נספח – פריטי עבודה 2

שינויים שבוצעו

קובץ דרישות:

מודול עובדים:

- הוספת דרישות לניהול הרשאות ופירוטם בהתאם לתחומי האחריות שנקבעו

מודול הובלות:

- הוספת דרישה להעברת הודעות למנהל כוח אדם לגבי הובלות קרובות

מודול מלאי:

- הוספת תחומי אחריות למחסנאי

כללי:

- הוספת דרישה לתמיכה מרובת משתמשים ממקומות פיסיים שונים

תרשים מחלקות:

מודול עובדים:

- הסרת Constraint כתוצאה מהחלטת עיצוב למזג את האחריות למשמרת. מוריד Coupling

מודול ספקים:

- שינוי הארגומנטים שמקבלת השיטה orderItem כתוצאה מהאינטגרציה עם מודול הובלות
- הוספה לOrder שדה status כתוצאה מאינטגרציה עם מודול הובלות
- הורדת שדה מAgreementItem, עכשיו נלקח דרך אינטגרציה עם מודול מלאי

מודול הובלות:

- הסרת מחלקות כתוצאה מהאינטגרציה עם מודול מלאי, שם המידע מנוהל בצורה מדוייקת יותר

מודול מלאי:

- שינוי הארגומנטים שמקבלת השיטה orderArrived כתוצאה מדרישת ההרשאות שנוספה למחסנאי

תרשימי Seq/Col:

מודול עובדים:

- הוספת תרשים Login והוספתו לתהליכים שדורשים התחברות תחילה

מודול מלאי:

- שינוי תרשים של orderArrived

תרשים ERD:

- מיזוג התרשימים כתוצאה ממיזוג הרביעיות

אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

הפקולטה למדעי ההנדסה
המחלקה להנדסת מערכות מידע

Faculty of Engineering Sciences
Dept. of Information Systems Engineering

Ben-Gurion University of the Negev

“Superly”

Requirements specification

1. Functional Requirements

ID	Module	Description	Priority	Risk	Status	Feasable
1.	HR	The system **MUST** support management of employees only by the HR manager. Creation: Employee can be 1 of any employee type Updatable information: Name, Salary, Bank Details, Certifications.	MH	Low	Done	Yes
2.	HR	The system **CANNOT** allow creation of 2 employees with the same ID.	MH	Low	Done	Yes
3.	HR	The system **MUST** support management of licenses for carriers only by HR Manager.	MH	Low	Done	Yes
4.	HR	The system **MUST** store employee information on a DB	MH	Low	Done	Yes
5.	HR	The system **MUST** support management of shifts only by HR Manager. Updatable information: shift manager, counts, assigned employee IDs .	MH	High	Done	Yes
6.	HR	The system **CANNOT** allow creation of shifts with no shift manager	MH	Low	Done	Yes
~~7.~~	~~HR~~	~~The system **SHOULD** be allow setting of how many workers of each type each shift needs~~	~~NTH~~	~~Low~~	~~Done~~	~~Yes~~
8.	HR	The system **CANNOT** allow creation of shifts with no carriers, cashiers, storekeepers or sorters	MH	High	In Progress	Yes
9.	HR	The system **MUST** show which employees are available for each shift when scheduling	MH	High	Done	Yes
10.	HR	The system **SHOULD** show how many shifts each available employee has done the past month for each shift type	NTH	High	Done	Yes
11.	HR	The system **MUST** store shift history in DB	MH	Low	Done	Yes
12.	HR	The system **SHOULD** notify about upcoming incomplete shifts	NTH	Low	In Progress	Yes
13.	HR	The system **Must** require from the user to Login when it opens.	MH	Low	In Progress	Yes
14.	Suppliers	The system **MUST** support 3 kinds of suppliers (routine, by order, not transporting)	MH	Low	Done	Yes
15.	Suppliers	The system **MUST** save the supplier's information (id, bank account, paying agreement, contacts and contact's information)	MH	Low	Done	Yes
16.	Suppliers	The system **MUST** support management of supplier cards only by storekeeper and store manager.	MH	Low	Done	Yes
17.	Suppliers	The system **MUST** save if the supplier has fixed supplying days and when they are.	MH	Low	Done	Yes
18.	Suppliers	The system **MUST** save details about the supplied items (cost per item, cost when buying in bulk)	MH	Low	Done	Yes

19.	Suppliers	The system **MUST** show what items are purchased from each supplier	MH	Low	Done	Yes
20.	Suppliers	The system **MUST** give the ~~corporate management~~ storekeeper and the store manager the ability to edit the agreement's details (supplying days, prices, changing items)	NTH	Low	Done	Yes
21.	Inventory	The system **MUST** store the following product info: product ID, item name, category, price, weight, and manufacturer	MH	Low	Done	Yes
22.	Inventory	The system **MUST** store the following information about stock per each product: product ID, amount in each storefront, amount in each store warehouse, and each store's min amount	MH	Low	Done	Yes
23.	Inventory	The system **MUST** store the following information about sales per each product and category: active sales, passed sales, future sales, percent off in sale	MH	Low	Done	Yes
24.	Inventory	The system **MUST** send an alert to the storekeeper when product is getting under the minimum amount in a certain branch. specifying product ID and store ID	MH	Low	Done	Yes
25.	Inventory	The system **MUST** allow only the storekeeper to change the minimum and maximum value of each product in a branch.	MH	Low	Done	Yes
26.	Inventory	The system **MUST** be able to produce a report of all products in all branches that are below their minimum amount. The report should include the following: store ID, product ID, product name, product current amount in store, product current amount in warehouse, product current amount in the branch (store+warehouse), product minimum amount in the branch, product maximum amount in the branch	MH	Low	Done	Yes
27.	Inventory	The system **MUST** track amounts of product left in each location (each store and each warehouse), including: 1) buying product 2) removing damaged/expired products 3) moving products from the warehouse to the store 4) product has been returned to the store from a customer 5) product has been arrived from supplier.	MH	Low	Done	Yes
28.	Inventory	The system **MUST** be able to produce a report of all products in a chosen branch based on categories. The report should include the following: store ID, **product ID**, product name, category, product current amount in store, product current amount in warehouse, product current amount in the branch (store+warehouse), product minimum amount in the branch, product maximum amount in the branch.	MH	Low	Done	Yes
29.	Inventory	The system **SHOULD** be able to produce the history of the completed purchases from the suppliers, for a certain product. The history should include the following: StoreID, Supplier, Date, Amount, before discount price, after discount price, description.	NTH	Low	Done	Yes
30.	Inventory	The system **MUST** be able to produce the history of the completed purchases from the suppliers, for a certain product that was bought with a discount. The history should include the following: StoreID, Supplier, Date, Amount, before discount price, after discount price, description.	MH	Low	Done	Yes
31.	Inventory	The system **MUST** be able to create/cancel sales that apply to certain products and/or categories to specified dates. Sales	MH	Low	Done	Yes

		include: sale ID, percentage, start date, end date, Products IDs, categories IDs.				
32.	Inventory	The system **MUST** be able to produce the history of the sales to customers by product or by category. The history should include the following: sale ID, percentage, start date, end date, Products IDs, categories IDs.	MH	Low	Done	Yes
33.	Inventory	The system **MUST** be able to manage category (create, read, update, delete) only by storekeeper: update: change parent category, change name, change products in category	MH	Low	Done	Yes
34.	Inventory	The system **MUST** be able to manage product (create, read, update, delete) only by storekeeper: update: change category, change name, change price, add to a certain store, remove from a certain store, add new supplier to product, remove supplier from product	MH	Low	Done	Yes
35.	Inventory	The system **MUST** allow employees to report about damaged/expired products. The user will insert the following information: store ID, product ID, amount, user ID, description.	MH	Low	Done	Yes
36.	Inventory	The system **MUST** be able to supply a report of damaged/expired products needs to be able to be sorted by store, product, category.	MH	Low	Done	Yes
37.	Inventory	The system **MUST** be able to produce a report of damaged products/expired products/both products which were reported on certain range of dates. The report must include the following information: product ID, store ID, amount, user ID, description, date, expired/damaged.	MH	Low	Done	Yes
38.	Inventory	The system **MUST** be able to add and to remove branches to and from the chain.	MH	Low	Done	Yes
39.	Suppliers & Inventory	The system **MUST** support automatic orders, initiated whenever stock running low.	MH	Low	Done	Yes
40.	Suppliers & Inventory	The system **MUST** support automatic orders in fixed times (routine).	MH	Low	Done	Yes
41.	Suppliers & Inventory	The system **MUST** support that orders cannot be edited in the last 24 hours before the order's planned deliveries.	MH	Low	Done	Yes
42.	Suppliers & Inventory	The system **MUST** support choosing the best supplier for a specific order, choosing the cheapest one.	MH	Low	Done	Yes
43.	Suppliers & Inventory	The system **MUST** store the following order information: order id, store id, Supplier id, arrival and creation time, status, Products ids, Products amounts, original price, discount and final price.	MH	Low	Done	Yes
44.	HR & Transport	The system **MUSN'T** allow carrying out transport when no storekeepers are present at the transport arrival time.	MH	High	In Progress	Yes
45.	Transport	The system **MUST** support management of a transports documents in the transport management database only by the logistics manager.	MH	Low	Done	Yes
46.	Transport	The system **MUST** provide to the carrier a destination	MH	Low	In	Yes

		document for each destination.			Progress	
47.	Transport	The system **MUST** warn the logistics manager if the actual weight exceeds the maximum weight.	MH	Low	Done	Yes
48.	HR & Transport	The system **MUSN'T** allow placement of a carrier without a proper license for the truck selected for transportation.	MH	Low	Done	Yes
49.	Transport	The system **MUST** store the following transport info: transport ID, date, time of departure, truck number, driver name, weight of the truck, source and destinations.	MH	Low	Done	Yes
50.	Transport	The system **MUST** store the following destination document info: document ID, transport ID, list of products provided in this destination.	MH	Low	Done	Yes
51.	Transport	The system **MUST** store the following site info: site ID, address, phone number and name of the contact person.	MH	Low	Done	Yes
52.	Transport	The system **MUST** store the following truck info: license number, model, net weight, maximum capacity weight.	MH	Low	Done	Yes
53.	Transport	The system **MUST** support redesign the route in case of weight deviation only by the logistics manager.	MH	High	In Progress	Yes
54.	Transport	The system **MUST** update the transport document about the redesign of the route only by the logistics manager.	MH	Low	Done	Yes
55.	Transport	The system **MUST** support manage and order the transports only by the logistics manager.	MH	Low	Done	Yes
56.	Transport	The system **MUST** store the following transport document info: date, time of departure, truck number, driver name, weight of the truck, source, destinations and about redesign if done.	MH	Low	Done	Yes
57.	Transport	The system **MUST** support the redesign of downloading or replacing one of the destinations, or replacing a truck, or removing some of the products from the truck only by the logistics manager.	MH	High	In Progress	Yes
58.	Transport	The system **MUST** provide delivery of a product when there is a shortage.	MH	High	Backlog	No
59.		NEW Requirements				
60.	Suppliers &Transport	The system **MUST** match for every new Order a transport.	MH	High	Done	Yes
61.	Suppliers &Transport	The system **MUST** schedule the transport for the nearest appropriate date.	MH	High	In Progress	Yes
62.	Transport & HR	The system **MUST** alert the HR manager if there is no appropriate date for the execution of transport in the coming week.	MH	High	In Progress	Yes
63.	Suppliers & HR	The system **MUST** allow cancelation of an order only by HR-manager or Storekeeper or Logistic manager	MH	Low	In Progress	Yes
64.	Inventory	The system **MUST** allow only the storekeeper to confirm arrival of orders to the warehouse.	MH	Low	Done	Yes
65.	Inventory	The system **MUST** allow only the storekeeper to update quantities of products in stock according to the arrived order.	MH	Low	Done	Yes
66.	Inventory	The system **MUST** allow only the storekeepers to report defective / expired / missing products in an order that has	MH	Low	Done	Yes

		arrived (the report will be documented, and the rest of the products will be updated in the inventory system).				
67.	Suppliers	The system **MUST** allow ordering a product from a supplier according to it's supplier ID.	MH	Low	In Progress	Yes
68.	HR & Inventory	The system **MUST** allow inventory management only by storekeeper.	MH	Low	Done	Yes
69.	HR & Suppliers	The system **MUST** allow only the storekeeper to order from suppliers.	MH	Low	Done	Yes
70.	HR& Transport& Suppliers& Inventory	The system **MUST** allow the store manager to Produce reports and view all the data.	MH	Low	In Progress	Yes
71.	System	The system MUST support multiple users at the same time. These users may log in from different physical locations	MH	Low	In Progress	Yes
72.						
73.						

2. Non-Functional Requirements

ID	Module	Description	Priority	Risk	Status	Feasible
1.	Inventory	The system **MUST** support each product belonging to exactly one category, and each category having exactly one parent category or none, but can have any number of sub-categories. (sub-category and parent category are in a two-way connection)	MH	Low	Done	Yes
2.	Transport	The system **MUST** support weigh of the truck before the truck leaving the source.	MH	Low	Done	Yes
3.	System	The system CAN support interaction through graphical user interface	NTH	Low	In Progress	Yes

Appendix A Terms

Table 1: Terms

#	Term	Description
1.	Management	Set of action which include creation, reading, editing (information) and removal of defined class.
2.	Employee	Person with work contract for working for ‘Superly’
3.	Employee information	ID, Name, Bank details, Salary, Employment conditions, Starting date, Certifications
4.	Employee types	Cashier, Storekeeper, Carrier, Sorter, HR Manager, Logistics Manager, Transport Manager
5.	Shift	Concept used to describe a fixed work period.
6.	Shift Information	Date, Shift Type, Shift Manager’s ID, Count of Employees Needed for each Employee Type, Assigned Employee ID’s for each Employee Type
7.	Shift types	Morning shift, Evening shift
8.	Constraint	Concept used to describe when an Employee is available for work
9.	Constraint Information	Date, Shift Type, Available Employees for the Shift
10.	Target amount	The number of items needed for certain product in a specific store after it gets under its minimum amount.
11.	License types	B, C, C1, C + E
12.	Truck model	Van, Semi-Trailer, Double-Trailer, Full-Trailer.
13.	Shipping areas	South, North, East, West, Southwest, Southeast, Northwest, Northeast.
14.	Transport	Delivering products from supplier to destination.
15.	Redesign	When the transport exceeds the possible weight for it the transport returns to redesign.
16.	Transport Document	A document describing the transport after it is completed.
17.	Destination Document	A document describing the products delivered in a particular shipment to the destination.
18.	Shortage	When inventory runs out and exceeds the minimum quantity set for the product.
19.	Appropriate date	The closest date that possible according to the supply days of the supplier and to the carrier's schedule.
20.	Supplier card	Contains: name, address, bank number, contacts, manufacturers, paying agreement.
21.	Product supplier ID	The ID this product has at the supplier system.

Appendix B Open Questions

Table 3: Questions

#	Topic	Issue
1.	Document	Will there be an additional need for the documents?
2.		
3.		

Appendix C Assumptions

Table 2: Assumptions

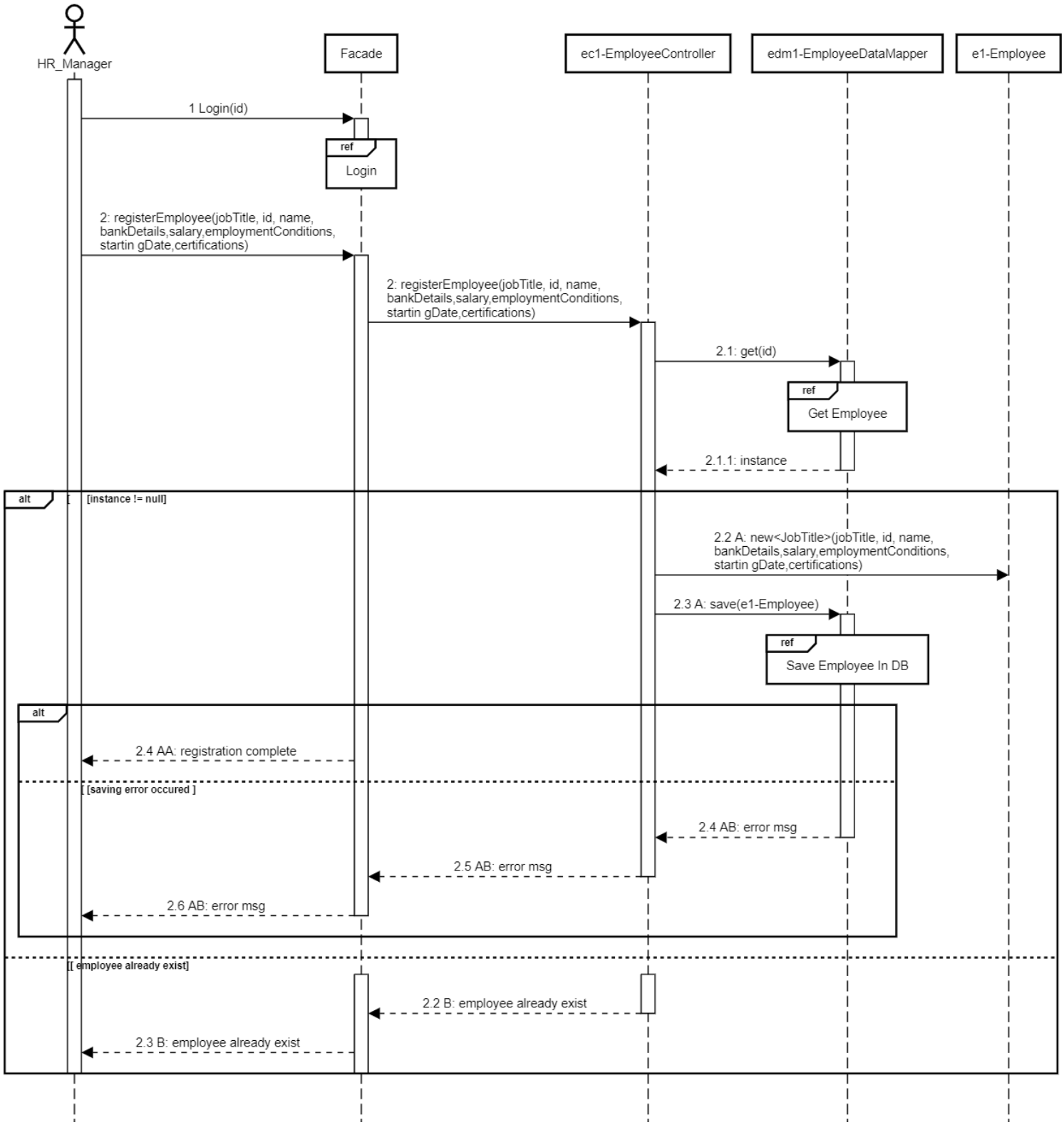
#	Topic	Assumption
1.	Suppliers	The contact's info is only phone number and name.
2.	Suppliers	We can't change orders from by order suppliers in the last 24 hours before the shipment.
3.	Suppliers	A supplier has only one type of agreement, from : by order, routine, not transporting.
4.	Suppliers	When stocks running low, we choose the cheapest supplier to order from, not considering his arrival time.
5.	Suppliers	We can add new supplier for the system without agreement. (Maybe we want just to save his information for future deals).
6.	Suppliers	We assume that when adding an item to agreement, the supplier manager knows both the id this item has at the supplier catalog and the id this item has at inventory.
7.	Inventory	There is predefined target amount per product per store which is above the min amount, such that when creating an order because of low stock the order amount should be target-min
8.	Suppliers+Inventory	The System is shut down at the end of each workday, and turned on at the beginning of the day. At the beginning of each day we send the suppliers the updated amounts for the next day's orders if needed.
9.	Suppliers+Inventory	When a delivery arrives at the warehouse, the orderID and supplierID is known (attached to the driver's receipt)
10.	Suppliers+Inventory	Each supplier can have different ID for the same product (For example: Bamba can have ID 1 in Superly, ID 5 in supplier 1 and ID 1 in supplier 2)
11.	Suppliers	The "best" supplier is the cheapest one regardless to arriving time.
12.	Suppliers+Inventory	Printing an order on screen equals to sending the order to the corresponding supplier.
13.	Suppliers+Inventory	Sending an order that has already been sent to a supplier equals to informing the supplier about updating the order.

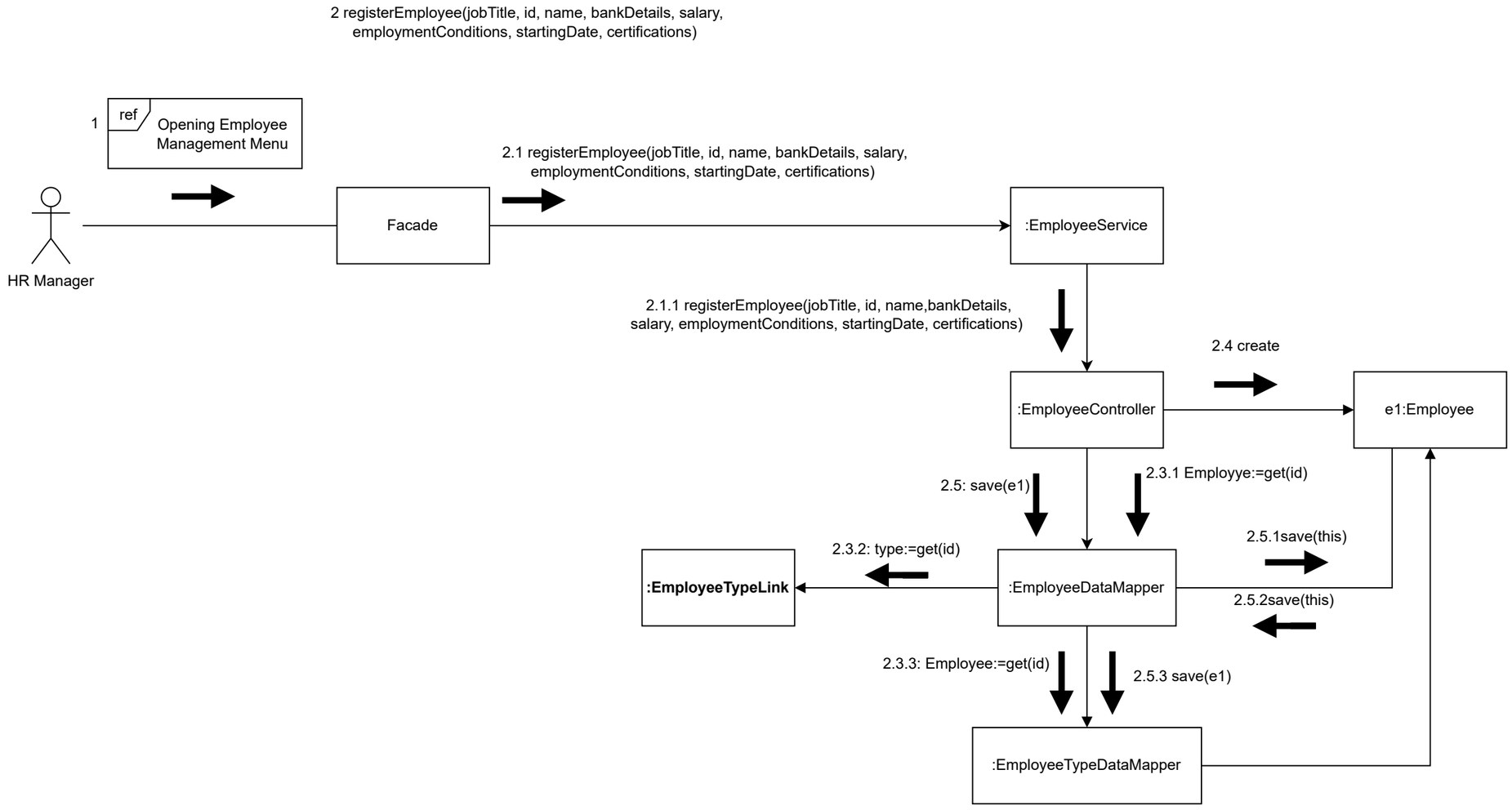
Version History

Table 2: Version History

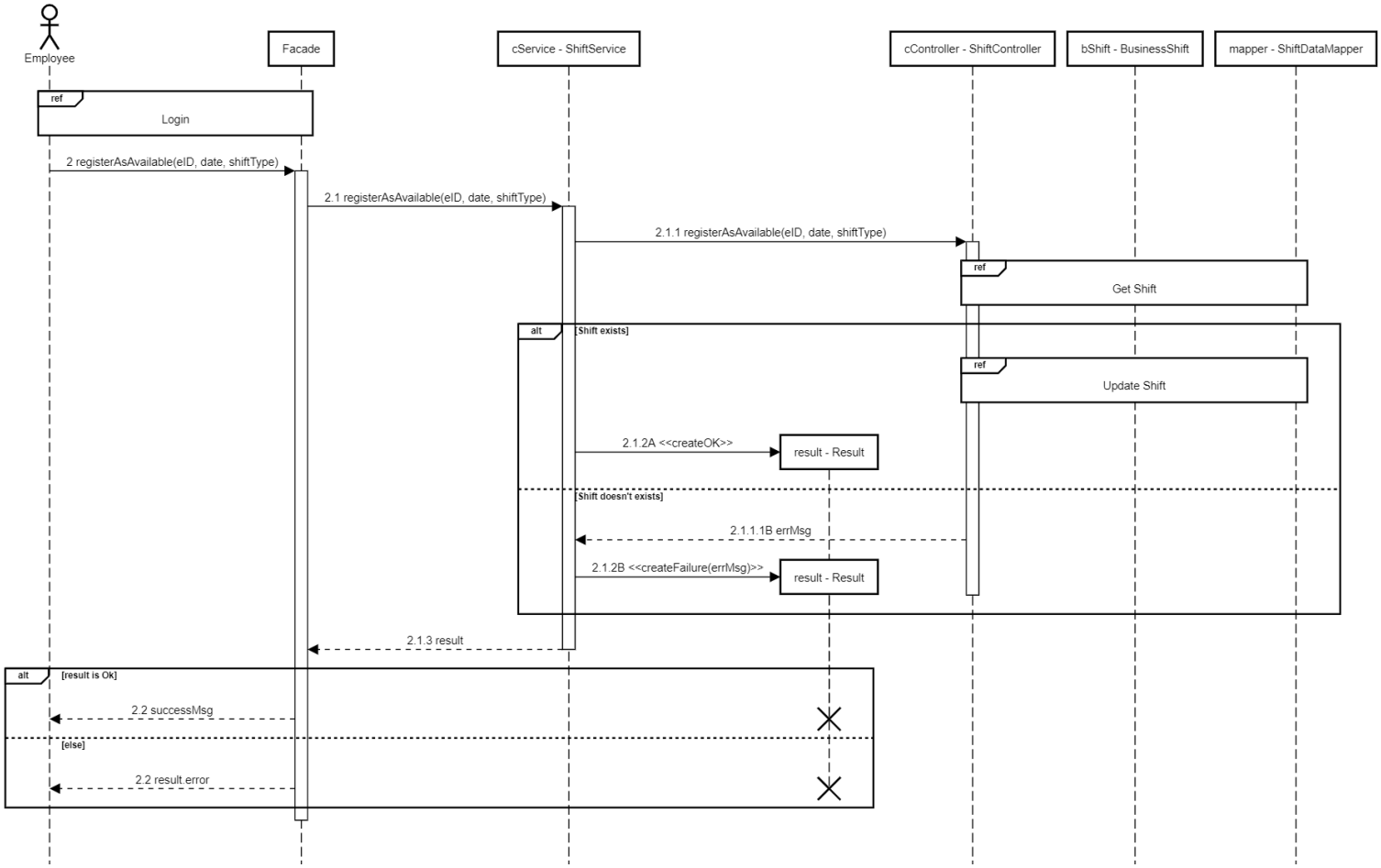
Date	Author	Remarks
25/04/2022	Roi Tiefenbrunn	First draft
09/05/2022	Yonatan Diga	Added suppliers & Inventory requirements and suppliers assumptions.
11/05/2022	Yonatan Diga	Improved requirements
14/05/2022	Tomer Ravkaie	Improved Requirements and Assumptions
15/05/2022	Roi Tiefenhbrunn	Deleting of non-functional reqs for HR-module Adding term ‘Employee’, ‘Employee Information’, ‘Employee types’, ‘Shift’, ‘Shift Information’, ‘Shift Types’ Adding requirements 12, 42
15/05/2022	Roi Tiefenbrunn	Completing Requirements 1, 4, 5, 11 Adding Requirement 2 Adding the terms ‘Constraint’, ‘Constraint Information’ Removed Requirement 7
15/05/2022	Chai-Shalev Hadad	Add transport requirements. Fix the incorrect requirements. Add terms and open question.
02/06/2022	Yonatan Diga	Added requirements for assignment 3
03/06/2022	Yonatan Diga	Updated permissions.
11/06/2022	Yonatan Diga	Updated requirements.
12/06/2022	Roi Tiefenbrunn	Added multi-user and GUI requirements as they are requested by the client

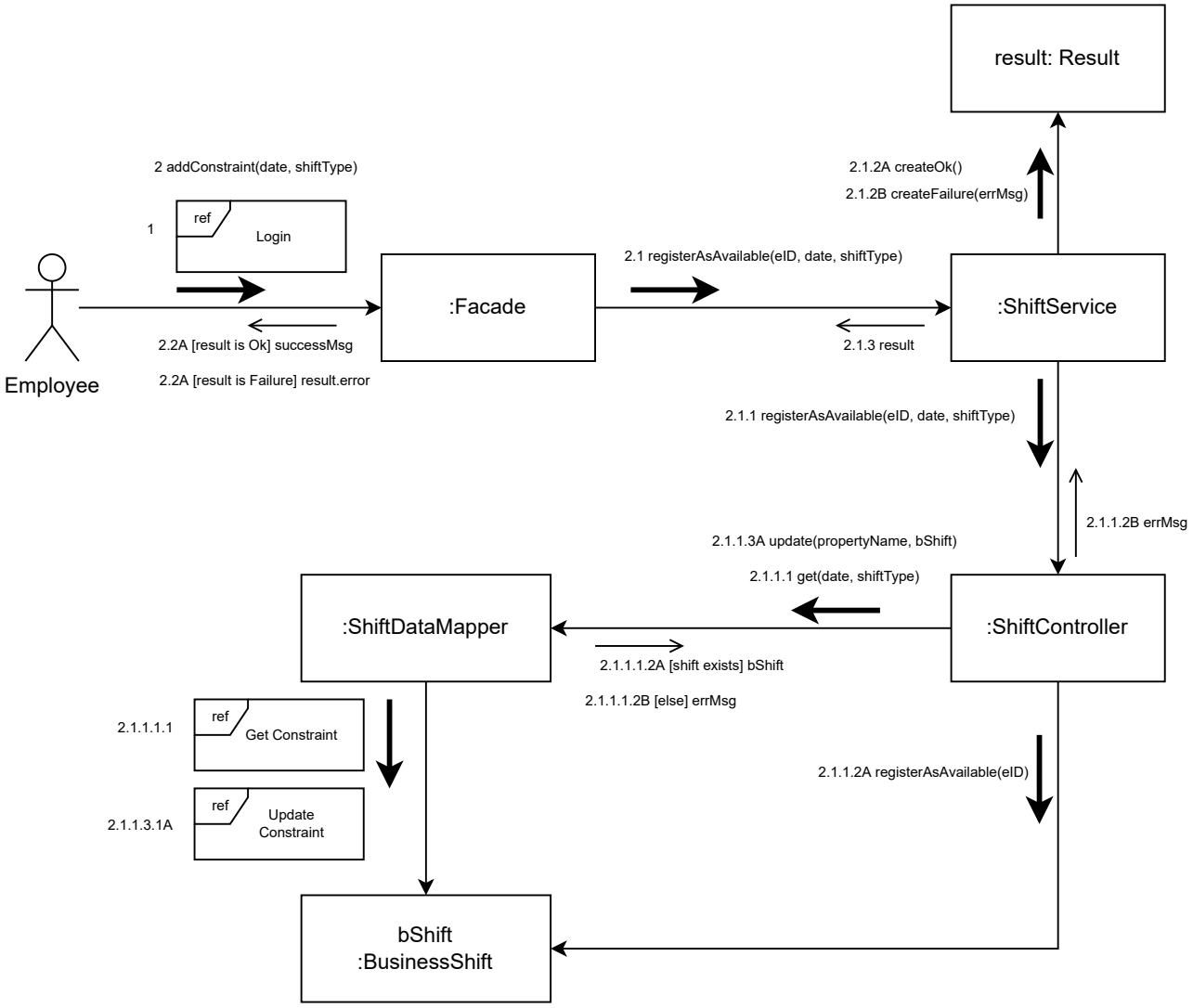
New Employee Registration



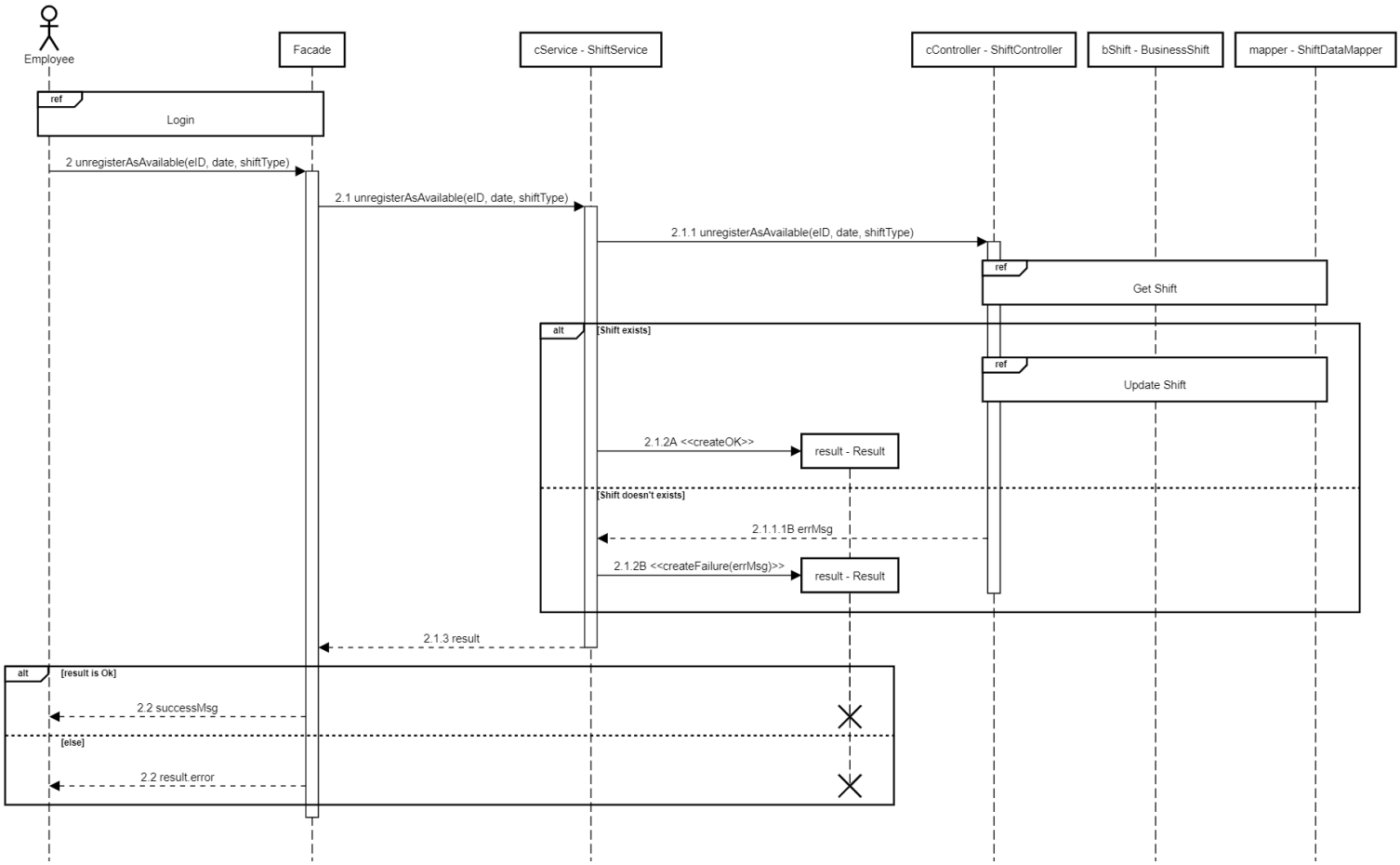

2 registerEmployee(jobTitle, id, name, bankDetails, salary,
employmentConditions, startingDate, certifications)
1
ref
Opening Employee
Management Menu
HR Manager
Facade
2.1 registerEmployee(jobTitle, id, name, bankDetails, salary,
employmentConditions, startingDate, certifications)
:EmployeeService
2.1.1 registerEmployee(jobTitle, id, name,bankDetails,
salary, employmentConditions, startingDate, certifications)
2.4 create
:EmployeeController
e1:Employee
2.5: save(e1)
2.3.1 Employye:=get(id)
2.3.2: type:=get(id)
:EmployeeTypeLink
:EmployeeDataMapper
2.5.1save(this)
2.5.2save(this)
2.3.3: Employee:=get(id)
2.5.3 save(e1)
:EmployeeTypeDataMapper

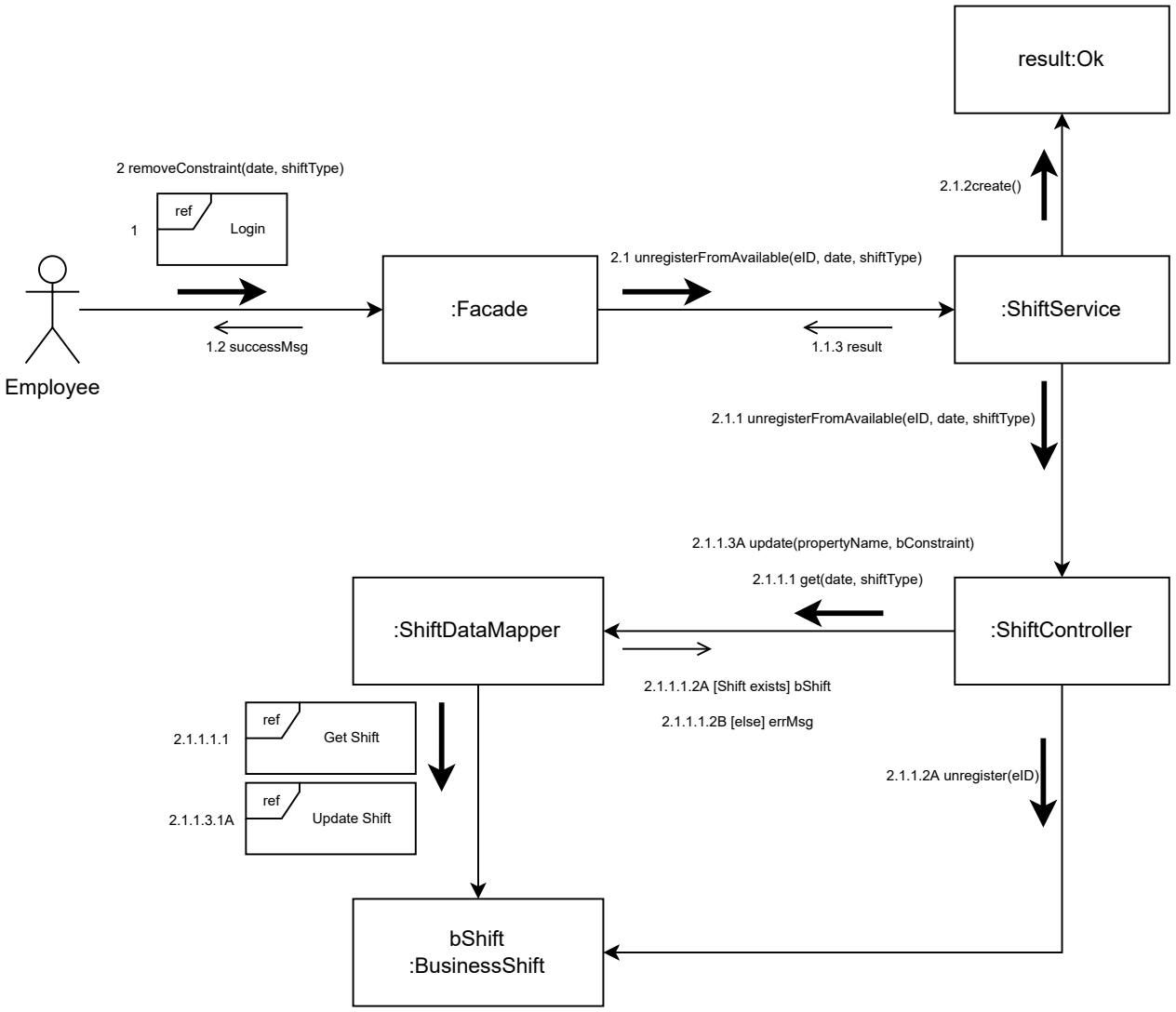
Adding Constraint



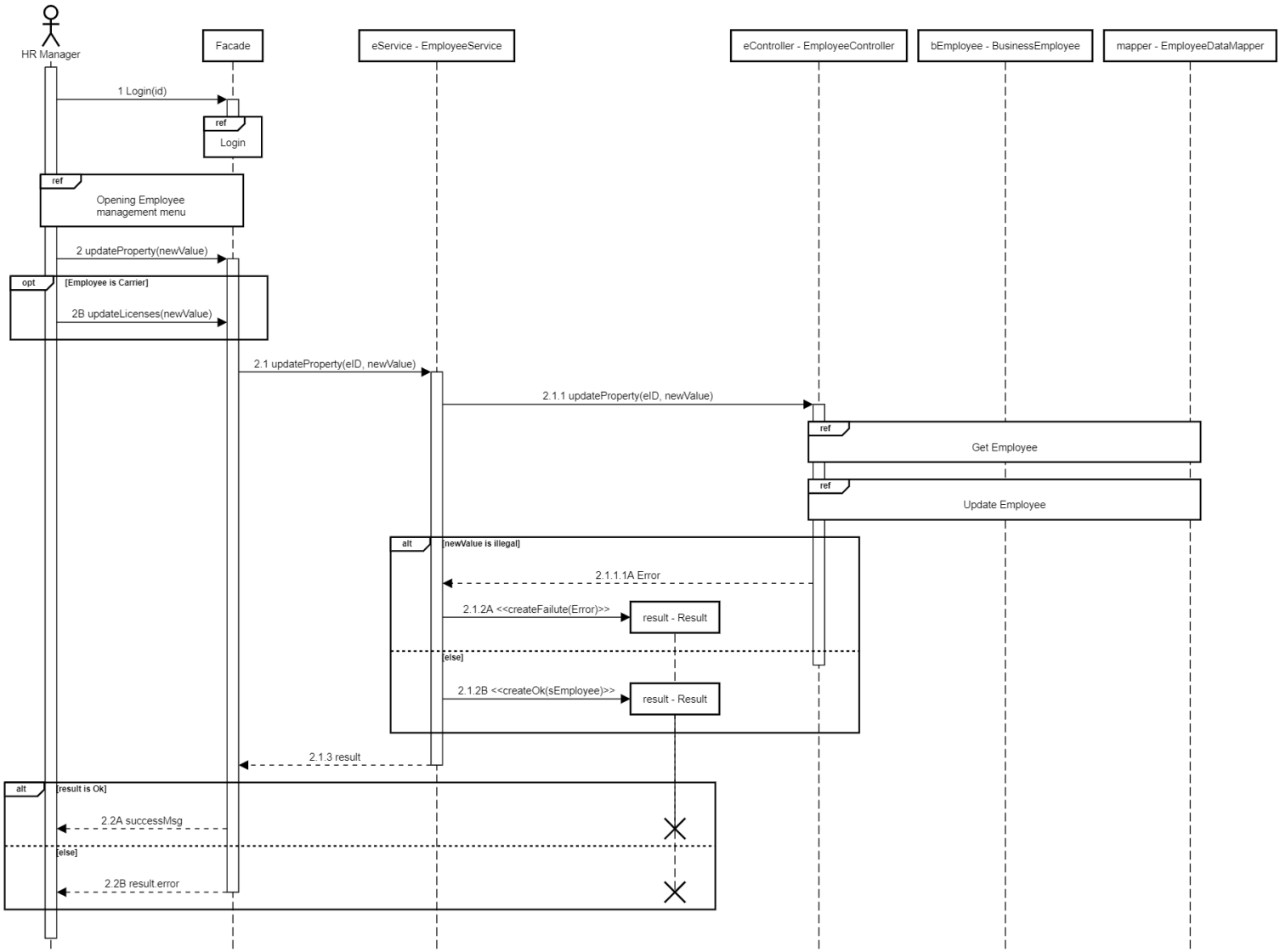

result: Result
2 addConstraint(date, shiftType)
2.1.2A createOk()
2.1.2B createFailure(errMsg)
1
ref
Login
2.1 registerAsAvailable(eID, date, shiftType)
:Facade
:ShiftService
Employee
2.2A [result is Ok] successMsg
2.2A [result is Failure] result.error
2.1.3 result
2.1.1 registerAsAvailable(eID, date, shiftType)
2.1.1.2B errMsg
2.1.1.3A update(propertyName, bShift)
2.1.1.1 get(date, shiftType)
:ShiftDataMapper
:ShiftController
2.1.1.1.2A [shift exists] bShift
2.1.1.1.2B [else] errMsg
2.1.1.1.1
ref
Get Constraint
2.1.1.2A registerAsAvailable(eID)
2.1.1.3.1A
ref
Update Constraint
bShift
:BusinessShift

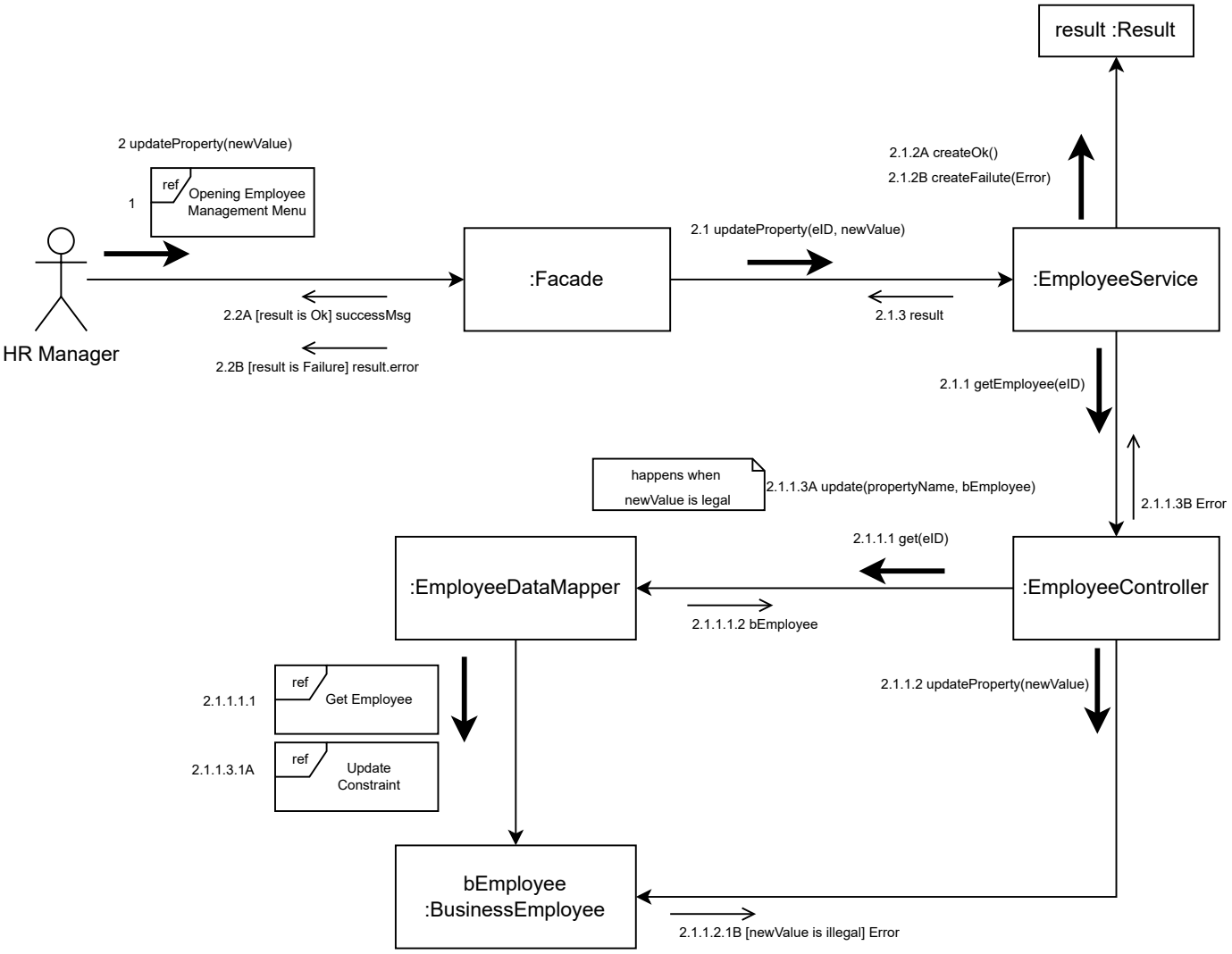
Adding Constraint

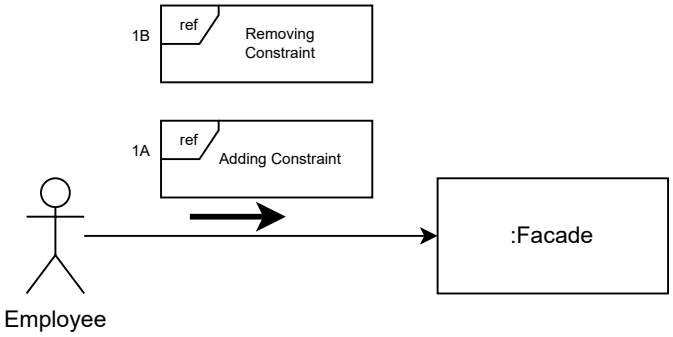



result:Ok
2 removeConstraint(date, shiftType)
2.1.2create()
1
ref
Login
2.1 unregisterFromAvailable(eID, date, shiftType)
:Facade
:ShiftService
1.2 successMsg
1.1.3 result
Employee
2.1.1 unregisterFromAvailable(eID, date, shiftType)
2.1.1.3A update(propertyName, bConstraint)
2.1.1.1 get(date, shiftType)
:ShiftDataMapper
:ShiftController
2.1.1.1.2A [Shift exists] bShift
2.1.1.1.2B [else] errMsg
2.1.1.1.1
ref
Get Shift
2.1.1.2A unregister(eID)
2.1.1.3.1A
ref
Update Shift
bShift
:BusinessShift

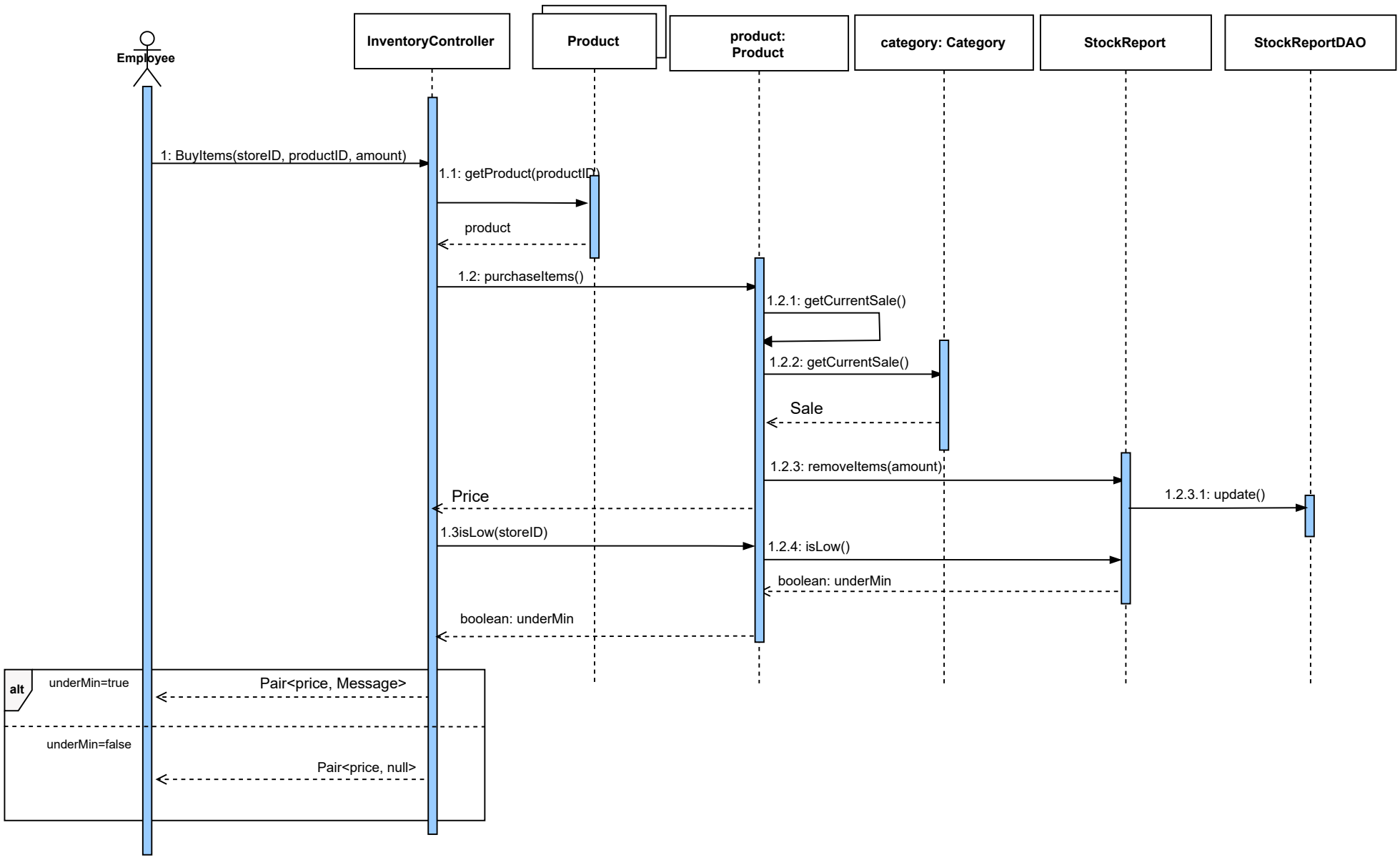
Updating Employee Details

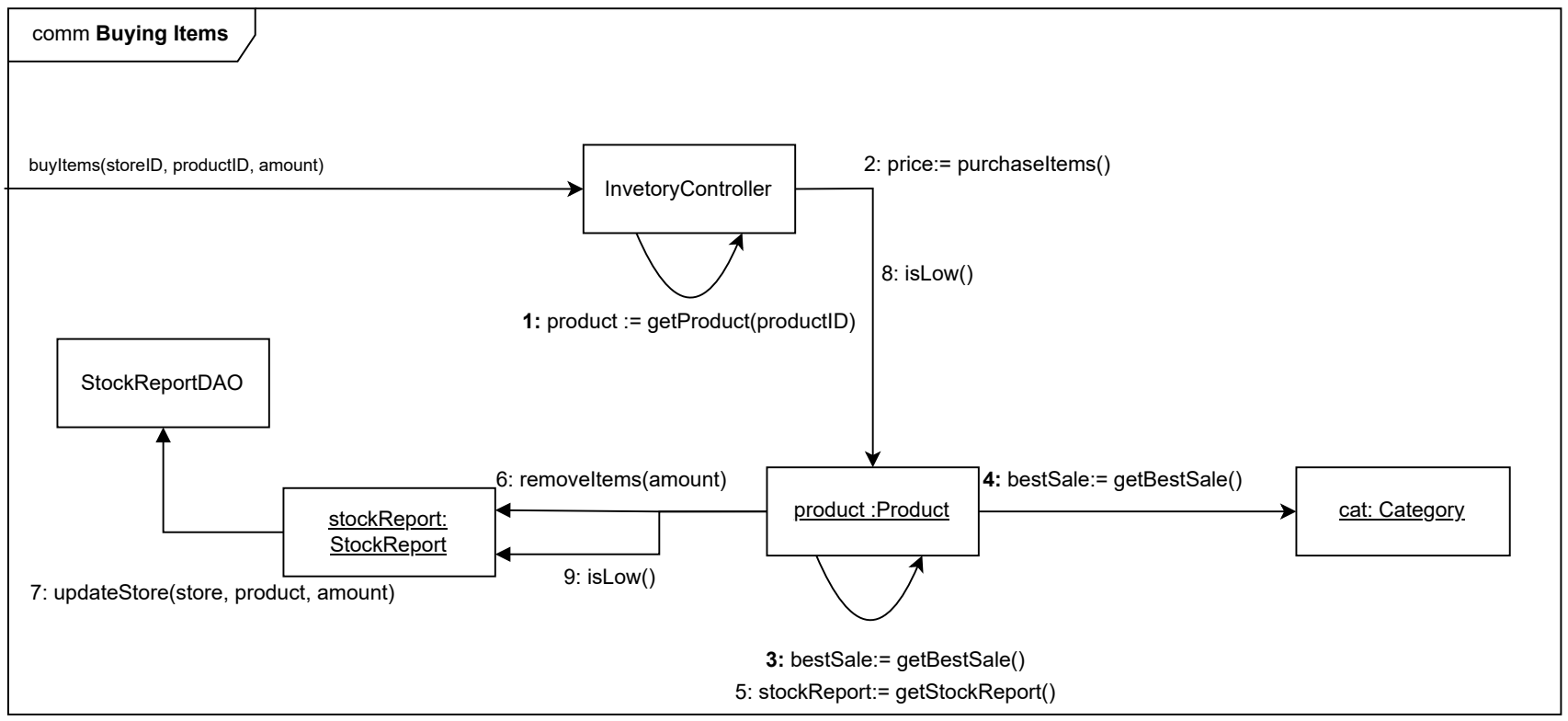



result :Result
2 updateProperty(newValue)
1
ref
Opening Employee Management Menu
2.1.2A createOk()
2.1.2B createFailute(Error)
2.1 updateProperty(eID, newValue)
:Facade
:EmployeeService
HR Manager
2.2A [result is Ok] successMsg
2.2B [result is Failure] result.error
2.1.3 result
2.1.1 getEmployee(eID)
happens when newValue is legal
2.1.1.3A update(propertyName, bEmployee)
2.1.1.3B Error
2.1.1.1 get(eID)
:EmployeeDataMapper
:EmployeeController
2.1.1.1.2 bEmployee
2.1.1.1.1
ref
Get Employee
2.1.1.2 updateProperty(newValue)
2.1.1.3.1A
ref
Update Constraint
bEmployee
:BusinessEmployee
2.1.1.2.1B [newValue is illegal] Error

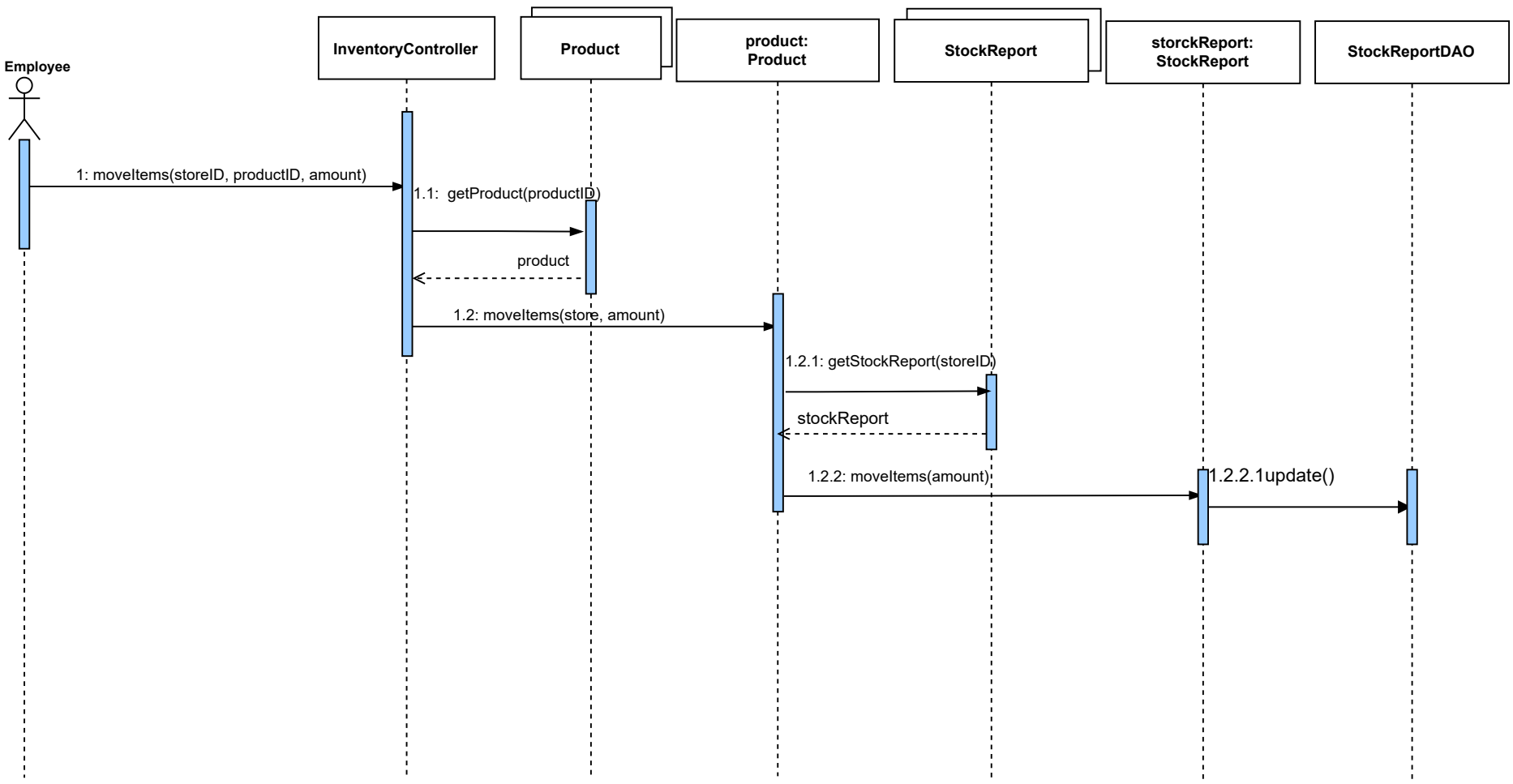

1B
ref
Removing
Constraint
1A
ref
Adding Constraint
:Facade
Employee

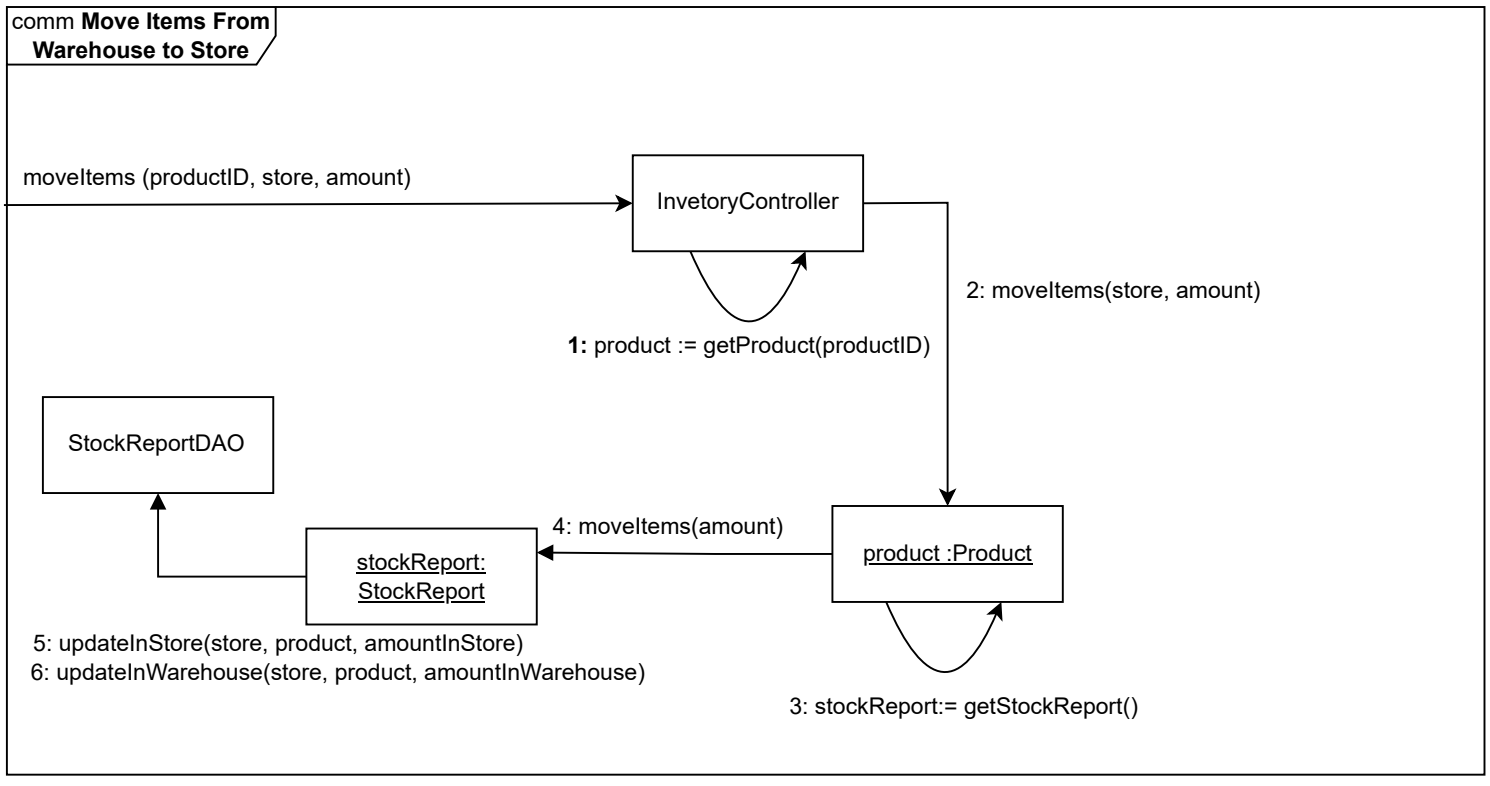
Buying Items



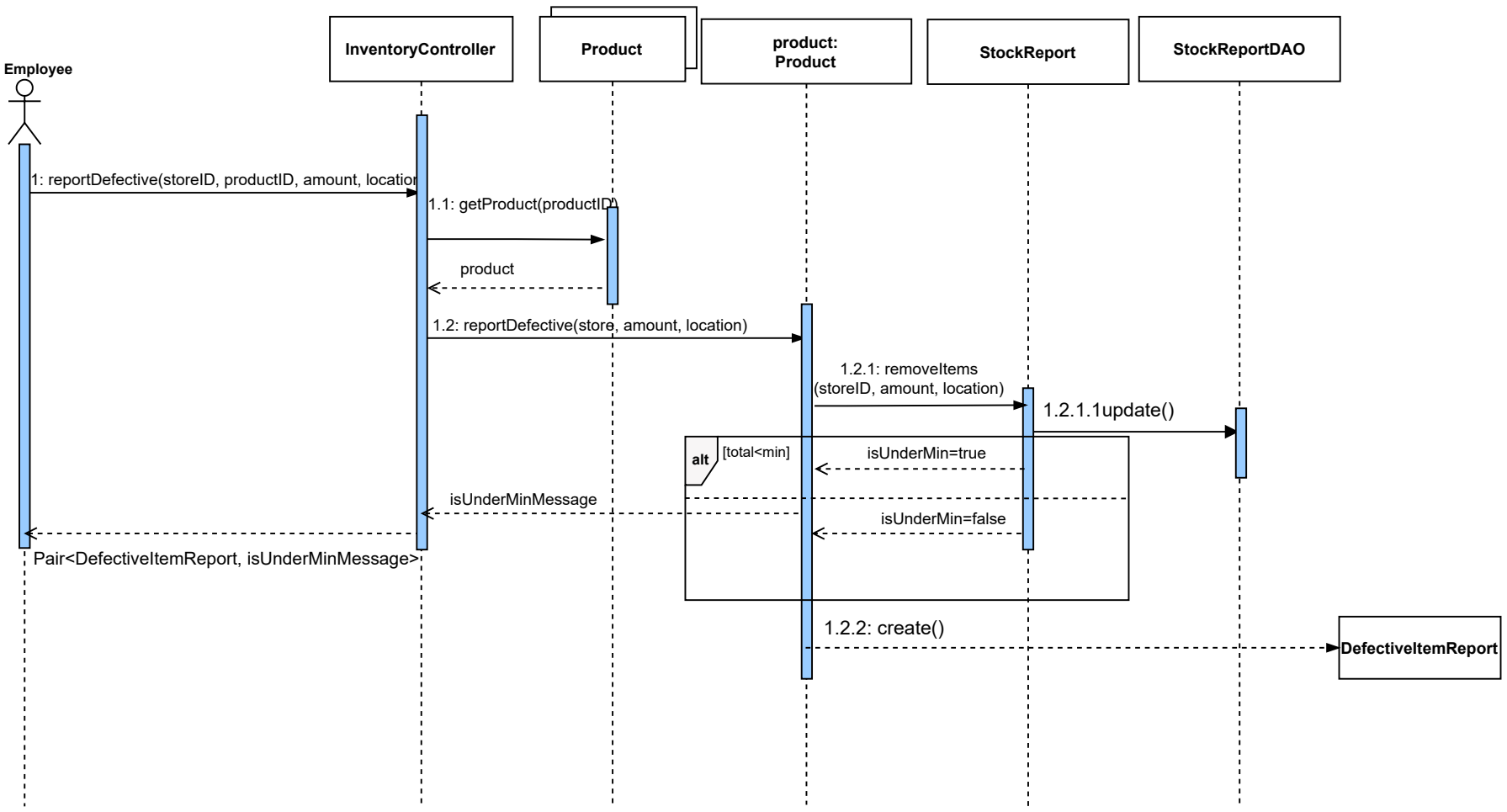

comm **Buying Items**
buyItems(storeID, productID, amount)
InvetoryController
2: price:= purchaseItems()
8: isLow()
1: product := getProduct(productID)
StockReportDAO
6: removeItems(amount)
4: bestSale:= getBestSale()
stockReport:
StockReport
product :Product
cat: Category
9: isLow()
7: updateStore(store, product, amount)
3: bestSale:= getBestSale()
5: stockReport:= getStockReport()

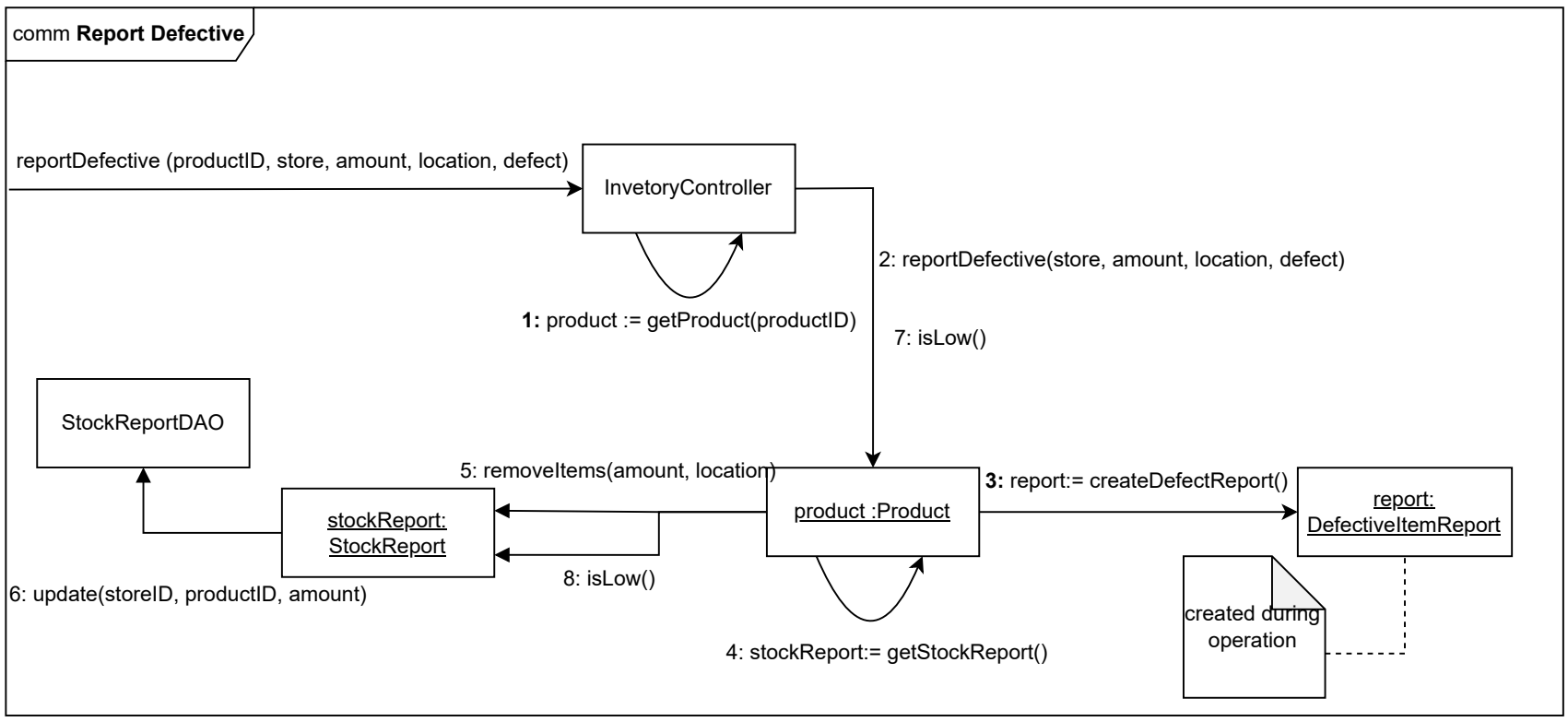
Move Items from warehouse to store



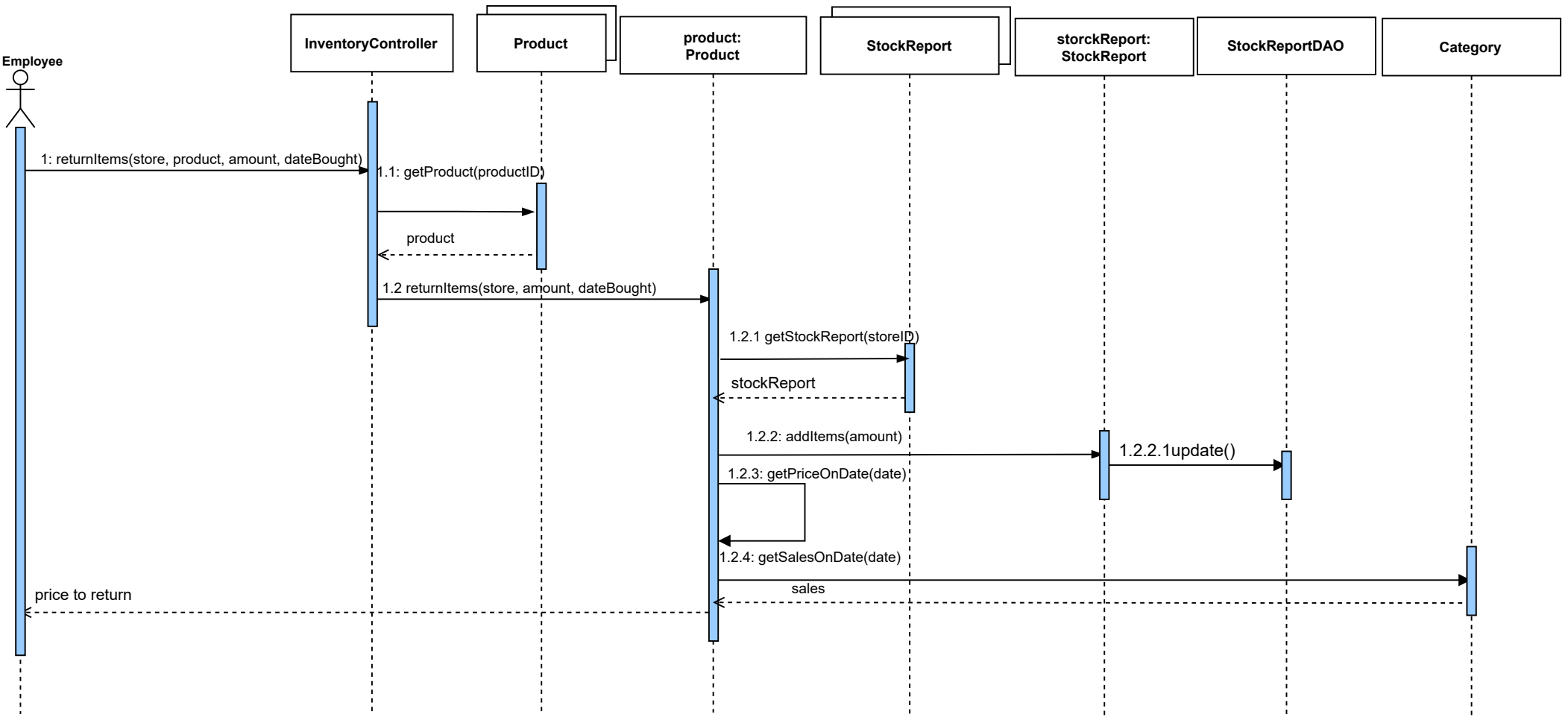

comm **Move Items From Warehouse to Store**
moveItems (productID, store, amount)
InvetoryController
2: moveItems(store, amount)
1: product := getProduct(productID)
StockReportDAO
4: moveItems(amount)
stockReport: StockReport
product :Product
5: updateInStore(store, product, amountInStore)
6: updateInWarehouse(store, product, amountInWarehouse)
3: stockReport:= getStockReport()

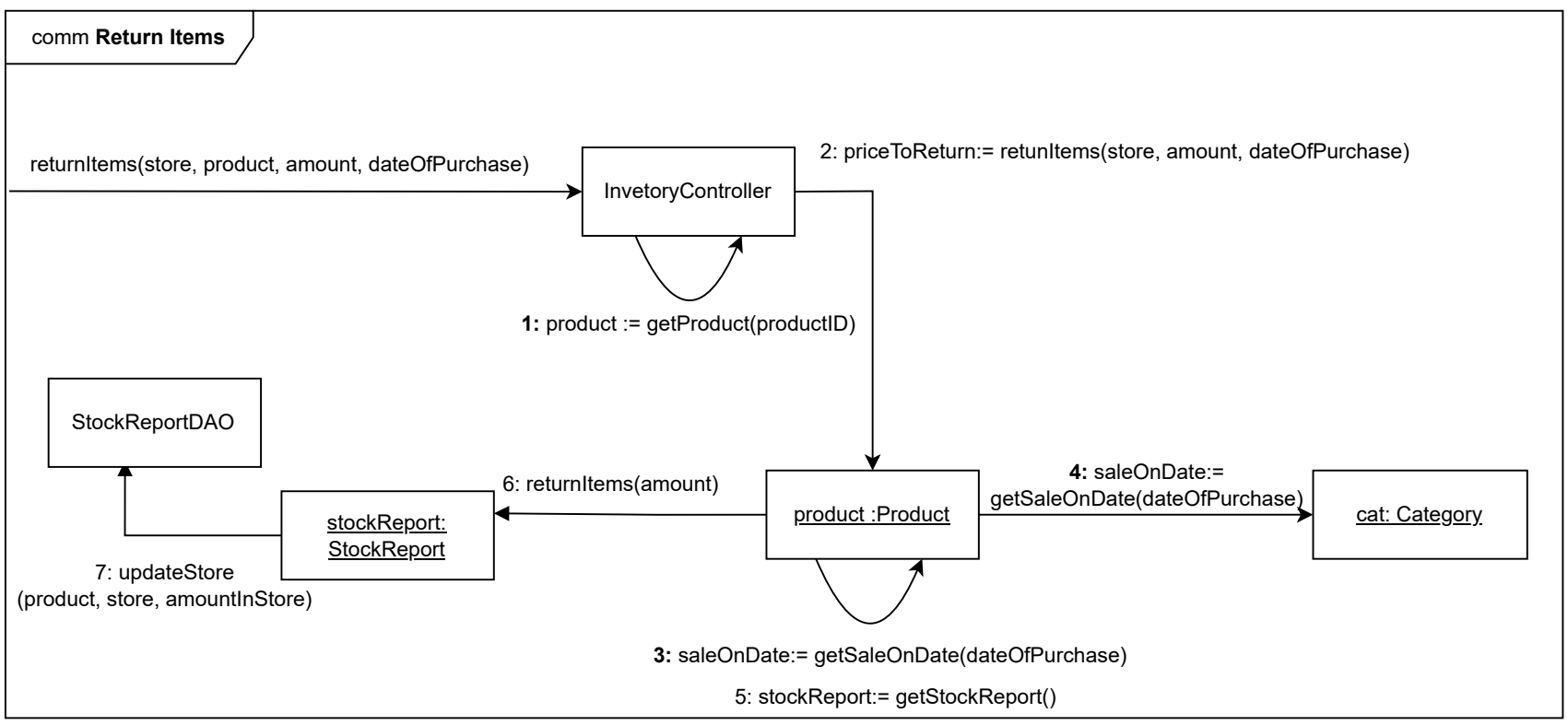
Report Defective

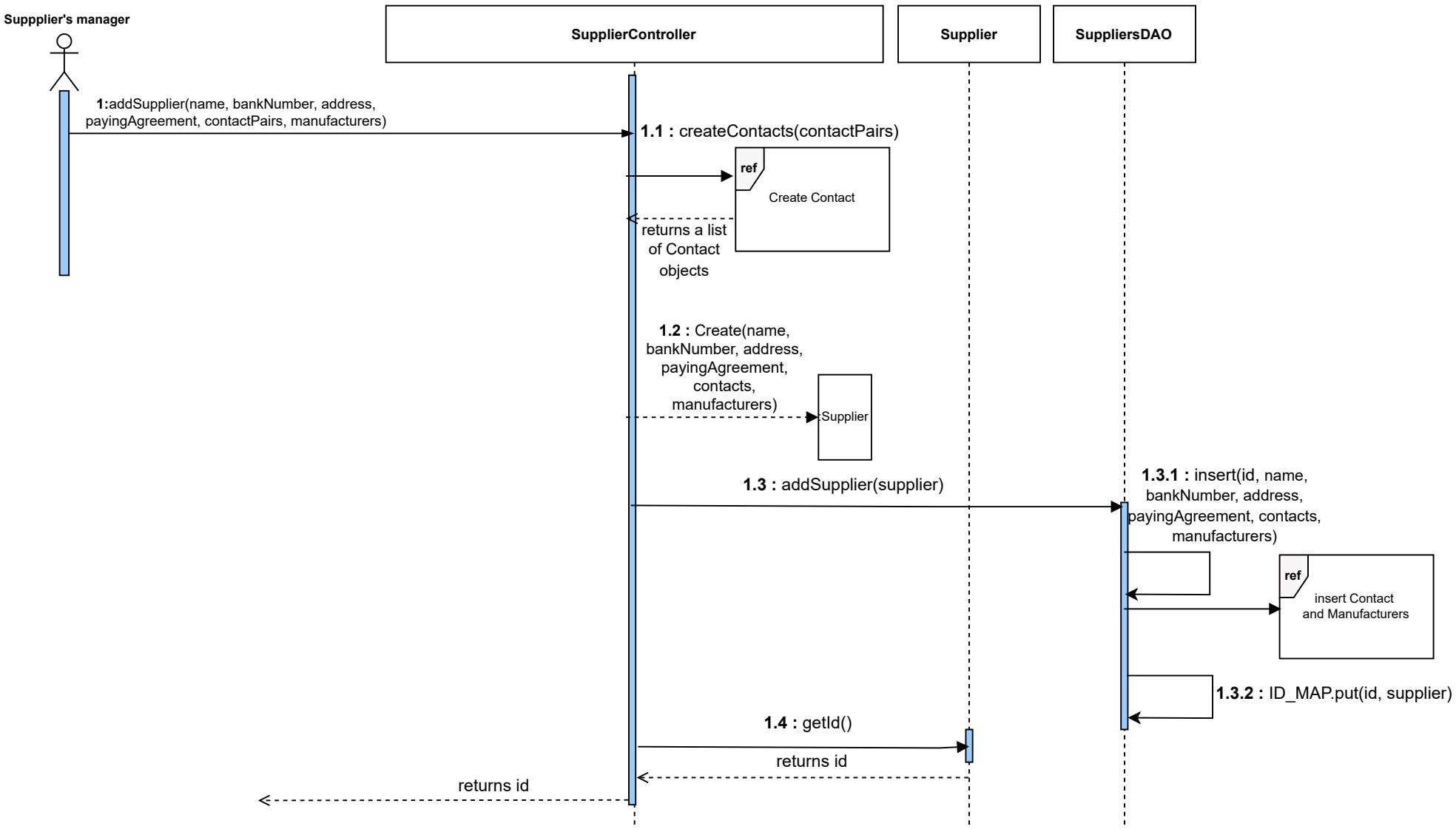


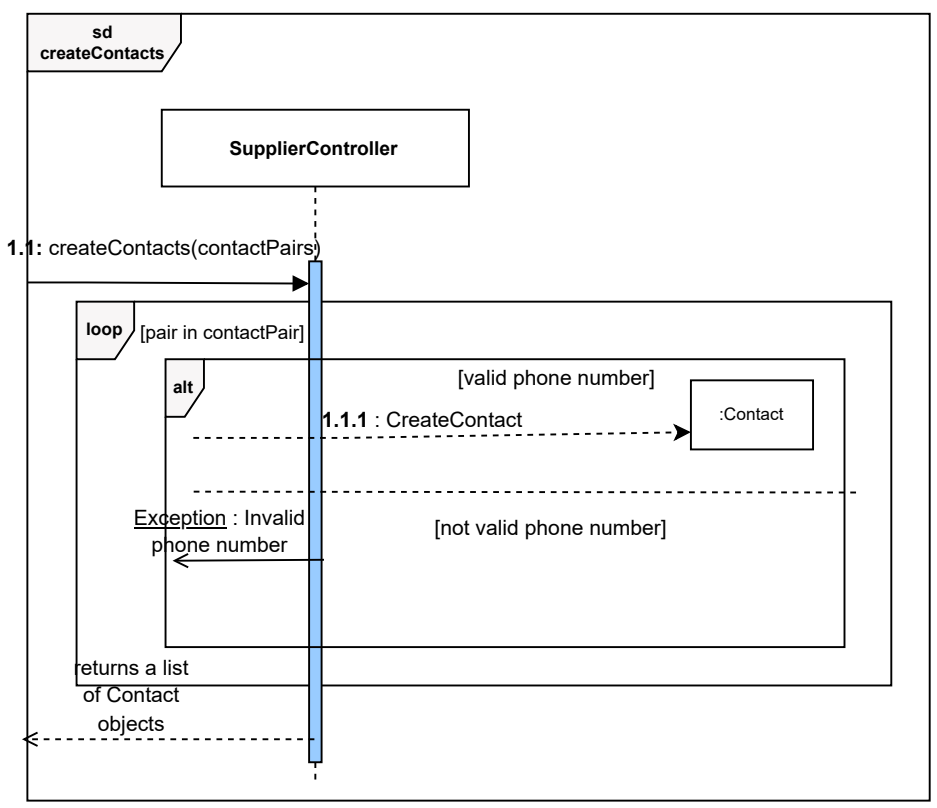

comm **Report Defective**
reportDefective (productID, store, amount, location, defect)
InvetoryController
2: reportDefective(store, amount, location, defect)
1: product := getProduct(productID)
7: isLow()
StockReportDAO
5: removeItems(amount, location)
3: report:= createDefectReport()
stockReport:
StockReport
product :Product
report:
DefectiveItemReport
8: isLow()
6: update(storeID, productID, amount)
created during
operation
4: stockReport:= getStockReport()

ReturnItems

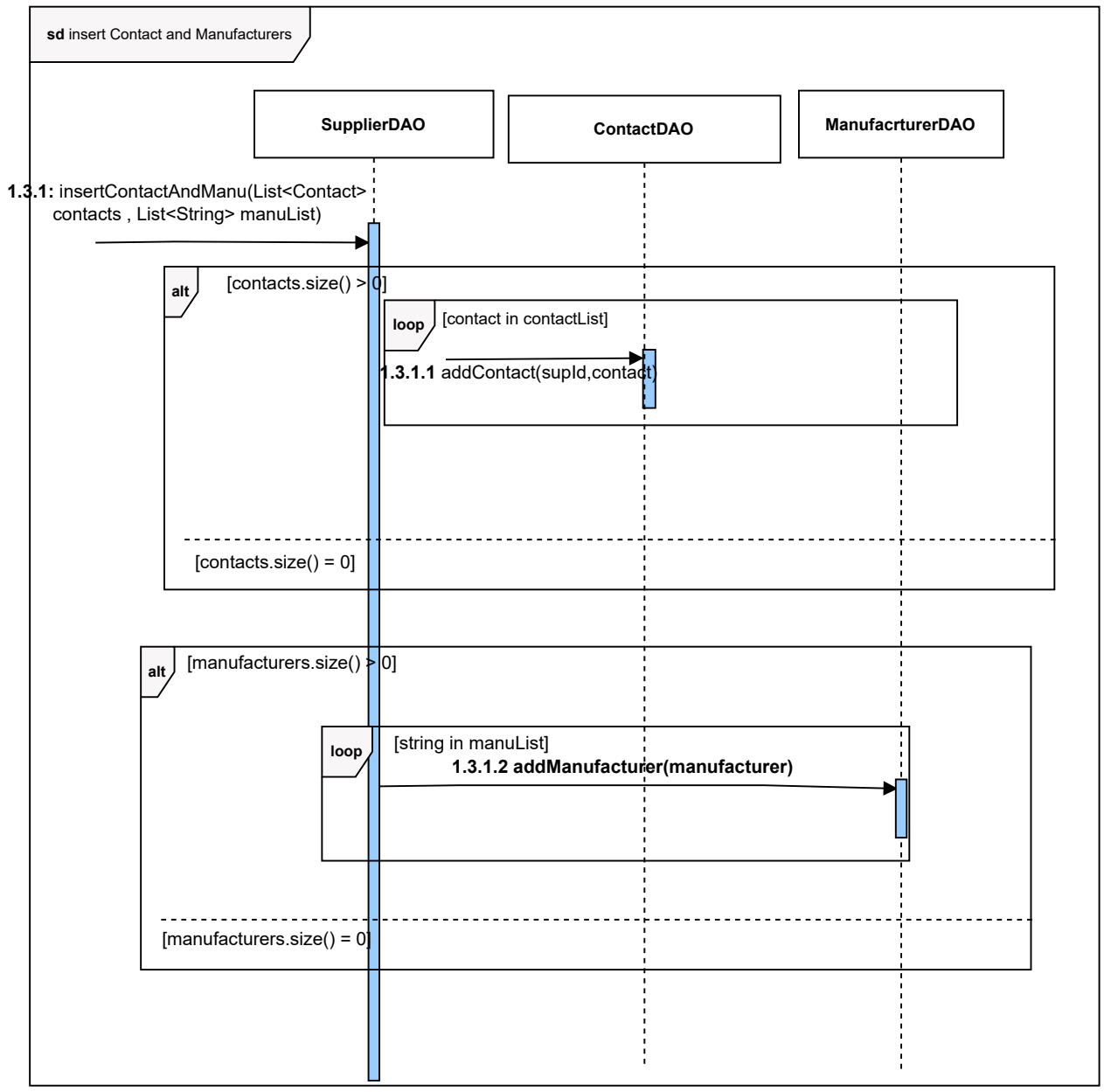


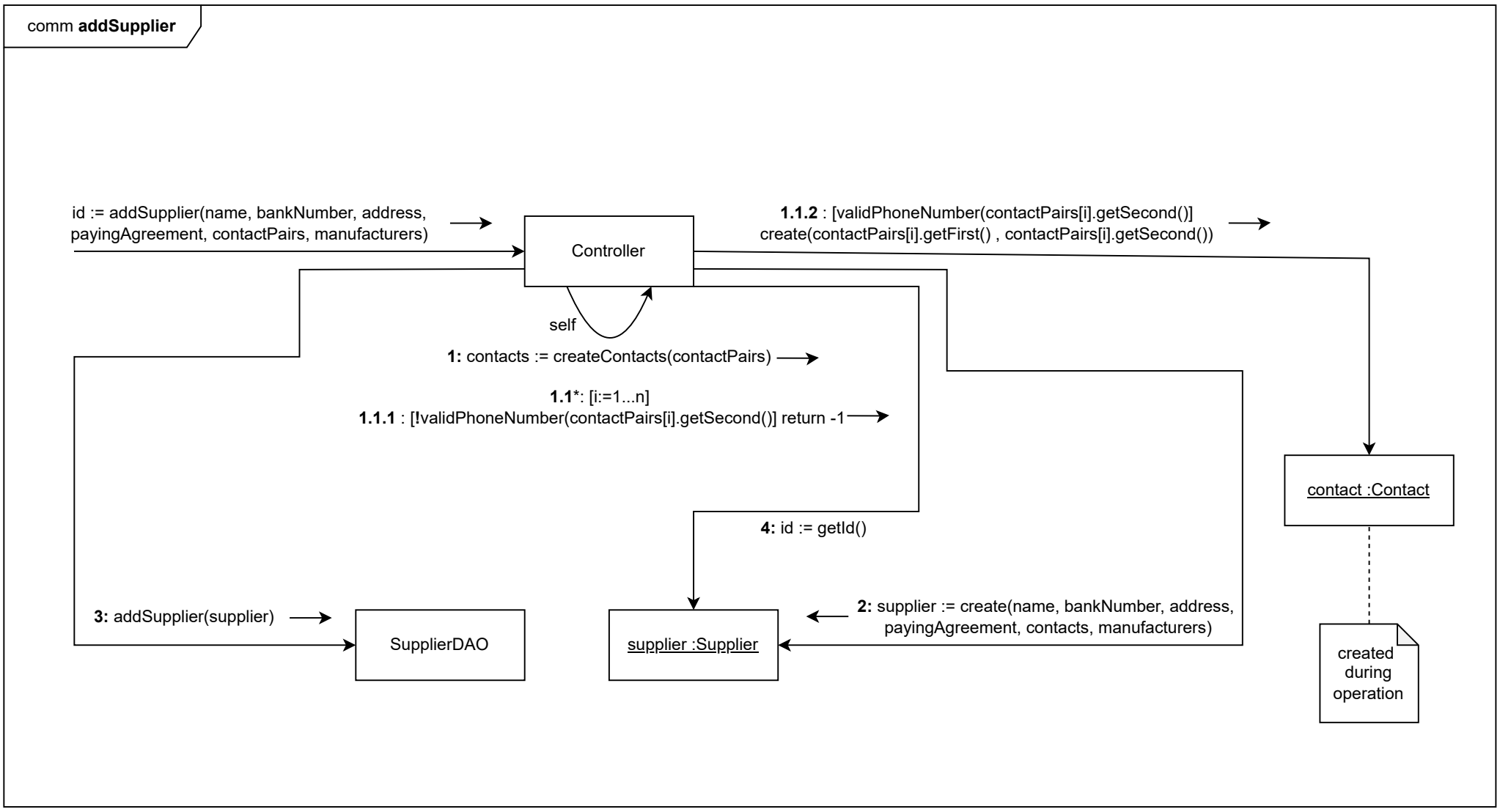

comm **Return Items**
returnItems(store, product, amount, dateOfPurchase)
InvetoryController
2: priceToReturn:= retunItems(store, amount, dateOfPurchase)
1: product := getProduct(productID)
StockReportDAO
6: returnItems(amount)
4: saleOnDate:=
getSaleOnDate(dateOfPurchase)
stockReport:
StockReport
product :Product
cat: Category
7: updateStore
(product, store, amountInStore)
3: saleOnDate:= getSaleOnDate(dateOfPurchase)
5: stockReport:= getStockReport()



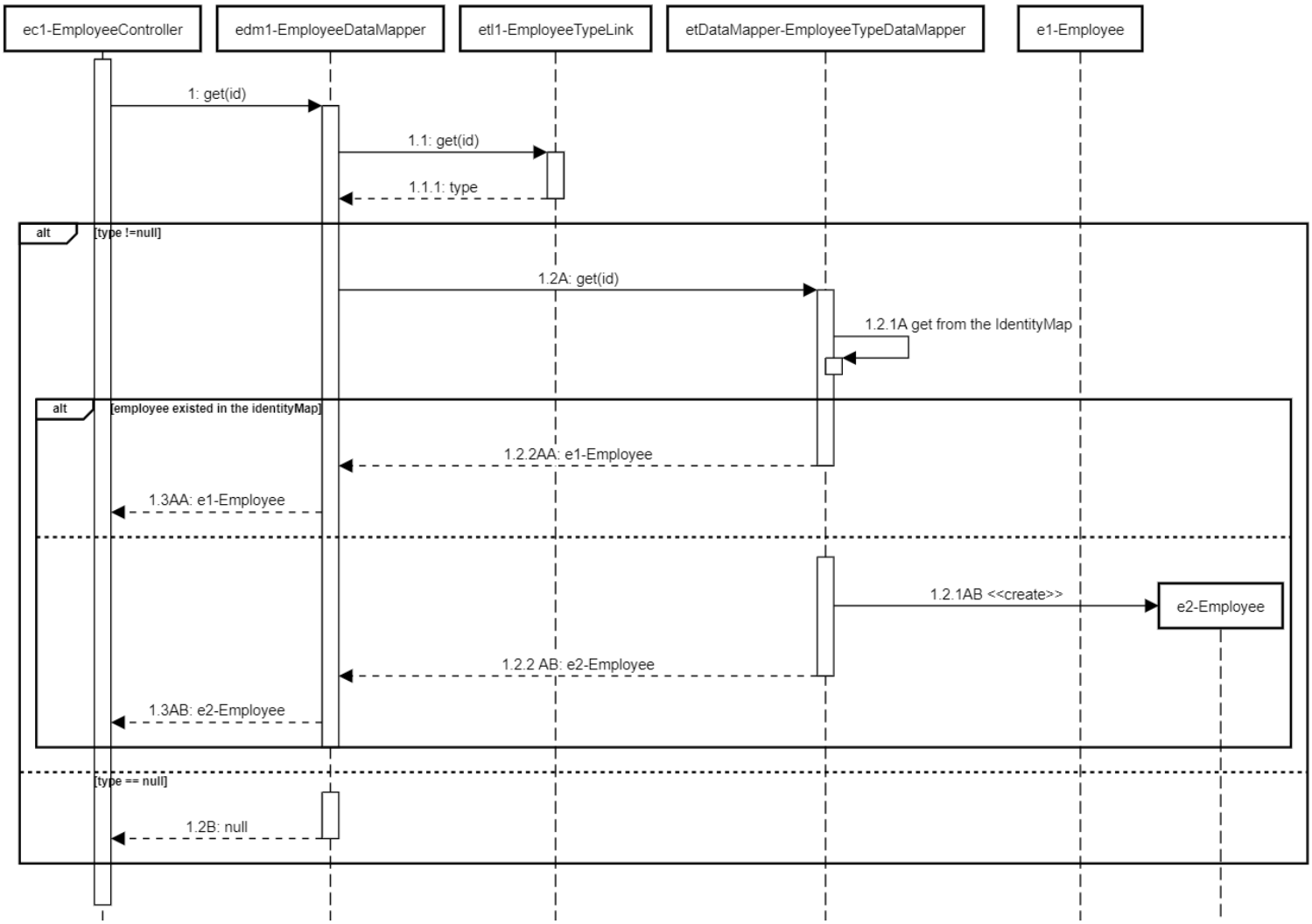


When adding a Supplier to the system, the user doesn't have to add agreement, and in fact adding agreement is a different and seperated use case, so we didn't write this here

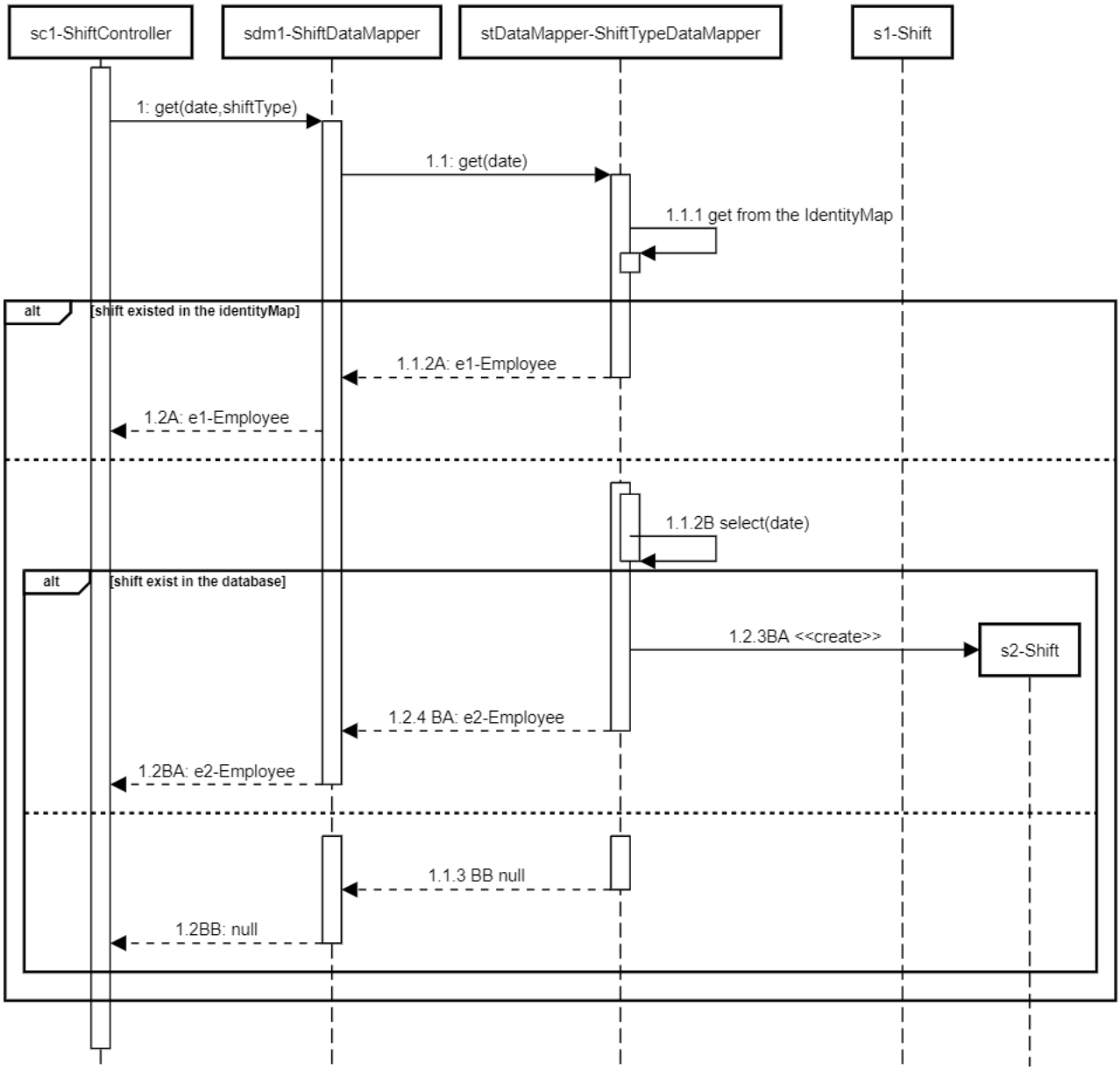

sd insert Contact and Manufacturers
SupplierDAO
ContactDAO
ManufacrturerDAO
1.3.1: insertContactAndManu(List<Contact>
contacts , List<String> manuList)
alt
[contacts.size() > 0]
loop
[contact in contactList]
1.3.1.1 addContact(supId,contact)
[contacts.size() = 0]
alt
[manufacturers.size() > 0]
loop
[string in manuList]
1.3.1.2 addManufacturer(manufacturer)
[manufacturers.size() = 0]


comm addSupplier
id := addSupplier(name, bankNumber, address,
payingAgreement, contactPairs, manufacturers)
Controller
1.1.2 : [validPhoneNumber(contactPairs[i].getSecond())]
create(contactPairs[i].getFirst() , contactPairs[i].getSecond())
self
1: contacts := createContacts(contactPairs)
1.1*: [i:=1...n]
1.1.1 : [!validPhoneNumber(contactPairs[i].getSecond()] return -1
contact :Contact
4: id := getId()
3: addSupplier(supplier)
SupplierDAO
supplier :Supplier
2: supplier := create(name, bankNumber, address,
payingAgreement, contacts, manufacturers)
created
during
operation

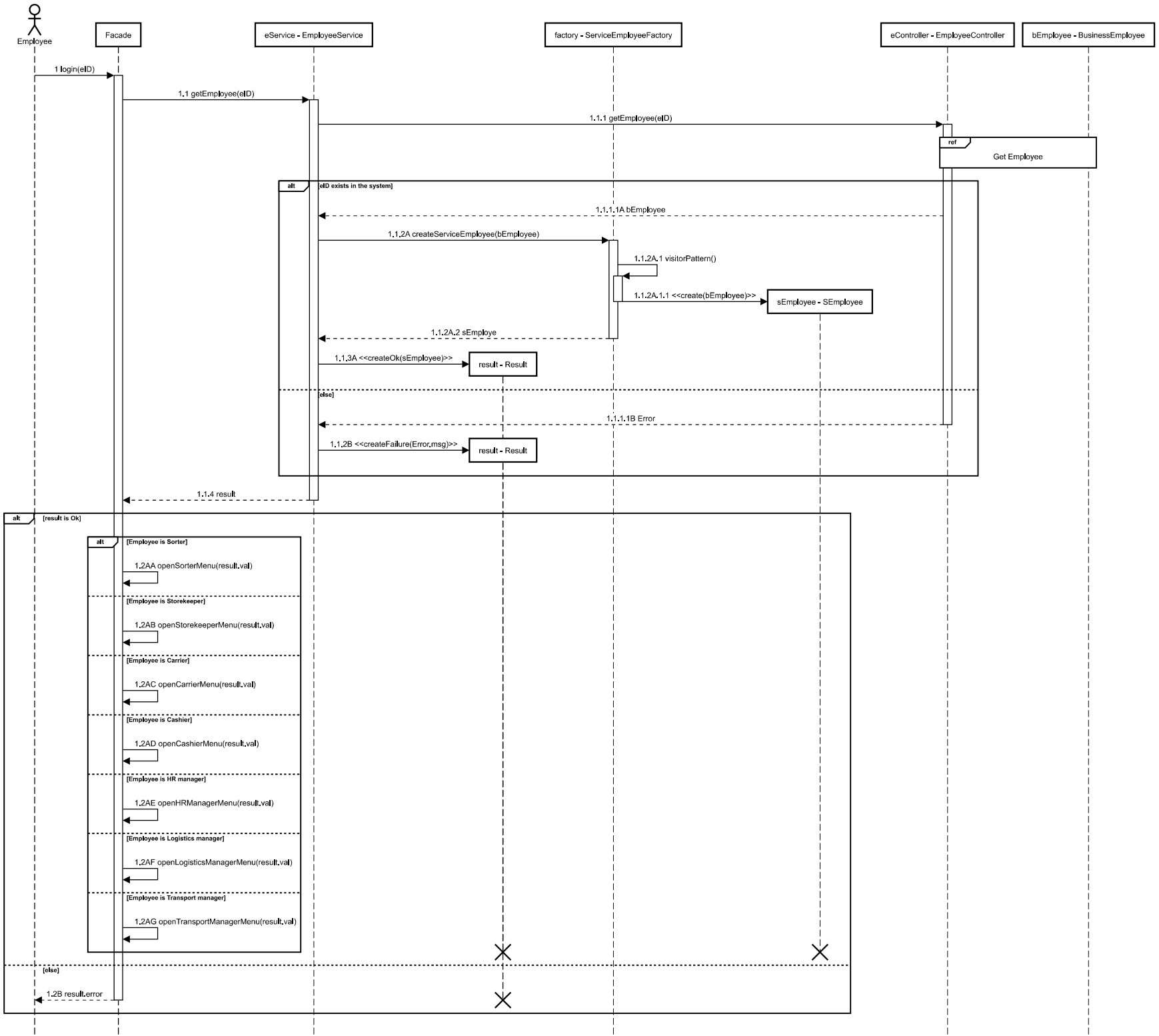
Get Employee



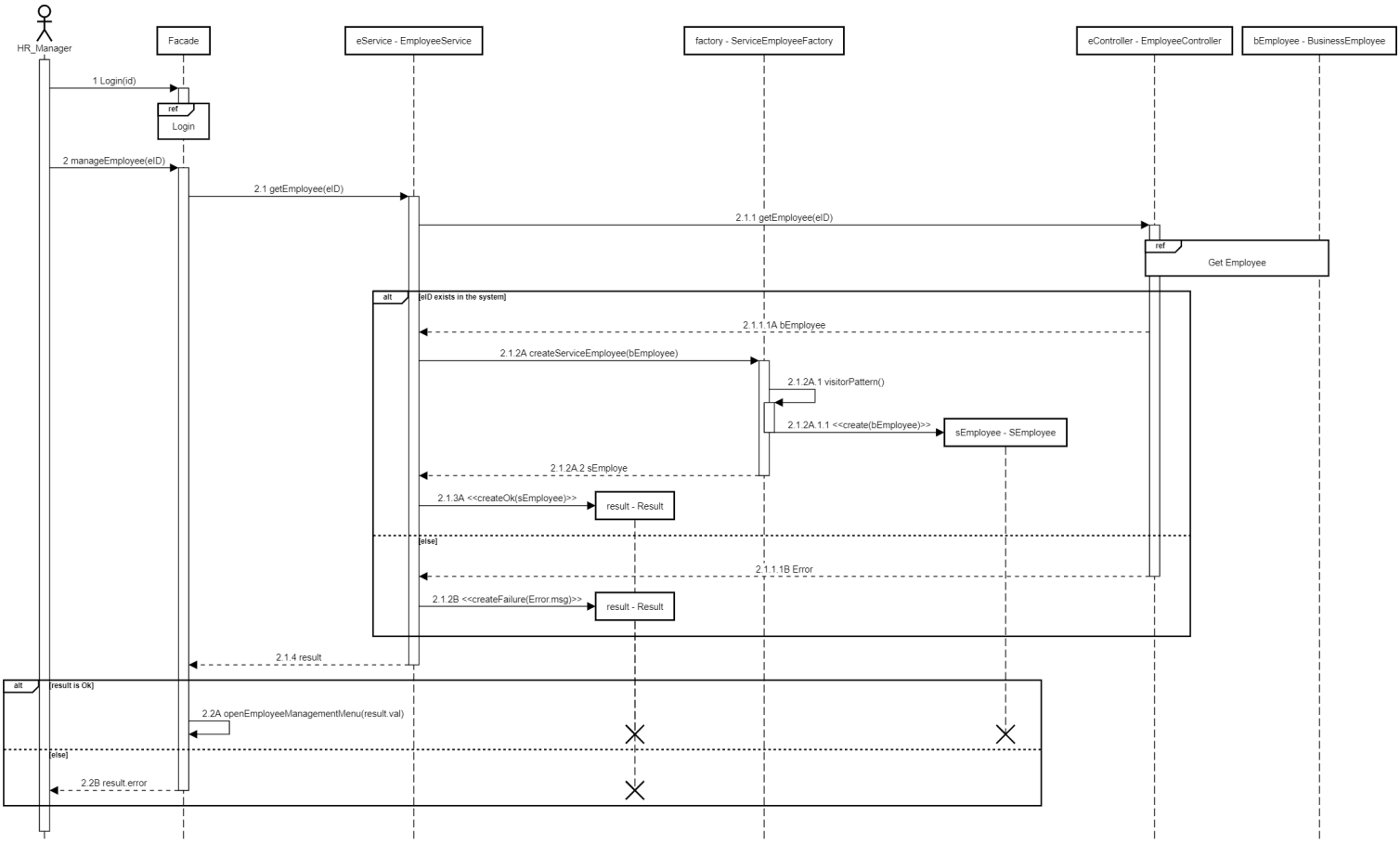
Get Shift


sc1-ShiftController
sdm1-ShiftDataMapper
stDataMapper-ShiftTypeDataMapper
s1-Shift
1: get(date,shiftType)
1.1: get(date)
1.1.1 get from the IdentityMap
alt
[shift existed in the identityMap]
1.1.2A: e1-Employee
1.2A: e1-Employee
1.1.2B select(date)
alt
[shift exist in the database]
1.2.3BA <<create>>
s2-Shift
1.2.4 BA: e2-Employee
1.2BA: e2-Employee
1.1.3 BB null
1.2BB: null

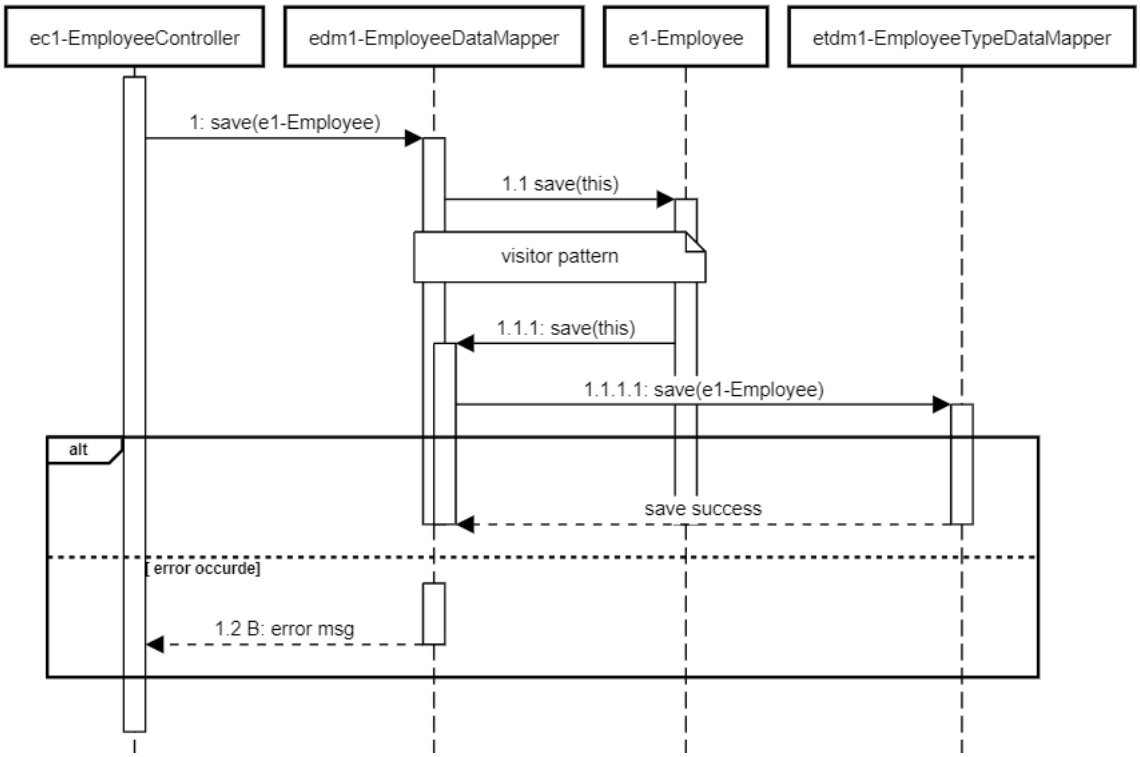
Login


Employee
Facade
eService - EmployeeService
factory - ServiceEmployeeFactory
eController - EmployeeController
bEmployee - BusinessEmployee
1 login(eID)
1.1 getEmployee(eID)
1.1.1 getEmployee(eID)
ref
Get Employee
alt
[eID exists in the system]
1.1.1.1A bEmployee
1.1.2A createServiceEmployee(bEmployee)
1.1.2A.1 visitorPattern()
1.1.2A.1.1 <<create(bEmployee)>>
sEmployee - SEmployee
1.1.2A.2 sEmploye
1.1.3A <<createOk(sEmployee)>>
result - Result
[else]
1.1.1.1B Error
1.1.2B <<createFailure(Error,msg)>>
result - Result
1.1.4 result
alt
[result is Ok]
alt
[Employee is Sorter]
1.2AA openSorterMenu(result.val)
[Employee is Storekeeper]
1.2AB openStorekeeperMenu(result.val)
[Employee is Carrier]
1.2AC openCarrierMenu(result.val)
[Employee is Cashier]
1.2AD openCashierMenu(result.val)
[Employee is HR manager]
1.2AE openHRManagerMenu(result.val)
[Employee is Logistics manager]
1.2AF openLogisticsManagerMenu(result.val)
[Employee is Transport manager]
1.2AG openTransportManagerMenu(result.val)
[else]
1.2B result.error

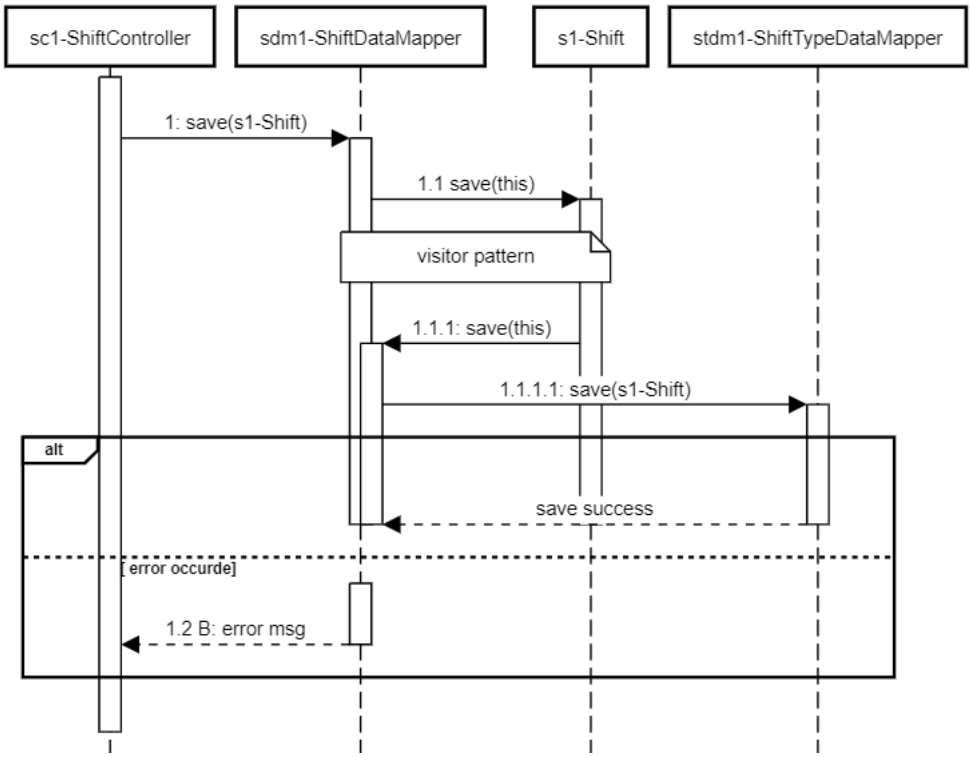
Opening Employee Management Menu



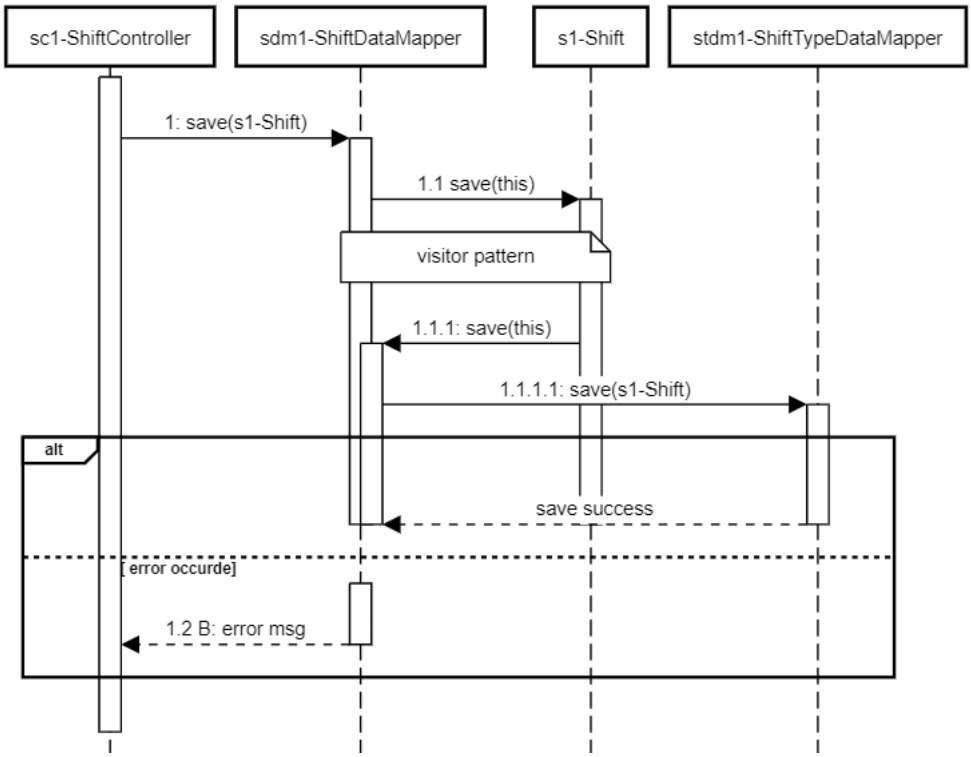
Save Employee In DB

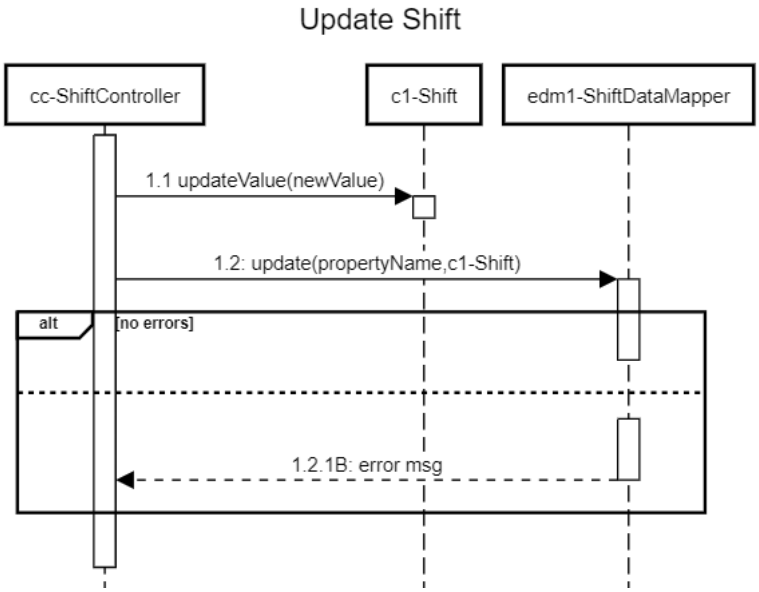


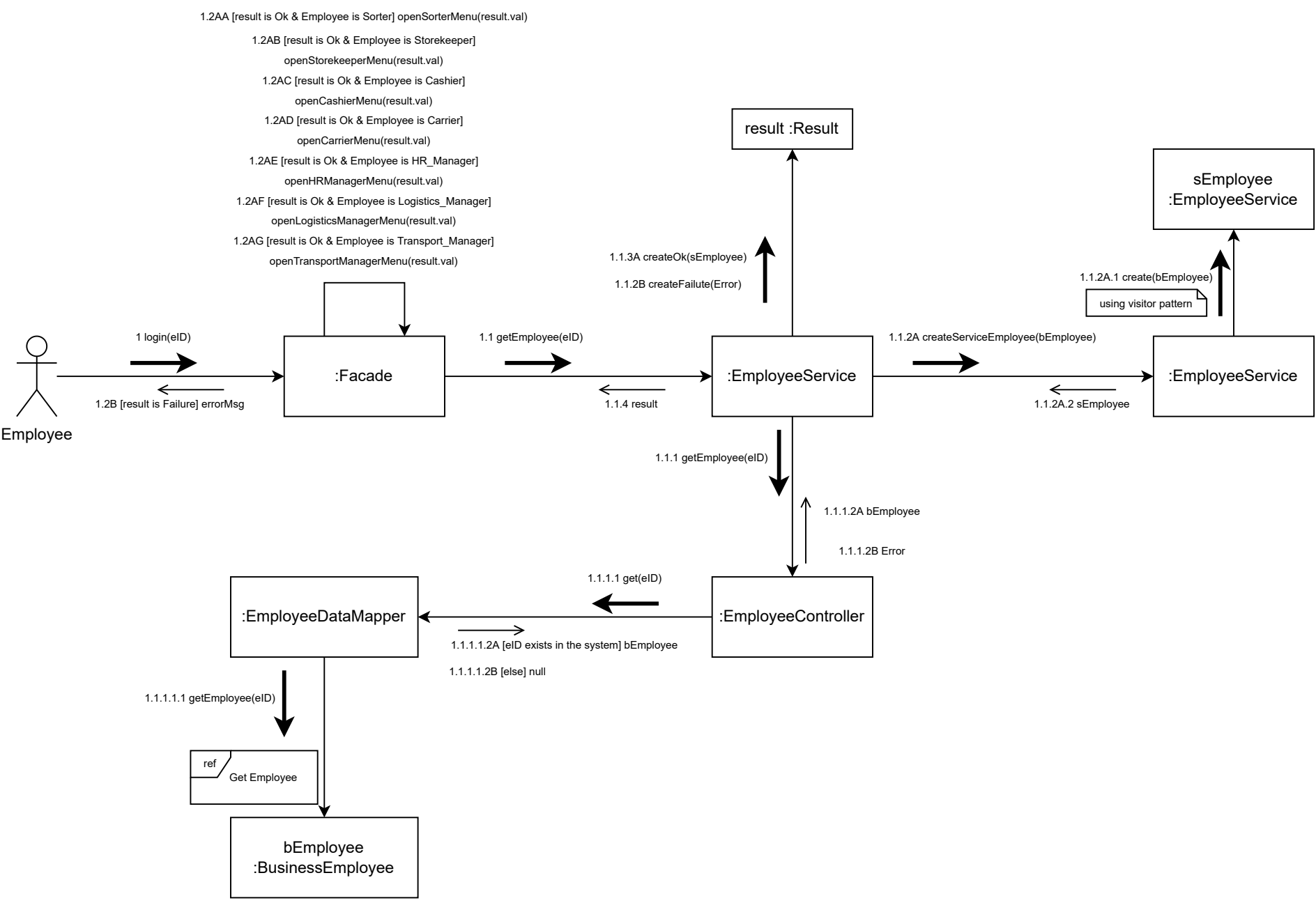
Save shift In DB

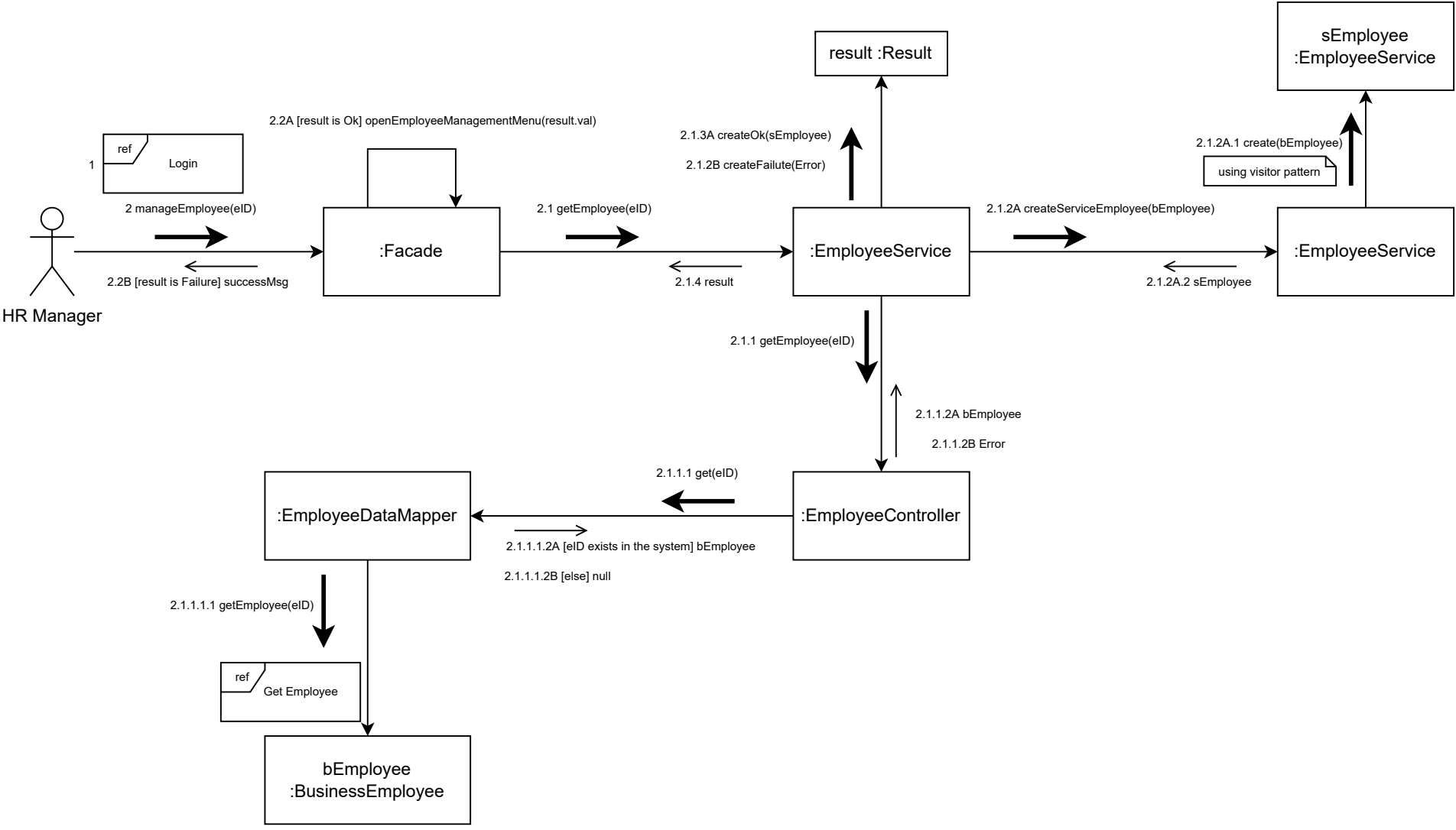


Save shift In DB




Update Shift
cc-ShiftController
c1-Shift
edm1-ShiftDataMapper
1.1 updateValue(newValue)
1.2: update(propertyName,c1-Shift)
alt
[no errors]
1.2.1B: error msg


1.2AA [result is Ok & Employee is Sorter] openSorterMenu(result.val)
1.2AB [result is Ok & Employee is Storekeeper]
openStorekeeperMenu(result.val)
1.2AC [result is Ok & Employee is Cashier]
openCashierMenu(result.val)
1.2AD [result is Ok & Employee is Carrier]
openCarrierMenu(result.val)
1.2AE [result is Ok & Employee is HR_Manager]
openHRManagerMenu(result.val)
1.2AF [result is Ok & Employee is Logistics_Manager]
openLogisticsManagerMenu(result.val)
1.2AG [result is Ok & Employee is Transport_Manager]
openTransportManagerMenu(result.val)
result :Result
sEmployee
:EmployeeService
1.1.3A createOk(sEmployee)
1.1.2B createFailute(Error)
1.1.2A.1 create(bEmployee)
using visitor pattern
1 login(eID)
1.2B [result is Failure] errorMsg
:Facade
1.1 getEmployee(eID)
1.1.4 result
:EmployeeService
1.1.2A createServiceEmployee(bEmployee)
1.1.2A.2 sEmployee
:EmployeeService
Employee
1.1.1 getEmployee(eID)
1.1.1.2A bEmployee
1.1.1.2B Error
1.1.1.1 get(eID)
:EmployeeDataMapper
:EmployeeController
1.1.1.1.2A [eID exists in the system] bEmployee
1.1.1.1.2B [else] null
1.1.1.1.1 getEmployee(eID)
ref
Get Employee
bEmployee
:BusinessEmployee


HR Manager
1
ref
Login
2 manageEmployee(eID)
2.2B [result is Failure] successMsg
2.2A [result is Ok] openEmployeeManagementMenu(result.val)
:Facade
2.1 getEmployee(eID)
2.1.4 result
result :Result
2.1.3A createOk(sEmployee)
2.1.2B createFailute(Error)
:EmployeeService
2.1.2A createServiceEmployee(bEmployee)
2.1.2A.2 sEmployee
:EmployeeService
2.1.2A.1 create(bEmployee)
using visitor pattern
sEmployee
:EmployeeService
2.1.1 getEmployee(eID)
2.1.1.2A bEmployee
2.1.1.2B Error
:EmployeeController
2.1.1.1 get(eID)
2.1.1.1.2A [eID exists in the system] bEmployee
2.1.1.1.2B [else] null
:EmployeeDataMapper
2.1.1.1.1 getEmployee(eID)
ref
Get Employee
bEmployee
:BusinessEmployee

managerId
carrierCount
cashierCount
hr_manager_count
storekeeperCout
Shift
date
sorterCount
Constraint
logisticManagerCount
(0,n)
(1,n)
can work in (constraint)
(0,n)
(1,1)
manager
(0,n)
works in
(1,n)
Is a
MorningShift
EveningShift
ID
Name
bank details
salary
employee conditions
Employee
starting date
certifications
Is a
Sorter
Carrier
Cashier
HR_Manager
logistic manager
Storekepper
transport_Manager
1
Licenses
ID
Weight
Transport ID
truck number
carrier name
Transport Document
1
1
Contain
Documentation
ID
Status
Truck Weight
Date
Transport
1
*
1
Placed
1
Truck
Contain
License Number
model
net weight
Max Capacity Weight
*
Destination Document
1
Documentation
ID
Transport ID
Weight
Date
Amount
Defect
ReportID
EmployeeID
DefectiveItem
StoreID
Description
ProductID
is detailed by
0,n
details
Shelves
is about
belongs to
parent category of
child category of
0,n
0,1
ProductID
Name
Location
Store
inWareHouse
ID
Name
Category
1
belongs to
CategoryID
is in
1
contains
0,n
Product
Weight
0,n
applies on
manufacturer
0,n
applies on
1
Price
describes order of
1
tracks
applies to
applies to
manages
is included in
0,n
Percent
0,n
Sale
is tracked by
Store
EndDate
is included in
0,n
StockReport
AmountInStore
minAmount
ID
StartDate
includes
AmountInWarehouse
targetAmount
FinalPrice
Discount
Weight
is ordered in
(0,n)
OrderItem
PricePerUnit
IDBySupplier
(1,n)
description
defectiveItems
Contains
Quantity
Manufacturer
Bank
ID
Address
PaymentAgreement
Name
AgreementType
Supplier
supplied by
(1,1)
(1,n)
(1,1)
Supplies
supplies
OrderID
(1,1)
(0,n)
Order
1
*
Phone
SupplierContacts
(0,n)
Has
StoreID
Name
CreationTime
SupplierID
Working Agreement
Status
ArrivalTime
Name
ProductID
AgreementItem
(1,1)
Agreement
Manufacturer
Day
PricePerUnit
Routine
ByOrder
SelfTransport
Time

Use Case Diagram:

Use Case Diagram detailing

the following use cases:

a. Registering new employee

b. Updating employee details and constraints

c.
Inventory updating and warning on low stock

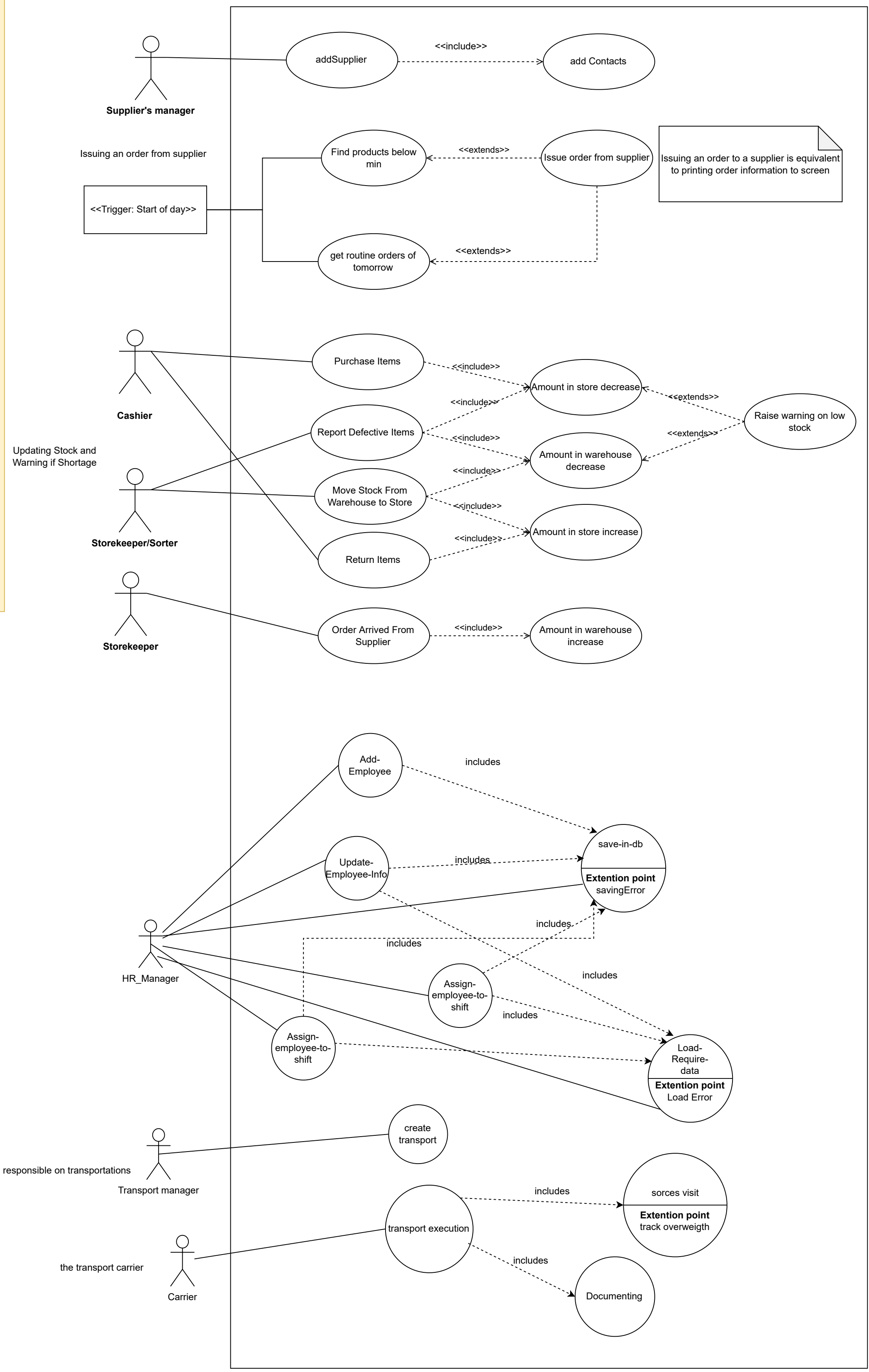
d.
Adding new supplier

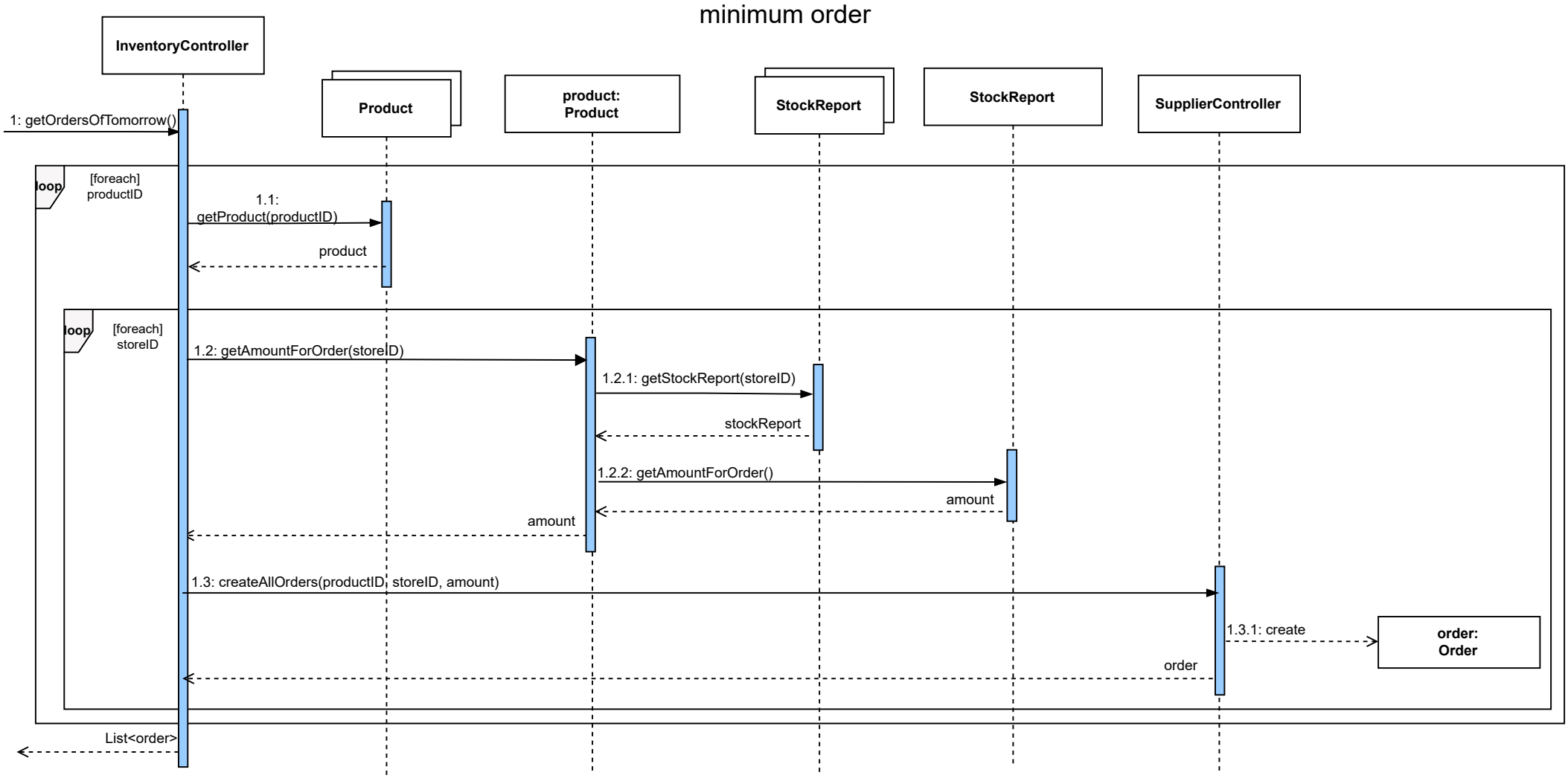
e. Carrying
out routine order from supplier

f.
Carrying out order due to low stock

g.
Assigning employee to shift

h.
Carrying out a transport





Use case name: Issuing an order from supplier due to lack of product in a specific store.

Textual description: The shift manager start the system at the beggining of each day, then all the orders of products under minimum amount are being issued to their corresponding supplier (printed on screen).

List of actors: supply manager/shift manager (the one that should see the orders report on the screen every morning).

Pre-conditions: The system must load the following data: suppliers and products

Post-conditions: The amount of the product decreased in the store (new amount=old amount-required amount<minimum amount => new order has been created and printed on screen). amountInDeliveries.contains(orderID, required amount)=true, in the corresponding stockReport.

Name: Orders On The Way

Textual description: this use case should be initiated once in a day, when the system is started and it returns to the inventory department all the orders that are scheduled for tomorrow, and all the orders requested by the inventory department due to shortage in certain items.

List of Actors: Storekeeper.

Pre-Conditions: requested items must be delivered by least one supplier.

Post-Conditions: inventory items on-the-way quantities are updated according to the orders and shortages are deleted.

Main success scenario (Pseudo-code):

User

1) Start the system at the beginning of the work day.
2) System orders automatically from suppliers.
3) The new orders are available to watch in Supplier System.

Inventory Model

startOfDay:

1) Find the items to order, the requested amount for each one and what store request this order.
2) Ask supplier model to order requested items and routine orders (call function createAllOrders).
3) Update all products amount in inventory.

Supplier Model

createAllOrders(mapOfShorage):

1) Declare List<Order> *finalOrders*.
2) Declare List<Order> *autoOrders*.
3) Load all existing routine orders from Data Base, add all to *finalOrders*.
4) For every routine supplier:
   a) If the supplier has a scheduled delivery for tomorrow, order from him the last order.
   b) Add order to *autoOrder*.
5) For every item in mapOfShortage:
   a) Search for the item in all the orders in *autoOrders*, and subtract the quantity ordered of this item to the requested store.
   b) Add any order contains the item to *finalOrders*, remove it from *autoOrders*.
   c) Update requested quantity of the item in mapOfShortage d) If the requested quantity for the item is 0, delete item from mapOfShortage.
6) For each item in mapOfShortage:
   a) Search for supplier that sell this item with the cheapest price. b) If there is an order from the supplier to the requested store in *finalOrders*: update the order to contain the item in the wished quantity. Else: If there is an order in *autoOrder*s from supplier, delete the order and make a new order for the item, add it to *finalOrders*. Else: make a new order for the item, add it to *finalOrders*.
7) Add all orders from *autoOrder*s to *finalOrders*.
8) Return *finalOrders*.

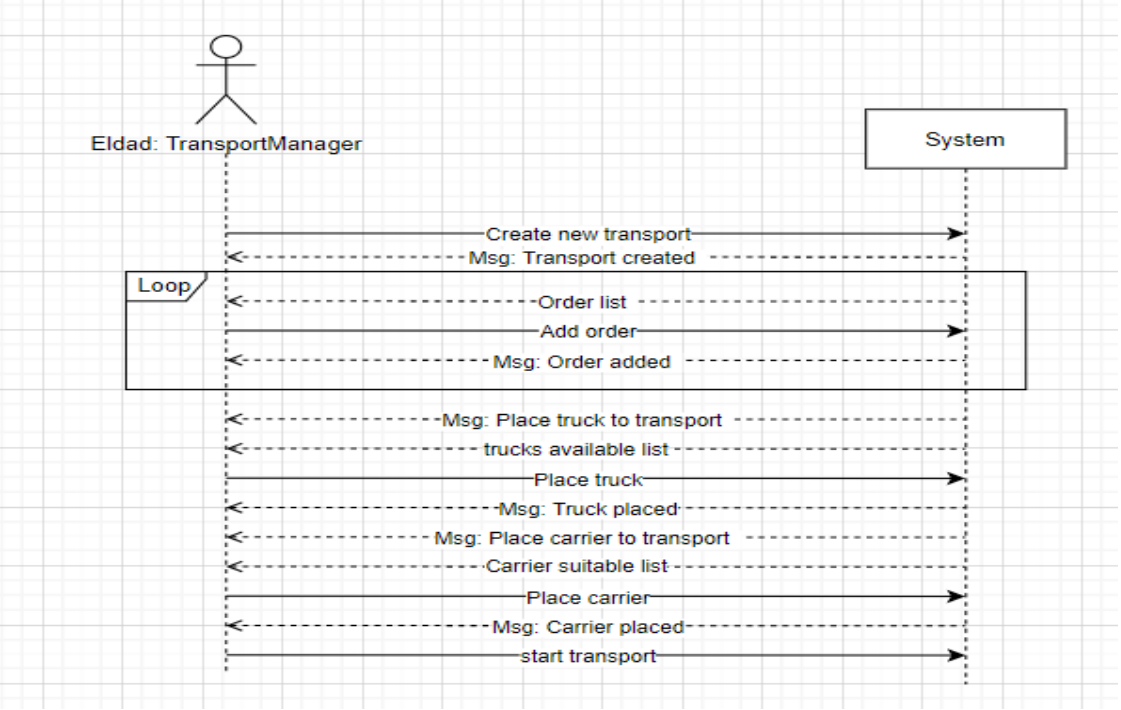
Alternative\Extantions:

a) System failure when initiated:
   1) System will present the error message to the user that booted the system.
   2) System will ask the user to restart it.

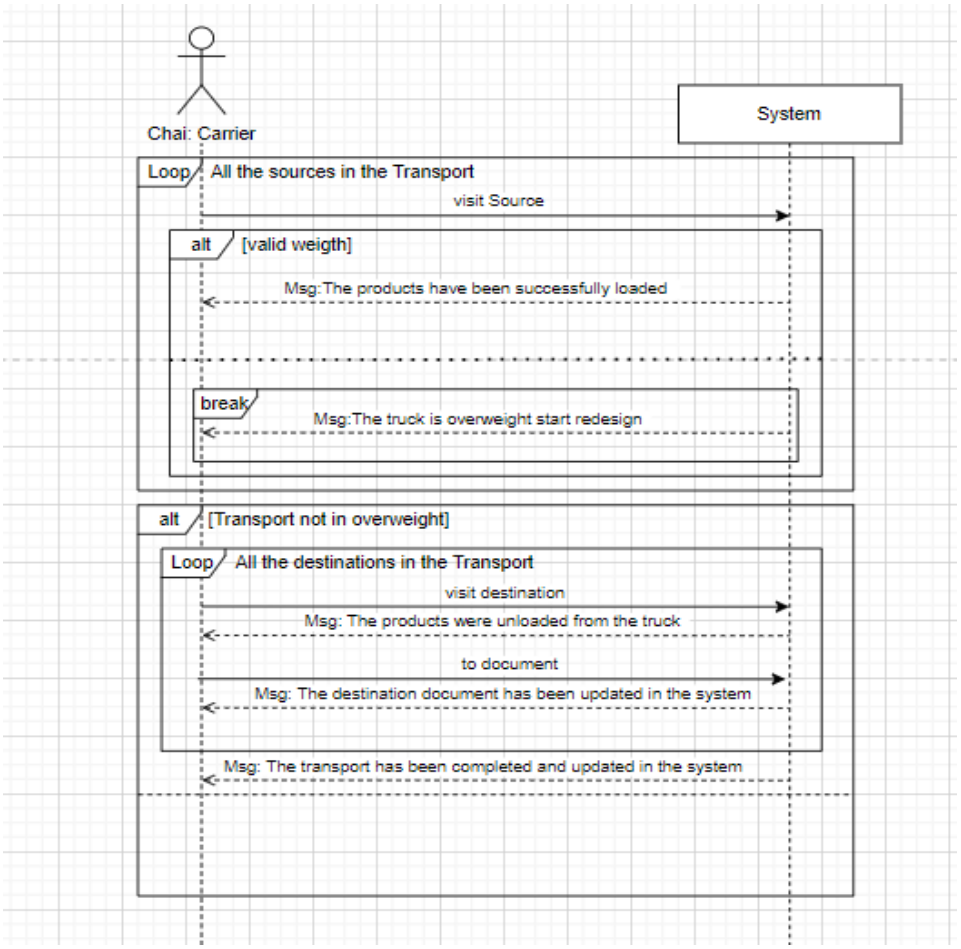
:Use case

Use case name	Carrying out its transport
Textual Description	Creating and carrying out its transport.
List of Actors	Transport manager and Carrier.
Pre-conditions	1. There is an order for transportation. 2. There is a truck and a carrier with a suitable license for this truck. 3. There are sources and destinations of transportation.
Post-conditions	1. In the completed transports archive the order exist. 2. The truck and driver will be available for other transports.
Main success scenario	1. Transport Manger create new transport. 2. Transport manager adds transport order to the transport. 3. Transport manager chooses truck for the transport. 4. Transport manager chooses driver that suitable to this transport. 5. The carrier updates the truck weight in each source. 6. The carrier updates about his visit in each destination, and the destination document will save in the archive. 7. When the carrier finishes his ride the transport will finish, and transport document will save in the archive.
Alternatives/Extensions	In case that the truck is in overweight the alert will send to the carrier and the redesign of the transport will be performed. The transport will end and start over. In the redesign of the transport the system will operate to reduce the weight of the truck so the transport will not be in over weight again.

sequence diagram for Transport manager with the system



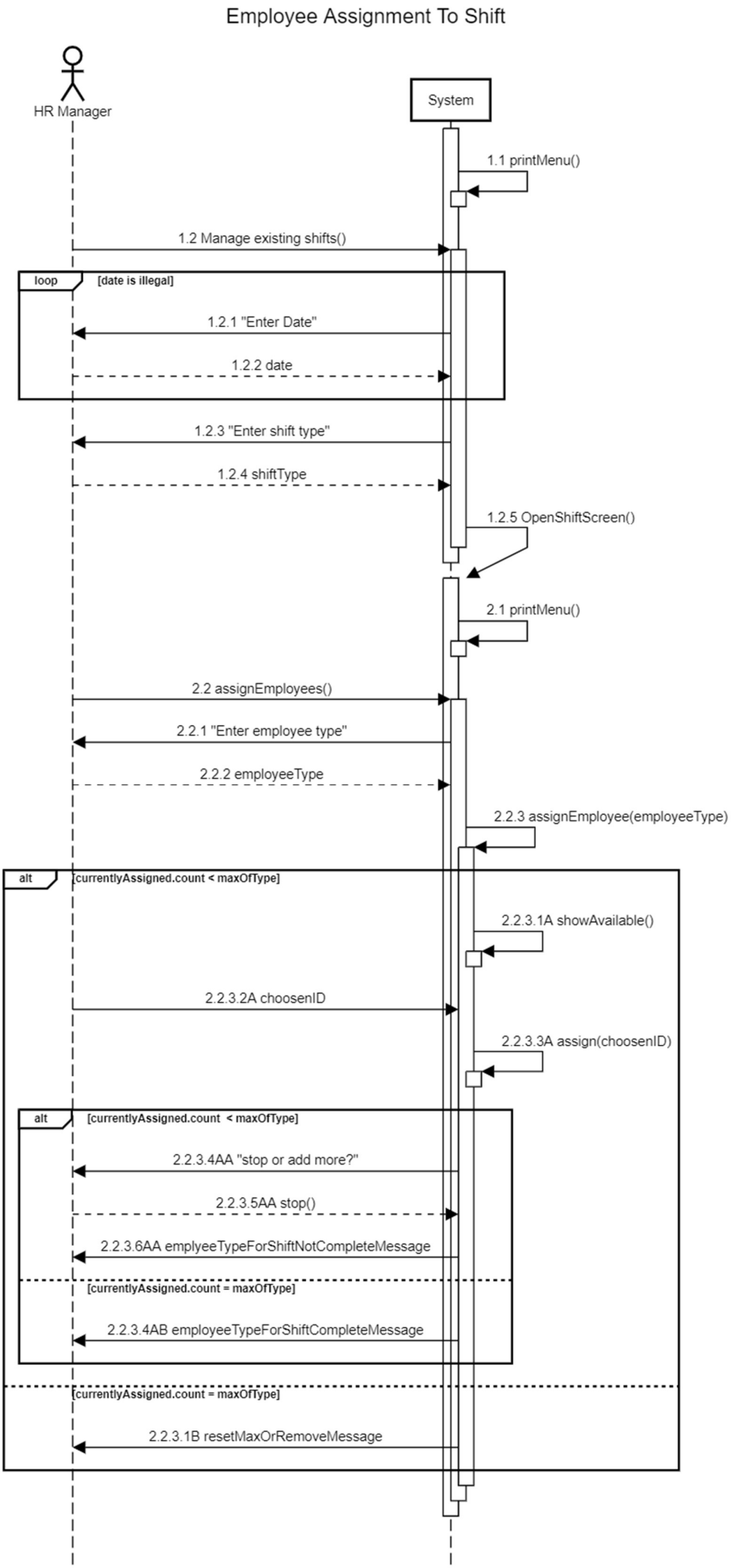
sequence diagram for Transport manager with the system

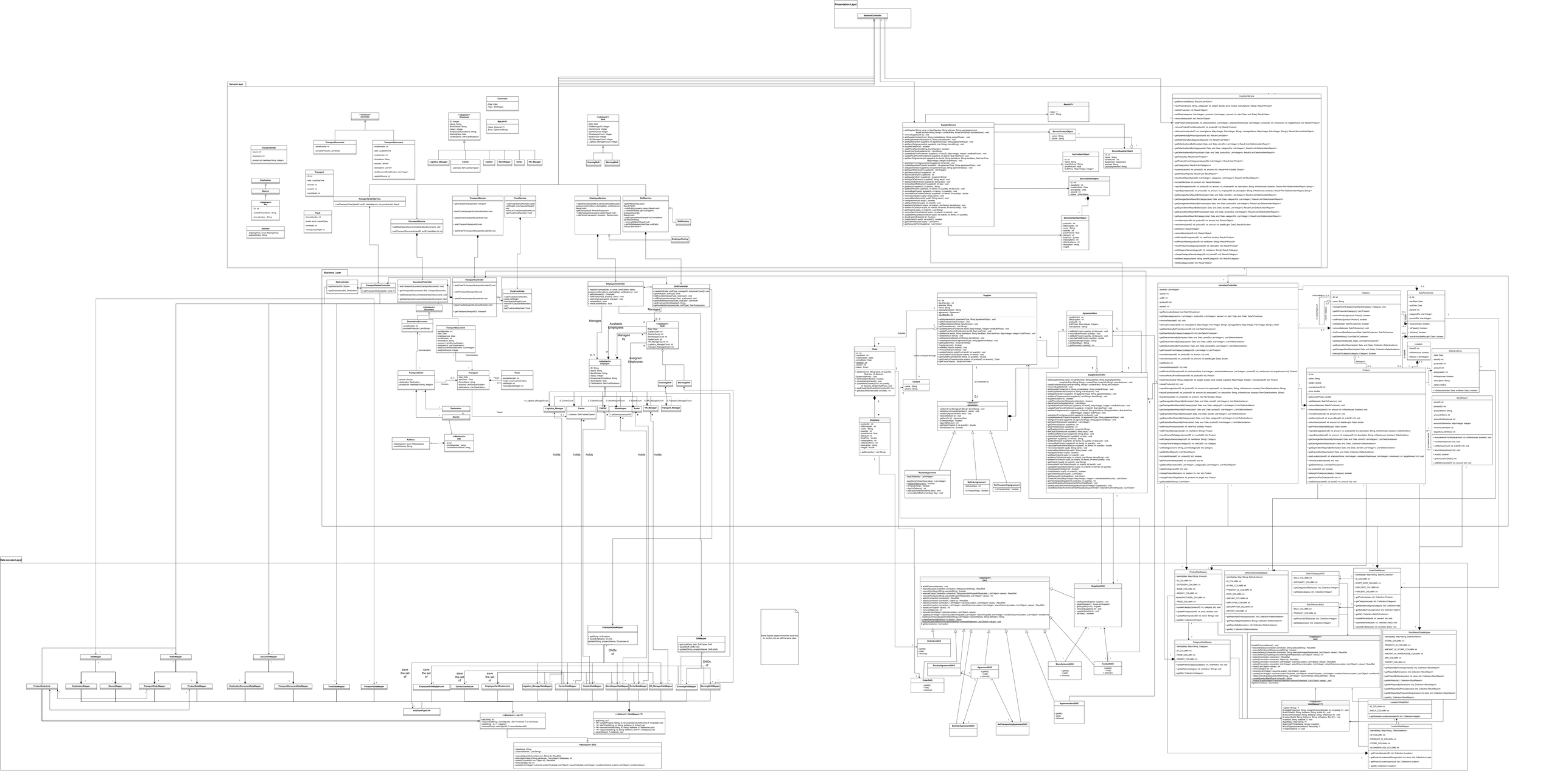


Detailed Use Cases:

Use case name	Assigning-Employee-to-Shift
Textual Description	The HR manager assigns a specific employee to a specific shift.
List of Actor	HR Manager
Pre-Conditions	The HR manager is in the ShiftMenu, The shift must exist in the system; The employee has to have a constraint matching shift date and type; The employee must not be assigned to the shift
Post-Conditions	The shift has the employee assigned to it; the database includes a record which details the assignment
Main success scenario	1) The HR manager chooses the shift through the ShiftMenu 2) The HR manager chooses to add to the shift employees of the type of the wanted employee 3) System validates number of already assigned employees of chosen type is less than set max number for this type 4) System shows available employees for chosen shift of chosen type 5) HR Manager chooses the employee out of the list of available employees 6) System checks new number of currently assigned to set max 7) HR Manager stops the assignment process
Alternative/Extensions	3') Number of assigned employees matches set number 4') System notifies HR manager of the situation and offers to reset the number of employees for the shift or remove already assigned employee
	7'') System stops the assignment process

When assigning an employee to a shift the system will interact with the manager in order to produce identifying data for the shift and the employee. The system will assign and save the assignment in the system if conditions are met.




Presentation Layer
BackendController
Service Layer
Business Layer
Data Access Layer
SupplierService
InventoryService
EmployeeService
ShiftService
TransportService
TruckService
DocumentService
SupplierController
InventoryController
EmployeeController
ShiftController
TransportController
TruckController
DocumentController
SiteController
Supplier
Product
Category
Discount
Location
Shift
Employee
Transport
Truck
Destination
Source
Document
Order
Contact
Contract
Address
Result<T>
ServiceContactObject
ServiceSupplierObject
ServiceSupplierItemObject
ServiceProductObject
ServiceDiscountObject
ServiceOrderObject
ServiceOrderItemObject
ServiceLocationObject
ServiceCategoryObject
ServiceDocumentObject
ServiceTransportObject
ServiceTruckObject
ServiceEmployeeObject
ServiceShiftObject
ServiceContractObject
ServiceSiteObject
ServiceDestinationObject
ServiceSourceObject
ServiceAddressObject
ServiceDriverLicenseObject
ServiceTruckTypeObject
ServiceTransportDocumentObject
ServiceDocumentItemObject
ServiceDocumentTypeObject
ServiceDocumentStatusObject
ServiceDocumentItemTypeObject
ServiceDocumentItemStatusObject
ServiceDocumentItemTypeObject
ServiceDocumentItemStatusObject
Data Access Layer
SupplierDAO
InventoryDAO
EmployeeDAO
ShiftDAO
TransportDAO
TruckDAO
DocumentDAO
SiteDAO
ContractDAO
OrderDAO
RoutineAgreementDAO
ByOrderAgreementDAO
NotTransportingAgreementDAO
SupplierDataMapper
InventoryDataMapper
EmployeeDataMapper
ShiftMapper
TransportDataMapper
TruckDataMapper
DocumentDataMapper
SiteDataMapper
ContractDataMapper
OrderDataMapper
RoutineAgreementDataMapper
ByOrderAgreementDataMapper
NotTransportingAgreementDataMapper
Some classes appear more than once only for comfort, but are still the same class.

Service Layer

Service-service
iService-truckS
transService-TransportS
DocService-DocumentS
sService-SiteS
orderService-orderS

Facade- facade

TruckService- truckS
controller-truckC

TransportService-TransportS
controller- transportC

DocumentService- DocumentS
controller- DocumentC

SiteService- SiteS
controller- siteC

OrderService- orderS
controller- orderC

ShiftService-shiftS

EmployeeService-employeeS

SupplierService

InventoryService

the srervice objects are not in the object diagdam because they always create and disappear so they are not in the memory

Business Layer

MorningShift1 - MorningShift
- Date: 27-05-2020
- ShiftManagerID: 5
- CarrierCount: 1
- CashierCount: 1
- StorekeeperCount: 1
- SorterCount: 1
- HR_ManagerCount: 1
- Logistics_ManagerCount: 0

EveningShift1 - EveningShift
- Date: 25-04-2022
- ShiftManagerID: 3
- CarrierCount: 1
- CashierCount: 1
- StorekeeperCount: 1
- SorterCount: 1
- HR_ManagerCount: 1
- Logistics_ManagerCount: 1

Olsk - Sorter
- ID: 11
- Name: Olsk Niv
- BankDetails: A654
- Salary: 2300
- EmploymentConditions: "ID: 11
Job Title: Sorter
Starting Date: 16-08-2020
Salary per shift: 20"
- Certifications: []

Rut - Cashier
- ID: 3
- Name: Rut Tif
- BankDetails: A123
- Salary: 2300
- EmploymentConditions: "ID: 3
Job Title: Cashier
Starting Date: 16-08-2020
Salary per shift: 30000"
- Certifications: []

Ein - Carrier
- ID: 13
- Name: Ein
- BankDetails:A 252
- Salary: 2300
- EmploymentConditions: "ID: 13
Job Title: Carrier
Starting Date: 16-08-2021
Salary per shift: 30000"
- StartingDate: 16-08-2021
- Certifications: []
- Licenses: [B,C]

Rubin - HR_Manager
- ID: 53
- Name: Rubin
- BankDetails: 252
- Salary: 23000
- EmploymentConditions: "ID: 53
Job Title: HR Manager
Starting Date: 16-08-2021
Salary per shift: 23000"
- StartingDate: 16-08-2021
- Certifications: []

Trunk1
licenseNumber-123456
model-B
netWeight-3
maxCapacityWeight-4
currWeight-3

available
assigned workers
drive on
carry the products

TruckController-truckC
TransportController-transportC
DocumentController-DocumentC
EmployeeController-employeeC
SiteController-siteC
OrderController-orderC
ShiftController- shiftC

Address- destAddress
shippingArea- south
exactAddress- Ashkelon Barnes1

Destination6-Ashkelon1
id-2
address-destAddress
contactName- Israel
phoneNumber-0527854321

is taking place in

Address- sourceAddress
shippingArea- south
exactAddress- TLV Hashfon 54

Source-TLV1
id-1
address-sourceAddress
contactName- Avi
phoneNumber-0501234567

TransportOrder-order
id - 1
src- TLV1
dst-Ashkelon1
products,list <product,100>

order the product

Products- product
id - 1
name- milk
weight-1

Data Access Layer

EmployeeDataMapper
DocumentMapper
SiteMapper
OrderMapper
TransportDataMapper
transport_Nr= (1, transport
TruckDataMapper
truck_M = [123456, 5]
DestinationDocumentDataMapper
destinationDoc_M= (<1,destdoc>)
TransportDocumentDataMapper
transportDoc_Nr= (<1,Docum)
DestinationMapper
destinations_M= (<2,Ashkelon1>)
SourceMapper
sources_M= (<1,TLV1>)
TransportOrderMapper
to_IM= (<1,order>)
ProductDataMapper
product_IM= (<1,product>)
ProductOrderLink
SiftMapper
+ getLocalDate: date, Shift[]:yyes Shift
+ save(Shift: shift):void
+ update(String :propertyName, Shift: shift)
EmployeeShiftMapperLink
CarrierLicensesLink
EmployeeCertificationLink
CarrierDataMapper
Carrier_IM = (13, Ein)
CashierDataMapper
Cashier_IM = (3, Rut)
SorterDataMapper
Sorter_M = (11, Olsk)
EveningShiftMapper
EveningShift_IM = (15/04/2022Evening, EveningShift1)
MorningShiftMapper
Morning_IM = [25/04/2022Morning, MorningShift1]
employerTypeLink
Logistics_ManagerDataMapper
Logistics_Manager_IM = (24, Oran)
StoreKeeperDataMapper
StoreKeeper_IM = (87, Stir)
SorterDataMapper
Sorter_M = (11, Olsk)
HR_ManagerDataMapper
HR_Manager_IM = (53, Rubin)

SupplierController

Avi_valbanan:Supplier
- id: 2
- bankNumber = 244556
- address: Vinged 40
- city: Supplier : 50
- payingAgreement : "Check"
- agreement : Agreement
- manufacturers : ["Dairy", "Eg

Lisr:Contact
- name : Lisr
- contact's phone : 0508644567

Holds
1
0..1

ByOrderAgreement
- deliveryDays : 3

provides

Tomato: AgreementItem
- proxPerUnit : 9.20
- productId : 13
- idBySupplier : 50
- bulkPrice : [100 : 5 , 200 : 20]
- manufacturer : Osem

Bamba:OrderItem
- productId : 13
- idBySupplier : 100
- name : "Tomato"
- amount : 20
- proxPerUnit :9.20
- discount : 0
- price : 184.0
- missingAmount : 0
- defectiveItems : 0
- description : ""
- weight = -1

contains
supplies

Order
- id : 1
- supplierId : 0
- creationDate : 23/4/2022
- arrivalDate : 26/4/2022
- price : 0
- status : waiting

<<abstract>>
DAO

SuppliersDAO
OrderItemDAO
OrderDAO
AgreementDAO
ManufacturersDAO
ContactDAO
ByOrderAgreementDAO
NotTransportingAgreementDAO
RoutineAgreementDAO
AgreementItemDAO

InventoryController: io
storeIds = [1,2,3,4,5,6]
storeID = 7
productID = 104
catID - 25
saleID = 17

stores
stored

Category: Vegetables
ID = 5
Name = Vegetables

belongs to
subcategories

Category: Organic
ID = 2
Name = Organic

sub-category
applier
records

Product: Tomato
id = 13
name = Tomato
weight = -1
manufacturerID = 13
price = 6.20

Product: Tomato
id = 4
name = Tomato
weight = -1
manufacturerID = 14
price = 7.20

taxes
manager
manages

Location: store73
StoreID = 4
inWarehouse = false
Shelves = 73

Location: warehouse45
StoreID = 4
inWarehouse = true
Shelves = 205

recorded
applied
SaleToCustomer: TomatoSale
id = 3
startDate = 30/04/2021
endDate = 07/05/2021
percent = 33
categoryIDs = []
productIDs = [4]

SaleToCustomer: OrganicSale
id = 3
startDate = 23/04/2022
endDate = 07/05/2022
percent = 10
categoryIDs = [2]
productIDs = []

StockReport: sr-5-4
storeID = 4
productID = 4
productName = Tomato
amountInStore = 37
amountInWarehouse = 89
minAmountInStore = 100
targetAmountInStore = 29

StockReport: sr-6-4
storeID = 6
productID = 4
productName = Tomato
amountInStore =20
amountInWarehouse = 26
minAmountInStore = 75
targetAmountInStore = 19

StockReport: sr-4-13
storeID = 4
productID = 13
productName = Tomato
amountInStore = 37
amountInWarehouse = 15
minAmountInStore = 100
targetAmountInStore = 29

StockReport: sr-5-13
storeID = 5
productID = 13
productName = Tomato
amountInStore = 17
amountInWarehouse = 29
minAmountInStore = 50
targetAmountInStore = 109

Location: store04
StoreID = 4
inWarehouse = false
Shelves = 72

Location: warehouse04
StoreID = 4
inWarehouse = true
Shelves = 204

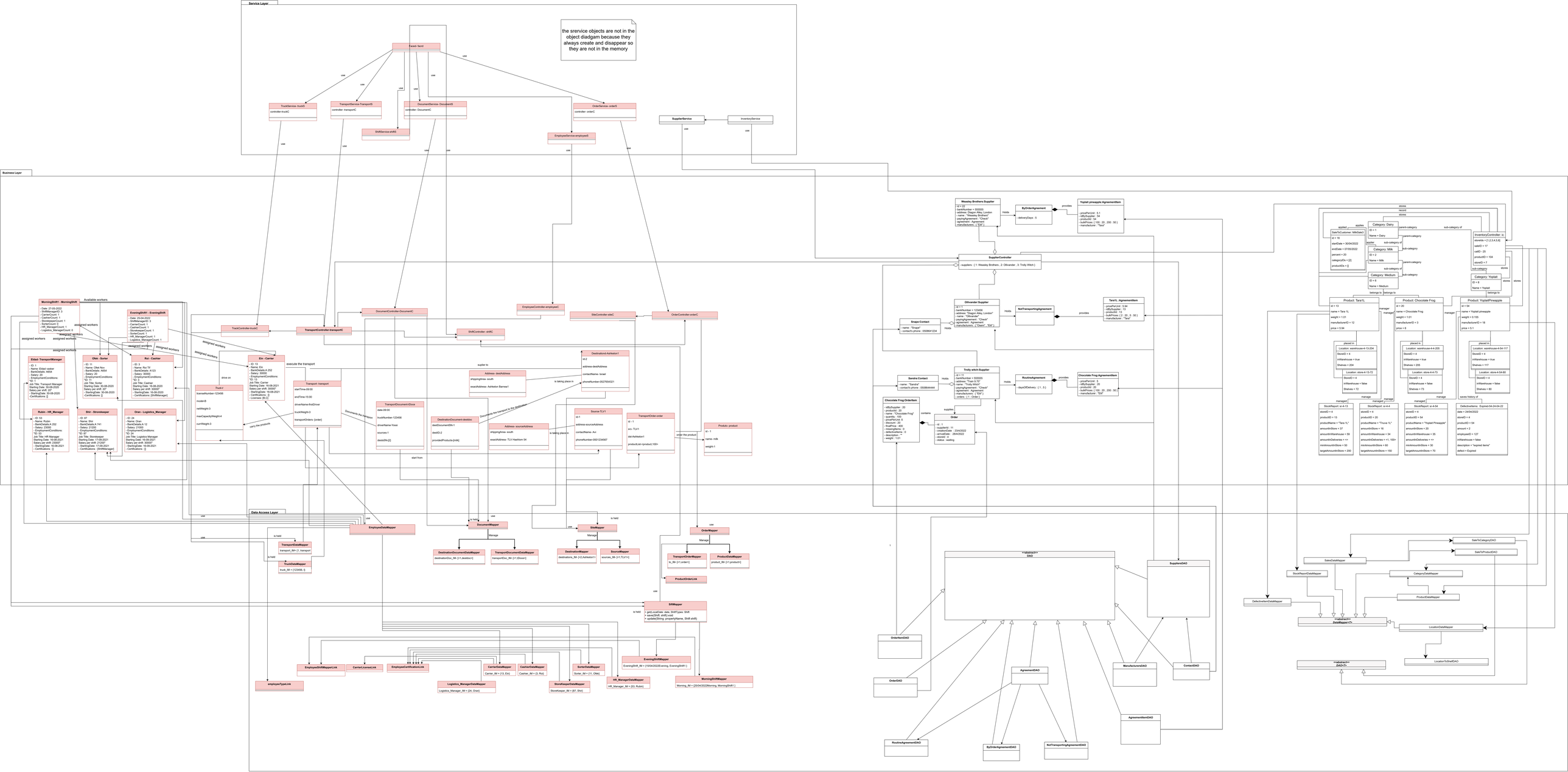
Location: warehouse05
StoreID = 5
inWarehouse = true
Shelves = 98

Location: store05
StoreID = 5
inWarehouse = false
Shelves = 62

DefectiveItems: Expired22
date = 26/04/2022
storeID = 5
productID = 4
amount = 2
employeeID = 127
description = "were forgotten in corner"
defect = Expired

DefectiveItems: Damaged22
date = 25/04/2022
storeID = 5
productID = 13
amount = 20
employeeID = 127
description = "box fell, mushy"
defect = Damaged

SaleToCategoryDAO
SaleToProductDAO
SalesDataMapper
StockReportDataMapper
CategoryDataMapper
ProductDataMapper
DefectiveItemDataMapper
<<abstract>>
DAO<T>
LocationDataMapper
Location/ToShelfDAO


Service Layer
the srervice objects are not in the object diadgam because they always create and disappear so they are not in the memory
Facad- facid
use
TruckService- truckS
controller-truckC
TransportService-TransportS
controller- transportC
DocumentService- DocumentS
controller- DocumentC
OrderService- orderS
controller- orderC
ShiftService-shiftS
EmployeeService-employeeS
SupplierService
InventoryService
Business Layer
MorningShift1 - MorningShift
- Date: 27-03-2022
- ShiftManagerID: 3
- CarrierCount: 1
- CashierCount: 1
- StorekeeperCount: 1
- SorterCount: 1
- HR Manager Count: 1
- Logistics_ManagerCount: 0
EveningShift1 - EveningShift
- Date: 25-04-2022
- ShiftManagerID: 3
- CarrierCount: 1
- CashierCount: 1
- StorekeeperCount: 1
- SorterCount: 1
- HR Manager Count: 1
- Logistics_ManagerCount: 1
Available workers
assigned workers
Eldad- TransportManager
- ID: 1
- Name: Eldad vasker
- BankDetails: 4654
- Salary: 20
- EmploymentConditions: TO: 1
Job Title: Transport Manager
Starting Date: 16-08-2020
Salary per shift: 20
- StartingDate: 30-08-2020
- Certifications: []
Ofek - Sorter
- ID: 11
- Name: Ofek Nov
- BankDetails: 9654
- Salary: 20
- EmploymentConditions: TO: 1
Job Title: Sorter
Starting Date: 30-08-2020
Salary per shift: 20
- StartingDate: 30-08-2020
- Certifications: []
Roi - Cashier
- ID: 3
- Name: Roi Tif
- BankDetails: 123
- Salary: 30000
- EmploymentConditions: TO: 24
Job Title: Cashier
Starting Date: 16-08-2020
Salary per shift: 30000
- StartingDate: 16-08-2020
- Certifications: [EveningAgent]
Rubin - HR_Manager
- ID: 53
- Name: Rubin
- BankDetails:A 252
- Salary: 23000
- EmploymentConditions: TO: 53
Job Title: HR Manager
Starting Date: 17-09-2021
Salary per shift: 23000
- StartingDate: 16-08-2021
- Certifications: []
Shir - Storekeeper
- ID: 87
- Name: Shir
- BankDetails:A 741
- Salary: 21200
- EmploymentConditions: TO: 87
Job Title: Storekeeper
Starting Date: 17-09-2021
Salary per shift: 21200
- StartingDate: 17-09-2021
- Certifications: [ShiftManager]
Oran - Logistics_Manager
- ID: 24
- Name: Oran
- BankDetails:A 12
- Salary: 21000
- EmploymentConditions: TO: 24
Job Title: Logistics Manager
Starting Date: 16-08-2021
Salary per shift: 35000
- StartingDate: 16-08-2021
- Certifications: []
Data Access Layer
EmployeeDataMapper
TransportDataMapper
transport_IMA {1, transport
TruckDataMapper
truck_IM = {123456, {}
DocumentMapper
Manage
DestinationDocumentDataMapper
destinationDoc_IM- {<1,destdoc>}
TransportDocumentDataMapper
transportDoc_IM- {<1,tDocer>}
SiteMapper
Manage
DestinationMapper
destinations_IM- {<2,Ashkelon1>
SourceMapper
sources_IM- {<1,TLV1>}
OrderMapper
Manage
TransportOrderMapper
to_IM- {<1,order>}
ProductDataMapper
product_IM- {<1,product>}
ProductOrderLink
ShiftMapper
+ getLocalDate- date, ShiftTypes: Shift
+ saveShift(Shift, shift):void
+ updateString: propertyName, Shift shift)
EmployeeShiftMapperLink
CarrierLicensesLink
EmployeeCertificationLink
employeeTypeLink
CarrierDataMapper
Carrier_IM = {3, Ein}
CashoDataMapper
Cashier_IM = {3, Roi}
Logistics_ManagerDataMapper
Logistics_Manager_IM = {24, Oran}
SorterDataMapper
Sorter_IM = {11, Ofek}
StoreKeeperDataMapper
StoreKeeper_IM = {87, Shir}
EveningShiftMapper
EveningShift_IM = {15/04/2022Evening, EveningShift1 }
HR_ManagerDataMapper
HR_Manager_IM = {53, Rubin}
MorningShiftMapper
Morning_IM = {25/04/2022Morning, MorningShift1 }
Weasley Brothers Supplier
- id = 20
- address: Diagon Alley, London"
- name : "Weasley Brothers"
- payingAgreement : "Check"
- agreement : Agreement
- manufacturers : ["Gryf]
Holds
provides
ByOrderAgreement
- deliveryDays : 5
Yoplait pineapple AgreementItem
- prisePerUnit : 5.1
- idBySupplier : 54
- productId : 54
- bulkPrices : [100 , 20 , 200 : 50]
- manufacturer : "Tara"
SupplierController
- suppliers : [1: Weasley Brothers , 2: Olivander , 3: Trolly Witch]
Olivander Supplier
- id = 1
- bankNumber = 123456
- address: "Diagon Alley, London"
- name : "Olivander"
- payingAgreement : "Check"
- agreement : Agreement
- manufacturers : ["Dawn" , "Eld"]
NotTransportingAgreement
Tara1L -AgreementItem
- prisePerUnit : 5.94
- idBySupplier : 13
- productId : 13
- bulkPrices: [2 , 20 , 5 : 50]
- manufacturer : "Tara"
Snape-Contact
- name : "Snape"
- contact's phone : 0509841234
Holds
Trolly witch.Supplier
- id = 11
- bankNumber = 555555
- address: "Tsar, 9 70"
- name : "Trolly Witch"
- payingAgreement : "Check"
- agreement : Agreement
- manufacturers : ["Eld"]
- orders : [1 , Order]
RoutineAgreement
- daysOfDelivery : [1 , 5]
Chocolate Frog AgreementItem
- prisePerUnit : 9
- idBySupplier : 20
- productId : 20
- bulkPrices: [100 , 20 , 200 : 50]
- manufacturer : "Elit"
Sandra-Contact
- name : "Sandra"
- contact's phone : 0505864444
Chocolate Frog.OrderItem
- idBySupplier : 20
- name : "Chocolate Frog"
- quantity : 100
- prisePerUnit : 5
- total : 500
- finalPrice : 400
- missingItems : 0
- defectiveItems : 0
- description : ""
- weight : 1.01
contains
supplies
Order
- id : 1
- supplierId : 11
- orderDate : 23/4/2022
- arrivalDate : 25/4/2022
- storeId : 4
- status : waiting
stores
records
Category: Dairy
ID = 1
Name = Dairy
Category: Milk
ID = 2
Name = Milk
Category: Medium
ID = 6
Name = Medium
Category: Yoplait
ID = 8
Name = Yoplait
applied
SaleToCustomer: MilkSale3
- id = 16
- startDate = 30/04/2022
- endDate = 07/05/2022
- percent = 20
- categoryIDs = [2]
- productIDs = []
InventoryController: ic
- storeIds = [1,2,3,4,5,6]
- saleID = 17
- catID - 25
- productID = 104
- storeID = 7
belongs to
Product: Tara1L
- id = 13
- name = Tara 1L
- weight = 1.01
- manufacturerID = 12
- price = 5.94
Product: Chocolate Frog
- id = 20
- name = Chocolate Frog
- weight = 1.01
- manufacturerID = 3
- price = 6
Product: YoplaitPineapple
- id = 54
- name = Yoplait pineapple
- weight = 0.155
- manufacturerID = 18
- price = 5.1
placed in
Location: warehouse-4-13-204
StoreID = 4
inWarehouse = true
Shelves = 204
Location: store-4-13-72
StoreID = 4
inWarehouse = false
Shelves = 72
Location: warehouse-4-4-205
StoreID = 4
inWarehouse = true
Shelves = 205
Location: store-4-4-73
StoreID = 4
inWarehouse = false
Shelves = 73
Location: warehouse-4-54-117
StoreID = 4
inWarehouse = true
Shelves = 117
Location: store-4-54-80
StoreID = 4
inWarehouse = false
Shelves = 80
manager
managed
StockReport: w-4-13
storeID = 4
productID = 13
productName = "Tara 1L"
amountInStore = 37
amountInWarehouse = 58
amountInDeliveries = +x
minAmountInStore = 50
targetAmountInStore = 200
StockReport: w-4-4
storeID = 4
productID = 20
productName = "Truva 1L"
amountInStore = 18
amountInWarehouse = 34
amountInDeliveries = +1, 100=
minAmountInStore = 60
targetAmountInStore = 150
StockReport: w-4-54
storeID = 4
productID = 54
productName = "Yoplait Pineapple"
amountInStore = 20
amountInWarehouse = 35
amountInDeliveries = +x
minAmountInStore = 30
targetAmountInStore = 70
DefectiveItems: Expired-54-24-04-22
date = 24/04/2022
storeID = 4
productID = 54
amount = 2
employeeID = 127
inWarehouse = false
description = "expired items"
defect = Expired
saves history of
SaleToCustomer
SaleToCategoryDAO
SaleToProductDAO
SalesDataMapper
StockReportDataMapper
CategoryDataMapper
ProductDataMapper
DefectiveItemDataMapper
LocationDataMapper
LocationToShelfDAO
<<abstract>>
DAO<T>
SuppliersDAO
OrderItemDAO
OrderDAO
AgreementDAO
ManufacturersDAO
ContactDAO
AgreementItemDAO
RoutineAgreementDAO
ByOrderAgreementDAO
NotTransportingAgreementDAO
Truck-r
licenseNumber-123456
model-0
netWeight-3
maxCapacityWeight-4
currWeight-3
drive on
Available workers
assigned workers
Eln - Carrier
- ID: 13
- Name: Eln
- BankDetails:A 252
- Salary: 30000
- EmploymentConditions: TO: 13
Job Title: Carrier
Starting Date: 16-08-2021
Salary per shift: 30000
- StartingDate: 16-08-2021
- Certifications: [B,C2]
- Licenses: [B,C2]
execute the transport
Transport- transport
startTime-09:00
endTime-15:00
driveName-firstDriver
truckWeight-3
transportOrders- [order]
carry the products
Documents the transport
TransportDocument-tDoce
date-09:00
truckNumber-123456
driveName-Yossi
sources-1
destsSN-[2]
Documents the transport to the destination
DestinationId-Ashkelon1
id-2
address-destAddress
-contactName- Israel
phoneNumber-0527654321
is taking place in
Address- destAddress
shippingArea- south
exactAddress- Ashkelon Barnea1
supplier to
DestinationDocument-destdoc
destDocumentSN-1
destID-2
providedProducts-[milk]
Address- sourceAddress
shippingArea- south
exactAddress- TLV Hashkon 54
is taking place in
Source-TLV1
id-1
address-sourceAddress
contactName- Avi
phoneNumber-0501234567
TransportOrder-order
id - 1
dst-Ashkelon1
productList->product,100>
order the product
Produto- product
id - 1
name- milk
weight-1
start from