Author: Ben Scherer
Class: CIS 1111
Date: 8/3/2017
Project: Final

Description

Movie database program. Leverages flat file delimited text file. Results are then read into a vector utilizing a custom struct. Search/Update/Delete/Add are the supported functions.

Usage:

1. Search/Modify Records
    a. Search database
        i. By Title
        ii. By Name
        iii. By Year
        iv. By Genre
    b. Choose Record
        i. Update Record
        ii. Delete Record
2. Add Records

Dependencies:

Movies.csv

Delimited text file. Database that is loaded/written too. Needs to be in the same directory and the program

Header Files:

#include <iostream>

Used for cout

#include <iomanip>

used to manipulate cout

#include <string>

needed for string variable

#include <math.h>

used for basic arithmatic

#include <limits>

user for numeric_limits

#include <vector>

needed to use vectors

#include <fstream>

file handling

#include <sstream>

used for string bufffer

#include <cstddef>

Functions:

```
void displayWelcome();
void movieDetails(vector<Movie> &movies, int movieIndex);
        display details about movie
int getIntInput(string questionToAsk, string errorMsg);
        validates int input and int range
int getIntInput(string questionToAsk, string errorMsg, int lowRange, int highRange);
        validates int input and int range
string getInput(string questionToAsk, string errorMsg);
        validates string input
Movie parseData(string movieString);
        parses data from comma delimted string and returns Movie object
vector<Movie> loadDB(string strFileName);
        loads data from flat file to vector
Movie addMovie();
        add movie to database
vector<int> searchTitle(vector<Movie> movies);
        search for movies by title
vector<int> searchGenre(vector<Movie> movies);
        search for movies by genre
vector<int> searchActor(vector<Movie> movies);
        search for movies by actor
vector<int> searchYear(vector<Movie> movies);
        search for movies by year
void searchMovies(vector<Movie> &searchVector);
```

main function for handling the different search types

char getSentinel(string message,string errMessage);

Input validation for sentinel. simple y/n

Movie updateMovie(Movie mvObj);

update movie

void displayMovie(Movie mvObj);

display movie

float getFloatInput(string questionToAsk, string errorMsg, int lowRange, int highRange);

void updateDB(vector<Movie> movies,string file);

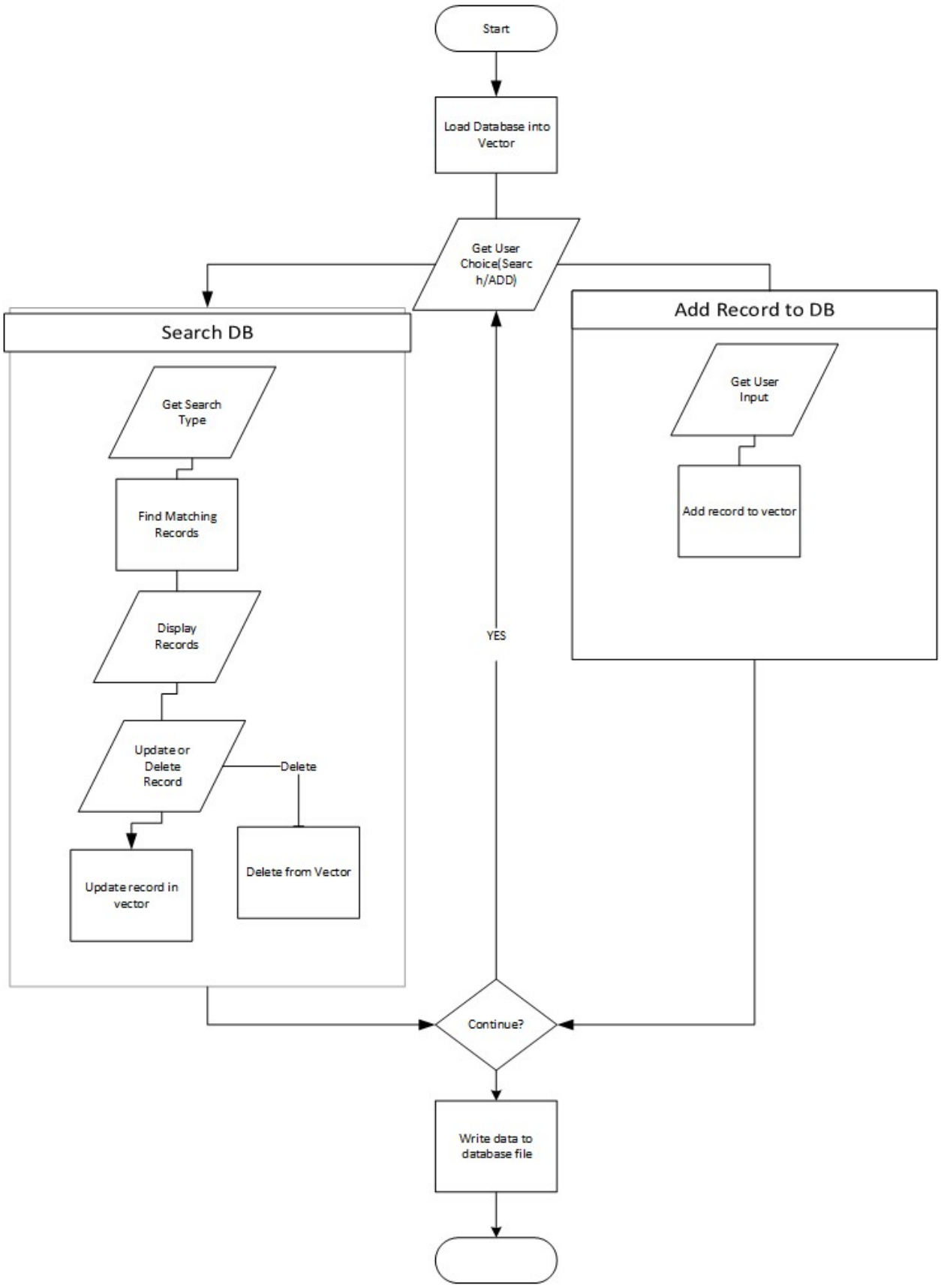
write changes to file

Structures/Classes:

```
//data structure for each individual movie
struct Movie {
	string title; //name of movie
	string studio; //name of studio
	string contentRating;
	int year; //year of release
	float rating; //10 star rating system
	string genre; //genre of movie
	string releaseDate; //date of release
	string writers; //writers of movie
	string runtime; //runtime
	string directors; //directors
	string actors; //actors


};
```

Flow:


Start
Load Database into Vector
Get User Choice(Search/ADD)
Search DB
Get Search Type
Find Matching Records
Display Records
Update or Delete Record
Delete
Update record in vector
Delete from Vector
YES
Add Record to DB
Get User Input
Add record to vector
Continue?
Write data to database file