Sprint 1: Design Inspection, Code Inspection, Unit Testing

Jacob Riggs, Julian Ng, Christian McKain, Cole Compton, Adrian Taubner

Design Inspection

Product: The Endless Abyss

Date: 9/20/2019

Authors: Development Team - Team 21

Defect #	Description	Severity	How Corrected
1	Trophies have a description field in the database, but no field to display this description in the trophy room	Workaround	Added a title attribute to the trophy elements so that the description is displayed when the user hovers over a trophy
2	Java used for the development of the backend was not satisfactory for the criteria of the project.	Critical	Changed the backend to be coded in Javascript and use Node.js
3	Connection to frontend from backend using Web Sockets seemed overbearing.	Workaround	Decided to use REST method and created a handler for RESTful requests.
4	Design of database changed to accomodate for not having a lot of space to work with without paying	Workaround	Decided to store the images related to the trophies on the frontend and reference the trophies in the

	money.		database using strings.
5	DigitalOcean hosting costs 40$ and we don’t need the extra performance for our web app	Workaround	Decided to host our web app on Google Firebase instead of DigitalOcean

Code Inspection

Product: The Endless Abyss

Date: 9/20/2019

Authors: Development Team - Team 21

Defect #	Description	Severity	How Corrected
1	Login Requests are not being received by server	Critical	Standup server listener
2	All informational toasts are not displaying	Workaround	Confirm import status on component and check the version number for new changes
3	Trophy Room expects an exact number of trophies	Workaround	Use v-for to generate a number of trophies equal to the number returned by the get-trophies request so that trophies can be added after the game has been released
3	There is a lack of div tags to differentiate parts of a component	Important	Backtracking to existing pages and create well label tags to help speed up CSS development time in the future
4	Using individual component pages will make it difficult to impose login	Important	Implement a global private and public page that checks for account token

	credentials		
5	Implementing a Java backend was convoluted and tough to use	Critical	Changed design to implement a Node.js backend instead
6	Reactive rendering in the trophy room and leader board	Workaround	Instead of using manual rendering Vue supports reactive rendering and should be used for readability and efficiency
7	Attempted to connect the backend to the database without considering the port.	Critical	Made sure the port was correct for the hosting service we are using.

Unit Testing

Module: Create Account

Date: 9/20/2019

Authors: Development Team - Team 21

Defect #	Description	Input	Output	Severity	How Corrected
1	Non matching passwords during account creation	Asd asdd	Account created (toast) EXPECTED (Toast) your passwords do not match (error font)	critical	Found the check statement was === when it should have been !==
2	Looking at the create account it states login	none	Labeled login EXPECTED Labeled Create Account	Workaround	Edit displayed label in affected pages
3	Create account accepts non alphanumeric characters	@#$%=	New Account Created EXPECTED No special symbols can be added to this field (toast)	Critical	Changing the form fields in account creation, and login to not add non alphanumeric characters to the field

Module: Trophy Room

Date: 9/20/2019

Authors: Development Team - Team 21

Defect #	Description	Input	Output	Severity	How Corrected
1	Trophies overlap other elements	User navigates to the trophy room page	Expected: Trophies display underneath page title Actual: Page title is covered by trophies	Workaround	Modified flex and size css tags6

Module: Login

Date: 9/20/2019

Authors: Development Team - Team 21

Defect #	Description	Input	Output	Severity	How Corrected
1	Non alphanumeric characters can be added to the username and password field and not fail on login submit	!@#$%	That is an incorrect login, please try again (toast) EXPECTED Special Characters can not be used	workaround	Have fields test what characters are being used before adding them to the field
2	Either field can be filled to a character size larger that what the server is willing to accept	Aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaa > 26 characters	An error has occurred please try again EXPECTED That is an incorrect login, please try again (toast)	Important	Apply a max character length username and password fields

Module: Server

Date: 9/20/2019

Authors: Development Team - Team 21

Defect #	Description	Input	Output	Severity	How Corrected
1	Connection to database being refused.	host: 'remotemysql.com', user: 'adrian.taubner@gmail.com', password: 'oBIw78M3tx', port: '3306'	ERROR: CONNECTION TO DATABASE REFUSED DUE TO PERMISSIONS	Critical	Login with the credentials of the database rather than to the hosting site: host: 'remotemysql.com', user: 'oBIw78M3tx', password: 'IgLMVTpxnE', port: '3306'
2	RESTful requests being handled incorrectly on the backend.	JSON with dictionary: [Hello : World]	Error with structure of JSON first character ‘u’	Critical	The parsing of the JSON not occuring correctly and body-parser can be used to fix this.

3	Attempting to receive data from a GET request.	var obj = req.body;
		console.log(obj.Hello);
			Error with req.body not existing.	Critical	Incorrectly attempting to retrieve body data from a GET request.