

```
1 package Model;
2
3 import java.util.ArrayList;
10
11 public class TicketBooker {
12     static int availableLowerBerths = 2;//normally 21
13     static int availableMiddleBerths = 2;//normally 21
14     static int availableUpperBerths = 2;//normally 21
15     static int availableRacTickets = 2;//normally 18
16     static int availableWaitingList = 2;//normally 10
17     static Queue<Integer> rac=new LinkedList<Integer>();
18     static Queue<Integer> wait=new LinkedList<Integer>();
19     static HashSet<Integer> Bookedticket=new HashSet<Integer>();
20     static List<Integer> availableLowerBerthshas=new ArrayList<Integer>(Arrays.asList(1,2));
21     static List<Integer>availableMiddleBerthshas=new ArrayList<Integer>(Arrays.asList(1,2));
22     static List<Integer>availableUpperBerthshas=new ArrayList<Integer>(Arrays.asList(1,2));
23     static List<Integer>availableRacTicketshas=new ArrayList<Integer>(Arrays.asList(1,2));
24     static List<Integer>availableWaitingListhas=new ArrayList<Integer>(Arrays.asList(1,2));
25     static HashMap<Integer,User> has=new HashMap<Integer,User>();
26     public void cancel_ticket(int id, User s) {
27         int number=s.number;
28         if(s.berthPreference.equals("U"))
29         {
30             availableUpperBerths=availableUpperBerths+1;
31             availableUpperBerthshas.add(number);
32             System.out.println("_____Successfully Cancel Your Ticket_____");
33         }else if(s.berthPreference.equals("L"))
34         {
35             availableLowerBerths=availableLowerBerths+1;
36             availableLowerBerthshas.add(number);
37             System.out.println("_____Successfully Cancel Your Ticket_____");
38         }else if(s.berthPreference.equals("M"))
39         {
40             availableMiddleBerths=availableMiddleBerths+1;
41             availableMiddleBerthshas.add(number);
42             System.out.println("_____Successfully Cancel Your Ticket_____");
43         }else if(s.berthPreference.equals("RAC"))
44         {
45             availableRacTickets=availableRacTickets+1;
46             availableRacTicketshas.add(number);
47             System.out.println("_____Successfully Cancel Your Ticket_____");
48         }else {
49             availableWaitingList=availableWaitingList+1;
50             availableWaitingListhas.add(number);
51             System.out.println("_____Successfully Cancel Your Ticket_____");
52         }
53         has.remove(Integer.valueOf(id));
54
55     }
56
57
58     public void available_ticket() {
59         System.out.println("Available Lower Berths " + availableLowerBerths);
60         System.out.println("Available Middle Berths " + availableMiddleBerths);
61         System.out.println("Available Upper Berths " + availableUpperBerths);
62         System.out.println("Available RACs " + availableRacTickets);
63         System.out.println("Available Waiting List " + availableWaitingList);
```

```
64         System.out.println("_____");
65
66     }
67
68     public void book_ticket() {
69         if(has.isEmpty())
70         {
71             System.out.println("_____No Yet Register Now_____");
72         }else {
73             for(User s:has.values())
74             {
75                 System.out.println("User ID      :"+s.passengerId);
76                 System.out.println("User Name   :"+s.name);
77                 System.out.println("User Age    :"+s.age);
78                 System.out.println("User Berths :"+s.number+s.alloted);
79                 System.out.println("_____");
80             }
81         }
82     }
83 }
84
85 public void booked(User s,int berthInfo,String allotedBerth) {
86     s.number=berthInfo;
87     s.alloted=allotedBerth;
88     has.put(s.passengerId, s);
89     Bookedticket.add(s.passengerId);
90     System.out.println("_____Booked Successfully_____");
91 }
92
93 }
94
```