

# Design Inspection

Author: Whole development team

Defect #	Description	Severity	How it was corrected
1	The sign in/ sign out button did not properly update once a user signed out, as the button still showed the option to sign out	3	The code was updated so that the button displayed the option to sign in once you logged out
2	The text of the search button on the navigation bar was the same color as the navigation bar	3	The color and text of the button were updated so they were visible on the navigation bar
3	Sign in and sign up page were on same page which was confusing for users	3	Split into two pages with the default landing page going to the Sign In page
4	Navigation bar overlapped elements below preventing users from being able to access other elements of the website	1	A buffer was added to below the navigation bar so that it didn't overlap other elements
5	Profile button would not link to the profile page	2	A Link was added around the button so that it would redirect to the profile page
6	Cancel button on Sign Up page did not do anything	2	The Cancel button on the Sign Up page now redirects back to the Sign In page
7	It was difficult for users to navigate back to the Profile page	2	After signing in as a user, a Profile button now shows up on the Navigation Bar to make it easier for

			users to find
8	Dropdown to add ingredients in Profile page was not visible	2	Fixed dropdown to show users ingredient options
9	Pages did not scale to different screen sizes and would not display properly on smaller screens	2	Made elements in components scale to screen size so they look okay on a wider range of screen sizes
10	Recipe schema did not have all parameters to represent a recipe object	1	Using a json file to store all of our temporary recipe objects.

## Code Inspection

Defect #	Description	Severity	How it was corrected
1	Function for logging in user did not check for password only checked if username exists	3	Switched login method to google sign in, allowing google to take care of authentication while still maintaining our own backend
2	Methods and callback functions missing closing parenthesis, brackets,etc. resulting in syntax error.	3	All closing parenthesis and brackets were corrected
3	Returning wrong http	2	Looking up http

	status code		status codes and changing the return statement to return correct status code
4	Hard coding urls, which resulted in errors when trying to deploy to heroku.	3	Fixed by using heroku as primary backend interface.
5	Hard coding other context based information that was dynamic and changes with the environment	3	Fixed by having adding environment file that contained all information being used by the application.
6	Cross origin request errors.	3	Fixed by using a middleware that set the correct headers.
7	Using a local mongodb instance for all database transactions resulting in errors of inconsistency in the data among every developers computer.	2	Using a centralised online database for all transactions.
8	The user could upvote or downvote a recipe an infinite number of times	3	The code will be updated to only allow the user to upvote or downvote a recipe once (Not implemented yet).

## Unit Testing Report

This system relies on the following modules:

- Database module

- Backend module
- Authentication module

### Database Module

The Database module is the module responsible for storing user and recipe data such as username, ingredients, instructions etc. The database module was tested manually.

Product	Bazaar Web Application		
Module	Database Module		
Date			
Author	Development Team		
Defect #	Description	Severity	How it was corrected
1	We created a test user. Doing this we discovered we were storing passwords in plaintext in our database instead of hashing them before publishing.	2	We switched to google authentication and login, using OAuth2.0

### Backend Module

The Backend module is the module responsible for directing the user throughout the web application as well as modifying data and communication between the frontend and the database. The backend module was tested by the supertest and mocha frameworks.

Product	Bazaar Web Application		
Module	Backend Module		
Date			
Author	Development Team		
Defect #	Description	Severity	How it was corrected
1	We created a test that calls "/search" without a search query, the database	2	Json objects with appropriate error messages were added in

	returned a json objects without error messages.		
2	We created a test that calls “/search” without a search query, the application returned the wrong http status code.	2	Http status codes were researched and corrected

### Authentication Module

The Authentication module is the module responsible for allowing users to create accounts and login to Bazaar. The authentication module was tested partly through the mocha framework and partly through manual testing.

Product	Bazaar Web Application		
Module	Authentication Module		
Date			
Author	Development Team		
Defect #	Description	Severity	How it was corrected
1	We created a test that loads “/auth/google”, in doing so we discovered that the application was not able to redirect to the google sign in page.	3	Adding the correct redirect urls to the google cloud platform for the application
2	We created a test that logged a returning user into Bazaar, in doing so we found that Bazaar would recreate the user in the database because it thought the user was a new user.	3	Unique Google ids were stored as part of the user schema to identify returning users.
3	We created a test that successfully	3	We fixed this defect by adding parameters

	logged in a user to find that a lack of a callback function caused google to fail to redirect to Bazaar's home page after the login.		to the Google passport strategy to allow Google to redirect to Bazaar homepage with the appropriate user information.
--	--	--	---

### Account Module

The account module is the module responsible for the account page of the application where users can view their settings and change their username. This module was tested via manual testing.

Product	Bazaar Web Application		
Module	Account Module		
Author	Development Team		
Defect #	Description	Severity	How it was corrected
1	The Change Username option would allow for user to re-submit a new username of no characters or just a space character	3	A check is made to ensure that username is of actual characters. The UI will also alert the user and not send the request.
2	If a user inputs a username into the change username box that is already taken by another user, no error message is shown to indicate that.	2	Not yet implemented but in the future we would have to handle the error message that the backend gives and display that in the frontend page.

### Navigation Module

The navigation module contains the navigation bar, which contains the search bar, and the signin/sign out buttons. This module was tested by manual testing.

Product	Bazaar Web Application		
Module	Navigation Module		
Author	Development team		
Defect #	Description	Severity	How it was corrected
1	The search bar on the navbar would allow a search query of 0 length, causing a page to be rendered filled with nothing.	3	The submit button next to the search bar will only allow interaction if the query in the search bar is of length > 0
2	The search bar on the navbar would allow a search query of an item that could not be found, causing a page to be rendered filled with nothing without any other message shown.	3	Not yet implemented

### Sign In/Sign Up Module

The Sign In/ Sign Up Module contains the pages for signing into the app, as well as a page to create an account for Bazaar. These modules were created by manual testing.

Product	Bazaar Web Application		
Module	Sign In/Sign Up Module		
Author	Development team		
Defect #	Description	Severity	How it was corrected
1	User can hit submit button and successfully sign in without entering in Email or Password	1	Input is checked to make sure that the length is greater than 0 before submit button allows user to

	fields.		log in.
2	When user is signing up, the reenter password field was not being checked against the first password field.	2	Before user can successfully sign in, the password and reenter password fields are checked to ensure that they are equivalent.

### Recipe Viewer Module

The Recipe Viewer module aggregates other UI components in order to display data. These modules were created by manual testing.

Product	Bazaar Web Application		
Module	Recipe Viewer Module		
Author	Development team		
Defect #	Description	Severity	How it was corrected
1	If no recipes are available in database, user gets redirected to a blank page instead of receiving an error message.	2	Before Recipe Viewer page is shown, there is a check made to ensure that there are recipes to view.