



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Department of Computer Science
Faculty of Computing

GROUP PROJECT

ASSIGNMENT 2

LINKED LISTS

Programme	:	Bachelor of Computer Science (<i>Computer Network and Security</i>)
Subject Code	:	SECJ2013
Subject Name	:	DATA STRUCTURE AND ALGORITHMS
Session-Sem	:	2023/2024-1

Prepared by	:	1) Aidan Andrew Daniel (A22EC0036) 2) Muhammad Nabil Iman bin Mohd Fazli (A22EC0046) 3) Muhammad Haziq bin Saamsol (A22EC0214)
Section	:	03
Group	:	GOATS
Lecturer	:	Mrs Lizawati binti Mi Yusuf

Table of Content

Objectives.....	3
Synopsis.....	3
Flow Chart.....	4
Linked list Implementation.....	5
Add Node Functionality.....	5
1. Add node at the front.....	5
2. Add node at the middle (specific position).....	5
3. Add node at the back.....	6
Delete Node Functionality.....	6
Head Node Deletion (Unlinking):.....	6
Middle or Last Node Deletion (Unlinking):.....	6
Final touches (& Deletion):.....	6
Find Node Functionality.....	7
Sorting Node Functionality.....	7

Objectives

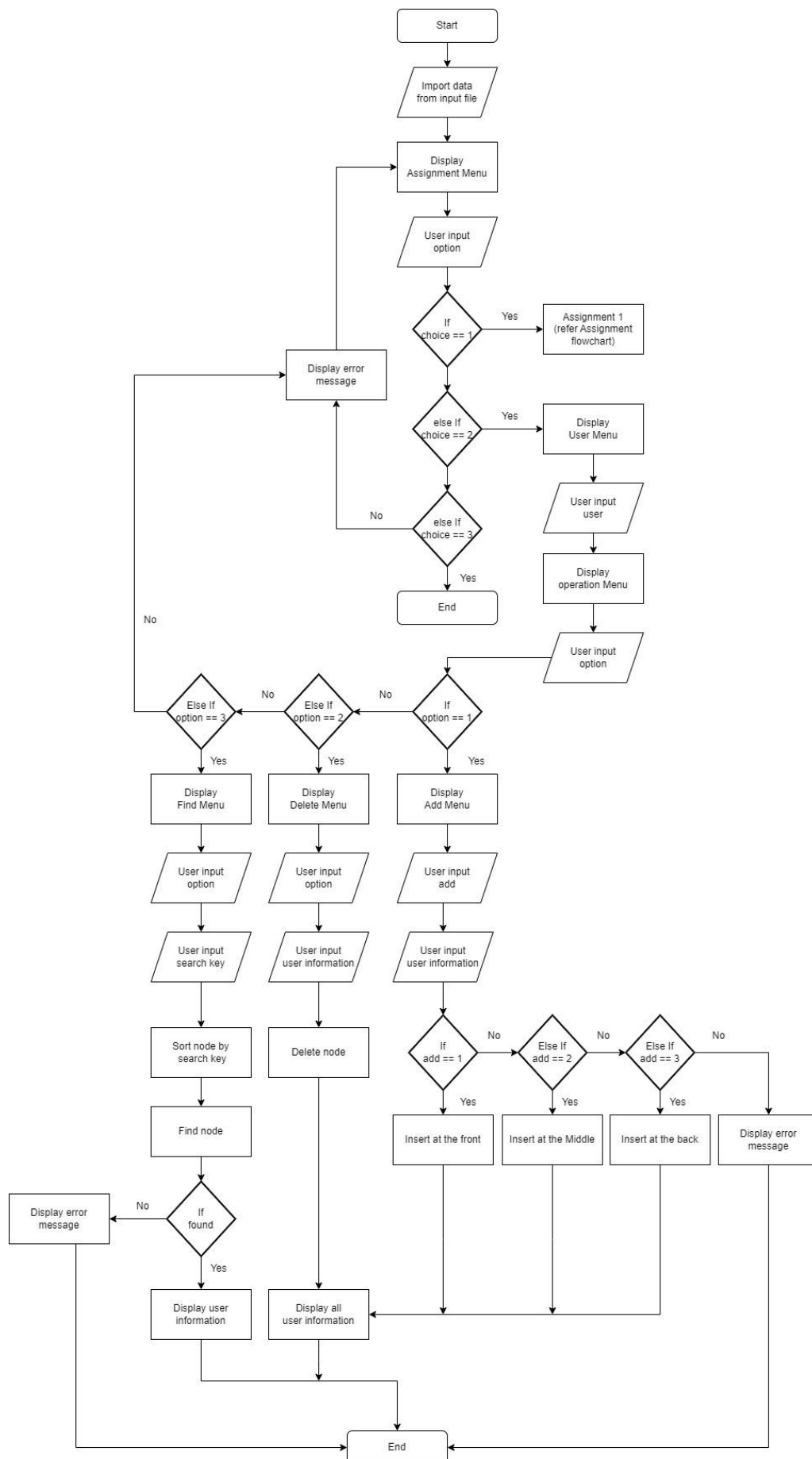
- To provide an easy management system for recording patients and staff of hospitals.
- To provide a sorting and searching function in the system.
- To provide efficiency and effectiveness with high accuracy in sorting and searching by implementing a data structure algorithm.

Synopsis

A hospital management system is a system which develops to provide efficient management of large data with high accuracy. We implemented various data structure algorithms in this system which give high performance. This system allowed users to sort in ascending and descending and also to search by name or IC or ID. Before the search process the system will sort the data according to the key the user chooses. So this makes the searching faster. So in this system we have divided the program into two parts : Assignment 1 and Assignment 2.

In Assignment 1, we implement the sorting and searching algorithm by using an array while in Assignment 2, we implement the algorithm using a linked list which has unlimited size. Arrays have limited size while linked lists have unlimited size which we can insert data. Linked list is suitable for large data while array is suitable for small size data. At both assignments, we implemented encapsulation concepts and data hiding concepts , which give security, flexibility and modularity.

Flow Chart



Linked list Implementation.

Linked list used in the Hospital Management System is doubly linked list which node is the class Patient or class Doctor, which have added a next pointer and prev pointer as private attribute for each class to implement data hiding which enhance the security of the data.

Add Node Functionality

In this assignment 2 program, it has 3 add functions which add nodes into the list but each function will add a specific position. Since we use doubly linked lists it is easier to add nodes into the list. Additionally, the attributes of the patient class and doctor class have the same number of attributes and data type, so in each function we can add either patient or doctor. For all the functions it will check if the list is empty or not. If empty, the list head pointer will point to the new node that the user wants to insert. Before the add function is called the user needs to choose user type (patient/ doctor) then insert the user information and only if the user wants to insert at the middle user will need to give the position.

Type of Add function :

1. Add node at the front

To insert from the front, the new node next pointer needs to point to the head of the list first and then the head prev pointer needs to point to the new node. Lastly, the head is changed to the new node to keep the head at front.

2. Add node at the middle (specific position)

To insert from the middle, the user will input the position where the user wants the node to be in the list. So, in the function we will loop through the list until we get to the position or until the node next pointer points to the null.

If the position is greater than the number of nodes in the list, an error message will be displayed.

If the position is equal to 1, the addFront function will be called to insert at the front.

Also if the position is equal to the number of nodes in the list, the addBack function will be called to insert at the back.

3. Add node at the back

To insert from the back, it is almost the same as insert at the front but we need to traverse the list until the node next pointer points to the null. So, the new node prev pointer will point to the current node that we transverse, the current node next pointer will point to the new node.

Delete Node Functionality

The deleteNode function allows the user to remove a node, either a Patient or a Doctor, based on their IDs. Each are specified in the code with either 'p' for patient or 'd' for doctor. The function will first scan through the respective linked list (patients or doctors) to locate the node with the matching the key, ID.

Head Node Deletion (Unlinking):

If the node to be deleted is the first node or the head of the list, the head pointer is updated to point to the next node. If the list becomes empty, the head is set to NULL.

Middle or Last Node Deletion (Unlinking):

If the node is in the middle or at the end of the list, the function updates the links of the surrounding nodes to allow for the node to be deleted. This is done by adjusting the next and prev pointers of nodes.

Final touches (& Deletion):

Once the node is successfully unlinked from the list, the function deletes the node. The function returns an integer (1 or 0) indicating the success of the deletion operation.

Find Node Functionality

The findNode function allows the user to find a node based on the search key entered and type of user. The search key uses the two attributes which is name and IC/ID (Based on the type of user, and the user indicates 1 for patient and 2 for doctor).

The function accepts two parameters which are searchKey and user. The function then checks the linked list of patients/doctors, comparing the searchKey input with the names or IC/IDs stored in the list.

If found: The function displays the data of the patients/doctors

If not found: The function displays an error message saying the key is not matched with any data in the list.

Sorting Node Functionality

Sorting Linked list function will implement the bubble sort algorithm, in which users need to choose whether to sort by name or IC/ID for the patient and doctor. This function will be implemented before the findnode function is called. So, When the list is sorted it is more faster to find the node.

