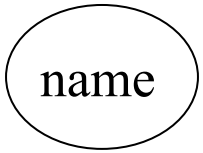
Use case diagrams

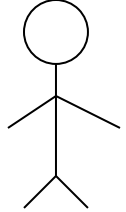
Dr. Muhammad Taimoor Khan
Taimoor.khan@nu.edu.pk

1. Use Case



- Use cases specify desired behavior.
- A use case is a description of a set of sequences of actions, including variants, a system performs to yield an observable result of value to an actor.
- Each sequence represent an interaction of actors with the system.

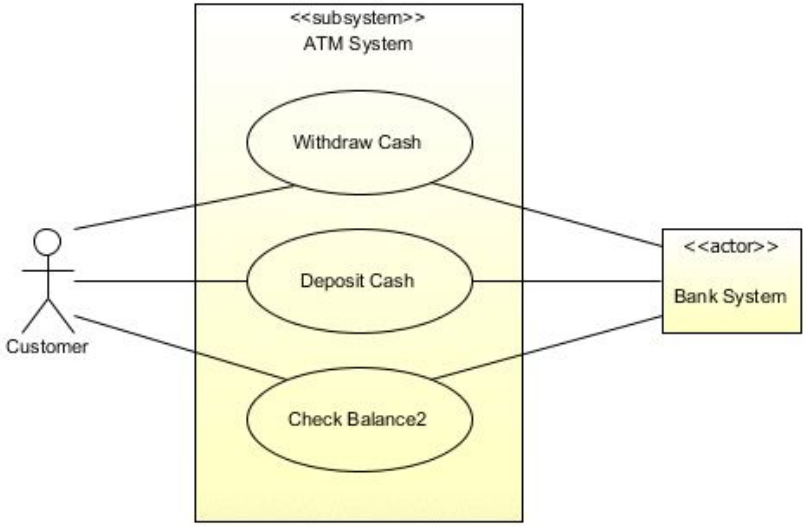
2. Actors



- Actors represent a set of roles that users of use case play when interacting with these use cases.
- Actors are entities which require help from the system to perform their task or are needed to execute the system's functions.
- Actors are not part of the system.

System Actor

- Actors can be an automated systems
- Such a system already exist and will communicate the system to be built
- Can be primary or secondary user

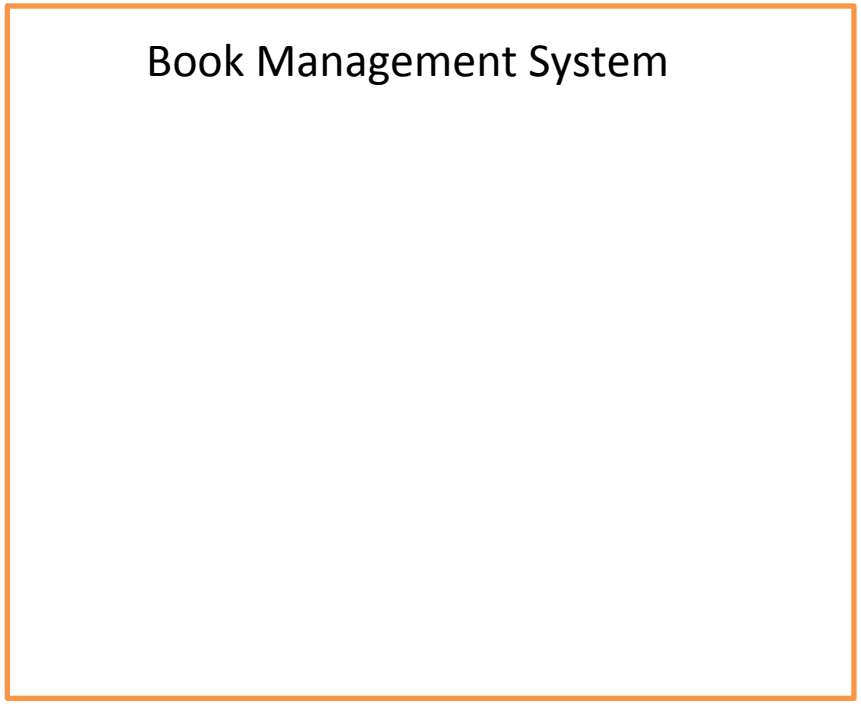


3. Association

- Shows presence or absence of possible communication between an actor and a use case
- It presents a static view of association i.e., not giving any information about the sequence and count of messages communicated
- It doesn’t have a direction and therefore doesn’t indicate the orientation of communication (Its too early for that!)

4. System Boundary

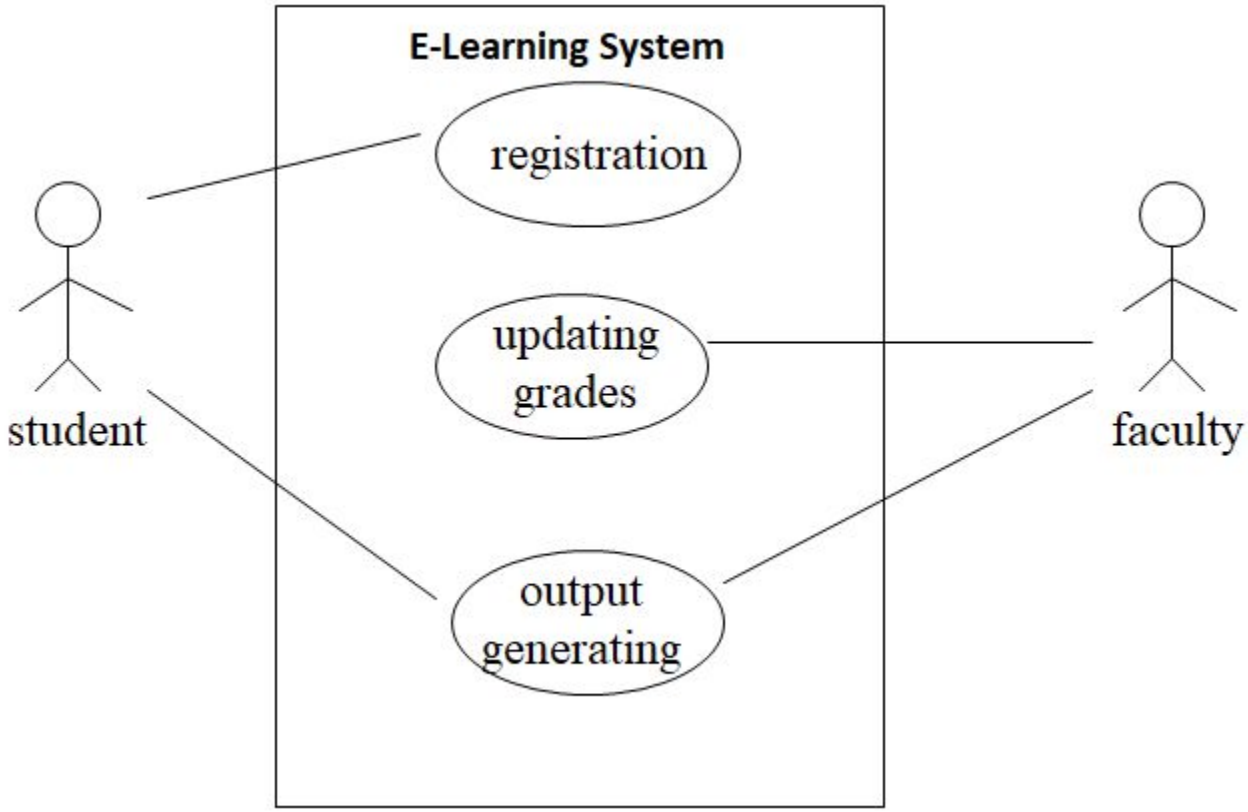
- It has everything to be built within it
- It defines scope of the system
- Everything outside the boundary is system's environment



5. System Name:

Within Boundary

Example of Use Case Diagram



Relationships between Use Cases

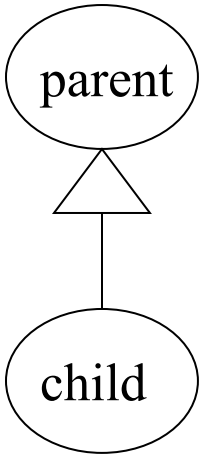
1. Generalization - use cases that are specialized versions of other use cases.

2. Include - use cases that are included as parts of other use cases.

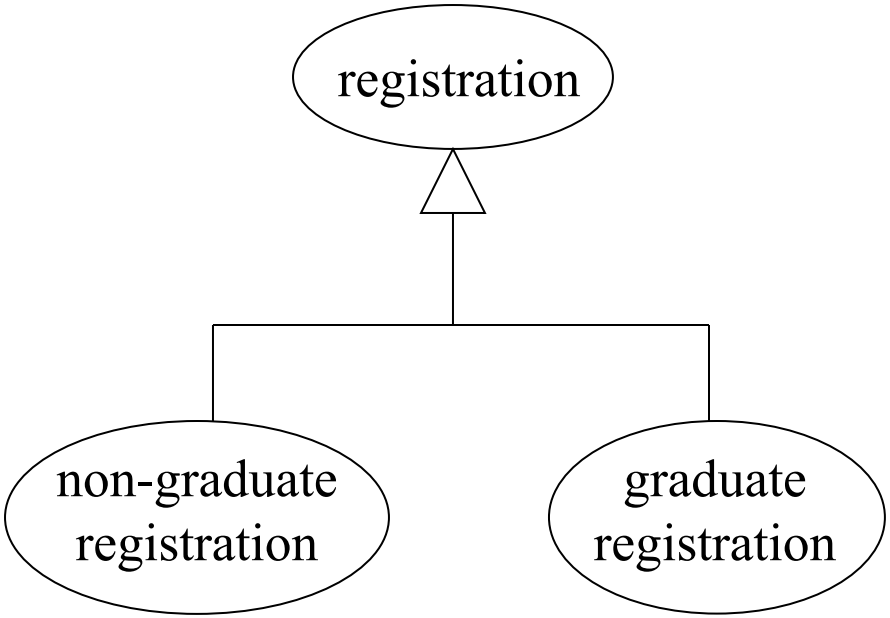
3. Extend - use cases that extend the behavior of other core use cases.

1. Generalization

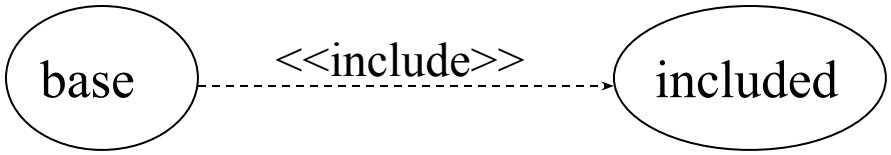
- The child use case inherits the behavior and meaning of the parent use case.
- The child may add to or override the behavior of its parent.



More about Generalization



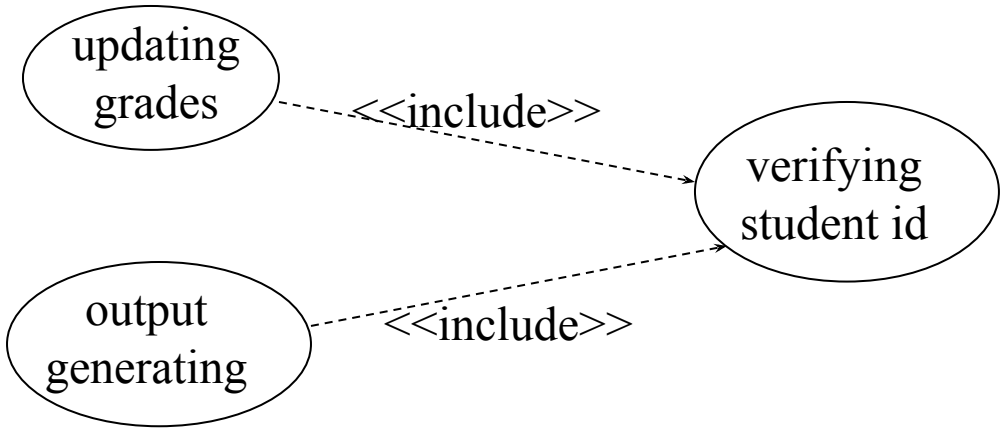
2. Include



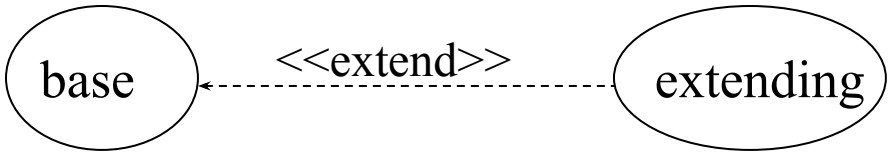
- The base use case explicitly incorporates the behavior of another use case at a location specified in the base
- The included use case never stands alone. It only occurs as a part of some larger base that includes it
- Usually needed at more than one base use case

More about Include

- Enables to avoid describing the same flow of events several times by putting the common behavior in a use case of its own.



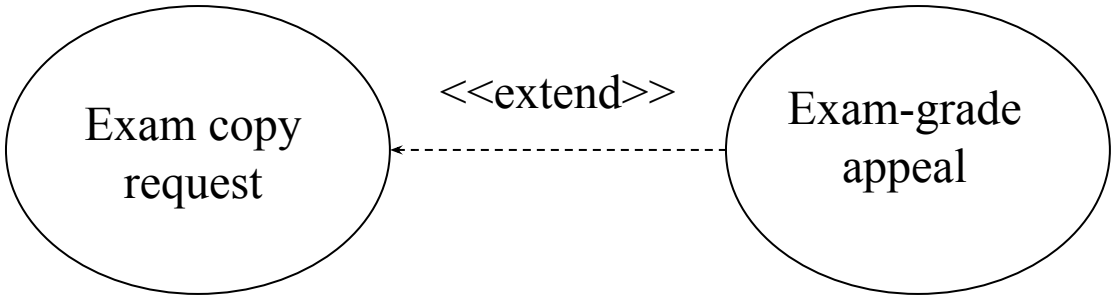
3. Extend



- The base use case implicitly incorporates the behavior of another use case at certain points called extension points.
- The base use case may stand alone, but under certain conditions its behavior may be extended by the behavior of another use case.

More about Extend

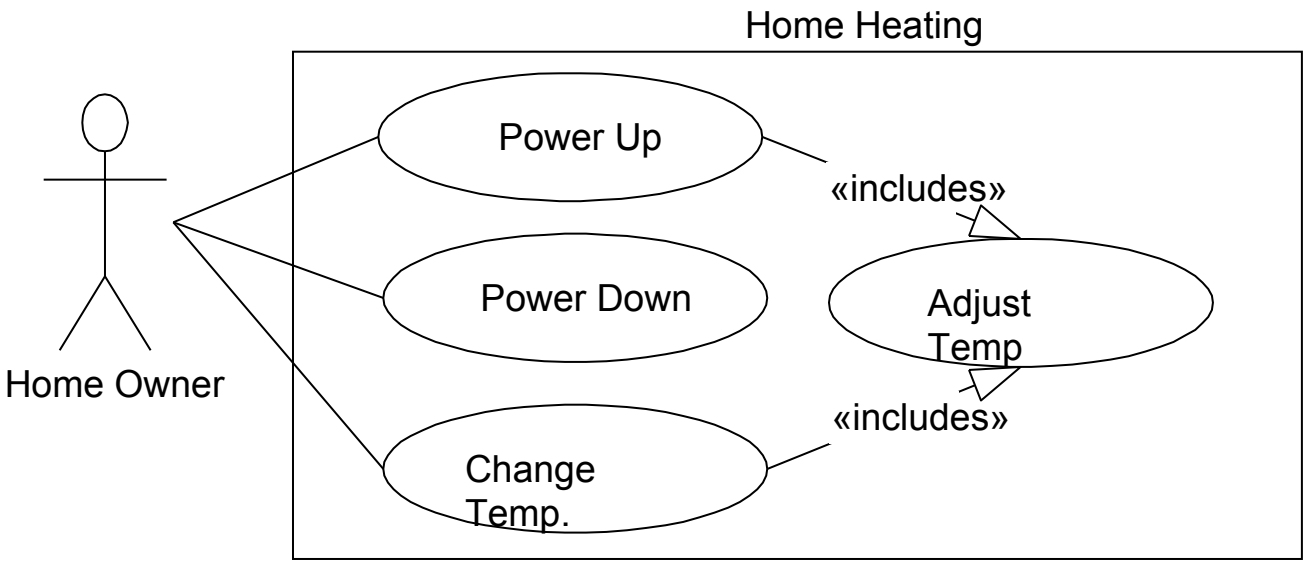
- Enables to model optional behavior or branching under conditions.

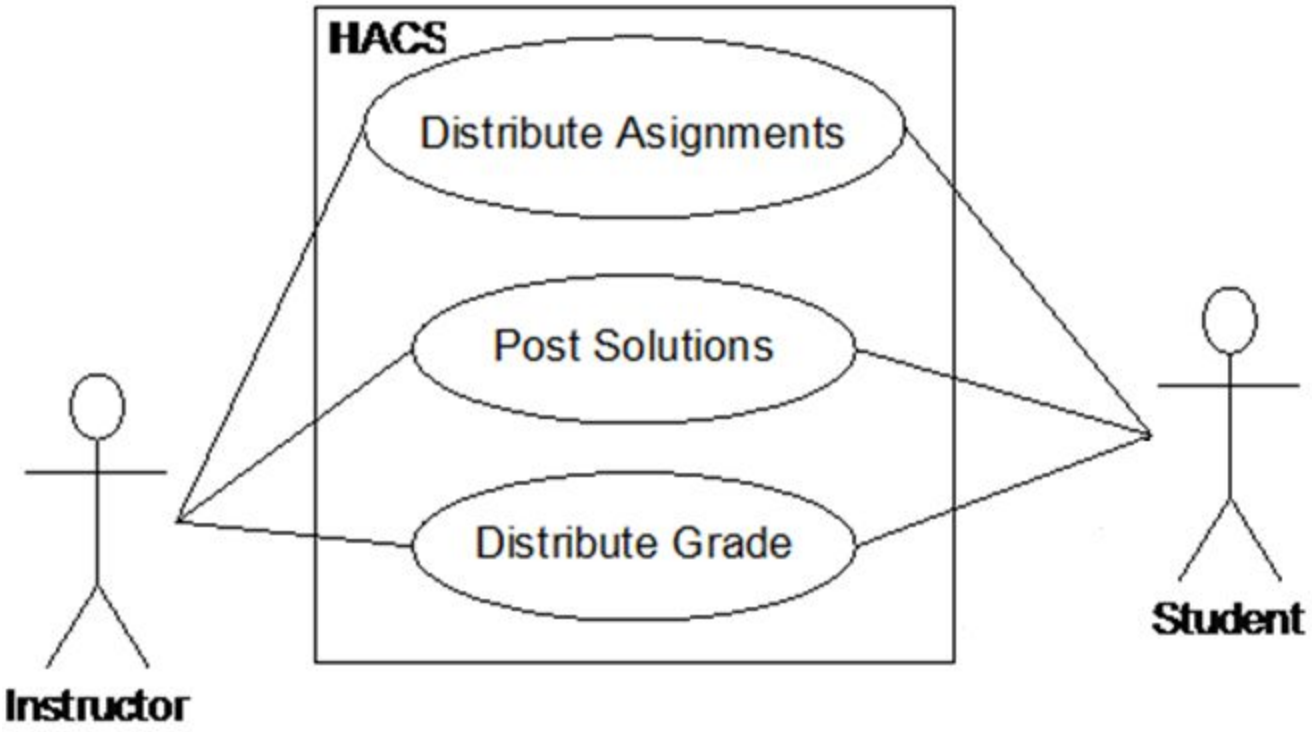


Relationships between Use Cases and Actors

- Actors may be connected to use cases by associations, indicating that the actor and the use case communicate with one another using messages.

Modified Home Heating




HACS
Distribute Asignments
Post Solutions
Distribute Grade
Instructor
Student

Example- Money Withdraw

- Use Case: Withdraw Money
- Author: ZB
- Date: 1-OCT-2004
- Purpose: To withdraw some cash from user's bank account
- Overview: The use case starts when the customer inserts his credit card into the system. The system requests the user PIN. The system validates the PIN. If the validation succeeded, the customer can choose the withdraw operation else alternative 1 –. The customer enters the amount of cash to withdraw. The system checks the amount of cash in the user account, its credit limit. If the withdraw amount in the range between the current amount + credit limit the system dispense the cash and prints a withdraw receipt, else alternative 2 –.

Example- Money Withdraw (cont.)

- Actors: Customer
- Pre Condition:
  - The ATM must be in a state ready to accept transactions
  - The ATM must have at least some cash on hand that it can dispense
  - The ATM must have enough paper to print a receipt for at least one transaction
- Post Condition:
  - The current amount of cash in the user account is the amount before the withdraw minus the withdraw amount
  - A receipt was printed on the withdraw amount
  - The withdraw transaction was audit in the System log file

Example- Money Withdraw (cont.)

- Alternative flow of events:
  - Customer authorization failed. Display an error message, cancel the transaction and eject the card.
  - Customer has insufficient funds in its account. Display an error message
  - Customer exceeds its legal amount. Display an error message

Example- Money Withdraw (cont.)

- One method to identify use cases is actor-based:
  - Identify the actors related to a system or organization.
  - For each actor, identify the processes they initiate or participate in.

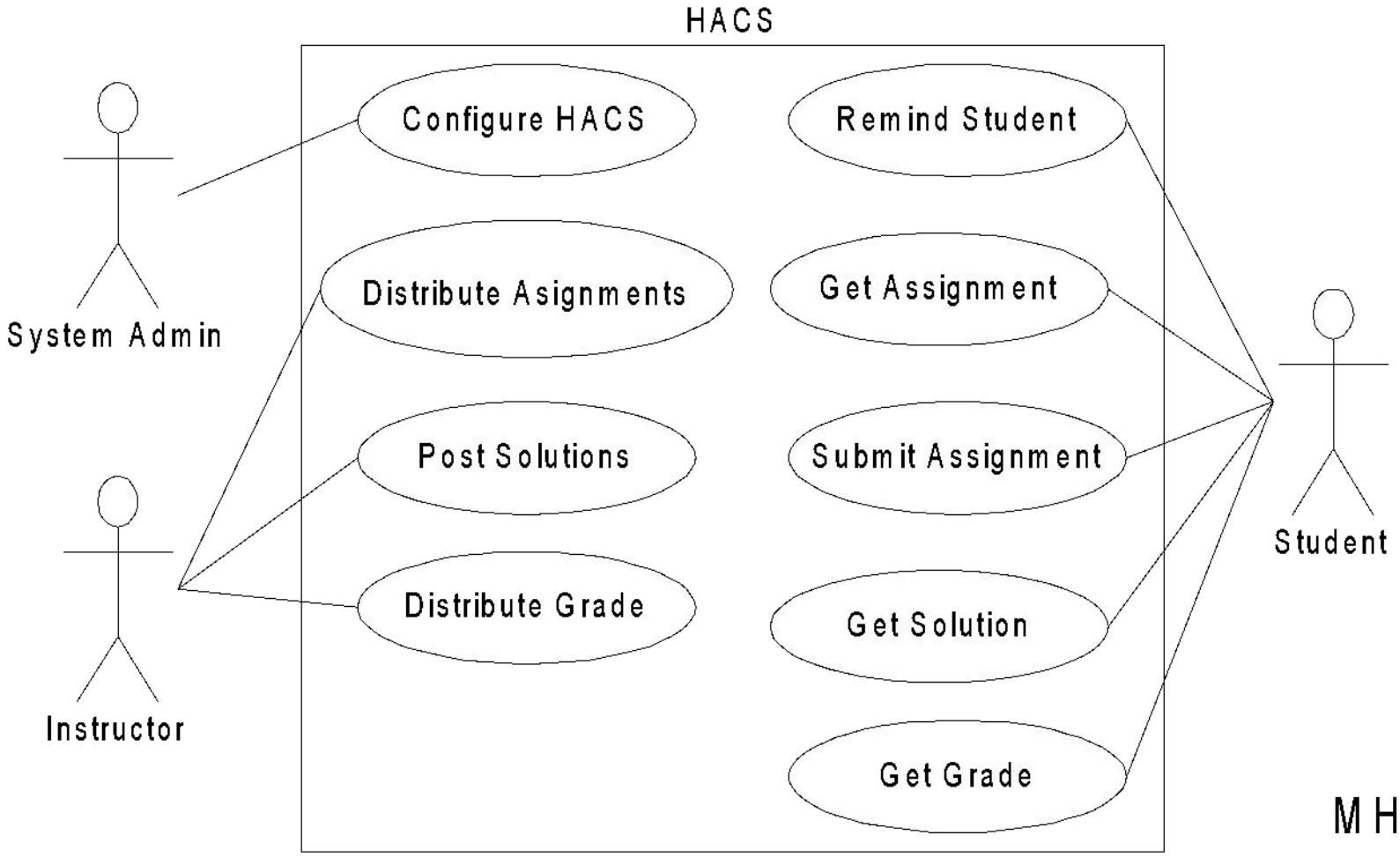
- A second method to identify use cases is event-based:
  - Identify the external events that a system must respond to.
  - Relate the events to actors and use cases.

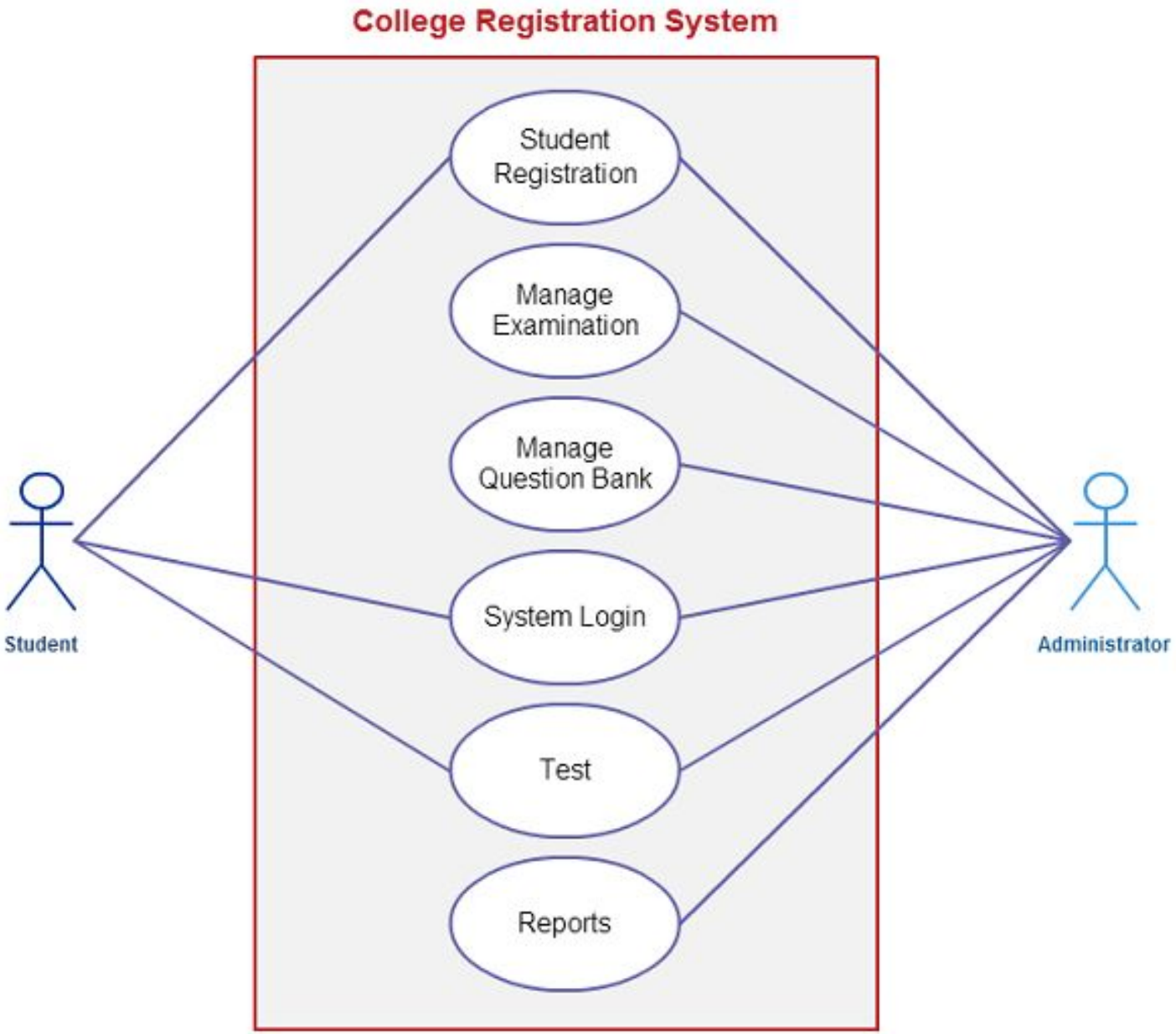
Sample Systems

- Online purchase
- University admission
- Joining sports club
- Transport system
- Flight reservation
- Cellular company
- Recruitment agency ...

Multiple Use-cases in a single Use-case Diagram

(do not try until you have good practice of it)




College Registration System
Student Registration
Manage Examination
Manage Question Bank
System Login
Test
Reports
Student
Administrator