

CS543 Team Programming Assignment:

Hashtag Image Sharing Service (HISS)

Due date: Dec. 04. 2016 (Sunday) 11:59 PM

Overall Description:

A hashtag is used widely to label or metadata tag an image on social network and microblogging services. Users create and use hashtags by placing # sign in front of a word or a phrase without spaces.



Figure 1. An example of hashtags

For this programming assignment, you and your teammates are to build a hashtag image sharing service (HISS) that enables a user to receive a notification when an image with a hashtag of the user's interest is newly uploaded.

Sample Scenario:

Alice likes dogs. She wishes to receive a notification whenever a new image of a dog becomes available. Thus, she subscribes to #dog on HISS, which will allow Alice to receive a notification whenever a new image with #dog hashtag becomes available.

Bob takes a photo of his dog and uploads it to HISS with a #dog hashtag. HISS then sends out a notification to Alice, thus allowing Alice to view Bob's image.

Requirements:

- You are required to use publish-subscribe messaging pattern to build HISS. A user subscribes to hashtags of interest and publishes a hashtag when sharing an image. For instance, Alice subscribes to 'dog', and Bob publishes an image using a 'dog' hashtag. It is okay to assume that an image only contains a single hashtag.

If you feel that uploading/downloading image through publish-subscribe messaging pattern is infeasible (e.g. due to large data size), you may implement other methods to upload/download an image and publish/subscribe metadata required to access the image. For example, you could upload an image to the cloud of your choice (or your local server) and publish a link (URL) of the image as a publish-subscribe message

payload for the client to load the URL.

- A user may subscribe to one or more hashtags of interest. A user can also subscribe and unsubscribe to one or more hashtags any time. Once a user unsubscribes to a specific hashtag, she should no longer receive notifications for that hashtag.
- Implementation choices are up to you and your teammates. You could implement end-user clients as Android application, web, windows executable, etc. A broker for publish-subscribe messaging pattern can be implemented as a separate server running on a desktop, laptop, cloud, etc. It is okay to use a third-party library for publish-subscribe communication (i.e. no need to implement broker on your own!).

While you can implement HISS in any way you want to, please keep in mind that TAs must be able to follow your implementation details and demo for grading 😊.

Submission:

- A zip file containing all your source codes (or source repository) and executables for an end-user client and a server running a publish/subscribe broker. Please include a short README to setup and run executables.
- A report (3 pages max, free template) including, but not limited to,
 - Architecture of the HISS with a short description of each component
 - Implementation details
 - Each member's job responsibilities
 - Screenshots

Extra credit (+2):

- A support for multiple hashtags of a single image.
E.g. Bob publishes an image with #dog #puppy. Alice subscribes to #dog #puppy. In such a case, Alice should only receive one notification despite of having multiple hashtags matched.
- Nice, simple and intuitive UI

References (optional):

- MQTT publish-subscribe messaging libraries
 - <https://eclipse.org/paho/>
 - <https://mosquitto.org/>