Movie Recommendation System Management App

Author: Student#1708
Group: 223
Submission Day: 10.06.2023

2023, Moscow HSE, bachelor's Program DSBA

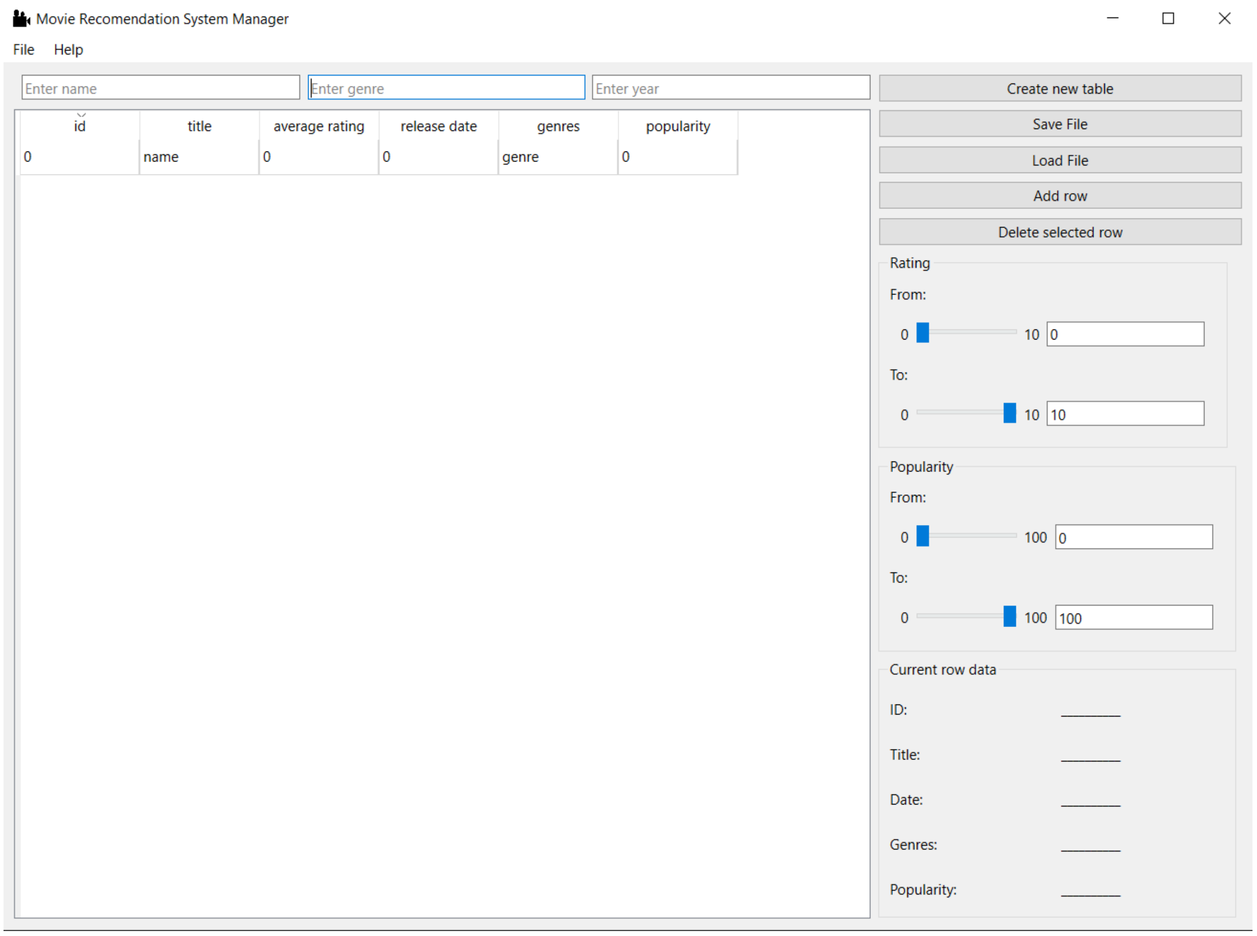
Problem.

The main idea was to create a C++ QT-application, which can read a dataset from proper .csv file and help one to find data, that is needed, via filter systems. Also, this application needs to be able to add and remove entries from table.

So, my project solves two problems:

- It helps to manage data in sphere of movies.
- It helps to choose movie based on filters.

Implementation.



id	title	average rating	release date	genres	popularity
0	name	0	0	genre	0

Rating
From: 0 — 10 0
To: 0 — 10 10

Popularity
From: 0 — 100 0
To: 0 — 100 100

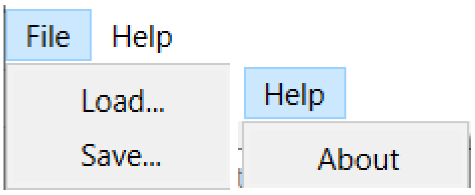
Current row data
ID: _______
Title: _______
Date: _______
Genres: _______
Popularity: _______

That’s the main window of Application. It is sizeable and pretty much very simple UI… But there is an idea:

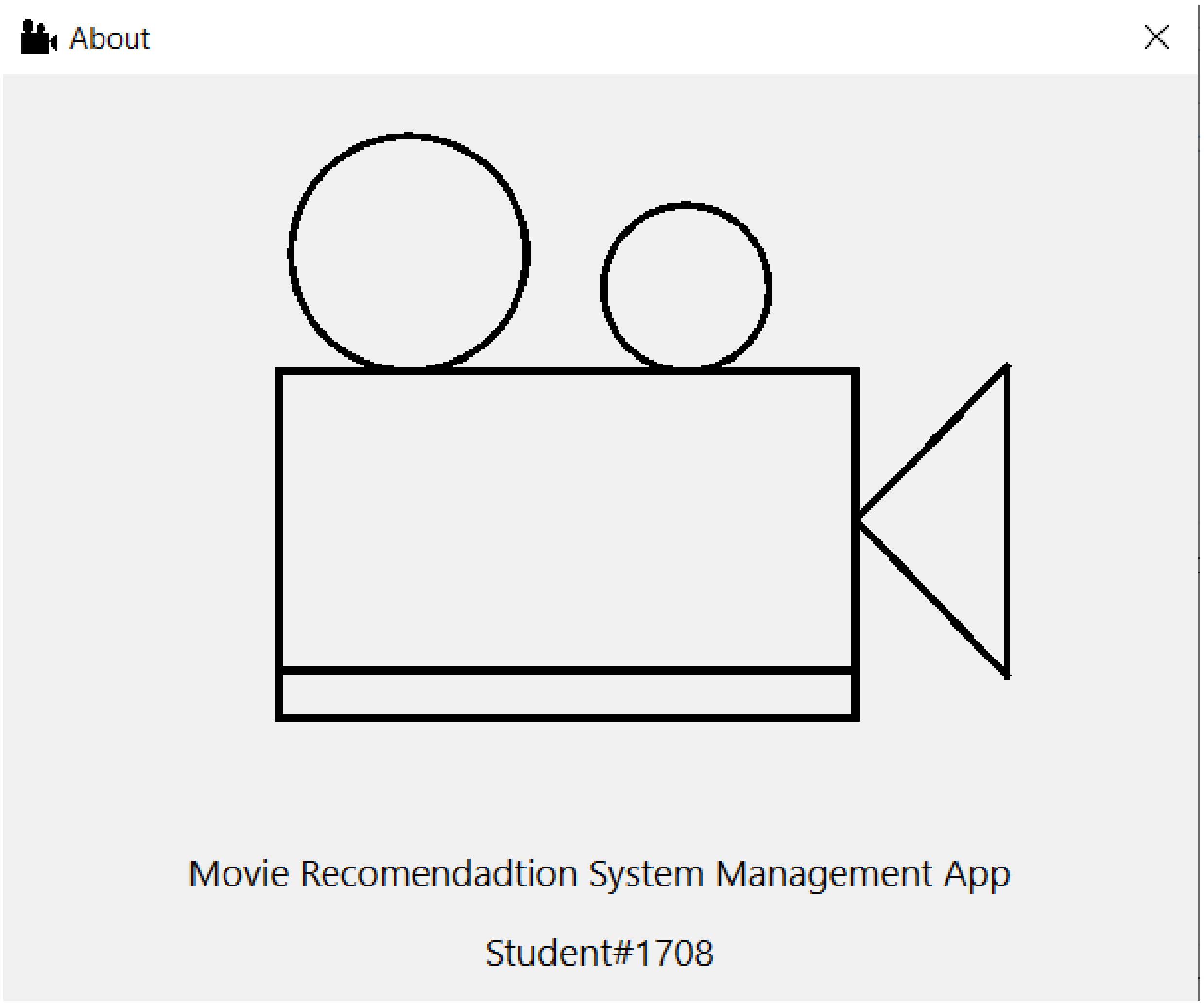
“`Simple is better than complex.`
`Complex is better than complicated.`”

Let’s start with title bar. There you can see icon of application and application name. Icon was added via .rc file in application folder. Name is added via QT Designer function.

Next goes tool bar.



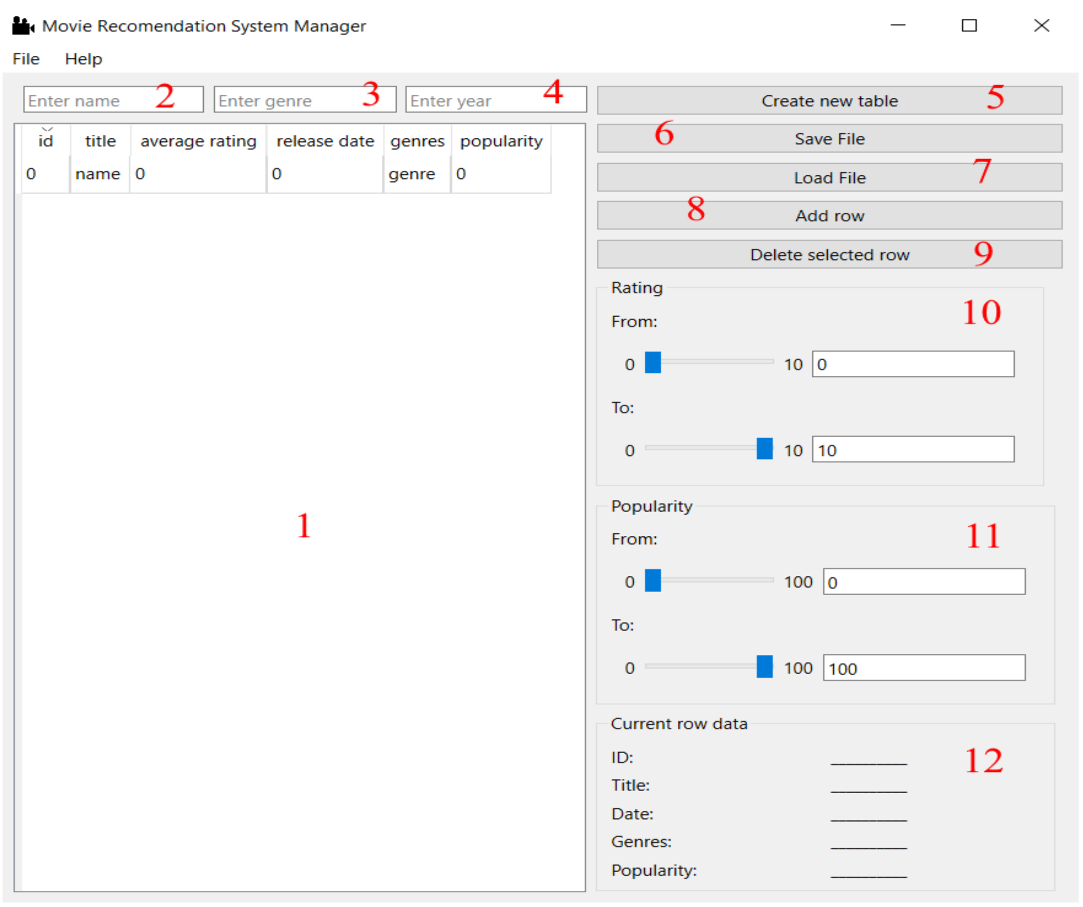
There are two dialog menus. File menu gives option to load table from .csv or save table to .csv file.
Help menu contains About window with application name, student ID and application icon. This window is not sizeable. The code for this icon is provided nearby. It is very simple and uses only lines and elipses.



Movie Recomendadtion System Management App

Student#1708

Now let’s go back to the main window.
Bellow title bar a client area is located.It consists of a many elements, lets look at each of them on the next page.

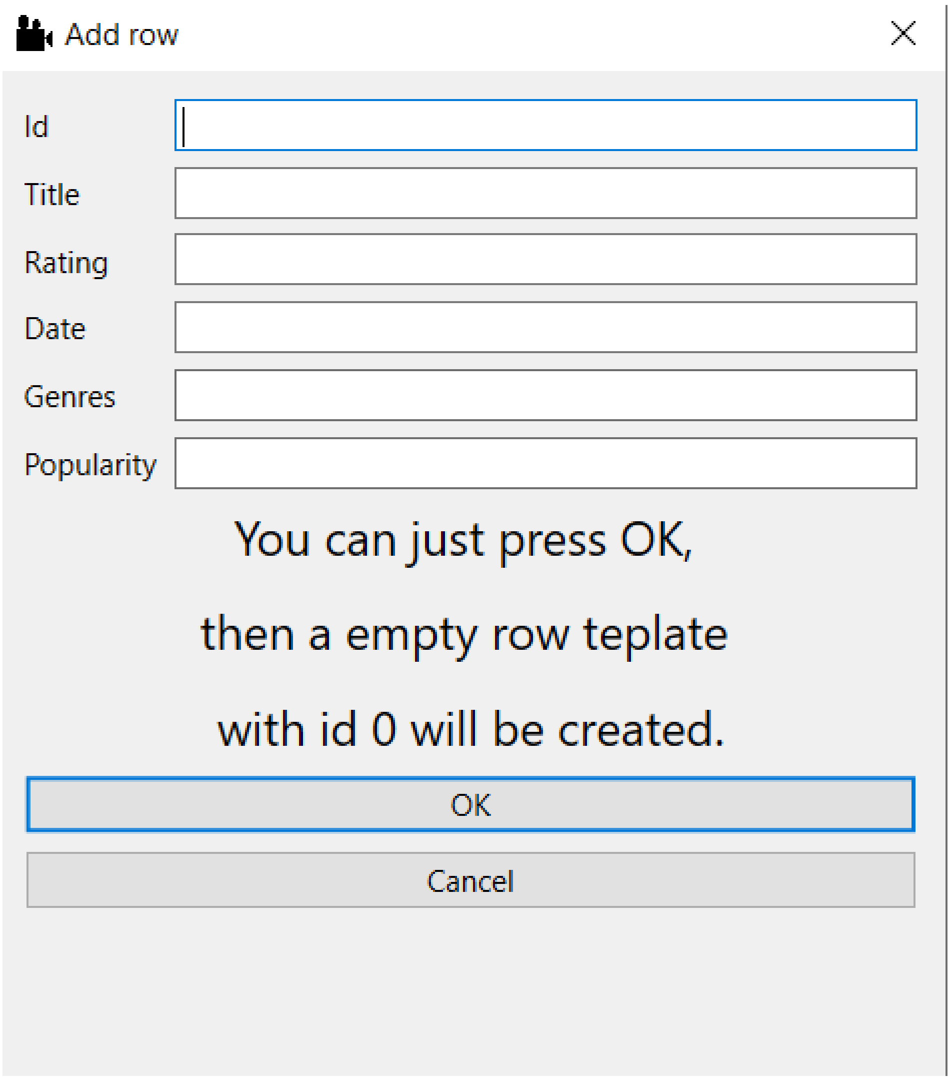


id	title	average rating	release date	genres	popularity
0	name	0	0	genre	0

Rating From:	0	10	0
Rating To:	0	10	10
Popularity From:	0	100	0
Popularity To:	0	100	100

1. QTableView element that show QAbstractTableModel subclass. When you open the app, new table is created automatically. Data can be modified inplace, all restrictions are alredy implemented (look at dataset description in Discussion part).
2. Filter for a name of movie.
3. Filter, where one can specify a genre, in which one is interested.
4. Filter for a year.
5. A button, that creates a new table (starting one).
6. Button to save a table to a .csv file.
7. Button to load a table from a .csv file.
8. Button that adds a new row into current table.
9. Button that removes a selected row from a table. After removing selection is removed. Nothing will happen if no row is selected.
10. Sliders to filter movie by rating. Step is equal to 1 (custom proxy model form QSortFilterProxyMode).
11. Sliders to filter movie by popularity. Step is equal to 10 (custom proxy model form QSortFilterProxyMode).
12. Data from selected row, so one can be sure what data is selected. Here labels are used, which are restored to default value in some cases.

Additional window opens when one is trying to create a new row.



You can just press OK,

then a empty row teplate

with id 0 will be created.

As it says, one is not obliged to write data. It still will work. If not appropriate data entered, it will be changed to the appropriate type of data and row still will be added to the table, so user can fix the row later.

For more information about impementation one can look into the code itself. There are many comments, that describe what is happening.

Discussion.

So, let’s look at problems with dataset. Data set provided with specification, was pretty much unrelated to it. Also, all table data was in .json format and it would be very hard to read such dataset, while implementation of complicated reading system is not a part of my project. So, I created a new one, that has only columns requested from specification and no .json style and with ‘;’ as separator. Except the first one – rank. Because it was assumed that rank must be based on views, but we don’t have data about views in dataset.

Let’s look at its columns:

- ID – nonnegative integer. Assumed to be unique, but this is not required.
- Title – string, no other restrictions, except not using ‘;’ in name and special symbols (such as ‘\t’ or ‘\n’).
- Average rating – float value from 0 to 10.
- Release date (only year – spec. req.) – nonnegative integer.
- Genres – string, no other restriction, except not using ‘;’ as separator and special symbols (such as ‘\t’ or ‘\n’).
- Popularity – float value from 0 to 100.

It is very important for initial .csv to be appropriate, but application should be working in almost every case, if .csv file is readable (no problems with separators).

So, what about special features from specification?

- **Display information on request –** I don’t really understand, what was needed, but user can set filters and see data in the table and special widget.
- **Give recommendations on the 5 most preferred films** – unfortunately, I don’t have user preferences. But user can set filters and see data in the table and special widget.
- **Top 10 best films of a given year** – user can set filter on year and then sort table by rating.
- **Sorting by parameter** – this is implemented, every column of table can be sorted and on each column a filter can be applied.

For more information about impementation one can look into the code itself. There are many comments, that describe what is happening.

Conclusion.

The program is working and doing its work very well. Specification was not about analysis, but about finding movie to watch (recommendation system) and there is almost nothing to improve.

But in case of table management, I think one method could be added (still, I don't know when it may be useful) – it is to delete multiple rows at the same time. Also, create a new table button may be copied to file menu, but it still has not much sense.

But if one wants to extend application to data analysis sphere, the next can be added:

- Table statistics – number of entries (rows), counting of values (ex. How many movies are in horror genre?), data on years (ex. Pie plot with genres on year), more data about rating and popularity.
- Ability to create sub table and save it in another .csv file.

And at this point I'm left of ideas.

But in general, I'm satisfied with the result.