

6.170 Assignment 4: Network Stickies

Betser, Dina M
username: dbetser

Object Model: 8/8

Comment: You have captured all the objects with the correct level of abstraction

State Machine: 7/7

Comment: Good SM

Design Notes: 16/18

Comment: Good design notes. Some of the below points could have been discussed in this section. How are IDs assigned to stickies (Client and server must agree on ID; also, some kind of global or per-user uniqueness of sticky IDs must be enforced), Saving sticky content -- automatic vs manual, how does communication with server happen. ID reuse, when does the saving to server occur (is it manual or automated)?, Concurrent updates. E.g. website open in 2 tabs. Delete a note in one, then edit that same note in the other tab.

Specification: 10/10

Comment: Good

Implementation - Code: 21/25

Comment: Good, robust approach to the engineering, but you skimmed on some of the details. You should always put HTML attribute values in quotes (even if they're one word), and having a blank action is not valid HTML. It would be nice if one could add more than one note at a time, and it would have been preferable not to reload the page upon updating the sticky. Instead, you should fire off the AJAX request to the server, which will update its state about the sticky, then use JavaScript to update the client's information (for dragging, no additional work would be necessary, since the sticky is now in the correct position)! Your issue with jumping stickies is related to how the .position() method works. If you read the docs, you see that it is relative to the offset's parent (sticky_pane, for you), since you don't specify in your CSS how your elements should be laid out (see the CSS position attribute). You can either use .offset() with absolute positioning or set the position attribute in the sticky_note class. Also, you are violating separation of concerns by storing state about the sticky notes in the DOM (notably, the ID). It would be preferable to use the .data function of jQuery, which lets you associate data with an element in a way that does not pollute the DOM. Finally, always use type="password" when accepting passwords. Good work on the model and automated tests, though!

Code Notes: 4/4

Comment: Good notes

Testing: 10/10

Comment: Good test suite

Implementation - Output: 16/18

Comment: 500 error when you try to edit a note in one window after it has been deleted in another. While this doesn't actually crash the app, it's showing up in the console and should probably be handled more gracefully. When the drag movement of a note ends, it kind of jumps from where it was stopped.

Assignment 4 Grade: 92/100