



Entwicklung einer Foto-App für Android

Studienarbeit

im Studiengang Angewandte Informatik
an der Dualen Hochschule Baden-Württemberg Mannheim

von

Tobias Dorra und Philipp Pütz

November 2015

Bearbeitungszeitraum
Kurs

4 Wochen
TINF14-AIBC

Inhaltsverzeichnis

Abkürzungsverzeichnis	i
Abbildungsverzeichnis	ii
1 Einführung	1
1.1 Funktionen	1
1.2 Projektumfang	2
2 Benutzeroberfläche	7
2.1 Beschreibung der Benutzeroberfläche	7
2.2 Verwendete Libraries	8

Abkürzungsverzeichnis

Abbildungsverzeichnis

2.1	App-Startansicht	7
2.2	Schematische Darstellung	7
2.3	Navigation Drawer	8
2.4	Schematische Darstellung	8

1 Einführung

Bei gemeinsamen Freizeitaktivitäten mit Freunden entstehen oft viele Fotos. Diese werden dann entweder gar nicht oder erst Wochen später miteinander geteilt.

Unser Ziel ist es, die Benutzererfahrung beim Teilen der Bilder zu verbessern. Das soll mit einer eigenen Foto-App passieren. Anstatt die gesammelten Bilder erst nach der Veranstaltung auszuwählen und dann an die entsprechenden Kontakte zu schicken, sollen die geschossenen Bilder bereits während der Veranstaltung automatisch mit den anderen anwesenden Freunden geteilt werden.

1.1 Funktionen

1.1.1 Organisation der Fotos

Die Fotos werden in sogenannten „Streams“ gruppiert. Jeder Stream repräsentiert dabei ein gemeinsames Erlebnis mit Freunden. Streams können umbenannt oder gelöscht werden. Beim Löschen eines Streams werden alle enthaltenen Bilder ebenfalls gelöscht.

Um einen Stream mit Fotos zu befüllen, kann entweder die Gerätekamera genutzt werden, oder es werden Fotos aus der Android-Galerie importiert. Die Fotos werden mit Name und Vorschaubild im Stream angezeigt. Außerdem existiert eine Detailansicht, in der man das Foto vergrößert betrachten kann. Fotos können umbenannt oder gelöscht werden. Ausgewählte Fotos können wieder zurück in die Android-Galerie exportiert, oder über die Teilen-Funktion anderen Apps zugänglich gemacht werden.

1.1.2 Teilen von Fotos

Zu einem Stream können mehrere Benutzer hinzugefügt werden. Die Fotos in einem Stream werden live zwischen allen Teilnehmern synchronisiert.

Alle Teilnehmer in einem Stream sind gleichberechtigt. Es gibt also keinen „Streamadministrator“ mit besonderen Rechten. Jeder Teilnehmer eines Streams kann Bilder hinzufügen, Bilder löschen und die Teilnehmerliste verwalten. Insbesondere ist jeder Teilnehmer auch dazu berechtigt, weitere Teilnehmer hinzuzufügen.

Entfernt ein Benutzer einen Stream, so bleibt der Stream bei den anderen Teilnehmern noch vorhanden. Bei diesen wird lediglich der Benutzer aus der Teilnehmerliste entfernt.

Wird ein Benutzer nachträglich zu einem Stream hinzugefügt, so erhält er nicht nur die neuen Bilder, sondern bekommt auch automatisch Zugriff auf alle Bilder, die bereits vor seinem Eintritt im Stream vorhanden waren.

1.1.3 Nutzung alternativer Methoden zur Datenübertragung

Wenn man gezwungen ist, das Mobilfunknetz zu benutzen, ist die Internetverbindung oft schlecht. Außerdem haben viele Smartphone-Nutzer ein begrenztes Datenvolumen. Daher sollen die Daten nicht über das Internet, sondern über alternative Verbindungen übertragen werden. Diese beinhalten:

1. Bluetooth
2. Bluetooth Low Energy (eventuell, falls technisch machbar)
3. Wi-Fi Direct

1.2 Projektumfang

1.2.1 Umgesetzte Funktionalitäten

Die Anwendung, die im Rahmen dieses Projektes entwickelt wurde, stellt lediglich einen Prototyp dar. Es werden also nicht alle beschriebenen Funktionen umgesetzt. Bei der Architektur wurde jedoch darauf geachtet, dass die Anwendung um die beschriebenen Funktionalitäten erweiterbar ist. Die folgenden Funktionalitäten wurden im Prototyp umgesetzt:

ID	Beschreibung
F-10	Bilder können zur App hinzugefügt werden
F-10.1	Bilder können über die Kamera aufgenommen werden

ID	Beschreibung
F-10.2	Bilder aus der Galerie geladen werden
F-20	Bilder können verwaltet werden
F-20.1	Bilder können gelöscht und umbenannt werden
F-20.2	Bilder haben verschiedene Attribute (Bildtitel, Aufnahmeinformationen)
F-30	„Streams“ (Ordner) gruppieren Bilder
F-30.1	Streams können erstellt, gelöscht und umbenannt werden
F-40	Bilder werden in einer listen ähnlichen Darstellung mit Vorschaubild und Bildinformationen angezeigt
F-50	Bilder können in einer Vollbildansicht angezeigt werden
F-50.1	Aus der Vollbildansicht können Bilder in die Galerie exportiert und per E-Mail bzw. zu teilen
F-60	Die Architektur ist erweiterbar
D-10	Alle Appdaten (Streams, Bilder, Datenbanken) werden in einen speziellen Verzeichnis gespeichert
D-10	Gespeicherte Bilder erhalten als Zusatz das aktuelle Datum
D-20	Alle Bild- und Streaminformationen werden in einer Datenbank gespeichert
UI-10	Es existiert eine Hauptansicht
UI-10.1	Die Hauptansicht zeigt Vorschaubilder von aufgenommenen und aus der Galerie geladenen Bildern an
UI-10.2	Zu jedem Bild werden in der Hauptansicht der Bildtitel und Aufnahmeinformationen angezeigt
UI-10.3	Durch antippen der Bilder startet eine neue Activity die das Bild in einer Vollbildansicht anzeigt
UI-10.4	Durch einen „long tap“ auf ein Bild wird ein Kontext Menü angezeigt
UI-10.4.1	Im Kontext Menü stehen folgende Optionen zur Verfügung: Bild umbenennen, Bild löschen, Bild in Galerie exportieren

ID	Beschreibung
UI-10.5	Ein Floating-Action Button Menü bietet weitere Optionen
UI-10.5.1	Über das Menü können Bilder mit der Kamera aufgenommen werden
UI-10.5.2	Über das Menü können Bilder aus der Galerie geladen werden
UI-10.6	Ein „Navigation Drawer“ bietet die Möglichkeit neue „Streams“ anzulegen, die ähnlich wie Ordner, Bilder gruppieren
UI-10.6.1	Die Hauptansicht zeigt den Inhalt eines Streams an
UI-10.7	Über das Hauptmenü können Streams umbenannt werden
UI-10.8	Über das Hauptmenü können Streams und alle zugehörigen Bilder gelöscht werden
UI-10.9	Eine Standardansicht animiert den Nutzer zum anlegen eines Streams, wenn kein Stream angelegt ist
UI-20	Es existiert eine Vollbildansicht für Bilder
UI-20.1	Die Vollbildansicht wird durch an tippen eines Bildes in der Hauptansicht geöffnet
UI-20.2	Es ist möglich ein Bild aus der Vollbildansicht in die Galerie zu exportieren
UI-20.3	Es ist möglich ein Bild aus der Vollbildansicht zu teilen (Versand per E-Mail falls E-Mail Client vorhanden)

1.2.2 Offene Funktionalitäten

Die im folgenden aufgelisteten Funktionalitäten wurden bisher nicht implementiert und werden für weitere Versionen vorgemerkt.

ID	Beschreibung
F-70	Es können Benutzer zu Streams hinzugefügt werden

ID	Beschreibung
F-70.1	Neue Benutzer erhalten Zugriff auf vorhandene und zukünftige Bilder im Stream
F-80	Bilder werden zwischen den verschiedenen Nutzern live synchronisiert
F-90	Es existiert ein Art Synchronisationslog der die unterschiedlichen Aktivitäten der Benutzer untereinander abgleicht
F-100	Benutzer können aus Streams austreten
F-100.1	Tritt ein Benutzer aus einem Stream aus, so werden lokal alle Bilder des Streams und der Stream selbst gelöscht
F-100.2	Ausgetretene Benutzer werden bei anderen Benutzern aus der Streamteilnehmerliste entfernt
F-110	Es stehen verschiedene Synchronisationsmöglichkeiten zur Verfügung:
F-110.1	Bluetooth
F-110.2	Bluetooth Low Energy (falls verfügbar)
F-110.3	Wi-Fi Direct
F-120	Alle Streamteilnehmer können über eine Teilnehmerliste eingesehen werden
F-130	In der Vollbildansicht kann an Bilder herangezoomt werden
F-130	Durch Gesten kann in der Vollbildansicht zum nächsten oder vorherigen Bild gewechselt werden
F-140	Das gesamte UI ist für verschiedene Endgeräte (Tablets, Android TVs) optimiert
D-10	Der Synchronisationslog und die Teilnehmerlisten werden in der Datenbank gespeichert
UI-30	In der Hauptansicht eines Streams ermöglicht ein Menübutton die Teilnehmerliste einzusehen und Teilnehmer hinzuzufügen
UI-40	Der Menüpunkt „Einstellungen“ im Navigation Drawer ermöglicht die Synchronisationsmethode und allgemeine Synchronisationseinstellungen festzulegen

2 Benutzeroberfläche

2.1 Beschreibung der Benutzeroberfläche

2.1.1 Appstart

Nach dem erstmaligen Start der App durch anklicken des App Icons gelangt der Benutzer zur in Abbildung 2.1 gezeigten Benutzeroberfläche.

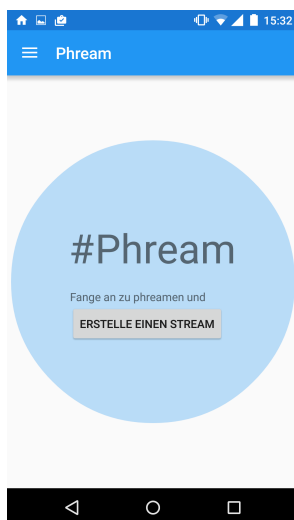


Abbildung 2.1: App-Startansicht



Abbildung 2.2: Schematische Darstellung

Die in Abbildung 2.2 eingezeichnete Statusbar ist standardmäßig in jeder Ansicht der App zusehen und zeigt allgemeine Statusinformationen zum Androidgerät an. Die Action Bar wird von der App selbst gestaltet und beinhaltet verschiedene Optionen, wie beispielsweise Menüs. Der größte Teil der App nimmt das Fragment in Anspruch. In diesem werden die verschiedenen Informationen der App dargestellt. Das Fragment ist dabei eine Art selbständige „Sub-Activity“, die einen eigenen Lifecycle, inklusive eigener Action Listener, hat. Das Fragment wird dynamisch durch einen Manager bei Benutzeraktionen ausgetauscht.

In der aktuell angezeigten Startansicht (Abbildung 2.1), die beim erstmaligen Start oder wenn Stream (Ordner) angelegt angezeigt wird, beinhaltet das Fragment den dargestellten Text, den Button zum anlegen eines Streams und den im Hintergrund platzierten Kreis. Durch anklicken des Buttons, öffnet sich ein Eingabefenster, in dem ein neuer Streamname eingegeben werden kann. Nach Bestätigung des Streamnamens wird der Stream angelegt und erscheint daraufhin in Navigation Drawer.

2.1.2 Navigation Drawer

Der Navigation Drawer (Abbildung 2.3) kann durch klicken auf das Menüicon oben links neben dem Appnamen oder durch einen Swipe vom linken Bildschirmrand geöffnet werden. In der daraufhin eingeblendeten Ansicht hat der Benutzer die Möglichkeit zwischen verschiedenen Streams zu wechseln und neue Streams anzulegen. Durch anklicken eines Streamnamens wird dieser aktiv und wird daraufhin in der Hauptansicht angezeigt. Durch einen swipe zum linken Bildschirmrand oder durch anklicken des Pfeils links neben dem Appnamen kann der Navigation Drawer geschlossen werden.

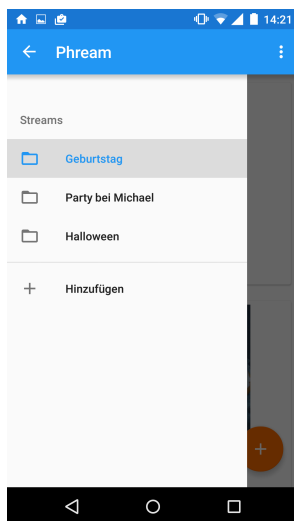


Abbildung 2.3: Navigation Drawer



Abbildung 2.4: Schematische Darstellung

In der schematischen Darstellung (Abbildung 2.4) ist zu sehen, dass der Navigation Drawer das aktuelle Fragment „überdeckt“ und den restlichen Teil des Fragments abdunkelt. In der Action Bar wird dabei der Button zum öffnen des Navigation Drawer zu einem Pfeil.

2.2 Verwendete Libraries