**CMPS 312 – Mobile Application Development Final Makeup Practical Exam - 13 December 2023 (duration 2h)**

**Objective:** Showcase your mobile development skills by crafting an Android application named “ScreenScore” 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 **ScreenScore** with all the necessary dependencies.
2. Implement a **ScreenScore** app that allows users to explore Movies and their ratings. And 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 **Ratings** 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.

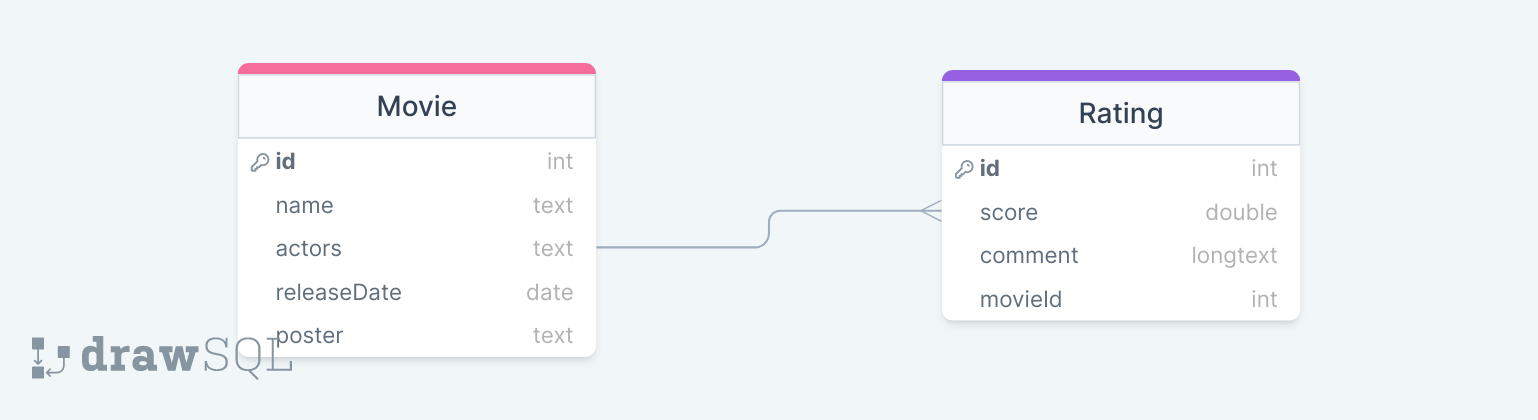


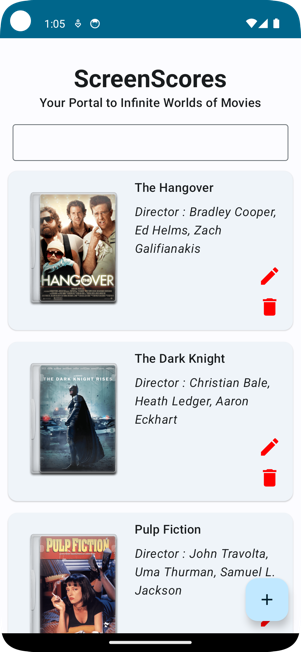
Figure 1 ScreenScore ER Diagram

1. **ScreenScoreDao Interface and ScreenScoreDatabase [20]** Create the database class and the ScreenScoreDao Interface with necessary functions and annotations to allow users to:

* Get all movies
* Get all movies that match specific search query either in the name or actors
* Add a new movie
* Delete a movie
* Update a movie
* Get a movie
* Get all ratings for specific movie

1. **ScreenScoreRepository [10]** Implement the ScreenScoreRepository with all necessary methods listed above. Also, inside the repository, initialize the database if it is empty using the provided JSON files.
2. **ScreenScoreViewModel [20]** Create a ScreenScoreViewModel 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 [40]** Implement the UI and navigation as described below:
   * **Initial Movie Display:** Upon launching the app, the interface should present a list of movies arranged in a Lazy Vertical Grid, similar to the illustration provided in the figure 2 below.
   * **Movie Deletion:** Users should have the capability to delete a specific movie and, in the process, remove all Ratings linked to that Movie.

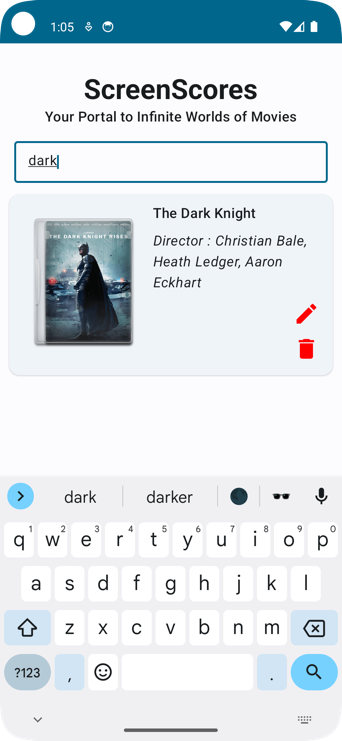


Figure 2 Movie List Screen

* **Searching:** Users should be able to search any movie using either the name of the movie or the name of the actors in the movie as shown in figure 3.

Figure 3 Movie Search

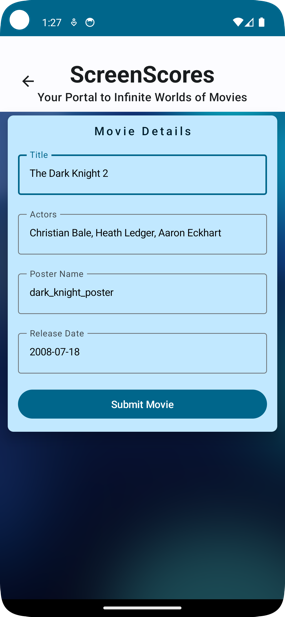
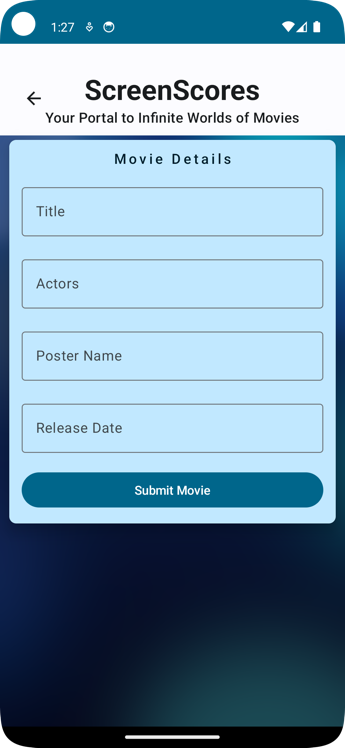
* + **Edit Button Functionality:** On the Movie List Screen, an "Edit" button should be present. Clicking on this button will direct the user to the Movie editor screen shown in figure 4, allowing them to modify Movie details.
  + **Add Movie:** Clicking on an the add button should allow the user to add a new Movie as shown in Figure 5. **[Hint you can use the same screen for add and edit]**

Figure 4 Edit Screen / Add Movie Screen



A screen shot of a movie

Description automatically generated

Figure 5 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 6, providing users with comprehensive information about the selected movie and all its user rating infos.

Figure 6 Details Screen