Practical assignment 2 – Part A

Deliveries module + HR module

Submitted by:

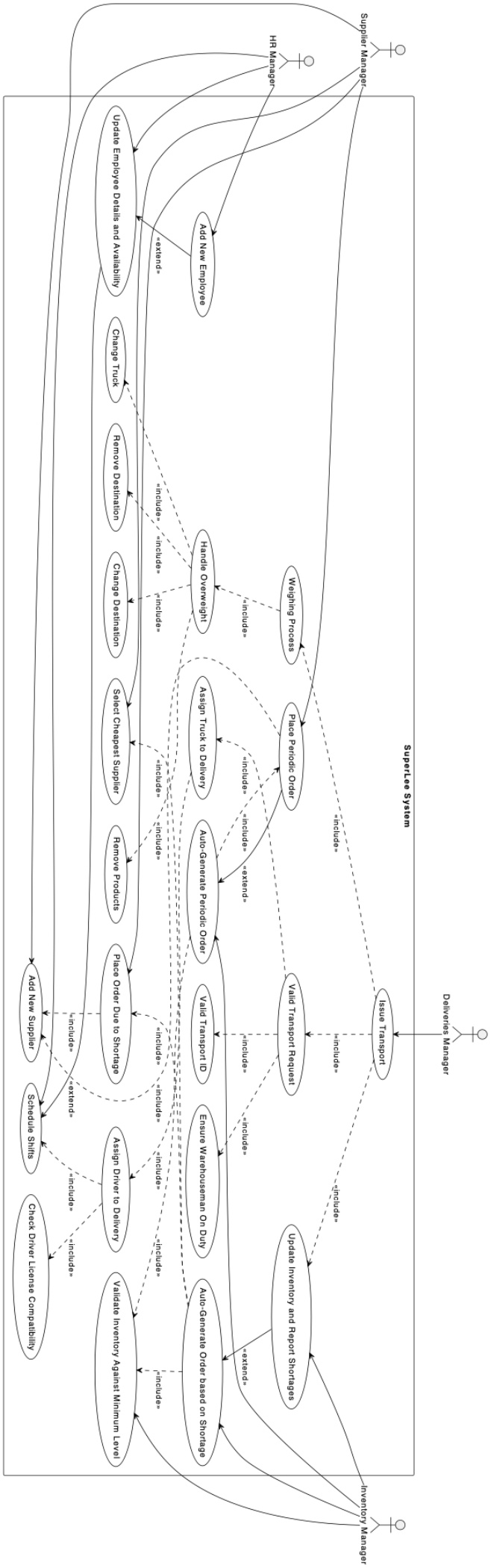
Shahar Shamir 318861820

Lital Kupchick 318567914

Hadar Ofer 208583781

Bar Pesso 315175554

1.1.


Supplier Manager
HR Manager
Deliveries Manager
Inventory Manager
SuperLee System
Update Employee Details and Availability
Add New Employee
Change Truck
Remove Destination
Change Destination
Select Cheapest Supplier
Remove Products
Add New Supplier
Schedule Shifts
Check Driver License Compatibility
Handle Overweight
Weighing Process
Assign Truck to Delivery
Place Periodic Order
Auto-Generate Periodic Order
Place Order Due to Shortage
Assign Driver to Delivery
Valid Transport ID
Valid Transport Request
Issue Transport
Ensure Warehouseman On Duty
Validate Inventory Against Minimum Level
Update Inventory and Report Shortages
Auto-Generate Order based on Shortage
«extend»
«include»

1.2.

Use case name	Assigning employees to shifts
Textual Description	After the HR Manager creates the shifts for the upcoming week, he selects a specific shift to assign employees. The system displays the required roles for that shift and lists all available and qualified employees. The HR Manager assigns each employee to the appropriate role based on availability and qualifications.
List of Actors	HR Manager
Pre-conditions	-The HR Manager is identified and authenticated. - Shifts for the relevant week have already been created and are active. - The HR Manager has defined all required roles for the selected shift. - There are employees who are both qualified and available for the required roles in that shift.
Post-conditions	- All required roles for the selected shift have been successfully assigned to qualified employees.
Main success scenario	1. The HR Manager logs into the system 2. The system displays a list of available shifts 3. HR Manager selects a shift 4. System displays required roles for the selected shift 5. WHILE there are unassigned roles: 5.1 The system displays list of unassigned roles 5.2 HR Manager selects a role to assign 5.3 System displays available employees for the selected role 5.4 HR Manager selects an employee for the role 5.5 System confirms assignment 6. After all roles are processed: The system verifies if all roles were assigned and includes a Shift Manager 7. System displays confirmation that the shift is valid and saved
Alternatives / Extensions	Extension 1: If the selected shift is invalid, the system displays an error message and returns to the shift selection. Extension 2: If no required roles are defined for the selected shift, the system displays a message and exits the use case. Extension 3: If no available and qualified employees are found for the selected role, the system displays a message and returns to role selection. Extension 4: If the HR Manager chooses to skip the role assignment, no assignment is made, and the system returns to role selection. Extension 5: If the selected employee is invalid, the system displays an error and prompts for a valid selection. Extension 6: If not all roles are assigned or a Shift Manager is missing, the system displays a warning and exits the use case.

PSAUDO-CODE:

```
BEGIN UseCase "Assigning Employees to Shifts"
  HR Manager logs into the system
  DISPLAY list of available shifts
  HR Manager selects a shift
  IF invalid shift selection THEN       // E1
     DISPLAY error message
     EXIT
  ENDIF
  DISPLAY required roles for selected shift
  IF no roles defined THEN              // E2
     DISPLAY message
     EXIT
  ENDIF
  WHILE there are unassigned roles DO
     DISPLAY list of unassigned roles
     HR Manager selects a role
     IF invalid role selection THEN    // E4
        DISPLAY error message
        CONTINUE
     ENDIF
     DISPLAY list of available employees for role
     IF no available employees THEN    // E3
        DISPLAY message
        CONTINUE
     ENDIF

     HR Manager selects an employee

     IF skip assignment THEN          // E4
        CONTINUE
```

```
    ENDIF
    IF invalid employee selection THEN // E5
      DISPLAY error message
      CONTINUE
    ENDIF
    ASSIGN employee to role
  END WHILE

  VERIFY all roles assigned and Shift Manager present

  IF not valid THEN                // E6
    DISPLAY error message
    EXIT
  ENDIF
  DISPLAY confirmation of successful shift assignment
END UseCase
```

Use case name	Issuing a Transport
Textual Description	The system allows the Transport Manager to initiate and issue a new Transport. The system guides the manager through either manual or automatic assignment of a driver and truck, verifies availability, validates the transport ID and weight, and provides corrective options if needed. Once all validations pass, the transport is created and stored in the system's database.
List of Actors	Transport Manager
Pre-conditions	The Transport Manager must be logged into the system. The system must have available trucks, drivers, and warehouse workers.
Post-conditions	A new Transport is created and saved. A driver and truck are assigned. All transport details are stored in the system's database.
Main success scenario	1. The Transport Manager selects the option to issue a transport. 2. The system prompts for assignment mode: automatic or manual. 3. The Transport Manager chooses an assignment mode. 4. The system collects transport details (driver, truck, destination, weight, etc.). 5. The system validates: - Driver license compatibility - Driver and warehouseman availability in the shift - Transport ID 6. The system checks whether the cargo weight exceeds the truck's capacity. 7. If all validations pass, the system creates and stores the new transport. 8. The system confirms the successful issuance of the transport.
Alternatives / Extensions	Extension 1: If the assignment mode is not selected, the system prompts the manager to choose a mode. Extension 2: If the driver is not qualified or unavailable, or if no warehouseman is assigned, the system displays an error message. Extension 3: If the transport ID is invalid, the system rejects the request and displays an error. Extension 4: If the cargo weight exceeds the limit, the system provides options to: - Remove a destination - Change the destination - Change the truck - Remove products

PSUEDO CODE

```
BEGIN UseCase "Issuing a Transport"
        AUTHENTICATE Transport Manager
        IF not logged in THEN
                DISPLAY "Access denied" ; EXIT
        ENDIF
        PROMPT "Choose Assignment Mode (Auto / Manual)"
        GET assignmentMode
        IF assignmentMode is NULL THEN
                DISPLAY "Please select assignment mode" ; EXIT
        ENDIF
        IF assignmentMode == "Manual" THEN
                PROMPT Manager to select Truck & Driver
        ELSE
                SYSTEM auto-assigns Truck & Driver
        ENDIF
        PROMPT for:
        - Destinations
        - Estimated Weight
        - Transport ID
        VALIDATE:
        - Driver license ↔ Truck type
        - Driver and warehouseman available in shift
        - Valid Transport ID
        IF validation fails THEN
                DISPLAY "Invalid transport request" ; EXIT
        ENDIF
        IF Estimated Weight > Truck Capacity THEN
                DISPLAY "Transport is overweight"
                DISPLAY correction options:
```

1. Remove Destination

2. Change Destination

3. Change Truck

4. Remove Products

GET correctionOption

APPLY correctionOption

REVALIDATE weight

IF still overweight THEN

DISPLAY "Transport cannot be issued" ; EXIT

ENDIF

ENDIF

CREATE & SAVE Transport

DISPLAY "Transport issued successfully"

END UseCase

2.1. Contract for Use Case: Assigning Employees to Shifts

Contract CO1:

Operation Name	assignEmployeeToShift(Scanner sc)
Reference	Use Case: Assign employees to shift.
Pre-conditions	-The HR Manager has created active shifts for the upcoming week. -The HR Manager has defined all required roles for the selected shift. -There are employees in the system who are both available and qualified for the required roles.
Post-conditions	-Each required role in the selected shift was assigned to a qualified employee (association formed). -The assigned employee was linked to the role within the shift (association formed). -The shift stored the assignment of each employee to their assigned role(attribute modification). -The assigned role was removed from the list of unassigned roles (attribute modification).

Contract CO2:

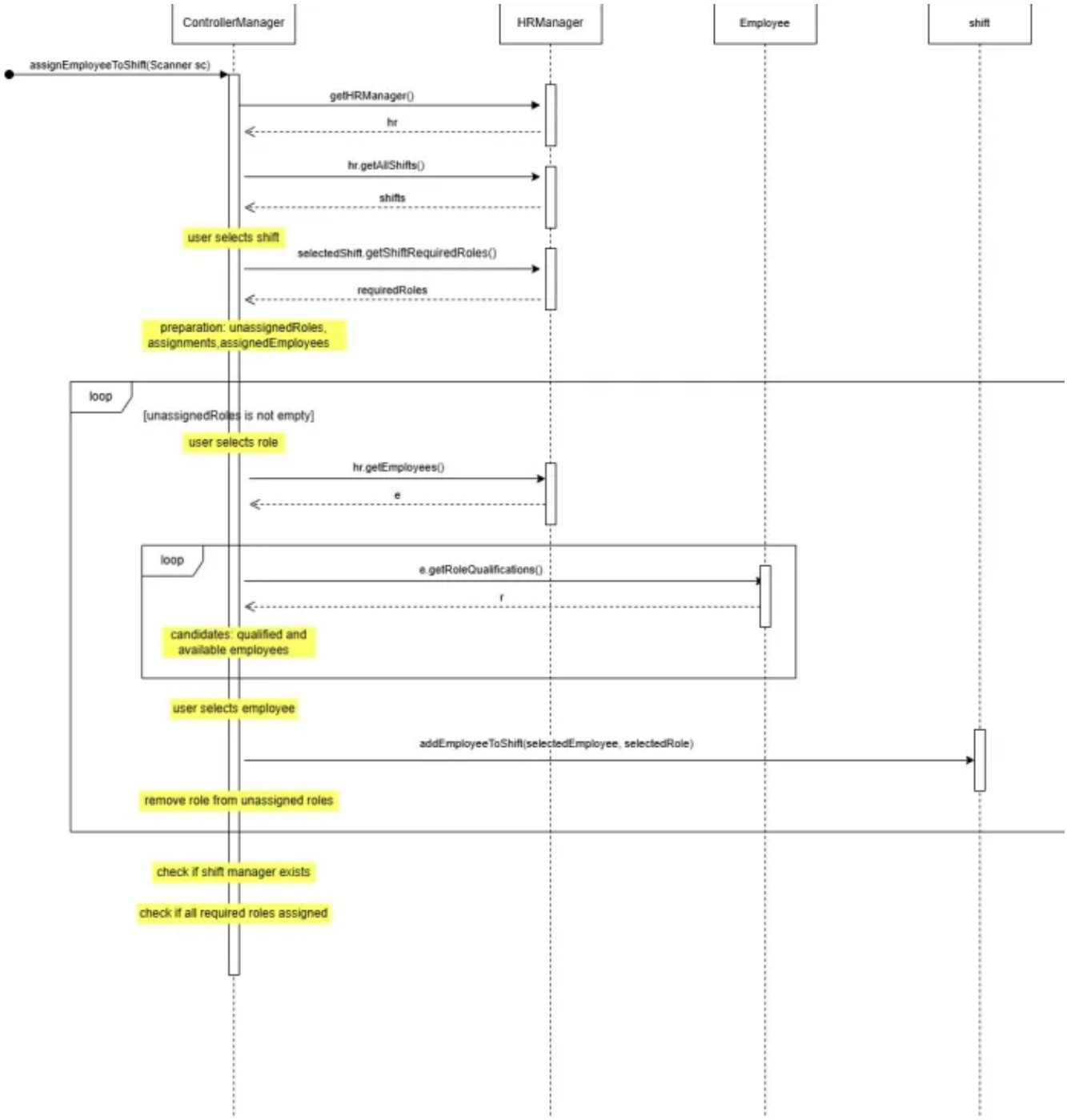
Operation Name	addEmployeeToShift(Employee employee, Role role)
Reference	Use Case: Assign employees to shift.
Pre-conditions	- The role is part of the selected shift required roles. - The employee has not already been assigned to the shift.
Post-conditions	- The employee was assigned to the selected shift in the specified role (association formed). - The shift stored the assignment of each employee to their assigned role (attribute modification). -The assigned role was removed from the list of unassigned roles (attribute modification).

Contract CO3 for Use Case: Issuing a Transport

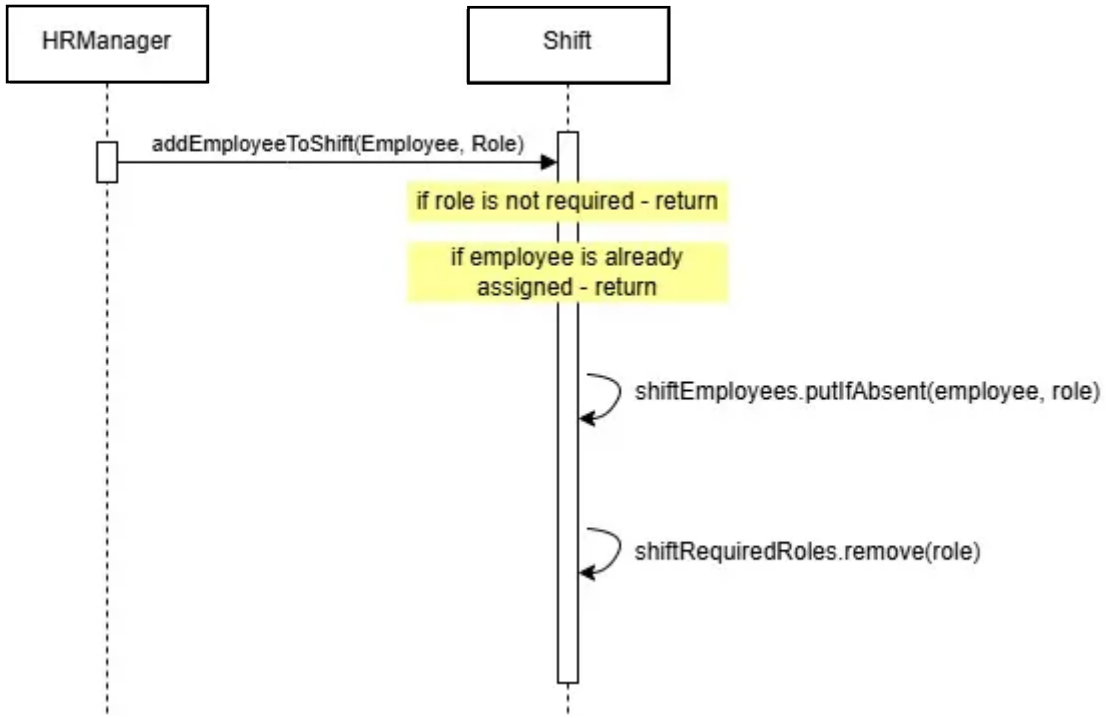
Operation Name	createTransport(TransportRequest request, TruckService t_service, DriverService d_service, ReportService reportService, ShipmentAreaService shipmentAreaService)
Reference	Use Case: Issuing a Transport
Pre-conditions	-Truck and Driver must be available - Driver must have suitable license type for the truck - Sites must be in the same shipment area - Weight must not exceed the truck's limit - A warehouseman must be on site during the shift in which the transport arrives
Post-conditions	- A new Transport was created and saved in the database - A new Transport report was created and saved in the database - The truck and driver assigned to the transport are marked as unavailable - System confirmed success

2.2. Sequence Diagram

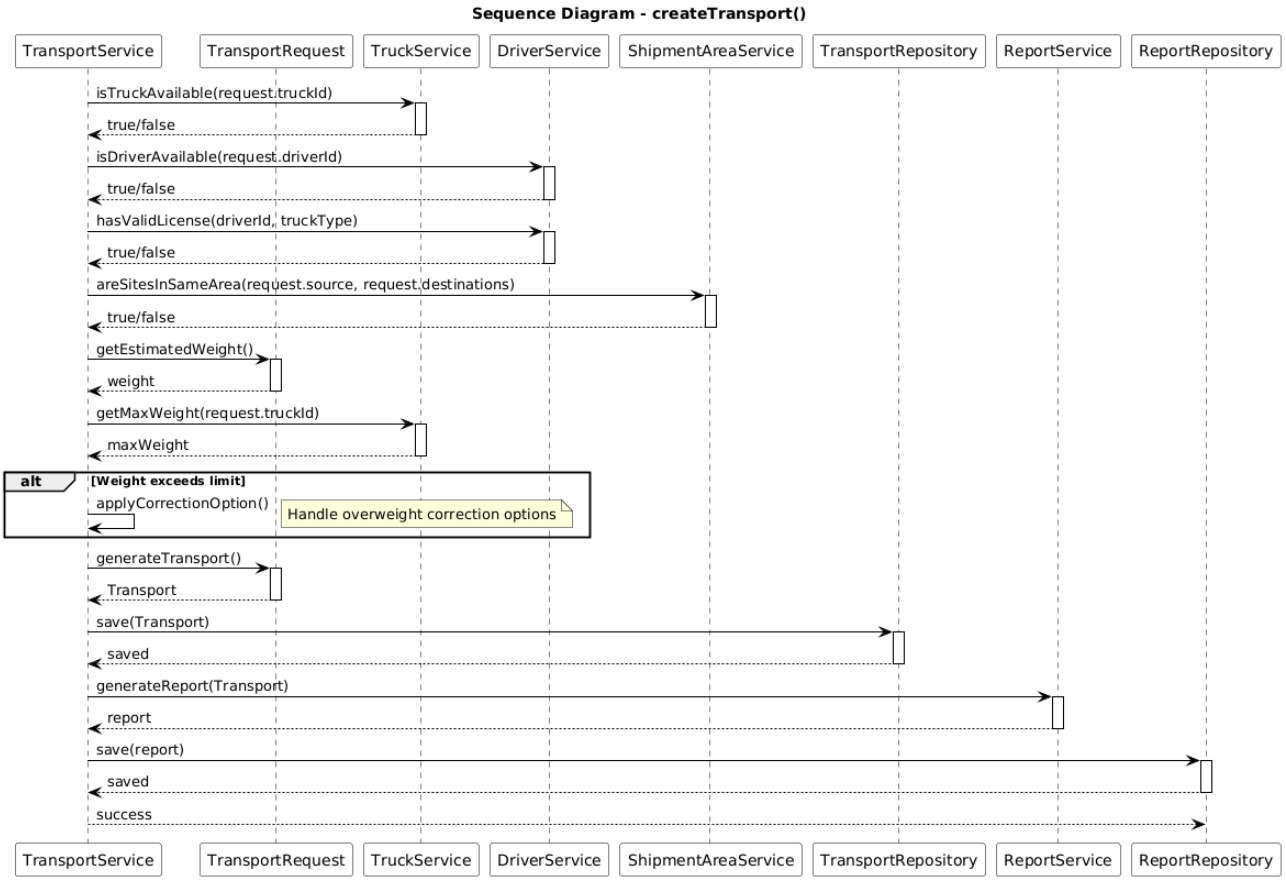
Contract 1 – assignEmployeeToShift:



Contract 2 – addEmployeeToShift:



Contract CO3 – createTransport:



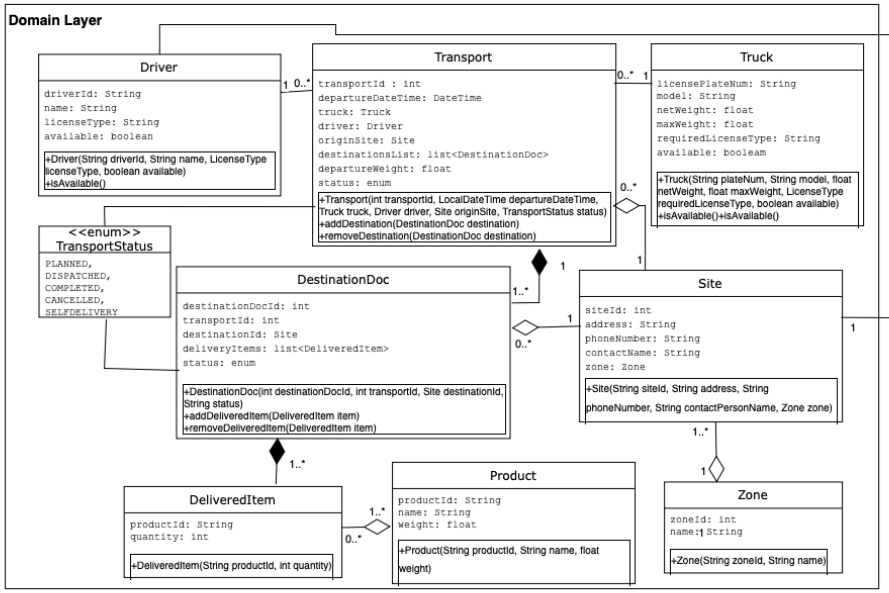
3.1. Class Diagram

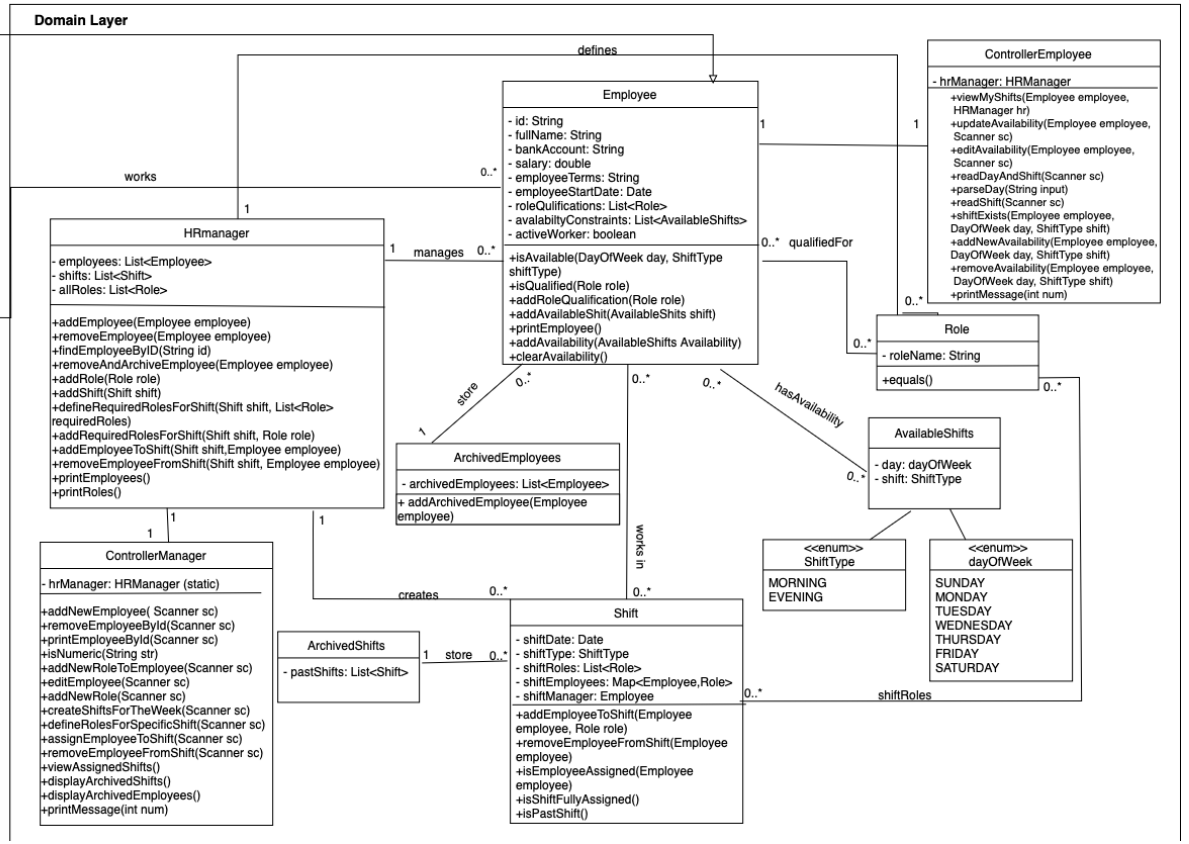
What was added?

- Methods – as required in the assignment.
- Inheritance connection between Driver and Employee, since a driver is an employee with an additional field of license type.
- A one-to-many relationship from Site to Employee, since an employee can work at only one site, while a site can be associated with multiple employees or none at all.

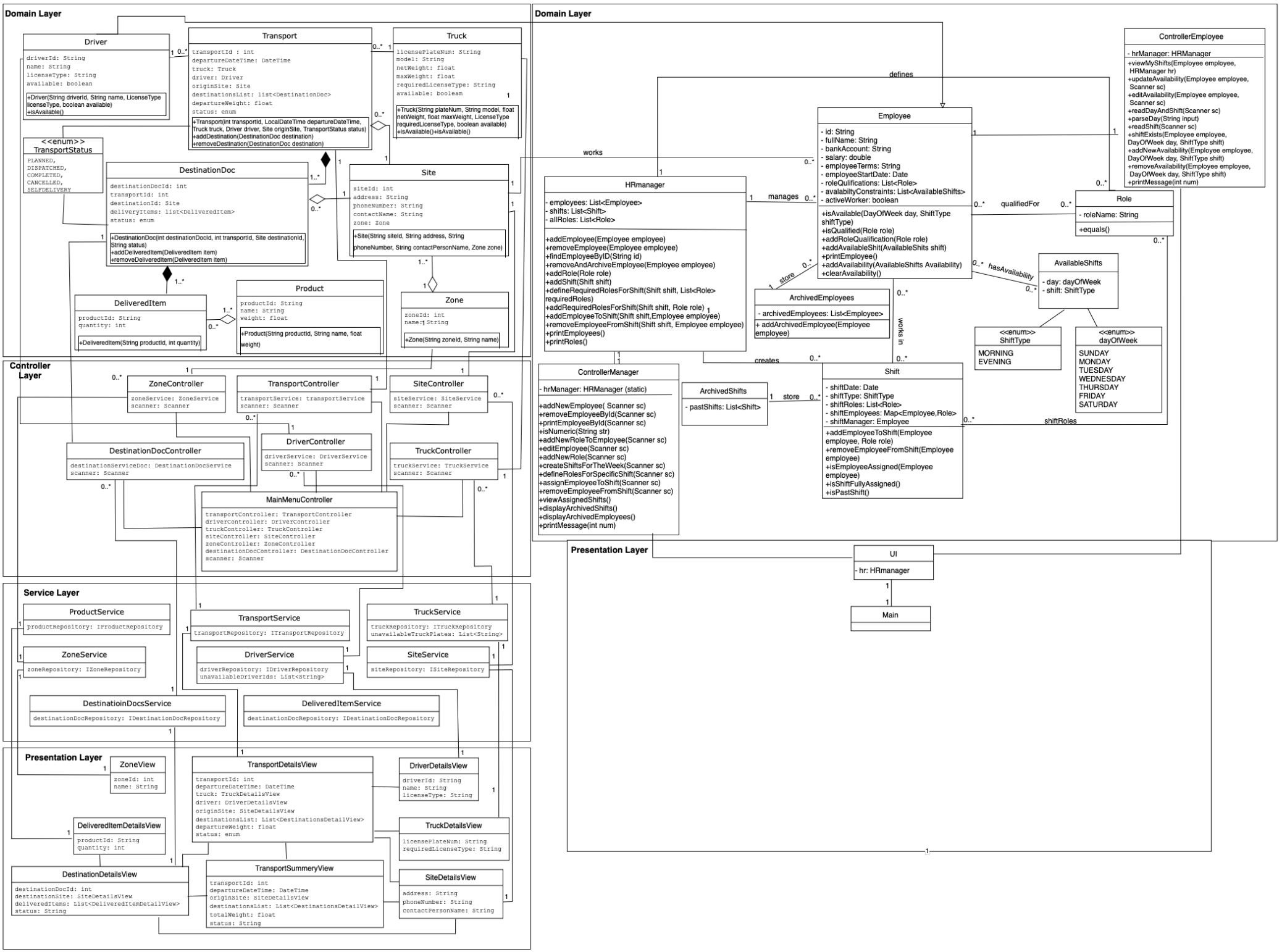
Since the full diagram is large and the details are difficult to read, we will first present only the domain layer classes that include the new connections. The complete diagram will be shown on the following page.

The Partly Diagram (Only Domains of both modules):





The full class diagram is in the next page:


Domain Layer
Driver
driverId: String
name: String
licenseType: String
available: boolean
+Driver(String driverId, String name, LicenseType licenseType, boolean available)
+isAvailable()
Transport
transportId : int
departureDateTime: DateTime
truck: Truck
driver: Driver
originSite: Site
destinationsList: list<DestinationDoc>
departureWeight: float
status: enum
+Transport(int transportId, LocalDateTime departureDateTime, Truck truck, Driver driver, Site originSite, TransportStatus status)
+addDestination(DestinationDoc destination)
+removeDestination(DestinationDoc destination)
Truck
licensePlateNum: String
model: String
netWeight: float
maxWeight: float
requiredLicenseType: String
available: booleam
+Truck(String plateNum, String model, float netWeight, float maxWeight, LicenseType requiredLicenseType, boolean available)
+isAvailable()+isAvailable()
<<enum>>
TransportStatus
PLANNED,
DISPATCHED,
COMPLETED,
CANCELLED,
SELFDELIVERY
DestinationDoc
destinationDocId: int
transportId: int
destinationId: Site
deliveryItems: list<DeliveredItem>
status: enum
+DestinationDoc(int destinationDocId, int transportId, Site destinationId, String status)
+addDeliveredItem(DeliveredItem item)
+removeDeliveredItem(DeliveredItem item)
Site
siteId: int
address: String
phoneNumber: String
contactName: String
zone: Zone
+Site(String siteId, String address, String phoneNumber, String contactPersonName, Zone zone)
DeliveredItem
productId: String
quantity: int
+DeliveredItem(String productId, int quantity)
Product
productId: String
name: String
weight: float
+Product(String productId, String name, float weight)
Zone
zoneId: int
name: String
+Zone(String zoneId, String name)
Controller Layer
ZoneController
zoneService: ZoneService
scanner: Scanner
TransportController
transportService: transportService
scanner: Scanner
SiteController
siteService: SiteService
scanner: Scanner
DriverController
driverService: DriverService
scanner: Scanner
DestinationDocController
destinationServiceDoc: DestinationDocService
scanner: Scanner
TruckController
truckService: TruckService
scanner: Scanner
MainMenuController
transportController: TransportController
driverController: DriverController
truckController: TruckController
siteController: SiteController
zoneController: ZoneController
destinationDocController: DestinationDocController
scanner: Scanner
Service Layer
ProductService
productRepository: IProductRepository
TransportService
transportRepository: ITransportRepository
TruckService
truckRepository: ITruckRepository
unavailableTruckPlates: List<String>
ZoneService
zoneRepository: IZoneRepository
DriverService
driverRepository: IDriverRepository
unavailableDriverIds: List<String>
SiteService
siteRepository: ISiteRepository
DestinatioinDocsService
destinationDocRepository: IDestinationDocRepository
DeliveredItemService
destinationDocRepository: IDestinationDocRepository
Presentation Layer
ZoneView
zoneId: int
name: String
TransportDetailsView
transportId: int
departureDateTime: DateTime
truck: TruckDetailsView
driver: DriverDetailsView
originSite: SiteDetailsView
destinationsList: List<DestinationsDetailView>
departureWeight: float
status: enum
DriverDetailsView
driverId: String
name: String
licenseType: String
DeliveredItemDetailsView
productId: String
quantity: int
TruckDetailsView
licensePlateNum: String
requiredLicenseType: String
DestinationDetailsView
destinationDocId: int
destinationSite: SiteDetailsView
deliveredItems: List<DeliveredItemDetailView>
status: String
TransportSummeryView
transportId: int
departureDateTime: DateTime
originSite: SiteDetailsView
destinationsList: List<DestinationsDetailView>
totalWeight: float
status: String
SiteDetailsView
address: String
phoneNumber: String
contactPersonName: String
Domain Layer
defines
works
manages
qualifiedFor
hasAvailability
store
creates
works in
Employee
- id: String
- fullName: String
- bankAccount: String
- salary: double
- employeeTerms: String
- employeeStartDate: Date
- roleQulifications: List<Role>
- avalabiltyConstraints: List<AvailableShifts>
- activeWorker: boolean
+isAvailable(DayOfWeek day, ShiftType shiftType)
+isQualified(Role role)
+addRoleQualification(Role role)
+addAvailableShit(AvailableShits shift)
+printEmployee()
+addAvailability(AvailableShifts Availability)
+clearAvailability()
HRmanager
- employees: List<Employee>
- shifts: List<Shift>
- allRoles: List<Role>
+addEmployee(Employee employee)
+removeEmployee(Employee employee)
+findEmployeeByID(String id)
+removeAndArchiveEmployee(Employee employee)
+addRole(Role role)
+addShift(Shift shift)
+defineRequiredRolesForShift(Shift shift, List<Role> requiredRoles)
+addRequiredRolesForShift(Shift shift, Role role)
+addEmployeeToShift(Shift shift,Employee employee)
+removeEmployeeFromShift(Shift shift, Employee employee)
+printEmployees()
+printRoles()
ControllerEmployee
- hrManager: HRManager
+viewMyShifts(Employee employee, HRManager hr)
+updateAvailability(Employee employee, Scanner sc)
+editAvailability(Employee employee, Scanner sc)
+readDayAndShift(Scanner sc)
+parseDay(String input)
+readShift(Scanner sc)
+shiftExists(Employee employee, DayOfWeek day, ShiftType shift)
+addNewAvailability(Employee employee, DayOfWeek day, ShiftType shift)
+removeAvailability(Employee employee, DayOfWeek day, ShiftType shift)
+printMessage(int num)
Role
- roleName: String
+equals()
AvailableShifts
- day: dayOfWeek
- shift: ShiftType
ArchivedEmployees
- archivedEmployees: List<Employee>
+ addArchivedEmployee(Employee employee)
<<enum>>
ShiftType
MORNING
EVENING
<<enum>>
dayOfWeek
SUNDAY
MONDAY
TUESDAY
WEDNESDAY
THURSDAY
FRIDAY
SATURDAY
ControllerManager
- hrManager: HRManager (static)
+addNewEmployee(Scanner sc)
+removeEmployeeById(Scanner sc)
+printEmployeeById(Scanner sc)
+isNumeric(String str)
+addNewRoleToEmployee(Scanner sc)
+editEmployee(Scanner sc)
+addNewRole(Scanner sc)
+createShiftsForTheWeek(Scanner sc)
+defineRolesForSpecificShift(Scanner sc)
+assignEmployeeToShift(Scanner sc)
+removeEmployeeFromShift(Scanner sc)
+viewAssignedShifts()
+displayArchivedShifts()
+displayArchivedEmployees()
+printMessage(int num)
ArchivedShifts
- pastShifts: List<Shift>
Shift
- shiftDate: Date
- shiftType: ShiftType
- shiftRoles: List<Role>
- shiftEmployees: Map<Employee,Role>
- shiftManager: Employee
+addEmployeeToShift(Employee employee, Role role)
+removeEmployeeFromShift(Employee employee)
+isEmployeeAssigned(Employee employee)
+isShiftFullyAssigned()
+isPastShift()
shiftRoles
Presentation Layer
UI
- hr: HRmanager
Main

3.2. Requirements Document:

ID	Module	Functional / Non-Functional	Description	Priority	Risk	Status
1	HR	Functional	The system MUST support registration of new employees.	MH	Low	done
2	HR	Functional	The system MUST manage for each employee the following details: full name, ID number, bank account, salary, employment terms, and employment start date.	MH	Low	done
3	HR	Functional	The system MUST allow editing and deleting employee records.	MH	Low	done
4	HR	Functional	The system MUST support associating employees with available shifts (morning/evening) based on their availability and qualifications.	MH	Low	done
5	HR	Functional	The system MUST support defining the roles that each employee is qualified to perform.	MH	Low	done
6	HR	Functional	The system MUST allow the HR manager to view a list of employees and filter them by availability or roles.	MH	Low	done
7	HR	Functional	The system MUST maintain a history of past shift assignments.	MH	Low	done
8	HR	Functional	The system MUST support two types of shifts: morning and evening.	MH	Low	done
9	HR	Functional	The system MUST support appointing one certified shift manager per shift.	MH	Low	done
10	HR	Functional	The system MUST allow the HR manager to define which roles are required for each shift.	MH	Low	done
11	HR	Functional	The system MUST allow searching employees by ID.	MH	Low	done
12	HR	Functional	The system MUST support the creation of morning or evening shifts only.	MH	Low	done
13	HR	Functional	The system MUST allow users to log in using their roles.	MH	Low	done
14	HR	Functional	The system will allow filtering drivers by availability in the shift.	MH	Low	
15	HR + Deliveries	Functional	The system MUST prevent the creation of a delivery if there is no available and qualified driver assigned to the relevant shift, or no warehouseman is scheduled to receive it.	MH	Low	
16	HR + Deliveries	Functional	The system will limit the amount of deliveries according to time left for the driver shift.	MH	Low	
17	HR + Deliveries	Functional	The system will check if there's a warehouse-man in all the destinations for the transportation.	MH	Low	
18	HR + Deliveries	Functional	The system MUST support appointing one certified delivery manager per shift.	MH	Low	
19	HR + Deliveries	Functional	The system MUST ensure that the assigned driver is available and scheduled for the same shift as the delivery.	MH	Low	
20	HR + Deliveries	Functional	The system MUST verify that a warehouseman is scheduled for every shift that includes a delivery.	MH	Low	
21	HR + Deliveries	Functional	The system MUST allow defining the working hours for morning and evening shifts.	MH	Low	
22	HR + Deliveries	Functional	The system MUST support storing the site ID (workplace) for each employee.	MH	Low	
23	HR + Deliveries	Functional	The system MUST allow the HR Manager to create shifts for specific sites.	MH	Low	

24	HR + Deliveries	Functional	The system MUST allow the HR Manager to assign employees to shifts per site.	MH	Low	
25	Deliveries	Functional	The system will allow you to enter truck data including license plate number, model, net weight and maximum weight.	MH	Low	done
26	Deliveries	Functional	The system will allow you to enter site data including address, phone number and contact name.	MH	Low	done
27	Deliveries	Functional	The system will allow creating a delivery that includes date, truck departure time, truck number, driver name, source, and destinations.	MH	Low	done
28	Deliveries	Functional	The system will generate a numbered document for each delivery destination.	MH	Low	done
29	Deliveries	Functional	The system will store the document number in the system for each destination, to enable tracking and future reference.	MH	Low	done
30	Deliveries	Functional	The system will allow entering and saving the actual truck weight (measured manually before departure) in the delivery form.	MH	Low	done
31	Deliveries	Functional	The system will alert the user if the weight of the truck exceeded the maximum weight.	MH	Low	done
32	Deliveries	Functional	The system will allow indicating whether the supplier delivers goods independently (i.e., without requiring a delivery from the company).	MH	Low	done
33	Deliveries	Functional	The system will allow removing a destination from a delivery in case of overweight.	MH	Low	done
34	Deliveries	Functional	The system will allow replacing a destination in case of overweight.	MH	Low	done
35	Deliveries	Functional	The system will allow changing the assigned truck in case of overweight.	MH	Low	done
36	Deliveries	Functional	The system will allow removing items from the delivery in case of overweight.	MH	Low	done
37	Deliveries	Functional	The system will update the delivery document if destinations are removed or changed.	MH	Low	done
38	Deliveries	Functional	The system will update the delivery document if the assigned truck is changed.	MH	Low	done
39	Deliveries	Functional	The system will update the delivery document if items are removed from the delivery.	MH	Low	done
40	Deliveries	Functional	The system will prevent assigning a driver to a delivery if the driver's license type does not match the required license type for the selected truck.	MH	Low	done
41	Deliveries	Functional	The system will allow defining different delivery zones.	MH	Low	done
42	Deliveries	Functional	The system will list the items delivered for each numbered document.	MH	Low	done
43	Deliveries	Functional	The system MUST support adding an estimated arrival time for each delivery.	MH	Low	
44	HR	Functional	The system MUST enforce assignment rules: only qualified employees can be assigned to specific roles.	MH	Medium	done
45	HR	Functional	The system MUST allow shift creation and assignment of employees to shifts by the HR manager.	MH	High	done
46	Deliveries	Functional	The system will support assigning multiple transports per driver per day.	MH	High	done
54	Deliveries	Non-Functional	The system will maintain a delivery management database that record every delivery.	MH	High	done

55	Deliveries	Non-Functional	The system will optimize the distribution of goods.	MH	High	
47	HR	Functional	The system SHOULD notify the HR manager if a shift has missing roles or insufficient staff.	NTH	Low	done
48	HR	Functional	The system SHOULD allow the HR manager to define and manage job roles in the system (add, delete, update).	NTH	Low	done
49	HR	Non-Functional	The system SHOULD restrict access to sensitive data to authorized users only.	NTH	Low	done
50	HR + Deliveries	Functional	The system SHOULD notify the delivery manager if no suitable driver is available for a new delivery.	NTH	Low	done
51	HR + Deliveries	Functional	The system SHOULD suggest available drivers for a delivery based on license and availability.	NTH	Low	done
52	Deliveries	Functional	The system will display a visual schedule of deliveries by day and time.	NTH	Low	done
53	Deliveries	Functional	The system will allow filtering deliveries by delivery zone.	NTH	Low	done
56	Deliveries	Non-Functional	The system will allow filtering track by the weight they can carry.	NTH	Low	done
57	Deliveries	Non-Functional	The system will allow filtering tracks by the type of driver license needed.	NTH	Low	done
58	Deliveries	Non-Functional	The system will calculate the weight of a delivery by destination.	NTH	Low	done
59	Deliveries	Non-Functional	The system will allow filtering deliveries by weight.	NTH	Low	done
60	Deliveries	Non-Functional	The system will allow filtering drivers by type of driver license.	NTH	Low	done
61	Deliveries	Non-Functional	The system will allow filtering deliveries by destination.	NTH	Low	done
62	Deliveries	Non-Functional	The system will allow filtering deliveries by status.	NTH	Low	done
63	Deliveries	Non-Functional	The system will allow filtering deliveries by driver.	NTH	Low	done
64	Deliveries	Non-Functional	The system will allow search for a delivery by delivery number.	NTH	Low	done
65	Deliveries	Functional	The system will generate separate product reports for each destination within a transport at any time.	NTH	High	done
66	Deliveries	Non-Functional	The user interface will be intuitive and easy to use for non-technical users.	NTH	High	
67	Deliveries	Non-Functional	The system will suggest an optimal delivery route based on destination locations.	NTH	High	

Explanation of the Changes:

The changes seen in the tables were made after considering the interface between the models and reviewing the forum.

Definition of Terms:

Term	Translation
Transport	A transport is carried out by a Delivery Manager. Includes: transport ID, planned departure time, truck, driver, origin site, destination list, departure weight and status.
Transport Summery	A summary of transport details: transport ID, planned departure date, origin site, destination list, truck weight, status.
Destination Doc	Delivery document for one destination. Contains: docID, transport ID, destination ID, product list, status.
Site	A site can be a pickup or delivery point. Fields: site ID, address, phone number, contact person, delivery zone.
Zone	A delivery zone groups destinations to optimize transport times. Includes: zone ID and name.
Product	A product stored at sites for delivery. Includes: product ID, name, weight of product.
Delivered Item	A product within a shipment. Includes product ID and quantity.
Truck	A truck in the system used for deliveries. Fields: ID, model, net weight, max weight, required license type, availability.
Driver	The driver is a system employee. Has employee details and license type.
Warehouseman	Warehouseman is a system employee. Handles trucks during loading and unloading of shipments.
Delivery Manager	Delivery Manager plans transport and makes decisions when issues arise (e.g., overweight, out of zone, no licensed driver, etc.).
Employee	A person who works for the organization and is registered in the system with personal and employment details: full name, ID number, bank account, salary, employment terms, and employment start date.
Shift	A defined working period within a day, divided into morning and evening segments.
Shift Type	A classification that defines whether the shift occurs in the morning or evening.
Role	A job function assigned to employees, representing the tasks they are qualified to perform.
Shift Manager	A required role in every shift, responsible for supervising other employees and performing managerial tasks.
Shift Assignment	The process of assigning employees to specific roles in scheduled shifts, based on availability and qualifications.
HR Manager	A system user responsible for managing employees, shift planning, and role assignments.
Archive Shifts	The system's record of past shifts and assignments, used for reference and tracking.
Archive Employees	The system's record of past employees and assignments, used for reference and tracking.

System Assumptions:

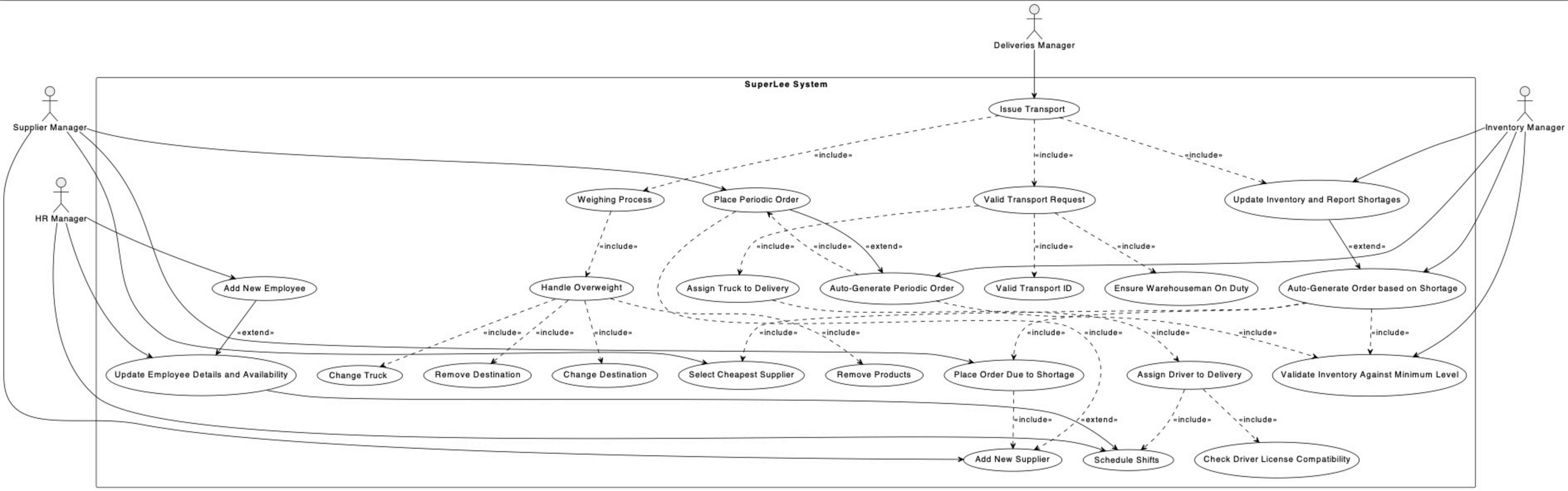
Topic	Issue	Answer
Data Archiving Policy	What is the retention period for old delivery data and how is it archived?	We'll keep deliveries up to 2 years back.
Delivery Delay Handling	How should the system handle unexpected delays during transport?	The deliveries manager decide what to do.
Delivery Manager	Is the deliveries manager is for all the deliveries or is it for a single delivery?	The deliveries manager is for all the deliveries.
Driver Scheduling	Can a driver perform more than one delivery per day and how is it managed?	Yes, the deliveries manager will decide and manage.
Formatting	What is the format of the phone number? (05.../+972...)	We'll inmplement as it's a +972 number
Mobile Access To System	Will the system be accessible via mobile devices for drivers or managers?	For now we'll implement it only as an app on the computer.
Real-Time Optimization	Should the system dynamically re-optimize the schedule during execution?	Only the deliveries manager can change it.
Weighing Sequence Integration	Does the system support weighting the truck before or after each site visit?	The truck is weighted after we have loaded it with goods and before it leaves for its destination(s).
Zone Definition Criteria	What logic determines which destinations belong to the same delivery zone?	The deliveries manager decide that.
Zone Deletetion	What happens when we delete zones?	The system will pop a message that will let the manager decide weather to change the zone for those areas or delete those stores.
Shifts	Whats the shifts hours?	The morning shift will be 07:00-15:00 and the evening shift is 15:00-23:00.
Delivery Manager	How many in a shift?	There will be 1 delivery manager in a shift and if there isn't one in the shift, the system will alert about it.
Workers in Sites	A worker belongs to 1 site?	Yes.
Multiple Roles	Can an employee hold multiple roles?	The client responded: Yes.
Multiple Roles per shift	Can an employee be assigned to more than one role within the same shift?	The client responded: No.
Weekly Shift Limits	Is there a limit to how many shifts an employee can work per week?	No.
Shift Modification Authority	Does the HR manager have the authority to remove employees from an assigned shift?	The client responded: Yes.
HR Manager connect	Who can connect to the system as HR manager?	Only those with an approved password.
Employee Availability Constraints	Are each employee's shift constraints constant, or can they be changed over time?	The client responded: No. Constraints are resubmitted weekly by each employee on Thursdays.
Employee Data Editing	Would you like the system to support editing existing employees?	Yes.
Employee Data Management	Would you like the system to support deleting employees who no longer work for the company?	The client responded: Yes.
Employee Deletion Handling	After deleting an employee, should the system retain the record in a separate archive?	Yes. His data should be kept.

Passwords:

Passwords for system access:
System Administrator – admin
Transport Manager – transport123
Password for HR Manager access:
HRManager-1111

In the next pages there'll be the appendices:

1. Use-Case
2. Full class diagram
3. HR module first assignment
4. Deliveries module first assignment


Deliveries Manager
SuperLee System
Supplier Manager
HR Manager
Inventory Manager
Issue Transport
Weighing Process
Place Periodic Order
Valid Transport Request
Update Inventory and Report Shortages
Add New Employee
Handle Overweight
Assign Truck to Delivery
Auto-Generate Periodic Order
Valid Transport ID
Ensure Warehouseman On Duty
Auto-Generate Order based on Shortage
Update Employee Details and Availability
Change Truck
Remove Destination
Change Destination
Select Cheapest Supplier
Remove Products
Place Order Due to Shortage
Assign Driver to Delivery
Validate Inventory Against Minimum Level
Add New Supplier
Schedule Shifts
Check Driver License Compatibility
«include»
«extend»

Domain Layer

Driver
driverId: String
name: String
licenseType: String
available: boolean
+Driver(String driverId, String name, LicenseType licenseType, boolean available)
+isAvailable()

Transport
transportId : int
departureDateTime: DateTime
truck: Truck
driver: Driver
originSite: Site
destinationsList: list<DestinationDoc>
departureWeight: float
status: enum
+Transport(int transportId, LocalDateTime departureDateTime, Truck truck, Driver driver, Site originSite, TransportStatus status)
+addDestination(DestinationDoc destination)
+removeDestination(DestinationDoc destination)

Truck
licensePlateNum: String
model: String
netWeight: float
maxWeight: float
requiredLicenseType: String
available: booleam
+Truck(String plateNum, String model, float netWeight, float maxWeight, LicenseType requiredLicenseType, boolean available)
+isAvailable()+isAvailable()

<<enum>> TransportStatus
PLANNED,
DISPATCHED,
COMPLETED,
CANCELLED,
SELFDELIVERY

DestinationDoc
destinationDocId: int
transportId: int
destinationId: Site
deliveryItems: list<DeliveredItem>
status: enum
+DestinationDoc(int destinationDocId, int transportId, Site destinationId, String status)
+addDeliveredItem(DeliveredItem item)
+removeDeliveredItem(DeliveredItem item)

Site
siteId: int
address: String
phoneNumber: String
contactName: String
zone: Zone
+Site(String siteId, String address, String phoneNumber, String contactPersonName, Zone zone)

DeliveredItem
productId: String
quantity: int
+DeliveredItem(String productId, int quantity)

Product
productId: String
name: String
weight: float
+Product(String productId, String name, float weight)

Zone
zoneId: int
name: String
+Zone(String zoneId, String name)

Controller Layer

ZoneController
zoneService: ZoneService
scanner: Scanner

TransportController
transportService: transportService
scanner: Scanner

SiteController
siteService: SiteService
scanner: Scanner

DestinationDocController
destinationServiceDoc: DestinationDocService
scanner: Scanner

DriverController
driverService: DriverService
scanner: Scanner

TruckController
truckService: TruckService
scanner: Scanner

MainMenuController
transportController: TransportController
driverController: DriverController
truckController: TruckController
siteController: SiteController
zoneController: ZoneController
destinationDocController: DestinationDocController
scanner: Scanner

Service Layer

ProductService
productRepository: IProductRepository

TransportService
transportRepository: ITransportRepository

TruckService
truckRepository: ITruckRepository
unavailableTruckPlates: List<String>

ZoneService
zoneRepository: IZoneRepository

DriverService
driverRepository: IDriverRepository
unavailableDriverIds: List<String>

SiteService
siteRepository: ISiteRepository

DestinatioinDocsService
destinationDocRepository: IDestinationDocRepository

DeliveredItemService
destinationDocRepository: IDestinationDocRepository

Presentation Layer

ZoneView
zoneId: int
name: String

TransportDetailsView
transportId: int
departureDateTime: DateTime
truck: TruckDetailsView
driver: DriverDetailsView
originSite: SiteDetailsView
destinationsList: List<DestinationsDetailView>
departureWeight: float
status: enum

DriverDetailsView
driverId: String
name: String
licenseType: String

DeliveredItemDetailsView
productId: String
quantity: int

TruckDetailsView
licensePlateNum: String
requiredLicenseType: String

DestinationDetailsView
destinationDocId: int
destinationSite: SiteDetailsView
deliveredItems: List<DeliveredItemDetailView>
status: String

TransportSummeryView
transportId: int
departureDateTime: DateTime
originSite: SiteDetailsView
destinationsList: List<DestinationsDetailView>
totalWeight: float
status: String

SiteDetailsView
address: String
phoneNumber: String
contactPersonName: String

Domain Layer

ControllerEmployee
- hrManager: HRManager
+viewMyShifts(Employee employee, HRManager hr)
+updateAvailability(Employee employee, Scanner sc)
+editAvailability(Employee employee, Scanner sc)
+readDayAndShift(Scanner sc)
+parseDay(String input)
+readShift(Scanner sc)
+shiftExists(Employee employee, DayOfWeek day, ShiftType shift)
+addNewAvailability(Employee employee, DayOfWeek day, ShiftType shift)
+removeAvailability(Employee employee, DayOfWeek day, ShiftType shift)
+printMessage(int num)

defines

Employee
- id: String
- fullName: String
- bankAccount: String
- salary: double
- employeeTerms: String
- employeeStartDate: Date
- roleQulifications: List<Role>
- avalabiltyConstraints: List<AvailableShifts>
- activeWorker: boolean
+isAvailable(DayOfWeek day, ShiftType shiftType)
+isQualified(Role role)
+addRoleQualification(Role role)
+addAvailableShit(AvailableShits shift)
+printEmployee()
+addAvailability(AvailableShifts Availability)
+clearAvailability()

works

HRmanager
- employees: List<Employee>
- shifts: List<Shift>
- allRoles: List<Role>
+addEmployee(Employee employee)
+removeEmployee(Employee employee)
+findEmployeeByID(String id)
+removeAndArchiveEmployee(Employee employee)
+addRole(Role role)
+addShift(Shift shift)
+defineRequiredRolesForShift(Shift shift, List<Role> requiredRoles)
+addRequiredRolesForShift(Shift shift, Role role)
+addEmployeeToShift(Shift shift,Employee employee)
+removeEmployeeFromShift(Shift shift, Employee employee)
+printEmployees()
+printRoles()

manages

qualifiedFor

Role
- roleName: String
+equals()

store

ArchivedEmployees
- archivedEmployees: List<Employee>
+ addArchivedEmployee(Employee employee)

hasAvailability

AvailableShifts
- day: dayOfWeek
- shift: ShiftType

works in

<<enum>> ShiftType
MORNING
EVENING

<<enum>> dayOfWeek
SUNDAY
MONDAY
TUESDAY
WEDNESDAY
THURSDAY
FRIDAY
SATURDAY

creates

ControllerManager
- hrManager: HRManager (static)
+addNewEmployee(Scanner sc)
+removeEmployeeById(Scanner sc)
+printEmployeeById(Scanner sc)
+isNumeric(String str)
+addNewRoleToEmployee(Scanner sc)
+editEmployee(Scanner sc)
+addNewRole(Scanner sc)
+createShiftsForTheWeek(Scanner sc)
+defineRolesForSpecificShift(Scanner sc)
+assignEmployeeToShift(Scanner sc)
+removeEmployeeFromShift(Scanner sc)
+viewAssignedShifts()
+displayArchivedShifts()
+displayArchivedEmployees()
+printMessage(int num)

ArchivedShifts
- pastShifts: List<Shift>

store

Shift
- shiftDate: Date
- shiftType: ShiftType
- shiftRoles: List<Role>
- shiftEmployees: Map<Employee,Role>
- shiftManager: Employee
+addEmployeeToShift(Employee employee, Role role)
+removeEmployeeFromShift(Employee employee)
+isEmployeeAssigned(Employee employee)
+isShiftFullyAssigned()
+isPastShift()

shiftRoles

Presentation Layer

UI
- hr: HRmanager

Main

ID	Module	Functional/Non-Functional	Description	Priority	Risk	Status
1	HR	Functional	The system MUST support registration of new employees.	MH	Low	done
2	HR	Functional	The system MUST manage for each employee the following details: full name, ID number, bank account, salary, employment terms, and employment start date.	MH	Low	done
3	HR	Functional	The system MUST allow editing and deleting employee records.	MH	Low	done
4	HR	Functional	The system MUST support associating employees with available shifts (morning/evening) based on their availability and qualifications.	MH	Low	done
5	HR	Functional	The system MUST support defining the roles that each employee is qualified to perform.	MH	Low	done
6	HR	Functional	The system MUST allow the HR manager to view a list of employees and filter them by availability or roles.	MH	Low	done
7	HR	Functional	The system MUST allow shift creation and assignment of employees to shifts by the HR manager.	MH	High	done
8	HR	Functional	The system MUST enforce assignment rules: only qualified employees can be assigned to specific roles.	MH	Medium	done
9	HR	Functional	The system MUST maintain a history of past shift assignments.	MH	Low	done
10	HR	Functional	The system MUST support two types of shifts: morning and evening.	MH	Low	done
11	HR	Functional	The system MUST support appointing one certified shift manager per shift.	MH	Low	done
12	HR	Functional	The system MUST allow the HR manager to define which roles are required for each shift.	MH	Low	done
13	HR	Functional	The system SHOULD notify the HR manager if a shift has missing roles or insufficient staff.	NTH	Low	done
14	HR	Functional	The system SHOULD allow the HR manager to define and manage job roles in the system(add, delete, update).	NTH	Low	done
15	HR	Non-Functional	The system SHOULD restrict access to sensitive data to authorized users only	NTH	Low	done
16	HR	Functional	The system MUST allow searching employees by ID.	MH	Low	done
17	HR	Functional	The system MUST support the creation of morning or evening shifts only.	MH	Low	done

18	HR	Functional	The system MUST allow users to log in using their role.	MH	Low	done

#	Topic	Issue	Client's Response
1	Multiple Roles	Can an employee hold multiple roles?	The client responded: Yes.
2	Multiple Roles per Shift	Can an employee be assigned to more than one role within the same shift?	The client responded: No.
3	Weekly Shift Limits	Is there a limit to how many shifts an employee can work per week?	No.
4	Shift Modification Authority	Does the HR manager have the authority to remove employees from an assigned shift?	The client responded: Yes.
5	HR Manager connect	Who can connect to the system as HR manager?	Only those with an approved password.
6	Employee Availability Constraints	Are each employee's shift constraints constant, or can they be changed over time?	The client responded: No. Constraints are resubmitted weekly by each employee on Thursdays.
7	Employee Data Editing	Would you like the system to support editing existing employees?	Yes.
8	Employee Data Management	Would you like the system to support deleting employees who no longer work for the company?	The client responded: Yes.
9	Employee Deletion Handling	After deleting an employee, should the system retain the record in a separate archive?	Yes. His data should be kept.

בר פסו 315175554
הדר עופר 208583781

מטלה 1

שאלה 1 - דרישות

ID	Module	Functional / Non - Functional	Description	Priority	Risk	Status
1	Deliveries	**Functional**	The system will allow you to enter truck data including license plate number, model, net weight and maximum weight.	MH	L	In Progress...
2	Deliveries	**Functional**	The system will allow you to enter site data including address, phone number and contact name.	MH	L	In Progress...
3	Deliveries	**Functional**	The system will allow creating a delivery that includes date, truck departure time, truck number, driver name, source, and destinations.	MH	L	In Progress...
4	Deliveries	**Functional**	The system will generate a numbered document for each delivery destination.	MH	L	In Progress...
5	Deliveries	**Functional**	The system will store the document number in the system for each destination, to enable tracking and future reference.	MH	L	In Progress...
6	Deliveries	**Functional**	The system will allow entering and saving the actual truck weight (measured manually before departure) in the delivery form.	MH	L	In Progress...
7	Deliveries	**Functional**	The system will alert the user if the weight of the truck exceeded the maximum weight.	MH	L	In Progress...
8	Deliveries	**Functional**	The system will allow indicating whether the supplier delivers goods independently (i.e., without requiring a delivery from the company).	MH	L	In Progress...
9	Deliveries	**Functional**	The system will allow removing a destination from a delivery in case of overweight.	MH	L	In Progress...
10	Deliveries	**Functional**	The system will allow replacing a destination in case of overweight.	MH	L	In Progress...
11	Deliveries	**Functional**	The system will allow changing the assigned truck in case of overweight.	MH	L	In Progress...
12	Deliveries	**Functional**	The system will allow removing items from the delivery in case of overweight.	MH	L	In Progress...
13	Deliveries	**Functional**	The system will update the delivery document if destinations are removed or changed.	MH	L	In Progress...
14	Deliveries	**Functional**	The system will update the delivery document if the assigned truck is changed.	MH	L	In Progress...
15	Deliveries	**Functional**	The system will update the delivery document if items are removed from the delivery.	MH	L	In Progress...
16	Deliveries	**Functional**	The system will prevent assigning a driver to a delivery if the driver's license type does not match the required license type for the selected truck.	MH	L	In Progress...
17	Deliveries	**Functional**	The system will allow defining different delivery zones.	MH	L	In Progress...
18	Deliveries	**Functional**	The system will list the items delivered for each numbered document.	MH	L	In Progress...
19	Deliveries	**Functional**	The system will support assigning multiple transports per driver per day.	MH	H	In Progress...
20	Deliveries	Non-Functional	The system will maintain a delivery management database that record every delivery.	MH	H	Can't be done
21	Deliveries	Non-Functional	The system will optimize the distribution of goods.	MH	H	Can't be done
22	Deliveries	**Functional**	The system will display a visual schedule of deliveries by day and time.	NTH	L	Can't be done
23	Deliveries	**Functional**	The system will allow filtering deliveries by delivery zone.	NTH	L	In Progress...
24	Deliveries	Non-Functional	The system will allow filtering track by the weight they can carry.	NTH	L	In Progress...
25	Deliveries	Non-Functional	The system will allow filtering tracks by the type of driver license needed.	NTH	L	In Progress...
26	Deliveries	Non-Functional	The system will calculate the weight of a delivery by destination.	NTH	L	In Progress...
27	Deliveries	Non-Functional	The system will allow filtering deliveries by weight.	NTH	L	In Progress...
28	Deliveries	Non-Functional	The system will allow filtering drivers by type of driver license.	NTH	L	In Progress...
29	Deliveries	Non-Functional	The system will allow filtering deliveries by destination.	NTH	L	In Progress...
30	Deliveries	Non-Functional	The system will allow filtering deliveries by status.	NTH	L	In Progress...
31	Deliveries	Non-Functional	The system will allow filtering deliveries by driver.	NTH	L	In Progress...
32	Deliveries	Non-Functional	The system will allow search for a delivery by delivery number.	NTH	L	Can't be done
33	Deliveries	**Functional**	The system will generate separate product reports for each destination within a transport at any time.	NTH	H	Can't be done
34	Deliveries	Non-Functional	The system will suggest an optimal delivery route based on destination locations.	NTH	H	Can't be done
35	Deliveries	Non-Functional	The user interface will be intuitive and easy to use for non-technical users.	NTH	H	Can't be done

בר פסו 315175554
הדר עופר 208583781

סיבות למה חלק מהדרישות אינן ניתנות למימוש:

Description	Reason It May Not Be Feasible
The system will maintain a delivery management database that record every delivery.	Storing all delivery records long-term requires advanced server and storage resources. The system may need to delete old data to maintain performance.
The system will optimize the distribution of goods.	True optimization requires advanced algorithms, real-time data (traffic, truck availability...), and potentially AI – which may be too complex or costly to implement initially.
The system will display a visual schedule of deliveries by day and time.	Creating a dynamic, visual schedule requires a sophisticated UI. Budget or technical limitations may prevent this feature in early versions.
The system will allow search for a delivery by delivery number.	If delivery numbers are inconsistent or not properly logged, the search feature may yield inaccurate results or require advanced logic that's not implemented.
The system will generate separate product reports for each destination within a transport at any time.	Generating dynamic reports by destination requires a well-structured data model. If such structure isn't implemented, the reports can't be created accurately.
The system will suggest an optimal delivery route based on destination locations.	Requires integration with mapping and real-time traffic APIs, which might not be available due to technical, regulatory, or financial constraints.
The user interface will be intuitive and easy to use for non-technical users.	Creating a user-friendly interface requires UX/UI design, user testing, and research. Limited time or budget may result in a more technical interface.

שאלות שעלו לנו מדרישות הלקוח ותשובות עבורן:

Topic	Issue	Answer
Data Archiving Policy	What is the retention period for old delivery data and how is it archived?	We'll keep deliveries up to 2 years back.
Delivery Delay Handling	How should the system handle unexpected delays during transport?	The deliveries manager decide what to do.
Delivery Manager	Is the deliveries manager is for all the deliveries or is it for a single delivery?	The deliveries manager is for all the deliveries.
Driver Scheduling	Can a driver perform more than one delivery per day and how is it managed?	Yes, the deliveries manager will decide and manage.
Formatting	What is the format of the phone number? (05.../+972...)	We'll inmplement as it's a +972 number
Mobile Access To System	Will the system be accessible via mobile devices for drivers or managers?	For now we'll implement it only as an app on the computer.
Real-Time Optimization	Should the system dynamically re-optimize the schedule during execution?	Only the deliveries manager can change it.
Weighing Sequence Integration	Does the system support weighting the truck before or after each site visit?	The truck is weighted after we have loaded it with goods and before it leaves for its destination(s).
Zone Definition Criteria	What logic determines which destinations belong to the same delivery zone?	The deliveries manager decide that.
Zone Deletetion	What happens when we delete zones?	The system will pop a message that will let the manager decide weather to change the zone for those areas or delete those stores.

הנחות שעשינו לגבי המערכת:

- מנהל מערכת ואחראי הובלות הם שני תפקידים שונים, יש 1 מכל תפקיד. מנהל המערכת קובע מי הוא אחראי ההובלות.
- מנהל מערכת רשאי למנות אחראי הובלות ולצפות בכל המידע אך לא לשנות אותו.
- אחראי הובלות רשאי להוסיף/לעדכן/למחוק מידע לגבי הובלות, נהגים, משאיות, אזורי שילוח, קבלת החלטות במקרה של חריגה במשקל/מעבר ב2 אזורי שילוח באותה הובלה/אין נהג מתאים למשאית/אין משאית מתאימה למשלוח וכו'.

- אזור שילוח מיוצג ע"י שם ומספר, ביחד הם יהיו ייחודיים.
- נשקול את המשאית לאחר העמסה בלבד! בעת הגעה ליעד נניח שכל הסחורה הרלוונטית ירדה.
- הוספנו שדה "סטטוס" אל מסמך ההובלה.
- תיעוד ספקים עצמאיים – הסטטוס של הDestinationDoc יהיה SELFDELIVERY.
- נמלא מסמך הובלה עבור כל יעד בהובלה.
- ניתן לעדכן את ההובלה לאחר שיצאה כבר. נעדכן אם ההובלה התעכבה/בוטלה מכל סיבה שהיא.
- משאית ריקה אינה נחשבת להובלה.
- כשיש מוצר פגום – מראש לא נעביר אותו אל היעד ונמחק אותו מההזמנה.
- לנהגים במערכת יש סוג רשיון אחד בלבד.
- <u>סיסמאות לכניסה למערכת:</u>
  מנהל המערכת – admin
  אחראי הובלות – transport123

שאלה 2 – תרשים מחלקות

על מנת שהדיאגרמה תהיה יותר קריאה, החיבורים בצבעים –

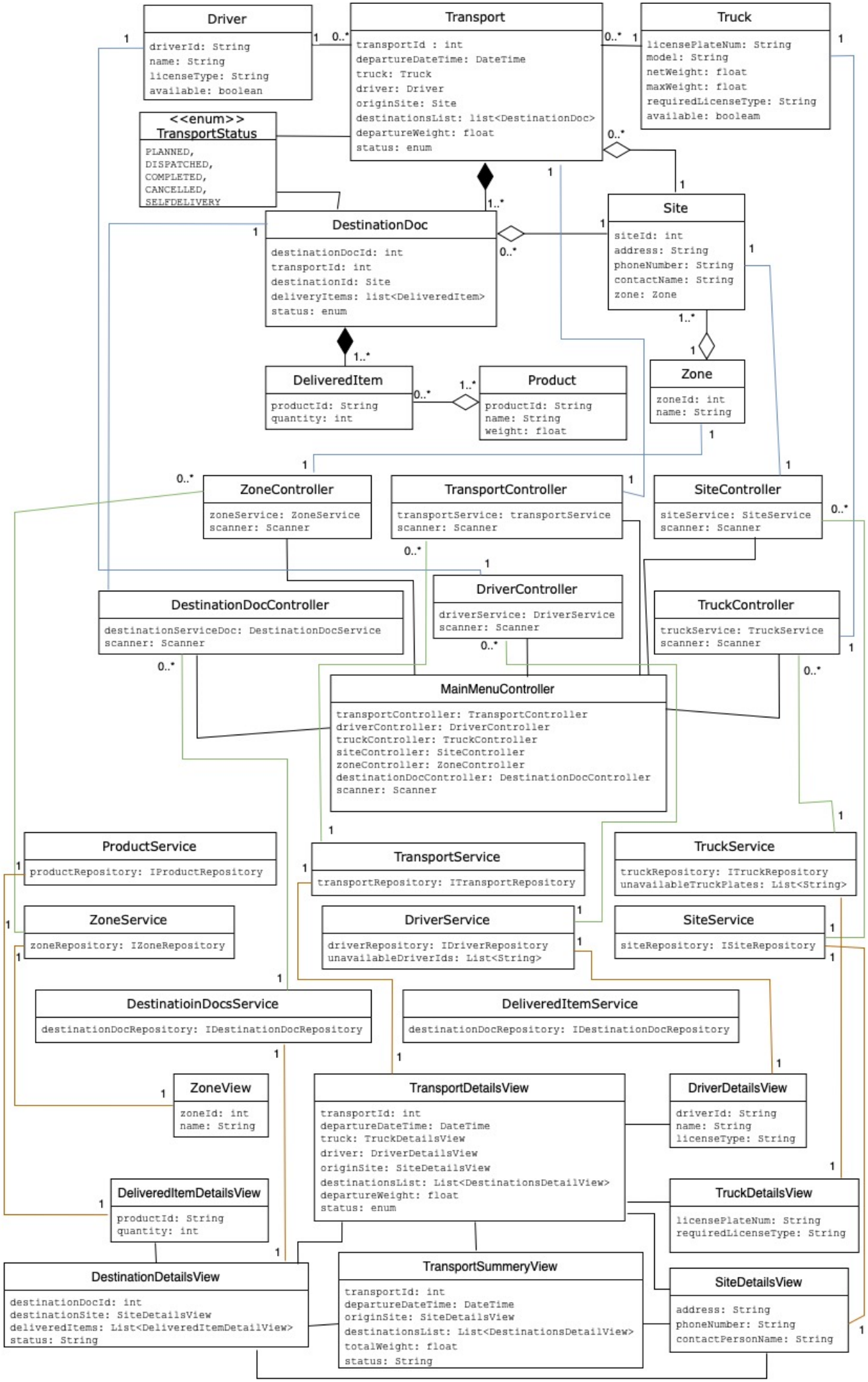
בצבע **כחול** – חיבורים בין Domain ל-Controller

בצבע **ירוק** – חיבורים בין Controller ל-Service

בצבע **סגול** – חיבורים בין Service ל-Presentation

בצבע **שחור** – חיבורים בין שכבה לעצמה (Domain ל-Domain וכו')

הדיאגרמה נמצאת בעמוד הבא.


Driver
driverId: String
name: String
licenseType: String
available: boolean
Transport
transportId : int
departureDateTime: DateTime
truck: Truck
driver: Driver
originSite: Site
destinationsList: list<DestinationDoc>
departureWeight: float
status: enum
Truck
licensePlateNum: String
model: String
netWeight: float
maxWeight: float
requiredLicenseType: String
available: booleam
<<enum>>
TransportStatus
PLANNED,
DISPATCHED,
COMPLETED,
CANCELLED,
SELFDELIVERY
DestinationDoc
destinationDocId: int
transportId: int
destinationId: Site
deliveryItems: list<DeliveredItem>
status: enum
Site
siteId: int
address: String
phoneNumber: String
contactName: String
zone: Zone
DeliveredItem
productId: String
quantity: int
Product
productId: String
name: String
weight: float
Zone
zoneId: int
name: String
ZoneController
zoneService: ZoneService
scanner: Scanner
TransportController
transportService: transportService
scanner: Scanner
SiteController
siteService: SiteService
scanner: Scanner
DriverController
driverService: DriverService
scanner: Scanner
DestinationDocController
destinationServiceDoc: DestinationDocService
scanner: Scanner
TruckController
truckService: TruckService
scanner: Scanner
MainMenuController
transportController: TransportController
driverController: DriverController
truckController: TruckController
siteController: SiteController
zoneController: ZoneController
destinationDocController: DestinationDocController
scanner: Scanner
ProductService
productRepository: IProductRepository
TransportService
transportRepository: ITransportRepository
TruckService
truckRepository: ITruckRepository
unavailableTruckPlates: List<String>
ZoneService
zoneRepository: IZoneRepository
DriverService
driverRepository: IDriverRepository
unavailableDriverIds: List<String>
SiteService
siteRepository: ISiteRepository
DestinatioinDocsService
destinationDocRepository: IDestinationDocRepository
DeliveredItemService
destinationDocRepository: IDestinationDocRepository
ZoneView
zoneId: int
name: String
TransportDetailsView
transportId: int
departureDateTime: DateTime
truck: TruckDetailsView
driver: DriverDetailsView
originSite: SiteDetailsView
destinationsList: List<DestinationsDetailView>
departureWeight: float
status: enum
DriverDetailsView
driverId: String
name: String
licenseType: String
DeliveredItemDetailsView
productId: String
quantity: int
TruckDetailsView
licensePlateNum: String
requiredLicenseType: String
DestinationDetailsView
destinationDocId: int
destinationSite: SiteDetailsView
deliveredItems: List<DeliveredItemDetailView>
status: String
TransportSummeryView
transportId: int
departureDateTime: DateTime
originSite: SiteDetailsView
destinationsList: List<DestinationsDetailView>
totalWeight: float
status: String
SiteDetailsView
address: String
phoneNumber: String
contactPersonName: String