

פרוייקט סופר-לי

מודול ספקים ומלאי

מגישים:

יאיר בר-יוסף 214267874

נועה ארגמני 318859014

ילנה נסירוב 321790891

אור דהן 213390685

טבלה 1: דרישות

ID	מודל	פונקציונלי / לא פונקציונלי	דרישה	Priority	Risk	Status
1.	מלאי	פונקציונלי	על המערכת לשלוח התראות מקדימות עבור מוצרים שעומדים להיגמר במלאי.	NTH	Low	Done
2.	מלאי	פונקציונלי	על המערכת לאפשר להפיק דוחות על הפריטים הקיימים במלאי.	MH	Low	Done
3.	מלאי	פונקציונלי	כל דוח על הפריטים הקיימים במלאי שהמערכת צריכה לספק יכולול את המספר הקטלוגי, שם המוצר, היצרן של כל פריט, הכמות הנוכחית של כל פריט, הכמות במדפים והכמות במחסן.	MH	Low	Done
4.	מלאי	פונקציונלי	על המערכת לספק דוח על פי קטגוריה או קטגוריות.	MH	Low	Done
5.	מלאי	פונקציונלי	כל דוח לפי קטגוריה שהמערכת צריכה לספק יכולול שמות הקטגוריה ואת המידע של כל הפריטים הנמצאים תחת אותה קטגוריה.	NTH	Low	Done
6.	מלאי	פונקציונלי	על המערכת לספק דוחות של פריטים פגומים.	MH	Low	Done
7.	מלאי	פונקציונלי	כל דוח של פריטים פגומים שהמערכת צריכה לספק יכולול את המספר הייחודי של הפריט, שם הפריט, הקטגוריה, סוג הפגם, אם המוצר פג תוקף אז את תאריך התפוגה שלהם	MH	Low	Done
8.	מלאי	פונקציונלי	על המערכת לאפשר לספק דוחות עבור מוצרים חסרים.	NTH	Low	Done
9.	מלאי	פונקציונלי	על המערכת לאפשר הוספה ומחיקה של פריטים מהמלאי.	MH	Low	Done
10.	מלאי	פונקציונלי	על המערכת לספק דוח של מבצעים.	MH	Low	Done
11.	מלאי	פונקציונלי	כל דוח של מבצעים שהמערכת צריכה לספק יכולול את שם המבצע, אחוז ההנחה, תאריך ההתחלה ותאריך הסיום של המבצע.	NTH	Low	Done
12.	מלאי	פונקציונלי	על המערכת לתמוך באחסון פריטים במחסן ובמדפי החנות בלבד.	MH	Low	Done
13.	מלאי	פונקציונלי	על המערכת לאפשר עדכון על פגימות המוצר.	MH	Low	Done
14.	מלאי	פונקציונלי	ברגע שהנחה מסוימת נגמרת, אז היא תימחק מהמערכת.	MH	High	Done
15.	מלאי	פונקציונלי	על המערכת לאפשר לקבוע את הסף לכמות המינימלית עבור כל מוצר.	MH	Low	Done
16.	מלאי	פונקציונלי	על המערכת לאפשר הוספה ומחיקה של פריטים מקטגוריה.	NTH	Low	Done
17.	ספקים	לא פונקציונלי	המערכת צריכה לאפשר שמירה של קטגוריות המוצרים מהן החברה מזמינה מכל ספק.	NTH	Low	Done
18.	ספקים	פונקציונלי	המערכת תתמוך באפשרות שמספר ספקים יוכלו לספק מאותו יצרן.	MH	Low	Done
19.	ספקים	פונקציונלי	המערכת צריכה לשמור לכל ספק: שם ספק, מספר ספק, כתובת ואת שמות היצרנים איתם	NTH	Low	Done

הוא עובד.						
20.	ספקים	פונקציונלי	המערכת תתמוך בכך שספק מסוים יוכל לספק מספר מוצרים מיצרנים שונים	MH	Low	Done
21.	ספקים	פונקציונלי	על המערכת לשמור כרטיס ספק לכל ספק הרשום במערכת.	MH	Low	Done
22.	ספקים	פונקציונלי	המערכת תשמור בכרטיס הספק את מספר ח"פ, חשבון בנק, ותנאי תשלום של הספק.	MH	Low	Done
23.	ספקים	פונקציונלי	המערכת תשמור בכרטיס הספק את שמות אנשי הקשר של הספק ופרטי ההתקשרות עמם.	NTH	Low	Done
24.	ספקים	פונקציונלי	המערכת תתמוך בכך שלחברה יהיו הסכמים שונים עם ספקים שונים ועם תנאים עסקיים (תנאי תשלום, כתבי כמויות) שונים.	MH	Low	Done
25.	ספקים	פונקציונלי	המערכת חייבת להיות מסוגלת לשמור את מספר הימים הקבוע לסיפוק, (או כלום, אם אין כאלה).	MH	Low	Done
26.	ספקים	פונקציונלי	המערכת חייבת לשמור הסכם עם כל ספק המתאר: ימי סיפוק מוצרים, תאריך חתימת ההסכם, פרטי פריטים הכלולים בהסכם, כמספר קטלוגי, סוג, כמות, מחיר לכל פריט ו"כתב כמויות" (אם קיים).	MH	Low	Done
27.	ספקים	פונקציונלי	המערכת תנהל לכל פריט: ספק, מספר קטלוגי של הספק, מחיר מחירון, הנחה בעת רכישות בהיקף גדול ("כתב כמויות").	NTH	Low	Done
28.	ספקים	פונקציונלי	המערכת חייבת להיות מסוגלת לשמור "כתב כמויות" (אם קיים), לכל ספק, שיתאר הנחות על מוצר, בהתאם ל: כמות המוצר שנקנה בהזמנה, מספר מוצרים מינימלי בהזמנה, עלות כוללת של הזמנה ועוד סוגי הנחות נוספים על פי דרישה.	MH	Low	Done
29.	ספקים	פונקציונלי	המערכת חייבת לתמוך בניהול הזמנות מספקים, כאשר כל הזמנה תכלול: מס' הזמנה, שם ספק, מס' ספק, תאריך הזמנה, פריטים, כמות לכל פריט, מחיר כולל ומצב הזמנה.	MH	Low	Done
30.	ספקים	פונקציונלי	המערכת חייבת לתמוך בעריכה של הסכם עם הספק.	MH	Low	Done
31.	ספקים	פונקציונלי	המערכת חייבת תמיד לעשות את ההזמנה הכוללת שתעלה הכי פחות לפי מחיר הזמנה סופי.	NTH	High	Done
32.	ספקים	פונקציונלי	על המערכת לעדכן את המשתמש, על כשלון ביצוע עסקה, כאשר כשלון ביצוע עסקה יקרה במידה ויש חוסר במלאי של הספקים, אי תקינות בהזמנה, כשלון מערכת, כל מקרה אחר בו העסקה נכשלה.	NTH	High	Done
33.	ספקים	לא פונקציונלי	על המערכת לעדכן את המשתמש במקרה של כשלון מערכת.	NTH	Low	Done
34.	ספקים	לא פונקציונלי	על המערכת לאפשר חזרה לפעילות רגילה לאחר תקלות או מקרי קיצון, (recovery).	MH	High	Done
35.	ספקים	לא	מצב המערכת צריך להיות ניתן לשחזור לאחר	MH	High	Done

			פונקציונלי	סגירה (יזומה או בלתי יזומה), (persistence).
36.	ספקים	פונקציונלי	המערכת חייבת לתמוך בביטול של הסכמים.	Done Low MH
37.	מלאי+ ספקים	פונקציונלי	על המערכת לבצע הזמנה אוטומטית מספק עקב חוסר במוצר כלשהו.	Done High MH
38.	מלאי+ ספקים	פונקציונלי	על מנת להוציא הזמנה עקב חוסר על המערכת לקבל מהמשתמש את פרטי המוצר ממערכת המלאי ואת הכמות הנדרשת.	Done Low MH
39.	מלאי+ ספקים	פונקציונלי	על המערכת לבצע הזמנה אוטומטית מספק באופן תקופתי.	Done High MH
40.	ספקים	פונקציונלי	על המערכת לספק הזמנה תקופתית בזמני הגעה קבועים.	Done Low MH
41.	מלאי+ ספקים	פונקציונלי	על מנת להוציא הזמנות תקופתיות על המערכת לעדכן את ההזמנה יום לפני מועד האספקה.	Done Low MH
42.	מלאי+ ספקים	פונקציונלי	על המערכת לוודא שרמת המלאי הצפויה להגיע תעלה על הכמות המינימלית שמוגדרת לכל פריט.	Done Low NH
43.	ספקים	פונקציונלי	כאשר נעשית הזמנה למוצר שמסופק על ידי יותר מספק אחד, על המערכת לבדוק איזה ספק יביא את המחיר הטוב ביותר ולהזמין ממנו.	Done Low NH
44.	מלאי	פונקציונלי	על המערכת לתת גישה למחסנאי לניהול מלאי הזמנות מספקים.	Done Low NH
45.	ספקים	פונקציונלי	על המערכת לתת גישה לאחראי קשרי ספקים לניהול כרטיסי ספק חוזים עם הספקים וכדומה.	Done Low NH
46.	מלאי+ ספקים	פונקציונלי	על המערכת לאפשר למנהל חנות לשלוף דו"חות ונתונים מכל סוג.	Done Low NH
47.	מלאי+ ספקים	פונקציונלי	על המערכת לאפשר גישה למערכת באמצעות ממשק גרפי (GUI).	Done High NH
48.	מלאי+ ספקים	פונקציונלי	על המערכת לתת אפשרות גישה למשתמש דרך GUI או CLI.	Done Low MH
49.	מלאי+ ספקים	פונקציונלי	על המערכת לאפשר גישה למערכת באמצעות CLI.	Done High NH
50.	מלאי+ ספקים	פונקציונלי	על המערכת להציג אך ורק את האופציות המתאימות לתפקיד הנבחר.	Done Low MH

שינויים: הוספת דרישות 44-50.

טבלה 2: הנחות

#	נושא	שאלה	תשובה
1.	קביעת כמות נדרשת בהזמנה	איך נקבעת כמות נדרשת להזמנה של פריט, האם מעבר לכמות מינימלית יש כמות אידיאלית או שהמינימלית היא האידיאלית?	הכמות שצריכה להיות בחנות צריכה להיות בשאיפה יותר גדולה במעט מהמינימום, תהיה אפשרות להזין ערך עבור כמות אקסטרה שרצויה עבור כל מוצר.
2.	זמן הגעה של הזמנה תקופתית	אם ההזמנה היא תקופתית היא תמיד מגיעה באותו יום בשבוע ולפי כך אפשר לעדכן את המלאי? או צריך לבדוק מתי מספקים לי אותה?	אפשר להניח שאספקה תקופתית תמיד מגיעה באותו הזמן.
3.	מספר ספקים בהזמנה	האם מספר ספקים יכולים להיות מעורבים באותה הזמנה?	לא, בהזמנה מעורב ספק אחד בלבד.
4.	בחירת ספק להזמנה עקב חוסר	על פי מה נבחר ספק להזמנה עקב חוסר?	בהינתן שישנו חוסר במוצר מסוים המערכת תייצר את ההזמנה על פי מועד ההגעה המוקדם ביותר, כלומר למצוא את הספק שייתן לי את המשלוח המהיר ביותר, גם אם זה יותר יקר מהאופציה הזולה ביותר.
5.	משתתפים בהזמנה האוטומטית	אילו בעלי תפקידים משתתפים כעת בתהליך ההזמנה האוטומטית?	כעת תהליך ההזמנה האוטומטי משתף את הספקים (כי צריך שהם יגידו לנו את מועדי האספקה של המוצרים שלהם) וגם את המחסנאים שמעדכנים את המלאי ואת החוסרים.
6.	יזימה של הזמנה	מי יוזם הזמנה קבועה ומי יוזם הזמנה עקב חוסר?	הזמנה קבועה מתואמת מול הספק, והזמנה עקב חוסר היא יוזמה של מחסנאי.

טבלה 3: מושגים

#	מושג	תיאור
1.	Supplier	ספק הוא ארגון הפועל בשיתוף פעולה עם החברה. ייתכן שיש לחברה מספר מגוון של ספקים שכולם מספקים פריטים שונים. בנוסף, עשויים להיות 2 או יותר ספקים המספקים את אותו פריט לחברה. לכל ספק יש קטלוג משלו, אשר מתאר את הפריטים, ההנחות והמחיר ליחידה עבור כל פריט בקטלוג.
2.	Contract	חוזה בין החברה לספק, זהו למעשה כניסה יחידה של קטלוג הספק.
3.	Order	כפי שפורט לעיל, החברה עובדת עם ספקים שונים מהם היא יכולה להזמין פריטים. כל הזמנה במערכת קשורה לספק ספציפי, והיא עשויה להכיל רק פריטים המופיעים בקטלוג הספק. פרטי ההזמנה הם כפי שמופיעים בקובץ המצורף למטלה.
4.	Item	פריט שנמכר בחברה.
5.	Category	סופר-לי מסווגת את הפריטים שלה לפי קטגוריות. לכל קטגוריה עשויות להיות תת קטגוריות. החברה עשויה לכלול הנחה גלובלית עבור כל הפריטים באותה קטגוריה.
6.	Report	יש כל מיני דוחות במערכת: באופן כללי, דו"ח הוא רשימה של פריטים שכולם חולקים מאפיין משותף, למשל: מוצר עומד לפוג, מוצרים פגומים, מוצרים בכמות נמוכה וכולי.

Use Case .1

Use Case Diagram .A

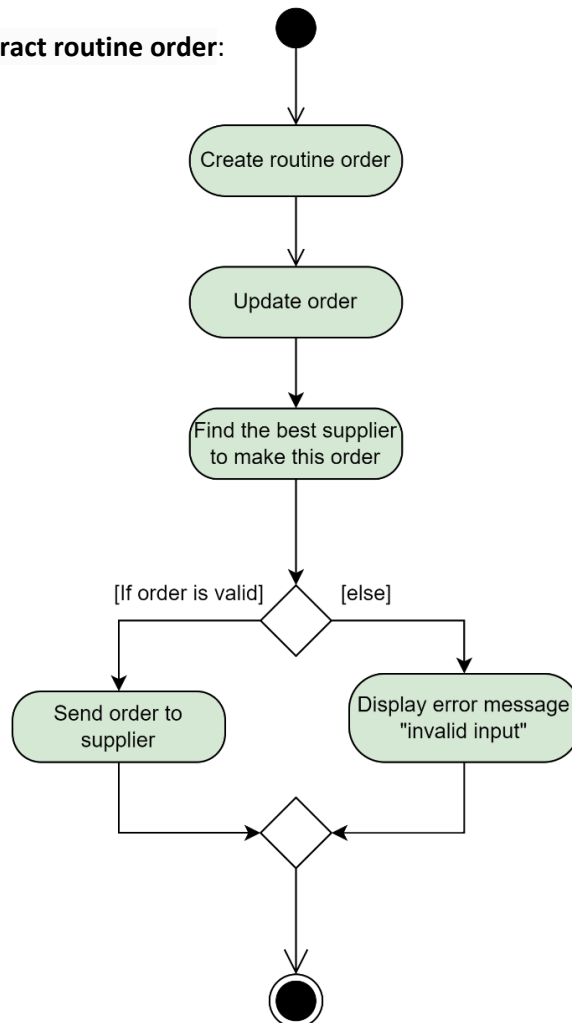


שינויים: מחיקת ספק בתור שחקן.

Use Case Description .B

Use case name	Extract routine order
Textual Description	הוצאת הזמנה תקופתית מספק
List of Actors	inventory
Pre-conditions	<ul style="list-style-type: none"> ההזמנה עודכנה יום לפני ביצוע ההספקה זמן ההגעה הינו זמן ההגעה המוגדר מראש
Post-conditions	<ul style="list-style-type: none"> ההזמנה בוצעה רמת המלאי הצפויה עם הגעת ההזמנה עולה על רמת מלאי המינימום המוגדר לכל פריט
Main success scenario	<ul style="list-style-type: none"> יום לפני ההזמנה התקופתית עודכנה ההזמנה כנדרש נבחרו ספקים עם המחיר הטוב ביותר ע"פ כתבי הכמויות ביום ההזמנה הספק הכין את כל המוצרים בהצלחה המשלוח התקבל במלאי רמת המלאי שהגיעה עם ההזמנה עלתה על רמת מלאי המינימום המוגדר
Alternatives/Extensions	<ul style="list-style-type: none"> התראה על שגיאה במקרה של הזמנת פריטים שלא קיימים אצל אף אחד מהספקים אי אישור ההזמנה במקרה שהגיע כמות נמוכה ממה הצפוי

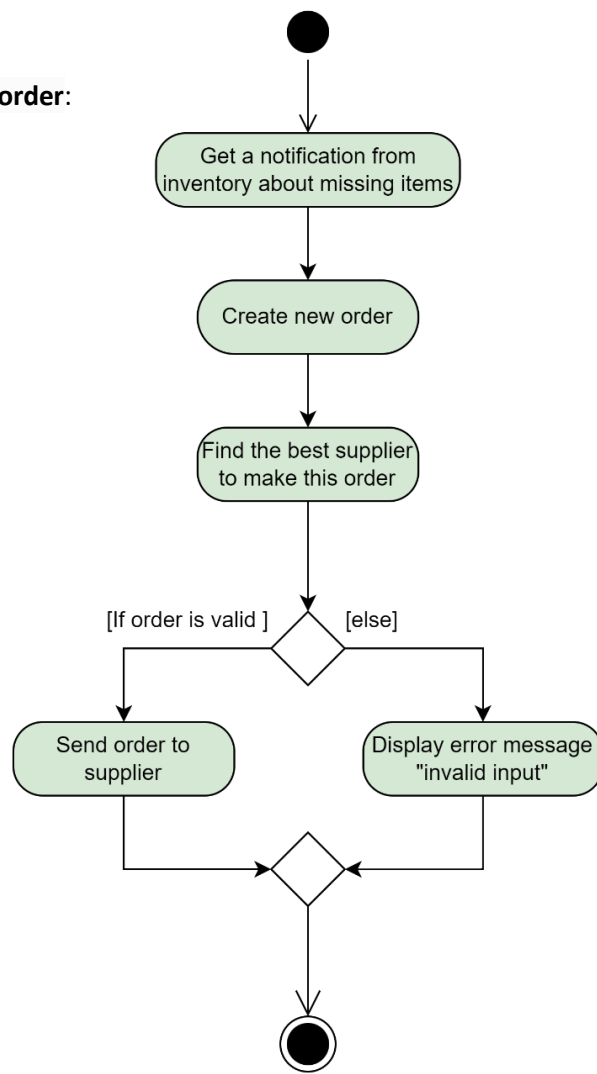
Activity diagram for **Extract routine order**:



Use case name	Send alert about missing items and make an order
Textual Description	הוצאת הזמנה מספק עקב חוסר
List of Actors	User
Pre-conditions	<ul style="list-style-type: none"> • התבצעה התראה על חוסר במוצר מסוים
Post-conditions	<ul style="list-style-type: none"> • הזמנה בוצעה • רמת המלאי הצפויה עם הגעת ההזמנה עולה על רמת מלאי המינימום המוגדר לכל פריט
Main success scenario	<ul style="list-style-type: none"> • התבצעה התראה על חוסר במוצר מסוים • נחבר את הספק המהיר ביותר ולאחר מכן נתחשב בספק הזול ביותר לבצע ממנו את ההזמנה • המשלוח התקבל במלאי • רמת המלאי שהגיעה עם ההזמנה עלתה על רמת מלאי המינימום המוגדר
Alternatives/Extensions	<ul style="list-style-type: none"> • התראה על שגיאה במקרה של הזמנת פריטים שלא קיימים אצל אף אחד מהספקים • אי אישור ההזמנה במקרה שהגיע כמות נמוכה ממה הצפוי

Activity diagram for

Extract missing Items order:



שינויים: שינוי התנאים בתרחישים.

Sequence & Collaboration Diagrams .2

Contracts .A

C01: UpdateItemsToOrder

Operation	UpdateItemsToOrder(items: List <Item>, amounts : List<integer>, dateOfDelivery: LocalDate)
Cross References	Use Case: Update items to order
Preconditions	There are items in the data base.
Postconditions	items and amount were updated.

C02: FindBestSuppliers

Operation	FindBestSuppliers (items: List <Item> , amounts : List<integer>)
Cross References	Use Case: Find the best supplier for the order
Preconditions	<ul style="list-style-type: none">• There are suppliers in the data base.• There are items in the data base.
Postconditions	A List<Supplier> instance suppliers was returned from the data base.

C03: CreateRoutineOrder

Operation	CreateRoutineOrder(items: List <Item> , amounts : List<integer>, suppliers: List<Supplier>)
Cross References	Use Case: Order
Preconditions	<ul style="list-style-type: none">• There are suppliers in the data base.• There are items in the data base.
Postconditions	<ul style="list-style-type: none">• An Order instance order was created with Boolean field isRoutine as true and enum field routineTime as specificDay (instance creation)• Order was saved in the data base.• supplier was associated with the current Order (association formed).• Supplier.gotOrder became true (attribute modification)

C04: CheckAmount

Operation	CheckAmount (orderId: int)
Cross References	Use Case: Order
Preconditions	There is an underway order.
Postconditions	If the amount level expected upon arrival of the order exceeds the minimum amount level defined for each item than the order is received.

C05: ReceiveOrder

Operation	ReceiveOrder (order: Order)
Cross References	Use Case: Order
Preconditions	<ul style="list-style-type: none">• There is an underway order.• The order was updated a day before the making the supply.• the arrival time is the predefined arrival time.• The amount level expected upon arrival of the order exceeds the minimum amount level defined for each item.
Postconditions	<ul style="list-style-type: none">• Order status was updated to delivered.

C06: GetMissingItemNotification

Operation	GetMissingItemNotification (Item item)
Cross References	Use Case: Send alert about missing items and make an order
Preconditions	<ul style="list-style-type: none">• There are items in the data base.• The amount of an item is lower than his minimum amount.
Postconditions	Notify to suppliers that an item is run of stock.

C07: FindBestSupplier

Operation	FindBestSupplier (Item item, Amount amount = default)
Cross References	Use Case: Find the best supplier for the order
Preconditions	<ul style="list-style-type: none">• There are suppliers in the data base.• There are items in the data base.
Postconditions	A supplier instance was return from the data base.

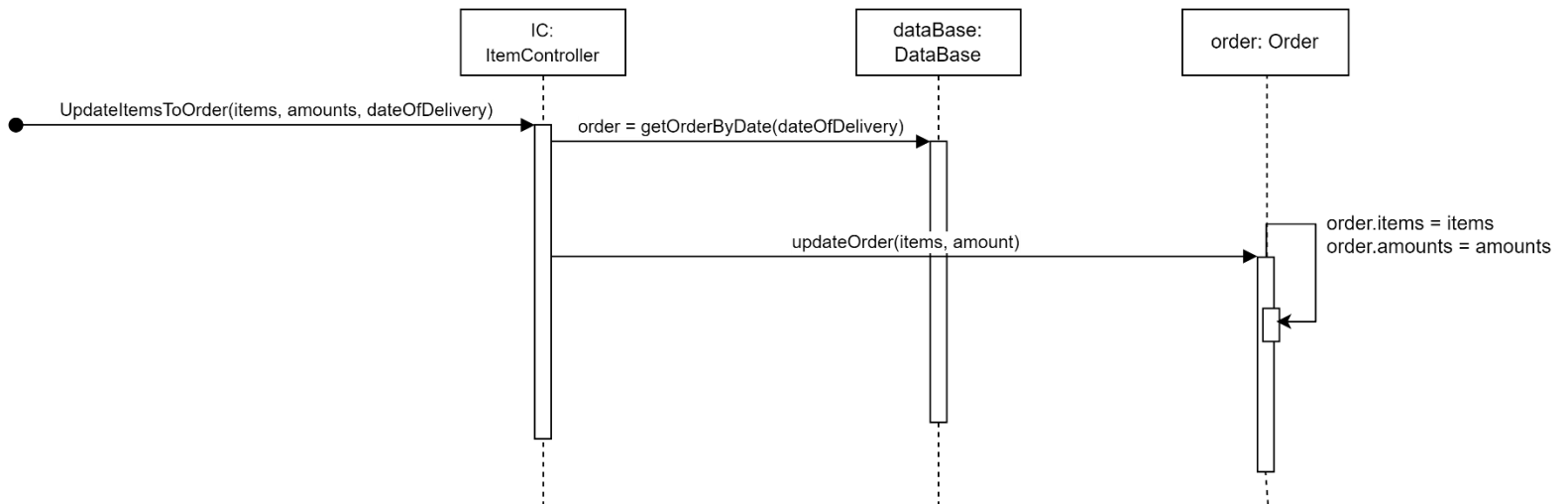
C08: CreateMissingItemsOrder

Operation	CreateMissingItemsOrder(Item item, Amount amount = default)
Cross References	Use Case: Order
Preconditions	<ul style="list-style-type: none">• There are suppliers in the data base.• There are items in the data base.
Postconditions	<ul style="list-style-type: none">• An Order instance order was created with Boolean field isRoutine as false (instance creation)• Order was saved in the data base.• Supplier.getOrder became true (attribute modification)

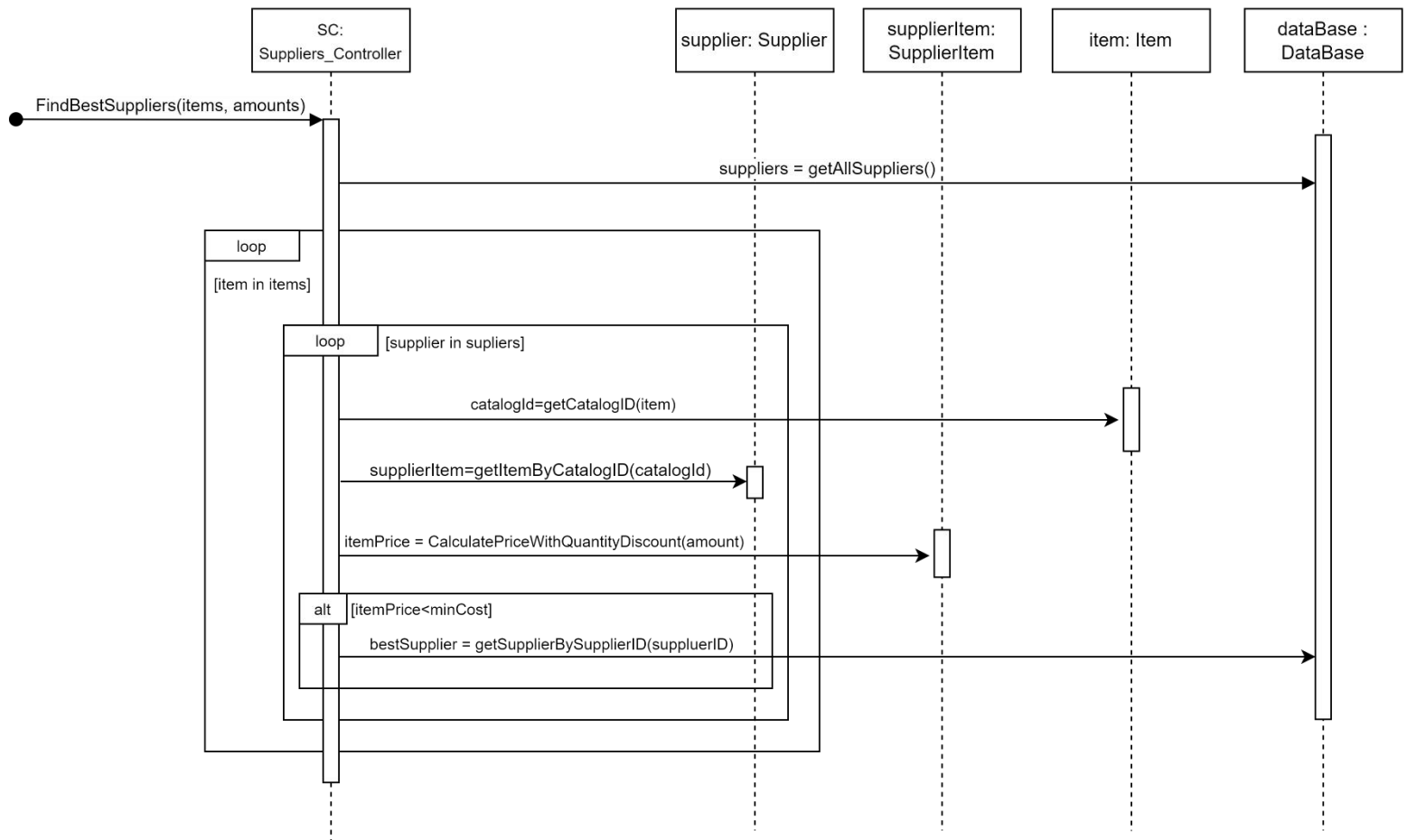
שינויים: הוספת תנאים מקדימים לחוזים.

Sequence Diagrams .B

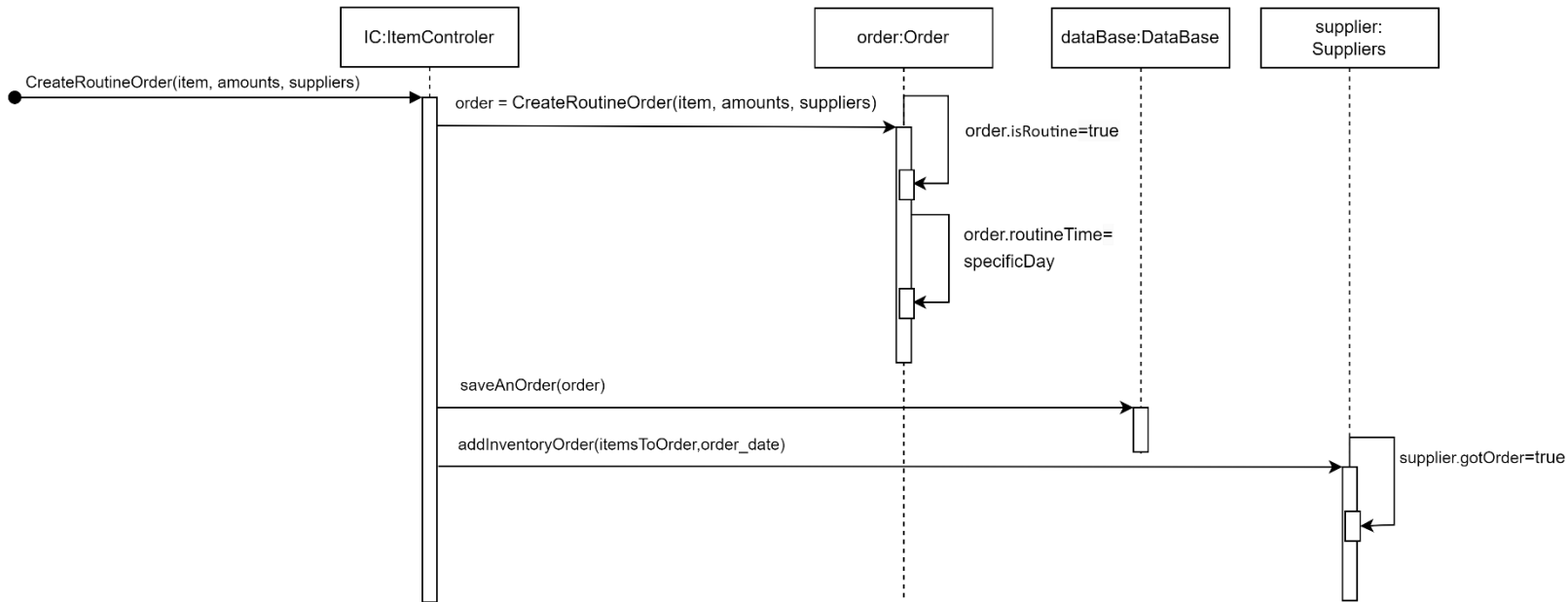
Sequence diagram 1: UpdateItemsToOrder



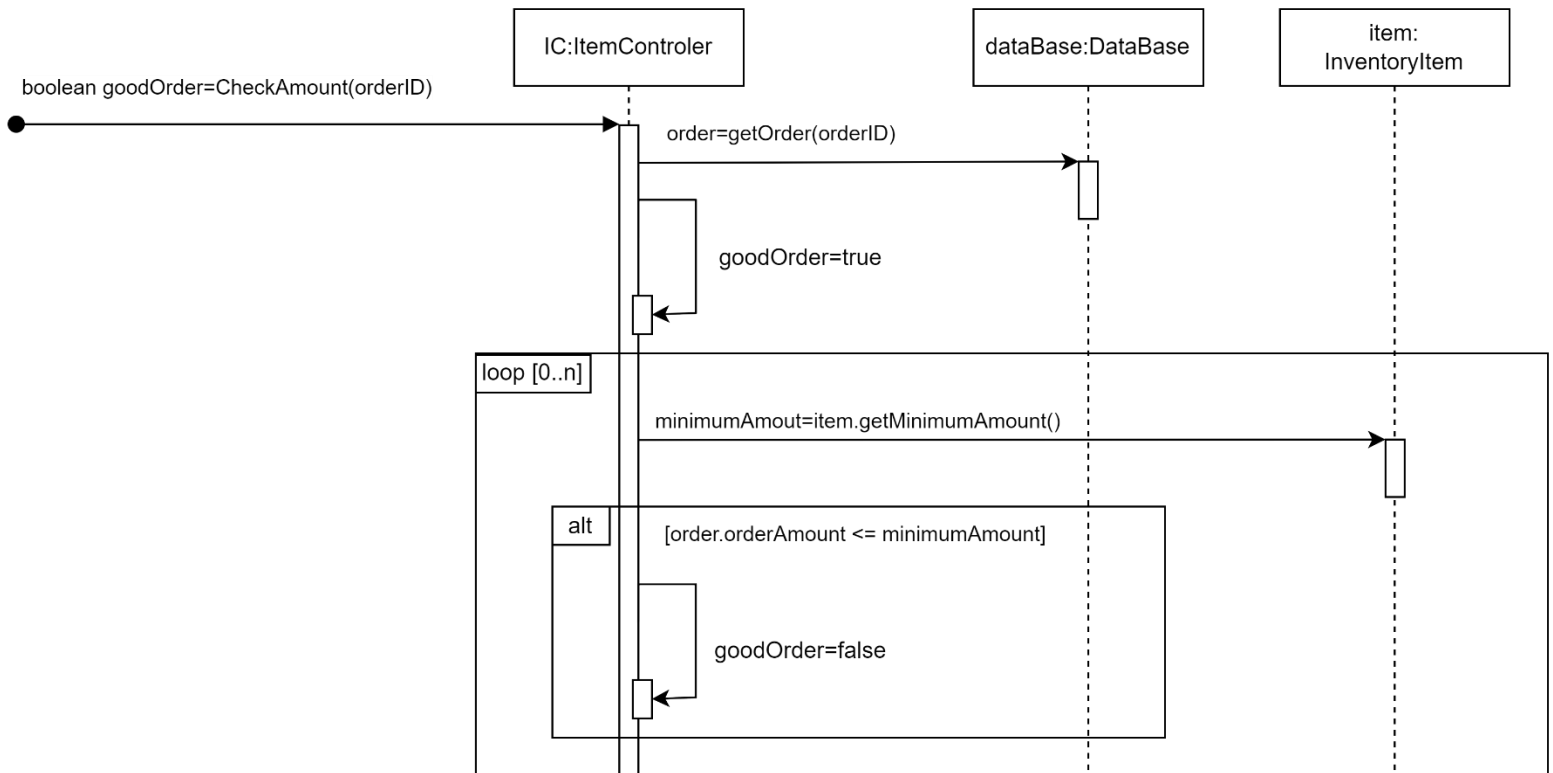
Sequence diagram 2: FindBestSuppliers



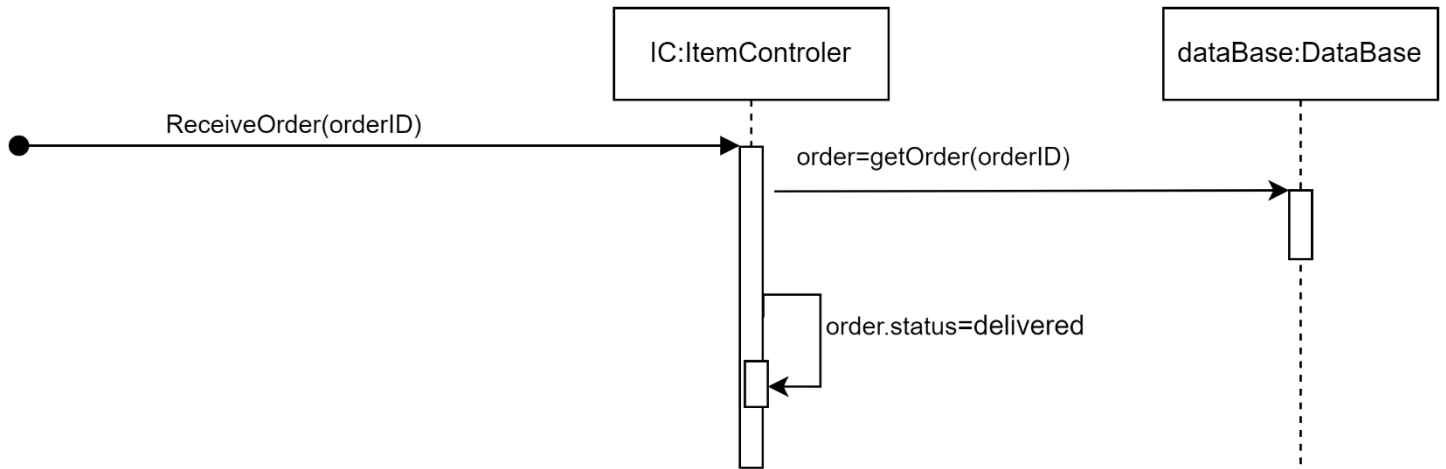
Sequence diagram 3: CreateRoutineOrder



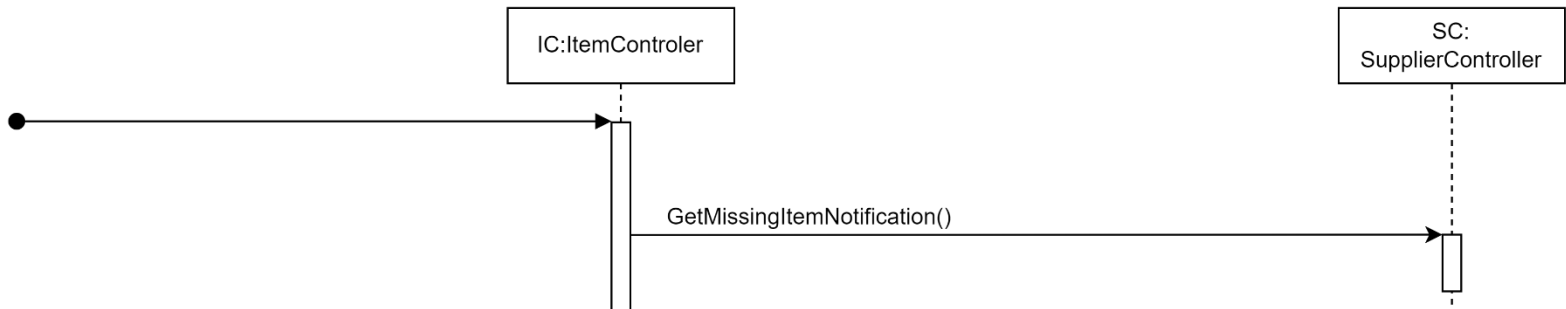
Sequence diagram 4: CheckAmount



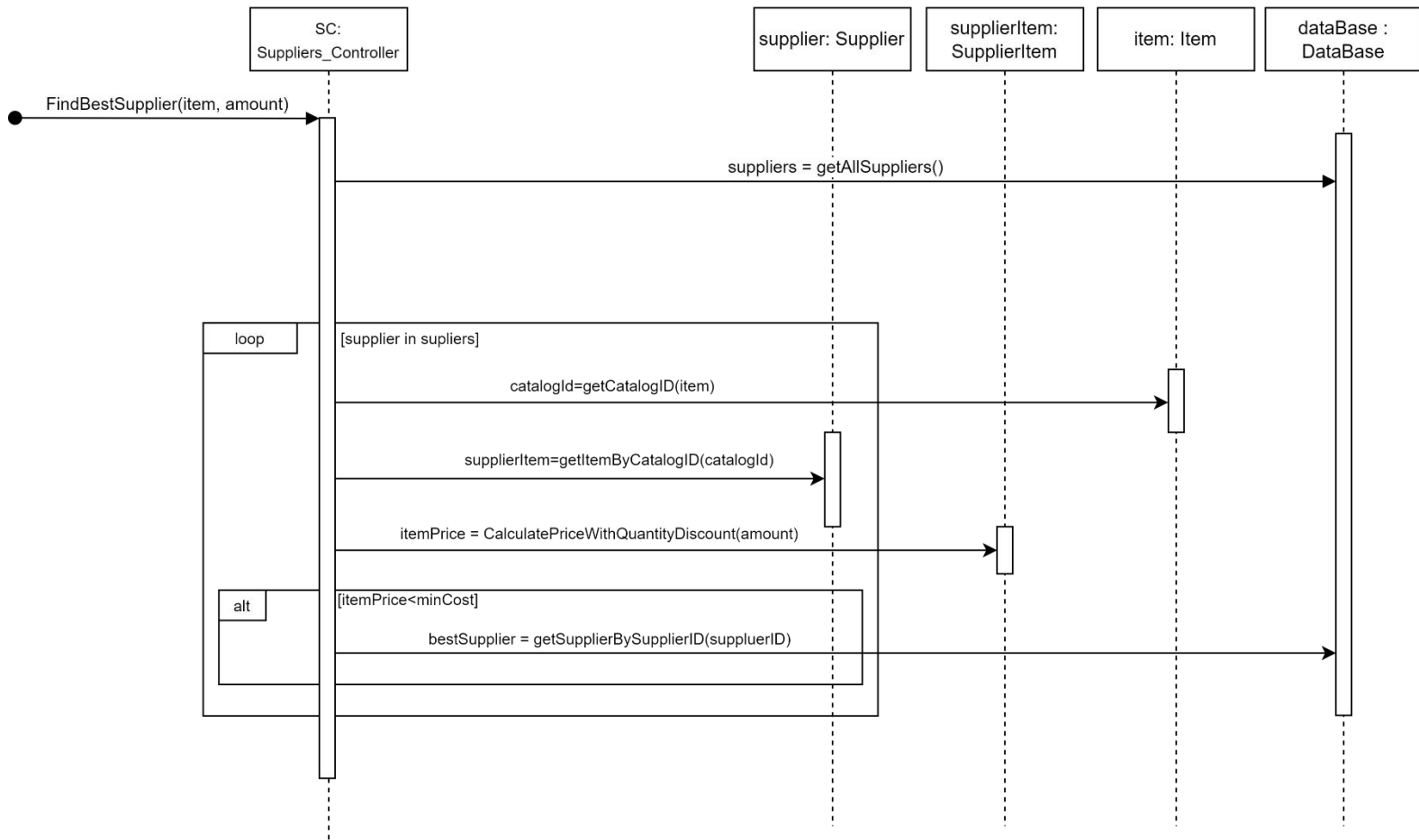
Sequence diagram 5: ReceiveOrder



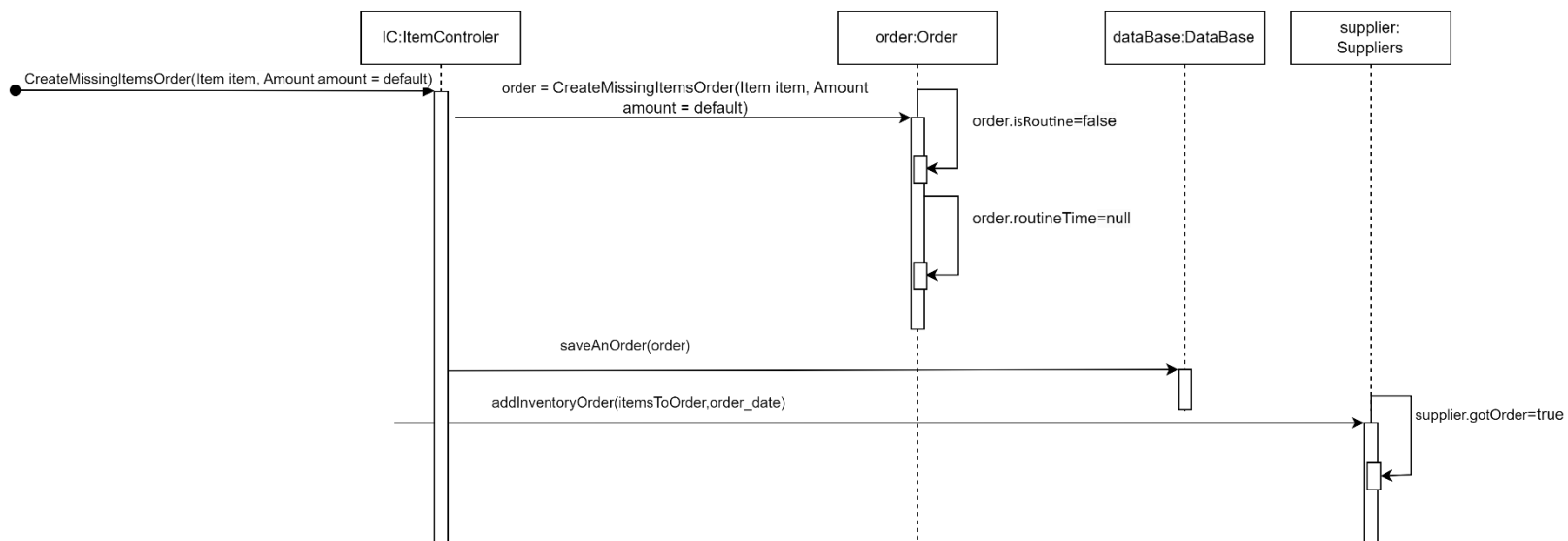
Sequence diagram 6: GetMissingItemNotification



Sequence diagram 7: FindBestSupplier



Sequence diagram 8: CreateMissingItemsOrder



ERD .3

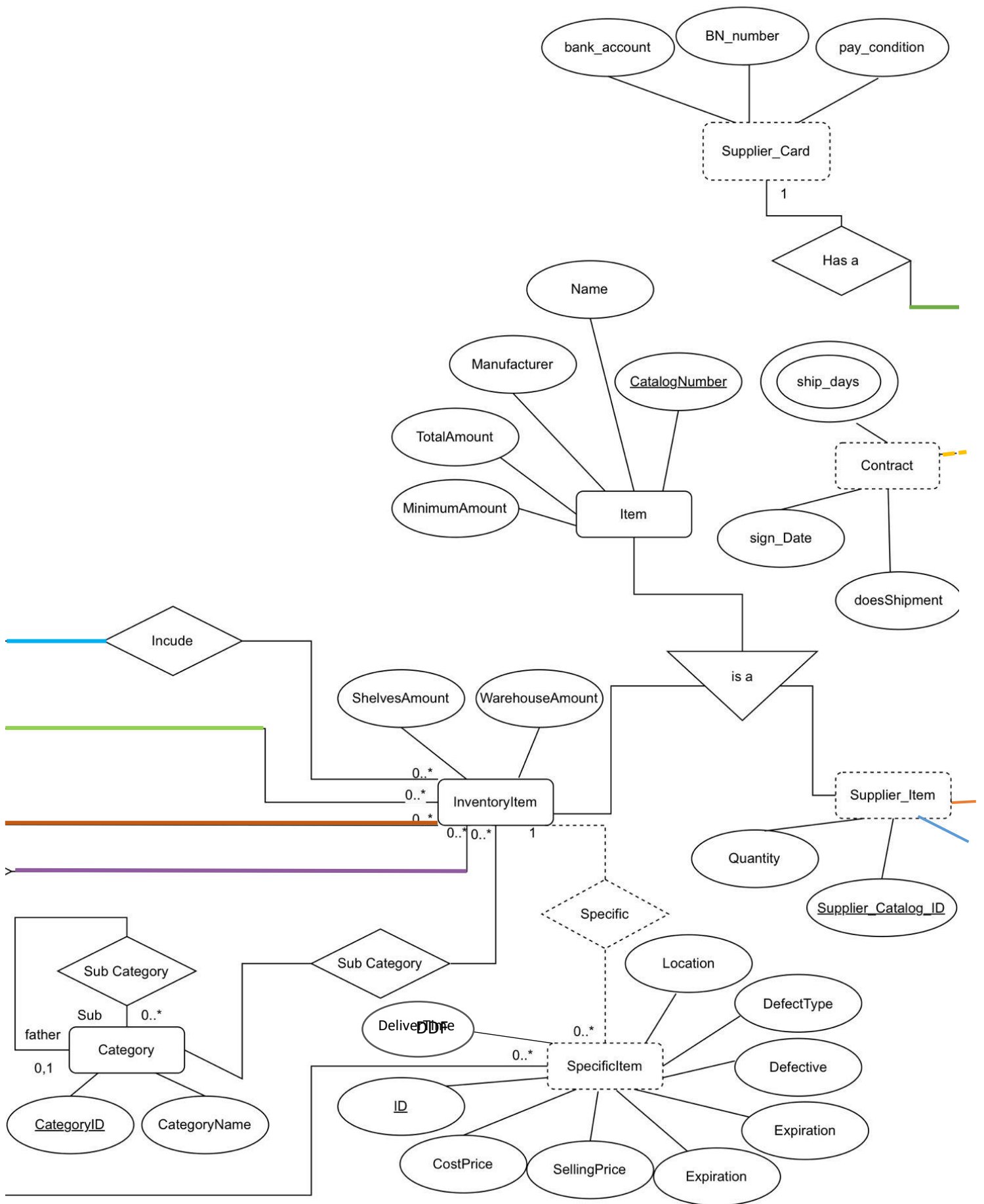
This diagram, labeled 'ERD .3', provides a detailed view of the database schema. It includes the following entities and their attributes:

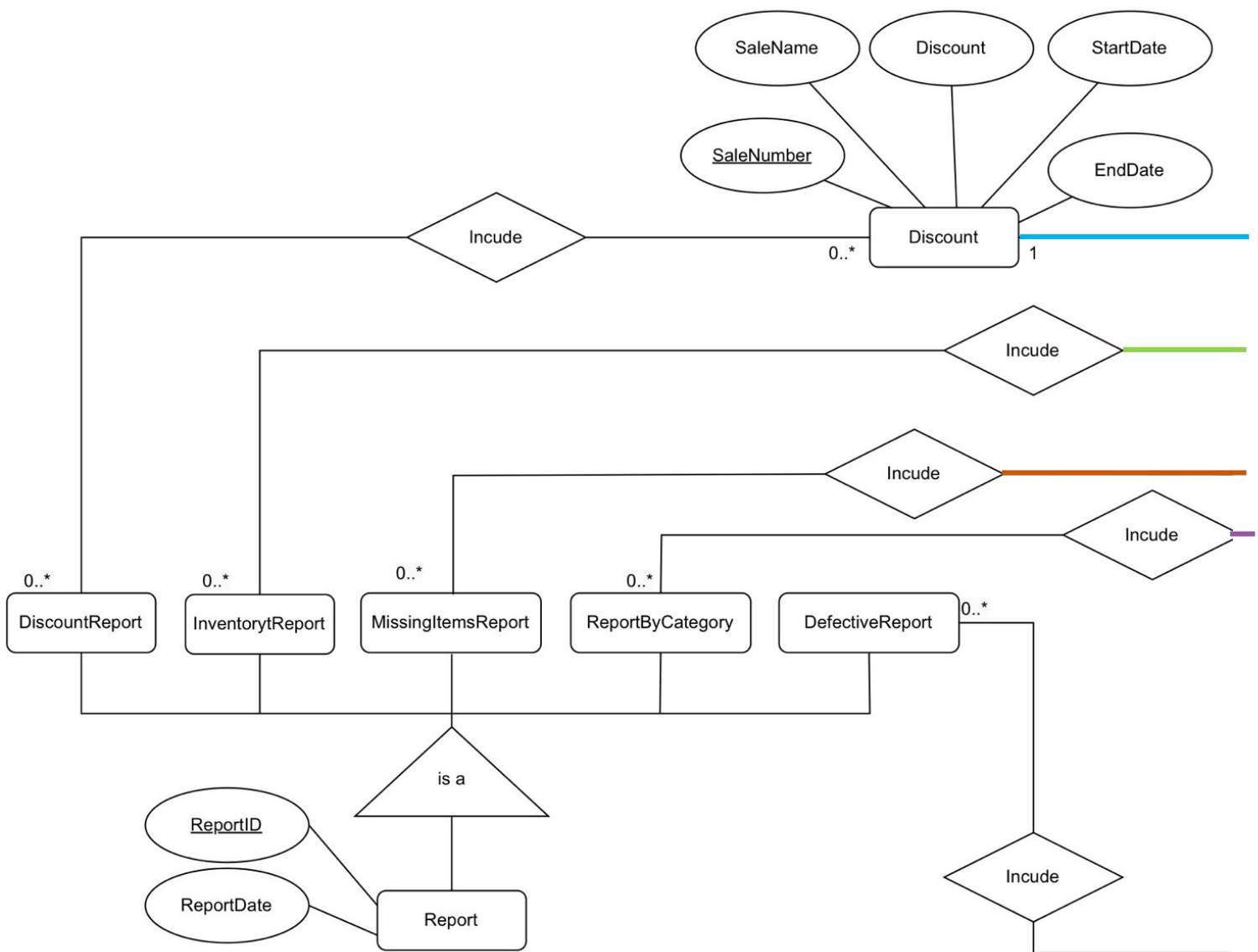
- Contact** (dashed rectangle): name, phone.
- TotalOrderDiscount** (dashed rectangle): price, discount, type.
- Supplier** (rectangle): supplier_ID, supplier_name, adress, address.
- Order** (rectangle): orderId, Status, dateOrderIssued, supplierName, totalQuantity, price_before_discount, orderAmount, final_price, total_discount, CatalogID, total_price_without_discount.
- OrderForInventory** (dashed rectangle): supplierName, address, orderDate, phone_number, itemsInSupplierOrder (double oval).
- manufacture** (rectangle): Id, name.
- Item_for_Order** (dashed rectangle): orderAmount, final_price, total_discount, CatalogID, total_price_without_discount.
- discountByTotalQuantity** (dashed rectangle): Discount, Quantity.

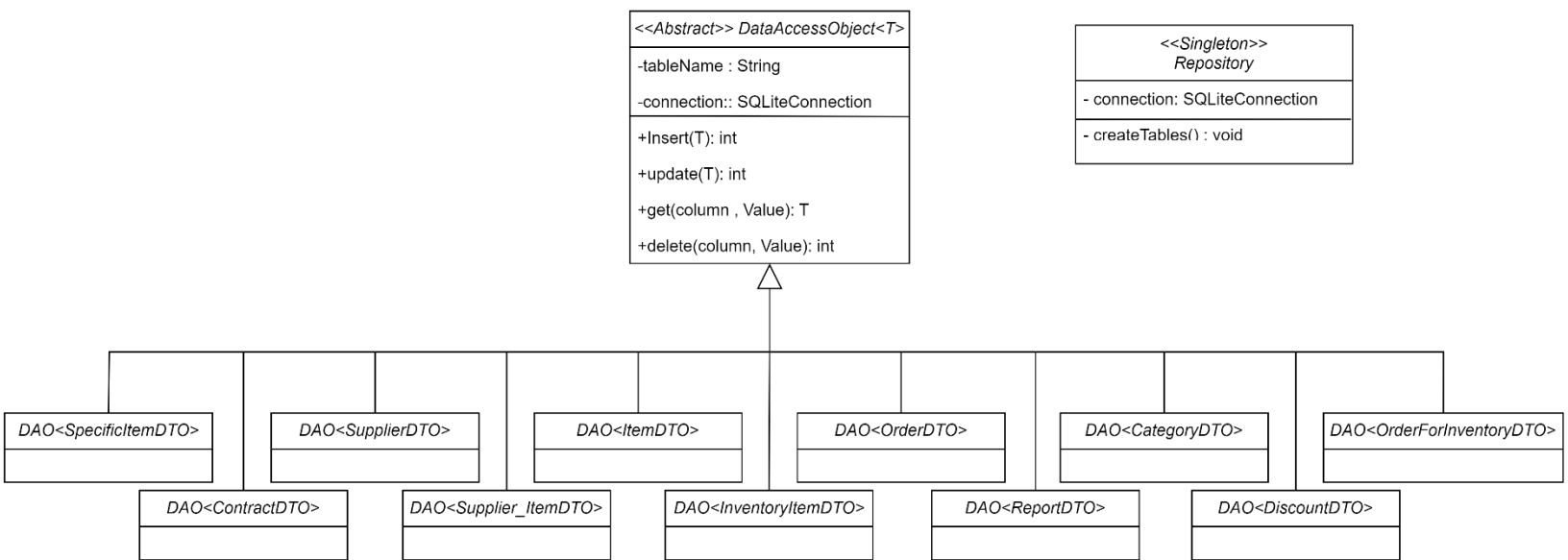
The relationships (diamonds) and their cardinalities are:

- Contact with**: Contact (1..*) to Supplier (1).
- Discount Order by**: Supplier (1) to TotalOrderDiscount (0..*).
- Has a**: Supplier (1) to Contact (1). Highlighted with a green line and a dashed orange line.
- supplies**: Supplier (1) to Order (0..*). Highlighted with an orange line.
- Makes an**: Supplier (1) to Order (0..*).
- Makes**: Supplier (1) to OrderForInventory (0..*).
- Supplies from**: Supplier (1) to manufacture (1..*).
- hold**: Order (1) to discountByTotalQuantity (0..*). Highlighted with a blue line.
- include**: Order (1) to Item_for_Order (1..*).

)







All mappers are associated to their respective DAOs

ItemDAO
- dao : ItemDAO
+ buildItem(ItemDAO) : Item
+ addItem(Item) : int
+ updateItem(Item) : void
+ getItem(int) : Item
+ updateMinAmount(int): void
+ updateAmount(int): void
+ remove(Item): void

OrderForInventoryDAO
- dao :OrderForInventoryDAO
+ buildOrder(OrderForInventoryDAO) : OrderForInventor
+ addOrder(Order,int) : int
+ remove(Order) : void
+ getOrder(int) : Order

Supplier_Item_DAO
- dao :Supplier_ItemDAO
+ addSupplier_Item(Supplier_Item, int) : int
+ buildSupplier_Item(Supplier_ItemDTO) : Supplier_Item
+ remove(Supplier_Item) : void
+ removeDiscount(Supplier_Item, int) : void
+ addDiscount(Supplier_Item, int, int) : void
+ getItemContracts(int) : List
+ getAll_Items() : List

SpecficItemDAO
- dao :SpecficItemDAO
+ addSpecficItem(SpecficItem, int) : void
+ deleteSpecficItem(SpecficItem, int) : void
+ buildSpecficItem(SpecficItemDTO) : SpecficItem
+ updateSpecficItem(SpecficItem) : void
+ getAllItems() : List
+ getSpecficItemByID(int) : SpecficItem
+ removeAll() : void

OrderDAO
- dao : OrderDAO
+ buildOrder(OrderDAO) : Order
+ addOrder(Order,int) : int
+ updateOrder(Order) : void
+ remove(Order) : void
+ getOrder(int) : Order
+ addItemToOrder (Order,ProdInOrder, int) : int
+ updateItemOrder(Order,ProdInOrder,int) : int
+ removeItemFromOrder(Order,Contract) : int

InventoryItemDAO
- dao :InventoryItemDAO
+ addItem(InventoryItem) : void
+ buildItem(InventoryItemDTO) : InventoryItem
+ updateItem(InventoryItem) : void
+ remove(InventoryItem) : void
+ getItemByCatalogNumber(int) : InventoryItem
+ addSpecficItemsToItem(List, int) : void
+ getAllItems() : List
+ getSpecficItemsByCatalogNumber(int) : List
+ removeSpecficItemsFromItem(int) : void
+ removeAll() : void

DiscountDAO
- dao : DiscountDAO
+ buildDiscount(DiscountDTO) : Discount
+ addDiscount(Discount) : void
+ updateDiscount(Discount) : void
+ getDiscountBySaleNumber(int) : Discount
+ removeDiscount(int) : void
+ removeAll() : void
+ getItemsBySaleNumber(int) : List
+ addItemToDiscount(List, int) : void
+ getAllDiscounts() : List

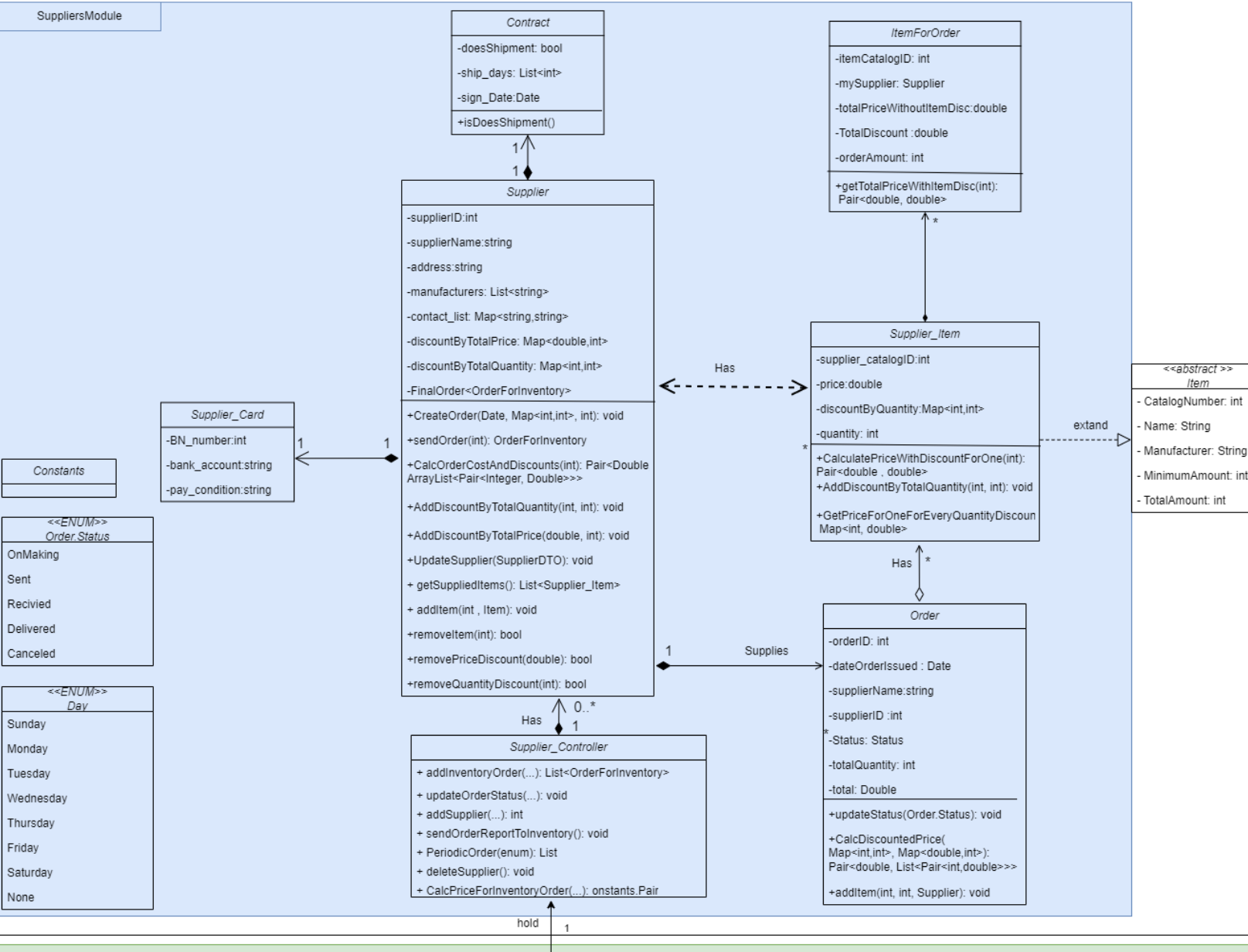
ContractDAO
- dao : ContractDAO
+ buildContract(ContractDAO) : Contract
+ addContract(Contract) : int
+ updateContract(Contract) : void
+ getContract(int) : Contract
+ remove(Contract): void

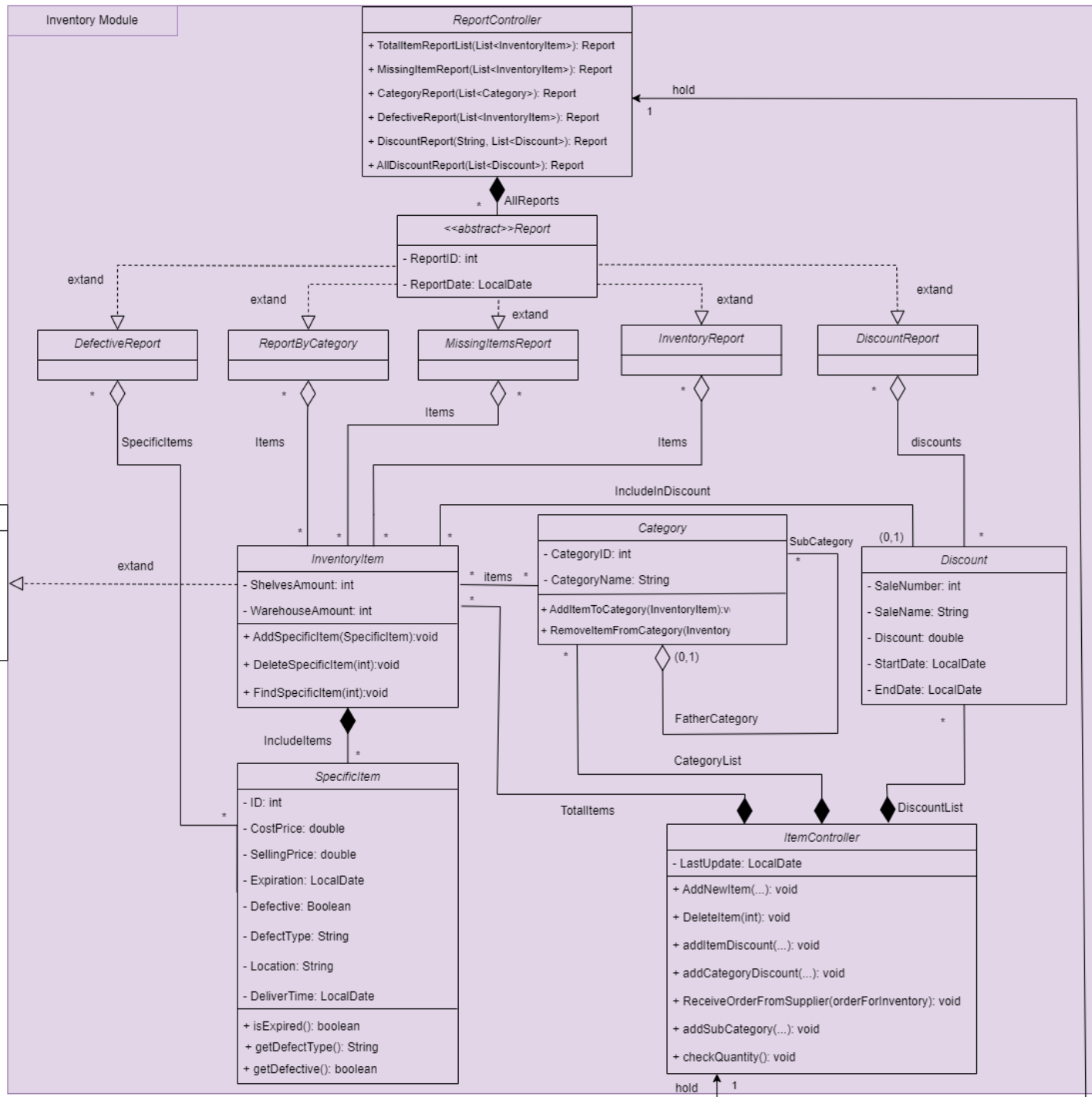
ReportDAO
- dao : ReportDAO
+ buildReport(ReporyDTO) : Report
+ addReport(Report) : void
+ updateReport(Report) : void
+ getReportByID(int) : Report
+ removeAll() : void
+ addItemToReport(List, int) : void
+ addDiscountsToReport(List, int) : void
+ addSpecficItemsToReport(List, int) : void

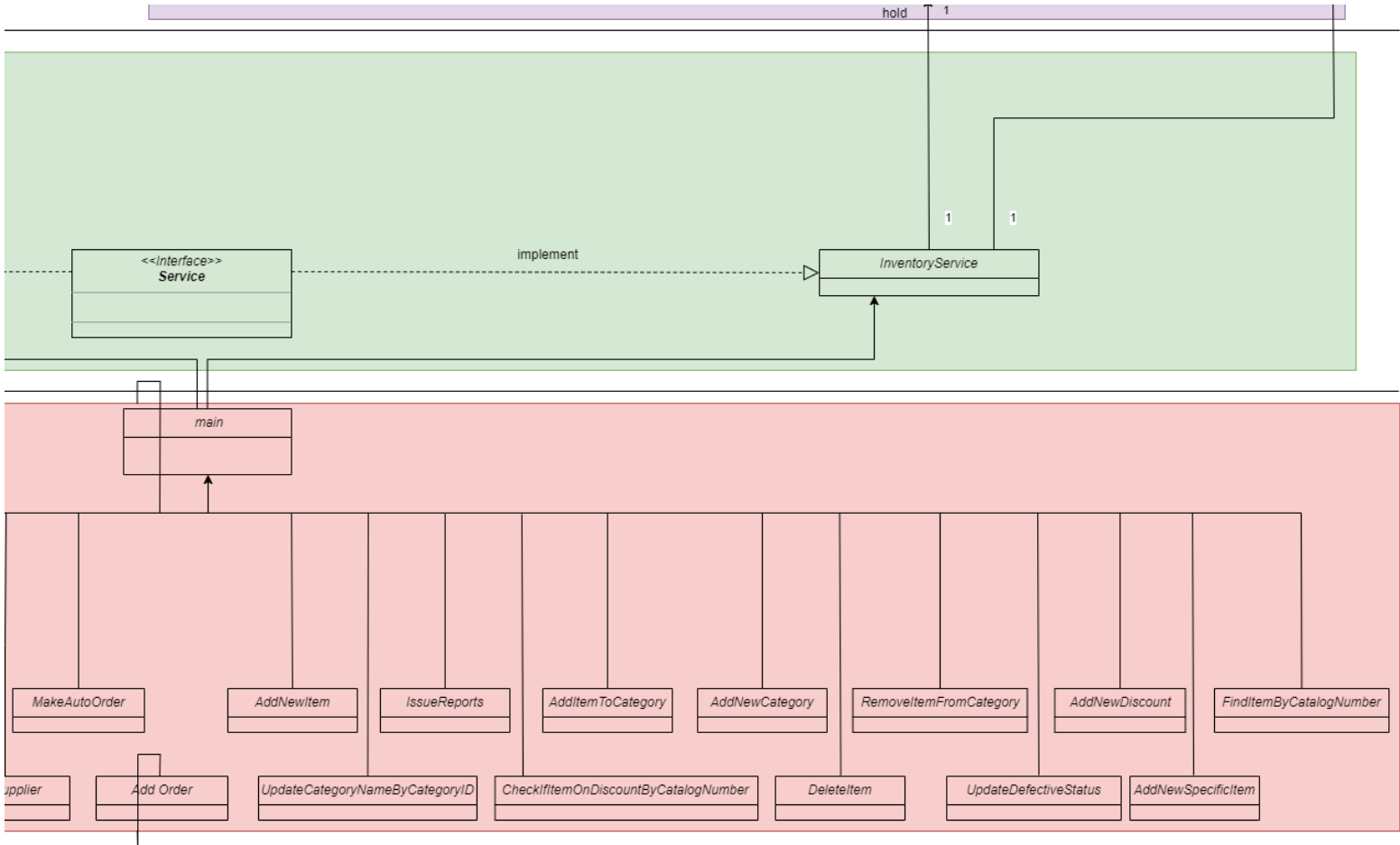
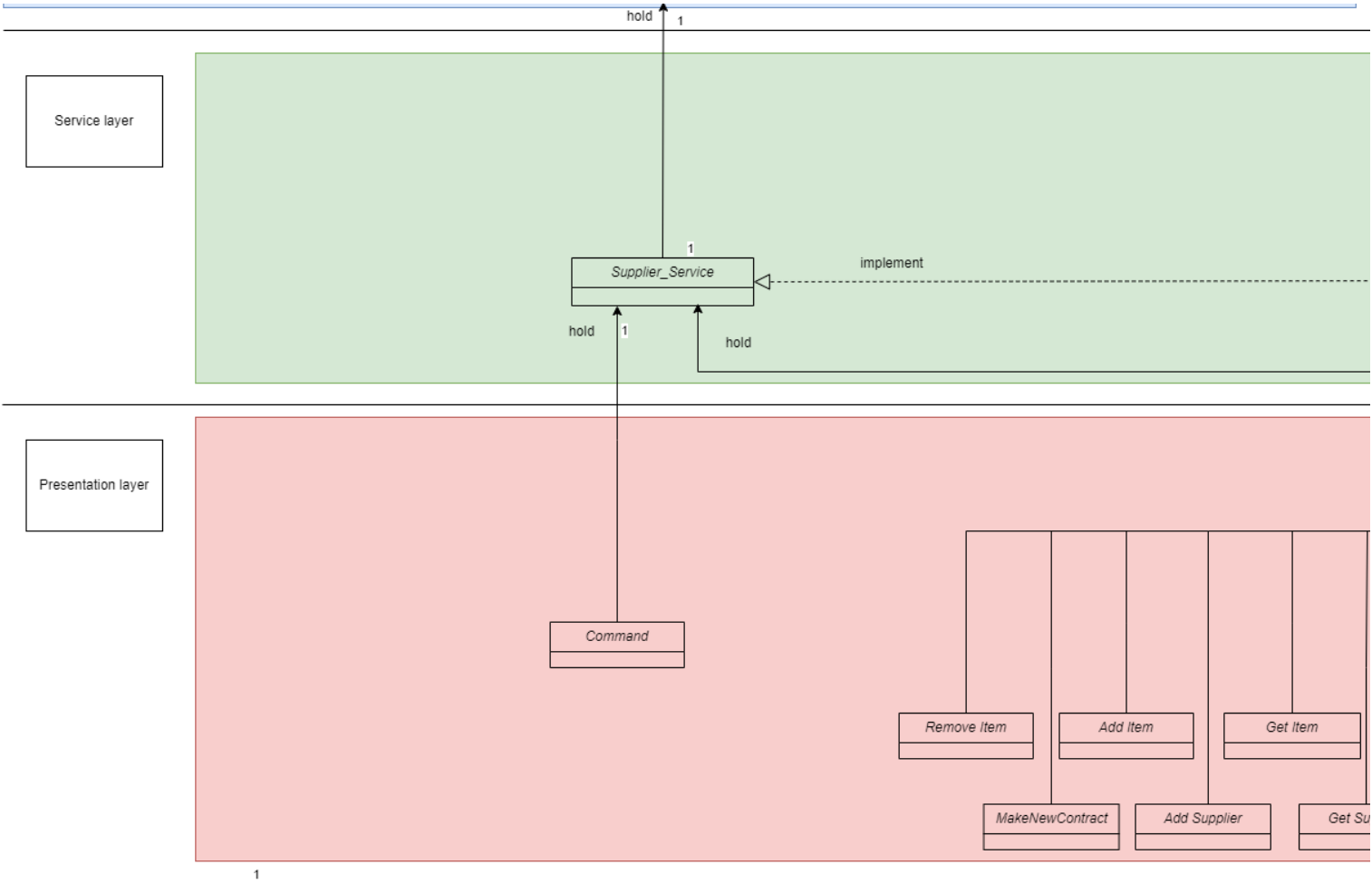
CategoryDAO
- dao : categoryDAO
+ buildCategory(CategoryDTO) : Category
+ addCategory(Category) : void
+ updateCategory(Category) : void
+ getAllCategories() : List
+ getCategoryByld(int) : Category
+ findItemsByCategory(int) : List
+ findSubCategoriesByCategory(int) : List
+ removeAll() : void
+ addTotItemsInCategory(List, int) : void

SupplierDAO
- dao : SupplierDAO
+ buildSupplier(SupplierDAO) : SupplierItem
+ addSupplier(Supplier) : int
+ updateSupplier(Supplier) : void
+ deleteSupplier(Supplier) : void
+ getSupplier(int) : Supplier
+ removeDiscount(Supplier,double) : void
+ addDiscount(Supplier,double,int): void

Business layer





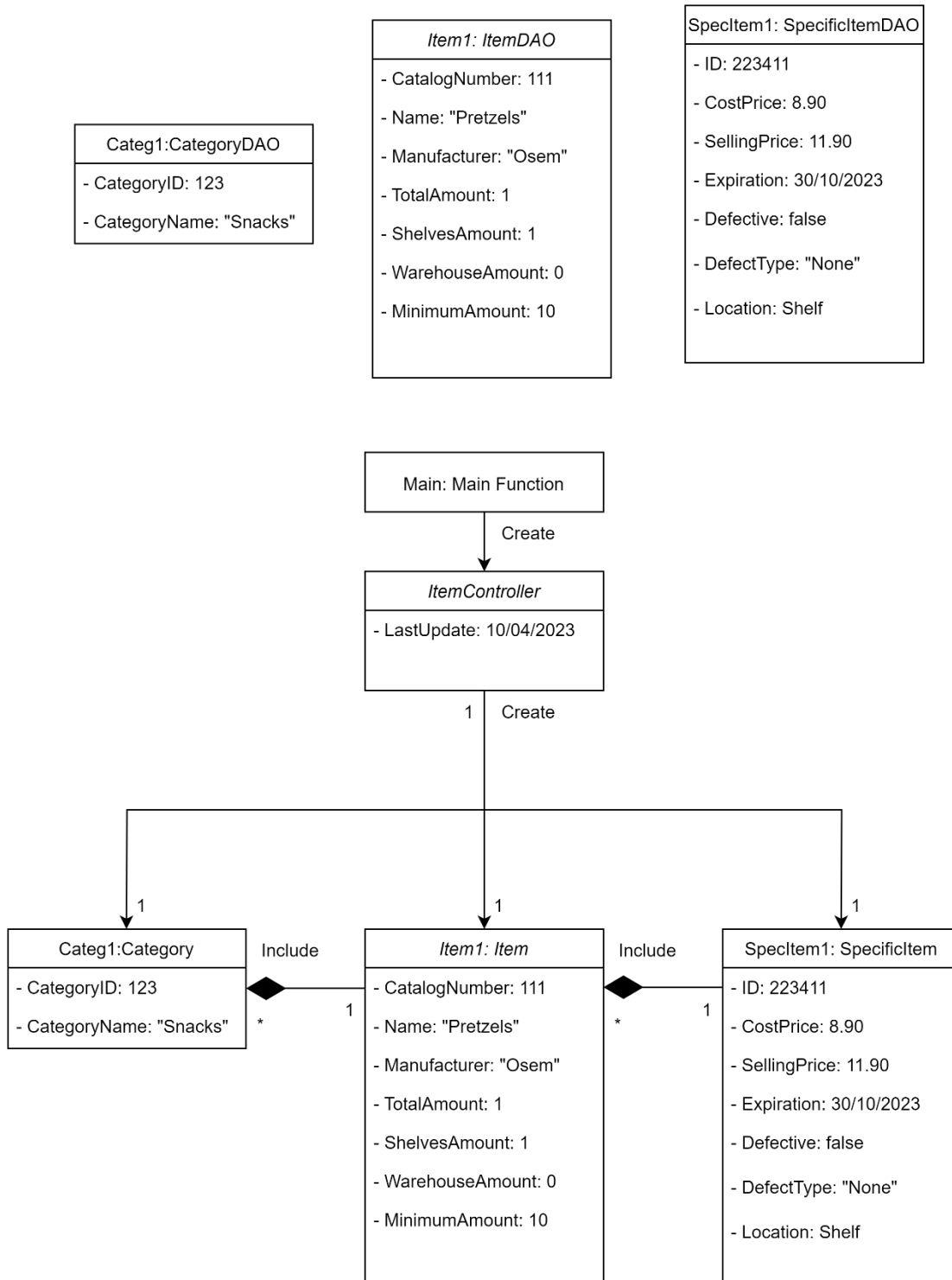


תרשים אובייקטים:

תרחיש 1: (מלאי)

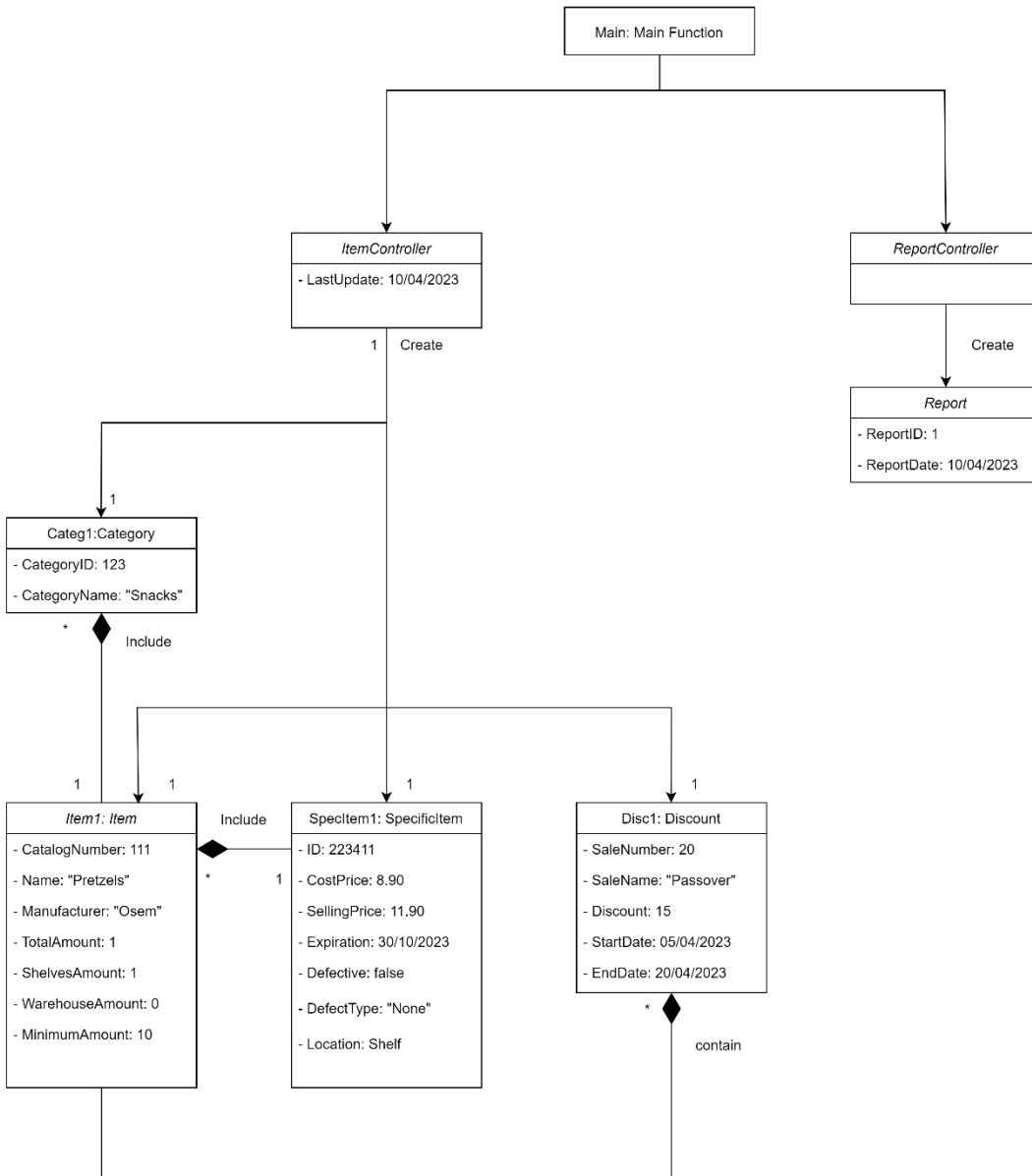
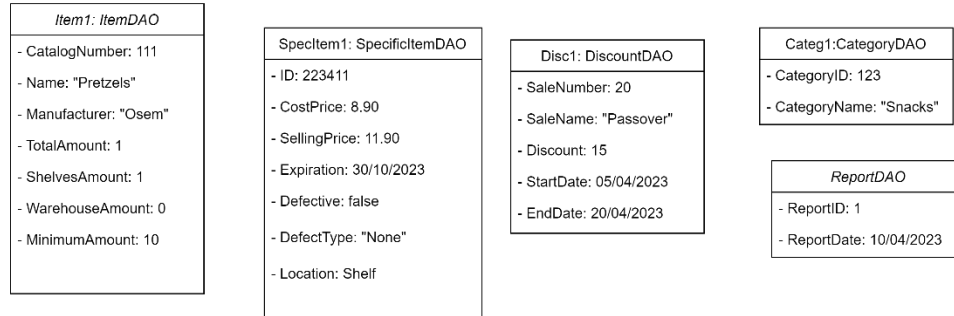
היום הגיע לסופר "לי" מוצר כללי חדש בשם "בייגלה", מנהל המחסן הוסיף את המוצר כללי של "ביילה" למערכת ובנוסף את המוצר הספציפי שקיבלנו במשלוח.

מנהלת הסופר התרגשה מהמוצר החדש שהגיע לסניף היום ולכן פתחה קטגוריה חדשה במיוחד בשבילו בשם "חטיפים"



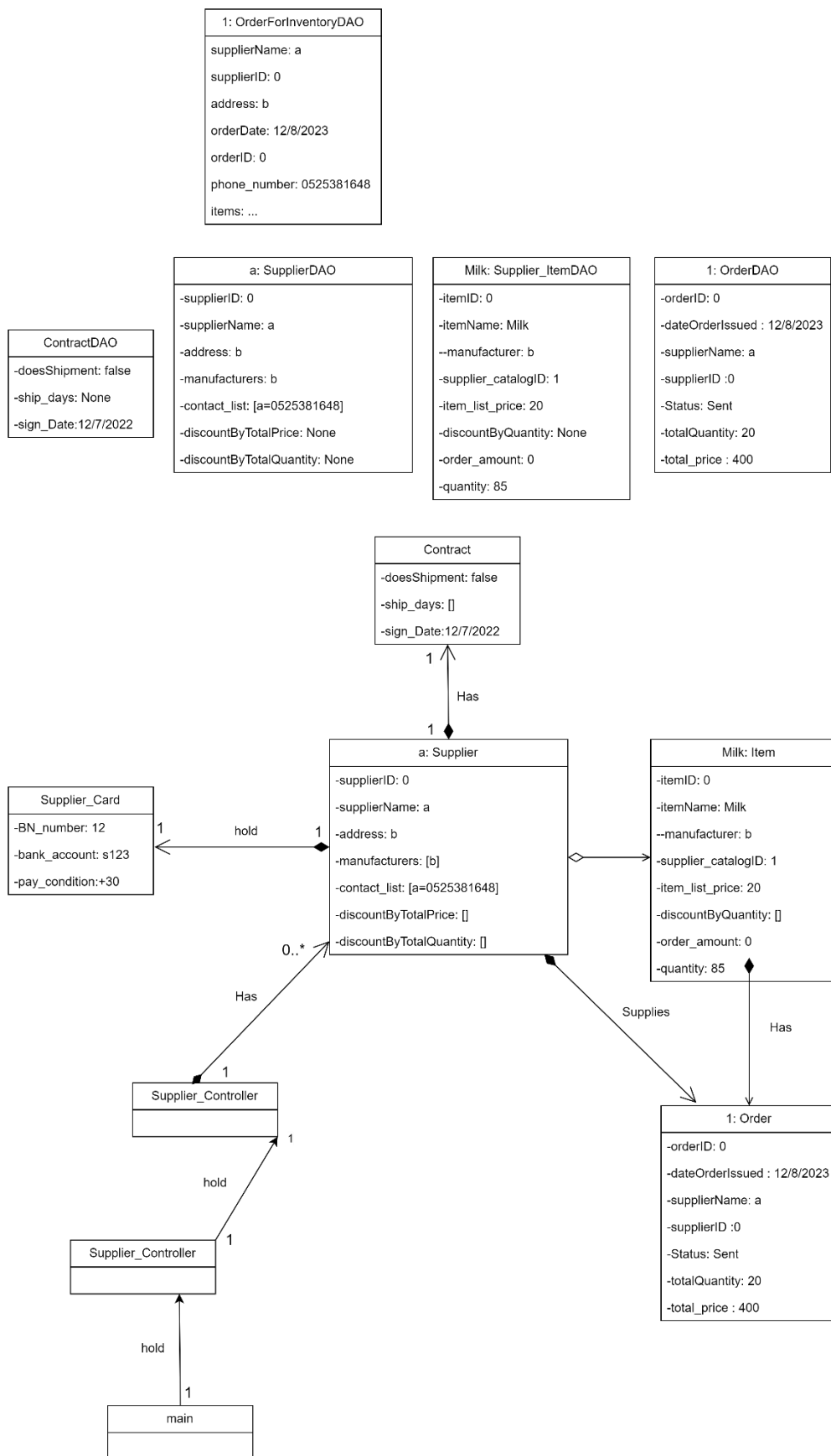
תרחיש 2: Strongly Satisfiable (מלאי)

בהמשך לתרחיש הקודם, לאור המוצר החדש שהגיע לסניף סופר "לי" החליט לפנק במבצע חדש על "בייגלה", מנהלת הסופר הכניסה את ההנחה החדשה למערכת.
בשעות הצהריים הגיעה לקוחה לחנות והבחינה במוצר "בייגלה" והחליטה לרכוש אחד לעצמה.
בסוף היום מנהלת הסופר הוציאה דוח מכירות עבור כל הפריטים אשר נמכרו באותו היום.



תרחיש 1: (ספקים)

הוספת ספק חדש למערכת



תרחיש 2: (ספקים)

