***Project Title Here***

****

**CAPITAL UNIVERSITY OF SCIENCE & TECHNOLOGY**

B.S.COMPUTER SCIENCE, 3rd Semester

Section 2/3

DATA STRUCTURES

 (CS-2143)

**Course Instructor**: Shahzad Qaiser

**Submitted by:**

Afshan Liaqaut(Team Leader)

Reg # BCS201019

Rimsha Liaquat

Reg # BCS201018

1. Project Title:

Railway Reservation Management System

1. Abstract:

In the bus reservation system, first of all, according to destination passengers will be given their tickets according to their bus number. we are making a system in which tickets are given to the passengers according to the first come ,first serve rule. Each passenger will be allotted with a unique ticket number. In the case ,if any passenger will cancel his or her ticket ,then that ticket will be given to very next passenger in the Queue.

1. Problem Statement:

Tickets are sold for different routes. The tickets are first

come, first serve. A person may cancel his/her ticket and

the free seat is now available for next person in queue.

4. List of Requirements:

* Singly Linklist
* Queue
* File handling
* Enqueue
* Dequeue
* Search
* Show
* Constructors

5. Data Structures (With Justification):

We have used the data structure ***Linked List with Queue*** according to the requirement of the problem. We used singly link list instead of doubly because here in this problem we are using Queue using the rule: “ first come, first serve “. So, We don’t need to delete any data from end or some where from middle so, we don’t need ***prev*** (previous Node) to move back in the list. So that, no extra space is reserved in the memory.

6. File Handling:

//file handling.

Bus\_reservation\_system obj;

fstream obj1;

obj1.open("bus.txt",ios::out|ios::app);

if(!obj1)

{

cout<<"error in creating object."<<endl;

return 0;

}

cout<<" file open successfully."<<endl;

// permanent Data storage of all Passengers in file .

obj1<<" Data stored in file permanently is :";

obj1<<endl<<"Bus\_Ticket\_no "<<Bus\_Ticket\_no<<endl;

obj1<<"Bus\_no "<<Bus\_no<<endl;

obj1<<"Fair "<<Fair<<endl<<endl;

obj1<<endl<<"Total seats in a bus. "<<Total\_seats<<endl;

obj1<<"Seat\_no "<<Seat\_no<<endl;

obj1<<endl<<"departure\_time "<<departure\_time<<endl;

obj1<<"Booking\_date "<<Booking\_date<<endl;

obj1<<"Traveling\_date "<<Traveling\_date<<endl<<endl;

obj1<<endl<<"departure\_from "<<Departure\_from<<endl;

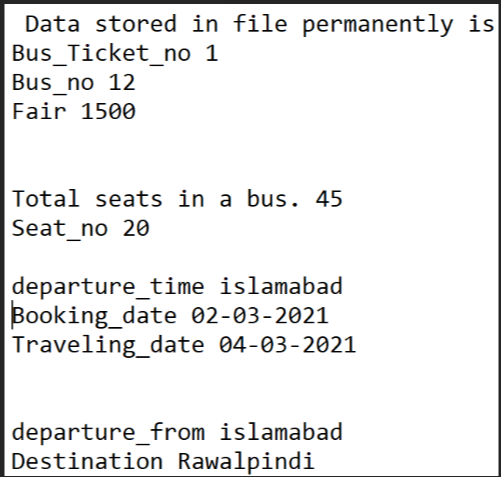
obj1<<"Destination "<<Destination<<endl;

7. UML (Basic Class Diagram)

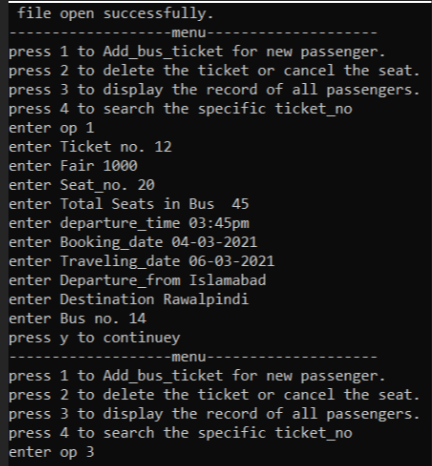
|  |
| --- |
| ***Bus\_reservation\_system*** |
| |  | | --- | | ***Node***  - bus\_ticket\_no: int  - total\_seats: int - seat\_no:int - fair: int - booking\_date: string - traveling\_date: string  - departure\_from: string - departure\_time: string-departure\_from: string - destination: string - bus\_no: string  - next: Node\* |   - rear: Node\*  - front: Node\*  -int count; |
| + Bus\_reservation\_system()  + Add\_bus\_ticket(): void  + delete\_ticket(): void  + display(): void  + search\_data():void |

8. Project Outputs:

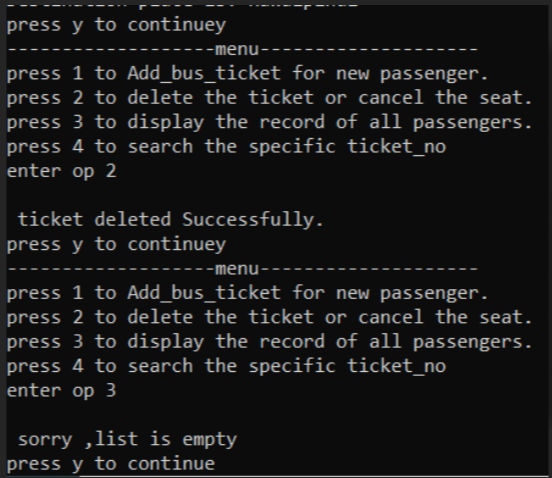
***Data stored in File:***



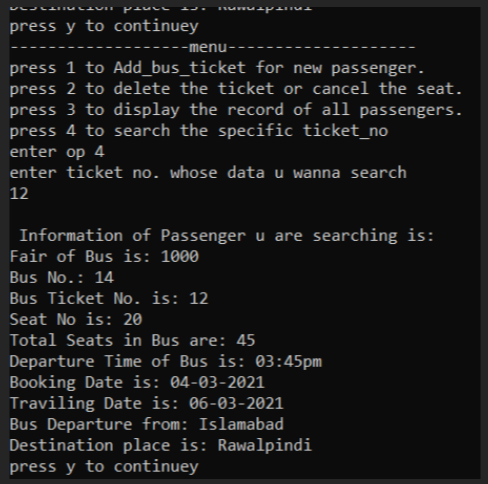
Enqueue Data:



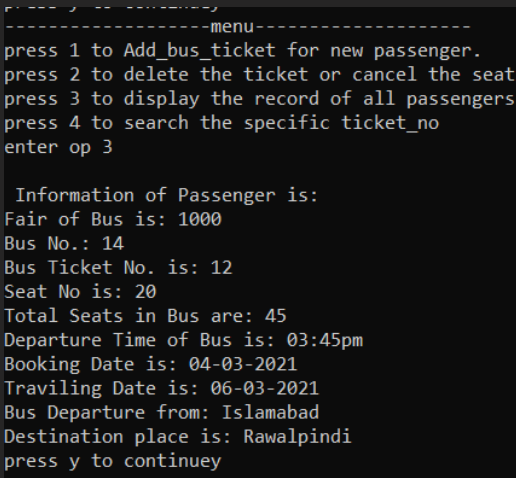
***Dequeue Data:***



***Search Data is:***



***Display Data:***



***9. Complete Code:***

#include <iostream>

#include <fstream>

using namespace std;

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

//structure of node Type

struct Node

{

int bus\_ticket\_no;

int total\_seats;

int seat\_no;

int fair;

string booking\_date;

string traveling\_date;

string departure\_time;

string departure\_from;

string destination;

string bus\_no;

Node\* next;

};

class Bus\_reservation\_system

{

private:

Node \*rear,\*front;

int count;

public:

//Set default constructor

Bus\_reservation\_system()

{

front=rear=NULL;

count=0;

}

// Reserve the seats in Bus.

void Add\_bus\_ticket(int tick\_no,int f,int S\_no,int total\_seats,string depart\_time,string b\_date,string travel\_date,string depart,string destiny,string bus\_n)

{

Node \* n=new Node;

n->fair=f;

n->bus\_no=bus\_n;

n->bus\_ticket\_no=tick\_no;

n->seat\_no=S\_no;

n->total\_seats=total\_seats;

n->departure\_time=depart\_time;

n->booking\_date=b\_date;

n->traveling\_date=travel\_date;

n->departure\_from=depart;

n->destination=destiny;

n->next= NULL;

//if Bus is empty.

if (front==NULL)

{

front=rear=n;

count++;

}

//If there already exist passengers in BUS.

else

{

rear->next=n;

n->next=NULL;

rear=n;

count++;

}

}

//If passenger cancels the Bus Ticket

void delete\_ticket()

{

// if No passenger is in Bus i.e bus is empty.

if(count==0 && front==0)

{

cout<<" \n sorry ,list is empty \n";

}

//If one passenger is in Bus.

else if(count==1)

{

front=rear=NULL;

count--;

}

//if more than one passenger is in Bus.

else

{

Node\* n=new Node;

n=front;

front=front->next;

count--;

}

}

void display()

{

// if Bus is empty

if(front==NULL && count==0)

{

cout<<"\n sorry ,list is empty "<<endl;

}

else

{

//information of all passengers in Bus.

Node \* n=new Node;

n=front;

while(n!=NULL)

{

cout<<"\n Information of Passenger is: "<<endl;

cout<<"Fair of Bus is: "<<n->fair<<endl;

cout<<"Bus No.: "<<n->bus\_no<<endl;

cout<<"Bus Ticket No. is: "<<n->bus\_ticket\_no<<endl;

cout<<"Seat No is: "<<n->seat\_no<<endl;

cout<<"Total Seats in Bus are: "<<n->total\_seats<<endl;

cout<<"Departure Time of Bus is: "<<n->departure\_time<<endl;

cout<<"Booking Date is: "<<n->booking\_date<<endl;

cout<<"Traviling Date is: "<<n->traveling\_date<<endl;

cout<<"Bus Departure from: "<<n->departure\_from<<endl;

cout<<"Destination place is: "<<n->destination<<endl;

n=n->next;

}

}

}

//if want to search a person by ticket Number.

void search\_data(int ticket\_no)

{

if(front==0 && count==0)

{

cout<<"sorry, empty"<<endl;

}

else

{

Node\* n=new Node;

n=front;

while(n!=NULL)

{

if(n->bus\_ticket\_no==ticket\_no)

{

cout<<"\n Information of Passenger u are searching is: "<<endl;

cout<<"Fair of Bus is: "<<n->fair<<endl;

cout<<"Bus No.: "<<n->bus\_no<<endl;

cout<<"Bus Ticket No. is: "<<n->bus\_ticket\_no<<endl;

cout<<"Seat No is: "<<n->seat\_no<<endl;

cout<<"Total Seats in Bus are: "<<n->total\_seats<<endl;

cout<<"Departure Time of Bus is: "<<n->departure\_time<<endl;

cout<<"Booking Date is: "<<n->booking\_date<<endl;

cout<<"Traviling Date is: "<<n->traveling\_date<<endl;

cout<<"Bus Departure from: "<<n->departure\_from<<endl;

cout<<"Destination place is: "<<n->destination<<endl;

}

n=n->next;

}

}

}

};

int main(int argc, char\*\* argv)

{

int Bus\_Ticket\_no,Fair,Seat\_no,Total\_seats;

int x; //x variable for searching.

string departure\_time;

string Booking\_date;

string Traveling\_date;

string Departure\_from;

string Destination;

string Bus\_no;

//file handling.

Bus\_reservation\_system obj;

fstream obj1;

obj1.open("bus.txt",ios::out|ios::app);

if(!obj1)

{

cout<<"error in creating object."<<endl;

return 0;

}

cout<<" file open successfully."<<endl;

char ch;

int op ;

do{

cout<<"-------------------menu--------------------"<<endl;

cout<<"press 1 to Add\_bus\_ticket for new passenger. "<<endl;

cout<<"press 2 to delete the ticket or cancel the seat. "<<endl;

cout<<"press 3 to display the record of all passengers. "<<endl;

cout<<"press 4 to search the specific ticket\_no "<<endl;

cout<<"enter op ";

cin>>op;

switch(op)

{

case 1:

//input Info. of passengers.

cout<<"enter Ticket no. " ;

cin>>Bus\_Ticket\_no;

cout<<"enter Fair ";

cin>>Fair;

cout<<"enter Seat\_no. ";

cin>>Seat\_no;

cout<<"enter Total Seats in Bus ";

cin>>Total\_seats;

cin.ignore(1000,'\n');

cout<<"enter departure\_time ";

getline(cin,departure\_time);

cout<<"enter Booking\_date ";

getline(cin,Booking\_date);

cout<<"enter Traveling\_date ";

getline(cin,Traveling\_date);

cout<<"enter Departure\_from ";

getline(cin,Departure\_from);

cout<<"enter Destination ";

getline(cin,Destination);

cout<<"enter Bus no. ";

getline(cin,Bus\_no);

// permanent Data storage of all Passengers in file .

obj1<<" Data stored in file permanently is :";

obj1<<endl<<"Bus\_Ticket\_no "<<Bus\_Ticket\_no<<endl;

obj1<<"Bus\_no "<<Bus\_no<<endl;

obj1<<"Fair "<<Fair<<endl<<endl;

obj1<<endl<<"Total seats in a bus. "<<Total\_seats<<endl;

obj1<<"Seat\_no "<<Seat\_no<<endl;

obj1<<endl<<"departure\_time "<<departure\_time<<endl;

obj1<<"Booking\_date "<<Booking\_date<<endl;

obj1<<"Traveling\_date "<<Traveling\_date<<endl<<endl;

obj1<<endl<<"departure\_from "<<Departure\_from<<endl;

obj1<<"Destination "<<Destination<<endl;

// Insert Operation

obj.Add\_bus\_ticket(Bus\_Ticket\_no,Fair,Seat\_no,Total\_seats,departure\_time,Booking\_date,Traveling\_date,Departure\_from,Destination,Bus\_no);

break;

case 2:

// Deletion Operation

obj.delete\_ticket();

cout<<"\n ticket deleted Successfully."<<endl;

break;

case 3:

// Show information

obj.display();

break;

case 4:

// Search operation

cout<<"enter ticket no. whose data u wanna search"<<endl;

cin>>x;

obj.search\_data(x);

break;

default :

// Default case

cout<<"Invalid No."<<endl;

}

cout<<"press y to continue";

cin>>ch;

}while(ch=='y');

//closing the file.

obj1.close();

return 0;

}