

Startup Submission 04: Dynamic UI Mockups - Report

Team: BBQ Forte

Key React Components:

SideBar: Provides navigation to all content via react-router. The sidebar also has a state that contains the current user's data allowing it to pass it down to linked components along with displaying the user's name.

Navbar: Displays the top navbar with react-router functionality. Has a state that contains the current user's data allowing it to pass it down to linked components along with displaying the user's name. Also, it has a search bar that allows user to search for relevant terms in the existing playlists.

Carousel: This component displays five (may be more or less) featured contents as a sliding banner on the main page. "Featured contents" include popular games, new releases, or other random steam games/playlists.

News/Updates: This component displays the six most recent new/update entries, which includes playlist updates, new game playlist releases, site maintenances, etc.

PlaylistFeed: Manages the display of Playlist and Recommend in a feed format. Has a method, `onChildChanged`, that is passed to Playlist as a prop so that Playlist can delete itself then pass the new state back up to its parent component. PlaylistFeed also implements creating new, empty Playlists. Eventually there will be options (hopefully) to import and export playlists to/from Spotify. There will also be options to associate playlists to games pulled from the Steam API rather than just typing in a name.

PlayList: Uses Bootstrap panels to display editable playlists. Playback capabilities are not yet enabled because it will require integration with Spotify. Each playlist can search the Spotify database (using a GET in `server.js`) for tracks. For now this just returns data such as title, track, artist, Spotify URL, and album. Playlist uses Bootstraps ``collapse`` CSS extensively. Each Playlist is initially displayed with the song list collapsed. Also, the ``Add New Playlist`` panel at the top expands when the user clicks it to show the form for creating playlists. Playlists can remove themselves from the database when the user clicks the X.

SongList: This component implements the Spotify song search and is called in the Playlist `render()`. The Spotify API only returns 20 songs, which this will display after getting the list. (See Project Feature Bugs below if the song search doesn't work on the first try).

Song: This component displays data for individual songs and is called in the Playlist `render()` method. Because of a quirk of ReactJS, each song has to have its own `<table>` even though it is just rows of the table. This component is also capable of deleting itself. If it is called by

SongList, it will NOT display the remove icon/button because it should not be able to remove search results.

Highest-Rated: This component displays the highest rated playlists which depends on the number of votes.

Most-Popular: This component displays the most popular playlists which depends on the number of votes.

New-Release: This component displays the newest playlists which depends on the time of playlist creation.

Rising: This component displays the rising playlists which depends on the time when the playlist is created and the number of votes within that time.

Search Results: This component displays the search result after the user has entered search terms in the search

Settings: This component allows the user to modify their experience of using the app by changing certain parameters.

Profile: *****KNOWN BUG**** Cannot get currentPlaylist into state so it is mocked up in DB, also cannot get savedPlaylists into state so cannot determine favorite artists and games. These are left static.* This component renders the whole profile page, allows the user to edit their fullname and about sections. This section is incomplete because of 15+ hours attempting to figure out the bug with the state.

Honors Feature React Components:

Logan:

Forum: This component displays a list of topics, and relevant information on the topics (such as post and thread counts). Although I created a method to pull information from the Forum JSON object, the information of specific objects in the topics array, such as topics[0] was not able to be retrieved so I manually entered the starting data into the state field. Since information from the topics could not be pulled, I could not pull information for the Last Discussion field. Uses the getForum() function.

Topic: This component displays a list of threads, and relevant information on the threads, such as replies and views. Since view count is not a part of the thread object, it was not mocked up. Like with the forum component, I had trouble pulling directly from the topics array, so I could not

dynamically create a list of threads, although a class for a table row (that could be created dynamically using map) was created. Uses the `getTopic()` function.

Thread: This component displays a list of posts. The situation with this component is akin to the situation with the Topic component, posts could not be added dynamically due to trouble pulling the information from the database.

New Thread: This component allows the user to enter a title and content for a new thread (the content goes in the thread's first post field). Again, I was not able to finish implementing this feature due to issues with the database. The information does not add to the database as desired. Uses the `postThread()` function.

New Post: This component is similar to the new thread component, except the content the user enters is a post that is added to a thread's post field. I regrettably could not finish implementing this feature, again due to issues with the database. The information does not add to the database as desired. Uses the `postComment()` function.

Matthew:

Recommend: Displays songs recommended for the user on the playlistfeed component on the user's playlists page. This portion of the page does not render if the user has not selected "show recommendations" in the settings page.

Stanley:

Private Chat - Recent Conversations: This component displays a user's "recent conversations" with other users. The user may click on the other user to switch the chat box to that user's conversation or the "x" to delete the conversation from the list. When clicking on a new user from the "live help" menu (see below), the recent conversations menu should be updated with a new entry of the selected user if it does not exist already (see bugs section).

Private Chat - Chat Box: This component displays a user's conversation (stored messages) with a given user. The user may send new messages to the other user, which will be displayed in the chat box.

Private Chat - Live Help: This component displays a list of users that are organized by genre (this is decided based on their preferences designated on their user profile). The user may click on one of these names to switch the chat box to display the corresponding conversation (and messages) with the new user.

Founder Responsibilities:

Brendan Kelly: Edited the Profile component to include users profile data pulled from the database. ****KNOWN BUG**** Cannot get currentPlaylist into state so it is mocked up in DB, also cannot get savedPlaylists so cannot determine favorite artists and games. These are left static. Most of this component is incomplete because of 15+ hours of trying to debug the known bug above,
<https://piazza.com/class/ij4c6lc9mnu9o?cid=276>
To be completed: editing changes the DB, uploading profile pictures, updating followers/following. Check private piazza post for more details.

Jess Hendricks: Created the initial Sidebar and Navbar components and added react-router functionality. Also created the first version of `initialData` for everyone to edit and use.
Implemented the PlaylistFeed, Playlist, SongList (including setting up the GET for songs with Spotify), and Song components along with several functions in server.js related to their functionality.
Altered the CSS styling for Playlists to display them as panels with rounded corners that are collapsable.

Ka Wo Fong: Edited the Navbar components to incorporate the searching functionality and created the search result component.
Implemented the highest rated, most popular, new release, and rising pages to displays corresponding playlists in each page.
Edited database.js to include mock data for highest rated, most popular, new release, and rising pages.
Created multiple functions in server.js to retrieve mock data from database.js
I have authored getPlaylistWithAuthor(), getNewRelease(), getMostPopular(), getHighestRated(), getRising(), and searchPlaylist() in server.js

Logan Rennick: Fixed formatting of the About Us page. Implemented the Forum honors feature. Most of the Forums component is unfortunately still incomplete due to the bug in accessing the Forums object that all the Forums pages (Forums, Topics, Thread, New Post, New Thread) rely on.

Matthew Zenzie: Introduce settings page, and integrate with the database, starting each user with default settings. Made sure settings propagate out and effect the pages they should effect.

Stanley Lok: Implemented the Carousel and News/Updates Sections that are displayed on the home page.
Implemented the Private Chat honors feature, allowing the user to store chat conversations with other users.
Transferred much of the site's content to the mock database as 'initialData' for use along with multiple server functions to retrieve the mock data from the database.
Edited the ER Diagram (see last page) based on grading feedback and mock database format.

Product Feature Cuts/Bugs:

Bug: Playlist: There is a challenging bug (detailed in [Issue #10](#)) where the first time a search of songs is attempted nothing happens. This *seems* to only occur the first time the feature is used after playlist.js or song-list.js is altered. Otherwise it has been very difficult to replicate and seems to be somewhat rare.

Bug: Private Chat: There is a bug where after clicking on a user in the private chat UI (either in the Recent Conversations menu or Live Help menu), the chat box does not immediately switch to show the selected user's conversation. Instead, the user must send a message to the old user to switch the chat box to the selected user. This also causes the problem where the first message is stored in the old user's message log (all subsequent messages are stored in the correct user's log given that you do not click on another user). There is also a bug where clicking on a user that does not exist in the Recent Conversations menu does not add the user to that menu. There is probably a problem dealing with asynchrony, which will be looked into soon.

Bug: Search Bar: There is a bug retrieving the query parameters from the URL in search-result.js component. Right now, the search term is set to "Elite". However, it should search whatever terms the user enters.

Bug: Updating Highest-Rated, Most-Popular, Rising, and New-Release: There is a bug in updating the playlists on these page because of missing properties in our mock database which we will fix soon.

Bug: Settings: The checkboxes won't always take the proper checked state at the start. The value is still set correctly in the database though.

Bug: Recommendations: Adding a song doesn't properly delete the object from the list of recommended songs. Furthermore, songs are currently held only within playlists, so additional songs are mocked for the moment. In the finished product, the addRecommendation function will cause the server to perform a number of additional actions.

Bug: Forums: There is a bug in retrieving information from the Forums JSON object. This bug affects the Forums, Topics, Thread, New Thread, and New Post pages. I intend to fix these issues by finding a way to get the getForum and getTopic methods to work as desired, and to add mapping functions to the topics and thread pages.

Bug: New Thread and New Post: Information inputted in content fields is not able to update/ be added to the Forums JSON object. If the previous bug can be ironed out I expect this bug to be ironed out much easier. Additionally, I will need to make sure the event listeners are attached to the content fields/ button, and I need to figure out how to route from the new thread / new post page to the newly created thread/post.

Cut: Additional Thread/Post Pages: Since I wasn't able to dynamically add threads and posts to the topics and threads respectively, it wasn't feasible to implement the "navigator" feature (although the navbar interface is still on the topic and thread pages). While working on fixing the other bugs for the forums component I will try to implement this feature. If it isn't feasible, the navbars will be removed.

Cut: Friends Tab in Private Chat: We have decided to move away from having a friend's list and instead have a following/followers system. Thus, the friends tab no longer fits in the private chat. This section has been revised into only the "Genre Help" tab, which has been renamed to "Live Help."

Cut: Popout Private Chat: This was an extra part of the honors feature that would improve user accessibility to the private chat function, but I could not figure out a way to implement this smoothly. This feature may return later, but it is unlikely.

Updated ER Diagram:

