# Parking Management System For Theatres In POC

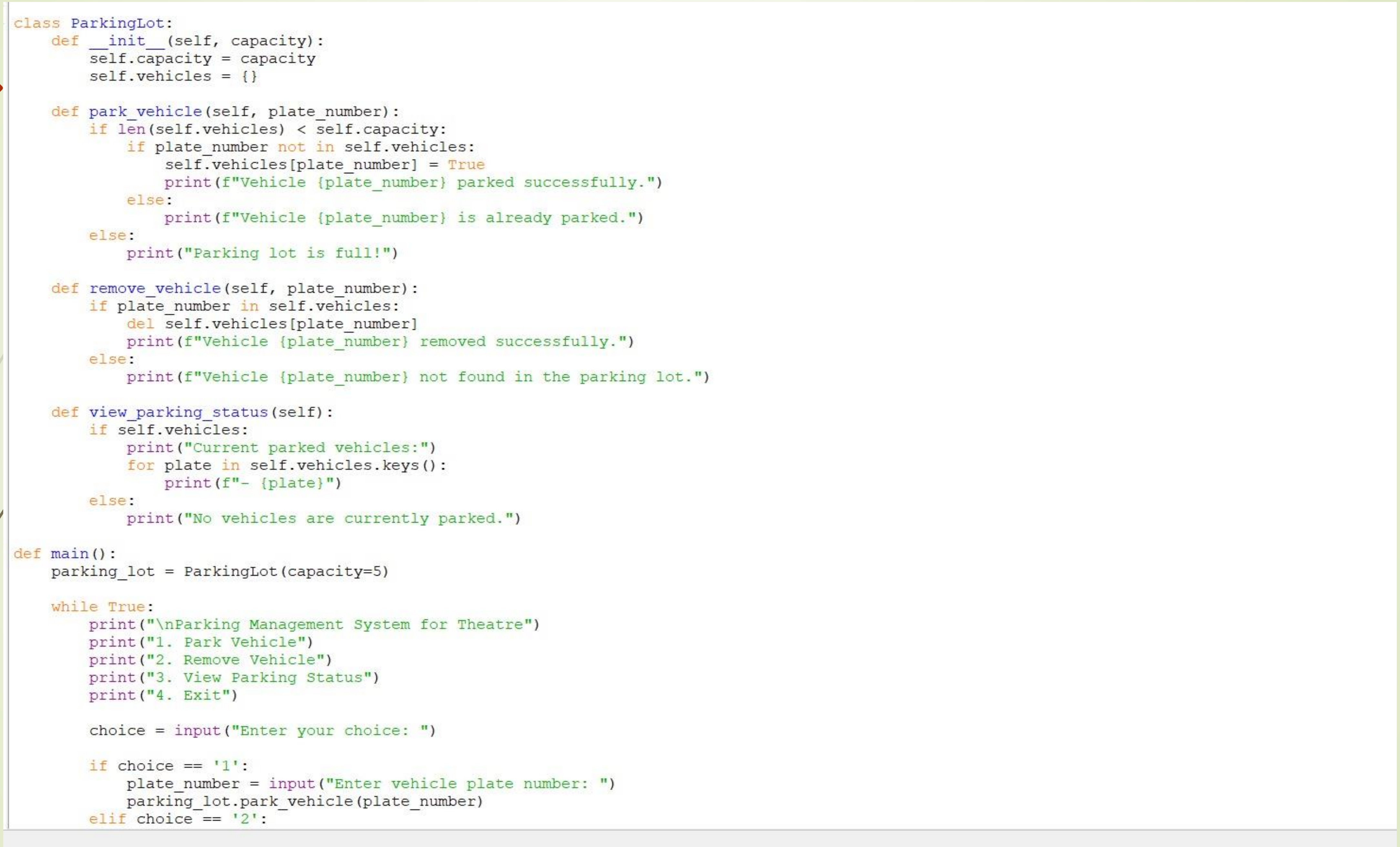
# Introduction

- A Parking Management System (PMS) for theatres aims to solve common challenges like traffic congestion , long wait times for parking spots, and confusion about available spaces .
- The goal of this Proof Of Concept (POC) is to demonstrate a parking management that enhances user experiences and optimizes parking space for theatres .
- It will include real-time availability , reservation features ,and seamless payment options.

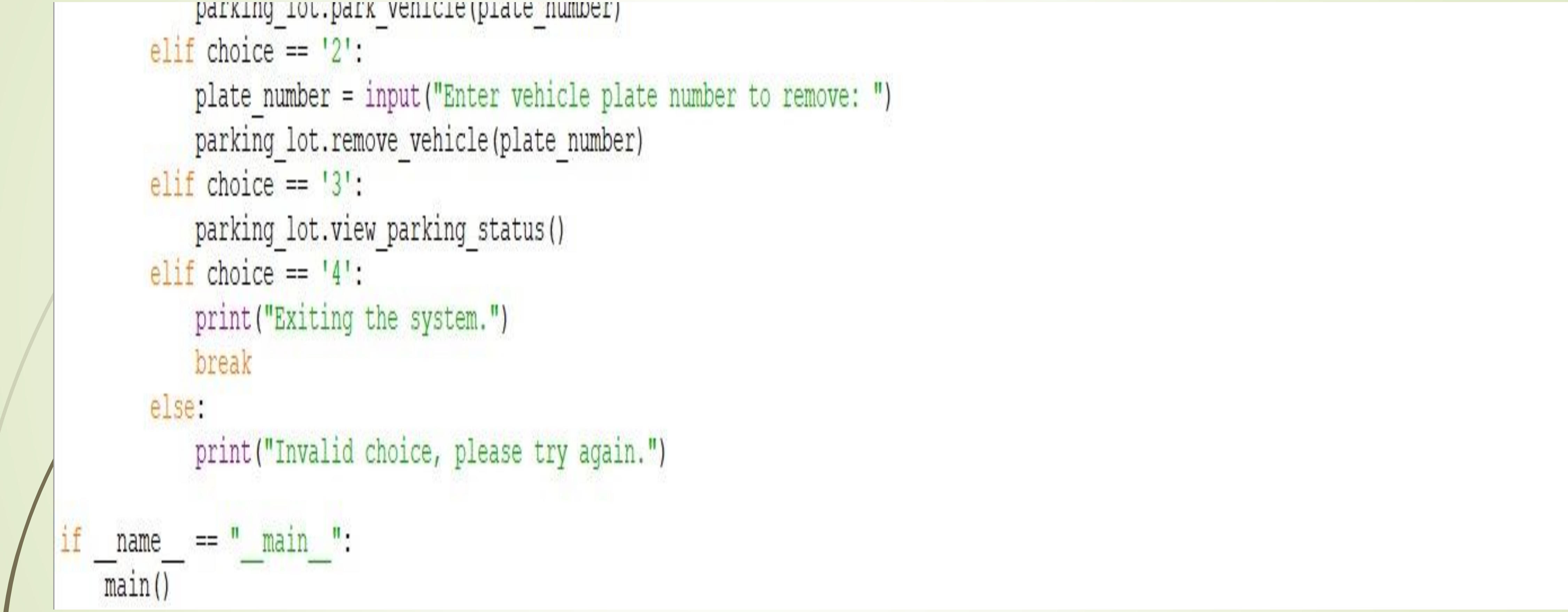
# Components of a parking management system

- **1. Reservation system**
- *** Online Booking :** Allow customers to reserve parking spots in advance via a website or app.
- *** Real-Time Availability :** Display available spots in real time to avoid overbooking.
- **2. Access Control**
- *** Automated Gates :** use RFID and QR code scanners at entry/exit points to automate access.
- ***Mobile Entry:** Enable customers to enter using mobile tickets or codes .
- **3. Safety and Security**
- *** Surveillance Cameras :** Install cameras for security monitoring .
- ***Lighting and Signage :** Ensure the parking area is well-lit and has clear signage for navigation .

- **4. Payment Processing**
- *** Integrated Payment Systems:** Support various payment methods (credit/debit cards, mobile payments).
- *** In-App Payment:** Allow payments through the theatres app for seamless interactions .
- **5. Customer Support**
- *** Help Desk Integration :** Provide a hotline or chat feature for assistance .
- *** FAQs and Guides :** Offer resources for common inquires about parking and reservations .
- **6. Integration With Theatre Operations**
- *** Show Timings:** Sync parking availability with show schedules to optimize space.
- *** Dynamic Pricing :** Adjust rates based on demand and peak times .

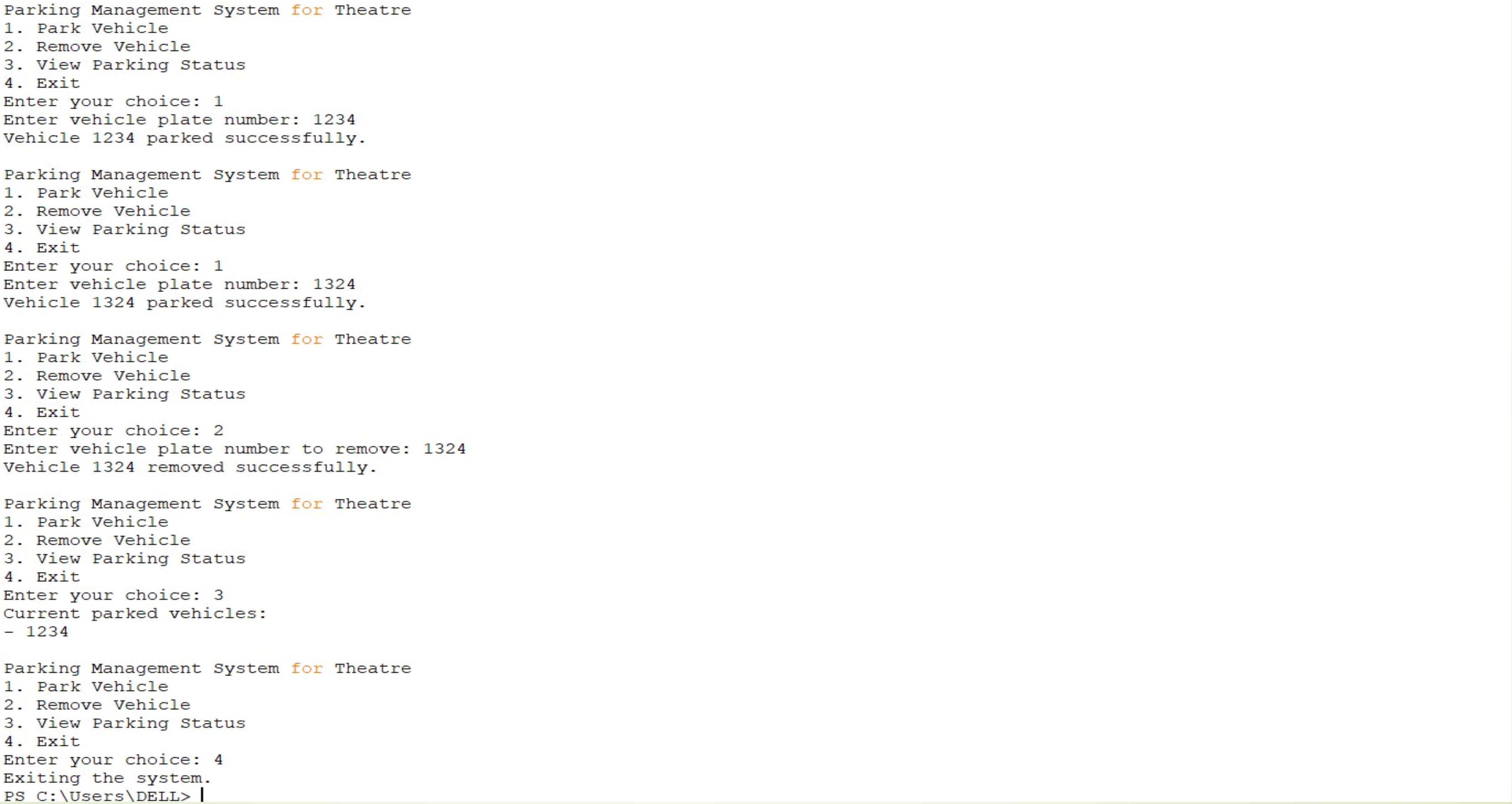


```python
class ParkingLot:
    def __init__(self, capacity):
        self.capacity = capacity
        self.vehicles = {}

    def park_vehicle(self, plate_number):
        if len(self.vehicles) < self.capacity:
            if plate_number not in self.vehicles:
                self.vehicles[plate_number] = True
                print(f"Vehicle {plate_number} parked successfully.")
            else:
                print(f"Vehicle {plate_number} is already parked.")
        else:
            print("Parking lot is full!")

    def remove_vehicle(self, plate_number):
        if plate_number in self.vehicles:
            del self.vehicles[plate_number]
            print(f"Vehicle {plate_number} removed successfully.")
        else:
            print(f"Vehicle {plate_number} not found in the parking lot.")

    def view_parking_status(self):
        if self.vehicles:
            print("Current parked vehicles:")
            for plate in self.vehicles.keys():
                print(f"- {plate}")
        else:
            print("No vehicles are currently parked.")

def main():
    parking_lot = ParkingLot(capacity=5)

    while True:
        print("\nParking Management System for Theatre")
        print("1. Park Vehicle")
        print("2. Remove Vehicle")
        print("3. View Parking Status")
        print("4. Exit")

        choice = input("Enter your choice: ")

        if choice == '1':
            plate_number = input("Enter vehicle plate number: ")
            parking_lot.park_vehicle(plate_number)
        elif choice == '2':
```



```
            parking_lot.park_vehicle(plate_number)
        elif choice == '2':
            plate_number = input("Enter vehicle plate number to remove: ")
            parking_lot.remove_vehicle(plate_number)
        elif choice == '3':
            parking_lot.view_parking_status()
        elif choice == '4':
            print("Exiting the system.")
            break
        else:
            print("Invalid choice, please try again.")

if __name__ == "__main__":
    main()
```

# Output:



```
Parking Management System for Theatre
1. Park Vehicle
2. Remove Vehicle
3. View Parking Status
4. Exit
Enter your choice: 1
Enter vehicle plate number: 1234
Vehicle 1234 parked successfully.

Parking Management System for Theatre
1. Park Vehicle
2. Remove Vehicle
3. View Parking Status
4. Exit
Enter your choice: 1
Enter vehicle plate number: 1324
Vehicle 1324 parked successfully.

Parking Management System for Theatre
1. Park Vehicle
2. Remove Vehicle
3. View Parking Status
4. Exit
Enter your choice: 2
Enter vehicle plate number to remove: 1324
Vehicle 1324 removed successfully.

Parking Management System for Theatre
1. Park Vehicle
2. Remove Vehicle
3. View Parking Status
4. Exit
Enter your choice: 3
Current parked vehicles:
- 1234

Parking Management System for Theatre
1. Park Vehicle
2. Remove Vehicle
3. View Parking Status
4. Exit
Enter your choice: 4
Exiting the system.
PS C:\Users\DELL>
```

- **Benefits**

***Increased Efficiency :** Reduces time spent searching for parking.

***Enhanced Customer Experience :** streamlined processes lead to happier patrons.

***Revenue Optimization :** Better management can lead to increased parking revenue.

***Data Insights :** Collecting data helps in making informed decisions for future improvements.

**Conclusion**

Implementing a parking management system tailored for theatres can significantly improve the overall experience for patrons while optimizing space and resources for the theatre.