# Use Cases:

Customer:

1. Take Ticket
2. Park Car
3. Scan and Pay Ticket

Parking Attendant:

1. Login/Logout
2. Take Payment
3. View System
4. Update System

System:

1. Assign a parking spot to a vehicle
2. Remove a vehicle from a parking spot
3. Show Parking Full Message
4. Show Available Parking Spot Message

Admin/Misc

1. Add Parking Level/Floor
2. Add Parking Spot
3. Add Parking Attendant
4. Add Modify Parking Rate
5. Add Entrance/Exit Panels
6. Parking Display Board

# Design Pattern:

ParkingLot: Our system will have only one object of this class. I am going to use the Singleton pattern.

# Class/ Entity List:

## Enumeration

### ParkingSpotType:

1. Handicapped
2. Compact
3. Large
4. Motorbike
5. Electric

### VehicleType:

1. Car
2. Truck
3. Electric
4. Van
5. Motorbike

### ParkingTicketStatus:

1. Active
2. Paid
3. Lost

### AccountStatus:

1. Active
2. Closed
3. Canceled
4. Blacklisted
5. None

## Classes

### Location:

1. StreetAddress
2. City
3. State
4. ZipCode
5. Country

### Person:

1. Name
2. Address
3. Email
4. Phone

### ParkingRate:

1. Duration: int
2. Rate: Double

### ParkingLot:

1. Fields:
   1. ID
   2. Address
2. Methods:
   1. addParkingFloor()
   2. addEntrancePanel()
   3. getNewParkingTicket()
   4. isFull()

## ParkingFloor:

1. Fields
   1. Name: string
2. Methods
   1. updateDisplayBoard()
   2. addParkingSlot()
   3. assignVehicleToSlot()
   4. freeSlot()

### ParkingSpot:

1. Fields
   1. ID: Number
   2. Free: Bool
   3. Type:ParkingSlotType
2. Methods
   1. updateDisplayBoard()
   2. addParkingSlot()
   3. assignVehicleToSlot()
   4. freeSlot()

### Vehicle:

1. Fields
   1. LicenseNumber
   2. Type: VehicleType
2. Methods
   1. AssignTicket()