Pixol Report

**Advance Programming**

Course Instructor

**Mr. Shakeel Anjum**

Submitted By

**14F\_8028**

Date

**October 11, 2016**

Fall 2016



Department of Computer Science

FAST–National University of Computer & Emerging Sciences Chiniot – Faisalabad Campus

Table of Contents

[Acknowledgement 3](#_Toc463913161)

[Abstract 3](#_Toc463913162)

[Introduction 4](#_Toc463913163)

[Problem Statement 4](#_Toc463913164)

[Solution Design 5](#_Toc463913165)

[Techniques Used 7](#_Toc463913166)

[References 8](#_Toc463913167)

# Acknowledgement

In the name of Allah, the most beneficent and the most merciful. In the very beginning we first thank our maker **“ALLAH THE ALMIGHTY”** Who is the creator of all. Because of his blessings we were able to start our project and hereby complete it within the required time. It’s been with us the great right hand of our honorable teacher **Mr. Shakeel Anjum** who helped us in every moment whenever we were in a pinch. It was him who believed in us and made us believe in ourselves to complete this project.

Side by side we pay regards of gratitude to our parents as they and their prayers for our success have always been a pillar of strength for us in our life. Significantly, for this particular project the special prayers by our worthy parents are the cause of the success for all of us in the entire project.

# Abstract

The efficiently managed and well organized efforts of our group made this project a useful utility console application into production that is named as Pixol. Pixol is a command line based application that enables the user to download his/her pictures from their Instagram account. These pictures are downloaded by this application this option wasn’t given by any other software before. Our group has done their best to provide such a solution for the users to coup up with this need while backing up their data from their social account. These pictures are then stored in a directory and can be used as needed.

# Introduction

Pixol is a project chosen by our group for the 1st mid-term project of the course “Advance Programming”. It is basically an application that runs on command line to download the uploaded images on the **Instagram** account of an individual. **Instagram** is an online mobile photo-sharing, video-sharing, and social networking service that enables its users to take pictures and videos, and share them either publicly or privately on the app, as well as through a variety of other social networking platforms, such as Facebook, Twitter, Tumblr, and Flickr. It is therefore able to download the images on an account even if it is a private account. These images are downloaded in a JPEG format that can be reused if needed. Basically for both types of accounts the API platform of the Instagram is used to interact with the application.

In a deeper view the Instagram provides a mode known as a “sandbox mode” to interact with the application within a certain range of parameters and the restrictions. Every new app created on the Instagram Platform starts in sandbox mode. This is a fully functional environment that allows you to test the API before. The users and media available in Sandbox mode are real Instagram data (i.e. what is normally visible in the Instagram app). So, the data is restricted to 10 users and 20 most recent media from those particular users, following these restrictions imposed on our application as well, we are able to download all the images of an individual if he/she has a public account and 20 most recent pictures if he/she has an account that is private. When the application will be approved be the Instagram and goes live then the application is able to download all the images of that any private account.

# Problem Statement

If we pass eye around us, nearly seventy to eighty percent of the people now have a social account to enhance the social image of themselves. Some use it as a time pass but most of the ratio contains people who use the social accounts to increase their interaction in the society. This is considered as a plus point in an individual’s life. In a nutshell, the social networks are a source for an individual to pass on an image of themselves that they approve of. Therefore, Instagram is also one of the social identities of an individual. Instagram is used to post photos that are shared with people that are following a particular account.

These social identities are like an asset for a person to share their daily routine and want that data to be stored on such sites. This is the behavior that led us to make such an application that backs up the data uploaded by an individual from a social platform.

Why do we need to back up the data on Instagram? The answer comes from the limit that is associated with such accounts which allows a user to upload limited amount of pictures. It is a figure of **fifteen hundred** images that Instagram allows for a single individual to upload on their site after which it asks user to delete some of their existing pictures from their account to upload new ones.

Therefore, it a big problem to those people who have high social status and want it to be maintained on a daily basis. So in order to tackle with this problem, our group searched for a solution if it is available and it solves the problem of such users, the solution must solve the problem with a technique that is easily usable by the users. However, it was of no hope as no backup methods are available or provided by the makers of such platforms for their users to back up their data quickly with no difficulty. So, our group decided to make a project to provide a solution to the problem of picture limits imposed on their accounts. The solution is to download the pictures from a particular account on Instagram and store it in a directory. Thus, this can be used by all Instagram users to deal with this problem.

# Solution Design

Starting with the initial behavior of the application, it is providing a **“Pixol.exe”** file that is to be executed in order to run the application. ”Pixol.exe” file has to be used with certain type of arguments that are necessary to run the application correctly. Basically it needs to run the **“Pixol.exe”** file via the command line. It has two options; one for public accounts and the other for private accounts. In the case of public account, the first parameter is the –public account type specifier and the next parameter is the username of the account the pictures are going to be downloaded from. Similarly, for the private account the first parameter is the –private account type specifier and the next parameter is the username followed by a space and then the password of an account. Then user can also specify minimum likes limit for an image to qualify for download using “–likes” specifier followed by minimum number of likes the image must have.

The next step is the code that integrates the following application to be executed in a normal flow. The account type specifier decides which method of getting images will be used. In case of –public the public **getPublicPhotos(string targetprofile, int likes)** begins to execute and in case for the –private the **getPrivatePhotos(string username, string password, int likes)** method is executed. In both cases, the API of the Instagram is used in the application to deal with the artifacts. Let’s discuss the public part first the method **getPublicPhotos(string targetprofile, int likes)** is called that picks up a public URL placed in the “app.config” file and the parameter “targetprofile” is passed to that URL and the request is made to that URL to get information of a public account. The program first creates a directory with the targetprofile name if it doesn’t exist already to download the images. **RestSharp** is used as an API client to interact with the Instagram API, GET method is used for the request.

When the request is executed after assigning the final URL to the client, as a result, a response is received from the server in the form of a script that is known as **JSON (Java Script Object Notation) script**. JSON is a collection of the name value pairs that in various languages recognized as objects and the ordered list of values named as the arrays. This JSON script returned is then deserialized into a **“RootObject”** class instance that was made using the **online JSON to c# Class generator**. The public script contains three data items as a status, an array of items and a Boolean variable more available. The items are basically information of the images stored in the account that are going to be downloaded using this application.

One by one the bytes are downloaded using URL or images in the collection of items and these bytes are then converted into a Memory stream that is then used to make a bitmap object which is then saved in the jpeg format as an image. This is the process through which the images of the public account are downloaded in a particular directory.

The other part is to get the images from a private account. In this part a method named as getPrivatePhotos(string username, string password, int likes) is called. This requires an **access token** to be granted that is then used to make requests to the Instagram API on user’s behalf. These tokens are unique to a user and should be stored securely. Access tokens may expire at any time in the future.

Therefore to receive the access token we have to direct the user to our authorization URL where the user account allows our app to get access token for his account by doing automated login. Then program uses a URL in the “app.config” file to which the access token is passed and the JSON script is received that is of different data from the public one. It contains a “pagination object”, a “meta object” and an array of “datum” type which contains the information of all the media uploaded by the user. Apart from this the overall procedure to download images is the same as that of a public account. So, the pictures are downloaded in the same way as of the procedure denoted above in the case for the public profiles of Instagram.

# Techniques Used

The techniques used are of a huge value to our project that made it useful. We completed our project in allocated time using some third party libraries and APIs, which massively increased the simplicity of the code. I.e. **RestSharp, NEWSOFT JSON.**

Firstly, the basic technique is to use the RestSharp as a client API, which makes it easy to request to the Instagram API and get response in simple code lines. The ease of access to the API made the task go smarter. In this way a response can also be received using the GET method of which the **IRestResponce** is used to save the response that is received on the particular request message sent via the RestSharp client.

As the response is now received which contains the JSON script as content that is a hierarchy of the object that are stored on that particular account on the Instagram. Using NEWSOFT JSON deserializer the JSON script which was containing the hierarchy of objects containing information of the accounts was deserialized into C# class called RootObject to access each and every item of the script and then from that particular item it is easier to access the image. These images are then in downloaded through another client instance.

When an image is downloaded in the form of byte array it is then converted in a memory stream. As this memory stream will then be used to make an image using the bitmap object. This bitmap object can then be easily treated in numerous ways of your choice. In this project we are just saving the particular bitmap image in a directory that is being made by the similar name of the username. Furthermore different operations can be performed on this bitmap if necessary in the future requirement of the project.

# References

The following are the references through which the required information is taken that makes as the significant part of the project.

* [https://www.instagram.com](https://www.instagram.com/)
* [http://json2csharp.com](http://json2csharp.com/)
* [http://restsharp.org](http://restsharp.org/)