

## Praktikum iOS-Entwicklung im Wintersemester 2016/2017 Übungsblatt 7

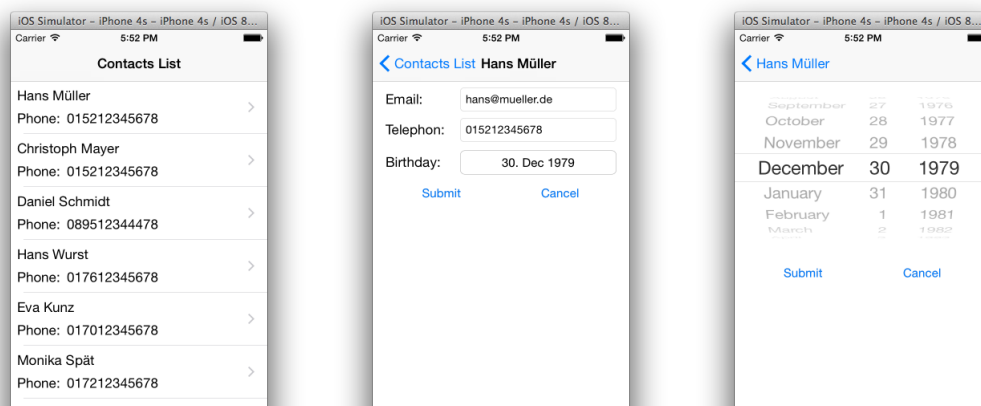
**Abgabetermin:** 20.12.2016, spätestens 23:59 Uhr  
(Gilt nur für die Teilnehmer, die ihre Lösungen am Besprechungstermin vorstellen. Dies geschieht in Form einer ca. 5-minütigen Powerpoint-Präsentation. Die Abgabe muss daher die Präsentationsdatei enthalten und erfolgt über Uniworx.)

**Besprechung:** 21.12.2016

### Aufgabe 15: (H) Kontaktliste

In der Vorlesung zum Praktikum haben Sie im Umgang mit Collection Views bereits ein Konzept zur Darstellung von Kollektionen kennengelernt. Handelt es sich bei einer Kollektion z.B. um eine einfache Liste von Objekten, bietet sich als weniger komplexe Alternative die Verwendung von Table Views an. Informationen zur Programmierung von Table Views finden Sie im *Table View Programming Guide for iOS* (<https://developer.apple.com>).

In dieser Aufgabe sollen Sie mit Hilfe einer Table View eine iOS-App entwickeln, mit der man eine Liste bestehender Kontakte anzeigen und deren Einträge modifizieren kann. Die gewünschten Anzeigen der Anwendung sind in Abbildung 1 dargestellt.



(a) Tabellenansicht der Kontakte    (b) Detailansicht der Kontaktdaten    (c) Detailansicht des Geburtsdatums

Abbildung 1: Darstellung der Anwendung in seinen unterschiedlichen Zuständen auf dem iPhone 4s.

Halten Sie sich bei der Umsetzung der Anwendung an folgende Anforderungen:

- Entwickeln Sie zur Verwaltung einzelner Kontakte eine Klasse `Contact`. Ein Kontakt soll sich durch einen Vor- und einen Nachnamen, eine Telefonnummer, eine Email Adresse und ein Geburtsdatum definieren.
  - Implementieren Sie das Datenmodell der Anwendung in Form eines Singletons. Entwickeln Sie dazu eine Klasse `ContactStore`, die den globalen Zugriff auf die Liste der Kontakte erlaubt. Fügen Sie ihrer Singleton-Instanz bei der Initialisierung zehn erfundene Kontakte hinzu.
  - Implementieren Sie die Tabellenansicht ihrer Kontaktliste in einer Table View in Anlehnung an Abbildung 1a.
  - Der Click auf einen Listeneintrag soll den Nutzer zur Detailansicht der Kontaktdaten navigieren. Implementieren Sie die Detailansicht in Anlehnung an Abbildung 1b. Setzen Sie die Anzeige der Email Adresse und der Telefonnummer als editierbare Textfelder um. Verwenden Sie für die Darstellung des Geburtsdatums einen Button.
  - Der Click auf das Geburtsdatum soll den Anwender zu einer weiteren Detailansicht navigieren, der ihm die Modifikation des Geburtsdatums erlaubt. Implementieren Sie diese Detailansicht in Anlehnung an Abbildung 1c.
  - Sowohl die Detailansicht eines Kontakts, als auch die Detailansicht für das Geburtsdatum soll einen Submit- und einen Cancel-Button zur Verfügung stellen. Sowohl der Click auf einen Submit bzw. Cancel-Button, als auch die Verwendung der Navigation Bar soll den Nutzer direkt zur vorhergehenden Ansicht navigieren. Der Click des Submit-Buttons in der Detailansicht des Geburtsdatums soll lediglich eine Modifikation des angezeigten Geburtsdatums in der Detailansicht des Kontakts bewirken! Die eigentliche Modifikation des Kontakts im Datenmodell soll erst beim Click des Submit-Buttons in der Detailansicht eines Kontakts erfolgen!
  - Verwenden Sie bei der Modellierung ihrer Anzeigen geeignete Layout-Constraints.
- a. Implementieren Sie die Anwendung mit Swift.
  - b. Implementieren Sie die Anwendung mit Objective-C

## Aufgabe 16: (H) Flickr Search Extended

In der Vorlesung zum Praktikum wurde die Anwendung *FlickrSearch* entwickelt. Das zugrundeliegende Xcode Projekt können Sie sich auf der Website zur Vorlesung herunterladen. In dieser Aufgabe sollen Sie die Anwendung um einige Funktionen erweitern. Betten Sie die Anwendung zur Bearbeitung der Aufgaben in eine Instanz vom Typ `UINavigationController` ein.

**Achtung:** Sie benötigen einen eigenen Schlüssel für die Verwendung der Flickr API. Diesen können Sie unter [https://www.flickr.com/services/api/misc.api\\_keys.html](https://www.flickr.com/services/api/misc.api_keys.html) beantragen. Tragen Sie ihren Schlüssel in der Datei `Flickr.m` an der entsprechend Stelle ein.

- a. In dieser Teilaufgabe soll die Collection View der Anwendung in ihrer Darstellung und Editierbarkeit erweitert werden.
  - (i) Collection Views lassen sich mit Instanzen der Klasse `UICollectionViewReusableView` zusätzlich dekorieren.

Fügen Sie der Collection View eine Instanz der Klasse `UICollectionViewReusableView`

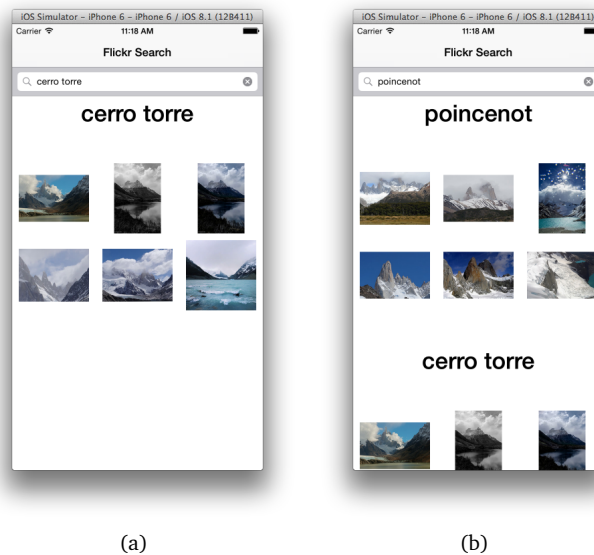


Abbildung 2: Darstellung der Anwendung nach der Eingabe von Suchbegriffen auf dem iPhone 6.

hinzu, so dass sie für eine Sektion den dazugehörigen Suchbegriff als Überschrift darstellt. Halten Sie sich bei der Formatierung der Überschrift an die Beispiele aus Abbildung 2.

- (ii) Durch die Implementierung weiterer Funktionen des `UICollectionViewDelegate`-Protokolls lassen sich einer Collection View sehr einfach die Editierfunktionen `Cut`, `Copy` und `Paste` hinzufügen. Sind die Funktionen entsprechend implementiert, verursacht eine `LongPress`-Geste die Darstellung eines Popup-Menüs mit den gewünschten Editierfunktionen. Die dabei verwendete Instanz der Klasse `UIMenuController` lässt sich auch für eigene Zwecke erweitern bzw. verändern.

Fügen Sie nun der Collection View eine Editierfunktion `Delete` zum Löschen einzelner Bilder hinzu. Verwenden Sie dazu die von der Funktion `sharedMenuController()` der Klasse `UIMenuController` zurückgelieferte Instanz. Abbildung 3 veranschaulicht das gewünschte Ergebnis.

- b. In dieser Teilaufgabe sollen Sie der Anwendung eine weitere Ansicht zur Betrachtung einzelner Bilder hinzufügen. Ein `Single Tap` auf die Vorschau eines Bildes in der Collection View soll zur Anzeige einer neuen Detailansicht führen. In dieser Ansicht soll nicht die Vorschau (Thumbnail), sondern das Originalbild angezeigt werden. Implementieren Sie die gewünschte Funktionalität unter Berücksichtigung folgender Anforderungen:
- Wird die Vorschau zum ersten mal angetippt, muss zunächst das Originalbild über die Funktion `loadImageForPhoto` der Klasse `Flickr` geladen werden.
  - Ein Aktivitätsindikator soll den Ladevorgang visualisieren.
  - Ist der Ladevorgang abgeschlossen, soll das Originalbild vollständig und unverzerrt angezeigt werden. Der Ablauf der Navigation ist in Abbildung 4 dargestellt.
  - Tritt ein Fehler auf (z.B. keine Verbindung zum Internet, ...) soll in der Detailansicht anstelle des Originalbildes lediglich die Vorschau angezeigt werden. Dem Anwender soll dieses Verhalten mit einer Warnung visualisiert werden. Der Ablauf der Navigation im Falle eines Fehlers ist in Abbildung 5 dargestellt.

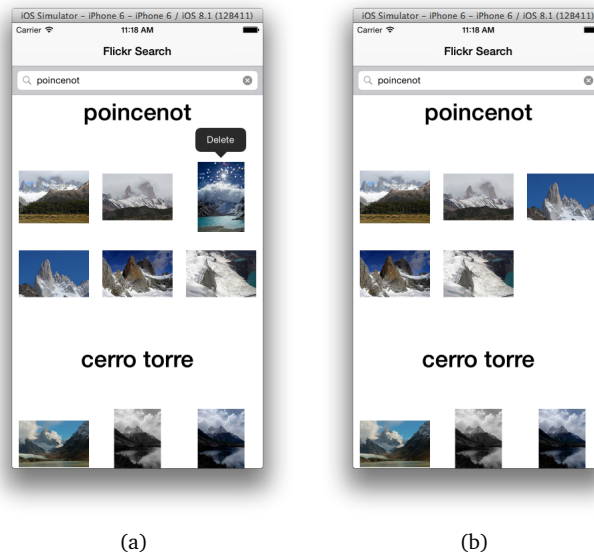


Abbildung 3: Darstellung der Anwendung auf dem iPhone 6 vor (3a) und nach (3b) dem Löschen eines Bildes.

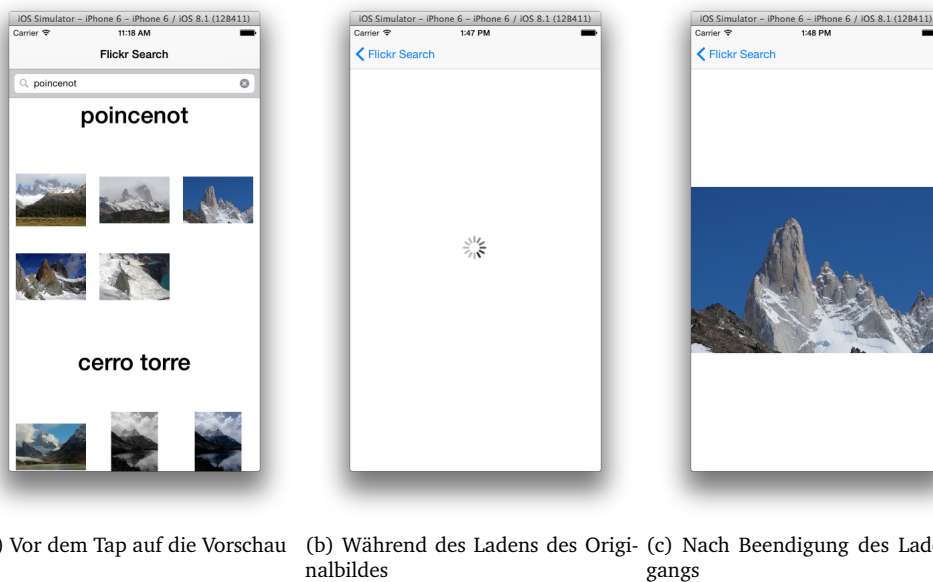
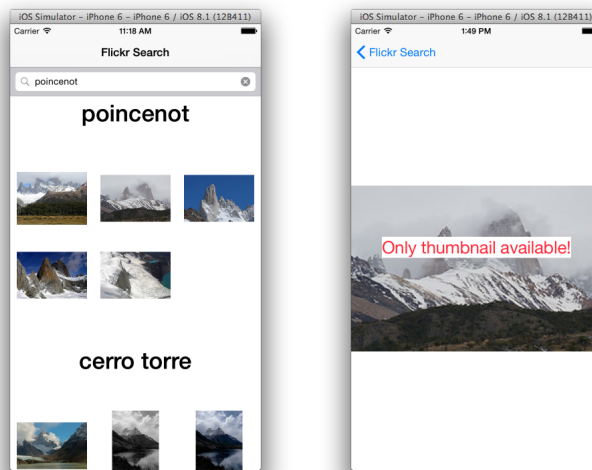


Abbildung 4: Darstellung der Navigation und der dabei auftretenden Zustände beim Ladevorgang auf dem iPhone 6.



(a) Vor dem Tap auf die Vorschau

(b) Anzeige des Thumbnails

Abbildung 5: Darstellung der Navigation unter Annahme eines fehlerhaften Ladevorgangs auf dem iPhone 6.