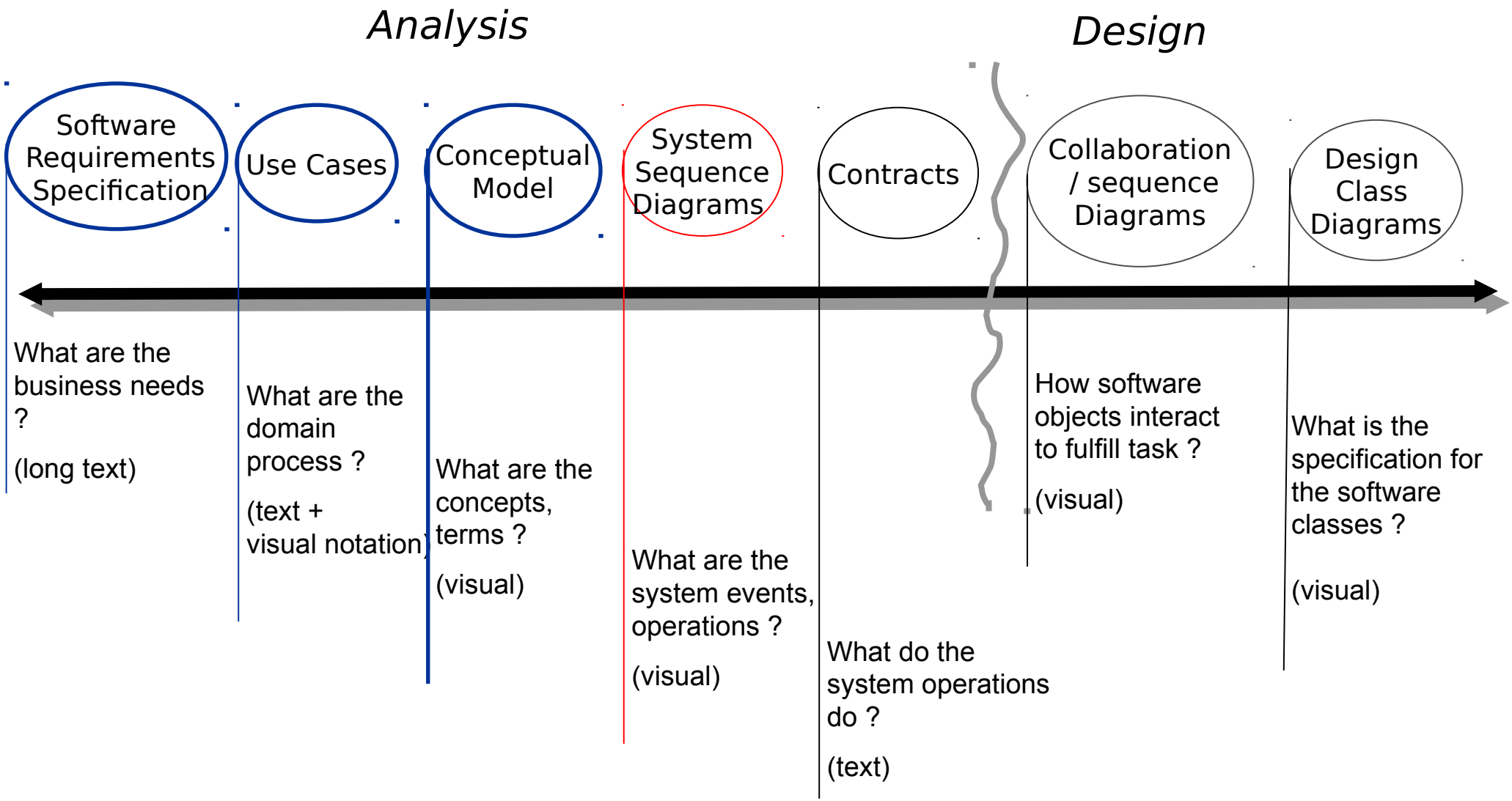
מבוא לתיכנות מונחה עצמים – תהליכי ניתוח ותיכנון

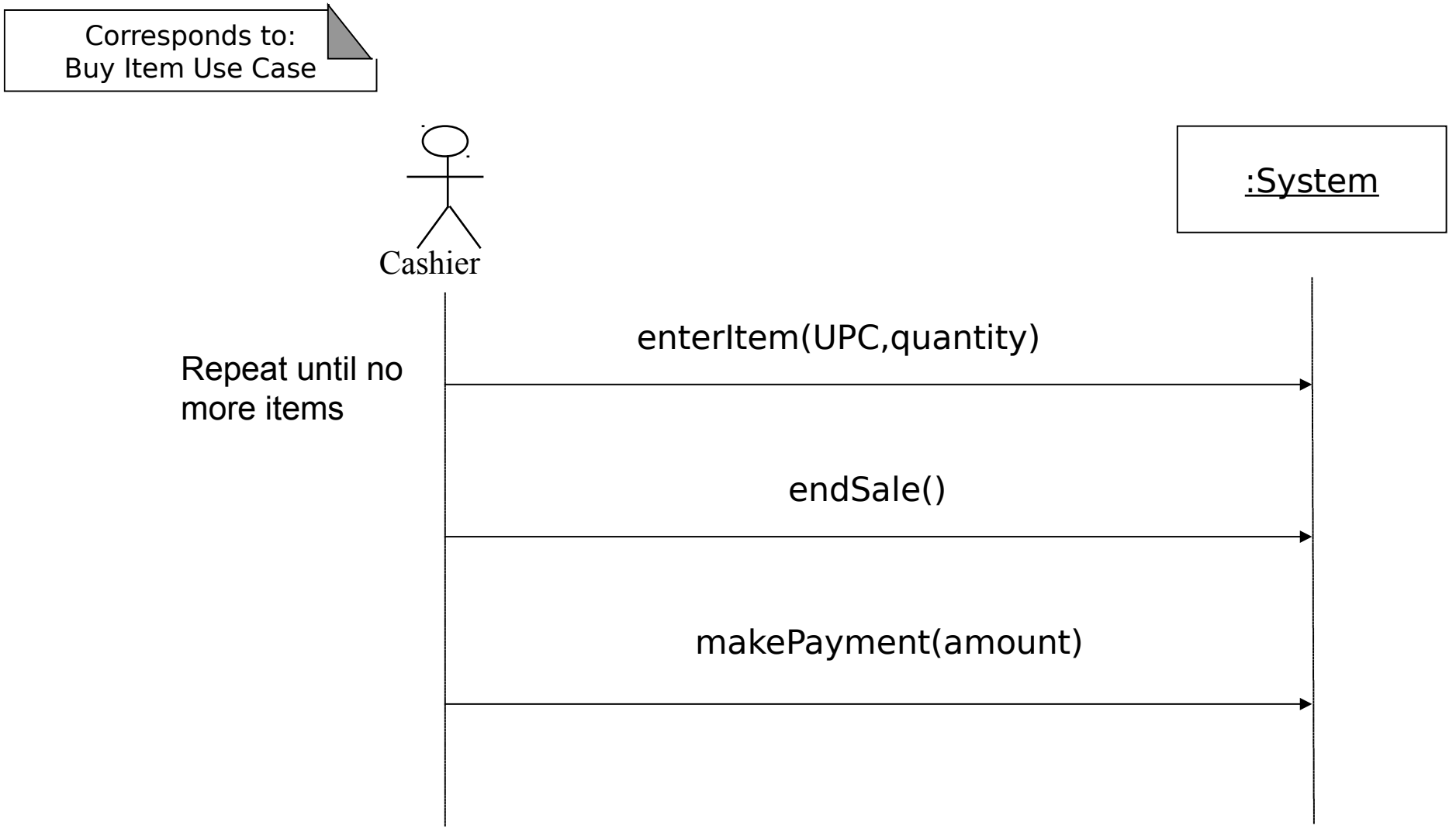
מקורות:

- Craig Larman, “Applying UML and Patterns”
- Dr. Rahmi Marasli, CMU course
- Dr. Mira Balaban, BGU course
- GWT Lecturer, “UML Tutorials”,
  https://www.youtube.com/watch?v=y7grsHY9Fa0&list=PLoWne5q-c9E_Q2_eAUZKPDA5K0V-O5zXs

תרשים סדרת-פעולות



תרשים סדרת פעולות - דוגמה



תרגום תרחיש-שימוש לתרשים-סדרת-פעולות

Actor Actions:

1. This use case begins when a Customer arrives at a POST checkout system with items to purchase

2. **Cashier records the identifier for each item.** If there is more than one of the same item, **Cashier enters the quantity**

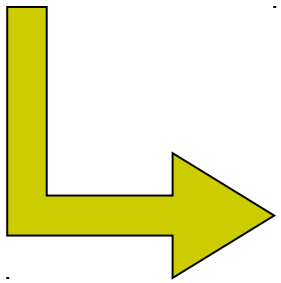
4. **On completion** of item entry, **Cashier indicates to POST that item entry is complete**

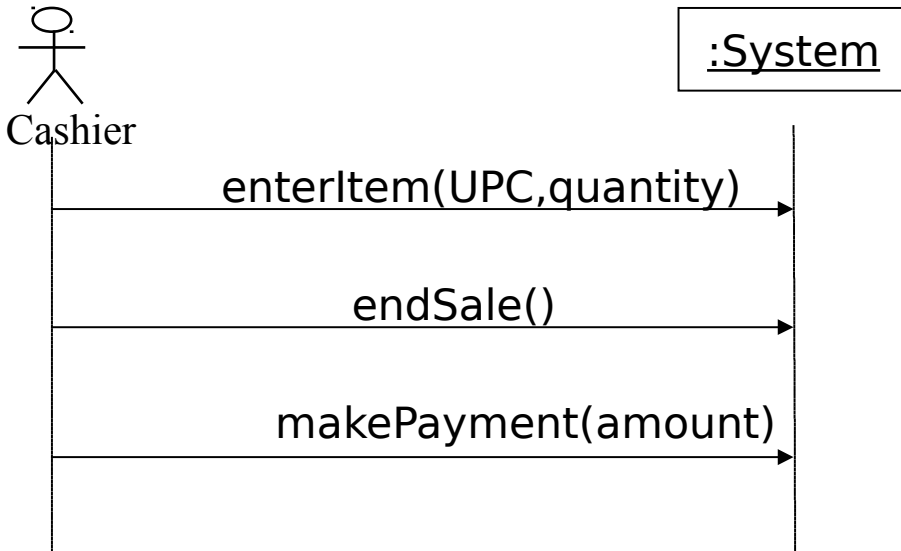
6. **Cashier tells Customer the total**

System Response:

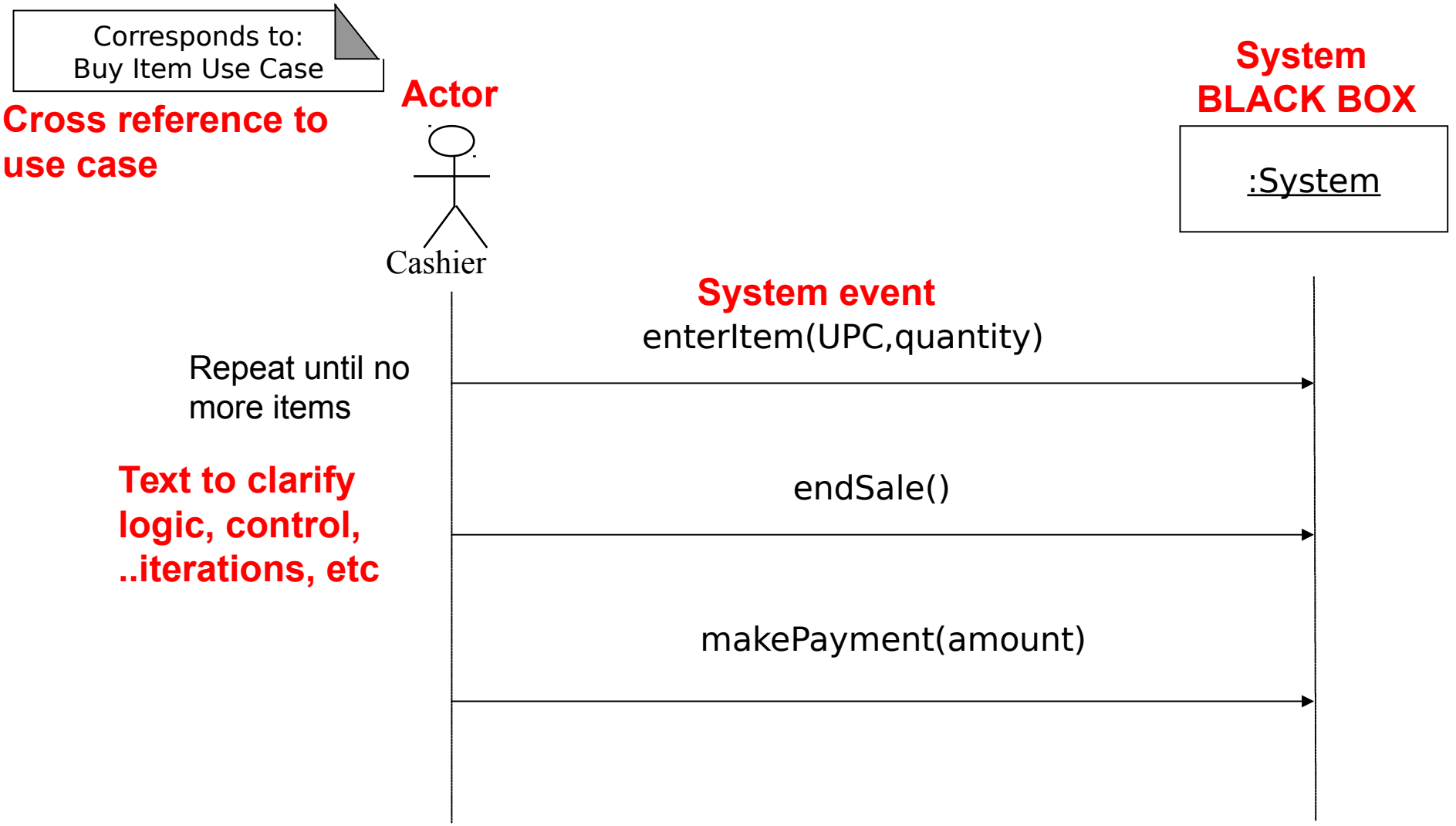
3. Determines item price and adds item info to the running sales transaction
   Description and price of the current item is presented

5. Calculates and presents sale total

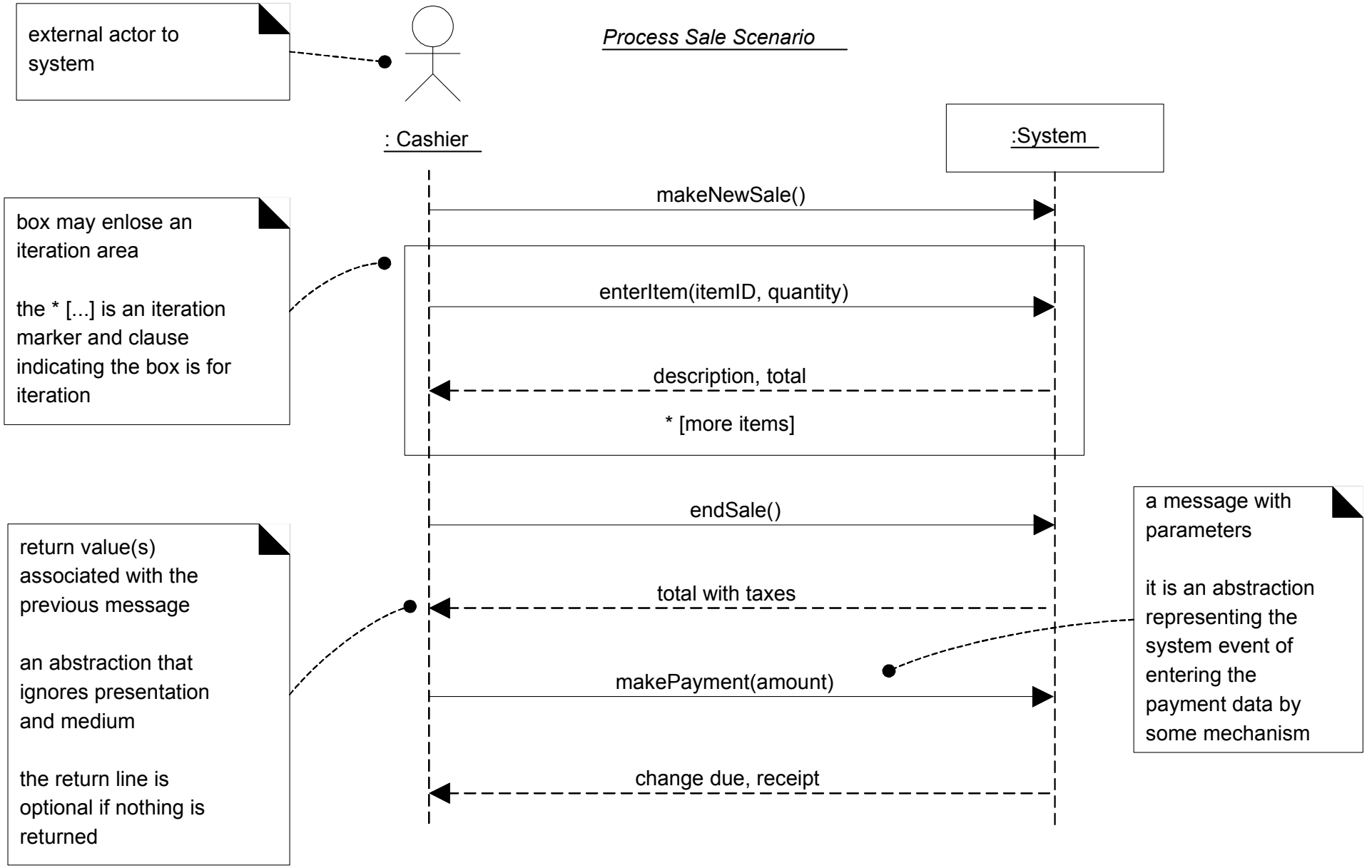




תרשים סדרת פעולות



הוספת הפעולות שמבצעת המערכת



איך לקרוא לפעולות בתרשים?

- לפי הכוונה ולא לפי הפעולה הפיסית. דוגמאות:

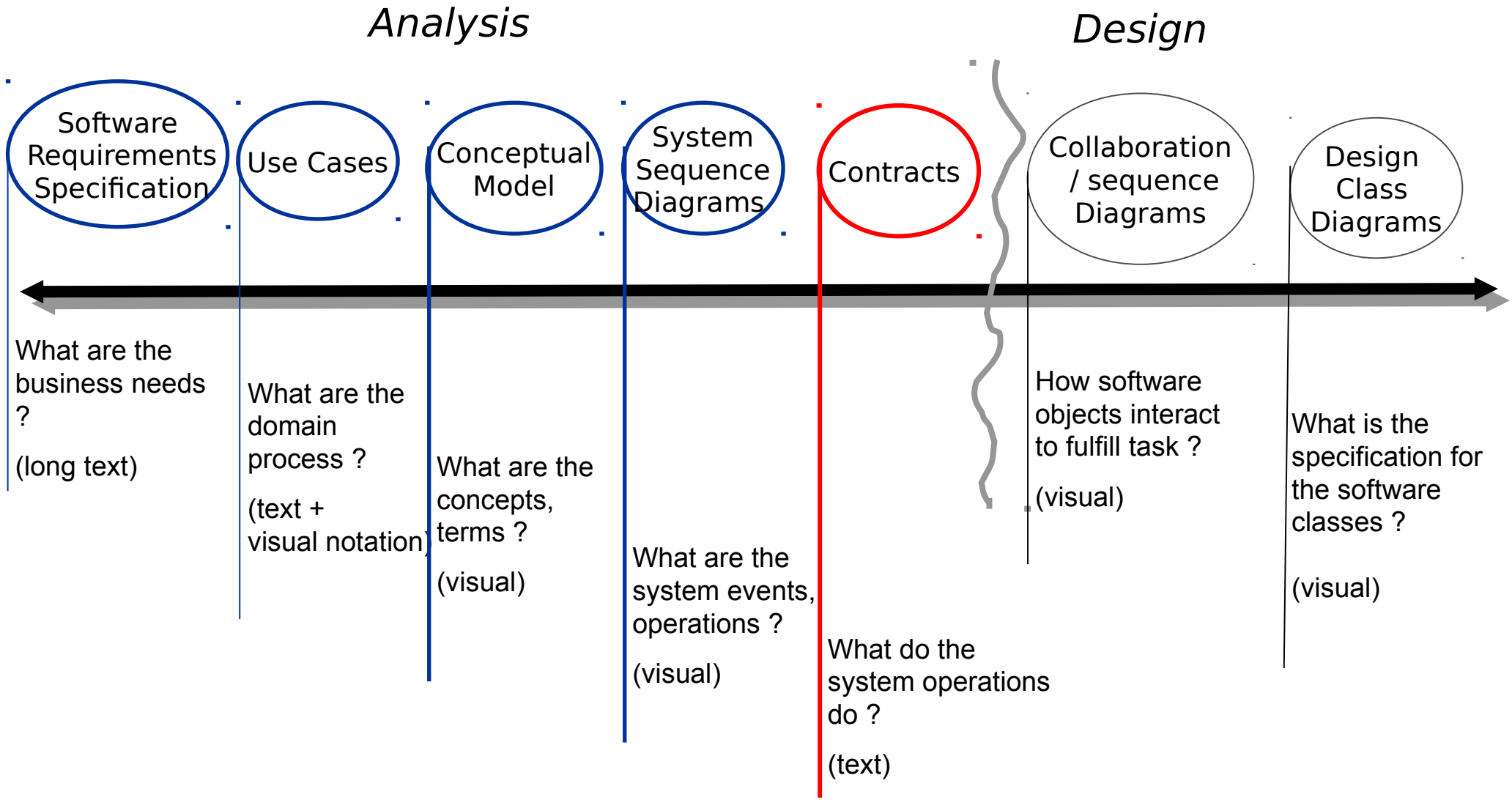
יותר טוב	פחות טוב
endSale()	ESCPressed()
makePayment(amount)	enterAmountTendered(amount) enterPayment(amount)

חוזי-פעולות:

מה כל פעולה אמורה לבצע.

לכל פעולה יש **דרישות** (pre-conditions)

והבטחות (post-conditions).



חוזים – דוגמה – הפעולה enterItem

- **Name**: enterItem(upc:number, quantity:integer)
- **Responsibility**: Enter (record) sale of an item and add it to sale. Display item description and price.
- **Type**: System
- **Cross References**: System Functions: R1.1, R1.3, R1.9.
  Use Cases: Buy Items
- **Notes**: Use fast dbase access
- **Exceptions**: If UPC is invalid, indicate an error
- **Pre-Conditions**: UPC is known to system - דרישות

חוזים – דוגמה – הפעולה enterItem

- **Post-Conditions - הבטחות**
  - If new sale, a *Sale* was created (*instance creation*)
  - If new sale, new *Sale* was associated with *POST* (*association formed*)
  - A *SalesLineItem* was created (*instance created*)
  - *SalesLineItem* was associated with *Sale* (*association formed*)
  - *SaleLineItem.quantity* was set to *quantity* (*attribute modification*)
  - *SaleLineItem* was associated *with ProductSpecification based on UPC* match (*association formed*)

חוזים – סעיפים בחוזה

- **Name**: Name of operation, and parameters
- **Responsibilities**: An informal description of responsibilities this operation must fulfill
- **Type**: Name of type (concept, software class, interface)
- **Cross References**: System function reference numbers, use cases, etc.
- **Notes**: Design notes, algorithms, etc.
- **Exceptions**: Exceptional cases

חוזים – סעיפים בחוזה

- **Output**: Non-UI outputs such as messages or records that are sent outside of system

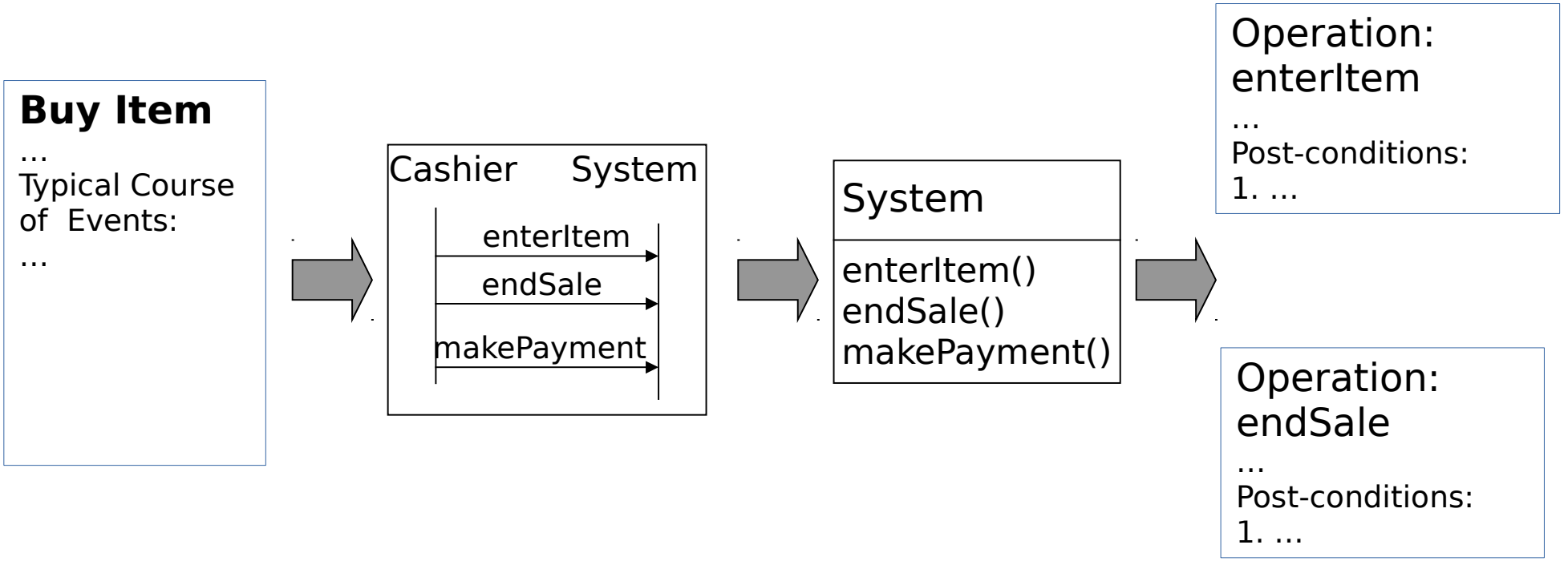
- **Pre-Conditions**: Assumptions about the state of system before execution of operation

- **Post-Conditions**:
  - The state of system after completion of operation

איך כותבים חוזה?

- מזהים פעולות בתרשים-סדרת-פעולות.
- לכל פעולה שמתבצעת ע"י המערכת, יוצרים חוזה.
- כותבים את סעיף ה"אחריות" (Responsibility) – מה תפקיד הפעולה בסדרת-הפעולות?
- כותבים את סעיף ה"הבטחות"(Post-conditions) – מה המצב אחרי שהפעולה הסתיימה בהצלחה?
- כותבים את סעיף ה"דרישות" (Pre-conditions) – מה צריך להיות המצב לפני שהפעולה התחילה, כדי שהיא תוכל להתבצע בהצלחה?

מתרחישי-שימוש לחוזי-פעולות



Use Case

System Sequence Diagram

System Operations

Contracts

חוזי-פעולה – דוגמאות נוספות

Contract for makePayment

Contract

Name:	makePayment (amount : Number or Quantity)
Responsibilities:	Record the payment, calculate balance and print receipt.
Type:	System
Cross References:	System Functions: R2.1 Use Cases: Buy Items
Notes:	
Exceptions:	If the sale is not complete, indicate an error. If the amount is less than the sale total, indicate an error.
Output:	
Pre-conditions:	
Post-conditions:	

- A *Payment* was created (instance creation).
- *Payment.amountTendered* was set to *amount* (attribute modification).
- The *Payment* was associated with the *Sale* (relationship formed).
- The *Sale* was associated with the *Store*, to add it to the historical log of completed sales (relationship formed).

Contract for StartUp

Contract

Name:	startUp()
Responsibilities:	Initialize the system.
Type:	System
Cross References:	
Notes:	
Exceptions:	
Output:	
Pre-conditions:	
Post-conditions:	

- A *Store, POST, ProductCatalog and ProductSpecifications* have been created (*instance creation*).
- *ProductCatalog* was associated with *ProductSpecifications* (*association formed*).
- *Store* was associated with *ProductCatalog* (*association formed*).
- *Store* was associated with *POST* (*association formed*).
- *POST* was associated with *ProductCatalog* (*association formed*).

Contract for endSale

Contract

Name:	endSale()
Responsibilities:	Record that it is the end of entry of sale items, and display sale total.
Type:	System
Cross References:	System Functions: R1.2 Use Cases: Buy Items
Notes:	
Exceptions:	If a sale is not underway, indicate that it was an error.
Output:	
Pre-conditions:	UPC is known to the system.
Post-conditions:	

- *Sale.isComplete* was set to *true* (attribute modification).

מתרחיש-שימוש לחוזה-פעולה - דוגמה

מערכת כניסה (login) לרשת מחשבים

תרחיש שימוש

במערכת הפעלה מרובת משתמשים (רשת מחשבים כמו באוניברסיטה) :

משתמש מצטרף למערכת שכבר נמצאת בפעולה . המשתמש מודיע למערכת שהוא רוצה לבצע Login למחשב מסויים . המערכת מוודאת ששם המשתמש והסיסמא שלו מתאימים . (בהסתמך על מאגר משתמשים מורשים הקיים במערכת)

אם הסיסמא לא נכונה , המערכת מוציאה הודעת שגיאה .

אם שם המשתמש והסיסמא תואמים , המערכת מייצרת חשבון פעיל עבור המשתמש הנוכחי ומקשרת אותו למחשב המבוקש .

כשהמשתמש רוצה לסיים את עבודתו , הוא מגיש בקשה ל Logout . החשבון הפעיל נסגר , והמערכת מעדכנת במאגר המשתמשים את סה"כ זמן העבודה של המשתמש .

Use case : login to system / logout of system

Use case : login to system / logout of system

Actors : user

Purpose : creating a new account for user , and connecting it to a specific computer
Closing the account , and presenting the total time of usage , when the user wants to logout

Overview : a user wants to login to a specific computer in the system . he enters his login and password . the system checks the correctness of the password m, based on a list of authorized users .
If the password is wrong , the system will display an error message . else the system will create an active account for the user , and will connect it to the specific computer.
When the user wants to finish his work , he asks the system to logout , the account is closed , and the system will save the total time of usage .

Type : primary and essential

Cross references : …

Typical cours of events

Actor action	System response
1. the user requests to log into a specific computer (enters user name , password and computer's name)	
	2. the system checks the correctness of the password. If the password is wrong , the system will display an error message . else the system will connect the user to the requested computer .
3. the user gives a request to logout	
	4. the system closes the account , and saves the total time of usage .

login contract

name : login (UserName:string ,Password:string ,ComputerName:string ,Time:time)

responsibilities : create a new user account , and connect it to the requested computer

type : system

cross reference : use case : login to system / logout of system

notes : use super fast database accses

exeptions : if the password is wrong , indicate an error

output :

preconditions :

- the specific computer is known to the system
- user login and password is known to the system

postconditions :

- an account was created (instance creation)
- the account was linked to the system (link formed)
- the account was linked to the requested computer (link formed)
- account start-time was set (attribute modification)