

Design Inspection

Product	ListAssist		
Date	2/11/018		
Author	Development Team		
Defect #	Description	Severity	How to prevent
1	When a signed in user is viewing a list shared by another user, the former user should not have access to modify the list contents	2	Ensure the public version of the list does not have modification options for the list contents [not implemented yet]
2	Upon modifying the database, the user cookie could have expired or become invalid	3	Pass the cookie into any requests that the user wishes to make and check it against what's stored on the server [not implemented yet]
3	If a user clicks the sign up with Google button and they don't have an account on the server, one should be created for them	2	Check if their Google credentials are already in the database and if not, create the account [not implemented yet]

Code Inspection

Product	ListAssist		
Date	2/11/2018		
Author	Development Team		
Defect #	Description	Severity	How Corrected
1	Possible to enter no name when creating a new list	3	Check to make sure value (name of list) given is not null and not empty
2	When managing a user, the callback functions could be null	2	Check if the callback function is null

			and if so, return from the function
3	When modifying the database, the query string used could contain invalid characters	2	Check if the query string has any unsupported sql characters
4	Several null reference errors caught when toggling between different made lists	2	Check nullity
5	Some files use spaces instead of tabs for a non-uniform coding style	1	Always use tabs. It just makes sense.

Unit Testing

Methodology

In order to automate unit testing of our NodeJS backend, we are using the Mocha testing framework with the Chai assertion library. There are tests, separated by module in the test folder. When mocha is run (using npm test), all of the tests can be run automatically. For those tests that are for the http-server, the chai-http library takes care of creating a dummy server and making a http request.

Currently, since the front-end is not implemented with javascript, there is no front end test automation in place. In some cases, there were no tests we are able to implement since they were only markup. In the cases that there was functionality to test, we tested them manually. We hope to also implement automation on the front end.

Module	Amazon Web Scraper
Date	2/11/2018
Author	Drew Atkinson
Note	The Amazon web scraper contains two functions, one to parse an amazon item and one to parse a wish list. Each of these functions are exported in their respective files in the util folder. These are automated with Mocha and Chai, which assert that the proper fields, or errors are thrown.

Defect #	Description	Severity	How Corrected
1	When given an incorrect url to the file, there should be an error thrown and the JS promise should be rejected. This was discovered by the automated unit testing by passing an invalid url.	2	Check for a null result from osmosis, and manually reject the promise. (itemscraper.js line 17)

Module	Creating Lists		
Date	2/11/2018		
Author	Bhavana Kakubal		
Defect #	Description	Severity	How Corrected
1	No error thrown when non-unique list name is requested.	2	Not yet corrected. Add code to check uniqueness of list name. This is not possible yet because the front end page for creating lists is not yet connected to the database.

Module	Getting User Accounts		
Date	2/11/2018		
Author	Ryan Everett		
Defect #	Description	Severity	How Corrected
1	When given a token to retrieve user info, a test user is returned.	2	Not yet corrected. Connect Database

			functions to user/get function
--	--	--	--------------------------------

Module	User login		
Date	2/11/2018		
Author	Ryan Everett		
Defect #	Description	Severity	How Corrected
1	authenticator.js does not check for null when attempting to get login payload	3	Not yet corrected. Check for a null login
2	Authenticator.js may still add the user to tracking even if the callback returns null.	3	Make sure a returned token from authenticator is not null

Product	Routing		
Date	2/11/2018		
Author	Ryan Everett		
Defect #	Description	Severity	How Corrected
1	Express should serve the app main page at '/' instead of the express default page.	2	Changed default routing to the app page. Removed unused index.ejs file

Product	Database Connection		
Date	2/11/2018		

Author	Will Johnston		
Defect #	Description	Severity	How Corrected
1	Username in Users table in database can be blank, can be greater than 64 characters. Found through automated test cases	3	Checked for empty strings and too long usernames in the database connection.

