Programming Assignment – 2

Arya Girisha Rao

Movies and TV Series Database

For this assignment, I built a web-service to search IMDB movies and TV series data and rank them by the users.

Features of the Application –

1. User can search Movies and TV series.
2. User can provide their rating for any Movies/TV Series present in the collection.
3. The rating is updated with the average of existing values and new values.

Technical details of the Application –

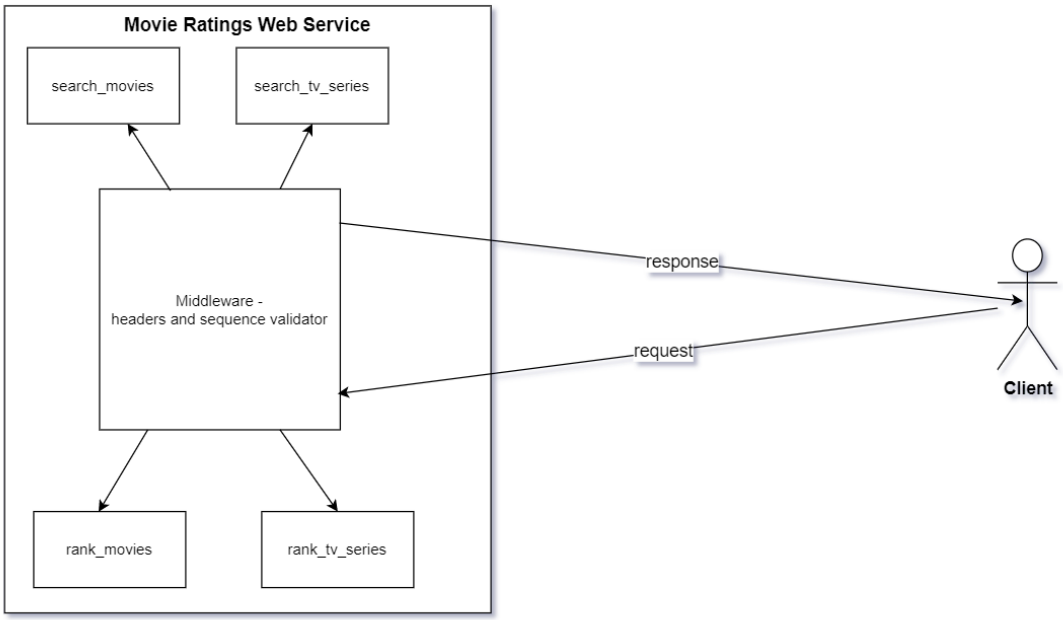
1. The application contains 3 data sets. All the dataset is present in ***input_data folder*** -
    a. Top movies data (TOP_MOVIES_PROCESSED.csv)
    b. Top TV Series data (TOP_TV_SERIES_PROCESSED.csv)
    c. Users' data to validate the username and password (USER_DATA.csv)
2. ***MongoDB*** is used as backend database. The application provides a script (db_helper.py) provide option to load the data sets into local DB connection.
3. ***Python*** and ***FastAPI*** is used to develop the web service and client module to demonstrate web service invocation. All the source code is present in ***src*** folder and client modules are present in ***src/testing*** folder.

Structure and Design of the Application -

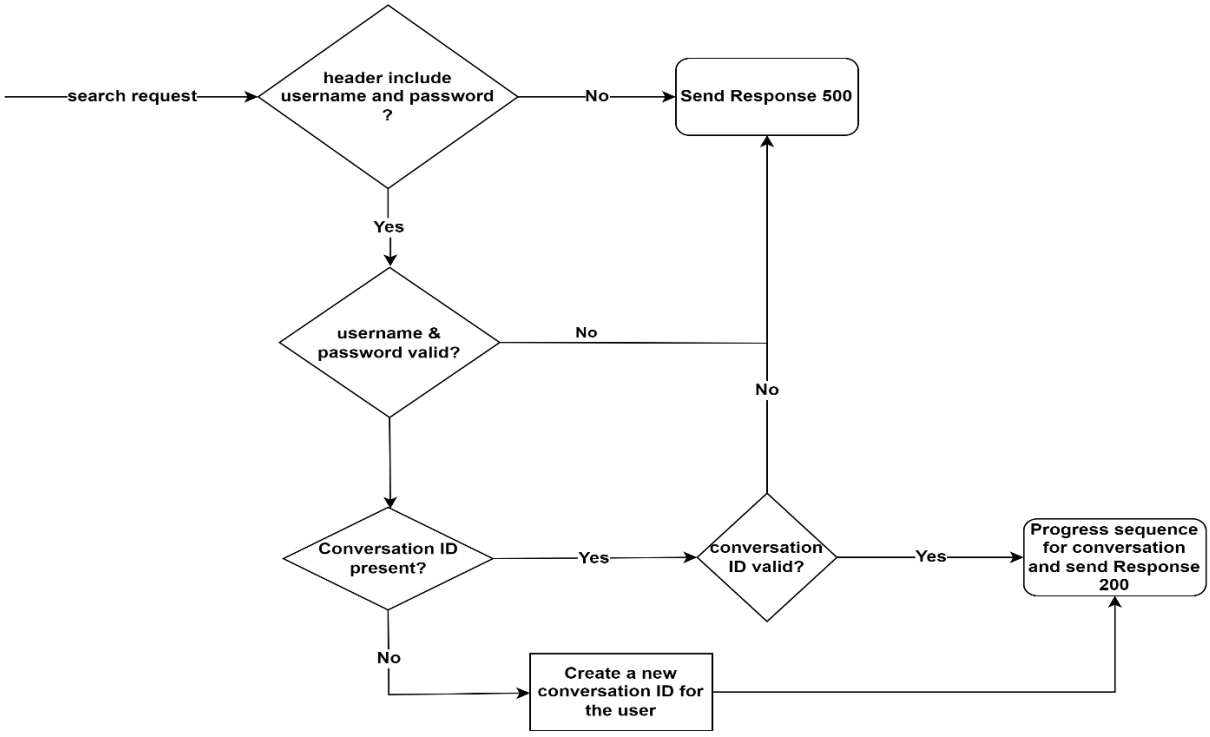
1. The designed web-service includes a middleware (endpoint: ***middleware_manager***) which acts as an intermediary between the client and services. The middleware receives all the input requests, validates the requests by ***processing only the header*** and then forwards the request to required service end-points. It receives the response from the end-point and sends it to the client.
2. There are ***four services*** – search movies, rank movies, search tv series, and rank tv series.
3. The co-ordination policy of the web service – search operation needs to be performed before rank operation. Also search and rank should be for the same type of the data (movies/tv series) – i.e. searching movies and trying to rank tv series is an invalid sequence.
4. A user can ***have more than 1 conversation*** with the service. Each service keeps track of all the search operations performed in that conversation so validation can be done.
5. Whenever a search request is received, the middleware checks the header and validates the username and password. If the validation fails, it returns a Response of 500. If the validation passes, it moves to validating the conversation ID. If the client has not provided the ID, it'll assign a new one otherwise it validates whether the conversation ID matches to the username. If validation is passed, the search operation is marked complete and now user can make a post request.

6. Whenever a post request is received, the middleware the checks the header and validates username, password, and conversation ID. Then it checks if the conversation ID is marked complete. If the search operation is completed, it allows the update of the rating else the client is informed to perform the search first and then send a post request.

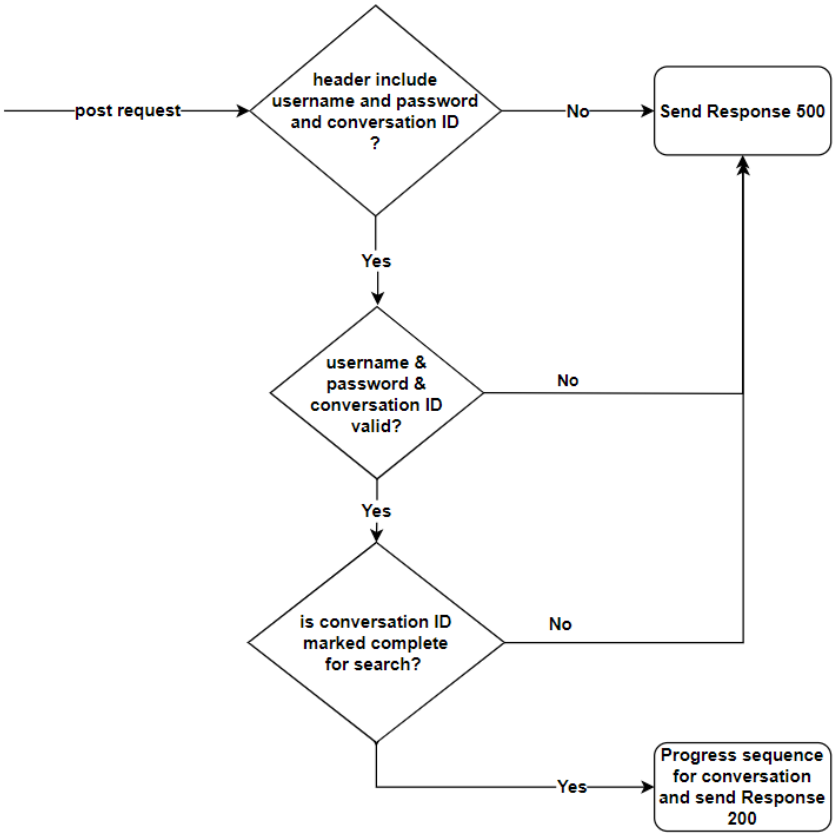
System Representation of Web service -



Get request processing flowchart from the middleware -



Post request processing flowchart from the middleware -



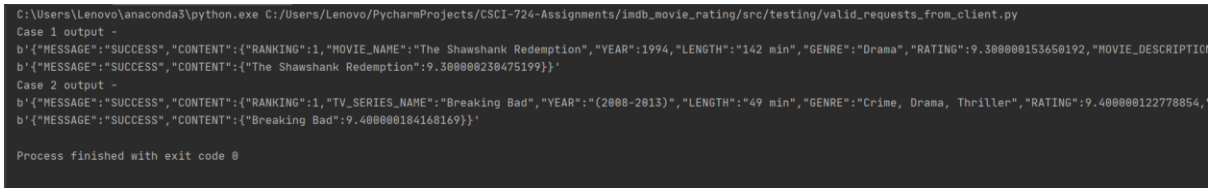
Invalid cases tested from client side in the file -src\testing\invalid_requests_from_client.py-

1. **Case 1 – function name - invalid_conversation_1()** – The request does not contain username and password.
2. **Case 2 – function name - invalid_conversation_2() –** The request contains an invalid username and password.
3. **Case 3 – function name – invalid_conversation_3() –** The client makes search request first and then rank request but the second request does not contain conversation ID.
4. **Case 4 – function name - invalid_conversation_4() -** Client trying to rank a movie before searching for the movie.
5. **Case 5 – function name - invalid_conversation_5() -** Client trying to rank a movie after searching for the TV series.

Valid cases tested from client side in the file -src\testing\valid_requests_from_client.py-

1. **Case 1 – function name - valid_conversation_1()** - Client tries to search movies with valid request and then tries to rank movie with valid request including conversation ID received from the server.
2. **Case 2 – function name - valid_conversation_2()** - Client tries to search movies with valid request and then tries to rank TV series with valid request including conversation ID received from the server.

Valid case output screenshot –



Invalid case output screenshot –