```
package TrainTicketBookingApp;

import java.util.ArrayList;
import java.util.List;

public class Main {

	public static void main(String[] args) {
		// TODO Auto-generated method stub

		List<String> stoppages = new ArrayList<>();
		stoppages.add("A");
		stoppages.add("B");
		stoppages.add("C");
		stoppages.add("D");
		stoppages.add("E");

		Train train = new Train("VandeBharat", 1010, stoppages);
		User user1 = new User("user1", "A", "E", 8);

		User user2 = new User("user2", "C", "D", 2);

		User user3 = new User("user2", "A", "B", 2);



		train.getAvailability("A", "B"); // Expected 8
		train.getAvailability("B", "C"); // Expected 8
		train.getAvailability("A", "C"); // Expected 8
		train.getAvailability("D", "E"); // Expected 8"

		System.out.println("-------------------------");

		Ticket ticket1 = train.bookticket(user1);
		System.out.println(ticket1);
		System.out.println("-------------------------");

		train.getAvailability("A", "B"); // Waiting
		train.getAvailability("B", "C"); // Waiting
		train.getAvailability("A", "C"); // Waiting
		train.getAvailability("D", "E"); // Waiting

		System.out.println("-------------------------");
		Ticket ticket2 = train.bookticket(user2);
		System.out.println(ticket2);

		System.out.println("-------------------------");

		ticket1 = train.cancelTicket(1, 2);
		System.out.println(ticket1);

		System.out.println("-------------------------");

		train.getAvailability("A", "B"); // Expected 2
		train.getAvailability("B", "C"); // Expected 2
		train.getAvailability("A", "C"); // Expected 2
		train.getAvailability("D", "E"); // Expected 2

		System.out.println("-------------------------");
```

```
                System.out.println("------------------------");

                train.totalBookingInfo();

                System.out.println("------------------------");


                ticket1 = train.cancelTicket(1, 2);
                System.out.println(ticket1);


                System.out.println("------------------------");

                train.totalBookingInfo();

                System.out.println("------------------------");


                System.out.println("------------------------");

                train.getAvailability("A", "B"); // Expected 4
                train.getAvailability("B", "C"); // Expected 4
                train.getAvailability("A", "C"); // Expected 4
                train.getAvailability("D", "E"); // Expected 4

                System.out.println("------------------------");
        }

}
```

```java
package TrainTicketBookingApp;

import java.util.List;

public class Ticket {
    private int pnr;
    private String source;
    private String destination;
    private int noOfTickets;
    private boolean isConfirmed;
    private boolean isWaitinglist;
    private List<Integer> seatNums;

    public Ticket(int pnr, String source, String
destination, int noOfTickets, boolean isConfirmed,
            boolean isWaitinglist, List<Integer>
seatNums) {
        super();
        this.pnr = pnr;
        this.source = source;
        this.destination = destination;
```

```
        this.noOfTickets = noOfTickets;
        this.isConfirmed = isConfirmed;
        this.isWaitinglist = isWaitinglist;
        this.seatNums = seatNums;
    }

    public int getPnr() {
        return pnr;
    }

    public void setPnr(int pnr) {
        this.pnr = pnr;
    }

    public String getSource() {
        return source;
    }

    public void setSource(String source) {
        this.source = source;
    }

    public String getDestination() {
        return destination;
    }

    public void setDestination(String destination) {
        this.destination = destination;
    }

    public int getNoOfTickets() {
        return noOfTickets;
    }

    public void setNoOfTickets(int noOfTickets) {
        this.noOfTickets = noOfTickets;
    }

    public boolean isConfirmed() {
        return isConfirmed;
    }

    public void setConfirmed(boolean isConfirmed) {
        this.isConfirmed = isConfirmed;
    }
```

```
    public boolean isWaitinglist() {
        return isWaitinglist;
    }

    public void setWaitinglist(boolean isWaitinglist) {
        this.isWaitinglist = isWaitinglist;
    }

    public List<Integer> getSeatNums() {
        return seatNums;
    }

    public void setSeatNums(List<Integer> seatNums) {
        this.seatNums = seatNums;
    }

    @Override
    public String toString() {
        return "Ticket [pnr=" + pnr + ", source=" +
source + ", destination=" + destination + ",
noOfTickets="
                + noOfTickets + ", isConfirmed=" +
isConfirmed + ", isWaitinglist=" + isWaitinglist + ",
seatNums="
                + seatNums + "]";
    }


}


package TrainTicketBookingApp;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.List;
import java.util.Map.Entry;
import java.util.Queue;

public class Train {

    private String name;
```

```
    private int iD;
    private List<String> stoppages;
    private static final int MAX_SEATS = 8;
    private static final int MAX_WAITING_SEATS = 2;
    private int reminingSeats = MAX_SEATS;
    private int confirmedSeats = 0;
    private String source;
    private String destination;
    HashMap<String, List<Integer>>
tickettodifferentdestinations = new HashMap<>();
    private User user;
    String ticketSource;
    String ticketDestination;
    int ticketCount;
    int currentWaitingList = 0;
    boolean iSCheckingfromBooking = false;
    private int pnrNum = 0;
    private List<Integer> seatnumber = new
ArrayList<>();
    private int seatNumCount = 0;
    private boolean istrainWaitingLisrtAvailable =
false;
    private HashMap<Integer, Ticket> totalTicketInfo =
new HashMap<>();
    private Queue<Ticket> pnrQueue = new LinkedList<>();

    Train(String name, int iD, List<String> stoppages) {
        this.name = name;
        this.iD = iD;
        this.stoppages = stoppages;
        for (int i = 0; i < stoppages.size(); i++) {
            ArrayList<Integer> NumOfTickets = new
ArrayList<>();
            for (int j = 0; j < stoppages.size(); j++) {
                if (i < j) {
                    int count = i == j ? 0 : MAX_SEATS;
                    NumOfTickets.add(count);
                } else {
                    NumOfTickets.add(0);
                }

            }
```

```
tickettodifferentdestinations.put(stoppages.get(i),
NumOfTickets);

        }
        for (int i = 0; i < MAX_SEATS; i++) {
            seatnumber.add(i);
        }
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getiD() {
        return iD;
    }

    public void setiD(int iD) {
        this.iD = iD;
    }

    public List<String> getStoppages() {
        return stoppages;
    }

    public void setStoppages(List<String> stoppages) {
        this.stoppages = stoppages;
    }

    public static int getMaxSeats() {
        return MAX_SEATS;
    }

    public static int getMaxWaitingSeats() {
        return MAX_WAITING_SEATS;
    }

    public int getAvailability(String source, String
destination) {
        source = source.toUpperCase();
```

```
        destination = destination.toUpperCase();

        if (destination.toCharArray()[0] <
source.toCharArray()[0]) {
            System.out.println("Not allowed");
            return 0;
        }
        if (source.equals(destination)) {
            System.out.println("Source and Destination
should not be same");
            return 0;
        }
        ArrayList<Integer> noOfSeatAvailableList =
(ArrayList<Integer>)
tickettodifferentdestinations.get(source);
        int numOfSeats = 0;
        switch (destination) {
        case "A":
            numOfSeats = noOfSeatAvailableList.get(0);

            break;
        case "B":
            numOfSeats = noOfSeatAvailableList.get(1);
            break;
        case "C":
            numOfSeats = noOfSeatAvailableList.get(2);
            break;
        case "D":
            numOfSeats = noOfSeatAvailableList.get(3);
            break;
        case "E":
            numOfSeats = noOfSeatAvailableList.get(4);
            break;

        default:
            break;
        }

        if (numOfSeats > 0 && !iSCheckingfromBooking) {
            System.out
                    .println(" Available Seats are :
Between " + source + " and " + destination + " Is :" +
numOfSeats);
            return numOfSeats;
```

```
        } else if (numOfSeats > 0 && numOfSeats >
currentWaitingList) {
            if (!iSCheckingfromBooking) {

                System.out.println(
                        " Available Seats are : Between
" + source + " and " + destination + " Is :" +
numOfSeats);
            }
            return numOfSeats;
        } else if (!iSCheckingfromBooking) {
            System.out.println("No tickets available ,
only Waiting list ");
        }
        return 0;

    }

    public int generatePnr() {
        pnrNum++;
        return pnrNum;
    }

    public void totalBookingInfo() {

        for (Entry<Integer, Ticket> entry :
totalTicketInfo.entrySet()) {
            System.out.println(entry.getValue());
        }

    }

    public boolean getTrainWaitingListStatus() {
        return this.istrainWaitingLisrtAvailable;
    }

    public Ticket cancelTicket(int pnr, int noOfSeats) {
        Ticket ticket = totalTicketInfo.get(pnr);
        ArrayList<Integer> seats = (ArrayList<Integer>)
ticket.getSeatNums();
        int comfirmedSeats = seats.size();
        if (comfirmedSeats < noOfSeats) {
            System.out.println("Please give proper
info : ");
            return null;
```

```
        }
        ticket.setNoOfTickets(comfirmedSeats -
noOfSeats);

        for (int i = 0; i < noOfSeats; i++) {
            seats.remove(comfirmedSeats - i - 1);
        }
        ticket.setSeatNums(seats);
        seatNumCount = seatNumCount - noOfSeats;
        String from = "FromTicketCancellation";
        System.out.println("Cancelled " + noOfSeats + "
successfully");

        boolean isWaitingListAvailable =
this.getTrainWaitingListStatus();


this.updateTicketAvailability(ticket.getSource(),
ticket.getDestination(), noOfSeats, from);
        if (isWaitingListAvailable) {
            confirmWaitingList(pnrQueue);
        }

        return ticket;
    }

    public Ticket confirmWaitingList(Queue<Ticket>
waitingList) {
        for (Iterator iterator = waitingList.iterator();
iterator.hasNext();) {
            Ticket ticket = (Ticket) iterator.next();
            // System.out.println(ticket);
            int noOfTickets = ticket.getNoOfTickets();
            String destination =
ticket.getDestination();
            String source = ticket.getSource();
            int pnr = ticket.getPnr();

            if (noOfTickets <= currentWaitingList) {
                String from = "TicketBooking";
                this.updateTicketAvailability(source,
destination, noOfTickets, from);
                this.currentWaitingList =
this.currentWaitingList - noOfTickets;
```

```
                List<Integer> seats = new ArrayList<>();
                for (int i = 0; i < this.ticketCount; i+
+) {
                    if (seatNumCount >= 8) {
                        seatNumCount = seatNumCount %
MAX_SEATS;
                    }
                    seats.add(++seatNumCount);

                }
                ticket.setSeatNums(seats);
                ticket.setConfirmed(true);
                ticket.setWaitinglist(false);

                totalTicketInfo.put(pnr, ticket);
                this.iSCheckingfromBooking = false;
            }

        }
        return null;
    }

    public Ticket bookticket(User user) {
        this.iSCheckingfromBooking = true;
        this.user = user;
        this.ticketSource = user.getUserSource();
        this.ticketDestination =
user.getUserDestination();
        this.ticketCount = user.getNoOfTickets();

        if (getAvailability(ticketSource,
ticketDestination) < this.ticketCount
                && this.currentWaitingList >
this.ticketCount) {
            System.out.println("No tickets available");
            return null;
        } else if (this.currentWaitingList
<this.ticketCount
                && getAvailability(ticketSource,
ticketDestination) < this.ticketCount) {
            System.out.println("You have been given to
waiting list");
            int pnrGenetedNum = this.generatePnr();
            List<Integer> seats = new ArrayList<>();
            this.istrainWaitingLisrtAvailable = true;
```

```
            currentWaitingList = currentWaitingList +
this.ticketCount;


            Ticket ticket = new Ticket(pnrGenetedNum,
this.ticketSource, this.ticketDestination,
this.ticketCount,
                    false, true, seats);
            totalTicketInfo.put(pnrGenetedNum, ticket);
            pnrQueue.add(ticket);
            return ticket;
        } else {
            System.out.println("Ticket booked
Successfully");
            String from = "TicketBooking";
            updateTicketAvailability(this.ticketSource,
this.ticketDestination, this.ticketCount, from);
            int pnrGenetedNum = this.generatePnr();
            List<Integer> seats = new ArrayList<>();
            for (int i = 0; i < this.ticketCount; i++) {
                if (seatNumCount >= 8) {
                    seatNumCount = seatNumCount %
MAX_SEATS;
                }
                seats.add(++seatNumCount);

            }
            Ticket ticket = new Ticket(pnrGenetedNum,
this.ticketSource, this.ticketDestination,
this.ticketCount, true,
                    false, seats);
            totalTicketInfo.put(pnrGenetedNum, ticket);
            this.iSCheckingfromBooking = false;
            return ticket;
        }

    }

    public void updateTicketAvailability(String source,
String destination, int count, String callingFrom) {
        source = source.toUpperCase();
        destination = destination.toUpperCase();
        int sourceIndex = source.toCharArray()[0] - 'A';
        int destinationIndex = destination.toCharArray()
[0] - 'A';
```

```
        for (int i = 0; i < this.stoppages.size(); i++)
{
            int numOfSeats = 0;

            char c = (char) (i + 'A');
            String key = c + "";
            ArrayList<Integer> noOfSeatAvailableList =
(ArrayList<Integer>)
tickettodifferentdestinations.get(key);
            for (int j = 0; j <
noOfSeatAvailableList.size(); j++) {
                if (j > sourceIndex && i < sourceIndex)
{

                    numOfSeats =
noOfSeatAvailableList.get(j);
                    if
("FromTicketCancellation".equals(callingFrom)) {
                        numOfSeats = numOfSeats +
count;
                    } else if
("TicketBooking".equals(callingFrom)) {
                        numOfSeats = numOfSeats -
count;
                    }

                    if (numOfSeats <= 0) {
                        numOfSeats = 0;
                    }

                    noOfSeatAvailableList.remove(j);
                    noOfSeatAvailableList.add(j,
numOfSeats);

                } else if (j > sourceIndex && i <
destinationIndex) {

                    numOfSeats =
noOfSeatAvailableList.get(j);

                    if
("FromTicketCancellation".equals(callingFrom)) {
                        numOfSeats = numOfSeats +
count;
```

```
                        } else if
("TicketBooking".equals(callingFrom)) {
                            numOfSeats = numOfSeats -
count;
                        }

                        if (numOfSeats <= 0) {
                            numOfSeats = 0;
                        }

                        noOfSeatAvailableList.remove(j);
                        noOfSeatAvailableList.add(j,
numOfSeats);

                    }

                }
                this.iSCheckingfromBooking = false;
                tickettodifferentdestinations.put(key,
noOfSeatAvailableList);

            }

        }
}


package TrainTicketBookingApp;

public class User {
    private String name;
    private String userSource;
    private String userDestination;
    private int noOfTickets;

    public User(String name, String source, String
destination, int noOfTickets) {

        this.name = name;
        this.userSource = source;
        this.userDestination = destination;
        this.noOfTickets = noOfTickets;
    }

    public String getName() {
```

```
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getUserSource() {
        return userSource;
    }

    public void setUserSource(String userSource) {
        this.userSource = userSource;
    }

    public String getUserDestination() {
        return userDestination;
    }

    public void setUserDestination(String
userDestination) {
        this.userDestination = userDestination;
    }

    public int getNoOfTickets() {
        return noOfTickets;
    }

    public void setNoOfTickets(int noOfTickets) {
        this.noOfTickets = noOfTickets;
    }

}
```