Ahmad Afridi

107576832

17 April 2019

Design Document 2421

**Problem Description:** This program is designed to create a simple database system. The database is designed to handle multiple records of actors and actresses and their respective movie, and whether they won an award or not. Then by printing the files to a console the user will be able to manipulate those files by adding or deleting records. This will be achieved using an ostream operator that will be able to change the records directly in the file.

**Overall Software Architecture**

External Data File

BST

Tree

Modify

Function

Search

Function

Main

Menu

Delete

Function

Export

Function

Sort

Function

Partial Search

Function

The basic software architecture is all intertwined between the main menu. The ones in red are the functions that appear on the menu, the one in blue is the external file that gets sent to the project folder.

**Input Requirements:** The input requirements for this code will be 2 CSV files. One csv file contains the year, the type of award, whether the actor won or not by using 0 or 1, the name of actor or actress, and the name of the film. The second CSV is a film or pictures CSV which provides the name of the movie, the year the movie was produced, how many awards the movie was nominated for, the rating, the duration of the film in minutes not hours, the two types of genre, genre 1 and genre 2,the month of its release, the Metacritic score which is a score based on how well the film was on a scale from 0-100, and the finally a brief synopsis of what the movie is about. The other inputs will be from the user on how they want to add or delete records from the csv files. Those will be inputs of strings and one Boolean for the whether the actor won an award or not. The length of the user input should be limited to the size of the variable holds. For example, the name string should not be 170 characters long since it is just a name and the bool should just be a 1 or 0 for true or false.

**Output Requirements:** The output will the two csv files. The first CSV file will output the year, the type of award, whether the actor won or not by using 0 or 1, the name of actor or actress, and the name of the film. Then the user will be asked to enter what record they would like to search either from the movie csv or the actor csv. This output will be sorted through the binary search tree and will be printed to the console. Then the user will be able to modify the records in the console. Then the user will also be able to delete the records in the console. The user will also be able to export there input the rest of the csv file as well to another csv or text file.

**DATA STRUCTURES:** The main data structure is the binary search tree that holds all the information about the csv file. There are many smaller sub structures that are present like the Key() structure. This key function holds on to the main category that is required for the project. For example, the key category of the actor csv is the name category. After that all the categories and their structures are put inside the data category aka the data () function. The parent, left, and right functions are all individual nodes part of a bigger data structure. These variables are private, but they have individual setters and getters to set them to either the left or right of the tree. The tree can have multiple left and right nodes but can only have one parent. This is the most efficient way to create a tree, that is why you create the parent, then comes the left node followed by the right node.

**USER INTERFACE:** To navigate through the menus, the user will first be able to search if a record exists, for example they will search for an actor/actress by name or movie by name then the user will have the option of deleting, modifying or adding a record. Then it will show where the record has been added or modified or deleted. I will most likely use a switch statement to allow the user to make those modifications in the menu from the csv.

**Status of Application:** The application has been complied on xcode and does provide full functionality on the cse grid of 4.8.

**Problem Solution Discussion:** The first step in order to create the program was to create nodes and convert them into linked lists which in turn you need to convert to a binary tree by using the binary tree hpp file and header file. Then you must load the data of the csv file using ifstream operators and multiple pointers and multiple getliness to retrieve the data from both the pictures csv and actor’s csv. After the getline has been used you must assign the data to the respective node by using the pointers. This will load all of the data into the binary tree. Next for the majority of the functions they will use the cin.get to retrieve the input from the user. Then use a switch case statement to acquire the users input and to call the functions that modify a record, add a record, search a record, etc. This is the overall structure that was used to create the project and solve the problem. On a side not, the Binary tree will always sort by Key(key is the head of the data) not by data.

**User Interface Diagram**

Modify and return to menu

Ask for the field to modify

Search for the record

Ask for the field to sort

Sort by the field and return to menu

Modify a record

Functon

Sort by field

Main

Menu

Search a record

Import

Ask for the name of the file

Ask for the field to search in

Delete a record

Function

Export

Function

Add a record

Ask for search within the search result

Search for the record

Find the result and return to menu

Make A new file

delete and return to menu