uc

Let's assume that
order does not
require control of
physical products
availability

<<Subsystem>>
Point of Sale Terminal

Scan UPC

<<extend>>

<<include>>

**Choose a
product**

<<include>>

**Choose a
quantity**

**Validity
control**

<<include>>

Calculate tax

1..*

1

Customer Entity

**Payment by
cash drawer**

<<include>>

Checkout

1

<<include>>

**Payment by
credit /
check**

1

1

1

<<extend>>

Payment processing system

**Verify
payment**

1

**Receive a
receipt
(invoice)**

**Provide
Product
Catalog**

1

1

Manager Entity

**Start up the
terminal**

1

1..*

**Hold
information**

Store Entity

pkg

Model

Payment
- type : Enum
- total : int
- payed : boolean

Customer
- name : String
- creditCardNumber : String
+ getName() : String
+ setName(name : String) : void
+ getCreditCardNumber() : String
+ setCreditCardNumber(number : String) : void

ProductSpecs
- description : String
- price : int
- UPC : int
+ getDescription() : String
+ setDescription(desc : String) : void
+ getPrice() : int
+ setPrice(price : int) : void
+ getUPC() : int
+ setUPC(UPC : int) : void

ProductCatalog
+ validateUPC(UPC : int) : boolean
+ addProduct(product : ProductSpecs) : void
+ removeProduct(UPC : int) : ProductSpecs
+ loadProducts() : void
+ getProduct(UPC : int) : ProductSpecs

CLI

CLI
+ runStore() : void

Logic

SalesItem
- quantity : int
+ setQuantity(quantity : int) : void
+ getQuantity() : int
+ getSpecs() : ProductSpecs

Store
+ initPOSTs() : void
+ getPosts() : ArrayList

In CLI, I could implement such method as "readCommand" and use "createTransaction" to gradually create whole order by customer step-by-step by adding results from commands to Transaction object; since transactions are loaded automatically, this is not a part of CLI assignment, but it is ready for GUI use.

<<Default Package>>

Start
+ main() : void

Transaction
+ printInvoice() : String
+ pay() : boolean
+ addSalesItem(UPC : int) : void
+ removeSalesItem(UPC : int) : SalesItem

POST
- id : String
- transactionsFilename : String
- loadProductCatalog() : void
- loadTransactions() : void
+ createTransaction() : Transaction
+ getId() : String

Util

ProductReader
- filename : String
+ nextProduct() : ProductSpecs

TransactionReader
- filename : String
+ nextTransaction() : Transaction

1..*   1   1   1   1   1   1   1   1   0..1   1..*   1   1   1   1   1   1   1

sd 3 Sequence Diagram0