Activity 3 – Presentation Layer Patterns

Ryan Coon

CST-361

Professor Mark Smithers

May 29, 2022

Screenshots:

Graphical user interface, application

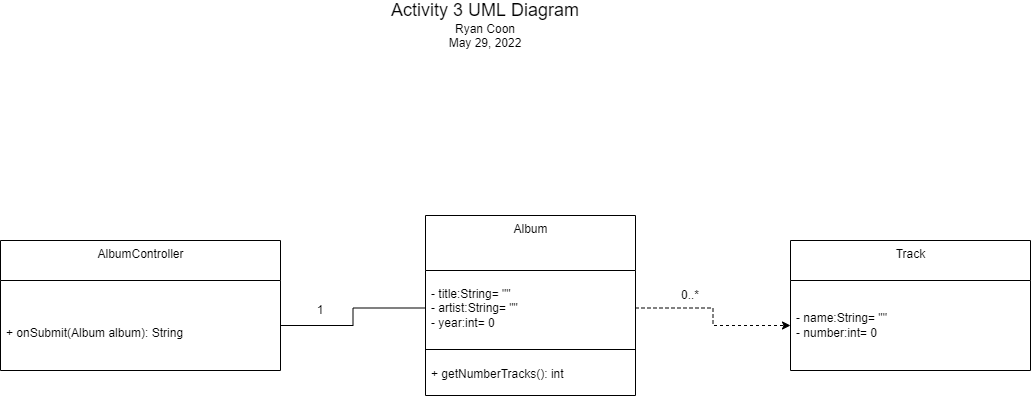
Description automatically generated

Graphical user interface, application, Word

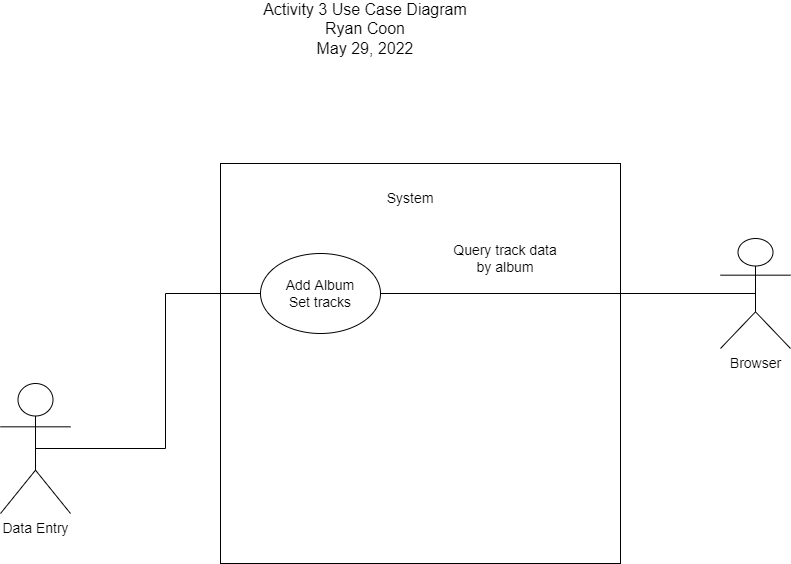
Description automatically generated

Design Report:

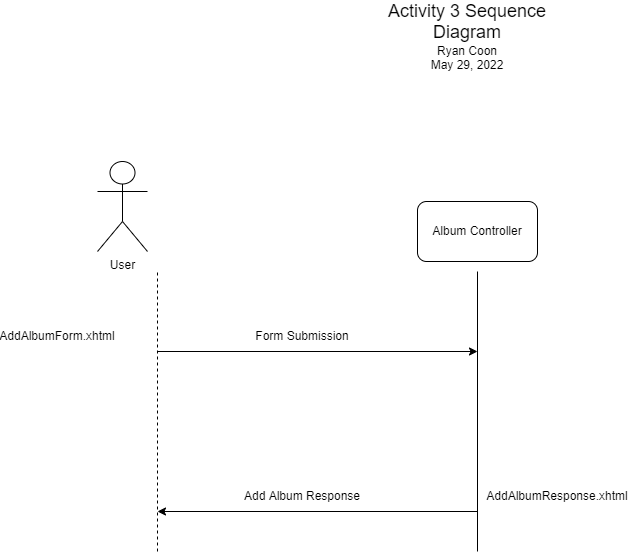
UML Class Diagram



Use Case Diagram



Sequence Diagram

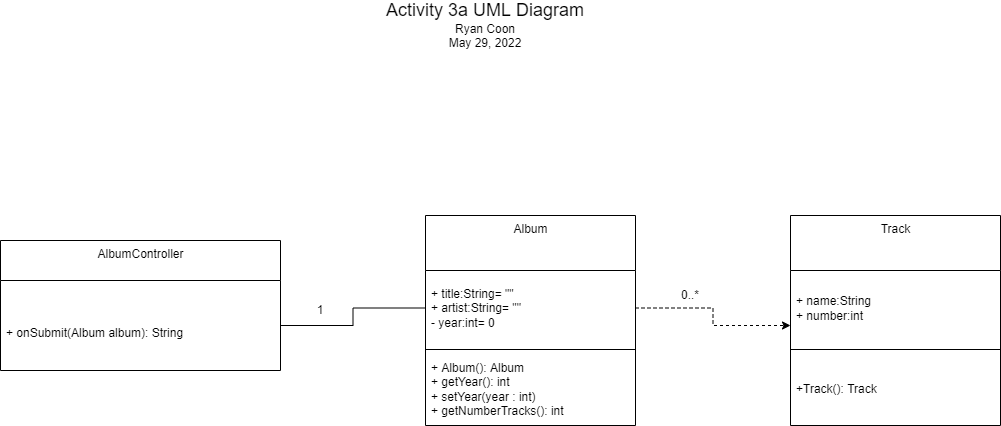


MVC Design Pattern:

The MVC or Model-View-Controller design pattern allows for the separation of concepts between the individual parts of an application. The model elements are representations of the data that will be manipulated through requests. The View portion controls what will be displayed to the user after all the request processes have been completed. The Controller portion manages the requests from a user, deciding what business logic or data operations to apply before sending a designated response back to the user agent. In activity 3, the Track and Album classes are parts of the Model layer. The AlbumController class is a part of the Controller layer. Lastly the JSF templating pages (AddAlbumForm.xhtml and AddAlbumResponse.xhtml) are a part of the View layer of the application.

Activity 3a

UML Class Diagram



Use Case Diagram

Diagram

Description automatically generated

Sequence Diagram

Diagram

Description automatically generated

MVC Type I

MVC Type I involves tight coupling of the view and controller layers. All of the business logic and session management logic are embedded directly into the templating pages themselves.

MVC Type II

MVC Type II is a pattern that moves all of the business logic out from the presentation layer and moves it into the controller layer(in a servlet class).

MVC for Assignment 3

The type of MVC used for this assignment was MVC Type II. Following the prompt, we divided the business logic and connected it to the backing bean behind the UI form using a controller servlet. The resulting UI page only knows what element is accessible because the servlet class paced the Album class into the persistent FacesContext object.