

## Design Defect Log 2

Product: Chat Resource / Database Manager

Date: 10/08/2016

Author: Noah, Garrett

Moderator: Garrett

Inspectors: Nate, Riley

Recorder: Nate, Riley

Defect #	Description	Severity	How Corrected
1	When sending a message the message is sent as a parameter instead of header/http request body. These parameters are cleartext even over SSL creating a security issue.	2	Changed the parameters to headers in the request body
2	Sending message was a get request and should be a post request	1	Changed request to a post request so that a request body could be added for future design changes.

Product: User Resource / Database Manager

Date:10/08/2016

Author: Riley, Nate

Moderator: Garrett

Inspectors: Noah, Garrett

Recorder: Noah, Garrett

Defect #	Description	Severity	How Corrected
1	There is no learning language parameter in the user database schema	1	We changed the database schema to have a learning language field.
2	There is no easy way to get the list of possible matches for a specific user.	1	We changed the learning language and native language field to be comma-delimited so that we can easily parse the string from the database.

Product: Auth Resource / Database Manager

Date: 10/08/2016

Author: Riley, Nate

Moderator: Garrett

Inspectors: Noah, Garrett

Recorder: Noah, Garrett

Defect #	Description	Severity	How Corrected
1	There is no easy way to distinguish auth tokens from Facebook and Google	2	Add a column to the table to keep track of which authority the token is from and have auth pass that to database when storing tokens

Product: Flashcard Resource / Database Manager

Date: 10/08/2016

Author: Riley, Nate

Moderator: Garrett

Inspectors: Noah, Garrett

Recorder: Noah, Garrett

Defect #	Description	Severity	How Corrected
1	When storing flashcards in the database, if multiple users used the same set id, those two sets would be treated as one	1	Added the user Id to each flashcard within the database.

## Inspection Defect Log

Product: Authorization

Date: 10/10/2016

Author: Riley, Nate

Moderator: Garrett

Inspectors: Noah, Garrett

Recorder: Noah, Garrett

Defect #	Description	Severity	How Corrected
1	Redundant calls checking if the body of the post message for an authentication request was empty	3	Removed redundant call
2	Authentication requests without a valid verb (login, register, logout) would return successful responses.	2	If the verb is not recognized, return an error
3	If the verb for an authentication request does not contain a trailing '/', the request is being executed twice	1	Removing the second set of calls to complete the request
4	The request body for authentication is being parsed twice	2	Remove the extra parsing calls
5	An invalid Facebook authentication token sends a valid response before sending an invalid response	1	Simplify the conditionals when checking Facebook authentication so that it always follows the intended branch
6	Google authentication is implemented but never being called	1	Call the Google verification if the provided auth token is from Google

Product: User

Date:10/12/2016

Author: Riley, Nate

Moderator: Garrett

Inspectors: Noah, Garrett

Recorder: Noah, Garrett

Defect #	Description	Severity	How Corrected
----------	-------------	----------	---------------

	None found		
--	------------	--	--

Product: Database

Date: 10/08/2016

Author: Riley, Nate, Noah, Garrett

Moderator: Garrett

Inspectors: Garrett

Recorder: Garrett

Defect #	Description	Severity	How Corrected
1	If the database returned a user with an invalid or null user id, no error is shown	3	Return an error if the user does not have a user id.
2	For the matching algorithm, the SQL query only allows exact matches of a user's entire list of languages	2	Edit the query to match individual languages instead of the full list
3	When retrieving flashcards, the same response is returned for an empty list as an invalid list	2	Return nil for an invalid list and an empty list only for a valid empty list

Product: Chat

Date: 10/09/2016

Author: Noah, Garrett

Moderator: Garrett

Inspectors: Garrett

Recorder: Garrett

Defect #	Description	Severity	How Corrected
1	If there are more than MAXINT64 image or picture files stored on the server with the same name, the server will crash.	3	When renaming a file to avoid collisions, if MAXINT64 is reached, begin appending letters to the end of the name.

Product: Flashcards

Date:10/10/2016

Author: Noah

Moderator: Garrett

Inspectors: Garrett

Recorder: Garrett

Defect #	Description	Severity	How Corrected
1	Flashcard front and back are vulnerable to SQL injection	1	Base64 encode the front, back, and setID for each flashcard before storing it in the database

## Testing Defect Log

Product: Chat

Date: 10/11/2016

Author: Noah

Moderator: Garrett

Inspectors: Garrett, Noah

Recorder: Garrett, Noah

Defect #	Description	Severity	How Corrected
	No defects		

Product: User

Date: 10/13/2016

Author: Riley

Moderator: Garrett

Inspectors: Garrett, Noah

Recorder: Garrett, Noah

Defect #	Description	Severity	How Corrected
1	The list of user's selected from the database do not always match the current user's native language to learning language requirement	2	Fix the sql query to search for both the learning to native and native to learning language requirement.
2	The list of users returned is sorted from largest difference in age to smallest instead of smallest difference in age to largest.	2	Fix the sorting algorithm to sort increasing instead of decreasing.
3	When there are no matches a crash occurs.	1	Add a guard to the sql query statement.
4	When there are no matches returned by the database manager a crash occurs	1	Add a guard to the json encoding call.

Product: Authorization

Date: 10/13/2016

Author: Riley

Moderator: Nate

Inspectors: Nate

Recorder: Nate

Defect #	Description	Severity	How Corrected
1	The register endpoint would not return the proper session_token, a default value was returned.	2	Modified the Database Manager for creating user to SELECT the user's info after being INSERTED.
2	The logout endpoint would return a successful logout if the request was accurate but would not revoke the user's session token.	1	Fixed the mysql query in logoutUserWith in DatabaseManager to set the user's session_token field in the database to NULL.
3	User was able to register multiple times with the same Facebook or Google account (authority_account_id)	1	Added a precheck in createUserWith in DatabaseManager to make sure that the authority_account_id has not been previously used.
4	The register endpoint would return a 200 status code if the user attempted to register with an already existing authority_account_id	1	Added a badRequestResponse call to the handler for nil response from createUser in DatabaseManager
5	The login endpoint was not updating the user's session_token in the database.	1	Fixed the sql query in loginUserWith in DatabaseManager to update with new session_token provided by user given that the session_token is valid with OAuth.

6	If user attempted to logout with an invalid session_token in the HTTP header the server would respond with an HTTP 200 status code.	2	Added a badRequestResponse call the the handler for nil in the bool response from lougoutUserWith in DatabaseManager.
---	---	---	---

Product: Database

Date: 10/12/2016

Author: Riley

Moderator: Garrett

Inspectors: Garrett

Recorder: Nate

Defect #	Description	Severity	How Corrected
1	The session_token column name is misspelled in the SQL query when checking for valid auth	1	Fix the misspelling
2	Multiple users are allowed by the database to have the same auth token	2	Make the auth token column unique
3	Multiple users can be created using the same Facebook or Google account	2	Add check that the Facebook or Google account id are not already associated with a user

Product: Flashcards

Date: 10/13/2016

Author: Noah

Moderator: Garrett

Inspectors: Garrett, Noah

Recorder: Garrett, Noah

Defect #	Description	Severity	How Corrected
1	When requesting flashcards, the front and the back was not being decoded	2	Decode from base64 before sending.



	from base64.		
2	Not checking character length on set ids, so the database would truncate the value.	3	Return an error to the user if the set id is longer than 50 characters
3	Not checking if the front, back, and set id are empty	3	Check for empty string for each front, back and set id
4	Invalid json formatting did not notify the user they have failed	2	Send invalid flashcard response