

Final Project

Movie Rating & Comment Website

INFSCI 2560

December 13, 2017

Peng Jin (pej23)

Xinyi Chen (xic98)

Xinyi Li (xil192)

WeiQi Yu (wey46)

Table of Contents

Part I: Overview	1
Part II: Data Source	1
Part III: Authentication	1
Part IV: Movie Rating Survey	2
Part V: Homepage	3
Part VI: Account	3
Part VII: Search	3
Part VIII: Movie Detail.....	3
Function:	3
Part VIII: Recommendation.....	6

Part I: Overview

The goal of our project is to develop a movie community with two prominent features: interaction and intelligent. To be more specific, in this movie community user can rate movies and recommendations are going to be made by the personal rating history. There are four main part in our project which includes Gathering data source, Creating User authentication, account and using history page, Designing and Building home page and movie detail page, and last but not least, achieving the recommendation function.

This paper includes our initial idea,, thinking process, and implementation of some functions. The code of our project are included in the file.

Part II: Data Source

We first considered to use data from twitflicks directly, but there is a question that although there are many comments, the information of movies is not enough. Information such as cast and directors are not included. Fortunately, we found “The Movie DB” website which offers APIs of movie information. We wrote a crawler to download all the information on twitflicks and reassigned the rating to about 2000 users since our recommendation algorithm is based on rating similarity, but each user in original data has rated only one movie. After that, we used the mapping table of movie id and movie name from TMDb and downloaded the additional information of movies according to the id. Finally, we generated four csv files and imported them directly to the database after creating the schema.

Part III: Authentication

The authentication section consists of two pages (login.jsp, register.jsp) and one servlet (Authenticatoin.java). The authentication process mainly uses ajax to transfer data between front end and backend.

In login page, user is required to fill in the username and password, and only after both blanks are filled can “submit” button be clicked. Then the script will send POST ajax request to servlet with parameters “login”, “username” and “password”. The servlet receives the login signal and checks the database whether the username matches the password, and there is another process which is used to check whether the user has completed the “rating survey”. The mapping of result and status are as follows. The script in login page then parses the result and chooses to redirect to index page or survey page.

Result Tag Value	Status
0	Success
1	Fail
2	Success, but need to complete the survey

Table1. Mapping of Result and Status

The register page is very similar to the login page. The most significant difference is that the user is required to fill in more information such as email and name. We use regular expression to check whether the email is valid, and we also check whether two password inputs are the same. After user fills in all blanks and clicks the “submit” button, the following process are the same as the one in the login page.

Part IV: Movie Rating Survey

In the part of movie rating survey “recommendbase.jsp”, new users should rate 10 movies to create records which will be used in the recommendation algorithm.

In the “recommendbase.java” servlet, we randomly select 8 movies which are showing on the website from our movie database. Then the output in xml format will be received by the front-end through AJAX. After that, we use DOM to generate all nodes which are used for showing the movies, and these codes can be seen in “randommovie.js”.

In the “star.js” document, the first 4 functions are used for controlling the performance of rating stars. Once the user clicks on the stars, it will call “submitrating” function, and then connect

back-end servlet to insert data into the database by AJAX. After completing the ratings of 10 movies, the user will be directed to our home page and begin to browse our webpages. The personal movie recommendation pages will be unlocked.

Part V: Homepage

Our homepage consists of three sections, which are welcome image, featured movies and customized recommendations. When the page is loaded, the script will check the session for “uid” attribute to judge whether the user has logged in. If the user has not logged in, the recommendation section will show the default movies. Otherwise, the script will first check whether the user has completed the survey by “already” flag which should be set in Authentication servlet or rating(survey) servlet. If there is a flag, the script send ajax request to “homePageRec.java” servlet and obtain the information of movies recommended.

Part VI: Account

The account section consists of two parts, account setting and rating&comment history. The account setting will send ajax request to Authentication servlet to get the personal info and update the info. The rating&comment history section will send ajax request to History.java servlet and use the information to generate a table.

Part VII: Search

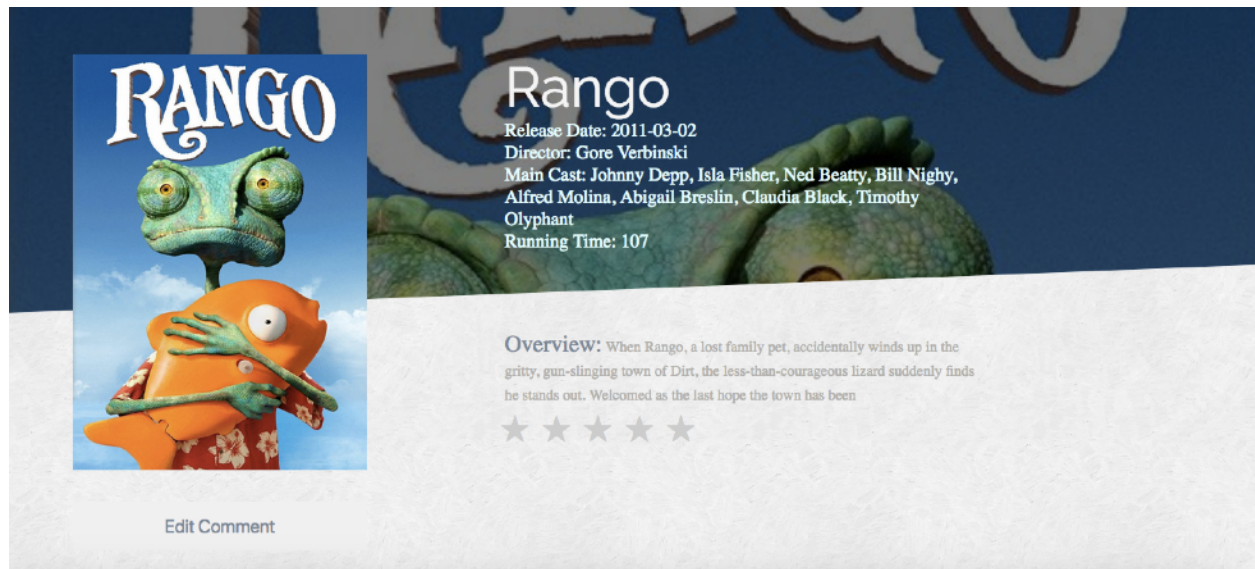
The search section has search.jsp page and Search.java servlet. Once the page is loaded, the script will get parameters from url such as “name” and “page” and send them to servlet. If “page” doesn’t exist, it will be set to 1. The servlet has implemented multiple search and fuzzy search by splitting words with whitespace and using regular expressions in SQL queries.

Part VIII: Movie Detail

Function:

This website shows the detail information about a specific movie the user interests and allow the user to leave comment and rate the movie. For the details, the the Name, Release Date, Director, Main Cast,

Running Time and Overview of the movie. It also presents all comments under this movie. The current user's previous comment will be presented as well if he has commented after the page load. All those data is obtained from the 'comments' and 'users' table from database by selvert and Ajax .Also, the user can edit his comment and resend it. If he never review this movie, he can review it and the comment will be placed on the top of the comment list.



Picture 1. Movie Details presented by current page



Picture 2. Comments under the selected movie

Once the user has commented the movie, the commented will be send to selvert to servlet by Ajax and update or insert into database on has the user commented or rated this movie before or not. Users can rate the movie by click the rating star. The rating score will be passed to selvert by Ajax and update or insert into database.

User can watch the trailer by click the "Watch Trailer" button.

When I was working on this part of the project, there some problems I had and here is how I solved them.

The first challenge for me is to design the rating system. The color of the stars need to be changed, and it's depend on the rating that many stars will changed.



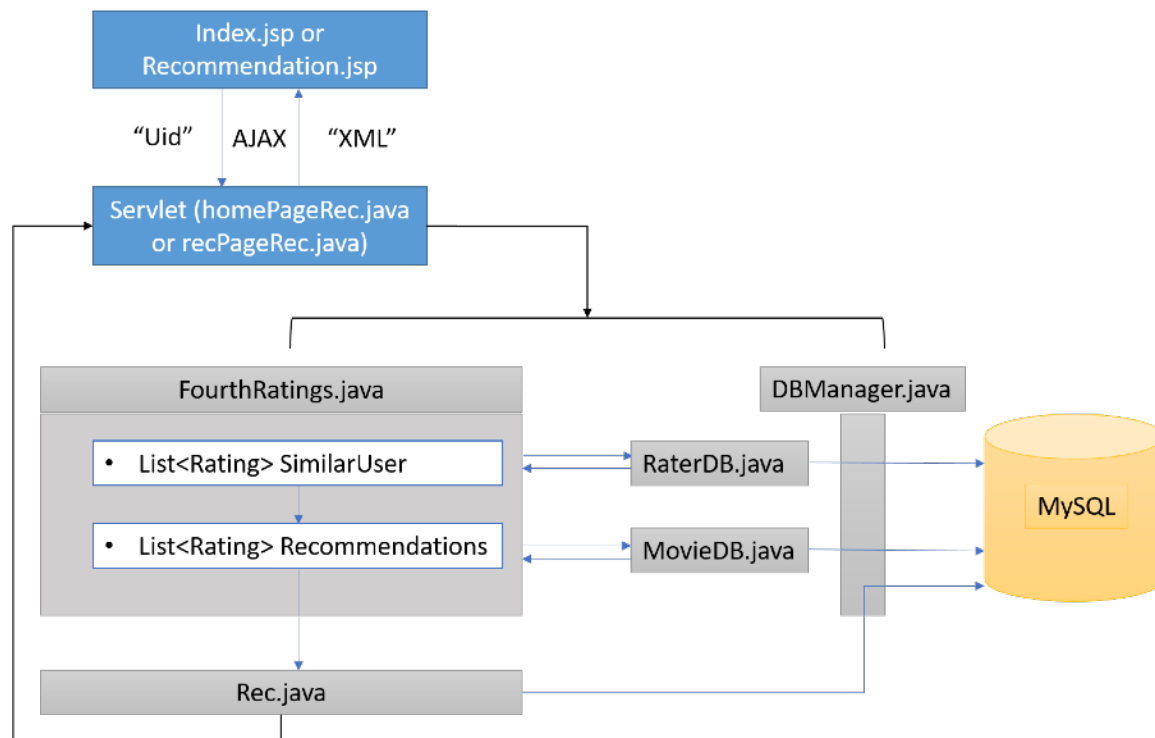
Picture 3. Put mouse on the star, before click action



Picture 4. After click, a rating behavior is completed

To achieve this function, I design a function called "stars" which also takes the rating value.

Part VIII: Recommendation



Picture 5. The recommendation section structure

In "FourthRating":

- Compare "my" rating histories with any users in the database; calculate their "similarity values" stored in a List
- Candidate movies for "me" are from top similar users' watching history. Their rating multiply by the "similarity" produce a "weighted rating" of a movie
- Return a list storing the "MovieIDs" and the "weighted ratings"

Then, the "Rec.java" get all the movie information of the "MovieIDs" in the list from MySQL, pass all those information to the front-end by XML