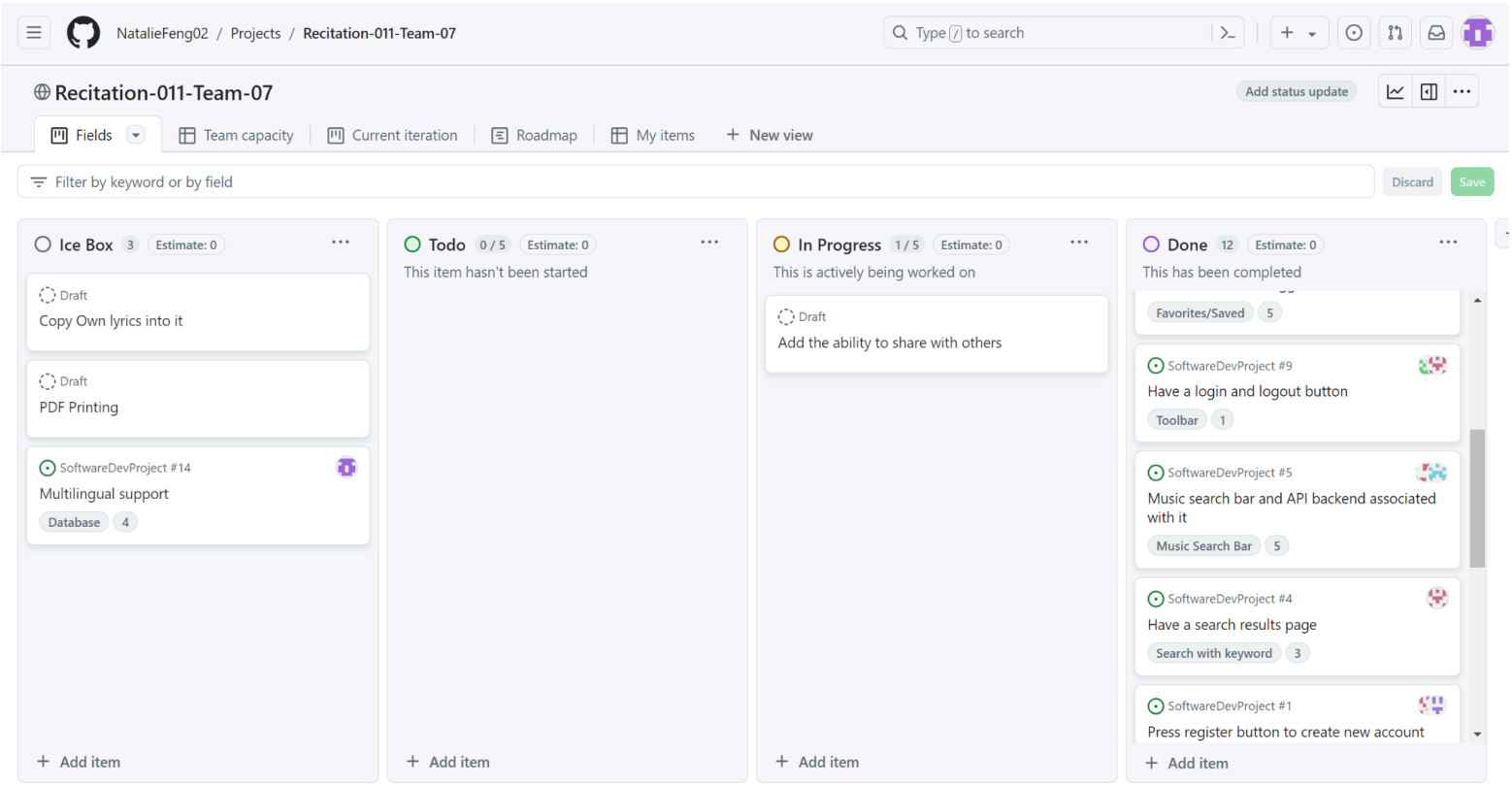
LyricSense

Members

- Max Buchalski (mabu7182)
- Daniel Busch (dabu1055)
- Lydia Clark (lydclar)
- Natalie Feng (NatalieFeng02)
- Charles Keely (charles-keely)
- Gabriel Vitti (gabrielvitti)

LyricSense is an application powered by the ChatGPT, Spotify, and lyrics.ovh APIs that allows the user to find their favorite songs and get an in-depth analysis of both the lyrics and the background information surrounding how the song was made in the first place. After creating an account, the user can enter their favorite artist, album, or song in the search bar to look for a song to be analyzed. Once the search completes and the user selects a song, a request is sent to ChatGPT to analyze the meaning of the lyrics and give those results back to the user. If the user likes the response they've gotten, they can save this analysis to their account and quickly go back to it later without having to search for the artist again. A similar feature exists for saving and viewing the background information. Additional features include the ability to reanalyze a song to get a different analysis/background, searching up other friends to see if they also use the website, and the ability to update the user's username, email, or password.

Project Tracker: https://github.com/users/NatalieFeng02/projects/1



Video Demo: https://drive.google.com/file/d/1XLP4h7GJHpzY_InxtogYBrVdXTIpaWMX/view

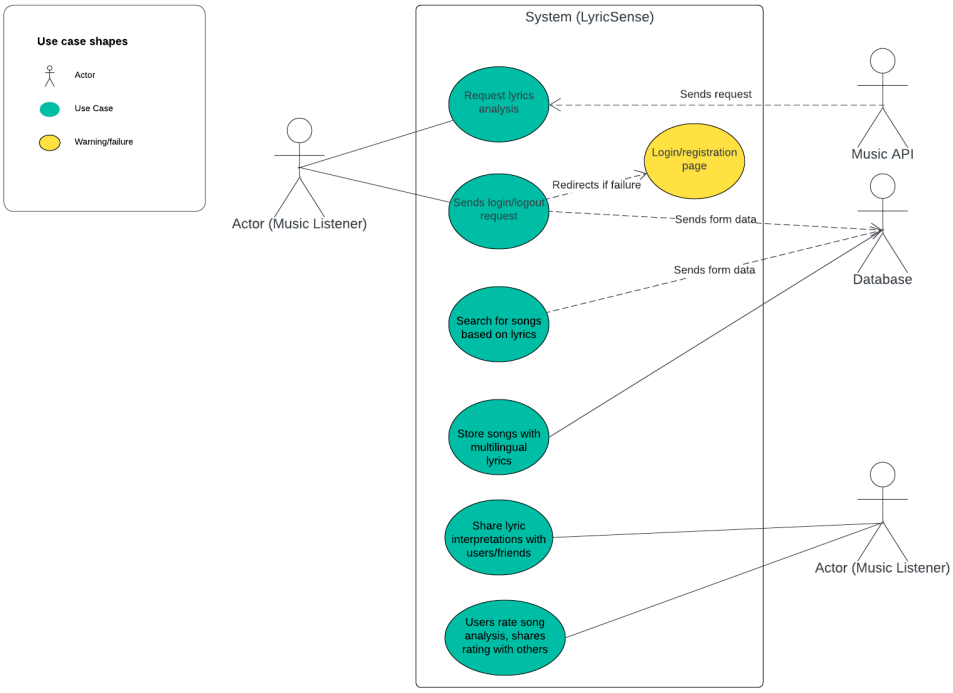
VCS: https://github.com/NatalieFeng02/SoftwareDevProject

Contributions:

- Max Buchalski: Made the presentation, logout page, wrote the original API for logout, login, create, and account information. Wrote the original hbs files for those all and did some work to help on them throughout the project, as well as making the navs and making sure they work correctly on each page and all the view pages.
- Daniel Busch: Created and implemented database to store user information and song analyses/backgrounds. Implemented unit testing with Mocha/Chai. Wrote endpoints and built pages for saved analysis, saved background, and saved analyses.

- Lydia Clark: Added styling/consistency to the login, create account, forget password, account not found, and password reset successful pages. Implemented endpoints for logging in, and the navbar. Made the home page dynamic to display different features depending on the user's logged in status. Added error messaging to login/account creation so users understand what issues are.
- Natalie Feng: Worked on the register page, the login page, and the account information page.
- Charles Keely: Implemented the design and most of the backend for the analysis and background pages. Implemented the search/results pages. Embedded the dynamic album display for the front page. Helped bugfix in teammate’s code. Implemented the loading page with a dynamic color changing loading symbol and nav bar. Created the Figma
- Gabriel Vitti: Worked on the iTunes search API, later shifted to working on the Spotify API. Created the “users search” page and functionality to add and remove people from your friends list.

Use Case Diagram:



Wireframes:

https://www.figma.com/file/MV7RbVFRbzuVgTFY5vsX6c/LyricSense-Website?type=design&node-id=0-1&mode=design

Test Results: In our first test, we were simply testing the functionality of creating an account and login pages. At the time of testing with users, we did not have functionality to alert users on whether or not that username already existed or if they were missing a parameter. We also had users ask us whether or not they needed special characters or not in their login which we hadn't specified. While we unfortunately did not have a chance to update the "hints" on creating a user account such as having a minimum number of characters or the requirement of special characters, we were able to implement alerts that helped guide the user if their account creation failed. In addition, we also have an alert for the login page if a user inputs the wrong credentials.

Our second test involved letting the users actually try to get a song analysis for a song that they liked. To keep the experience consistent as the search results page may look different depending on if a song has already had an analysis, we removed any saved analyses after each user test. Overall, our users had a positive experience with navigating these pages. The search bar gave hints on how to search, the results page was clearly labeled so the users knew how to read this page, and although not explicitly told, the users intuitively knew to click on the name of a song to get an analysis. One important piece of feedback we were given was a way to “mark” songs already having an analysis/background in the results page. While most of our users went to the “Saved Analyses” page in the corner of the site, some users wanted to get an analysis of multiple songs from their favorite artists and kept forgetting which songs they already did an analysis on. In one user test, they accidentally did a second analysis because they thought they did another song from the same artist. As a result of this, we implemented “Analysis” and “Background” buttons that dynamically appear next to the search results of a song if that user had already gotten an analysis or background for that song. So, if they search for an artist/album and see those buttons, they know they have already done that song and can safely skip it.

Due to time constraints, our third test involved the simple ability of being able to find a friend on the site and adding them. While the friend functionality does not serve much use currently, we still wanted to test the framework of a friends system and get early feedback on it for further development. To test this, we had pairs of user testers both log in to an account provided by us (Due to the database only being local) and have them discover for themselves how to add a friend. While we did have to point some users on where to go, the adding process was overall intuitive to them. Most users would input the full username of their friend, but some only did a partial entry and were still given the same results. Notably, we made the “Users” page as a feature on the nav bar so that users will have an easier time finding this. We got feedback from a couple of users on potentially having a “display” of users at the bottom of the user search

page so you could scroll through that and find their friends, but we did not implement this both due to time constraints and concerns over privacy (Whether or not a user wishes to be “known”).

As a whole, we were very happy with the feedback we received. We had to make some adjustments to how users actually navigate and interact with the website, but the overall experience was very positive from our users.

Deployment: To use this website, follow the GitHub link to the repository and clone the code. An env file will also be needed which includes the ChatGPT API key. After getting these necessities, launch Docker, go to the SoftwareDevProject/ProjectSourceCode directory in your favorite shell/terminal, and type “docker compose up”. If errors occur, a second docker compose up” may be necessary as the database is being initialized which some unit tests depend on. Once the docker containers are running, navigate to http://localhost:3000/ in your favorite browser to access the site. While the main page can be viewed, an account must be created in order to use the features of the site.