

Product	Song Wars - Team 17		
Date	10/13/2017		
Author	Ben Maxfield, Tom Fanella, Terry Lam, Christian Lock, Austin Miller		
Defect #	Description	Severity	How corrected
<b>DESIGN INSPECTION</b>			
1	Users shouldn't be able to recommend songs multiple times a week.	1	A new SQL table was created managing users' song recommendations, which will be cleared each week.
2	Backend shouldn't have to make multiple SQL queries to make song recommendation.	3	Client saves the Spotify User ID that is now returned with access token on login.
3	Need to be able to properly assign positional hierarchy in the bracket for songs to be displayed.	1	Establish a round and position column in the bracket table to distinguish where each song belongs in the bracket and in which round.
4	Bracket data (songs voted on during the past week) should be able to be migrated and stored whenever historical data is needed	1	Add a bracket_id column to distinguish what bracket the data belongs to.
5	Need a way to access historical data from any particular week	2	Create a separate SQL table that will append each week's bracket data, which will be able to be queried by bracket_id.
6	Users should be able to remain logged in after repeated visits to the site.	2	Store the user's id and authorization token in the LocalStorage of the browser and use it to authenticate on repeated visits.
<b>CODE INSPECTION</b>			
4	<code>export const authServer = code =&gt; {   return new Promise((resolve,   reject) =&gt; {</code>	3	<code>export const authServer = code =&gt; {</code>

	<pre> let uri = '{URI}'; fetch(uri) .then(data =&gt; data.json()) .then(json =&gt; {   resolve(json); }) .catch(err =&gt; reject(err)); }); } </pre>		<pre> return new Promise((resolve, reject) =&gt; {   if (getAccessToken()) {     reject('Already Signed in');   }    let uri = '{URI}';   fetch(uri)   .then(data =&gt; data.json())   .then(json =&gt; {     resolve(json);   })   .catch(err =&gt; reject(err)); }); } </pre>
5	<pre> &lt;BrowserRouter&gt;   &lt;Route exact path='/auth' component={Auth} /&gt;   &lt;Route path='/' component={App} /&gt; &lt;/BrowserRouter&gt; </pre>		<pre> &lt;BrowserRouter&gt;   &lt;Switch&gt;     &lt;Route exact path='/auth' component={Auth} /&gt;     &lt;Route path='/' component={App} /&gt;   &lt;/Switch&gt; &lt;/BrowserRouter&gt; </pre>
6	<pre> const Auth = () =&gt; {   let uriSearch = this.props.location.search; // ... } </pre>		<pre> const Auth = ({ location }) =&gt; {   let uriSearch = location.search; // ... } </pre>
UNIT TESTING INSPECTION			
Test #	Description	Severity	Defects Found
1	Tested to ensure duplicate song recommendations from the same user would not succeed.	1	N/A
2	Tested to ensure that the lambda for past week's bracket data returned the necessary database entries.	1	N/A

3	Tested to ensure that voting lambda GET for past week's bracket operated as expected, getting a random matchup for the user to vote on.	1	Ran into issue not duplicating matchups for a user.
4	Tested to ensure that voting lambda POST for past week's bracket operated as expected, recording the user's vote on a matchup.	1	N/A
5	Tested the redirection from the authentication page to the landing page after a user has authenticated.	1	A case testing a user that had already authenticated failed.