JAVA PROJECT REPORT

(Project Term January-May 2023)

Streamlining Hospitality: Automating Hotel Management System for Enhanced Customer Experience

Submitted by

NAME	ROLL NO.	Registration No.
Aditya Chandrayan	26	12103164
Arjun Sharma	74	12103353

Project Group Number ………….

Course Code :-CSE310

Submitted to: -

Under the Guidance of

Dr. Ranjith Kumar A

School of Computer Science and Engineering
Lovely Professional University

Phagwara,Punjab

DECLARATION

We hereby declare that the project work entitled ("Hotel Booking Management") is an authentic record of our own work carried out as requirements of Capstone Project for the award of B.Tech degree in Computer Science from Lovely Professional University, Phagwara, under the guidance of (Name of Faculty Mentor), during August to November 2022. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Name of Student 1: -Aditya Chandrayan
Registration Number :- 12103164

Name of Student 2: - Arjun Sharma
Registration Number:- 12103353

(Aditya Chandrayan)
Date: 20/04/23

(Arjun Sharma)
Date: 20/04/23

TABLE OF CONTENTS

INTRODUCTION

Hotel Management System is a system that provides us to reserving rooms, checking whether the rooms are vacant are or not etc by using online browsing. This system is very useful to all especially for business people.

For Business people they don't have sufficient time for these then they can use these type of online Hotel Management Systems. By this project we will reduce the faults in bills of their expenditure and decrease time of delay to give the bills to the customers. We can also save the bills of the customer. By this project we can also include all the taxes on the bills according to their expenditures. It has a scope to reduce the errors in making the bills. Computerized bill can be printed within fraction of seconds. Online ordering of Booking is possible by using this software. This Project is based on php . If anyone wants to book the room for few days then they can specify the specific number by seeing the types of rooms we have. The bill of this online booking is based on the type of room they can select is displayed.

SOFTWARE REQUIREMENTS:

The major so requirements of the project are as flow

Operating System : Windows xp

HARDWARE REQUIREMENTS:

The hardware requirements that map towards the software are

RAM: 256M

Processor: Intel

Mouse

Keyboard

Abstract

The Project HOTEL BOOKING SYSTEM is a application that allows the hotel manager to handle all hotel activities online. Interactive GUI and the ability to manage various hotel bookings and rooms make this system very flexible and convenient. The hotel manager is a very busy person and does not have the time to sit and manage the entire activities manually on paper. This application gives him the power and flexibility to manage the entire system from a single online system. Hotel booking project provides room booking .

The given code creates a user interface for booking a hotel reservation using Java Swing. It provides text fields for entering the name, date, time, and number of guests, and a button to submit the reservation. This code can be used as a part of a larger hotel booking system project, which would typically include additional functionality for storing and retrieving reservation data, managing room availability, and handling payments.

Scope of the Project

The scope of a hotel management system project in Java can vary depending on the specific requirements of the project. However, a typical hotel management system would include the following functionalities:

Reservation management: This module would allow users to book hotel rooms and manage reservations.

Room management: This module would allow hotel staff to manage room availability, check-in, check-out, and room assignments.

Guest management: This module would allow hotel staff to manage guest information, including contact details, check-in and check-out times, and payment details.

Billing and payment: This module would handle the billing process, including generating invoices, managing payments, and maintaining financial records.

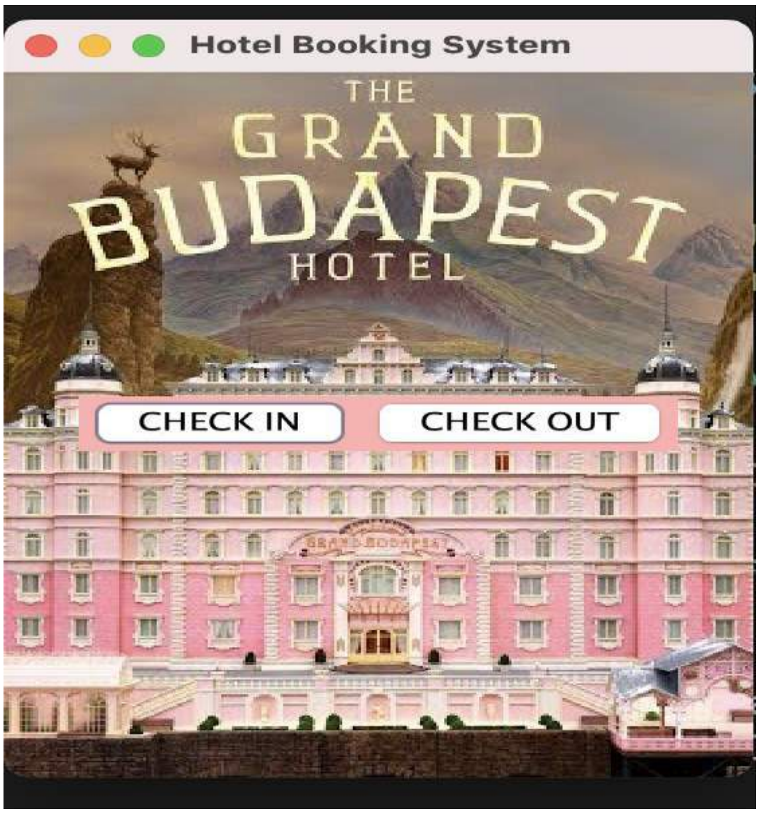
Reporting and analytics: This module would provide reports on room occupancy, revenue, and other metrics to help hotel management make informed decisions.

To develop a hotel management system in Java, one would need to have a good understanding of Java programming language . It would also require knowledge of software development best practices, such as software design patterns, version control, and testing methodologies. The scope of the project can be adjusted depending on the needs of the hotel and the available resources.

Modules:-

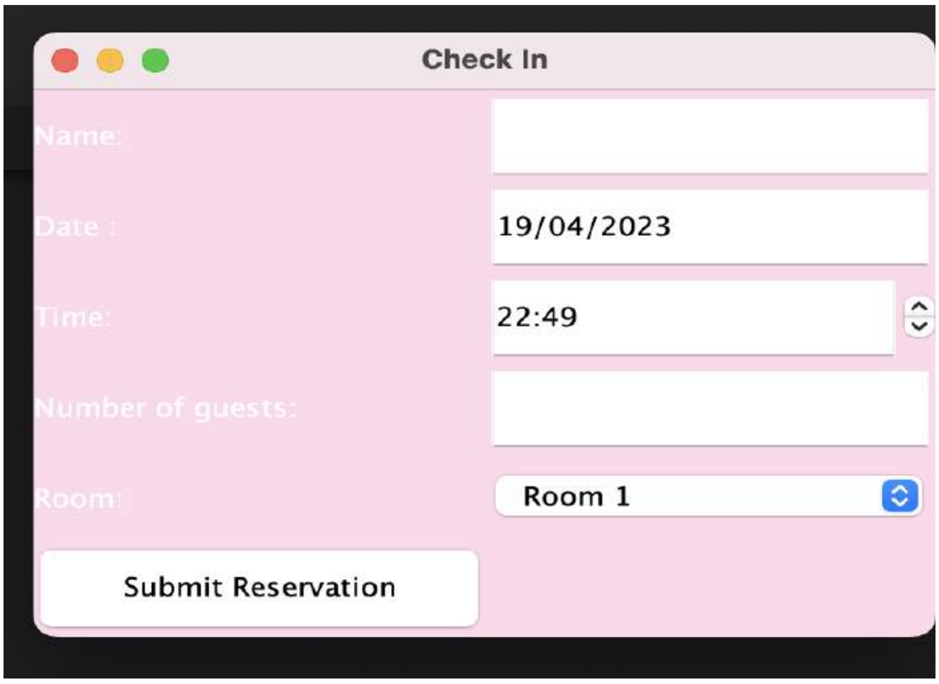
1. Homepage

This module helps to make a user interface for booking a reservation. Name the file Booking.java.



Two Button added in login module.

1. Check in: - It will open the check in page for the entry of new guests.


Check In

Name:

Date : 19/04/2023

Time: 22:49

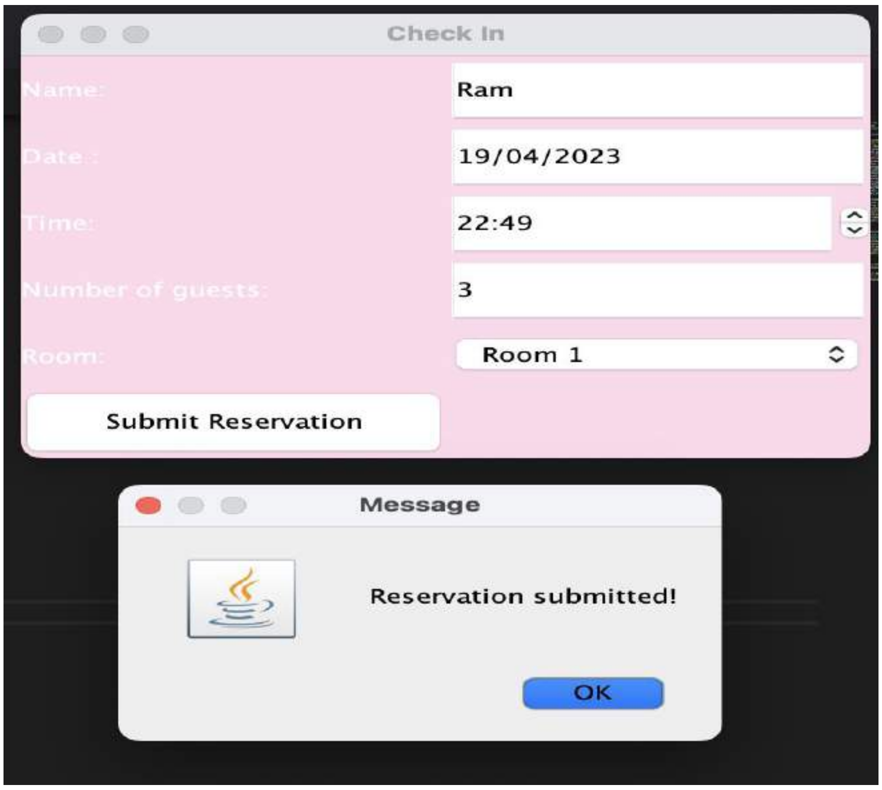
Number of guests:

Room: Room 1

Submit Reservation

Check in:-

It include input fields for the guest name, date, time, number of guests, and available room selection. Upon submission, the code checks room availability and writes reservation data to a CSV file, and displays the total bill.


Check In

Name: Ram

Date : 19/04/2023

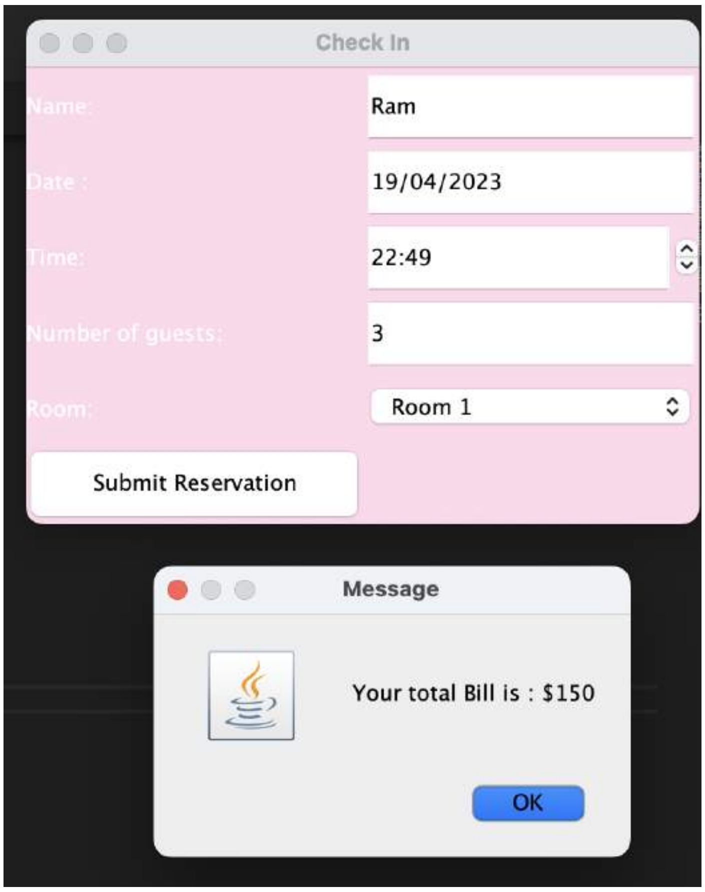
Time: 22:49

Number of guests: 3

Room: Room 1

Message

Reservation submitted!


Check In

Name: Ram

Date : 19/04/2023

Time: 22:49

Number of guests: 3

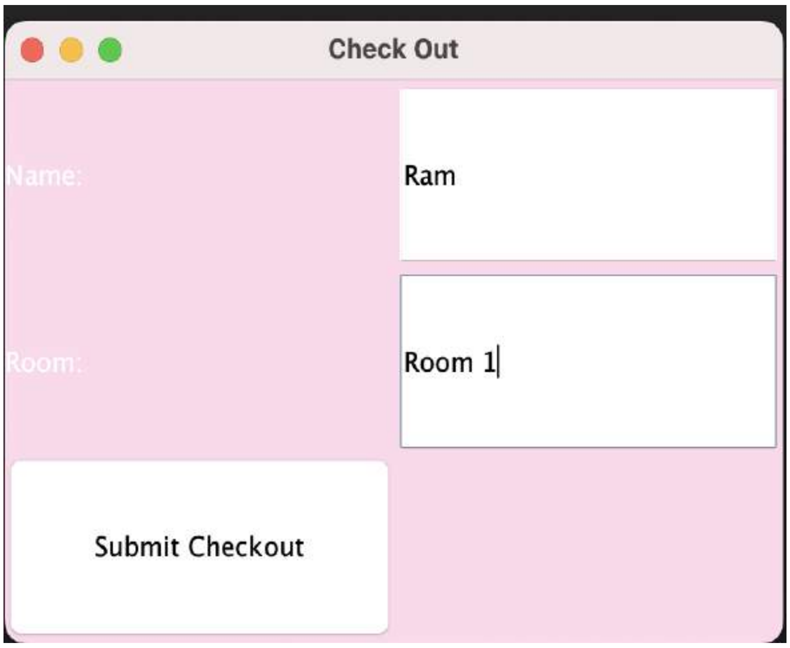
Room: Room 1

Message

Your total Bill is : $150

2. Check out: - It will open for the maintaining the exit details of the guests.

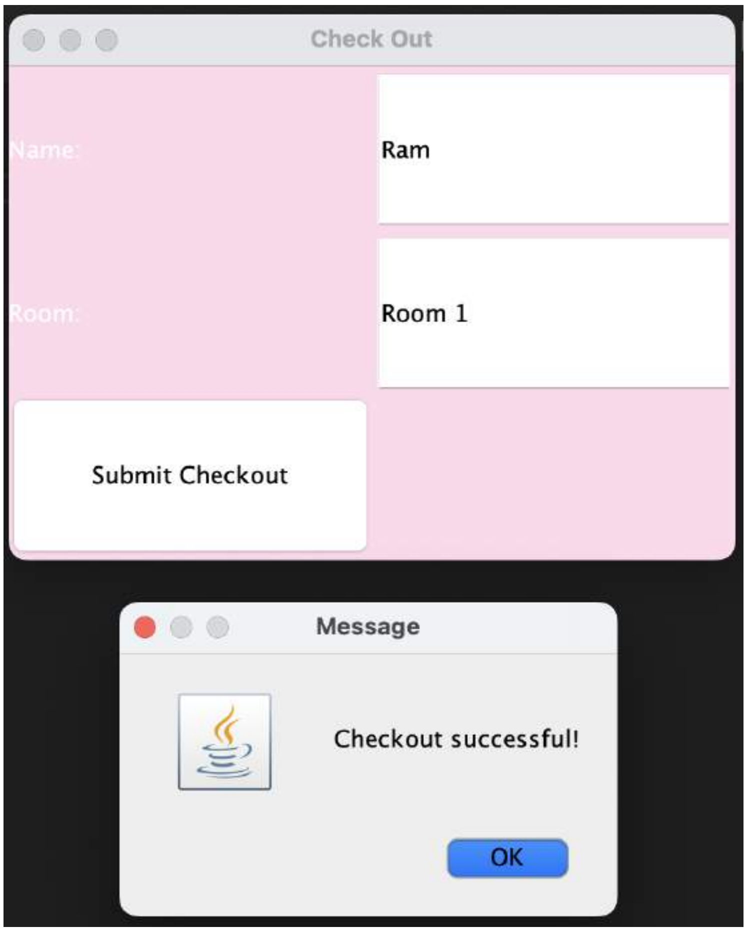
It include the input fields for guest name and room number. Upon submission, the code removes the corresponding reservation data from a CSV file, displaying a success message or error message if the reservation is not found or invalid.



Check Out

Name: Ram

Room: Room 1



Check Out

Name: Ram

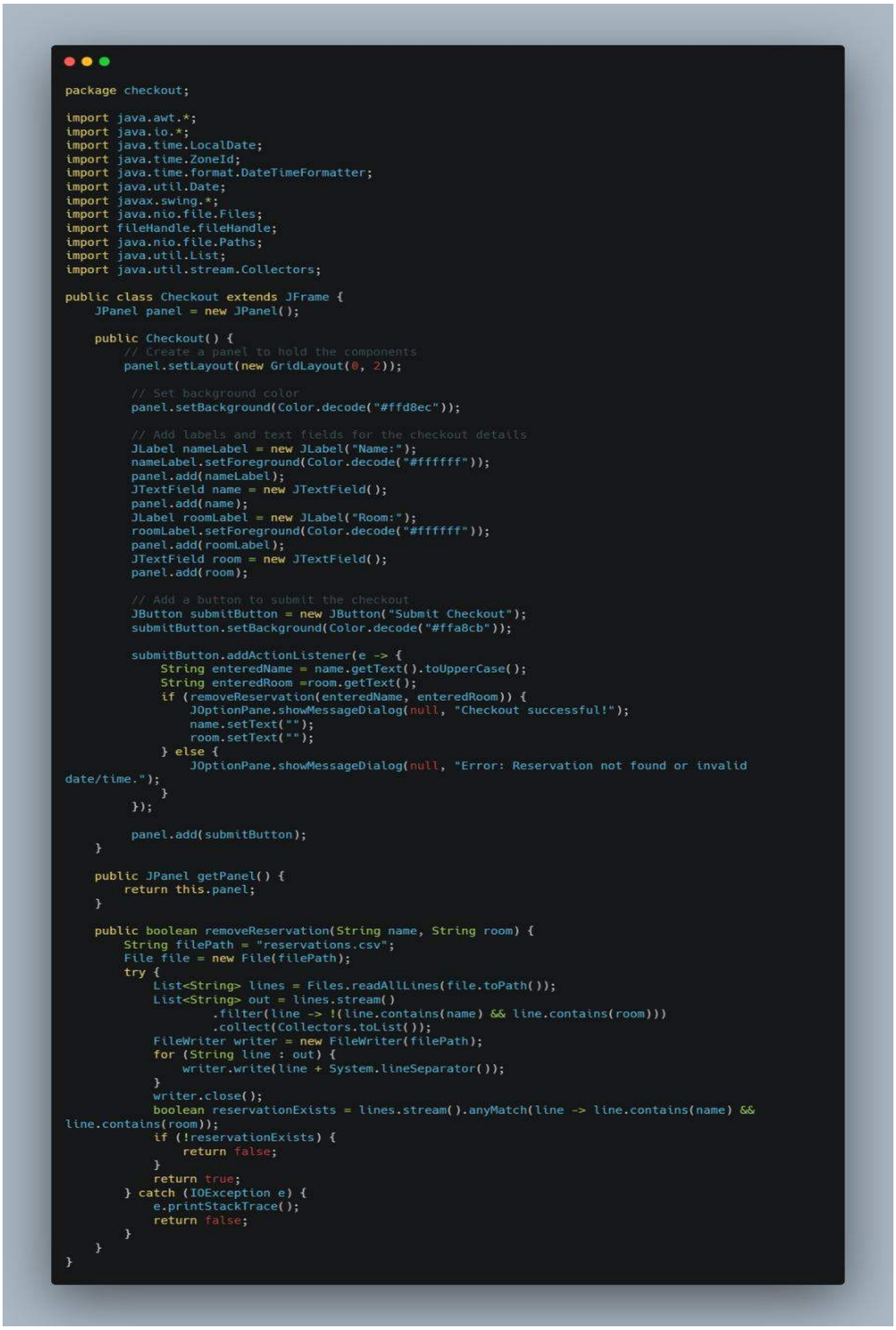
Room: Room 1

Message

Checkout successful!

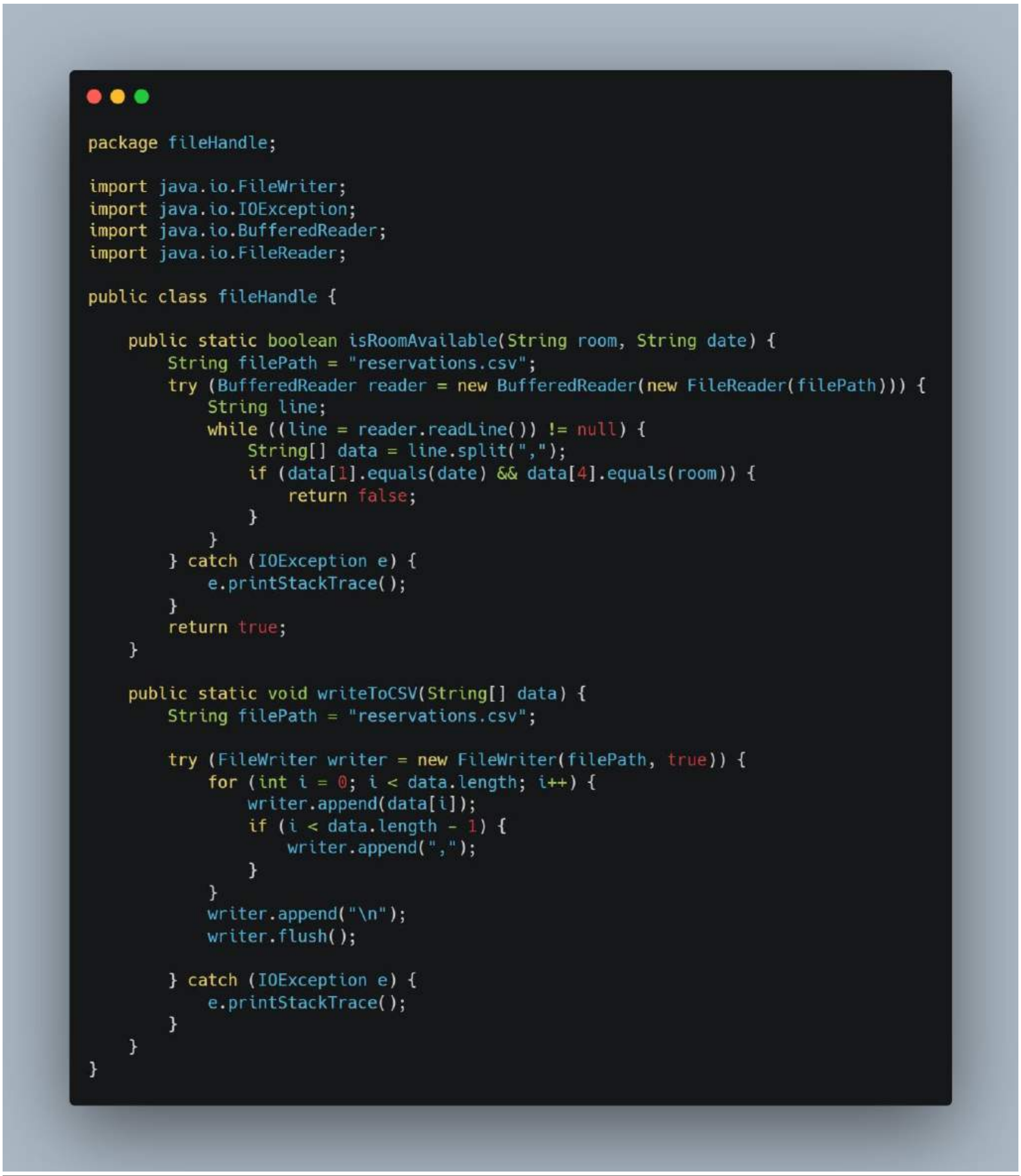
Source code:-

Checkout.java



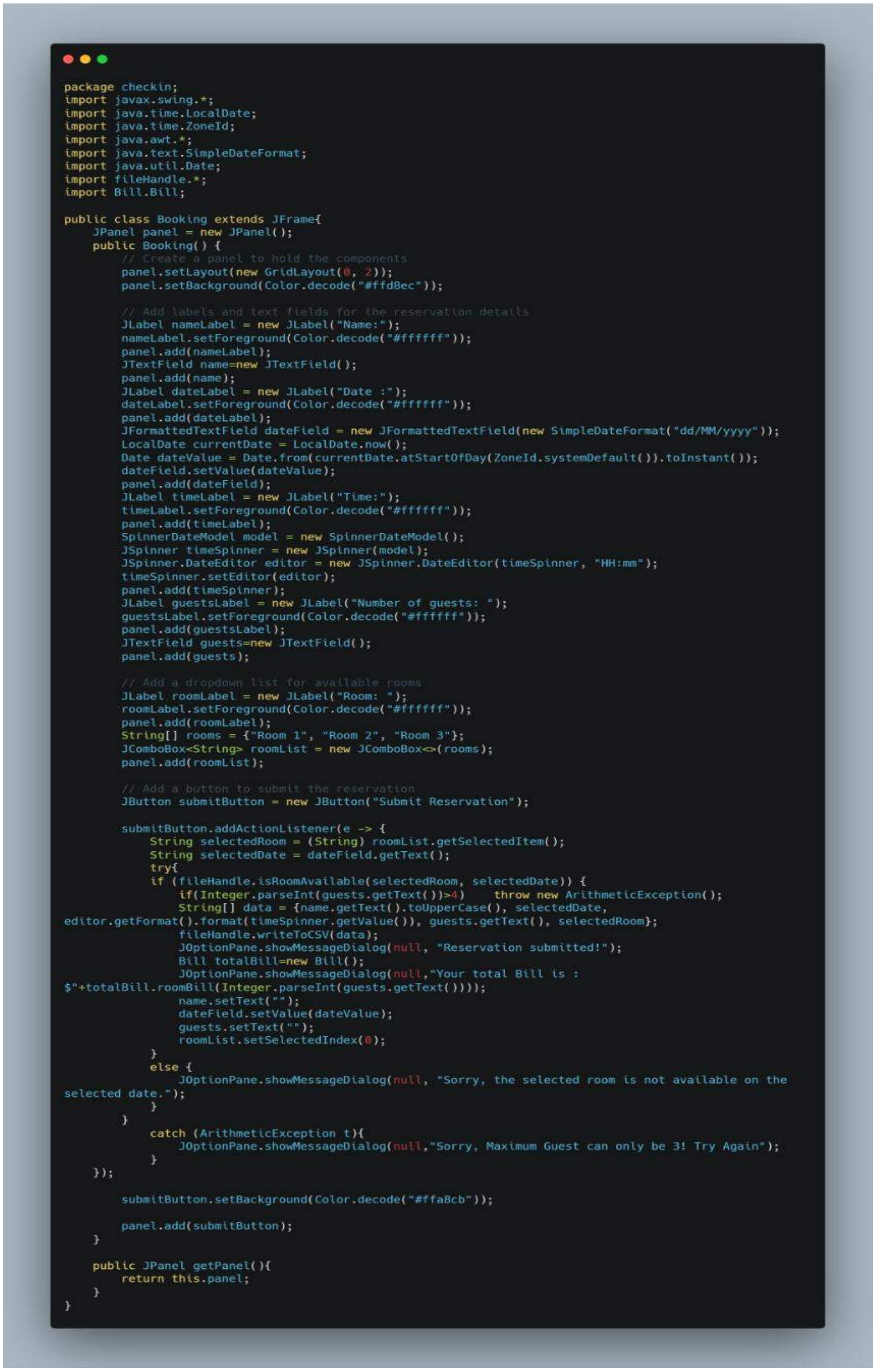
```java
package checkout;

import java.awt.*;
import java.io.*;
import java.time.LocalDate;
import java.time.ZoneId;
import java.time.format.DateTimeFormatter;
import java.util.Date;
import javax.swing.*;
import java.nio.file.Files;
import fileHandle.fileHandle;
import java.nio.file.Paths;
import java.util.List;
import java.util.stream.Collectors;

public class Checkout extends JFrame {
    JPanel panel = new JPanel();

    public Checkout() {
        // Create a panel to hold the components
        panel.setLayout(new GridLayout(0, 2));

         // Set background color
         panel.setBackground(Color.decode("#ffd8ec"));

         // Add labels and text fields for the checkout details
         JLabel nameLabel = new JLabel("Name:");
         nameLabel.setForeground(Color.decode("#ffffff"));
         panel.add(nameLabel);
         JTextField name = new JTextField();
         panel.add(name);
         JLabel roomLabel = new JLabel("Room:");
         roomLabel.setForeground(Color.decode("#ffffff"));
         panel.add(roomLabel);
         JTextField room = new JTextField();
         panel.add(room);

         // Add a button to submit the checkout
         JButton submitButton = new JButton("Submit Checkout");
         submitButton.setBackground(Color.decode("#ffa8cb"));

         submitButton.addActionListener(e -> {
             String enteredName = name.getText().toUpperCase();
             String enteredRoom =room.getText();
             if (removeReservation(enteredName, enteredRoom)) {
                 JOptionPane.showMessageDialog(null, "Checkout successful!");
                 name.setText("");
                 room.setText("");
             } else {
                 JOptionPane.showMessageDialog(null, "Error: Reservation not found or invalid
date/time.");
             }
         });

         panel.add(submitButton);
    }

    public JPanel getPanel() {
        return this.panel;
    }

    public boolean removeReservation(String name, String room) {
        String filePath = "reservations.csv";
        File file = new File(filePath);
        try {
            List<String> lines = Files.readAllLines(file.toPath());
            List<String> out = lines.stream()
                    .filter(line -> !(line.contains(name) && line.contains(room)))
                    .collect(Collectors.toList());
            FileWriter writer = new FileWriter(filePath);
            for (String line : out) {
                writer.write(line + System.lineSeparator());
            }
            writer.close();
            boolean reservationExists = lines.stream().anyMatch(line -> line.contains(name) &&
line.contains(room));
            if (!reservationExists) {
                return false;
            }
            return true;
        } catch (IOException e) {
            e.printStackTrace();
            return false;
        }
    }
}
```

Filehandel.java



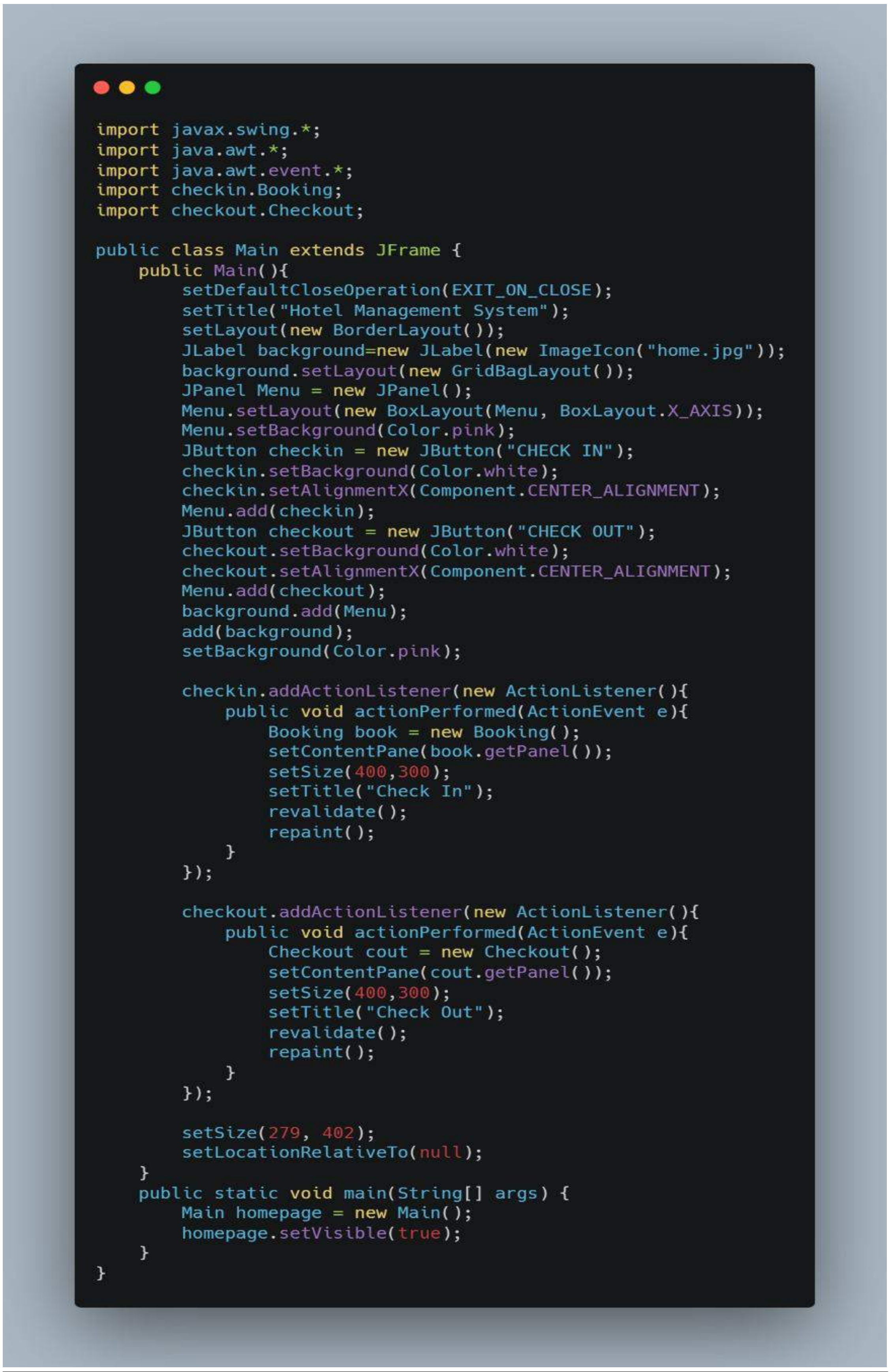
```java
package fileHandle;

import java.io.FileWriter;
import java.io.IOException;
import java.io.BufferedReader;
import java.io.FileReader;

public class fileHandle {

    public static boolean isRoomAvailable(String room, String date) {
        String filePath = "reservations.csv";
        try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) {
            String line;
            while ((line = reader.readLine()) != null) {
                String[] data = line.split(",");
                if (data[1].equals(date) && data[4].equals(room)) {
                    return false;
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        return true;
    }

    public static void writeToCSV(String[] data) {
        String filePath = "reservations.csv";

        try (FileWriter writer = new FileWriter(filePath, true)) {
            for (int i = 0; i < data.length; i++) {
                writer.append(data[i]);
                if (i < data.length - 1) {
                    writer.append(",");
                }
            }
            writer.append("\n");
            writer.flush();

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Booking.java



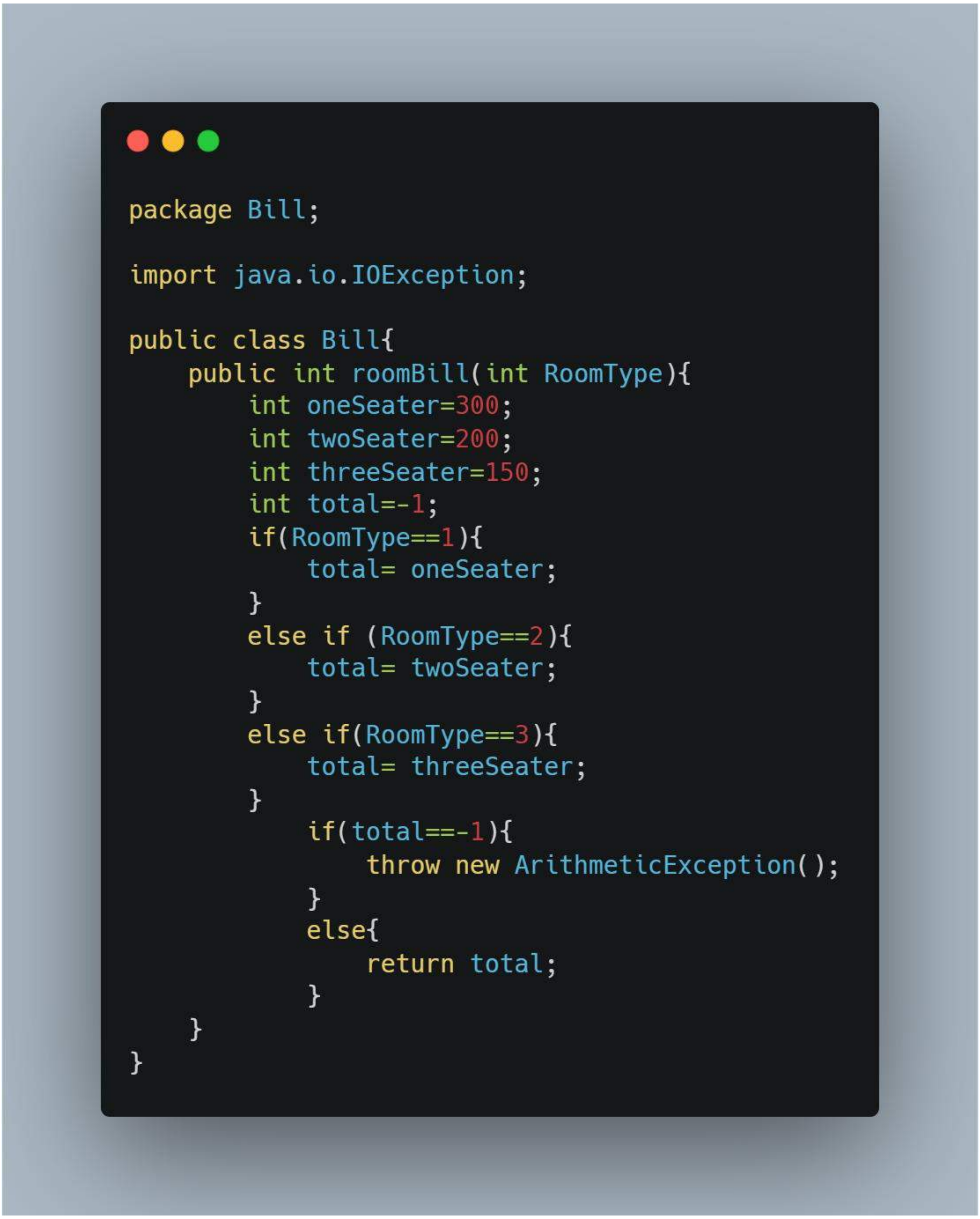
```java
package checkin;
import javax.swing.*;
import java.time.LocalDate;
import java.time.ZoneId;
import java.awt.*;
import java.text.SimpleDateFormat;
import java.util.Date;
import fileHandle.*;
import Bill.Bill;

public class Booking extends JFrame{
    JPanel panel = new JPanel();
    public Booking() {
        // Create a panel to hold the components
        panel.setLayout(new GridLayout(0, 2));
        panel.setBackground(Color.decode("#ffd8ec"));

        // Add labels and text fields for the reservation details
        JLabel nameLabel = new JLabel("Name:");
        nameLabel.setForeground(Color.decode("#ffffff"));
        panel.add(nameLabel);
        JTextField name=new JTextField();
        panel.add(name);
        JLabel dateLabel = new JLabel("Date :");
        dateLabel.setForeground(Color.decode("#ffffff"));
        panel.add(dateLabel);
        JFormattedTextField dateField = new JFormattedTextField(new SimpleDateFormat("dd/MM/yyyy"));
        LocalDate currentDate = LocalDate.now();
        Date dateValue = Date.from(currentDate.atStartOfDay(ZoneId.systemDefault()).toInstant());
        dateField.setValue(dateValue);
        panel.add(dateField);
        JLabel timeLabel = new JLabel("Time:");
        timeLabel.setForeground(Color.decode("#ffffff"));
        panel.add(timeLabel);
        SpinnerDateModel model = new SpinnerDateModel();
        JSpinner timeSpinner = new JSpinner(model);
        JSpinner.DateEditor editor = new JSpinner.DateEditor(timeSpinner, "HH:mm");
        timeSpinner.setEditor(editor);
        panel.add(timeSpinner);
        JLabel guestsLabel = new JLabel("Number of guests: ");
        guestsLabel.setForeground(Color.decode("#ffffff"));
        panel.add(guestsLabel);
        JTextField guests=new JTextField();
        panel.add(guests);

        // Add a dropdown list for available rooms
        JLabel roomLabel = new JLabel("Room: ");
        roomLabel.setForeground(Color.decode("#ffffff"));
        panel.add(roomLabel);
        String[] rooms = {"Room 1", "Room 2", "Room 3"};
        JComboBox<String> roomList = new JComboBox<>(rooms);
        panel.add(roomList);

        // Add a button to submit the reservation
        JButton submitButton = new JButton("Submit Reservation");

        submitButton.addActionListener(e -> {
            String selectedRoom = (String) roomList.getSelectedItem();
            String selectedDate = dateField.getText();
            try{
            if (fileHandle.isRoomAvailable(selectedRoom, selectedDate)) {
                if(Integer.parseInt(guests.getText())>4)    throw new ArithmeticException();
                String[] data = {name.getText().toUpperCase(), selectedDate,
editor.getFormat().format(timeSpinner.getValue()), guests.getText(), selectedRoom};
                fileHandle.writeToCSV(data);
                JOptionPane.showMessageDialog(null, "Reservation submitted!");
                Bill totalBill=new Bill();
                JOptionPane.showMessageDialog(null,"Your total Bill is :
$"+totalBill.roomBill(Integer.parseInt(guests.getText())));
                name.setText("");
                dateField.setValue(dateValue);
                guests.setText("");
                roomList.setSelectedIndex(0);
            }
            else {
                JOptionPane.showMessageDialog(null, "Sorry, the selected room is not available on the
selected date.");
            }
        }
            catch (ArithmeticException t){
                JOptionPane.showMessageDialog(null,"Sorry, Maximum Guest can only be 3! Try Again");
            }
    });

        submitButton.setBackground(Color.decode("#ffa8cb"));

        panel.add(submitButton);
    }

    public JPanel getPanel(){
        return this.panel;
    }
}
```

Main.java



```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import checkin.Booking;
import checkout.Checkout;

public class Main extends JFrame {
    public Main(){
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setTitle("Hotel Management System");
        setLayout(new BorderLayout());
        JLabel background=new JLabel(new ImageIcon("home.jpg"));
        background.setLayout(new GridBagLayout());
        JPanel Menu = new JPanel();
        Menu.setLayout(new BoxLayout(Menu, BoxLayout.X_AXIS));
        Menu.setBackground(Color.pink);
        JButton checkin = new JButton("CHECK IN");
        checkin.setBackground(Color.white);
        checkin.setAlignmentX(Component.CENTER_ALIGNMENT);
        Menu.add(checkin);
        JButton checkout = new JButton("CHECK OUT");
        checkout.setBackground(Color.white);
        checkout.setAlignmentX(Component.CENTER_ALIGNMENT);
        Menu.add(checkout);
        background.add(Menu);
        add(background);
        setBackground(Color.pink);

        checkin.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                Booking book = new Booking();
                setContentPane(book.getPanel());
                setSize(400,300);
                setTitle("Check In");
                revalidate();
                repaint();
            }
        });

        checkout.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                Checkout cout = new Checkout();
                setContentPane(cout.getPanel());
                setSize(400,300);
                setTitle("Check Out");
                revalidate();
                repaint();
            }
        });

        setSize(279, 402);
        setLocationRelativeTo(null);
    }
    public static void main(String[] args) {
        Main homepage = new Main();
        homepage.setVisible(true);
    }
}
```

Bill.java



```java
package Bill;

import java.io.IOException;

public class Bill{
    public int roomBill(int RoomType){
        int oneSeater=300;
        int twoSeater=200;
        int threeSeater=150;
        int total=-1;
        if(RoomType==1){
            total= oneSeater;
        }
        else if (RoomType==2){
            total= twoSeater;
        }
        else if(RoomType==3){
            total= threeSeater;
        }
            if(total==-1){
                throw new ArithmeticException();
            }
            else{
                return total;
            }
    }
}
```

Conclusion

In conclusion, the development of a hotel booking management system is a complex task that requires a range of technical skills and expertise. However, a well-designed system can offer many benefits to hotel businesses, including improved booking efficiency, better inventory management, and increased revenue.

Through the development process, it is important to consider the specific needs of the hotel business and its customers. Key features such as online booking, real-time availability, and automated notifications can enhance the user experience and increase customer satisfaction

It is also important to consider potential challenges such as data security, scalability, and system integration when designing and implementing the system.

Overall, a hotel booking management system in Java can be a valuable tool for streamlining hotel operations, improving customer service, and driving business growth.