Merchant Monetary System

Data Structure

Project Supervisor

Mr. Samyan Qayyum Wahla

Group ID ($G11$)

Project Member

Syed Hashir	2021-CS-1
Kabir Ahmed	2021-CS-4
M. Hamad Hassan	2021-CS-33

Department of Computer Science
University of Engineering and Technology, Lahore
Pakistan

Data Strucuture

The following section shows the reason for choosing the data structure in the particular use case with a brief explanation.

0.1 Use Case 1:LogIn

Use Case ID	U01
Data Structure Used	Linked List
Time Complexity	In Worst Case: Search: O(n), Insertion: O(1), Deletion: O(n)
Space Complexity	O(n)
Pseudocode	**Search:**
	LIST-SEARCH(L,k)
	1 x=L.head
	2 while x $\neq$ NIL and x:key $\neq$ k
	3 x = x.next
	4 return x
	Insert:
	LIST-INSERT(L, x)
	1 x.next=L.head
	2 if L.head $\neq$ NIL
	3 L.head.pre = x
	4 L.head = x
	5 x.pre = NIL
	Delete:
	LIST-DELETE(L,x)
	1 if x.pre $\neq$ NIL
	2 x.pre.next=x.next
	3 else L.head D x.next
	4 if x.next $\neq$ NIL
	5 x.next.pre =x.pre

Justification for the use of data structure	In mentioned use case required a linear-dynamic data structure. Doubly LinkedList provides an efficient way to search the specific information from a large amount of data and then compare it with input information to produce the required result. It allows you to move back and forth in the list to get the required result.
Available choices	Array List,Hash Table
Comparison	The array list worst and average case time complexity is O(n). It takes contiguous memory. The hash table is best in the average case, but in the worst case time, complexity rise to O(n). It takes contiguous memory for storing the hash function value. In the average and worst case, the linked list insertion and deletion take O(1) time. In the average and worst case, it takes O(n) time for deletion. It did not require contiguous memory allocation. Array list, hash table, and linked list space complexity O(n) are the same. Array list, hash table, and linked list space complexity O(n) are the same.

Use Case ID	**Data Structure Used**	**Justification for the use of data structure**
U01	Linked List	In the U01 (LOGIN), search and compare the user from the list so when the user data is found it returns the action.
U04	Linked List	In the U04 (Account Details), Grid of the added users shown lists where all the users are stored(added)
U05	Linked List	In the U05 (Update Account), Update data of the user present in the Linked List
U06	Linked List	In the U06 (Add Product), Add the product data in the List.
U07	Linked List	In the U07 (View Product), View the product data in the Grid that are stored in the list at the back-end.
U08	Linked List	In the U08 (Update Product), Update the product data in the list where the data of the products are added.
U09	Linked List	In the U09 (Add Rider), Add the Rider data in the List. Selection of the list is because there is the ease in the deletion and search in the data of list
U10	Linked List	In the U10 (Update Rider), update the rider data. Selection of the list is to search is to ease.
U11	Queue	In the U11(Order Product), To place the order we use the mechanism of First in and First Out (first order item will be placed first)
U12	Stack	In the U12 (Email), To send the mail and view the mail (first send mail is shown in the last and the most recent one in the first)
U13	Linked List	In the U13 (Add Warehouse), Add the detail data of the warehouse in the list.

U14	Linked List	In the U14 (Detail Warehouse), Select the desired warehouse and delete the data of the warehouse and also delete the data from the list and selection of list is that to delete the warehouse other indexes of list easily manage.
U15	Linked List	In the U15 (Edit Warehouse), Select the data from the list and Edit the detail data of the warehouse in the list.
U16	Linked List	In the U16 (Order Status), Data is selected and data of the desired Order is updated in the list.
U17	BST	In the U17 (Route Finder), Routes are found according to the points (nodes) so selection of BST is due to the ease of the data finding.
U18	Linked List	In the U18 (Add Shopkeeper), Add the shopkeeper data in the list because there is an ease for the deletion and searching.
U19	Linked List	In the U19 (Add Payment), payment of the specific shopkeeper is added on the list to search and edit the details in the list.
U20	Linked List	In the U20 (Add Expenses Amount), Add the Expenses data in the List. Because there is an ease to update the specific data in the list and search or delete it in list.
U21	Linked List	In the U21 (Create Account), Linked list is used to add user.