

Playlist Musicale

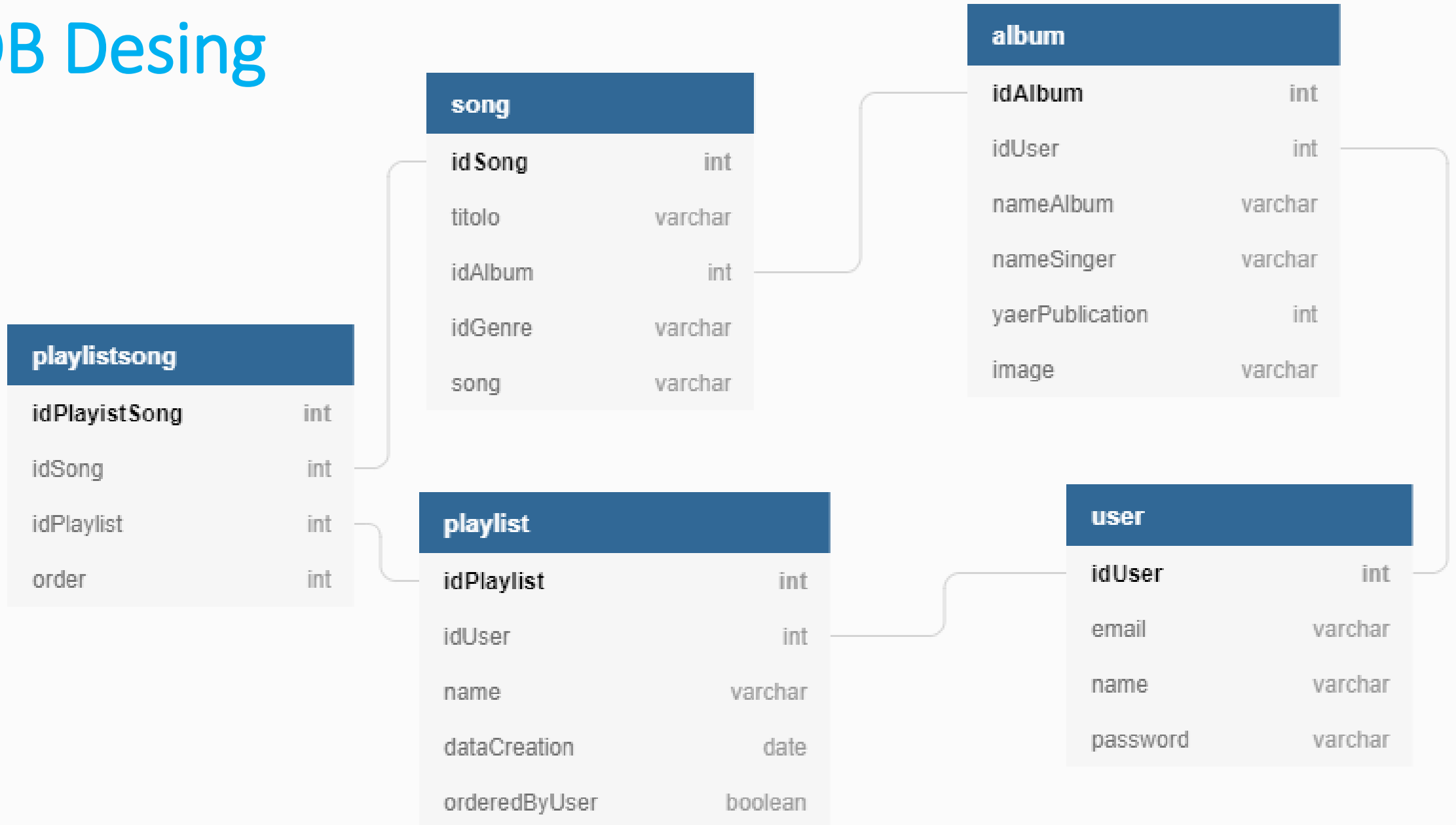
RIA

Cordioli Matteo

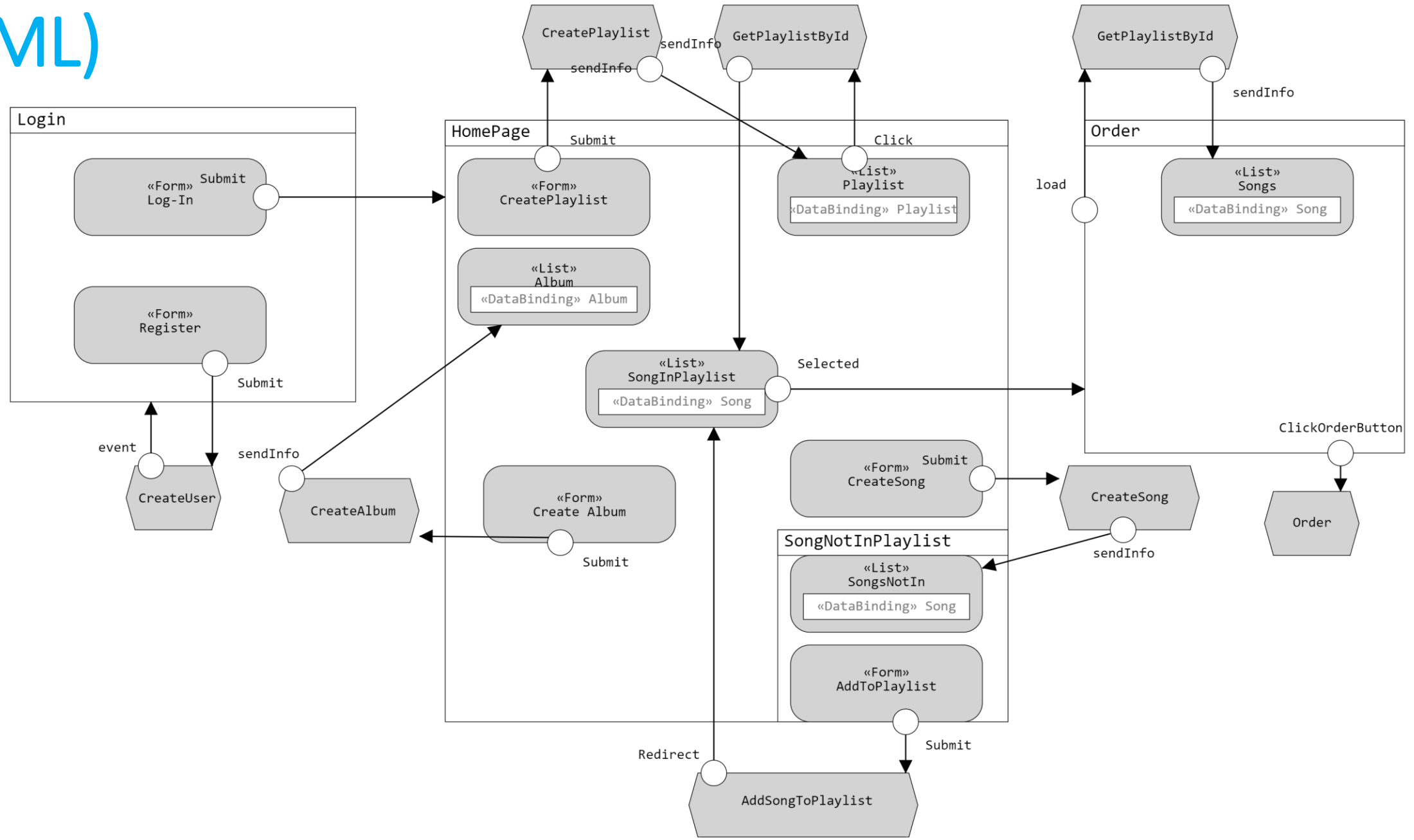
10611332

913598

DB Desing



Application Desing (IFML)



Components:

Beans:

- User
- Playlist
- Song
- Album

Controller

- Login (Post)
- Createuser (Post)
- CreatePlaylist (Post)
- CreateAlbum (Post)
- CreateSong (Post)
- AddSongToPlaylist (Post)
- GetFile (Get)
- GetAlbums(Get)
- GetPlaylists(Get)
- GetPlaylistById(Get)
- GetSongNotIn(Get)
- ModifyOrderSong(Post)
- Logout(Get)

*Session
User*

Html Pages:

- Index
- Home
- Order

CSS:

- Default
- Reset
- Css
- CssJs

JavaScript:

- Utility
- LoginManager
- PlaylistManager
- OrderManager

Utility:

- CheckVariable
- ConnectionManager
- ErrorManager

DAO:

- UserDao
 - newUser(name, password, email)
 - getUserWithId(idUser)
 - existUserWith(email, password)
- PlaylistDAO
 - playlistOfUser(idUser)
 - newPlaylist(name)
 - getPlaylist(idPlaylist)
 - getAllSongOfPlaylist(idPlaylist)
 - getAllSongOfNotIn(idUser, idPlaylist)
 - **updatePlaylistOrdered(idPlaylist)**
- SongDAO
 - newSong(idAlbum, name, file, genre)
 - getSongWithId(idSong)
- AlbumDAO
 - newAlbum(name, singer, cover, datePublication)
 - albumOfUser(idUser)
 - getAlbumWithId(idAlbum)
- PlaylistSongDAO
 - newLinkSongPlaylist(idSong, idPlaylist)
 - **updateSongOrder(idSong, order, idPlaylist)**

Components:

Session:
nameUser
idPlaylist
namePlaylist

- Utility:
 - Contains the function used everywhere (es. MakeCall)
- LoginManager:
 - EventListener associate to the button in the index page, to Login and to Register
- PlaylistManager:
 - PageOrchestrator: load the default settings in the start function, manage the not in playlist song box and the album option
 - ListPlaylist: manage the list of playlist, autoclick during onLoad event the first one and color the selected one
 - ListSong: manage the songs of the current playlist, next and prec event, selected one and color the selected one
 - Song: manage the current song, load the mp3 audio file
 - Manage all the buttons for newSong, newPlaylist, newAlbum, orderPlaylist
- OrderManager:
 - PageOrchestrator: load the default settings in the start function and load the songs of the playlist selectee in the homepage
 - OrderList: build the list and attach the drag and drop event
 - Manage the button of submit to save the data

Event Handler:

Client Side		Server Side	
Event	Action	Event	Action
Index→ loginForm→ Submit	Check bad format, makeCall	POST loginForm	Check bad format, save user if valid in session
Index→ registerForm→ Submit	Check bad format, makeCall	POST registerForm	Check bad format, insert user in db
HomePage→load	Update: playlists and songs lists, songSelected	GET (no param, already in sesion)	Get playlists of curr user Get songs of the first playlist Get songs of curr user not in the first playlist
HomePage→ PlaylistList→ clickPlaylist	Update UI in PlaylistList, update SongList and songSelected	GET idPlaylist	Check bad format Get songs of 'idPlaylist' Get songs of curr user not in 'idPlaylist'
HomePage→ SongList→ clickSong	Update UI in SongList and songSelected	-	-
HomePage→ SongNotIn→ submit	Update SongList and SongNotIn	POST idSong, idPlaylist	Check bad format, insert in db

Client Side		Server Side	
Event	Action	Event	Action
HomePage→ AlbumForm→ submit	Check bad format, makeCall	POST AlbumForm	Check bad format, insert album in db
HomePage→ SongForm→ submit	Check bad format, makeCall	POST SongForm	Check bad format, insert song in db
HomePage→ PlaylistForm→ submit	Check bad format, makeCall	POST PlaylistForm	Check bad format, insert playlist in db
HomePage→ OrderPlaylist→ click	Set windows location Order.html	-	-
HomePage→ Logout→ click	Set Windows location index.html, clear session	-	-
Order→ load	Update orderSongList	GET idPlaylist	Check bad format, Get songs of 'idPlaylist'
Order→ saveOrder→ click	Check bad format	POST idPlaylist, orderForm	Check bad format Update songs of 'idPlaylist' Update 'idPlaylist'
Order→ Logout→ click	Set Windows location index.html, clear session	GET (no param)	Invalidate session