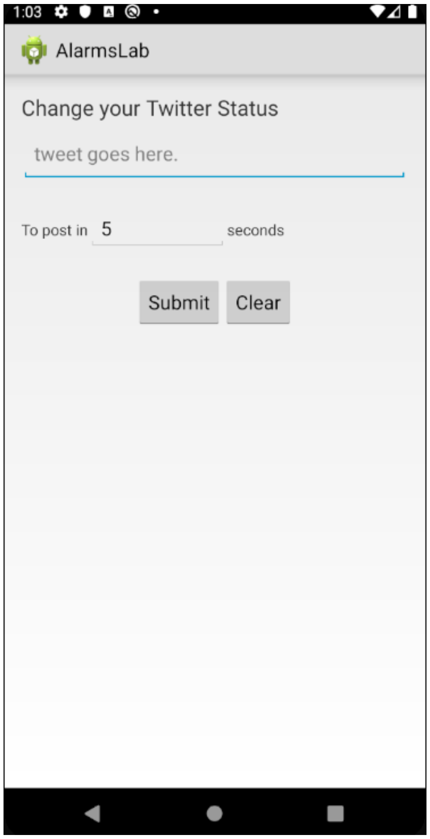
# Lab 8 - Alarms

## Objectives:

Familiarize yourself with Alarms and Networking. Create an application that uses Alarms, Networking, and the AsyncTask class.

Once you've completed this lab you should have a better understanding of Alarms, Networking and the AsyncTask class. You should know how to use and create alarms using the AlarmManager service, how to use Networking support classes to send HTTP POST/GET requests to other services on the Internet, and how to use the AsyncTask class to perform short tasks on a separate thread. For now, we have already implemented the Networking potion of the code. But we highly encourage you to go through that part of the code to get better understanding of it.

## Overview:



1:03

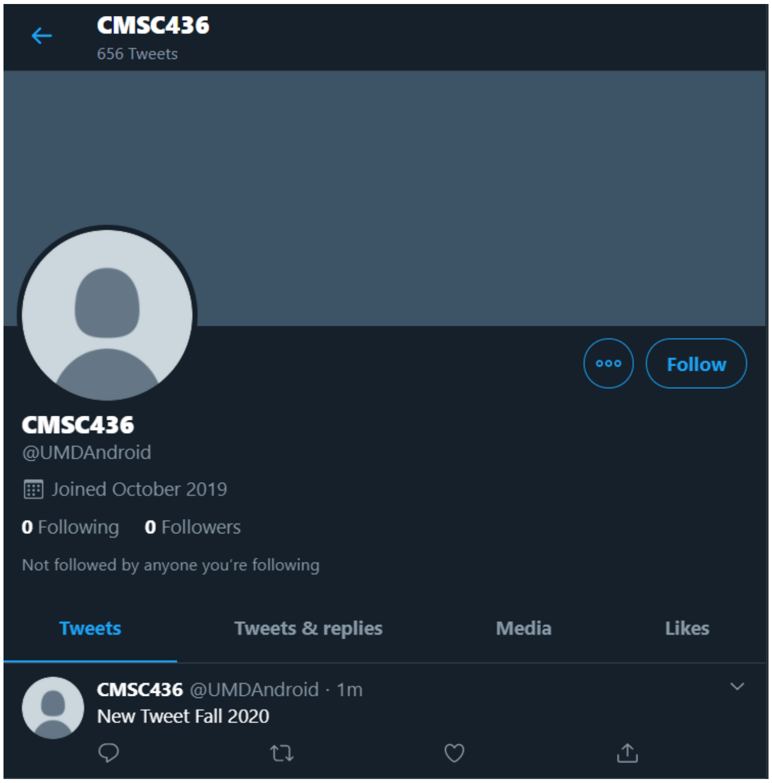
AlarmsLab

Change your Twitter Status

tweet goes here.

To post in 5 seconds

Using the app depicted below, the user will enter a Twitter status update and then set a delivery time, measured as some number of seconds from the current time. After this, if the user hits the submit button, the code should set an alarm to go off at the specified time. When the alarm goes off, it should start the AlarmTweetService, which will do the work of posting the new status to Twitter. You can see your posts on the www.twitter.com/UMDAndroid Twitter handle.



CMSC436
656 Tweets

CMSC436
@UMDAndroid
Joined October 2019
0 Following 0 Followers
Not followed by anyone you're following

CMSC436 @UMDAndroid · 1m
New Tweet Fall 2020

See the screencast of the app in operation.

## Implementation Notes:

1) Checkout the Lab8_Alarms branch from the upstream repo. Make sure to push it to your origin right away before you start implementing any code.

2) Note that since we will be sharing a Twitter account, you should put some unique identifier on your tweets. Otherwise, you won't be able to distinguish your tweets from those of your classmates.

3) Modify the AlarmCreateActivity.kt file, particularly the set() method, so that it sets an Alarm to start the AlarmTweetService.

## Submission:

To submit your implementation, save and commit your local changes and push to your origin. Make sure to log into your GitLab account and verify that your changes are there.

## Optional Notes:

In the lab, for the sake of simplicity we have hard-coded the Twitter API tokens. But this should not be done for several reasons. For example: in case of our scenario, anyone could steal those keys and use it to post derogatory comments on the Twitter handle. In the worst-case scenario an attacker can try to leak user data in the form of personal details(Name, Phone numbers, Addresses). For more information, you can see the following links about why keys should be stored securely and what are the risks of using hard-coded keys.

https://www.zdnet.com/article/secret-tokens-found-hard-coded-in-hundreds-of-android-apps/

https://www.lordcodes.com/articles/protecting-secrets-in-an-android-project

https://guides.codepath.com/android/Storing-Secret-Keys-in-Android