

# Design Inspection, Code Inspection and Unit Testing 2

## - 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	Connection values should not be hardcoded into the index.js file.	1	Use Nconf and a config.json to dynamically pull these values.
2	The front and the backend should not have separate config.json files when they will always be started in parallel	3	Create a central config.json which starts both servers.
3	MiddleMan/index.js was growing too large with functions to package db data that needed to be inserted.	3	Move to a separate utils.js function that exports functions for packaging db data.
4	User repo downloads were cluttering the root dir of our application.	2	Create a UserRepositories folder with a better file structure for storing user repo downloads.
5	The database did not contain a boolean value for if a User's repository was monitored.	1	Add an isMonitored boolean to the UserRepo table.

6	Database queries were inserted without testing for SQL injection.	2	Create a validate_db.js file that validates inputs to the DB before they're entered.
7	The repoID in the server is using test data instead of things returned from Github API.	1	Create a function to receive the user data from github API and process it for testing.
8	Githubhook npm package involved creating another server to listen on another port	3	Changed to express-github-hook npm package to avoid issues having communication being done on different ports.
9	MiddleMan/index.js was growing too large with functions to communicate with GitHub and the webhook	3	Moved them to separate files and called them using functions in the index.js file
10	Access Token was being used on client side for user identification	2	Changed to using session id for user identification, and stored access token on the backend
11	Multiple functions were executing the same task with different values, and so code was being repeated	3	Bunched together similar code into functions and used functional paradigms to reduce repetition.
12			
13			

## 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	DB query values were being used before they had returned due to the asynchronous behavior of js.	2	Use promises to handle function flow control.
2	Calling the /monitor/... route wasn't triggering the initial repo download	3	Add a function call to clone the repo and kickoff testing.
3	Due to a typo in creating user repo folders, the names of these folders included single quotes around it.	3	Remove single quotes from string interpolation.
4	A response that was already JSON formatted was being parsed into JSON again.	3	Remove the second parsing and just return.
5	Post call overwrote the creation of a webhook	3	Remove the initial post call and call the webhook instead
6	When receiving information about the webhook, incorrect variables and data were being given	3	Created a new function to listen for updates to the branch and use the data being received there
7	When we wanted to receive the payload of a webhook, no information was being returned	2	Changed the content type to 'application/x-www-form-urlencoded' instead of 'application/json'
8	When try to test the database function in the server, nothing is returned.	1	Calling the function in the right scope and passing the parameter with the right referencing.
9	The client was not checking for all the status codes	2	Add a function that checks for responses and accordingly initiates events
10	The structure of the JSON being sent from the API wasn't formatted to the planned standard	1	Rewrote the response of the API according to the standard.
11	API routes were being called in many places in the client and were being newly created every single	3	Created a helper function to be called when a route to a certain endpoint was

	time by string concatenation		needed to be called.
12	Github webhook not pushing proper payload information	3	Changed to a different npm package to properly receive payload
13	Access token was not being properly retrieved, returned a could not be found error	3	Changed the variable for secret key to properly update so that the http request was sent the correct information

### 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	The code to get the accessToken had been inserted into the DB.	3	Validate inputs such that only strings that follow the accessToken format could insert.
2	Errors were not being handled if a non existent repo id was inputted to /monitor/...	2	Handle the promise rejection and report the error back to the user.
3	Repos that changed from public to	2	If the repo is marked as

	private or vice versa we're not being updated in the database.		public, use public repo downloads first, if auth errors occur, then use private repo download and update the table if successful.
4	The list of repos being displayed for the user did not filter out the repos that were not being monitored	2	Added a conditional to check whether a repository was being monitored before displaying it.
5	User was not being directed to a login page when they were logged out	2	Added a check before any component mounted on any page to redirect user if session ID did not exist in the database.
6	When repo was clicked, it gave details of another repo.	2	Fixed off by one error.
7	There were empty spaces between repositories being displayed	3	Filtered out repositories that weren't being monitored before mapping over them using the inbuilt map function.
8	User login did not persist on different tabs	3	Stored session ID and user data in local storage to ensure persistence during the browser session.
9	Adding a repository to be monitored was not working as repo_id being sent to the backend was null.	2	Ensured repo_id being sent to be monitored was the correct one that the user clicked.
10	Response from server was causing error when checking for repository details	2	Parsed JSON before using it.
11	Webhook was not receiving a post request when a monitored repository had a pull request to the master branch	2	Matched post request path to the one stored on GitHub and corrected them so they were the same.
12	Removing a repository from monitoring was not working as the	2	Added an onClick function to the button to add

	button displayed did nothing		functionality
13			
14			
15			