Project 1 – Netflix Menu

```
PS C:\Users\danie\Documents\Fresno State\Fall 2022\CSCI 41\Project 1 - Netflix Menu>  & 'c:\Users\danie\.vscode\extensions\ms-vscode.cpptools-
1.12.4-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-0uhb5jbo.155' '--stdout=Microsoft-MIEngine-Out-qct
du03p.jj4' '--stderr=Microsoft-MIEngine-Error-5olfauak.p5s' '--pid=Microsoft-MIEngine-Pid-zyceru5t.gsy' '--dbgExe=C:\msys64\mingw64\bin\gdb.ex
e' '--interpreter=mi'
The movie The Batman was found
The movie Interstellar was not found
The movie The Hangover was found

All Movies on Netflix:
Mission: Impossible - Fallout (7/27/2018)
The Batman (3/4/2022)
Cars (6/9/2006)
Monsters, Inc. (11/2/2001)
21 Jump Street (3/16/2012)
The Hangover (6/2/2009)

The Batman's index is 2
Category Comedy was found
Category Animation was successfully removed

Category Animation was not found
PS C:\Users\danie\Documents\Fresno State\Fall 2022\CSCI 41\Project 1 - Netflix Menu>
```

In my project, I created a CPP file “Project 1” with a main() function, where I included the header files containing all the classes and structures needed for the Netflix Menu.

The first header file “Date.h” contains the Date class and provides a Date object that will be used for the movies’ release dates.

The second header file “movie.h” contains the structure “movieNode”, handling the movies in the class “movies”, which handles the movie lists (implemented as double linked lists) as well as the functions used to interact with the movie lists.

The third header file “category.h” contains the structure “categoryNode”, handling the categories in the class “categories”, which handles the category lists (also implemented as double linked lists) as well as the functions used to interact with the category lists and the movie list within each category.

When we run the program, we get the output shown in the image above. If we take a look at the main() function in Project 1.cpp, I added three categories to the Netflix Menu: Action, Animation and Comedy. Next, I added 2 movies with their release dates to each respective category, which I checked for some of them whether they have been successfully added using the “searchMovie()” function. In addition, I checked the case for a movie that has not been added.

On the following lines, I printed all the movies, with their respective release dates, from all categories available on the Netflix Menu using the “printAllMovies()” function. Next, I returned the index of “The Batman” movie in the list of its category using the “getMovieIndex()” function, which turned out to be at index 2. Afterwards, I used the “searchCategory()” function to check whether or not the Comedy category is available on the Netflix Menu, and the output was positive. I also tried utilizing the “removeCategory()” function to remove the Animation category from the Netflix Menu. To check whether or not the removal was successful, we use the ”searchCategory()” function again by passing the Animation category as a string, and the output then confirmed that the Animation category was not found.

Unfortunately, I wasn't able to test the function "searchMovie()" using Binary Search as the implementation wasn't successful. I ended up implementing it with linear search instead, but the code shown below is the closest I was able to get for Binary Search: (not included in the Project zip file)

```
bool categories::binarySearchMovie(categoryNode* cat, string title, int low, int high) {

    movieNode* middle = cat->movieList->getHead();

    bool found = false;
    if (low > high) {
        found = false;
    }
    else {
        int mid = (low + high) / 2;
        for (int i = 0; i < mid; i++) {
            middle = middle->next;
        }
        if (middle->data == title) {
            found = true;
        }
        else {
            if (middle->data < title) {
                found = binarySearchMovie(cat, title, mid + 1, high);
            }
            else {
                found = binarySearchMovie(cat, title, low, mid - 1);
            }
        }
    }
    return found;
}

void categories::searchAllMovies(string title) {
   categoryNode* current = head;
   while (current != nullptr) {
        if (binarySearchMovie(current, title, 0, current->movieList->getNumMovies())) {
            cout << "The movie " << title << " was found" << endl;
            return;
        }
        current = current->next;
   }
   cout << "The movie " << title << " was not found" << endl;
}
```