Polytechnic Institute of NYU April 20, 2013
Computer Science and Engineering
CS 6083, Spring 2013

Project #2 (due before or after the final exam)

In the second project, you have to create a web-based user interface for the *jingo* database designed in the first project. In particular, users should be able to register, create a profile, log in, post notes, and comment on notes they have received. Users should be able to define filters for receiving notes and be able to send and answer friend requests.

Note that you have more freedom in this second project to design your own system. You still have to follow the basic guidelines, but you can choose the actual look and feel of the site, and offer other features that you find useful. In general, design an overall nice and functional system. There will be some extra points available for a nice and smooth design. If you are doing the project in a group, note that both students have to attend the demo and know ALL details of the design. So work together with your partner, not separately. Also, slightly more will be expected if you are working in a team. Start by revising your design from the first project as needed. In general, part of the credit for this project will be given for revising and improving the design in the first project.

A note about the interface(s) you are expected to build for this project. There are really two interfaces. One is the interface that a mobile user would use – in this case, the actual current time and estimated location would be sent to the site to determine which notes are visible, and when a user posts a note, this location would also be used. The other one is an interface based on maps where the user can choose their current location and the current time, and the system would then display the notes visible at that time in that location. This second interface is needed for two reasons: (1) a user may want to post notes about places they find, say, on the internet (like a new restaurant) even when they are not near the place, or might want to explore what notes would be visible if they were to travel to a place, and (2) we want to be able to do the demo without having the TAs run through Brooklyn with you. For the second interface, learn how to use map APIs such as Google Maps or others to allow users to select a (fake) location on a map, and how to user a slider or other gadget to choose a time. For the first project, can you figure out how to get the estimated current location of your laptop, and the current time, so you can supply it to your system? Note that in the database backend, the types of queries run in both interfaces would basically be the same.

Users should be able to perform all operations via a standard web browser. This should be implemented by writing a program that is called by a web server, connects to your database, then calls appropriate stored procedures that you have defined in the database (or maybe send queries), and finally returns the results as a web page. You can implement the interface in several different ways. You may use frameworks such as PHP, Java, Ruby on Rails, or VB to connect to your backend database. Contact TAs Aditya Guntupalli (aguntu01@students.poly.edu) or Shi Li (sli18@students.poly.edu) for technical questions.

Every group is expected to demo their project to one of the TAs at the end of the semester. If you use your own installation, make sure you can access this during the demo. One popular choice is to use a local web server, database, and browser on your laptop, which means you need to bring your own laptop to the demo. (In this case, your project does not have to be available on the public Internet; it is enough to have it run locally on your laptop). Also, one thing to consider is how to keep state for a user session and how to assign URLs to content – it might be desirable if users could bookmark a note, e.g., as a page displaying a particular piece of content or the results of a search (but not sure this makes sense). Grading will be done on the entire project based on what features are supported, how attractive and convenient the system is for users, your project description and documentation (important), and the appropriateness of your design in terms of overall architecture and use of the database system.

Describe and document your design. Log some sessions with your system. Bring your description (documentation) and the logs in hardcopy to the demo. You should also be able to show your source code during the demo. The documentation should consist of 15-20 pages of carefully written text describing and justifying your design and the decisions you made during the implementation, and describing how a user should use your system. Note that your documentation and other materials should cover both Projects 1 and 2, so you should modify and extend your materials from the first project appropriately.