UML Class Diagram

Product *

Sale *

Acquisition *

Partner *

Batche *

WarehouseManager

Warehouse

- - date : int
- - idtransfer : int
- - balance : int
- - partners : Partner [ ]
- - products : Product [ ]
- - transactions : Transaction [ ]
- - batches : Batche [ ]

+ advance Date (days : int) : Void
+ show all products ( ) : String
+ show all batches ( ) : String
+ show batches by partner (id partner : String) : String
+ show batches by product (id product : String) : String
+ show partner (idpartner : string) : String
+ show all partners ( ) : String
+ register partner (id partner : String , name : String , adress : String) : Void
+ toggle product notifications (idpartner : String , id product : String) : Void
+ show partner acquisitions (id partner : String) : String
+ show partner sales (id partner : String) : String
+ show transaction (id transaction : int) : String
+ register breakdown transaction(idpartner : String , id product : String , amount : String) : Void
+ register sale transaction (idpartner : String , id product : String , limdate : int amount : int) : Void
+ register acquisition transaction(idpartner : String , idproduct : String , cost : int , amount : int) : Void
+ look up payments by partner (id partner : String) : String
+ look up product batches under given price (price : double) : String

WarehouseManager

--filename : String

-- warehouse : Warehouse

+ save ( ) : void

+ saveAs (filename : String) : void

+ load (filename : String) : void

+ importfile (textfile : String) : void

+ advanceDate (days : int) : void

+ showallproducts ( ) : String

+ showallbatches ( ) : String

+ show batches by partner (id partner : String) : String

+ showbatches by product (id product : String) : String

+ show all partners ( ) : String

+ show partner (id partner : String) : String

+ register partner (id partner : String, name : String, adress : String) : void

+ toggle product notifications (id partner : String, idproduct : String) : void

+ show partner acquisitions (id partner : String) : String

+ show partner sales (id partner : String) : String

+ show transaction (id transaction : int) : String

+ register breakdown transaction (id partner : String, idproduct : String, amount : String) : void

+ register sale transaction (id partner : String, idproduct : String, limdate : int, amount : int) : void

+ register acquisition transaction (id partner : String, id product : String, cost : int, amount : int) : void

+ lookup payments by partner (id partner : String) : String

+ lookup product batches under given price (price : double) : String

Product
- id : String
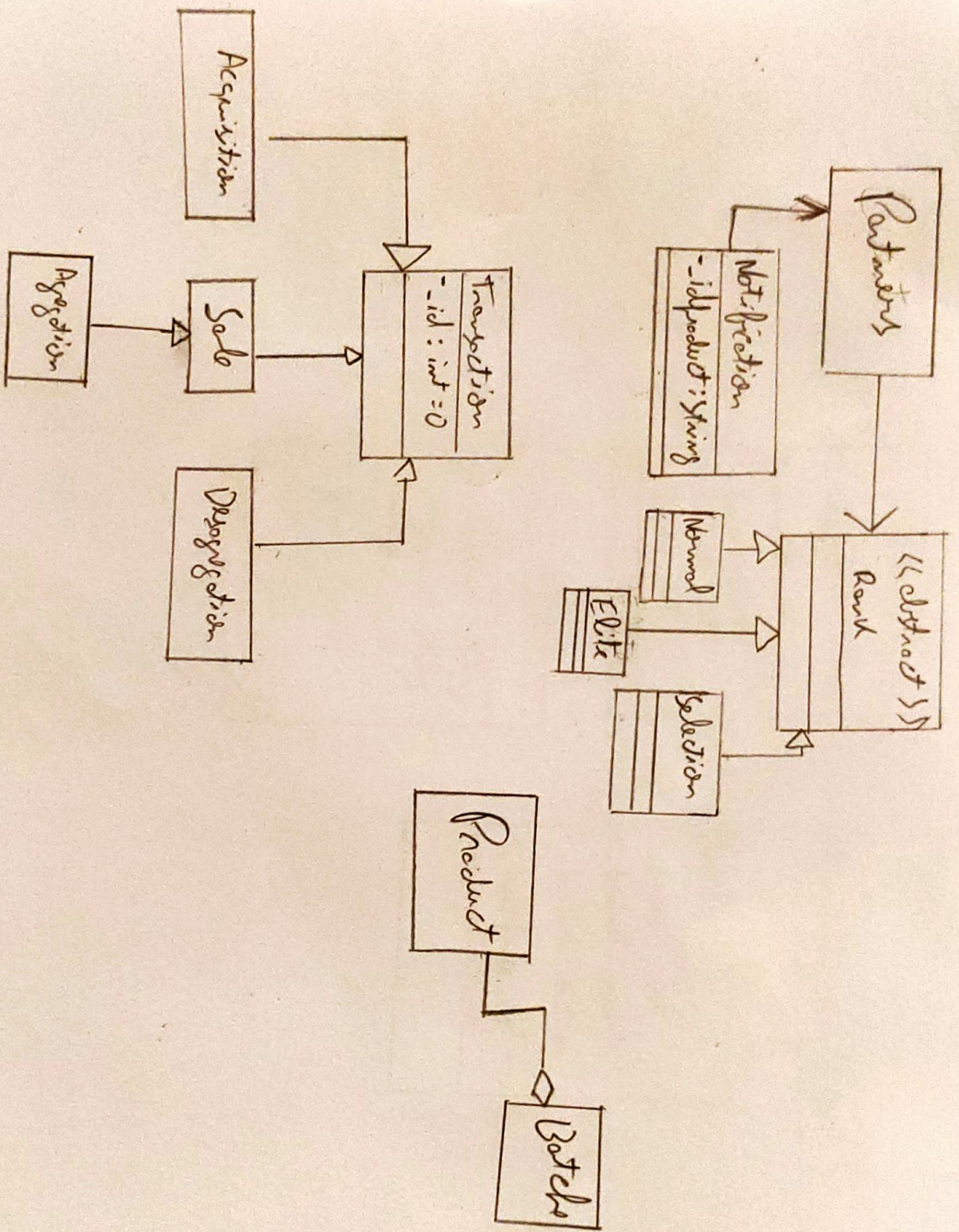- price : double
- amount : int
- recipe : Integer[]
- conflexity : String

Acquisition
- id partner : String
- id product : String
- amount : int
- totalprice : double
- order : int

Sale
- id partner : String
- id product : String
- amount : int
- limit : double
- order : int

Desgregation
- id partner : String
- id product : String
- amount : int
- price : double
- order : int
- disaggregate(id:product):void

Agregation
- Aggregate(id:product):void

Batch
- amount : int
- product : Product
- partner : Partner
- unitprice : int

UML class diagram (hand-drawn):

- **Acquisition**
- **Aggestion**
- **Sale**
- **Desaggestion**
- **Transaction**
  - -id : int = 0
- **Partners**
- **Notification**
  - -idproduct : string
- **Rank** «abstract»
- **Normal**
- **Elite**
- **Selection**
- **Product**
- **Batch**

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

Artur Monteiro

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

Diogo Pereira