Turnitin Originality Report

Processed on: 31-Dec-2023 21:25 +08
ID: 2265819142
Word Count: 1359
Submitted: 1

ASSIGNMENT2_CODERA By Joseph Lau Yeo KAI

Similarity Index	Similarity by Source	
5%	Internet Sources:	2%
	Publications:	1%
	Student Papers:	4%

2% match (student papers from 30-Jan-2023)
Submitted to Southern New Hampshire University - Continuing Education on 2023-01-30

1% match (student papers from 18-Aug-2023)
Submitted to The University of Texas at Arlington on 2023-08-18

1% match (Internet from 20-Jun-2019)
https://www.scribd.com/document/351511335/Recommendation-on-Graphs

1% match (student papers from 30-Apr-2012)
Submitted to University of Arizona on 2012-04-30

1% match (student papers from 13-Aug-2023)
Submitted to Asia Pacific University College of Technology and Innovation (UCTI) on 2023-08-13

Department of Computer Science Faculty of Computing Assignment 2 (Hospital Management System) Programme Subject Code Subject Name Session-Sem : Bachelor of Computer Science (Data Engineering) : SECJ2013 : Data Structures and Algorithms : 2023/2024-1 Prepared by : 1) Neo Zheng Weng (A22EC0093) 2) Joseph Lau Yeo Kai (A22EC0055) 3) Lee Yik Hong (A21BE0376) Section : 02 Group : Codera Lecturer : Ms.Lizawati binti Mi Yusuf Github link : https://github.com/jjn7702/SECJ2013- DSA/tree/main/Submission/sec02/Codera/Assignment2 Objectives ? Develop a hospital management system to manage the patient record. ? Implement linked list data structure and algorithm to build this hospital mangament system. ? Allow healthcare workers to add or delete patients, search or sort patient record, and view patient medical history. Synopsis The hospital management system has been developed to manage the patient records in a hospital effectively. This system will act as a tool to help the healthcare professionals to monitor patient information, contacts, and medical history in a better way. This system will allow those healthcare workers to add, delete, search, sort , and display all the patient records. Linked list data structure is used with some operations to perform the functions of this proposed system because linked list is better than array in the modification without moving larger amount of data. The patient record will include with patient name, identity card number, age, contact number, diagnosed disease, and diagnosed date. Firstly, add a new patient function will provide options to insert the data at the beginning, at any position of the middle, or at the end of the patient record. Next, delete a patient function will provide options to delete the first data, any data in the middle, or the last data of patient record. Moreover, search patient function will allow the healthcare staff to search patient medical record based on name, identity card number, or age. Furthermore, sort patient function will allow the healthcare staff to sort the patient list in assending order based on name, age, or diagnosed date. Lastly, the complete patient medical record can be displayed to the healthcare workers. The system uses merge sort algorithm with linked list structure in assending order because it has stable and predictable performance. This is because merge sort has an O(n log n) time complexity, where "n" is the number of elements to be sorted. In a nutshell, the Hospital Management System helps minimize the workload for healthcare workers in replacing manage patient records manually. The system contributes to efficient and accurate healthcare services. Class Diagram The relationships in the class diagram are: ? Patient and Node: Composition, as each Node contains an instance of Patient. ? List and Node: Composition, as List contains a collection of Node instances to construct the linked list. ? List and Patient: Association, as List uses instances of Patient through Node to manage patient information in the linked list. Pseudocode 1. Start 2. Define a Patient class with attributes such as name, IC (identity card number), age, gender, contact number, disease, and diagnosis date. 3. Display a loading page and the main page with options: 1. Add a new patient 2. Delete a patient 3. Search for a patient 4. Sort patients 5. Display patient list 6. Logout 4. Get user choice. 5. If opt is 1: 5.1 Display <> [1] Add at the beginning [2] Add at the middle(any position) [3] Add at the end [4] Back 5.2 If choice is 1: 5.2.1 Input the patient info at the beginning of the table. 5.3 If choice is 2: 5.3.1 Input the patient info at the middle of the table. 5.4 If choice is 3: 5.4.1 Input the patient info at the end of the table. 5.5 If choice is 4, the system back to the main page. 6. If opt is 2: 6.1 Display <> [1] Delete the first patient list [2] Delete the middle(any position)patient list [3] Delete the last patient list [4] Back 6.2 If choice is 1: 6.2.1 Delete the patient info at the beginning of the table. 6.3 If choice is 2: 6.3.1 Input the patient info at the middle of the table. 6.4 If choice is 3: 6.4.1 Input the patient info at the end of the table. 6.5 If choice is 4, the system back to the main page. 7. If opt is 3: 7.1 Display <

> [1] Search by name [2] Search by IC [3] Search by age [4] Back 7.2 If choice is 1: 7.2.1 Search the patient by name. 7.3 If choice is 2: 7.3.1 Search the patient by IC. 7.4 If choice is 3: 7.4.1 Search the patient by age. 7.5 If choice is 4, the system back to the main page. 8. If opt is 4: 8.1 Display <> [1] Sort by name [2] Sort by age [3] Sort by date [4] Back 8.2 If choice is 1: 8.2.1 Sort the patient table by name. 8.3 If choice is 2: 8.3.1 Sort the patient by age. 8.4 If choice is 3: 8.4.1 Sort the patient by date. 8.5 If choice is 4, the system back to the main page. 9. If opt is 5: 9.1 Display <> [1] Display current list [2] Back 9.2 If choice is 1: 9.2.1 Display the patients list. 9.3 If choice is 2, the system back to the main page. 10. If opt is 6, system logged out. 11. If opt is not 1-6, display an error message. 12. Repeat steps 4-9 until the user selects option 6: Logout. Description on Linked List Operation Adding a Patient: The insert, insertEnd, and insertMiddle functions are used to add a new patient to the list. They all perform same functions which is add new patient records at different position. The insert function will add the node at the beginning of the list. The next pointer of the new node is set to the current head of the list, then the head is updated to the new node. For the insertEnd function, the node is added at the end of the list. This is done by travelling to the last node and next pointer is set to the new node. The insertMiddle function can help to add the node at a middle position. The function move to the desired location and update next pointers of the adjacent nodes to insert the new node. Deleting a Patient: The removeFront, removeEnd, and removeMiddle functions are used to remove a patient from the list. The removeFront function deletes the first patient info in the list. The head is the second node and therefore the system deletes the first node. The head is set to the set The removeEnd function will help to delete the last node in the list. The last second node is set to NULL, then the last node is deleted. The removeMiddle function deletes a node at a position. The node before the node want to be deleted position will be updated for its upgoing pointer, in order to delete that node. Searching for a Patient: The findNodeName, findNodeIC, and findNodeAge functions are used to find a patient in the list. Each function will go through the list based on the input. If input match the list, then the system returns that patient info. If no patient found, the system returns NULL. Sorting Patients: The MergeSort, MergeSortDate, and MergeSortAge functions are used to sort the list of patients. Merge Sort is a Divide and Conquer algorithm that divides the input array into two parts, sorts the two halves independently, and then merges them. The MergeSort function is a recursive function that continuously divides an array until it reaches the individual elements (base case: if first < last). Once the elements are created individually, they are recombined in the sorted order using the merge function. The sorting is done based on three types sorting, name, date and age of the patient. Displaying the Patient List: The dispList function is used to display the current list of patients.