**LAB 4**

**22F-3858**

**TASK 4:**

**Code :**

#include <iostream>

#include <string>

using namespace std;

struct QueueNode

{

string passengerName;

string pickUpLoc;

string destination;

int waittime; // in minutes left

QueueNode\* next;

QueueNode(string name, string loc, string des, int time)

{

passengerName = name;

pickUpLoc = loc;

destination = des;

waittime = time;

next = NULL;

}

};

class DynQueue

{

QueueNode\* front;

QueueNode\* rear;

int numItems;

public:

DynQueue()

{

front = NULL;

rear = NULL;

numItems = -1;

}

bool isEmpty()

{

return (numItems == -1);

}

void enqueue(string name, string loc, string des, int time)

{

QueueNode\* newNode = new QueueNode(name, loc, des, time);

newNode->next = NULL;

if (isEmpty())

{

front = newNode;

rear = newNode;

}

else

{

rear->next = newNode;

rear = newNode;

}

numItems++;

}

bool dequeue()

{

QueueNode\* temp;

if (isEmpty())

{

cout << "The queue is empty.\n";

return false;

}

temp = front->next;

delete front;

front = temp;

numItems--;

return true;

}

void makeNull()

{

while (!isEmpty())

{

dequeue();

}

}

void display() {

if (isEmpty()) {

cout << "\nNo data in queue" << endl;

return;

}

QueueNode\* temp = front;

int count = 0;

cout << "\nPending Rides:";

do {

cout << "\n Name : " << temp->passengerName << endl << "PickUp Location :" << temp->pickUpLoc << endl << "Destination :" << temp->destination << endl << "Estimated Time wait(minutes) :" << temp->waittime;

temp = temp->next;

count++;

cout << endl;

} while (count <= numItems);

}

void findRide(string name) {

if (isEmpty()) {

cout << "\nNo data in queue" << endl;

return;

}

QueueNode\* temp = front;

int count = 0;

bool found = false;

do {

if (temp->passengerName == name) {

found = true;

cout << "\n Name : " << temp->passengerName << endl << "PickUp Location :" << temp->pickUpLoc << endl << "Destination :" << temp->destination << endl << "Estimated Time wait(minutes) :" << temp->waittime;

}

temp = temp->next;

count++;

} while (count <= numItems);

if (found == false) {

cout << "\nPassenger with having name \"" << name << "\" not found...\n";

}

}

~DynQueue()

{

makeNull();

}

};

int main()

{

DynQueue d1;

d1.enqueue("ali", "fast", "fsd", 30);

d1.enqueue("ahmad", "fast", "sgd", 20);

d1.enqueue("Aslam", "LHR", "FSD", 10);

d1.enqueue("Jameel", "LHR", "FSD", 10);

d1.findRide("ahmads");

d1.findRide("Jameel");

cout << "\n Before Deletion :" << endl;

d1.display();

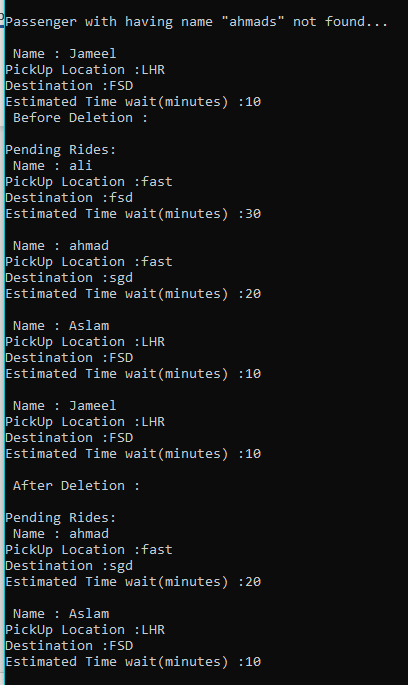
cout << "\n After Deletion :" << endl;

d1.dequeue();

d1.display();

return 0;

}

OUTPUT :  


A screen shot of a computer

Description automatically generated