Overview

I created a Twitter account specifically for this coursework on the 17th January of this year; hence the inactivity – lack of friends, followers, likes and posts. The language that I decided to use was Java.

Figure 1 – Twitter Page

Application created

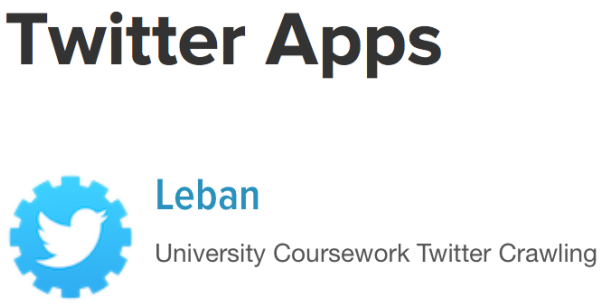

Figure 2i – New Application 'University Coursework Twitter Crawling'

Figure 2ii – Account Information

Figure 2iii – Account Information

Figure 3i and 3ii shows the API key, API secret, access token and access token secret. Vital information is kept hidden by red squares.

Part A – Profile

```
private static void getProfile(User user) {
    print("Profile:");

    print("Date: " + user.getCreatedAt());
    print("UserName: " + "@" + user.getScreenName());
    print("Name: " + user.getName());
    print("ID: " + user.getId());
    print("Profile Image: " + user.getProfileImageURL());
    print("Following: " + user.getFriendsCount());
    print("Followers: " + user.getFollowersCount());
    print("Language: " + user.getLang());
    print("AccessLevel: " + user.getAccessLevel());
    print("Favourites: " + user.getFavouritesCount());
}
```

Figure 3i – Profile Code

When the above method fragment code is called it prints out most of my twitter profile. This includes the date created, username, my actual name, if of my twitter account, URL to my profile image, following and follower's count, language, access level and favourites. To note, the 'print' is a method that simply returns sout.

Output:

```
Profile:
Date: Wed Jan 17 15:58:09 GMT 2018
UserName: @lebanmohamed1
Name: Leban
ID: [redacted]
Profile Image: http://abs.twimg.com/sticky/default_profile_images/default_profile_normal.png
Following: 15
Followers: 11
Language: en
AccessLevel: 2
Favourites: 4
```

Figure 3ii – Output of Profile

Part B – Timeline

```
private static void getTimeline(User user, Twitter twitter) throws TwitterException {
    print("----------------------------------------");
    print("Timeline:");

    List<twitter4j.Status> statuses = twitter.getHomeTimeline();
    for (twitter4j.Status status : statuses) {
        print("@" + status.getUser().getScreenName() + " - " + status.getText());
    }
}
```

Figure 4i – *Timeline Code*

The above piece of code continuously prints out my twitter feed by matching the users name to their statement. This iteration continues until it goes through all my timeline (given that mine is fairly small I'm unaware of the limitation of this).

Output:

```
@yousfk_ - @muzaba80 Hello dude, you are cool. Friendship?
@YousufB9 - No one cares https://t.co/V04AVbt3ZI
@i_ilyasm - Just setting up my Twitter. #myfirstTweet
@Ilyas_1503 - RT @DonGorganLipz: Top 5 wettest black youts of all time https://t.co/8iUUX75dBs
@Ilyas_1503 - RT @aliladiere: Levels. https://t.co/c1WQhIjdVS
@Ilyas_1503 - RT @thesolesupplier: Retweet if you're feeling this Air Max 97 'Atmos' 🐘
```

Figure 4ii – Timeline Output

Part C – Followers

- Indegree = 15

The indgree are those I'm followed to, but they are not following me. So I'm connected to them but they are not connected to me.
This can be illustrated via a directed graph where I'm connected to Java buy Java isn't connected to me:

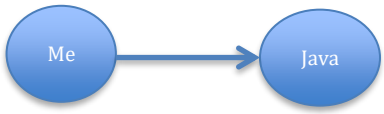

Figure 5i– Followers Example Directed

```
private static void getFollowers(User user, Twitter twitter) throws TwitterException {
    print("----------------------------------------");
    print("Followers:");

    IDs followerID = twitter.getFollowersIDs(user.getScreenName(), -1);
    long[] followerIDs = followerID.getIDs();
    for (long s : followerIDs) {
        print("@" + twitter.showUser(s).getScreenName());
    }
}
```

Figure 6ii– Followers Code

The above code iterates through all of the people I follow and prints their names out one after the other. It does this by holding my followers ID into an array that then loops through converting their ID's to their usernames.

Output:

```
Followers:
@i_ilyasm
@SajPapakhaAli
@wengerin1997
@yousfk_
@HishamTraboulsi
@YousufB9
@ShakaChow
@average_moe
@Ilyas_1503
@HiHasan22
@Rohim04614724
```

Figure 5iii– Followers Output

Part D – Following

- Outdegree = 11

The outdegree are those who follow me, I could or could not be following them back. So we could possibly be connected to each other (friends of one another).

This can be illustrated via an undirected graph where Ilyas and I are connected to each other:

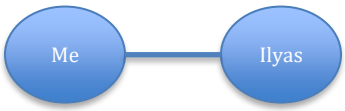

Figure 6i– Following Undirected Example

Or:

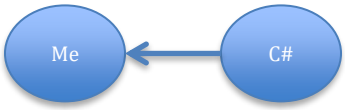

Figure 6ii– Following Directed Example

```
private static void getFollowing(String[] args, Twitter twitter) throws TwitterException {
    print("----------------------------------------");
    print("Following:");

    List<Long> followingID = new ArrayList<Long>();
    IDs ids;

    if (0 < args.length)
        ids = twitter.getFriendsIDs(args[0], -1);
    else
        ids = twitter.getFriendsIDs(-1);

    for (long id : ids.getIDs()) {
        followingID.add(id);
    }

    for (int i = 0; i < followingID.size(); i++) {
        print("@" + twitter.createFriendship(followingID.get(i)).getName());
    }
}
```

Figure 6iii– Following Code

The code gets all users ids that I follow storing them into a List, then iterates through their IDs to get their usernames.

Output:

```
Following:
@Ilyas M
@Musab Adam
@Sajjad Ali
@ismail.
@Yousf
@Adil
@Mahdi Chowdhury
@Yousuf B
@Asstronaut
@Ilyas
@Shakib Chowdhury
@Masud Ahmed
@Hisham Traboulsi
@Rohim
@H.Malik
```

Figure 6iv– Following Output