Presentation Layer DeliveryManagerService + deliveryManager: DeliveryManagerImp UserInteractionUtil UserInterface ~ scanner: Scanner implements Buisnes Layer <<enumeration>> TruckType <<enumeration>> DeliveryStatus - contactName: String + Refrigerated: - address: String + DELIVERED: - name: String + IN_TRANSIT: - deliveryZone: int + CANCELLED: - contactPhone: String + NOT_STARTED: License + computeDistance(Site otherSite) : int - weightAllowedTons: int <<enumeration>> - truckTypesAllowed: Set<TruckType> **Availability** + Busy: + isLicensed(Truck truck) : boolean Driver + Available: DeliveryStop ı- name: String - items: Map<String, Integer> - id: String - shipmentInstanceID: int - availability: Availability - status: DeliveryStatus - phone: String - origin: Site - license: License Truck - destination: Site +freeDriver(): void - model: String truckTypeRequired: TruckType +isLicensed(Truck truck) : boolean - netWeightTons: int - estimatedArrivalTime: Timestamp - maxWeightTons: int + updateArrivalTime(Timestamp dispatchlTime) : void - availability: Availability - type: TruckType SiteGenerator - licensePlate: String + sitesList: List<Site> +freeTruck(): void - ZONES: int[] - NAMES: String[] DeliveryException DeliveryForm - formld: int - deliveryFormsController: DeliveryFormsController - stopToCancel: DeliveryStop - maxWeightAllowed: int - dispatchWeightTons: int DriverController 1- originSite: Site - truck: Truck <<interface>> - instance: DriverController WeightMeasurer TruckController 1- driverDAO: DriverDAO - deliveryManager: DeliveryManagerImpl HR_Deliveries_interfaces - hrManager: HRIntegrator - trucks: HashSet<Truck> - driver: Driver - drivers: HashSet<Driver> - weightMeasurer: WeightMeasurer - instance: TruckController - truckDAO: TruckDAO - hrManager: HRIntegrator + pickDriver(Truck truck, Timestamp startTime, Timestamp finishTime) : Driver - dispatchTime: Timestamp <<interface>> + notifyHR(Timestamp startTime, Timestamp finishTime, Driver driver) : void + pickTruck(TruckType requiredType) : Truck HRIntegrator - destinationSitesToVisit: List<DeliveryStop> + pickTruck(TruckType requiredType, int requiredWeight) : Truck - destinationSitesVisited: List<DeliveryStop> + visitDeliveryStop(DeliveryStop deliveryStop) : void + performWeightCheck() : void + startJourney() : void <<interface>> <-interface>>
DeliveryIntegrator + completeJourney() : void + updateArrivalTimes() : void + updateArrivalTimesgetStopsByTime(Timestamp startTime() : List<DeliveryStop> + cancelForm() :void DeliveryManagerImpl <<interface>>
TripReplanner - pendingDeliveryStops: Set<DeliveryStop> deliveryCount: int truckController: TruckController <<interface>> DeliveryManager - driverController: DriverController tripReplanner: TripReplanner deliveryFormsController: DeliveryFormsController instance: DeliveryManagerImpl DeliveryFormsController + replanDelivery(DeliveryForm form) : void - instance: DeliveryFormsController + addDeliveryStop(DeliveryStop deliveryStop): void - pendingDeliveryForms: Set<DeliveryForm> + removeDeliveryStop(int deliveryId): void - driverController: DriverController + replaceTruck(DeliveryForm form): void deliveryFormCount: int - deliveryFormDAO: DeliveryFormDAO - completedDeliveryForms: Set<DeliveryForm> - truckController: TruckController (Timestamp startTime+ getDeliveryByArrivalTime(Timestamp startTime, Timestamp finishTime, String store) : Set<DeliveryStop> + CreateForm (stops:List<DeliveryStop>, origin: Site) : void + getTruckType(List<DeliveryStop> destinationSitesToVisit) : TruckType + startDeliveryForm(DeliveryForm deliveryForm) : void + terminateDeliveryForm(DeliveryForm deliveryForm) : void DAL TruckDAO DeliveryFormDAO - conn: Connect Connect + conn: Connection - instance: Connect