IEEE COMPUTER SOCIETY
Mansoura Student Branch Chapter

HOTEL BOOKING SYSTM

by Shahd Tarek

ABOUT ME

- Shahd Tarek
- BIS Student At MET Academy
- Level 2
- Linkedin: Shahd Tarek
- GitHub: shahd-tareq

IEEE COMPUTER SOCIETY
Mansoura Student Branch Chapter

INTRODUCTION

Hotel Booking System is a console based application built in Dart using Object Oriented Programming. It allows guests to view available rooms, make bookings, and pay via different payment methods.

CLASS

1-Guest

2-Room

3-Hotel

4-Booking

5-Payment

CLASS - GUEST

Class Role: Represents a customer who interacts with the hotel system to make reservations and manage bookings.

Properties:

- name: String - Full name of the guest. Used to identify the user.
- email: String - Guest's email for communication.
- phoneNumber: String - Contact number for the guest.

Methods:

- displayInfo(): Displays the guest's full information.

```
class Guest {
  String name;
  String email;
  String Number;

  Guest({required this.name, required this.email, required this.Number});

  void displayInfo() {
    print("Guest Info:");
    print("Name: $name");
    print("Email: $email");
    print("Phone: $Number");
  }
}

```

ABSTRACT CLASS - ROOM

Class Role: Serves as the base class for different types of rooms in the hotel.

Properties:

- roomNumber: int - Unique number for the room.
- pricePerNight: double - The cost per night for the room.

Methods:

- displayDetails(): Displays the room's number and price per night.

```dart
abstract class Room {
  int roomNumber;
  double priceNight;

  Room({required this.roomNumber, required this.priceNight});

  void displayRoomDetails();
}

class SingleRoom extends Room {
  SingleRoom({required int roomNumber})
    : super(roomNumber: roomNumber, priceNight: 500);

  @override
  void displayRoomDetails() {
    print("Single Room:Room ($roomNumber) | \$${priceNight}>night");
  }
}

class DoubleRoom extends Room {
  DoubleRoom({required int roomNumber})
    : super(roomNumber: roomNumber, priceNight: 900);

  @override
  void displayRoomDetails() {
    print("Double Room:Room ($roomNumber) | \$${priceNight}>night");
  }
}

class SuiteRoom extends Room {
  SuiteRoom({required int roomNumber})
    : super(roomNumber: roomNumber, priceNight: 1200);

  @override
  void displayRoomDetails() {
    print("Suite Room :Room ($roomNumber )| \$${priceNight}>night");
  }
}

```

CLASS - BOOKING

Class Role: Represents a reservation made by a guest for a specific room and duration in the hotel.

Properties:

- guest: Guest - The guest making the booking.
- room: Room - The room being booked by the guest.
- nights: int - The number of nights the guest intends to stay.

Methods:

- calculateTotal(): Calculates the total cost of the booking by multiplying the price per night by the number of nights.
- displayBooking(): Displays the details of the booking including guest, room, and dates.

```dart
import 'guest.dart';
import 'room.dart';

class Booking {
  Guest guest;
  Room room;
  int nights;

  Booking({required this.guest, required this.room, required this.nights});

  double get totalPrice => room.priceNight * nights;

  void printDetails() {
    print("\n Booking Summary:");
    guest.displayInfo();
    room.displayRoomDetails();
    print("Nights: $nights");
    print("Total Price: \$${totalPrice}");
  }
}

```

ABSTRACT CLASS - PAYMENTMETHOD

Class Role:

Provides an interface for all payment methods that can be used to pay for bookings.

Methods:

- pay(double amount): Abstract method that processes the payment (to be implemented in subclasses).

```
abstract class PaymentMethod {
  void pay(double amount);
}

class CreditCardPayment extends PaymentMethod {
  String cardNumber;

  CreditCardPayment(this.cardNumber);

  @override
  void pay(double amount) {
    print("Paying \${amount} using Credit Card: $cardNumber");
  }
}

class CashPayment extends PaymentMethod {
  @override
  void pay(double amount) {
    print("Paying \${amount} in cash.");
  }
}

```

SUBCLASSES - CREDITCARDPAYMENT / CASHPAYMENT

Class Role:

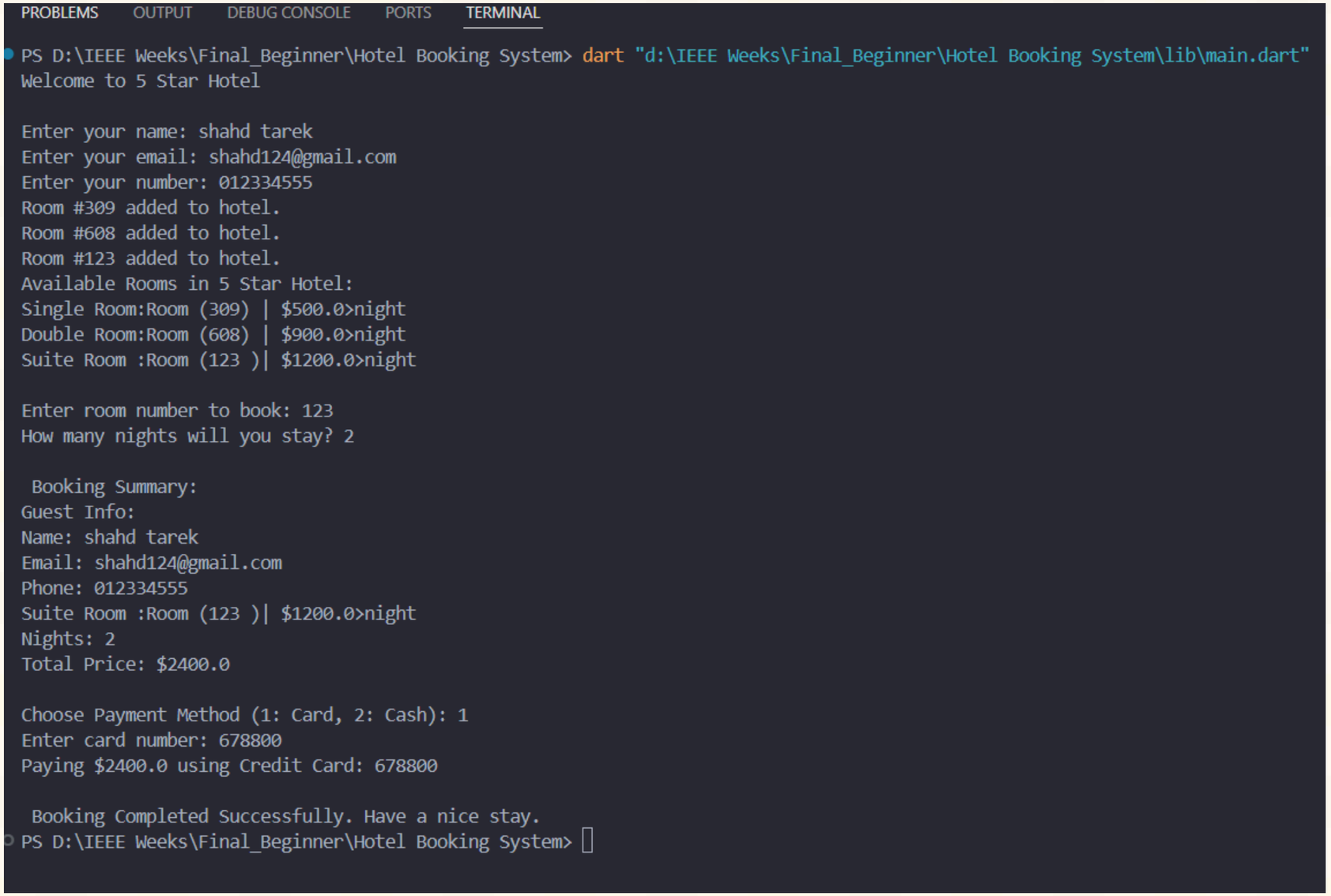
Implements different payment methods to process bookings:

- CreditCardPayment: Handles payments using credit card details.
- CashPayment: Handles payments made with cash.

Methods:

- **pay(double amount): Implemented in both subclasses to process payments according to the selected method.**

OUTPUT THIS SYSTEM



RELATIONSHIPS

- GUEST ↔ BOOKING (ONE TO MANY): A GUEST CAN MAKE MULTIPLE BOOKINGS.
- HOTEL ↔ ROOM (ONE TO MANY): A HOTEL CONTAINS MANY ROOMS.
- BOOKING ↔ ROOM (ONE TO ONE): A BOOKING IS ASSOCIATED WITH ONE SPECIFIC ROOM.
- BOOKING ↔ PAYMENTMETHOD (ONE TO ONE): A BOOKING USES ONE PAYMENT METHOD.

SUMMARY

THIS SYSTEM DEMONSTRATES REAL-WORLD HOTEL OPERATIONS USING CORE OOP PRINCIPLES SUCH AS:

- INHERITANCE (ROOM TYPES, PAYMENT TYPES)
- ABSTRACTION (ROOM AND PAYMENTMETHOD BASE CLASSES)
- COMPOSITION (BOOKING LINKS GUEST AND ROOM)
- POLYMORPHISM (PAYMENT HANDLING IN DIFFERENT PAYMENT TYPES)


THANK YOU