**Report**

***General overview:***

* On the main screen the user has an option to choose between 5 different options:
  + - “1” – search for a movie. We enter a few keywords, and the user sees all titles that match all those keywords. The user is also able to select a movie from the list, and the number of votes, the names of cast/crew members and their characters are displayed for this particular movie.
    - “2” – search for a genre. The user provides a genre and a minimum vote count and sees all titles under the provided genre (again case-insensitive match) that have the given number of votes or more. The result is sorted based on the average rating with the highest rating on top.
    - “3” – search for cast/crew members. The user provides a cast/crew member name and sees all professions of the member and for each title the member had a job, the primary title, the job and character (if any).
    - “4” – add a movie. The user adds a row to title\_basics by providing a unique id, a title, a start year, a running time and a list of genres.
    - ‘5’ - Add a cast/crew member. The user provides a cast/crew member id, a title id, and a category.
    - “q” – exits from the system

***Detailed design:***

* The program tsv-2-json program will handle changing tsv files into json files, this will not be touched by the rest of the general program
* The program load-json program will load a json file to the mongo database for the use of the program below, which will not be touched for the rest of the program below.
* The main() function is responsible for organizing all of the functionality of the program and providing the user interaction in hiding through the program and providing the structure of how the program should be run. There are functions that will be called for the functionality that is specified below and all of this will be in a loop until the user wishes to exit. Lastly, this function provided the exit transitions for this program.
* **Functionality after going into the program:**
  + Search for titles is implemented by search\_movie, this function takes care of the program function of making the user enter one or more keywords and find all titles matching the keywords are returned. Another thing with this function is the matching is defined as in the spec and uses the AND semantics. Lastly, the user can select a title and see the rating, the number of votes, the names of cast/crew members and their characters (if any). This function also uses other functions such as seeDetailedInfo() and search\_by\_key\_words() for more functionality.
  + Search for genres is implemented by searchForGenres(), with this function it provides that the user can provide a genre and a minimum vote count and all titles under the provided genre are case-insensitive with the given number of votes or more are returned. The results are sorted in descending order of the average rating as well.
  + Search for cast/crew members is implemented by searchForCast(), the user can search for the information. The user can provide a cast/crew member name and the function will give all professions of the member are returned and as well as for each title the member had a job, the primary title, the job and character (if any) are returned.
  + Add a movie is implemented by add\_movie() function and the user can add a row to title\_basics by providing a unique id, a title, a start year, a running time and a list of genres. This function after taking those inputs will return both the primary title and the original title are set to the provided title, the title type is set to movie and isAdult and endYear are set to Null.
  + Add a cast/crew member is implemented by a function add\_cast\_member(name\_basics, title\_basics, title\_principals). The user should be able to add a row to title\_principals by providing a cast/crew member id, a title id, and a category. The provided title and person ids should exist in name\_basics and title\_basics respectively (otherwise, proper messages should be given), the ordering should be set to the largest ordering listed for the title plus one (or 1 if the title is not listed in title\_principals) and any other field that is not provided (including job and characters) set to Null.

***Testing strategy:***

* In the beginning, test all queries in order to break everything into small pieces and be sure that everything works correctly using print and checking with the database visually to see for correctness . Then, test separately all the functions and all possible inputs including invalid input and design edge cases. Then try to connect all the functions into one program and check whether the data is transmitted appropriately between functions, and design all connections between functions. Check control functions and all possible pathways in a program depending on user input. Many of the bugs found are incorrect inputs and are fixed along the way after finding.

***Group work break-down strategy:***

* Oleksandr Kasych (kasych): Did everything together with the other group member pair programming (total time spent around 25h)
* Harry Zhao (wpzhao): Did everything together with the other group member pair programming (total time spent around 25h)

Discuss software together before starting writing functions. Discuss the main menu. Implement everything through pair programming. Together go through all edge cases, and test our merged program.

The method of coordination is in person and through Discord.