Case Study-1

Parking Lot

System Requirements (Excepted and can be added more)

1. The parking lot should have multiple floors where customers can park their vehicles.

2. The parking lot should have multiple entry and exit points.

3. Customers can collect a parking ticket from the entry points and can pay the parking fee at the exit points on their way out.

4. Customers can pay the tickets at the automated exit panel or to the parking attendant.

5. Customers can pay via both cash and credit cards.

6. Customers should also be able to pay the parking fee at the customer's info portal on each floor. If the customer has paid at the info portal, they don't have to pay at the exit.

7. The system should not allow more vehicles than the maximum capacity of the parking lot. If the parking is full, the system should be able to show a message at the entrance panel and on the parking display board on the ground floor.

8. Each parking floor will have many parking spots. The system should support multiple types of parking spots such as Compact, Large, Handicapped, Motorcycle, etc.

9. The Parking lot should have some parking spots specified for electric cars. These spots should have an electric panel through which customers can pay and charge their vehicles.

10. The system should support parking for different types of vehicles like car, truck, van, motorcycle, etc.

11. Each parking floor should have a display board showing any free parking spot for each spot type.

12. The system should support a per-hour parking fee model. For example, customers have to pay Rs 20 for the first hour, Rs 10 for the second and third hours, and Rs 5 for all the remaining hours.

MINUTES OF 8/10 :

CLASSES:

Admin, Customer, Floor, Spot, Customer Panel..

Interfaces : Spot,Payment,Vehicle

WORK:

Varun and Rajendra.
Floor,Admin,Customer: Manjunath
Vehicle: Sahithi
Electric and non-Electric spot: Aditya

Spot interface:
Methods:
Assign: Assigns the Spot if empty
isVacant-To check if a spot is vacant(returns true if vacant)
getSpotId-returns the spotId

Spot classes :ElectricSpot, CompactSpot, LargeSpot, HandicappedSpot, MotorcycleSpot
Espot implements Spot,Payment:
Variables:int spotId, int capacity;
Methods: charge-prints “charging your vehicle”
ok leave it i ll implement someone implement the payment thing fast!!!!!!

Requirements: should show different options(print them simply) take input for the choice
Add an extra option (i.e cash) when payment is done at exit.

Class Payment{
public void getFee(){
if(time <= 60){
fee = 20; }
else if(time > 60 && time <= 120){
fee = 30;}

```
            else if(time > 120 && time <= 180){
                fee = 40;}
        else{
                fee = 40 +Math.ceil(( time - 180 )/60)*5;}

                System.out. println("You have to pay "+fee+" rupees");
        }
public void payViaCash(){
                System.out. println("Payment is done via cash");
        }
public void payViaCreditCard(){
                System.out. println("Payment is done via credit card");
        }
                DONE
```

```
Interface Vehicle{
                public void parkInSpot();
                        }

Class Car implements Vehicle{

        public void parkInSpot(CompactSpot s){
                CompactSpot.park(s);
                }
        }

Class Truck implements Vehicle{
        public void parkInSpot(LargeSpot s){
                CompactSpot.park(s);
                }
                        }

Class Van implements Vehicle{
        public void parkInSpot(LargeSpot s){
                CompactSpot.park(s);
                }
                        }

Class MotorCycle implements Vehicle{
        public void parkInSpot(CompactSpot s){
                CompactSpot.park(s);
                }
                        }
```

Floor Class:(done!!!) some work related to extension is pending.

Variables:

Array of Nspots,Espots;

Static maxcapacity,curcapacity;

Methods:

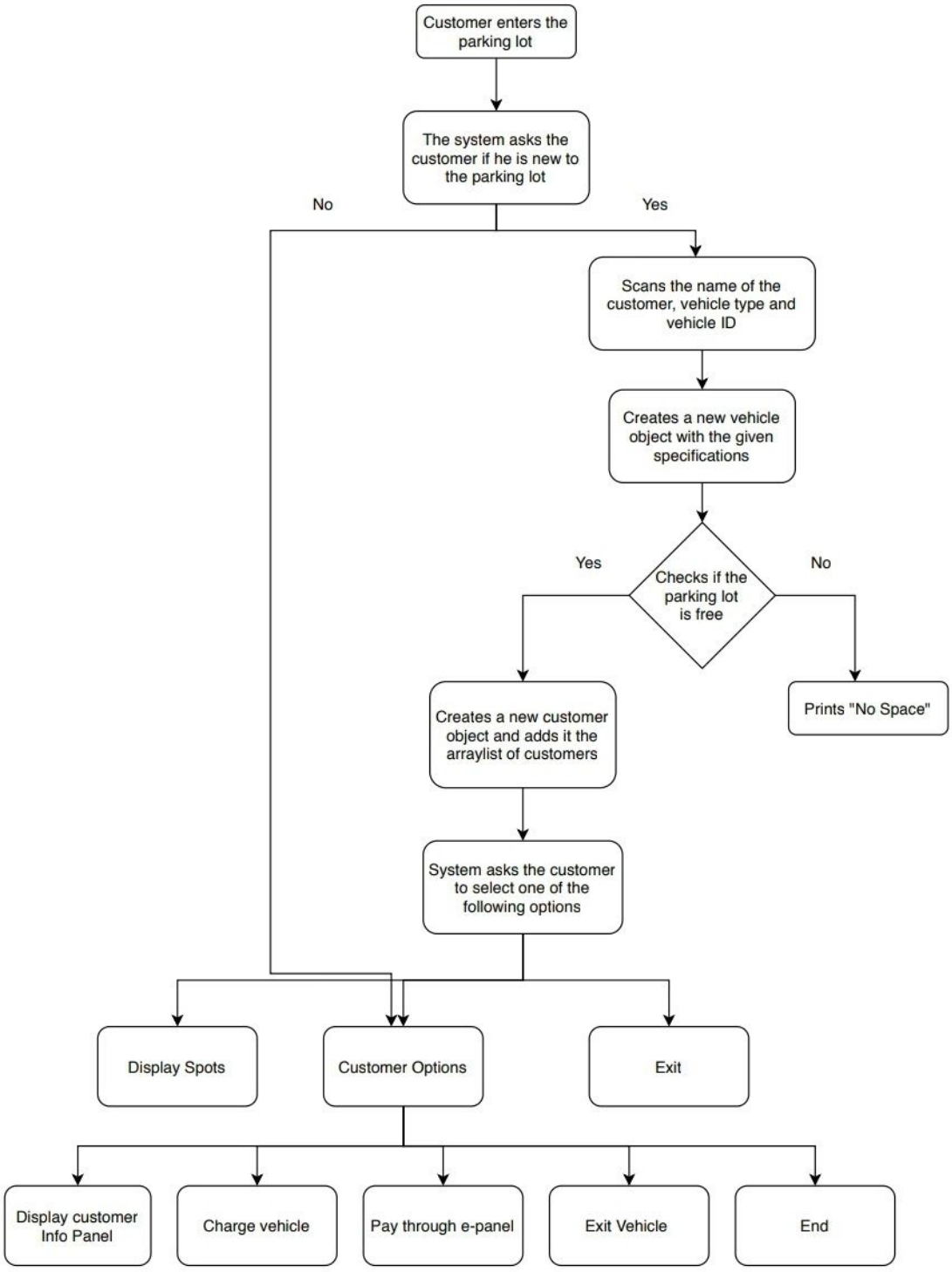
getElectricalSpot -> returns the position of an unoccupied Espot

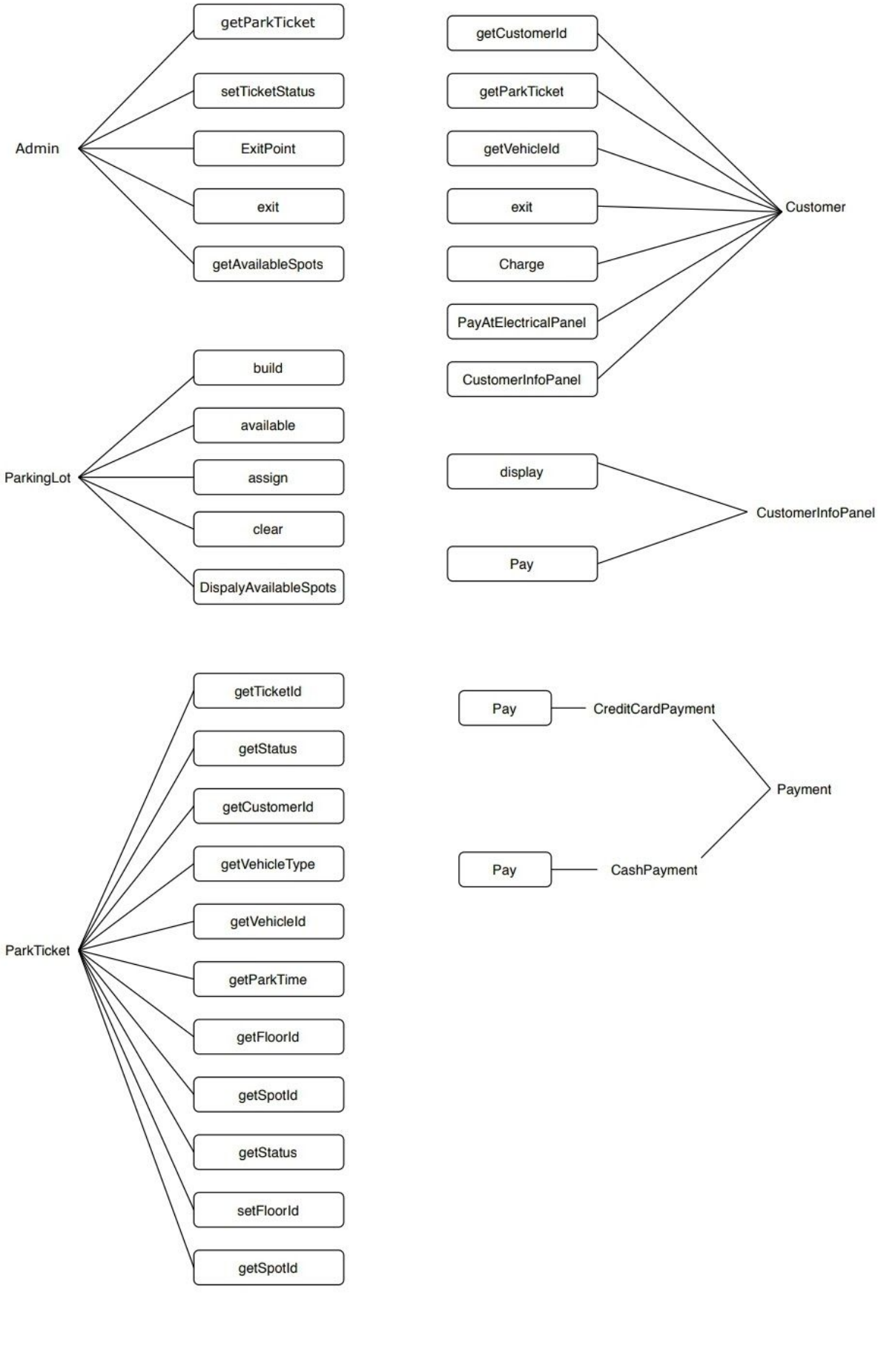
getNonElectricalSpot -> returns the position of an unoccupied Nspot

AddNspot -> adds another Nspot if possible

AddEspot -> adds another Espot if possible

PROGRAM FLOW




Admin
getParkTicket
setTicketStatus
ExitPoint
exit
getAvailableSpots
ParkingLot
build
available
assign
clear
DispalyAvailableSpots
ParkTicket
getTicketId
getStatus
getCustomerId
getVehicleType
getVehicleId
getParkTime
getFloorId
getSpotId
getStatus
setFloorId
getSpotId
getCustomerId
getParkTicket
getVehicleId
exit
Charge
PayAtElectricalPanel
CustomerInfoPanel
Customer
display
Pay
CustomerInfoPanel
Pay
CreditCardPayment
Pay
CashPayment
Payment