

Design Inspection, Code Inspection and Unit Testing

- Team 19

SquareCI

Members: Brandon Marx, Jonah Heeren, Jun Soo Kim, Shulin Ye, Vritant Bhardwaj

1. Design Inspection Defects

Product	Square CI		
Date	September 19th, 2017		
Authors	Vritant Bhardwaj, Jonah Heeren		
Moderators	Brandon Marx		
Inspectors	Jun Soo Kim, Shulin Ye		
Recorders	Vritant Bhardwaj, Jonah Heeren		
Defect #	Description	Severity	How Corrected
1	The server allowed CORS as the client was running at a different port. This shouldn't be the case as it is insecure.	1	Changed the client's proxy to the server's port.
2	The access token received from GitHub was being stored in the client and not securely in the server.	2	Changed to using session ID in the client and matched it with the access token from GitHub in the server.
3	The database's config in config.json contains secure information and should not be pushed to the repository	2	Added config.json to .gitignore
4	The node modules were heavy and taking up a lot of space in the git repository with every commit.	3	Added the directory containing the node modules to gitignore.
5	The sessionID was not being refreshed, and so anyone on another system could access a user's	1	Made session ID have a refresh time, after which it would become invalid for a

	information..		user.
6	Interacting with the GitHub API for user authentication was not secure as a response from a random server to the same endpoint couldn't be differentiated.	2	Added a randomly generated string to the request sent to GitHub API and checked for it on receiving a response.
7	Asking the server for user information using the user's email as the key should not be done.	1	Switched to using a unique session ID to assure the security of the user's information.
8	The Client Secret for the GitHub API should not be stored on the client as it is the only key that GitHub identifies our app with, and shouldn't be accessible to others.	1	The server stores the Client Secret in the environment and communicates with the GitHub API to send and receive data.
9	The user shouldn't be directed directly to the GitHub login authorisation page from the root route, so that on logging out they can be directed to the root route.	3	Changed the GitHub login authorisation redirection to happen at the '/login' route.
10	The user's data shouldn't have to be asked from the server by all routes using the session ID.	3	Created a global variable that exists across the routes that holds user information and persists using the browser's local storage.
11	The repository pulled from the GitHub branch is incorrectly stored and can't be accessed in order to run test cases	1	Create a standardized path variable so all other modules no where to access the stored code.

2. Code Inspection Defects

Product	Square CI
Date	September 20th, 2017
Authors	Jun Soo Kim, Brandon Marx

Moderators	Shulin Ye		
Inspectors	Vritant Bhardwaj, Jonah Heeren		
Recorders	Jun Soo Kim, Brandon Marx		
Defect #	Description	Severity	How Corrected
1	The responses received in the client router from the server were not being checked before being used.	1	Added type checking as well as checks for proper structure of the objects
2	React component's state was being changed by directly changing value of the state variable, and so it wouldn't activate an event.	2	The setState method was called instead to actively update the DOM.
3	When sending a request for data from the client to the server, the response was stored right after calling the request, which would lead to null pointer exceptions.	1	Used promises to make sure that when data was used, it would contain values returned from the server
4	The children of the React Components had no keys, and so would slow down the updates made on them.	3	Added unique keys to each child in a list of components.
5	Session ID would disappear on refreshing the page as it was being stored on a global variable, and so user would log out.	2	Stored session on local storage as well, and checked for an existing session ID and updated the global variable.
6	Navigating to another route in the client would refresh the window, and clear the user data stored in the global variable, which would mean fetching user data on every route.	3	Added a component wrapper on all pages that ensured local links (internal navigation) wouldn't refresh the page.
7	Redirecting GitHub's Authentication callback to the server would lead to the client not responding.	1	Redirected the callback to the client and sent the response to the server from client.
8	The GitHub Authentication would lead to the client not responding if the server was not running.	2	Ensure that the server is always running. Inform developers if the instance

			goes down.
9	The GitHub webhook does not always cause a pull of the master branch.	2	Ensure that the logic for the webhook has been properly implemented. Ensure use of correct webhook.
10	The master branch is pulled when there are not any new updates.	3	Ensure that the logic for the webhook has been properly implemented. Ensure use of correct webhook.
11	The entirety of the GitHub branch is not pulled.	2	Correct logic on the repo downloading function in order to ensure the entire branch is downloaded.

3. Unit Testing Defects

For Square CI, we automated our tests for the server and the client separately using two tools. Mocha for the server and Jest for the client. This is mainly because Mocha was easier to learn and more lightweight, but at the same time testing React Components was done incomparably better by Jest.

Along with Mocha, an add on library Chai was used for an assertion library. We chose it for us to be able to test our API with HTTP requests.

The main modules of our application are broadly the server and the client. In the client, there is the router and the components. And in the server, there is the API, the GitHub polling system and the Test running system.

Product	Square CI		
Date	September 21st, 2017		
Authors	Vritant Bhardwaj, Jun Soo Kim, Brandon Marx, Shulin Ye		
Defect #	Description	Severity	How Corrected
1	After given access to an accounts github credentials, attempting to pull from a repository fails.	1	Fixed scope in GitHub authorization and requested repository access
2	After given access with correct	2	Don't hardcode 'master' but

	github credentials, pulling any branch will only pull master		use input given from user for certain branch and append to url
3	When given details on a nonexistent Github repository the server halts	1	included robust exception catching methods to handle being given incorrect information.
4	When given authentication to GitHub the repository branch does not download.	2	Added checks to ensure correct access token was being used to acquire repository from GitHub
5	When given a request to create a new repository, the Restful API displays "Failure"		Ensured that user selected repositories only from their own existing ones.
6	When given a nonexistent Route, the Restful API would return "cannot get"	2	Redirect all unspecified routes to a 404 page
7	User given endpoints were not functional.	1	Report to user on client that server endpoints are not valid
8	The code was not deployed on the specified server endpoint after running the tests	1	Added checks to ensure that the code was deployed to the correct server (alpha, beta or prod) depending on the tests that passed
9	The test logs generated were overwriting previous test logs	2	Added a new table to store test logs of tests being run on user's repositories
10	The details of the failed tests were not being shown on the client end.	2	Redirected verbose outputs of mocha into the table to ensure that details of failed tests would be shown
11	Notification for tests being run after the first time were not being shown	3	Changed the push notification to be triggered at every significant update of test logs.
12	The repositories on the /repositories route were not being shown	2	Checked for error responses from the server

			and let the user know of no repositories existing
13	Logout would not log out the user.	1	Ensured that local storage of the browser no longer held session ID when logged out.
14	Unlinking repository crashed client	1	Created route in server to handle unlinking repository with checks for errors.
15	/repos could be accessed even when user was not logged in	1	Ensured that sessionID was searched for in every route and redirected to login when not found