

Final Report

Database Simulation

Adam Smith

106969176

4-22-17

Summary of provided functions:

Mainmenu function reads database from file and displays the menus and sub menus for the user to traverse.

Menu functions are addrecord, deleterecord, search for records, modify records, output to file, and display with print in order tree traversal.

The searchforrecord function calls search functions which calls sort functions based on user desire.

The modify record calls a function I made inside the BStree that finds the specified node and edits it.

In the future I would streamline the search functions to take up less lines.

Status of Program:

All the functions are working properly and tested for user input error.

The database works with the small and large file.

The program compiled and ran successfully on the cse grid.

Design Document

1. Database Simulation

Adam Smith

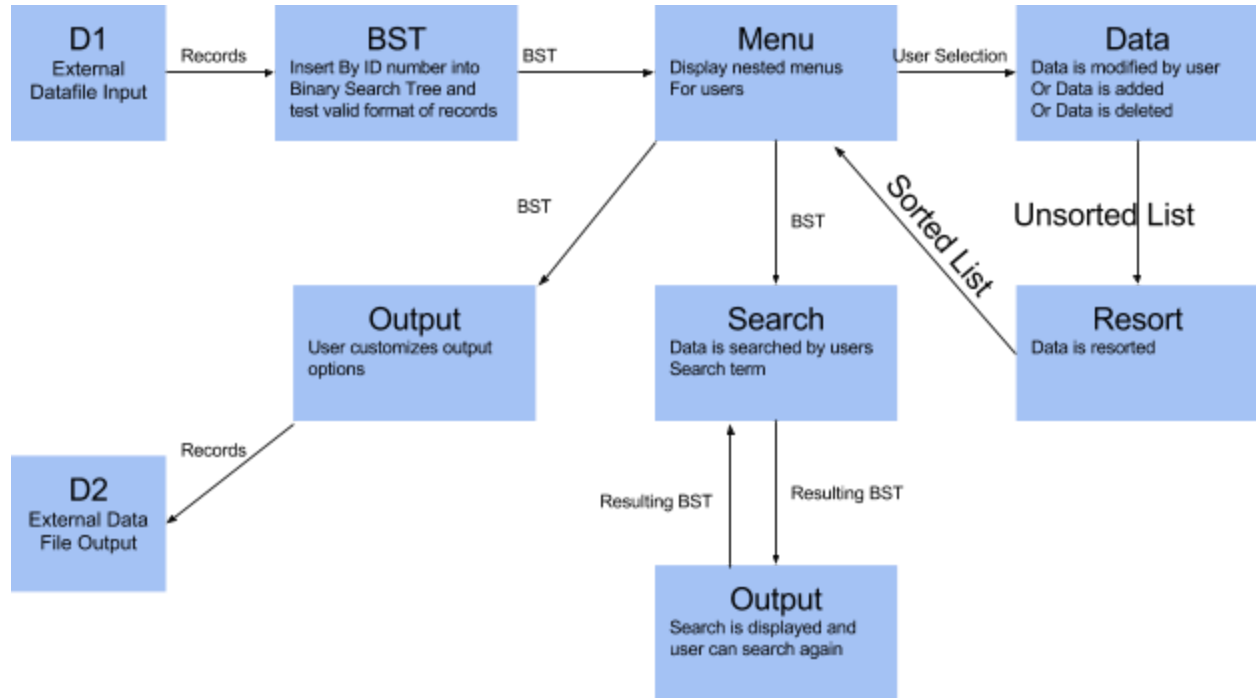
106969176

4-2-17

2. Problem Description

This program is going to be a simple database system with multiple records stored in a file. The program should be able to search and sort and modify records.

3. Overall Software Architecture



4. Input Types from file and user

Contents of each record.

1. ID Number - string , must be 9 digits
2. First name - string
3. Middle name (or initial) - string
4. Last name - string
5. Company name - string
6. Home phone - string, must be 13 chars
7. Office phone - string, must be 13 chars
8. Email - string, must contain @ symbol
9. Mobile number - string, must be 13 chars
10. Street address - string
11. City - string
12. State - string
13. Zip code - string
14. Country - string

15. List of affiliates - vector of strings per affiliate

User can search any items in each record , and search within searches.

5. Output Types from user

The user is able to customize which records to output and the parts of each record to output, as well as the way the output is sorted by.

- 16. ID Number -string
- 17. First name - string
- 18. Middle name (or initial) - string
- 19. Last name - string
- 20. Company name - string
- 21. Home phone - string, must be 13 chars
- 22. Office phone - string, must be 13 chars
- 23. Email - string, must contain @ symbol
- 24. Mobile number - string, must be 13 chars
- 25. Street address - string
- 26. City - string
- 27. State - string
- 28. Zip code - string
- 29. Country - string
- 30. List of affiliates -vector of strings per affiliate

Records are separated with |

6. Problem Solution Discussion

I plan to use a recursive algorithm to traverse the binary search tree using the id number. This breaks the tree into smaller trees until the record is found. Searching for any other part of the records will require searching the entire tree since it is not sorted by the other parts. I plan to accomplish this with an iterative algorithm to traverse vectors until the correct record is found.

7. Data Structures

I've considered arrays, vectors, stacks, and linked lists for sub containers. Arrays would not be practical for this application because the initial size is unknown and the size will be changing. Stacks would not work because we would need easy access to any element and in a stack you can only access the top element at a time. Vectors and linked lists do not have this problem. I choose vectors over linked lists because we need to access data in the middle frequently which takes longer for linked lists than for vectors.

8.

