

# MyCritic Defect Logs

Anubhav Saxena, Avadhoot Joshi, Brandon Kent, Jacob Conley, Jordan Hagedorn

## Design Inspection Defects

Defect #	Description	Severity	Solution	Who Found/Fixed
1	Clickable text does not change cursor style on hover, making it look like plain, non-hyperlinked text	3	Added onmouseover cursor effects to linked text	Anubhav
2	Game images download instead of displaying when embedded in game page	3	Discarded game images	Anubhav
3	Ratings have many decimal places, unnecessary to understand the rating	3	Rounded all rating scores received from API to zero decimal places	Avadhoot
4	On the search page, if an element had not been rated yet it showed undefined	3	If an element is null set rating to No Rating	Jacob
5	On the user page, the users review didn't have a title for the review so you didn't know what it was for.	2	Modify the database to include a title element in the reviews table	Jacob
6	Media rating numbers were very ambiguous and hard to understand	3	Added text instead of numbers, and added a score out of 5 based on the index of the text selected	Anubhav
7	After deleting an account they can still click on myprofile and logout	3	Delete their session or log them out	Jacob
8	After a user is deleted, you can manually go to their userpage and see the text "'s Reviews"	3	Delete reviews for a deleted user from the reviews table and have a check for if a user exists	Jacob
9	Text in front of images is often not visible because of the color and size	3	Changed font colors, sizes, and animations, to	Avadhoot

	formatting		make sure they are visible in all situations	
11	Certain clickable text is not underlined, making it impossible to know that it is clickable, such as the username in a review	2	Made clickable text to be underlined in most situations, to indicate they are links	Avadhoot
12	The cover image provided by the game API was tiny, and upon embedding, downloaded instead of displaying	3	The API-received game id was parsed and sent to a separate URL to retrieve a larger image for the game item	Jordan Anubhav
13	The reviewForm was accessible by any user, whether or not they were logged in, causing crashes when they submitted reviews	2	Add conditional in EJS to only display the form if logged in	Jordan
14	Certain API queries were slow and impossible to cache as they took place client side	3	Move all API functionality to the node server and generate XML calls to interact with them from the client	Jacob Anubhav Jordan
15	The feed list did not allow any way to get from the list to the reviewers profile page.	3	Made the feed list objects clickable and redirect to the reviewer's page	Brandon
16	Users were able to follow themselves and thus had their own reviews on the feed.	3	Added a check to make sure they could not follow themselves	Brandon/Jacob
17	Users were able to follow another user multiple times and thus create duplicates in the database.	3	Made the follow/unfollow buttons immediately change so they could not be clicked multiple times.	Jacob

## Code Inspection Defects

Defect #	Description	Severity	Solution	Who Found/Fixed
----------	-------------	----------	----------	-----------------

1	If a search result doesn't exist for a media type none of the media types would output their results	2	Have a check before displaying the output that the result exists	Jacob
2	For changing passwords it asks for email and password allowing the potential to change anyone's username and password	1	Remove the field asking for email and use the session email that's stored.	Jacob
3	If a user is not logged in, the reviewForm attempts to load and append itself, but silently crashes	2	Added a check for the content of reviewForm on the client side, and only attempted to append if a user was logged in	Anubhav
4	The song list sometimes crashes silently and sometimes takes a long time to load, with no way for a user to know if it is slow or has crashed	2	Added loading text between the api fetch and display, added a thrown error upon failing	Anubhav
5	The reviewForm did not pass the title of the media to the server when adding reviews	2	Add a hidden form element that extracts the title and sends it with the users review	Jordan
6	While processing the reviewForm in the server, extracting the username caused problems with Node asynchronicity, making reviews have null authors	2	Add a promise function to ensure the username is found before submitting the review	Jordan
7	The caching server was caching results before the entire JSON response was processed, making later calls to the cache incorrect and crashing because of parse failures	1	Made an array of promises and upon resolution of them all, the cache is updated with the complete value	Jordan
8	The unfollow button will remain as the only option on another user's profile even after it has been clicked and they are no longer being followed in the database.	2	An empty array was being treated as a string and thus causing an if statement that was comparing the array's length to misbehave. Changed the array to be treated properly.	Brandon/Jacob
9	The feed list would load in up to 20	2	Added an additional	Brandon

	objects and would continue to try to access more even if there were less than 20, causing the server to crash.		boolean that would stop creating more feed objects once there were no more to be read in.	
--	--	--	---	--

## Unit Testing Defects

Unit tests were done using TestCafe, This mimics user interaction and asserts for the expected outcome.

Defect #	Description	Severity	Solution	Who Found/Fixed
1	Asserting the average score on the search results failed because the score wasn't the expected score	3	The loop was selecting the wrong average, changed it to select the right one.	Jacob
2	After calling an API request, succeeding similar calls did not pull the data from the cache	3	The variables from setting the cache and getting from the cache needed to be the same	Jordan
3	The Assertion that the filter selected displays the right type failed	3	Changed the selection values to return the right media type	Jacob