

# Implementation

## Importing libraries

```
from datetime import datetime
import pickle
import os
```

## Classes

```
class Employee:
    def __init__(self, name, salary, department, passport, age, email, profit=0):
        """
        Initialize an instance of the Manager class.

        :param name: The name of the manager
        :param salary: The salary of the manager
        :param department: The department the manager belongs to
        :param passport: The passport number of the manager
        :param age: The age of the manager
        :param email: The email address of the manager
        :param managerId: The ID of the manager
        :param isDeleted: Whether the manager has been deleted or not (default False)
        """
        self.name = name
        self.salary = salary
        self.department = department
        self.passport = passport
        self.age = age
        self.profit = profit
        self.email = email

class Manager(Employee):

    def __init__(self, name, salary, department, passport, age, email, managerId, isDeleted=False):
        """
        Initialize an instance of the SalesMan class.

        :param name: The name of the salesperson
```

```
:param salary: The salary of the salesperson
:param department: The department the salesperson belongs to
:param passport: The passport number of the salesperson
:param age: The age of the salesperson
:param email: The email address of the salesperson
:param empld: The ID of the salesperson
:param managerId: The ID of the manager the salesperson reports to
:param isDeleted: Whether the salesperson has been deleted or not (default False)
"""
```

```
super().__init__(name, salary, department, passport, age, email)
self.managerId = managerId
self.isDeleted = isDeleted
```

```
class SalesMan(Employee):
```

```
def __init__(self, name, salary, department, passport, age, email, empld, managerId, isDeleted=False):
    """
```

Initialize an Employee object with the given name, salary, department, passport, age, email, employee ID, manager ID, and deleted status.

Parameters:

name (str): The name of the employee.

salary (float): The salary of the employee.

department (str): The department the employee works in.

passport (str): The passport number of the employee.

age (int): The age of the employee.

email (str): The email address of the employee.

empld (int): The ID of the employee.

managerId (int): The ID of the employee's manager.

isDeleted (bool, optional): Whether the employee has been deleted from the system. Defaults to False.

```
"""
```

```
super().__init__(name, salary, department, passport, age, email)
self.empld = empld
self.managerId = managerId
self.isDeleted = isDeleted
```

```
class Car:
```

```
def __init__(self, name, carId, type, price, model, isDeleted=False):
    """
```

Initialize a Car object with the given name, car ID, type, price, and model.

Parameters:

name (str): The name of the car.

carId (int): The ID of the car.

type (str): The type of the car.

price (float): The price of the car.

model (str): The model of the car.

isDeleted (bool, optional): Whether the car has been deleted from the system. Defaults to False.

```
"""
```

```
self.carId = carId
```

```
self.type = type
```

```
self.price = price
```

```
self.isDeleted = isDeleted
```

```
self.model=model
```

```
self.name=name
```

**class Sales:**

```
"""
```

Initialize a Sale object with the given car ID, employee ID, and date.

Parameters:

carId (int): The ID of the car being sold.

emplId (int): The ID of the employee making the sale.

date (str): The date of the sale in YYYY-MM-DD format.

```
"""
```

```
def __init__(self, carId, emplId, date):
```

```
self.carId = carId
```

```
self.emplId = emplId
```

```
self.date = date
```

**class Driver:**

```
def __init__(self):
```

```
"""
```

Initialize an empty SalesSystem object with empty lists for saleList, managersList, salesManList, and carList.

```
"""
```

```
self.saleList = []
```

```
self.managersList = []
```

```
self.salesManList = []
```

```
self.carList = []
```

