung der meisten Funktionen von „CostPatrol“ benötigen die Activities Zugriff auf Persistenzoperationen. Die Datenzugriffsschicht bietet daher Schnittstellen zur Speicherung und Bereitstellung der Domänenobjekte an. Eine genauere Beschreibung der Domänenobjekte folgt nach dem Activity-Unterkapitel. Für die Umsetzung der Datenzugriffsschicht ist das Entwurfsmuster der Data Access Objects (DAO) vorgesehen. Dieses sorgt einerseits für eine Entkopplung der eigentlichen Datenspeicherung von der Anwendung. Andererseits abstrahiert das Muster den Zugriff auf die zugrundeliegende Datenhaltungskomponente der Anwendung. Auf die konkrete Umsetzung wird im letzten Unterkapitel näher eingegangen.

³ <http://developer.android.com/guide/topics/resources/providing-resources.html>

3.1. Activities

Im Paket `de.hornickel.costpatrol.activity` sind die Activities enthalten, die jeweils einer Bildschirmansicht der mobilen Applikation entsprechen. Eine im Vorfeld durchgeführte Analyse zeigte eine Reihe von Gemeinsamkeiten zwischen den geplanten Activities auf, die beim Entwurf der Klassenstruktur in Form gemeinsamer Basisklassen berücksichtigt wurden.

Abbildung 14 zeigt ein vereinfachtes Klassendiagramm der Activities, welche die vier Listansichten realisieren. Gemeinsame Funktionen der `StatisticListActivity`, `ExpenseSeries-`

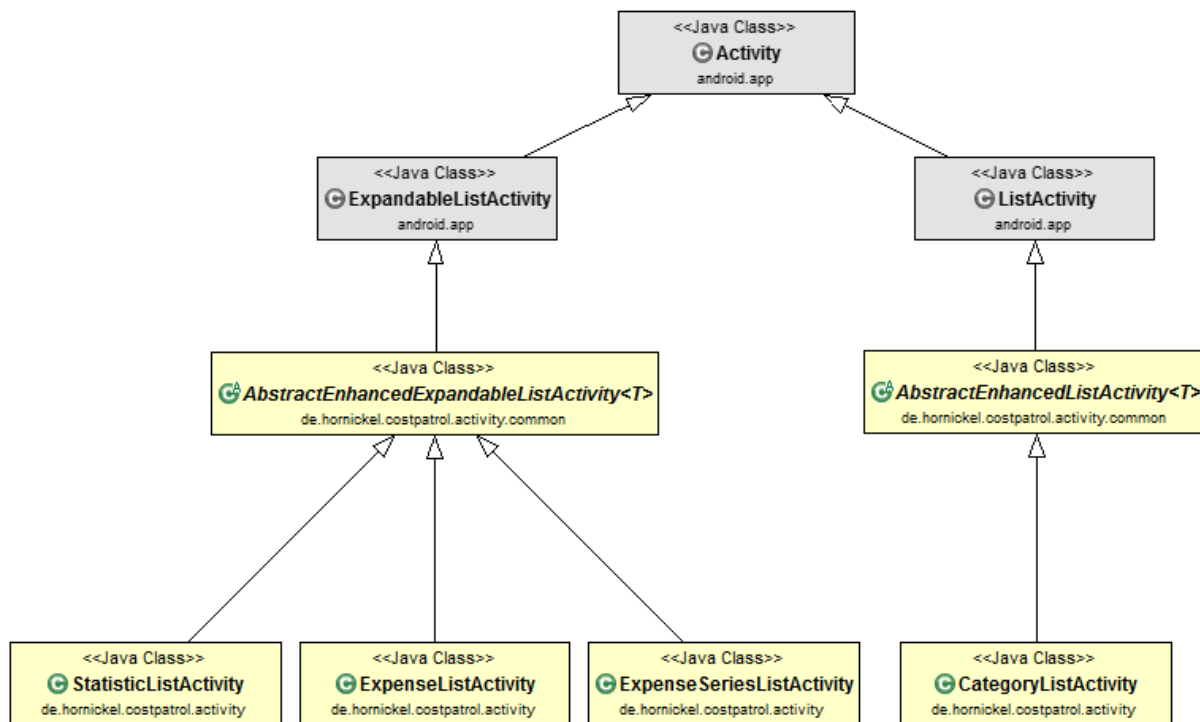


Abbildung 14 - Vereinfachtes Klassendiagramm der ListActivities

`ListActivity` und der `ExpenseListActivity` stellt deren abstrakten Superklasse `AbstractEnhancedExpandableListActivity` zur Verfügung. Die Klasse ist selbst eine Ableitung der `ExpandableListActivity` des Android-Frameworks. Im Wesentlichen erfüllt diese Klasse, die im Kapitel 3.1.1 beschriebenen übergreifenden Anforderungen für Listendarstellungen. Beispielsweise sorgt diese Klasse für eine einheitliche Anzeige der ProgressBar beim Ladevorgang der Daten. Wird festgestellt, dass keine Daten vorhanden sind, wird, wie spezifiziert, ein Hinweis ausgegeben. Um den Ladevorgang asynchron in Hintergrund-Threads auszuführen, implementiert diese Klasse die Callback-Methoden des `CursorLoaders`. Die Aktivität des `CursorLoaders` wird dann in den Lifecycle-Methoden der Activity gesteuert.

Neben den exemplarisch angeführten übergreifenden Anforderungen für Listen, realisiert die Klasse `AbstractEnhancedExpandableListActivity` auch den kontextabhängigen Modus der ActionBar zum Löschen von Listeneinträgen. Dies erfolgt über die Implementierung der Methoden des Interfaces `android.widget.AbsListView.MultiChoiceModeListener`.

Vergleichbare Aufgaben übernimmt auch die Klasse `AbstractEnhancedListActivity`, die wiederum von der Android-Klasse `ListActivity` abgeleitet ist.

Die vier konkreten Unterklassen realisieren im Wesentlichen, die spezifischen Eigenschaften der jeweiligen Listenansicht. Dies wird über die Implementierung der abstrakten Methoden ihrer Superklasse vorgenommen.

Beim Klassenentwurf der beiden Activities zum Anlegen und Ändern von Ausgaben beziehungsweise Ausgabenserien wurde, wie zuvor bei den Listenansichten, die Vermeidung von Quellcodeduplikation angestrebt. Hierzu wurde die gemeinsame abstrakte Basisklasse `AbstractEntryActivity` konzipiert, die wiederum, wie in Abbildung 15 zu sehen ist, Frag-

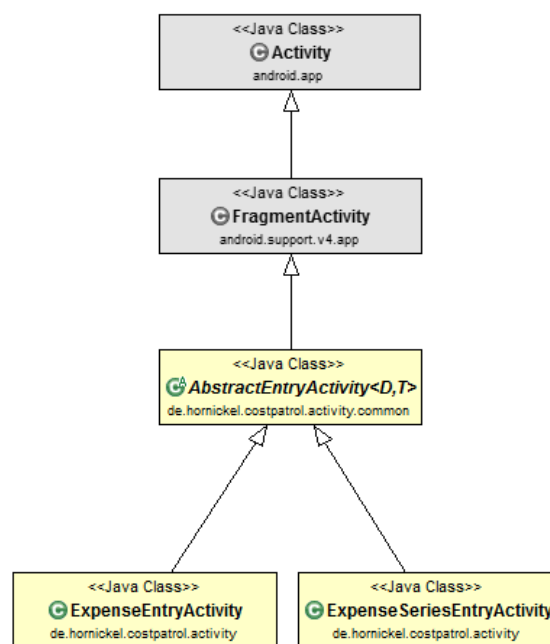


Abbildung 15 - Vereinfachtes Klassendiagramm der Activities zum Neuanlegen

mentActivity abgeleitet ist. Die Ableitung ist notwendig, da die Unterklassen `ExpenseEntryActivity` und `ExpenseSeriesEntryActivity` Funktionalitäten zur Darstellung des DatePickers als `DialogFragment` benötigen.

Wie auch bei den ListActivities übernimmt die abstrakte Basisklasse den Großteil der Funktionen. In den konkreten Unterklassen werden die spezifischen Eigenschaften der jeweiligen Oberfläche durch die Implementierung der abstrakten Methoden berücksichtigt.

3.2. Domänenobjekte

Ausgehend von den spezifizierten Anforderungen wurde ein Klassenmodell für die Domänenobjekte entwickelt (siehe Abbildung 16). Die Objekte sollen als einfache Java-Objekte (sogenannte Plain Old Java Object) umgesetzt werden. Eine Ausgabe wird dabei durch die Klasse `Expense` repräsentiert. Sie erbt ebenso wie die Klasse `ExpenseSeries` zur Abbildung von Ausgabenserien von der abstrakten Klassen `AbstractExpense`. Diese besitzt unter anderem ein Attribut vom dritten Domänenobjekt namens `Category`, das Kategorien

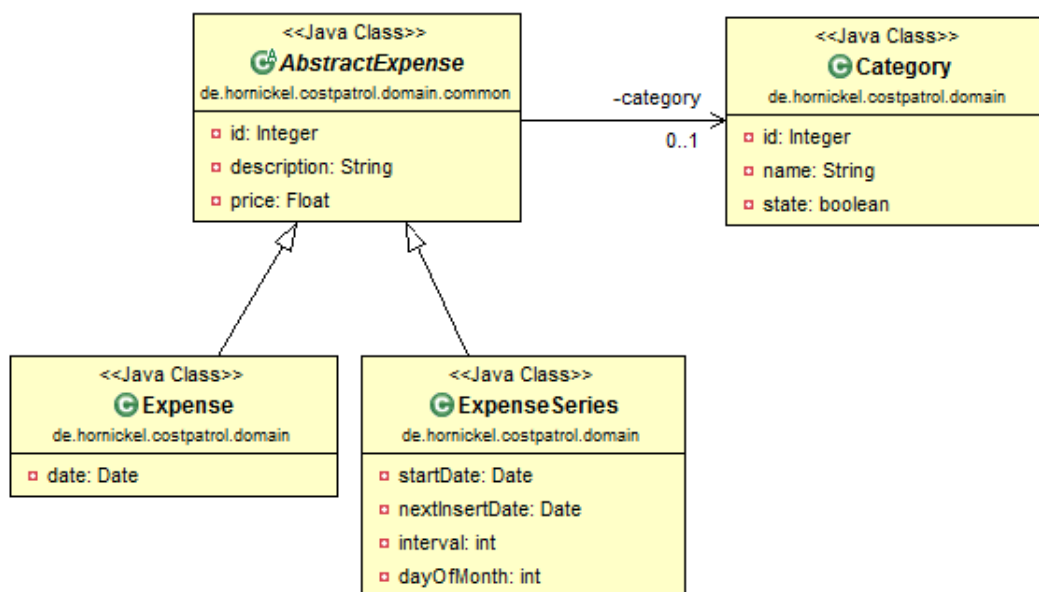


Abbildung 16 - Klassendiagramm des Domänenmodells

abbildet. Neben den im Modell aufgeführten Attributen, besitzen die Klassen außerdem Getter- und Setter-Methoden. Des Weiteren überschreiben die Klassen die `toString()`-Methode, um beispielsweise bei Bedarf in Log-Ausgaben deren Wertebelegung einsehen zu können.

3.3. Datenzugriffsschicht

Die geplante Datenzugriffsschicht bietet, wie aus deren Namen hervorgeht, Zugriff auf Daten der mobilen Applikation und kümmert sich um deren Speicherung in der SQLite-Datenbank. Für die Umsetzung ist wie bereits einleitend erwähnt das Entwurfsmuster der Data Access Objects (DAO) vorgesehen. Durch die Abstraktion des Zugriffs auf die zugrundeliegende Datenhaltungskomponente wird ermöglicht, Mock-Komponenten für Tests zu verwenden. Die folgende Abbildung 17 zeigt einen Ausschnitt des Klassendiagramms. Das Interface `IBaseDAO` definiert die Basisoperationen, die für jede davon abgeleitete Schnittstelle gelten. Dazu gehören die Methoden `save(T)` zum Speichern beim Anlegen oder nach einer Änderung, die Methode `getById(long)` zum Laden eines Objektes, mit der angegebenen ID, `delete(long)` zum Löschen, `getAll()` um alle Daten zu laden und `close()` zum Schließen der Datenbankverbindung.

Die Interfaces `ICategoryDAO`, `IExpenseSeriesDAO` und `IExpenseDAO` erben von `IBaseDAO` und definieren weitere benötigte Persistenzoperationen. Implementiert werden die Interfaces von den Klassen `CategoryDAOImpl`, `ExpenseSeriesDAOImpl` und `ExpenseDAOImpl` sowie deren Mock-Implementierungen, die aus Platzgründen nicht im Klassendiagramm dargestellt wurden.

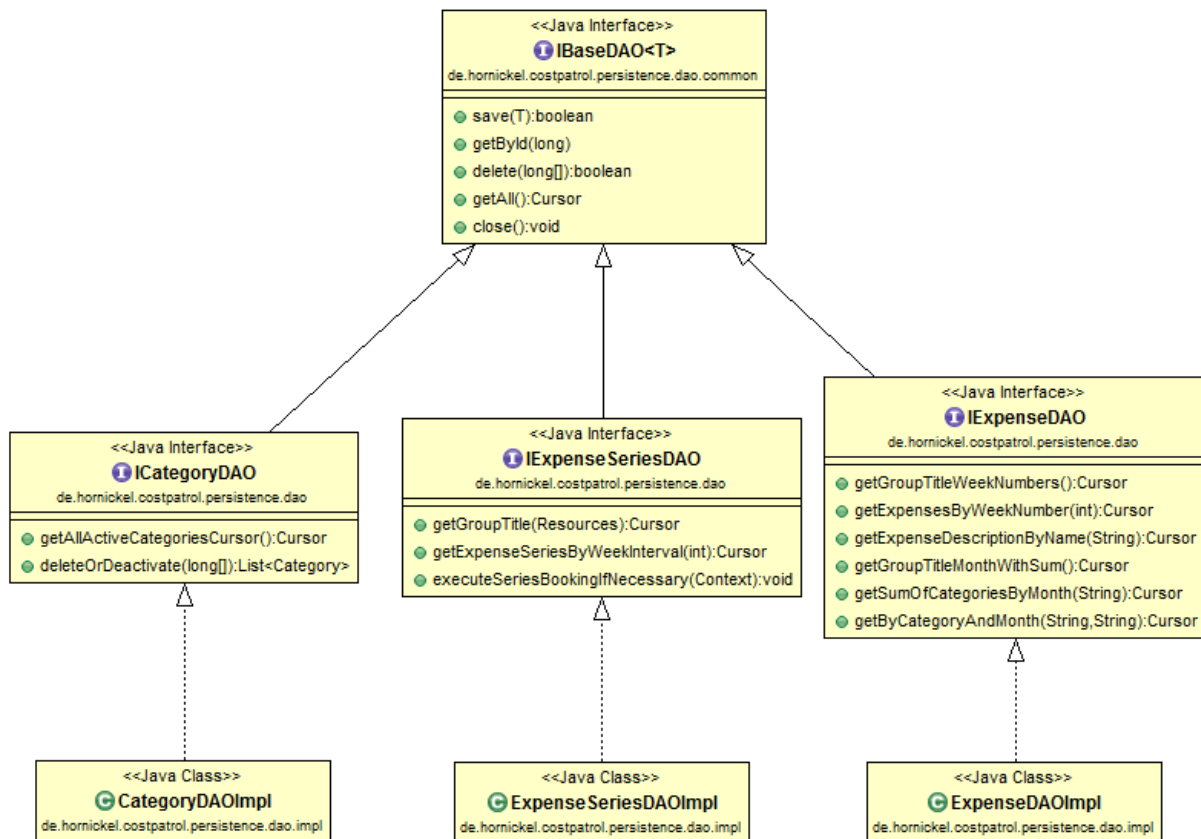


Abbildung 17 - Klassendiagramm des Domänenmodells

Der Zugriff der Activities auf die DAO-Objekte erfolgt über die generische Methode `createDAO()` der Klasse `DAOFactory`. Die Methode erzeugt die konkreten DAO-Implementierungen über Reflection anhand von Namenskonventionen und cached danach deren Instanzen. Dieser Ansatz ermöglicht die Hinzunahme von neuen DAOs ohne Anpassung der Erzeugungsmethode.

Außerdem hat die ausgelagerte Erzeugung der DAOs den Vorteil, dass die eigentlichen DAO-Implementierungen zur Laufzeit durch Mock-Implementierungen für automatisierte Tests ersetzt werden können.

4. Test

Zur Sicherstellung einer niedrigen Fehlerrate ist ein dreistufiges Testvorgehen geplant. Auf erster Stufe sind White-Box-Tests mit Hilfe von JUnit und dem Android Test Framework vorgesehen. Diese sollen nicht nur zur Verifikation einer gerade entwickelten oder geänderten Funktion dienen, sondern vor allem als Regressionstest eingesetzt werden. Bevor der Quellcode in die Versionsverwaltung überführt wird, soll ein sogenannter Smoke-Test in manueller Form durchgeführt werden, der die zweite Stufe repräsentiert. Hierbei soll mit geringem Aufwand sichergestellt werden, dass sich die grundlegenden Funktionen der App korrekt verhalten und die Navigation zwischen den wichtigsten Oberflächen gemäß der Spezifikation abläuft. Nach Fertigstellung jedes Releases sollen umfangreiche manuelle Tests folgen. Diese werden als Black-Box-Test durchgeführt und stellen die dritte Teststufe dar. In den folgenden Unterkapiteln wird nun für jede der drei Stufen das Vorgehen für einen Ausschnitt der Anwendungsfunktionalität detailliert beschrieben.

4.1. Stufe 1 - Junit-Tests

Im Rahmen der ersten Teststufe sollen einerseits klassische JUnit-Tests und andererseits Tests basierend auf dem Android Test Framework realisiert werden. Der Quellcode beider Testarten wird in einem eigenständigen Projekt namens `CostPatrolTest` in der Entwicklungsumgebung untergebracht. Die Paketstruktur und -bezeichnungen des Testprojekts richten sich nach den Konventionen der Android Developer-Webseite⁴. Demnach beginnen die Pakete des Testprojekts mit dem Paketnamen des Projektes der App, gefolgt von dem Suffix „.tests“.

Die JUnit-Tests konzentrieren sich auf die Sicherstellung von Funktionen, die durch reine Java-Sprachelemente realisiert wurden. Da JUnit in der Version 3 eingesetzt wird, muss jede Testklasse von `junit.framework.TestCase` erben. Jede Testmethode muss zudem mit dem Präfix „test“ beginnen. Beispiele für derartige Modultests stellen die folgenden Klassen dar, die in dem Testprojekt zu finden sind:

- `de.hornickel.costpatrol.util.tests.ValidationUtilsTest`
- `de.hornickel.costpatrol.util.tests.FormatterTest`
- `de.hornickel.costpatrol.domain.tests.ExpenseSeriesTest`

Während sich der `ValidationUtilsTest` und der `FormatterTest` auf einzelne triviale Funktionen der App konzentrieren, dient der `ExpenseSeriesTest` zur Verifikation eines Teils der Geschäftslogik. Der Test besteht aus 14 Testfällen und prüft die korrekte Berechnung der Datumswerte, an denen Ausgaben von Ausgabenserien automatisch gebucht werden. Jeder der Testfälle enthält eine kurze Beschreibung, eine Angabe der relevanten Parameter der Ausgangssituation und das erwartete Ergebnis in Form von Javadoc.

Die zweite Art der implementierten Tests fokussiert sich dagegen auf Funktionsabläufe der mobilen Applikation. Im Rahmen dessen sind Oberflächentests geplant, bei denen Eingaben simuliert werden. Anschließend wird geprüft, ob die Daten korrekt verarbeitet und an die Datenzugriffsschicht übergeben wurden. Zur Realisierung der Tests werden die Objekte, die

⁴ <http://developer.android.com/training/activity-testing/activity-basic-testing.html>

für die Persistenzaufgaben zuständig sind, ummantelt. Vor der Weiterleitung der Methodenaufrufe an die Originalobjekte werden die Parameter zwischengespeichert, so dass diese im Rahmen der automatisierten Tests überprüft werden können. Die folgenden beiden Tests der Eingabe-Activities verwenden diesen Mechanismus:

- `de.hornickel.costpatrol.activity.tests.ExpenseEntryActivityTest`
- `de.hornickel.costpatrol.activity.tests.ExpenseSeriesEntryActivityTest`

Zusätzlich soll auch die Datenzugriffsschicht automatisiert getestet werden. Es werden hierzu Domänenobjekte mit Testdaten erzeugt und über die Funktionalität der Datenzugriffsschicht gespeichert. Nach erneuten Laden der Daten werden sie mit den ursprünglichen Objekten verglichen. Beispiele für diese Tests werden durch folgende Klassen repräsentiert:

- `de.hornickel.costpatrol.persistence.dao.tests.ExpenseDAOImplTest`
- `de.hornickel.costpatrol.persistence.dao.tests.ExpenseSeriesDAOImplTest`
- `de.hornickel.costpatrol.persistence.dao.tests.CategoryDAOImplTest`

4.2. Stufe 2 - Smoke-Test

Nach jeder abgeschlossenen Quellcodeänderung soll ein Smoke-Test in manueller Form durchgeführt werden, dessen Inhalte unabhängig von dem Anwendungsbereich der Änderung sind. Die Durchführungszeit des Tests beträgt für geübte Benutzer etwa 4 Minuten. Der Test soll dabei die folgende Aktionen enthalten und wird als erfolgreich gewertet, wenn stets das beschriebene Soll-Verhalten des Systems eintritt:

Schritt	Aktion	Soll-Verhalten
1	Neuinstallation der App	App lässt sich fehlerfrei installieren
2	Start der App (im Anzeigemodus Portrait)	App startet und zeigt Hinweis, dass keine Daten vorhanden sind. Evtl. ist zuvor kurz die ProgressBar eingeblendet.
3	Klick auf das Icon in ActionBar zum Anlegen einer Ausgabe	Oberfläche zum Anlegen von Ausgaben öffnet sich. Aktuelles Tagesdatum erscheint im Datumsfeld. Erste Kategorie ist ausgewählt. Die unteren beiden Eingabefelder sind leer.
4	Eingabe von Testdaten: <ul style="list-style-type: none"> • Datum von Vortag • Zweite Kategorie in Auswahlliste • Kurzbeschreibung „Test“ • Preis „1.9“ 	Eingaben lassen sich vornehmen.
5	Änderung des Anzeigemodus auf Landscape. (Danach Rückkehr in Modus Portrait)	Alle Komponenten sind sichtbar und enthalten die eingegebenen Werte, unabhängig vom Anzeigemodus.
6	Speicherung der Ausgabe über einen Klick auf das ActionBar-Icon	Oberfläche der Startansicht erscheint. Liste enthält einen Eintrag bestehend aus der Kalenderwoche des gestrigen Tages, der aufgeklappt ist. Darunter wird zuvor erfasste Ausgabe mit den korrekten Werten dargestellt.

7	Änderung des Anzeigemodus auf Landscape. (Danach Rückkehr in Modus Portrait)	Liste der Ausgaben wird dargestellt, unabhängig vom Anzeigemodus. Dargestellte Inhalte entsprechen dem vorherigen Schritt.
8	Klick auf den Listeneintrag der Ausgabe	Oberfläche zum Ändern der Ausgabe erscheint (identisch mit Oberfläche zum Neuanlegen). Oberflächenkomponenten erhalten die im Schritt 4 eingegeben Daten.
9	Änderung des Preises auf „10.00“ und Speicherung	Änderung kann vorgenommen werden. Danach ist Startansicht zu sehen. Geänderter Preis wird angezeigt.
10	Öffnen der Ausgabenstatistik über das ActionBar-Icon	Ansicht der Ausgabenstatistik wird angezeigt. Liste enthält einen Eintrag bestehend aus dem Monat des gestrigen Tages und Summe „10,00 €“, der aufgeklappt ist. Darunter wird die in Schritt 4 ausgewählte Kategorie angezeigt. Summe der Kategorie entspricht ebenfalls „10,00 €“.
11	Änderung des Anzeigemodus auf Landscape. (Danach Rückkehr in Modus Portrait)	Liste entspricht den Angaben im vorherigen Schritt, unabhängig vom Anzeigemodus.
12	Klick auf Kategorie	Detaildialog öffnet sich. Titel des Dialoges gibt die Kategorie und den Preis korrekt wieder. Darunter wird ein Listeneintrag mit erfassten Daten korrekt angezeigt.
13	„OK“ klicken	Dialog schließt sich und die Ausgabenstatistik ist zu sehen.
14	Auswahl der Up-Navigation	Startansicht der App ist zu sehen.
15	Long-Klick auf den Listeneintrag der Ausgabe	ActionBar wechselt in den kontextabhängigen Modus. Links erscheint die Zahl „1“ und rechts ist ein Icon zur Ausführung der Löschung sichtbar. Die Hintergrundfarbe der selektierten Zeile hebt sich mit einem hellen Blauton ab.
16	Klick auf Lösch-Icon	Kontextabhängiger Modus der ActionBar wird ausgeblendet. Es erscheint ein Hinweis, dass keine Daten vorliegen.
17	Klick auf Aktion „Kategorie verwalten“ der ActionBar	Anzeige der Kategorien öffnet sich. Alle Kategorien haben den Status aktiv.
18	Änderung des Anzeigemodus auf Landscape. (Danach Rückkehr in Modus Portrait)	Liste entspricht den Angaben im vorherigen Schritt, unabhängig vom Anzeigemodus
19	Klick auf das Icon in ActionBar zum Neuanlegen einer Kategorie	Der Dialog zum Neuanlegen von Kategorien erscheint. Das Eingabefeld ist leer.
20	Eingabe der Bezeichnung „Test“	Eingabe lässt sich vornehmen.
21	Speicherung der Kategorie über einen Klick auf „OK“	Anzeige der Kategorien erscheint. Liste enthält ergänzend zu den vorherigen Einträgen, die erfasste Kategorie „Test“, deren Status aktiv ist.
22	Klick auf Kategorie	Dialog zum Ändern der Kategorien öffnet sich. Das

		Textfeld enthält den Wert „Test“. Switcher-Komponente für Status ist eingeblendet
23	Änderung des Status auf inaktiv und Speicherung durch Klick auf „OK“	Änderung lässt sich vornehmen. Nach dem Speichern ist die Anzeige der Kategorien zu sehen. Geänderte Kategorie hat nun den Status „inaktiv“.
24	Long-Klick auf den Listeneintrag der Kategorie „Test“	ActionBar wechselt in den kontextabhängigen Modus. Links erscheint die Zahl „1“ und rechts ist ein Icon zur Ausführung der Löschung sichtbar. Die Hintergrundfarbe der selektierten Zeile hebt sich mit einem hellen Blauton ab.
25	Klick auf Lösch-Icon	Kontextabhängiger Modus der ActionBar wird ausgeblendet. Anzeige der Kategorien ist zu sehen. Kategorie „Test“ ist nicht mehr in der Liste zu finden.
26	Betätigen der „Zurück“-Taste	Startansicht der App ist zu sehen.
27	Klick auf Aktion „Ausgabenserien verwalten“ der ActionBar	Anzeige der aufgelisteten Ausgabenserien erscheint. Hinweis, dass keine Daten vorhanden sind ist zu sehen.
28	Klick auf das Icon in ActionBar zum Anlegen einer Ausgabeserie	Oberfläche zum Anlegen von Ausgabenserien öffnet sich. RadioButton „Monatlich“ ist ausgewählt. Aktuelles Tagesdatum erscheint im Datumsfeld, daneben der aktuelle Tag im Monat. Erste Kategorie ist ausgewählt. Die unteren beiden Eingabefelder sind leer.
29	Eingabe von Testdaten: <ul style="list-style-type: none"> • Monatlich • Datum des Monatsersten des Vormonats • Erste Tag im Monat • Vierte Kategorie in Auswahlliste • Kurzbeschreibung „Test2“ • Preis „15“ 	Eingaben lassen sich vornehmen.
30	Speicherung der Ausgabe über einen Klick auf das ActionBar-Icon	Evtl. ist kurz ein Progress-Dialog eingeblendet. Anzeige der Ausgabeserien erscheint danach. Liste enthält Gruppe „Monatlich“, darunter einen Eintrag. Eintrag besteht aus Text „Nächste Buchung am <Datum>“, Kategorie, Beschreibung und Preis die im Schritt 29 eingegeben wurden. <Datum> ist der Monatserste des Folgemonats.
31	Auswahl der Up-Navigation	Startansicht der App ist zu sehen. Liste enthält durch automatische Buchung zwei Einträge. Der obere vom ersten Tag des aktuellen und der untere vom ersten Tag des Vormonats. Beide zeigen die Kategorie, die Beschreibung und den Preis, die im Schritt 29 eingegeben wurden korrekt an.

Tabelle 4 - Manuelle Schritte des Smoke-Tests

4.3. Stufe 3 - Manuelle Tests

Die manuellen Tests der dritten Teststufe grenzen sich durch die Detaillierung der Testfälle und dem daraus resultierenden Aufwand von den zuvor angeführten manuellen Smoke-Tests ab. Die Testfälle wurden von den spezifizierten Anwendungsfällen abgeleitet und so konzipiert, dass beim Test der Validierung alle Grenzwerte geprüft werden. Im Folgenden wird dieser Vorgang nun für die Funktion „Kategorie anlegen“ exemplarisch dargestellt. Die Vorbedingung der Testfälle ist jeweils der geöffnete Dialog zur Eingabe einer Kategorie.

Testfall ID	Beschreibung	Aktion	Soll-Verhalten
TF_01	Gültige Kategorie	1. Eingabe der Bezeichnung „Wohnung“ 2. Bestätigung mit „OK“	Kategorie wird gespeichert und erscheint danach in Kategorieliste
TF_02	Gültige Kategorie mit Mindestlänge (Grenzwertbetrachtung)	1. Eingabe der Bezeichnung „Bad“ 2. Bestätigung mit „OK“	Kategorie wird gespeichert und erscheint danach in Kategorieliste
TF_03	Gültige Kategorie mit Maximallänge (Grenzwertbetrachtung)	1. Eingabe der Bezeichnung „Arbeitsplatz-computer“ 2. Bestätigung mit „OK“	Kategorie wird gespeichert und erscheint danach in Kategorieliste
TF_04	Ungültige Kategorie , da ohne Eingabe	1. Bestätigung mit „OK“	Dialog bleibt geöffnet und es erscheint eine Validierungsmeldung mit dem Hinweis „Geben Sie eine Bezeichnung ein“. Nach Korrektur durch Eingabe von „Bad“ wird Kategorie gespeichert und erscheint danach in Kategorieliste.
TF_05	Ungültige Kategorie , da Eingabe zu kurz (Grenzwertbetrachtung)	1. Eingabe der Bezeichnung „Ba“ 2. Bestätigung mit „OK“	Dialog bleibt geöffnet und es erscheint eine Validierungsmeldung mit dem Hinweis „Geben Sie mindestens 3 Zeichen ein“. Nach Korrektur durch Ergänzung eines weiteren Zeichens wird Kategorie gespeichert und erscheint danach in Kategorieliste.
TF_06	Ungültige Kategorie , da Eingabe zu lang (Grenzwertbetrachtung)	1. Eingabe der Bezeichnung „Unterhaltungsprogramm“ 2. Bestätigung mit „OK“	Dialog bleibt geöffnet und es erscheint eine Validierungsmeldung mit dem Hinweis „Geben Sie höchstens 20 Zeichen ein“. Nach Korrektur durch Entfernen eines Zeichens wird Kategorie gespeichert und erscheint danach in Kategorieliste.
TF_07	Ungültige Kategorie , da nur Leerzeichen	1. Eingabe von drei Leerzeichen als Bezeichnung	Dialog bleibt geöffnet und es erscheint eine Validierungsmeldung mit dem Hinweis „Geben Sie eine

		2. Bestätigung mit „OK“	Bezeichnung ein“. Nach Korrektur durch Eingabe von „Bad“ wird Kategorie gespeichert und erscheint danach in Kategorieliste.
--	--	-------------------------	---

Tabelle 5 - Testfälle zur Funktion „Kategorie anlegen“