Bilkent University

Department of Computer Engineering

CS 353 Course Project

Group 9

Proposal Report

- Yavuz Faruk Bakman
- Arda Göktoğan
- Fatih Sevban Uyanık
- Duygu Nur Yaldız

Supervisor: Arif Usta

Table of Contents

1. Introduction

Online media platforms became an important part of human life. Platforms such as YouTube and Netflix are watched by millions of people each minute [1], [2]. Therefore, the demand of people for such media services can not be ignored. In this project, an online media platform will be proposed, designed and also implemented.

This report contains the detailed project description, which includes the purpose of the project and the way that the database will be utilized. Description of the project also contains the entities that exist both in the database. Then the functional requirements of the system will be discussed in terms of the set of operations that the system will serve for users and the set of operations that the system should handle. Afterwards, non functional requirements of the system will be explained, which will be crucial for favorable user experience. In the Limitations section, the regulations that users should follow will be clarified as well as the constraints of the system will be proposed. Finally, Entity/Relation diagram will be proposed as a model of conceptual design of the database.

2. Project Description

Lava is a platform that users can watch movies and series. It enables the users to make friends, to send film/series suggestions to their friends, and to see the activities of friends. With these features, Lava aims to be an interactive environment for users. Also, the users of this application will be able to comment on films or episodes of series. They can like or dislike movies, episodes and comments. In order to make it easy to find and select a movie or series for users, the application will categorize movies and series as well as providing a search tab, which users can search movies/series by directly their names or the name of a desired person acted or directed. Also, similar movies and series to what a user has watched and liked before will be suggested to the user. Users will be able to favor

genres, which the application will consider while suggesting movies/series. Lastly, Lava will provide users with the ability to create their own lists of movies and/or series to watch them later and to organize them according to their wish. Overall, the purpose of this project is to serve users an interactive, funny and customizable platform that they can watch movies and series.

2.1. Why and how a database is going to be used?

This project aims to function for large numbers of users with piles of media files as well as their relations. Disturbing the information regarding these relations into several files may cause problems. Since there are multiple restrictions about the data, In the long term, it is possible to occur some inconsistencies or redundancies among these files. For example, If any movie is removed from the system, the list of people that liked the removed movie should not be kept in the system anymore. This consistency is hard to provide by filesystems. Moreover, this program is expected to serve for many people at the same time, which means that data in the system will be accessed and manipulated from different users at the same time. Allowing multiple users to interact with file systems at the same may cause data concurrency problems. Third reason is restricting the permission of users with different roles. Having a filesystem will be harder to manage permissions of the users in the software level and may cause security problems. Therefore representing such complex data information in the Database system provides us to store and retrieve data more efficiently.

Database is going to be used in several ways. Firstly It aims to represent, store and make accessible complex information in the system regarding the users, media files, and their relations. Database queries are powerful because they allow people to search any movie/series based on features. They also will allow the system to make meaningful recommendations to the users, based on their interests and common attributes of the movies/series they had watched before.

2.2. Database Items

- **User:** The main part of the system is user. By the user entity, we store password and email information which is unique. Users can access the system with this information. Users' birth_date, language that users select and name information is stored in the database to provide better presentation to the users. Also, there is a role attribute. This attribute basically determines whether the user is a watcher or admin. Based on this attribute, user entities have different access right in the database system. Users can send and receive friendship requests. By the request, they can be friends and see their activities.
- **Movie:** Movies are one of the fundamental parts of the system. Users can watch movies, comment, like or dislike them, suggest other people, and add their channels. Every movie includes the information including the original language, the age limit which basically provides displaying proper contents to the user, the description of the movie, the duration, the country movie produced and have some tags that describe the movie.
- **Series:** Series is like movies. However, instead of having one media, it has episodes. It has the most attributes that movies have. For instance, countary, original languages etc.. But other attributes such as duration are in the weak entity of Episode. Series also can be suggested and might exist in a channel.
- **Episode (Weak Entity of Series):** It is a weak entity because it does not exist without its owner (series) and the partial key is season number and episode number together. Every episode has a description which basically summarizes what will happen in the episode. Also, it has the attributes that movies have such as duration, release time etc..
- **Film Member:** Film members are either director or actor/actress. They can exist in series' episodes and/or movies. By holding this information in the database, we will be able to provide the service that users can search movies/series by the

actor/actress or director name. Also, for every member there is a general information part. By this attribute, when a user searches for an actor they can see the kind of biography of the actor.

- **Genre:** There are different genres which are uniquely identified by their name. We want to keep genre as not an attribute of movies/series entity. Because users can favor a genre and see their favourite genres on a different page.
- **Movie Comment (Weak Entity of Movie and User):** It is a weak entity of movie and user entities. It has a partial key which is id. The comment has time information. Comments can be liked or disliked also one comment can be commented with another comment.
- **Episode Comment (Weak Entity of Episode and User):** It has almost the same properties as movie comment. But it is a weak entity of the episode not the movie.
- **Channel (Weak entity to User):** Users can create their own channels. These channels may include movies or/and series. They have a name which the user assigns and also the creation time information is holded.
- **Default List:** These are the movie lists created by the system. These lists may include movies/series. For instance, these lists can be named as the editor's choice, the movies in 2. World War.
- **Movie Subtitle (Weak entity to Movie):** There are different subtitles for one movie. The subtitle entity has attributes which are language and the subtitle's link which directs the subtitle file.
- **Episode Subtitle (Weak entity to Episode):** It is the samex" with movie subtitle but it is the weak entity of Episode instead of Movie.
- **Notification:** Users can get notification from the system. These notifications have text messages generated automatically. And it has the information. The message may be about a friends' activity or a new released movie.
- **Suggestion:** Suggestions include movies/series and they can be sent by users to each other. Users can also leave a message with the suggestion they send.

3. Requirements

3.1. Functional Requirements

Functional requirements of the system can be analysed in terms of the functions it serves for the users and the functions that serve internally.

User Side

A user in the system is defined by its unique e-mail, name, surname, password, birth date and selected language. A user can enter the system by unique e-mail and its associated password. Users also have roles such that admins and default users. Admin users are discussed in the next title.

Users can watch movies or series in the system. They can see the related information about any movie, series, or episodes as well as the actors played in that movie. When users enter the system, they can continue from watching where they left last time. They will also be able to search movies or series by their attributes.

Users are allowed to like/dislike movies and episodes, and make comments on them. They can also like/dislike the comments or make comments on other comments. These comments and likes/dislikes will be visible to all users. Any user can create lists of movies and series (called channels), based on their wishes to be able to watch them later or to group films in an organized way for any purpose. Since there are genres associated with series and movies, users can favor these genres. These favors will play a role on the notifications of the users, sent by the system.

Users can send friend requests to each other, and if accepted, users will be friends. Friends are allowed to make movie or series suggestions to each other. Users can also see activities of their friends on their wall (which will be the default

screen when users enter the system) These activities may consist of the likes/dislikes and comments as well as the movies or series they watch.

Admin Side

Admins in the system have all the permissions that any user has, and some extended permissions as expected. Admin in the system will be able to add new series, episodes and movies to the system. They can add, change or delete any information about the movies or series, as well as the subtitles associated with movies or episodes.

Admins can send notifications to all users, groups of users or individual users. They can create, change or delete a list of movies and series. Since the admins are also responsible for the surveillance of the system, they will be able to delete comments of users in case of inappropriate discourse.

System Side

System serves as a backbone to accomplish the user and admin functions discussed in the previous sections. Besides, the system plays an important role for recommending movies/series to users based on their interests. System can send notifications to users for film recommendations. These recommendations will be determined by the common attributes (genre, language, actors etc.) between the recommended movies or series and the movies and series that the user watched before.

3.2. Non-Functional Requirements

3.2.1. Performance

Since this application aims to serve movies and series, the performance of it is an important point. The users should be able to watch films and series without interruptions. The system should be able to serve 1000 movies/episodes simultaneously. Also, the system should respond to the requests at most 2 seconds.

3.2.2. User Friendliness

The user interface of the application should be easy to use and learn so that a new-coming user can learn how to use the application in 10-15 minutes.

3.2.3. Authentication and Security

Users need to sign up the application with their emails, and sign in to the system with these emails and the passwords that they assigned. The information and activities of users should not be shared with third party services without their permission. Also, only the admins should be able to add and remove movies/series from the system.

3.2.4. Modifiability

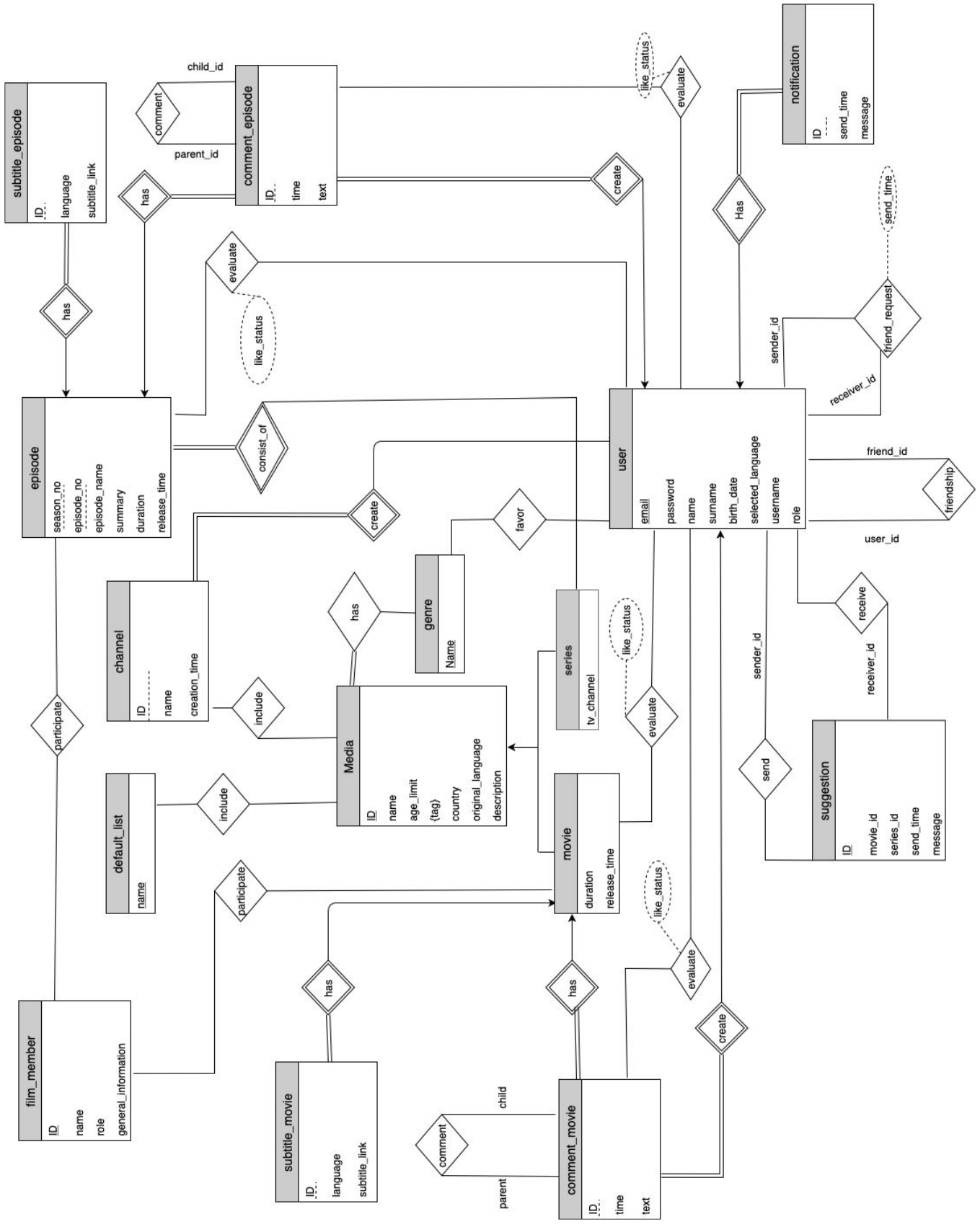
New movies and series are constantly being shot. Therefore, the application should be modifiable to deal with such new-coming data. Admin users can be able to add and remove the movies and series to the system.

4. Limitations

- Only admin users should be able to add and remove movies/series from the system.
- Users should not be able to send suggestions to the users that are not their friends.

- Users should not be able to send friendship requests to the users that are already their friends.
- Users should not be able to add the same movie/series to the same channel multiple times.
- Admin users are able to delete comments of other users but they are not allowed to change comments of other users or add new comments.
- Users should not be able to delete and modify other user’s comments.
- Each user can see and manipulate only its own channels.
- Admin can create default movie/series lists but other users cannot.

5. Entity/Relationship Diagram



6. Conclusion

To conclude, we have designed a movie/series system which is named Lava. This system is user friendly and provides different lots of functionalities such as sending suggestions to friends, creating your own movie/series channel etc.. To store the information of some of the system objects, a database system will be used. This system is designed in an elegant way. Therefore, it does not have artificial, unnecessary entities and it has necessary entities and relations between these entities. The ER diagram is provided to make the reader understand the design better. Also, the system has limitations. The users cannot change whatever they want. Both users and admins have to obey some limitations. This makes our system more reliable. The system will be provided as a website.

7. Website

Reports can be found at : https://ybakman.github.io/Lava

8. References

[1] “YouTube by the Numbers: Stats, Demographics & Fun Facts”. https://www.omnicoreagency.com/youtube-statistics/. [Accessed: Oct 13, .2020].

[2] “Netflix’s Cindy Holland Says Subscribers Watch an Average of Two Hours a Day”. https://variety.com/2019/tv/news/netflix-cindy-holland-subscribers-watch-average-two-hours-day-1203159868/. [Accessed: Oct 13, 2020].