

# **PROGETTO TECNOLOGIE INFORMATICHE PER IL WEB**

Playlist Musicale RIA

Camillo Nicolò De Sabbata -10667697

Elia Fantini - 10651951

## Analisi dei dati

Analisi della specifica: **pages**, **components**, **events**, **actions**.

Un'applicazione web consente la gestione di una Playlist di brani musicali. Playlist e brani sono personali di ogni utente e non condivisi. Ogni brano musicale è memorizzato nella base di dati mediante un titolo, l'immagine e il titolo dell'album da cui il brano è tratto, il nome dell'interprete (singolo o gruppo) dell'album, l'anno di pubblicazione dell'album, il genere musicale (si supponga che i generi siano prefissati) e il file musicale. L'utente, previo login, può creare brani mediante il caricamento dei dati relativi e raggrupparli in Playlist. Una Playlist è un insieme di brani scelti tra quelli caricati dallo stesso utente ordinati per data decrescente dall'anno di pubblicazione dell'album. Una Playlist ha un titolo e una data di creazione ed è associata al suo creatore. A seguito del **login**, l'utente **accede all'Homepage** che presenta l'**elenco delle proprie Playlist**, ordinate per data di creazione decrescente, una **form** per **caricare un brano** con tutti i dati relativi e una **form** per **creare una nuova Playlist** inizialmente vuota. Quando l'utente **clicca su una Playlist** nell'Homepage, **appare** la pagina **Playlist page** che contiene inizialmente una **tabella** di una riga e cinque colonne. Ogni cella contiene il titolo di un brano e l'immagine dell'album da cui proviene. Se la Playlist è inizialmente vuota compare un messaggio: *"La Playlist non contiene ancora brani musicali"*.

## Analisi dei dati

I brani sono ordinati da sinistra a destra per data decrescente dell'album di pubblicazione. Se la Playlist contiene più di cinque brani, sono disponibili comandi per vedere il precedente e successivo gruppo di brani. Se la pagina Playlist mostra il primo gruppo e ne esistono altri successivi nell'ordinamento, compare a destra della riga il **bottone Successivi**, che permette di **vedere il gruppo successivo**. Se la pagina Playlist mostra l'ultimo gruppo e ne esistono altri precedenti nell'ordinamento, compare a sinistra della riga il **bottone Precedenti**, che permette di **vedere i cinque brani precedenti**. Se la pagina Playlist mostra un blocco ed esistono sia precedenti sia successivi, compare a destra della riga il bottone Successivi e a sinistra il bottone Precedenti. La pagina Playlist contiene anche una **form** che consente di **selezionare e aggiungere un brano** alla Playlist corrente. A seguito dell'aggiunta di un brano alla Playlist corrente, l'applicazione visualizza nuovamente la pagina a partire dal primo blocco della Playlist. Quando l'utente **seleziona il titolo di un brano**, la pagina **Player** mostra tutti i **dati del brano** scelto e il **player audio** per la riproduzione del brano.

## Database Design

Database:

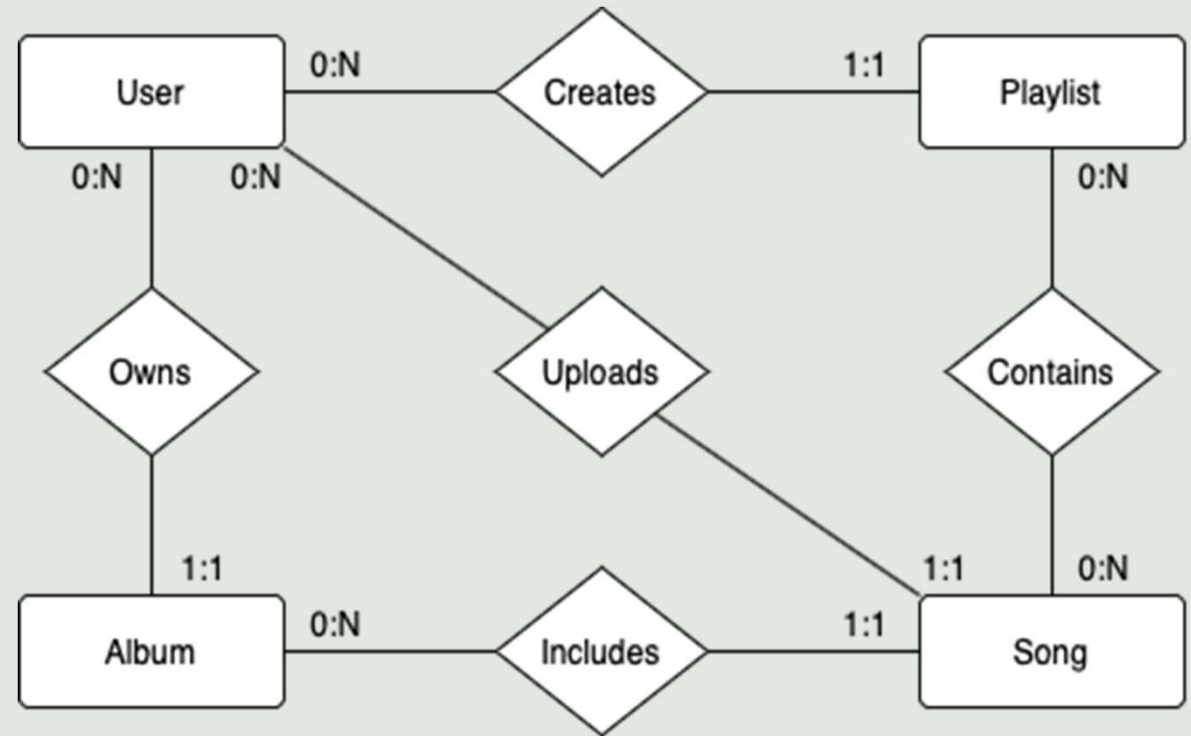
Album (album\_ID, album\_title, interpreter, publication\_year, image\_path, user\_ID)

Containment (containment\_ID, song\_ID, playlist\_ID, ordinal)

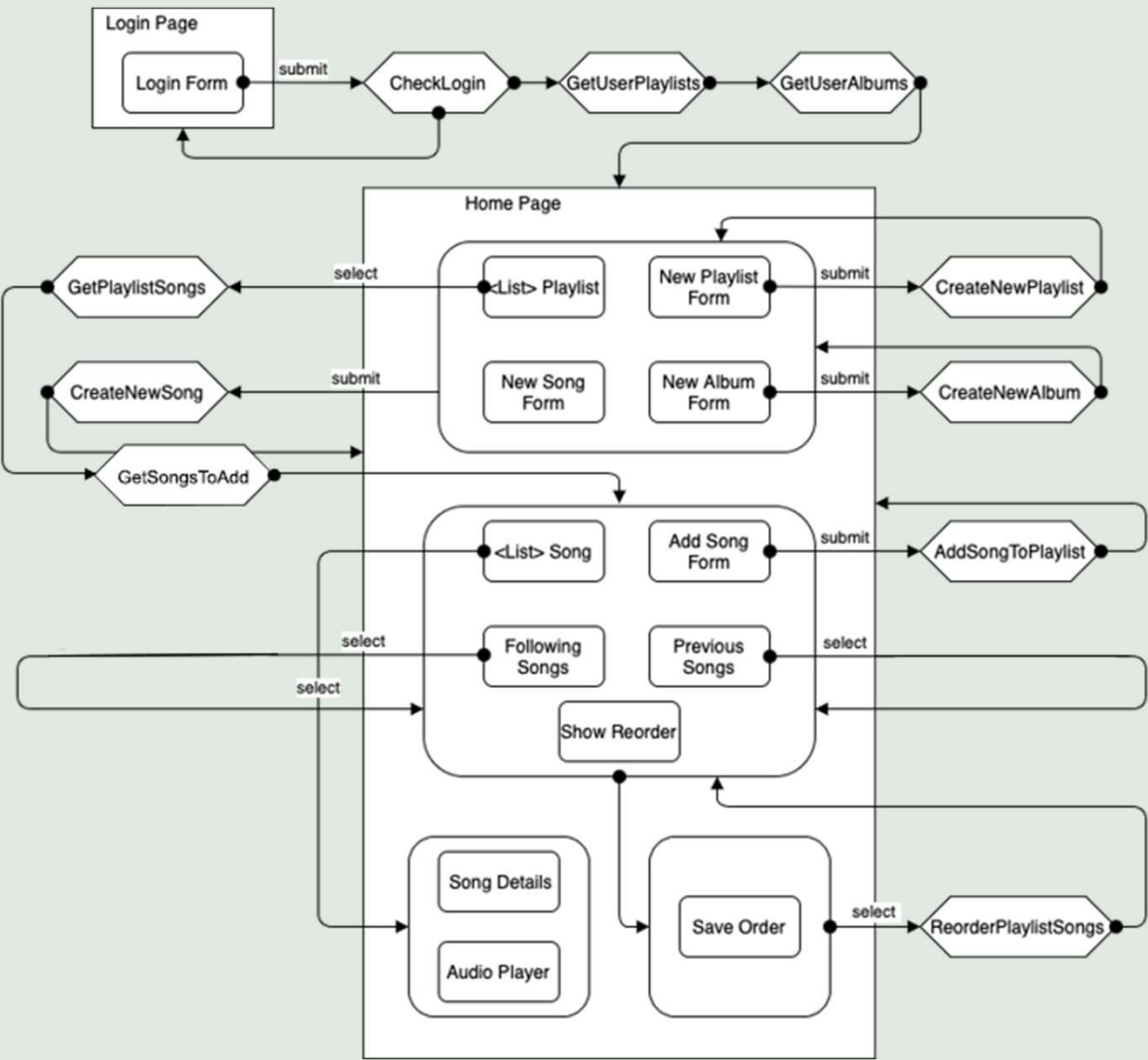
Playlist (playlist\_ID, playlist\_name, creator\_ID, creation\_date)

Song (song\_ID, song\_title, album\_ID, genre, file\_path, owner\_ID)

User (user\_ID, username, password)



# Application Design



## Server Components

### Model Objects (Beans):

User  
Playlist  
Song  
Album

### Views (templates):

Login  
Home Page

### Controllers (Servlets):

AddSongToPlaylist  
CheckLogin  
CreateNewAlbum  
CreateNewPlaylist  
CreateNewSong  
GetUserPlaylists  
GetUserAlbums  
GetPlaylistSongs  
GetSongsToAdd  
ReorderPlaylistSongs

### Data Access Objects:

UserDAO:

- checkCredentials(username,password)

PlaylistDAO:

- createNewPlaylist(playlistName, creationDate, creator)
- findPlaylistByUser(userId)
- checkPlaylist(userId, playlistId)

ContainmentDAO:

- createNewContainment(playlistId, songId, ordinal)
- findSongsByPlaylist(playlistId)
- changeSongOrdinal(ordinal, songId, playlistId)

SongDAO:

- createNewSong(songTitle, albumId, genre, userId, filePath)
- findSongsToAdd(userId, playlistId)
- checkSong(userId, songId)
- findSongById(songID)

AlbumDAO :

- createNewAlbum(albumTitle, interpreter, publicationYear, userId, filePath)
- findAlbumByUser(userId)
- checkAlbum(userId, albumId)

## Client Components

### Index

LoginForm:

- Gestione del submit e degli errori

### Homepage

UserPlaylists

- reset(): imposta le condizioni di iniziali visibilità dei vari sotto-componenti
- show(): richiede al server le playlists dell'utente
- update(): mostra le playlists dell'utente

NewAlbum form

- Gestione della creazione di un nuovo album

NewSong form

- show(): richiede al server gli album dell'utente
- update(): mostra le opzioni di caricamento della canzone
- Gestione del caricamento di una canzone

NewPlaylist form

- reset(): imposta le condizioni di iniziali visibilità dei vari sotto-componenti
- Gestione della creazione di una nuova playlist

PlaylistSongs

- reset(): imposta le condizioni di iniziali visibilità dei vari sotto-componenti
- show(): richiede al server le canzoni della playlist ad un dato

indice

- update(): mostra le canzoni della playlist ad un dato indice
- saveOrder(): salva nel server l'ordine attuale delle canzoni
- showReorder(): mostra la lista completa delle canzoni permettendo di riordinarle
- showForm(): richiede al server le canzoni che possono essere aggiunte
- updateForm(): mostra il form per l'aggiunta di canzoni alla playlist corrente

AddSong form

- Gestione dell'aggiunta di una canzone alla playlist

SongPlayer

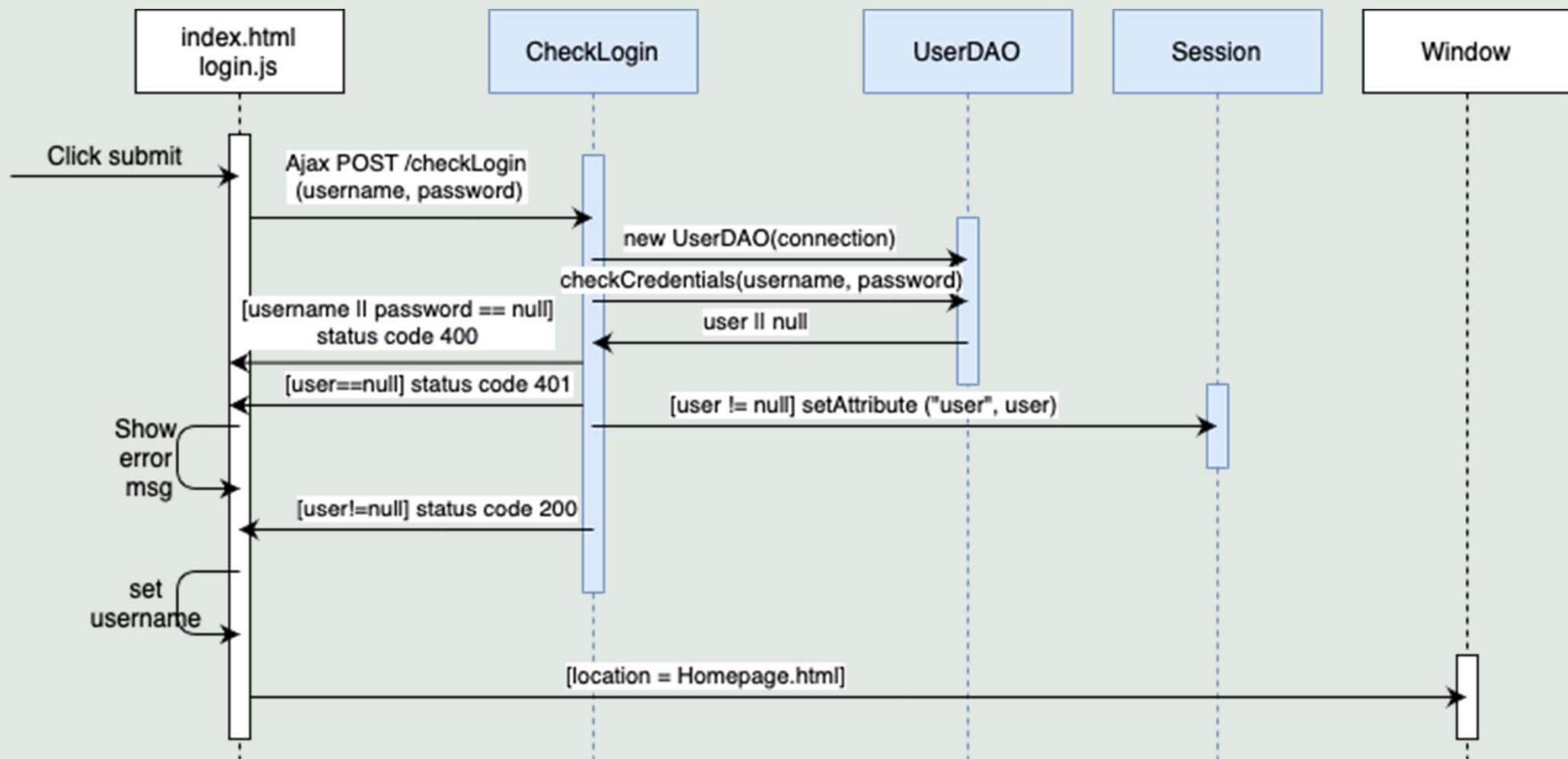
- reset(): imposta le condizioni di iniziali visibilità dei vari sotto-componenti
- show(): cerca la canzone da mostrare
- update(): mostra i dettagli e il player audio della canzone

PageOrchestrator

- start(): inizializza i componenti della pagina
- refresh(): gestisce il ricaricamento della pagina portandola alla situazione iniziale

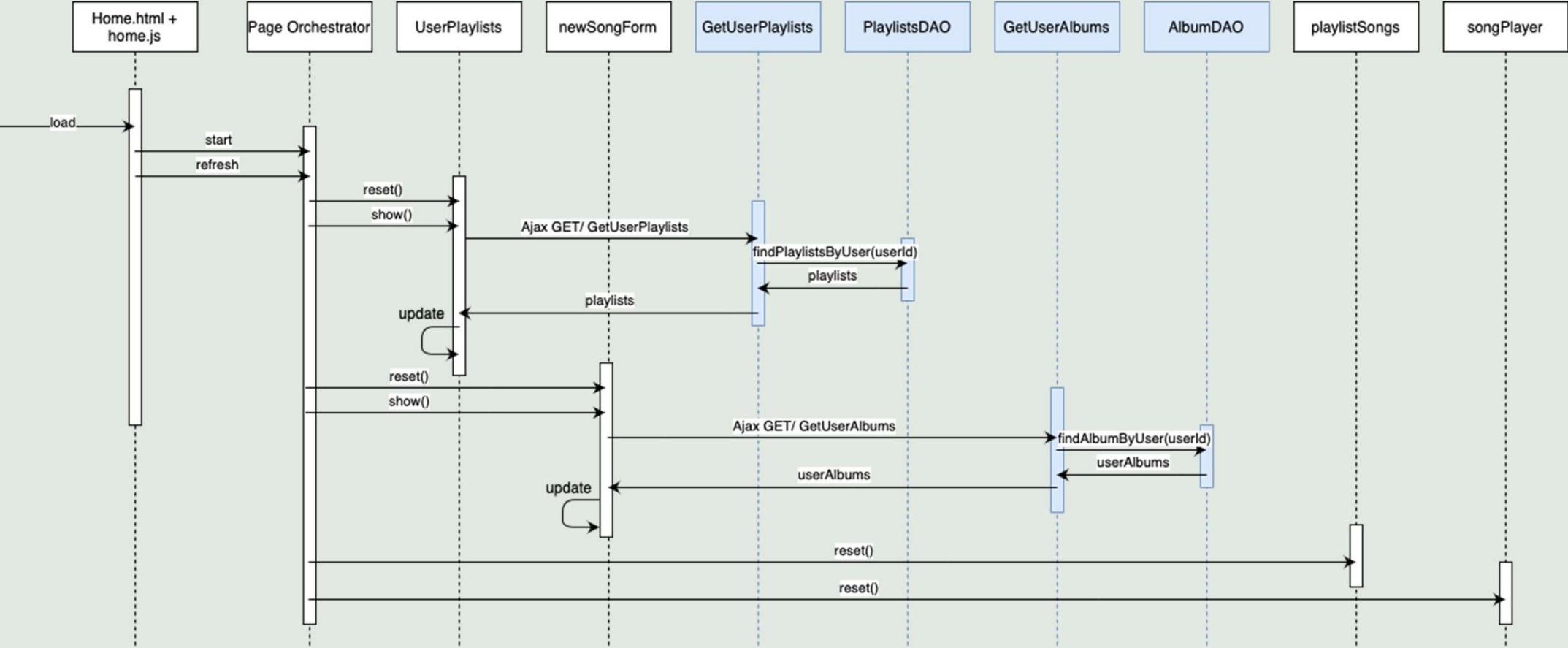


## Login

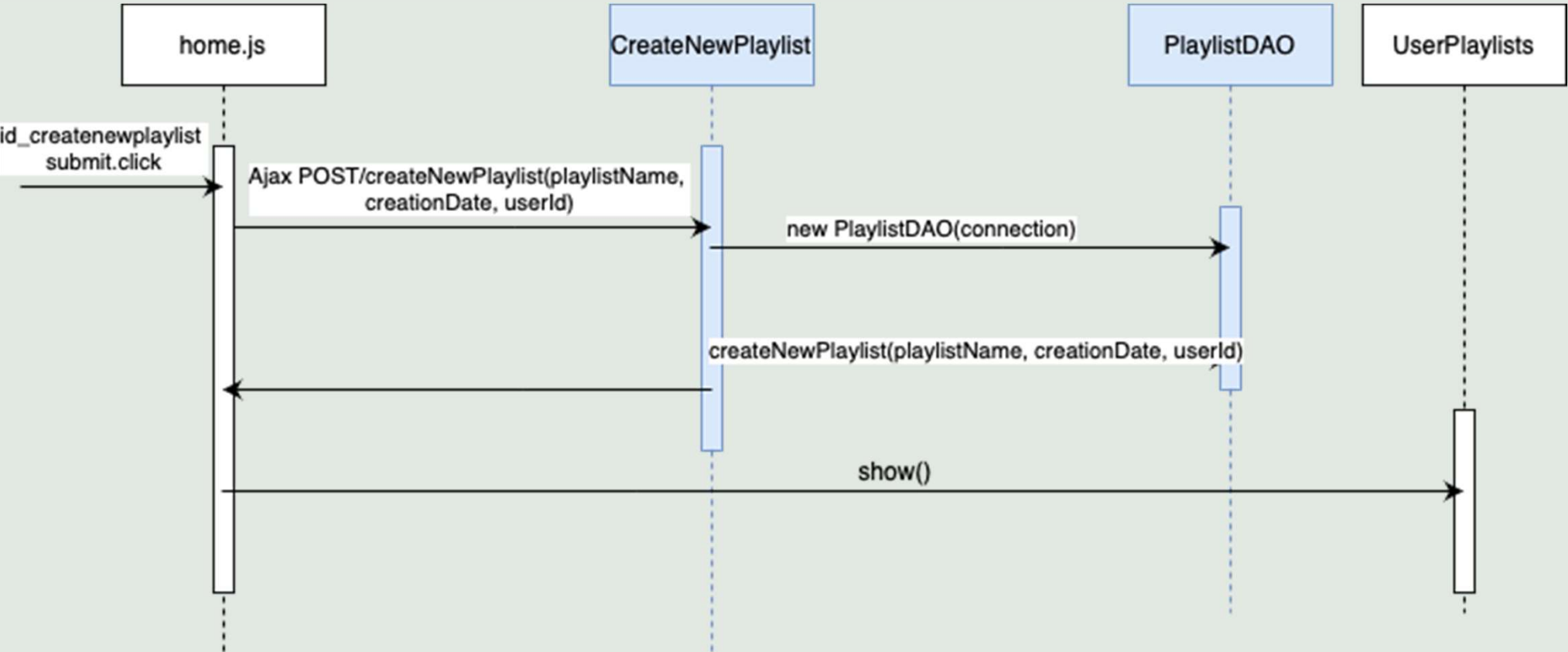




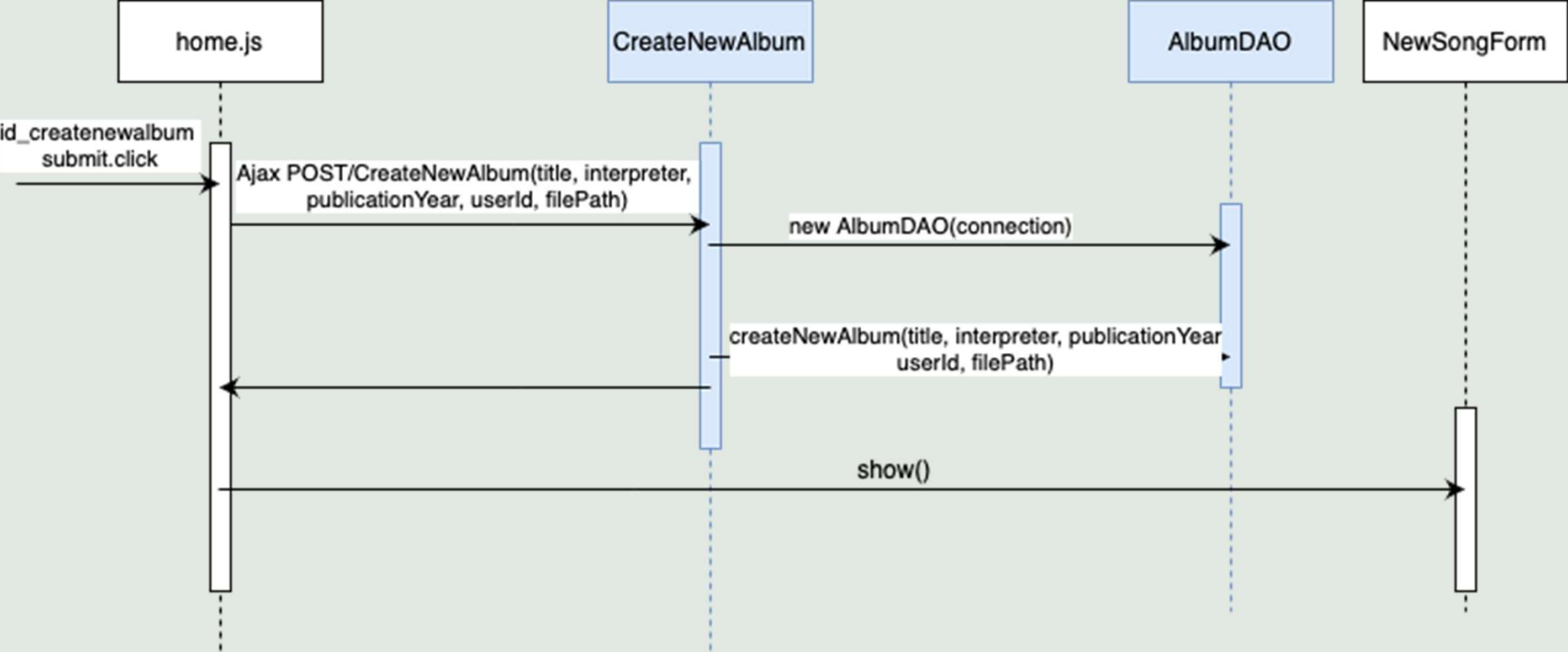
# Load Homepage



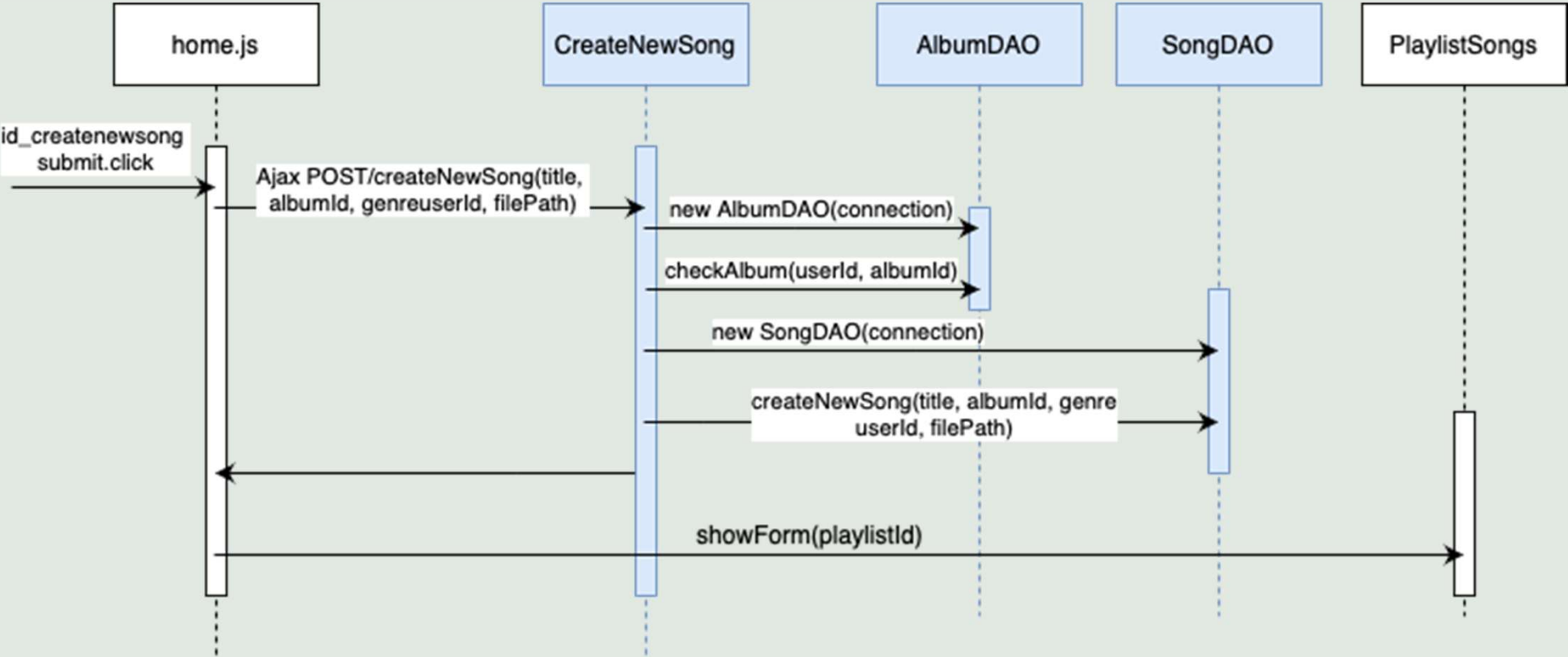
# Create New Playlist



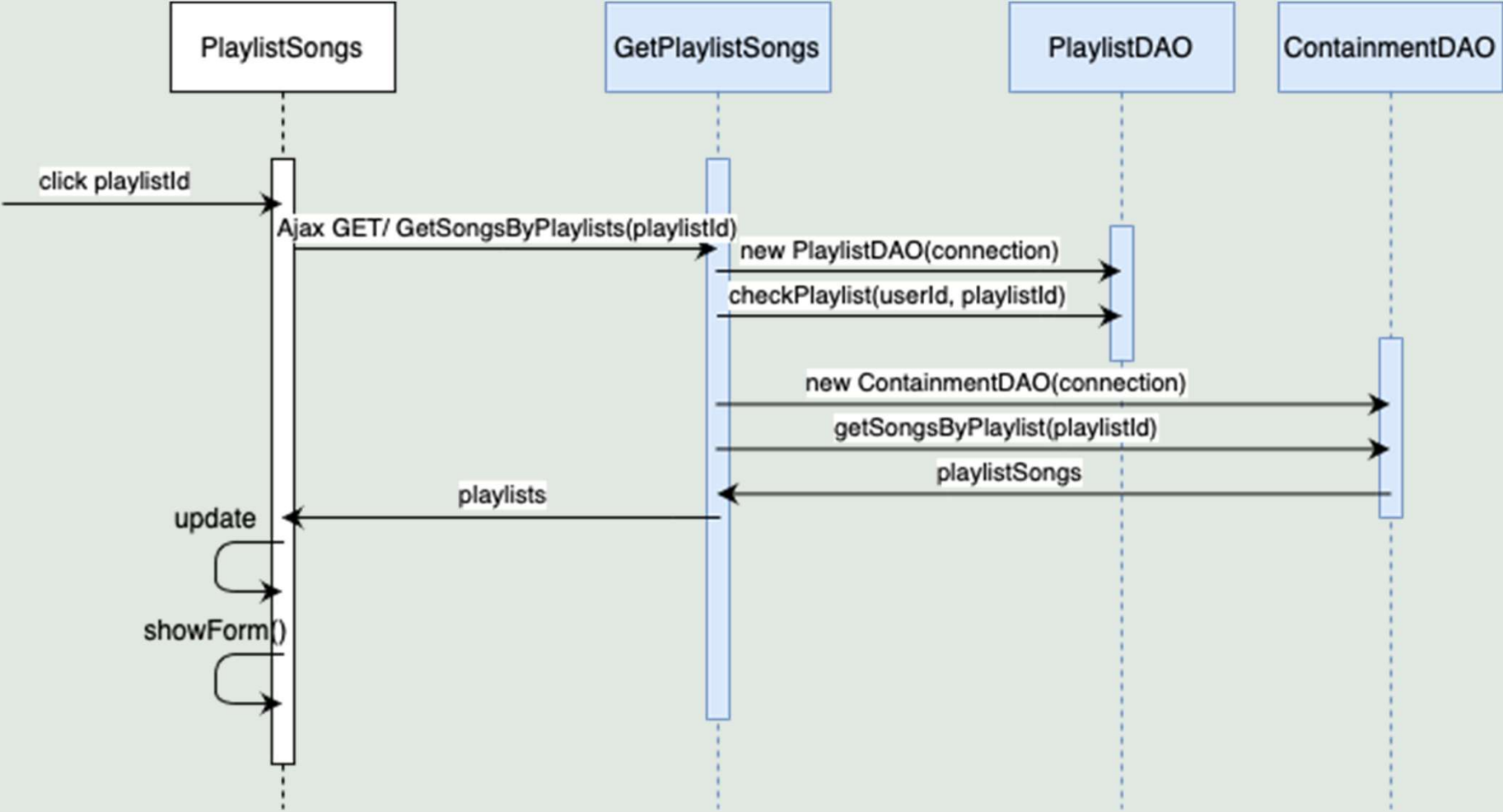
# Create New Album



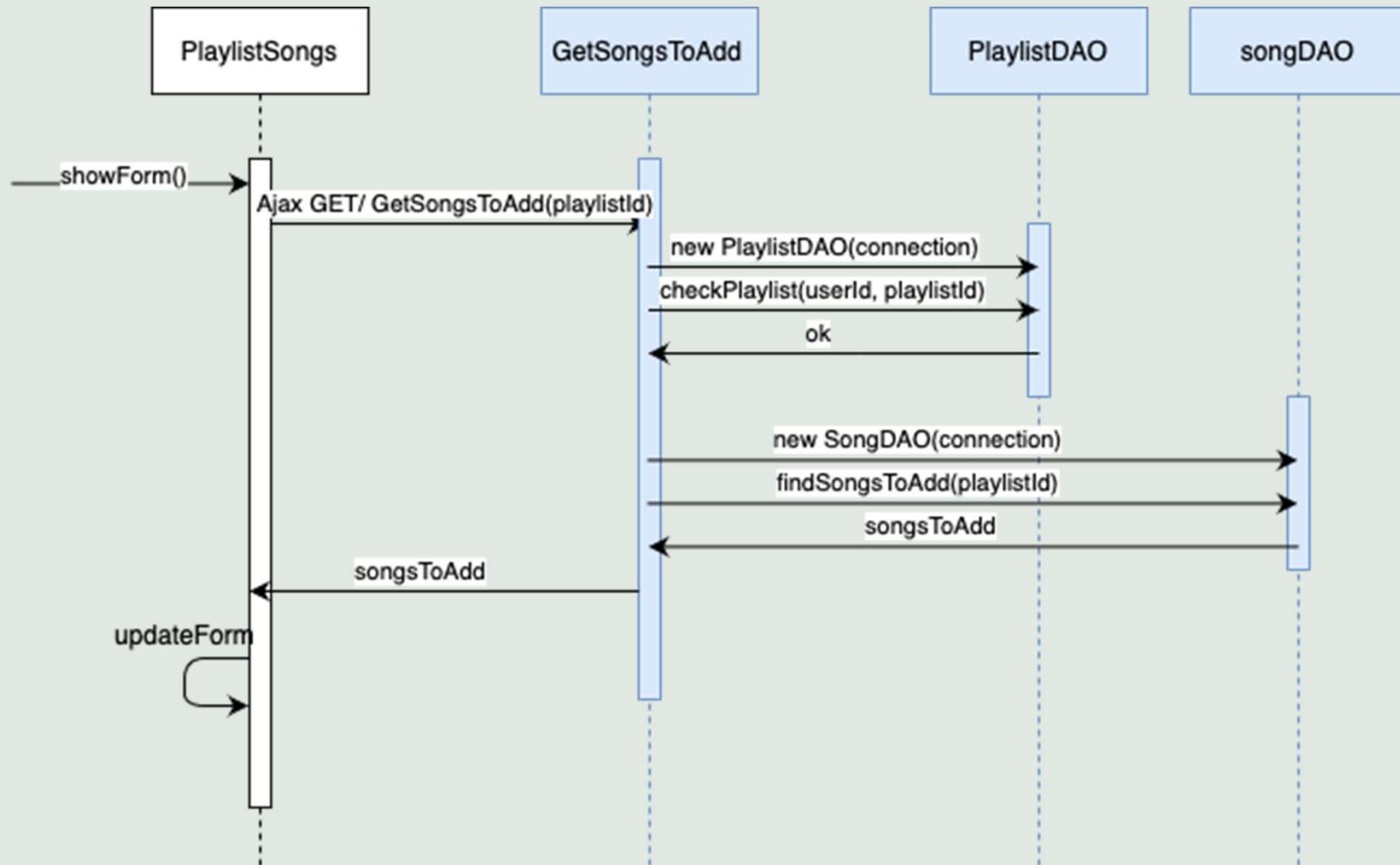
# Create New Song



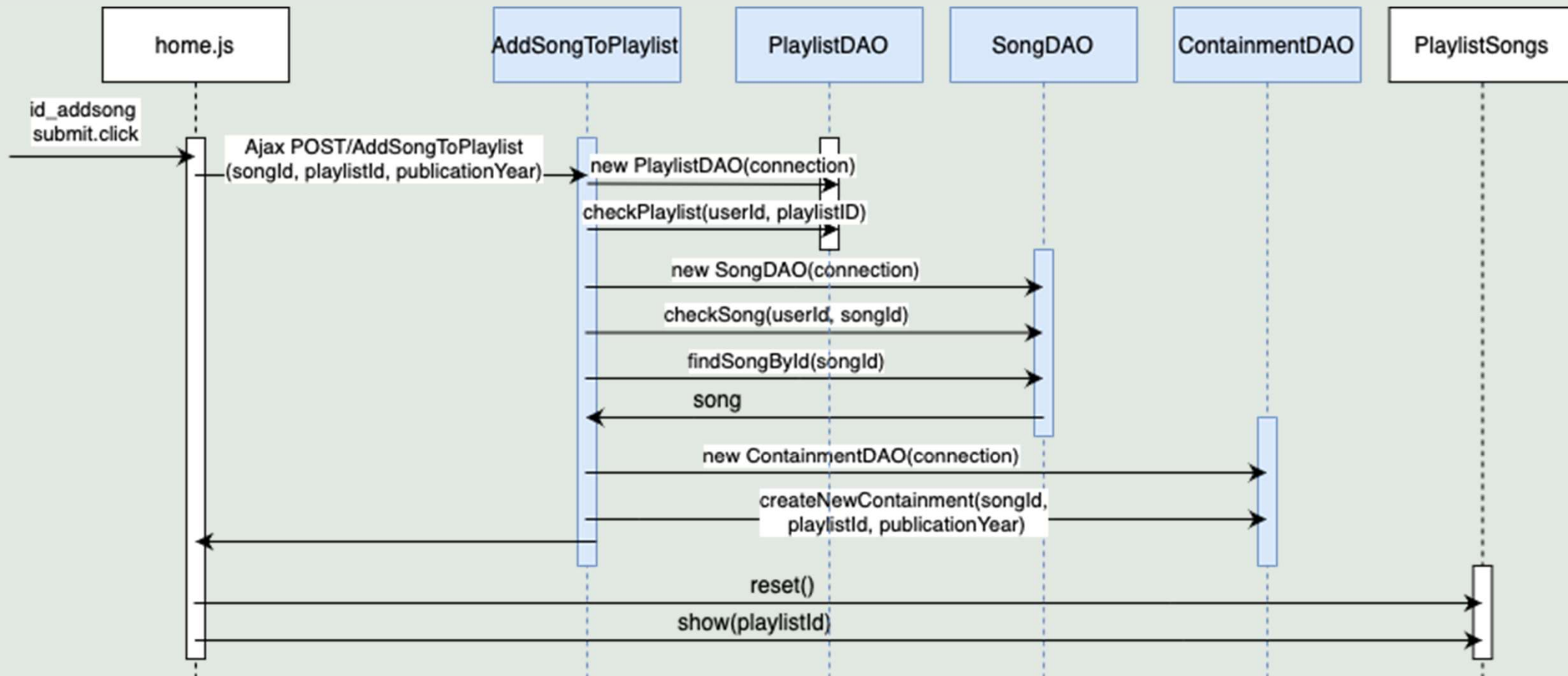
# Get Playlist Songs



## Get Songs To Add



## Add Song To Playlist





# Reorder Songs

