Programming Challenge

Functional requirements

You are tasked with creating a simple application that will present a stream (feed) of photos. We would like to be able to scroll through the photos vertically – like we are used to doing in a web browser.

The photo stream is available on our **RESTful JSON** web service that we have set up for the purpose of this exercise. You should use the following entry points:

- **challenge.superfling.com/** - content of the feed, that contains ids of the photos and their titles.
- **challenge.superfling.com/photos/** - photos, available by their id presented in the feed.

The application should:

- Read the feed content from the above web service and store the data in a Core Data table.
- Present the photos – downloaded form our web service – in a neat, scrollable, one column stream.
- There should be a title overlying each photo on the upper part – presenting text supplied by the service. We want to keep the title within 1 line of text.
- Each of the photos should be downloaded at only once during one application execution. But when the application is terminated and launched again, the photos should be downloaded again.

Technical requirements

It is enough that the application can be run on **iPhone 4S/5/5s** with **iOS7**. We may want to run it on **iPhone 6** or **iPad** to see how it is doing – but the primary devices are as described.

Evaluation criteria

The functionality of the application is simple because we want to make sure that:

- you care about the user experience
- you create elegant, clean and maintainable code
- you are able to design good objective architecture for your application
- you develop in a scalable manner
- you pay attention to details In case of doubts, use your own judgement to take appropriate decisions to create the best (in your opinion) solution – this is just a test to show your approach to problem solving and software development.

Deliverables

We expect, that you create a complete **Xcode** project, in a version-controlled repository (preferably with several commits so that we can see the process of creating the app).

We want to be able to open it and run immediately on our device!

Extension 1 - Graphical improvements

Functional requirements

We would like to add some additional features to make the presentation of the feed more attractive:

- When the user scrolls the feed to photos, that have not been yet downloaded, the app will obviously try to download them from the webservice. After an image has been downloaded, we want to present it with some animation - it should not just pop up. The proposed animation is to fade in the photo and to enlarge it slightly (during the fade in animation) - up to the desired size.
- When the user touches any of the images, it should perform a small animation - rotate over its horizontal axis (the rotation axis is in the center of the image, from left to right). The rotation can be complete or partial, but when finished, the animation should bring the photo to its original state (without any distortion).

Extension 2 - More Data

The feed contains information about users, who created the entries. We want to store this information in Core Data (in the appropriate data model), so that we can produce some statistics. As an example, generate the following information - based on the information stored in Core Data:

User Name	Number of posts	Average image size	Greatest photo width
User 1	12	558,045	2048
...	...	...	...

The application should present this report using NSLog commands, after all the photos have been downloaded (the user has scrolled the entire feed so that all images have beed downloaded from the webservice). The format is entirely up to you - any text format is acceptable, no table is required.

The report must be based only on the information stored in Core Data - so that it can be recalculated at any moment, using the method that you will create for this purpose.