

Product

- key: String
- supplier: Supplier
- price: int
- retailPrice: int
- stock: int
- interest: list<clear>

- remove - method 1: void
- change - price (1: divide)

Box

- service-level: String

Book

- title: String
- author: String
- ISBN: String

Container

- service-level: String

Client

- key: String
- name: String
- address: String
- transactions: List<Transaction>
- notifications: List<Notification>
- points: int

- +add-Transaction (Transaction: Object): void
- +update-points (points: int): void
- +toggle-notifications (notification: Product): void

Supplier

- key: String
- name: String
- address: String
- active: boolean
- transactions: List<Transaction>

- +add-transaction (Transaction: Object): void
- +toggle-activity (1: boolean)

Container
- service-level: String
- service-quality: String

Transaction
- id: int

Polygons
- product-key: String
- description: String
- area: String

+ add - transaction (Transaction object): void
+ toggle - activity 1: boolean

Order
- supplier: Supplier
- products: List<Product>
- total-cost: int

Sale
- comment: int
- product: Product
- paid: boolean
- dealer: int
- client: int
+ pay - Sale (Store Stock): void
+ get - price () : int

Storage

- filename: String
- store: Store = new Shell();

+ save(): void

+ saveAll(filename: String): void

+ load(filename: String): void

+ importFile(filePath: String): void

+ registerProduct(key: String, price: int, criticalLevel: int, supplierKey: String, serviceType: String): void

+ registerBook(key: String, title: String, author: String, ISBN: String, price: int, criticalLevel: int, supplierKey: String): void

+ registerContainer(key: String, material: int, criticalLevel: int, supplierKey: String, serviceType: String, serviceLevel: String): void

+ changeProductPrice(key: String, price: int): void

+ showAllProducts(): void

+ showAllClients(): void

+ showTransaction(id: int): void

+ registerSupplier(name: String, key: String, address: String): void

+ registerClient(key: String, name: String, address: String): void

+ registerOrderTransaction(supplierKey: String, productKey: String, quantity: int, price: int, status: boolean): void

+ registerSaleTransaction(clientKey: String, date: int, productKey: String, quantity: int): void

store

- date: int
- available - balance: float
- real - balance: float

register - product int x 1: product

register - client (1): client

register - supplier (1): supplier

register - sale - transaction (1): sale

register - order - transaction (1): order

register - product - books (1): book

register - product - container (1): container

calculate - balance (1): void

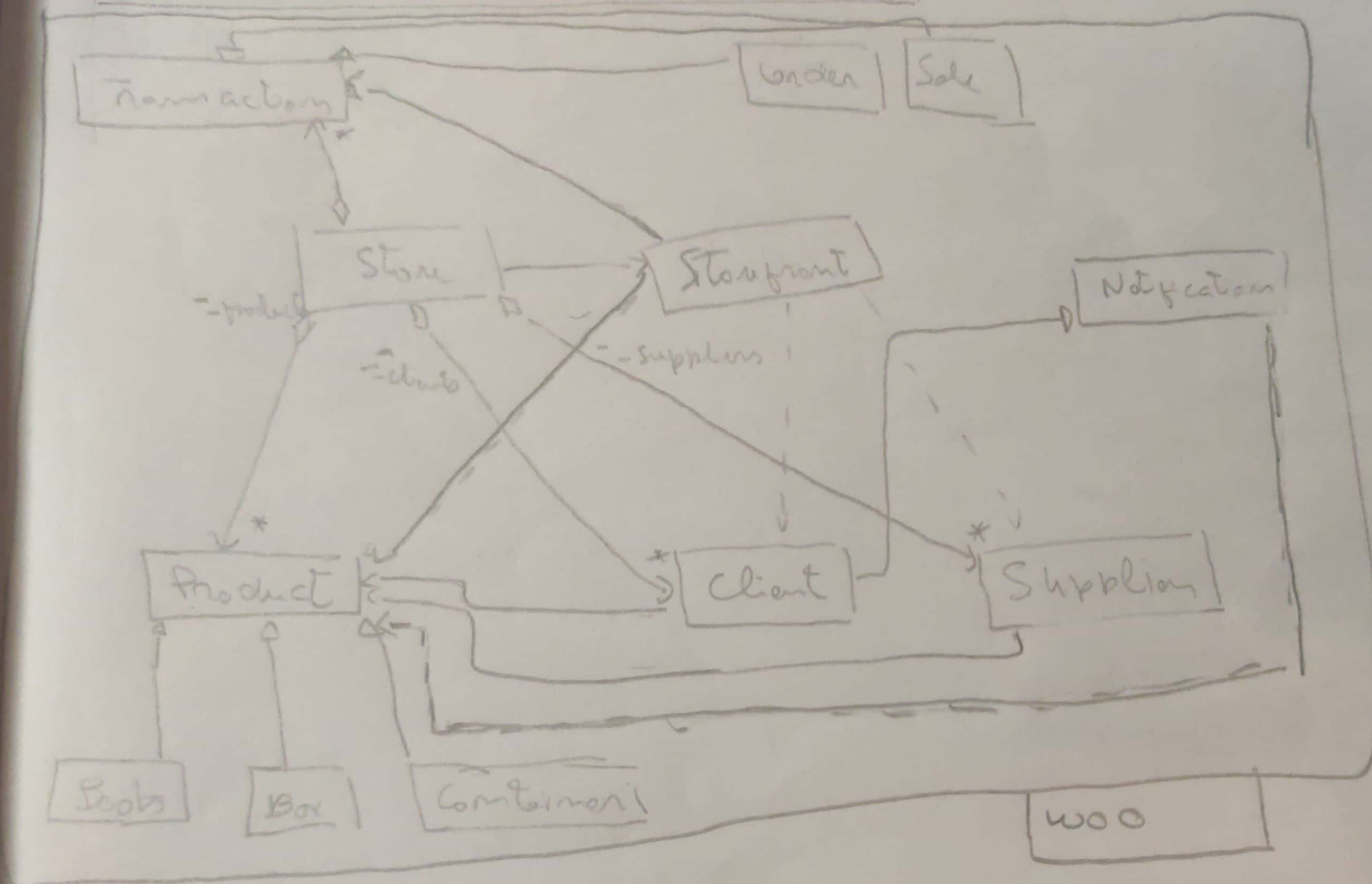
show - all - product (1): list < Product >

show - all - suppliers (1): list < Supplier >

show - transactions (1): list < Transaction >

show - supplier (1): Supplier

show - client (1): client



Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constam o grupo de trabalho.

André Matos

Declaro por minha honra que este diagrama foi realizado
apenas por elementos que constituem o grupo
do projeto.

André Matos

Declaro por minha honra que este diagrama
foi realizado apenas pelos elementos que
constituem o grupo do projeto.

Martins.