Python and Django Problem Statement

Parking Lot Management

Design an application to assist the parking assistance team (PAT) to help them accommodate customers with following facilities:

1. Parking is in multiple levels.
2. Each level has the same number of parking spaces.
3. Parking slots are categorized into two-wheeler and four-wheeler slots
4. The PAT facilitates customers to the available parking lots.

Application features

1. PAT can view the number of available parking slots in each floor.
2. PAT has access to information about all the parking spaces, regardless of their availability.
3. They can assign a random parking lot number from the available slots.
4. A parking fee can be generated for parking while unlocking the lot.
5. Customer has access to only availability of parking spaces in any level. (Only number of available spaces, not the lot numbers)
6. Customers can pre-book the parking lot for a timeslot. A random lot from available is allotted based on the category selected.
7. Customers can cancel their booking before the timeslot.
8. If the customer checks out of the lot beyond the timeslot, an additional fee is applied for late checkout.

Requirements

API

1. Number of available parking spaces for each category (2-wheeler/4-wheeler) in each level (GET)
    - Admins can get the number of available lots for each category
    - Public users can only check if available or not for category

    Input: None

    Output: Number of slots/Availability
2. Assign/Lock a parking space to a vehicle (POST)
    - Assign a random parking lot based on the input and update the availability in parking space table
    - Parking lot should be verified in the Parking table before assigning. Both In and Out times must be available for all entries of the lot number
    - Create a new entry in the ParkingHistory table whenever this API call is invoked and update the TWA/FWA field in ParkingSpace for the parking level in input

    Input: Vehicle category, Vehicle number and Parking level

    Output: Vehicle category, Vehicle number, Parking level, Parking lot number, Locking time, User ID
3. Unlock parking space (POST)

    Input: Vehicle number, Lot

    Output: Vehicle number, Parking lot number, locking time, unlocking time, Parking fees, User ID

Tables

1. ParkingSpace
    - ID
    - Level
    - TWA (Number of two-wheeler available slots)
    - FWA (Number of four-wheeler available slots)

2. User
    - ID
    - Name
    - Password (encrypted)
    - Role (ADMIN/PUBLIC)
3. ParkingHistory
    - ID
    - Level
    - Type (TW/FW)
    - VehicleNumber
    - Lot
    - In
    - Out
    - Fee