

```

package skillsdemo3hotel;
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import java.text.DecimalFormat;
import java.time.DayOfWeek;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
public class addingcode {
    static String[] cleaningRoster;
    static String[] guests;
    static Scanner sc = new Scanner(System.in);
    static DecimalFormat moneyFormat = new DecimalFormat("€###,##0.00");
    static String name;
    static String contactNumber;
    static String email;
    static String checkInDate;
    static String booking;
    static byte numOfNights;
    static float costOfStay;
    boolean validEmail;
    public static void main(String[] args) {
        cleaningRoster = new String[7]; // one entry for each day of the week
        guests = new String[100]; // array to keep track of guests staying in
the hotel
        boolean quit = false;
        printMenu();
        sc.close();
    }
    // print menu
    public static void printMenu() {
        int choice = 0;
        while (choice != 6) {
            System.out.println("Welcome to the Management System");
            System.out.println("1. Book in");
            System.out.println("2. Check out");
            System.out.println("3. Generate key code");
            System.out.println("4. Print cleaning roster");
            System.out.println("5. Change/Set the promo code");
            System.out.println("6. Quit");
            System.out.print("Enter your choice: ");
            try {
                choice = sc.nextInt();
            } catch (Exception e) {
                System.out.println("Invalid choice. Please try again.");
                sc.nextLine(); // consume the extra line left by nextInt()
                continue;
            }
            handleChoice(choice);
        }
    }
    public static void handleChoice(int choice) {
        switch (choice) {
            case 1:
                bookIn();
                break;
            case 2:

```

```

        checkOut();
        break;
    case 3:
        generateKeyCode();
        break;
    case 4:
        printCleaningRoster();
        break;
    case 5:
        changePromoCode();
        break;
    case 6:
        System.out.println("Goodbye!");
        break;
    default:
        System.out.println("Invalid choice. Please try again.");
        break;
    }
}

public static void bookIn() {
    // Prompt the user for information Dylan part
    System.out.println("Please enter your name:");
    name = sc.next();
    // Prompt the user for their contact number
    System.out.println("Please enter your contact number:");
    contactNumber = sc.next();
    // Prompt the user for their email address
    String emailAddress = "";
    boolean validEmail = false;
    while (!validEmail) {
        System.out.println("Please enter your email address:");
        emailAddress = sc.next();
        if (emailAddress.contains("@") && emailAddress.contains(".")) {
            validEmail = true;

            validEmail = true;
        }
        else {
            System.out.println("Invalid email address, please re-enter
your email address.");
        }
    }
    // dylan
    // Prompt the user for their room number
    int roomNumber = 0;
    boolean validRoomNumber = false;
    while (!validRoomNumber) {
        System.out.println("Please enter your room number (between 100 and
599):");
        roomNumber = sc.nextInt();
        if (roomNumber >= 100 && roomNumber <= 599) {
            validRoomNumber = true;
        } else {
            System.out.println("Invalid room number, please enter a number
between 100 and 599.");
        }
    }
}

```

```

// Prompt the user for the check-in date
System.out.println("Please enter the check-in date (DD/MM/YYYY):");
String checkInDate = sc.next();
// Prompt the user for the number of nights they will be staying
System.out.println("How many nights will you be staying?");
byte numOfNights = sc.nextByte();
// Calculate the cost of stay
float costOfStay = numOfNights * 105.0f;
System.out.print("Enter promo code (leave blank for none): ");
String inputPromoCode = sc.nextLine();
double discount = 0.0;
if (inputPromoCode.equals(inputPromoCode)) {
    discount = 0.1; // 10% discount for valid promo code
    System.out.println("Promo code applied.");
} else {
    System.out.println("Invalid promo code.");
}
// Update the cleaning roster for the days of the booking
LocalDate startDate = LocalDate.parse(checkInDate,
DateTimeFormatter.ofPattern("dd/MM/yyyy"));
for (int i = 0; i < numOfNights; i++) {
    DayOfWeek dayOfWeek = startDate.plusDays(i).getDayOfWeek();
    int dayIndex = dayOfWeek.getValue() - 1;
    cleaningRoster[dayIndex] += "\n" + "Room " + roomNumber + ": " +
name;

    // apply promo code

}

// Generate a booking reference number
int bookingReference = (int) (Math.random() * 1000000);
// Display the booking information
System.out.println("Booking successful! Here is your booking
information:");
System.out.println("Name: " + name);
System.out.println("Contact number: " + contactNumber);
System.out.println("Email address: " + emailAddress);
System.out.println("Room number: " + roomNumber);
System.out.println("Check-in date: " + checkInDate);
System.out.println("Number of nights: " + numOfNights);
System.out.println("Cost of stay: €" + costOfStay);
System.out.println("Booking reference number: " + bookingReference);
// Added the closing brace for the method#
// Prompt the user for their credit card details
boolean validCard = false;
long creditCardNumber = 0;
int expiryMonth = 0;
int expiryYear = 0;
while (!validCard) {
    System.out.println("Please enter your credit card number: to
confirm payment");
    creditCardNumber = sc.nextLong();
    if (creditCardNumber < 1000000000000000L || creditCardNumber >
9999999999999999L) {
        System.out.println("Invalid credit card number, please re-
enter your credit card number.");
    }
}

```

```

        continue;
    }
    System.out.println("Please enter the expiry month (MM):");
    expiryMonth = sc.nextInt();
    if (expiryMonth < 1 || expiryMonth > 12) {
        System.out.println("Invalid expiry month, please re-enter your
expiry month.");
        continue;
    }
    System.out.println("Please enter the expiry year (YYYY):");
    expiryYear = sc.nextInt();
    if (expiryYear < LocalDate.now().getYear() || expiryYear >
LocalDate.now().getYear() + 10) {
        System.out.println("Invalid expiry year, please re-enter your
expiry year.");
        continue;
    }
    validCard = true;
}

// Mask the credit card number
String maskedCardNumber = "XXXX-XXXX-XXXX-" +
String.valueOf(creditCardNumber).substring(12);
// Construct the email message
System.out.println( "Dear " + name + ",\n\n" +
    "On behalf of our entire staff, we would like to welcome you
to Hotel Siochan and hope that you " +
    "will have a memorable experience.\n\n" +
    "Your stay details are as follows:\n" +
    "Name: " + name + "\n" +
    "Contact number: " + contactNumber + "\n" +
    "Email address: " + emailAddress + "\n" +
    "Room number: " + roomNumber + "\n" +
    "Check-in date: " + checkInDate + "\n" +
    "Number of nights: " + numOfNights + "\n" +
    "Cost of stay: €" + costOfStay + "\n" +
    "Booking reference number: " + bookingReference + "\n" +
    "Credit card number: " + maskedCardNumber + "\n" +
    "Key code: " + roomNumber + bookingReference + "\n\n" +
    "If you need anything during your stay, our \n\n staff will be
pleased to assist you at any time. " +
    "Just call the front desk and we will be at your service.\n\n
You can also text us at 083-1234567 " +
    "or on Facebook Messenger hotelSiochan@facebook.com\n\n" +
    "Best regards,\n" +
    "Hotel Siochan staff");

}

// cleaning roster so that the employees can see what rooms are occupied
and which are not
public static void printCleaningRoster() {
    System.out.println("Cleaning Roster:");
    for (int i = 0; i < cleaningRoster.length; i++) {
        System.out.println(DayOfWeek.of(i + 1).toString() + ": " +
cleaningRoster[i]);
    }
}
}

```

```

// to allow the employee to check visitor out and ask if they used any extra
facility's
public static void checkOut() {
    System.out.println("Checking out a room");
    System.out.println("Please enter the room number:");
    int roomNumber = sc.nextInt();
    boolean hotelParking = false;
    boolean hotelSpa = false;
    float additionalCost = 0.0f;
    System.out.println("Did the guest avail of hotel parking? (y/n)");
    String input = sc.next();
    if (input.equalsIgnoreCase("y")) {
        hotelParking = true;
        additionalCost += 15.0f;
    }
    System.out.println("Did the guest avail of the hotel's spa?
(y/n)");
    input = sc.next();
    if (input.equalsIgnoreCase("y")) {
        hotelSpa = true;
        additionalCost += 60.0f;
    }
    if (additionalCost > 0.0f) {
        System.out.println("The additional cost for facilities is: " +
moneyFormat.format(additionalCost));
        System.out.println("Please enter the guest's credit card number
for payment:");
        String creditCardNumber = sc.next();
    }
    // updating cleaning roster when a visitor checkout
    for (int i = 0; i < cleaningRoster.length; i++) {
        if (cleaningRoster[i] != null) {
            String[] bookingDetails = cleaningRoster[i].split(":");
            String bookedRoomNumber = bookingDetails[0].substring(5);
            if (bookedRoomNumber.equals(Integer.toString(roomNumber)))
{
                System.out.println("Room " + roomNumber + " has been
checked out.");
                String checkInDate = bookingDetails[1].substring(1);
                DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("d/M/yyyy");
                LocalDate checkIn = LocalDate.parse(checkInDate,
formatter);
                int numOfDay =
Integer.parseInt(bookingDetails[2].substring(1));
                LocalDate checkOut = checkIn.plusDays(numOfDay);
                DayOfWeek checkOutDay = checkOut.getDayOfWeek();
                cleaningRoster[i] = null;
                System.out.println("Room " + roomNumber + " was checked
in on " + checkInDate + " and checked out on " + checkOut.format(formatter) +
" (" + checkOutDay + ").");
                break;
            }
        }
    }

    // Display a message indicating that the room has been checked out

```

```

        System.out.println("Room " + roomNumber + " has been checked
out.");
    }
    // making promo code
    public static String changePromoCode() {
        System.out.println("Enter the new promo code:");
        String newPromoCode = sc.next();
        String promoCode = newPromoCode;
        System.out.println("Promo code updated to: " + promoCode);

        String characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
        int length = 8;
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < length; i++) {
            int index = (int) (Math.random() * characters.length());
            sb.append(characters.charAt(index));
        }
        return "HOTEL10-" + sb.toString();
    }

    // key code maker for rooms
    public static void generateKeyCode() {
        System.out.println("Generating key code...");
        int keyCode = (int) (Math.random() * 1000000);
        System.out.println("Key code generated: " + keyCode);
    }
}

```

Data Dictionary

Variables

booking: A string that stores the booking reference number.
bookingReference: Integer variable to store the randomly generated booking reference number.
checkInDate: A string that stores the check-in date of the guest in the format "DD/MM/YYYY".
choice: Integer variable to store the user's choice in the main menu.
cleaningRoster: An array of strings that stores the names of guests staying in the hotel for each day of the week.
contactNumber: A string that stores the contact number of the guest.
contactNumber: A String variable that stores the contact number of the user.
costOfStay: A float that stores the total cost of the guest's stay.

creditCardNumber: A long variable that stores the entered credit card number.
dayIndex: Integer variable to store the index of the day in the cleaningRoster array.
dayOfWeek: DayOfWeek object to store the day of the week for a specific date.
discount: Double variable to store the discount amount based on the promo code.
email: A string that stores the email address of the guest.
emailAddress: String variable to store the email address entered by the user during email validation.
expiryMonth: An integer variable that stores the entered expiry month of the credit card.
expiryYear: An integer variable that stores the entered expiry year of the credit card.
guests: An array of strings that keeps track of guests staying in the hotel.
inputPromoCode: String variable to store the promo code entered by the user.
keyCode: An integer variable that stores the generated key code. The key code is randomly generated using the Math.random() method.
maskedCardNumber: A String variable that stores the masked credit card number for display.
moneyFormat: An instance of the DecimalFormat class used to format currency values.
name: A string that stores the name of the guest.
numOfNights: A byte that stores the number of nights the guest will be staying.
roomNumber: Integer variable to store the room number selected by the guest.
sc: An instance of the Scanner class used to read input from the user.
startDate: LocalDate object to store the parsed check-in date.
validCard: A boolean variable that represents whether the entered credit card details are valid or not.
validEmail: A boolean that indicates whether the email address provided by the guest is valid.
validRoomNumber: Boolean variable to check if the entered room number is valid.

Methods

`bookIn()`: A method that prompts the user for information such as name, contact number, email address, room number, check-in date, and number of nights, calculates the cost of stay, updates the cleaning roster, applies a promo code discount if provided, generates a booking reference number, and displays the booking information to the user.

`changePromoCode()`: A method that allows the user to change or set a promo code for the hotel.

`checkOut()`: A method that handles the check-out process for guests, calculates the total cost of the stay, updates the cleaning roster, and removes the guest from the guests array.

`generateKeyCode()`: A method that generates a key code for the guest's room and displays it to the user.

`handleChoice(int choice)`: A method that handles the user's choice from the main menu and calls the corresponding method accordingly.

`printCleaningRoster()`: A method that displays the cleaning roster for each day of the week.

`printMenu()`: A method that displays the main menu options to the user.

Algorithm – Entire Program

1. Declare and initialize static variables:
 - Declare and initialize a String array "cleaningRoster" with 7 elements, one for each day of the week.
 - Declare and initialize a String array "guests" with 100 elements, to keep track of guests staying in the hotel.
 - Declare and initialize a Scanner object "sc" to read input from the console.
 - Declare and initialize a DecimalFormat object "moneyFormat" to format currency values.
 - Declare and initialize various String, byte, and float variables to store guest information, such as name, contact number, email, check-in date, booking, number of nights, and cost of stay.
 - Declare a boolean variable "validEmail" to keep track of whether the entered email address is valid.
2. Define a "main" method:
 - Initialize "cleaningRoster" and "guests" arrays.
 - Set a boolean variable "quit" to false.
 - Call "printMenu()" method to display the menu to the user.
 - Close the Scanner object "sc".
3. Define a "printMenu()" method:
 - Display the menu options to the user using `System.out.println` statements.

- Use a while loop to repeatedly read the user's choice using the "sc.nextInt()" method.
 - Call the "handleChoice()" method with the user's choice to perform the corresponding action.
 - Continue looping until the user enters 6 to quit.
4. Define a "handleChoice()" method:
 - Use a switch statement to handle different menu choices:
 - Case 1: Call "bookIn()" method to handle guest check-in.
 - Case 2: Call "checkOut()" method to handle guest check-out.
 - Case 3: Call "generateKeyCode()" method to generate a key code.
 - Case 4: Call "printCleaningRoster()" method to print the cleaning roster.
 - Case 5: Call "changePromoCode()" method to change or set the promo code.
 - Case 6: Print a goodbye message and exit.
 - Default: Display an error message for invalid choices.
 5. Define a "bookIn()" method:
 - Prompt the user for guest information, such as name, contact number, email, room number, check-in date, and number of nights.
 - Validate the entered email address using a regular expression pattern.
 - Validate the entered room number to be between 100 and 599.
 - Calculate the cost of stay based on the number of nights.
 - Update the cleaning roster with the guest's information for each day of the booking.
 - Prompt the user for a promo code and apply a discount if it's valid.
 - Generate a booking reference number.
 - Display the booking information to the user.
 6. Define a "checkOut()" method:
 - Prompt the user for the guest's name or booking reference number.
 - Search the "guests" array for the guest's information.
 - If found, remove the guest's information from the "guests" array and update the cleaning roster.
 - Display the check-out information to the user.
 7. Define a "generateKeyCode()" method:
 - Generate a random key code.
 - Display the key code to the user.
 8. Define a "printCleaningRoster()" method:
 - Print the cleaning roster to the console.
 9. Define a "changePromoCode()" method:
 - Prompt the user for a new promo code.
 - Update the promo code with the new value.
 10. End of program.

Algorithm – My Code

Print Cleaning Roster

1. Create a method named printCleaningRoster that takes no arguments and has a void return type.
2. Within the method, use System.out.println to print the string "Cleaning Roster:" to the console.

3. Use a for loop to iterate over the cleaningRoster array. The loop variable i should start from 0 and go up to the length of cleaningRoster minus 1.
4. Inside the loop, use `DayOfWeek.of(i + 1)` to get the `DayOfWeek` enum value corresponding to the current index i incremented by 1. Convert it to a string using `toString()`.
5. Concatenate the string obtained from `DayOfWeek.of(i + 1).toString()` with the string ": " and the value at the current index of cleaningRoster array.
6. Print the concatenated string using `System.out.println` to the console.
7. After the loop completes, the cleaning roster will be printed with each day of the week followed by the corresponding value from the cleaningRoster array.

Validate Email Address

1. Import the necessary package `java.util.Scanner` for input reading.
2. Define a class named `SkillsDemo3Email` with a public static void `main(String[] args)` method as the entry point of the program.
3. Inside the main method, create a `Scanner` object named `input` to read input from the console.
4. Declare a `String` variable named `email` to store the email address entered by the user.
5. Declare a boolean variable named `valid` and initialize it to `false`. This variable will be used to control the loop for email address validation.
6. Start a while loop with the condition `!valid`, which means the loop will continue until a valid email address is entered by the user.
7. Inside the loop, use `System.out.println` to display the prompt "Please enter your email address:" to the console.
8. Use `input.nextLine()` to read a line of input from the user and store it in the `email` variable.
9. Call the `isValidEmail(email)` method with the `email` variable as an argument to check if the entered email address is valid. If it returns `true`, set `valid` to `true` to exit the loop, and display a message "Thank you! Your email address is: " followed by the entered email address using `System.out.println`.
10. If the entered email address is not valid, display a message "Invalid email address. Please try again." using `System.out.println`.
11. After the loop completes, the program will exit as a valid email address has been entered.
12. Define a public static boolean `isValidEmail(String email)` method outside the main method to check if an email address is valid.
13. Inside the `isValidEmail` method, declare two boolean variables `hasAtSign` and `hasFullStop` and initialize them to `false`. These variables will be used to check if the email address contains an '@' and a '.' respectively.
14. Use a for loop to iterate over each character in the email string. Use `email.charAt(i)` to get the character at index i.
15. Inside the loop, use an if statement to check if the current character is '@'. If it is, set `hasAtSign` to `true`.
16. Use another if statement to check if the current character is '.'. If it is, set `hasFullStop` to `true`.

17. After the loop completes, return `hasAtSign && hasFullStop`, which means the email address is valid if it contains both '@' and '.'.
18. The `isValidEmail` method is now complete, and it will return true if the entered email address is valid, and false otherwise.

Validate Room Number

1. Define a class named `SkillsDemo3Room` with a private nested static class `RoomNumberValidation` inside it.
2. Inside the `RoomNumberValidation` class, declare a private static integer array `roomNumbers` with a size of 500 to store the entered room numbers.
3. Declare a private static integer variable `index` and initialize it to 0. This variable will be used as an index to keep track of the number of room numbers entered.
4. Define a public static boolean method named `validateRoomNumber` with an integer parameter `roomNumber` to validate a given room number.
5. Inside the `validateRoomNumber` method, use an if statement to check if the `roomNumber` is less than 100 or greater than 599, which are the valid room number ranges. If it is, display a message "Invalid room number" using `System.out.println` and return false.
6. Use a for loop to iterate over the `roomNumbers` array from index 0 to index (exclusive), which represents the number of room numbers entered so far.
7. Inside the loop, use another if statement to check if the current element in the `roomNumbers` array is equal to the `roomNumber` being validated. If it is, display a message "Room number already entered" using `System.out.println` and return false.
8. If the `roomNumber` is not found in the `roomNumbers` array, assign it to the `roomNumbers` array at the index position, and increment `index` by 1 to keep track of the number of room numbers entered.
9. Return true to indicate that the `roomNumber` is valid and has been successfully added to the `roomNumbers` array.
10. The `validateRoomNumber` method is now complete, and it will return false if the `roomNumber` is invalid or has already been entered, and true otherwise. Note that the `roomNumbers` array will store up to 500 unique room numbers entered through successive calls to the `validateRoomNumber` method.

Loop until credit number or cancel the booking

1. Initialize a boolean variable `validCard` to false to represent whether the credit card details entered by the user are valid or not.
2. Declare and initialize `creditCardNumber`, `expiryMonth`, and `expiryYear` to 0 to store the user's credit card details.
3. Use a while loop with the condition `!validCard` to repeatedly prompt the user for their credit card details until a valid card is entered.
4. Inside the loop, use `System.out.println` to display a prompt asking the user to enter their credit card number.
5. Use `sc.nextLong()` (assuming `sc` is a `Scanner` object) to read the credit card number entered by the user as a long.

6. Use an if statement to check if the creditCardNumber is less than 1000000000000000L or greater than 9999999999999999L, which are the valid credit card number ranges. If it is not, display a message "Invalid credit card number, please re-enter your credit card number." using System.out.println, and use continue to go to the next iteration of the loop.
7. If the credit card number is valid, prompt the user to enter the expiry month (MM) using System.out.println.
8. Use sc.nextInt() to read the expiry month entered by the user as an integer.
9. Use another if statement to check if the expiryMonth is less than 1 or greater than 12, which are the valid expiry month ranges. If it is not, display a message "Invalid expiry month, please re-enter your expiry month." using System.out.println, and use continue to go to the next iteration of the loop.
10. If the expiry month is valid, prompt the user to enter the expiry year (YYYY) using System.out.println.
11. Use sc.nextInt() to read the expiry year entered by the user as an integer.
12. Use another if statement to check if the expiryYear is less than the current year or greater than the current year + 10, which are the valid expiry year ranges. If it is not, display a message "Invalid expiry year, please re-enter your expiry year." using System.out.println, and use continue to go to the next iteration of the loop.
13. If the expiry year is valid, set validCard to true to indicate that the credit card details are valid, and exit the loop.
14. The loop will continue to prompt the user for their credit card details until a valid card is entered. Once a valid card is entered, the loop will be exited, and the credit card details will be stored in the creditCardNumber, expiryMonth, and expiryYear variables for further processing.

Read in Guest details(with no validation) and calculate cost of stay.

1. Initialize a Scanner object sc to read input from the user.
2. Initialize a DecimalFormat object moneyFormat with the pattern "€###,##0.00" to format currency values.
3. Declare and initialize variables to store user inputs: name as a string to store the user's name, contactNumber as a string to store the user's contact number, emailAddress as a string to store the user's email address, validEmail as a boolean to keep track of whether the entered email address is valid or not, numOfNights as a byte to store the number of nights the user will be staying, and costOfStay as a float to store the cost of the stay.
4. Use System.out.println to display a prompt asking the user to enter their name.
5. Use sc.next() to read the user's name entered as a string and store it in the name variable.
6. Display a prompt asking the user to enter their contact number using System.out.println.
7. Use sc.next() to read the user's contact number entered as a string and store it in the contactNumber variable.

8. Use a while loop with the condition !validEmail to repeatedly prompt the user for their email address until a valid email address is entered.
9. Inside the loop, display a prompt asking the user to enter their email address using System.out.println.
10. Use sc.next() to read the user's email address entered as a string and store it in the emailAddress variable.
11. Use an if statement to check if the emailAddress contains "@" and "." to determine if it's a valid email address. If it is, set validEmail to true to exit the loop. Otherwise, display a message "Invalid email address, please re-enter your email address." using System.out.println, and continue with the next iteration of the loop.
12. Display a prompt asking the user to enter the number of nights they will be staying using System.out.println.
13. Use sc.nextByte() to read the number of nights entered by the user as a byte and store it in the numOfNights variable.
14. Calculate the costOfStay by multiplying numOfNights by 105.
15. The algorithm ends here, and you can continue with further processing of the collected user inputs or displaying the calculated cost of stay as needed.

Test Data

1. Booking in and booking out without spa or parking.
2. Checking if more than one person can be checked in and if email is put in wrong and checking the cleaning roster.
3. Test data check Generate key code and change/Set the promo code and test.
4. No person in book and out with spa and parking.

Test data 1

Booking in and booking out without spa or parking

Booking in and booking out without spa or parking

Input 1

Output please enter your name

Input harry

Output please enter your contact number

Input 09873845673

Output please enter email address

Input harry@gmail.com

Output Please enter room number (between 100 and 599)

Input 222

Output Please enter the check-in date (DD/MM/YYYY):

Input date 26/04/2023

Output How many nights will you be staying?

Input 2

Output Enter promo code (leave blank for none): Promo code applied.

output Name: harry

Contact number: 09873845673

Email address: harry@gmail.com

Room number: 222

Check-in date: 26/04/2023

Number of nights: 2

Cost of stay: €210.0

Booking reference number: 529319

Output Please enter your credit card number: to confirm payment

input4929643253685942

output Please enter the expiry month (MM):

input 04

output Please enter the expiry year (YYYY):

input 2023

output Dear harry,

On behalf of our entire staff, we would like to welcome you to Hotel Siochan and hope that you will have a memorable experience.

Your stay details are as follows:

Name: harry

Contact number: 09873845673

Email address: harry@gmail.com

Room number: 222

Check-in date: 26/04/2023

Number of nights: 2

Cost of stay: €210.0

Booking reference number: 529319

Credit card number: XXXX-XXXX-XXXX-5942

Key code: 222529319

If you need anything during your stay, our

staff will be pleased to assist you at any time. Just call the front desk and we will be at your service.

You can also text us at 083-1234567 or on Facebook Messenger [hotelSiochan@facebook.com](https://www.facebook.com/hotelSiochan)

Output

Enter your choice:

Input 2

Output Checking out a room

Output

Input 222

Output Did the guest avail of hotel parking? (y/n)

Input no

Output Did the guest avail of the hotel's spa? (y/n)

Input no

output Room 222 has been checked out.

Actual output of data

```
adding code java Application C:\Users\harry\p2\poo\plugins\org.eclipse.jdt.ui\openjakarta.jar\win32-x86_64-1700520
Welcome to the Management System
1. Book in
2. Check out
3. Generate key code
4. Print cleaning roster
5. Change/Set the promo code
6. Quit
Enter your choice: 1
Please enter your name:
harry
Please enter your contact number:
09873845673
Please enter your email address:
harry@gmail.com
Please enter your room number (between 100 and 599):
222
Please enter the check-in date (DD/MM/YYYY):
26/04/2023
How many nights will you be staying?
2
Enter promo code (leave blank for none): Promo code applied.
Enter promo code (leave blank for none):
Promo code applied.
Booking successful! Here is your booking information:
Name: harry
Contact number: 09873845673
Email address: harry@gmail.com
Room number: 222
Check-in date: 26/04/2023
Number of nights: 2
Cost of stay: €210.0
Booking reference number: 529319
Please enter your credit card number: to confirm payment
4929643253685942
Please enter the expiry month (MM):
04
Please enter the expiry year (YYYY):
2023
```



```
Dear harry,  
  
On behalf of our entire staff, we would like to welcome you to Hotel Siochan and hope that you will have a memorable experience.  
  
Your stay details are as follows:  
Name: harry  
Contact number: 09873845673  
Email address: harry@gmail.com  
Room number: 222  
Check-in date: 26/04/2023  
Number of nights: 2  
Cost of stay: €210.0  
Booking reference number: 529319  
Credit card number: XXXX-XXXX-XXXX-5942  
Key code: 222529319  
  
If you need anything during your stay, our  
  
staff will be pleased to assist you at any time. Just call the front desk and we will be at your service.  
  
You can also text us at 083-1234567 or on Facebook Messenger hotelSiochan@facebook.com
```

```
Enter your choice: 2  
Checking out a room  
Please enter the room number:  
222  
Did the guest avail of hotel parking? (y/n)  
n  
Did the guest avail of the hotel's spa? (y/n)  
n  
Room 222 has been checked out.
```

Test data 2

Checking if more than one person can be checked in and if email put in wrong and checking cleaning roster.

output

Input 1

Output please enter your name

Input bob

Output please enter your contact number

Input 0857368347

output Please enter your email address:

input bob@gmailcom

output Invalid email address, please re-enter your email address.

Output Please enter your email address:

inputbobgmail.com

output Invalid email address, please re-enter your email address.

Output Please enter your email address:

Input bob@gmail.com

Output Please enter your room number (between 100 and 599):

Input 223

Output Please enter the check-in date (DD/MM/YYYY):

Input 25/04/2023

Output How many nights will you be staying?

Output Enter promo code (leave blank for none): Promo code applied.

Input 2

Output text saying prices and other info

Input cad details

Output all info in a email format

Input 1

Output Name:

harry

Contact number:

09873845673

Email address:

harry@gmail.com

Room number:

222

Check-in date:

26/04/2023

Number of nights:

2

Output details of stay

Output r credit card number

Input 4929643253685942

Please enter the expiry month (MM):

04

output Please enter the expiry year (YYYY):

input 2023

output details in email and send to customer about stay

Actual Data

```
Welcome to the Management System
1. Book in
2. Check out
3. Generate key code
4. Print cleaning roster
5. Change/Set the promo code
6. Quit
Enter your choice: 1
Please enter your name:
bob
Please enter your contact number:
0857368347
Please enter your email address:
bob@gmailcom
Invalid email address, please re-enter your email address.
Please enter your email address:
bobgmail.com
Invalid email address, please re-enter your email address.
Please enter your email address:
bob@gmail.com
Please enter your room number (between 100 and 599):
223
Please enter the check-in date (DD/MM/YYYY):
25/04/2023
How many nights will you be staying?
2
Enter promo code (leave blank for none): Promo code applied.
Enter promo code (leave blank for none):
Promo code applied.
Booking successful! Here is your booking information:
Name: bob
Contact number: 0857368347
Email address: bob@gmail.com
Room number: 223
Check-in date: 25/04/2023
Number of nights: 2
Cost of stay: €210.0
Booking reference number: 723724
Please enter your credit card number: to confirm payment
4485753072920598
Please enter the expiry month (MM):
05
Please enter the expiry year (YYYY):
2027
```

```
Best regards,  
Hotel Siochan staff  
Welcome to the Management System  
1. Book in  
2. Check out  
3. Generate key code  
4. Print cleaning roster  
5. Change/Set the promo code  
6. Quit  
Enter your choice: 1  
Please enter your name:  
mark  
Please enter your contact number:  
08582353053  
Please enter your email address:  
mark@hotmail.com  
Please enter your room number (between 100 and 599):  
224  
Please enter the check-in date (DD/MM/YYYY):  
26/04/2023  
How many nights will you be staying?  
3  
Enter promo code (leave blank for none): Promo code applied.  
Enter promo code (leave blank for none):  
Promo code applied.  
Enter promo code (leave blank for none):  
Promo code applied.  
Booking successful! Here is your booking information:  
Name: mark  
Contact number: 08582353053  
Email address: mark@hotmail.com  
Room number: 224  
Check-in date: 26/04/2023  
Number of nights: 3  
Cost of stay: €315.0  
Booking reference number: 737052  
Please enter your credit card number: to confirm payment  
4532817841433176  
Please enter the expiry month (MM):  
09  
Please enter the expiry year (YYYY):  
2027
```

Dear mark,

On behalf of our entire staff, we would like to welcome you to Hotel Siochan and hope that you will have a memorable experience.

Your stay details are as follows:

Name: mark

Contact number: 08582353053

Email address: mark@hotmail.com

Room number: 224

Check-in date: 26/04/2023

Number of nights: 3

Cost of stay: €315.0

Booking reference number: 737052

Credit card number: XXXX-XXXX-XXXX-3176

Key code: 224737052

If you need anything during your stay, our

staff will be pleased to assist you at any time. Just call the front desk and we will be at your service.

You can also text us at 083-1234567 or on Facebook Messenger hotelSiochan@facebook.com

```
Best regards,  
Hotel Siochan staff  
Welcome to the Management System  
1. Book in  
2. Check out  
3. Generate key code  
4. Print cleaning roster  
5. Change/Set the promo code  
6. Quit  
Enter your choice: 4  
Cleaning Roster:  
MONDAY: null  
TUESDAY: null  
Room 223: bob  
WEDNESDAY: null  
Room 223: bob  
Room 224: mark  
THURSDAY: null  
Room 224: mark  
FRIDAY: null  
Room 224: mark  
SATURDAY: null  
SUNDAY: null
```

Test data 3 check Generate key code and change/Set the promo code and test

Name Mary

Number 08578543045

Email mary@hotmail.com

Room number 226

Check in date 25/04/2023

Num of nights 1

Promo code set hotel10

Card details 4428064615661669

expiry month :01

expiry year 2027

```
Welcome to the Management System
1. Book in
2. Check out
3. Generate key code
4. Print cleaning roster
5. Change/Set the promo code
6. Quit
Enter your choice: 5
Enter the new promo code:
hotel10
Promo code updated to: hotel10
Welcome to the Management System
1. Book in
2. Check out
3. Generate key code
4. Print cleaning roster
5. Change/Set the promo code
6. Quit
Enter your choice: 1
Please enter your name:
mary
Please enter your contact number:
08578543045
Please enter your email address:
mary@hotmail.com
Please enter your room number (between 100 and 599):
226
Please enter the check-in date (DD/MM/YYYY):
25/04/2023
How many nights will you be staying?
1
Enter promo code (leave blank for none): hotel10
Promo code applied.
Booking successful! Here is your booking information:
Name: mary
Contact number: 08578543045
Email address: mary@hotmail.com
Room number: 226
Check-in date: 25/04/2023
Number of nights: 1
Cost of stay: €105.0
Booking reference number: 462866
Please enter your credit card number: to confirm payment
4428064615661669
Please enter the expiry month (MM):
01
Please enter the expiry year (YYYY):
2027
```

```
Dear mary,

On behalf of our entire staff, we would like to welcome you to Hotel Siochan and hope that you will have a memorable experience.

Your stay details are as follows:
Name: mary
Contact number: 08578543045
Email address: mary@hotmail.com
Room number: 226
Check-in date: 25/04/2023
Number of nights: 1
Cost of stay: €105.0
Booking reference number: 462866
Credit card number: XXXX-XXXX-XXXX-1669
Key code: 226462866

If you need anything during your stay, our
staff will be pleased to assist you at any time. Just call the front desk and we will be at your service.

You can also text us at 083-1234567 or on Facebook Messenger hotelSiochan@facebook.com
```

Test Data 4

One person books in and out with spa and parking.

Name: Jeff

Number 085754234126

Email jeff@gmail.com

Room number 300

Check in date 25/04/2023

Num of nights 2

Promo code set none

Card details 4428064615661669

expiry month :02

expiry year 2027

spa : yes

parking ; yes

price euro 75

Actual Data

```
Welcome to the Management System
1. Book in
2. Check out
3. Generate key code
4. Print cleaning roster
5. Change/Set the promo code
6. Quit
Enter your choice: 1
Please enter your name:
jeff
Please enter your contact number:
085754234126
Please enter your email address:
jeff@gmail.com
Please enter your room number (between 100 and 599):
300
Please enter the check-in date (DD/MM/YYYY):
25/04/2023
How many nights will you be staying?
4
Enter promo code (leave blank for none): Promo code applied.
Booking successful! Here is your booking information:
Name: jeff
Contact number: 085754234126
Email address: jeff@gmail.com
Room number: 300
Check-in date: 25/04/2023
Number of nights: 4
Cost of stay: €420.0
Booking reference number: 808512
Please enter your credit card number: to confirm payment
4428064615661669
Please enter the expiry month (MM):
02
Please enter the expiry year (YYYY):
2027
```

```
#Dear jeff,

On behalf of our entire staff, we would like to welcome you to Hotel Siochan and hope that you will have a memorable experience.

Your stay details are as follows:
Name: jeff
Contact number: 085754234126
Email address: jeff@gmail.com
Room number: 300
Check-in date: 25/04/2023
Number of nights: 4
Cost of stay: €420.0
Booking reference number: 808512
Credit card number: XXXX-XXXX-XXXX-1669
Key code: 300808512

If you need anything during your stay, our

staff will be pleased to assist you at any time. Just call the front desk and we will be at your service.

You can also text us at 083-1234567 or on Facebook Messenger hotelSiochan@facebook.com

Best regards,
```

```
Enter your choice: 2
Invalid choice. Please try again.
Welcome to the Management System
1. Book in
2. Check out
3. Generate key code
4. Print cleaning roster
5. Change/Set the promo code
6. Quit
Enter your choice: 2
Checking out a room
Please enter the room number:
222
Did the guest avail of hotel parking? (y/n)
y
Did the guest avail of the hotel's spa? (y/n)
y
The additional cost for facilities is: €75.00
Please enter the guest's credit card number for payment:
4428064615661669
Room 222 has been checked out.
Welcome to the Management System
1. Book in
2. Check out
3. Generate key code
4. Print cleaning roster
5. Change/Set the promo code
6. Quit
Enter your choice:
```

Documentation of all meetings

Hotel Siochan

Meeting #: 1

Date: 4th of April

Time: 18:30-18:55

Present: Dylan Coveney
Liam Lynch
Retrive Ncube

Task to be completed	To be completed by
Update cleaning roster when booking details are read in (Research arrays? Use a string?) - Liam	Liam
Update cleaning roster when a guest checks out - Liam	Liam
· Print cleaning roster - Dylan	Dylan
· Set up email (including formatting values) - Retrive	Retrive
· Print menu – (Book in, Check out, Generate key code, Print cleaning Roster, Change/Set the promo code) and loop until quit. - Liam	Liam
· Determine the method to be called from the chosen menu option. - Liam	Liam
· Validate email address (check and loop) - Dylan	Dylan
· Validate room number (check and loop) - Dylan	Dylan
· Validate credit card number	

(check and loop) – Coded once and called twice (for booking in and extra costs) - Retrive	Retrive
· Loop until credit card number or cancel the booking. - Dylan	Dylan
· Input a key code (must be validated) – Coded once and called twice - Retrive	Retrive
· Ask about (loop until Y/N) and calculate cost of extra facilities - Retrive	Retrive
· Deal with promo code is valid (loop until Y/N and calculate discount if necessary) - Retrive	Retrive
· Read in the guest details (with no validation) - Dylan	Dylan
· Calculate cost of stay - Dylan	Dylan
· Change/Set promo code - Retrive	Retrive

Notes: Tasks were assigned at this meeting. Some tasks had already been started or completed as there had been previous discussions through messages and they were assigned this way. The remaining tasks were distributed in this meeting

Date and time of next meeting: Undecided so far, we will be checking back every few days to see what we have done.

Hotel Siochan

Meeting #: 2

Date: 18th April 2023

Time: 9AM-10AM

Present: Liam Lynch
Dylan Coveney
Retrive Ncube

Review of previous meeting: In the previous meeting we assigned each of the tasks that we were required to do. Since the last meeting, Dylan and Liam completed the code they were required to do. Dylan is waiting on code from Retrieve to complete the final part of his code. Retrive was having trouble with her code and was unable to complete any of the code she was assigned in the previous weeks.

Tasks for next meeting:

Task to be completed	To be completed by
Complete last part of code once Retrive's code is finished	Dylan
Complete tasks previously assigned	Retrive

Notes:

Date and time of next meeting: 25th April 2023

Hotel Siochan

Meeting #: Final

Date: April 25th 2023 **Time:** 9AM-10AM

Present: Dylan Coveney
Liam Lynch
Retrive Ncube

Review of previous meeting: Since the last meeting the code for the entire project has been completed. The only work left to do is the algorithm, data dictionary and test data.

Task to be completed	Completed, partially completed or fully completed. If partially completed, indicate an estimated percentage of the work completed.
Algorithm, test data and data dictionary	Entire group.

Notes:

Sign off on this project:

Signed: Dylan Coveney
Liam Lynch
Retrive Ncube