**HR-Module** Application PresentationLayer worker - manager : PersonnelManager ServiceLayer FactoryService ShiftService EmployeeService BranchService <<Singleton>> PersonnelManager Domain Layer EmployeeController + employees: Hashmap<Integer,Contract> - Employees : HashMap<String, Employee> + ShiftsHistory: LinkedList<Integer> + addEmployee(id:integer,name:string,contract:contract,bank:bankaccount): String
+ removeEmployee(id:integer):string
+ addShiftToweek(shiftd:integer,id:integer):string - Name: String - Available employees : Hashmap<Integer , LinkedList<Integer>> +showAvailableShifts(id:integer):linkedlist<integer>
+addDriver(shift id , driver id): String
+checkLicense(Integer driverLicense, IntegerrequiredLicense) -schedule: HashMap<Integer, HashMap<Integer,String> <<Singleton>>
ShiftController +buildShift(id:integer,role:string):string +unbuildShift(shiftid:integer,id:integer):string + shifts: Hashmap<Integer,Shift> +showAvailableEmployees():string +checkEmployees():string +getSchedule():string Employee +addShift(shiftid:integer,time:localdate, +getShiftsHistory():Linkedlist<integer> minworker:integer,type:string,branchid:integer) + name: String :string +getShift(shiftid:integer):Shift + Dealdetails: Contract +addEmployee(shiftid:integer,id:integer):string + bankAccount: BankAccount - WeeklyShifts: HashMap<Integer,Boolean> - ShiftsHistory: LinkedList<Integer> -Employeeld : Integer <<Singleton>> -roles : LinkedList Branchontroller -BranchId : Integer - Branches : HashMap<Integer, Branch> licenseType: string + minimumWorkers: Integer at least +addShiftToWeek(Integer shiftId) : String one Shift +addBranch(branchid:integer,location:string) + employees:LinkedList<Integer> 1..\* Manager +removeBranch(branchid:integr):string +removeShiftFromWeek(Integer ShiftId) : String + type: String + ShiftMangerld: Integer BankAccount + Cancelations: LinkedList<Integer> - Username : String + time : LocalDate - Password : String + ShiftId: Integer - Balance : integer Branch Contract - Branchid : Integer + location: String + startdate: LocalDate +PossibleToDeliver(deliverytime,reqLicense) - roles : LinkedList<String> + salary: Integer - BranchId : Integer + employmentType: String + BranchEmployes :LinkedList<Integer> + branchID: Integer - BranchShiftList: LinkedList<Integer> + ContracrtId: Integer DataAccessLayer DTO'S EmployeeDTO BranchDTO DAO'S + name: String + location: String + ContractId : Integer - BranchId : Integer + BankUsername : String 1..\* + Role : String ;-----includes-----0..\* -BranchId : Integer <<Singleton>> ShiftDAO <<Singleton>> -Employeeld : Integer EmployeeDAO + connection : Connection + connection : Connection + Role : String + getConnection():Connection + getConnection():Connection + setConnection(Connection):void + setConnection(Connection):void +insert(product:ProductDTO):void +insert(employee:EmployeeDTO):void <<Singleton>>
WeeklyShiftsDAO <<Singleton>> + connection : Connection BranchDAO + getConnection():Connection + connection : Connection PersonnelManagerDTO ShiftDTO +setConnection(Connection):void + getConnection():Connection + Name : String -----includes · --> + minimumWorkers: Integer +insert(Weekly:WeeklyDTO):void + setConnection(Connection):void - Branchid : Integer +insert(branch:BranchDTO):void contained in WeeklyShiftsDTO <<Singleton>>
PersonnelDAO + type: String 0..\* + Employeld : Integer + connection : Connection + time : LocalDate <<Singleton>>
RolesDAO + BranchId : Integer RolesDTO + ShiftId: Integer + ShiftId : Integer + connection : Connection + getConnection():Connection includes + Employeeld : Integer + BranchId : Integer 0..\* 0..\* setConnection(Connection):void ShiftHistoryDAO includes + Role : String ShiftsHistoryDTO +insert(roles:RolesDTO):void + connection : Connection + Employeld : Integer + BranchId : Integer <Singleton>>
BankAccountDAO ContractDAO + ShiftId : Integer ContractDTO ScheduleDTO + connection : Connection + connection : Connection + startdate: LocalDate + ShiftId : Integer + getConnection():Connection ·----getConnection():Connection + salary: Integer + Employeeld : Integer + setConnection(Connection):void setConnection(Connection):void + BranchId : Integer +insert(contract:ContractDTO):void + employmentType: String +insert(Bank:BankDTO):void + HRName : String BankAccountDTO + branchID: Integer <<Singleton>> - Username : String ScheduleDAO + ContracrtId: Integer - Password : String + connection : Connection - Balance : integer SuperLee DataBase **Delivery - Module Presentation Layer** delivery Applecation - deliveryApp: mainService Service Layer mainService - driverController: DriverController - driverService: DriverService - delievryController: DelieveryController - deliverySerivce: DeliveryService - truckController: TruckContrller - locationController: LocationController - locationService: LocationService **DriverService** - driverController: driverController DeliveryService TruckService - deliveryController: DeliveryController - truckController: TruckController LocatioService - locationController: LocationController Domain Layer Controllers **Main Controller** deliveryController: DeliveryController - truckController: TruckController - driverController: DriverController - locationController: LocationController storeKeepers:List<StoreKeeper> + addDriver(driver:Driver) + addTruck( truck:Trck) + addLocation(location: Location) + checkWeight(deliveryld: int)
+ addStoreKeeper(storekeeper: Storekeeper) + updateShift(employeeld: int, shift: Shift)
- loadData() <<Singleton>> <<Singleton>> DriverController DeliveryController - deliveries: Map<Integer, Delivery> - drivers: List<Driver> + addDelivery(delivery: Delivery) + addDriver( driverDriver) - loadData() loadData() <<singleton>>TruckController -trucks: LinkedList<Truck> <<Singleton>>LocationController + addTruck(truck: Truck) - locations: Map<Integer,Location> - loadData() + addLocation(location: Location) - loadData() Objects Delivery Location - ld: int Driver - date: Time LocationId: int · humanld: int - time: Srtring address: string - truck: Truck name: string - driver: Driver contactNumber: int · licenseType: string contactName: string - source: Location - isAvaliable: boolean - quantity: Map<Item, Integer>
- docs: Map<Integer,LocItemDoc> weeklyshifts: map<int, shift> 1,\* - area: String LocItemDoc Item delieveryld: int Truck name: String - docld: int weight: int - locItems: Map<Item,Integer> number: int - destinationId: int - currentTruckWeight: int - model: string - addressLoc: String - truckWeight: int - maxWaight: int - isAvaliable: boolean - contactNumber: int - contactName: String DataAccessLayer DTO'S «abstract» DTO DriverDTO LocationDTO Extends - humanld: int - ld :int - name: string - address: string - licenseType: string contactNumber: stringcontactName: string - isAvaliable: int LocItemDocDTO - deliveryld: int DeliveryDTO driveryld: int - documentationId: int locationId: intcurrentTruckWeight: int - exitTime: string - arrivalTime: string - truckld: int - Qid: int TruckDTO - driverId: int - driverId: int - number: int - sourceld: int - model: string - area: String **ItemsQuantityDTO** - weight: int - maxWeight: int - Qld: int - isAvaliable: int - ItemName: string - quantity: int - deliveryld: int - itemsWeight: int DAO's DataDAO - tableName: string + Select(): List<DTO> Extends DriverDAO + Insert(driver: DriverDTO ): boolean

+ delete(driverId: int): boolean

LocationDAO

+ addLocation(location: LocationDAO): boolean + removeLocation(locationId: int): boolean + updateDelivery(deliveryId: int): boolean

ItemsQuantityDAO

+ add(toAdd: ItemsQuantityDTO): boolean

+ remove( id: int): boolean

LocItemDocDAO

+ add(toAdd:LocItemDocDAO): boolean

+ remove(id: int): boolean

DelieveryDAO

+ addDelivery(delivey: DeliveryDTO):boolean + removeDelivery(deliveryId: int): boolean + updateDriver(driver: int): boolean + updateTruck(truck: int): boolean

TruckDAO

+ addTruck(truck: TruckDAO): boolean + removeTruck(truckId: int): boolean