**SRS DOCUMENTATION FOR**

**VEHICLE PARKING SYSTEM**

**Introduction:**  
The **Vehicle Parking System** is designed to manage parking spaces efficiently by tracking vehicle information, managing the availability of parking spots, and calculating parking fees based on duration and vehicle type. It offers core functionalities including parking space assignment, fee calculation, and occupancy status updates with real-time data handling. The system utilizes collections for managing parking spots, inheritance for different vehicle types, and file handling for saving parking records.

**Objective:**

A Implement a vehicle parking system that manages the availability of parking spots, tracks vehicle details, and calculates parking fees. Use collections for managing parking spots, inheritance for different vehicle types, and file handling for saving parking records.

**Platform Specification:**

**HARDWARE:**

INTEL CORE 13,15,17 OR MAC,MINIMUM 4GB RAM

**SOFTWARE:**

JAVA,SQL,JDBC,JIRA

**Functional requirements:**

1. **Vehicle Registration and Parking Management**:
   * Allows new vehicles to be registered and assigns parking spots if available. Tracks occupancy status of each parking spot.
2. **Parking Fee Calculation**:
   * Calculates fees based on vehicle type, spot type, and parking duration.
3. **Parking Spot Management**:
   * Manages parking spots' availability and status in real-time.
4. **Admin Controls**:
   * Allows admins to view all parked vehicles, manage parking fees, and generate parking records for review and analysis.

**Schema diagram:**

**1.ParkingSlot:** Stores details of each parking slot in the system.

Columns:

slot\_id: INT (PK, AUTO\_INCREMENT) - Unique identifier for each parking slot.

slot\_number: VARCHAR(20) - Unique slot number assigned to each parking space.

floor: INT - Floor number where the parking slot is located.

status: ENUM('available', 'occupied') - Current status of the slot.

slot\_type: VARCHAR(20) - Type of slot (e.g., "Compact", "Large", "Handicapped").

**2. Vehicle:** Stores details of each vehicle entering the parking system.

Columns:

vehicle\_id: INT (PK, AUTO\_INCREMENT) - Unique identifier for each vehicle record.

license\_plate: VARCHAR(15) - License plate number of the vehicle.

vehicle\_type: VARCHAR(20) - Type of vehicle (e.g., "Car", "Bike", "Truck").

entry\_time: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP - Timestamp of vehicle entry.

slot\_id: INT (FK) - Foreign key referencing ParkingSlot(slot\_id).

Foreign Key:

slot\_id references ParkingSlot(slot\_id).

**3. Exit:** Stores details of each vehicle exit from the parking system.

Columns:

exit\_id: INT (PK, AUTO\_INCREMENT) - Unique identifier for each exit record.

vehicle\_id: INT (FK) - Foreign key referencing Vehicle(vehicle\_id).

exit\_time: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP - Timestamp of vehicle exit.

total\_fee: DECIMAL(10, 2) - Total parking fee calculated at exit.

Foreign Key:

vehicle\_id references Vehicle(vehicle\_id).

**4. Payment:** Stores payment transactions for parking fees.

Columns:

payment\_id: INT (PK, AUTO\_INCREMENT) - Unique identifier for each payment transaction.

exit\_id: INT (FK) - Foreign key referencing Exit(exit\_id).

payment\_amount: DECIMAL(10, 2) - Amount paid by the customer.

payment\_time: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP - Timestamp when payment was made.

payment\_method: VARCHAR(50) - Method of payment (e.g., "Cash", "Card", "UPI").

Foreign Key:

exit\_id references Exit(exit\_id).

**5. Rate:** Stores parking rates based on vehicle type and duration.

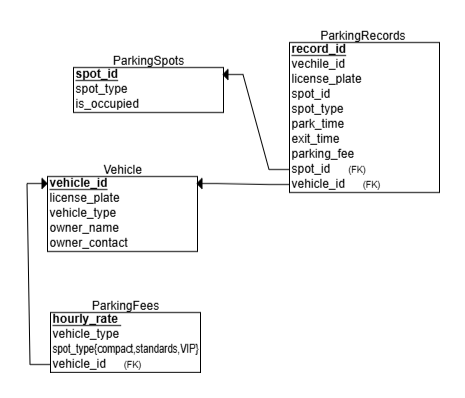
Columns:

rate\_id: INT (PK, AUTO\_INCREMENT) - Unique identifier for each rate record.

vehicle\_type: VARCHAR(20) - Type of vehicle (e.g., "Car", "Bike").

hourly\_rate: DECIMAL(10, 2) - Rate charged per hour.

daily\_rate: DECIMAL(10, 2) - Rate charged per day (if applicable).

SCHEMA:

**ER Diagram:**

