**Movie Details System**

**Purpose:**

The purpose of this system is to provide details of movie which client ask for. Details include name, overview, rating poster etc.

**Architecture:**

There is a main server which waits for client requests. Once a client gets connected, initial requests are transformed (described later) and then data is fetched from database (a back end server) to serve the client. Database stores the details of all movie. When a movie detail is provided, its poster is also provided. Poster is saved on the file system. Diagrammatical representation of architecture is like this:

DATABASE

CLIENT

MAIN SERVER

CLIENT

FILE SYSTEM

CLIENT

**Request Type:**

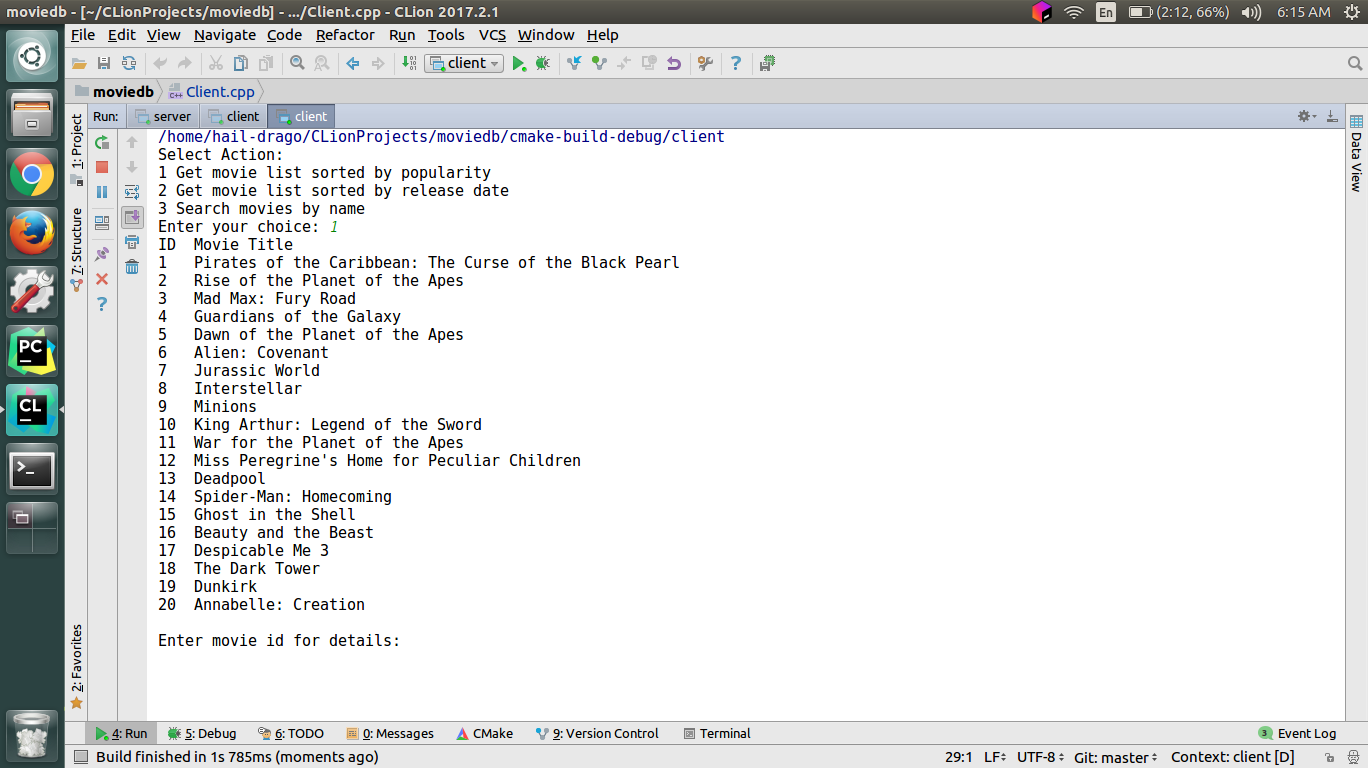
There is various different type of request in the system:

First when a client starts, it is asked by which means it wants to search movie. There are 3 option available:

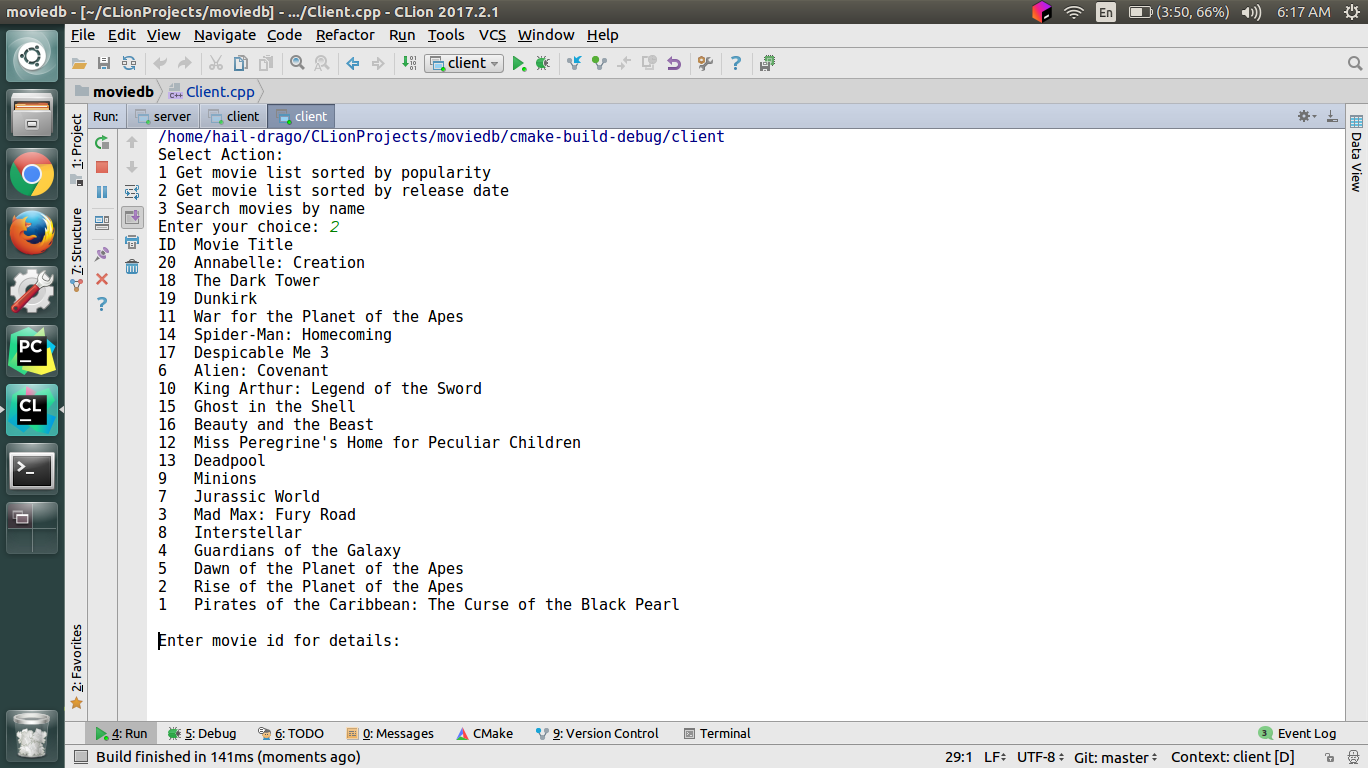
1. Get movie list sorted by Popularity.
2. Get movie list sorted by Release Date.
3. Search movie by Name.

All the 3 options are 3 different request. And the output corresponding to them are shown below:

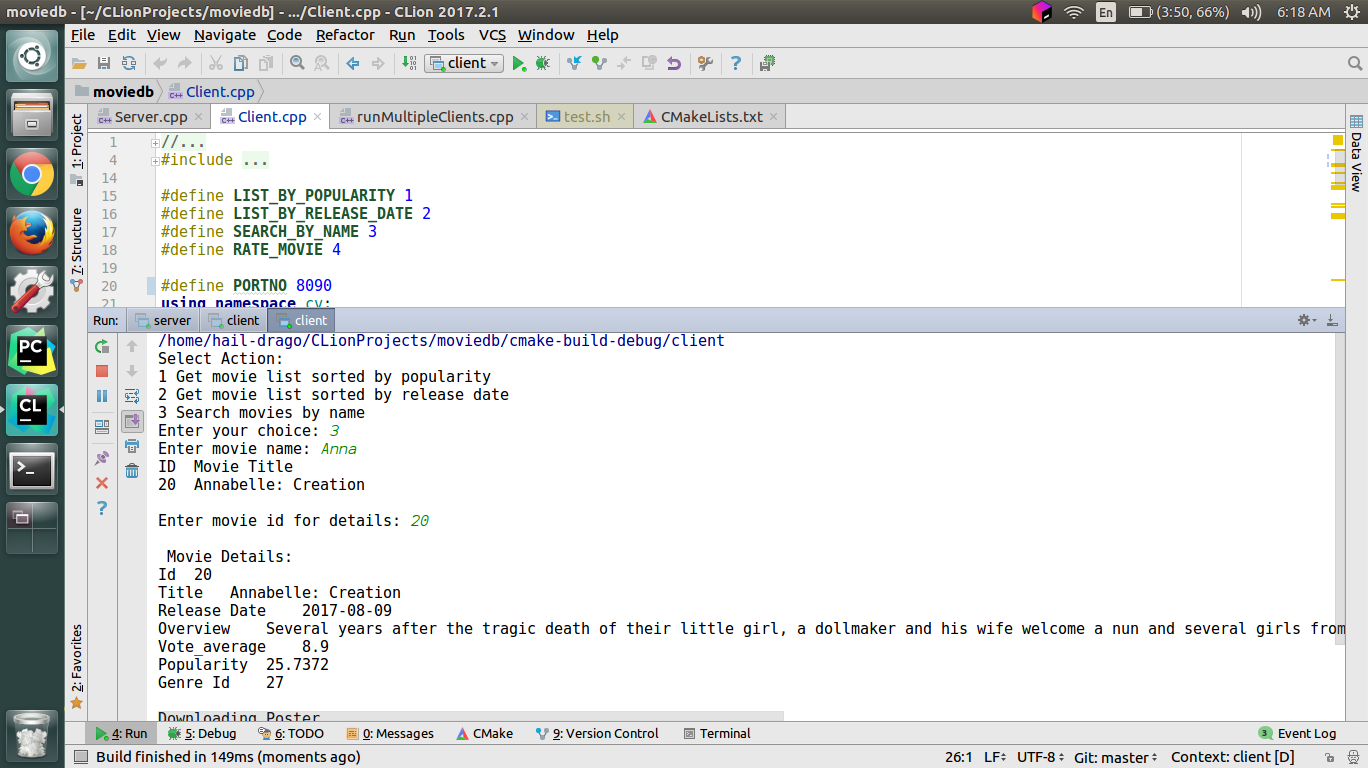
1. It returns list of movies in DB sorted by popularity

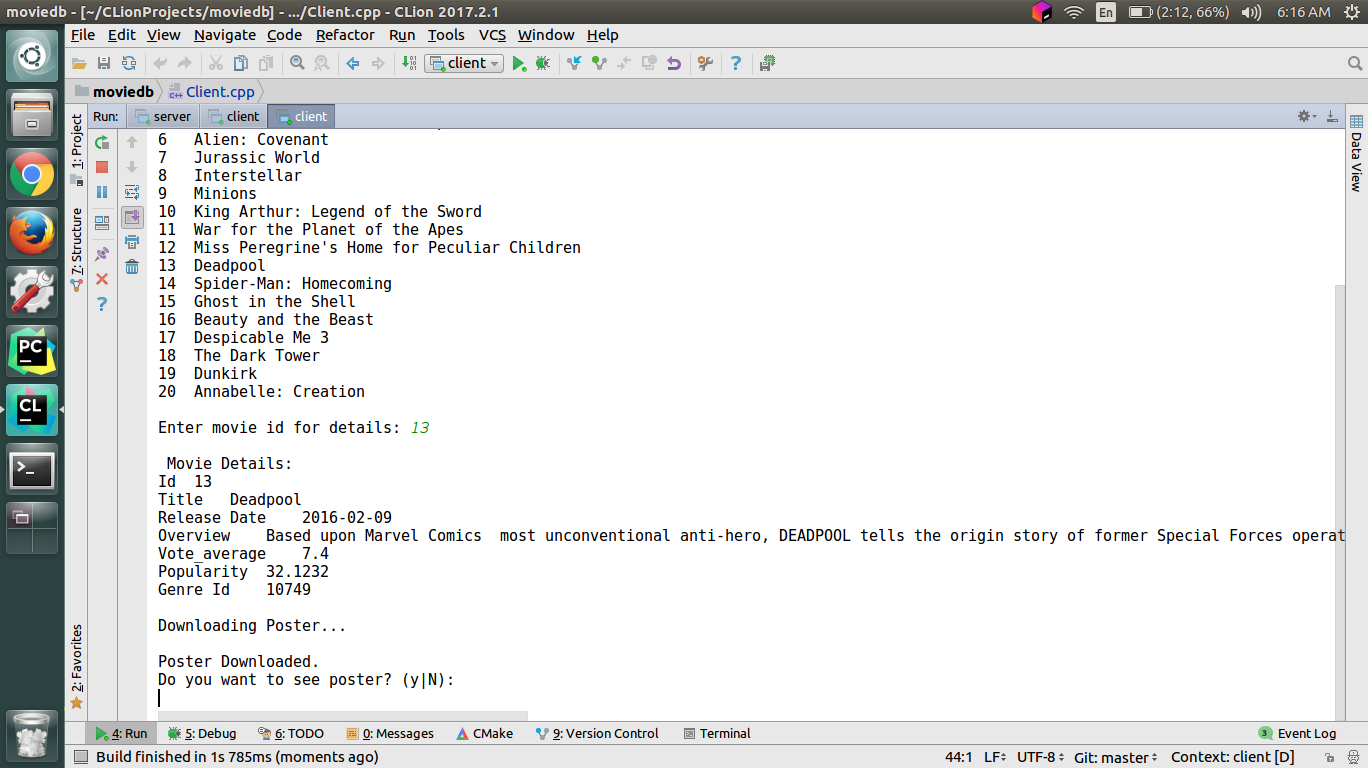


1. It returns list of movies in DB sorted by date



1. In this request, use is asked for a name and list of all movies matching that name are returned.



Once the list is there user enters the “Id” of movie which it wants to see. And data of that movie will be displayed and poster will be downloaded. So in this request, we are accessing both database and file system.

User can press y to see poster. After this client is asked if it wants to rate the movie or not, if yes then rating is taken and a new type of request is send to server. It updates the rating in db.

