Project Report

CSE 207

(Data Structures)

Submitted to

Amit Kumar Das
Senior Lecturer, Department of
Computer Science & Engineering
East West University

Submitted by

Ziaul Haque Rafi

ID: 2020-1-60-118

Kazi Jihan Hasan

ID: 2020-1-60-034

Group: 10
Section: 03

Date of Submission: 24.09.2021

Train Ticket Reservation System

Introduction:

In our project, we have made a train ticket reservation system using C language which will allow others to buy tickets through online. People can reserve their tickets for 8 different cities of Bangladesh. The program will also show available seats remaining. Moreover, the reservation history can also be seen using this program.

Structures and Functions:

```c
struct train
{
    int trainID;
    struct train *nextStation, *prevStation;
    char stationName;
    int seatAvailable;

} *head, *tail;

struct customer
{
    char name[25];
    int nid;
    int from;
    int destination;
    int price;
    struct customer *nextCustomer;

} *headQueue;
```

At first, we have made two structures: ‘train’ – will store different train’s information and ‘customer’ – will store all customer’s given information.

```c
void trainListSetUp()
{
    int i;
    head = (struct train *)malloc(sizeof(struct train));
    tail = (struct train *)malloc(sizeof(struct train));

    if(head == NULL || tail == NULL)
    {
        printf("Memory Error.\n");
        return;
    }
    else
    {
        struct train *newStation, *temp;
        int j = 1;

        head->trainID = j;
        head->nextStation = NULL;
        head->prevStation = NULL;
        head->seatAvailable = 5;
```

```
49          j++;
50          temp = head;
51
52          for(i = 2; i <= 8; i++)
53          {
54              newStation = (struct train *)malloc(sizeof(struct train));
55              temp->nextStation = newStation;
56              newStation->trainID = j;
57              newStation->nextStation = NULL;
58              newStation->prevStation = temp;
59
60              newStation->seatAvailable = 5;
61              temp = newStation;
62              j++;
63              if(i == 5)
64              {
65                  tail = newStation;
66              }
67          }
68
69          newStation->nextStation = head;
70          head->prevStation = newStation;
71      }
72  }
```

The function ‘trainListSetUp()’ creates the linked list by creating each node as stations and links them with each other for further use.

```
74  int seatBooking(int station)
75  {
76      struct train *temp;
77      temp = head;
78      int noSeat = 0;
79
80      while(noSeat == 0)
81      {
82          if(station == temp->trainID)
83          {
84
85              printf("Seat Remaining from this Station %d \n", temp->seatAvailable);
86
87              if(temp->seatAvailable == 0)
88              {
89                  noSeat = 1;
90                  printf("%d", temp->seatAvailable);
91                  return noSeat;
92              }
93              temp->seatAvailable--;
94              return 0;
95          }
96          else
97          {
98              temp = temp->nextStation;
99          }
100     }
101 }
```

The function ‘seatbooking()’ calculates the remaining seats and returns it to the main function as an integer. When there are no available seats, the function will return 1 which indicates no more seats available.

```
103  void seatFreeing(int station)
104  {
105      struct train *temp;
106      temp = head;
107      while (temp->nextStation == head)
108      {
109          if(station == temp->trainID)
110          {
111              temp->seatAvailable++;
112              return;
113          }
114          else
115          {
116              temp = temp->nextStation;
117          }
118      }
119  }
```

The function ‘seatFreeing()’ takes the destination as input from the customer and frees the seat in that train. Thus, it allows more customer to purchase the ticket from that station.

```
121  int ticketPrice(int from, int destination)
122  {
123      int price;
124      int diff;
125
126      diff = from - destination;
127
128      if (diff < 0)
129      {
130          diff = diff * (-1);
131          price = 500 * diff;
132      }
133      else
134      {
135          price = 500 * diff;
136      }
137      return price;
138  }
```

The function ‘ticketPrice()’ calculates the ticket price according to the distance between different stations and returns the ticket price as an integer.

```c
void ticketBooking()
{
    struct customer *newCustomer, *temp;

    if(firstCustomer == 0)
    {
labell:

        headQueue = (struct customer *)malloc(sizeof(struct customer));
        headQueue->nextCustomer = NULL;
        printf("\nWelcome First Customer!!!\n\nEnter Your Name: \n");
        scanf("%s", headQueue->name);

        printf("Enter Your NID:\n");
        scanf("%d", &headQueue->nid);
        printf("From: \n1.Dhaka\n2.Cumilla\n3.Chittagong\n4.Barisal\n5.Khulna\n6.Rajshahi\n7.Sylhet\n8.Dinajpur\n");
        scanf("%d", &headQueue->from);
        seatBooking(headQueue->from);
        printf("To: \n1.Dhaka\n2.Cumilla\n3.Chittagong\n4.Barisal\n5.Khulna\n6.Rajshahi\n7.Sylhet\n8.Dinajpur\n");
        scanf("%d", &headQueue->destination);
        headQueue->price = ticketPrice(headQueue->from, headQueue->destination);
        printf("The Ticket Price is : %d \n", headQueue->price);
        printf("Would you like to confirm the purchase?\n");
        printf("1.Yes\t2.No\n");
        int choice;
        scanf("%d", &choice);
        if(choice == 1)
        {

        }
        else
        {
            goto labell;
        }
        seatFreeing(headQueue->destination);

        firstCustomer = 1;
        printf("First Booking is Successful.\n\n");
    }
    else
    {
        int seat;
```

The function ‘ticketBooking()’ allows customers to input their information required to reserve a seat. There are two labels included in this function. The first label is for the first customer and the second one is for the next customers. If a customer does not confirm his/her ticket then it will return him/her to the first part of the current function.

```
234   void trainAvailable()
235   {
236
237       struct train *temp;
238       temp = head;
239
240       printf("The Departing from Dhaka has remaining seats : %d\n", temp->seatAvailable);
241       temp = temp->nextStation;
242       printf("The Departing from Cumilla has remaining seats : %d\n", temp->seatAvailable);
243       temp = temp->nextStation;
244       printf("The Departing from Chittagong has remaining seats : %d\n", temp->seatAvailable);
245       temp = temp->nextStation;
246       printf("The Departing from Barisal has remaining seats : %d\n", temp->seatAvailable);
247       temp = temp->nextStation;
248       printf("The Departing from Khulna has remaining seats : %d\n", temp->seatAvailable);
249       temp = temp->nextStation;
250       printf("The Departing from Rajshahi has remaining seats : %d\n", temp->seatAvailable);
251       temp = temp->nextStation;
252       printf("The Departing from Sylhet has remaining seats : %d\n", temp->seatAvailable);
253       temp = temp->nextStation;
254       printf("The Departing from Dinajpur has remaining seats : %d\n", temp->seatAvailable);
255       temp = temp->nextStation;
256       printf("\n");
257
258   }
```

The function ‘trainAvailable()’ will show the remaining seats of different train stations.

```
260   void viewCustomerHistory()
261   {
262
263       struct customer *temp;
264       temp = headQueue;
265
266       while(temp != NULL)
267       {
268
269           if(temp == headQueue)
270           {
273
274           printf("Customer Details: \n\n");
275           printf("Customer's Name : %s\n", temp->name);
276           printf("Customer's NID : %d \n", temp->nid);
277           printf("Customer's Ticket Price: %d\n", temp->price);
278           printf("Departure from: ");
279
280           if(temp->from % 2 == 0)
281           {
300
301           else
302           {
320
321           printf("To: ");
322
323           if(temp->destination % 2 == 0)
324           {
343           else
344           {
362
363           temp = temp->nextCustomer;
364       }
365       printf("\n\n");
```

The function ‘viewCustomerHistory()’ shows the customer’s given information which was stored in the queue while reserving tickets.

```c
int main()
{
    trainListSetUp();

    int option;
    int exit = 0;
    printf("           Welcome to Online Train Reservation Platform.          \n\n");

    while(exit == 0)
    {
        printf("1. Reserve a Train Ticket.\n");
        printf("2. View Available Train Details.\n");
        printf("3. View Customer History.\n");
        printf("4. Exit.\n\n");
        scanf("%d", &option);

        switch(option)
        {
        case 1:
            ticketBooking();
            break;
        case 2:
            trainAvailable();
            break;
        case 3:
            viewCustomerHistory();
            break;
        case 4:
            exit = 1;
            break;
        }
    }

    return 0;
}
```

Finally, we have made the main function which will call the ‘trainListSetup()’ function and then will ask the user for choosing an option from the list. We have used the ‘switch’ for choosing a choice. Case-1 will take the user to the ‘ticketBooking()’ function, Case-2 will take the user to the ‘trainAvailable()’ function, Case-3 will take the user to the ‘viewCustomerHistory()’ function and Case-4 will end the program.

Conclusion:

The project was about the train reservation system. By using this program, people can easily reserve their train tickets. Usage of linked list and queue have made the project more efficient.