Design Parking Lot 

Overview

There are two scenarios where you know the system or you dont know.

If you know about the system

1. You briefly tell what you know about the system
2. Ask if you are thinking of system is correct
3. Follow up questions
   a. Entity vs complete software system (If only entity design is required then no need to ask 2nd question
   b. If complete S/W system, then persistence or not (where should be put the data in memory or db)
   c. How the input will be given (command line or rest-API)

If you dont know about the system

1. Ask for the brief overview about the system so that you can come up with the necessary requirements
2. Follow up questions
   a. Entity vs complete software system (If only entity design is required then no need to ask 2nd question
   b. If complete S/W system, then persistence or not (where should be put the data in memory or db)
   c. How the input will be given (command line or rest-API)

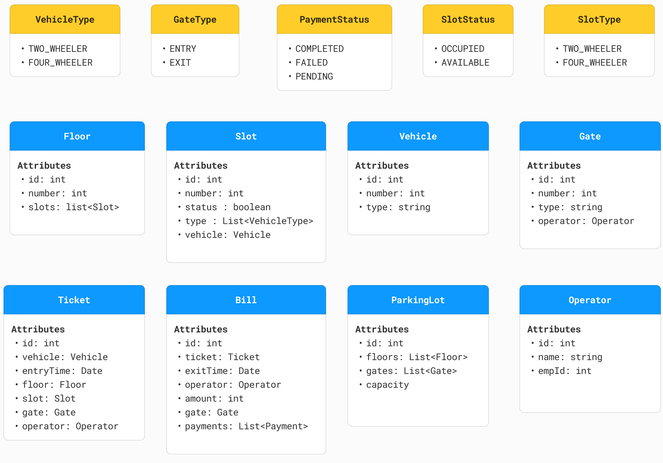
Requirement Gathering

1. Suggest ideas/ features
2. Visualization
   a. User Journey
   b. Physical structure

Requirements

1. Parking lot with multiple floors
2. Different types of vehicles can be there (2-wheelers or 4-wheelers)
3. Each spot can be dedicated to a set of vehicles
4. There can be multiple entry and exit gates
5. System should allow to easily change how the fees is calculated
6. At the time of entry a spot is assigned to the vehicle
7. We have to allow our system to be flexible with how they assign the spot(can be nearest to gate/randomly allocated)
8. Spot is released at the time of exit
9. System should support both online and offline payments
10. No pass system

Class Diagram



Notes for Schema Design

1. For every class that is representing entity in your class diagram, create a table for each of those
2. For primitive attributes in those classes(int, String, data etc) put them as it is as a column in the corresponding table.
3. For non-primitive attributes
   a. find the cardinality of relation
   b. Depending pn cardinality represent it

For 1:1 relation, put id of one side to another side
For 1:M relation, put id of one side to M side
For M:M relation, create new mapping table

Recommended, for every enum create a table with two column id and value
parking_spot_status
id value
1 AVAILABLE
2 BOOKED

Schema Design