OOM Project Report

Topic: Parking Lot Design

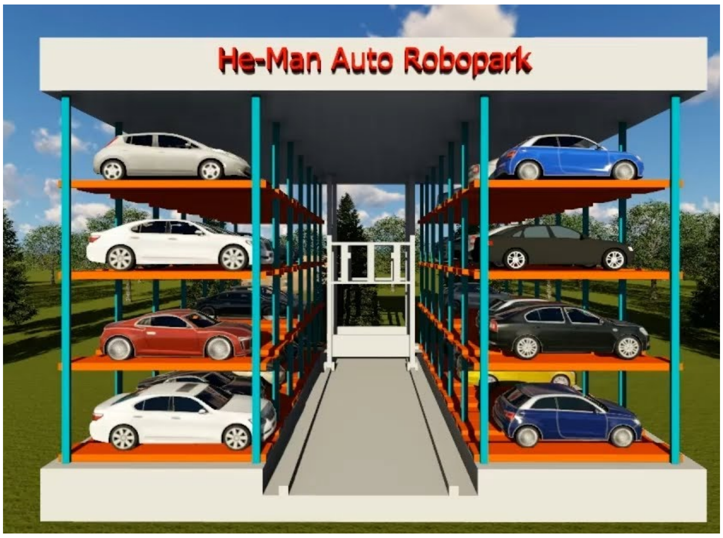
Submitted To: Saurabh Shrivastava

Made By:

- Ashna Agrawal - LCS2020028
- Koyna Mittal - LIT2020028
- Karishma Vanwari - LCS2020029
- Palak Goel - LIT2020030

PARKING LOT DESIGN

A parking lot or car park is a dedicated cleared area that is intended for parking vehicles. In most countries where cars are a major mode of transportation, parking lots are a feature of every city and suburban area. Shopping malls, sports stadiums, megachurches, and similar venues often feature parking lots over large areas.



System Requirements

We will focus on the following set of requirements while designing the parking lot:

1. Each parking floor will have many parking spots. The system should support multiple types of vehicles such as Large(Truck/bus), parking spots for Handicapped, Motorcycle/car, etc. on different floors.
2. Customers can collect the parking ticket from the parking attendant at the entry point and pay the attendant at the exit point.
3. When a vehicle is parked, a slot will be booked for it and when it leaves that slot will be freed for other vehicles.
4. 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.
5. The Parking lot should have some parking spots specified for electric cars. These spots should have an electric panel through which customers can charge their vehicles.
6. There should be separate accounts for admin and parking attendants and the admin should be able to see all the customer information and update the prices.
7. Customers can pay via both cash and credit cards for the parking ticket.

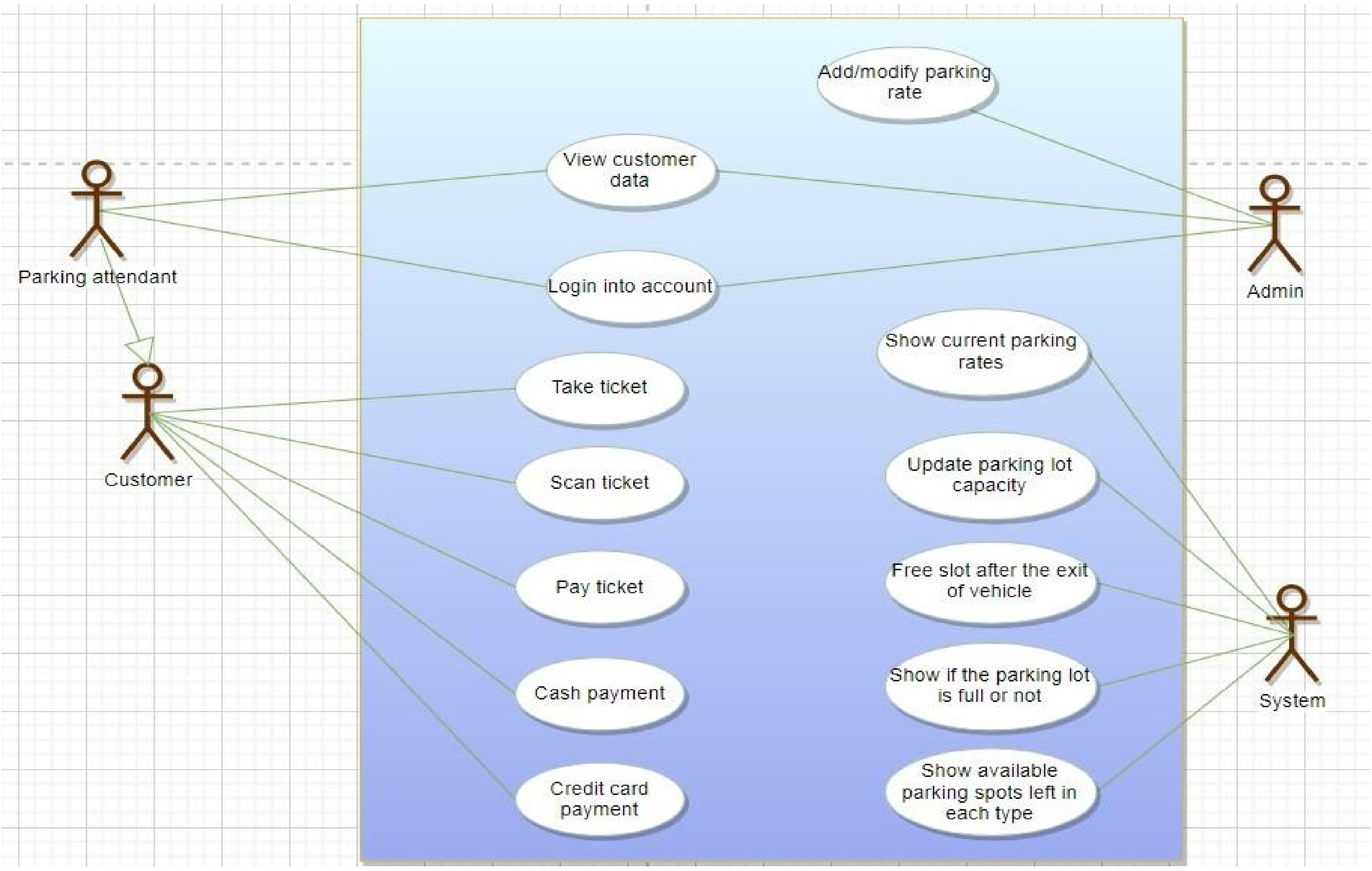
Use Case diagram

Here are the main actors in our system:

1. **Admin**: Mainly responsible for updating the prices if required and can also see the records of cars parked.
2. **Customer**: All customers can get a parking ticket and pay for it.
3. **Parking attendant**: Parking attendants can do all the activities on the customer's behalf and take cash for ticket payment.
4. **System:** To display messages on different info panels and assign and remove a vehicle from a parking spot.

Here are the top use cases for Parking Lot:

1. **Take ticket**: To provide customers with a new parking ticket when entering the parking lot.
2. **Scan ticket:** To scan a ticket to find out the total charge.
3. **Credit card payment:** To pay the ticket fee with a credit card.
4. **Cash payment**: To pay the parking ticket through cash.
5. **Add/Modify parking rate**: To allow admin to add or modify the hourly parking rate.
6. **Book Slot:** To book the slot of a vehicle according to its type.
7. **Free Slot:** To empty the slot after the vehicle is gone.

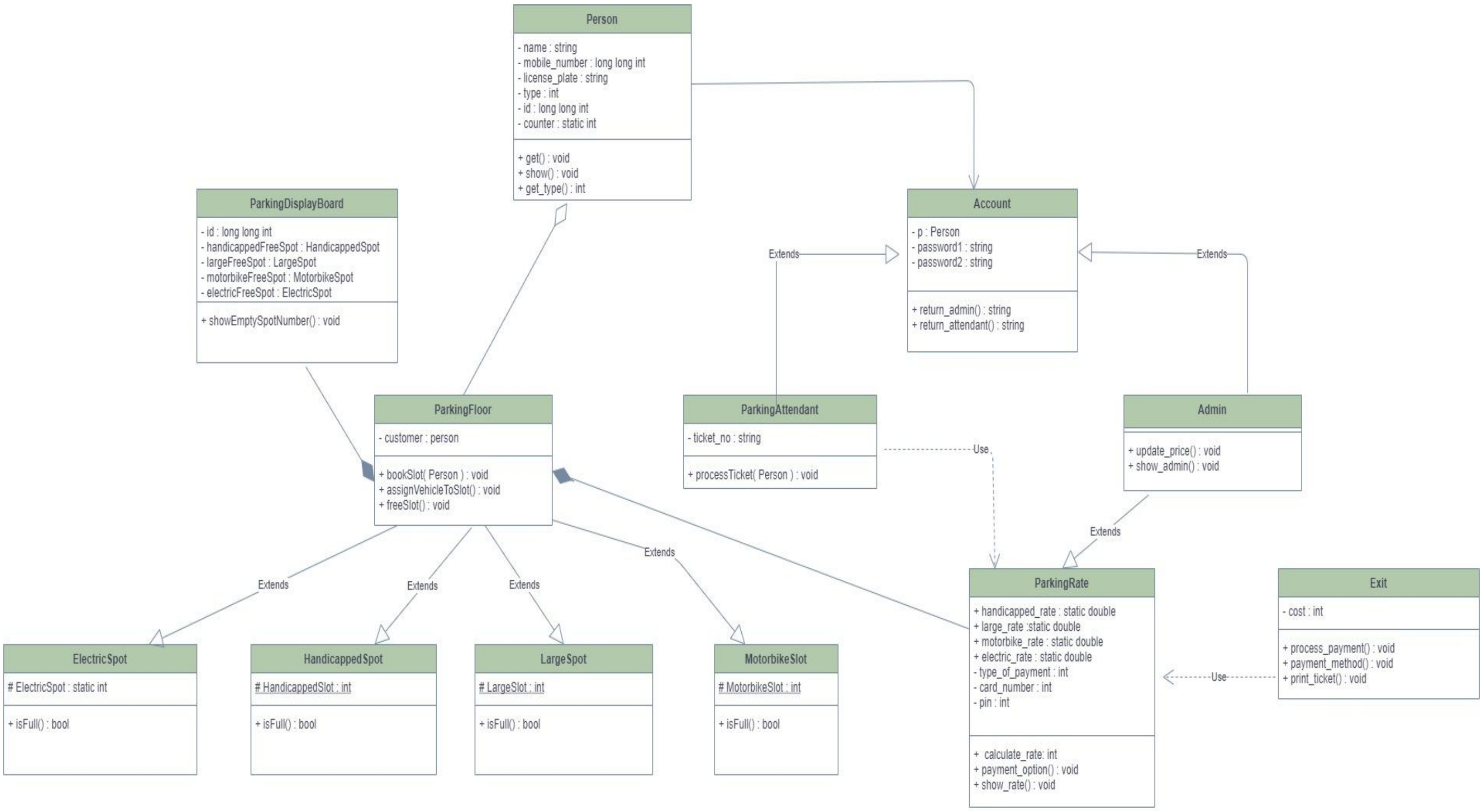

Add/modify parking rate
View customer data
Parking attendant
Login into account
Admin
Show current parking rates
Take ticket
Customer
Scan ticket
Update parking lot capacity
Pay ticket
Free slot after the exit of vehicle
System
Cash payment
Show if the parking lot is full or not
Credit card payment
Show available parking spots left in each type

USE CASE DIAGRAM

Class diagram

Here are the main classes of our Parking Lot System:

1. **Account:** We will have two types of accounts in the system: one for an Admin and a parking attendant.
2. **ParkingAttendant:** This class will encapsulate all the operations that an attendant can perform, like scanning tickets and processing payments.
3. **Admin:** This class will be responsible for updating the prices of different types of vehicles and also to get the information of all customers.
4. **Vehicle:** Vehicles will be parked in the parking spots. Our system will support different types of vehicles 1) Car/motorbike, 2) Truck/bus, 3) Electric.
5. **Person:** This class stores all the necessary information of the customer like id, name, mobile number, and license plate number.
6. **ParkingRate:** This class will help calculate and show the parking rate for different types of vehicles and the rate for electric vehicles and their charging.
7. **ParkingDisplayBoard:** This class shows the available free slots on various floors - handicapped slot, motorbike slot, electric slot, large slot.
8. **ExitPanel**: ExitPanel will facilitate payment of the ticket fee. The system will support credit card and cash transactions.
9. **ParkingFloor**: The parking lot will have many parking floors and it will book a slot, assign a vehicle to the slot, and be responsible to free a slot after the exit of the vehicle.



Conclusion

The Parking Management System is based on object-oriented concepts to achieve various requirements.
The services provided by smart parking have become the essence of building smart cities. This project focused on implementing an integrated solution for smart parking. The proposed system has several advantages, including detection of available free space, collecting and paying for the parking ticket, easy access to admin and parking attendant, etc.

Inheritance comes into the picture Parking Floor class inherits an electric spot, handicapped spot, Large spot, motorbike spot. Also, the parking attendant inherits from the account. Admin, another core entity of the entire system also inherits from 2 classes- parking rate and account. Incorporating the use of encapsulation at each step of the project to organize things by dividing each entity into a class of its own into individual header files. The use of association to link the heart of the program i.e. the user with the accounting system is also evident. Composition is also used to ensure that an entity does not exist without another to achieve proper functioning. Aggregation is also used where an instance of a class is needed along with the individual existence of another like in person and parking floor classes. The use of abstraction and various other concepts of object oriented programming is also evident throughout the project.