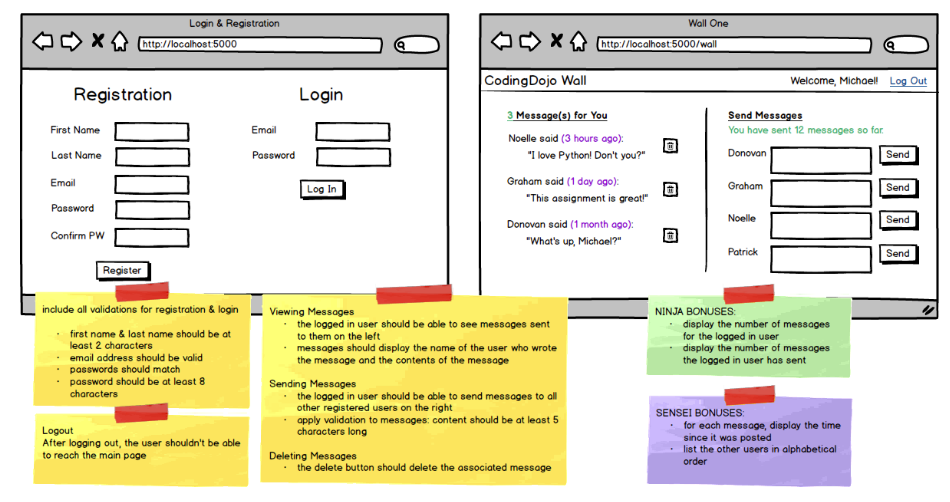
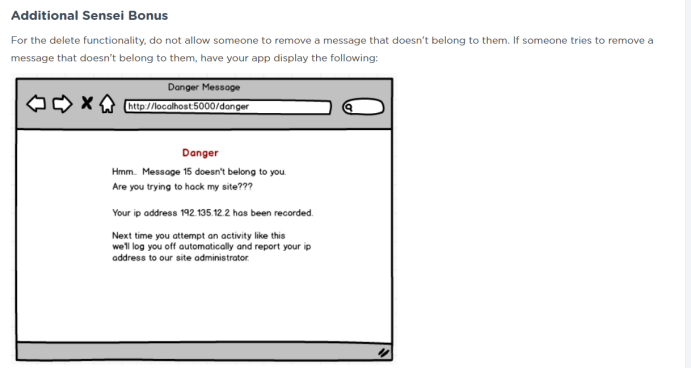
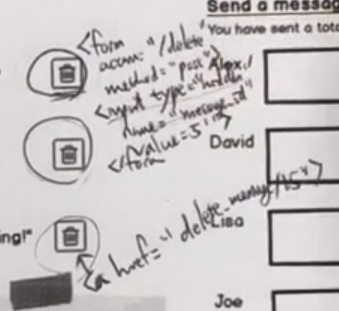
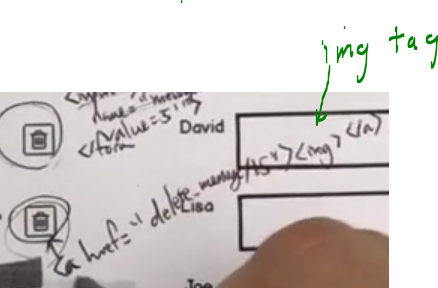
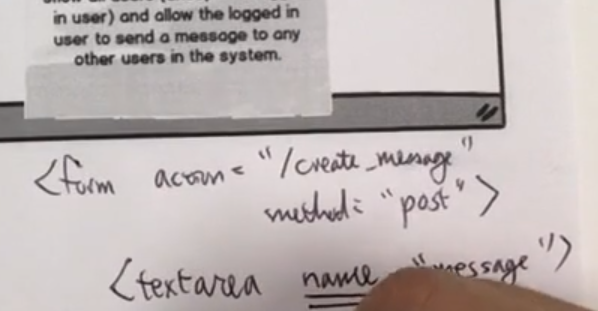
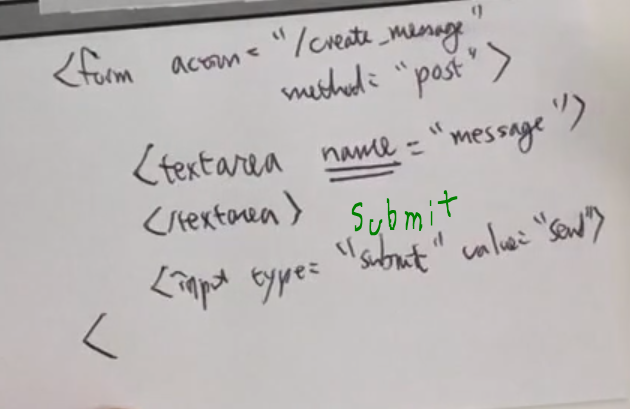
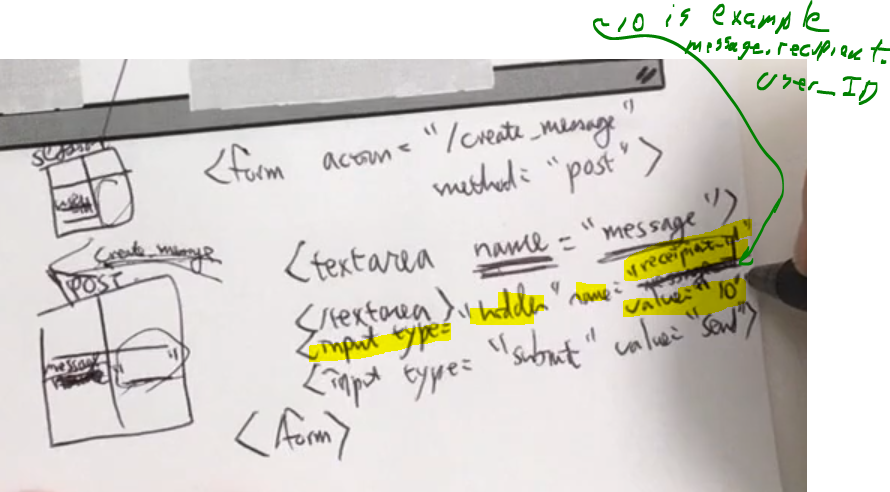
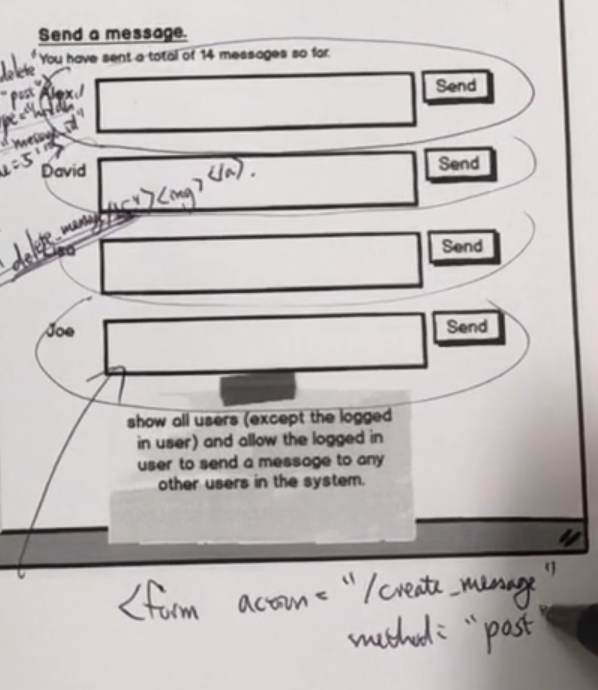
For this assignment, a user's "wall" is a list of their private messages. Once a user has logged in, they can view their wall and, on this same page, the logged in user can also send messages to other users. The yellow sticky notes indicate basic functionality. Review the green sticky notes for Ninja Bonuses and purple sticky notes for Sensei Bonuses.





From Michael’s video

* Recommends anchor tag for deletions
* 
* 
* He says send id of message directly in URL
* Four forms as below. He shows that action goes to/create\_message, method as post
* And a text area for message content:
* 
* 
* In below he adds the recipient ID via a hidden tag:
* 
* 

From Mattermost:

**nina\_t**

[4:21 PM](https://dojo.news/coding-dojo/pl/a1bj4ymmufnzimohg5jiqzty1h)

is anyone working on private wall yet? I wanted to see how your database was designed

I have two tables but before I forward engineer this baby I want to do a sanity check

**mike\_murray**

[4:21 PM](https://dojo.news/coding-dojo/pl/mbsn4j5ra7fit8h5eamm55potr)

i'l literally looking at that right now..

**kent\_h**

[4:25 PM](https://dojo.news/coding-dojo/pl/9cs9ekdy87djtrkjmfmqkwbdwr)

I will look at its db design, now

I would guess at least a users table and a messages table

The below might indicate using more than two tables...because having more than two relationships that are not self joings would mean more than 2 tables:

Include multiple one-to-many relationships

**kent\_h**

[4:38 PM](https://dojo.news/coding-dojo/pl/b3c5pxdjdjgh8bcphhuho7iwae)

In the video Michael talks about how many forms we have, but of course there is only one action per html page, so any of those send buttons really do the same thing....Amazon uses that by scattering "buy now" around their pages

**kent\_h**

[4:46 PM](https://dojo.news/coding-dojo/pl/pqkuduwb37gu3gumssw7r895ph)

I think in the real world it would be better to have one text area and dropdown to pick the recipient. THere are only four send buttons on the wireframe....what if there were 20 or more users...would the page just grow to have that many text areas?

I think the below is related to Michael's idea of having the message ID pass in the URL and thus when bad user figures that out he can try to delete any message:

do not allow someone to remove a message that doesn't belong to them. If someone tries to remove a message that doesn't belong to them, have your app display the following:

because the entire message displays on the "Wall One" page, it would seem it also could cause the page to grow excessively long...unless they are all inside of a div that has overflow with scroll bars...if that works

Still thinking about table count...

I do think Michael might want more than two tables, but I can't imagine what they would be....users...the regular stuff specific to a user...if we are tracking their postal address, then that would be another table, but I don't think so. The message table has a list of messages. Each has a sender and a receiver and content and date created...I don't know what else. If deleting a message just hides it forever, then another field to show messages are archived.

**kent\_h**

[5:11 PM](https://dojo.news/coding-dojo/pl/ms1yrtzanjd5djyb4qo798dzio)

Multiple forms with only one page send...how to know who is the recipient...

this page discusses: <https://stackoverflow.com/questions/547821/two-submit-buttons-in-one-form>

Stack Overflow

[**Two submit buttons in one form**](https://stackoverflow.com/questions/547821/two-submit-buttons-in-one-form)

I have two submit buttons in a form. How do I determine which one was hit serverside?



I hope users are smart enough to not populate two text boxes then click a third submit

**kent\_h**

[5:27 PM](https://dojo.news/coding-dojo/pl/wuxe5xqiijya5f778dftrekmur)

On the submit...Jan's answer on this page seems to match Michael's. Of course we get to iterate each form dynamically

<form method="post" action=""> <input type="hidden" name="product\_id" value="123"> <button type="submit" name="action" value="add\_to\_cart">Add to Cart</button> </form>

<https://stackoverflow.com/questions/8712398/multiple-forms-or-multiple-submits-in-a-page>

Stack Overflow

[**Multiple Forms or Multiple Submits in a Page?**](https://stackoverflow.com/questions/8712398/multiple-forms-or-multiple-submits-in-a-page)

I'm creating a page with the products sold in the website. I'd like to include an "add to cart" button near each product, which are listed with markup similar to this: &lt;h4 class="productHeading...

=-=-=-=-

**instructor\_wes**

[6:54 PM](https://dojo.news/coding-dojo/pl/hm37sktbpfb65jew4f58wgz4ew)

So @mike\_murray if you can help communicate to the group here. I'm not entirely sure how everyone's going about this query or who all is involved in this conversation at this point

But I think @ta\_nick is right

Having a table for "wall"doesn't quite seem necessary

Yeah I just got pulled away for a bit and I'll get set back up here shortly to replicate errors.

**instructor\_wes**

[6:55 PM](https://dojo.news/coding-dojo/pl/watpswypgpnrzq51uqyfic6uuo)

So the way I like to think about this private wall problem is that the database is just going to store all the data we absolutely need to store. We can then present that data in any way we'd like

If we want this page to look like a "wall", we can do so just by having access to users and messages

So the only thing we need to store with the message is the content of the message as well as the sender and the recipient. If we have that info available to us, we can do some pretty complex things with our user interface to make it look like there is a wall

**mike\_murray**

[6:58 PM](https://dojo.news/coding-dojo/pl/i3uosqwexidr7yiotyuxgfuxpc)

It originated when we were trying to identify what would he on what tables. How we could capture 4 elements of the wall. Owner, message, time and sender. Message and time easy. Link to one owner or to one sender. Ultimately we agreed to do single user(owner) to single wall. Single wall can have multiple messages. And single message to single user(author)

**image-7a0e4646-ee21-4dbb-bce5-e786d57d2c6f.jpg**

**instructor\_wes**

[6:58 PM](https://dojo.news/coding-dojo/pl/ouxsguta7by73cxj7eaazawxhy)

So the sender and recipient can both be one to many relationships between users and messages

you can have 2 separate one to many relationships between the same 2 tables

**mike\_murray**

[6:59 PM](https://dojo.news/coding-dojo/pl/r34y6jxctfr3tjgfb8eoc1sngc)

Ahh

**instructor\_wes**

[6:59 PM](https://dojo.news/coding-dojo/pl/dwto1zxjd78ptgcwe1krm4jfuo)

this wall thing works, but the wall needs to know exactly which user it's connected to

So I think this schema that you've shown is totally fine

Either way, the SQL JOINs are the hardest part

[7:00 PM](https://dojo.news/coding-dojo/pl/7zq7o8he3ifxpgaz94fmt71fpc)

Am I clearing anything up for you and the team or am I just muddling everything?

-==-=-=-=

Wes,

I essentially posted this earlier in Irvine channel.

Pasting summary here in case you prefer the smaller group

I am planning two: messages and users, if I need more to satisfy the multiple relationships requirement, then something like an address table for users. Does that sound okay?

I was confused about mention of a wall table.

Sorry about brevity. I am on phone at the Java meetup

**instructor\_wes**

[7:10 PM](https://dojo.news/coding-dojo/pl/yaksfysibtbmfp6x84dzee8m9o)

need to look into the multiple relationships question

Hey Kent you'll need to elaborate on the multiple relationships requirement you're referencing

You shouldn't need more than users and messages

**kent\_h**

[7:16 PM](https://dojo.news/coding-dojo/pl/t1a1tq393j8d38dqiras8sr9fw)

Ok. Stand by

[7:18 PM](https://dojo.news/coding-dojo/pl/jartwidebtr338uwetizxcx4yc)

This is a quotation from assignment:  
Include multiple one-to-many relationships

I think/hope a self join of two relationships solves

Each message connects to user table twice.

**instructor\_wes**

[7:22 PM](https://dojo.news/coding-dojo/pl/d69dhefhcb8t9du663aq5nowzw)

multiple one-to-many is just between users and messages

self join I suppose but not really because messages isn't a table whose sole responsibility is to connect users together

But yeah the diagram is similar to a self-join

=-=-=-=-=-=-

