Functions in driver class

Classes manipulation functions

```
def addSales(self, sale):
    """
Adds a new sale to the system.

Parameters:
sale (Sale): The sale to be added.

Returns:
bool: True if the sale was added successfully, False otherwise.
"""
    self.saleList.append(sale)
    print("okay")
    for car in self.carList:
        if car.carId==sale.carId and car.isDeleted == False:
            price=int(car.price)
    for emp in self.salesManList:
        if int(emp.empId)==int(sale.empId) and emp.isDeleted==False:
            print("okay3")
            emp.profit=emp.profit+(price*0.065)
            for man in self.managersList:
                if int(emp.managerId) == int(man.managerId):
                    print("okay4")
                    man.profit=man.profit+(price*0.035)
                    return True
    return False


def searchSalesByEmpId(self, empId):
    """
Searches for all sales made by a specific employee.

Parameters:
empId (str): The ID of the employee to search for.

Returns:
list: A list of all sales made by the specified employee.
"""
```

```
    return [sale for sale in self.saleList if sale.empId == empId]

def addCar(self, car):
    """
Adds a new car to the system.

Parameters:
car (Car): The car to be added.
"""
    self.carList.append(car)

def deleteCar(self, carId):
    """
Deletes a car from the system.

Parameters:
carId (str): The ID of the car to be deleted.

Returns:
bool: True if the car was deleted successfully, False otherwise.
"""
    for car in self.carList:
        if car.carId == carId:
            car.isDeleted = True
            return True
    return False

def modifyCar(self, carId, newPrice,newName,newModel,newType):
    """
Modifies the details of a car in the system.

Parameters:
carId (str): The ID of the car to be modified.
newPrice (float): The new price of the car.
newName (str): The new name of the car.
newModel (str): The new model of the car.
newType (str): The new type of the car.

Returns:
```

```
        bool: True if the car was modified successfully, False otherwise.
        """
        for car in self.carList:
            if car.carId == carId and car.isDeleted==False:
                car.price = newPrice
                car.name=newName
                car.model=newModel
                car.type=newType
                return True

    def searchCar(self, carId):
        """
        Searches for a car in the system by ID.

        Parameters:
        carId (str): The ID of the car to search for.

        Returns:
        Car or None: The car object if found, None otherwise.
        """
        for car in self.carList:
            if car.isDeleted==False:
                if car.carId == carId:
                    return car
        return None

    def addSalesMan(self, salesMan):
        """
        Adds a new salesperson to the system.

        Parameters:
        salesMan (SalesPerson): The salesperson to be added.
        """
        self.salesManList.append(salesMan)

    def deleteSalesMan(self, empId):
        """
        Deletes a salesperson from the system.
```

```
Parameters:
empId (str): The ID of the salesperson to be deleted.

Returns:
bool: True if the salesperson was deleted successfully, False otherwise.
"""

    for salesMan in self.salesManList:
        if int(salesMan.empId) ==int(empId) and salesMan.isDeleted==False:
            salesMan.isDeleted = True
            return True
    return False

def modifySalesMan(self, empId, newSalary,newAge,newDep,newName,newPass):
    """
Modifies the details of a salesperson in the system.

Parameters:
empId (str): The ID of the salesperson to be modified.
newSalary (float): The new salary of the salesperson.
newAge (int): The new age of the salesperson.
newDep (str): The new department of the salesperson.
newName (str): The new name of the salesperson.
newPass (str): The new passport number of the salesperson.

Returns:
bool: True if the salesperson was modified successfully, False otherwise.
"""
    for salesMan in self.salesManList:
        if int(salesMan.empId) == int(empId) and salesMan.isDeleted==False:
            salesMan.salary = newSalary
            salesMan.name=newName
            salesMan.age=newAge
            salesMan.department=newDep
            salesMan.passport=newPass
            return True
    return False

def searchSalesMan(self, empId):
```

```
    """
Searches for a salesperson in the system by ID.

Parameters:
empId (str): The ID of the salesperson to search for.

Returns:
SalesPerson or None: The salesperson object if found, None otherwise.
"""
    for salesMan in self.salesManList:
        if int(salesMan.empId) == int(empId) and salesMan.isDeleted==False:
            return salesMan
    return None

def addManager(self, manager):
    """
Adds a new manager to the system.

Parameters:
manager (Manager): The manager to be added.
"""
    self.managersList.append(manager)

def deleteManager(self, managerId):
    """Deletes a manager from the managersList by setting isDeleted flag to True
and returns True. If no manager with the given ID is found, returns False.
Also checks if the manager is linked with any salesperson before deleting.

Args:
- managerId (int): ID of the manager to be deleted

Returns:
- bool: True if manager is deleted successfully, False otherwise
"""
    for emp in self.salesManList:
        if int(emp.managerId)== int(managerId) and emp.isDeleted==False:
            return False
    for manager in self.managersList:
        if int(manager.managerId) == int(managerId) and manager.isDeleted==False:
```

```
            manager.isDeleted = True
            return True
    return False

def modifyManager(self, managerId, newSalary,newAge,newDep,newName,newPass):
    """Modifies the details of a manager with the given managerId in the managersList
and returns True. If no manager with the given ID is found, returns False.

Args:
- managerId (int): ID of the manager to be modified
- newSalary (float): New salary of the manager
- newAge (int): New age of the manager
- newDep (str): New department of the manager
- newName (str): New name of the manager
- newPass (str): New passport of the manager

Returns:
- bool: True if manager is modified successfully, False otherwise
"""
    for manager in self.managersList:
        if int(manager.managerId) == int(managerId) and manager.isDeleted==False:
            manager.salary = newSalary
            manager.name=newName
            manager.age=newAge
            manager.department=newDep
            manager.passport=newPass
            return True
    return False




def search_manager_by_id(self,manid):
    """Searches for a manager with the given ID in the managersList and returns True
if found, False otherwise.

Args:
- manid (int): ID of the manager to be searched
```

```
Returns:
- bool: True if manager with the given ID is found, False otherwise
"""
    for manager in self.managersList:

        if int(manager.managerId) == int(manid):
            return True
    return False
```