NanoTwitter 1.0

1. Add more features:
   1) Use Web Socket to send live notifications
   2) Support reply. Reply will send a notification to the target user.
   3) Made Pagination on home page. Click “load more” to load more tweets. The response time is reduced a lot by this! For example, at first we have 1000 tweet id to query on this page, now we only have to query 30 of them each time. (the tweet ids are cached in redis)
   4) We have finished the client library and have implemented more unit tests.
   5) Cache the home page for no logged in users (we haven’t done it before). The home page only display 50 newest tweets, so in Redis, we persist all information of that 50 tweets (not only ids) and the response time is reduced to 10%!

2. We implement our nanotwitter by heavy javascript requests. The page is not generated by pure erb. Actually we only have 3 to 4 erb files, and each file is like a single page application, when you click button, something will happen. This also means that it is impossible to test our app like the other group. We can only provide the API to query the home timeline, and this will only return json, not the entire page. While due to the fact that the bottleneck of the home page is on home timeline query, we think that is ok.