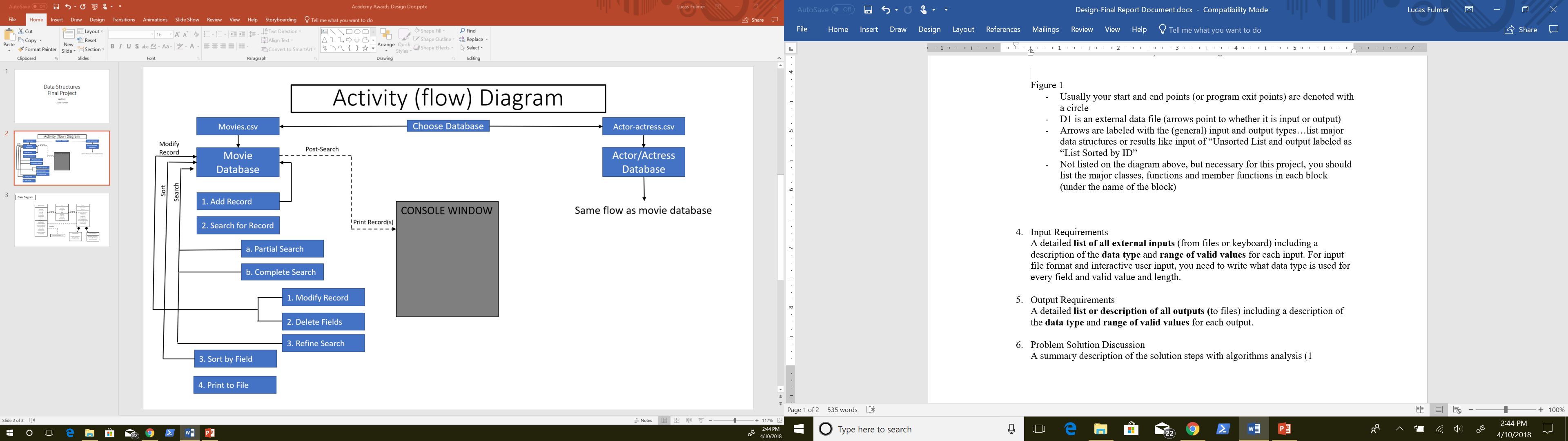
**Project Design/Implementation Document**

Title: Academy Awards Database

Author: Lucas Fulmer

ID: 108963697

Date: 2 May 2018

1. **Status of Application**  
   The program compiles, runs, and has been tested on CSE grid. The program meets all requirements of the assignment and is fully functional. One notable feature is that the search algorithms are all CASE SENSITIVE. If you search for something and don’t receive a return, there is likely an incorrect usage of lower case letters.
2. **Problem Description**This program will develop two databases for Academy Award recipients. One database will contain actors and actresses who have won awards and the other database will contain movies that have won awards. The program will allow the user to sort, search, and modify the databases. It also allows the user to print the entire database before or after modifications.
3. **Overall Software Architecture**Each database (1. Movie 2. Actor/Actress) will use a Binary Search Tree (BST) as the main abstract data type (ADT). The two main classes to be used will be: 1. MovieFile and 2. ActorFile, each with one BST corresponding to the class. The program itself will use a text base menu to allow the user to sort and search based on multiple criteria. The program will allow the user to further filter their search, in the event that there are multiple returns for a given search.  
                
   
4. **Input Requirements**

**Intro Menu:**

Keyboard Input (requires the user to choose integer 1 or 2)

1. Enter Movie Database

(data input – pictures.csv)

1. Enter Actor/Actress Database

(data input - actor\_actress.csv)

**Database Menu:**

Keyboard Input (requires the user to choose integer 1 – 5)

1. Add Record
2. Search for Record (keyboard input character ‘a’, ‘b’, ‘c’)
   1. Modify Record
   2. Delete Record
   3. Refine Search
3. Sort Database by Field (keyboard input ‘a’, ‘b’, ‘c’, ‘d’)
   1. Name
   2. Year
   3. Award
   4. Film
4. Print Database to file
5. Exit

1. **Output Requirements**Each database (movies and actor/actress) will be printed to a .csv file. This can occur before or after user-modifications to the database.
2. **Problem Solution Discussion**  
   The program will begin by reading-in the pictures.csv and actor\_actress.csv into the appropriate database. As the files are being read-in each entry will be placed into the appropriate binary search tree (initially sorted by “Title” for pictures and “Name” for actor/actress). The user will then be given the option to enter into either the picture or actor database. After this, the user may add or edit entries, sort and search the database, or print the database to a .csv file.

1. **Data Structures**The main DS for this program will be the two binary search trees. At this point it is somewhat unclear as to which additional ADTs the program may need, but I foresee using a vector for search refinement. If the user searches for an entry and then wishes to refine the search, I believe it may be simpler to add the entries from the original search into a vector. A second search of the BST will require a search of every leaf, whereas if we add the entries to a vector, we will only need to search those specific entries. Either way the refined search will be Big O(n), but in reality (n) will be much smaller if we keep track of the original search’s returns.