The application of the Internet movie database is a forum for movie enthusiasts, containing a database of movies, actors, directors, etc. Users can search for movies, actors, and directors. They can also rate movies and view ratings of movies. This application bears similarity to the IMDB database (http://www.imdb.com/), but not limited to the services that IMDB provides.

The Internet movie database that you design and develop should meet the following requirements: (i) minimum functionalities, (ii) good performance for common user operations, and (iii) a significant extension the project, e.g., an interesting, useful service or a unique aspect of your database system. Grading of the project combines all three requirements using the scheme shown below.

<table>
<thead>
<tr>
<th>Grading scheme:</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%</td>
<td>Correct implementation of minimum functionalities listed in Section 1</td>
</tr>
<tr>
<td>10%</td>
<td>Performance of required features</td>
</tr>
<tr>
<td>30%</td>
<td>A major extension of the project</td>
</tr>
</tbody>
</table>

The three requirements are explained in detail in Sections 1, 2, 3, respectively. The data set that we provide for building the movie database is described in Section 4.

1 Minimum Functionality Requirements
Your Internet movie database (referred to as your application in the rest of the document) must support the following features.

1.1 Searches
Your application should support three types of searches. These searches can be performed by any user without having to login (i.e., without being registered on the site).

A. Simple searches based on properties of movies. For this category, your application should support the following queries:
- Find all movies that have ratings in the input rating range.
- Find all movies that were released in the input year range.
- Find movies with title given as input. Users should be able to search for both exact matches and substring matches (using the LIKE construct in SQL).

B. Complex searches involving multiple entities. The following queries involve the correlation of two or more entities.
- Find movies with a particular actor, actress, or director given as input. Users should be able to search for both exact matches and substring matches (using the LIKE construct).
- Find movies that have all the input actors (and/or actresses) in it.
- Find all actors and/or actresses who have worked with a given director.
C. Top K searches
- Show top k (k ≥ 1) movies as sorted by rating. In case of a tie, show any top k movies.

1.2 User Login and Rating of movies
As mentioned before, users can rate a movie in the forum. The database stores average rating of a movie based on up-to-date rating information.

User login
For users to rate movies, they need to register to the site and create logins for themselves. After they login to the site with the correct user name and password, they can rate movies.

Rating of a movie
A user can rate a movie in the range of 1-10. Every time a new rating is generated, the average rating for the movie is updated in the database. (Hint: you may want to store the average rating of a movie and the count of users who have rated this movie so far for such updates). It is up to you, the owner of the database, to decide if you allow a user to update/delete his rating of a movie.

1.3 SQL Interface for the Administrator
The database administrator is a special user with the user name “admin”. He has unrestricted access to all the data in the database.

Searches by the Administrator
Besides the search interface for regular users, you web site should also support the direct input of SQL queries for use by the administrator.

Updates by the Administrator
Information about the movies, actors, actresses, directors, and mpaa ratings in the movie database can only be updated with an administrator login. (Hint: you can use the same SQL interface for both searches and updates by the administrator.)

1.4 Constraints
While many of the constraints to which you need to pay attention are described above, they are summarized and re-iterated below.

Loading the database
- At the beginning, the database is loaded with initial values from the dataset supplied with the project (explained more in Section 4). In the loading process, you may also want to create values of other attributes that are in the schema of your database but not in the initial data sets (e.g., the movie id). If an attribute is present in more than one table in your schema (e.g., the movie id may appear in multiple tables), please make sure that values of this attribute are consistent across all tables.
Updating the database

- After the initial loading, only the administrator is allowed to add tuples representing new movies, new actors, new actresses, or new directors to the relevant tables, and to update or delete existing tuples about movies, actors, actresses, and directors.
- The database can also be updated with the information of user login or rating when a user registers to the site or rates a movie.

Need of user login

- A user can search the database without having to login.
- But login is needed to be able to rate movies.

Rating movies

- To ensure fair rating, each user is allowed to rate a movie only once.
- In addition, a user under the age of 18 is only allowed to rate certain movies based on the mppa-rating of a movie and the user's age. In this project, please use the following table as the guideline for deciding who can rate what movies.

<table>
<thead>
<tr>
<th>Rating</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>Anyone can rate the movie</td>
</tr>
<tr>
<td>PG</td>
<td>Children under 10 years of age cannot rate the movie</td>
</tr>
<tr>
<td>PG-13</td>
<td>People under 13 years of age cannot rate the movie</td>
</tr>
<tr>
<td>R</td>
<td>People under 17 years of age cannot rate the movie</td>
</tr>
<tr>
<td>NC-17</td>
<td>People under 18 years of age cannot rate the movie</td>
</tr>
</tbody>
</table>

2 Performance Requirements

Your application is expected to run with reasonable performance. While there are no specific requirements in terms of response time, most operations on the web site, including user registration, user login, searches, and updates, should be completed with no obvious delay.

If you observe significant delay in common operations, it is a good indicator that you should review your design and implementation to identify the causes of slow performance. As general guidelines, you may want to consider the following things:

- Can you refine the database schema to improve performance? You may want to apply the theory of normalization that we learned in class.
- Can you create additional indexes to improve performance? Refer to the lecture on physical design and tuning for a detailed discussion of this topic. The SQL command for creating an index is `CREATE INDEX index_name ON table_name ( column_name(s) )`.
- Did you implement the functionalities of your application in the right place of your program? For example, a join between two tables should be performed in the database backend. Performing the join in your PHP code is a mistake—even though you may still provide correct answers, the performance of your application will eventually suffer.
3 Project Enhancement

Your application should have a significant extension beyond the basic functionalities described above. The following list outlines several ideas regarding the project enhancement.

− You can create a user community where users write and share reviews. In addition, a user may maintain a list of his favorite movies and a to_watch list. Other users can view these lists. You can further allow users to find the common set of favorite movies among them.
− Apart from the basic function of rating movies, users may want to rate actors, actresses and directors, and also search for actors, actresses, and directors in a rating range.
− You may want to augment the security of your application, e.g., developing mechanisms to protect the database from a variety of attacks.
− You may also want to integrate your application with other data sources online. Examples include (1) integrating your application with movie reviews from “imdb” or “new york times”, (2) fetching posters and trailers of movies from their official websites, (3) integrating your application with showtime tables online, or (4) integrating your application with online movie rental services.
− Your application can also have a notification service, e.g., using XML/RSS feeds to report to interested users the latest addition to the movie database or top 10 movies this week.

You are also encouraged to propose ideas of your own. Just be creative and have fun!

4 Data Sets

You will be provided with several data sets in the form of text files. It is an important initial step of your project to parse the data in those files and load it into the database that you will create based on your designed schema. After the data is loaded into the database, your movie database is ready to accept requests for search, update, and deletion.

The format of each data set is explained in order below.

1. movies.txt
2. directors.txt
3. actors.txt
4. actresses.txt
5. mpaa-rating.txt
6. rating.txt

movies.txt
Each line contains two fields, separated by ‘|’.
− The part before ‘|’ is the movie name and is consistent across all files.
− The part after ‘|’ is the year.

directors.txt
The information about a director is presented by the director’s name in the first line and each movie directed by him/her in a separate, subsequent line. Any line that does not begin with ‘|’ is
a director’s name. All subsequent lines that begin with ‘|’ are movies directed by this director, until you hit another line that does not begin with ‘|’.

**actors.txt**  
This file is in the exact same format as directors.txt.

**actresses.txt**  
This file is in the exact same format as directors.txt.

**mpaa-rating.txt**  
Each line contains two fields, separated by ‘|’.  
- The first field is the movie name.  
- The second field is the mpaa-rating for the movie. You will need this to make sure users can rate movies depending on their age.

**rating.txt**  
Each line contains three fields, separated by ‘|’.  
- The first field is the count of the number of people who have rated the movie. You will need this to calculate the rating of a movie as more people rate it.  
- The second field is the actual rating, averaged across all the ratings from all the people who rated the movie.  
- The third field is the movie name.

A sample data set is available on the course web site under the “projects” page. The text files for this project, which are much larger than the sample files, will be uploaded to the “/usr/ftp/pub/cs445/projects/movie/” directory on the edlab machines.