CS 514 Project 3: Music Library
Marisa Tania
Professor Chris Brooks
December 11, 2020

The music server runs on port 8081

1. LOGIN

USER'S GUIDE:
Go to http://localhost:8081/login and to successfully login, there are three users in the music.db users table:

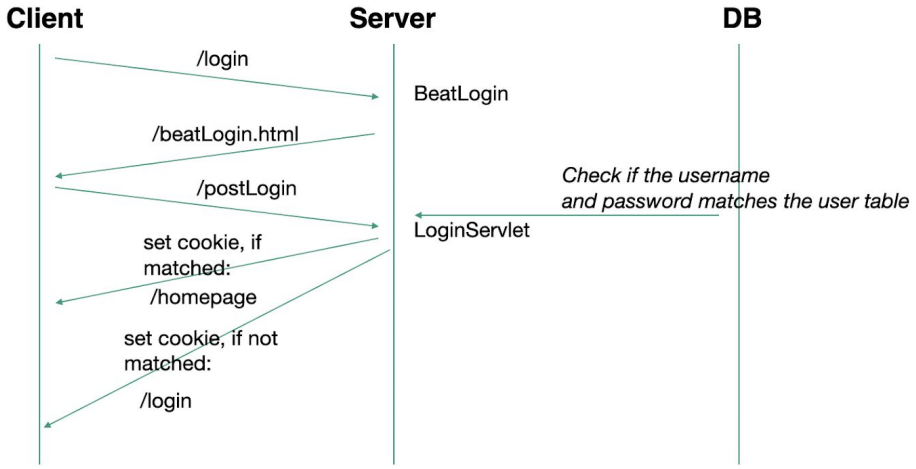
ID	Username	Password
1	Marisa	mt123
2	Lodie	l123
3	Nemo	n123

If the username or password is incorrect then the LoginServlet will redirect to http://localhost:8081/postLogin and notify the user to try to login again.
If you try to go to other pages without logging in, you'll be redirected to the login page, so make sure to log in first.

DEVELOPER:

- Client-side sends a request /login
- BeatLogin in the Music Server response with its doGet method that reads through the login.html page.
- The login.html is a form with method post and action "/postLogin"
- The server will respond with LoginServlet that get the parameter username and password from the HTML form and store the username in a 24-hour cookie.
- The LoginServlet's doPost method opens up music.db and checks the provided password against what is stored. If this is a match, go to the homepage. If it is not, send a response back to login.html, with a message indicating that the login failed.

Task 1: Login



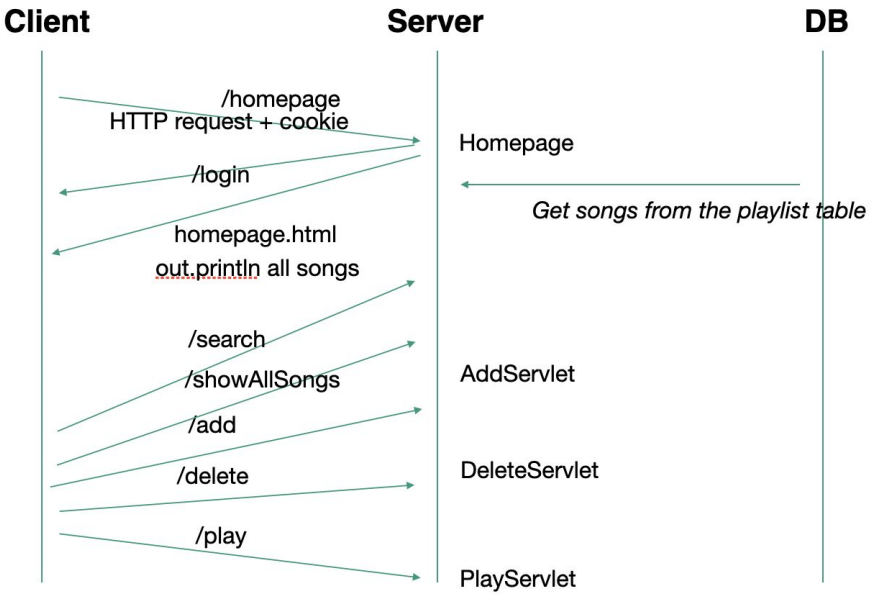
2. HOMEPAGE

USER'S GUIDE:
Once you logged in, you'll be redirected to the Beat Play Homepage http://localhost:8081/homepage that has 5 features: Search, Show, Add, Delete, Play. Note that the homepage displays the playlist table in music.db.

DEVELOPER:

- Client-side sends a request /homepage
- BeatLogin in the Music Server response with its doGet method that checks first if the user is logged in or not, if they are, it will read through the homepage.html page, and print out songs from the playlist table in the database. If not, then it will send a response to the login page, just like all servlets in this music server.
- homepage.html has a search bar with the action "/search" that will call the SearchServlet, and four buttons show, add, delete, and play that will go to the designated localhost URL on click.

Task 2: Homepage Library



3. SEARCH

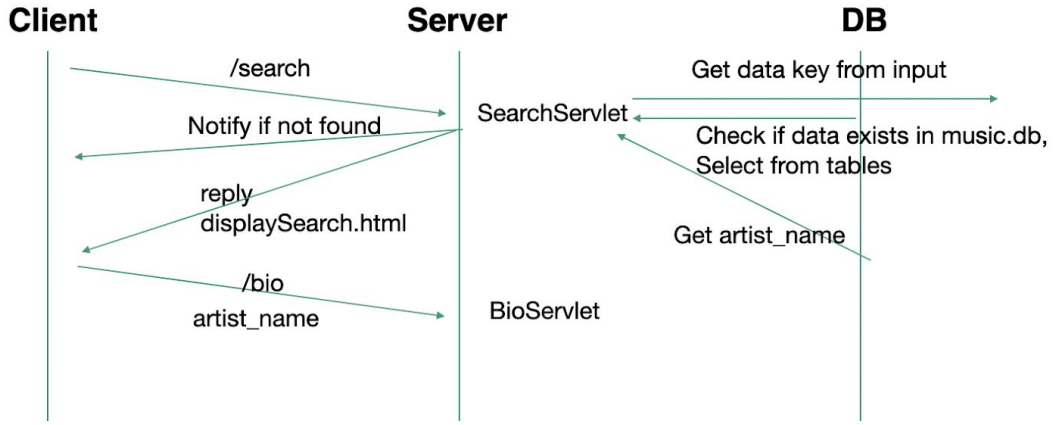
USER'S GUIDE:
At the top of the homepage, there's a search bar. You can search for songs based on artist names, song names, or album names. When you click the magnifying glass search button, you'll be redirected to http://localhost:8081/search. If the data is in the songs, albums, or artists table, then you'll see a table with a list of songs. (Check the show button to see the current all songs in the music.db)

If you click "GET BIO" button, you'll see the biographical information of the artist.

DEVELOPER:

- Client-side enter a search either by artist, song, or albums and sends a request /search.
- The servlet checks the cookie to make sure the user is logged in.
- The server responds with its doPost method that opens the music DB database page and prints out songs that match the search.
- The query is using inner join on the songs, albums, and artists as they are connected with foreign keys.
- The "GET BIO" button will send a hidden form request to the BioServlet with the artist_name as the parameter and value.

Task 3: Search



4. GET BIO

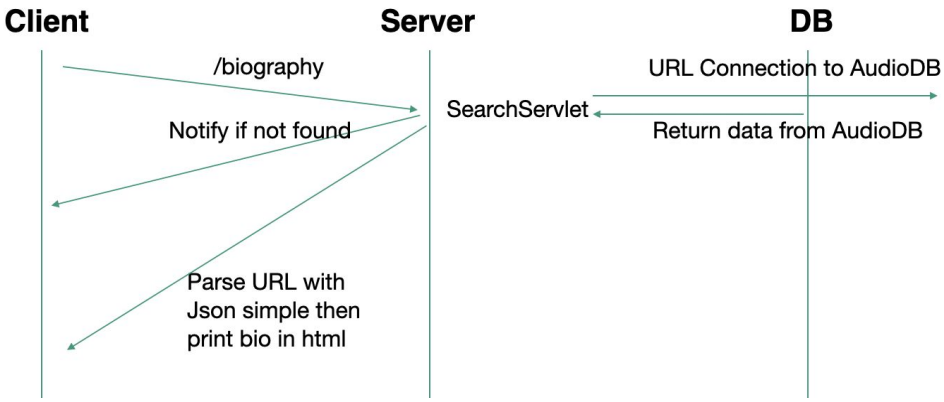
USER'S GUIDE:

Once you clicked the "GET BIO" button on the search page, you'll be redirected to http://localhost:8081/biography where you can read the biography of the artist.
You can go back to the homepage or go to other pages by clicking the website header at the top.

DEVELOPER:

- Client-side sends a request /biography, the servlet checks the cookie to make sure the user is logged in.
- If logged in, BioServlet responds with its doGet method that gets the request parameter "artist_name". If not, sends to login page.
- The server opens a URL Connection that calls out to The AudioDB, get back the read the results into a StringBuilder, and then hand the toString to a simple JSON parser.
- The JSON parser returns a JSON object with the key "artist" that we convert to JSON Array.
- We loop through the JSON array and get the "strBiographyEN" key and put it into a StringBuilder.
- doGet will convert the StringBuilder into an HTML code and print them out.

Task 4: Biography



5. SHOW

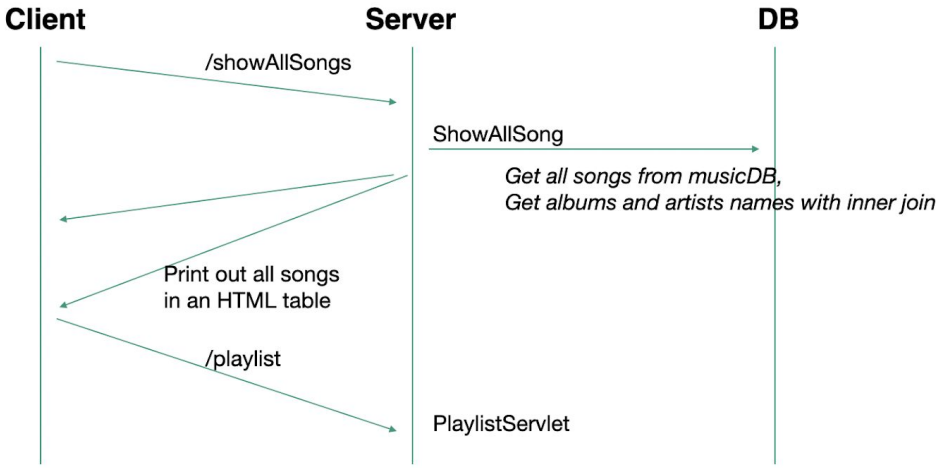
USER'S GUIDE:

The http://localhost:8081/showAllSongs displays all songs in the music.db songs table. When you click on the add button next to a song row, under the playlist column, that row will be added to the playlist table.

DEVELOPER:

- A request /show is sent to the server, which invokes the showAllSongs' doGet servlet.
- The servlet checks the cookie to make sure the user is logged in.
- If they are, the servlet fetches all the songs out of the SQLite DB and returns an HTML table using the inner join on songs, albums, and artists table.
- If they are not logged in, send them to the login.html page.
- Each row in the HTML table has a hidden form input that will send a form action to the Playlist servlet and send the song name, artist name, and album name on that row.

Task 5: Show



6. PLAYLIST

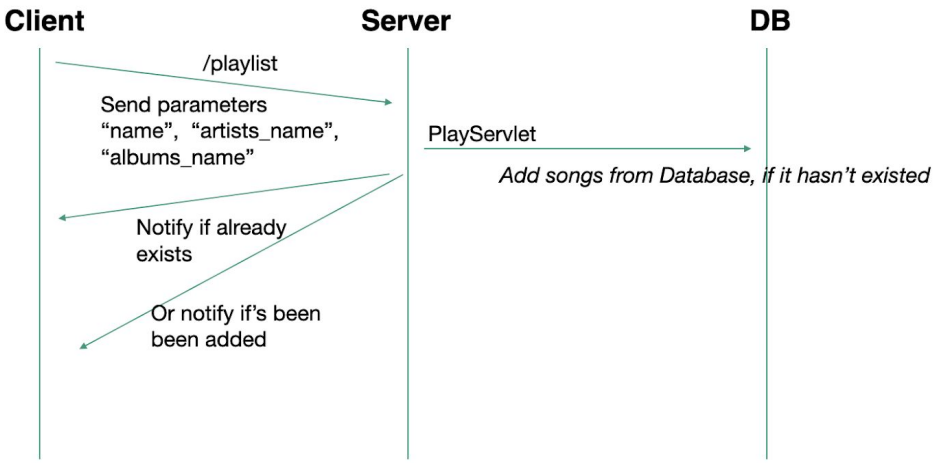
USER'S GUIDE:
You'll be notified if the song has been added or the song already exists in the playlist on http://localhost:8081/playlist.

Go back to the homepage to see if the song is now in your playlist.

DEVELOPER:

- A request /playlist is sent to the server, which invokes the PlaylistServlet' doPost servlet.
- The servlet checks the cookie to make sure the user is logged in. If not send the response to /login
- If logged in, get the request parameter for songs name, artists name, and albums name.
- Open the database and check if the song is already in the playlist.
- If it doesn't exist, add the song to the playlist table and notify the user with a confirmation message. If not also notify the user.

Task 6: Playlist



7. ADD

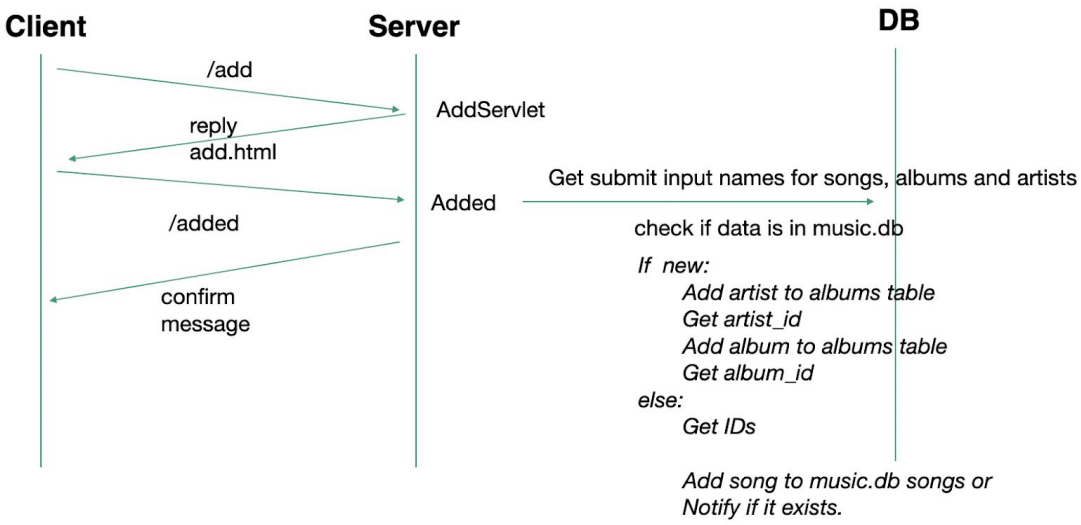
USER'S GUIDE:
The http://localhost:8081/add print out an HTML form to add a song in the song table. Fill in the song name, the album name, and the artist name. When you press the add button, you'll be redirected to http://localhost:8081/added and notified if the song has been added to the database or if the song already exists in the music.db songs table.

Click show to show all songs and check if the song is indeed in the table.

DEVELOPER:

- A request /add is sent to the server, which invokes the AddServlet' doGet servlet.
- Check if logged in, if yes sends an HTML form. If not send the user to the login page.
- The form takes in a song name, an artist name, and an album name.
- Send form information to the Added servlet
- Added servlet use the get parameter to get the song name, artist name, and album name.
- First, Added checks the artists table and see if the artist already exists. If it does, then get the artist id, if not add the artist to the table then get the id.
- Second, Added checks the albums table and see if the album already exists. If it does, then get the album id, if not add the album to the table then get the id.
- Third, Added checks the songs table and see if the songs already exist. If it does, then notify the user, if not add songs to the table using the song name, the artist id, and the album id as values.

Task 7: Add Song



8. DELETE

USER'S GUIDE:

The http://localhost:8081/delete print out an HTML form to delete a song in the song table. Fill in the song name and the artist name. When you press the delete button, you'll be redirected to http://localhost:8081/deleted and notified if the song has been added to the database or if the song already exists in the music.db songs table.

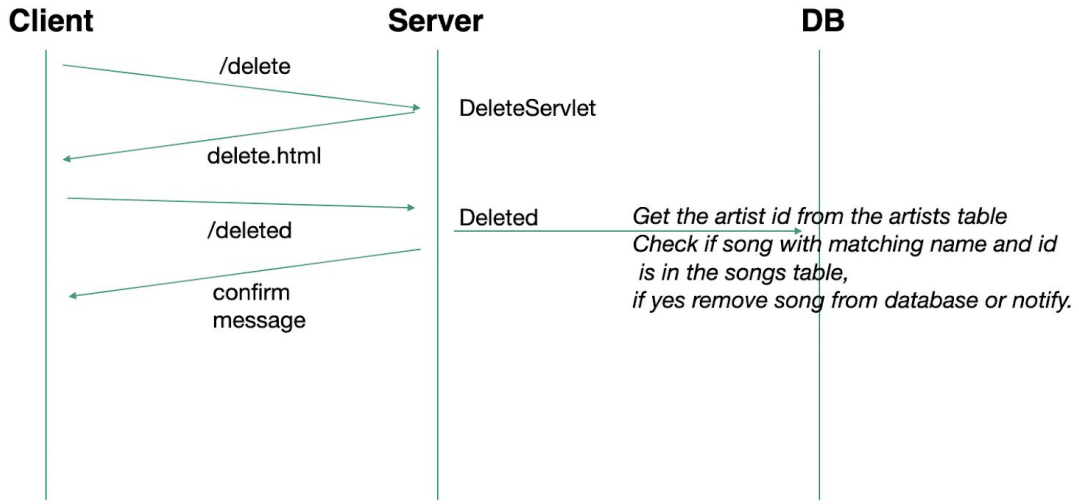
Click show to show all songs and check if the song is no longer in the table.

DEVELOPER:

- A request /delete is sent to the server, which invokes the DeleteServlet' doGet servlet.
- Check if logged in, if yes sends an HTML form. If not send the user to the login page.
- The form takes in a song name and an artist name

- Send form information to the Deleted servlet doPost
- doPost uses the get parameter to get the song name and album name.
- First, Deleted will get the artistID from the artists table
- Second, Deleted check the song table and see if the song with the same song name and artist id exists.
- If it does, then delete the song from the songs table. If not notify the user.

Task 8: Delete Song



9. PLAY

USER'S GUIDE:

The http://localhost:8081/play currently displays 3 songs as an example that the program works. When you click the song title, the music player will change to that song. You can play pause, move forward, or download the song.

DEVELOPER:

- A request /play is sent to the server, which invokes the PlayServlet' doGet servlet.
- Check if logged in, if not send the user to the login page.
- For playing music, PlayServlet gets file names from the files table in the music.db.
- Use the file names to fetch the mp3 from the content directory
- Use javascript to change the music player to the current song on click.

Task 9: Play Song