0360-322 Winter 2008 Assignment 2
Due: Mar 7, 11:59PM
Email your submission

Part I   True/False

For this part, clearly indicate if each assertion is TRUE or FALSE. _You also need to explain briefly why your answer is TRUE or FALSE._

1. The topics in the subject of object-oriented analysis and design include not only analysis and design from an object-oriented perspective, but also requirement analysis.
   Answer:

2. The term "Iterative development process" means the Unified Process (UP).
   Answer:

3. The Vision document is an executive summary of a project that describes briefly the project, as a context for the major players to establish a common vision of the project. Therefore, it should not be changed during construction phase in the UP.
   Answer:

4. You must create a domain model when you apply agile UP in a project.
   Answer:

Part II   Short Answers

1. What artefact(s) in the UP could possibly capture the functional requirements of a project?
Your Answer:

2. Explain in your own words the terms (1) *use case,* (2) *use case instance, and* (3) *scenario.* What is the relationship among these terms? Use your own words, simply copying and pasting from textbook is not acceptable.
 Your Answer:

3. Does a SSD show every system event that the external actor can possibly generate in a use case? Explain your answer.
Your Answer:
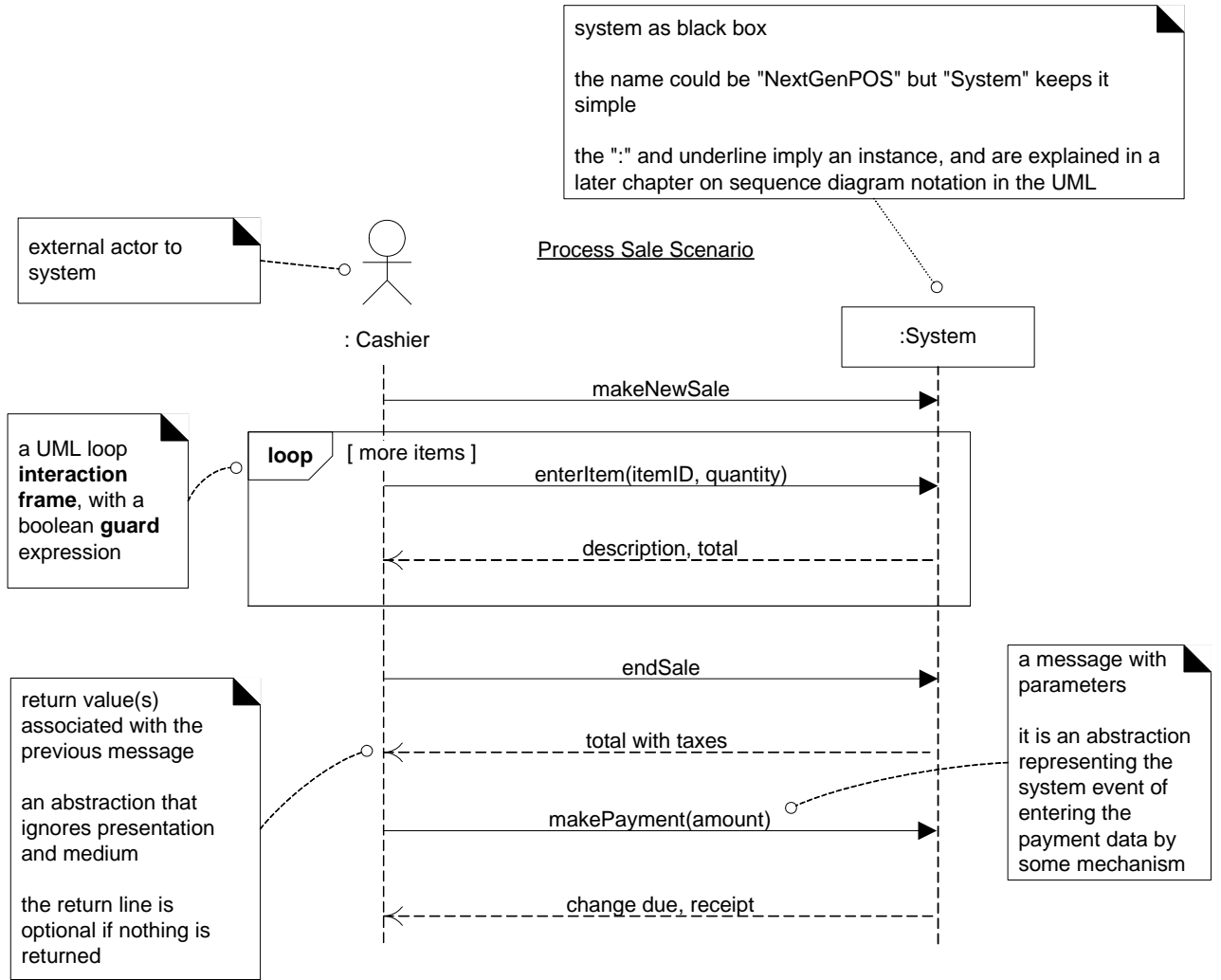
Part III   Modeling Question

Assume that you are in the team developing the NextGen POS System. A simple basic cash-only scenario of the Process Sale use case was studied in the first iteration of the elaboration phase in the book, in which a SSD and a partial domain model were created (see Appendix). Also some production-quality programming was done as well.

Suppose you are the leader and responsible for all the activities in the first iteration, instead of choosing the cash-only scenario of the Process Sale use case, you decide to choose the happy path of the credit-only scenario of Process Sale use case for the first iteration of the elaboration phase (see Appendix).
1) Explain how you would organize all activities that could possibly happen in this iteration.
2) Draw the SSD.
3) Show the expanded domain model.

For questions 2) and 3), it is not enough to simply show the SSD and the domain model as in the appendix, you also need to provide sufficient information to explain how and why you produce the SSD and domain model.
Your Answer:

APPENDIX

```
                                      Records-sale-of
┌─────────────────┐
│                 │        ┌──────────┐       ┌──────────┐            ┌──────────────┐
│                 │        │  Ledger  │       │ Product  │            │   Product    │
│                 │        │          │       │ Catalog  │  Contains  │  Description │
│                 │        ├──────────┤       ├──────────┤            ├──────────────┤
│                 │        │          │       │          │ 1      1..*│ itemID       │
│                 │        └──────────┘       └──────────┘            │ description  │
│                 │           1      1                                │ price        │
│            0..1 │                          1                        └──────────────┘
│      ┌──────────┴──┐   Records-                                          │
│      │   Sales     │   accounts-           Used-by         Describes  │ *
│      │  LineItem   │   for                 *                          │
│      ├─────────────┤                   ┌──────────┐                   │
│      │ quantity    │              1     │  Store   │   Stocks          │
│      └─────────────┘                    ├──────────┤        1..*   ┌──────┐
│          │ 1..*                         │ name     │ 1      *      │ Item │
│   Contained-in                          │ address  │──────────────├──────┤
│          │ 1                            └──────────┘               │      │ 1..*
│      ┌─────────┐  Logs-                      │ 1                   └──────┘
│      │  Sale   │  completed              Houses
│      ├─────────┤ *                       1..*
│      │ dateTime│   Captured-on          ┌──────────┐
│      │ / total │   0..1          1      │ Register │
│      └─────────┘                        ├──────────┤
│        1    1                           │ id       │
│   Paid-by   Is-for                      └──────────┘
│      1      1                                │ 1
│  ┌───────────┐  ┌──────────┐            ◄ Works-on
│  │CashPayment│  │ Customer │                         1
│  ├───────────┤  ├──────────┤              ┌──────────┐
│  │amountTendered│ │          │             │ Cashier  │
│  └───────────┘  └──────────┘              ├──────────┤
│                                           │ id       │
                                            └──────────┘
```

system as black box

the name could be "NextGenPOS" but "System" keeps it simple

the ":" and underline imply an instance, and are explained in a later chapter on sequence diagram notation in the UML

external actor to system

Process Sale Scenario

: Cashier

:System

makeNewSale

a UML loop **interaction frame**, with a boolean **guard** expression

loop [more items]

enterItem(itemID, quantity)

description, total

endSale

return value(s) associated with the previous message

an abstraction that ignores presentation and medium

the return line is optional if nothing is returned

total with taxes

makePayment(amount)

change due, receipt

a message with parameters

it is an abstraction representing the system event of entering the payment data by some mechanism

4

Main Success Scenario (or Basic Flow) of Process Sale:

1. Customer arrives at POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules.

   Cashier repeats steps 3-4 until indicates done.

5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. ***Customer pays and System handles payment.***
8. System logs completed sale and sends sale and payment information to the external Accounting system (for accounting and commissions) and Inventory system (to update inventory).
9. System presents receipt.
10. Customer leaves with receipt and goods (if any).

7b. Paying by credit:

1. Customer enters their credit account information.
2. System displays their payment for verification.
3. Cashier confirms.

   3a. Cashier cancels payment step:

   1. System reverts to "item entry" mode.
4. System sends payment authorization request to an external Payment Authorization Service System, and requests payment approval.

   4a. System detects failure to collaborate with external system:

   1. System signals error to Cashier.

   2. Cashier asks Customer for alternate payment.

5. System receives payment approval, signals approval to Cashier, and releases cash drawer (to insert signed credit payment receipt).

   5a. System receives payment denial:

   1. System signals denial to Cashier.

   2. Cashier asks Customer for alternate payment.

5b. Timeout waiting for response.

       1. System signals timeout to Cashier.

       2. Cashier may try again, or ask Customer for alternate payment.

6. System records the credit payment, which includes the payment approval.
7. System presents credit payment signature input mechanism.
8. Cashier asks Customer for a credit payment signature. Customer enters signature.
9. If signature on paper receipt, Cashier places receipt in cash drawer and closes it.