Express Within Regions

**Index Page**

|  |  |
| --- | --- |
| **Problem Introduction** | **Page No 1** |

* Problem Introduction……………………………………………………………. Page 2
* Operations included in the application………………………………………….. Page 3
* Choice of Data Structures……………………………………………………….. Page 4
* Code …………………………………………………………………………….. Page 5
* limitations and Future Scope……………………………………………………. Page 6
* Subject Importance……………………………………………………………… Page 7

|  |  |
| --- | --- |
| **Problem Introduction** | **Page No 2** |

In nowadays, we have seen multiple problems. From them we tried to focus on the issues in Uzbekistan. One of the bigger problems is an Express system that is needed among the regions in Uzbekistan. We all know that there already exist express packaging systems at other countries and they are already have developed a lot in their countries. But when it comes to Uzbekistan we need to improve express packaging system between our regions. For much faster, safest and trustworthy delivery.

As we know we can’t start creating a program without gathering ideas that this problem holds. So we have researched the problems and saw the short comings of the express system, the first is a being trustworthy, then safest, price, and also how fast it will arrive. From those problems we have tried to find the possible ways to implement solutions to make the express system worthy of their anticipation.

|  |  |
| --- | --- |
| **Operations included in the application** | **Page No 3** |

In the application that we have created some operations which mainly includes sorting techniques, data base using arrays, comparing strings and other methods.

For sorting techniques, we used bubble sorting to sort out the regions and the users to be seen.

For data base we used arrays to save the data beforehand and also to add data information too while in the program. From this we can see the users and regions that already exist beforehand that is implemented into the source.

We used an operation to compare strings and other methods similar to this to see the correct information was inputted correctly. We can compare it with the information stored into the system and see if it matches. When it does we can successfully login in to the system as the customer.

We have also included and operation to calculate how much the cost would be for the delivery to be made. To do this we find the weight of the product and multiply it according to the type of value it is going to be. This value will vary according to the situation.

|  |  |
| --- | --- |
| **Choice of Data Structures** | **Page No 4** |

In the program we have chosen to use the following data structure:

Bubble sorting technique – we used this data structure to sort the available people’s last name, and see them in a sorted order. This sorts by with the simplest algorithm that works repeatedly swapping the adjacent elements if they are in wrong order. With the help of this we sorted the elements in correct orders.

Linked list – We implemented this to the program be represented by a pointer to the first node of the linked list. While the first node is the head, and when the link is empty the value is NULL. In short it can be said that we used it to see a collection of objects “Nodes” to be stored randomly into the memory.

Array – We used arrays to store information into them and to make it easier to access the data that is stored inside the array. The collection of data inside the array we used for were string, integers, and characters.

Queue – We used queue in resources where scheduling was required. With the help of the two pointers rear and front ends, we used them to insert and remove an element from and to the queue respectively.

|  |  |
| --- | --- |
| **Code** | **Page No 5** |

This is our application’s code

#pragma once

#include <iostream>#include <string>#define MAX 100using namespace std;

void sorting(char arr[][MAX], int n)

{ char temp[MAX];

for (int j = 0; j < n - 1; j++) // Sorting strings using bubble sort

{ for (int i = j + 1; i < n; i++)

{ if (strcmp(arr[j], arr[i]) > 0)

{

strcpy\_s(temp, arr[j]);

strcpy\_s(arr[j], arr[i]);

strcpy\_s(arr[i], temp);

} } }}

class Order

{

public:

int id;

string from\_reg;

string to\_reg;

double massInKg;

Order(int i, string from, string to, double mass)

{

id = i;

from\_reg = from;

to\_reg = to;

massInKg = mass; }

int getId() { return id; }

int getmass() { return massInKg; }

string getFrom() { return from\_reg; } string getTo() { return to\_reg; }};

int order()

{ int choice;

char from\_region[][MAX] = { "Samarkand", "Tashkent", "Republic of Karakalpakistan",

"Syrdarya","Surkhandarya","Khorezm","Namangan","Fergana","Kashkadarya","Navoi","Djizzak",

"Andijan","Bukhara" };

char to\_region[][MAX] = { "Samarkand", "Tashkent", "Republic of Karakalpakistan",

"Syrdarya","Surkhandarya","Khorezm","Namangan","Fergana","Kashkadarya","Navoi","Djizzak",

"Andijan","Bukhara" };

int n = sizeof(from\_region) / sizeof(from\_region[0]);

//sorting regions

sorting(from\_region, n);

sorting(to\_region, n);

for (int i = 0; i < n; i++)

cout << i+1 << "Regions " << from\_region[i] << endl;

start:

cout << "Choose from where you want to dostavka\n";

int from;

cin >> from;

cout << "Choose to where you want to dostavka\n";

int to;

cin >> to;

cout << "What is weight of your good?";

double weight;

cin >> weight;

int id;

cout << "What is the recieved id of your good?";

cin >> id; Order order(id, from\_region[from-1], from\_region[to-1], weight); cout << "Order's id " << order.getId()<<endl; cout << "Order will be sent from " << order.getFrom() << endl;

cout << "Order will be recive to " << order.getTo() << endl;

cout << "Order's mass " << order.getmass() << endl;

int accept;

start1:

cout << "Do you accept?" << endl;

cout << "1) yes\n";

cout << "2 no\n";

cin >> accept;

if (accept == 1)

return order.getmass();

if (accept == 2)

goto start;

else {

cout << "Plese input the correct one\n";

goto start1;

}

system("pause");

return id;

}

class Queue

{

public:

int front, rear, size;

unsigned capacity;

int \*array;

};

Queue \*createQueue(unsigned capacity)

{

Queue \*queue = new Queue();

queue->capacity = capacity;

queue->front = queue->size = 0;

queue->rear = capacity - 1;

queue->array = new int[(

queue->capacity \* sizeof(int))];

return queue;

}

int isFull(Queue \*queue)

{

return (queue->size == queue->capacity);

}

int isEmpty(Queue \*queue)

{

return (queue->size == 0);

}

int enqueue(Queue \*queue, int id)

{

if (isFull(queue))

{

for (int i = 0; i < queue->capacity; i++)

{

int item = queue->array[queue->front];

queue->front = (queue->front + 1) % queue->capacity;

queue->size = queue->size - 1;

}

cout << "Dequeueing the queue" << endl;

}

queue->rear = (queue->rear + 1) % queue->capacity;

queue->array[queue->rear] = id;

queue->size = queue->size + 1;

cout <<"The product with id:"<< id << " enqueued to queue\n";

system("pause");

return id;

}

int dequeue(Queue \*queue)

{

if (isEmpty(queue))

return INT\_MIN;

int item = queue->array[queue->front];

queue->front = (queue->front + 1) % queue->capacity;

queue->size = queue->size - 1;

return item;

}

int front(Queue \*queue)

{

if (isEmpty(queue))

return INT\_MIN;

return queue->array[queue->front];

}

int rear(Queue \*queue)

{

if (isEmpty(queue))

return INT\_MIN;

return queue->array[queue->rear];

}

int registration()

{

bool loop2 = true, loop1 = true;

int choice;

int start = 4, order = 4;

char input[MAX];

string input2;

char lastname[][MAX] = { "Johnson", "Walton", "James", "White" };

char firstname[][MAX] = { "John", "Dan", "Alex", "Mark" };

string password[MAX] = { "J12345", "D12345", "A12345", "M12345" };

string username[MAX] = { "Joker1", "Dice12", "wonder58","mg46" };

case1:

system("cls");

cout << "Please choose option\n"

<< "1. Sign in\n"

<< "2. Sign Up\n"

<< "3. View Users\n";

cin >> choice;

switch (choice)

{

case 1:

system("cls");

cout << "Sign In\n";

while (loop1)

{

cout << "Input Username: ";

cin >> input;

for (int i = 0; i < 100; i++)

{

//username correct!

if (input == username[i])

{

loop1 = false;

while (loop2)

{ cout << "Input password: ";

cin >> input;

for (int j = 0; j < 100; j++)

{

// password correct

if (input == password[i])

{

order = j;

cout << "Login sucessful!\n";

loop2 = false;

break;

}

}

// password incorrect

if (loop2)

cout << "Password incorrect!\n"; } }}

// username incorrect

if (loop1)

cout << "Username incorrect!\n"; }

break;

case 2:system("cls");

cout << "Sign Up\n";

cin.getline(firstname[start], MAX);

cout << "Input first Name: ";

cin.getline(firstname[start], MAX);

cout << "Input laste Name: ";

cin.getline(lastname[start], MAX);

cout << "Input Username: ";

getline(cin, username[start]);

cout << "Input password: ";

getline(cin, password[start]);

start++;

cout << "Signup successfully!\n";

system("pause");

goto case1;

break;

case 3:system("cls");

cout << "Users in unsorted order: " << endl;

for (int i = 0; i < start; i++)

{

cout << lastname[i] << " " << firstname[i] << endl;

}

//call bubble sorting function

sorting(lastname, start);

cout << "Users in sorted order: " << endl;

for (int i = 0; i < start; i++)

{cout << i + 1 << "th user is " << lastname[i] << endl;

} system("pause");

goto case1;break;

}

system("pause");

return 0;

}

#include<iostream>

#include"registration.h"

using namespace std;

class Node {

public:

int data;

Node\* next;

};

// This function prints contents of linked list

// starting from the given node

void printList(Node\* n)

{

while (n != NULL) {

cout << n->data << " ";

n = n->next;

}

}

int calculatePrice(int mass) { cout << "Price of your order is: ";

if (mass == 0) { cout << 10000; }

else if (mass > 0) {cout << mass \* 10000;

}

cout << "SUM" << endl;

system("pause");

return 0;

}

// Driver code

int main()

{

Node\* head = NULL;

Node\* second = NULL;

Node\* third = NULL;

Node\* forth = NULL;

// allocate 3 nodes in the heap

head = new Node();

second = new Node();

third = new Node();

forth = new Node();

Queue \*queue = createQueue(10);

head->data = registration(); // firstly user has to be registred!

head->next = second;

second->data = order();// Order details

second->next = third;

third->data =enqueue(queue,second->data); //Go to queue where it sends when the numb of orders will be equla to 10!

third->next = forth;

//forth->data = recieve(); //Go to queue where it sends when the numb of orders will be equla to 10!

forth->data = calculatePrice(second->data);

forth->next = NULL;

printList(head);

return 0;

}

|  |  |
| --- | --- |
| **Limitations and Future Scope** | **Page No 6** |

As we know not every program can be perfect. We also cannot say that our program works 100 percent efficiently, because it does have its down sides as well. The limitation that our application has is that our program cannot detect whether a person’s order is fake or not. A child can order a good to be delivered to another region, but our program wouldn’t know.

First of all, our project aims to help the citizens in Uzbekistan to safely order from the express with low cost and fast delivery to the destination. This will benefit the people that needs help to move objects from one region to another region in Uzbekistan.

With the limited time we can help some amount of people in the time frame. We can still grow with the help of our users while hearing their feedbacks and trying our best to apply them into our project. As for our application itself we can create a platform on a PC and mobile phones to have consumers easily access out application from anywhere with ease. This program will contain user friendly atmosphere which will not be complex for the users to understand it.

When the users use our project we can guarantee safe, fast, on time, and cheap delivery. This will increase the trust between us the consumers. So as mentioned above we did face limitations in our program itself. But when we go deeper we will also have some difficulties at the beginning for the users to trust our program. Because we have to build trust step by step without rushing. If there will be any changes in the future that will be needed, we will add or delete those if necessary.

In Short for this program to be success we need to work as a team, do this project without rushing, so we will have time to think about what is going and see our mistakes in the process.

|  |  |
| --- | --- |
| **Subject Importance** | **Page No 7** |

The impact of our application might not be big as it cannot effect a lot of people’s lives, but we can say for sure that this application of ours can benefit people that are in need of help when they need an express ordering application in Uzbekistan between the regions. People in Uzbekistan can easily use the application without much worries about and can trust in us when needed.

If we come to the importance that this is in computer science, we can say that the importance might not be high because these kinds of applications already do exist. But we do know that this application has not been improved a lot in Uzbekistan. However, in a sense, we can include that this application will be important in computer science in Uzbekistan.

Data structure is of high importance in computer science, since it can have lots of advantages which allows information to be stored on hard disks, provides means for management of large datasets, for design of efficient algorithms, easier processing of data, and so on. We can count most of the advantages which helps everyone. As it is hard to make changes to data structures only advanced people can do them, which you can also say it is secure in its own way as well. So from some of the benefits we can see the importance of data structure in computer science.

As we continue to study this course we can perfectly mention that our perspective of computer science and data structure will increase exponentially. From this we will learn the ability to distinguish the methods that can be used in computer science. And it will also help us to set our goal step by step, without messing up in an incorrect order. As we know that gaining new knowledge will never be useless, we can always find ways to implement it in our future or it might even help us in difficult situations as well. The knowledge we will gain from this course will help us in our future carrier when trying to be programmers. This knowledge will also help us build our fundamental levels strong which will later on will benefit us.