

# Gleam

Design Inspection, Code Inspection, Unit Testing

Group 25

Derek Shu, Jie (Claire) Li , Emily Ou, Shafer Hess

## Design Inspection

Date: September 19th, 2019

Author: Development Team

Defect #	Description	Severity	How Corrected	Automatic
1	Users could access "My Posts" without error message while not logged in	Important	Respective page now displays a "Please Login" message to access the page	No
2	Users could access "My Bookmarks" without error message while not logged in	Important	Respective page now displays a "Please Login" message to access the page	No
3	Users could access "My Comments" without error message while not logged in	Important	Respective page now displays a "Please Login" message to access the page	No
4	Users could submit new posts while not logged in	Important	Disabled "New Post" button if no user session is active (user is not logged in)	No
5	Users could click "My Profile" dropdown while not logged in	Important	Disabled the "My Profile" dropdown unless there is a user session active	No

6	Users could access an invalid URL without error message for our application	Workaround	Set unmatched and undefined routes to display a "404" page	No
7	Information was being sent to the API through GET params (query string)	Critical	Changed method of sending information from query string of GET to body of POST	No
8	Changing to other pages results in post list with wrong index range. (After redirected to other pages, posts appeared in previous pages show again)	Important	Deferred paginating results to backend API and send pagination information back to front-end to avoid overlap.	No
9	No error message for a failed user login (user not found)	Workaround	Added a "Incorrect username or password" error if user is not found	No

## Code Inspection

Date: September 19th, 2019

Author: Development Team

Defect #	Description	Severity	How Corrected	Automatic
1	Variables in various source files were created but not used: "Divider", "Message", etc.	Workaround	Removed unnecessary variables from source code	No
2	Components in various source files were imported but not used: "PostList", "Input"	Workaround	Removed "PostList" and "Input" from imports	No
3	Functions in various source files were created but never called: "handleReset"	Workaround	Deprecated unnecessary functions that were no longer required	No
4	MyPost:: Use GET instead of POST to query postList of a specific user, which requires password and username	Critical	change GET to POST	No
5	PostContent: fetch function use wrong url: <code>`http://13.58.109.119:3001/view`</code>	Critical	change the url from <code>`http://13.58.109.119:3001/view`</code> to <code>`http://13.58.109.119:3001/posts/view`</code>	No

6	fetchCommentList() is called without constraint, which results in endless function invocations	Important	use content variable as flag: content is set originally to undefined and then changed according to the response. Only when content===undefined, is the fetchCommentList() called	No
7	PostContent: the index used to parse url in slice() function is wrong: an extra ':' is generated in postId	Critical	increase the index by 1	No
8	Post results were not being updated when user pressed "Enter" on the search bar	Important	Forced React to re-render post list on receiving an "Enter" press event	No
9	Post results were not being updated when pagination page was updated	Important	Forced React to re-render post list on page change by adding a representative state variable	No

10	"Sign In" button was still displayed after user successfully logged in	Important	Added logic to conditionally render the "Sign In" button or "Log Out" button	No
11	User welcome message in the navigation bar was not displaying after login	Workaround	Added logic to store username for the current session in order to display the welcome message	No
12	New post submissions by a user would not show up under "My Posts" until page was refreshed because the render preempted the database update	Workaround	Restructured execution flow to only redirect to "My Pages" once the post was successfully submitted	No
13	Inefficient querying of homepage posts.	Workaround	Moved Post querying outside of conditionals to reduce the number of queries made to database	No

## Unit Testing

### Back-End

Module: user\_controller

Date: September 19th, 2019

Author: Emily Ou

Defect #	Description	Input	Output	Severity	How Corrected	Automatic
1	Default 'index' route can retrieve all users in database in production	GET request to /users endpoint	Actual: Returns all users in database  Expected: No results in production database	Important	Added check for rails environment, No longer returns users in production	No
2	User would be created even if username already existed in the database	POST request with username and password payload	Actual: User is created  Expected: Failed to create user	Critical	Added check to see if username already exists during registration, if so send error to front end	No

Module: posts\_controller

Date: September 19th, 2019

Author: Shafer Hess and Emily Ou

Defect #	Description	Input	Output	Severity	How Corrected	Automatic
1	User entry retrieved from database is not checked to be valid user before creating post	POST request body with new post parameters (username, password, title, content)	Actual: Post is created without validating user  Expected: error message if the user is not validated, success message if user is	Critical	Created the is_validated method in the user model to verify user in database	No
2	Request with pageType == bookmarks were not being properly routed	POST payload with bookmarks pageType	Actual: API responds without any content  Expected: API returns user bookmarks	Critical	Fixed conditionals and updated processing of pageType variable in the posts_controller file	No
3	No pagination on requests with searchQuery attached	POST payload with searchQuery != ""	Actual: API internal server error  Expected:	Important	Fixed syntax error in data accessing. data["pageNumber"] -> data["pageNumber"]	No

			Results filtered with query, paginated			
4	Pagination not working in bookmarks request with attached search query	POST payload with pageType = bookmarks and searchQuery != ""	Actual: Pagination error and Internal Server Error  Expected: all User bookmarks paginated and filtered by query	Critical	Bookmark retrieval and pagination refactored to paginate_by_sql instead of find_by_sql. Allowed for more detailed queries	No
5	When searching for bookmarks with a query there is a 'too few arguments' error for the find_by_sql function	POST payload with pageType = bookmarks and searchQuery != ""	Actual: Internal Server Error  Expected: all User bookmarks paginated	Critical	Switched from using variables to store the SQL query and added the query directly as an argument	No
6	The 'view' route for retrieving post data doesn't verify proper post_id	POST payload with post_id = a selected post	Actual: No post error or incorrect content  Expected: Error if post doesn't exist	Workaround - Posts are stored and sent by dev front end	Ongoing	No



7	The 'view' route does not properly increment post view count	POST payload with pos_id = a selected post	Actual: Post view count is static  Expected: views is incremented every click	Important	Ongoing - solution in development	No
---	--	--	---	-----------	-----------------------------------	----

Module: bookmarks\_controller

Date: September 19th, 2019

Author: Shafer Hess

Defect #	Description	Input	Output	Severity	How Corrected	Automatic
1	When retrieving user object, code never checks for null response	POST payload with username password and post_id	Actual: No response  Expected: Error message	Important	Added check for is_validated before attempting to create bookmark entry	No
2	Adding a bookmark never checks if the user has previously registered	POST payload with the username of the user and the post_id they're bookmarking	Actual: No error message for duplicate bookmarks  Expected: Error message if bookmark already exists	Important	Ongoing - Query to check if user_id and post_id match already exists in development	No

## Front-End

Module: Login/Registration UI

Date: September 19th, 2019

Author: Derek Shu

Defect #	Description	Input	Output	Severity	How Corrected	Automatic
1	Users could submit blank username and password fields on registration	Blank inputs	Actual: Registration accepted  Expected: Failed registration	Critical	Disabled registration submission button unless both fields are populated	No
2	Users could submit blank username and password fields on login	Blank inputs	Actual: Login accepted  Expected: Failed login	Critical	Disabled login submission button unless both fields are populated	No
3	Wrong message was being displayed for registration vs. login	Alternate between clicking "Login" and "Register"	Actual: Displayed "Invalid Login" for registration  Expected: Display "Invalid Username" for registration	Workaround	Added state logic to differentiate between "login" and "register" so the correct message would be displayed	No

## Module: 'My Posts' Page

Date: September 19th, 2019

Author: Derek Shu and Jie Li

Defect #	Description	Input	Output	Severity	How Corrected	Automatic
1	My Posts page would not paginate results	POST request with username, password, and pageNumber and pageType	Actual: API would return 10 result count and not paginate past 10 results Expected: API returns total number of posts and returns 10 at a time	Critical	Corrected total post counting commands to accurately reflect total user posts	No
2	My Posts page would not properly search for user query	Any search query into the search bar	Actual: All posts the user has made were returned regardless of query  Expected: Results are filtered by keywords in title	Critical	Updated SQL query to include 'LIKE' clause to accurately search titles	No
3	Loading in the fetched posts would cause an unlikable jarring UI effect	Loading of any list of posts at any point of execution	Actual: Space between top and bottom pagination would shrink and expand  Expected: Have a placeholder loader rendered until all the necessary data has been retrieved	Workaround	Updated returned JSX to conditionally render placeholders or the post list	No

## Module: 'My Bookmarks' Page

Date: September 19th, 2019

Author: Derek Shu and Jie Li

Defect #	Description	Input	Output	Severity	How Corrected	
1	My Bookmarks page would not paginate results	POST request with username, password, and pageNumber and pageType	Actual: API would return 10 result count and not paginate past 10 results Expected: API returns total number of posts and returns 10 at a time	Critical	Corrected total post counting commands to accurately reflect total user posts	No
2	My Bookmarks page would not properly search for user query	Any search query into the search bar	Actual: All posts the user has made were returned regardless of query  Expected: Results are filtered by keywords in title	Critical	Updated SQL query to include 'LIKE' clause to accurately search titles	No
3	Loading in the fetched posts would cause an unlikable jarring UI effect	Loading of any list of posts at any point of execution	Actual: Space between top and bottom pagination would shrink and expand  Expected:	Workaround	Updated returned JSX to conditionally render placeholders or the post list	No

			Have a placeholder loader rendered until all the necessary data has been retrieved			
--	--	--	--	--	--	--

Module: 'View Post' Page

Date: September 19th, 2019

Author: Jie Li

Defect #	Description	Input	Output	Severity	How Corrected	Automatic
1	The post content does not correspond to the current postId because it did not catch record's postId correctly	postId	Actual: It shows the content of the n <sup>th</sup> post in the post list of the first page  Expected: Show the content of the post with the input postId	Critical	take use of the forEach loop in PostList component, which contains the post information (including postId) of the clicked row	No
2	Cannot view post detail without logging in	postId	Actual: Showed a blank page  Expected: Show the corresponding post content even without logging in	Critical	Delete the username & password checking in post controller	No