Rahul Rajagopalan
6.170

Network Stickies Design Notes

The major design question left open was where data processing should be done. This is a common question for all web applications that involve communication between a client and a server. It was specified that assuming that users will have JavaScript enabled is valid (see http://piazza.com/class#fall2011/6170/48) Therefore, I decided to use a “rich client” model; the clients would do most data of the processing using JavaScript and the server's job would simply be to persist the data. The primary advantages of this option are faster response time to user input, and less load on the server. The main drawback is that it excludes clients who do not have JavaScript enabled, but this was stated to not be a requirement of the assignment.

For the user interface, I used the Model-View-Controller design pattern. This has the advantage of decoupling the internal data model from the view, and making it relatively simple to add new views without changing the model. It was a natural fit for an application like this that depends heavily on user input.

To keep users from seeing others' notes, I wrote a simple register/login system that runs on the server. This is familiar to users from many other websites, and achieves the goal of making notes private.

To persist the data, I had the client code periodically serialize the NoteSet object backing the data as JSON, and send that data to the server to be saved. When the user reloads the page, the JSON string from before is downloaded from the server, and parsed back into a NoteSet object. This minimizes the amount of work that the server does and is simple to implement in JavaScript, as JSON closely models the structure of JavaScript data objects.

The data was saved using Python's shelve module. This has the advantage of being quite simple to use – its interface is almost exactly the same as a Python dictionary. Another option would have been a SQLite database; it would have probably had better performance, but be more complicated to use. Currently, Network Stickies does not have noticeable performance issues, so shelve suffices.