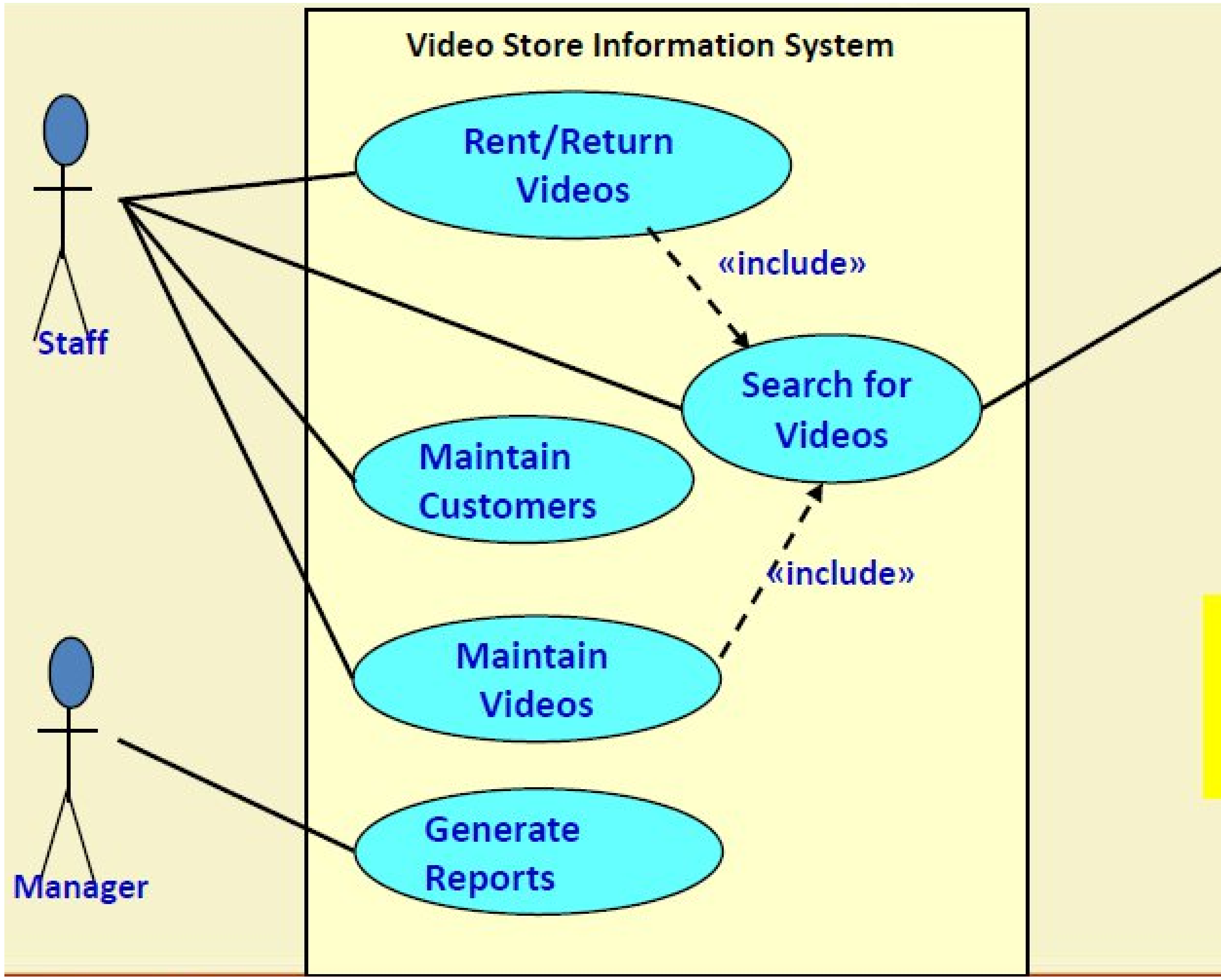
More on UML

Example 1: Video Store Information System Use Case Diagram

Video Store Information System supports the following business functions:

- Recording information about videos the store owns.
  - This database is searchable by staff and all customers
- Information about a customer's borrowed videos
  - Access by staff and customer. It involves video database searching.
- Staff can record video rentals and returns by customers.
  - It involves video database searching.
- Staff can maintain customer and video information.
- Managers of the store can generate various reports.



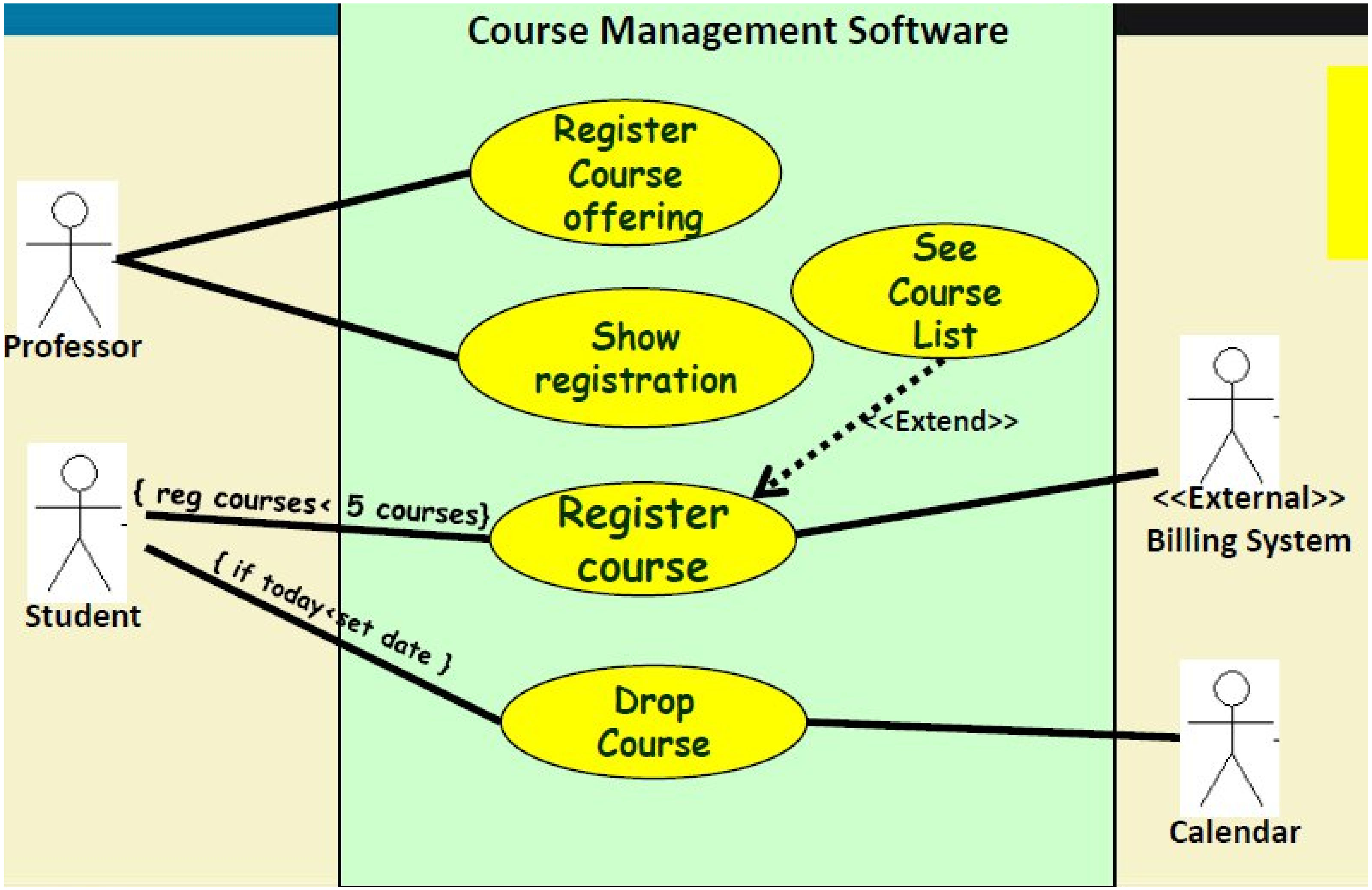
Example 1: Solution

Example 2: Course Management Software Use Case Diagram

- At the beginning of each semester,
  - Each professor shall register the courses that he is going to teach.

- A student can select up to four-course offerings.
  - **During registration a student can request a course catalogue showing course offerings for the semester.**
  - **Information about each course such as professor, department and prerequisites would be displayed.**
  - **The registration system sends information to the billing system, so that the students can be billed for the semester.**

- For each semester, there is a period of time during which dropping of courses is permitted.
- Professors must be able to access the system to see which students signed up for each of their course offerings.

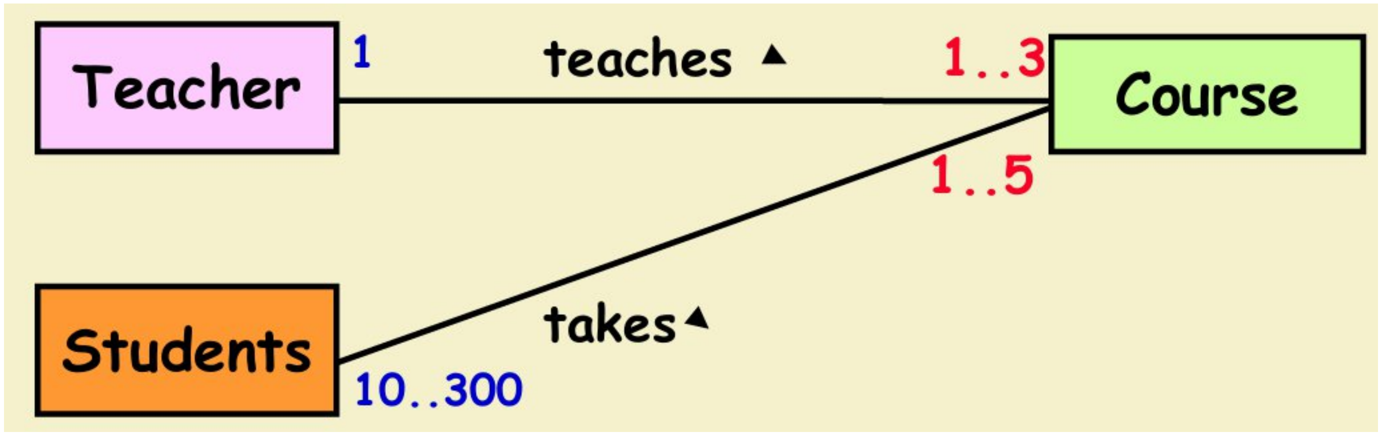

Course Management Software
Register Course offering
Show registration
See Course List
<<Extend>>
Register course
Drop Course
Professor
Student
{ reg courses< 5 courses}
{ if today<set date }
<<External>> Billing System
Calendar

Example 3: Class Diagram
Association – Multiplicity Example

- A teacher teaches 1 to 3 courses (subjects)
- Each course is taught by only one teacher.
- A student can take between 1 to 5 courses.
- A course can have 10 to 300 students.

Example 3: Class Diagram
Association – Multiplicity Example

- A teacher teaches 1 to 3 courses (subjects)
- Each course is taught by only one teacher.
- A student can take between 1 to 5 courses.
- A course can have 10 to 300 students.

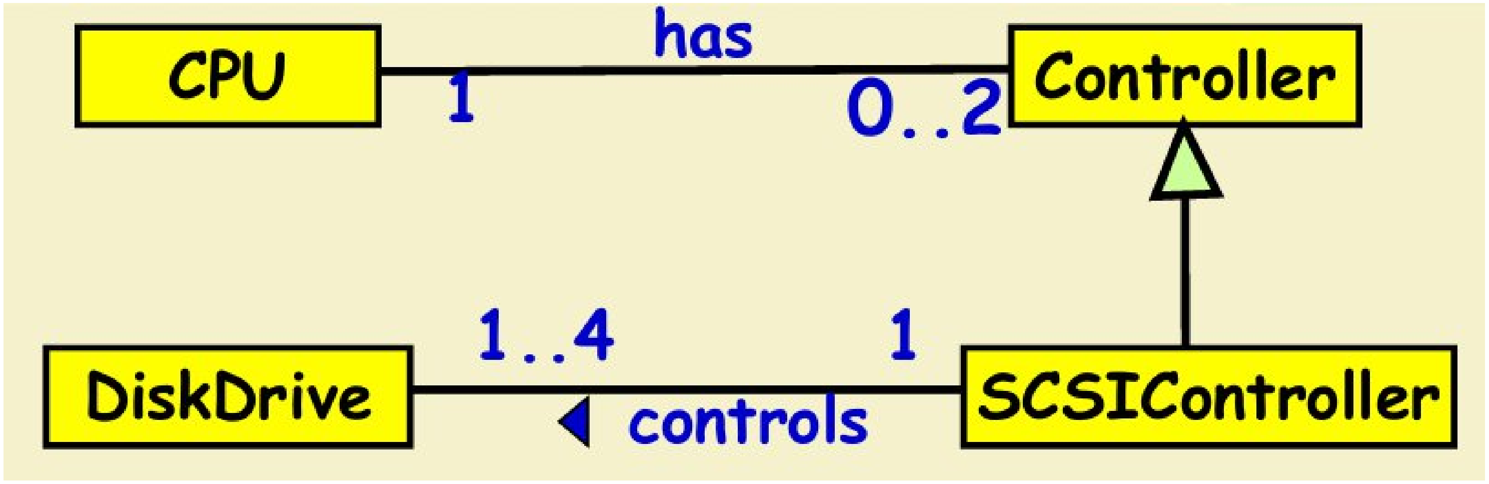


Example 4: Class Diagram
Association – Multiplicity Example

- 1 CPU has 0 to two Controllers
- 1-4 DiskDrives controlled by 1 SCSIController
- SCSIController is a (specialized) Controller

Example 4: Class Diagram
Association – Multiplicity Example

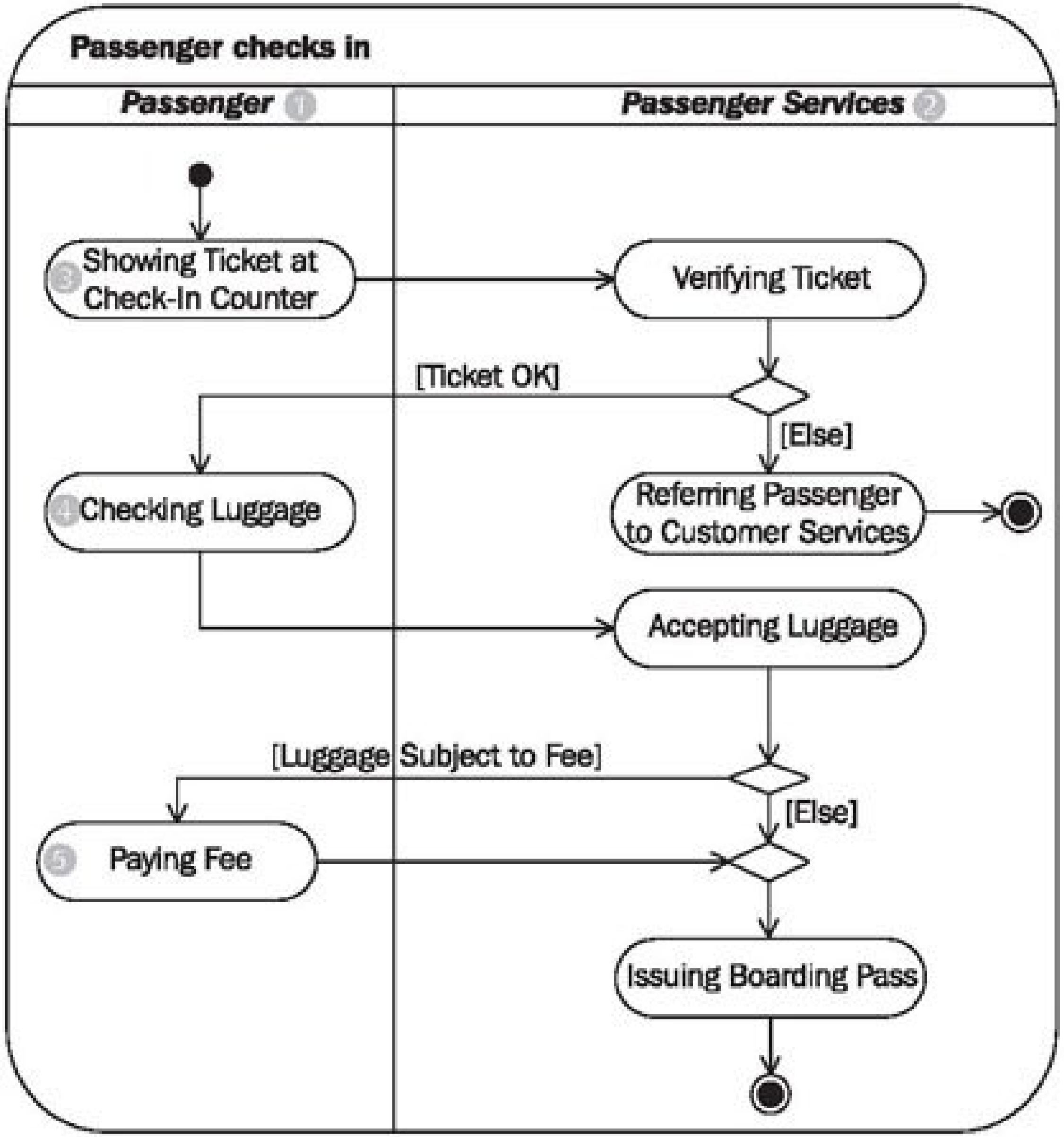
- 1 CPU has 0 to two Controllers
- 1-4 DiskDrives controlled by 1 SCSIController
- SCSIController is a (specialized) Controller



Example 5-Draw the Activity Diagram for Check in Process at Airport

- The activity diagram is divided into two partitions: passenger and passenger services.
- Upon reaching the Airport, the passenger will be required to show your ID and ticket to the security officers at the entry gate. The Ticket and ID Documents are verified by the Official,
- After verification, the Passenger goes ahead with checking in process and luggage drop off. Once you enter inside airport, you can see multiple boards with airline names and flight numbers. Go to your airline counter & officials at the counter will check the required documents to verify your identity. The airline staff will weigh your check-in luggage and provide you with a boarding pass
- Travelers are allowed to check up to one bag, but there is a weight restriction on the bag. The Bag should be less than 15 kgs. If luggage is > 15 kg then fees needs to be paid. You will be issued a boarding pass which will have all the details of your travel like Flight number, seat number, scheduled boarding time of the flight.

Example 5-Draw the Activity Diagram for Check in Process at Airport



More on Sequence Diagrams

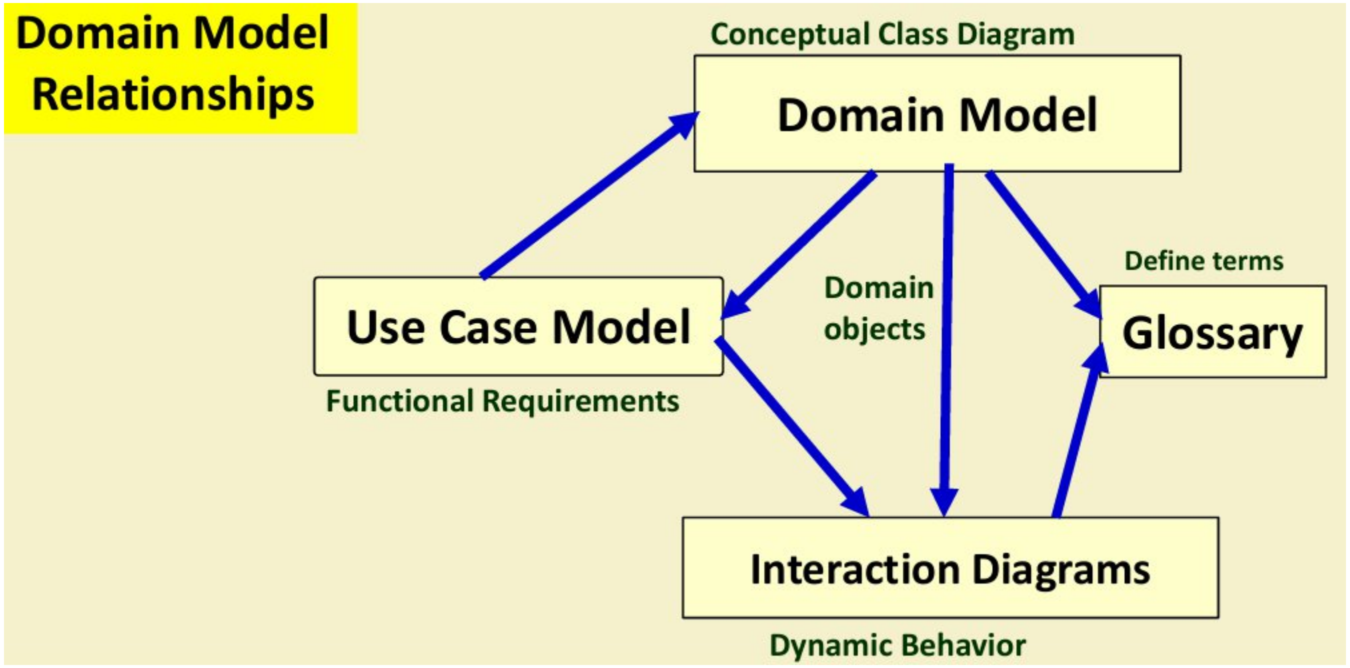
Sequence Diagrams

- How many interaction diagrams to draw?
  - Typically each interaction diagram realizes behaviour of a single use case

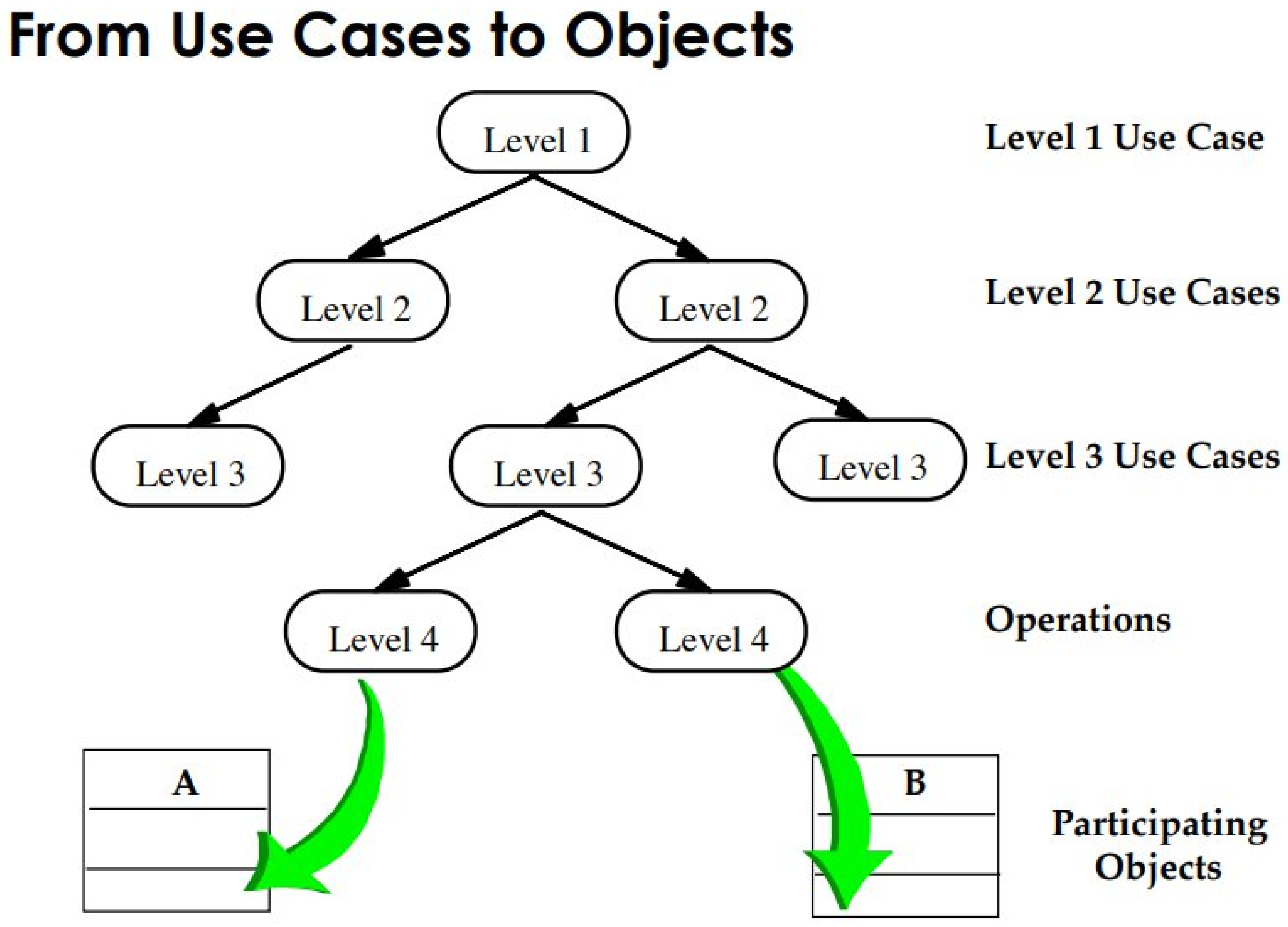
Draw one sequence diagram for each use case.

Domain Model Relationships

- Represent concepts or objects appearing in the problem domain.
- Also capture object relationships



From Use Cases to Objects



From Use Cases to Objects

Main goal: Find the important abstractions

- Steps during object modeling

1. Class identification

- Based on the fundamental assumption that we can find abstractions

2. Find the attributes
3. Find the operations
4. Find the associations between classes

- Order of steps
  - Goal: get the desired abstractions
  - Order of steps is secondary
- What happens if we find the wrong abstractions?
  - We iterate and revise the model.

From Use Cases to Objects

Starting from use cases and scenarios, analysis activities performed to obtain the analysis model are:

- Identifying entity objects
- Identifying boundary objects
- Identifying control objects
- Mapping use cases to objects
- Identifying associations among objects
- Identifying object attributes
- Modeling behavior with statecharts
- Modeling generalization relationships
- Reviewing the analysis model

Entity–control–boundary

- Entity–control–boundary (ECB), or
- Entity–boundary–control (EBC), or
- Boundary–control–entity (BCE)
- **is an architectural pattern used in use-case–driven object-oriented programming that structures the classes composing high-level object-oriented source code according to their responsibilities in the use-case realization.**

Categorizing Classes

- Three types of classes are to be identified:
  - Boundary class (Actor-use case pair)
  - Controller class (One per use case)
  - Entity class (Noun analysis)

Boundary Objects

- Handle interaction with actors: **User interface objects**

- Often implemented as screens, menus, forms, dialogs etc.
- Do not perform processing: But may validate input, format output, etc.

Identification of Boundary Objects

- Need one boundary object :For every actor-use case pair

Controller Objects

- **Overall responsibility to realize use case behavior:**
  - Interface with the boundary objects
  - Coordinate the activities of a set of entity objects

- **Embody most of the business logic required for use case execution:**
  - There can be more than one controller to realize a single use case

Controller Classes

- Controller classes coordinate, sequence, transact, **and otherwise control other objects…**
- In Smalltalk MVC mechanism:–These are called controller

Identification of Controller Objects

- Examine the use case diagram:
  - **Add one controller class for each use case.**
- Some controllers may need to be split into two or more controller classes if they get assigned too much responsibility.

Entity Objects

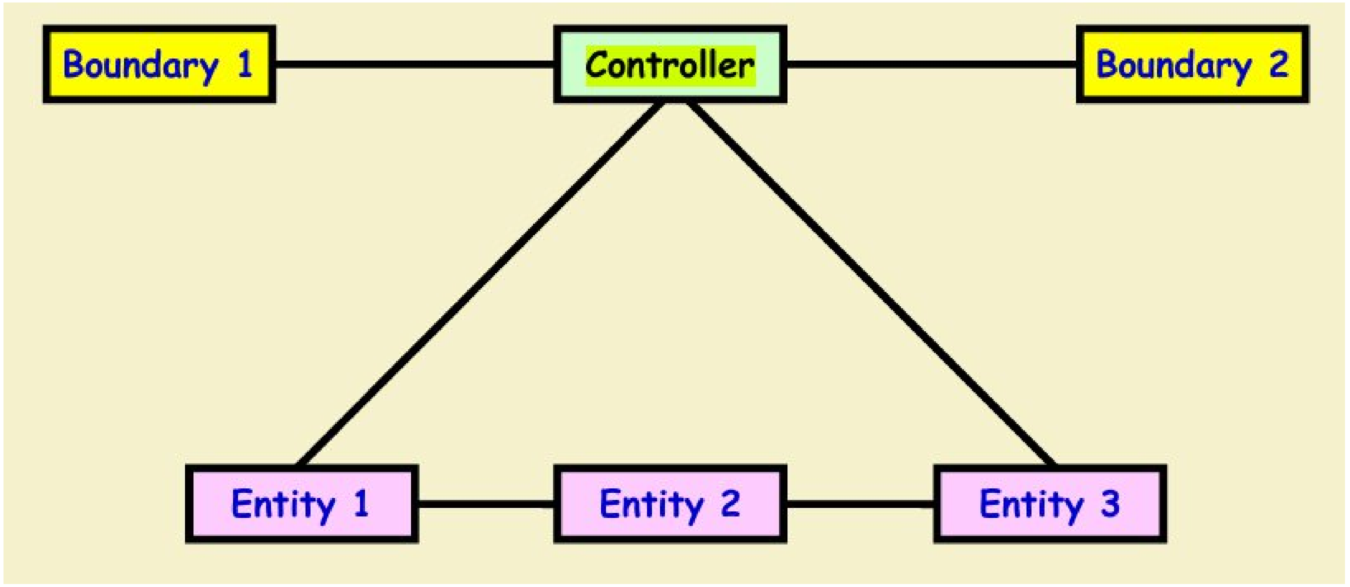
- **Hold information over long term:**
  - e.g. Book, BookRegister
- Normally are dumb servers:
  - **Responsible for storing data, fetching data etc.**
  - **Elementary operations on data such as searching, sorting, etc.**
- Often appear as **nouns** in the problem description…

Identification of Entity Objects by Noun Analysis

- Entity objects usually appear as nouns in the problem description.
- From the list of nouns, need to exclude:
  - **Users (e.g. accountant, librarian, etc)**
  - **Passive verbs (e.g. Acknowledgment)**
  - **Those with which you can not associate any data to store**
  - **Those with which you can not associate any methods**

Use Case Realization

- Realization of use case through the collaboration of Boundary, controller and entity object



ECB

- Entity represents long-lived information relevant for the stakeholders (i.e. mostly derived from domain objects, usually persistent);
- Boundary encapsulates interaction with external actors (users or external systems);
- Control ensures the processing required for the execution of a use case and its business logic, and coordinates, sequences controls other objects involved in the use case.

Example 1: Tic-Tac-Toe Computer Game

- A human player and the computer make alternate moves on a 3X3 square.
- A move consists of marking a previously unmarked square.
- The user inputs a number between 1 and 9 to mark a square
- Whoever is first to place three consecutive marks along a straight line (i.e., along a row, column, or diagonal) on the square wins.

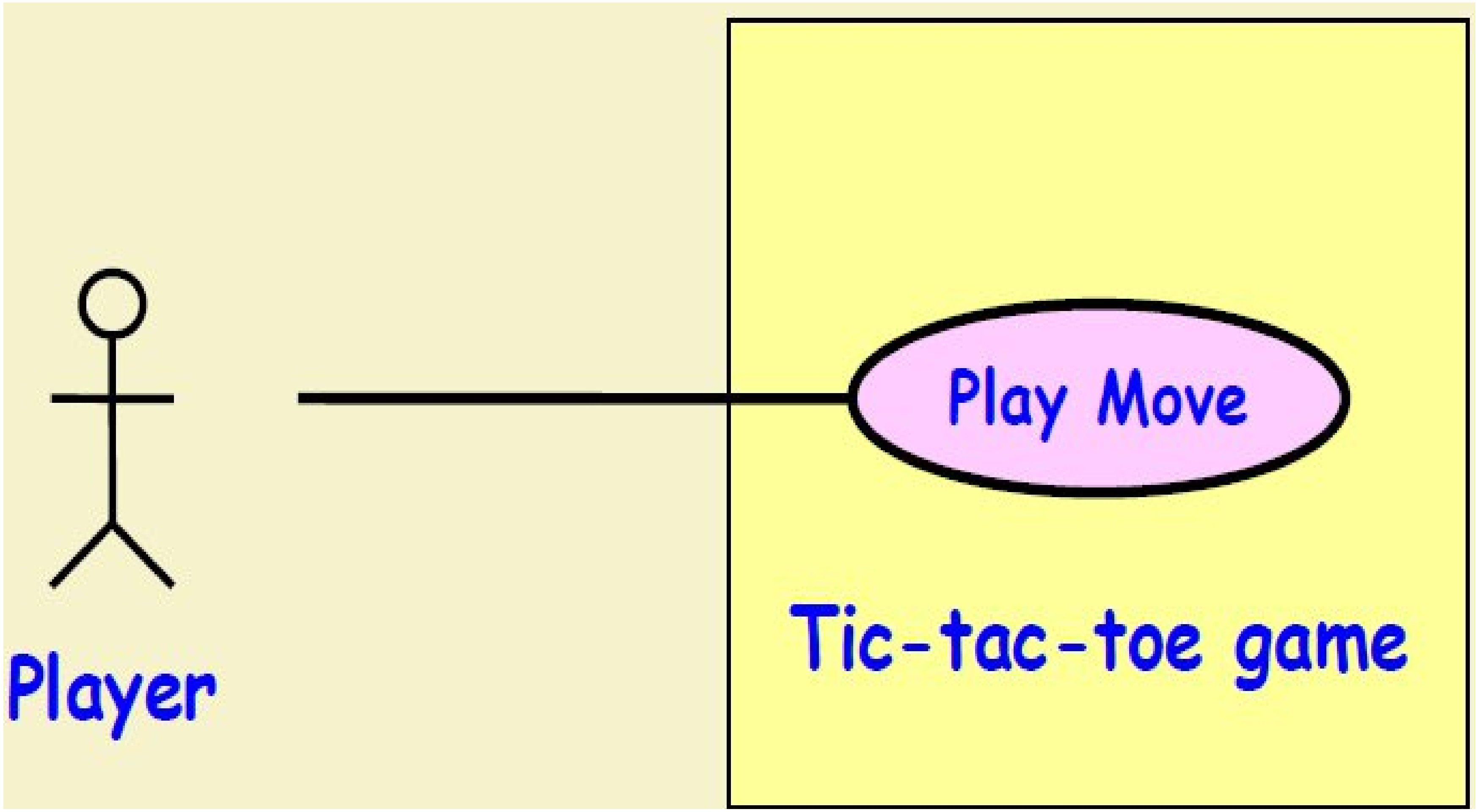
Example 1: Tic-Tac-Toe Computer Game cont…

- As soon as either of the human player or the computer wins, A message announcing the winner should be displayed.

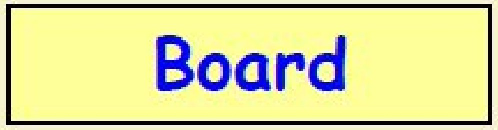
- If neither player manages to get three consecutive marks along a straight line, And all the squares on the board are filled up,
- Then the game is drawn.

- The computer always tries to win a game.

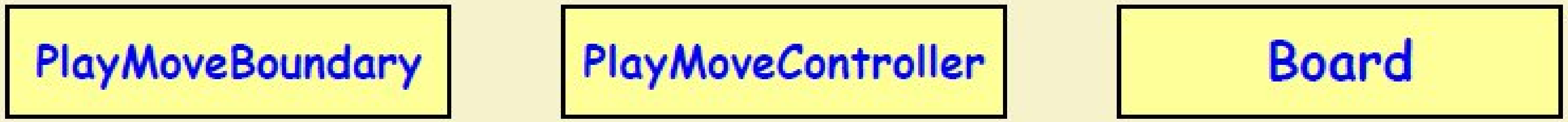
Example 1: Tic-Tac-TieUse Case Model



Example 1: Initial and Refined Domain Model



Initial domain model



Refined domain model

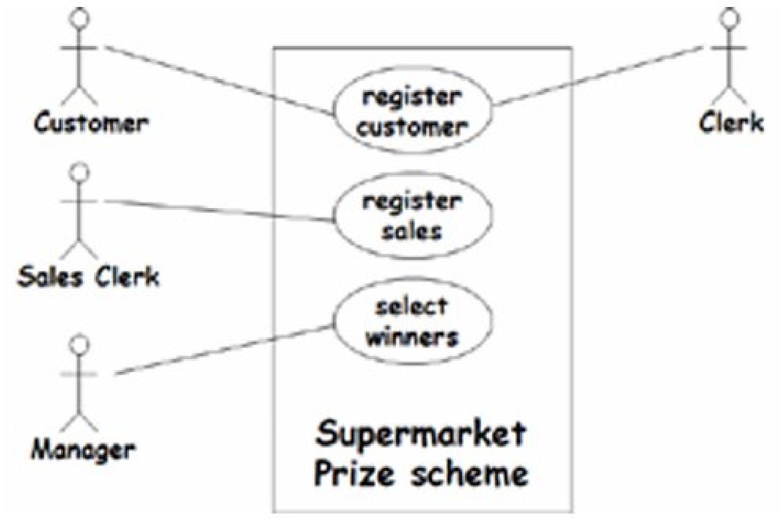
What is the number of boundary classes required for a software development project whose use case diagram is shown below?

a. 3

b. 4

c. 6

d. 8



What is the number of boundary classes required for a software development project whose use case diagram is shown below?

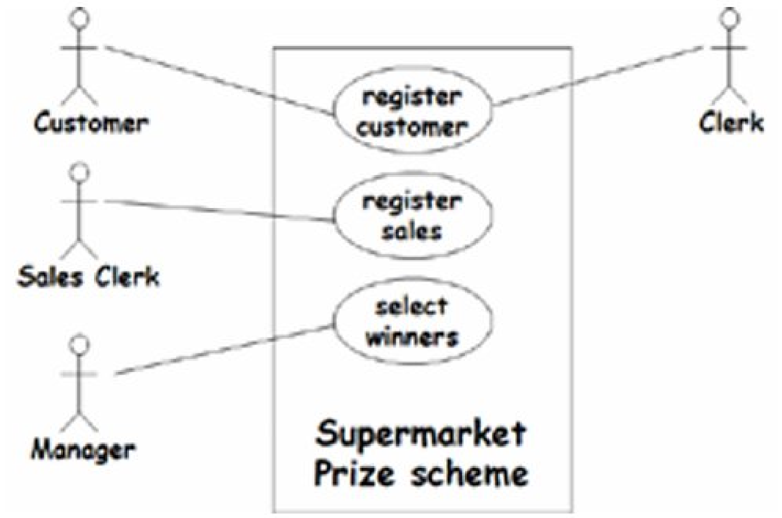
a. 3

b. 4

c. 6

d. 8

Ans b)



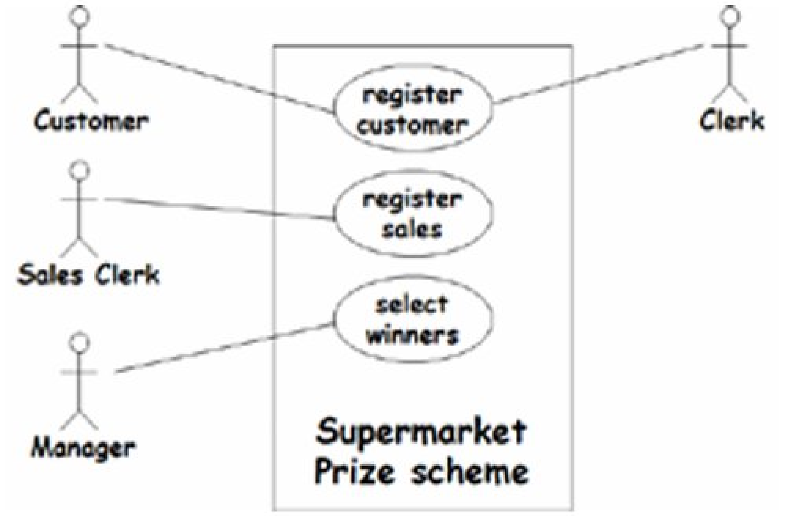
What is the number of controller classes required for a software development project whose use case diagram is shown below?

a. 3

b. 4

c. 6

d. 8



What is the number of controller classes required for a software development project whose use case diagram is shown below?

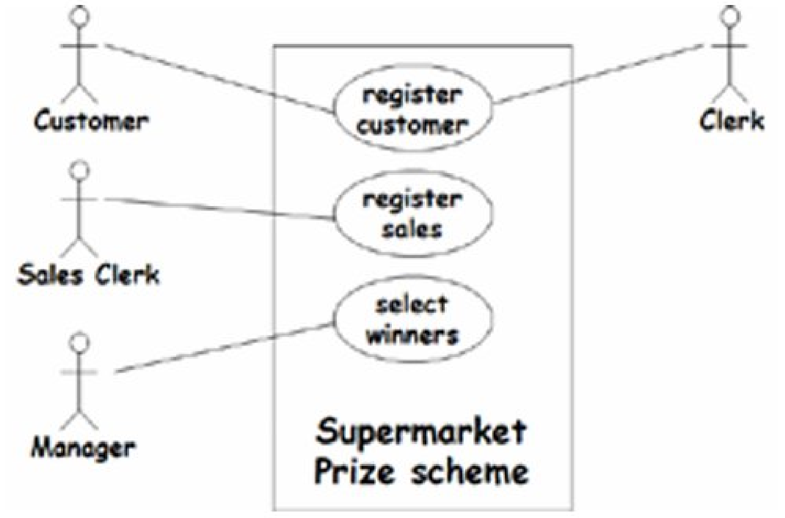
a. 3

b. 4

c. 6

d. 8

Ans 3)



Which one of the following does not serve as a guideline in identifying entity classes from a problem description?

a. Entity classes usually appear as data stores in a DFD model
b. Entity classes usually occur as group of objects that are aggregated
c. The aggregator of the objects of an entity class corresponds to a register in the physical world
d. Users are considered as entity classes

Which one of the following does not serve as a guideline in identifying entity classes from a problem description?

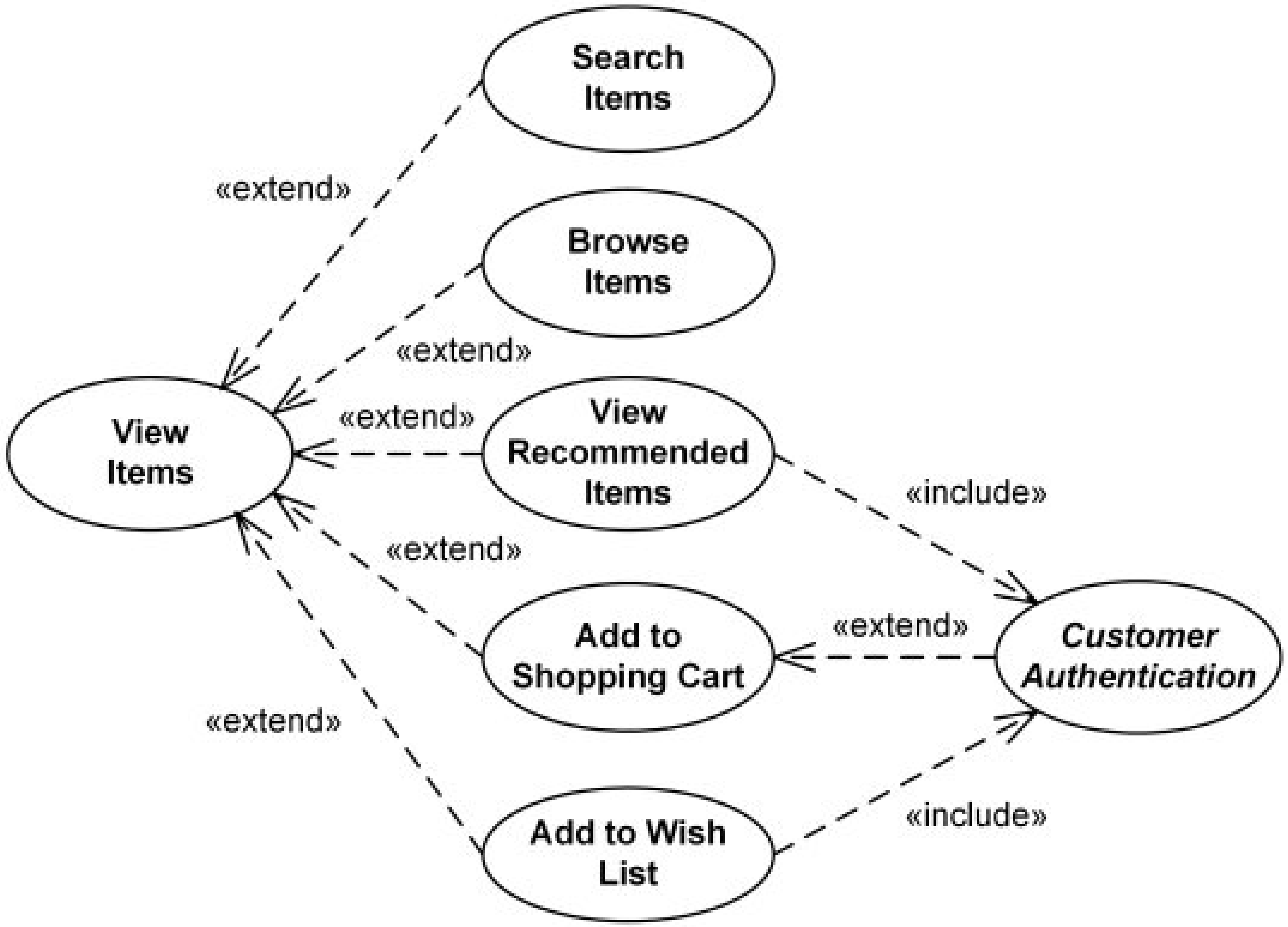
a. Entity classes usually appear as data stores in a DFD model
b. Entity classes usually occur as group of objects that are aggregated
c. The aggregator of the objects of an entity class corresponds to a register in the physical world
d. Users are considered as entity classes

Ans d)

Online Shopping Use Case Diagram Example

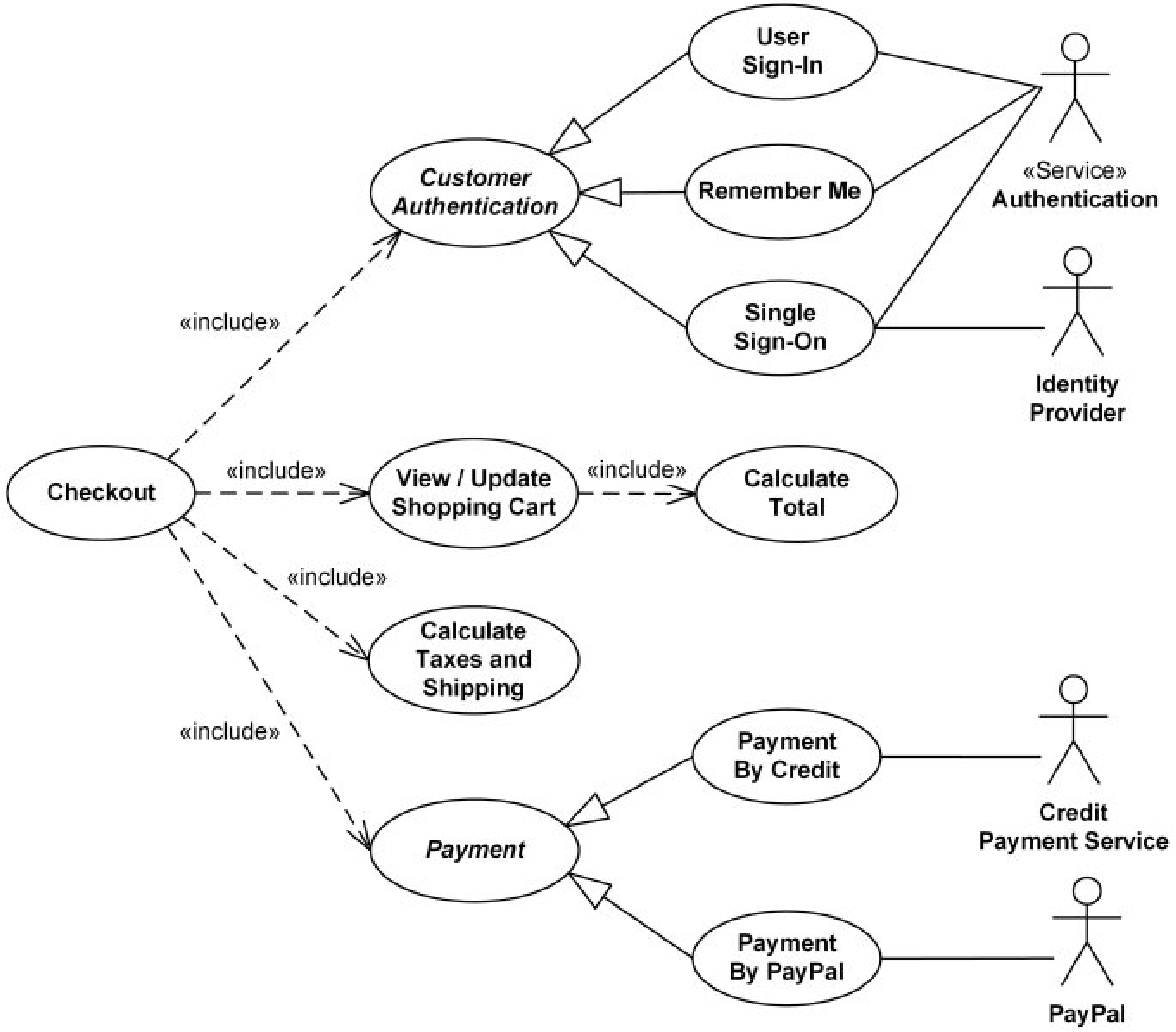
- View Items use case is extended by several optional use cases - customer may search for items, browse catalog, view items recommended for him/her, add items to shopping cart or wish list. All these use cases are extending use cases because they provide some optional functions allowing customer to find item.

- Customer Authentication use case is included in View Recommended Items and Add to Wish List because both require the customer to be authenticated. At the same time, item could be added to the shopping cart without user authentication.


Search Items
Browse Items
View Items
View Recommended Items
Add to Shopping Cart
Add to Wish List
Customer Authentication
«extend»
«extend»
«extend»
«extend»
«extend»
«extend»
«include»
«include»

Online Shopping
Use Case Diagram Example

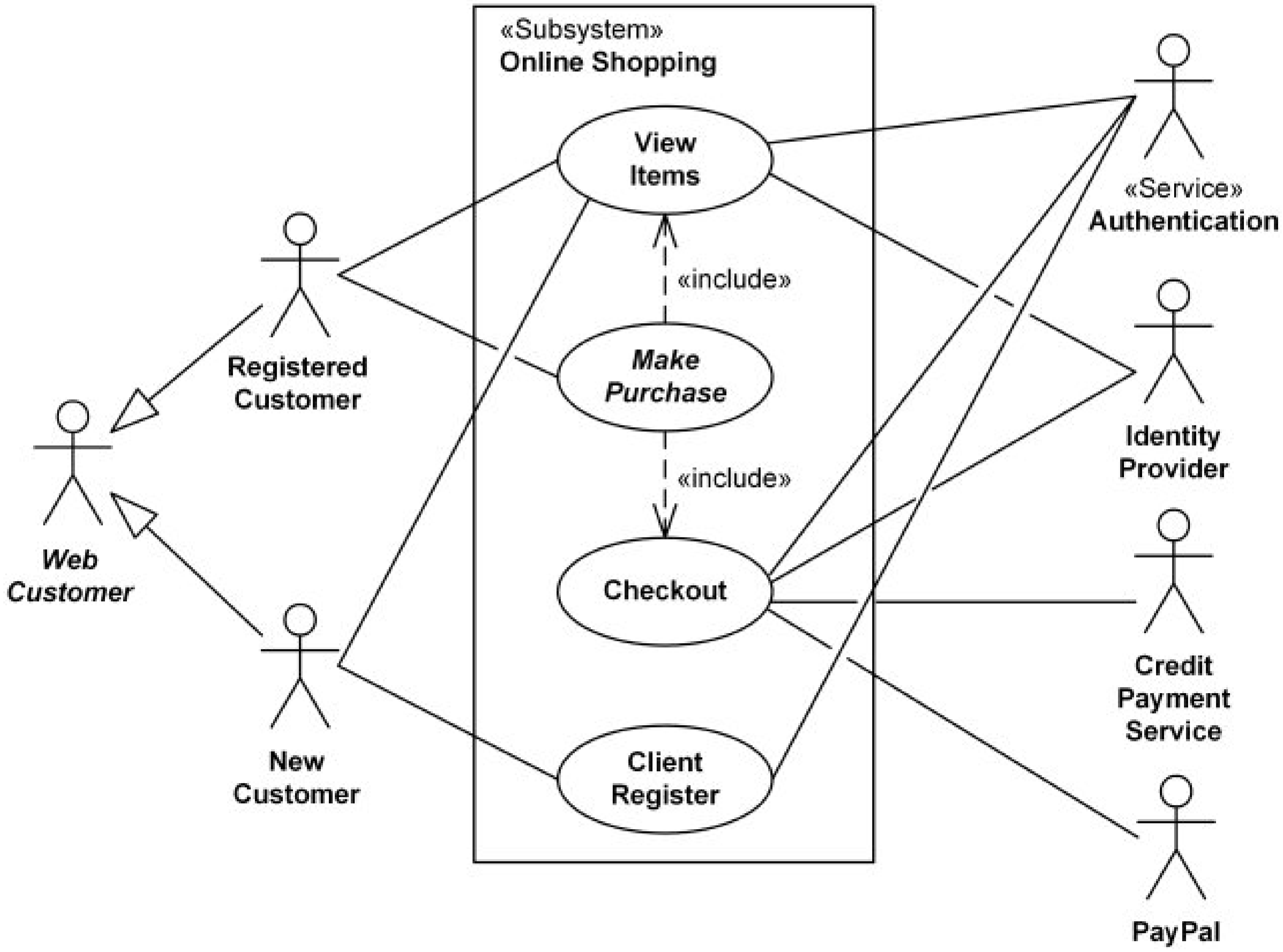
- **Checkout** use case includes several required uses cases- Customer authentication, View/Update Shopping Cart, Calculate Taxes & Shipping, Payment. Web customer should be authenticated. It could be done through user login page, user authentication cookie ("Remember me") or Single Sign-On (SSO). Web site authentication service is used in all these use cases, while SSO also requires participation of external identity provider.
- **Checkout** use case also includes **Payment** use case which could be done either by using credit card and external credit payment service or with PayPal.
- View/Update Shopping Cart includes calculation of total.


User Sign-In
Customer Authentication
Remember Me
Single Sign-On
«Service» Authentication
Identity Provider
«include»
Checkout
«include»
View / Update Shopping Cart
«include»
Calculate Total
«include»
Calculate Taxes and Shipping
«include»
Payment By Credit
Credit Payment Service
Payment
Payment By PayPal
PayPal

Online Shopping Use Case Diagram Example

- Web Customer actor uses some web site to make purchases online. Top level use cases are View Items, Make Purchase and Client Register. View Items use case could be used by customer as top level use case if customer only wants to find and see some products. This use case could also be used as a part of Make Purchase use case. Client Register use case allows customer to register on the web site, for example to get some coupons or be invited to private sales. Note, that Checkout use case is included use case not available by itself - checkout is part of making purchase. Web site authentication service is needed for View items, checkout and Client Register use cases. Identity Provider Service is needed for Checkout and View Items.
- Except for the Web Customer actor there are several other actors:- Registered Customer, New Customer, Credit Payment Service<<external system>>, PayPal<<external system>>

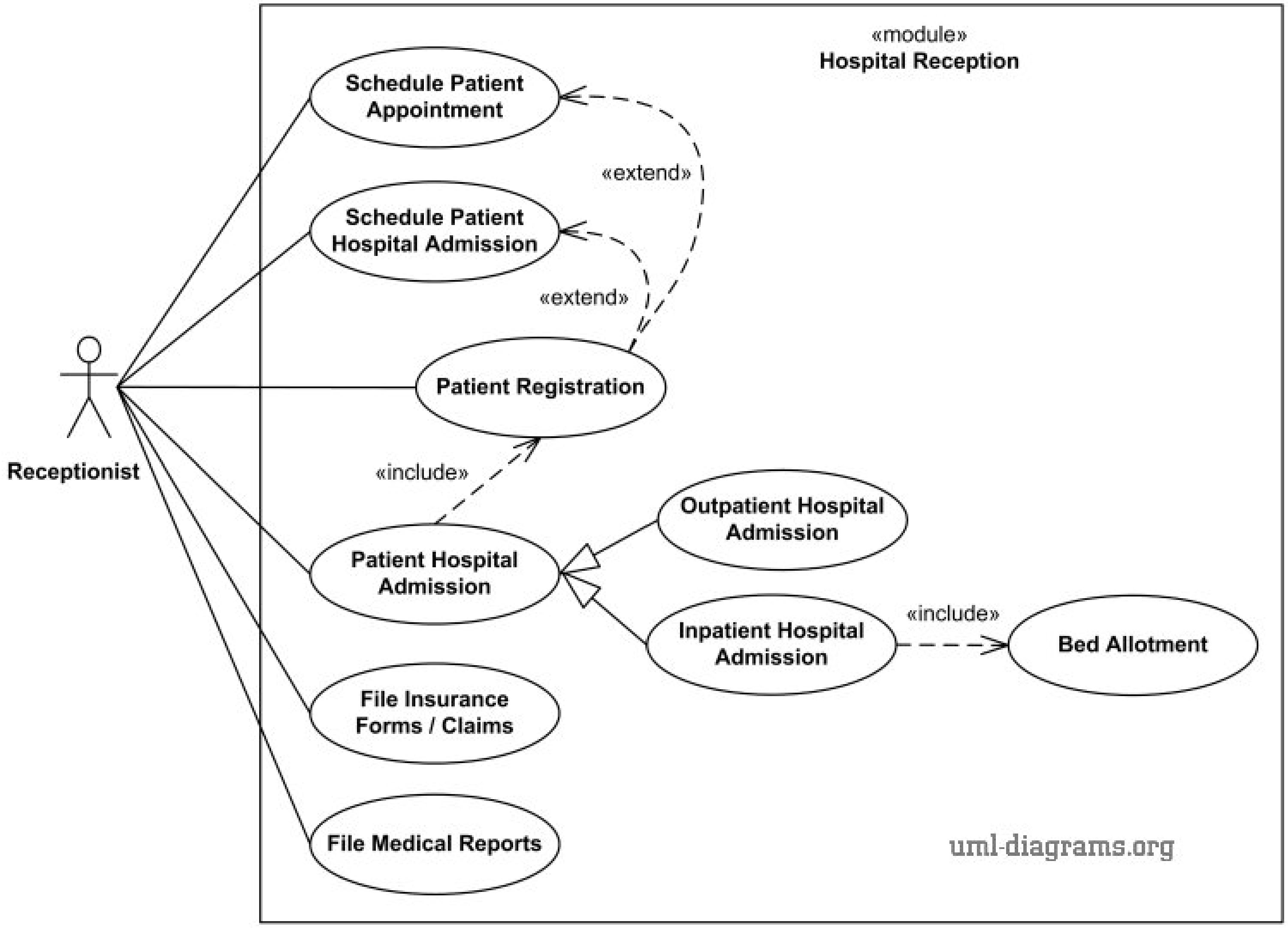
https://www.uml-diagrams.org/examples/online-shopping-use-case-diagram-example.html?context=uc-examples


«Subsystem»
Online Shopping
View Items
«include»
Make Purchase
«include»
Checkout
Client Register
Registered Customer
Web Customer
New Customer
«Service» Authentication
Identity Provider
Credit Payment Service
PayPal

Hospital Reception System

- **Hospital Management System** is a large system including several subsystems or modules providing variety of functions. UML use case diagram example below shows actor and use cases for a hospital's reception.
- ***Purpose:*** *Describe major services (functionality) provided by a hospital's reception.*
- **Hospital Reception** subsystem or module supports some of the many job duties of hospital receptionist. Receptionist schedules patient's appointments and admission to the hospital, collects information from patient upon patient's arrival and/or by phone. For the patient that will stay in the hospital ("inpatient") she or he should have a bed allotted in a ward. Receptionists might also receive patient's payments, record them in a database and provide receipts, file insurance claims and medical reports.

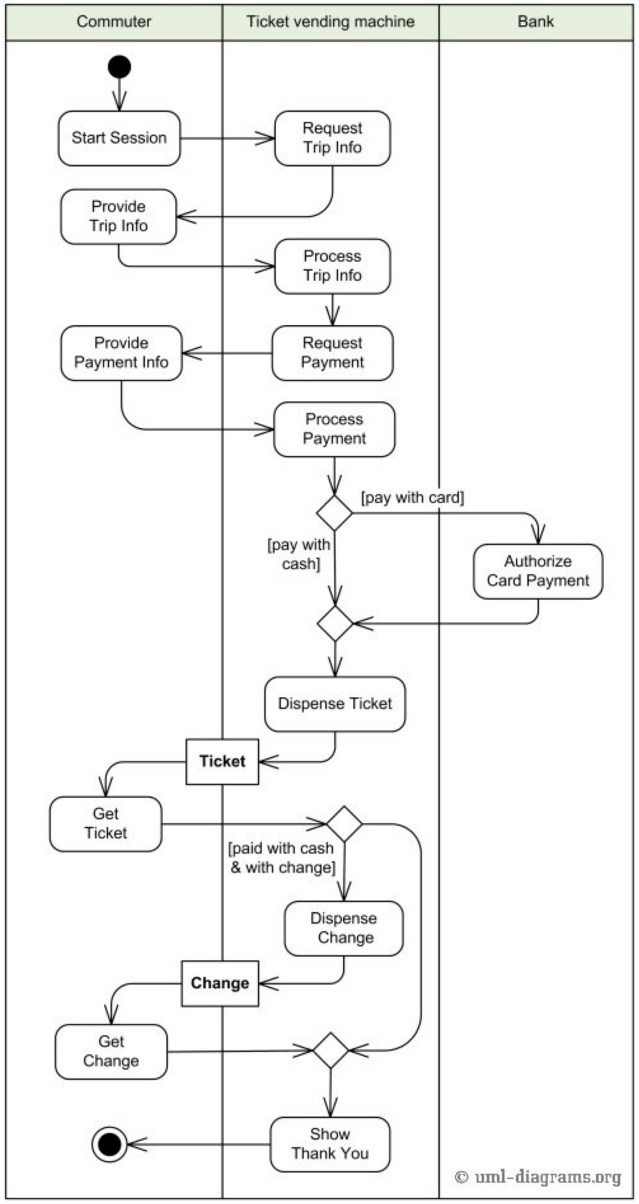
Hospital Reception System


«module»
Hospital Reception
Receptionist
Schedule Patient Appointment
Schedule Patient Hospital Admission
Patient Registration
«extend»
«extend»
«include»
Patient Hospital Admission
Outpatient Hospital Admission
Inpatient Hospital Admission
«include»
Bed Allotment
File Insurance Forms / Claims
File Medical Reports
uml-diagrams.org

Ticket Vending Machine
Activity Diagram for "Purchase Ticket" use case

- Activity is started by Commuter **actor** who needs to buy a ticket. Ticket vending machine will request trip information from Commuter. This information will include number and type of tickets, e.g. whether it is a monthly pass, one way or round ticket, route number, destination or zone number, etc.
- Based on the provided trip info ticket vending machine will calculate payment due and request payment options. Those options include payment by cash, or by credit or debit card. If payment by card was selected by Commuter, another actor, Bank will participate in the activity by authorizing the payment.
- After payment is complete, ticket is dispensed to the Commuter. Cash payment might result in some change due, so the change is dispensed to the Commuter in this case. Ticket vending machine will show some "Thank You" screen at the end of the activity.

Ticket Vending Machine
Activity Diagram for "Purchase Ticket" use case



References

- https://www.uml-diagrams.org/use-case-diagrams-examples.html
- https://www.uml-diagrams.org/examples/online-shopping-use-case-diagram-example.html?context=uc-examples