In [1]:

```
#bd.py
import sqlite3

DB_NAME = "parking.db"

def get_connection():
    return sqlite3.connect(DB_NAME)

def create_tables():
    conn = get_connection()
    cursor = conn.cursor()

    cursor.execute("""
    CREATE TABLE IF NOT EXISTS parking_slots (
        slot_number INTEGER PRIMARY KEY,
        is_available TEXT
    )
    """)

    cursor.execute("""
    CREATE TABLE IF NOT EXISTS parking_records (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        vehicle_number TEXT,
        slot_number INTEGER,
        entry_time TEXT,
        exit_time TEXT,
        status TEXT
    )
    """)

    conn.commit()
    conn.close()
```

In [2]:

```
#parking_slot.py
class ParkingSlot:
    def __init__(self, slot_number, is_available="YES"):
        self.slot_number = slot_number
        self.is_available = is_available
```

In [3]:

```
#parking_record.py
class ParkingRecord:
    def __init__(self, vehicle_number, slot_number, entry_time, status):
        self.vehicle_number = vehicle_number
        self.slot_number = slot_number
        self.entry_time = entry_time
        self.status = status
```

In [ ]:

```
#parking_dervice.py
from database.db import get_connection, create_tables
from datetime import datetime

TOTAL_SLOTS = 10
```

```python
def initialize_slots():
    create_tables()
    conn = get_connection()
    cursor = conn.cursor()

    cursor.execute("SELECT COUNT(*) FROM parking_slots")
    count = cursor.fetchone()[0]

    if count == 0:
        for i in range(1, TOTAL_SLOTS + 1):
            cursor.execute(
                "INSERT INTO parking_slots VALUES (?, ?)",
                (i, "YES")
            )
        conn.commit()
    conn.close()

def park_vehicle(vehicle_number):
    conn = get_connection()
    cursor = conn.cursor()

    cursor.execute(
        "SELECT slot_number FROM parking_slots WHERE is_available='YES' LIMIT
    )
    slot = cursor.fetchone()

    if not slot:
        print("❌ Parking Full")
        conn.close()
        return

    slot_number = slot[0]

    cursor.execute(
        "UPDATE parking_slots SET is_available='NO' WHERE slot_number=?",
        (slot_number,)
    )

    cursor.execute(
        "INSERT INTO parking_records VALUES (NULL, ?, ?, ?, ?, ?)",
        (
            vehicle_number,
            slot_number,
            datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
            None,
            "PARKED"
        )
    )

    conn.commit()
    conn.close()

    print(f"✅ Vehicle parked at slot {slot_number}")
```

```python
def exit_vehicle(vehicle_number):
    conn = get_connection()
    cursor = conn.cursor()

    cursor.execute(
        """
        SELECT slot_number FROM parking_records
        WHERE vehicle_number=? AND status='PARKED'
        """,
        (vehicle_number,)
    )
    record = cursor.fetchone()

    if not record:
        print("❌ Vehicle not found")
        conn.close()
        return

    slot_number = record[0]

    cursor.execute(
        """
        UPDATE parking_records
        SET exit_time=?, status='EXITED'
        WHERE vehicle_number=? AND status='PARKED'
        """,
        (datetime.now().strftime("%Y-%m-%d %H:%M:%S"), vehicle_number)
    )

    cursor.execute(
        "UPDATE parking_slots SET is_available='YES' WHERE slot_number=?",
        (slot_number,)
    )

    conn.commit()
    conn.close()

    print(f"✅ Vehicle exited from slot {slot_number}")

def view_status():
    conn = get_connection()
    cursor = conn.cursor()

    cursor.execute("SELECT * FROM parking_slots")
    slots = cursor.fetchall()

    print("\nParking Slot Status:")
    for slot in slots:
        print(f"Slot {slot[0]} → Available: {slot[1]}")

    conn.close()
```

In [5]:

```python
#menu.py
def show_menu():
    print("\n--- Vehicle Parking Management System ---")
    print("1. Park Vehicle")
    print("2. Exit Vehicle")
    print("3. View Parking Status")
    print("4. Exit")
```

In [ ]:

```python
#main.py
from services.parking_service import (
    park_vehicle,
    exit_vehicle,
    view_status,
    initialize_slots
)
from utils.menu import show_menu

def main():
    initialize_slots()

    while True:
        show_menu()
        choice = input("Enter your choice: ")

        if choice == "1":
            vehicle_no = input("Enter vehicle number: ")
            park_vehicle(vehicle_no)

        elif choice == "2":
            vehicle_no = input("Enter vehicle number: ")
            exit_vehicle(vehicle_no)

        elif choice == "3":
            view_status()

        elif choice == "4":
            print("Exiting system...")
            break

        else:
            print("Invalid choice!")

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

--- Vehicle Parking Management System ---

1. Park Vehicle
2. Exit Vehicle
3. View Parking Status

```
    4.  Exit Enter your choice: 1 Enter vehicle number: TS05TG1771 ✅ Vehicle
        parked at slot 5

--- Vehicle Parking Management System ---

    1.  Park Vehicle
    2.  Exit Vehicle
    3.  View Parking Status
    4.  Exit Enter your choice: Invalid choice!

--- Vehicle Parking Management System ---

    1.  Park Vehicle
    2.  Exit Vehicle
    3.  View Parking Status
    4.  Exit Enter your choice: 2 Enter vehicle number: TS05TT9875 ❌ Vehicle
        not found

--- Vehicle Parking Management System ---

    1.  Park Vehicle
    2.  Exit Vehicle
    3.  View Parking Status
    4.  Exit Enter your choice:
```