Requirements – Transport Module

ID	Module	Functional / Non-Functional	Description	Priority	Risk	Status
1	TRP	functional	The system will keep the following information for each transport: unique ID, date, departure hour, truck license number, driver ID, source, destinations, uniquely identifiable list of items for each destination, truck weight	HP	LR	done
2	TRP	functional	The system will keep the following information for each site: transport zone, address, phone number, contact person name, site type	HP	LR	done
3	TRP	functional	The system will keep the following information for each truck: license number, model, base weight, maximum weight, cooling capacity	HP	LR	done
4	TRP	functional	The system will keep the following information for each driver: unique ID, name, driving license type	HP	LR	done
5	TRP	functional	The system will allow adding new transports, drivers, trucks, sites, item lists	HP	LR	done
6	TRP	functional	The system will allow editing transports, drivers, trucks, sites and item lists.	HP	LR	done
7	TRP	functional	The system will allow deleting transports, drivers, trucks, sites and item lists.	HP	LR	done
8	TRP	non-functional	The truck's cooling capacities are: 1:"none",2:"cold",3:"frozen"	HP	LR	done
9	TRP	non-functional	The site types are the following: Logistical center, branch, supplier	HP	LR	done

ID	Module	Functional / Non-Functional	Description	Priority	Risk	Status
10	TRP	non-function	The types of driving licenses are defined as pairs of letter and number as such: 1-3 for cooling capacity and A-C for max weight. A : 0 - 10,000kg B : 0 - 20,000kg C : no limit 1 : no cooling 2 : cold 3 : frozen for example: C1, B3, B1 etc	HP	LR	done
11	TRP	non-function	The driver's license adequacy is ordered as such: A1<B1<C1<C2<C3 A2<B2<C2<C3 A3<B3<C3 which means that drivers can drive trucks that require a weaker license	HP	LR	done
12	TRP	functional	The system will reject the transport if the weight is higher than the maximum weight of the truck.	HP	LR	done
13	TRP	functional	The system will reject the transport if the driver's license is not adequate for the truck	HP	LR	done
14	TRP	functional	If a transport is rejected, the system will allow choosing between: changing or removing a destination, changing the truck, or changing the item list	HP	HR	done
15	TRP	functional	The system will enforce that the driver of the transport has the adequate license type for the truck according to the cooling capacity and the weight of the truck	HP	HR	done
16	TRP	functional	The system will not allow a truck to be assigned to two overlapping transports	LP	HR	backlog
17	TRP	functional	The system will not allow a driver to be assigned to two overlapping transports	LP	HR	backlog

ID	Module	Functional / Non-Functional	Description	Priority	Risk	Status
18	TRP	functional	The system will document every transport in the transport history database	HP	HR	backlog
19	TRP	functional	The system will document the list of items in the item history database	HP	HR	backlog

System assumptions (things we don't know but we assumed)**:**

1) As of now, we're assuming every driver and every truck is always available for a transport, regardless of whether they actually are. We are not enforcing this requirement because it's not in this module's jurisdiction to verify these details.

2) We're assuming 9 types of drivers' licenses exist, based on the max weight of the truck and the cooling capacity of the truck.

   A: 0-10 Tons, B: 0-20 Tons, C: no limit
   1: no cooling, 2: cold, 3: frozen

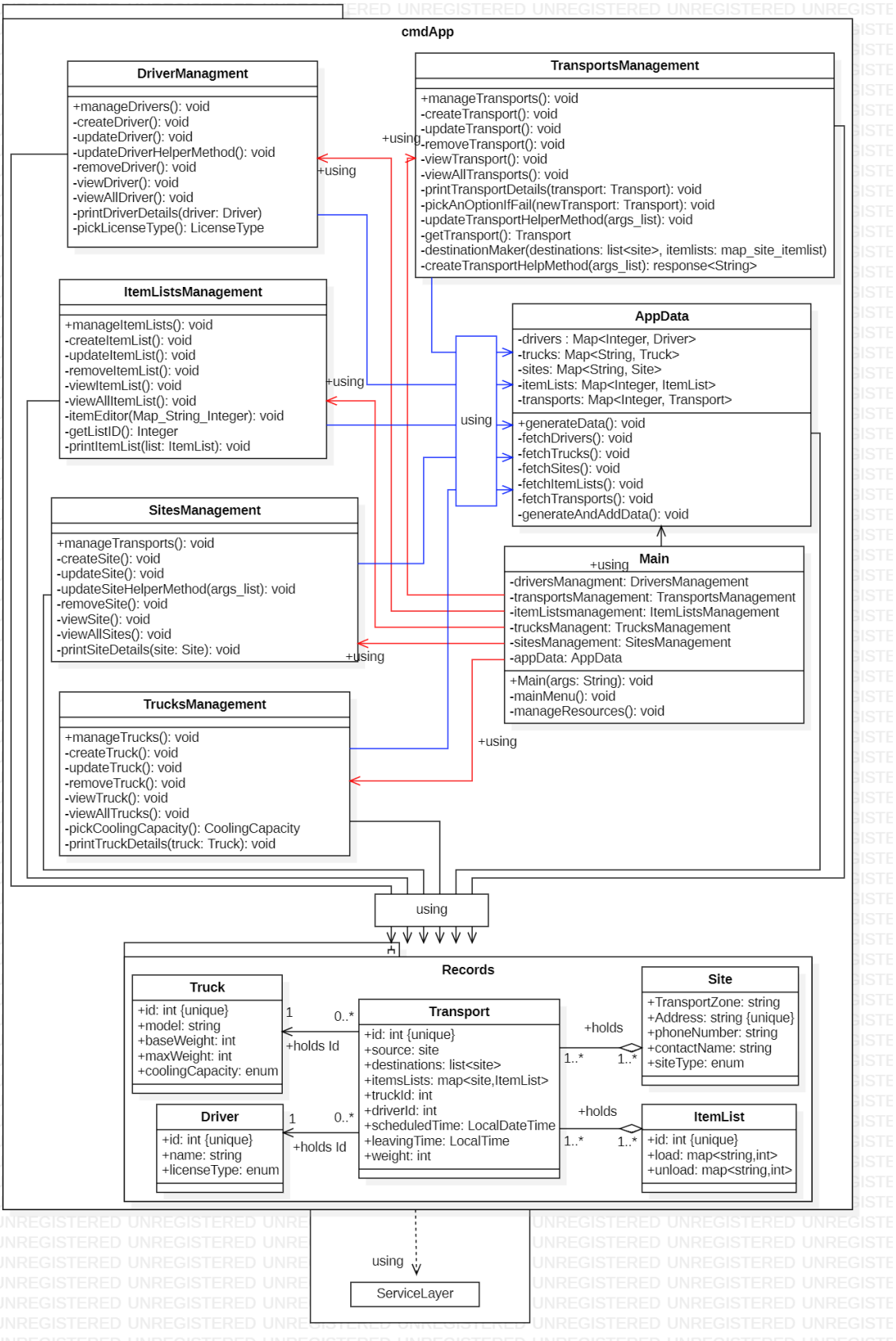
   every combination of these letters and numbers are a type of license number.
   for example: A1, B2, C3 etc.

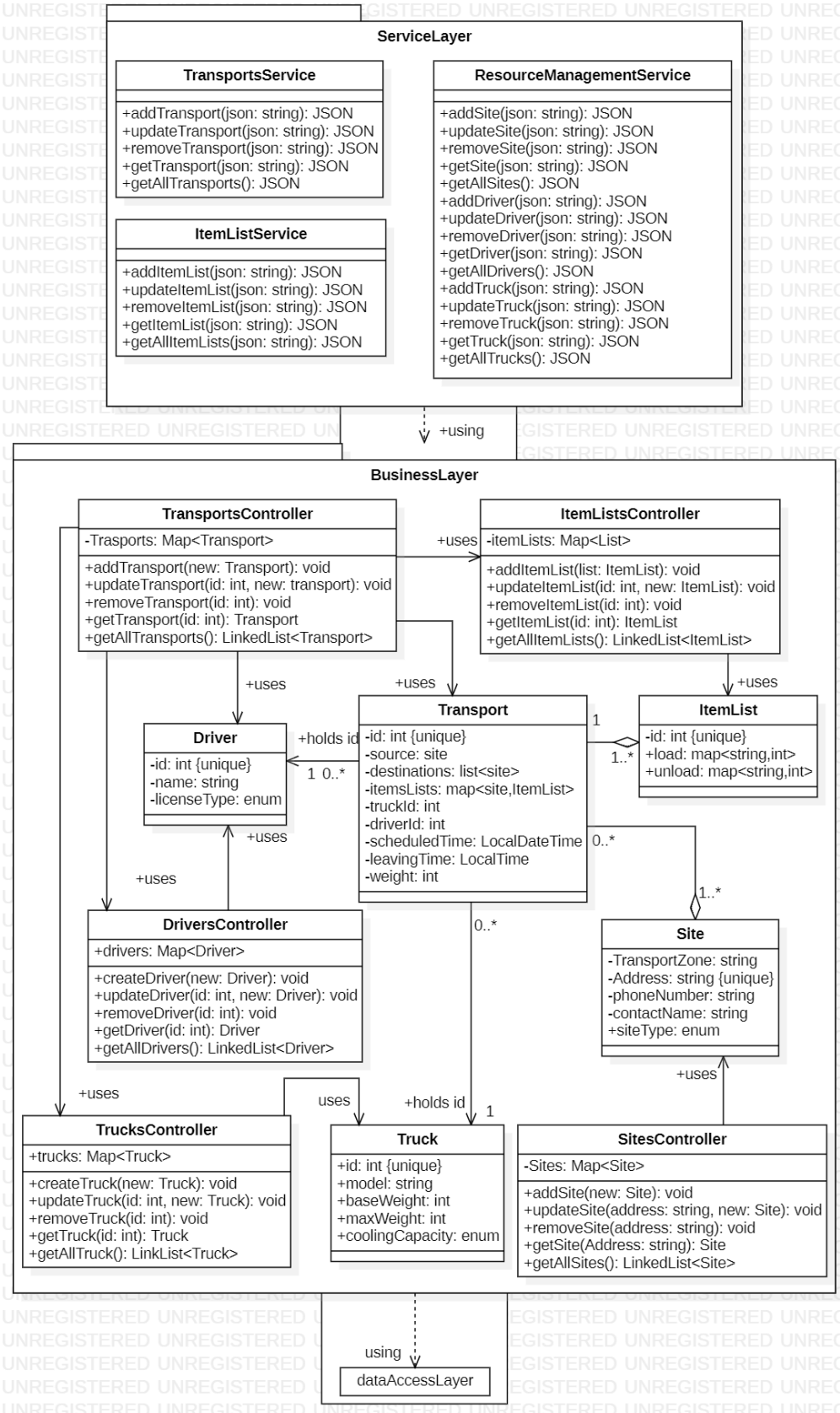
3) When the transport weight exceeds the max weight of the truck, we're allowing to change both the truck and the driver if necessary to accommodate the heavier truck.

Open questions:

Questions	
1	How do you know when a driver is available to be put in a transport?
2	How do you know when a truck is available to be put in a transport?
3	There is no information about when a transports ends, which means we can't know the answers to questions 1 and 2. How do we get this information?

Class diagram




ServiceLayer
TransportsService
+addTransport(json: string): JSON
+updateTransport(json: string): JSON
+removeTransport(json: string): JSON
+getTransport(json: string): JSON
+getAllTransports(): JSON
ItemListService
+addItemList(json: string): JSON
+updateItemList(json: string): JSON
+removeItemList(json: string): JSON
+getItemList(json: string): JSON
+getAllItemLists(json: string): JSON
ResourceManagementService
+addSite(json: string): JSON
+updateSite(json: string): JSON
+removeSite(json: string): JSON
+getSite(json: string): JSON
+getAllSites(): JSON
+addDriver(json: string): JSON
+updateDriver(json: string): JSON
+removeDriver(json: string): JSON
+getDriver(json: string): JSON
+getAllDrivers(): JSON
+addTruck(json: string): JSON
+updateTruck(json: string): JSON
+removeTruck(json: string): JSON
+getTruck(json: string): JSON
+getAllTrucks(): JSON
+using
BusinessLayer
TransportsController
-Trasports: Map<Transport>
+addTransport(new: Transport): void
+updateTransport(id: int, new: transport): void
+removeTransport(id: int): void
+getTransport(id: int): Transport
+getAllTransports(): LinkedList<Transport>
+uses
ItemListsController
-itemLists: Map<List>
+addItemList(list: ItemList): void
+updateItemList(id: int, new: ItemList): void
+removeItemList(id: int): void
+getItemList(id: int): ItemList
+getAllItemLists(): LinkedList<ItemList>
Driver
-id: int {unique}
-name: string
-licenseType: enum
+holds id
1 0..*
Transport
-id: int {unique}
-source: site
-destinations: list<site>
-itemsLists: map<site,ItemList>
-truckId: int
-driverId: int
-scheduledTime: LocalDateTime
-leavingTime: LocalTime
-weight: int
1
1..*
ItemList
-id: int {unique}
+load: map<string,int>
+unload: map<string,int>
0..*
1..*
DriversController
+drivers: Map<Driver>
+createDriver(new: Driver): void
+updateDriver(id: int, new: Driver): void
+removeDriver(id: int): void
+getDriver(id: int): Driver
+getAllDrivers(): LinkedList<Driver>
Site
-TransportZone: string
-Address: string {unique}
-phoneNumber: string
-contactName: string
+siteType: enum
TrucksController
+trucks: Map<Truck>
+createTruck(new: Truck): void
+updateTruck(id: int, new: Truck): void
+removeTruck(id: int): void
+getTruck(id: int): Truck
+getAllTruck(): LinkList<Truck>
uses
+holds id
1
Truck
+id: int {unique}
+model: string
+baseWeight: int
+maxWeight: int
+coolingCapacity: enum
SitesController
-Sites: Map<Site>
+addSite(new: Site): void
+updateSite(address: string, new: Site): void
+removeSite(address: string): void
+getSite(Address: string): Site
+getAllSites(): LinkedList<Site>
using
dataAccessLayer

Object Diagram 1 :

The following diagram describes the transport that take place on

14-05-23, performe by driver named Moshe (his id = 0), the transport will be carried out by a Toyota truck whose identification number is 4354 and will be at 09:00 in the morning .

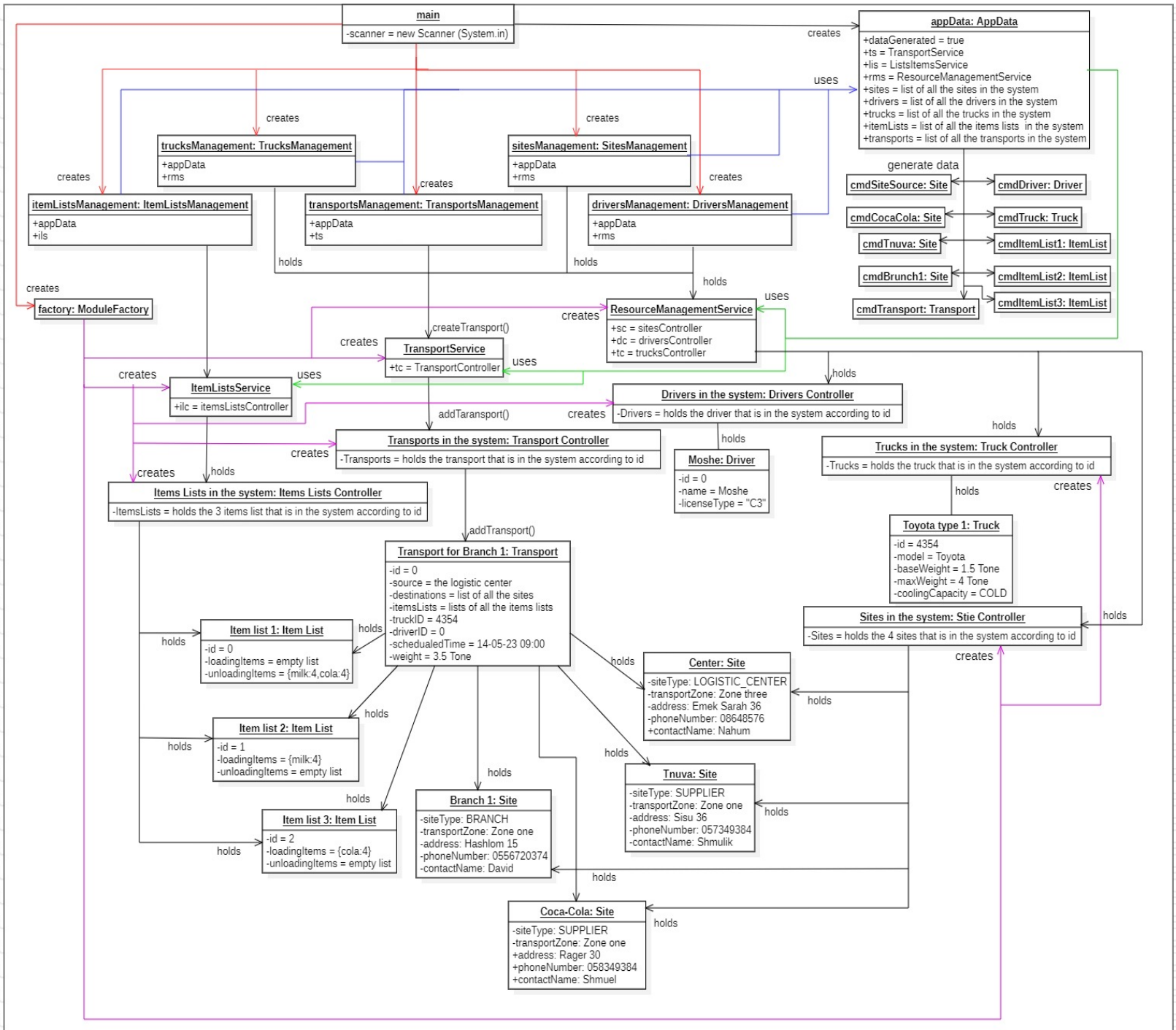
The following assumptions must be made about the scenario:

1) The driver has a driver's license suitable for the truck

2) The transport start from the logistics center

3) The system is described at the end of the transport - describes a snapshot of the system

4)It is assumed that all objects added to the transport exist in the system



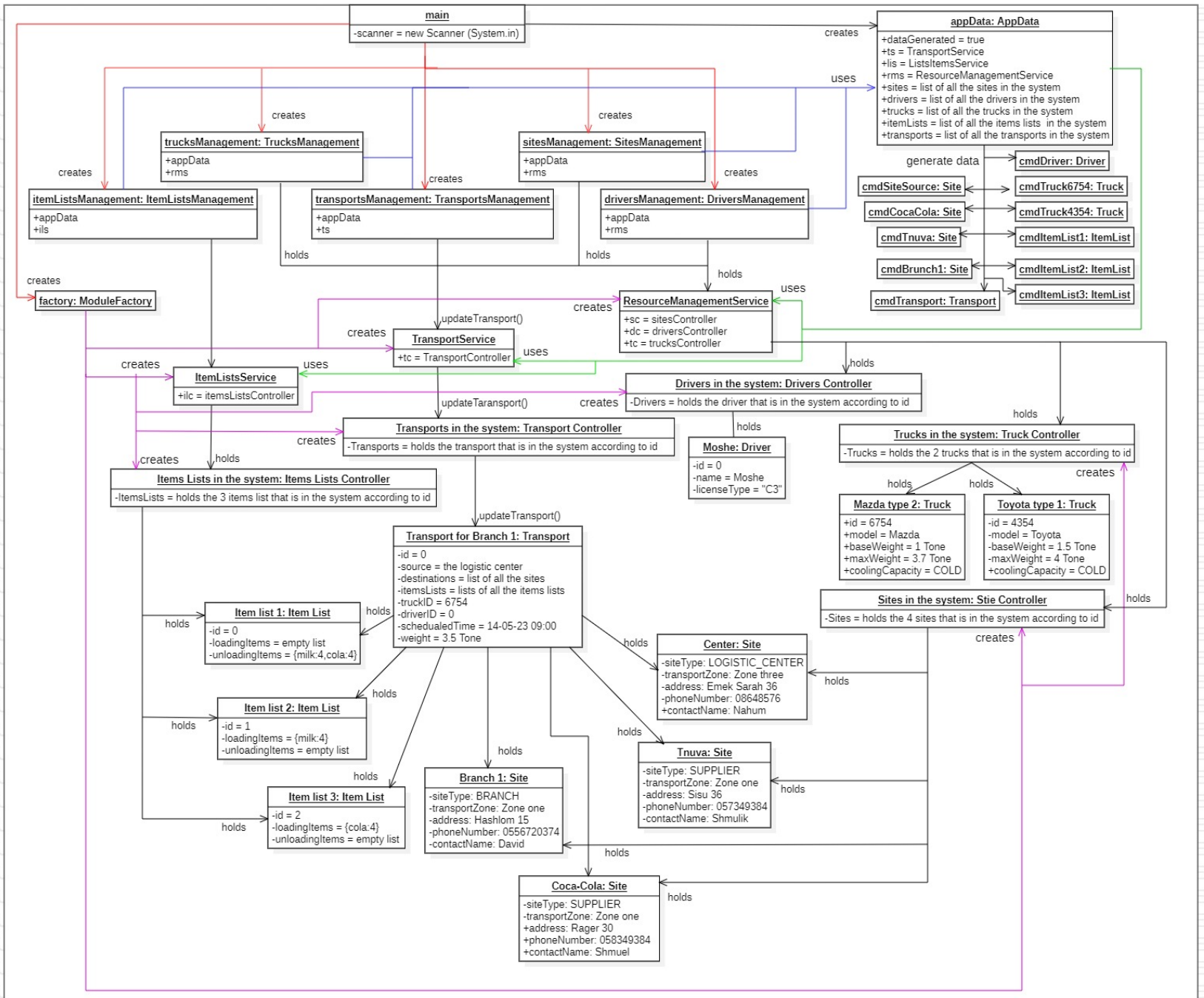
Object Diagram 2 :

On 13-05-23 the logistics center called Dani the transport manager and informed him that truck identification number 4354 had a flat tire, Dani looked in the system to see if there a truck that suitable for the transport that should be carried out on 14-05-23 and indeed found that truck number 6754 was suitable for that transport, so he updated in the system the new data about this transport.

The following assumptions must be made about the scenario:

1) The driver has a driver's license suitable for the new truck

2) The transport start from the logistics center

3) The system is described after the update - describes a snapshot of the system

4)It is assumed that truck number 6754 is exist in the system in addition to all the objects of the existing transport