Playlist Maker

Github Repository URL

Group 4
Kiara Chombo
Jeffrey Saeteros

Date of Submission: 12/12/2023

Table of Work

(Please write x in the boxes to mention what each student achieved in this project)

	Kiara Chombo	Jeffrey Saeteros	
Project Description	X	X	
Uses Cases Diagram(s)	X		
Sequence Diagrams	X		
Class diagram(s)	X		
Implementation	X	X	
Conclusion		X	

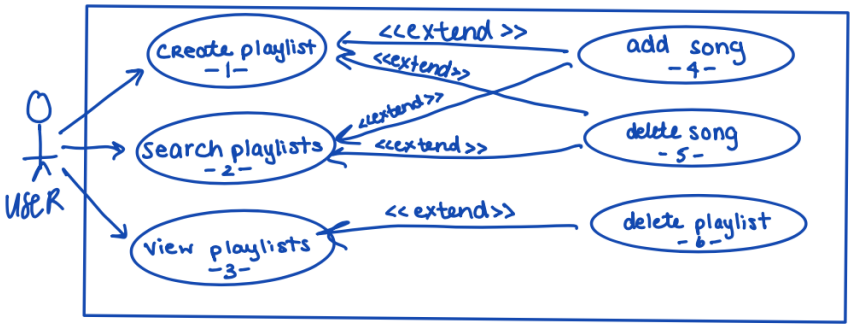
Table of Contents

System Analysis

Project Description:

The system we implemented for our final project is a Playlist Maker that uses MongoDB. Within this application the user will be able to create their own playlists, view the playlist they have already created, and search for existing playlists. Within creating their playlist the user will be required to name their playlist. Once the user chooses a title for their playlist, they are able to manually add songs by inputting the song title and also have the option to delete songs that they added within that playlist. Within the search playlist option, users can search for playlists that they have created previously. If the playlist exists then the user will be brought to that playlist and will have the options to add and delete songs from that playlist. Within the view playlists option, all of the user's created playlists will be displayed. Here, the user has the ability to delete any existing playlists. The goal of this project is to be able to create an application that is easy to use and stimulates the process of creating playlists similar to other platforms like Spotify. The system receives various user inputs through the GUI, such as playlist names, song names, and search queries. Inputs also include requests to our MongoDB database for creating, updating, deleting, and retrieving playlists and songs. The outputs of our program are primarily displayed on the GUI, including the lists of playlists and songs, confirmation messages, and error dialogs. The system outputs changes to our MongoDB database which reflects the user's actions such as adding or deleting songs or playlists. The program interfaces with a MongoDB database for data storage and retrieval. The GUI is designed using Swing and serves as the user interface. The program is constrained by the capabilities of the MongoDB Java driver and the Java Swing framework. Additionally, it relies on a stable network connection for database interactions.

Use Case Diagrams and Descriptions:



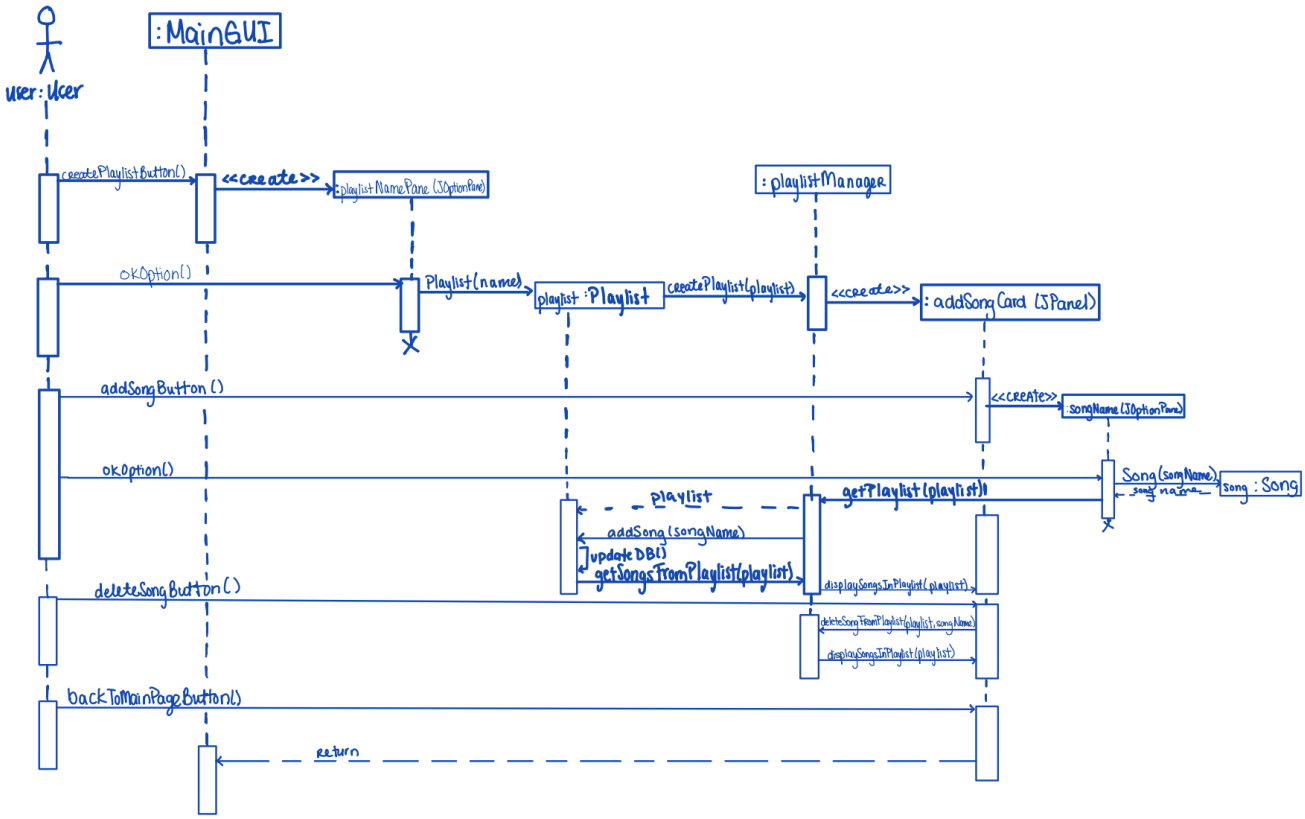
Descriptions:

UC Reference Name/Number: Create Playlist/1 Overview: This usecase allows the user to create new playlists. To create the playlist the user must name the playlist. Related Cases: N/A Actors: user	UC Reference Name/Number: Search Playlists/2 Overview: This usecase allows the user to search for a playlist that already exists in their library. Related Cases: N/A Actors: user	UC Reference Name/Number: View playlists /3 Overview: This usecase allows the user to view all the playlists that they have created Related Cases: N/A Actors: user
UC Reference Name/Number: Add Song /4 Overview: This usecase allows the user to add songs to their playlist during creating the playlist or searching for their playlist. Related Cases: create and search playlist Actors: user	UC Reference Name/Number: Delete Song /5 Overview: This usecase allows the user to delete songs to their playlist during creating the playlist or searching for their playlist. Related Cases: create and search playlist Actors: user	UC Reference Name/Number: Delde Playlist /6 Overview: This usecase allows the user to delete a specific playlist that they have created. Related Cases: create and search playlist Actors: user

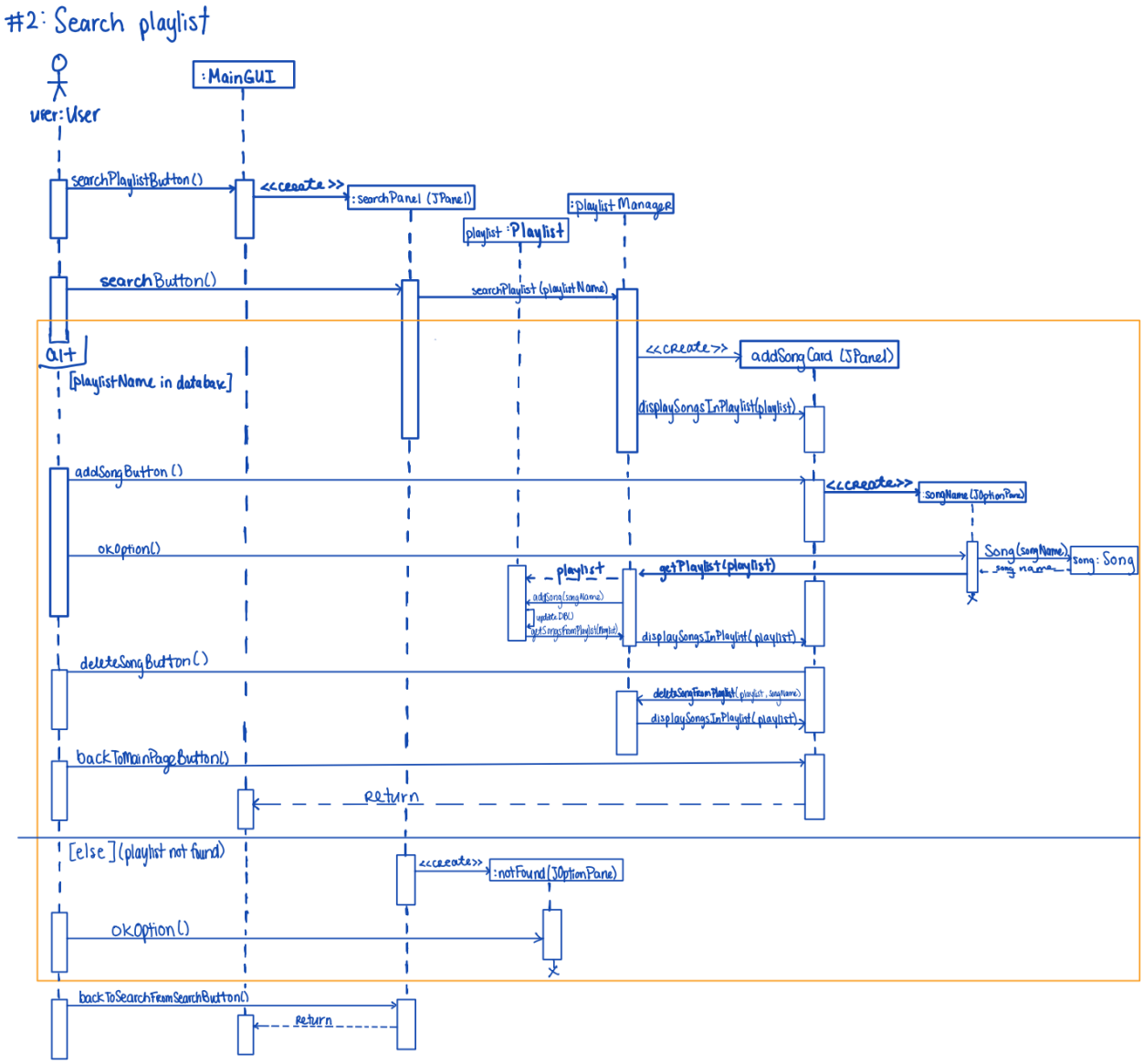
System Design

Sequence Diagrams:

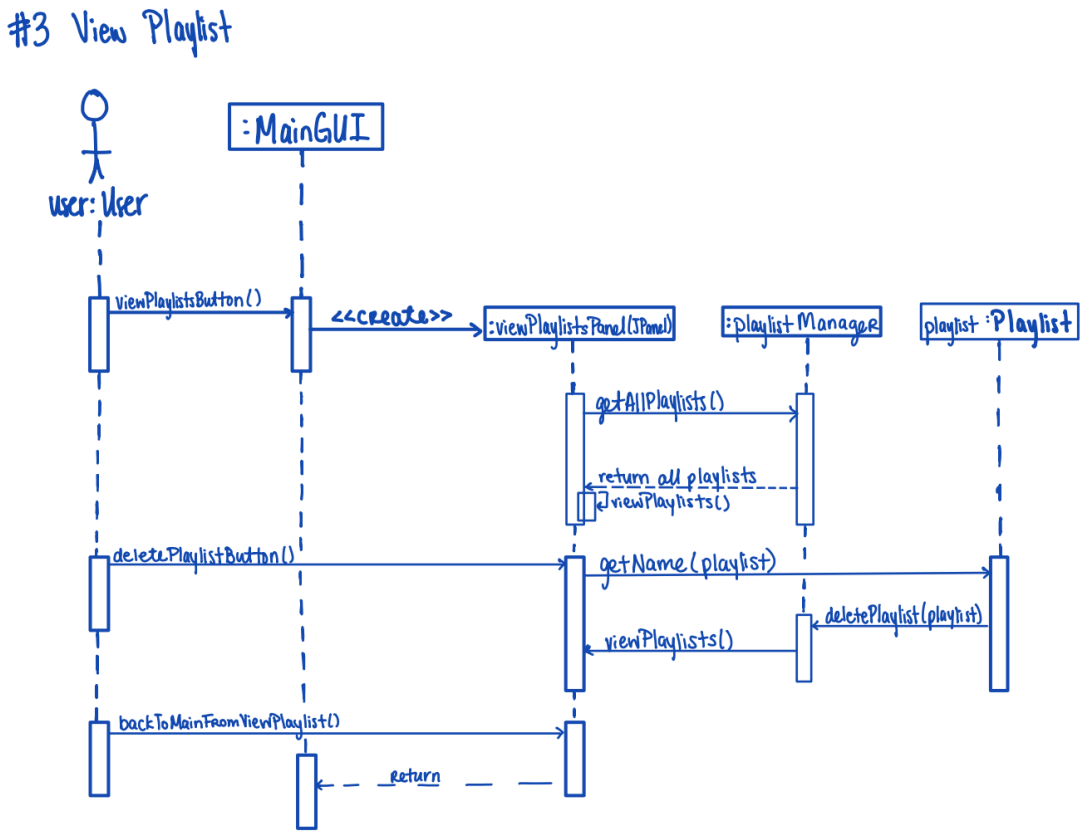
#1 Create Playlist:



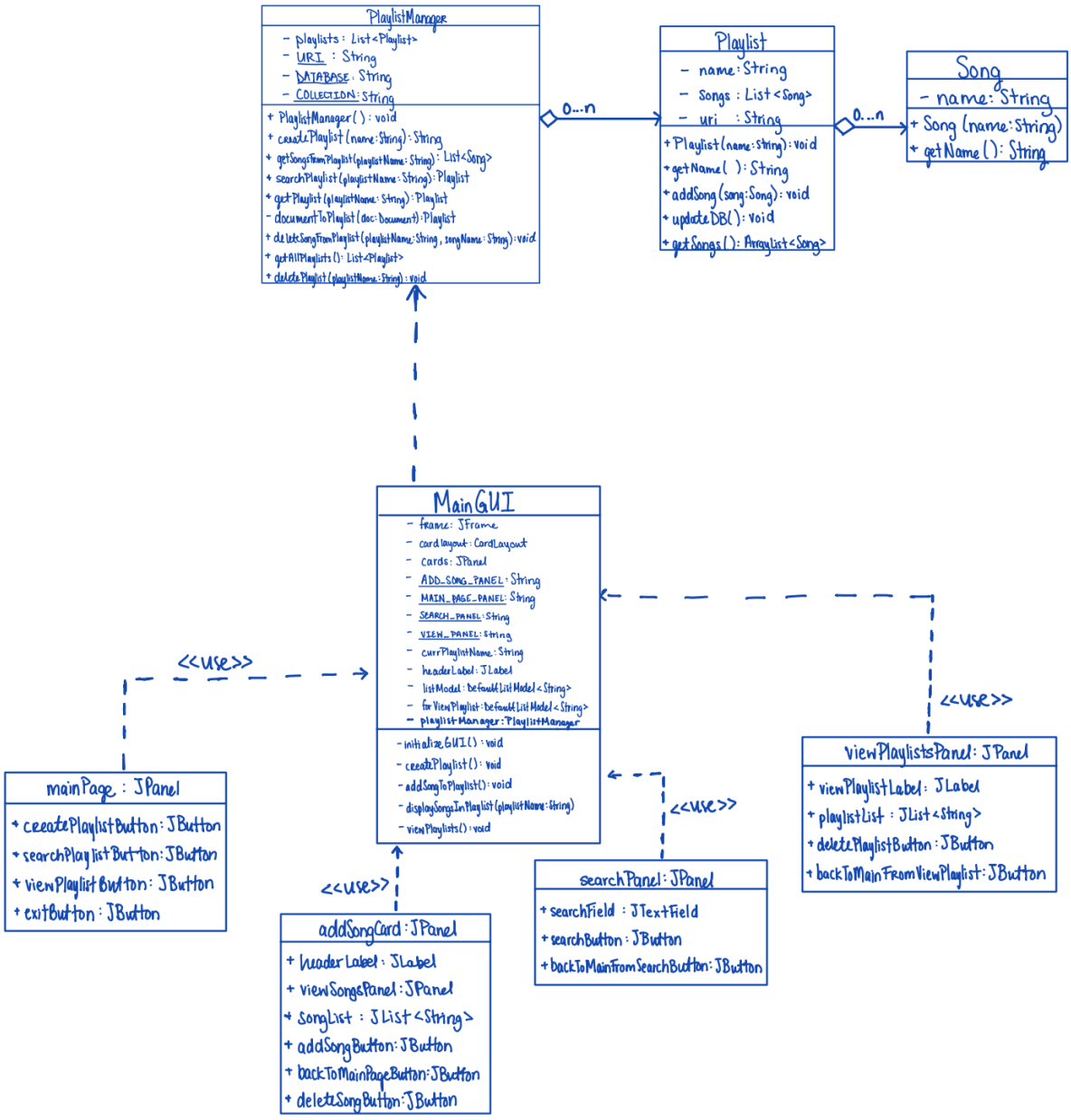
#2 Search Playlist:



#3 View Playlist:



Class Diagrams:



Conclusion

This project, centered around a Java-based playlist management system with a MongoDB backend, showcases the integration of a user-friendly graphical interface with robust database operations. Leveraging the strengths of Java Swing for the GUI and MongoDB for data storage and management, it provides an interactive platform for users to create, view, and delete playlists and songs. The system's design focuses on responsiveness and reliability, ensuring a seamless user experience while managing musical collections. Through this project, key aspects of software development, including user interface design, database connectivity, and application logic, are effectively brought together, demonstrating a practical application of programming skills in creating a useful and efficient tool for music organization and management.