**COIT20256 DATA STRUCTURES AND ALGORITHMS**

**Term 2, 2017**

**Assessment 2**

Lecturer and Tutor: Mahbub Ahmed

Prepared by:

Student Name: Prajalkumar Natvarlal Ramjiyani

Student Number: 12045947

Contents

[1. Reasons behind using ArrayList 3](#_Toc494321105)

[2. Reason behind selecting Insertion Sort Algorithm for Sorting 4](#_Toc494321106)

[3. Reason behind selecting Linear Search Algorithm for Searching 5](#_Toc494321107)

[4. Compare timings for sorting and searching algorithm 6](#_Toc494321108)

[5. Output Snapshots 7](#_Toc494321109)

[6. Challenges faced while working on this assignment 14](#_Toc494321110)

# Reasons behind using ArrayList

* In Array List search option is very fast compare to Linked List.
* When we are using get (int index) in array list it gives performance of O(1) while in linked list it given O(n).
* Array List uses the array data structure to maintain the data which is fast for searching. On other hand, linked list uses the double linked list structure so whenever search is required it travel through all elements for searching.
* Whenever we required to add and delete data from list frequently then we can use the linked list but for searching and sorting functions better to use array list as it work faster than linked list.

# Reason behind selecting Insertion Sort Algorithm for Sorting

* Insertion sort is one the most famous and stable sorting algorithm.
* It is faster for small number of records so it suite to us in this assignment as we have limited 1000 records max to sort.
* It also required small amount of memory to perform sorting which is also suitable for laptop.
* It performs the operation with O(n2) worst case time so it is adaptive for small size of data. Hance, I selected this light weight algorithm to perform sorting of small amount of data.

# Reason behind selecting Linear Search Algorithm for Searching

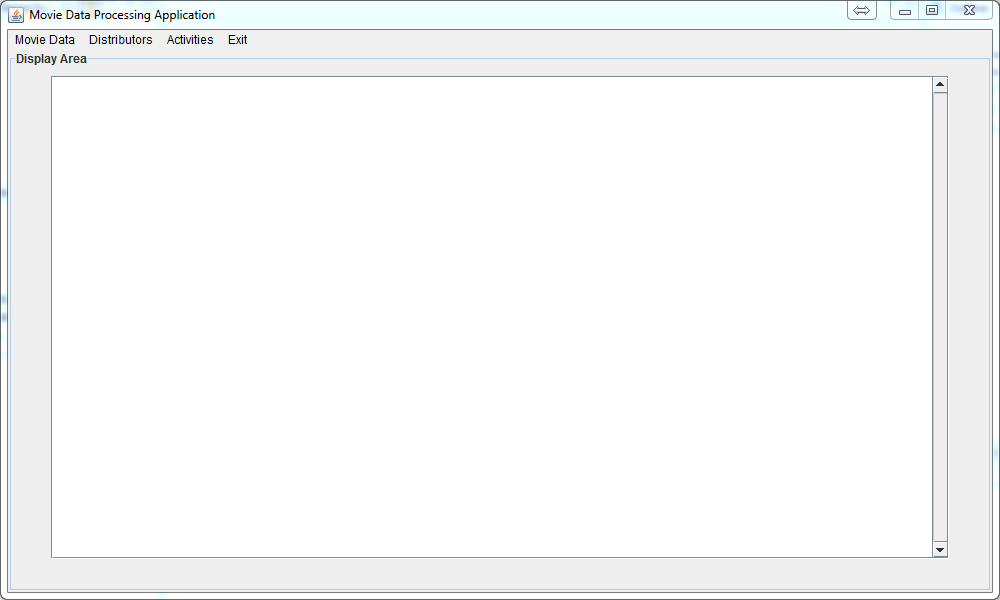
* Linear search performs worst case at n comparison where n is nothing but a size of the list. Main reason behind selecting this search algorithm is the list is unsorted. Here we are searching for a ticket price in this application/assignment and it is never sorted so we can’t utilize other good search algorithms like binary search which give good performance for sorted array. If we want to use it still we must have sort the array first then we can apply that algorithm.
* In real life scenario, Linear search is fastest search algorithm for unsorted list compare to other algorithms. In real implementation, we always use linear search.
* It is memory and resource efficient as it required only one element at a time to compare.

# Compare timings for sorting and searching algorithm

|  |  |  |  |
| --- | --- | --- | --- |
| Records | 10 movies | 100 movies | 1000 movies |
| Sorting Algorithm time in milliseconds | 0 | 2 | 30 |
| Searching Algorithm time in milliseconds | 0 | 0 | 1 |

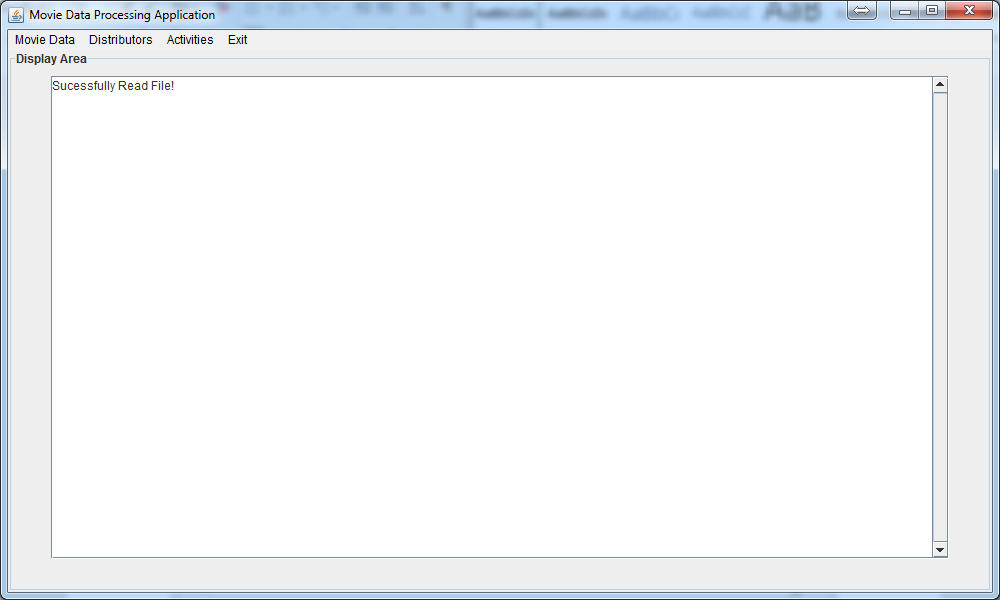
# Output Snapshots

Below snapshot 1 shows the home page of application



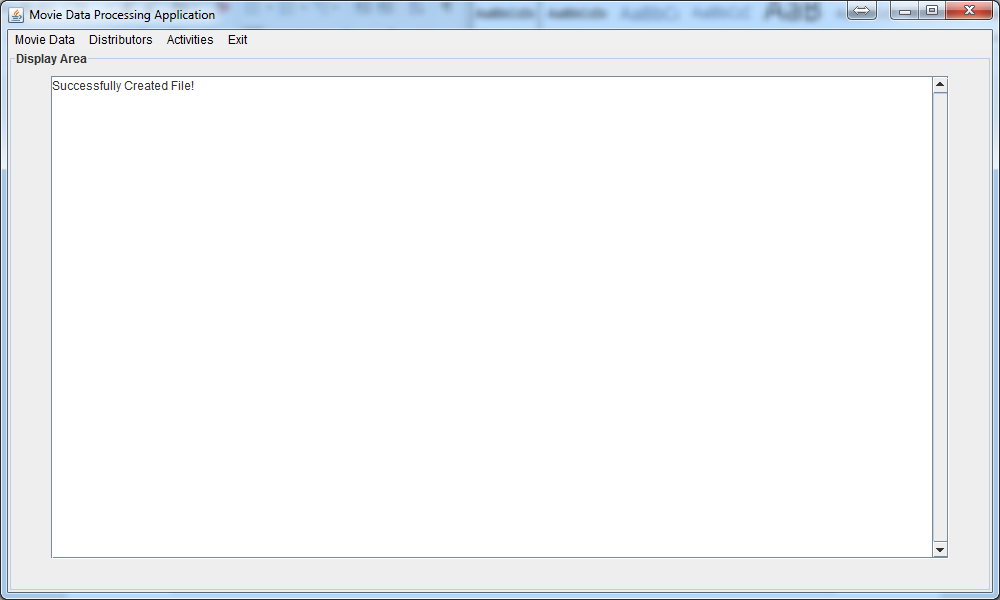
[Snapshot 1]

Below snapshot 2 describe the read functionality of application and indicate that file read successfully.



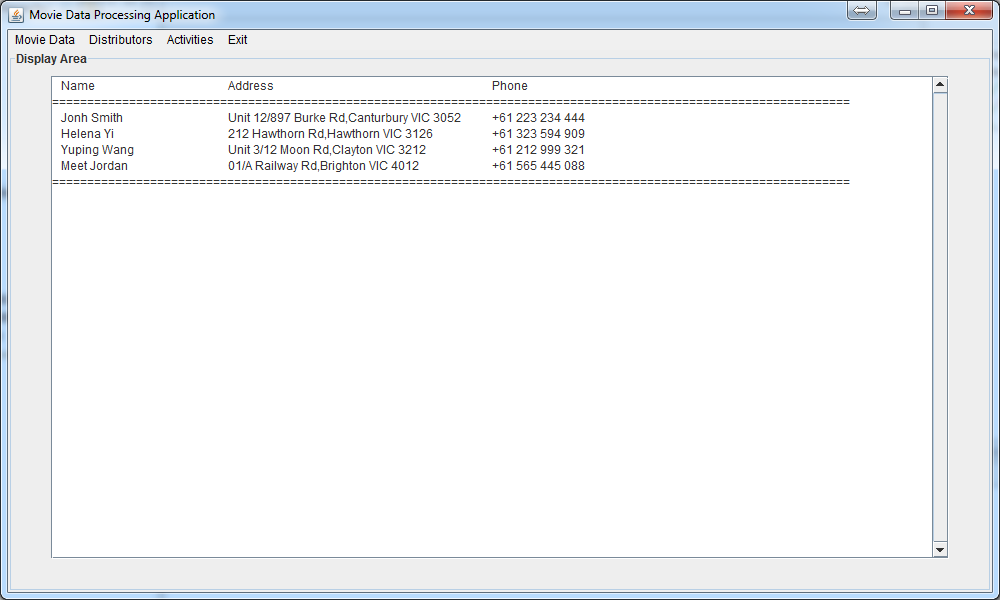
[Snapshot 2]

Below snapshot 3 gives the message of successfully file created by save functionality under Movie data menu.



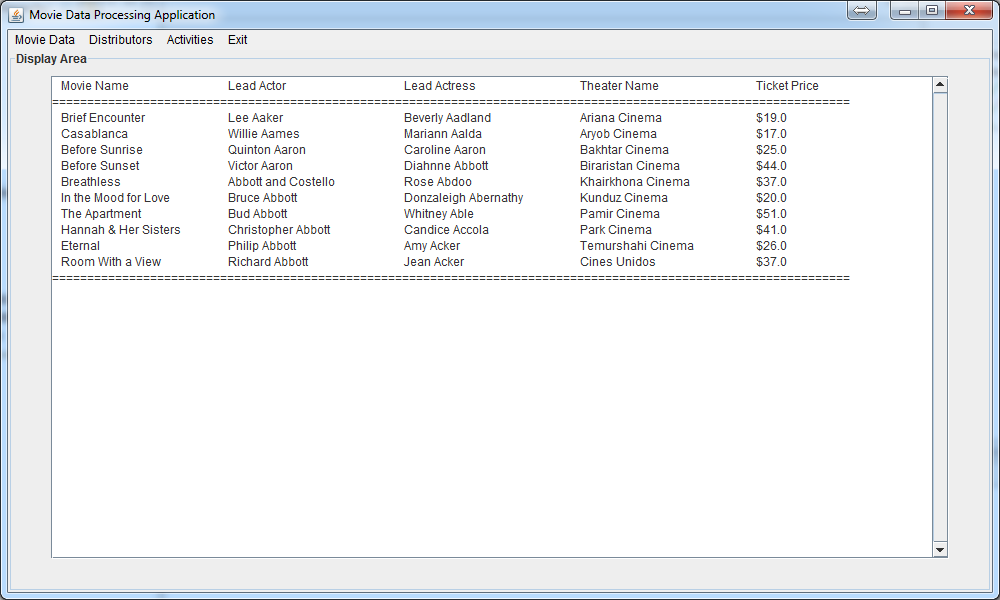
[Snapshot 3]

Below snapshot 4 describe the distributor details information by clicking on detail menu item under the Distributors menu.



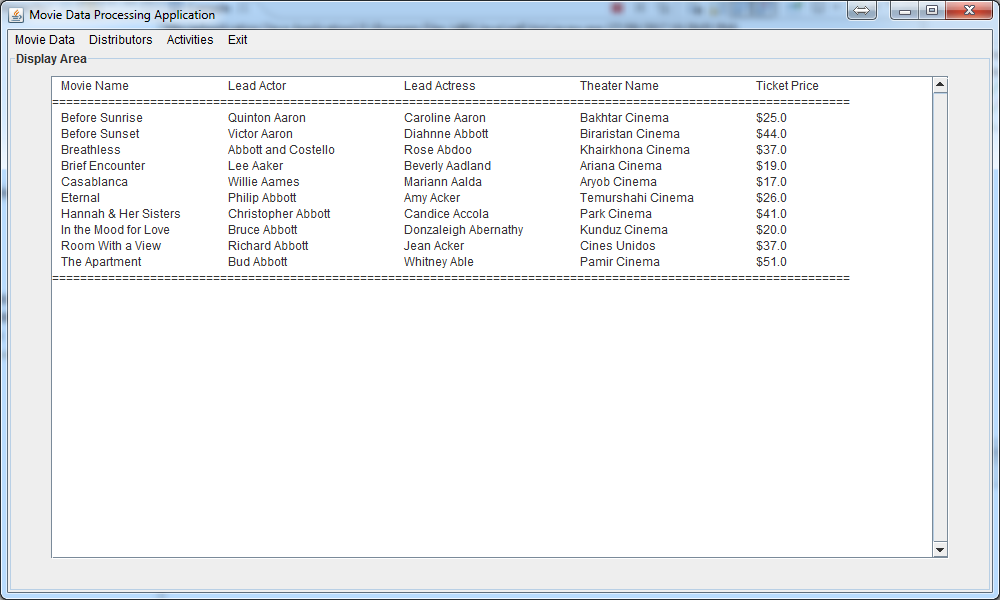
[Snapshot 4]

Below snapshot 5 screen appears while clicking on List movies menu item under the Activities menu.



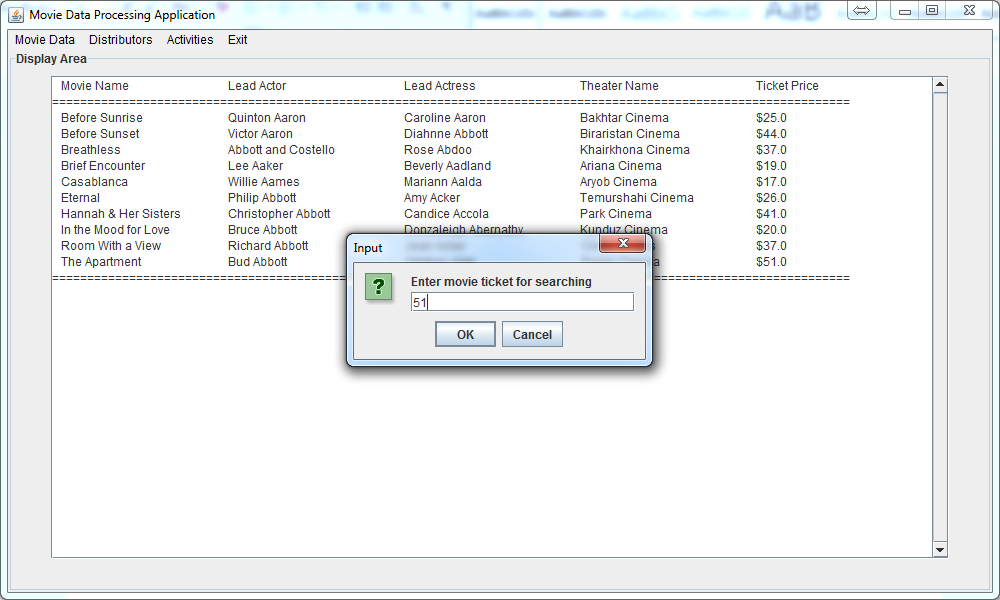
[Snapshot 5]

Below snapshot 6 shows the action for sort functionality on movie name with help of Sort by movies name menu item under Activities menu.



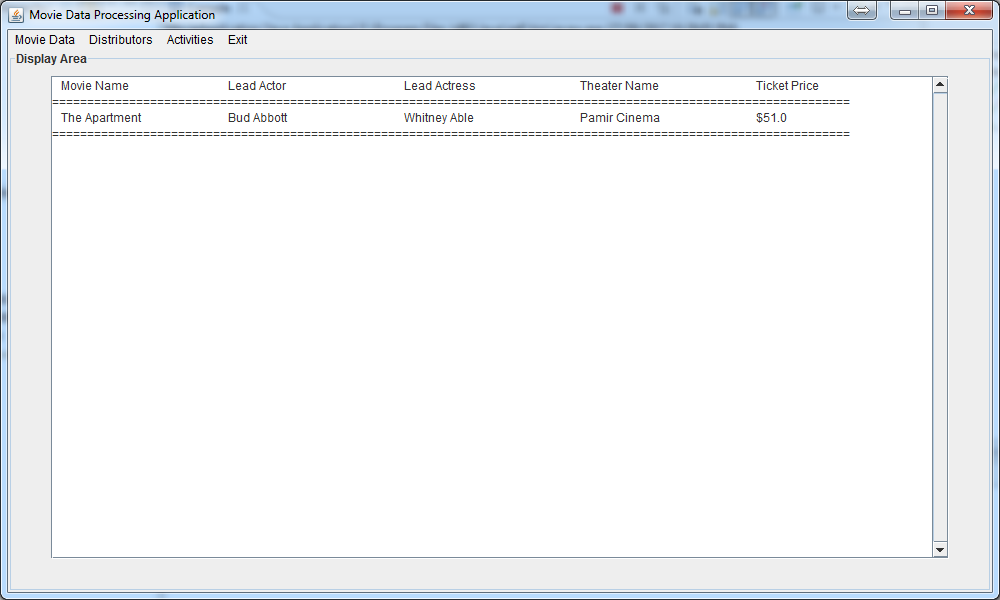
[Snapshot 6]

Below snapshot 7 gives information about search functionality of movie details by providing movie ticket price.



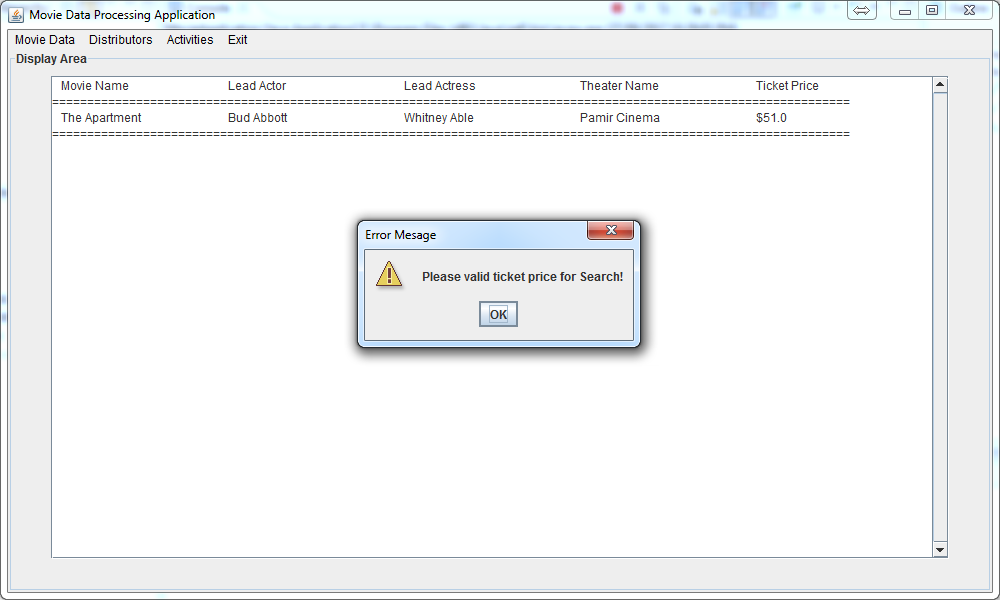
[Snapshot 7]

Below Snapshot 8 shows the result of selected ticket price 51 dollar in search window.



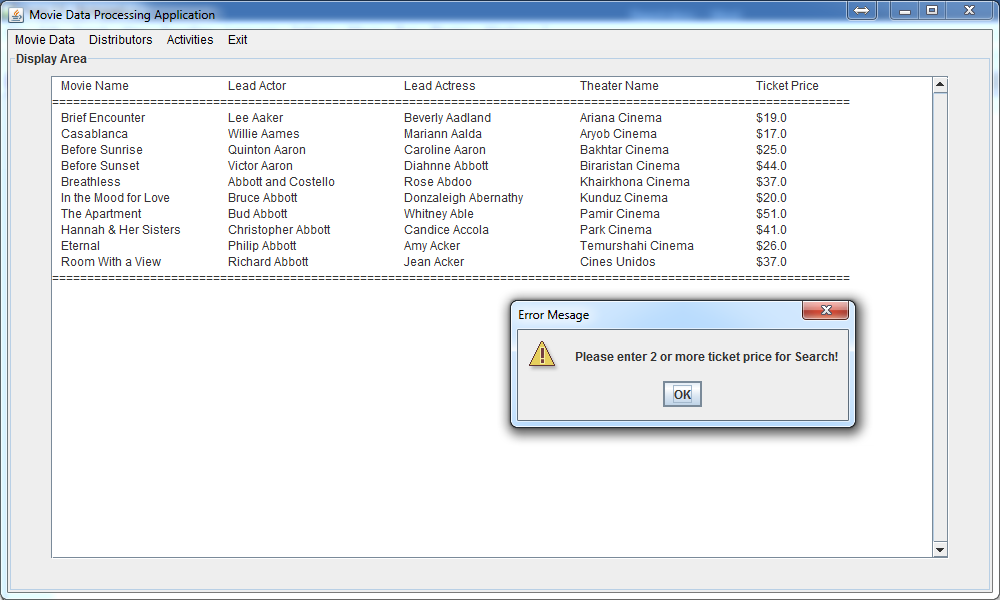
[Snapshot 8]

Below snapshot 9 shows the validation message for invalid input of ticket price. Like when someone type character instead of number value.



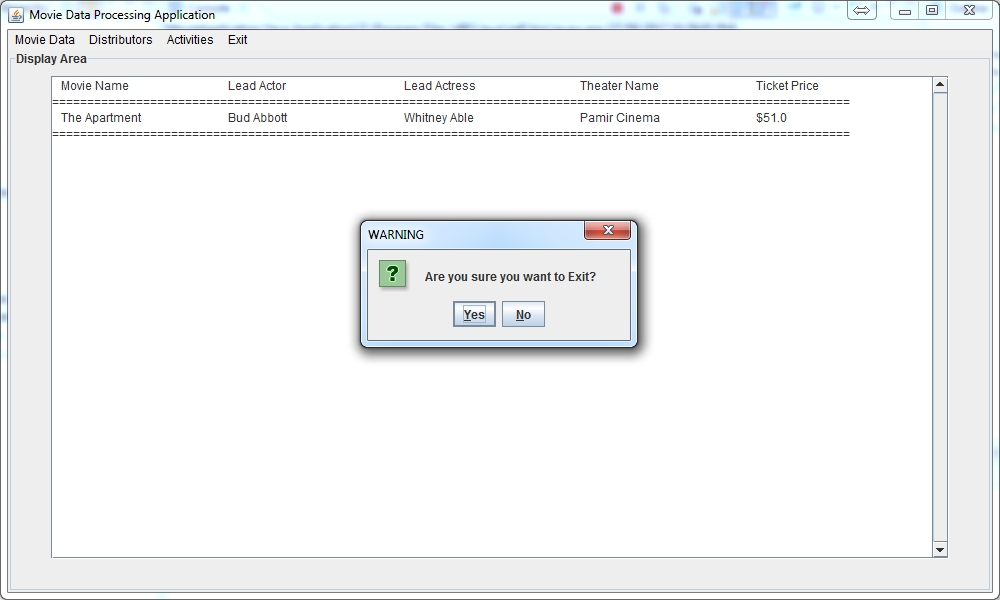
[Snapshot 9]

Below snapshot 10 shows the error message while entering less than 2 dollar value while searching by ticket price.



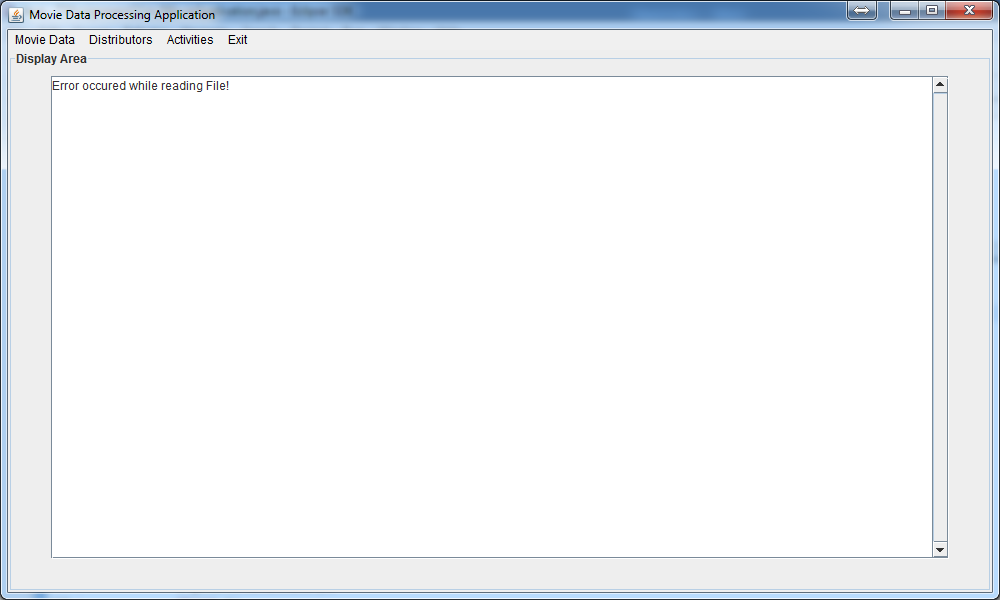
[Snapshot 10]

Below snapshot 11 shows the confirmation box while selecting exit menu item from exit menu.



[Snapshot 11]

Below snapshot 12 shows error message when something goes wrong while reading file. Demonstrating that exception handling has been done in this application.



[Snapshot 12]

# Challenges faced while working on this assignment

* Big issue to generate meaningful 1000 movies record. I achieved that with help of movies data online and using another program to combine the movies name and other details to make it 1000 records. I have not attached the program as it is not required to place it here. But I have attached movies files which I have generated using it.



* Another problem I faced is scroll bar, as described in this assignment scroll bar should be there to view all the movies in text area. But for me, it takes more than a day to figure out how can I put it there, but after long search on google I solved it. Google always rocks!
* I have not placed any reference as I have written text myself only which include algorithm selection itself for sorting and searching.