KARNATAK LAW SOCIETY’S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

**(APPROVED BY AICTE, NEW DELHI)**



*Course Project Report*

*On*

***HOTEL MANAGEMENT SYSTEM***

*Submitted in the partial fulfillment for the academic requirement**of*

***3rd Semester B.E.***

***In***

***Computer Science Engineering***

***Submitted by***

**NAME USN**

Harshita Nimbal2GI19CS049

Girija Kademani2GI19CS044

Irfan Kamate 2GI19CS052

Deepak Kumble2GI19CS038

**2020 – 2021(Odd Semester)**

**TABLE OF CONTENTS**

**Chapter 1: Introduction**

1.1) Problem Definition

1.2) Abstract

**Chapter 2: Design Details**

2.1) Initial class diagram

2.2) Final class diagram

**Chapter 3: Implementation**

3.1) Source code

3.2) Screenshots of output

**Conclusions**

**References**

**CHAPTER 1: INTRODUCTION**

**Problem Definition:**

To design a hotel service management software which can be used to manage activities like -storing customer details, booking rooms of four different types and ordering food for particular rooms, unbooking rooms and showing the bill. It can also be used to see different room features and room availability. It is a menu driven program and it runs until the user exits.

**Motivations to select this topic for course project:**

* The primary goal of our hotel management software is to take down the disadvantages of the manual handling of hotel operations.
* To learn proper UML notations for drawing class diagrams which help in the development of the application.
* Exploring the usefulness of the open source software Dia in drawing class diagrams using UML notations.

**Abstract of the project:**

**Functional requirements of the project:**

1. Allows the customer to book a room from 4 different types of rooms
2. Customers can order food for a their rooms
3. Checking for availability of a room
4. Displaying features of the room which the customers want to book
5. Unbooking the rooms
6. Giving the bill after the customers check out of the hotel

**List of Classes in the application domain**

Food, Single Room, Double Room, Not Available, Room, Hotel, Write and Hotel System

This is a Hotel Management tool which can be used to manage activities like storing customer details, booking rooms of four different types, ordering food for particular rooms, unbooking rooms and showing the bill. It can also be used to see different room features and room availability. It is a menu driven program and it runs until the user exits. File handling has been used to store the current status of the hotel (customer details, booked rooms, food ordered) in a file once the program exits so that when we restart the program, the old details are not lost. The program reads the file when it restarts to know the previous status of the hotel. Writing of file has been done in a separate thread as it can be done parallel. User defined exception is thrown if the user tries to book an already allotted room. Exception handling is properly done to deal with any kind of unexpected exception.

**Topics Covered-**

Classes and Objects, Inheritance, File Handling with Objects, Array List, implementing Interface, user defined exception and Exception handling.

**Advantages of Hotel Management System:**

In the modern, digital age, hotel software is used for a range of different purposes, but convenience, efficiency, accuracy and financial benefits are chief among them. Below, you will find five specific advantages associated with the use of hotel management software solutions within a property.

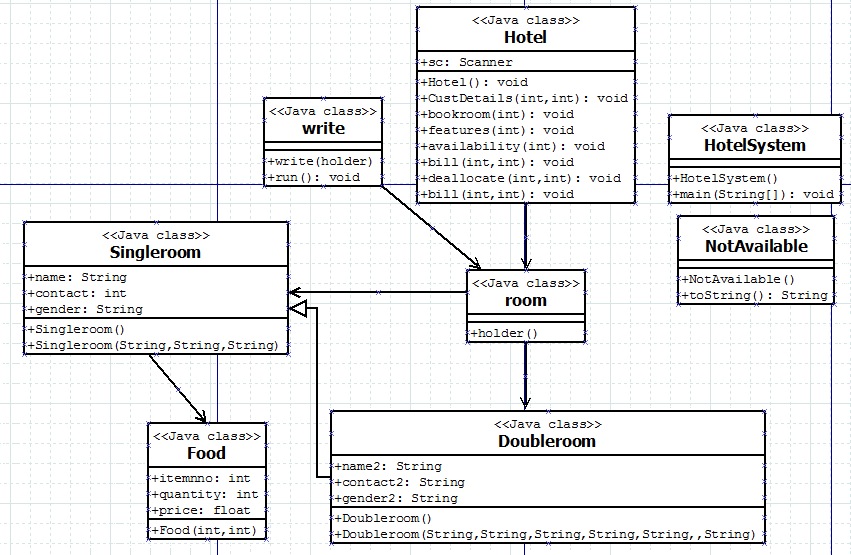
1. **Streamlined Processes:** A major benefit that can be gained from adopting a variety of hotel management software solutions is the ability to streamline processes. From automating many of the distribution tasks, to generating financial reports in an instant, software applications can make almost everything more efficient.
2. **Reduced Costs:** Although hotel software requires an initial financial investment, over a longer time period these applications and systems can lead to significant cost-savings. This is because less human involvement is required in many day-to-day processes, while money can also be saved in a number of other areas.
3. **Labour costs** can be reduced through automation, which can allow you to reduce the number of staff you employ, reduce the number of hours they work, or simply gain from increased productivity in other areas. Review management software can boost your reputation, saving on marketing costs, while channel managers can save on costs linked to distribution.
4. **Increased Revenue:** In addition to helping you to save money, hotel management software can also help to increase the amount of revenue you generate. For instance, revenue management systems can allow you to optimise pricing and distribution, using past data, existing data in the books and wider industry data to help you to accurately anticipate demand.
5. **Improved Customer Experience:** Another reason why hotel management software can be beneficial is because it can significantly enhance the customer experience you offer. One of the most obvious examples of this is the use of mobile hotel check-in and check-out systems, which allow for much more seamless arrivals and departures.
6. **Competitive Advantage:** Finally, one of the single most convincing reasons why you should utilise hotel management software solutions is the potential to gain a competitive advantage over rivals. Many software packages will provide valuable real-time industry insights, allowing you to be more competitive in terms of room rates, or package discounts.
7. Furthermore, hotel software can help you to build better relationships with customers, attract more positive feedback, generate customer loyalty and save money that could be spent on other areas that will boost your performance

**Disadvantages of Hotel Management System:**

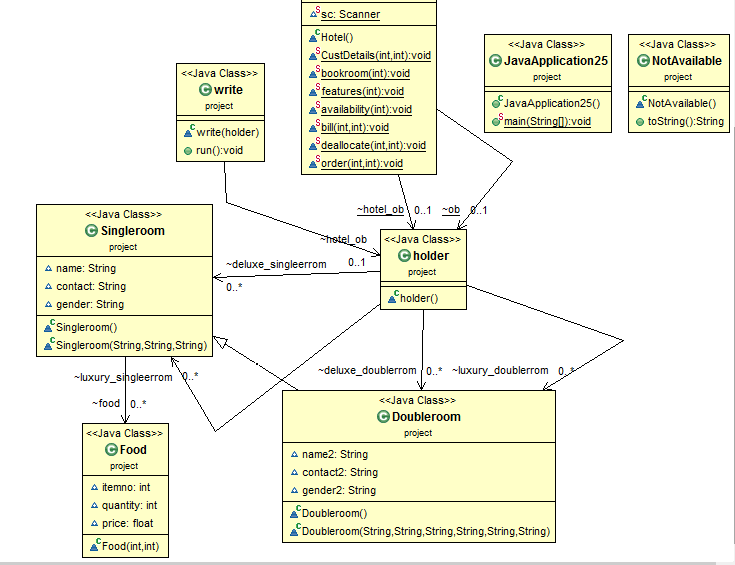
1. All the staff needs to be trained on the software
2. The software needs to be renewed each year
3. High risk of virus attack as the systems are connected to the internet most of the time
4. If there is a virus attack the stored information might get corrupt
5. If there is a power failure, the hotel runs a high risk of losing all the stored information
6. Most cities / countries do not have local support. Support is possible via online only and sometimes it takes time.

**CHAPTER 2: DESIGN DETAILS**

1. **Initial Class Diagram**



1. **Final Class Diagram**

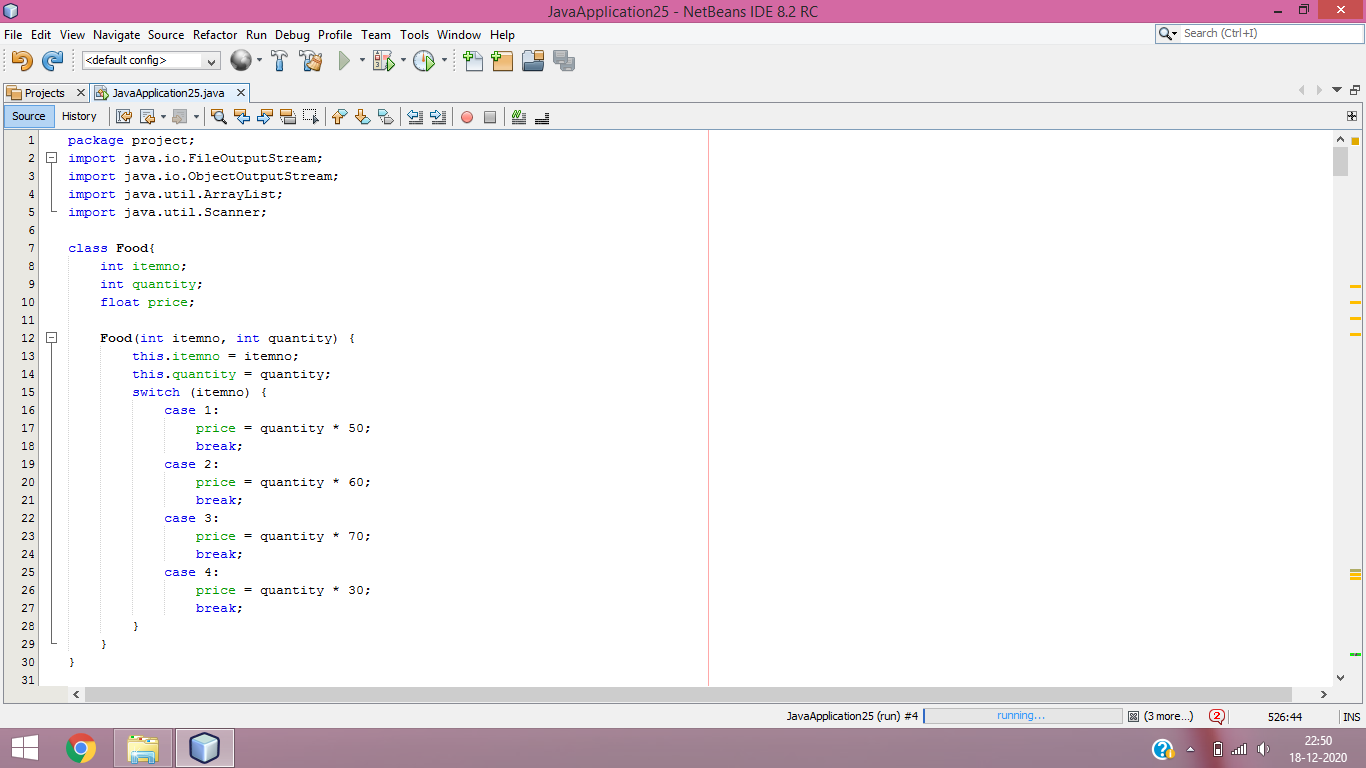


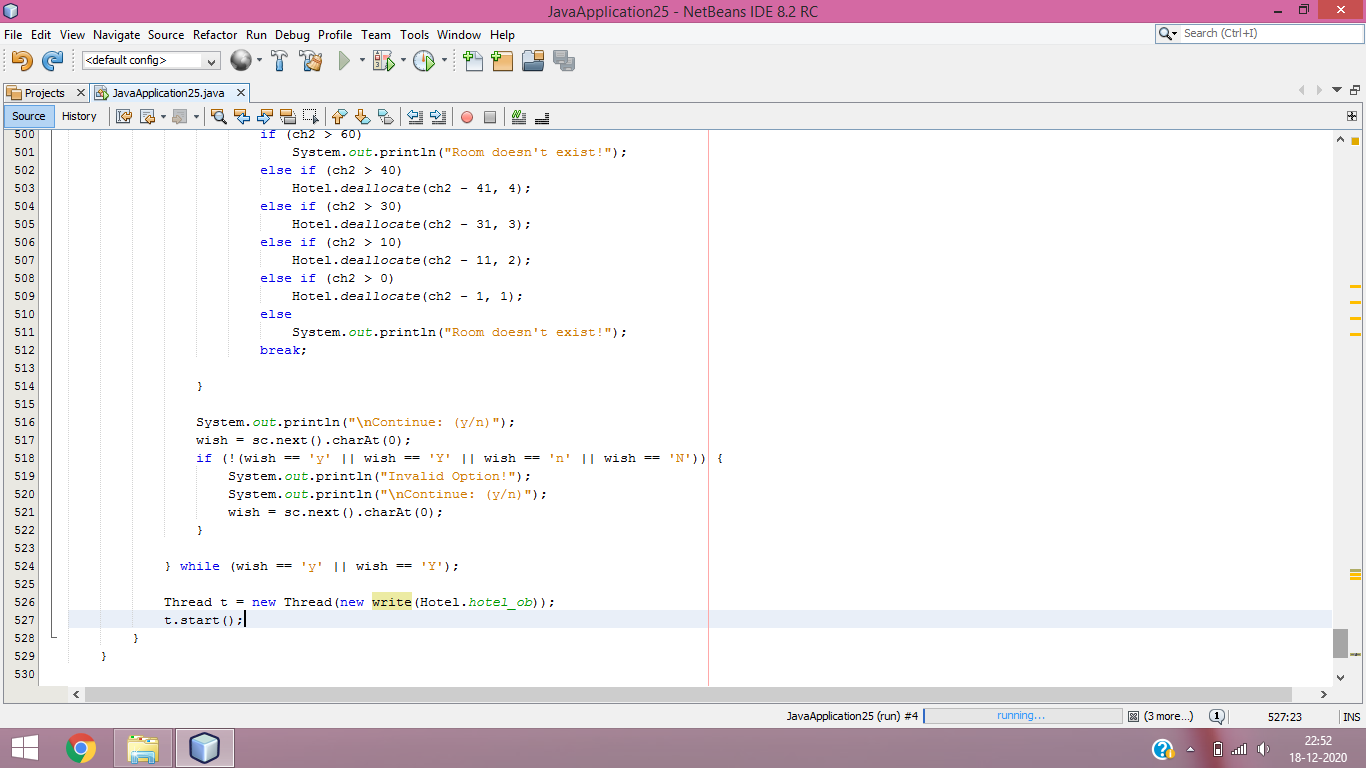
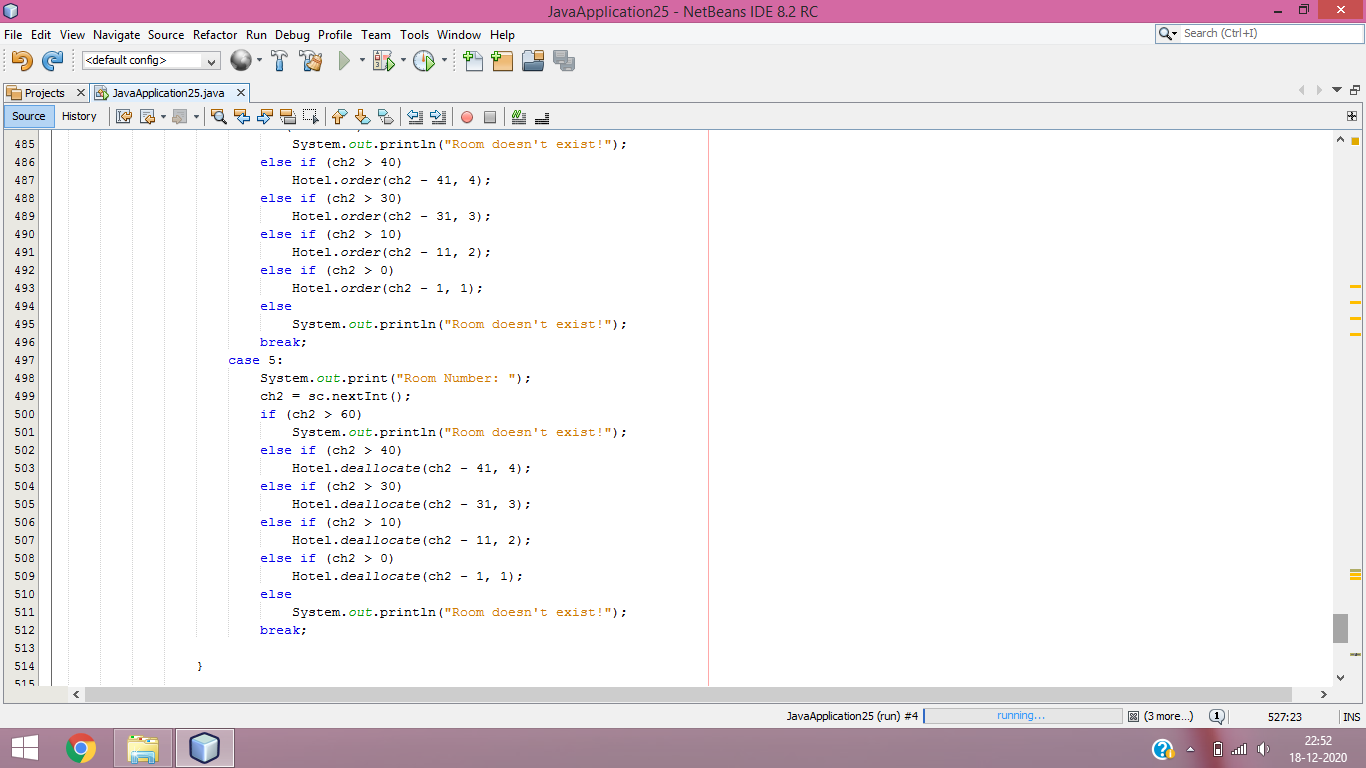
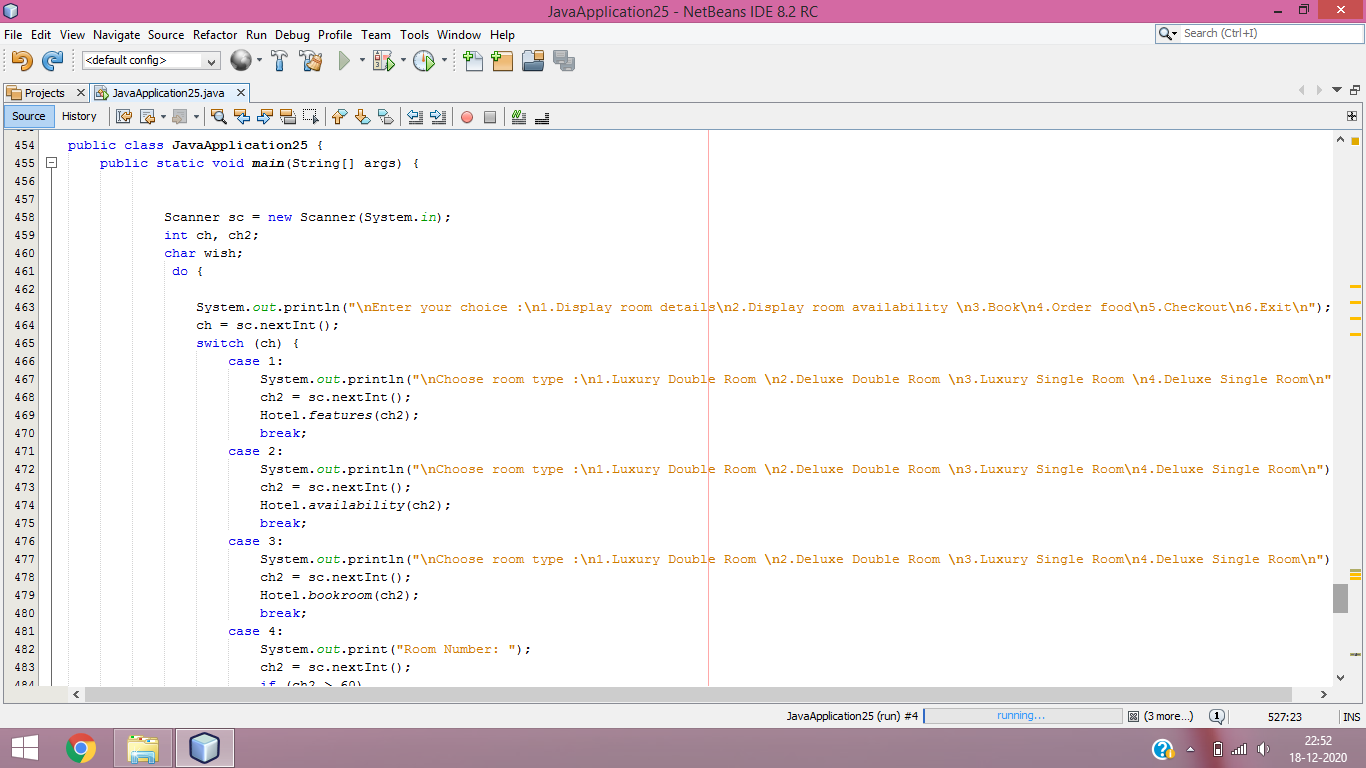
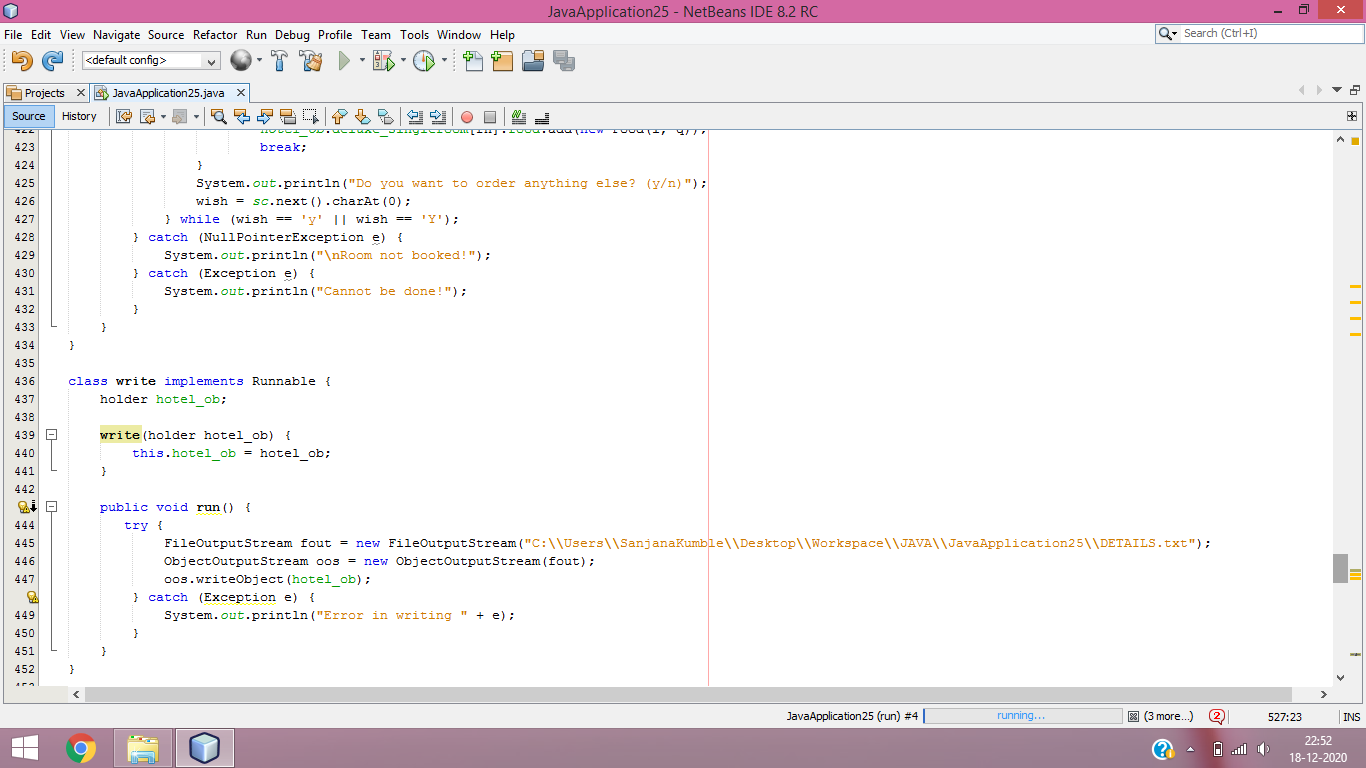
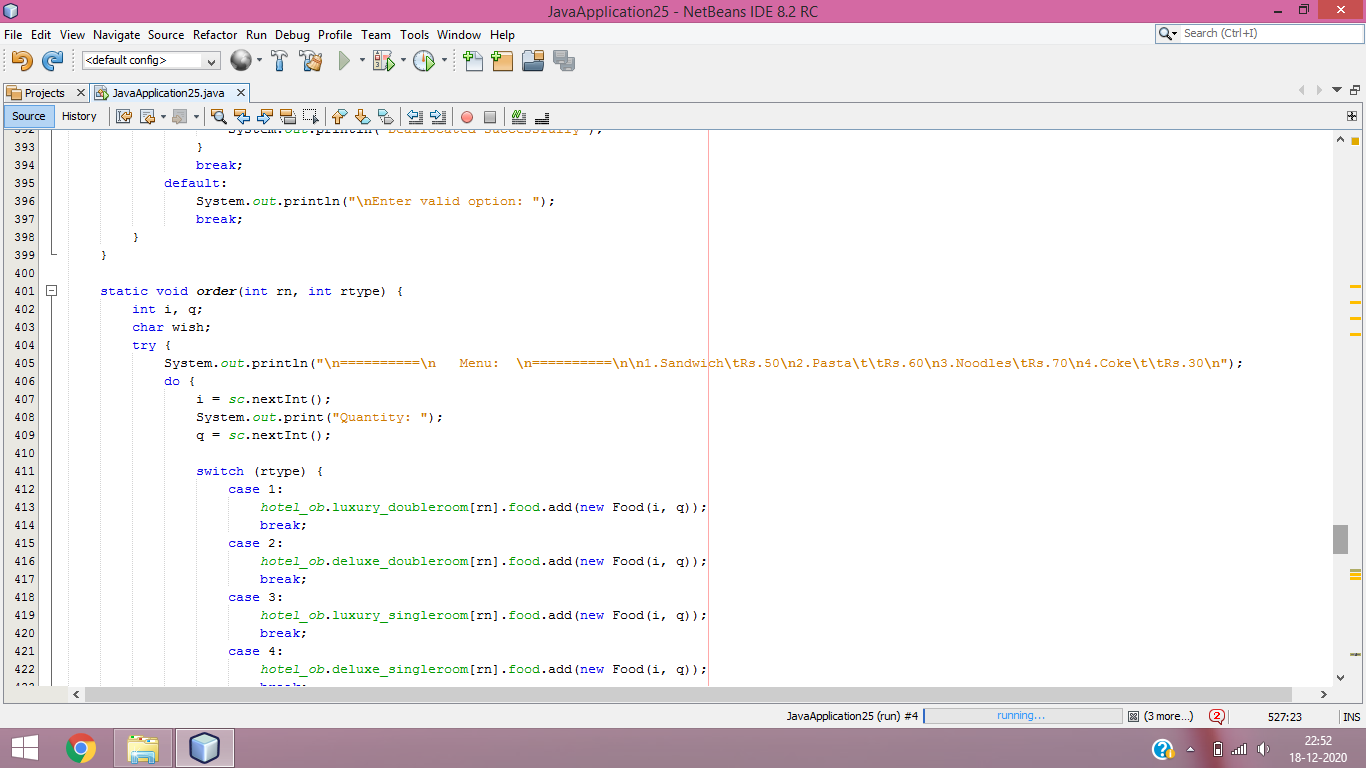
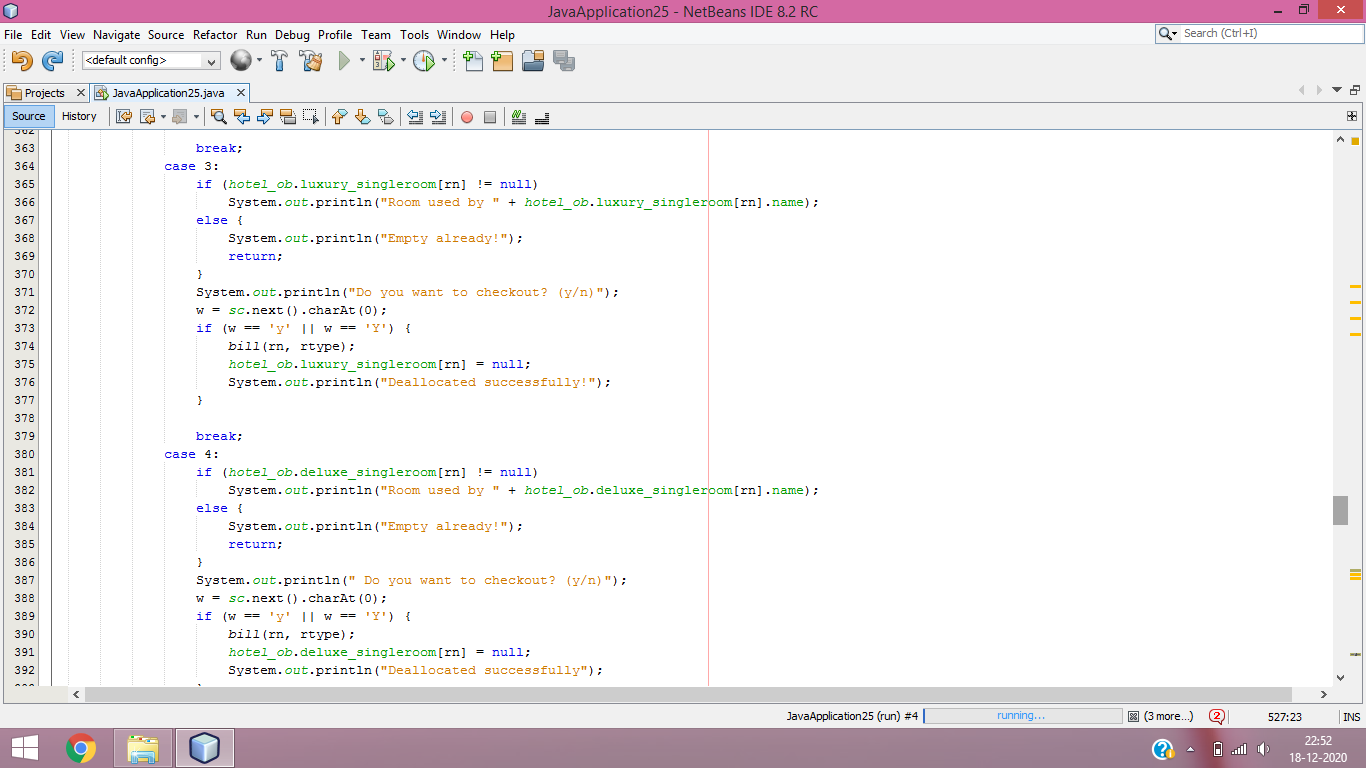
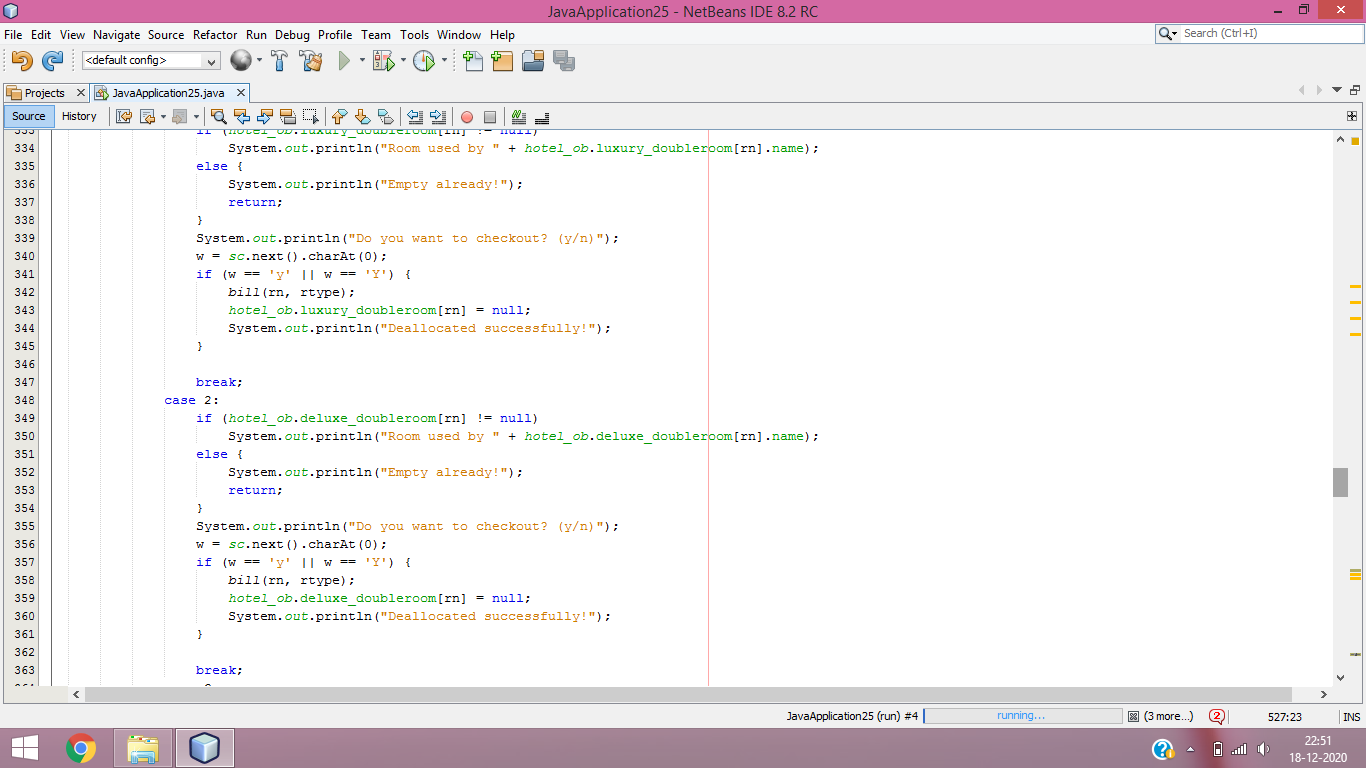
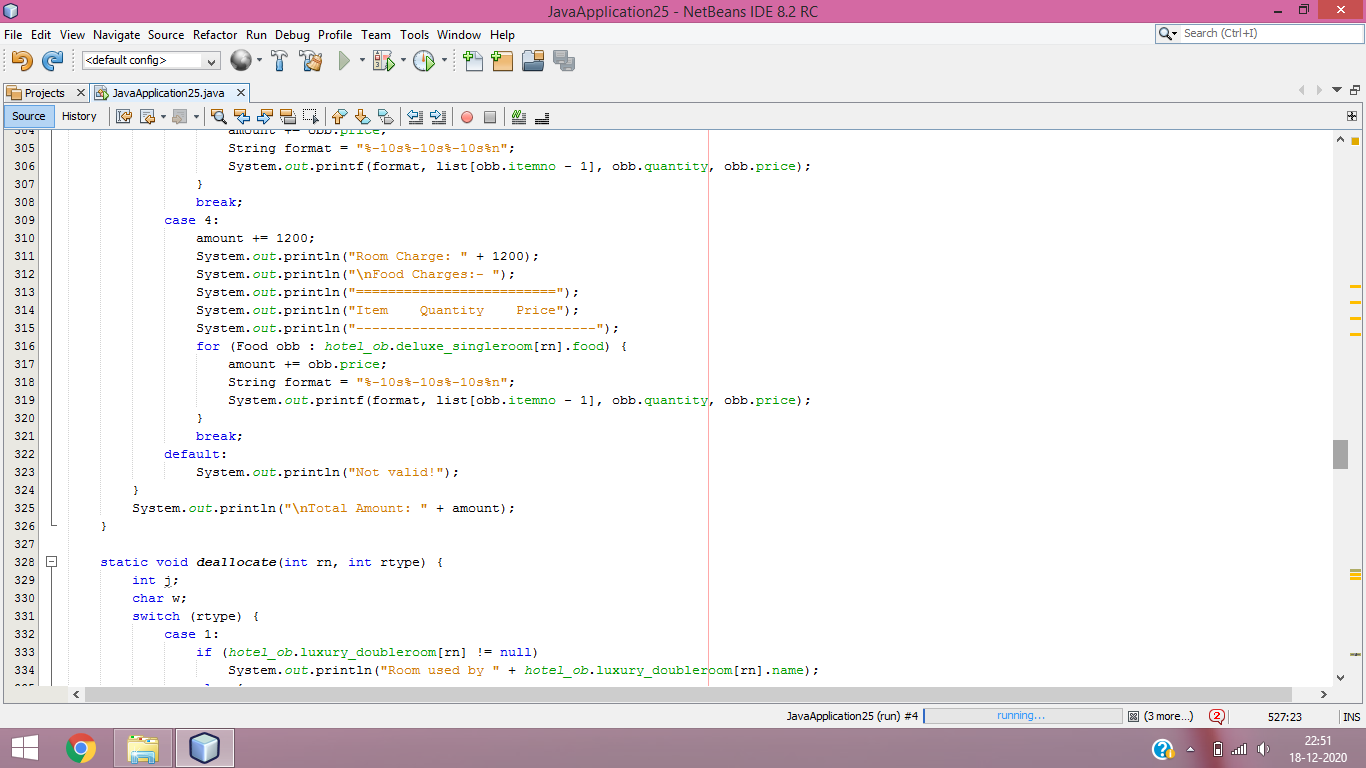
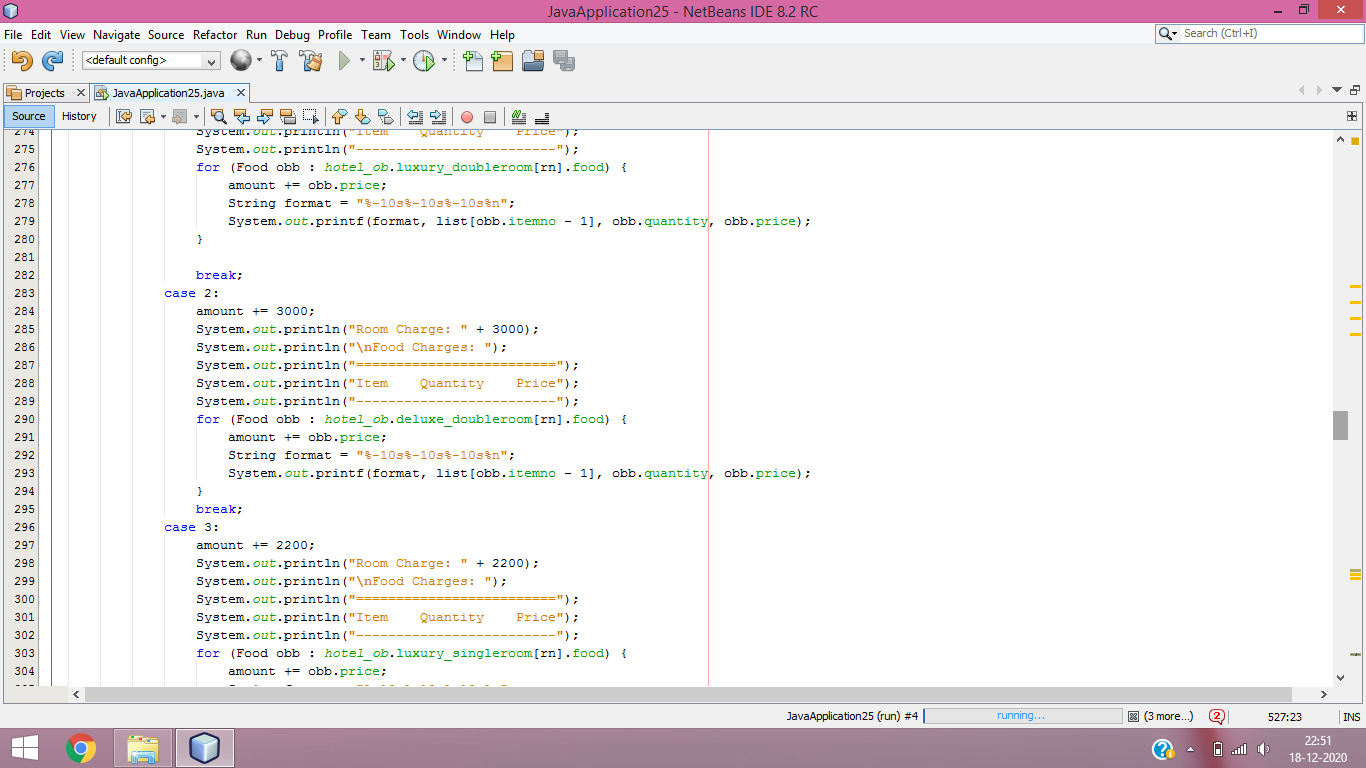
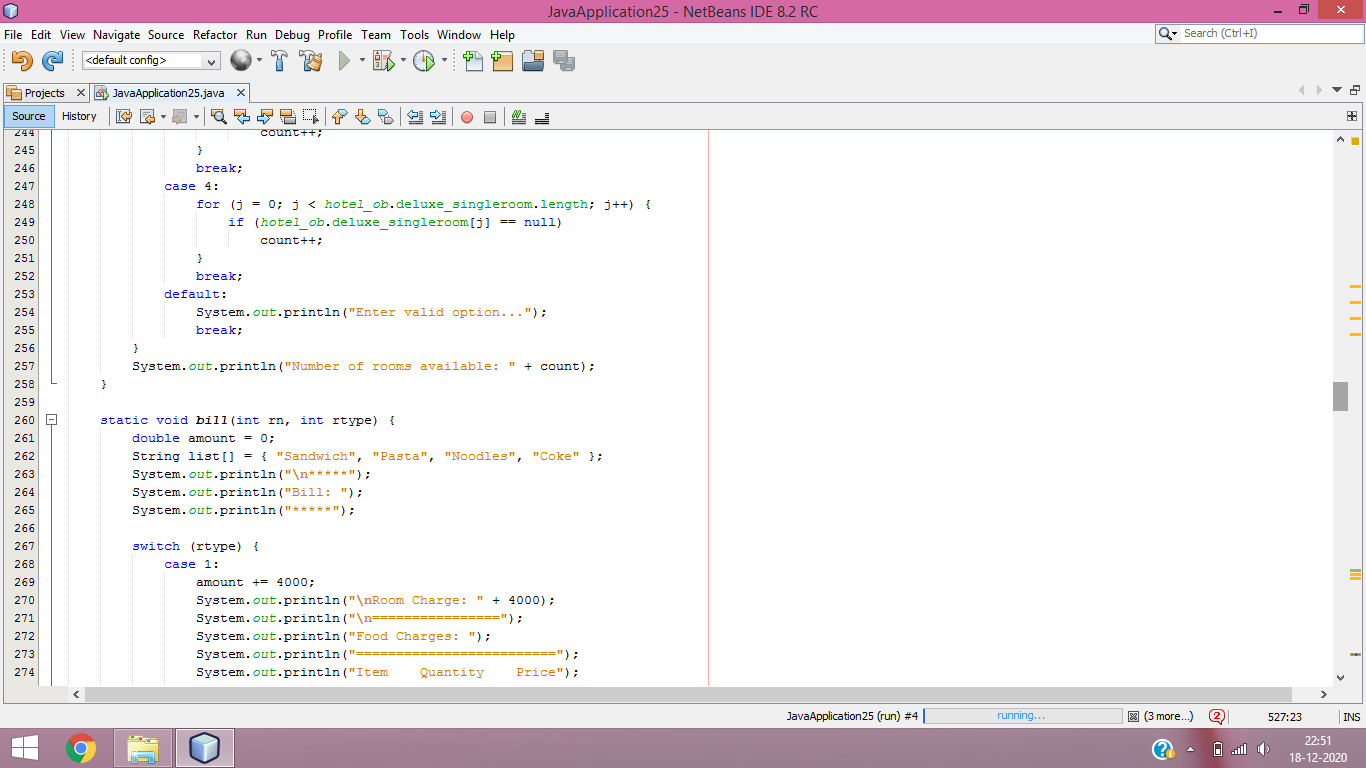
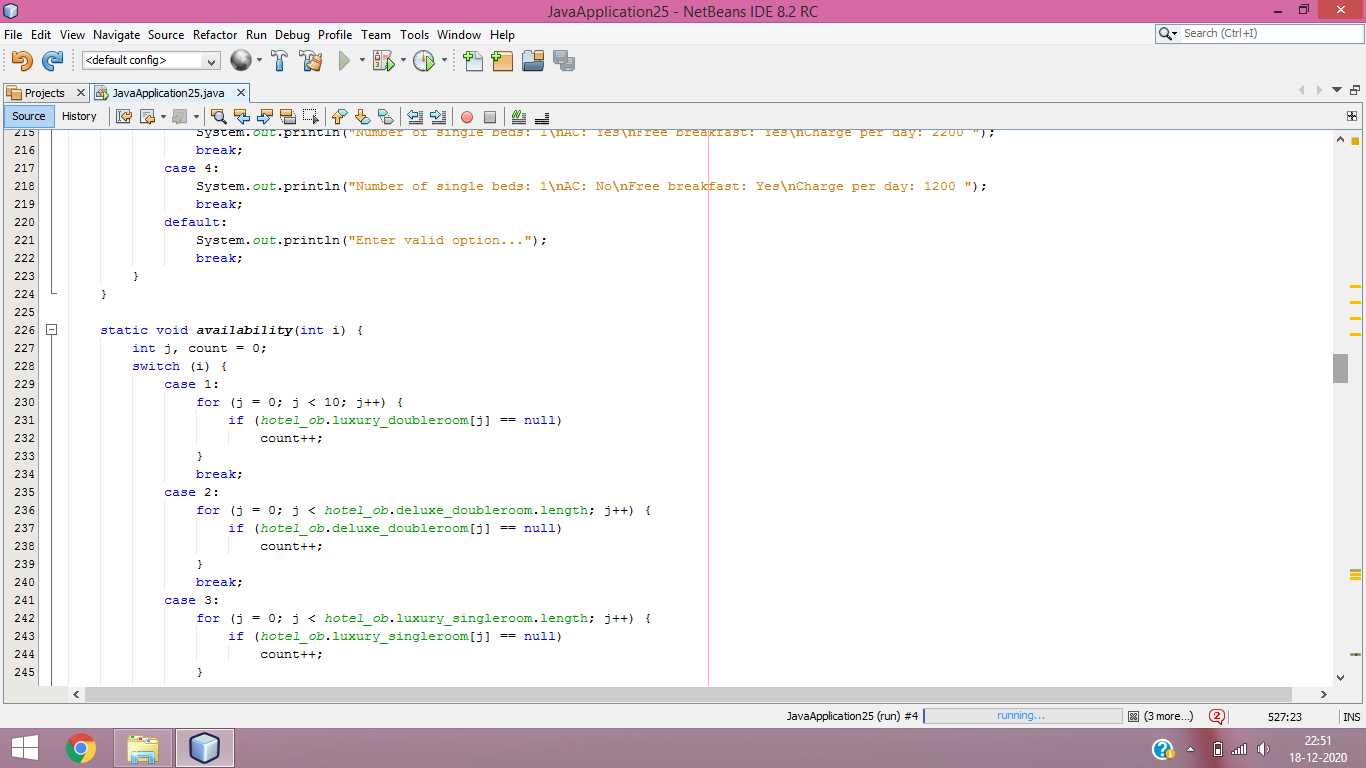
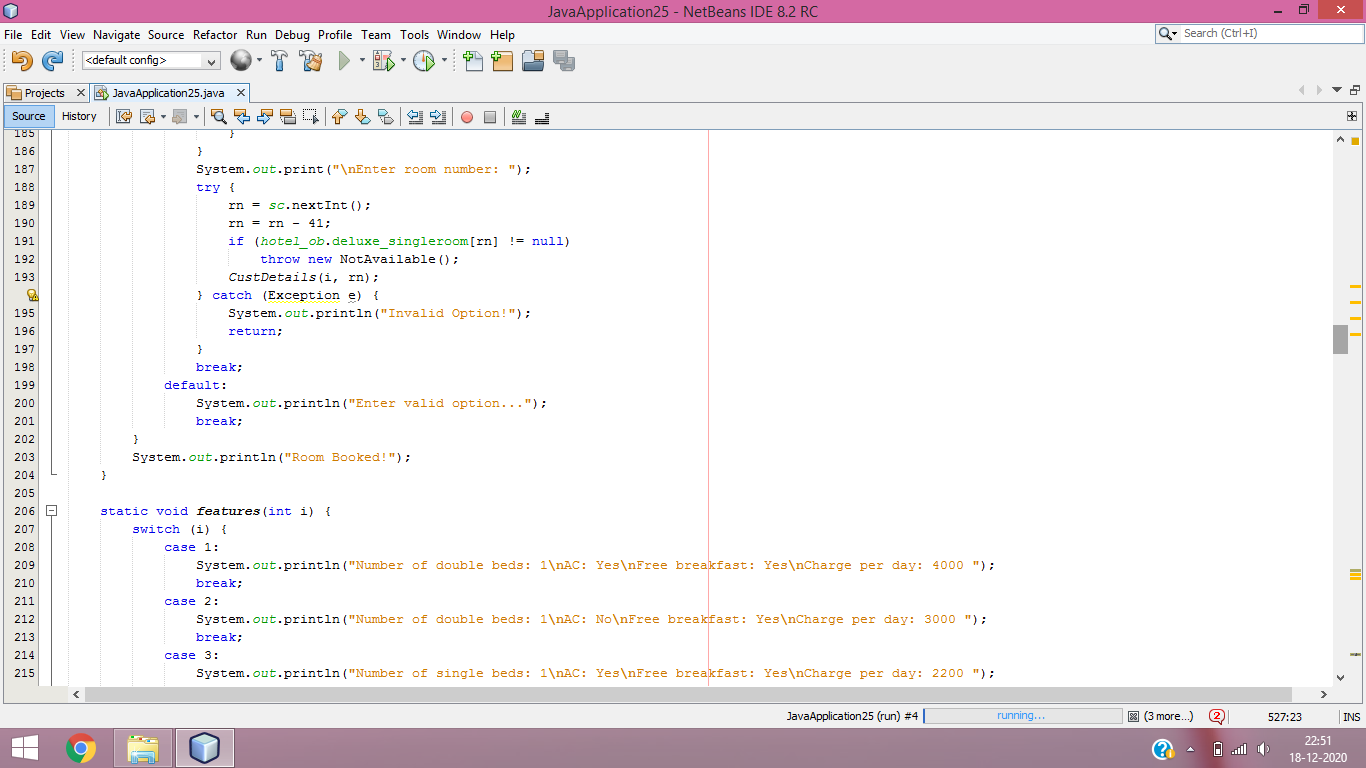
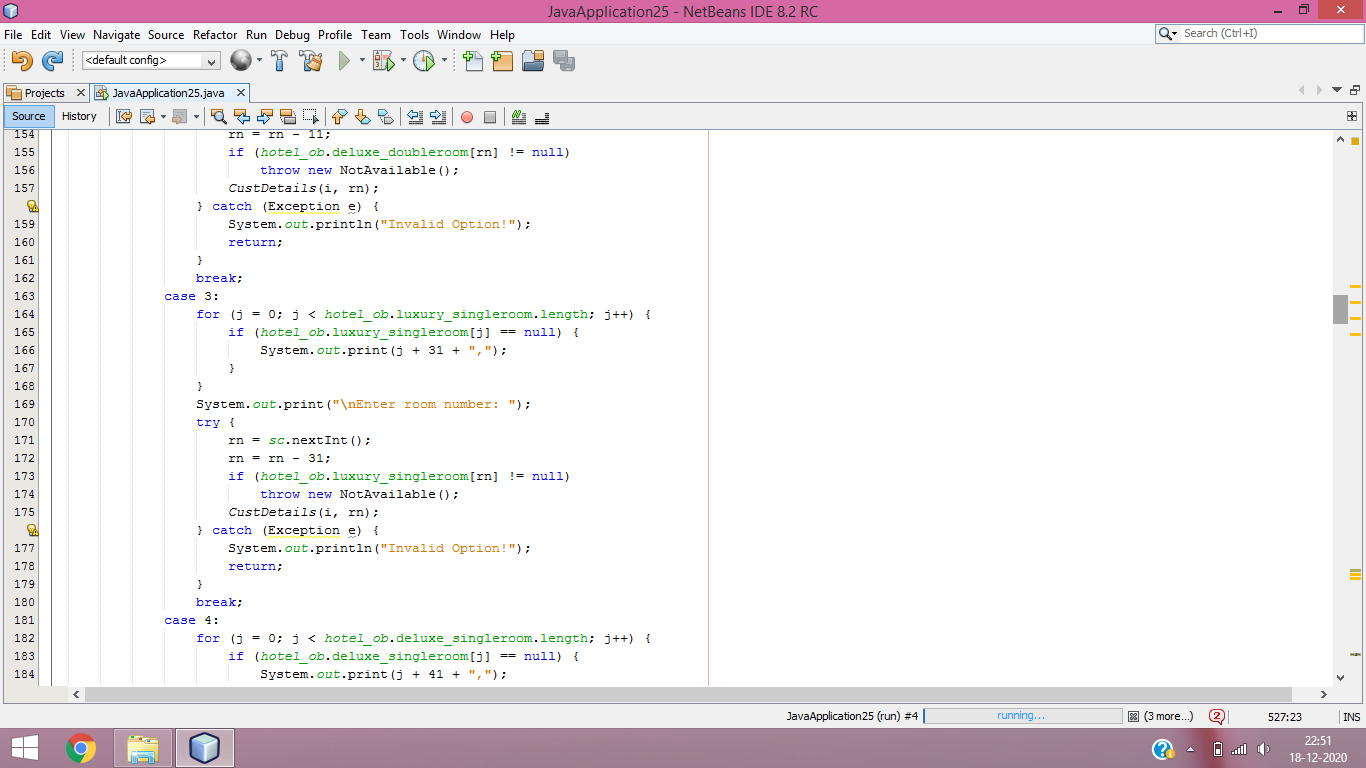
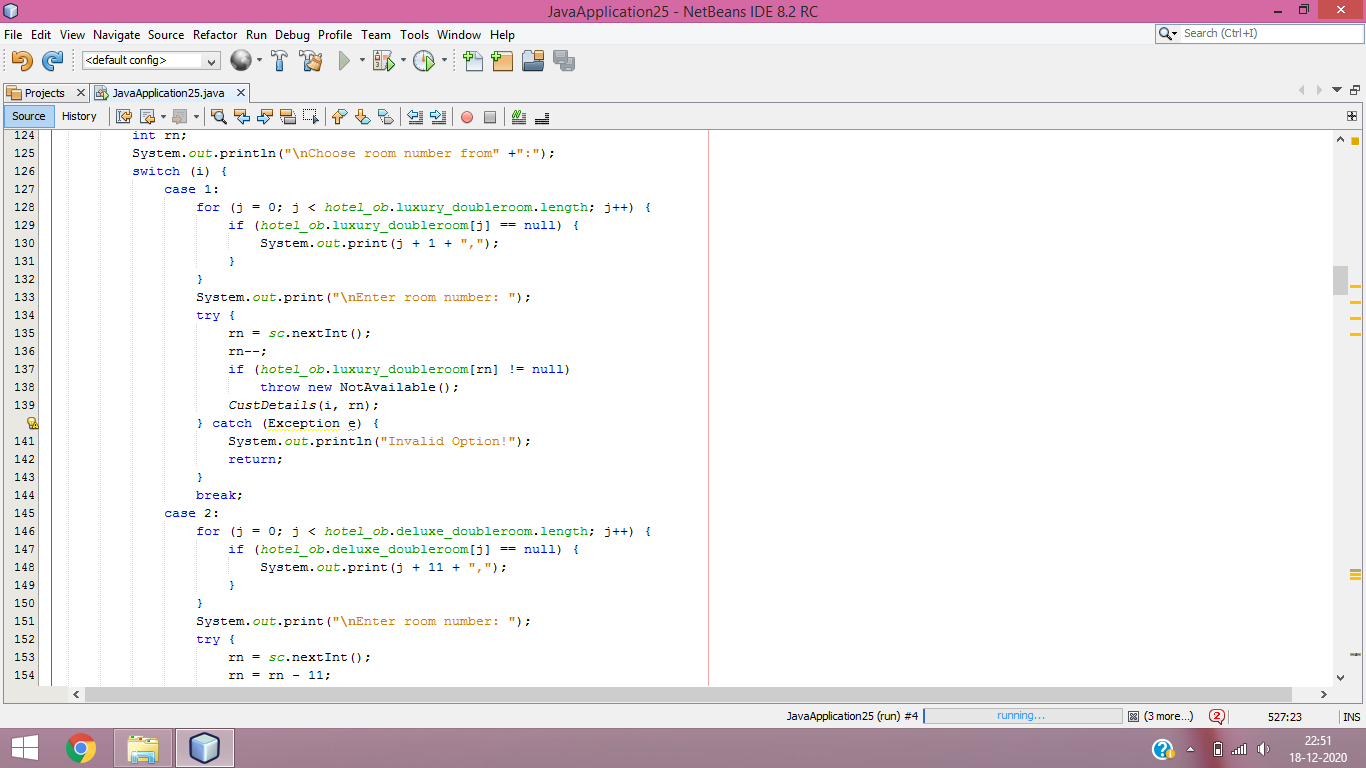
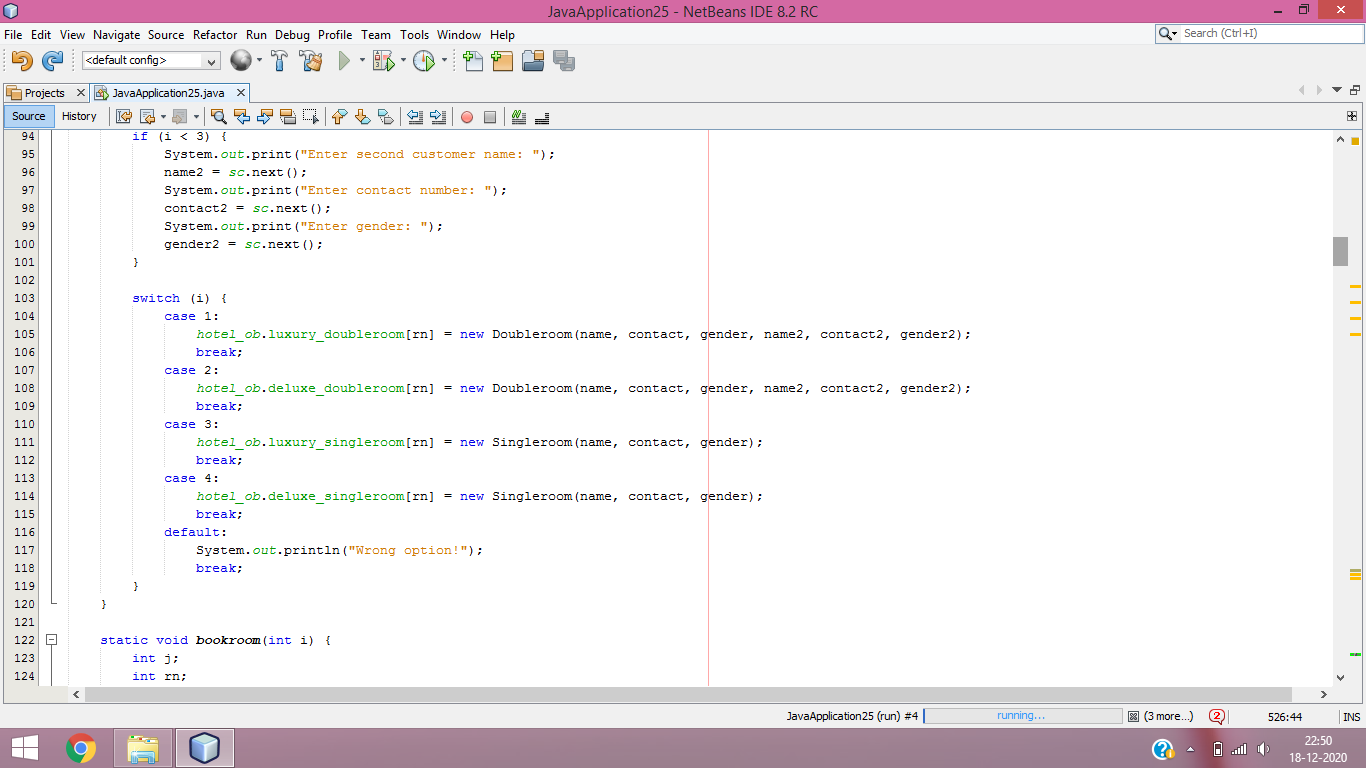
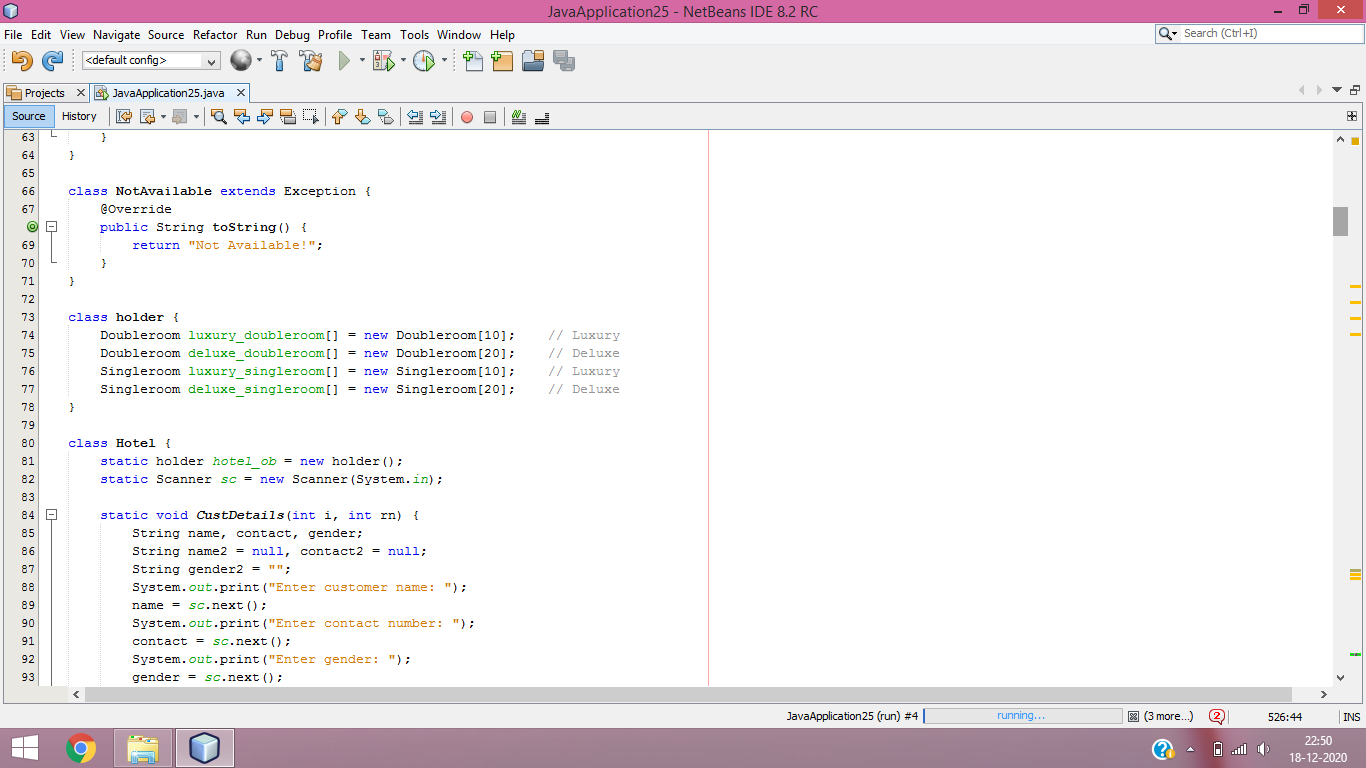
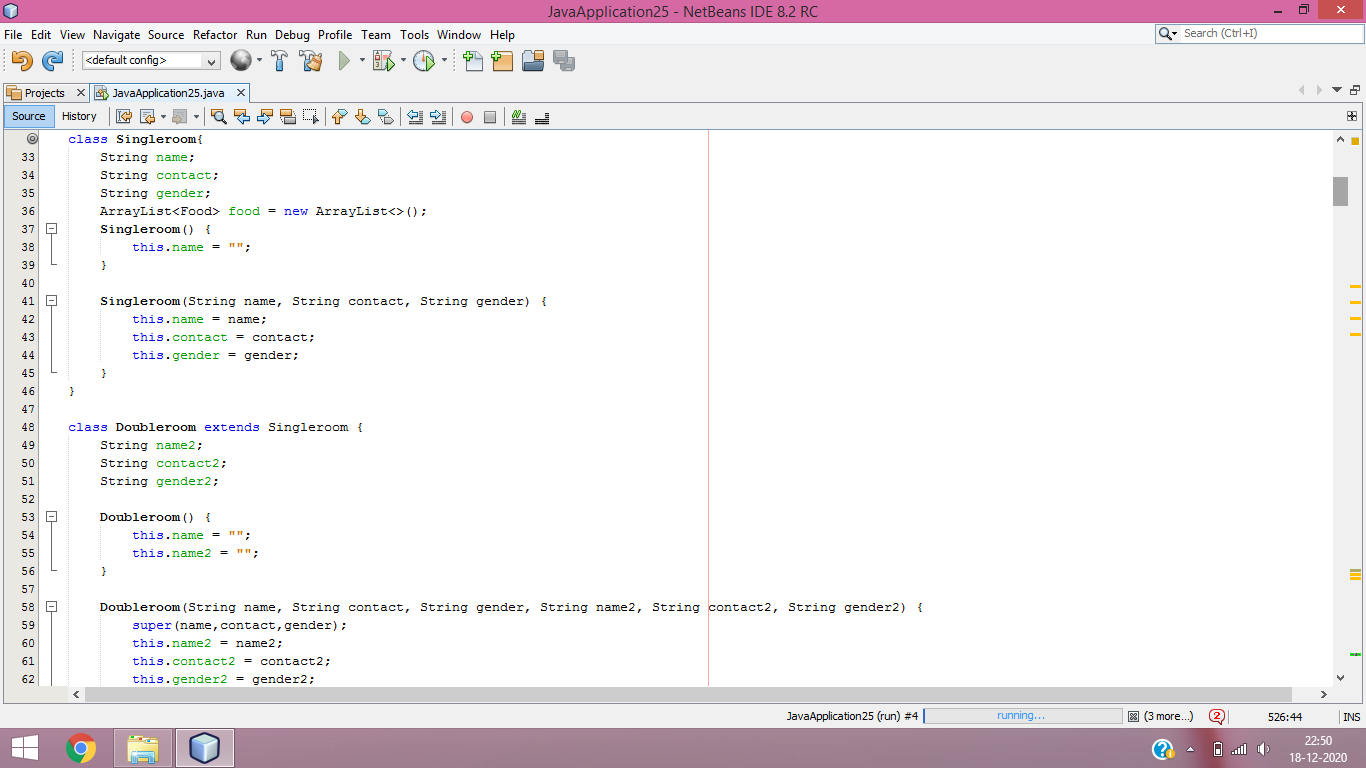
**Justification for selecting the final class diagram:**

* The second class diagram is selected as the final class diagram because it is more efficient as compared to the first one.
* Further, it also uses more proper UML notations compared to the first one.

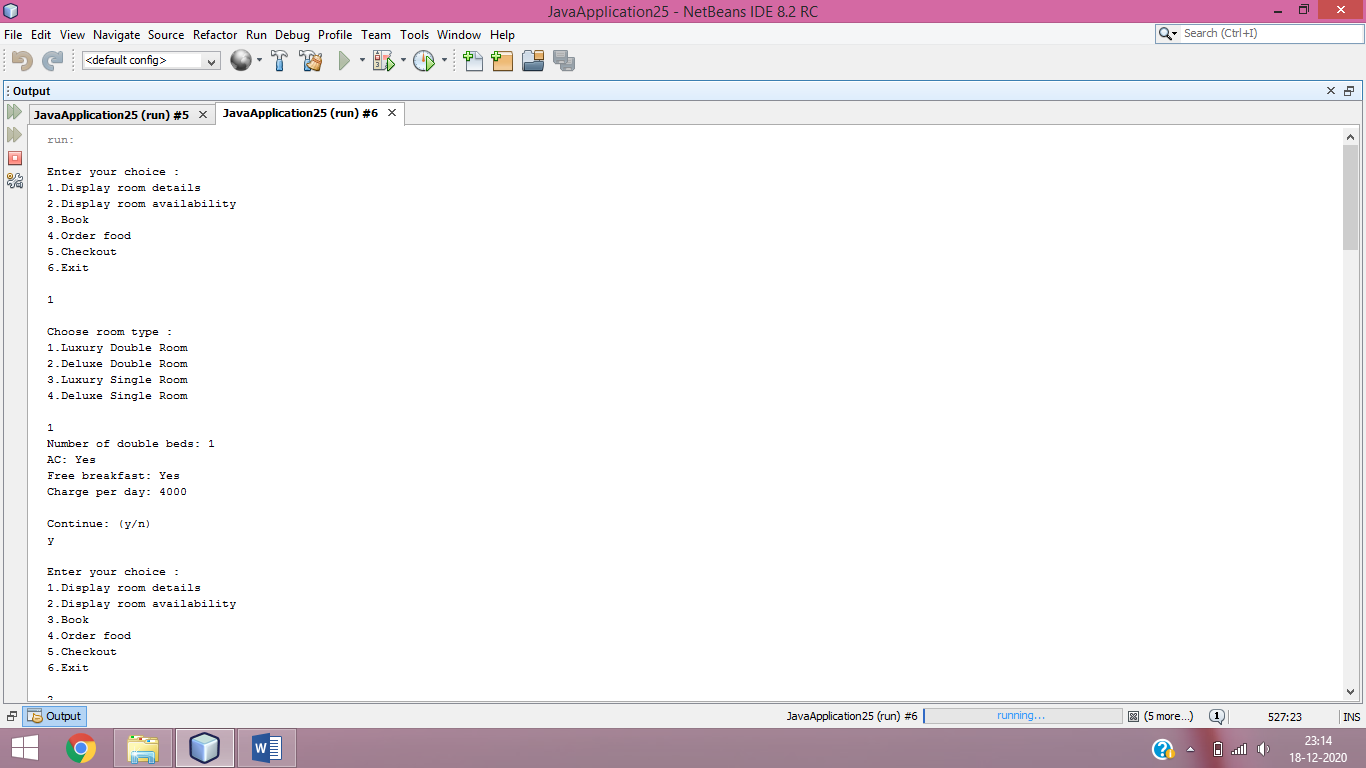
**CHAPTER 3: IMPLEMENTATION**

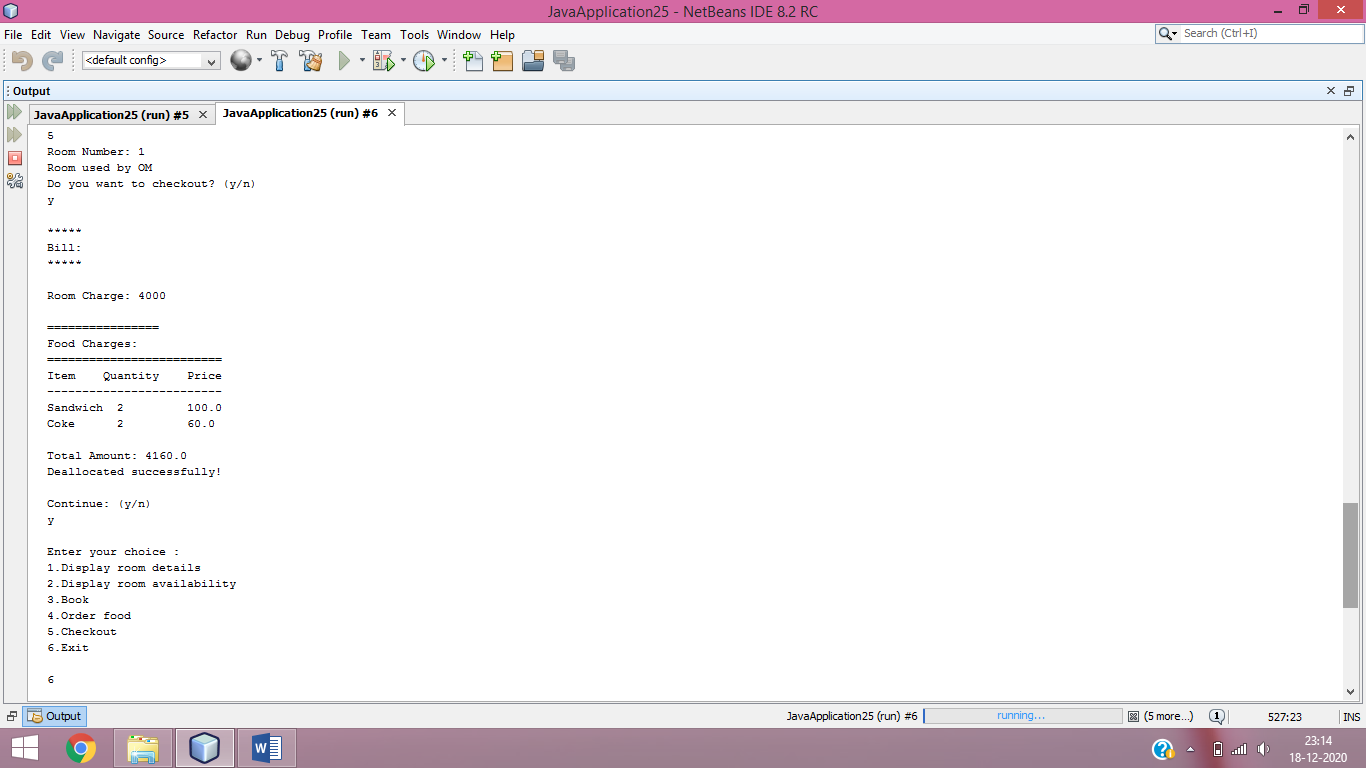
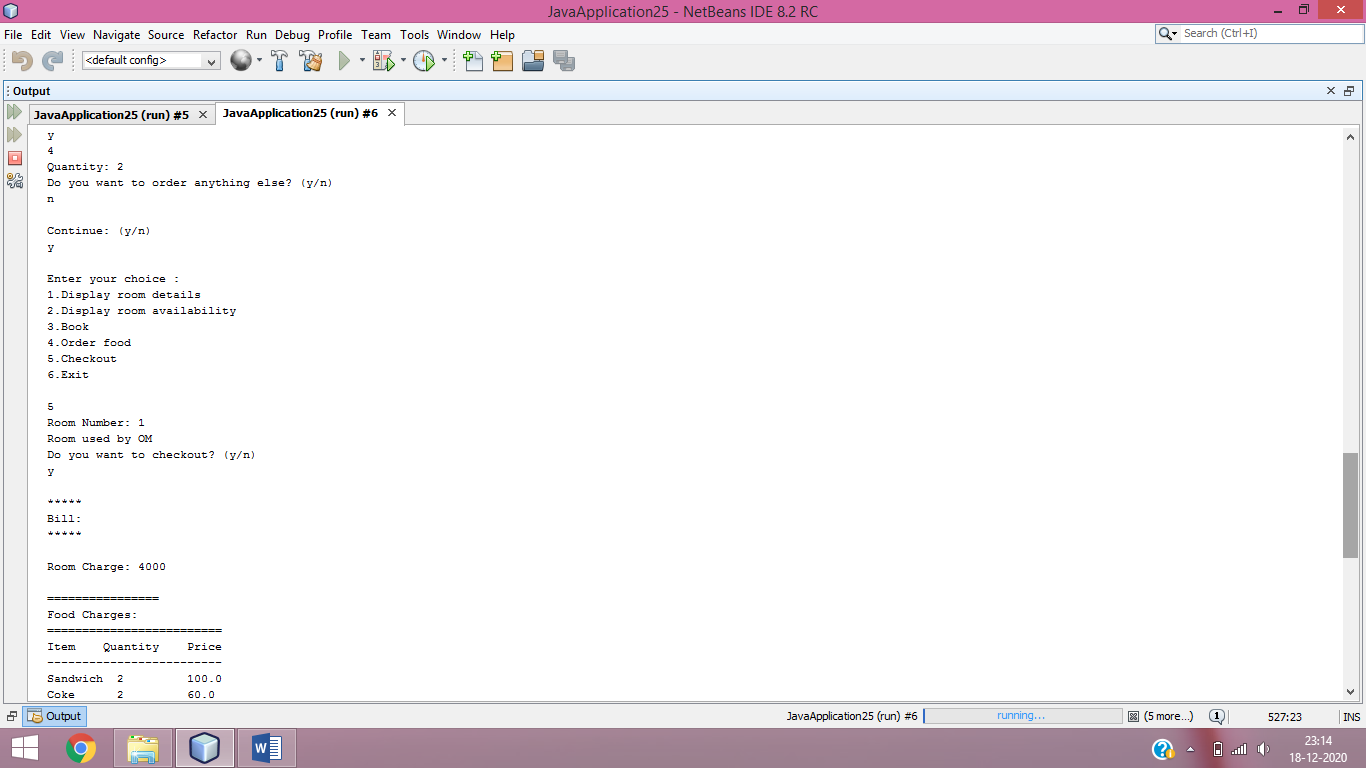
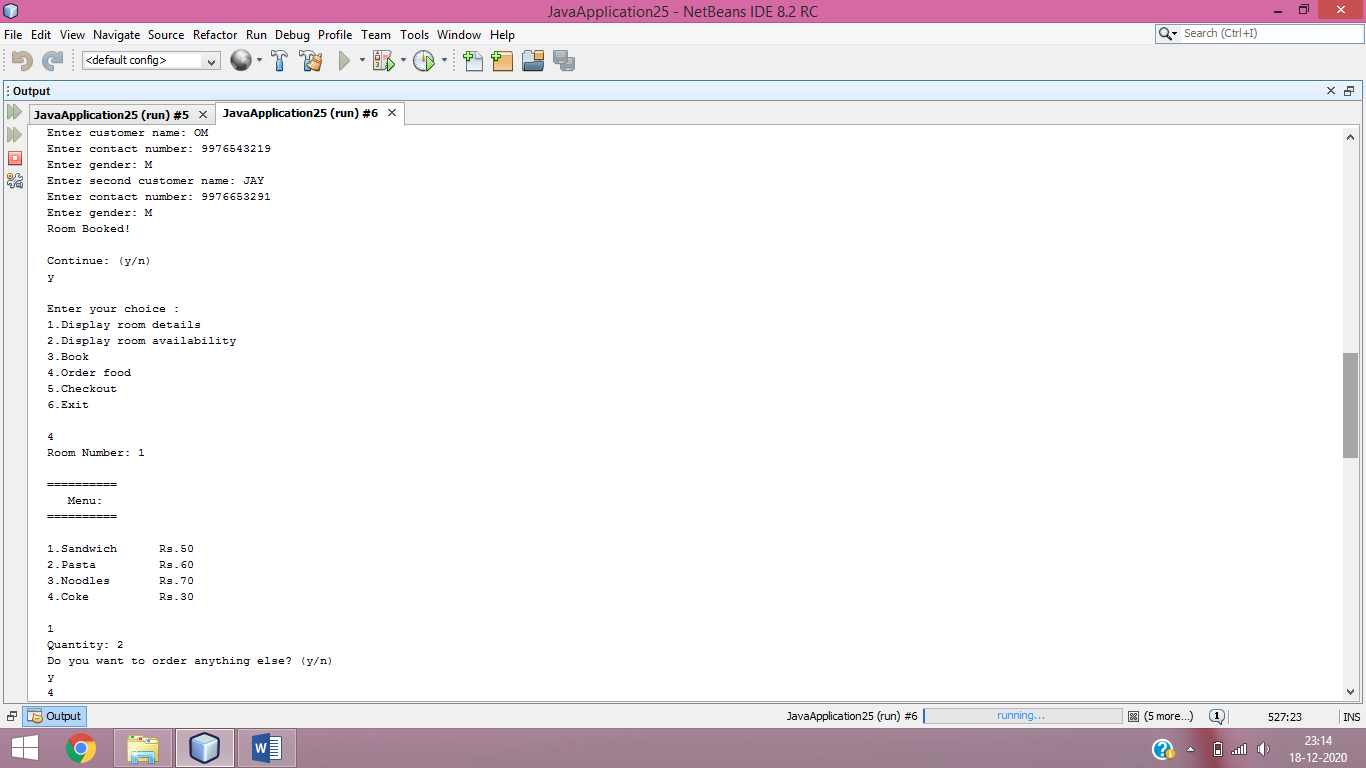
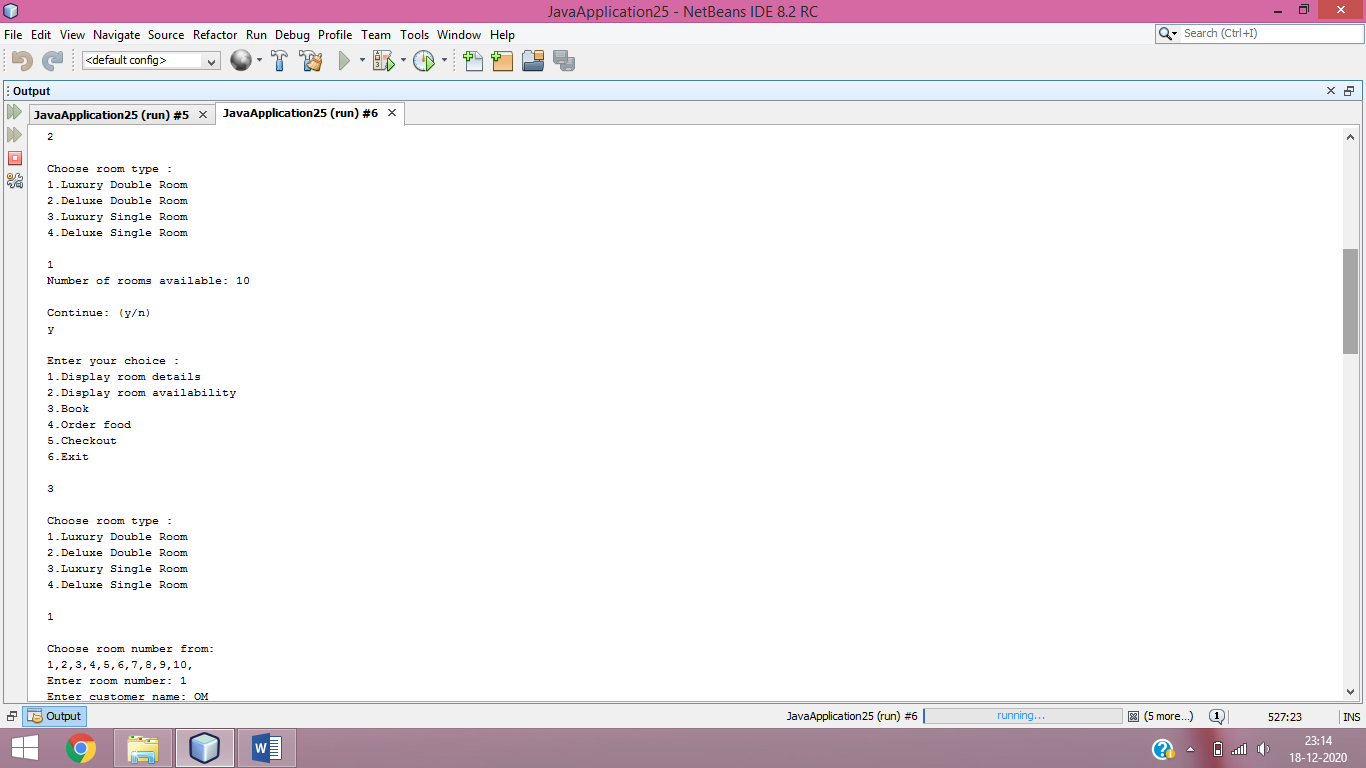
**Source Code:**

****

****

**Screenshots of Output:**

****

****

**CONCLUSION**

After working on this project, we come to the following conclusions:

* Dia is a very useful open source software to draw class diagrams easily and efficiently.
* Interfaces help us in understanding and implementing run time polymorphism easily.
* Java’s exception handling mechanism allows us to gracefully handle run time errors that would otherwise result in unexpected and abrupt termination of the program.
* A package allows to conveniently group related classes so that further maintenance of code becomes easier.
* The ArrayList class is a resizable [array](https://www.w3schools.com/java/java_arrays.asp), which can be found in the java.util package.

**REFERENCES**

* Java Programming: A Comprehensive Introduction by Herbert Schildt and Dale Skrien
* https://www.visual-paradigm.com/guide/uml-unified-modelling-language/uml-class-diagram-tutorial/
* https://developer.ibm.com/articles/an-introduction-to-uml/