**CMPS 312 – Mobile Application Development Final Practical Exam – 6 December 2023 (duration 2h)**

**Objective:** Showcase your mobile development skills by crafting an Android application named “FilmFusion” designed for Movie exploration.

**Task Overview:**

1. Sync the cmps312-lab repository and copy the Final Exam folder to your GitHub repo, which contains the base project named **FilmFusion** with all the necessary dependencies.
2. Implement a **FilmFusion** app that allows users to explore Movies by genres. Manage the app data in an SQLite database using Room Library.

# **Implementation Tasks:**

1. **Entity-Relations: [10]** Create two Entity classes with proper annotations named **Movie** and **Genre** as shown in the figure 1. Ensure that the IDs are auto-assigned by the database. Add foreign key annotations with cascade delete and update on the appropriate entity.

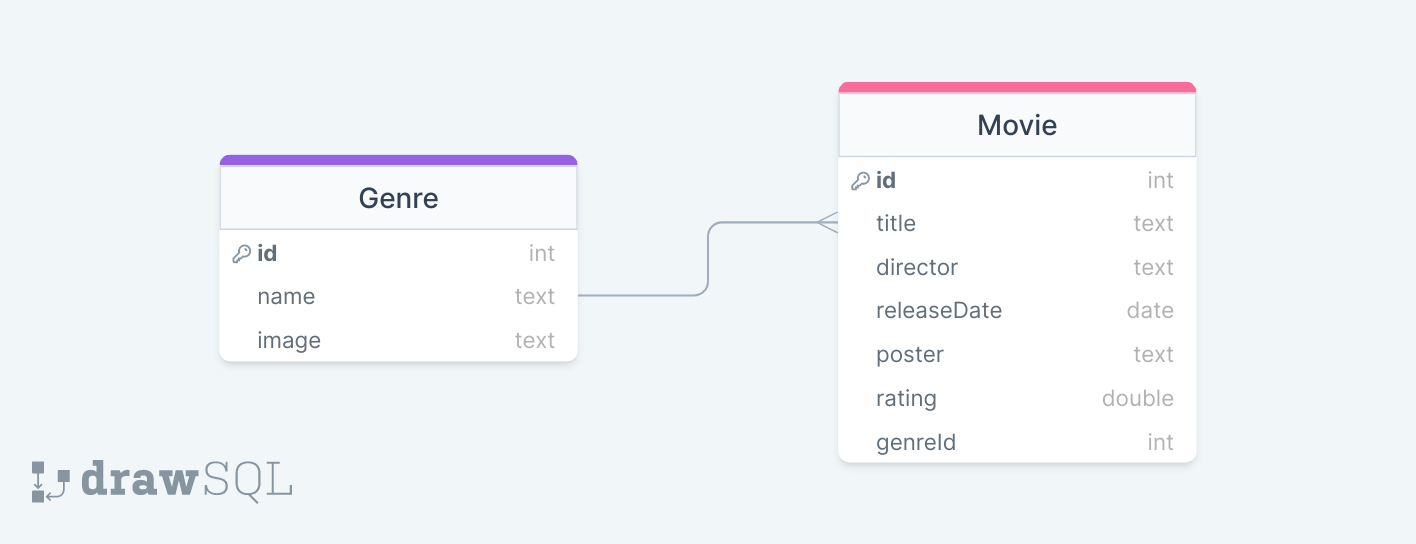


Figure FilmFusion ER Diagram

1. **FilmFusionDao Interface and FilmFusionDatabase [30]** Create the database class and the FilmFusionDao Interface with necessary functions and annotations to allow users to:

* Get all movies for specific Genre
* Add a new movie
* Delete a movie
* Update a movie
* Get a movie
* Add a new Genre
* Get all genres
* Delete a genre and all associated Movies

1. **FilmFusionRepository [10]** Implement the FilmFusionRepository with all necessary methods listed above. Also, inside the repository, initialize the database if it is empty using the provided JSON files.
2. **FilmFusionViewModel [20]** Create a FilmFusionViewModel to serve as an intermediary between the UI and the repository. The ViewModel should:

* Manage the interaction between the UI and the repository for Movie and genre data.
* Provide Flows and Observable data to observe changes in the underlying data.
* Handle user inputs and interactions, ensuring proper validation and data integrity.

1. **App UI and Navigation [30]** Implement the UI and navigation as described below:
   * **Initial Genre Display:** Upon launching the app, the interface should present a list of genres arranged in a Lazy Vertical Grid, similar to the illustration provided in the figure 2 below.
   * **Genre Deletion:** Users should have the capability to delete a specific genre and, in the process, remove all Movies linked to that genre.

Figure Genre Screen



* + **Genre Selection:** Users should be able to click on a genre from the displayed list. This action will lead them to the Movie List Screen shown in figure 3, where they can explore a collection of Movies associated with the selected genre.

Figure 3 Movie List Screen

* + **Edit Button Functionality:** On the Movie List Screen, an "Edit" button should be present. Clicking on this button will direct the user to the Movie editor screen, allowing them to modify Movie details.
  + **Add Movie:** Clicking on an individual Movie card within the Movie list screen should lead to the Movie details screen shown in figure 4, providing users with comprehensive information about the selected Movie.

Figure Edit Screen / Add Movie Screen



* + **Movie Details Display:** Clicking on an individual movie card within the movie list screen should lead to the movie details screen shown in figure 5, providing users with comprehensive information about the selected movie.

Figure Details Screen