Turnitin Originality Report

Processed on: 31-Dec-2023 19:45 +08
ID: 2265806048
Word Count: 1131
Submitted: 2

Assignment 2 DSA TechTurtles
By Azhar Yazid

Similarity Index	Similarity by Source	
6%	Internet Sources:	1%
	Publications:	5%
	Student Papers:	2%

2% match (student papers from 21-Dec-2012)
Submitted to iGroup on 2012-12-21

1% match (Kavita Srivastava. "chapter 3 Linear Data Structures and Their Applications", IGI Global, 2023)
Kavita Srivastava. "chapter 3 Linear Data Structures and Their Applications", IGI Global, 2023

1% match (D. Malhotra, N. Malhotra. "Chapter 4: Linked Lists", Walter de Gruyter GmbH, 2020)
D. Malhotra, N. Malhotra. "Chapter 4: Linked Lists", Walter de Gruyter GmbH, 2020

1% match (Internet from 07-Oct-2023)
https://1library.net/document/q7wml8wo-data-structures-topic.html

Objective The Courier Service System aims to: To utilize a singly linked list to manage packages, the administrator can add new packages with information like the tracking number, address, sender, recipient, category, and delivery status, among other tasks, using the system. To organize and manipulate the parcels as it provides the ability to add or remove items at any point, locate parcels by tracking number, and sort the list according to various parameters (sender name, address, tracking number, or beginning, end, or specific position). To manage arranging methods using sorting algorithms such as Bubble Sort, Selection Sort, and Insertion Sort and offers a menu-driven interface for users to engage with the parcel management capabilities. To develop a user-friendly and effective application for tracking and managing packages in a courier business. Synopsis First and foremost, the system allows the administrator to add a new parcel node whether at the beginning of the linked list, at the end of the linked list or add any specific position in the linked list. Secondly, administrators can delete existing parcel nodes the same as the adding process whether at the beginning of the linked list, at the end of the linked list or add any specific position in the linked list. Next, administrators can also find a parcel in the linked list by implementing a binary search algorithm. They can get the details of a particular parcel by entering tracking no. of the parcel. Besides, the system gives an option to sort the linked list of parcel nodes. There are 3 types of sorting options, sort by tracking number which implement bubble sort algorithm, sort by address which implement selection sort and sort by sender name which implement insertion sort algorithm. All of them come with option ascending and descending order. In addition, the administrator can view all lists of parcels with its details such as tracking number, address, status

and delivery option. Design (class design presented in a class diagram and/or algorithm design illustrated in pseudocode or a flow chart) Class Diagram Pseudocode Start The system reads the file data from ParcelData2.txt. If there is an error in the opening file, the system will terminate. The system will prompt the user to input a choice number between 1 to 6. Case 1: If user selects case 1, the user needs to enter a new parcel, the detail contains tracking number, address, sender's name, receiver's name, shipping option, and status delivery. If case 1, the parcel will be added at the beginning of the list. Else if case 2, the parcel will be added at the end. Else if case 3, the parcel will be added at a specific position. Else, the choice made by the user is invalid. Case 2 : If user enters case 2,the user will be prompted to delete a parcel. If case 1, the parcel will be added at the beginning of the list. Else if case 2, the parcel will be added at the end. Else if case 3, the parcel will be added at a specific position. Else, the choice made by the user is invalid. Case 3 : If the user enters choice 3, the user needs to enter an existing parcel tracking number to find the parcel using searchKey. If the user enters the wrong tracking number,the system will display "Parcel not found". Case 4 : If the user enters choice 4, the system will provide 3 sorting options: If the user selects case 1, the system will sort the parcel based on tracking number using bubble sort. The user will be given options to sort in ascending or descending order. If the user selects case 2, the system will sort the parcel based on address using selection sort. the user will be given options to sort in ascending or descending order. If the user selects case 3, the system will sort the parcel based on the sender's name using insertion sort. The user will be given options to sort in ascending or descending order. Else the system will display invalid sorting options. If the user selects case 5,the system will display all the parcels along with its details. If the user select case 6, the user will exit the program/system. Else the system will display "Invalid choice. Please enter a number between 1 and 6". End Description of how to implement data structure operations: linked lists. In this program, we apply a singly linked list for class Node and ParcelList. In the Node class, it defines a node in the linked list. Every node has a pointer to a Parcel class object 'data' and a pointer to the next node in the list assigned as 'next'. Then, the ParcelList class also uses a singly linked list in its class. It has a pointer to the head of the list 'head' and maintains a count of the number of nodes in the list 'count'. The implementation of linked lists is used for adding a parcel either at the beginning, at the end, or at a specific position in the list. Also, the same case can be said for deleting a parcel from the list, either deleting at the beginning, at the end, or at a specific position. Not to mention, in the Node and ParcelList classes, we have implemented Node insertion, deletion, find using searchkey and print. The Node Class applies encapsulation as the class encapsulates the individual elements of the linked list. Every node contains data, which is a Parcel class object and a reference to the next code. This encapsulation will assist in the elements' organization and managing process. The ParcelList class also applied encapsulation as it encapsulates the entire linked list. The encapsulation provides a clear interface for interacting with the linked list and veils the implementation details. The node insertion provides flexibility, meaning the ability to insert at any position, whether inserting a node at the beginning, end or at a specific position. Also, efficiency is another benefit as it allows different insertion strategies depending on the use case. Next, node deletion allows dynamic memory management. When a node is removed, it is important to free the associated memory, preventing memory leaks. As the same for node insertion, it is also efficient as it allows different deletion strategies. Moving on, the findNode used for finding a node are applied by using searchkey. In our code, we utilized findNode function to provide a way to search for a specific tracking number of a parcel. Last but not least, we implemented the displayAllNodes function to serve a human-readable representation of the linked list, so user's understanding of the state of the

system is improved, at the same time indirectly helps the user for code debugging, for example identifying issues and verify the operability of functions.