

**UNIVERSITETI AAB**

**FAKULTETI I SHKENCES KOMPJUTERIKE**

**INXHINIERIA E SOFTVERIT**

**Modelimi i Sistemeve**

**MSc. Fatime Gashi**

- ⦿ Modelet janë të dobishme për t'i kuptuar problemet si dhe për të komunikuar me të gjithë që janë të involvuar në problem. Ne ndërtojmë modele te sistemeve komplekse sepse nuk mund t'i kuptojmë sistemet e tilla si tërësi.
- ⦿ Modelimi e mundëson të kuptuarit më të mirë të kërkesave, dizajn më të pastër dhe sisteme me mirëmbajtje më të mirë.

- Modelimi është aktivitet që është përdorur në zhvillimin e softverit me vite. Definimi i modeleve e lehtëson ndarjen e aplikacionit kompleks apo të një sistemi të madh në pjesë të vogla dhe diskrete të cilat mund të studiohen individualisht.
- UML –ja e mundëson paraqitjen e sistemit duke përdorur simbole të cilat janë të lehta për t'u kuptuar e ato simbole quhen 9 diagramet e UML-se.

- Edhe pse ekzistojnë shumë teknika dhe vegla për modelim ne do të përqëndrohemi në modelimin e sistemeve dhe aplikacioneve të orientuara në objekte, duke e përdorur UML-ne (Unified Modelling Language) dmth Gjuhën e Unifikuar për Modelim.
- Pra UML e mundëson paraqitjen e karakteristikave të sistemit duke përdorur shënime.

- ⦿ Dhe shtrohet pyetja se përse UML-ja është opcion i preferuar, kurse përgjigjja është “Standardizimi”. Me heret janë përdorur metodologji të ndryshme si p. sh. Metodologjia e Rumbaugh, Booch, Jacobson, mirëpo problemi me këto ka qenë se edhe pse i kanë pasur përparësitë e tyre ato në esencë nuk kanë qenë të ngjajshme.
- ⦿ Dhe nëse keni punuar në disa projekte ku secili ka përdorur metodologji të ndryshme, atehere është dashur t’i njihni të gjitha metodologjitë. Kurse UML-ja i unifikon principet e dizajnit të të gjitha këtyre metodologjive në një gjuhë të vetme që mund të përdoret për të gjitha sistemet e orientuara në objekte.

- ⦿ Përderisa metodologjitë tjera janë përqëndruar më shumë në dizajn dhe dizajn të detalizuar, metodologjia e UML-së përqendrohet më shumë në kërkesa, analizë, dizajn dhe në implementim.
- ⦿ Për shembull nëse nevojitet të modelohen kërkesat për një sistem mund t'i përdorni Use Case diagramet pa i përdorur diagramet tjera të UML-së.

# Klasifikimi i UML Diagrameve – Statik, Dinamik, dhe Implementues

- Nje sistem i softverit mund te kete karakteristika statike, dinamike dhe implementuese,
- **Statike:** Kjo karakteristike tregon se prej cfare pjese eshte i perbere sistemi.
- **Dinamike:** Paraqet vetite e sjelljes se sistemit, p.sh. menytrat se si sillet sistemi si reagim ndaj ndodhive apo veprimeve te caktuara

- **Implementuese:** Kjo eshte karakteristike e re e sistemit qe i pershkruan elementet e ndryshme qe nevojiten per ta implementuar sistemin.
- prandaj edhe secili diagram i UML-se karakterizohen si ne vijim:
- **Statike**
- Use case diagrami
- Class diagrami



- **Dinamike**

- Object diagramami

- State diagramami

- Activity diagramami

- Sequence diagramami

- Collaboration diagramami

- **Implementuese**

- Component diagramami

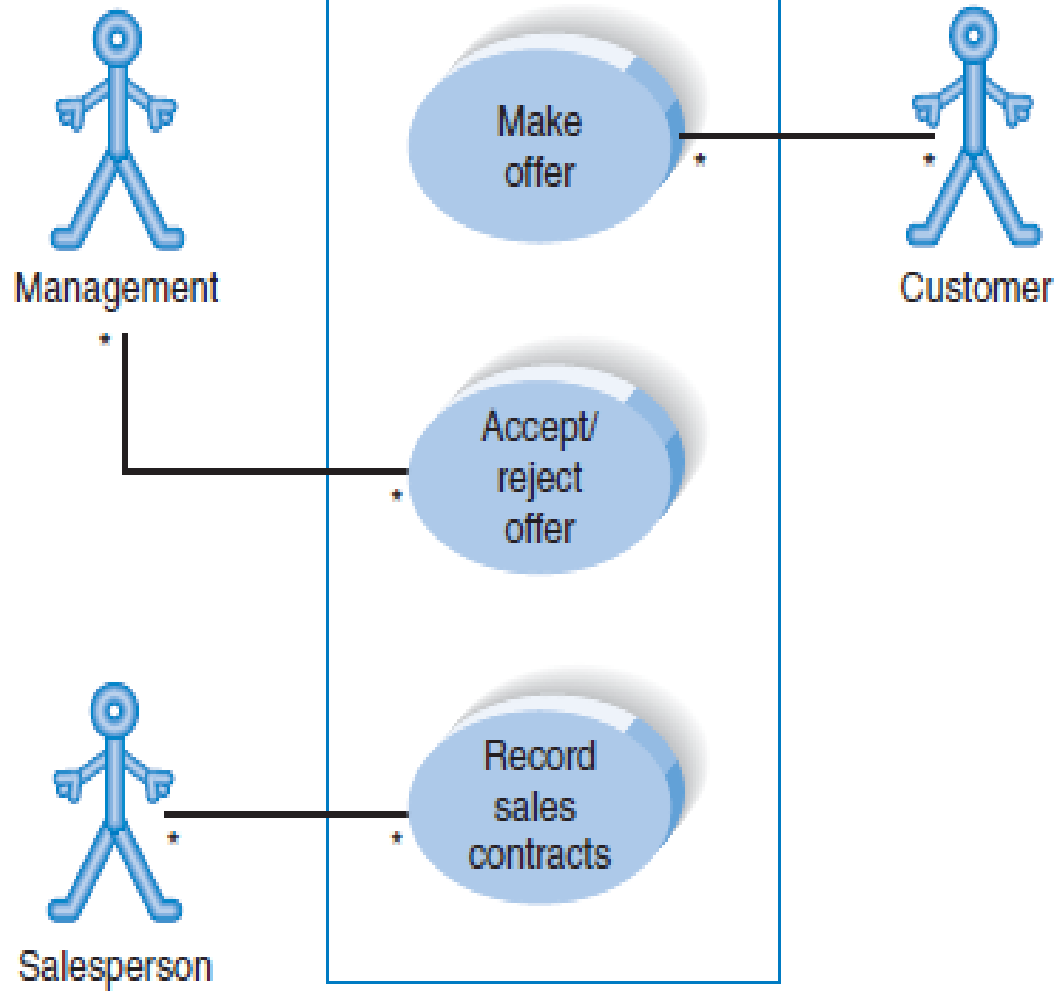
- Deployment diagramami

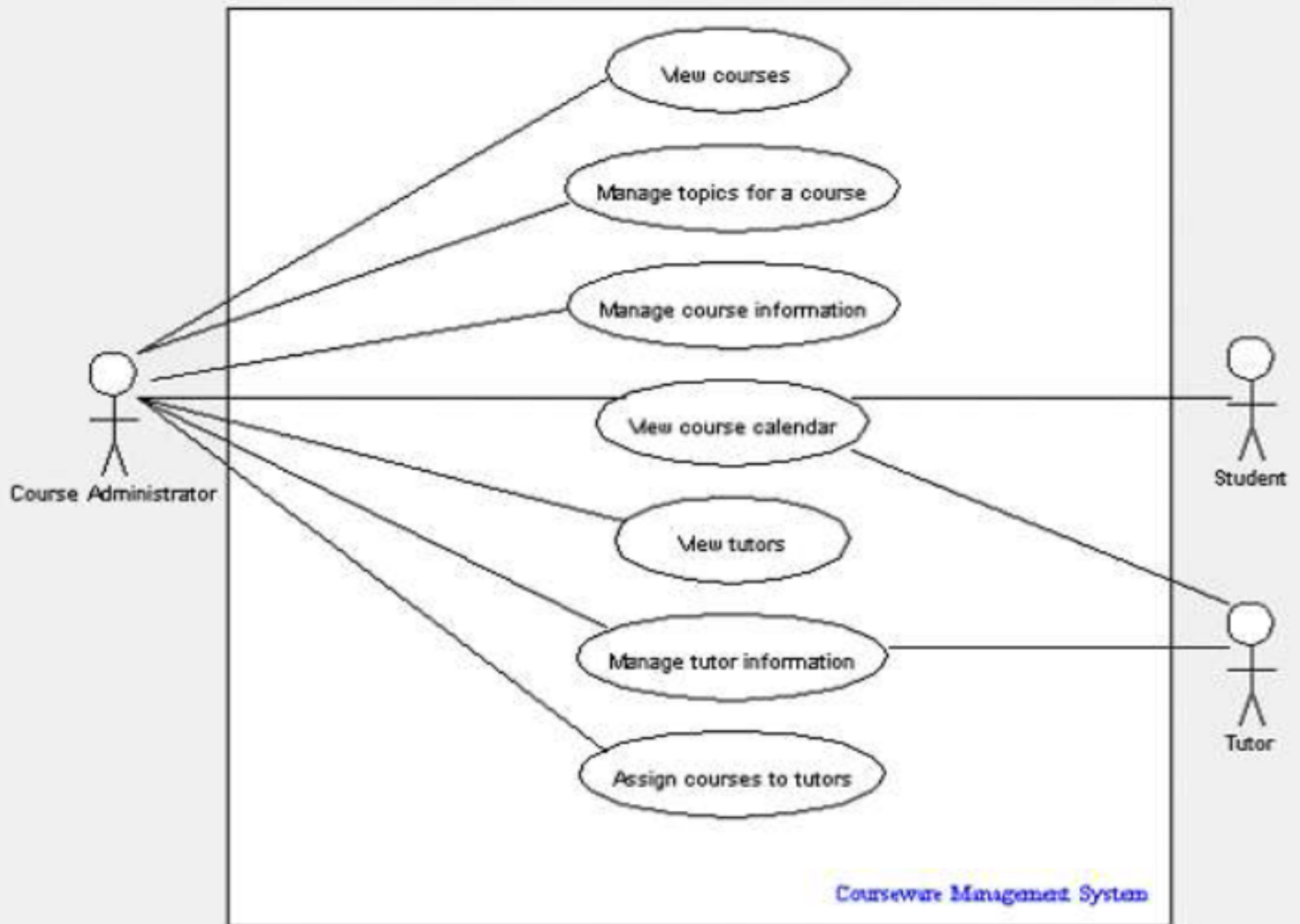
# Diagramet e UML-se

- Karakteristike e UML-se eshte se asnje diagram nuk mund t'i paraqese te gjitha elementet e ndryshme te nje sistemi. Prandaj, UML-ja perbehet nga nente diagrame qe mund te perdoren per ta modeluar sistemin, ne faza te ndryshme pergjate tere procesit te zhvillimit te tij.

- ⦿ Diagramet e UML-se jane:
- ⦿ **1. Use case diagrami** perdoret per t'i identifikuar elementet dhe proceset primare qe e formojne sistemin. Elementet primare njihen si “aktoret” dhe proceset quhen “use cases”. Diagrami use case tregon cilet aktore nderveprojne me secilin use case.
- ⦿ P.sh. ne figure eshte paraqitur nje Use Case diagram per sistemin e shitjes se veturave dhe nga diagrami mund te shihet se konsumatoret, personat per shitje, dhe personeli i menaxhmentit do ta perdor kete sistem per te bere oferta, pranuar/refuzuar oferta, dhe regjistruar kontrata te shitjes.

## Vehicle Sales System





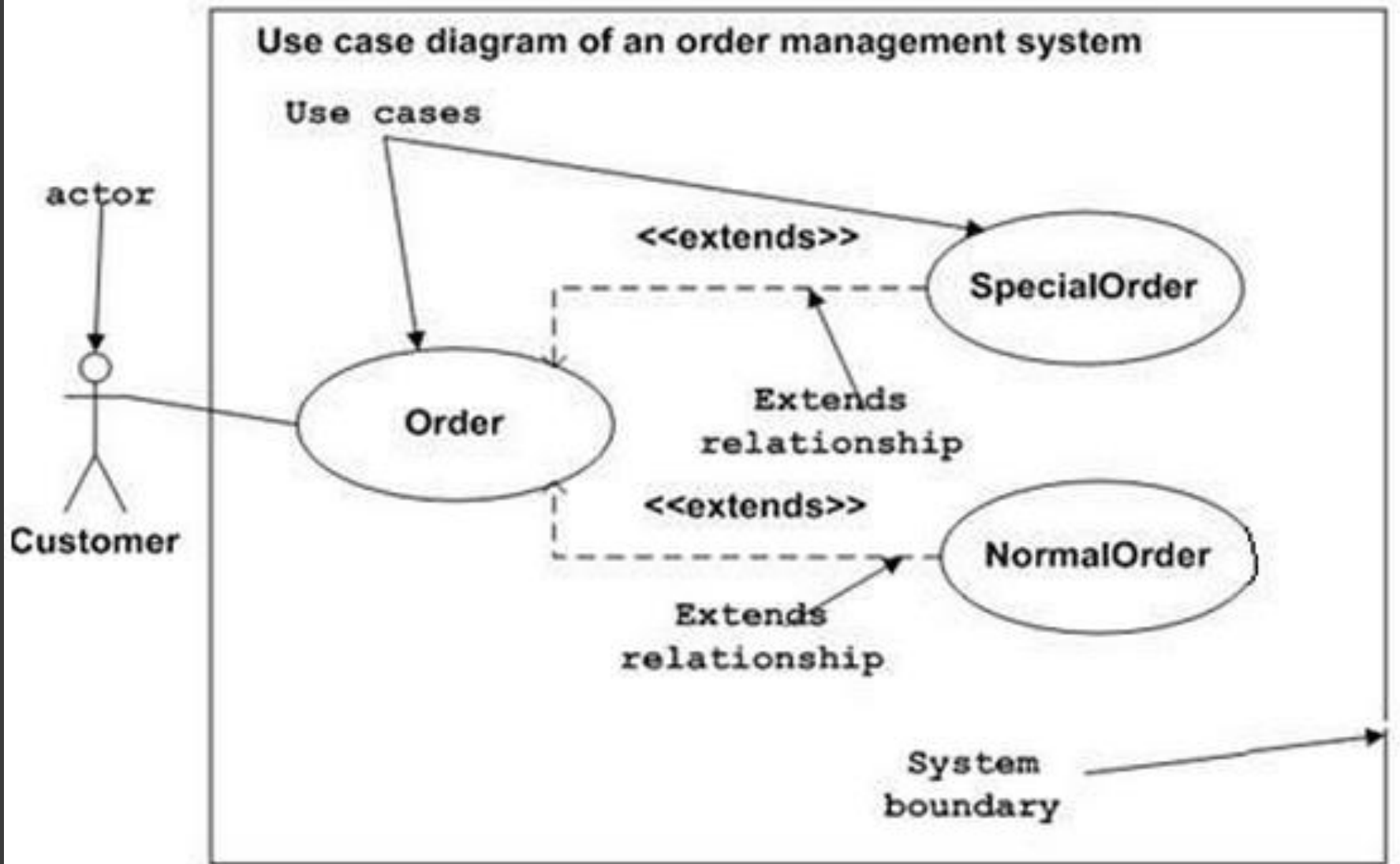
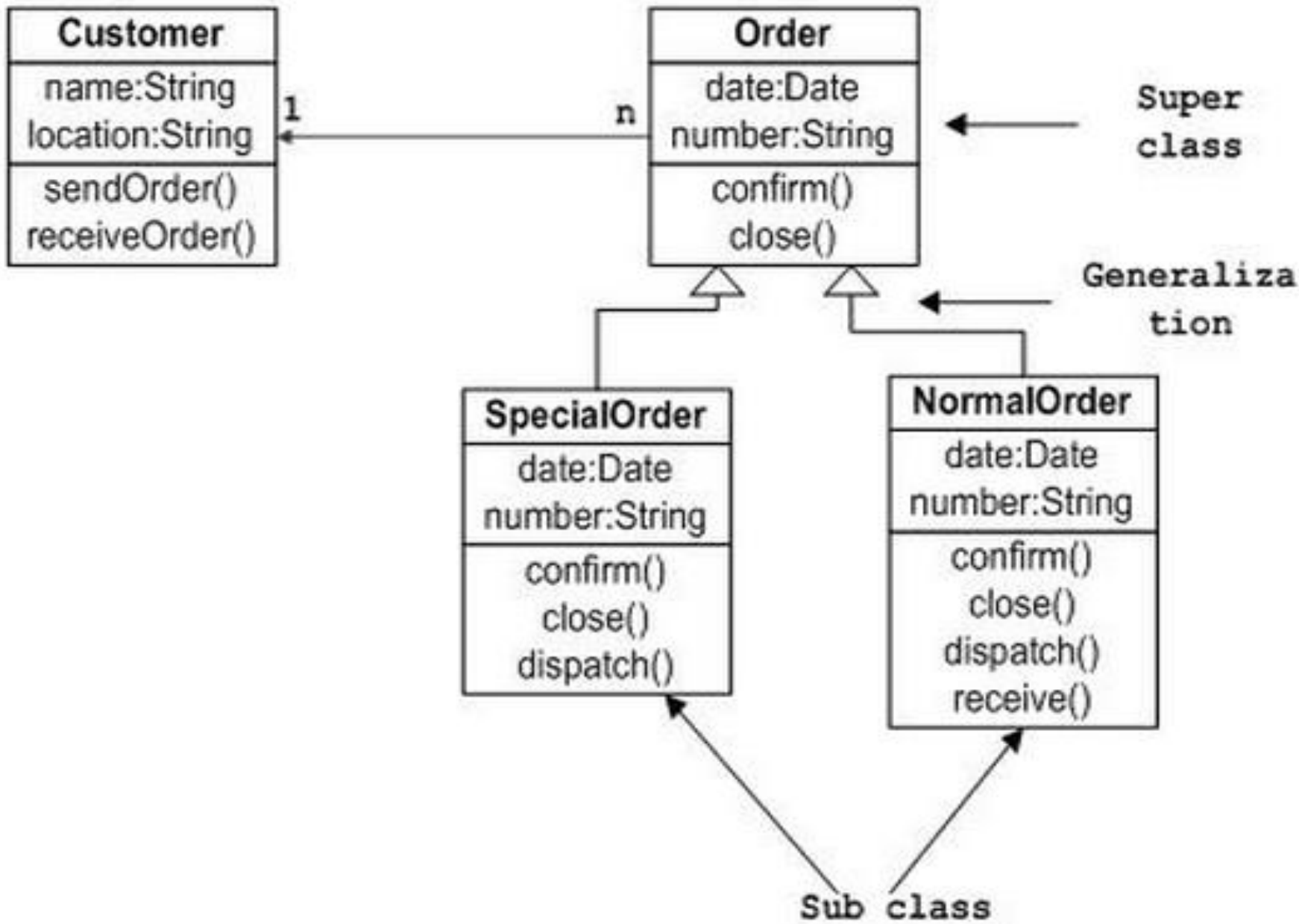
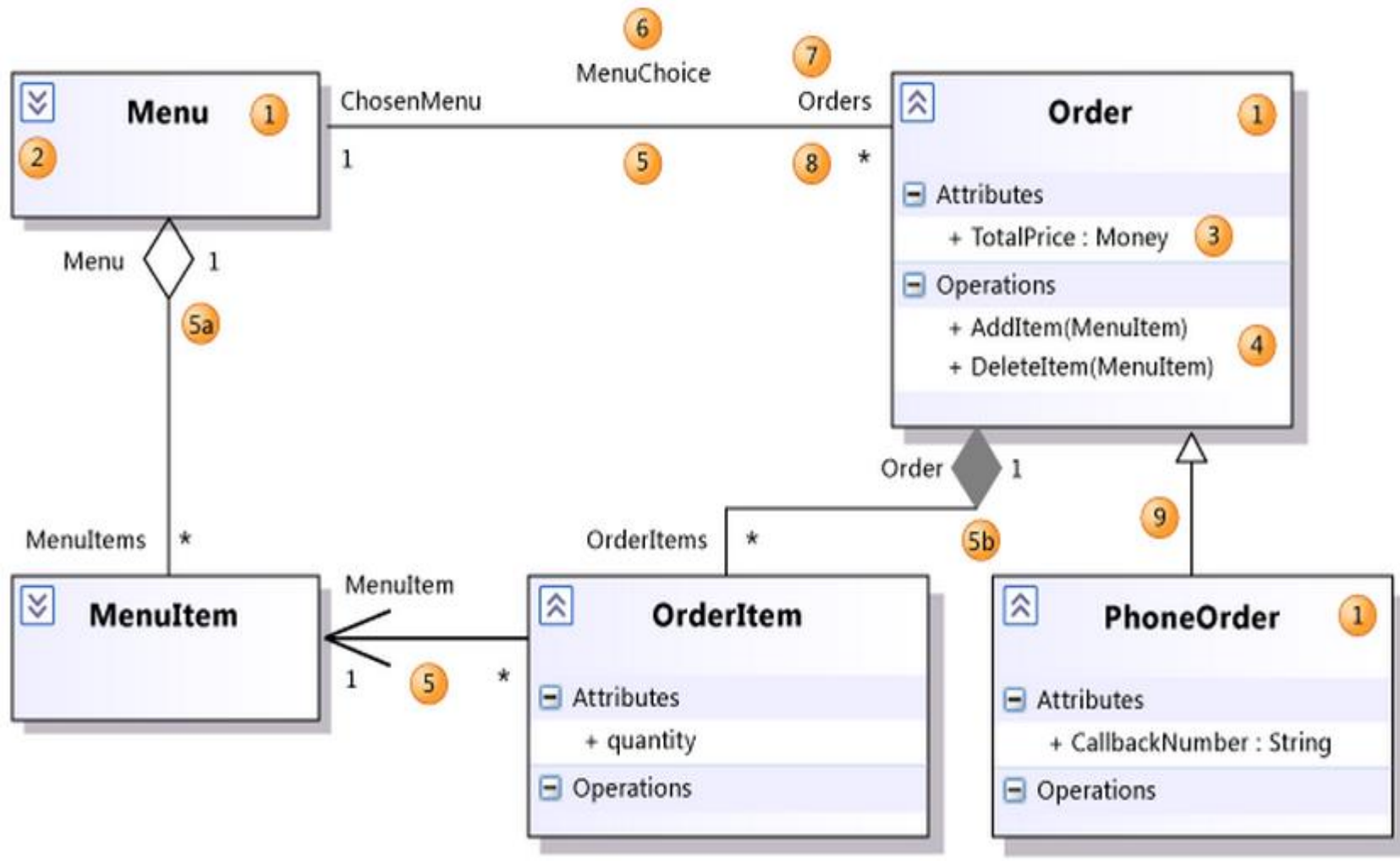


Figure: Sample Use Case diagram

- **2. Class diagrami** perdoret per ta permiresuar use case diagramin dhe per te definuar dizajn te detalizuar te sistemit. Class diagrami i klasifikon aktoret e definuar ne use case diagram ne nje komplet te klasave te nderlidhura.
- Relacioni ndermjet klasave mund te jete ose “eshte” ose “ka”. Secila klase ne class diagram mund te siguroje funksionalitete te caktuara. Keto funksionalitete te siguruara nga klasa, quhen “metoda” te klases. Perveq kesaj secila klase mund te kete “attribute” te caktuara qe ne menyre te vecante e identifikojne klasen.







- ③ **3. Object diagrami** eshte lloj i vecante i class diagramit. Objekti eshte nje shembull i klases.
- ③ Kjo ne esence domethene se objekti paraqet gjendjen e klases ne nje moment te caktuar kohor perderisa sistemi eshte duke funksionuar.
- ③ Object diagrami e paraqet gjendjen e klasave te ndryshme ne sistem dhe lidhjet e tyre ne nje moment kohor.

**Graduate School of Business: College**

univName="University of  
Chicago"  
noOfCourses: 1000  
capacity=2000  
areaName="Hyde Park"  
...

**College student no. 1:  
Student**

nameOfStudent="Sam"  
studentId=1  
age=28  
...

**College student no. 2:  
Student**

nameOfStudent="Nancy"  
studentId=2  
age=29  
...

units for testuser

### John's 1st: SavingsAccount

balance = 10,000.00  
id = 1234567890  
interestRate = 1,2  
minimumBalance =

### AgencyBank: Bank

bankname = AgencyBank  
IPaddress = 10.10.127.128  
username = John Doe  
password = johnny  
accounts = 3

### John's 2nd: CheckingAccount

balance = 1,254.76  
id = 987654321

### John's 3rd: CreditCardAccount

balance = 789.14  
id = 4445556667  
creditLimit = 5,000.00  
interestRateOnBalance =  
interestRateOnCashAdvance =

accounts

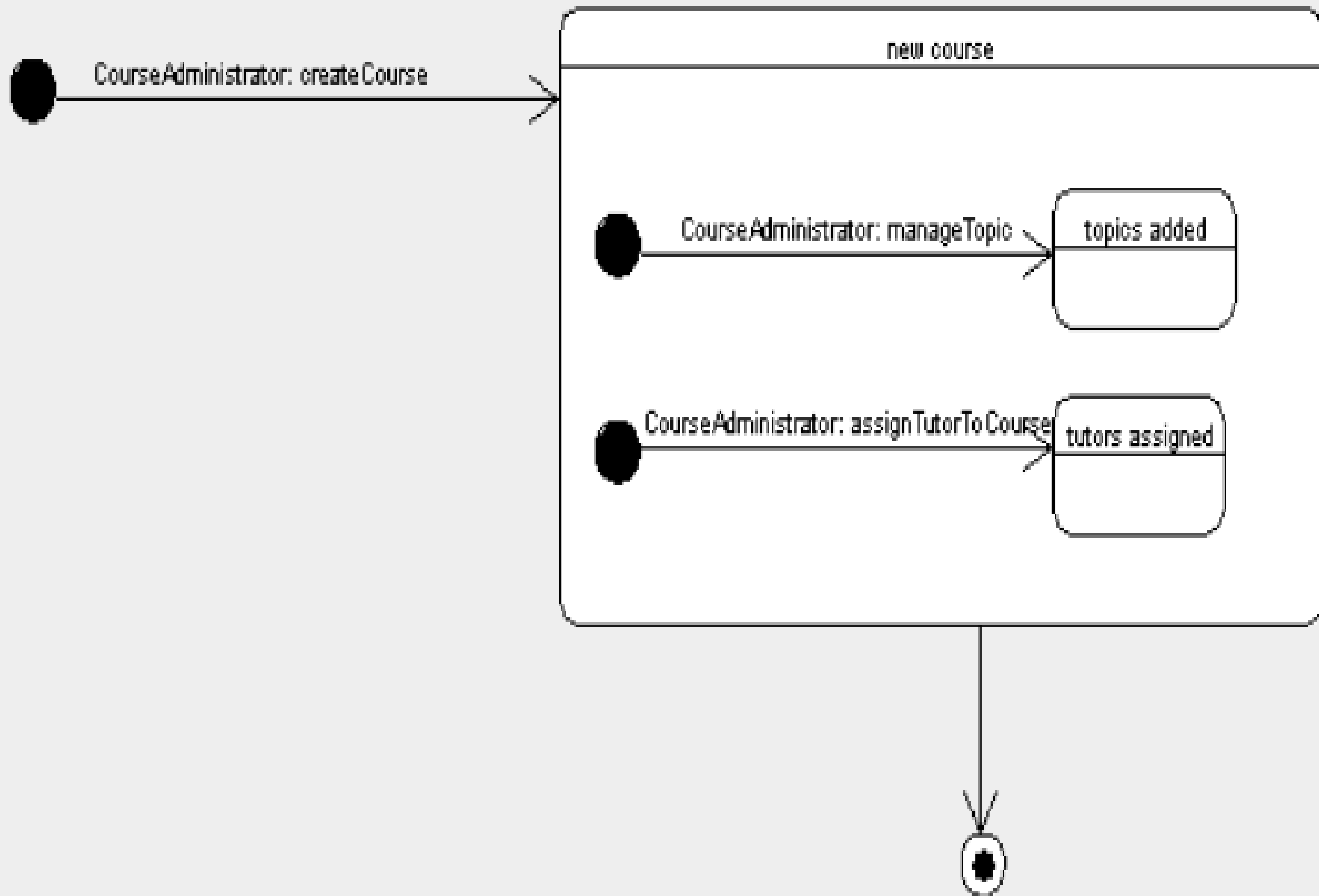
accounts

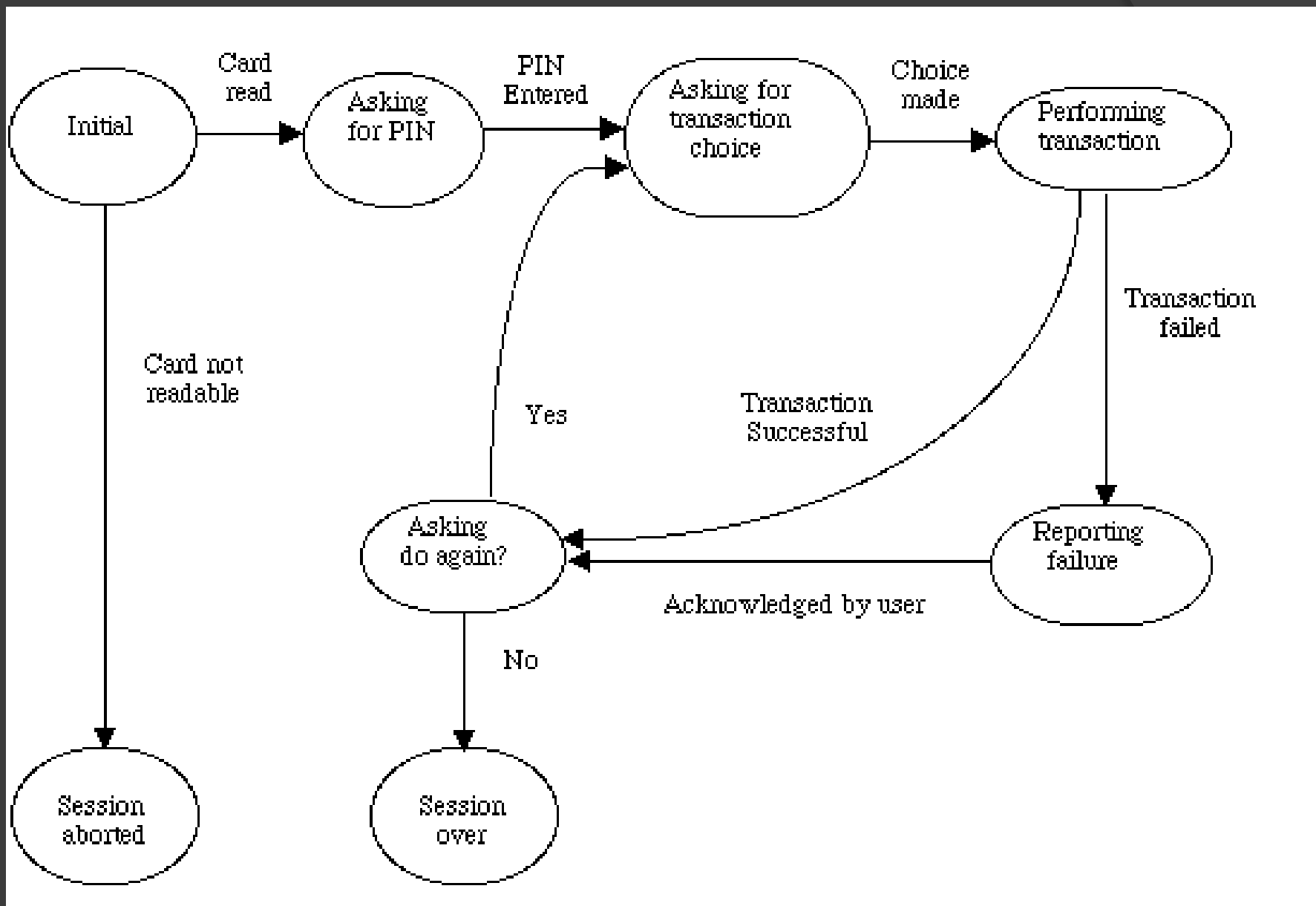
accounts

**4. State diagrami** paraqet gjendjet e ndryshme ne te cilat kalojne objektet e sistemit gjate ciklit te tyre jetesor.

Objektet ne sistem i nderrojne gjendjet si reagim ndaj ndodhive.

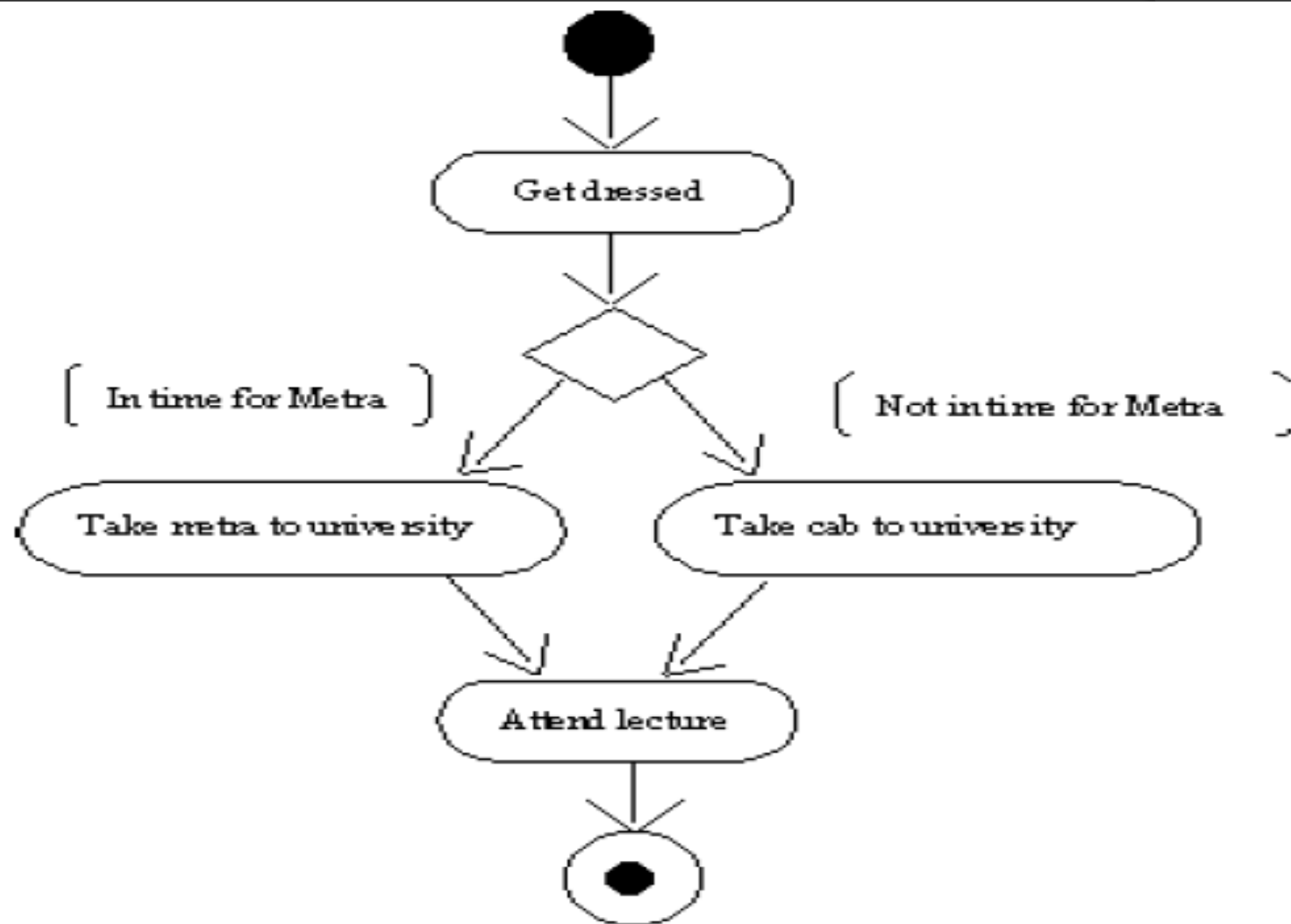
Gjithashtu state diagrami e paraqet tranzicionin e gjendjes se objektit nga gjendja fillestare deri ne gjendjen finale, si reagim ndaj ndodhive qe ndikojne ne sistem.

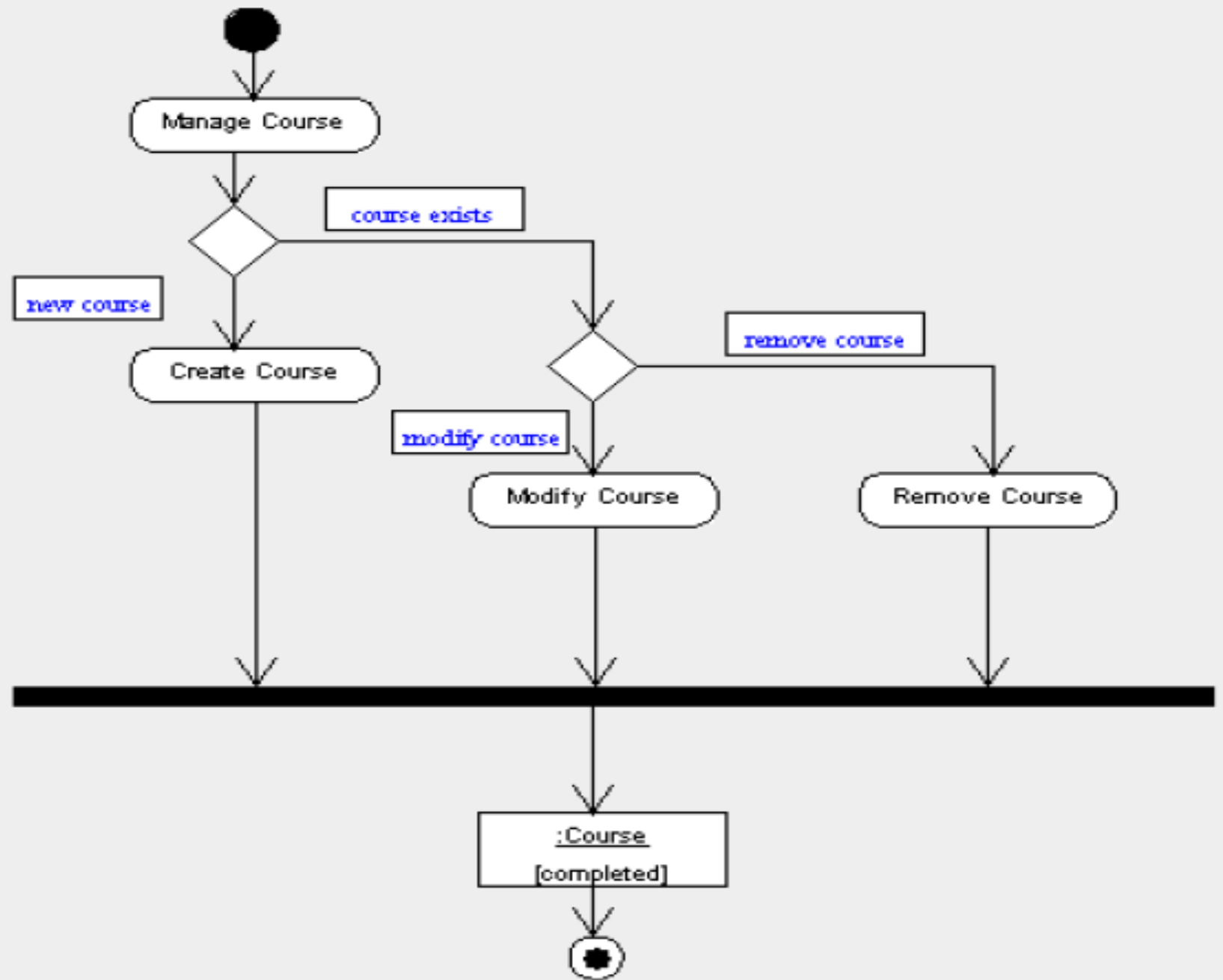




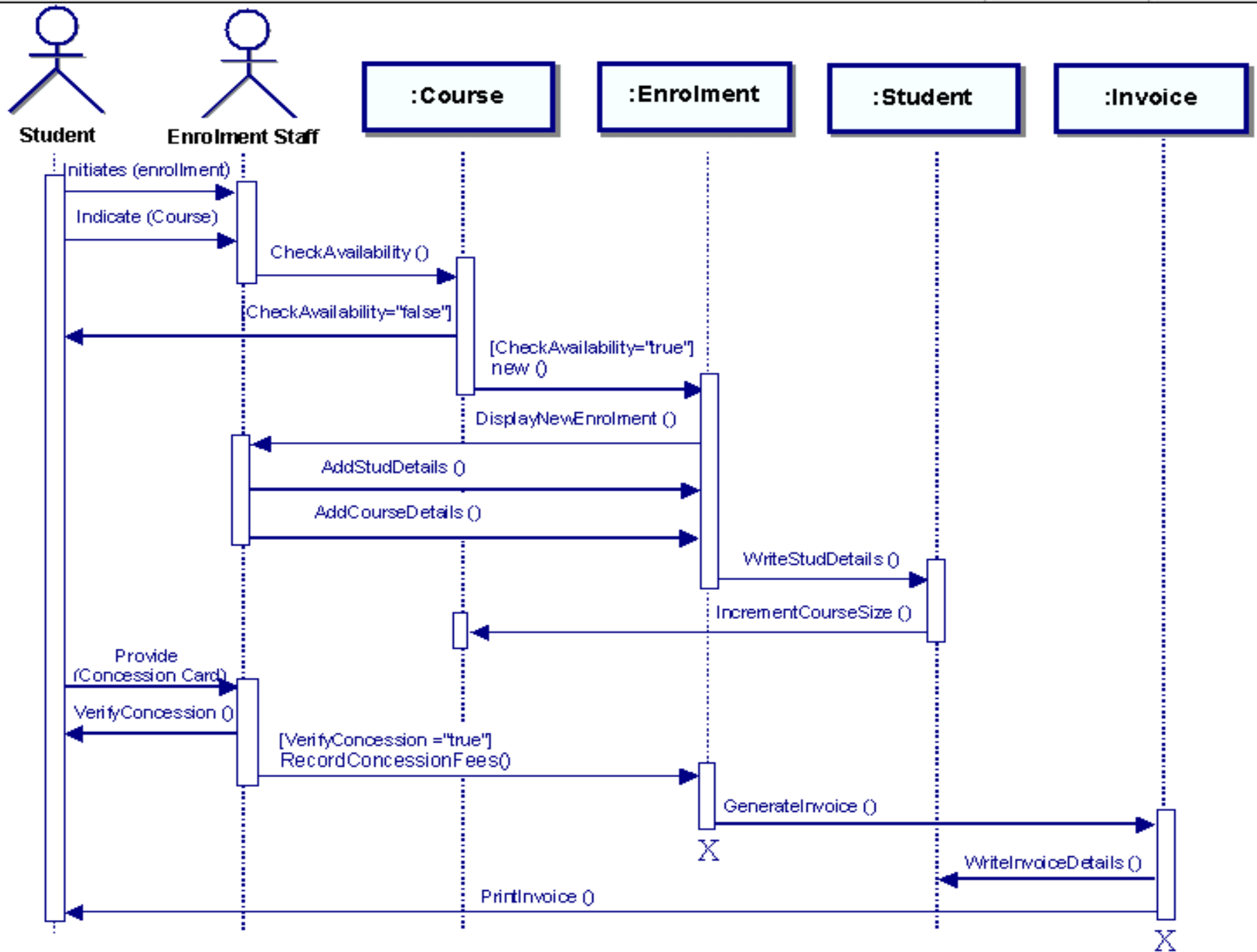
- ⑤ **5. Activity diagrami** - Rrjedhja e procesit ne sistem paraqitet me activity diagram. Activity diagram perbehet nga aktivitetet, akcionet, transicionet, gjendjet fillestare dhe finale.

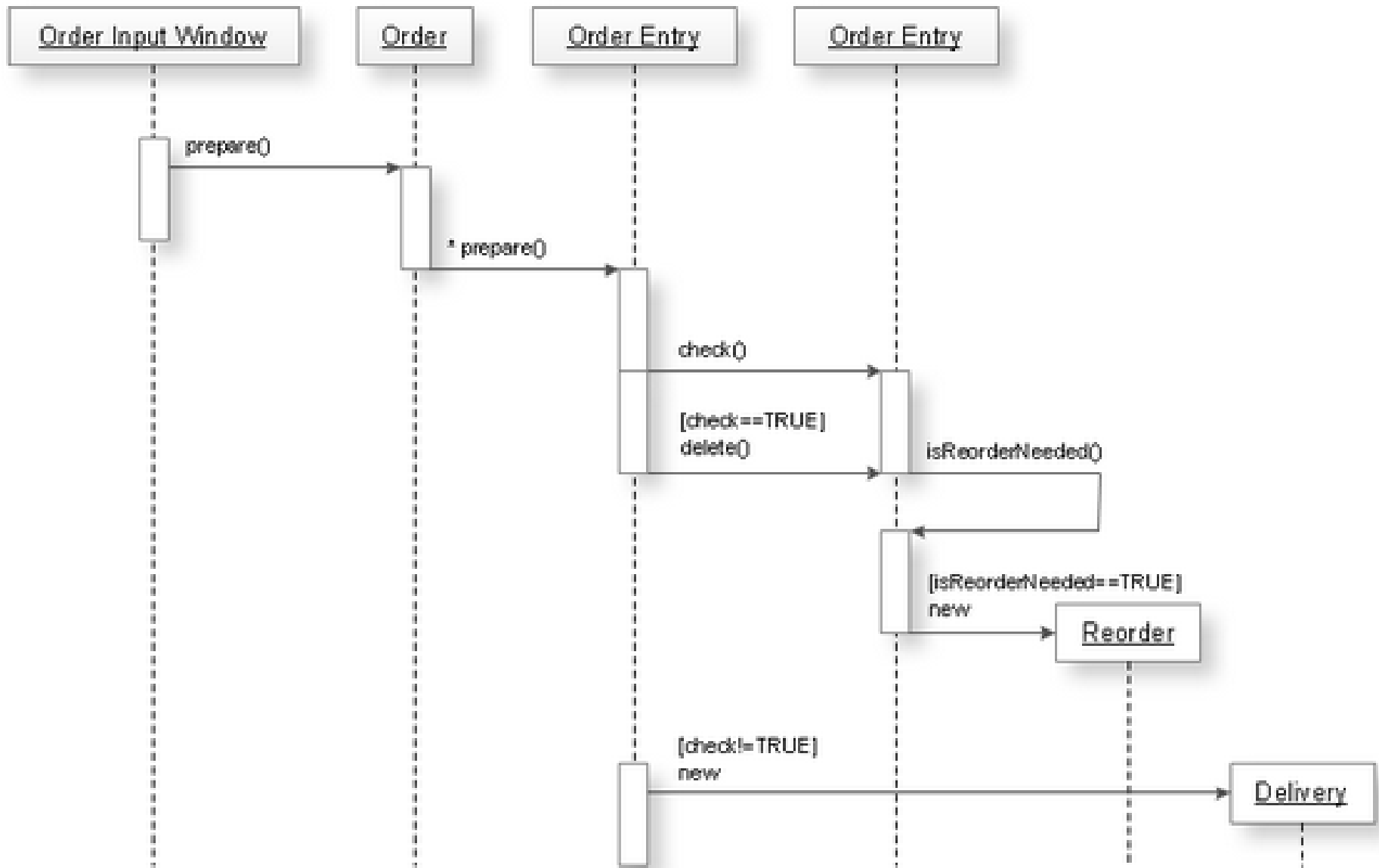




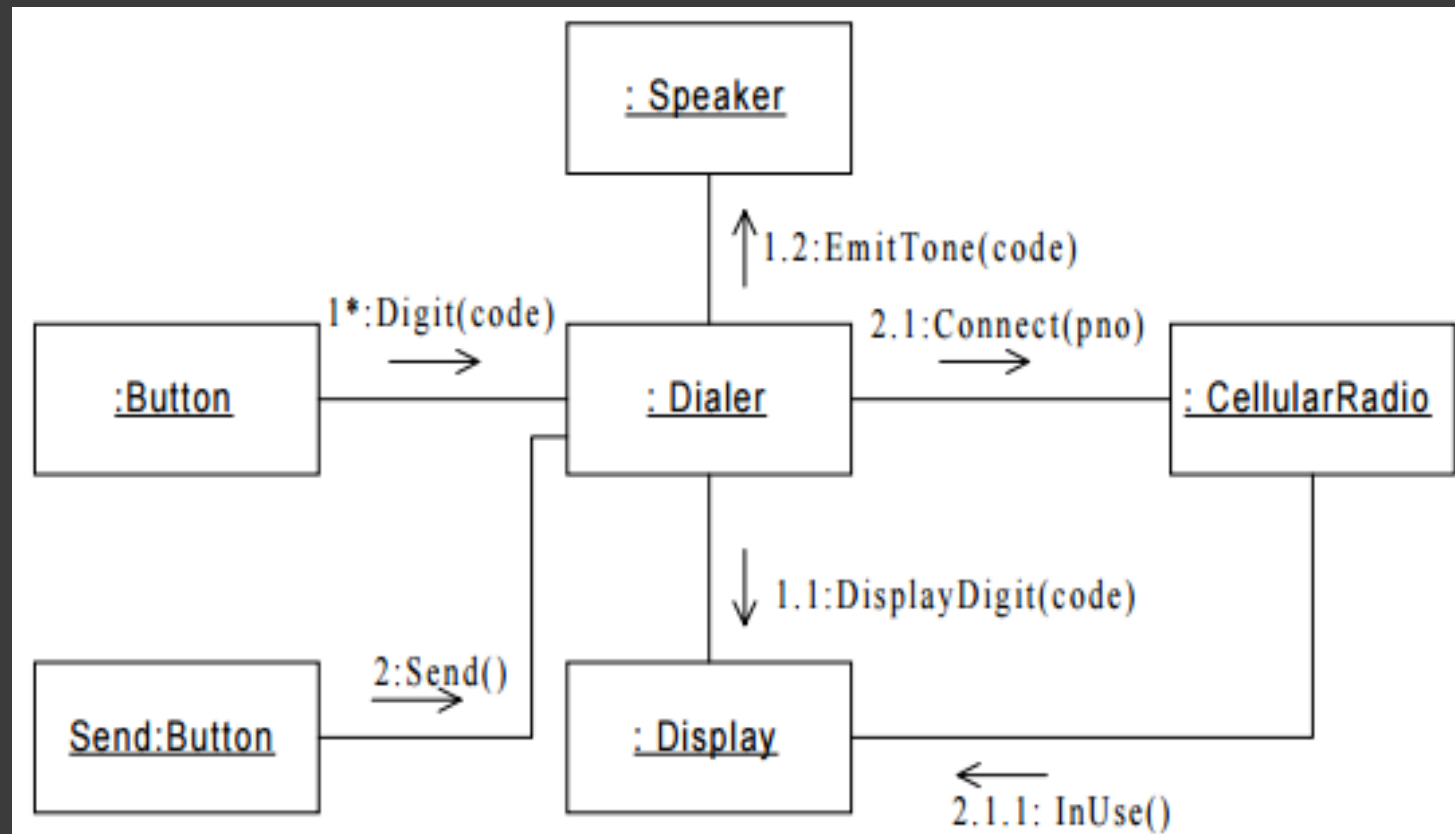


- ⑥ **6. Sequence diagrami** paraqet nderveprimin ndermjet objekteve te ndryshme ne sistem. Aspekt i rendesishem te ky diagram eshte se sekuenca ekzakte e nderveprimeve ndermjet objekteve paraqitet hap pas hapi. Objekte te ndryshme ne sequence diagram nderveprojne me njera tjetren duke derguar “mesazhe”.

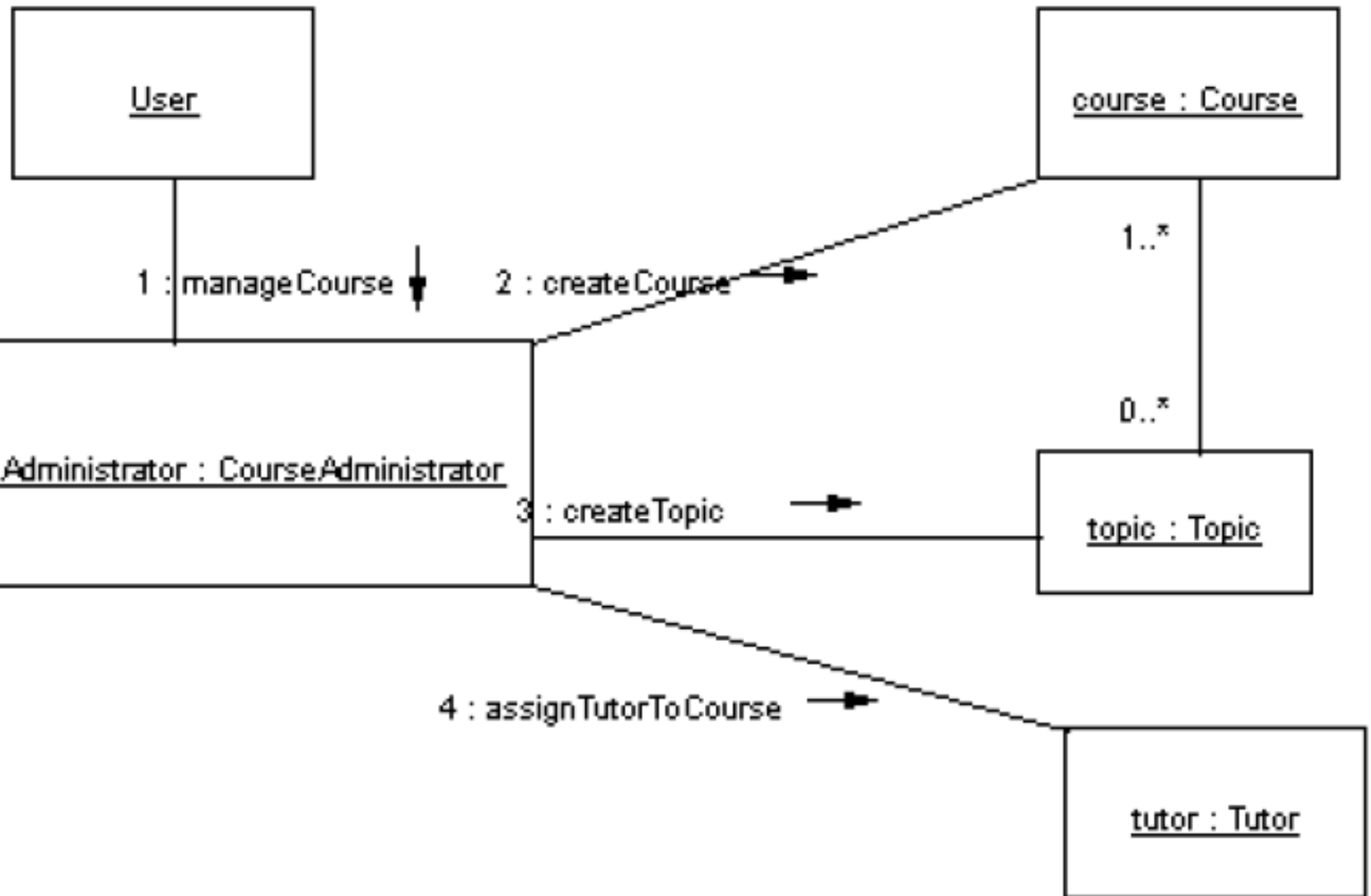




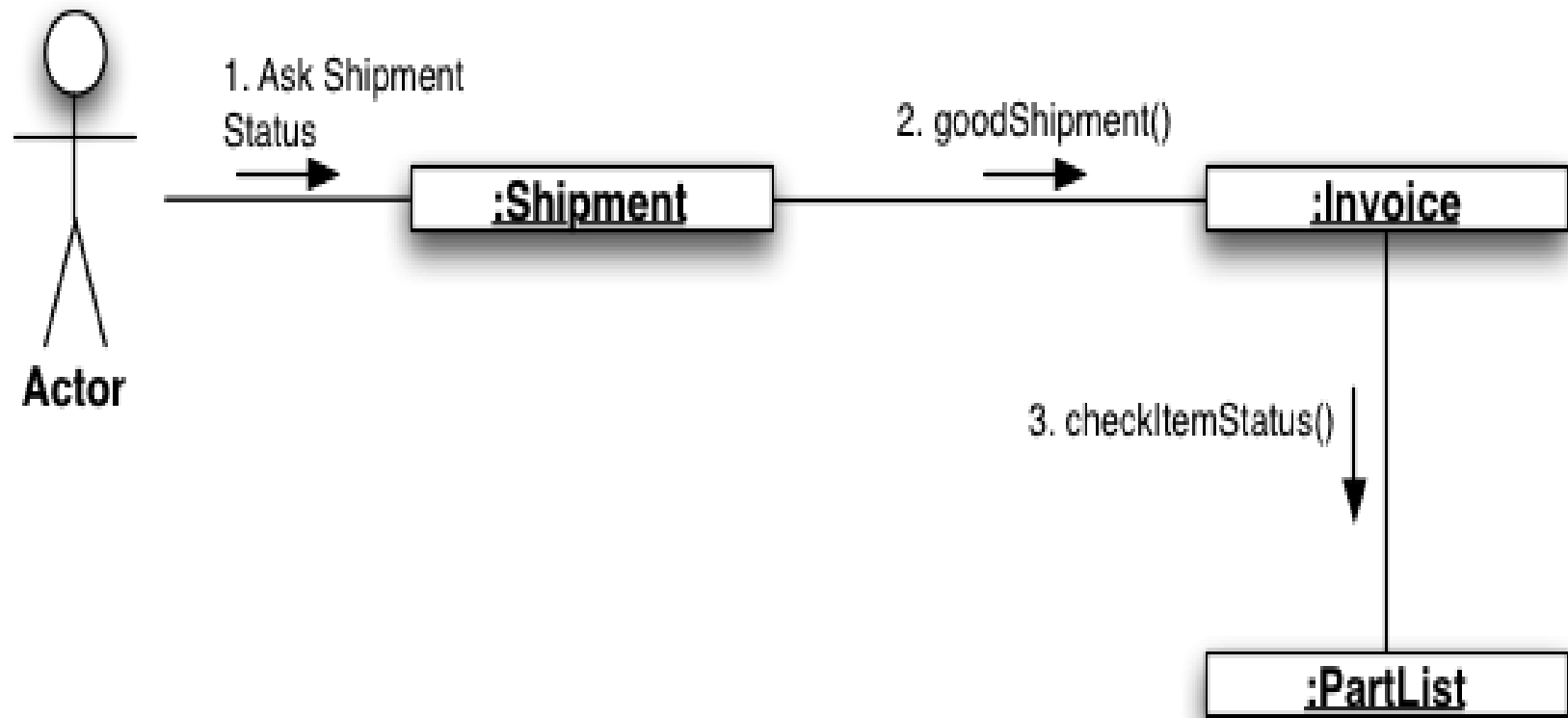
- ⑦ **7. Collaboration diagrami** i grupon nderveprimet ndermjet objekteve te ndryshme. Collaboration diagrami e mundeson identifikimin e te gjitha nderveprimeve te mundshme qe secili objekt ka me objektet tjera.



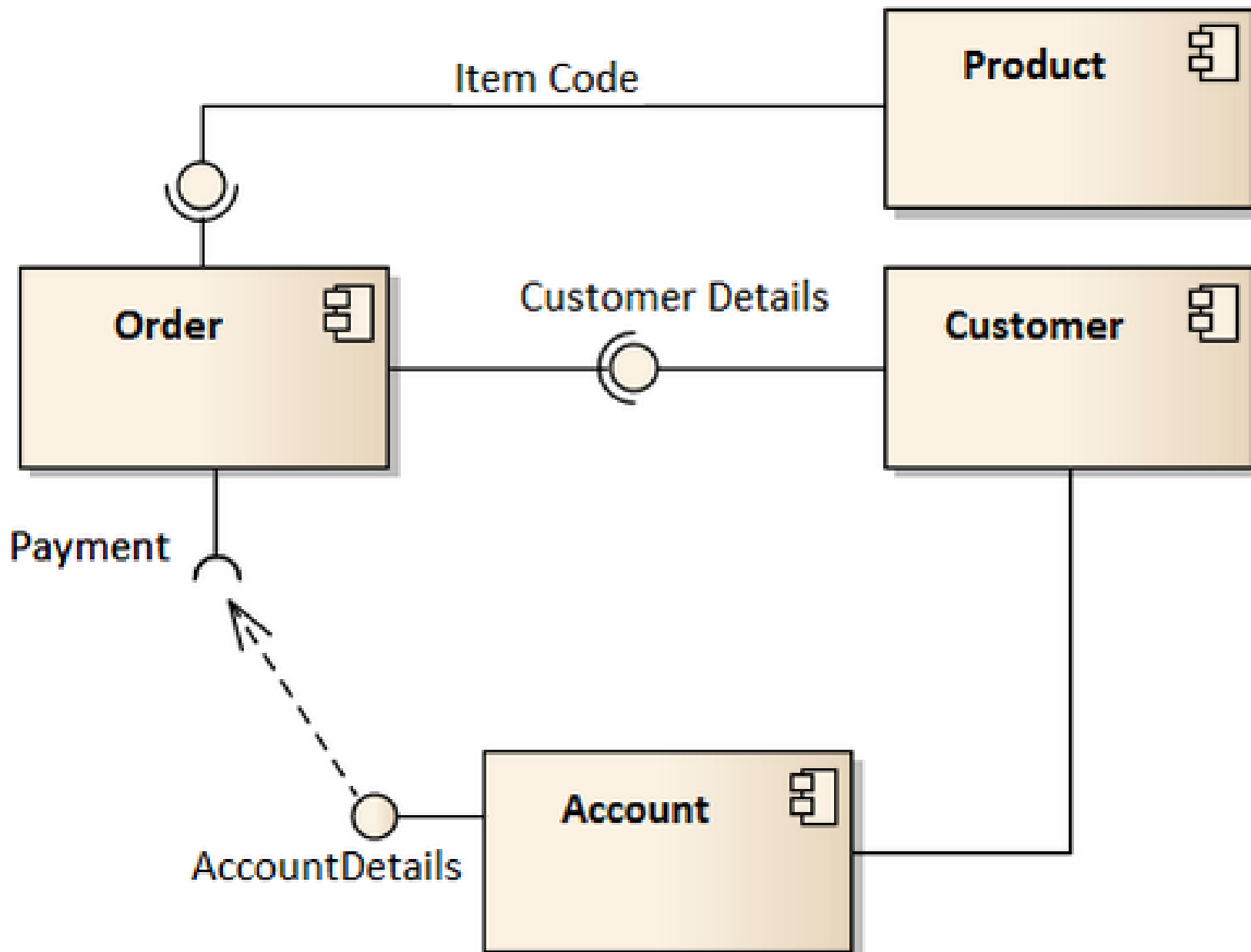
**Collaboration Diagram of “Make Phone Call” use case.**

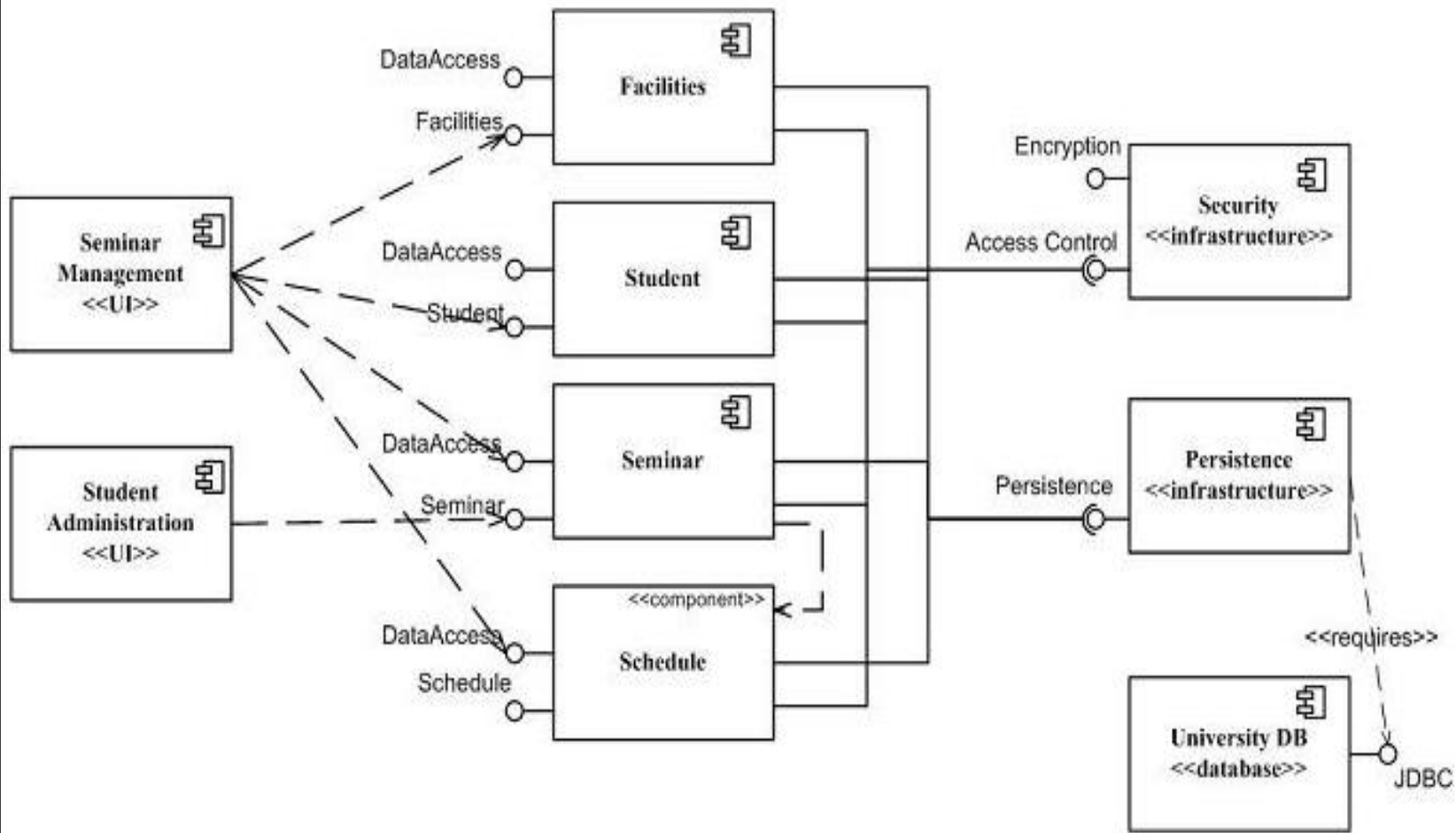




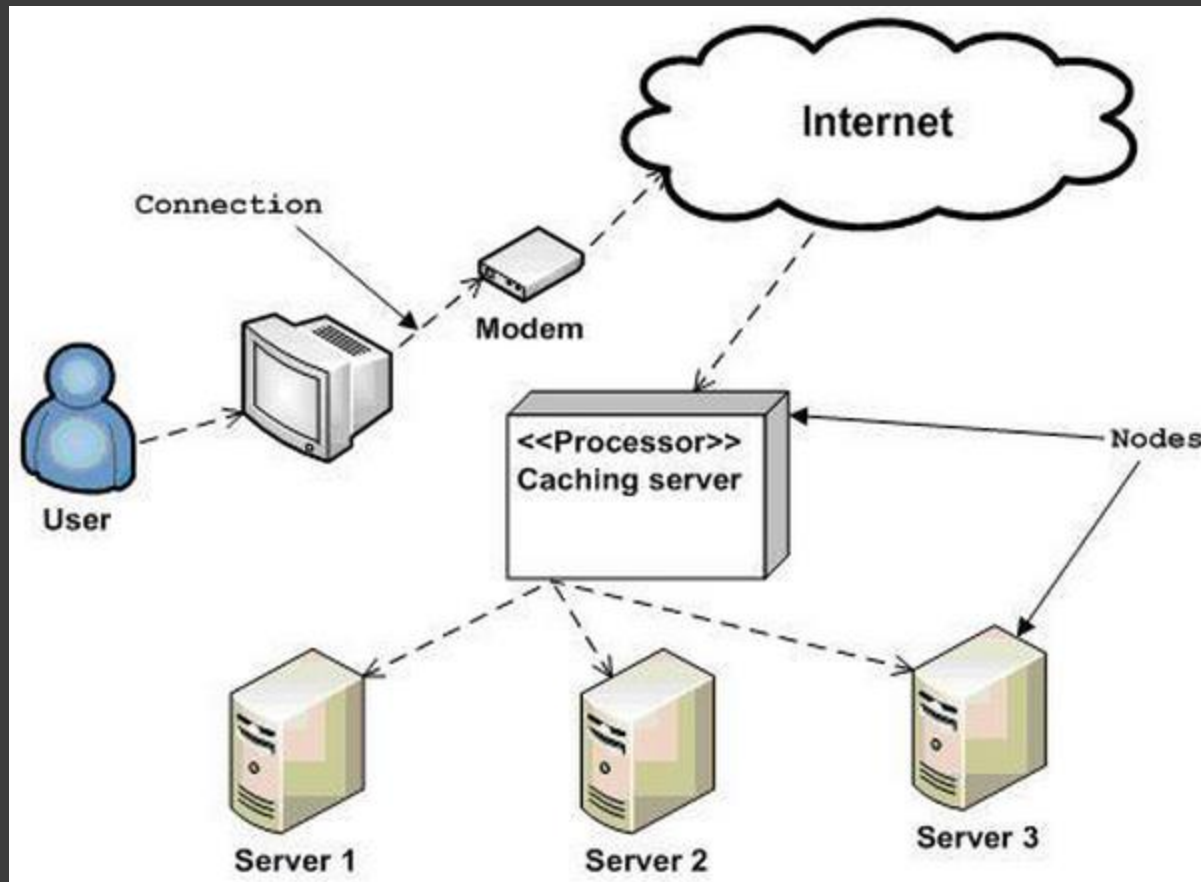


- ⑧ **8. Component diagrami** paraqet pjeset e niveli te larte qe e perbejne sistemin. Ky diagram tregon, ne nivel te larte, cilat komponente krijojne pjese te sistemit dhe si ato jane te nderlidhura.

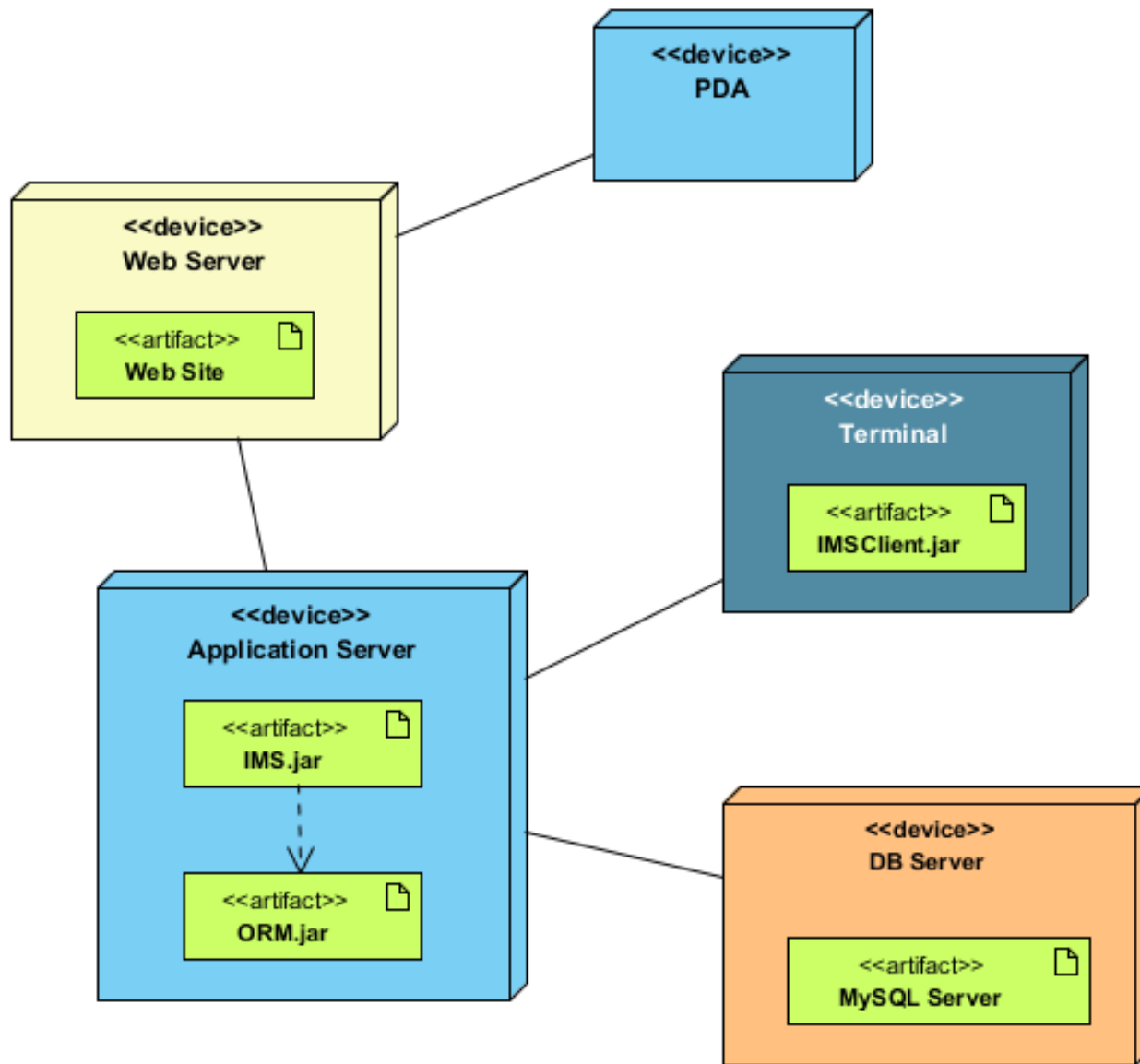




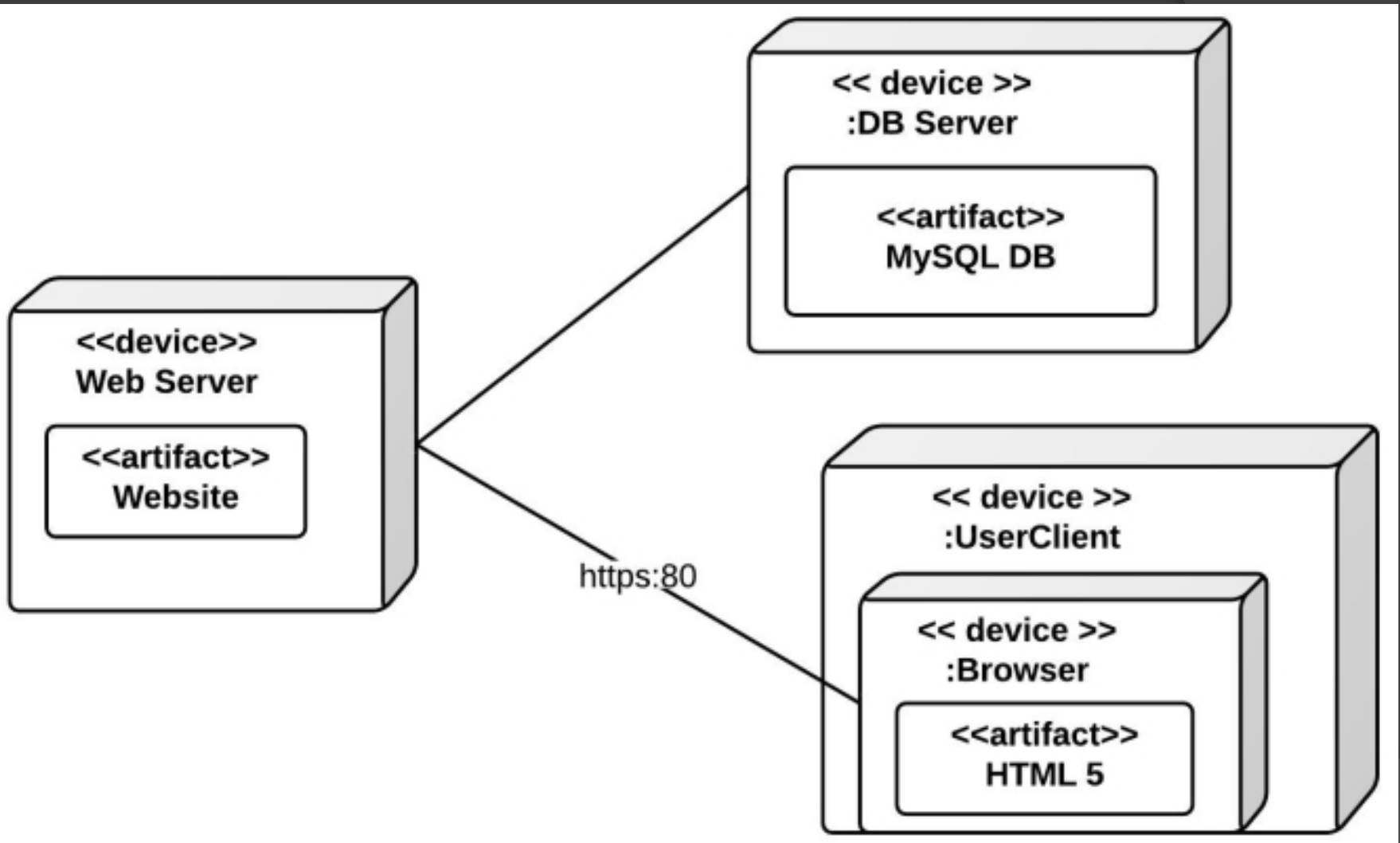
- 9. Deployment diagrami me se shumti perdoret kur sistemi te jete ndertuar dhe eshte i gatshem per perdorim.



Deployment diagram of an order management system



Deployment diagram



Deployment Diagram

# Mjetet e njohura te UML-se

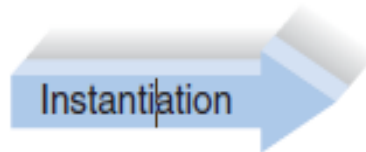
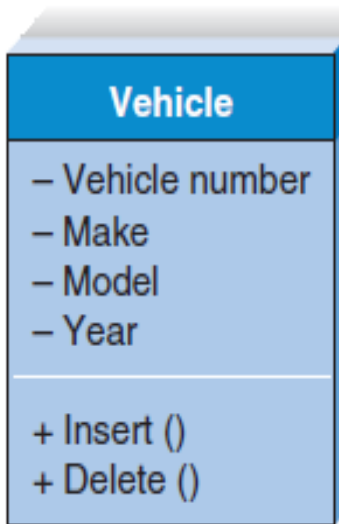
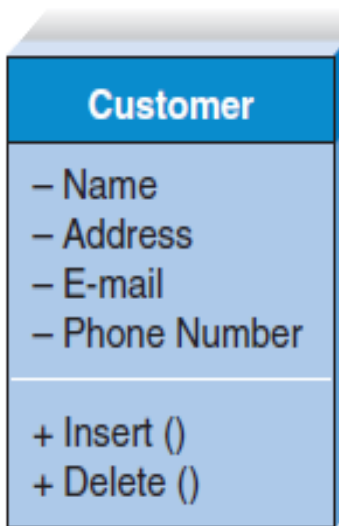
- ① **Microsoft Office Visio**
- ① **UModel**
- ① **Rational Rose**
- ① **Together**
- ① **Poseidon etj**



- **Klasat dhe objektet**
- Klasa është template e përgjithshme që e përdorim për t'i definuar dhe krijuar instancat specifike, apo objektet. Secili objekt është i asocuar me një klasë. Për shembull, të gjitha objektet që paraqesin informacione për pacientet mund t'i takojnë klasës së quajtur Pacienti, sepse kësaj klase i atribuon (p.sh. emri, adresa dhe data e lindjes) si dhe metodat (p.sh. mirëmbani informacionet dhe fshini hyrjet) të cilat i ndajnë të gjithë pacientet.

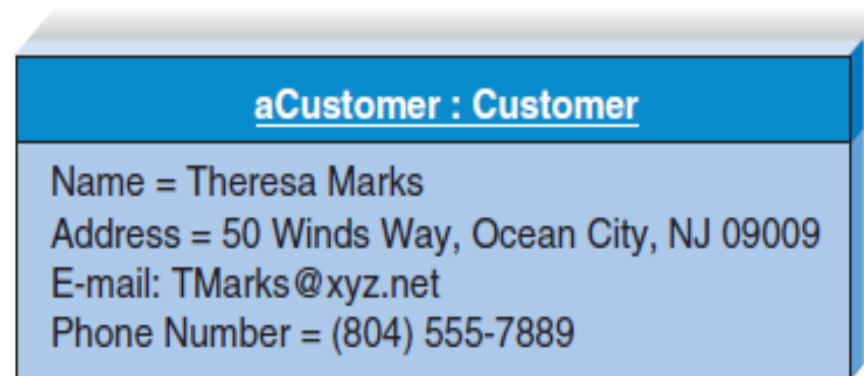
- Objekti është paraqitje e klases. Me fjale të tjera, objekti është person, vend, ngjarje apo diçka rreth së cilës dëshirojmë të marrim informacione. Nëse kemi qene duke e ndertuar një sistem të shitjes për vetura, atëherë klasat do të përfaqësojnë makinat, konsumatorët dhe ofertat. P.sh konsumatorët konsiderohen instance, apo objekte të klases së konsumatorëve. Secili objekt ka attribute që përshkruajnë informacion rreth objektit si emri, adresa, e-maili dhe numri i telefonit.

## Classes



## Objects

An instance of the Customer class



An instance of the Vehicle class

