|  |  |
| --- | --- |
|  | **DOKUZ EYLÜL UNIVERSITY**  **ENGINEERING FACULTY**  **DEPT. OF COMPUTER ENGINEERING** |

<RATE A MOVIE >

# CME 3201 Database Management Systems

# Term Project Report

Phase 2

2019-2020 FALL

2016510086 Mustafa ÖZSARAÇ

2017510033 Ayşe Özge ERİŞ

## Introduction

This report is prepared to give brief information about the Rate a Movie Project. The document contains the problem domain, solution and purpose of the project.

**Problem Domain**

People love movies and actors/actresses however there is one thing that people like more: discussing about them. The problem is that there is not an environment that allows people to share their opinions and ratings.

That’s why our purpose in order to solve this problem is to create a web site for movie lovers to rate movies, share their comments, see movie information such as movie’s director, cast, release date or creating watch lists for users etc.

Anyone who is willing to bring people who are interested in movies together could be the stakeholder of this project. They can invest in this project to support us to do better.

**Intended Audience**

-Anyone who loves movies.

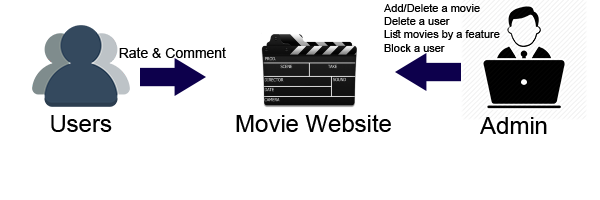
-Anyone who is curious about movies’ ratings.

-Anyone who is interested in discussing about movies.

Basically, this website will be for everyone who is keen on to movies.

## Purpose and Solution

The purpose of this project is to be solution to all those problems mentioned above. It will be a meeting point for all movie enthusiasts. Creating a practical and fun environment for people who are looking to movies.

**

The goal of this project is to create a website like “IMDB”. Users will be able to find all the information about movies or see watch lists of other users and their information.

## Overview

**2.1 Operations & Instructions**

**Register**

New users can click “Register” button on login screen to create their account with their selected username, name, surname, age and gender. All these will improve user’s experience in website.

**Login/Logout**

Users can login and logout to the system.

**Admin Panel**

Administrators of the website can use this panel to control, change, delete and access most of the information and other things in their website.

**View Comment**

Allows users to view comments of other users for movies.

**View Movie**

Allows users to view the movies.

**Rate Movie**

Allows users to rate the movies from a scale from 1 to 10.

**Add Comment**

Allows users to share their opinions about the movies.

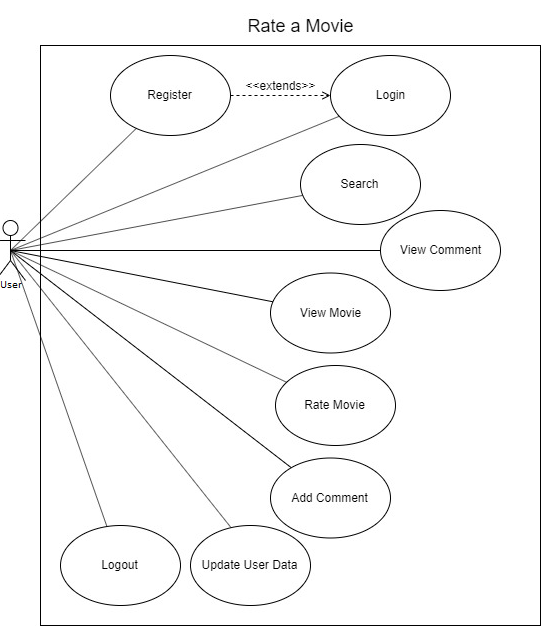
**Search**

Allows users to search for other users to see their profiles.

**Update User Data**

Users can edit their profiles by clicking “Edit Profile” button in their profile page.

**2.2 Use Case Diagram**



**2.3 Stakeholders**

Anyone who wants to bring people who are into movies together could be the stakeholder of this project. They can invest in this project to support us to do better. The roles of the stakeholders are as follows:

* Voting and Decision Making
* Providing Expertise
* Managing Industrial Crises
* Corporate Social Responsibility

**2.4 Entity List**

**User Entity**

1. User ID(auto-increment - int)
2. Nickname(varchar(16))
3. Password(varchar(16))
4. E-Mail(varchar(16))
5. Phone Number(Optional) (int(11))
6. Birthdate(date)
7. About(string)
8. Is Admin(bool)

**Movie Entity**

1. Movie ID(auto-increment - int)

2. Movie Title(varchar(256))

3. Genre(array(varchar(50)))

4. Release Date(date)

5. Rating Average(float)

6. Rating Count(int)

7. Crew(crew-int)

8. Age Limit(int)

**Comment Entity**

1. Comment ID(auto-increment - int)
2. Movie ID(int)
3. User ID(int)
4. Title(varchar(50))
5. Comment(varchar(50))
6. Entry Date(date)

**Genres Entity**

1. Genre ID(auto-increment - int)
2. Name(varchar(50))

**Actor Entity**

1. Actor ID(auto-increment - int)
2. Name (varchar(50))
3. Surname(varchar(50))

**Crew Entity**

1. Crew ID(auto-increment - int)
2. Name (varchar(50))
3. Surname(varchar(50))
4. Movie ID(int)

## Assumptions/Constraints/Risks

### Assumptions

With the help of the stakeholdersproviding expertise to this process, completing the project without any obstacles and on time is expected.

### Constraints

**Username**

While registering, users cannot have empty username. They must select a username that is not already in use.

**User Password**

All users must have a password for their account.

**User Age**

Users cannot select negative or over 100 as their age.

**User Gender**

Users can either pick male or female as their sex, or “–“ for empty.

**Rate Scale**

Users must rate the movies from 1 to 10 and only as an integer number.

**Obligations of the System**

Users cannot comment or rate movies unless they are logged in.

### Risks

The expectable risks are that the possibility of data loss from the database, undetectable bug in the application, unexpected breakdown of the serves and potential delay of the project’s handing over to the customers. We expect to overcome these possible problems with the help of our stakeholders’ expertise.

## Software Architecture

## Detailed System Design

### Entity-Relationship Diagram

### Relational Algebra Expressions

1. Write a query in SQL to list all the actors who have not acted in any movie between 1990 and 2000

SELECT \*

FROM Actor a, Crew b, Movie c

WHERE a.actor\_id=b.actor\_id

AND b.movie\_id=c.id

AND c.release\_date NOT BETWEEN 1990 and 2000;

πactors(σ1990<release\_date<2000(Actor) ⨝ Movie))

2)Write a query in SQL to list all the movies that has actor named “Ayşe Eriş”

SELECT title

FROM Actor a JOIN Movie b

WHERE a.actor\_name=”Ayşe Eriş”

πtitle(σname = “Ayşe Eriş”(Actor) ⨝ Movie))

1. Write a query in SQL to list all the movies with year and genres.

SELECT title, release\_date, name

FROM Movie JOIN Genre

Πtitle,release\_date,name (Genre ⨝ Movie)

4)Write a query in SQL to list all the movies’ rating that are higher than 7.

SELECT title

FROM Movie a

WHERE a.rating\_avg > 7

πtitle(σrating\_avg > 7 (Movie))

5)Write a query in SQL to list all the movie genres which are +18.

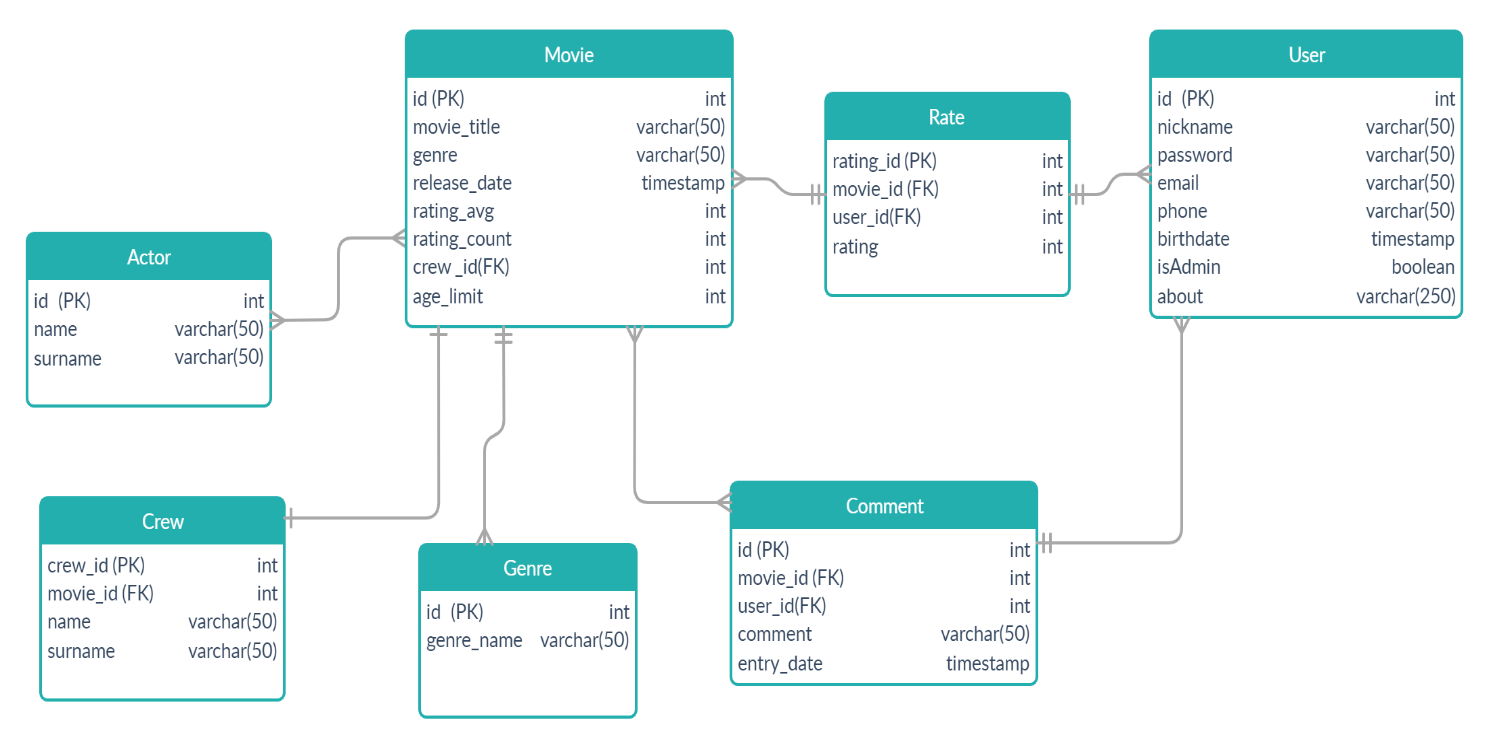
SELECT genres

FROM Movie a

WHERE a.age\_limit >= 18

πtitle(σage\_limit> 18 (Movie))

### Class Diagram



### CRUD Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Relation/Operations | Create | Read | Update | Delete |
| Contains |  | X | X | X |
| Rate | X | X | X | X |
| Share | X | X | X | X |
| Has |  | X |  |  |
| Starring | X | X | X | X |
| Includes | X | X | X | X |

### Database Schema

