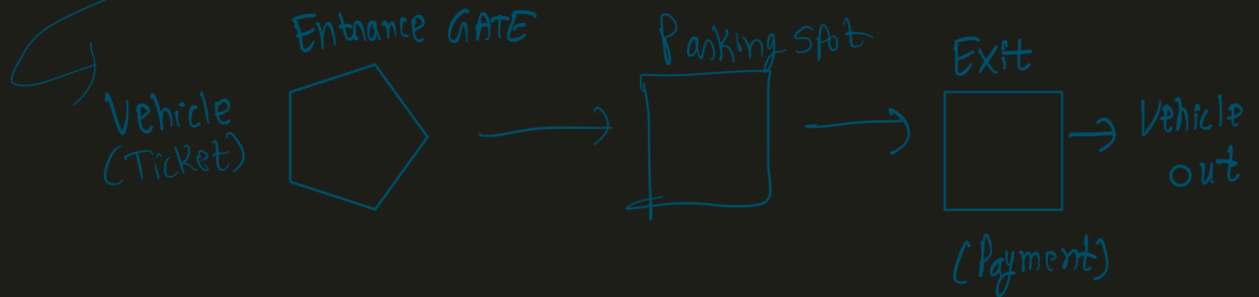


How to approach:

- ① make rough flow in mind & requirement clarification & objects clarity to interviewer



### Requirement Clarification

- ① How Many Entrance & exit  
(our code should be scalable to add more, entry & exits)  
→ 1 entrance  
→ 1 exit

- ② Different type of Parking Spot?  
\* Two wheelers  
\* four wheelers  
\* Three wheelers  
*currently these 2 only*

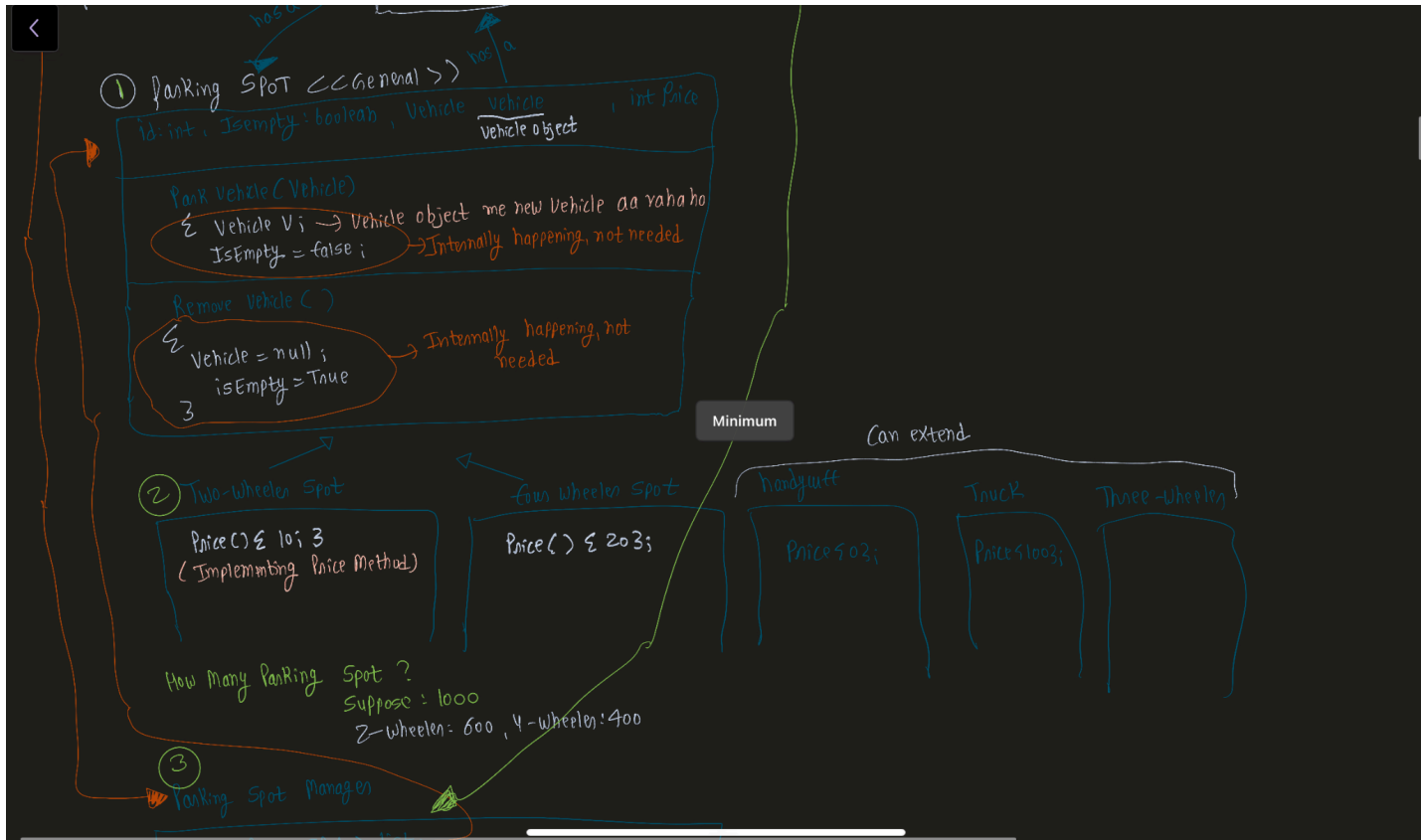
- ③ what are Payment strategy?  
\* Hourly  
\* Minutes  
\* Mixed

- ④ Parking Spot should be nearest to entrance
- ⑤ floors Parking?  
No

### Objects

- ① \* Vehicle *we are interested in* → Vehicle No. → enum { 2 wheelers, 4 wheelers }  
→ Vehicle Type  
\* Ticket → entry time, Parking Spot (2 or 4 wheelers)  
\* entrance GATE → find parking space, update parking space, generate ticket  
\* Parking Spot → id, is Empty, Vehicle, Price, Type, Nearest Parking Space  
\* exit GATE → cost calculate, payment, update parking spot

Minimum



&lt;

3

Parking Spot Managen

list &lt; ParkingSpot &gt; list;

PS Managen (list &lt; PS &gt; list)

Σ list = list *Carri list ko, Jo list aa vahi usse initialise kar dega*

\* it will help, In finding Parking Space, add Parking Space();  
 Remove Parking Space(); *Car use Minheap*  
 Park Vehicle();  
 Remove Vehicle();

4

Two-wheeler Managen

List &lt; PS &gt; list = £ 600 2-wheeler 3

Two wheeler Managen ()

Σ Super (list)

3 *Two wheeler Managen calls Super on Parent to get list*

is a

four-wheeler Managen

Minimum

List &lt; PS &gt; list = £ 400 4-wheeler PS 3

four wheeler Managen ()

Σ Super (list)

3

\* Parking Strategy for finding nearest Parking

find Parking Space(); *→ default**(Jo free mile waha)**has a*

<

④

Two-wheeler Manager

is-a

four-wheeler Manager

List<PS> list = £ 600 2-Wheeler 3

TwoWheelerManager()

£ super(list)

3 ↳ TwoWheelerManager  
Calls super on Parent  
to get list

List<PS> list = £ 400

4-Wheeler

PS 3

fourWheelerManager()

£ super(list)

3

\* Parking strategy for finding nearest Parking

findParkingSpace(); ↳ default

(Jo free mile waha)

Minimum

has a

Parking Strategy

Near to entrance  
find()

Near to entrance & elevator  
findSpace()

Default