```java
package hotel;

import java.io.*;
import java.lang.reflect.Array;
import java.time.LocalDate;
import java.util.*;

/**
 * An implementation of the Hotel interface for a hotel management system.
 *
 * ECM1410 - Object Oriented Programming
 * Continuous Assessment 2019 - Hotel management system
 *
 * @author1 680063381 - mtj202 - 008203
 * @author2 680062814 - da376 -
 * @version 12/03/2019
 *
 */

public class HotelImpl implements Hotel {
    /*
     * You are required to at least provide one constructor as follows
     * which initialises the hotel with all the data from the four text
     * files.
     */

    // I haven't included any Java documentation comments, but you need add them.

    public static void main(String[] args){
        HotelImpl init = new HotelImpl();
        init.HotelImpl("rooms.txt", "guests.txt", "bookings.txt", "payments.txt");
    }

    public void HotelImpl(String roomsTxtFileName, String guestsTxtFileName,
                          String bookingsTxtFileName, String paymentsTxtFileName){

        // Loads data from all of the four text files.

        importRoomsData(roomsTxtFileName);
        importGuestsData(guestsTxtFileName);
        importBookingsData(bookingsTxtFileName);
        importPaymentsData(paymentsTxtFileName);


        // // // // // // // // // TEST CHAMBERS // // // // // // // // //

        // Lol
        System.out.println("--== Epic ==--");

        addRoom(405, RoomType.SINGLE, 50, 2, "Shared bathroom");
        // Room 405 check
        System.out.println("\n\n--== Test: Room 405 (added) ==--\n");
        Room room = rooms.stream()
                .filter(r -> r.getRoomNumber() == 405)
                .findFirst().orElseThrow();

        System.out.println("Room Number " + room.getRoomNumber());
        System.out.println("Type " + room.getRoomType());
        System.out.println("Cost " + room.getRoomPrice());
        System.out.println("Capacity " + room.getRoomCapacity());
        System.out.println("Facilities " + room.getRoomFacilities());

        // Guest 10003 (VIP) check
        System.out.println("\n\n--== Test: Guest 10003 (VIP) ==--\n");
        Guest guest = guests.stream()
                .filter(g -> g.getGuestID() == 10003)
                .findFirst().orElseThrow();

        System.out.println("Guest ID " + guest.getGuestID());
        System.out.println("First Name " + guest.getfName());
        System.out.println("Last Name " + guest.getlName());
        System.out.println("Date Joined " + guest.getDateJoin());
        System.out.println("VIP Start Date " + guest.getVIPstartDate());
        System.out.println("VIP Expiry Date " + guest.getVIPexpiryDate());

        System.out.println("\n\n--== Test: Booking 100009 ==--\n");
        Booking booking = bookings.stream()
                .filter(b -> b.getID() == 100009)
                .findFirst().orElseThrow();

        System.out.println("Booking ID " + booking.getID());
        System.out.println("Guest ID " + booking.getGuestID());
        System.out.println("Room No. " + booking.getRoomNumber());
        System.out.println("Date Booked " + booking.getBookingDate());
        System.out.println("Check in " + booking.getCheckInDate());
        System.out.println("Check out " + booking.getCheckOutDate());

    }
    /*
     * Main Attributes
     */
    public List<Room> rooms = new ArrayList<>();
    public List<Guest> guests = new ArrayList<>();
    public List<Booking> bookings = new ArrayList<>();
    //public List<Payment> payments = new ArrayList<>();

    /*
     * Main Methods
     */

    public boolean importRoomsData(String roomsTxtFileName){
        try {
            BufferedReader reader = new BufferedReader(new FileReader(roomsTxtFileName));
            String line = reader.readLine();
            while (line != null) {
                String[] data = line.split(",");

                int roomNumber = Integer.valueOf(data[0]);
                RoomType roomType = null;

                switch (data[1]){
                    case "single":
                        roomType = RoomType.SINGLE;

                    case "double":
                        roomType = RoomType.DOUBLE;

                    case "family":
                        roomType = RoomType.FAMILY;

                    case "twin":
                        roomType = RoomType.TWIN;
                }
                double roomPrice = Double.valueOf(data[2]);
                int roomCapacity = Integer.valueOf(data[3]);
                String roomFacilities = data[4];

                line = reader.readLine();

                Room r = new Room(roomNumber, roomType, roomPrice, roomCapacity, roomFacilities);
                rooms.add(r);
            }
            reader.close();
            return true;

        } catch (IOException ex) {
            System.out.println(ex.getMessage());
        }
        return false;
    }

    public boolean importGuestsData(String guestsTxtFileName){
        try {
            BufferedReader reader = new BufferedReader(new FileReader(guestsTxtFileName));
            String line = reader.readLine();
            while (line != null) {
                String[] data = line.split(",");

                int guestID = Integer.valueOf(data[0]);
                String fName = data[1];
                String lName = data[2];
                LocalDate dateJoin = LocalDate.parse(data[3]);
                LocalDate VIPstartDate = null;
                LocalDate VIPexpiryDate = null;

                if (data.length == 6) {
                    VIPstartDate = LocalDate.parse(data[4]);
                    VIPexpiryDate = LocalDate.parse(data[5]);
                }
                line = reader.readLine();

                Guest g = new Guest(guestID, fName, lName, dateJoin, VIPstartDate, VIPexpiryDate);
                guests.add(g);
            }
            reader.close();
            return true;

        } catch (IOException ex) {
            System.out.println(ex.getMessage());
        }
        return false;
    }

    public boolean importBookingsData(String bookingsTxtFileName){

        try {
            BufferedReader reader = new BufferedReader(new FileReader(bookingsTxtFileName));
            String line = reader.readLine();
            while (line != null) {
                String[] data = line.split(",");

                int id = Integer.valueOf(data[0]);
                int guestID = Integer.valueOf(data[1]);
                int roomNumber = Integer.valueOf(data[2])
```

```java
;
                LocalDate bookingDate = LocalDate.parse(data[3]);
                LocalDate checkinDate = LocalDate.parse(data[4]);
                LocalDate checkoutDate = LocalDate.parse(data[5]);
                double totalCost = Double.valueOf(data[6]);

                line = reader.readLine();

                Booking b = new Booking(id, guestID, roomNumber, bookingDate, checkinDate, checkoutDate, totalCost); //Add correct parameters in Booking(id, guestID, ..etc)
                bookings.add(b);
            }
            reader.close();
            return true;

        } catch (IOException ex) {
            System.out.println(ex.getMessage());
        }
        return false;
    }

    public boolean importPaymentsData(String paymentsTxtFileName){
        /*try {
            BufferedReader reader = new BufferedReader(new FileReader(paymentsTxtFileName));
            String line = reader.readLine();
            while (line != null) {
                String[] data = line.split(",");

                // Add stuff here, same as before.

                line = reader.readLine();

                Payment p = new Payment(...);
                payments.add(p);
            }
            reader.close();
            return true;

        } catch (IOException ex) {
            System.out.println(ex.getMessage());
        }*/
        return false;
    }

    public void displayAllRooms(){}

    public void displayAllGuests(){}

    public void displayAllBookings(){}

    public void displayAllPayments(){}

    public boolean addRoom(int roomNumber, RoomType roomType, double price, int capacity, String facilities) {

        Room r = new Room(roomNumber, roomType, price, capacity, facilities);
        rooms.add(r);

        for(Room room: rooms){
            System.out.println("Room: " + room);
            if(room.getRoomNumber() == 401){
                System.out.println("Room exists!");
            } else{
                System.out.println("Room DOES NOT exist!");
            }
        }

        /* THE CODE BELOW WORKS (old version)
        try{
            Room room = rooms.stream()
                    .filter(r -> r.getRoomNumber() == roomNumber)
                    .findFirst()
                    .orElseThrow(() -> new NoSuchElementException());
            System.out.println("Error: Room " + roomNumber + " already exists.");

        } catch(NoSuchElementException ex){
            Room r = new Room(roomNumber, roomType, price, capacity, facilities);
            rooms.add(r);
            return true;
        }*/
        return false;
    }

    public boolean removeRoom(int roomNumber){return true;}

    public boolean addGuest(String fName, String lName, LocalDate dateJoin){return true;}

    public boolean addGuest(String fName, String lName, LocalDate dateJoin, LocalDate VIPstartDate, LocalDate VIPexpiryDate){return true;}

    public boolean removeGuest(int guestID){return true;}

    public boolean isAvailable(int roomNumber, LocalDate checkin, LocalDate checkout){return true;}

    public int[] availableRooms(RoomType roomType, LocalDate checkin, LocalDate checkout){return null;}

    public int bookOneRoom(int guestID, RoomType roomType, LocalDate checkin, LocalDate checkout){return 1;}

    public boolean checkOut(int bookingID, LocalDate actualCheckoutDate){return true;}

    public boolean cancelBooking(int bookingID){return true;}

    public int[] searchGuest(String firstName, String lastName){return null;}

    public void displayGuestBooking(int guestID){}

    public void displayBookingsOn(LocalDate thisDate){}

    public void displayPaymentsOn(LocalDate thisDate){}

    public boolean saveRoomsData(String roomsTxtFileName)
```

```java
{return true;}

    public boolean saveGuestsData(String guestsTxtFileName){return true;}

    public boolean saveBookingsData(String bookingsTxtFileName){return true;}

    public boolean savePaymentsData(String paymentsTxtFileName){return true;}

    /*
     *  Supporting Methods
     */

    // Checks whether a specified room, guest, booking or payment exists.
    // Note: for the 'ROOM' data type, the parameter 'id' is the room number.
    public static boolean doesExist(DataType type, int id){
        /*switch(type){
            case ROOM:
                code
            case GUEST:
                code
            case BOOKING:
                code
            case PAYMENT:
                code
        }*/
        return true;
    }
    /*
    doesContain()

    public static boolean doesContain(String[] arr, int targetValue) {
            for(String s: arr){
                if(s.equals(101)){
                    System.out.print("YES!");
                    return true;
                }
            }
            return false;
    }*/


}



```