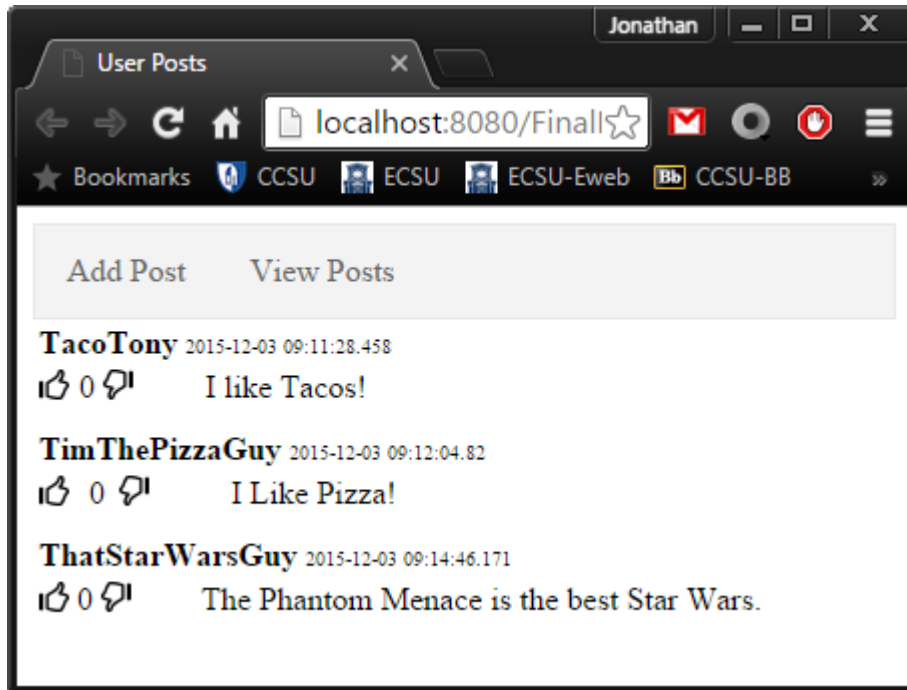
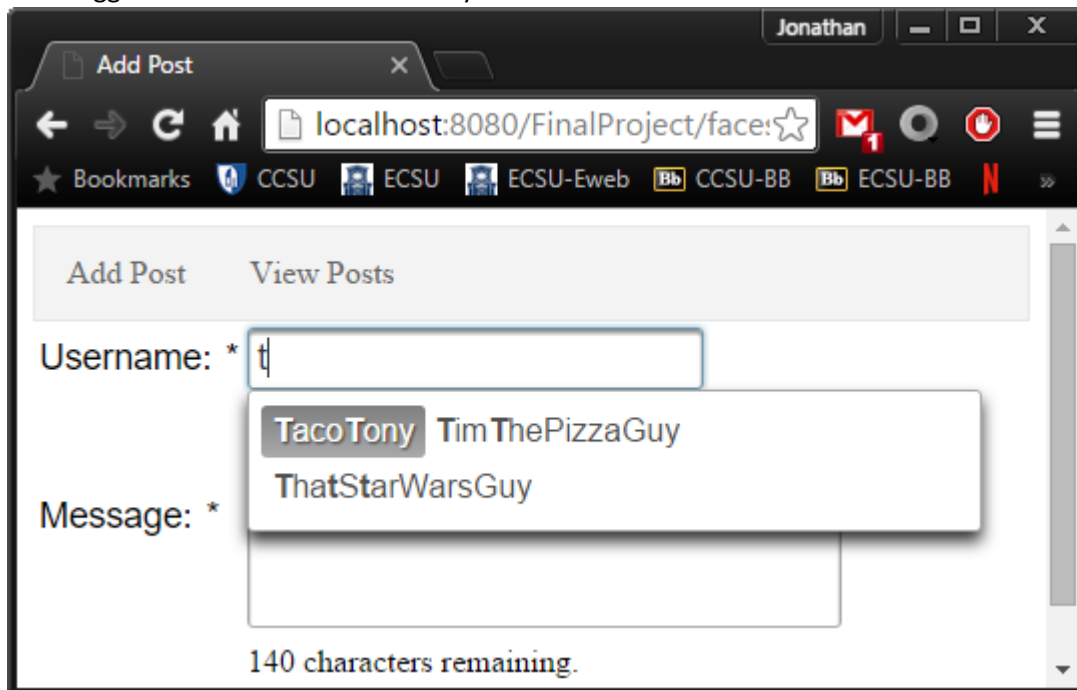


My web app is simple social media app where users can post messages with a username of their choosing.

1. When first visiting the site, the user is greeted by a list show all of the posts by other users.



2. Clicking the “Add Post” link in the top menu bar will direct the user to a page where they can create a post of their own. While typing a user name, AJAX is used to autocomplete the field with suggested usernames that already exist in the database.



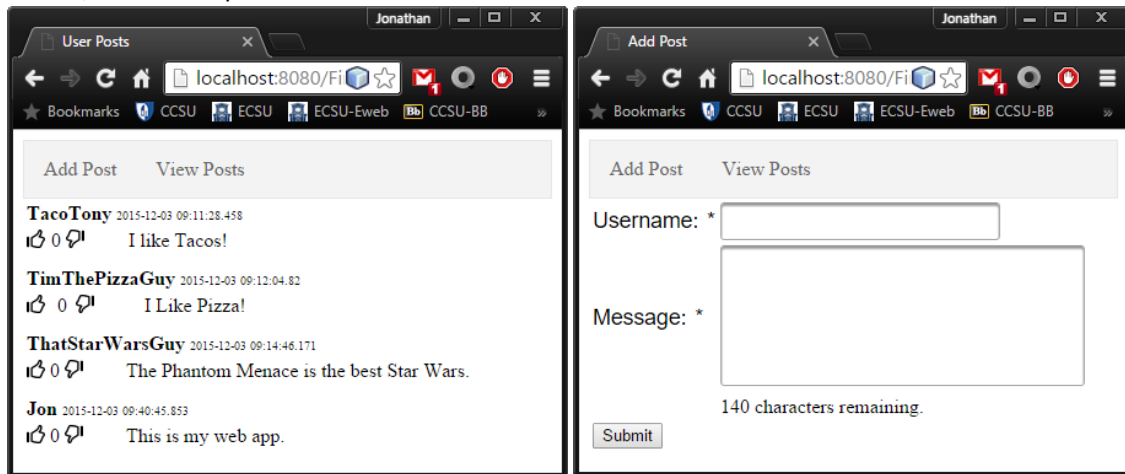
A combination of built in and custom validation are enforced when entering a username. A username must be between 3 and 15 characters as well as only be made up of alphanumeric characters. The user's message uses built in validation to only allow 140 characters.

The left screenshot shows the 'Add Post' form with a blue information message: 'Username: username may not contain spaces or special characters and must be between 3 and 15 characters long.' The username field contains 'Jon#\$%#@' and the message field contains 'Hi'. A red error message at the bottom says '138 characters remaining.' The right screenshot shows a red error message: 'Message: Validation Error: Length is less than allowable minimum of '2''. The username field contains 'Jon' and the message field contains '!'. A red error message at the bottom says '139 characters remaining.'

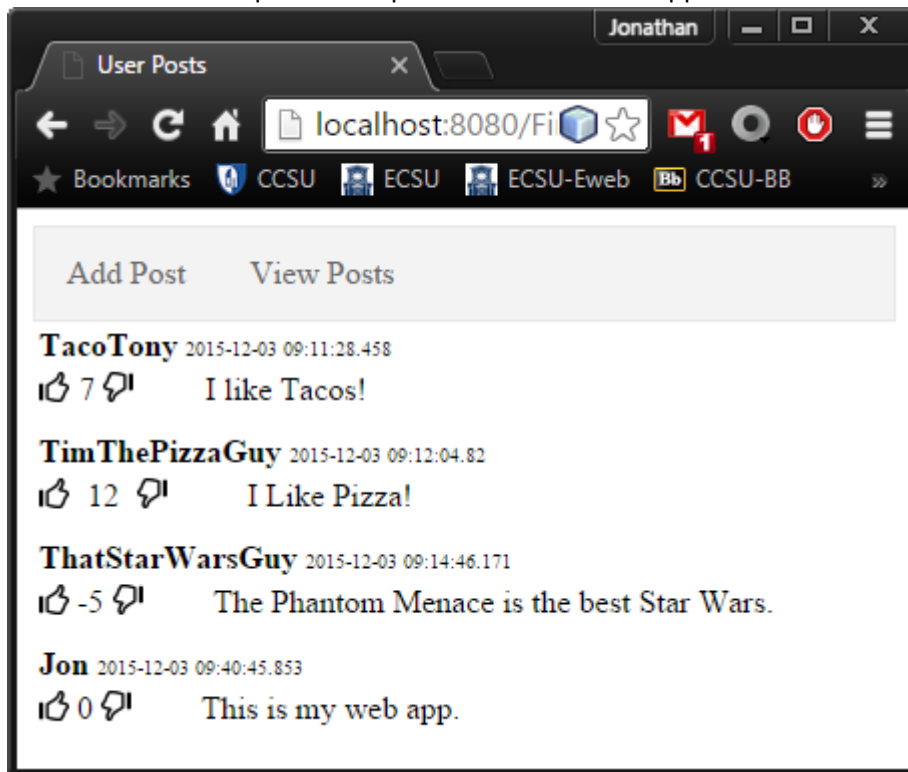
3. Once the user clicks the submit button, the user is brought to a page to confirm the post was successful. If something went wrong during the post creation then the user will be brought to an error page.

The left screenshot shows the 'Post Saved' page with a green message: 'Post saved successfully!'. Below it, the post details are displayed: 'Username: Jon', 'Message: This is my web app.', and 'Date posted: 2015-12-03 09:40:45.853'. The right screenshot shows an error page with a red message: 'ERROR!'.

4. The user can the menu bar to then select “Add Post”, to add another new post, or select “View Posts”, to view all posts.



5. Users may also “like” or “dislike” posts by other users by clicking the “thumbs-up” or “thumbs-down” buttons for a post. This updates a counter that appears between the two buttons.



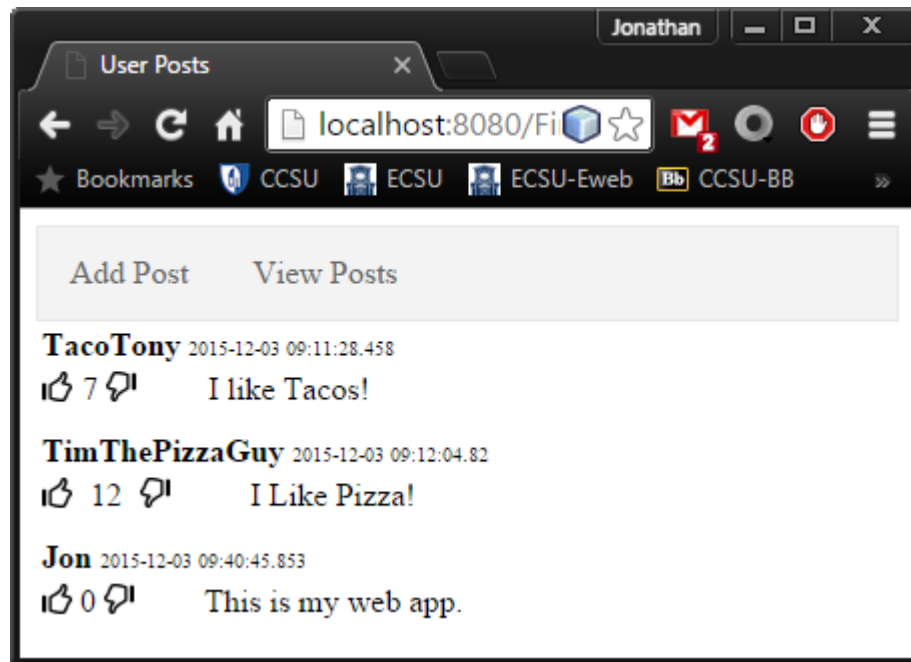
If a post receives too many dislikes the user's message will be deleted.

Before delete:

#	ID	USERNAME
1	1	TacoTony
2	3	TimThePizzaGuy
3	5	ThatStarWarsGuy
4	7	Jon

#	ID	DATEPOSTED	MESSAGE	NUMOFLIKES	SITEUSER
1		2 2015-12-03 09:11:28.458	I like Tacos!	7	1
2		4 2015-12-03 09:12:04.82	I Like Pizza!	12	3
3		6 2015-12-03 09:14:46.171	The Phantom Menace is the best Star Wars.	-5	5
4		8 2015-12-03 09:40:45.853	This is my web app.	0	7

After delete:



#	ID	USERNAME
1	1	TacoTony
2	3	TimThePizzaGuy
3	5	ThatStarWarsGuy
4	7	Jon

#	ID	DATEPOSTED	MESSAGE	NUMOFLIKES	SITEUSER
1		2 2015-12-03 09:11:28.458	I like Tacos!	7	1
2		4 2015-12-03 09:12:04.82	I Like Pizza!	12	3
3		8 2015-12-03 09:40:45.853	This is my web app.	0	7

Core requirements:

- **JPA entity beans for the model with at least one mapping of either many-to-many or one-to-many**
 - There is a one to many relationship between SiteUsers and Messages. A SiteUser bean may have one to many Messages.
- **Through application at least one place where you demonstrate understanding of each of these: create, update, delete, find (does not have to be same place)**
 - Create: A user can create a username and a message to be added to the database.
 - Update: Liking or disliking a post updates the NumOfLikes value in the Message table for the appropriate message.
 - Delete: A message will get deleted from the Message table if its NumOfLikes value reaches a specific value (for demoing purposes the value is -10).
 - Find: AJAX username autocomplete if user wishes to reuse an existing username in the database.
- **A complex flow element – a place where hitting “submit” on the page could take you one of several possible locations and the location must be determined by business logic (Class example was loan status determining whether to ask for more information about people, or home, or submit and save and show confirmation)**
 - Upon submitting a post the user can either be directed to a confirmation page or an error page.
 - The user may use the menu bar at the top of each page to navigate the app.
- **Use of validation both built in and custom validation through your chosen framework (i.e. JSF or Spring)**
 - Built in validation was used for the username field to limit the user from entering more than 15 characters.
 - Custom validation was used to only allow between 3 and 15 alphanumeric characters to be entered for a username.
 - Built in validation was used to limit a message between 2 and 140 characters.
- **AJAX**
 - Username field autocomplete.
- **Good MVC practices**
 - Model: SiteUser and Message managed entity beans.
 - View: addPost.xhtml, userPosts.xhtml, postSaved.xhtml, error.xhtml.
 - Controller: postController.java.