

iOS-Praktikum

Aufgabe 16 a)

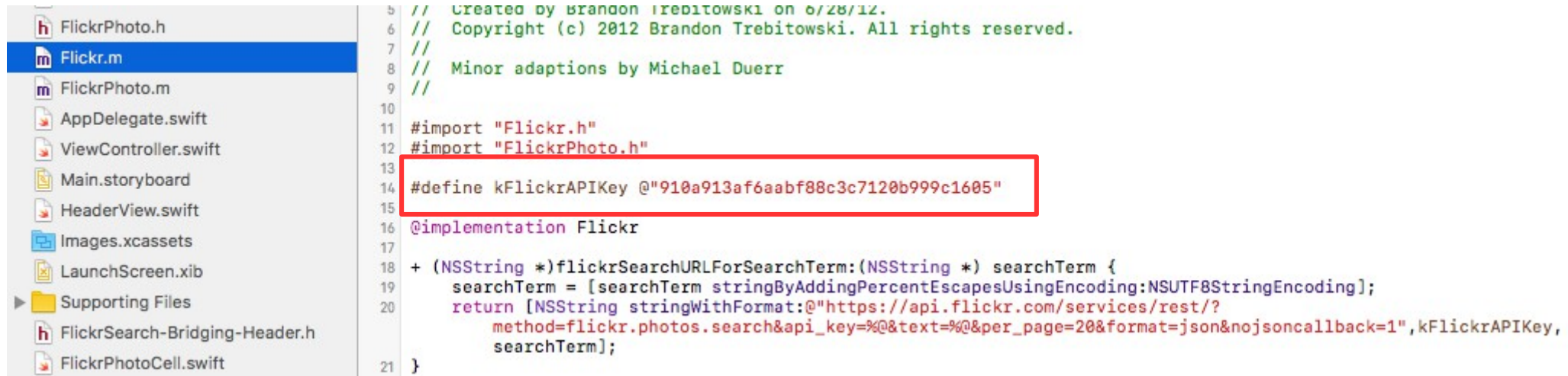
Erweiterung der FlickrSearch-Anwendung

Markus Zimmermann

i) Aufgabe: Anzeigen von Überschriften

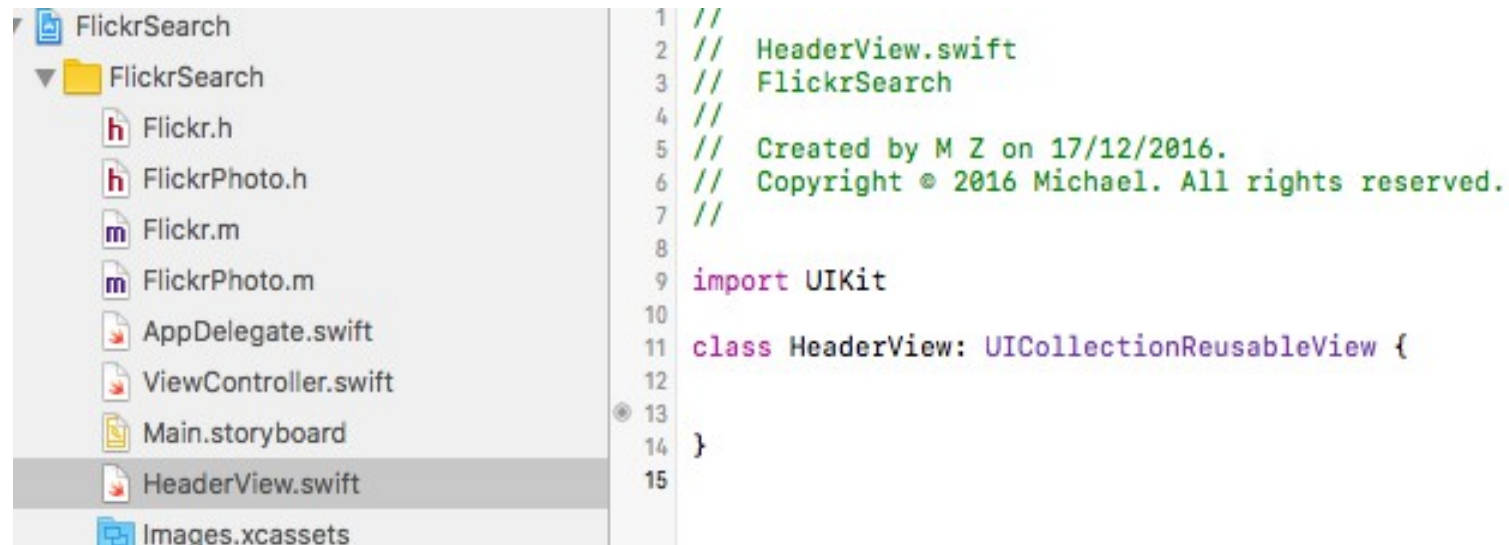


Schlüssel für Flickr API beantragen

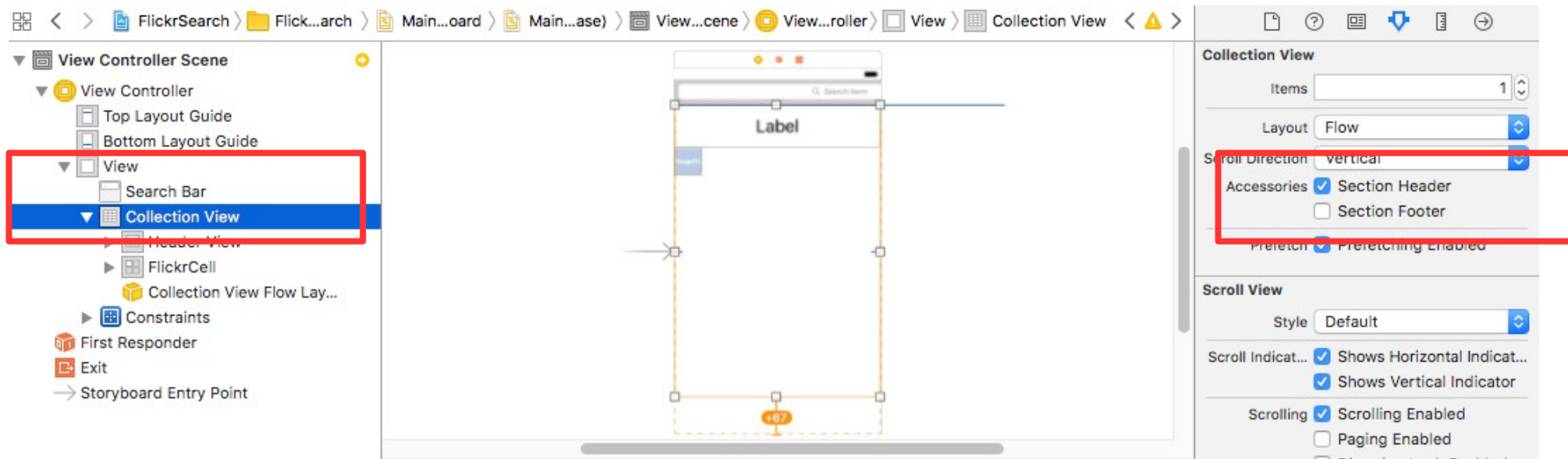


```
5 // Created by Brandon Trebitowski on 6/28/12.
6 // Copyright (c) 2012 Brandon Trebitowski. All rights reserved.
7 //
8 // Minor adaptations by Michael Duerr
9 //
10
11 #import "Flickr.h"
12 #import "FlickrPhoto.h"
13
14 #define kFlickrAPIKey @"910a913af6aabbf88c3c7120b999c1605"
15
16 @implementation Flickr
17
18 + (NSString *)flickrSearchURLForSearchTerm:(NSString *) searchTerm {
19     searchTerm = [searchTerm stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];
20     return [NSString stringWithFormat:@"https://api.flickr.com/services/rest/?
21         method=flickr.photos.search&api_key=%@&text=%@&per_page=20&format=json&nojsoncallback=1", kFlickrAPIKey,
22         searchTerm];
23 }
```

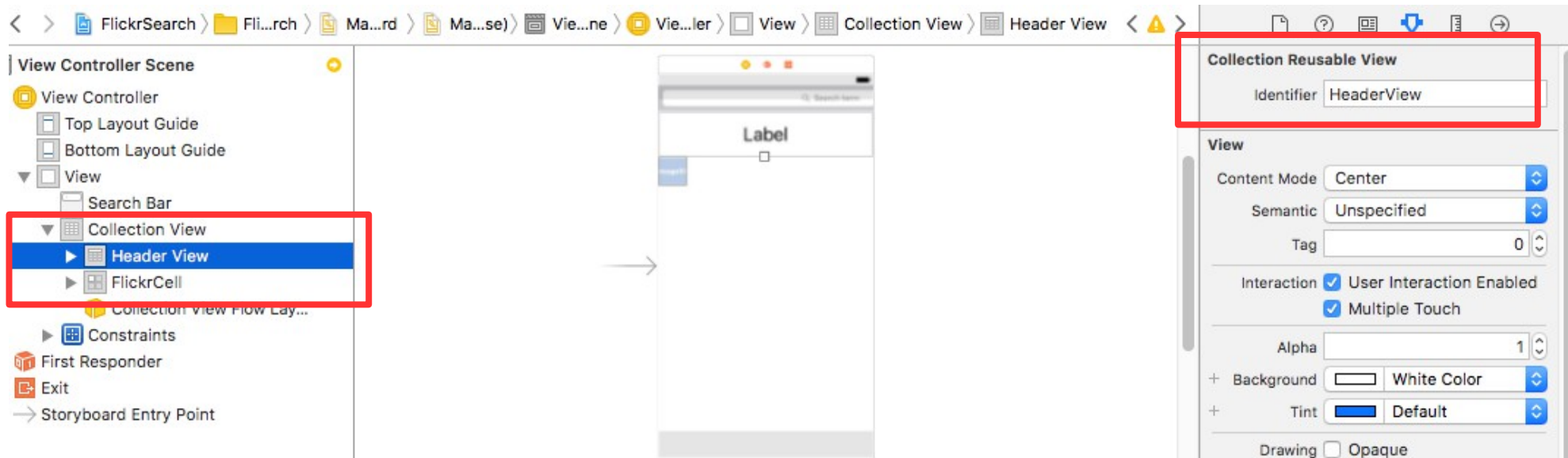
View für Header erstellen: ReusableView



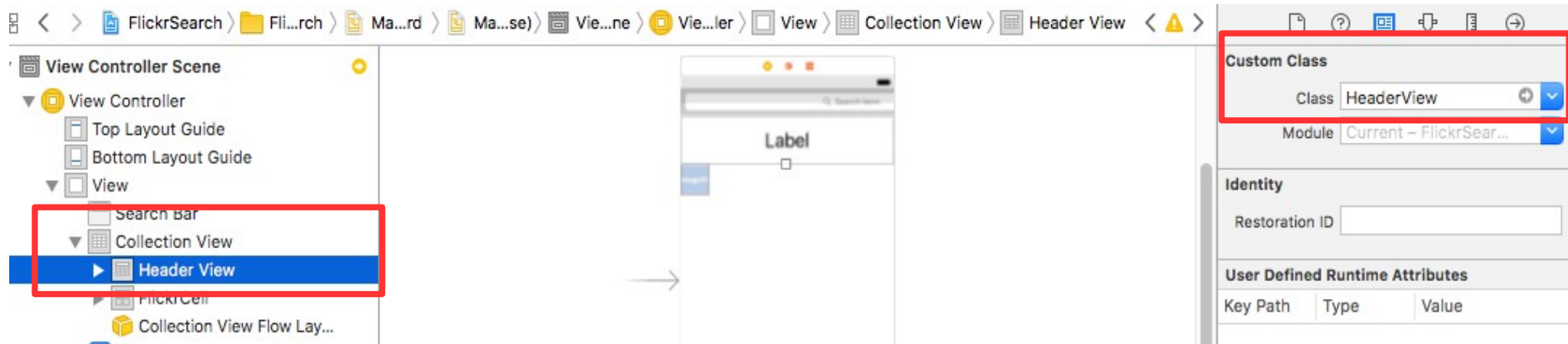
Storyboard bearbeiten: Section Header



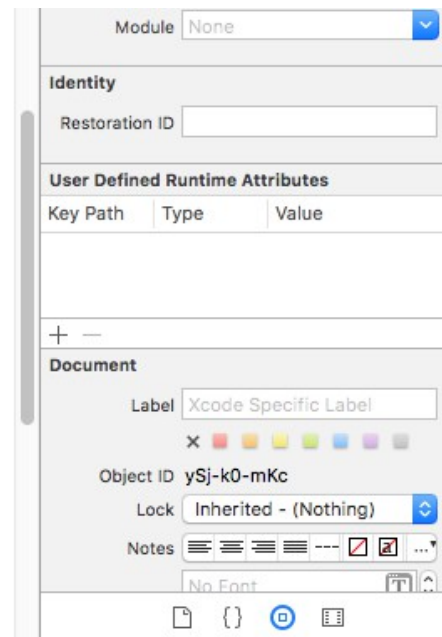
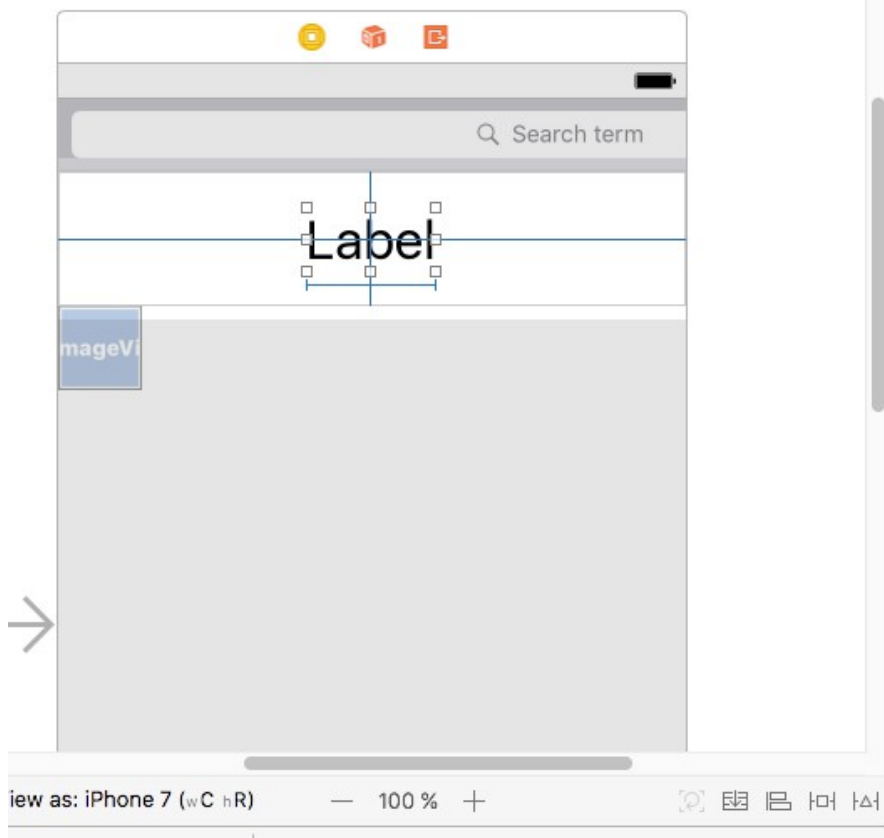
Storyboard bearbeiten: Identifier



Storyboard bearbeiten: Klasse verknüpfen



Label hinzufügen



Label - A variably sized amount of static text.

```
8
9 import UIKit
10
11 class HeaderView: UICollectionView {
12
13     @IBOutlet weak var headerLabel: UILabel!
14 }
15
```


Funktion in ViewController.swift

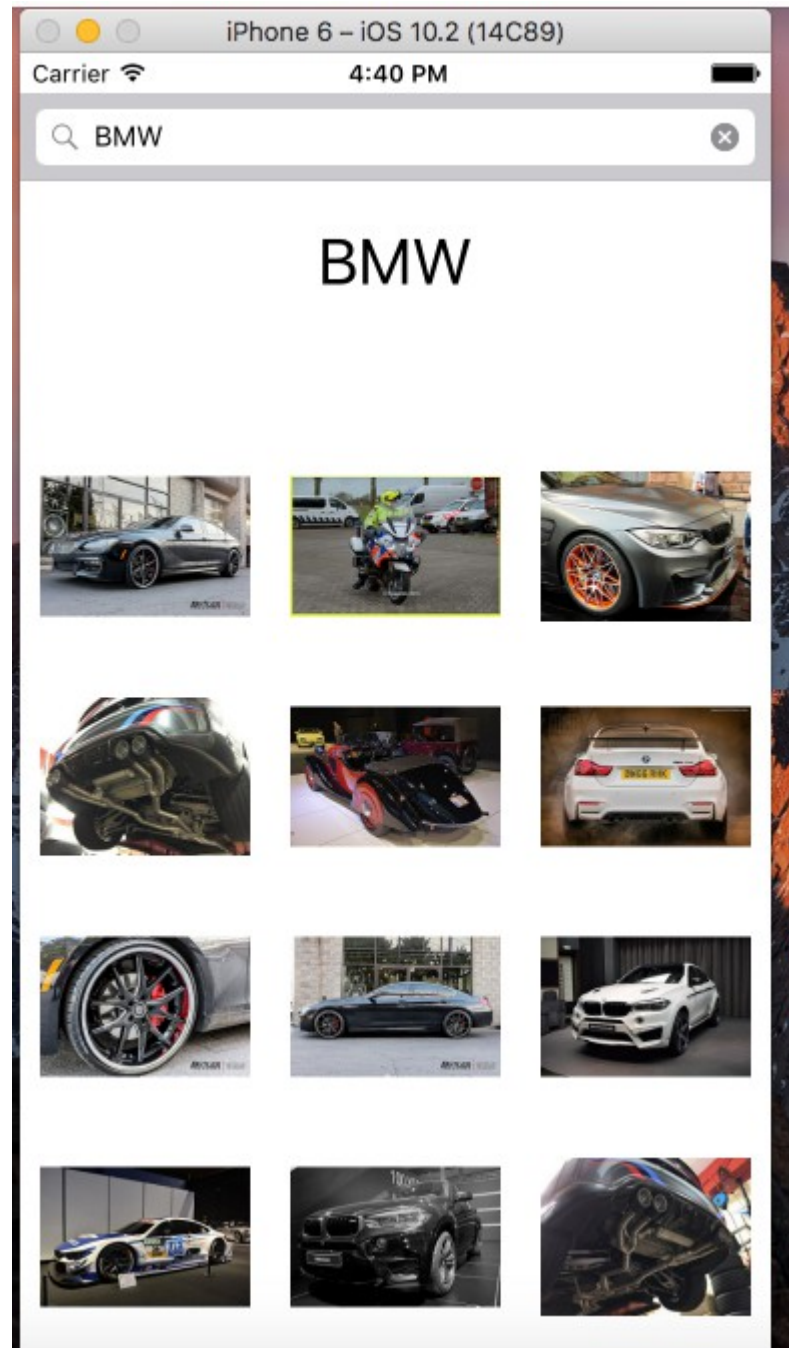
```
//Setting the Label
func collectionView(collectionView: UICollectionView, viewForSupplementaryElementOfKind kind: String,
    atIndexPath indexPath: NSIndexPath) -> UICollectionViewReusableView {

    //Get current header
    let header = collectionView.dequeueReusableSupplementaryViewOfKind(UICollectionElementKindSectionHeader,
        withReuseIdentifier: "HeaderView", forIndexPath: indexPath) as! HeaderView

    //Set text to current search term
    header.headerLabel.text = searches[indexPath.section]

    //Return headerView
    return header
}
```

Ergebnis:



ii) Aufgabe: Löschen von Fotos



LongPress – Geste hinzufügen

```
8
9 import UIKit
10
11 class ViewController: UIViewController, UICollectionViewDelegateFlowLayout , UISearchBarDelegate,
    UICollectionViewDelegate, UICollectionViewDataSource {
12
13     @IBOutlet weak var searchBar: UISearchBar!
14     @IBOutlet weak var collectionView: UICollectionView!
15
16     weak var currentIndexPath:NSIndexPath?
17
18     var searches = Array<String>()
19     var searchResults = Dictionary<String,Array<FlickrPhoto>>()
20     var flickr = Flickr()
21
22     override func viewDidLoad() {
23         //collectionView.registerClass(UICollectionViewCell.self, forCellWithReuseIdentifier: "FlickrCell")
24
25         //add long press gesture
26         let longPressRecognizer = UILongPressGestureRecognizer(target: self,
27                                                                 action: #selector(ViewController.showMenu))
28         self.collectionView?.addGestureRecognizer(longPressRecognizer)
29     }
30 }
```

ViewController.swift

Benötigte Methoden überschreiben

```
// override required functions
override func canBecomeFirstResponder() -> Bool {
    return true
}

func collectionView(collectionView: UICollectionView,
                    shouldShowMenuForItemAtIndexPath indexPath: NSIndexPath) -> Bool {
    return true
}

func collectionView(collectionView: UICollectionView, canPerformAction action: Selector, forItemAtIndexPath
                    indexPath: NSIndexPath, withSender sender: AnyObject?) -> Bool {
    return action == #selector(deleteSelectedPhoto)
}
```

ViewController.swift

Menu erstellen

```
// show menu
func showMenu(gesture: UILongPressGestureRecognizer){
    if gesture.state == UIGestureRecognizerState.Began {
        // indexPath for gesture position
        let indexPath =
            self.collectionView?.indexPathForItemAtPoint(gesture.locationInView(self.collectionView))

        if indexPath != nil {
            // select cell at indexPath
            let cell = self.collectionView?.cellForItemAtIndexPath(indexPath!)
            // create menu
            let menu = UIMenuController.sharedMenuController()
            let deleteMenuItem = UIMenuItem(title: "Delete", action: #selector(deleteSelectedPhoto))
            menu.setTargetRect(CGRectMake(0, 5, 60, 80), inView: (cell?.contentView!))
            menu.menuItems = [deleteMenuItem]
            menu.setMenuVisible(true, animated: true)
            currentIndexPath = indexPath
        }
    }
}
```

ViewController.swift

Delete - Funktion

```
// delete function
func deleteSelectedPhoto(){
    // path to deleted photo
    let path = currentIndexPath!

    // remove photo from datasource object
    searchResults[searches[path.section]]!.removeAtIndex(path.row)

    // remove item from collectionView
    self.collectionView.deleteItemsAtIndexPaths([path])

    // update
    self.collectionView.reloadData()
}
```

ViewController.swift

Ergebnis:

