**KLE Society's**

**KLE Technological University**



**COURSE PROJECT**

**Travel Agency Management System**

**Submitted by**

**Name Roll no SRN**

Suraj Baddi 525 01FE19BEC268

Abhisek Sunadholi 544 01FE19BEC290

Niranjan Bhuti 507 01FE19BEC248

Vishwas Banagar 517 01FE19BEC258

**------------------------------------------------------------------------------------------------------------------------**

**Faculty In-charge:**

**Prof. Vishal Pattanshetti**

**SCHOOL OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**Academic year 2021-22**

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Overview of the problem statement

1.2 Features of application

1.2.1 booking hotel for customer

1.2.2 booking transport for customer

1.2.3 search/edit/delete booking for customer

**2. DESIGN**

2.1 class diagram

2.2 Description of Each Class

2.3 Main Function

2.4 Use of Standard Design Patterns

**3. UNIT TEST PLAN**

3.1 getdetails

3.2 booking of hotel and transport

3.3 searching and deleting of records

3.4 view of records

**4. IMPLEMENTATION**

4.1 Results

1. **Introduction**
   1. Overview of Problem Statement

Our Travel agency management system consists of hotels (5 star, 3 star and normal), and transport services (by road and by train). The present systems are inadequate in providing information customers about tours plans. Often agencies are compelled to rely on local information sources and count on their own experience regarding time and cost. The current agencies are almost forced to gather information regarding the services through phone calls and are often poorly managed. We provide information of hotel and transport through online fixed price for our customers, through which they can choose the right package which also fits their budget.

* 1. Features of application
     1. Hotel Booking
* The customers have a wide range of options available from set of hotels
* The customers can choose the type of hotel they want.
* The customers are also availed with the real time price of the hotel of their desire and can book it instantly.
  + 1. Transport Booking
* Here, the customers are free to choose the mode of transportation they please
* The customer can choose from wide range of transport options available, be it road transport or train transport.
* The real time price of the tickets are also made available to the customers, they are free to choose the type of transport and book their tickets instantly.
  + 1. Search, Delete and View Bookings
* The customers can view or search their bookings once they have completed their payment.
* The customers are also free to delete their bookings and are eligible to get refund.

**2. DESIGN**

2.1 class diagram

|  |
| --- |
| **customer** |
| **+noofcustomers: int**  **+dd,mm,yyyy: int**  **+name:string**  **+phone:long long int**  **+address:string**  **+mail: string**  **+Booking():Booking;** |
| **getdetails():void**  **Createfile():void** |

|  |
| --- |
| **travelagency** |
| **+name:Static string**  **+phone:static long long int**  **+customer():customer;** |
| **showcustomerdetails():void** |

|  |
| --- |
| **Booking** |
| **+toh:int**  **+tot:int**  **+Transport():Transport;**  **+Hotel():Hotel;** |
| **+ sethoteldetails(): void**  **+ settransportdetails(): void**  **+ Searchdetails(): void**  **+ searchhoteldetails(): void**  **+ searchtraveldetails(): void**  **+ deleteTravRecord(): void**  **+ deleteHtlRecord(): void**  **+ Viewdetails():void**  **+ viewtraveldetails():void**  **+ viewhoteldetails():void** |

BOOKING

|  |
| --- |
| **<< abstract >>interface**  **Transport** |
| **+tname: string**  **+ tadd: string**  **+ destination: string**  **+ trans: string**  **+ modeoftransport: string**  **+ tdd,tmm,tyyyy: int**  **+ nooftickets: int** |
| **+ Assignprice():virtual void**  **+ Static Createl(TransportType type): Transport\*** |

|  |
| --- |
| **<< abstract >>interface**  **Hotel** |
| **+ hotel:String**  **+ destination:String**  **+ bname:String**  **+ add:String**  **+ typeofhotel:String**  **+ noofrooms:int**  **+ noofdays:int**  **+ cdd,cmm,cyyyy:int** |
| **+ Assignprice():virtual void**  **+ Static Createl(HotelType type): Hotel\*** |

|  |
| --- |
| **Normal** |
| **+nst:int** |
| **+ Assignprice():void**  **+ ShowHotel(string):void**  **+ confirmBooking():void**  **+ CreateHotelfile():void** |
|  |

|  |
| --- |
| **Tstar** |
| **+ tst:int** |
| **+ Assignprice():void**  **+ ShowHotel(string):void**  **+ confirmBooking():void**  **+ CreateHotelfile():void** |
|  |

|  |
| --- |
| **Fstar** |
| **+ fst:int** |
| **+ Assignprice():void**  **+ ShowHotel(string):void**  **+ confirmBooking():void**  **+ CreateHotelfile():void** |
|  |

|  |
| --- |
| **<< abstract >>interface**  **Transport** |
| **+tname: string**  **+ tadd: string**  **+ destination: string**  **+ trans: string**  **+ modeoftransport: string**  **+ tdd,tmm,tyyyy: int**  **+ nooftickets: int** |
| **+ Assignprice():virtual void**  **+ Static Createl(TransportType type): Transport\*** |

|  |
| --- |
| **Road** |
| **+ rt:int** |
| **+ Assignprice():void**  **+ ShowTransport(string):void**  **+ confirmTBooking():void**  **+ CreateTravelfile():void** |

|  |
| --- |
| **Train** |
| **+ tt:int** |
| **+ Assignprice():void**  **+ ShowTransport(string):void**  **+ confirmTBooking():void**  **+ CreateTravelfile():void** |

|  |
| --- |
| **Road** |
| **+ rt:int** |
| **+ Assignprice():void**  **+ ShowTransport(string):void**  **+ confirmTBooking():void**  **+ CreateTravelfile():void** |

|  |
| --- |
| **<< abstract >>interface**  **Transport** |
| **+tname: string**  **+ tadd: string**  **+ destination: string**  **+ trans: string**  **+ modeoftransport: string**  **+ tdd,tmm,tyyyy: int**  **+ nooftickets: int** |
| **+ Assignprice():virtual void**  **+ Static Createl(TransportType type): Transport\*** |
| **<< abstract >>interface**  **Transport** |
| **+tname: string**  **+ tadd: string**  **+ destination: string**  **+ trans: string**  **+ modeoftransport: string**  **+ tdd,tmm,tyyyy: int**  **+ nooftickets: int** |
| **+ Assignprice():virtual void**  **+ Static Createl(TransportType type): Transport\*** |
| **<< abstract >>interface**  **Transport** |
| **+tname: string**  **+ tadd: string**  **+ destination: string**  **+ trans: string**  **+ modeoftransport: string**  **+ tdd,tmm,tyyyy: int**  **+ nooftickets: int** |
| **+ Assignprice():virtual void**  **+ Static Createl(TransportType type): Transport\*** |

**2.2 Description of each class**

**1.Hotel**

|  |
| --- |
| **<< abstract >>interface**  **Hotel** |
| **+ hotel:String**  **+ destination:String**  **+ bname:String**  **+ add:String**  **+ typeofhotel:String**  **+ noofrooms:int**  **+ noofdays:int**  **+ cdd,cmm,cyyyy:int** |
| **+ Assignprice():virtual void=0**  **+ Static Createl(HotelType type): Hotel\*** |

* This is an Abstract class where we have a pure virtual function “Assignprice ()” which is used for assigning price of hotel and calculate price for total days and total members.
* It has 3 sub-classes: 5-star,3-star and normal
* The class Booking has a relationship with hotel.

|  |
| --- |
| **Normal** |
| **+nst:int** |
| **+ Assignprice():void**  **+ ShowHotel(string):void**  **+ confirmBooking():void**  **+ CreateHotelfile():void** |
|  |

|  |
| --- |
| **Tstar** |
| **+ tst:int** |
| **+ Assignprice():void**  **+ ShowHotel(string):void**  **+ confirmBooking():void**  **+ CreateHotelfile():void** |
|  |

1. **5 star,3 star and normal class**

|  |
| --- |
| **Fstar** |
| **+ fst:int** |
| **+ Assignprice():void**  **+ ShowHotel(string):void**  **+ confirmBooking():void**  **+ CreateHotelfile():void** |
|  |

* The Assignprice function is overridden from hotel class to all three classes. It assigns different price at location for different hotels.
* The ShowHotel function is overridden from hotel class to all three class. Displays the information of hotel.
* The ConfirmBooking function is overridden from hotel class to all three classes. It confirms the booking and total amount must be paid
* The CreateHotelfilefunction is overridden from hotel class to all three classes. The records are stored in the file.
* All the three class are in “is a” relationship with hotel class.

1. **Transport**

|  |
| --- |
| **<< abstract >>interface**  **Transport** |
| **+tname: string**  **+ tadd: string**  **+ destination: string**  **+ trans: string**  **+ modeoftransport: string**  **+ tdd,tmm,tyyyy: int**  **+ nooftickets: int** |
| **+ Assignprice():virtual void**  **+ Static Createl(TransportType type): Transport\*** |

* This is an Abstract class where we have a pure virtual function “Assignprice ()” which is used for assigning price of transport type and calculate price for total days and total members.
* It has 2 sub classes: road and train.
* The class Booking “has a” relationship with transport class.

1. **Road and train**

|  |
| --- |
| **Train** |
| **+ tt:int** |
| **+ Assignprice():void**  **+ ShowTransport(string):void**  **+ confirmTBooking():void**  **+ CreateTravelfile():void** |

|  |
| --- |
| **Road** |
| **+ rt:int** |
| **+ Assignprice():void**  **+ ShowTransport(string):void**  **+ confirmTBooking():void**  **+ CreateTravelfile():void** |

* The Assignprice function is overridden from transport class to other classes. It assigns different price at location.
* The ShowTransport function is overridden from transport class to other two classes. It Displays the information of transport type.
* The ConfirmTBooking function is overridden from transport classes. It confirms the booking of transport type and total amount must be paid
* The CreateTravelfile function is overridden from transport classes. The records are stored in the file.
* The 2 class Train and Road have “is a” relationship with transport.

**5.Booking**

|  |
| --- |
| **Booking** |
| **+toh:int**  **+tot:int**  **+Transport():Transport;**  **+Hotel():Hotel;** |
| **+ sethoteldetails(): void**  **+ settransportdetails(): void**  **+ Searchdetails(): void**  **+ searchhoteldetails(): void**  **+ searchtraveldetails(): void**  **+ deleteTravRecord(): void**  **+ deleteHtlRecord(): void**  **+ Viewdetails():void**  **+ viewtraveldetails():void**  **+ viewhoteldetails():void** |

* Toh is price of hotel, tot is price of transport booking class” has a “relationship with hotel and transport classes.
* It has member functions to sethoteldetails, settransportdetails, search, delete, view.
* Its member function sethoteldetails calculates the price for total members and total days and settransportdetails calculates the total price depending on the no. of tickets.
* The remaining classes are used to search, view and delete hotel and transport records.

1. **Customer**

|  |
| --- |
| **customer** |
| **+noofcustomers: int**  **+dd,mm,yyyy: int**  **+name:string**  **+phone:long long int**  **+address:string**  **+mail: string**  **+Booking():Booking;** |
| **getdetails():void**  **Createfile():void** |

* This class takes customer details as input.
* The customer class has a relationship with booking class.
* The getdails and createfile functions assign details createfile function create file for storing of details.

**7. Travel Agency**

|  |
| --- |
| **travelagency** |
| **+name:Static string**  **+phone:static long long int**  **+customer():customer;** |
| **showcustomerdetails():void** |

* Travel agency “has a” relationship with class customer.
* It contains the travel agency name and phone number.
* The member function is used to call the customer class.

**2.3 Main Function**

In main function we create a single object of travelagency.

The travel agency calls the booking class, in the booking class there is a switch menu for different options like sethoteldetails, settransportationdetails, Searchdetails, Viewdetails, Also create file to store the records.

According to the user input we will create a hotel and transport using factory design pattern. Once they get created, we ask our user which hotel they would like, 5 star,3 star or a normal hotel and in transporttype what kind of transport they would like road or train.

Once all the choices are made by the customer, hotel and transport will be confirmed and price will be calculated. Exceptions handling has been implemented where the customers have to choose the type of hotel and the type of transport.

**2.4 Use of standard Design patterns**

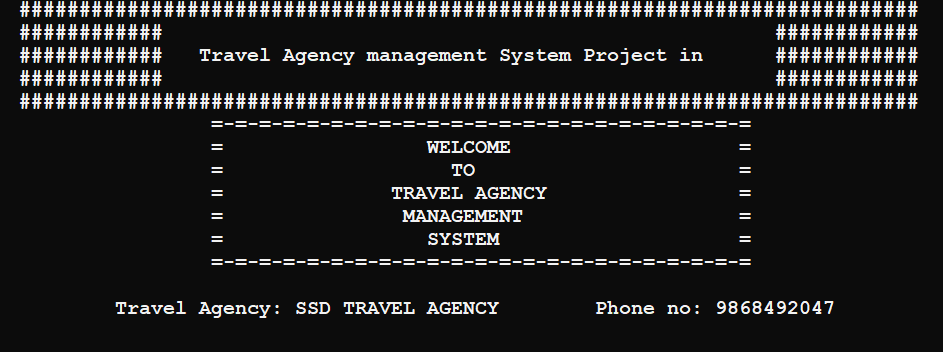
Factory design pattern is used for the above application.

***• Factory method pattern:***

• Definition: - The Factory Method pattern is a design pattern used to define a runtime interface for creating an object. It’s called a factory because it creates various types of objects without necessarily knowing what kind of object it creates or how to create it.

• Usage: - Factory method is suitable for this scenario because vehicle objects are to be created as per user demand, so to create the objects of the required hotel type and transport type during the run time interface it becomes easier.

**3. Unit Test plan**



* Getdetails ()-

The getdetails function involves giving customer details, analyzing the details booking the tickets

Given below is how a customer details are taken

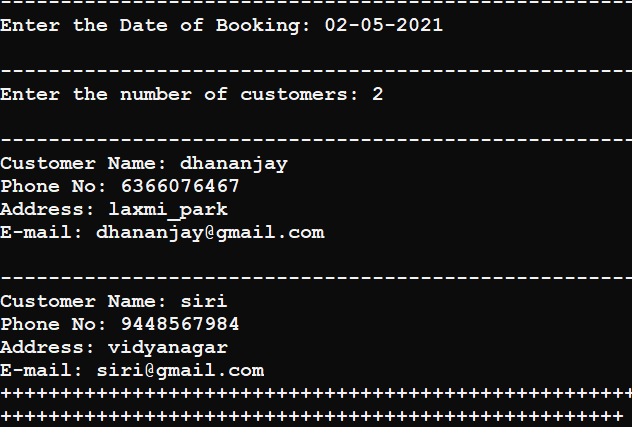
**Test case 1: Entering details of 2 customers**

**Input:**

The date of booking

Total number of customers

It takes the details of customer total customer.



Enter the Date of Booking: 02-05-2021

Enter the number of customers: 2

Customer Name: dhananjay

Phone No: 6366076467

Address: laxmi\_park

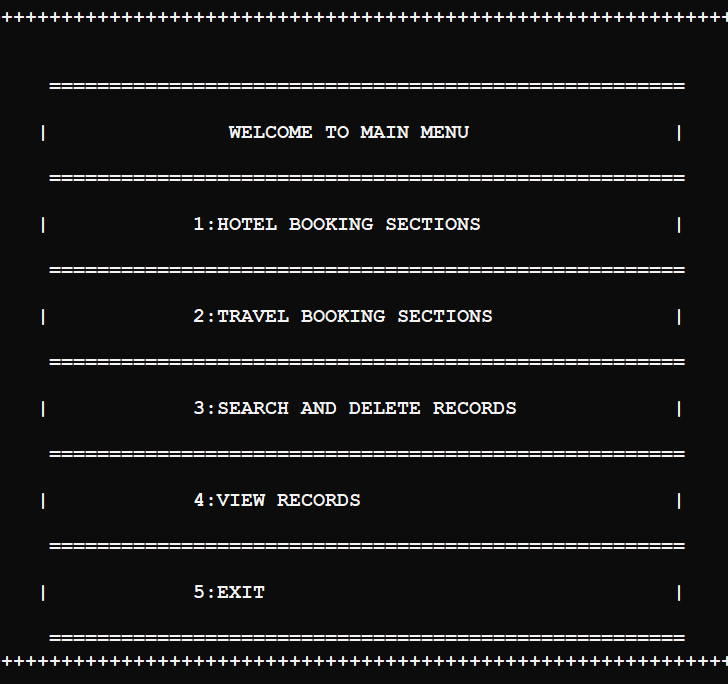
E-mail: dhananjay@gmail.com

Customer Name: siri

Phone No: 9448567984

Address: vidyanagar

E-mail: siri@gmail.com

****

***Enter the option to book***

**WELCOME TO MAIN**

**1: HOTEL BOOKING SECTIONS**

**2:TRAVEL BOOKING**

**3: SEARCH AND DELETE RECORDS**

**4: VIEW RECORDS**

**5: EXIT**

1

Please Enter Information for Hotel Reservation

Please enter the choice of your hotel......

1: 5\_Star

2: 3\_Star

3: Normal

1

PLEASE CHOOSE YOUR DESTINATION.

1: Coorg 2: Mysore

3: Chikamaglur 4: Udupi

5: Bangalore 6: Hampi

7: Murudeshwar 8: Gokarna

9: Dandeli 10: Pattadakal

Enter your choice to Reserve: 5

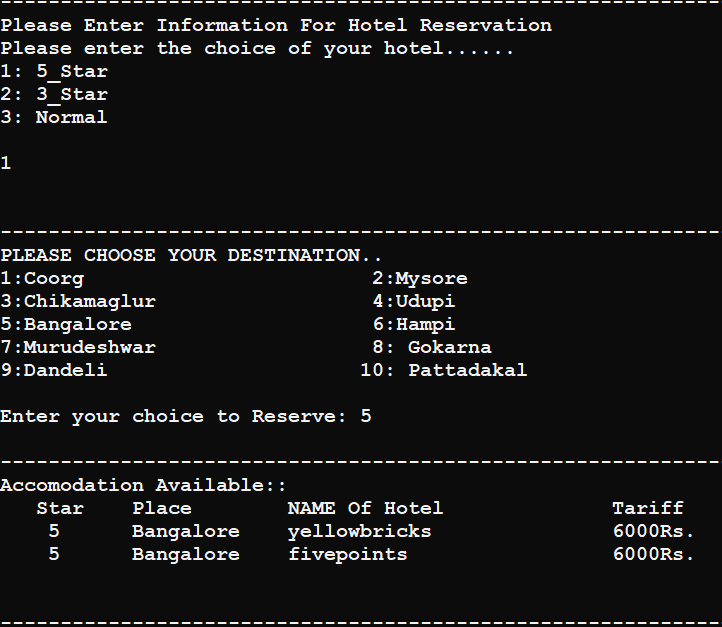
Accomodation Available:

Star Place NAME of Hotel Tariff

5 Bangalore yellowbricks 6000Rs.

5 Bangalore fivepoints 6000Rs.

Procedure of booking hotel screenshot

****

**Input:**

**------------------------------------------------------------------------------------------**

...PLEASE FILL IN THE DETAILS ...

HOTEL RESERVATION FORM

Booking Name: dhananjay

Address: laxmi\_park

Destination: Bangalore

Type of Hotel: 5Star\_Hotel

Name of Hotel: yellowbricks

Date of Check-in: 02-05-2021

No. Of Rooms: 2

No. of days to stay: 4

**Expected output:**

Please Wait while we make Reservation

Hotel Reservation

-----------------------

Name: dhananjay

Address: laxmi\_park

Type of Hotel: 5Star\_Hotel

Name of the hotel: yellowbricks, Bangalore

Date of Check-In: 2/-5/ 2021

No. of Rooms: 2

No. of Days to stay: 4

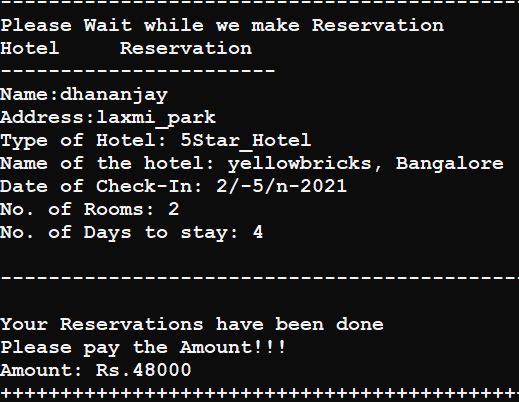
------------------------------------------------------------------------------------------

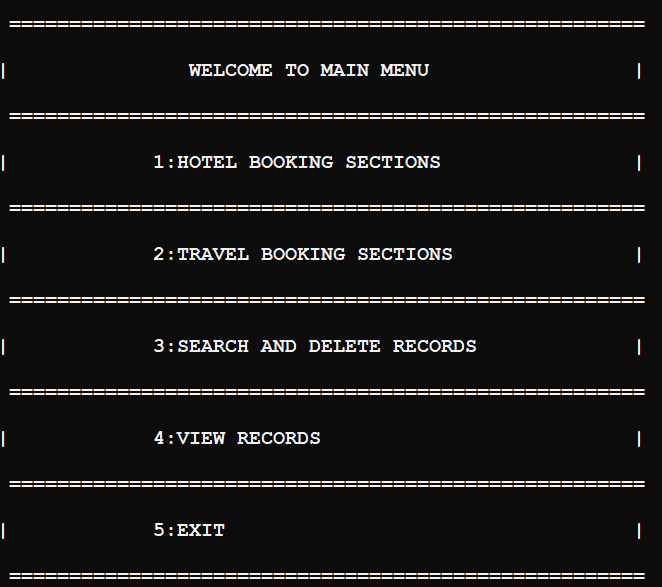
Your Reservations have been done

Please pay the Amount!!!

Amount: Rs.48000

**Actual output:**

****

****

**Input:**

2(for travel booking sections)  
------------------------------------------------------------------------------------------

Please Enter Information For Transport Reservation

Please enter the choice of your hotel......

1: Train

2: Road

2

**------------------------------------------------------------------------------------------**

PLEASE CHOOSE YOUR DESTINATION..

1: Coorg 2: Mysore

3: Chikamaglur 4: Udupi

5: Bangalore 6: Hampi

7: Murudeshwar 8: Gokarna

9: Dandeli 10: Pattadakal

Enter your choice to Reserve: 5

------------------------------------------------------------------------------------------

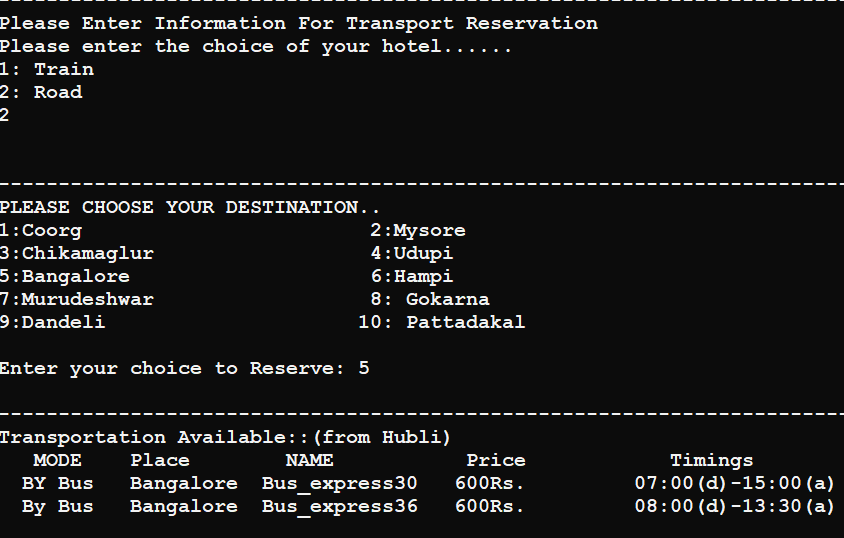
Transportation Available: :(from Hubli)

MODE Place NAME Price Timings

BY Bus Bangalore Bus\_express30 600Rs. 07:00(d)-15:00(a)

By Bus Bangalore Bus\_express36 600Rs. 08:00(d)-13:30(a)

**Procedure of booking screenshot**

****

**Input:**

**------------------------------------------------------------------------------------------**

...PLEASE FILL IN THE DETAILS ...

BUS RESERVATION FORM

Booking Name: dhananjay

Address: laxmi\_park

Mode of Transport: Road

Bus Name: Bus\_express30

Date of Travel: 02-05-2021

Destination: Bangalore

No. of Tickets: 2

**Expected output**

------------------------------------------------------------------------------------------

Please Wait while we make Reservation

Hotel Reservation

-----------------------

Name: dhananjay

taddress: laxmi\_park

Mode of Transport: Road

Bus name: Bus\_express30, Bangalore

Date of Travel: 2/-5/-2021

No. of Tickets: 2

**------------------------------------------------------------------------------------------**

Your Reservations have been done

Please pay the Amount!!!

Amount: Rs.1200

**Input:**

++++++++++++++++++++++++++++++++++++++++++++++++++++

4

VIEW RECORDS

1.VIEW HOTEL RECORDS

2.VIEW TRAVEL RECORDS

3: EXIT

2

TRAVEL RECORDS

dhananjay laxmi\_park Road Bus\_express30, Bangalore 2/-5/-2021 2 1200

VIEW RECORDS

1.VIEW HOTEL RECORDS

2.VIEW TRAVEL RECORDS

3: EXIT

1

HOTEL RECORDS

dhananjay laxmi\_park 5Star\_Hotel yellowbricks, Bangalore 2/-5/-2021 2 4 48000

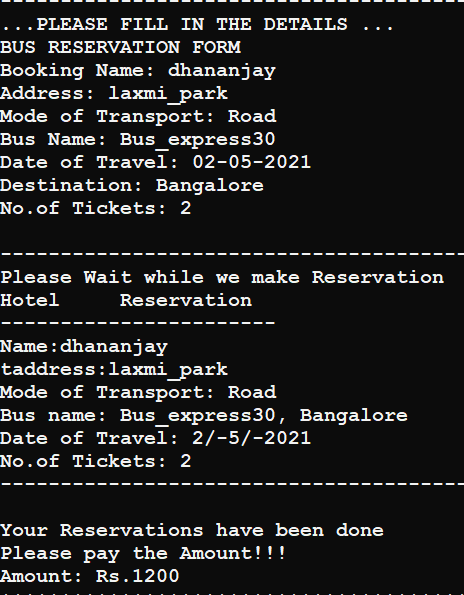
VIEW RECORDS

1.VIEW HOTEL RECORDS

2.VIEW TRAVEL RECORDS

3: EXIT

**Actual output**

****

Booking of hotel and transport has been done in output amount to be paid will be displayed.

**For searching and deleting the records**

We have member functions in booking class to searchhoteldetails, searchtraveldetails and deletehoteldetails, deletetraveldetails from the file.

**Input :**

**3**

****

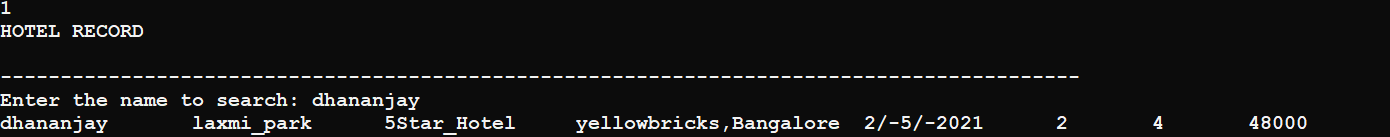
****

**Input:**

**1**

**1.Hotel records**

**Actual output:**

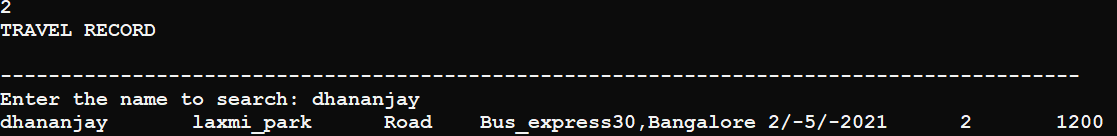
****

**Input:**

**2**

**Actual output:**

**2.Travel records**

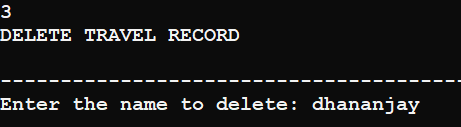
****

**Input:**

**3**

**Actual output:**

**3.Deleting travel records**

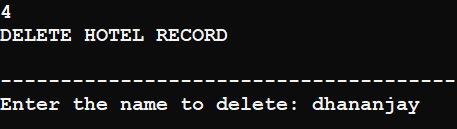
****

**Input:**

**4**

**Actual output:**

**4.Deleting hotel records**

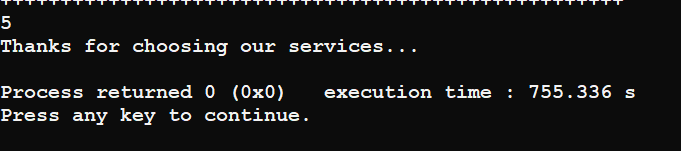
****

**Input:**

**5**

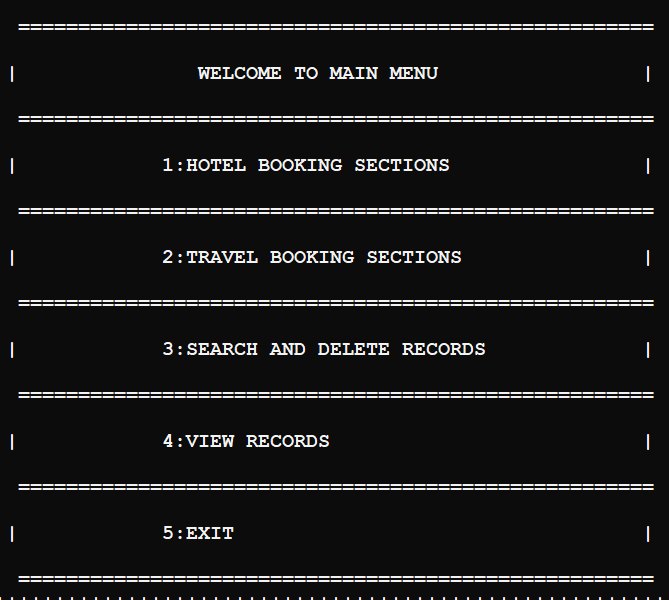
**When we press option 5**

It exits the program

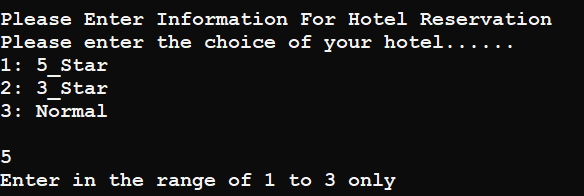
****

**4.Results**

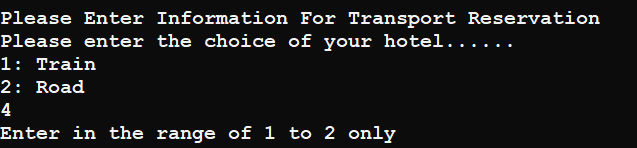
**1. Main menu**

****

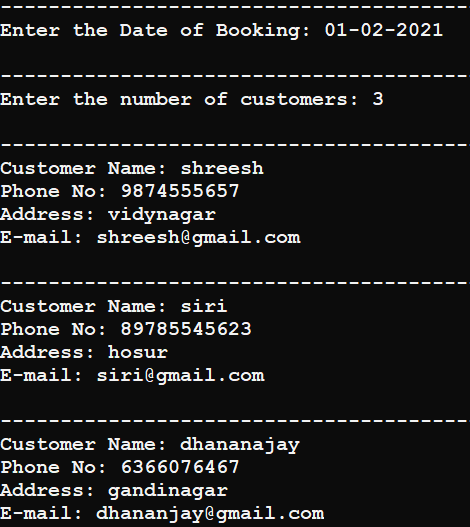
***1.Exception handling in hotel*: -if user enters the different choice it throws the exception handling error.**



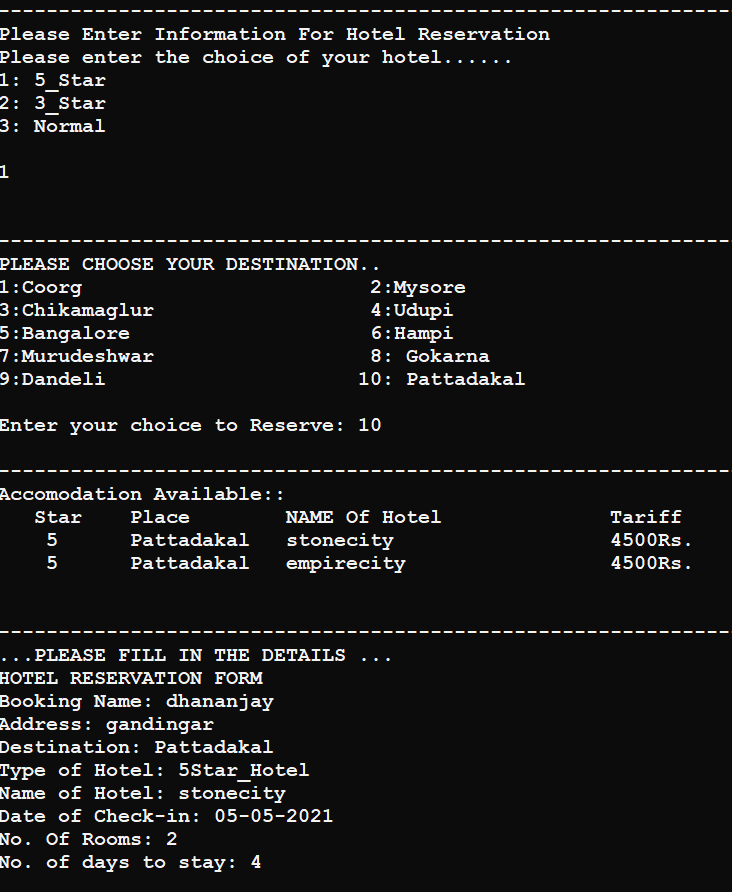
***2.Exception handling in transport*:- if user enters the different choice it throws the exception handling error .**



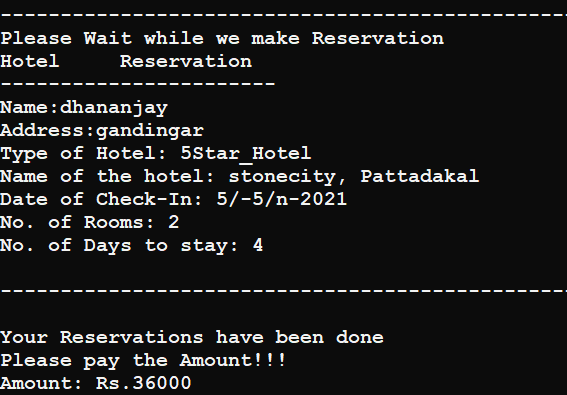
**3.input of 3 customers**



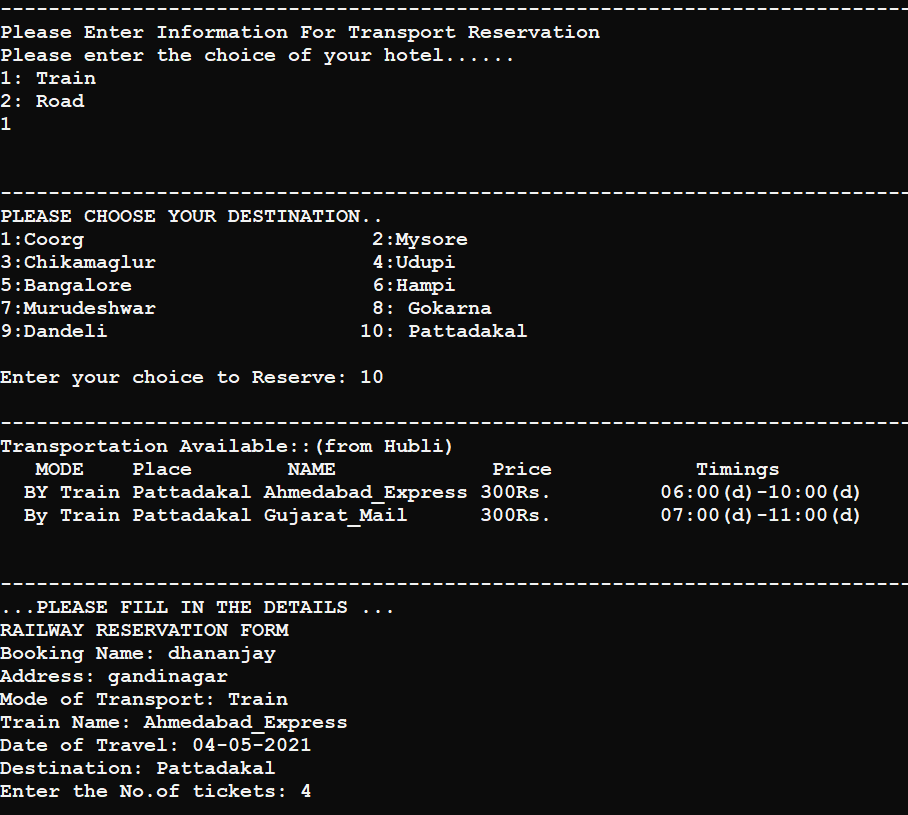
**4.booking hotel**



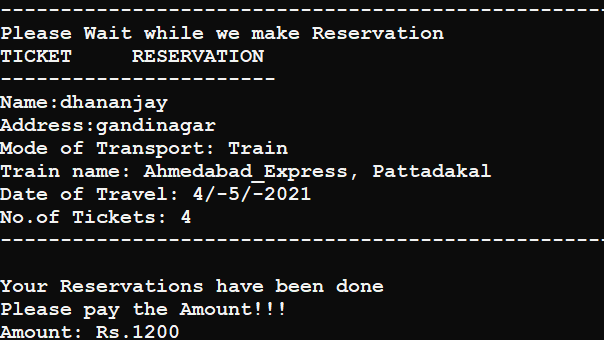
1. **Reservation of hotel and amount to be paid**



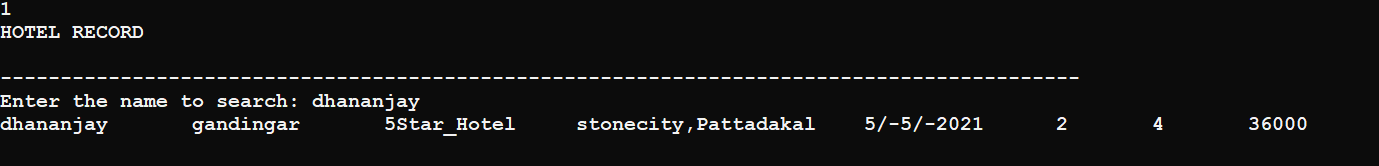
**6.booking of transport type**



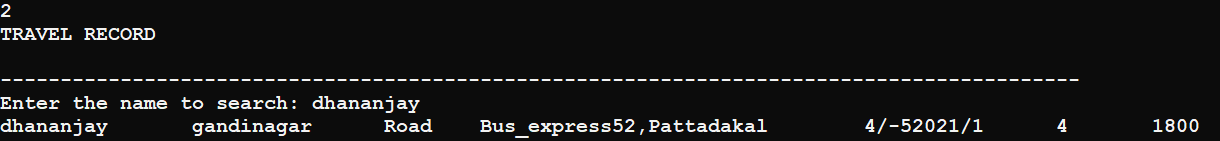
**7.reservation of transport and amount to be paid**



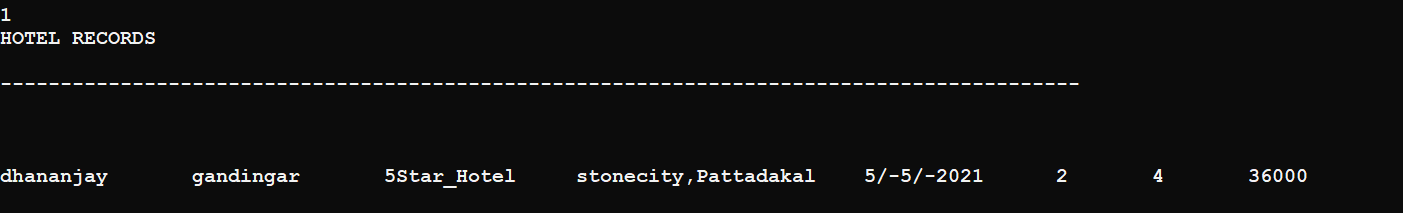
**Output:hotel record**

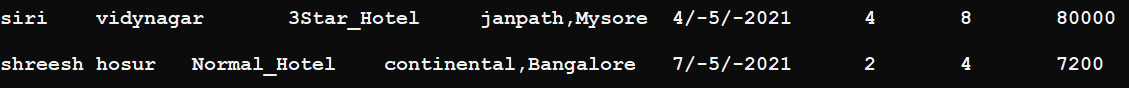


**Output : travel record**

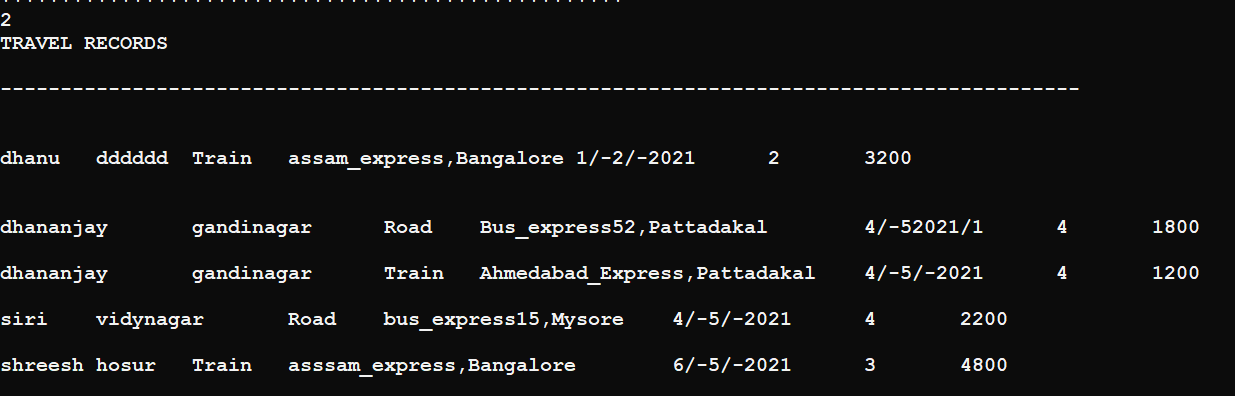


**Output : view hotel records**

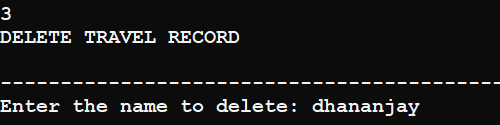


****

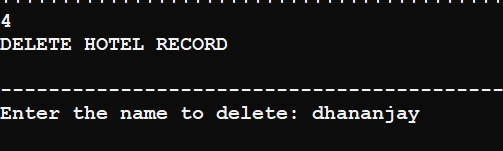
**Output : view travel records**



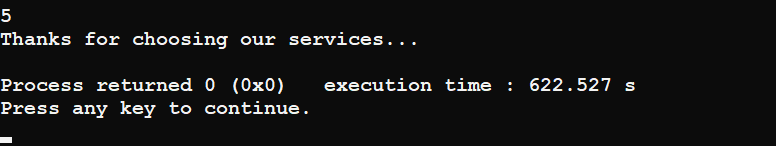
**Output:delete travel records**



**Output : delete hotel records**



**Exit**



|  |
| --- |
| sdsfdsThe Assignprice function is overridden from hotel class to all three class.it assign different price at location for different hotels. |
| **wwwwwww+ rt:int** |
| **+ Assignprice():void**  **+ ShowTransport(string):void**  **+ confirmTBooking():void**  **+ CreateTravelfile():void** |
| **Road** |
| **+ rt:int** |
| **+ Assignprice():void**  **+ ShowTransport(string):void**  **+ confirmTBooking():void**  **+ CreateTravelfile():void** |
| **Road** |
| **+ rt:int** |
| **+ Assignprice():void**  **+ ShowTransport(string):void**  **+ confirmTBooking():void**  **+ CreateTravelfile():void** |
| **Road** |
| **+ rt:int** |
| **+ Assignprice():void**  **+ ShowTransport(string):void**  **+ confirmTBooking():void**  **+ CreateTravelfile():void** |

|  |
| --- |
| **Road** |
| **+ rt:int** |
| **+ Assignprice():void**  **+ ShowTransport(string):void**  **+ confirmTBooking():void**  **+ CreateTravelfile():void** |
| **Road** |
| **+ rt:int** |
| **+ Assignprice():void**  **+ ShowTransport(string):void**  **+ confirmTBooking():void**  **+ CreateTravelfile():void** |