İhsan Doğramacı Bilkent University

Department of Computer Science

CS342 – DATABASE SYSTEMS

Online Movie Rental System

Project Proposal

Efe Ertürk 21902620

Oğuz Ata Çal 21903088

Yiğit Ekin 21901784

Arda Eren 21902505

1. Introduction

The topic of this project is designing an Online Movie Rental System. Our aim in this proposal is to clearly describe the project topic that we are designing and to discuss why and how we use a database for this system.

Then, we will specify functional and non-functional requirements for this system that detail the interaction between the different types of users and the system functionalities along with the system constraints. In the end of this section, we will list the programming language and frameworks we plan to use under pseudo-requirements heading.

Finally, we will display the conceptual design of our database using an entity-relationship model that will include strong and weak entities, relationships between entities, cardinality constraints, keys and attributes and more.

2. Project Description

The main aim of this Online Movie Rental system is to provide customers a platform to easily rent movies online. Customers will be able to create a request for a movie to be added to the system if their desired movie is not available. The customer can also write reviews, rate films and like or dislike other reviews. While writing a review, one can warn users if the review includes a spoiler to the film or not. The reviews can be displayed according to their dates, from newest to oldest, or according to net likes, or one can choose to view the reviews that do not have any spoiler in it and so on. There is also a friends system that allows customers to add other customers as friends and recommend films to them. The recommendation can include a message for your friends too.

Customers will be able to search movies by title, genre, production year, director and actors. They will also be able to see the movies they are currently renting, their rental history showing the movies they rented previously, the movies that their friends recommended to them, the most rented movies, the best rated movies (which has the best average rating) and finally the list of movies that they put on their favorites. The system will also keep track of the twenty movies that are newly registered to the system.

Employees will be able to satisfy the requests of customer requests and register new movies to the system. They will also have the authority to delete customer accounts.

For the new customers, there will be a promotion code assigned to them, which enables them to rent films at a discounted price.

3. Requirements

a. Functional Requirements

i. Customer

- Customers can search movies based on title, genre, production year, director and actors
- Customers can see the movies they are currently renting, their rental history showing the movies they rented previously, the movies that their friends recommended to them, the most rented movies, the best rated movies and finally the list of movies that they put on their favorites
- Customers can rate movies
- Customers can request a movie to be registered to the system
- Customers can add another customers as friends
- Customers can recommend movies to their friends with messages
- Customers can write reviews and rate movies, indicate whether the review will include spoilers or not.
- Customers can view reviews that are sorted by their date, net like count. They can choose to not see the reviews that have spoilers in them
- Customers can like or dislike the reviews made by other customers
- Customers can favorite movies

ii. Employee

- Employees can register new films based on customer requests
- Employees can delete customer accounts

b. Non-Functional Requirements

i. Security

- Passwords and card information will be kept hashed in the database using salt encryption (this way, passwords will not be seen even by developers)
- Employees and customers will have different access rights to functionalities of the system which prevents employees from adding a new rentable film to the system or employees to modify the data of the customers.
- Non-users or unauthorized users should not be directed to sensitive pages by copying and pasting the url to that page (for

example, a customer shouldn't be able to access employee pages just by copying and pasting the url)
- At database crushes, there should be no data loss
- System needs to be secure against web attacks such as SQL injections and XSS attacks
- When an account is being deleted, all personal data related to that account should be deleted
- After logoff, even from the same ip address, when a certain url is tried to be accessed, until a next authentication, these url should not be accessible

ii. Performance

- Optimized queries will be used to retrieve data from the database
- Logins should take less than 5 seconds when the correct data is provided
- Directing to other endpoints within the website should take less than 5 seconds
- Logout operation should take less than 5 seconds

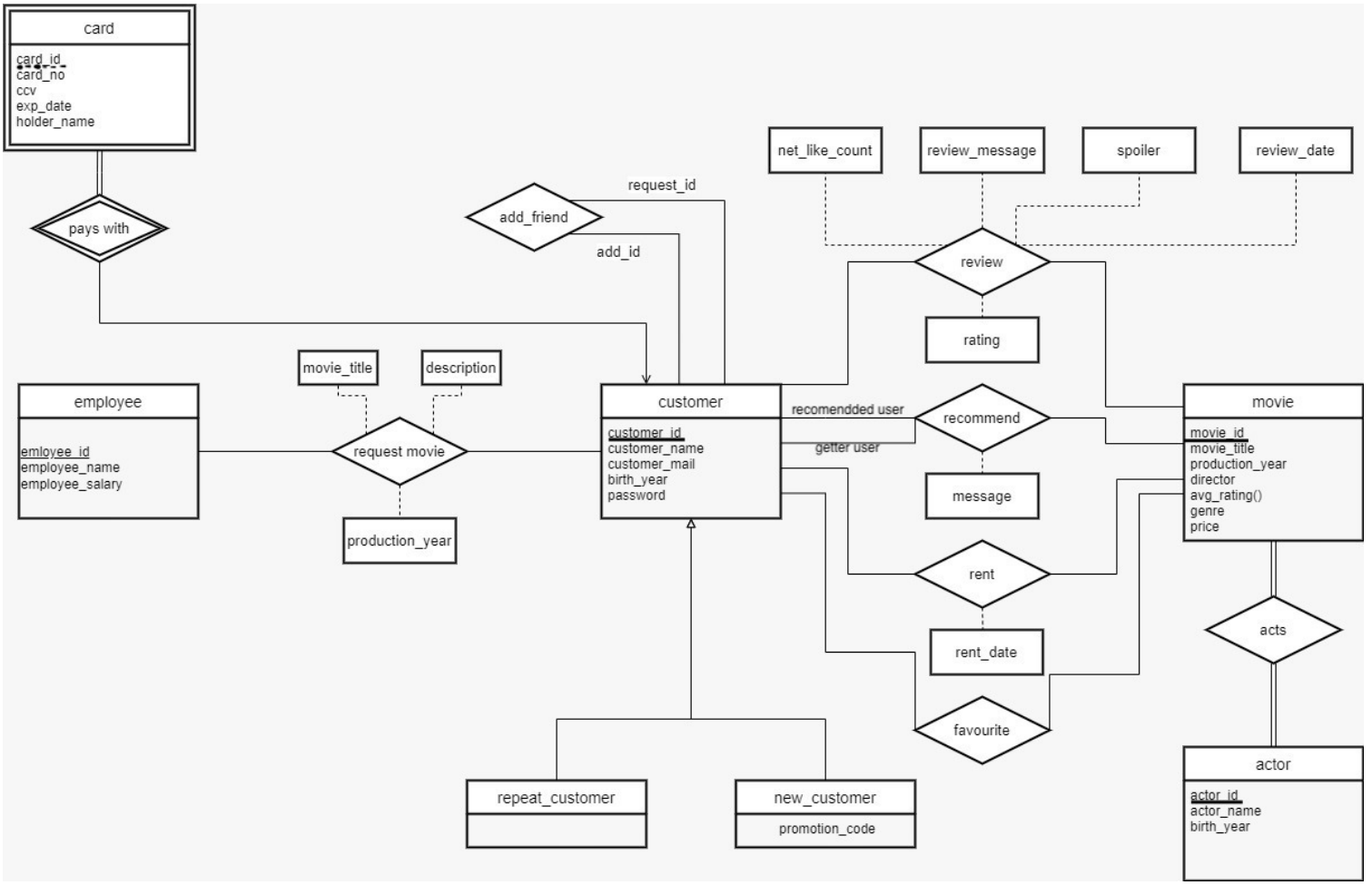
iii. Usability

- User friendly ui components will be used such as descriptionary labels, large buttons with hover effect, navbars, containers and in addition, responsive web design will be used.
- Use different font sizes to draw attention to headings (headings are larger in font size)
- Make the minimum font size (for bodies) 4 rem
- Use contrast color for background and texts of components (for example, beige color for background and dark gray for the texts)
- Give vivid colors to the component that need extra attention ("Delete customer" button can be red)
- Use mobile first approach (users are likely to use the mobile version of this system), responsive design should be implemented

c. Pseudo Requirements

- MySQL will be used as the query language
- JavaScript and Typescript will be used interchangeably for the frontend with the help React framework, Bootstrap4 will be used as a CSS framework and Java will be used with SpringBoot for the backend

4. Entity-Relationship Model



Link to the project website: https://cs353-group20-project-page.netlify.app/index.html