

National university of Computer and Emerging Sciences

Fall 2022

Data Structures and Algorithm

People's Red Bus Service

BS(CS)-3F



Group Members:

21K-4568 Muhammad Ali

21K-3175 Syed Muhammad Taqi Baquer

21K-3323 Muhammad Shaheer

Introduction:

The project is about peoples red bus service which provides service of travelling. The main reason behind doing this project is to help people find the shortest route between their source and destination.

The data structures used in this project are Graph, Priority Queue and Dijkstra Algorithm.

This report would highlight some salient features of the project.

Salient Features of Project:

1. Reservation

This code has a part which has been reserved by reservation. In this part customers Enter and then they get stored in the passenger List so that the authorities can access the data anytime. In this code, Reservation is implemented using most Convenient Data Structure named as Link List of Doubly type. Features which are part of this section is Book Ticket, Remove Passenger, Sort List of passengers using Bubble Sort on Link List, Displaying the Link List and Clearing the Link List for new shift of data.

2. Filing:

The second part has been reserved for Filing in which all the names of stop along with a Unique Stop number are stored in a File. There is one more file which shows edged connection between two stops and distance between them. Passenger List, Stops and Dataset files are attached for your reference.

3. Graph:

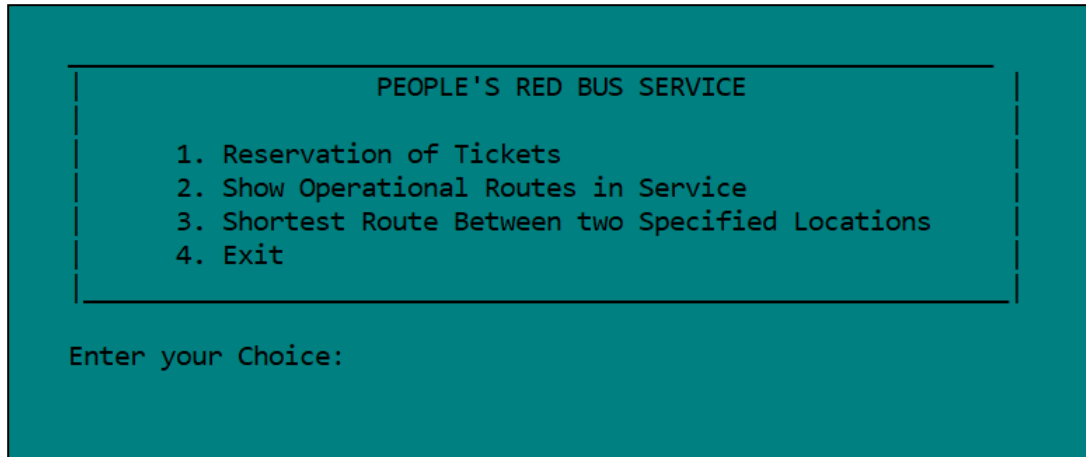
Another data structure used in this code is Graph we formed a graph with all the possible locations we had. Edges were then formed between them.

4. Dijkstra:

This was the crucial part of the code in which we had to find the shortest route between two specified locations. The shortest route was achieved using Dijkstra. Priorities were check and best ultimate Minimum distance was then achieved.

Outputs and their Meaning:

This output below shows the main menu of People Bus service system this is a menu driven program.



A screenshot of a terminal window showing the main menu for 'PEOPLE'S RED BUS SERVICE'. The menu is displayed within a rectangular box with a dashed border. Below the menu, the prompt 'Enter your Choice:' is visible.

```
PEOPLE'S RED BUS SERVICE

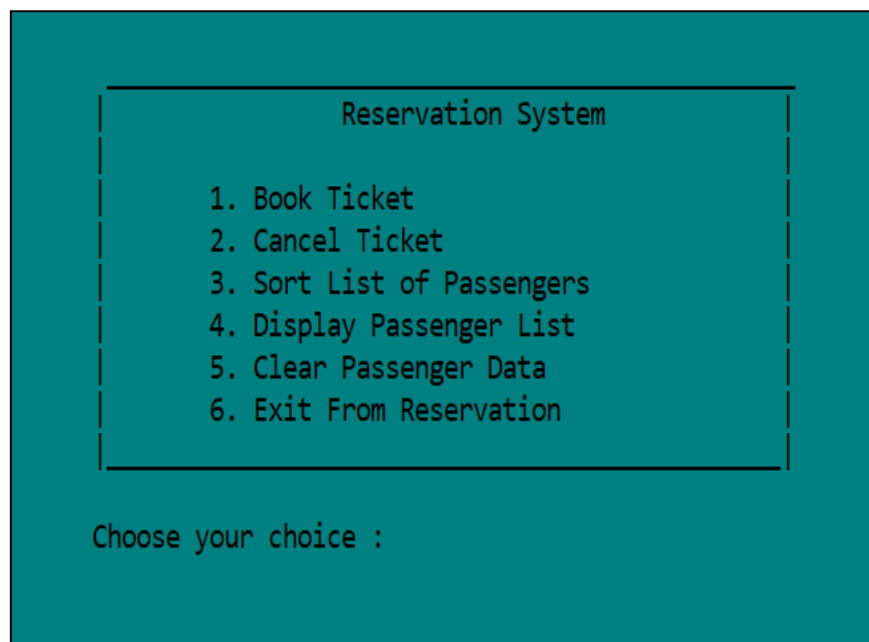
1. Reservation of Tickets
2. Show Operational Routes in Service
3. Shortest Route Between two Specified Locations
4. Exit

Enter your Choice:
```

Reservation:

A part of code deals with new persons entering the Bus. It just records their name and store the Passenger name with a unique ID.

Below is how this screen looks like,



A screenshot of a terminal window showing the 'Reservation System' menu. The menu is displayed within a rectangular box with a dashed border. Below the menu, the prompt 'Choose your choice :' is visible.

```
Reservation System

1. Book Ticket
2. Cancel Ticket
3. Sort List of Passengers
4. Display Passenger List
5. Clear Passenger Data
6. Exit From Reservation

Choose your choice :
```

Here is an example of entering a person named Fizza in the Passenger list.

```

                                Reservation System
                                -----
                                1. Book Ticket
                                2. Cancel Ticket
                                3. Sort List of Passengers
                                4. Display Passenger List
                                5. Clear Passenger Data
                                6. Exit From Reservation
                                -----

Choose your choice :  1

Enter Passenger Name: Fizza

||-----Passenger Successfully Entered-----||

```

This name would then be stored in a separate file named as PassengerList.txt

We can carryout more operations like Sorting the passenger list. We can cancel people's Ticket if they want to go out. Furthermore, it displays passenger list. One more Functionality is to clear the list so that new day shift should start, and previous passengers are removed.

Removing a Passenger through Unique seat Number:

```

                                Reservation System
                                -----
                                1. Book Ticket
                                2. Cancel Ticket
                                3. Sort List of Passengers
                                4. Display Passenger List
                                5. Clear Passenger Data
                                6. Exit From Reservation
                                -----

Choose your choice :  2

Enter Seat Number: 4

||-----Passenger Successfully Removed-----||

```

Displaying the Passengers:

```
Reservation System

1. Book Ticket
2. Cancel Ticket
3. Sort List of Passengers
4. Display Passenger List
5. Clear Passenger Data
6. Exit From Reservation

Choose your choice : 4

[Fizza- 1]->>[Ali- 2]->>[Dania- 3]->>[Sarim- 4]->>

Press any key to continue . . .
```

Displaying Customer after removal of a Passenger:

```
[Fizza- 1]->>[Ali- 2]->>[Dania- 3]->>[EmptySeat- 0]->>
```

Clearing Passenger List:

```
Reservation System

1. Book Ticket
2. Cancel Ticket
3. Sort List of Passengers
4. Display Passenger List
5. Clear Passenger Data
6. Exit From Reservation

Choose your choice : 5

||-----List Successfully Cleared-----||
```

Here is an output of Passengers when we clear the list:

```
Choose your choice : 4

[EmptySeat- 1]->>[EmptySeat- 2]->>[EmptySeat- 3]->>[EmptySeat- 0]->>

Press any key to continue . . .
```

Names of Stops:

This output shows the extract of output of stops along with their unique station.

```
19FiveStarChowrangi
20KDAChowrangi
21NazimabadEidGahGround
22Liaqtabad10Number
23EsaNagri
24CivicCentre
25NationalStadium
26KorangiRoad
27ShanChowrangi
28AllahWaliChowrangi
29NationalHighway5
30SteelMill
31BinQasimPort
32AbdullahGoth
33ChowkundiMorr
34FastUniversity
35BhainsColony
36ManzilPump
37Quaidabad
38MurghiKhana
39PrinceAlyBoysSchools
40MalirSessionCourt
41Malir15
42Numaish
43TajComplex
44AbdullahHaroon
```

Dijkstra Code:

We first need to enter source and destination. We need to make sure that Source and Destination entered should be same as stored in File.

```
Enter Source : Quaidabad
Enter Destination: bahadrabad
```

Below we can show list of shortest routes we assume Quaidabad as Zero and then find the shortest route using Dijkstra.

```
55          5
36          1
37          0
38          2
39          8
40          9
41         11
42         47
43         46
44         45
45         36
46         32
47         30
48         33
49         38
50         40
51         42
52         45
53        1061109567
54        1061109567
55        1061109567
56        1061109567
57        1061109567
58         21
59         16
60         11
61         21
Press any key to continue . . .
```

Final Output:

The final output shows the minimum distance possible between the routes.

It calculates Fare between the shortest routes.

It helps the passenger to find shortest and minimum fare possible for the specified route.

```
Press any key to continue . . .
Distance Between Quaidabad and bahadrabad: 21 Km
Fare for the given route: 210
```