SnapCity

Burhanuddin Shakir
17204504
COMP41550 - Project

Description

SnapCity is an app which downloads and displays geotagged images of a given location called The app has the following features:

- A map view to display the current location.
- A custom collection view to display the downloaded images
- A 3D touch preview of image
- A full screen view controller to display the picture with some information about picture

SnapCity uses MapKit to display the map view. On double tapping anywhere on the map, a pin is dropped and photos which are geotagged for that location are downloaded using the Flickr API for a 1km radius of the pin. The photos are then displayed in a collection view and as soon as they are available, they can be viewed with some information about them. Swipe down to dismiss the photo and a single tap reveals data about the pic. Share button can be used to share and download the image and navigation button opens apple maps to show direction from users current location to image's location. Peek and pop of image can be done on 3D touch supported devices.

Instructions:

- Download .zip file
- Open snap-city folder and click on **snap-city.xcworkspace**
- Project should open in xcode

App Architecture

MapVC:

The MapVC displays the map and collection view. It also implements dropping of pin, Calls to flickr API and image preview.

Implementation of pin dropping :

```
let touchPoint = sender.location(in: mapView)
    let touchCoordinate = mapView.convert(touchPoint, toCoordinateFrom: mapView)

    let annotation = DroppablePin(coordinate: touchCoordinate, identifier: "pin")
```

```
        mapView.addAnnotation(annotation)

        let coordinateRegion = MKCoordinateRegionMakeWithDistance(touchCoordinate,
regionRadius, regionRadius)

        mapView.setRegion(coordinateRegion, animated: true)
```

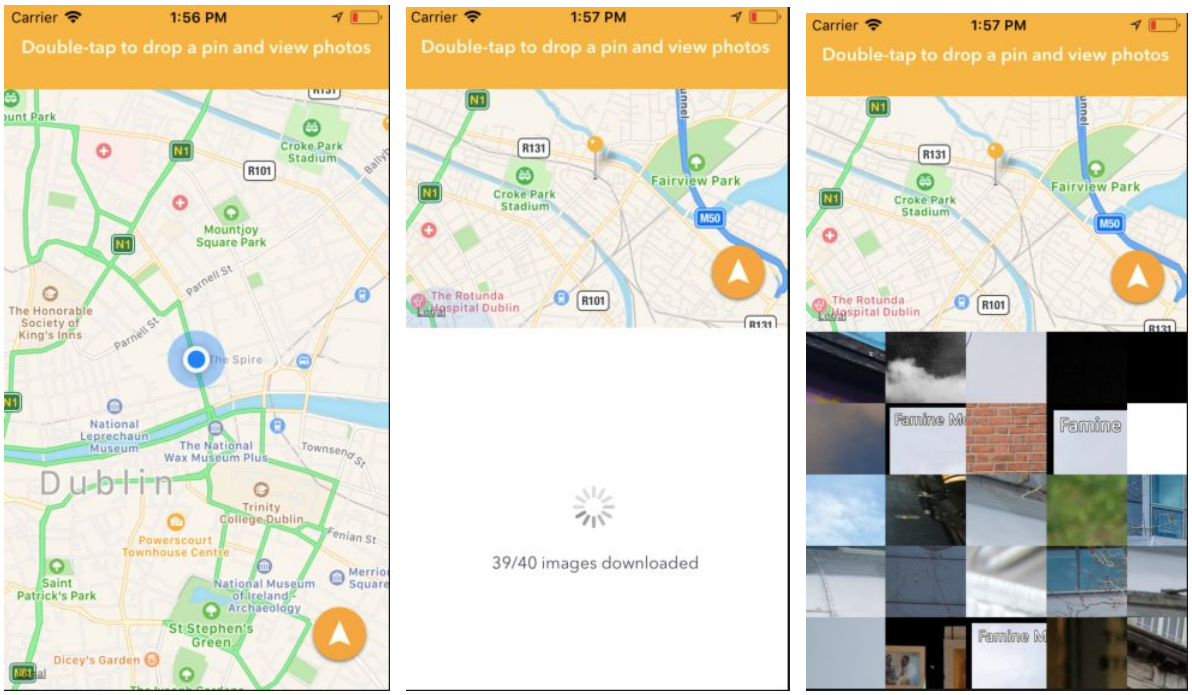
Using alamofire to retrieve images :

```
func retrieveImagesURL(forAnnotation annotation: DroppablePin, handler:@escaping (_ status :
Bool) -> ()){


        Alamofire.request(flickrUrl(forApiKey: api_key, withAnnotation: annotation,
andNumberOfPhotos: 40)).responseJSON { (response) in

            guard let json = response.result.value as? Dictionary<String,AnyObject> else {return}

            print(json)

            let photoDict = json["photos"] as! Dictionary<String,AnyObject>

            let photoDictArray = photoDict["photo"] as! [Dictionary<String,AnyObject>]

            for photo in photoDictArray{

                let postURL =
"https://farm\(photo["farm"]!).static.flickr.com/\(photo["server"]!)/\(photo["id"]!)_\(photo["secret"]!)_
h_d.jpg"

                let flickrImage = FlickrImage(url: postURL, description: photo["title"] as! String, id:
photo["id"]! as! String)

                self.imagesArray.append(flickrImage)
            }

            handler(true)

        }
    }
```

Implementing 3D touch for images :

```
func previewingContext(_ previewingContext: UIViewControllerPreviewing,
viewControllerForLocation location: CGPoint) -> UIViewController? {

    guard let indexPath = collectionView?.indexPathForItem(at: location), let cell =
collectionView?.cellForItem(at: indexPath) else { return nil}

    guard let popVC = storyboard?.instantiateViewController(withIdentifier: "PopUpVC") as?
PopUpVC else { return nil }
    popVC.initData(forImage: downloadedImages[indexPath.row], andData:
imagesArray[indexPath.row])

    previewingContext.sourceRect = cell.contentView.frame
    return popVC
  }

  func previewingContext(_ previewingContext: UIViewControllerPreviewing, commit
viewControllerToCommit: UIViewController) {

    show(viewControllerToCommit, sender: self)
  }
```

Screenshots



Popup VC

Popup VC displays images and its information

Sharing images :

```
// set up activity view controller
    let imageToShare = [ image! ]
    let activityViewController = UIActivityViewController(activityItems: imageToShare,
applicationActivities: nil)

    // present the view controller
    self.present(activityViewController, animated: true, completion: nil)
```

Showing directions in apple maps :

```
let location:CLLocationCoordinate2D = CLLocationCoordinate2D(latitude: Double((lat as
NSString).doubleValue), longitude: Double((long as NSString).doubleValue))


    let coordinate = CLLocationCoordinate2DMake(location.latitude, location.longitude)
    let mapItem = MKMapItem(placemark: MKPlacemark(coordinate: coordinate,
addressDictionary:nil))
    mapItem.name = "Destination/Target Address or Name"
    mapItem.openInMaps(launchOptions: [MKLaunchOptionsDirectionsModeKey :
MKLaunchOptionsDirectionsModeDriving])
```

ScreenShots of PopUp VC