Ryerson University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Title:	Software Design Architecture
Course Number:	COE692
Semester/Year (e.g.F2016)	W2024

Instructor:	Faezeh Ensan

Assignment/Lab Number:	3
Assignment/Lab Title:	

Submission Date:	February 25, 2024
Due Date:	February 25, 2024

Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
Cao	Andy	501105786	02	A.C
Chiu	Danielle	501116551	02	D.C

*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: http://www.ryerson.ca/senate/current/pol60.pdf

Sequence Diagram

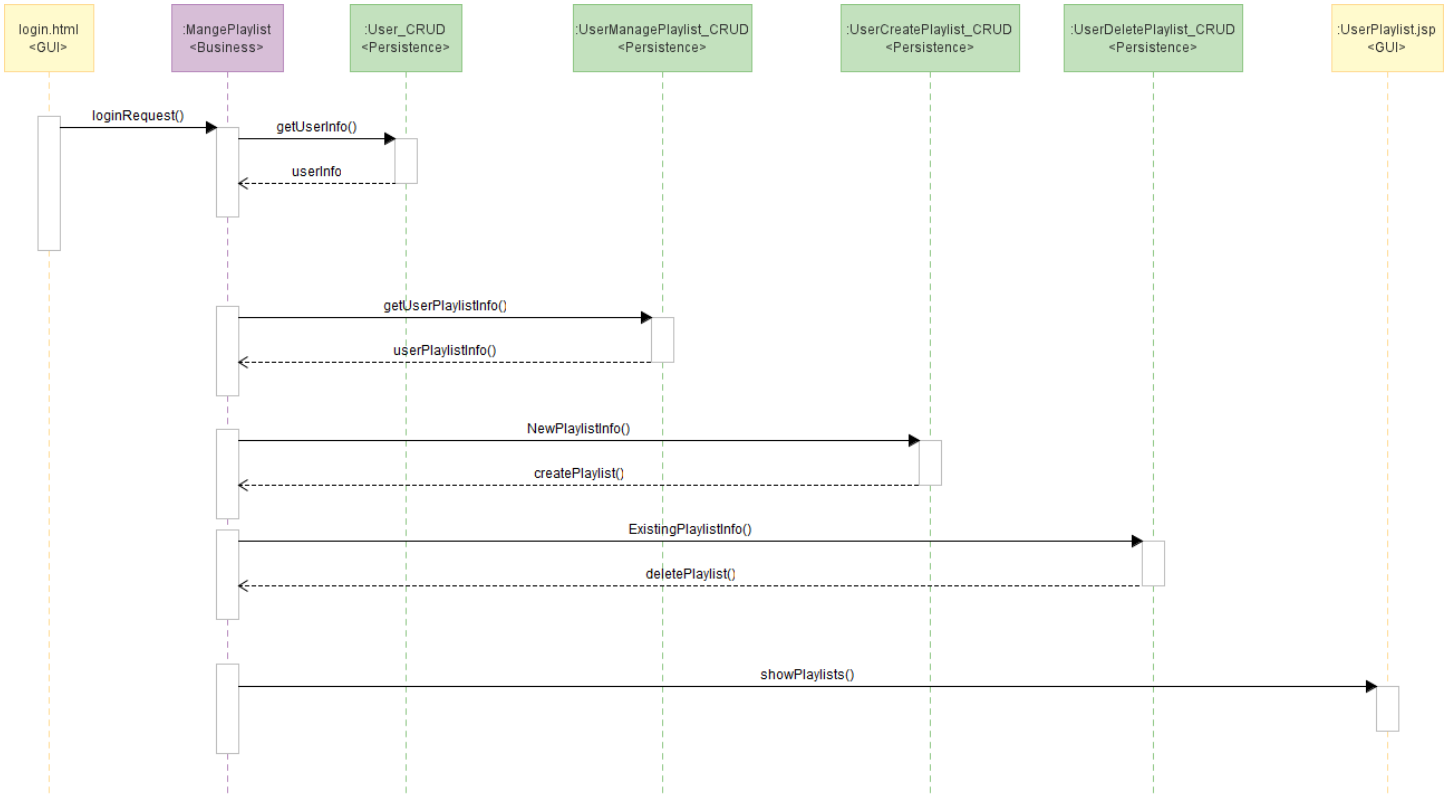

Figure 1: Sequence Diagram for use cases used in the prototype

Description

As shown in figure 1 above, this is a sequence diagram for the use cases that were created in the prototype during the previous lab. The use cases include the user login, the playlist manager, and the create playlist function. For the sequence diagram it starts off with the user login, this means that the system would need to be provided a username and password. The information would then be extended to User_CRUD to verify with the system, once successful, it would log the user in. Next, when the user selects the option to manage playlist, the extend object would have to get all the users existing playing list using the getUserPlaylistInfo() method from the UserManagePlaylistCRUD in the persistence layer. After all existing playlists are listed, the user would then be given the options to create or delete a playlist. Depending on the option chosen, the extend would call on NewPlaylistInfo() or ExistingPlaylistInfo() and the UserCreatePlaylist_CRUD and UserDeletePlaylist_CRUD would use that info to create or delete a playlist. Once that is all done, the extend would call the UserPlaylist,jsp and display the updated playlist that is on the user account.

Component Diagram

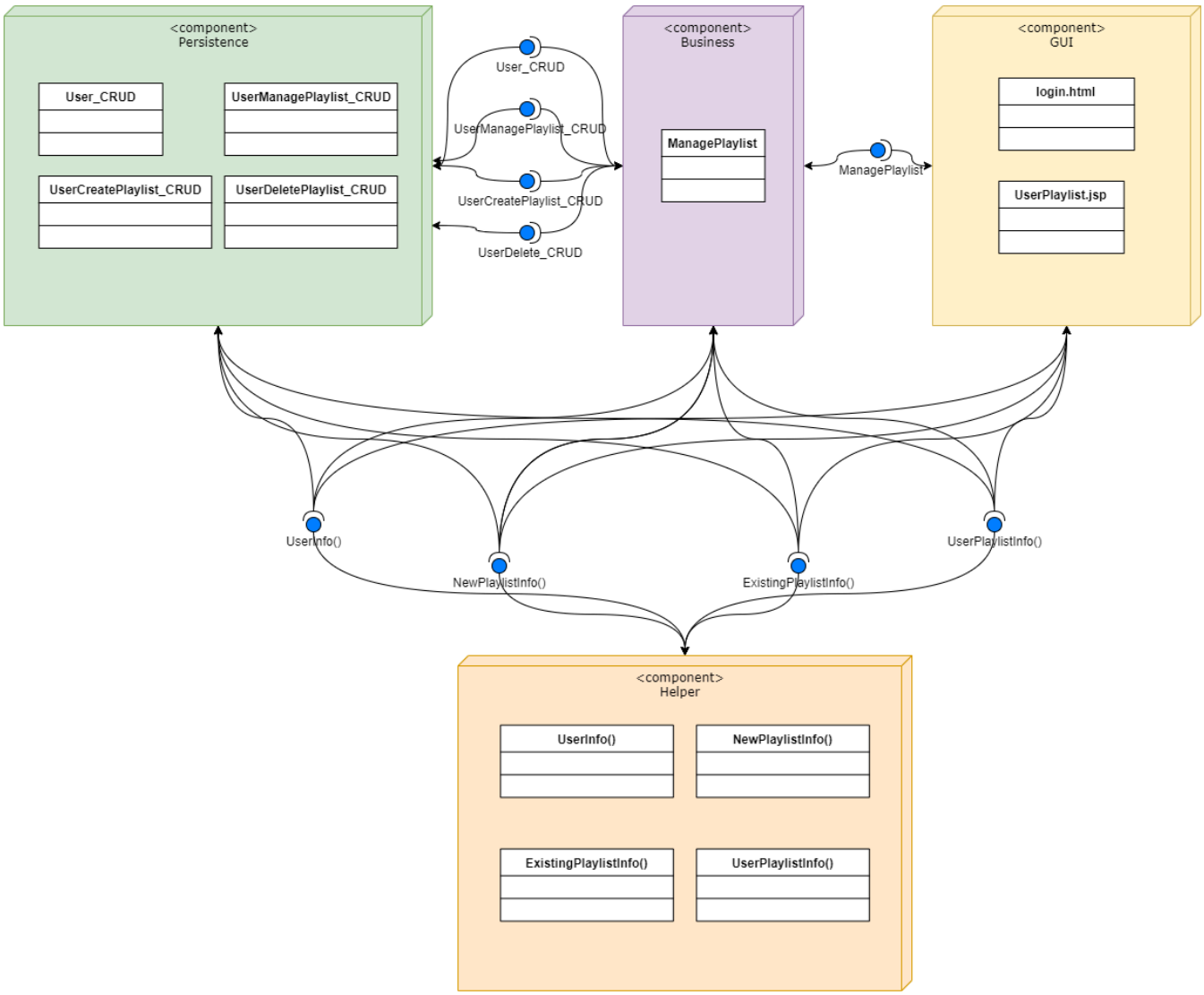

Figure 1: Component Diagram for use cases used in the prototype

Description

Currently the component consists of four main components:

- The first component is the persistence component, this component includes all the classes that are used to query the database and manage the information within it. This includes the functions of creating, reading, updating, and deleting tables and rows from the User and playlist created by the user
- The second component is the business component, this one includes the classes for managing the playlist
- The third one is the GUI component, this includes all the html forms and jsp files that are need for the system to display and interact with the user
- The last one is the helper component, this one includes all the classes used for passing and storing data between the other components.