G54CCS Labs, Exercise 5

Previous exercises have been quite tightly directed as to what you need to do and the result you should produce. This exercise is more open-ended and will require you to apply what you have learnt so far to build a simple cloud-hosted guestbook application. It leans heavily on material from exercises #3 and #4 in particular.

Note that this exercise is about trying to apply the cloud computing techniques you have learned so far. It is **not** about producing a heavily styled, graphically appealing website with flashy images, intrusive audio, or extensive use of Javascript.

The Guestbook

The basic guestbook application needs to support essentially two methods:

- *Add an entry.* A user should be able to add an entry to the guestbook, which should be stored along with some user name and the time the entry was added.
- *Retrieve all entries.* A user should be able to visit a page which displays all of the entries so far added to the guestbook, in the order in which they were added.

To implement these core features think about the *data model* (`models.py`) and the changes to the set of fields you require over exercise #4 (storage). Note that you will (obviously) need to store *multiple* entries this time, where the code discussed in exercise #4 ended up storing only a *single* row. Think also about the URLs (`urls.py`) through you wish to expose views on your data (`views.py`). Finally, you may need to make use of slightly more complex page templating that exercise #4 required: read the documentation on GAE page templates at `http://code.google.com/appengine/docs/python/gettingstarted/templates.html`.

Extensions

Having got these first two features — the core of the application — working, consider adding the following extensions:

Ex.1. Allow users to edit and update existing entries.

Ex.2. Store a a full edit history with each entry, and when entries are retrieved for display, allow each entry's full history to be displayed with the entry.

Ex.3. Enable users to vote for entries with which they agree/disagree.

Ex.4. Enable users to discuss and comment on entries and comments in a simple 'threaded' interface.