CMSI 370-01
INTERACTION DESIGN
Fall 2013

Assignment 1024 Feedback

Abdulrahman M. Alzaid *AbdulZaid*

3a — You have "one and a half" successful Ajax implementations, but the rest of your code signals that aspects of writing this type of code may still be unclear. The multiple calls to `GET /characters` are puzzling, and your `DELETE /character/:id` URL, if it worked correctly, would perform the delete at the database level but would not update the user interface properly. Plus, even if the requested Ajax implementations number just 2, trying out the others would still help communicating your understanding of this process. As of now, it looks like it can still use some improvement. (/)

3b — You don't have much here in terms of event-handling. You can grow this by expanding the functionality of your user interface. As mentioned before, even as a mockup there's a lot that you can do. Much more so now that there *is* a web service running behind all of these. (/)

3c — MVC understanding is the same as before, because it is the same code. (+)

4a — As mentioned previously, you only have one fully functional Ajax call. You also call `GET /characters` at least once more, for unknown reasons (but my guess is misunderstanding on what is fully going on). The problem with DELETE is not the network but an actual flaw in your choices for the way a character ID is tracked, then subsequently, if this is fixed, you need to give the user some visual feedback on the successful deletion. These are the main Ajax-related issues; additional notes can be found inline. (/)

4b — Separation of concerns remains the same as in the previous assignment, as it is the same code. (|)

4c — The formatting issues in your code have already been pointed out. They bear repeating here: "Your code starts out OK in terms of look and style, but boy does it deteriorate further down. Plus, you have tabs mixed into your code; stick to all spaces." (/)

4d — You did a good job with looking things up for the proper Ajax invocations. Now you need to learn a bit more about tracking the data properly, then updating the user interface appropriately when an Ajax response arrives. (+)

4e — Commit frequency and messages are appropriate for the work done. (+)

4f — Submitted to GitHub on time; *my.cs.lmu.edu* site also live at time of grading. (+)

Updated feedback based on commits up to 12/12/2013; only re-reviewed outcomes are included:

3a — You have certainly cleaned up many of your "back end" Ajax code; what's missing is the front-end. Most of your updates require a manual reload by the user in order for the changes to be seen. The ability to appropriately refresh your user interface when the data behind it change remains a significant part of knowing how user interfaces are constructed. (|)

3b — Your added Ajax functionality has improved the amount of event-handling that you do. Now you need to do the *right* event handling. You have the user interface events down, but not the Ajax callbacks: for example, PUT and DELETE don't do anything to the user interface when they return. They should. (|)

4a — Ajax gaps and that extra `GET /characters` call have been addressed. The aforementioned character ID tracking bug is there, now also affecting PUT, and as already mentioned you need to update the user interface when your Ajax calls return. (|)

4b — Separation of concerns remains the same as in the previous assignment, as it is the same code. (|)

4c — As with *4b*, this outcome gets the same proficiency as before, because it is the same code. (/)