

# USE CASE DIAGRAM

Scenario Based

# Requirements Specification vs Analysis Model

Both focus on the requirements from the user's view of the system

- The **requirements specification** uses natural language (derived from the problem statement)
- The **analysis model** uses a formal or semi-formal notation (we use UML)

# USE CASE DIAGRAM

- The use case represents the different ways in which a system can be used by the users
- A **use case** describes interactions of actors with the target system to perform a function.
- The use case model can be documented by drawing a use case diagram and writing an accompanying text elaborating the drawing (use case specification)

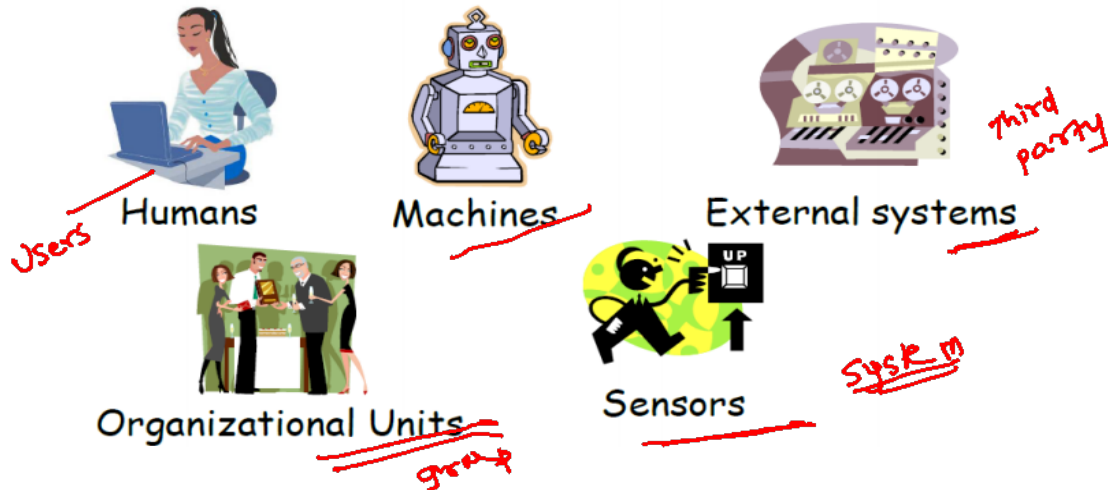
# Use Case Diagrams

- Use case diagrams are used to visualize, specify, construct, and document the (intended) behavior of the system, during requirements capture and analysis.
- Provide a way for developers, domain experts and end-users to Communicate.
- Serve as basis for testing.
- The use case diagram consists of
  - Actor
  - Use cases
  - Relationships

# Actors

External objects that produce/consume data:

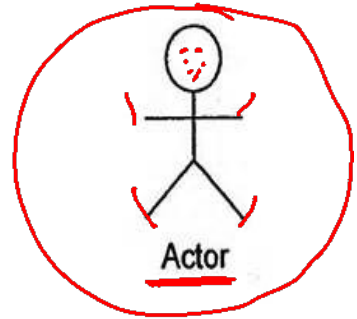
- Must serve as sources and destinations for data
- Must be external to the system



# ACTOR

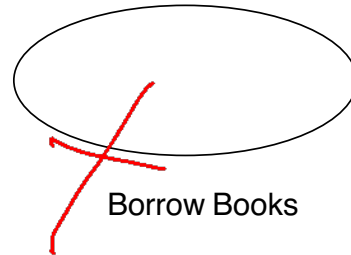
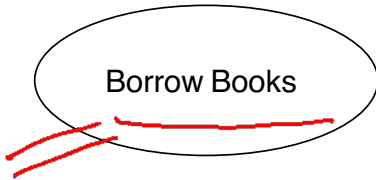
- Each Actor must be linked to a use case
- Represented by stick figure
- Labelled using a descriptive noun or phrase

Eg: Librarian, Doctor, Student, Customer



# USE CASE

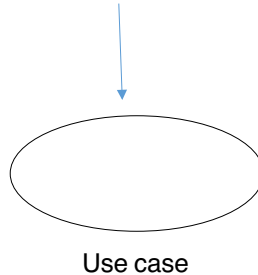
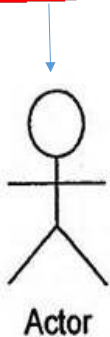
- Labelled using a descriptive verb-noun phrase
- Represented using an ellipse with name of the use case written inside the ellipse or written below the ellipse



- Consider the following scenario:

*“A patient calls the clinic to make an appointment for a yearly checkup”*

- A patient calls the clinic to make an appointment for a yearly checkup.



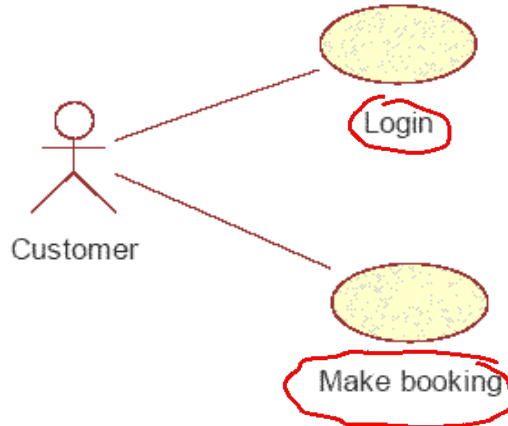


- The picture below is a **Make Appointment** use case for the medical clinic.
- The actor is a **Patient**. The connection between actor and use case is a **communication association** (or **communication** for short).



# RELATIONSHIPS

- Represents communication between actor and use case
- Depicted by a line connecting actor and use case



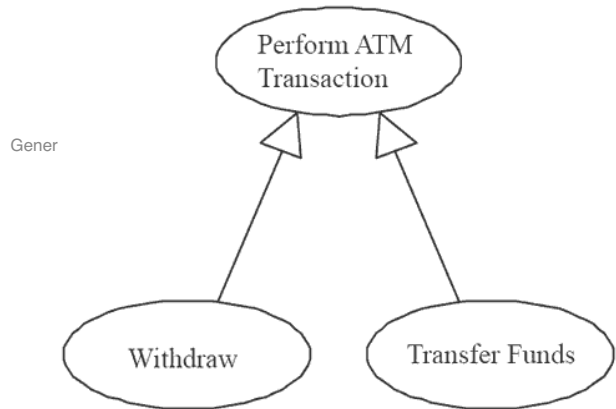
- Other Types of Relationships for Use Cases

- ~~Generalization~~ (in old version 1.0 but removed from 2.0)
- ~~Include~~
- ~~Extend~~

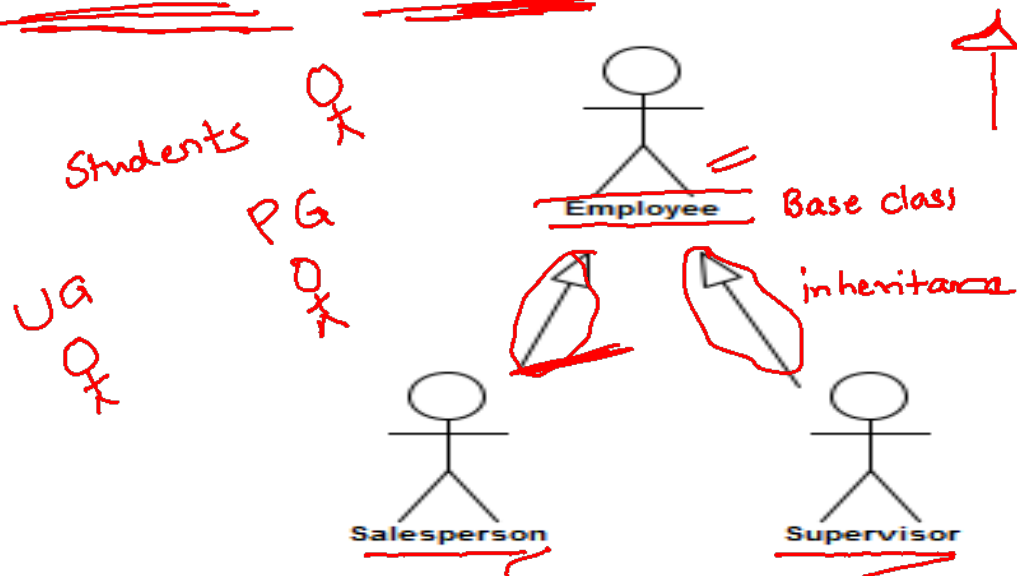
use 2.0

## Generalization Relationship

- refers to the relationship which can exist between two use cases and which shows that one use case (child) inherits the structure, behavior, and relationships of another actor (parent).
- The child use case is also referred to the more specialized use case while the parent is also referred to as the more abstract use case of the relationship.
- Generalization is possible within use cases and also between actors
- Represented by a line and a hollow arrow

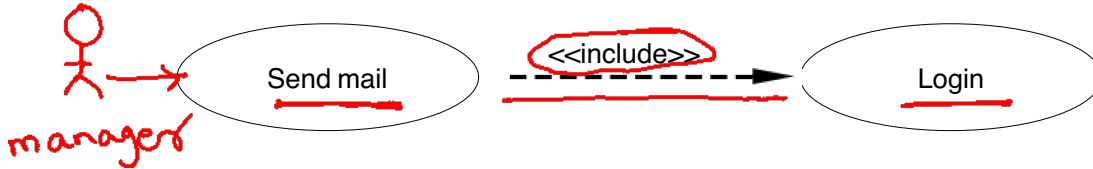


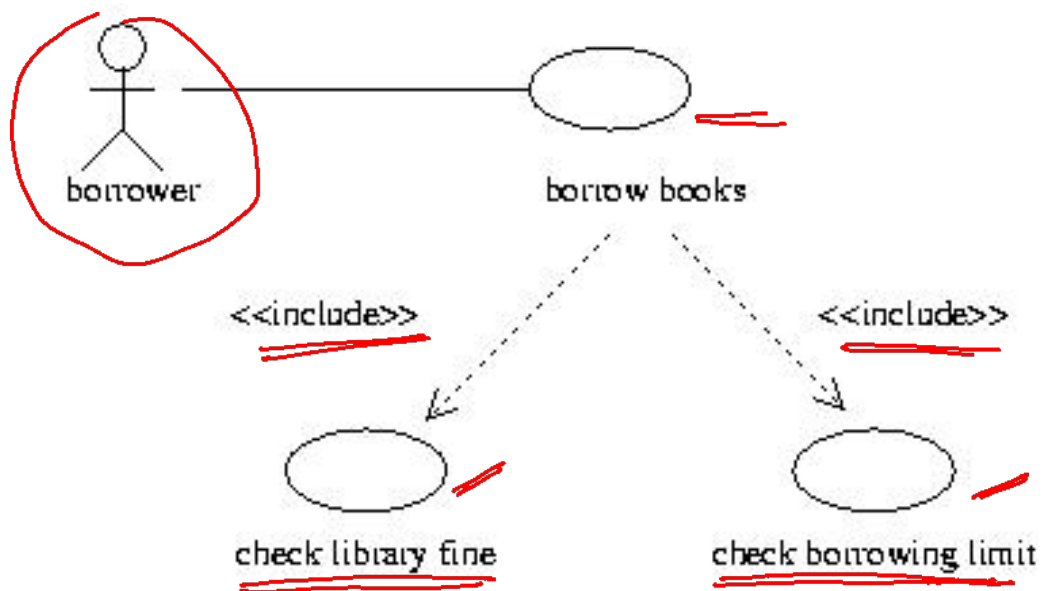
- Generalization between actors



# Include Relationship

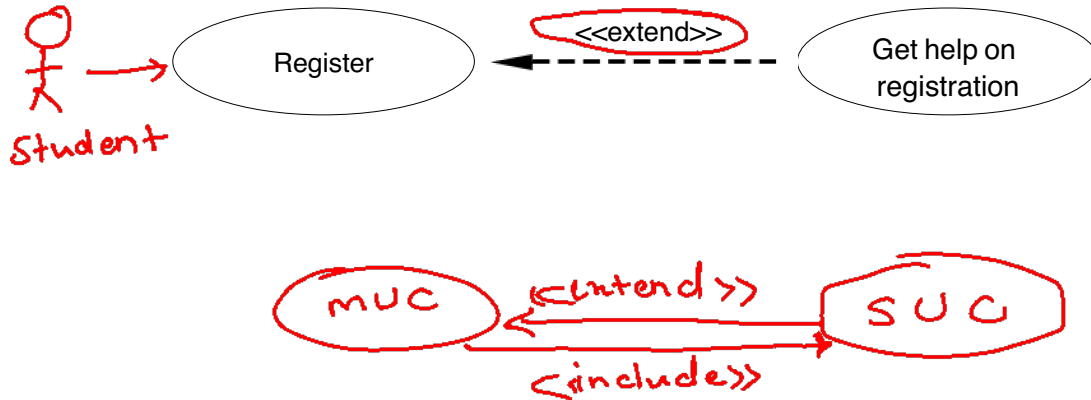
- Represents the inclusion of the functionality of one use case within another
- Arrow is drawn from the base use case to the used use case
- Write <<include>> above arrowhead line





## Extend relationship

- Represents the extension of the use case to include optional functionality
- Arrow is drawn from the extension use case to the base use case
- Write << extend >> above arrowhead line





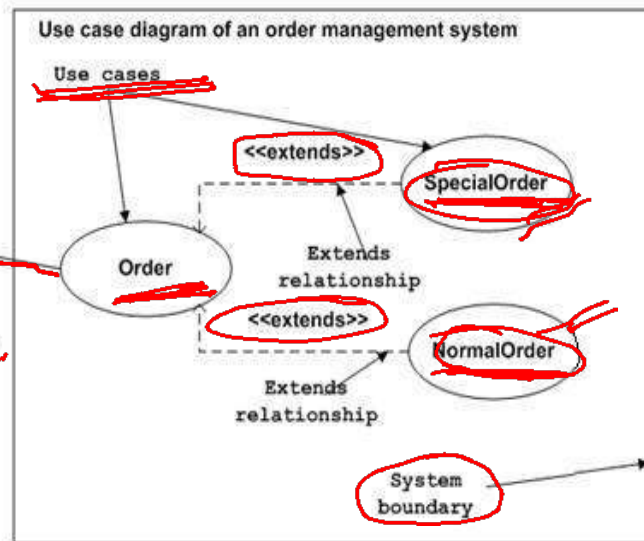
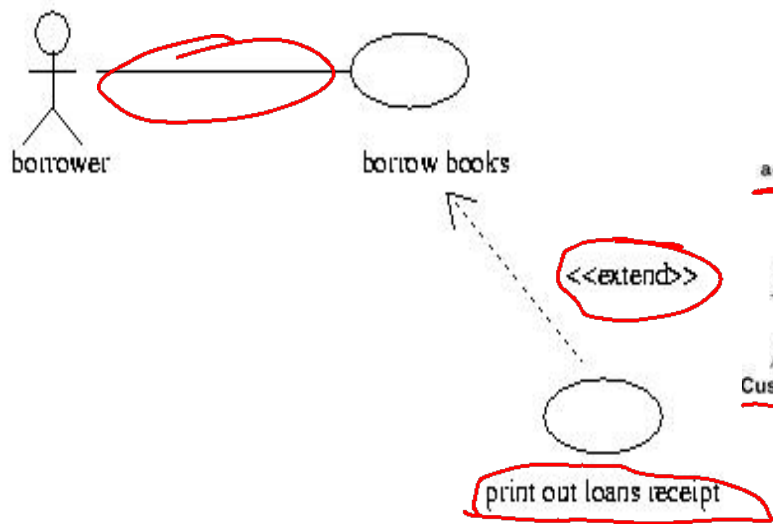
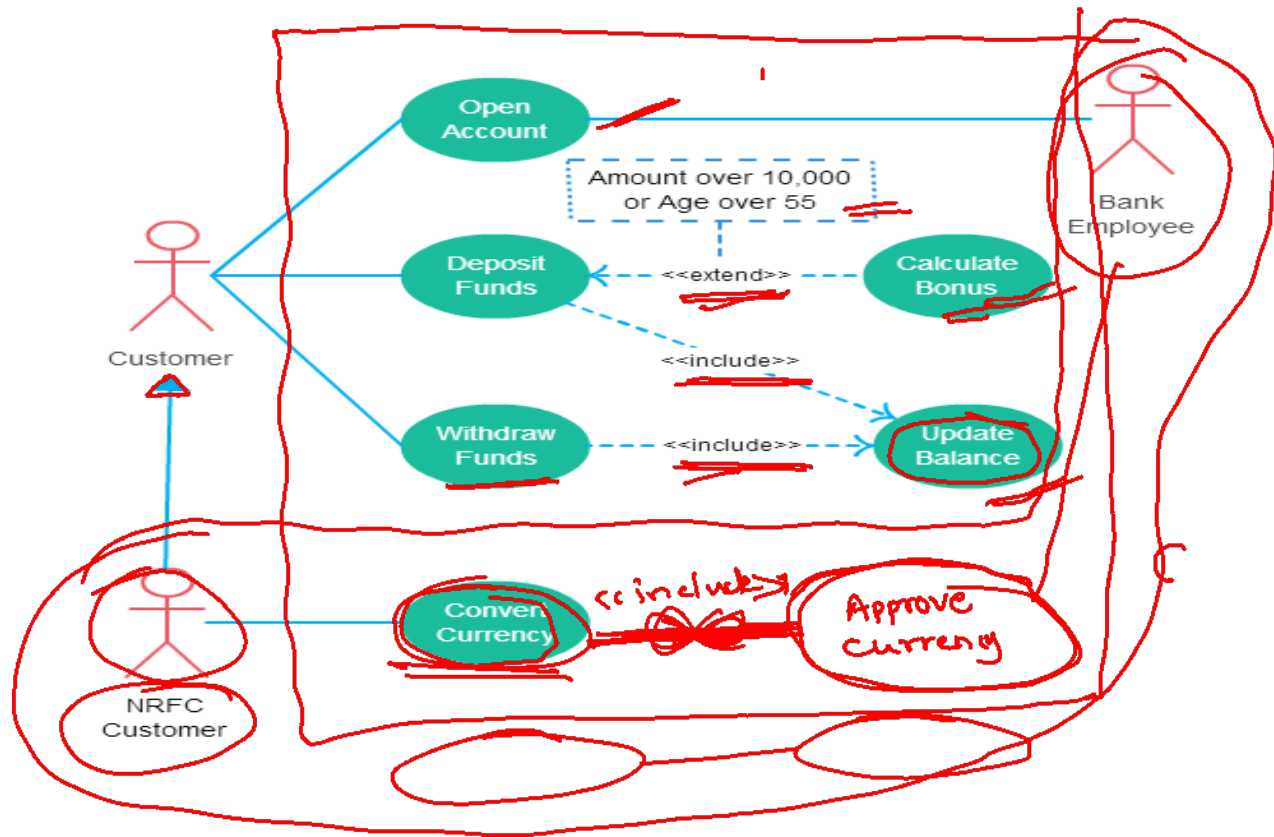
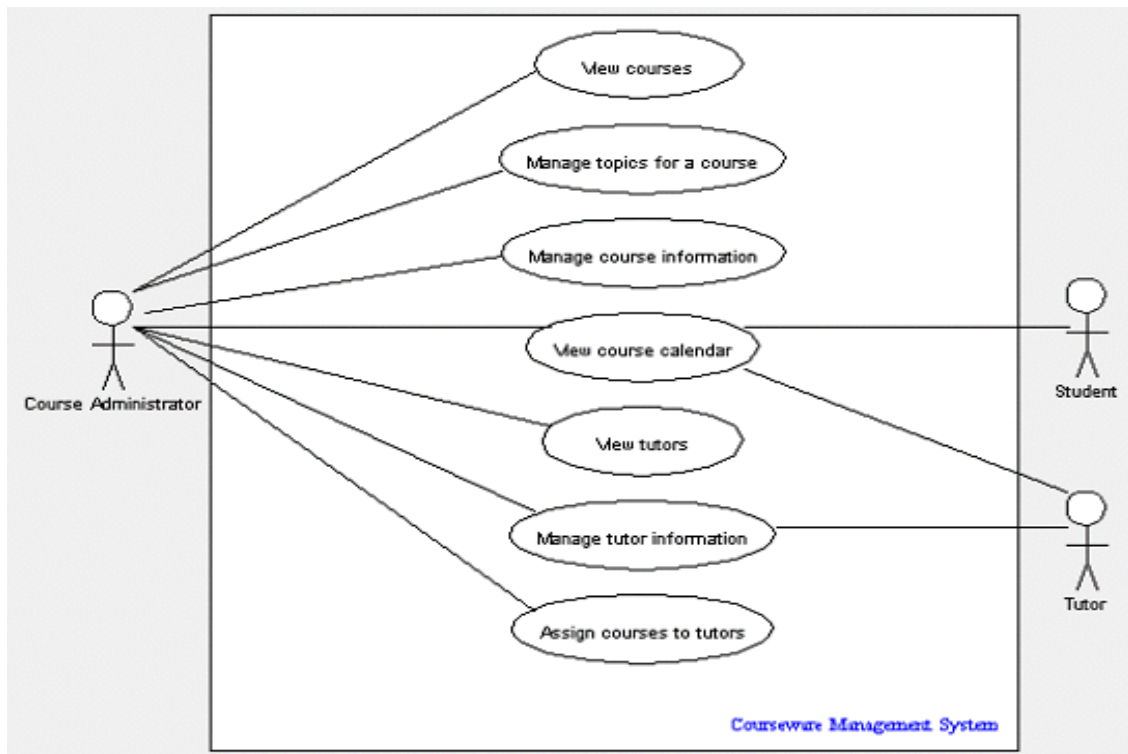


Figure: Sample Use Case diagram



# Use case diagram example



# Problem Statement 1

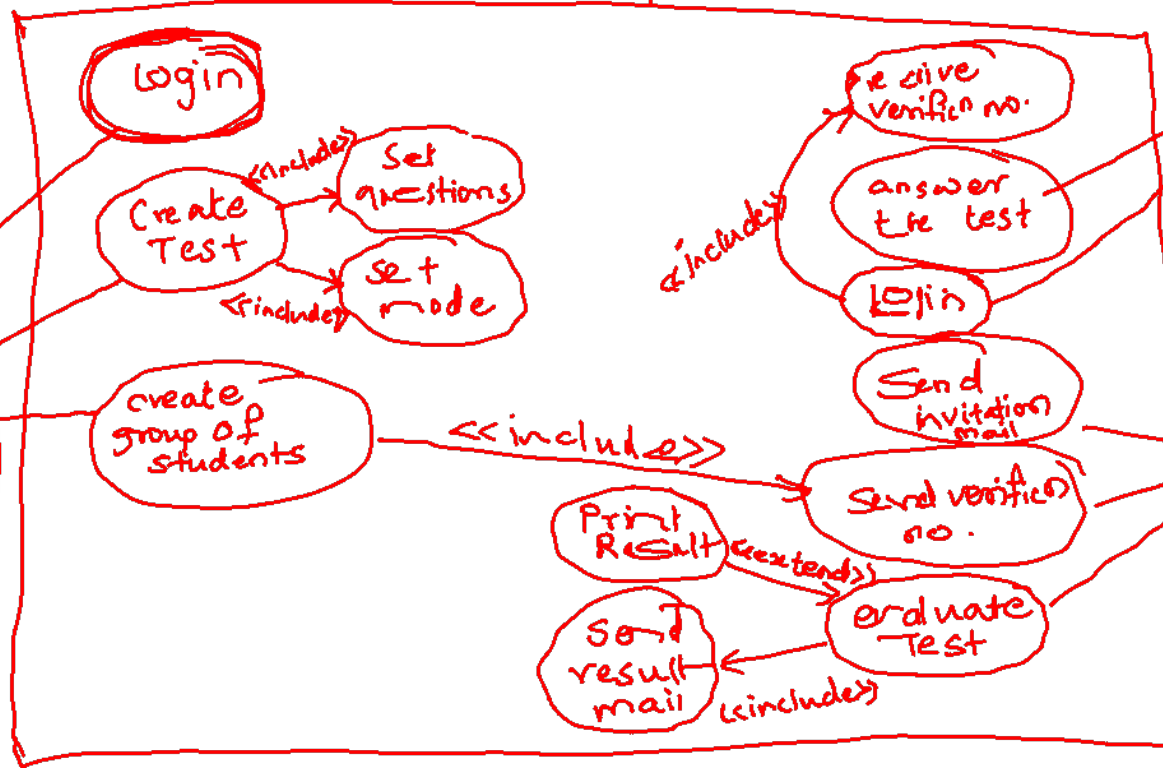
- ① The System is intended for faculty members to conduct online examination. Every registered faculty member can login to the system, create test, set the questions, set the mode for each test etc. Faculty member also decides which groups of students have to take a particular test and accordingly creates group of students. The system should send mail to the students inviting them to take the test. ②
- ③ Students can login using the verification number received and then answer the test questions. The System should evaluate student answers and send the result via mail or can print the result.

# Problem Statement 1

The System is basically intended for **faculty** members to conduct online examination. Every **registered faculty** member can login to the system, create test, set the questions, set the mode for each test etc. Faculty member also decides which groups of students have to take a particular test and accordingly creates group of students. The system should send mail to the students inviting them to take the test. **Students** can login using the verification number received and then answer the test questions. The **System** should evaluate student answers with that specified by faculty and send the result via mail.

Faculty

Registered  
Faculty



Student

System

## Problem Statement 2

An **auto rental company** wants to develop an automated system that would handle car reservations, customer billing, and car auctions. Usually a customer reserves a car, picks it up, and then returns it after a certain period of time. At the time of pick up, the customer has the option to buy or waive collision insurance on the car. When the car is returned, the customer receives a bill and pays the specified amount. In addition to renting out cars, every six months or so, the auto rental company auctions the cars that have accumulated over 20,000 miles.





# Problem Statement 3

Draw the use-case diagram for Hotel Information System. There are two types of customers: Tour-group customers and Individual customers. Both can book, cancel, check-in and check-out of a room by Phone or via the Internet. There are booking process clerk and reception staff who manage it. A customer can pay his bill by credit card or by cash. After frequent booking of hotel more than 5 times, customer can go for booking of special package of stay for 6<sup>th</sup> time. Clerk also maintains the resources available in the hotel and reception staff takes care of feedback collection from the customer.

# Problem Statement 4

A student may register for classes during a specified registration period. To register, a student must see their advisor. The advisor must approve each course that the student has selected. The advisor will use the registration system to determine if the student has met the course prerequisites, is in good academic standings and is eligible to register. If the advisor approves the courses, the advisor enters the student's college id into the course registration system. The course registration number for each course is entered. The course description, course number and section for those courses will automatically display. The system will check for schedule conflicts before saving the registrations. A bill for the courses will print in the financial administrator's office. The student should proceed to pick it up. Faculty can use the registration system to check enrollments in their classes, get a class list, check a student's transcript, look up a student's phone number and other such student information. The registrar can use the registration system to enter new classes for an upcoming semester, cancel a class, and check conflicts in classroom/faculty assignments. Admissions use the registration system to add new students. Enrollment services use the registration system to report on retention, update student information, and check fulfillment of graduation requirements for those students planning to graduate.

# Types of Requirements

- **Functional requirements**

- Describe the interactions between the system and its environment independent from the implementation
  - “An operator must be able to define a new game. “

- **Nonfunctional requirements**

- Aspects not directly related to functional behavior.
  - “The response time must be less than 1 second”

- **Constraints**

- Imposed by the client or the environment
  - “The implementation language must be Java “
- Called “**Pseudo requirements**” in the text book.

# Functional vs. Nonfunctional Requirements

## Functional Requirements

- Describe user tasks that the system needs to support
- Phrased as actions
  - “Advertise a new league”
  - “Schedule tournament”
  - “Notify an interest group”

## Nonfunctional Requirements

- Describe properties of the system or the domain
- Phrased as constraints or negative assertions
  - “All user inputs should be acknowledged within 1 second”
  - “A system crash should not result in data loss”.

# Types of Nonfunctional Requirements

- Usability
  - Reliability
    - Robustness
    - Safety
  - Performance
    - Response time
    - Scalability
    - Throughput
    - Availability
  - Supportability
    - Adaptability
    - Maintainability
- Implementation
  - Interface
  - Operation
  - Packaging
  - Legal
    - Licensing (GPL, LGPL)
    - Certification
    - Regulation

Constraints or  
Pseudo requirements

Quality requirements

# Nonfunctional Requirements: Examples

- “Spectators must be able to watch a match without prior registration and without prior knowledge of the match.”  
[?] *Usability Requirement*
- “The system must support 10 parallel tournaments”  
[?] *Performance Requirement*
- “The operator must be able to add new games without modifications to the existing system.”  
[?] *Supportability Requirement*

# USE CASE SPECIFICATION

(detailed version of a use case)

# Use Case Specification

1. **Brief Description** - (about the use case)
2. **Actors** – list out the actors involved
3. **Preconditions** - A textual description that defines any constraints on the system at the time the use case may start.
4. **Basic flow of events** - describes what "normally" happens when the use case is performed.
5. **Alternative flows** - covers behavior of an optional or exceptional character relative to normal behavior, and also variations of the normal behavior. Think of the alternate flows of events as "detours" from the basic flow of events.
6. **Postconditions** - A textual description that defines any constraints on the system at the time the use case will terminate.



# Use-Case Specification: Withdraw Cash

## 1. Brief Description

This use case describes how a Bank Customer uses an ATM to withdraw money from a bank account.

## 2. Actors

- Customer
- Bank

## 3. Preconditions:

- The bank Customer must possess a **bank card**.
- The network connection to the **Bank System** must be active.
- The system must have at least some cash that can be dispensed.
- The cash withdrawal **service option** must be available.

## 4. Basic Flow of Events

### 4.1 Insert Card

- The use case begins when the actor **Customer** inserts his/her **bank card** into the card reader on the ATM.
- The system allocates an **ATM session identifier** to enable errors to be tracked and synchronized between the ATM and the Bank System.

### 4.2 Read Card

- The system reads the **bank card information** from the card.

### 4.3 Authenticate Customer

- Authenticate the use of the **bank card** by the individual using the machine

### 4.4 Select Withdrawal

- The system displays the **service options** that are currently available on the machine.
- The Customer selects to withdraw cash.

### 4.5 Select Amount

- The system prompts for the amount to be withdrawn by displaying the list of standard withdrawal amounts.
- The Customer selects an amount to be withdrawn.

### 4.6 Confirm Withdrawal

- Customer gives conformation to withdraw cash

### 4.7 Eject Card

- The system ejects the Customer's **bank card**.
- The Customer takes the **bank card** from the machine.

### 4.8 Dispense Cash

## **5.Alternative Flows**

**5.1 Invalid User**

**5.2 Amount Exceeds Withdrawal limit**

**5.3 Insufficient cash**

**5.4 Money not removed**

## **6. Postconditions**

- The ATM has returned the card and dispensed the cash to the Customer and the withdrawal is registered on the Customer's account.
- The ATM has returned the card to the Customer and no withdrawal is registered on the Customer's account.
- The ATM has returned the card but has not supplied the amount of cash registered as withdrawn from the Customer's account. The discrepancy is registered in the ATM's log.
- The ATM has kept the card, no withdrawal has registered on the Customer's account and the Customer has been notified where to contact for more information.

Write a use case specification for

- i) **book ticket** use case in a **railway ticket booking system**.
- ii) **Order item** use case in an **online shopping website**.

Millennium Travel Corporation (MTC) travel agency plans to become a market leader by augmenting its human travel agents with an automated travel agent system for processing flight reservations. The automated travel agent will intermediate between travelers and the MTC corporate computing system, which interfaces with commercial airline reservation services. Like a human travel agent, it will assist travelers in booking, changing, and canceling flight reservations. If, for any reason, a traveler making a flight reservation travel request prefers human assistance, she will have the option to interact directly with a human travel agent.

The MTC automated travel agent system will process a wide range of flight reservation service requests. These include but are not limited to: inquiring about flights and airfares, making, changing, and canceling traveler profiles and accounts, booking, changing, confirming, and canceling flight reservations, generating travel itineraries.

A user with a valid system account and a valid travel account logs in to the system, requests to book a flight reservation, selects a flight, selects a payment method, and specifies delivery services for the flight tickets and travel itineraries.

The travel agent system must be capable of providing fast, accurate, and courteous ("user friendly") services for all requests supported. The system must be able to answer inquiries about flights and fares, generate, modify, and cancel traveler profiles and travel accounts, make, change, complete, and cancel reservations, obtain payment method and verify traveler credit line, generate travel itineraries and arrange for delivery of flight tickets and flight itineraries.