Software Re-Engineering

Use Cases
Use Case Diagrams

Dr. Imran Ali
Assistant Professor
School of Computing

Outline

- What is Use Case?
- Use Case Analysis?
- Use Case Diagram
- Identify Actors
  - What is Actor ?
- Use Case: A Scenario
- Components of Use Case
- Use Case Relationships
  - Relationships
  - Boundary
- Use Case Diagram Stereotypes
  - Generalization
  - Include
  - Extend
- Benefits of Use Cases
- Difficulties with Use Cases
- Case Studies
  - HACS
  - HACS Use Case Diagram
  - Home Heating System

Use Case ?

- A formal way of representing how a business system interacts with its environment
- Illustrates the activities that are performed by the users of the system
- A scenario-based technique in the UML (Unified Modeling Language)
- A sequence of actions a system performs that yields a valuable result for a particular actor.

Use Case Analysis

- What is an Actor?
- A user or outside system that interacts with the system being designed in order to obtain some value from that interaction
- Use Cases describe scenarios that define the interaction between users of the system (the actor) and the system itself.

Use Cases

- **Use case diagrams** :
- describe what a system does from the stand point of an external observer.
- The emphasis is on what a system does rather than how.
- Use case diagrams are closely connected to scenarios.
- A **scenario** is an example of what happens when someone (actor) interacts with the system.

Use Cases

- Here is a scenario for a medical clinic.

- A patient calls the clinic to make an appointment for a yearly checkup. The receptionist finds the nearest empty time slot in the appointment book and schedules the appointment for that time slot.

- We want to write a use case for this scenario.
- Remember: A **use case** is a summary of a single task or goal.

Use Case

- **Step 1: Identify the actors**
- As we read the scenario, find out the people or systems that are going to interact with the scenario.

- Example:
- "A patient calls the clinic to make an appointment for a yearly checkup. The receptionist finds the nearest empty time slot in the appointment book and schedules the appointment for that time slot."

Questions to Identify the People Actors

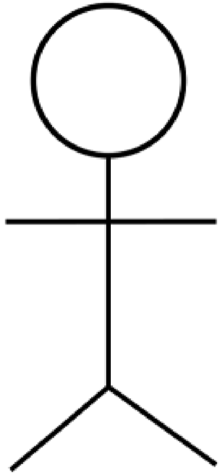
- Who is interested in the scenario/system ?
- Where in the organization is the scenario/system be used ?
- Who will benefit from the use of the scenario/system ?
- Who will supply information to this scenario/system, use information, and remove information ?
- Does one person play several different roles ?
  - May have an actor for each role
- Do several people play the same role ?
  - Only use one actor per role (no matter how many people play that role) e.g. one to many or use generalization of actor (e.g. new or old customer)

Questions to Identify the Other Actors

- What other entity is interested in the scenario/system?
- What other entity will supply information to the scenario/system, use this information, and remove this information?
- Does the system use an External resource ?
- Does the system interact with a legacy (Inheritance Interaction) system ?

Actors

- An Actor is an outside or external part to the system.
- It can be a:
  - Human
  - Peripheral device (hardware)
  - External system or subsystem
  - Time or time-based event
- Represented by:



Use Cases

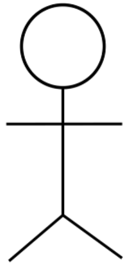
- A **use case** is a summary of scenario for a single task or goal.
- An **actor** is who or what initiates the events involved in the task of the use case.
- Actors are simply the roles that people or the objects play or interact with the system.

Use Case

- So as we read our scenario, what or who is the actor ?

- A patient calls the clinic to make an appointment for a yearly checkup. The receptionist finds the nearest empty time slot in the appointment book and schedules the appointment for that time slot.

As an external part of the system that interact with the system is **Patient**



Use Cases

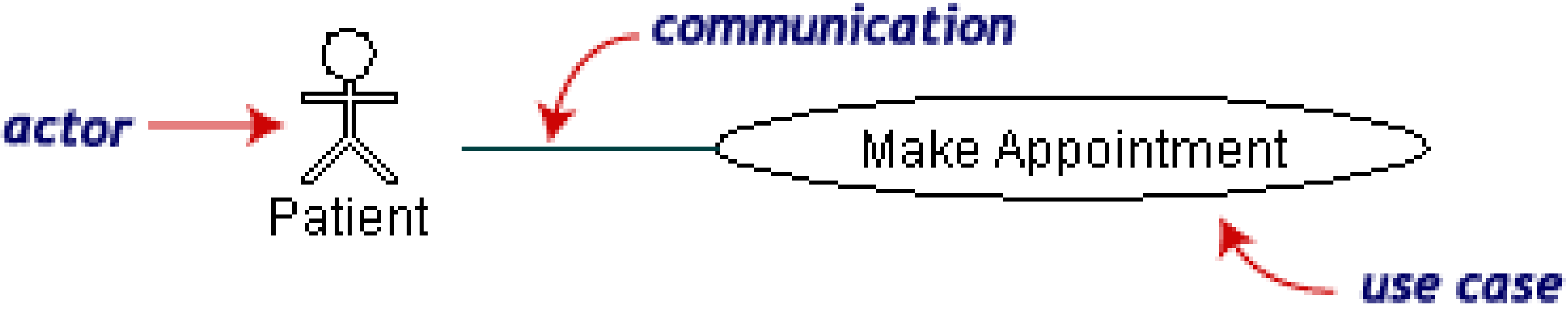
- The **use case** is a summary of scenarios for a single task or goal.

- So What is the Use Case ?

- The Use Case is **Make an Appointment.**
- It is a use case for the medical clinic.

Use Cases

- 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).
- Actors are stick figures
- Use cases are ovals
- Communications are lines that link actors to use cases.



Use Case Components

- The use case has three components.

1. The **use case** task, referred to as the use case that represents a feature needed in a software system.
2. The **actor(s)** who trigger the use case to activate.
3. The **communication** line to show how the actors communicate with the use case

Use Case Diagram- Use Case

- A major process performed by the system that benefits an actor(s) in some way

- Models a dialogue between an actor and the system

- Represents the functionality provided by the system

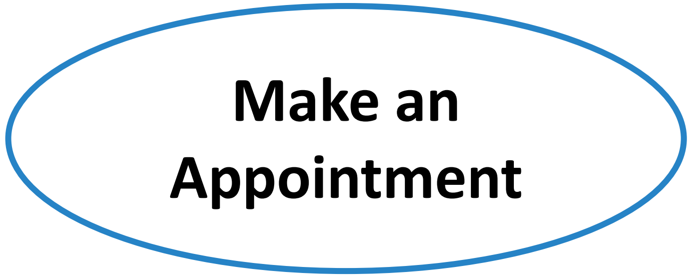
Use Case

- Each use case in a use case diagram describes one and only one function in which users (actors) interact with the system

  - May contain several “paths” that a user can take while interacting with the system
  - Each path is referred to as a scenario

Use Case

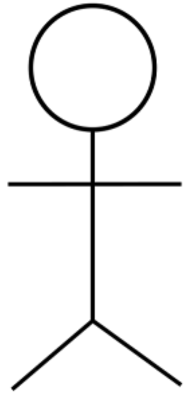
- Labelled using a descriptive verb-noun phrase
- Represented by an oval shape



Make an Appointment

Actor

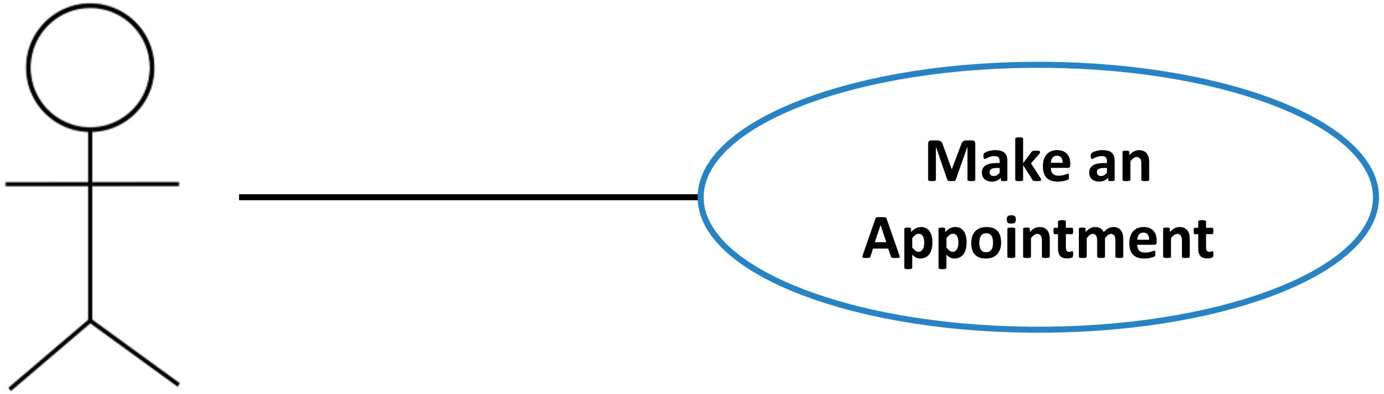
- Labelled using a descriptive noun or phrase
- Represented by a stick character



Patient, Doctor,
Nurse, Staff etc.

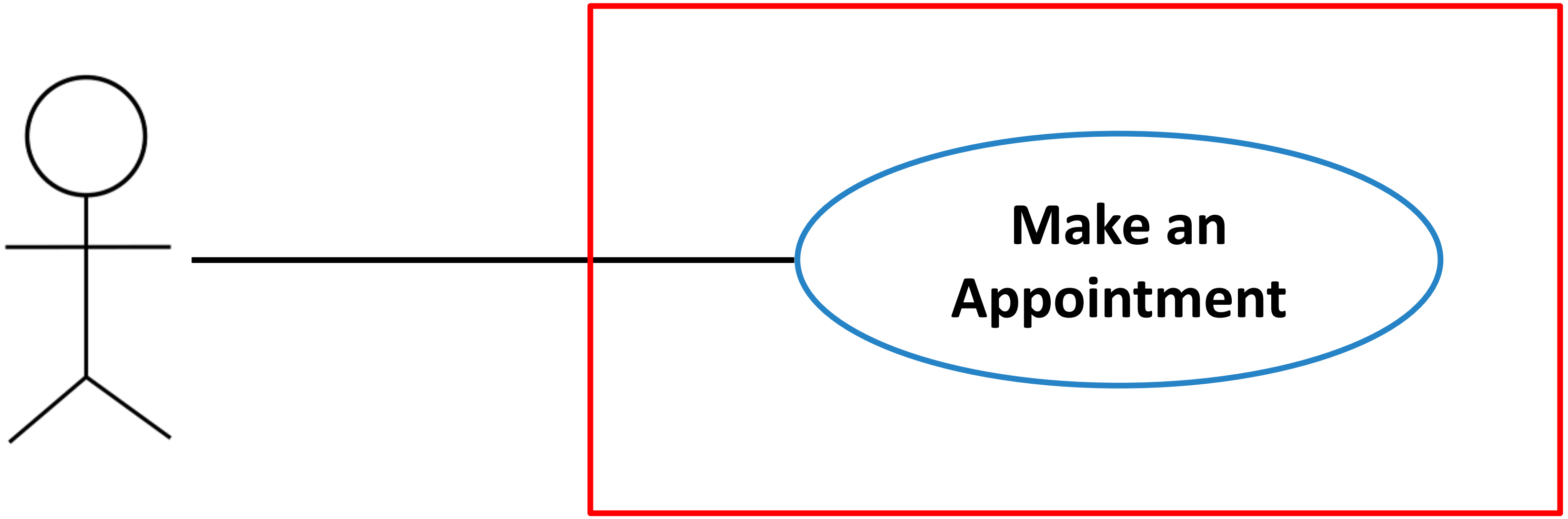
Use Case- Relationships

- Relationships
  - Represent communication between actor and use case
  - Depicted by line or double-headed arrow line
  - Also called association relationship

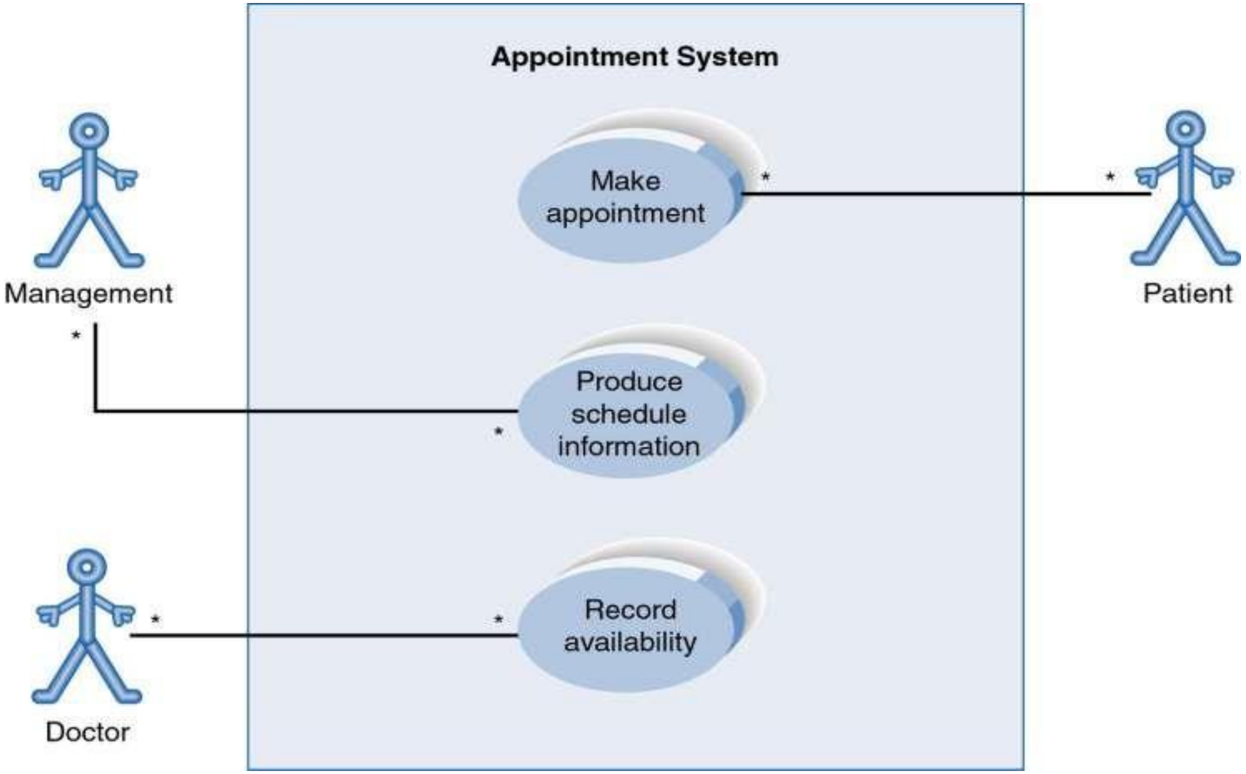


Use Case- Relationships

- Boundary
  - A boundary rectangle is placed around the perimeter of the system to show how the actors communicate with the system.



Use Case Diagram



A use case diagram is a collection of actors, use cases, and their communications.

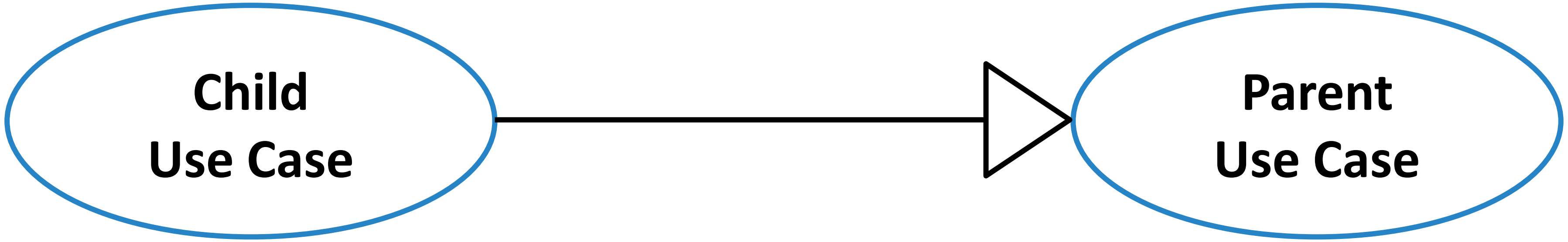
Use Case Diagram- Stereotypes

- UML defines three stereotypes of association between Use Cases:

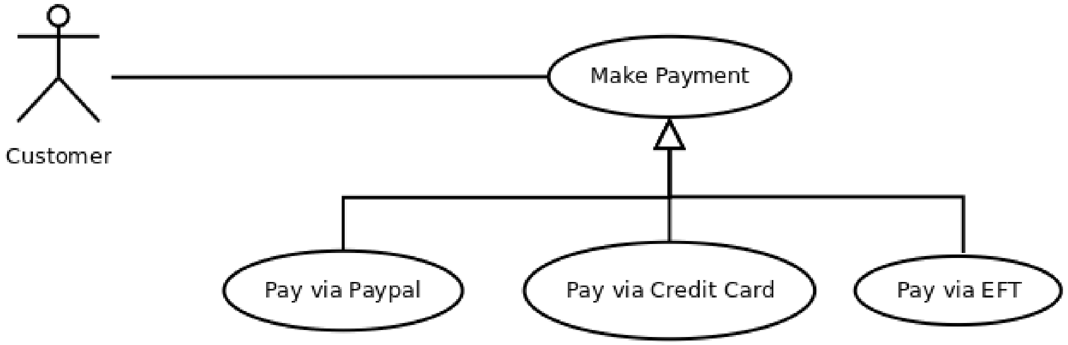
1. Generalization
2. «include»
3. «extend»

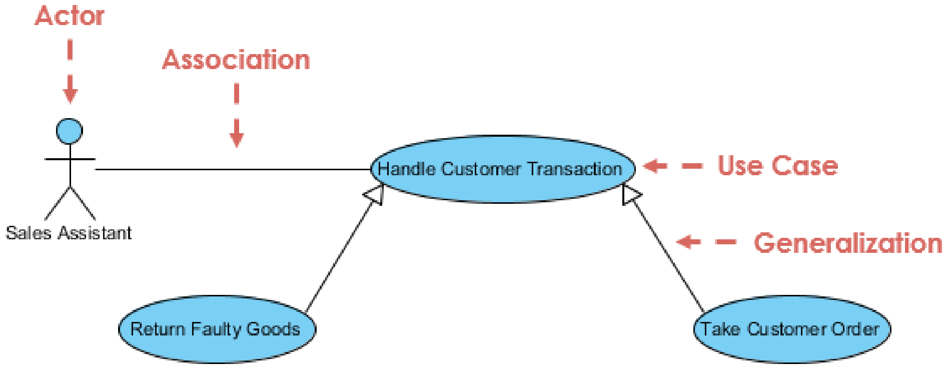
Use Case Diagram- Generalization

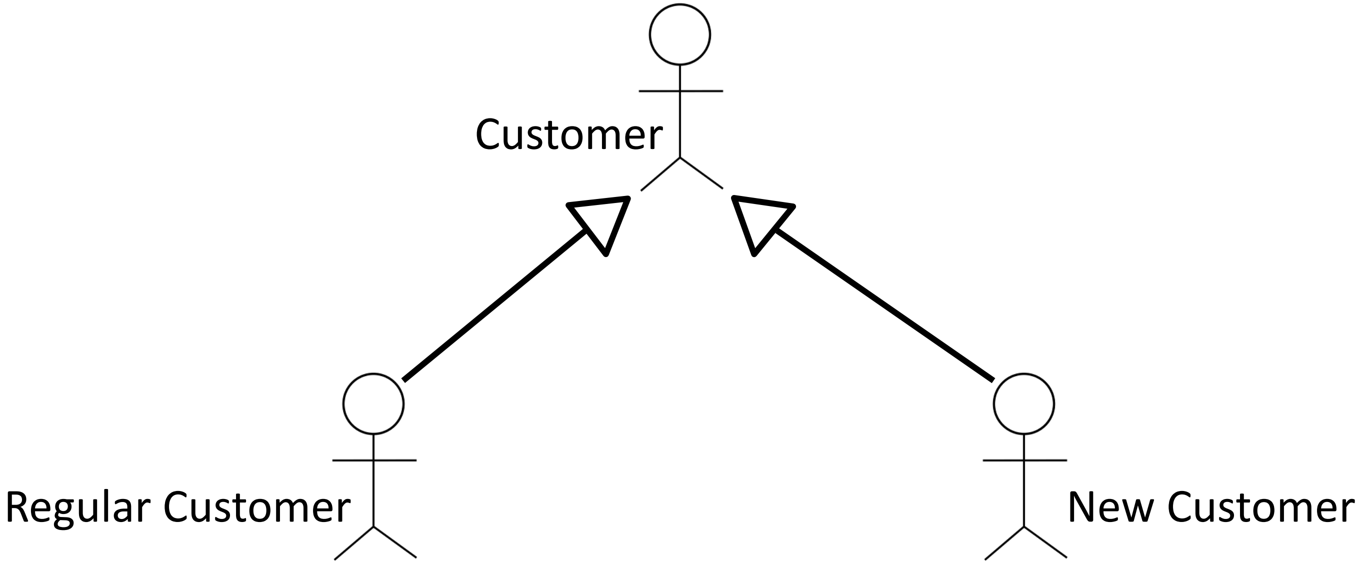
- Generalization Relationship
- Represented by a line and a hollow arrow
- A relationship from a child use case to a parent use case, specifying how a child can specialize all behavior and characteristics described for the parent.



Example of Relationships



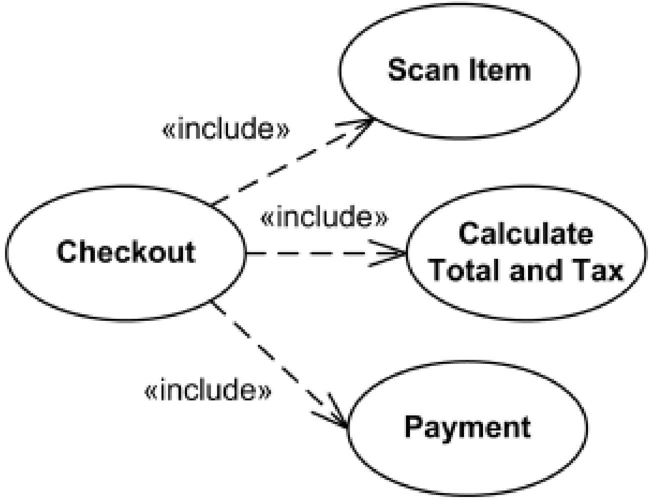


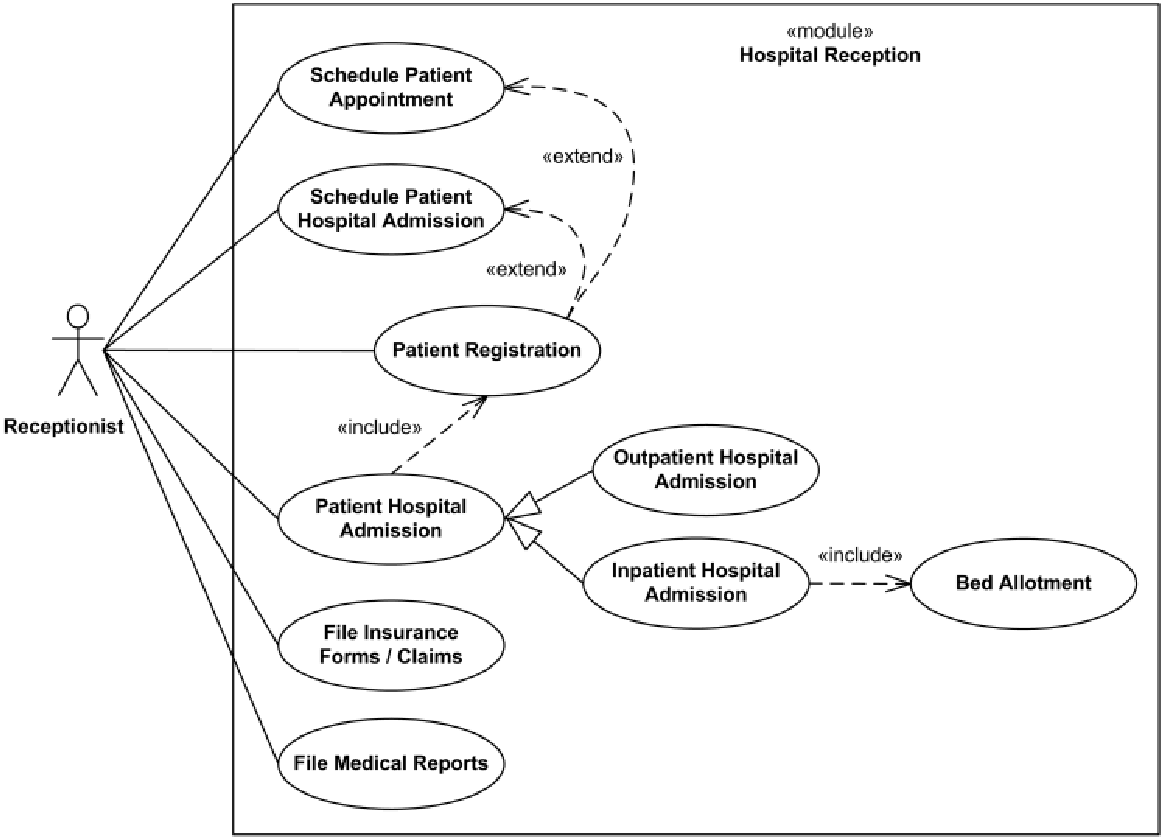


Use Case Diagram- Include

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

Example of Use Case Diagram-Include

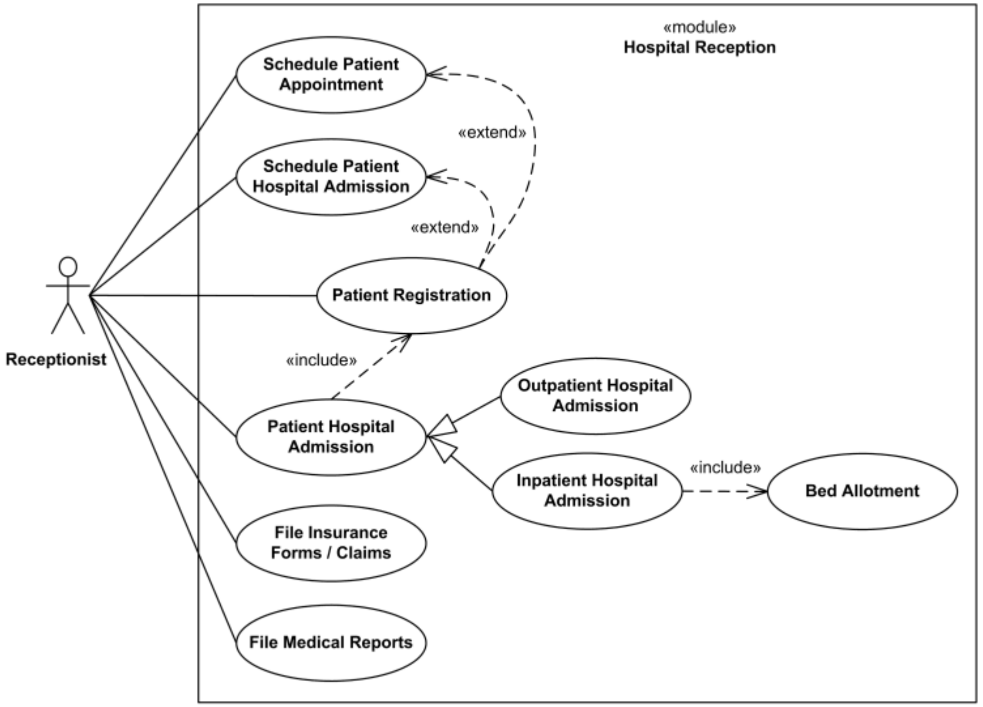


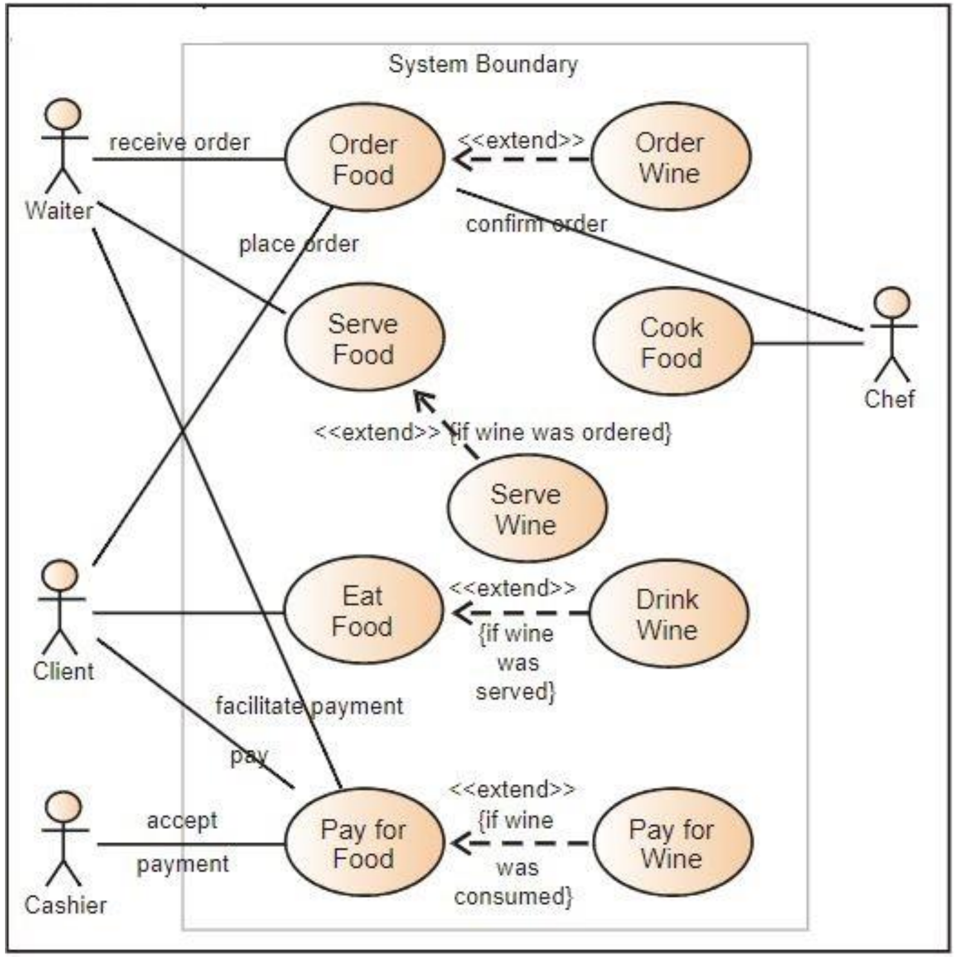

«module»
Hospital Reception
Schedule Patient Appointment
«extend»
Schedule Patient Hospital Admission
«extend»
Patient Registration
Receptionist
«include»
Outpatient Hospital Admission
Patient Hospital Admission
Inpatient Hospital Admission
«include»
Bed Allotment
File Insurance Forms / Claims
File Medical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

Use Case Diagram- Extend

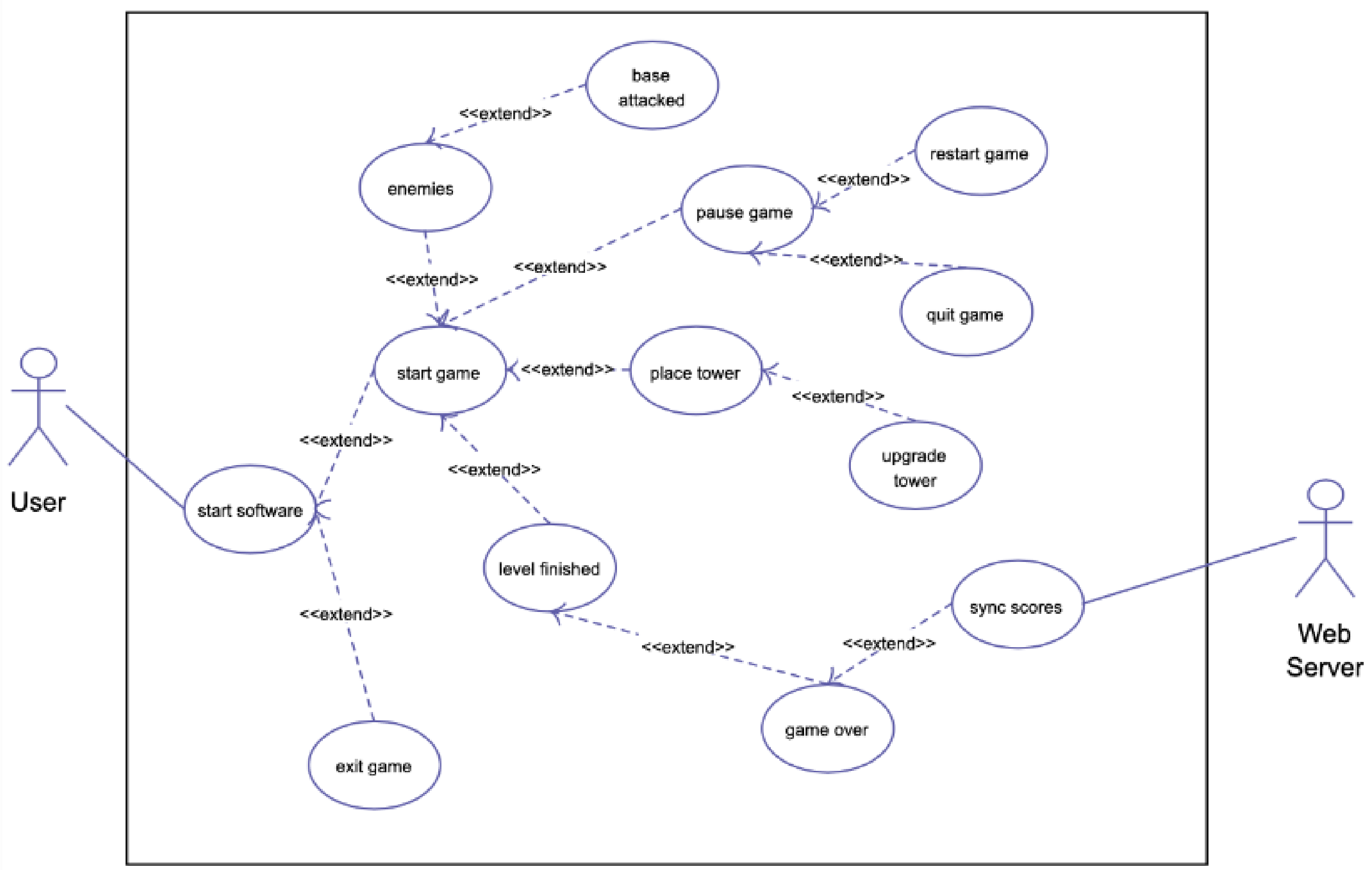
- Extend relationship
  - Represents the extension of the use case to include optional functionality
  - Arrow is drawn from the extension use case to the base use case
  - Write <<extend >> above arrowhead line

Example of Use Case Diagram-Extend

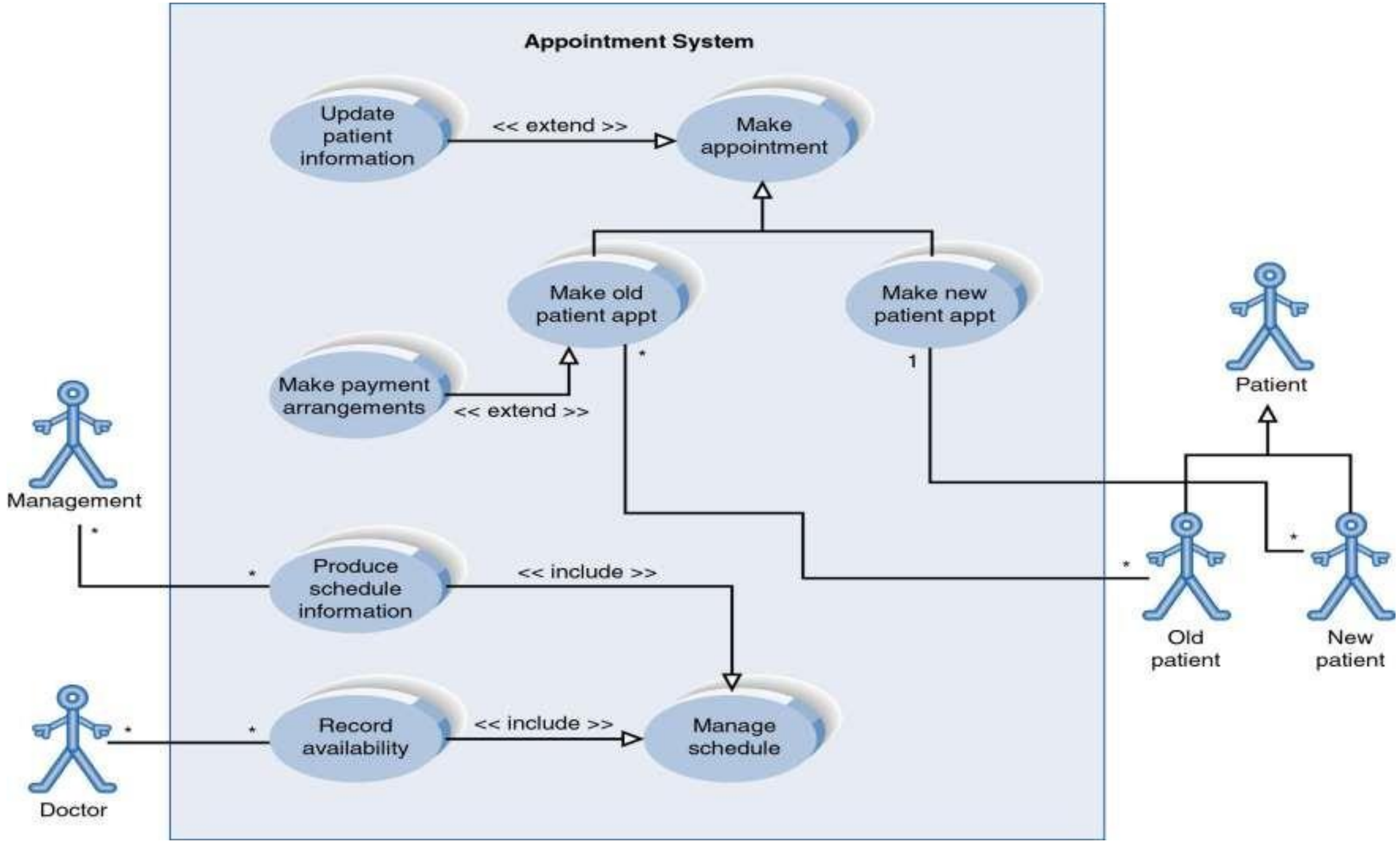

«module»
Hospital Reception
Schedule Patient Appointment
«extend»
Schedule Patient Hospital Admission
«extend»
Patient Registration
Receptionist
«include»
Outpatient Hospital Admission
Patient Hospital Admission
«include»
Inpatient Hospital Admission
Bed Allotment
File Insurance Forms / Claims
File Medical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



Game Example of Use Case Diagram-Extend



Example of Relationships



Benefits of Use Cases

- RUP's primary element in requirements capture
- Described using language of customer (domain language)
- RUP is Use Case Driven
- Easily-understood communication mechanism
- Make traceability of requirements easy.
- Provide summary of what the system should do at an abstract level.
- Easy to describe functional requirements.

Difficulties with Use Cases

- Transition from functional description to object description/class design.
- Makes reuse of class difficult.
  - Since UCs do not talk about classes, developers often wind up doing things their own way, making reuse difficult.
- Stating non-functional requirements are difficult

Case Study- HACS (Homework Assignment & Collection System)

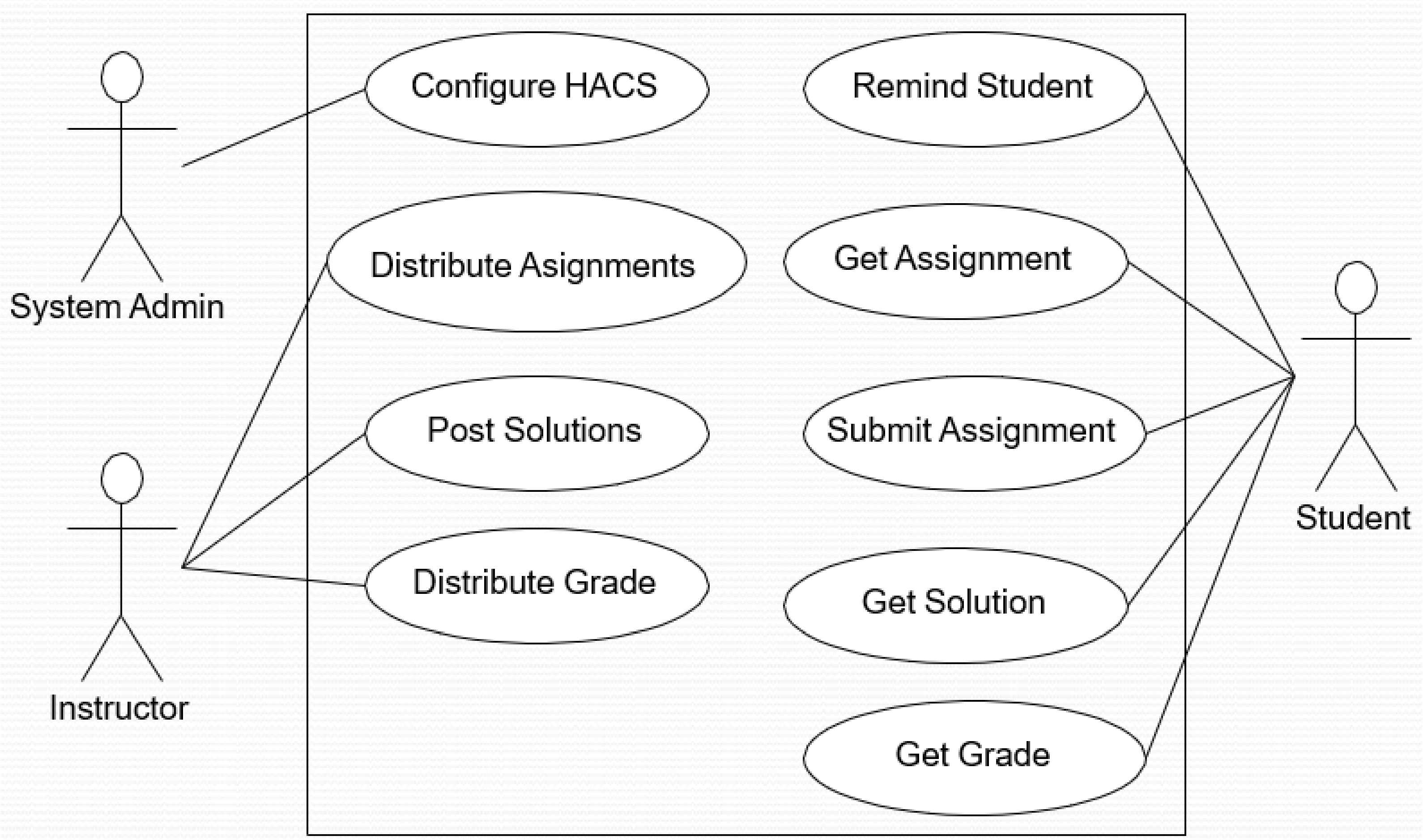
- Homework assignment and collection are an integral part of any educational system. Today, this task is performed manually. What we want the homework assignment distribution and collection system (HACS) to do is to automate this process.

- HACS will be used by the instructor to distribute the

  - Homework assignments, review the students' solutions, distribute suggested solution, and distribute student grades on each assignment.

- HACS shall also help the students by automatically distributing the assignments to the students, provide a facility where the students can submit their solutions, remind the students when an assignment is almost due, remind the students when an assignment is overdue.

HACS (Homework Assignment & Collection System)- Use Case Diagram



HACS- Use Case

Use case: Distribute Assignments

Actors: Instructor (initiator)

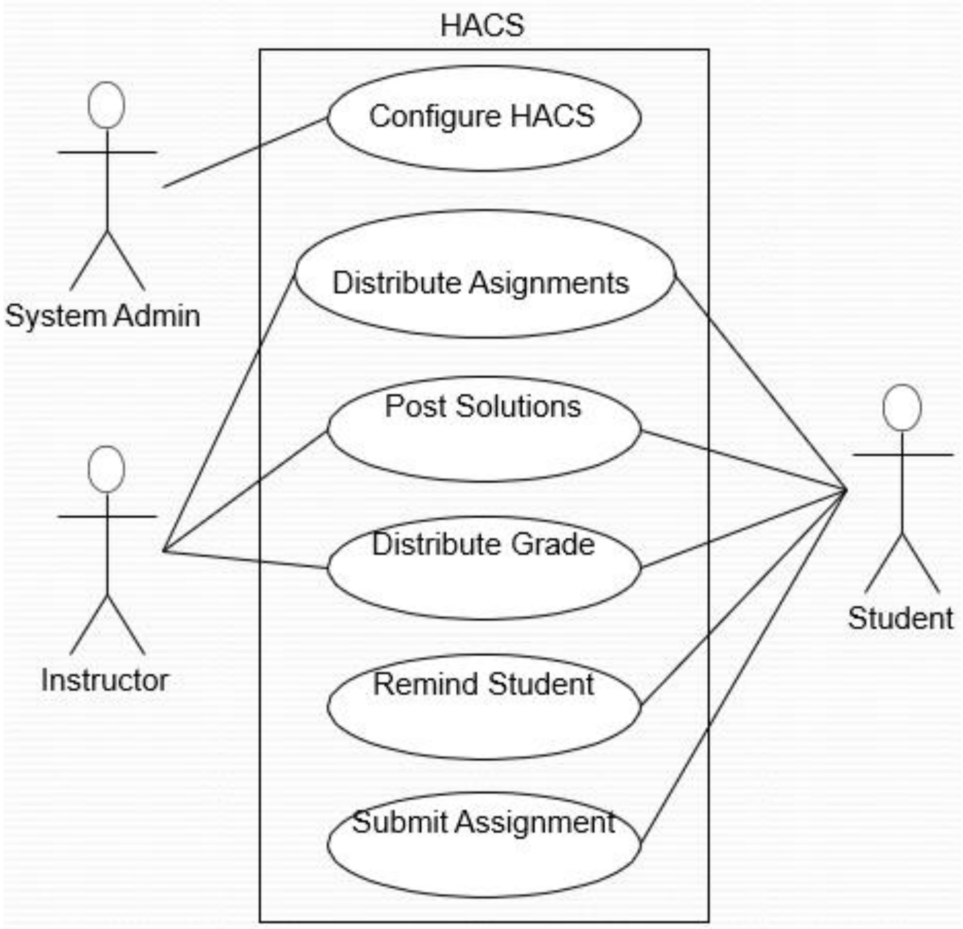
Type: Primary and essential

Description: The Instructor completes an assignment and submits it to the system. The instructor will also submit the due date and the class the assignment is assigned for.

Use-Cases: Configure HACS must be done before any user

(Instructor or Student) can use HACS

Alternate HACS



Alternate HACS- Use Case

Use case: Distribute Assignments

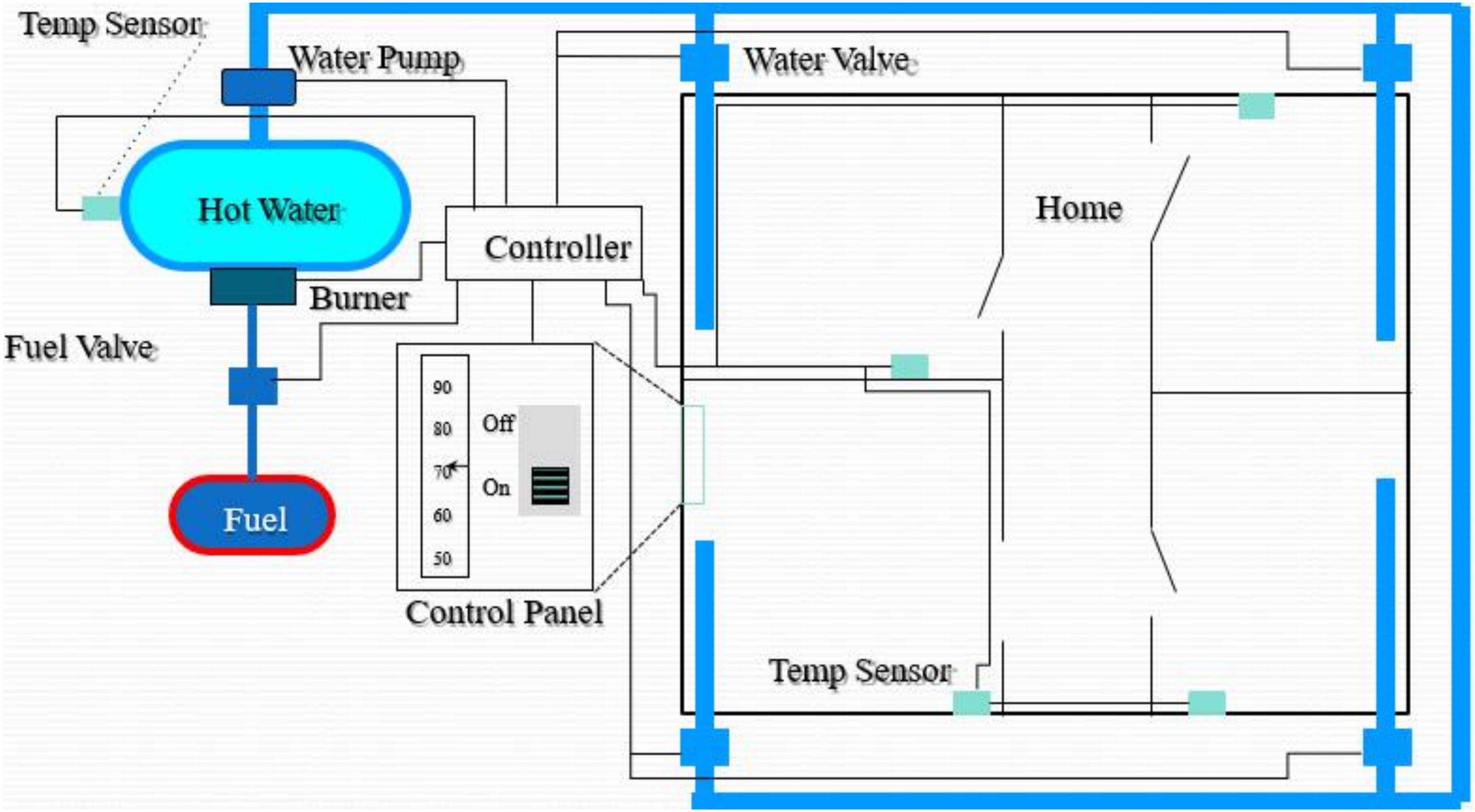
Actors: Instructor (initiator), Student

Type: Primary and essential

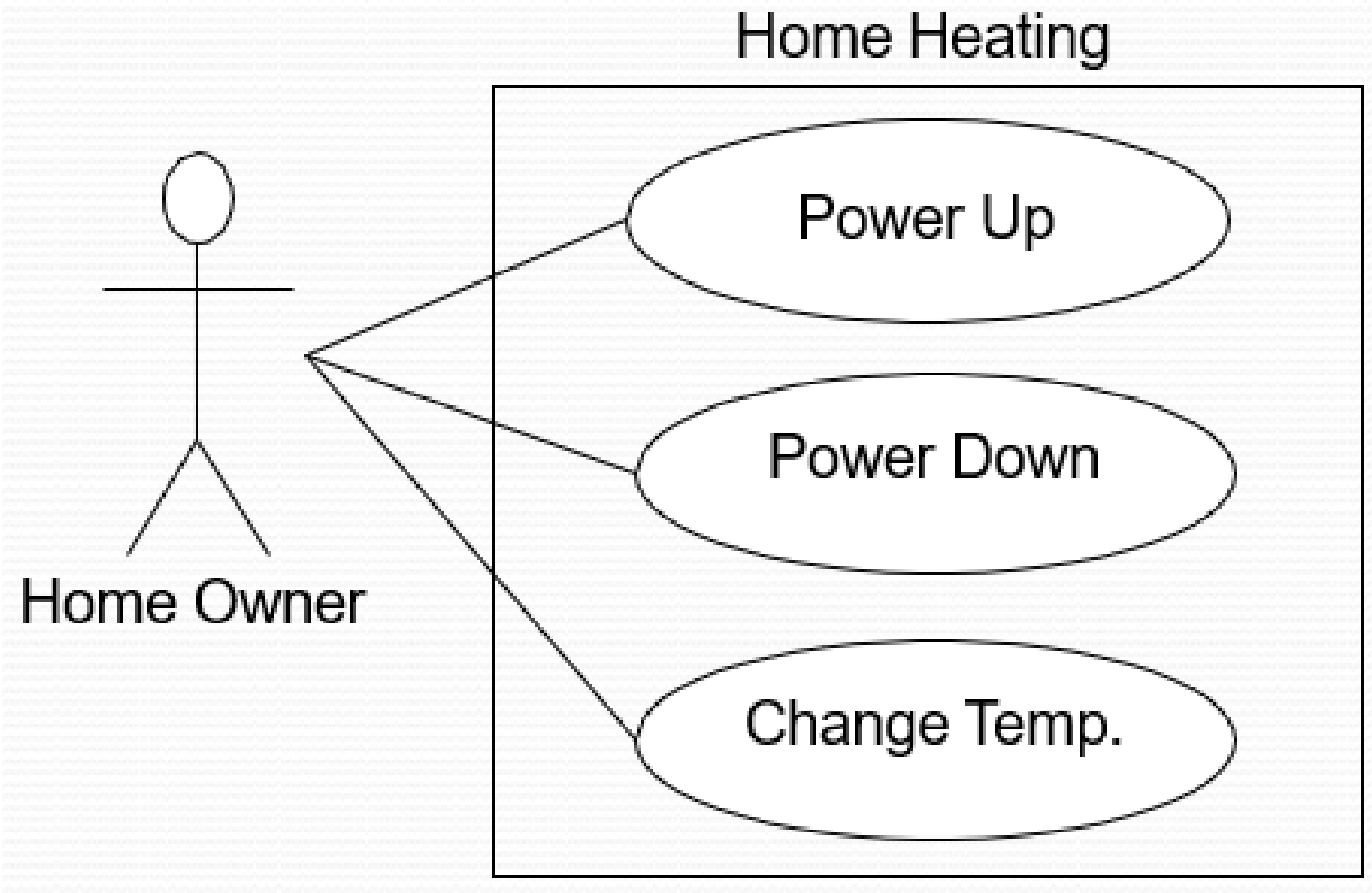
Description: The Instructor completes an assignment and submits it to system. The instructor will also submit delivery date, due date, and class assignment is assigned for. The system will at the due date mail the assignment to student.

Use-Cases: Configure HACS must be done before any user (Instructor or Student) can use HACS

Home Heating System



Home Heating System- Use Case Diagram



Home Heating System- Use Case

Use case: Power Up

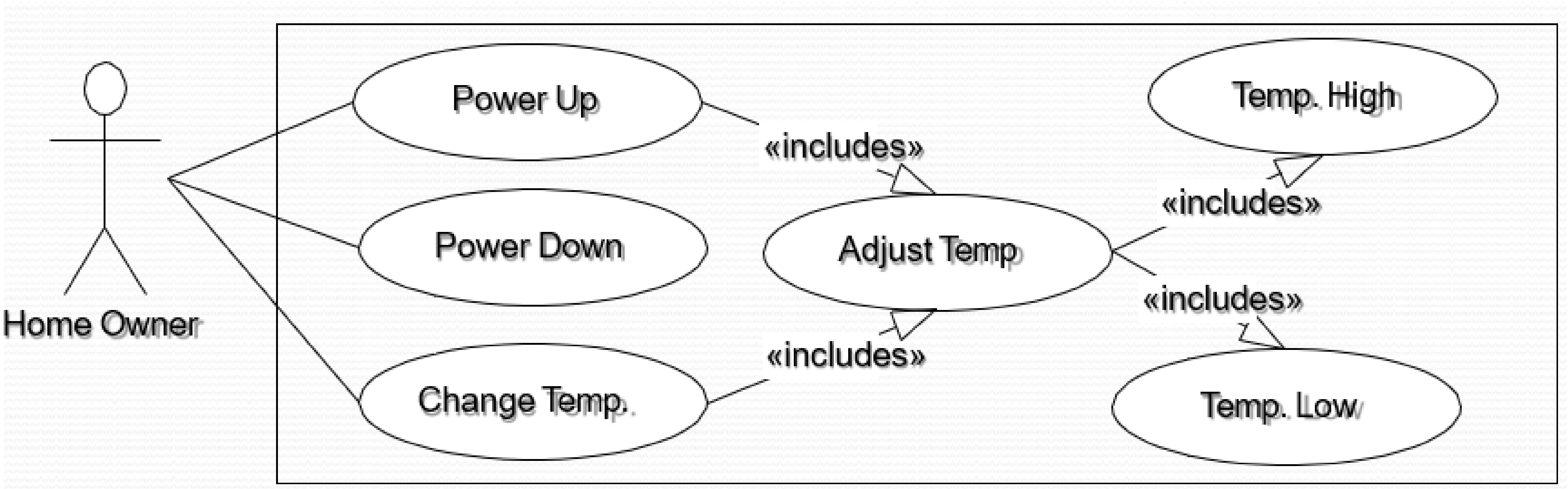
Actors: Home Owner (initiator)

Type: Primary and essential

Description: The Home Owner turns the power on. Each room is temperature checked. If a room is below the desired temperature the valve for the room is opened, the water pump started. If the water temp falls below threshold, the fuel valve is opened, and the burner ignited. If the temperature in all rooms is above the desired temperature, no actions are taken.

Use-Cases: None

Modified Home Heating System



Modified Home Heating System- Use Cases

Use case: Power Up

Actors: Home Owner (initiator)

Type: Primary and essential

Description: The Home Owner turns the power on. Perform Adjust Temp. If the temperature in all rooms is above the desired temperature, no actions are taken

Use-Cases: Perform Adjust Temp

Modified Home Heating System- Use Cases

Use case: Adjust Temp

Actors: System (initiator)

Type: Primary and essential

Description: Check the temperature in each room. For each room:

Below target: **Perform Temp Low**

Above target: **Perform Temp High**

Use-Cases: Temp Low, Temp High

Modified Home Heating System- Use Cases

Use case: Temp Low

Actors: System (initiator)

Type: Secondary and essential

Description: Open room valve, start pump if not started. If water temp falls below threshold, open fuel value and ignite burner.

Use-Cases: None