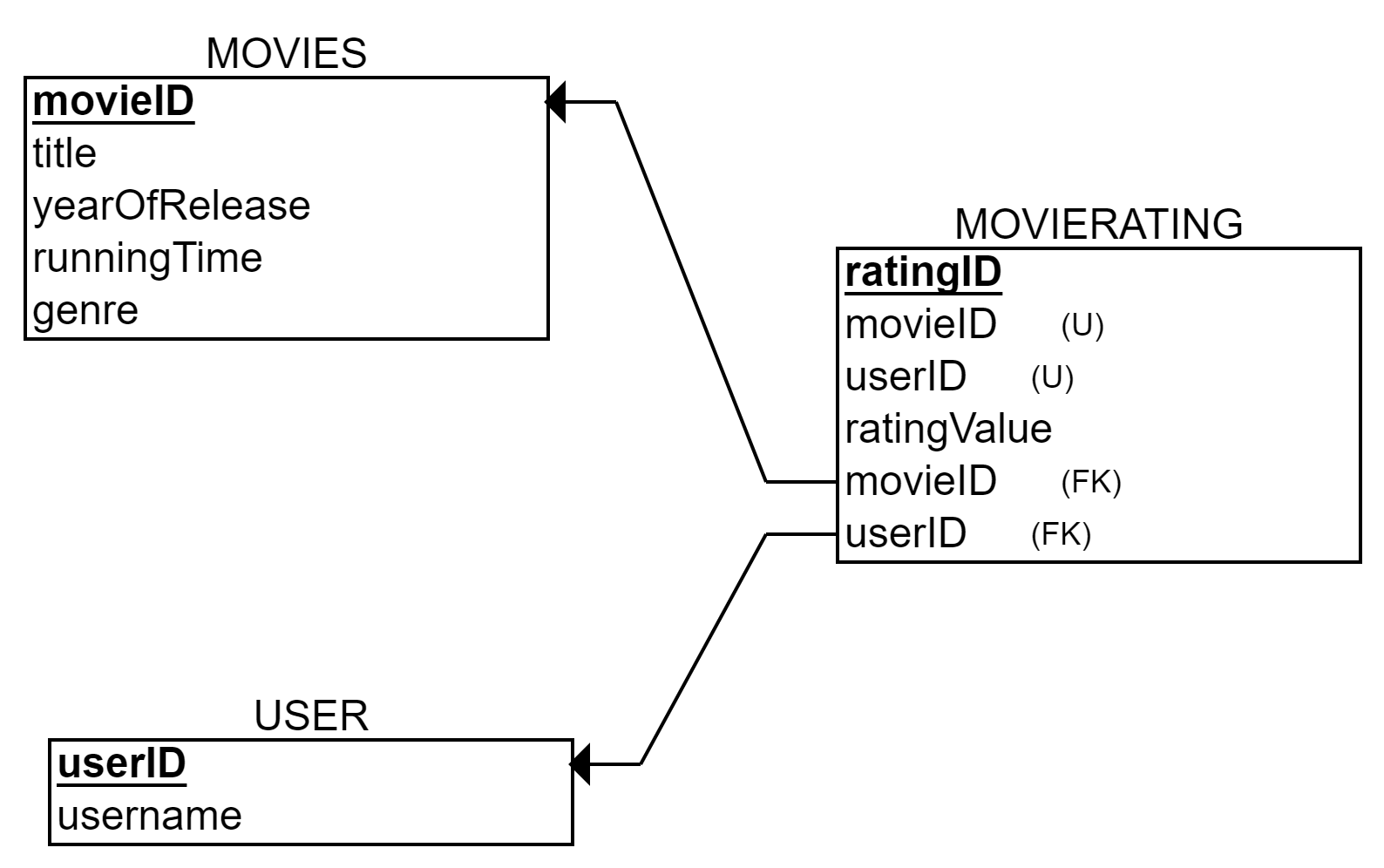
API Specification Solution:

**Database Structure**

To solve this Movie api challenge, I have a movie database with three tables

1. Movies (MovieId, Title, yearOfRelease, RunningTime, Genre)
2. Users (UserId, UserName)
3. Ratings (RatingId, UserId, MovieId, RatingValue)



Improvement: I would have prefer to have Genre as Code table

**C#**

I have created a simple web api project, created models and controllers (entity framework action). It is in memory databaseand I used Postman to test the solution.

1. To add movies: https://localhost:<portnumer>/api/movies/postMovies

Json example for movie: {"movieId":101,"title":"Gone with the Wind”, "YearOfRelease":1939,"runningTime":240,"genre":"Drama"},

1. To add users: https://localhost:<portnumer>/api/users

Json example for user: {"userId":208,"userName":"Ashley White"}

1. To add Ratings: https://localhost:<portnumer>/api/ratings/SaveRating

{"ratingId":14,"userId":208,"movieId":104,"ratingValue":3}

**This is my solution to the API D question**

**API A solution link: api/Ratings/SearchMovies**

This is post method and take SearchMovie model as input. I have used a search model to pass in the search input for this solution so that if we want to search a movie with more parameters we can easily do it. Example of search Model: { "searchString": "E.T", "genreStrign": "Drama", "year": 1930 }

**API B solution link: api/Ratings/ Top5Movies (GET)**

This returns details of top 5 movies based on average user ratings

**API C solution link: api/Ratings/Top%moviesByUser (GET)**

To get this solution I have use userId as search parameter. (It can be done username and then getting userId via username)