

UniSpot

Kevin Buhler, Blake Hamilton, Tristan Hunt,

Joanna Parker, Baxter Romero, Shuchi Shah

Table of Contents

Project Description	2
Project Tracker	3
Video	3
VCS	3
Contributions	4
<i>Kevin:</i>	<i>4</i>
<i>Blake:</i>	<i>4</i>
<i>Tristan:</i>	<i>4</i>
<i>Joanna:</i>	<i>5</i>
<i>Baxter:</i>	<i>5</i>
<i>Shuchi:</i>	<i>6</i>
Use Case Diagram	7
Test Results	8
Deployment	9

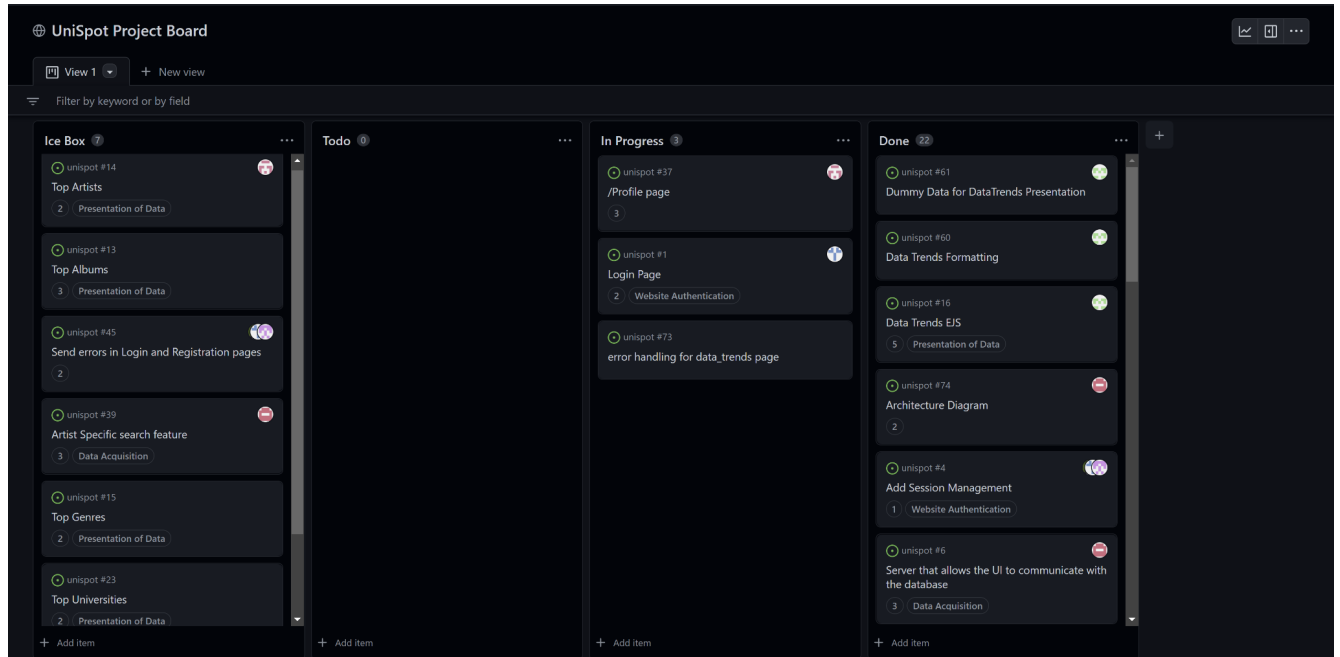
Project Description

UniSpot is a new way to connect with your university community! Create a student account with just your city and school to see your university's favorite songs. Check out the trends page for an interactive graph showing how popular a song has been over time. Once you have an account, you can search for the songs you listen to and increase their popularity! Your profile page will show your username and which songs you've liked recently. As a free, crowdsourced application, we only use your data to calculate song popularity at your university. We don't share it with anyone else and we always encrypt your passwords.

Future UniSpot growth will include a broader selection of music: albums and artists would be available as well as individual songs, and they would be incorporated into our trends feature. In addition, a verified artist account would give artists access to a ranked list of universities that listen to their music. Live concerts in the United States is a 52 billion dollar/year industry, so artists and record labels would love to get geographic data about where they can target listeners for concerts. This would likely be the primary revenue stream for our app.

Project Tracker

We used Github's [Project Board](#) feature to manage our workflow.



Video

Watch this [video](#) to see a live demo of the UniSpot website!

VCS

Our [Github repository](#) includes our source code, video demo, README.md, documentation, and project board.

Contributions

Kevin:

I set up all of the pages for our project and designed the user interface, like the footer, navbar, the welcome page, login and register pages, and home page. I was in charge of displaying everything on all pages except for the profile page. I used EJS and TailwindCSS to do most of this. I also worked in the backend to make sure that all data was ready to be presented in the frontend. I routed everything in the backend and also used some of Blake's queries to capture all of this information that we needed to display.

Blake:

I set up the directory structure for our project and worked on the creation of the database and the four tables in use for the database, as well as the final database diagram that our app uses and set up dummy data templates. I set up authentication for a third party API, then created a search route that uses the API to let our users search for songs. I then created a route to add songs from the search results to our database and a route to display the top songs at each university over different time frames, which uses complicated dynamic queries to run analytics.

Tristan:

I worked on the *data trends* page for our project, as well as contributing to project/presentation planning. I used ChartJs infrastructure to manually create a chart with a scrollable x-axis, and formatted the chart to fit our relevant information. I created a search bar form that submits a GET request to our SQL database. With Blake's help I developed the request

to query our database for the searched song, then return an array of size 12. Each entry of the array represents the number of transactions recorded with that song for each respective month. I displayed this array on the chart along with a card containing information about the song searched for. I added dynamic features to the chart so that it displays the current month as the last showing one, and the current year is displayed at the top. I also added dummy data for the song “Blank Space” to populate this chart for demonstration purposes in the final presentation.

Joanna:

I worked with Shuchi to set up and fill out our Github project board, and set up the initial database structure and diagram. I also took notes on all of the weekly meetings and kept track of requested changes from the TA meetings. I worked on the login and register routes as well as the users table in the database. That particular part of the project required a lot of debugging and resolving merge conflicts. I also did a lot of testing edge cases and new user testing, and I wrote the final project report.

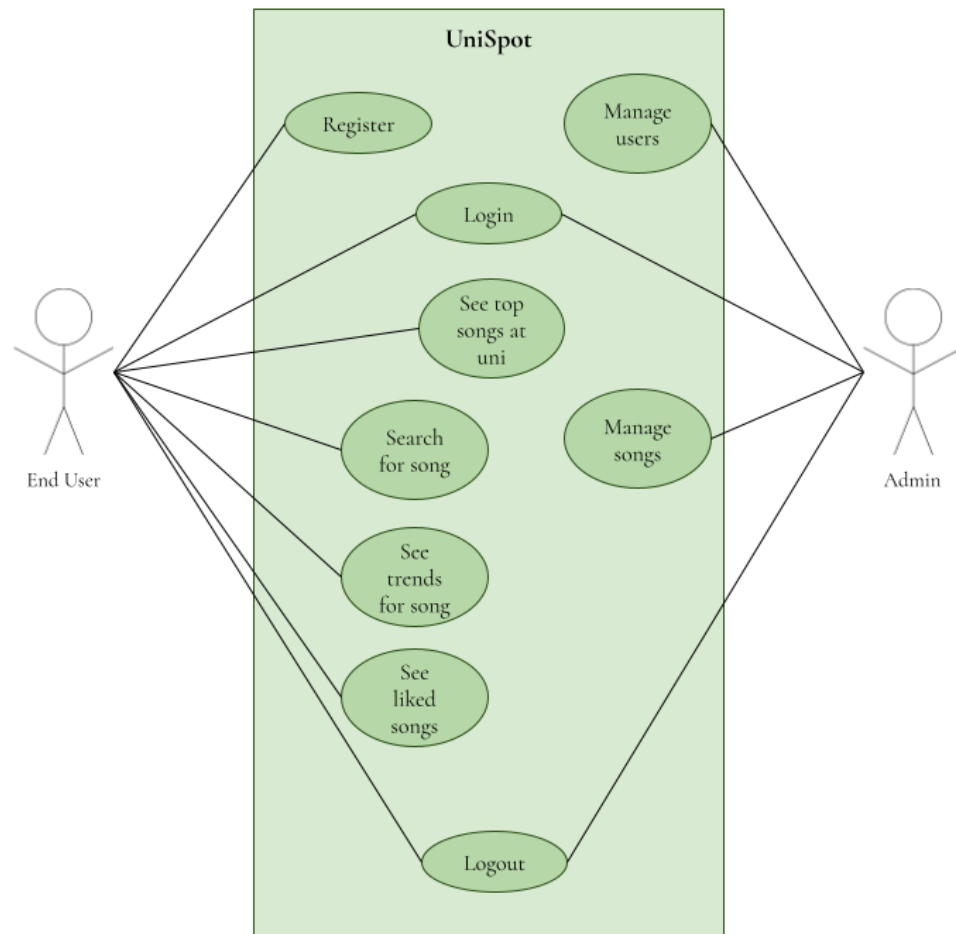
Baxter:

I was in charge of creating and uploading the release notes for Unispot on a weekly basis. Furthermore, I set up both the frontend and backend of the /profile page. I did this by creating user session management to display the user information. Furthermore, I used queries to populate and display cards on the front end reflecting all of the users “transactions” or liked songs.

Shuchi:

For this project I worked on creating our project board on Github. Along those lines I made sure our group was staying on timelines with weekly checkpoints. I also was in charge of feeding dummy data into the database. In which I added dummy data for all our tables. I was responsible for making sure our data was accurately getting calculated and displayed. I spent time on trying to debug the errors with our register and log in page. Another thing I worked on was pulling most of the information for the presentation.

Use Case Diagram



Test Results

We broke up our testing into three main sections: user authentication, the presentation of data, and the acquisition of data.

Testing user authentication primarily involved testing the login and register routes, and we tested continually through development. We ran into several issues, the biggest of which was a discrepancy in the database that caused white space to get added to the password in registration. Once that issue was resolved we were able to account for the following cases: correct usernames and passwords, incorrect usernames and passwords, usernames without passwords, and passwords without usernames. New users found our register and login pages intuitive to use.

Testing data presentation involved testing the front end of the main features of our website: the top 10 songs, search feature, trends page, and profile page, as well as website navigation. User acceptance testers found a couple of places where cards looked like clickable buttons but weren't. They also recommended that the search feature should not be case sensitive, or there should be an autocomplete functionality so the user doesn't have to enter the whole song name. Overall, they like the website design and found it mostly intuitive to use.

Acquisition of data was tested continually by the developers throughout the project. The bulk of this was testing the search songs feature: searching for a song in our database, searching for a song not in the database, and liking a song. One of the main things we accounted for was songs

with the same name. We wanted to display all songs that a user could be searching for rather than just the first song with that name in our database. On the front end, users found the search song feature to be intuitive and easy to use.

Deployment

We were not able to deploy our website due to issues with CU's Computer Science servers.