

Flashcards App



Technische Berufsschule Zürich TBZ

Modul: 335 Apps für Mobile Devices entwickeln

Projekt: Flashcards App - Dokumentation

Gruppe: Davina Golomingi, Fabienne Pfister, Victoria Meyer

Letzte Änderung: 19.06.2014

Inhaltverzeichnis

1.	Einleitung	3
1.1	Vorwort	3
2	Problembeschreibung	3
2.1	Use-Cases-Diagramm	4
3	Vorgehen (Methodik)	4
4	Architektur-Beschreibung	5
4.1	Story-Board	5
4.2	Lokal Ordner-Struktur	7
4.3	UML-Diagramm	8
5	Testbericht	9
5.1	Systemtest Testfall	9
6	Prozess und Fazit	10
6.1	Lokale Ordner-Struktur	10
6.2	Fazit der „Background Implementierung“	10
6.3	Optionen, Fotofunktionen	10
6.4	Fazit der Optionen-Fotofunktionen Implementierung	10
6.5	Allgemeiner Lernfazit	11

2. Einleitung

1.1 Vorwort

Da wir Lernende der Technischen Berufsschule Zürich (TBZ) uns generell für Android Apps Entwicklung interessieren, waren wir relativ schnell dazu entschlossen, in der TBZ-Studienwoche das Modul „335 Apps für Mobile Devices entwickeln“ zu besuchen.

Im Rahmen dieses Moduls wurde diese Dokumentation für die Android Flashcards App erstellt. Die App wurde selbständig während der TBZ-Studienwoche von uns entwickelt und dokumentiert. Der Inhalt dieser Dokumentation und die App werden am Schluss der Studienwoche präsentiert.

2 Problembeschreibung

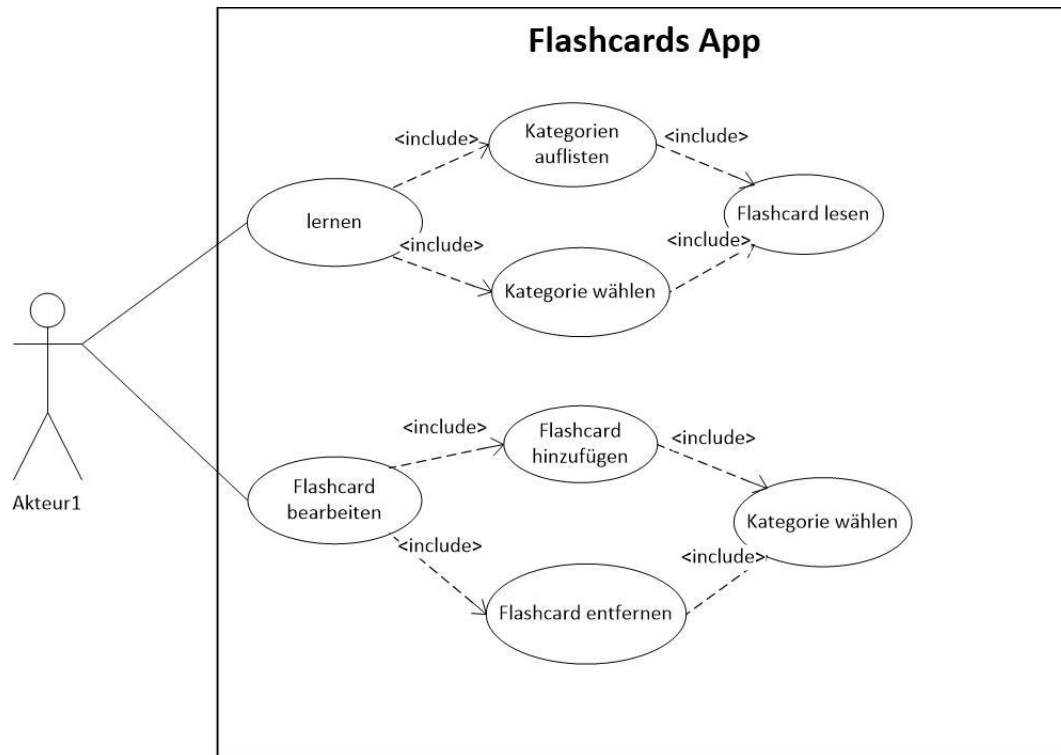
Karteikarten sind eine tolle Sache und als Hilfsmittel fürs Lernen dürften sie dem Großteil unserer Leserschaft bekannt sein. Leider ist das Konzept der Karteikarten besser gelungen als die Umsetzung. Der Gedanke dieses Karteikartenkonzepts mit dem ständigen Begleiter "Smartphone" zu verbinden ist nicht neu, aber sehr reichhaltig. Aus diesem Grund haben wir uns entschieden, eine Flashcards App zu entwickeln

Mittels Flashcards App sollen eingegebene Karteikarten einfach unterwegs auf dem Smartphone gelernt werden können. Man kennt dies vor allem vom Vokabeln lernen bei Fremdsprachen.

Die App läuft mit Android. Ebenfalls kann man die dazu gehörende Homeseite programmieren. Über die Homepage sollen die Karteikarten eingegeben und entfernt werden können. Die Karteikarten werden lokal in Kategorien-Verzeichnisse gespeichert. Mittels Knopfdruck soll es möglich sein, auf diese Kategorien-Verzeichnisse von der Homepage aus zuzugreifen, um die vorhandenen Karteikarten zu studieren.

Im Unterschied zur herkömmlichen Methode und bereits vorhandenen Apps werden die Karten nicht nur im Text gespeichert sondern es werden auch Bilder zu diesem Text abgelegt. Da viele Menschen visuell lernen, ist es von Vorteil, wenn sie auch mit Bildern lernen können.

2.1 Use-Cases-Diagramm



3 Vorgehen (Methodik)

Zuerst wollten wir eine grosse Applikation mit mehreren Seiten machen. Leider bemerkten wir schnell, dass dies in dieser kurzen Zeit kaum möglich ist. Hinzu kamen Unstimmigkeiten im Team, da wir überfordert waren. Als Hilfe haben wir nun Tutorials aus dem Internet verwendet.

Zusätzlich haben wir in der Gruppe wenige Erfahrungen im Android Programmieren und es besteht auch Zeitdruck. Aus diesen Gründen ist unsere Applikation kleiner geworden. Nun haben wir uns entschieden, mit dem Buch „Android 4 Apps entwickeln mit dem Android SDK“ zu arbeiten. Im Buch haben wir gute Beispiele gefunden, um die Flashcards App zu erstellen und diese gegebenenfalls auszubauen. Durch diese „Vereinfachung“ ist es uns möglich, in dieser kurzen Zeit ein realistischeres Projekt zu erarbeiten. Ebenfalls ist es uns möglich, qualitativ bessere Arbeit zu präsentieren.

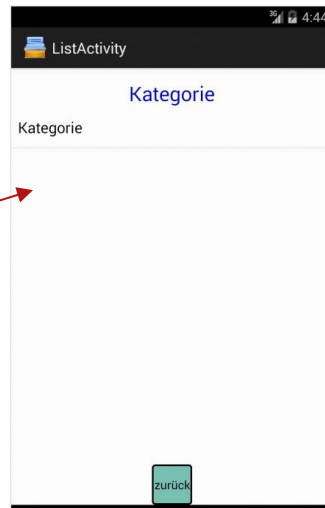
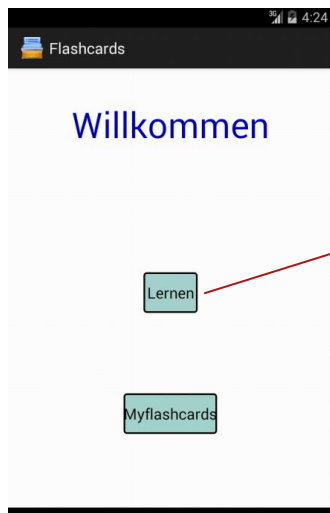
Das Testen der App wird über ein Android Handy durchgeführt und nicht über einen Emulator. Dies hat einige Vorteile, denn Emulatoren sind langsam und verursachen auch weitere Probleme. Zusätzlich muss ein Treiber installiert werden.

4 Architektur

Unsere Applikation beinhaltet vier Navigation-Seiten, die sind mit einander verbunden sind.

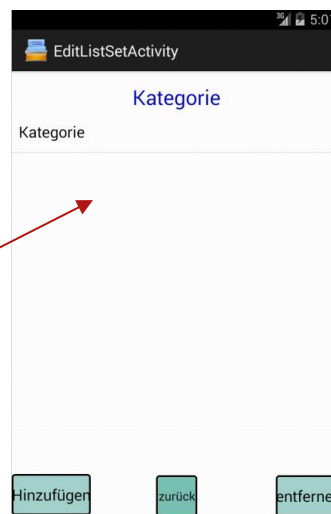
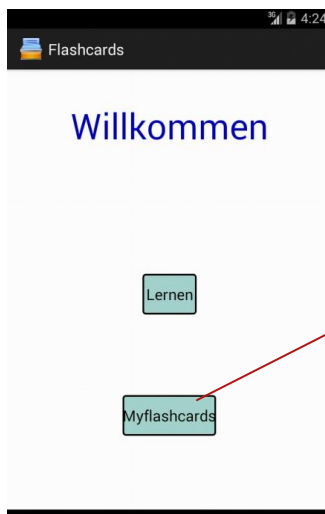
4.1 Story-Board

UC1 und UC2

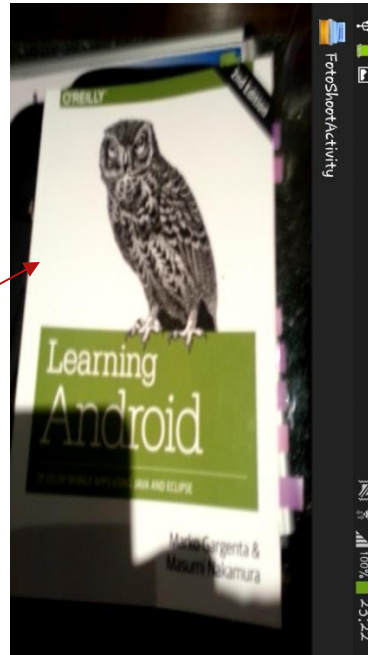
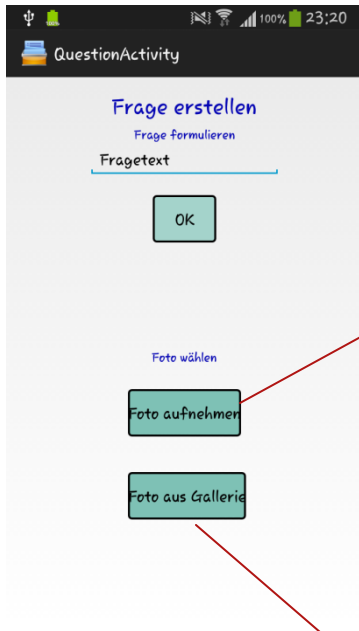


Beim drücken auf den Knopf lernen wird man direkte auf eine Liste der Kategorien weitergeleitet.

UC3 und UC4

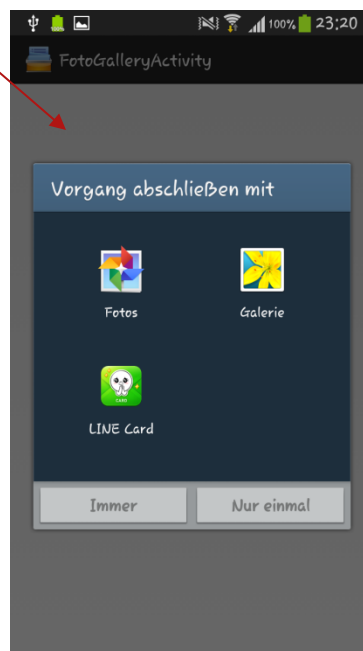


Beim anwählen vom Button „MyFlashcards“, erscheint die Liste der Kategorien. Dies ist die Übersicht der zu bearbeitenden Kategorien. Unten befinden sich 2 weitere Buttons, mit denen man eine Kategorie hinzufügen oder entfernen kann.



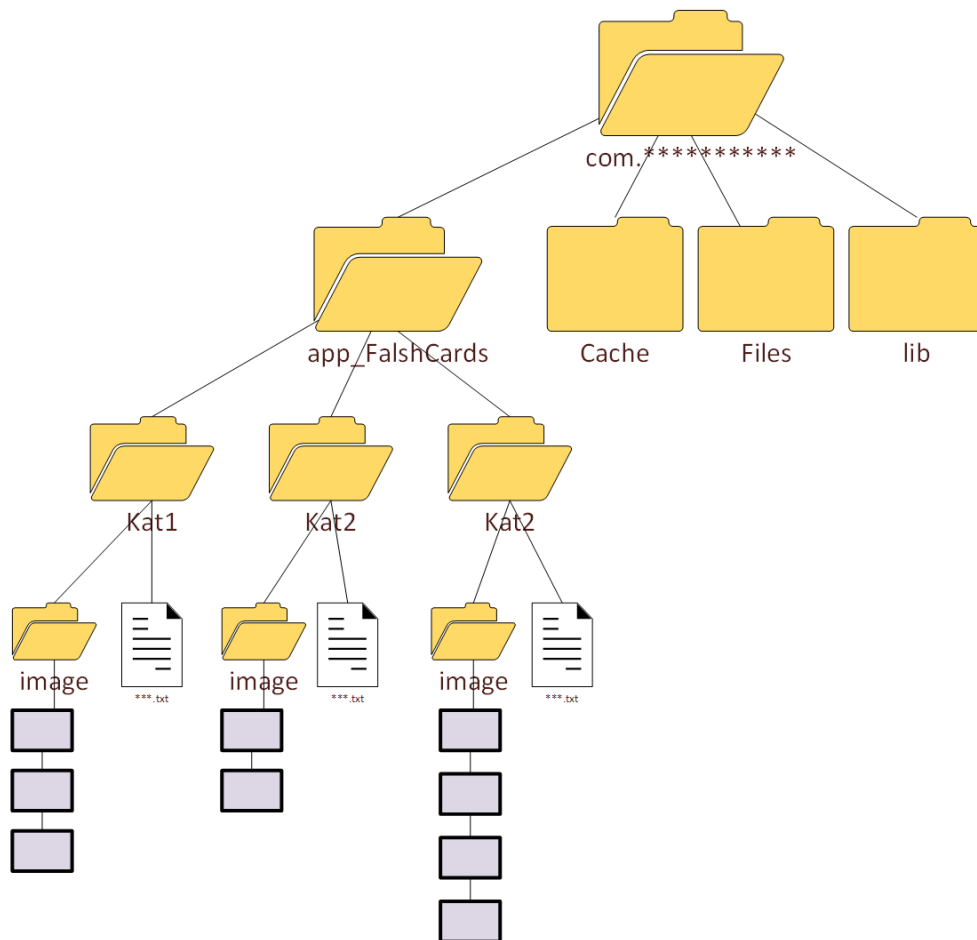
Wählt man „Foto aufnehmen“, so wird die Kamera aktiviert und man kann mit Tippen ein Foto schießen.

[Oben]
Wenn man bei der Liste von „MyFlashcards“ auf hinzufügen drückt, kommt man auf diese Ansicht. es ist möglich eine Frage als Text, oder als Foto einzugeben. Das Foto kann entweder direkt aufgenommen werden, oder aus der Galerie geladen werden.



Wird „Foto aus Galerie“ gewählt, kann man eine App mit Galerie-Funktion zur Anzeige der Fotos wählen um dort das Foto auszuwählen.

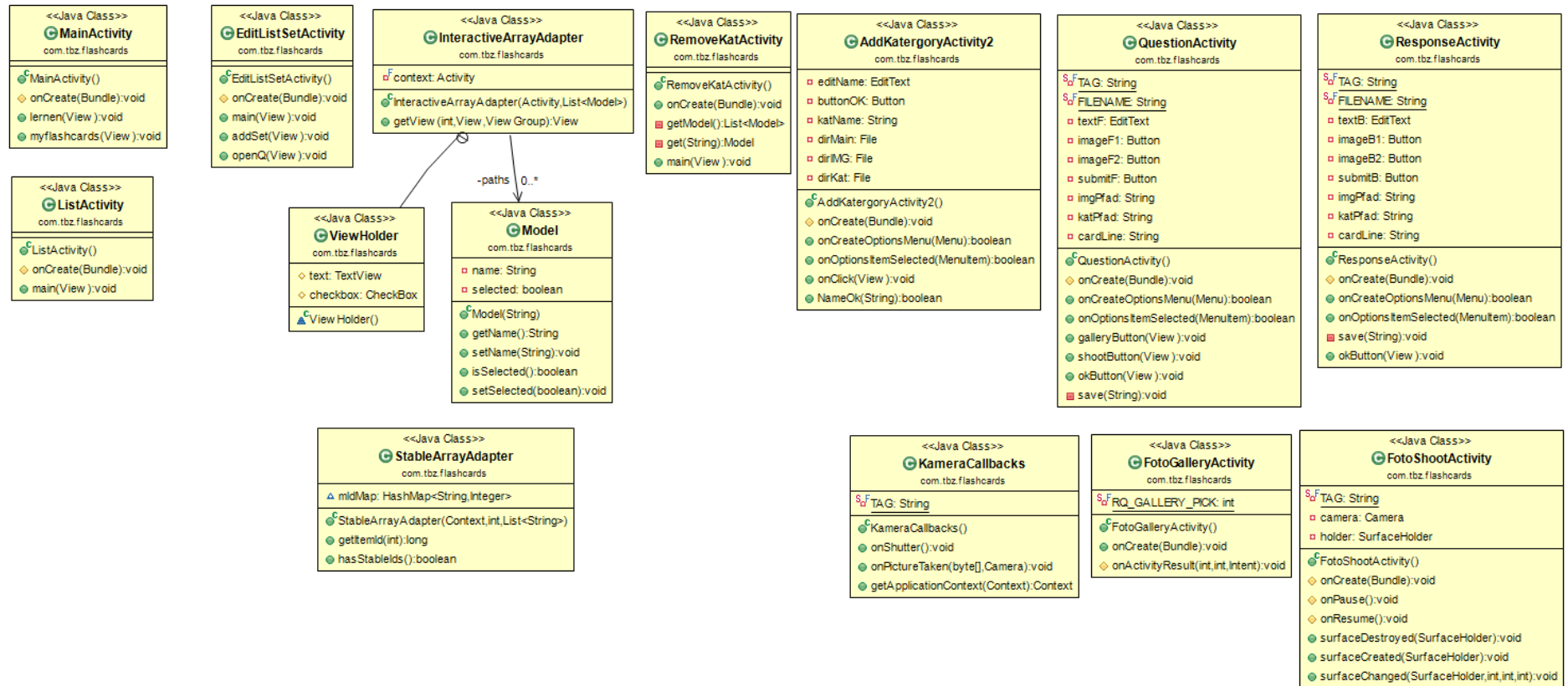
4.2 Lokal Ordner-Struktur



Wir haben uns für eine lokale Speicherung der Flashcards entschieden, weil es somit für den Benutzer einfach ist seinen Kategorienordner mit anderen zu teilen (man braucht nur den betroffenen Ordner zu kopieren und die App installiert zu haben). Man braucht für das Teilen der Flashcards somit keine Internetverbindung.

Bei der Implementierung der App konnte man die Erstellung der Ordner und Ihre dazugehörigen Files mit dem File Explorer verfolgen. Auf dem Handy konnten wir leider nicht ermitteln, wo sich der com-Ordner unserer Applikation auf dem lokalen Speicher befindet. Wir konnten uns aber vergewissern, dass die Speicherung auch auf dem Handy funktioniert, denn alle erstellten Ordner werden richtig angezeigt.

4.3 UML-Diagramm



5 Testbericht

Beim Testen können sowohl die Funktionalität des Programms als auch nichtfunktionale Eigenschaften, wie etwas das Verhalten unter großer Last, überprüft werden. Tests können zufällig ohne Vorbereitung, aber auch sorgfältig vorbereitet und strukturiert durchgeführt werden. Aus diesem Grund beschäftigen sich verschiedene Teststrategien/Konzepte mit der Frage, wie mit einer möglichst geringen Anzahl von Testfällen eine möglichst große Testabdeckung zu erreichen ist. Wir haben uns für den **Systemtest Testfall** entschieden.

5.1 Systemtest Testfall

Step	Aktivität	Resultat im System	Resultat für Benutzer
1	Die Applikation wurde gestartet , Klick auf „Lernen“	Eine Liste aller vorhandenen Kategorien wird bereitgestellt	Anzeige aller Kategorien
2	Eine Kategorie wird ausgewählt	Sucht die Flashcards dieser Kategorie	Anzeige aller Flashcards
3	Eine Flashcard wird ausgewählt	Holt eine bestimmte Flashcard dieser Kategorie ab	Anzeige einer Flashcard
4	Klick auf „zurück“	Die vorherige Ansicht wird aufgerufen	Die vorletzte View ist angezeigt
5	Klick auf „Myflashcards“	Eine Liste aller vorhandenen Kategorien und weitere Buttons (hinzufügen, zurück, entfernen) werden bereitgestellt	Erscheint die Liste der Kategorien. Dies ist die Übersicht der zu bearbeitenden Kategorien.
6	Klick auf „Hinzufügen“	Die QuestionActivity wird aufgerufen und man kann Text eingeben. Es werden weitere Buttons (Foto ausnehmen, Foto aus Gallery) angezeigt	Erscheint eine Seite mit einem Textfeld, in dem man den Text eingeben kann und die Buttons (Foto ausnehmen, Foto aus Gallery)
7	Klick auf „Foto ausnehmen“	Die Kamera wird aufgerufen.	Die Kamera wird aktiviert und man kann mit Tippen ein Foto schießen.
9	„Foto aus Gallery“ wird geklickt	Eine App mit Galerie-Funktion zur Anzeige der Fotos wird aufgerufen	Erscheint ein Fenster, in dem man die Gallery auswählen kann.
10	Die Gallery wird ausgewählt	Holt alle Fotos der Gallery ab	Alle Fotos der Gallery werden angezeigt
11	Die gewünschte Foto wird ausgewählt.	Holt ein bestimmtes Foto ab	Man kann das Foto sehen aber nicht hochladen

6 Prozess und Fazit

6.1 Lokale Ordner-Struktur

Wir haben uns für eine lokale Speicherung der Flashcards entschieden, weil es somit für den Benutzer einfach ist seinen Kategorienordner mit anderen zu teilen (man braucht nur den betroffenen Ordner zu kopieren und die App installiert zu haben). Man braucht für das Teilen der Flashcards somit keine Internetverbindung.

Bei der Implementierung der App konnte man die Erstellung der Ordner und Ihre dazugehörigen Files mit dem File Explorer verfolgen. Auf dem Handy konnten wir leider nicht ermitteln, wo sich der com-Ordner unserer Applikation auf dem lokalen Speicher befindet. Wir konnten uns aber vergewissern, dass die Speicherung auch auf dem Handy funktioniert, denn alle erstellten Ordner werden richtig angezeigt.

6.2 Fazit der „Background Implementierung“

Wir haben viel Zeit für das Design der inneren Architektur der Applikation investiert. Unsere Idee, dass das Grundgerüst möglichst logisch und simpel sein sollte, konnten wir umsetzen. Die Grundlage für den weiteren Aufbau steht demzufolge fest. Leider blieb und genau für den weiteren Aufbau nicht genug Zeit.

Unserer Meinung nach, war es wichtig, dass das Grundgerüst steht, bevor wir weitermachen. Niemand konnte sich am Anfang vorstellen wie zeitintensiv das Ganze werden sollte. Mit unseren Java-Vorkenntnissen waren wir am Anfang zuversichtlich, dass wir alles in der gegebenen Zeit erreichen würden. Wir haben unsere Ziele auf das Minimum beschränkt, trotzdem haben wir leider nicht alle erreicht. Dafür haben wir viel gelernt und etwas erreicht, was es in dieser Form noch nicht gab. (Das ist auch ein Grund, weshalb wir mehr Zeit brauchten, es gab wenig Dokumentation zu unserem Thema im Androidbereich.)

Ein weiterer Grund war, dass niemand von bereits existierendem Code duplizieren wollte, oder eine bereits existierende App nach Anleitung implementieren wollte. Im Nachhinein war das taktisch ungeschickt, wir hätten lieber ein Thema gewählt, zudem es zumindest mehr Dokumentation dazu gab, oder was wir schon in einer anderen Form im Voraus gemacht hatten (wie z.B. den Taschenrechner, den wir im Java-Kurs implementieren mussten). Dafür haben wir alles von Null her aufgebaut und alles selber kodiert, was ein grosser Vorteil für das Verständnis der Android-App-Entwicklung ist.

6.3 Optionen, Fotofunktionen

Es sind diverse Buttons verfügbar, welche eine einfache Usability garantieren. Man kann einfach zwischen den Menüs navigieren. Auf das Design wurde ebenfalls viel Wert gelegt. Der „zurück“-Button wurde jeweils mit einer Farbe eingefärbt, die anderen Funktionen einheitlich mit einer anderen.

Die Fotofunktionen sind dazu da, ein Foto auszuwählen und lokal abzuspeichern. Man kann ein Foto direkt aus der Galerie wählen, oder direkt eines schießen.

6.4 Fazit der Optionen-Fotofunktionen Implementierung

Die Buttons funktionieren alle, ausser der „Remove“-Button, welcher die entsprechende Activity noch nicht voll funktionsfähig ist.

Beim erstellen der Frage gibt es die Funktionen Text eingeben, Foto aus Galerie wählen oder Foto schießen. Alle diese Funktionen sind grundsätzlich durchführbar, allerdings konnten wir aufgrund des nicht Erreichens des Speicherorts den eingegebenen Text nicht weiterverarbeiten. Genauso wenn man ein Foto aus der Galerie wählt, wird es schlussendlich nicht abgespeichert. Das Foto kann man auch schießen, es wäre möglich das Bild auf der SD-Karte zu speichern, aber nicht am gewollten Speicherort. Deshalb ist keine Weiterverarbeitung möglich. Ziel wäre gewesen ein geschossenes Foto (original) auf der SD-Karte zur einfachen Wiederverwendung abzulegen und eine Version für die Karte, jene in Grösse reduziert. Die genau gleiche Funktion sollte dann ebenfalls für die Antwort verfügbar sein.

6.5 Allgemeiner Lernfazit

Das Projekt an sich war sehr spannend. Wir haben sehr viele Stunden für dieses Projekt investiert und sehr viel dabei gelernt, das überdeckt die Frustration, dass unsere App nicht fertig wurde, ein wenig.

Wir haben den Fokus darauf gelegt, die App selbst zu erstellen und nicht einfach sinnlos Code zu kopieren. Dies zog einen ausführlichen Lernprozess mit sich. Wir haben unsere Java-Kenntnisse mit Android-Spezifischen Eigenschaften erweitert. Die Architektur selbst ist schon ziemlich verschiedenen, was aber auch spannend zu erarbeiten war. Genau da lag auch eines der Probleme, wir mussten stets Dokumentationen und Lehrmittel erarbeiten, bis wir wieder einen Schritt vorwärts kamen.

Wir haben eine ausführliche Dokumentation erstellt. Sie umfasst alle erstellten Komponenten sowie fehlende Teilprojekte. Auf den geschriebenen Code wurde besonders viel Wert gelegt, mit vielen Kommentaren, die Standards eingehalten. Er ist ohne Duplikationen und mit geringer Kopplung generiert. Zusätzlich gibt es eine ausführliche Dokumentation für Benutzer beim Lesen der Texte, und es gibt ein JavaDoc, ein UML_Diagramm und sogar ein ReadMe zum verstehen des Aufbaus und allen Funktionalitäten.