

# Alarms and Networking

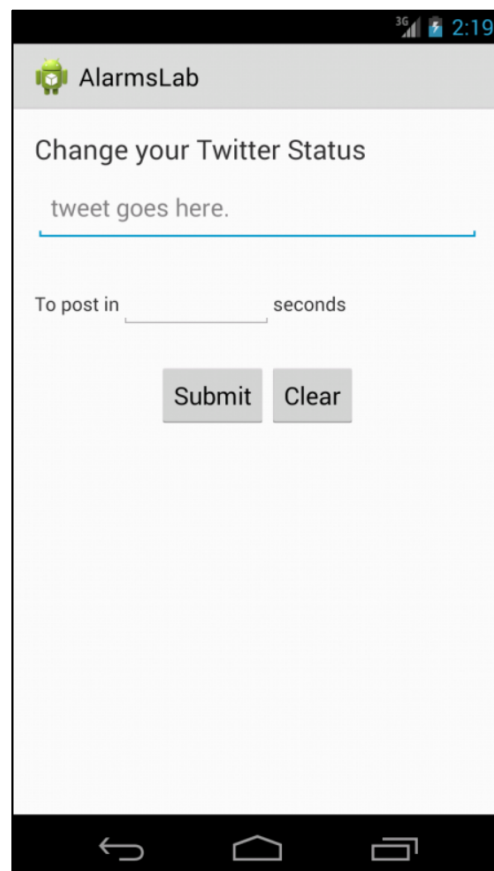
## Objectives:

Familiarize yourself with Alarms and Networking. Create an application that uses Alarms, Networking, and AsyncTasks.

Once you've completed this lab you should have a better understanding of Alarms, Networking and AsyncTasks. 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 AsyncTasks to perform short tasks on a separate thread.

## Overview:

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](http://www.twitter.com) website.

See the screencast of the app in operation.

### Implementation Notes:

- 1) Checkout the Lab6\_Alarms branch from the upstream repo. Make sure to push it to your origin right away before you start implementing any code.
- 2) Modify the code near the top of the AlarmTweetService so that not everyone is using the same Twitter Account. Note that since, we will be sharing several Twitter accounts, you should put some unique identifier on your tweets. Otherwise, you won't be able to distinguish your tweets from those of your classmates. Also modify AlarmTweetService to use an HttpURLConnection.
- 3) Modify AlarmCreateActivity, 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.