Object Oriented Software Engineering

UML Diagrams: Use Case Diagrams

Presented By:

T.A. **Asmaa Hamad El-saied**

E-mail: **eng.asmaa134@gmail.com**

Agenda

Use Case Diagram

Case Study

Use Case Diagrams

Use Case Diagram

- **What is the Use Case??**
  - A use case is a methodology used to identify, clarify, and organize system requirements (**functionality**).
- **Functional requirements**
  - Functional requirements are the services that the system should provide, They are product features or functions that developers must implement to enable users to accomplish their tasks. So, it's important to make them clear both for the development team and the stakeholders. Generally, functional requirements describe system behavior under specific conditions.

Use Case Diagram

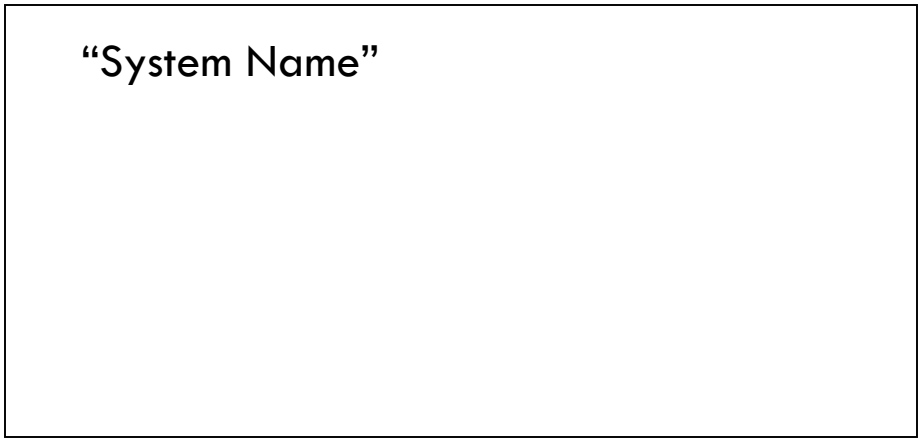
- **Non-functional requirements**
  - Constraints on the services or functions offered by the system such as:
    - Security
    - Reliability
    - Performance
    - Maintainability
    - Availability
    - Scalability
    - Usability

Use Case Diagram (cont.)

- **Use Case Diagram** is a way of visualizing the relationships
  - between actors and use cases (functions)
  - among use cases
- **Use Case Diagram has 4 major elements:**
  - The **system** described
  - The **actors** that the system interacts with
  - The **use-cases**, or services, that the system knows how to perform
  - The **relationships** between the above elements

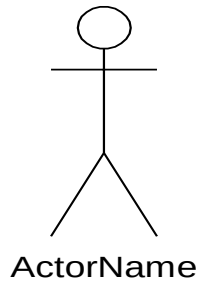
Use Case Diagram :System

- **System boundary:**
  - A boundary box is placed around the perimeter of the **system** to show how the actors communicate with the system.
  - The name of the system appears above or inside the box



Use Case Diagram : Actors

- **Actor** is a someone or something that interacts with the system (exchanges information with the system)
- Example:
  - Policeman – Enters data
  - Supervisor – Allowed to modify/erase data
  - Manager – Allowed to view statistics.
- **An actor** in a Use Case Diagram is represented as



Use Case Diagram : Actors

- **Actors** don't need to be human
  - ✓ May be an external system
  - – Examples:
    - ✓ GPS satellite: Provides the system with GPS coordinates
    - ✓ If actor is a system, in use case diagram it represented by by a rectangle 
- **An actor** has a name that reflects its role

Use Case Diagram :Actors

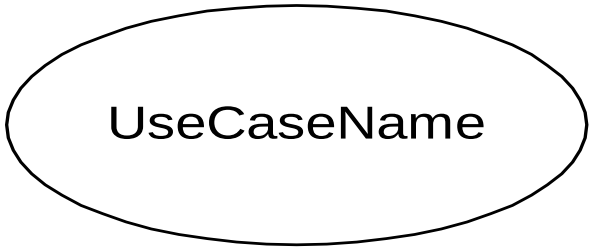
- **Types of actors**
  - **Primary Actors:** The Actor(s) using the system to achieve a goal. The **Use Case** documents the interactions between the system and the actors to achieve the goal of the primary actor.
  - **Secondary Actors:** Actors that the system needs assistance from to achieve the primary actor's goal.
  - A primary actor initiates an interaction with the system.
  - The system initiates interactions with secondary actors.

Use Case Diagram :Actors

- **Types of actors**
  - **For example:**
  - A user clicks the search button on an application's user interface. The application sends SQL query to a database system. The database system responds with a result set. The application formats and displays the result set to the user.
    - In this scenario the user is a primary actor because he initiates the interaction with the system (application).
    - The database system is a secondary actor because the application initiates the interaction by sending SQL query.

Use Case Diagram :Use case

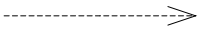
- **A use case** meets an actor's goal
- Use case name is a verb
- **A** use case in a Use Case Diagram is represented as a oval



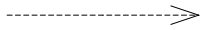
Use Case Description Template

Use Case Name	Name of the use case for the function
Actor(s) Names	The actors who interacting with this use case
Trigger	The main use of this function
Pre-condition	what are the conditions required to act this function
Basic Path (main success scenario)	Provide a bulleted list describe the steps for this function
Alternative Paths	If there is a different steps for this function in special cases
Post-condition	What are the outputs for this function

Use Case Diagram: Relationships

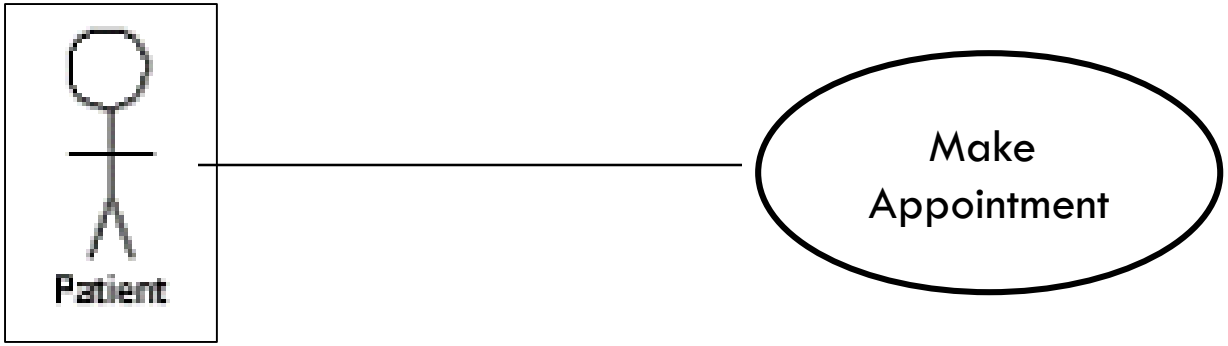
Construct	Description	Syntax
association	The participation of an actor in a use case. i.e., instance of an actor and instances of a use case communicate with each other.	
extend	A relationship from an *extension* use case to a *base* use case, specifying how the behavior for the extension use case can be inserted into the behavior defined for the base use case.	<<extend>> 
generalization	A taxonomic relationship between a more general use case and a more specific use case.	

Use Case Diagrams': Relationships

Construct	Description	Syntax
include	An relationship from a *base* use case to an *inclusion* use case, specifying how the behavior for the inclusion use case is inserted into the behavior defined for the base use case.	<<include>> 

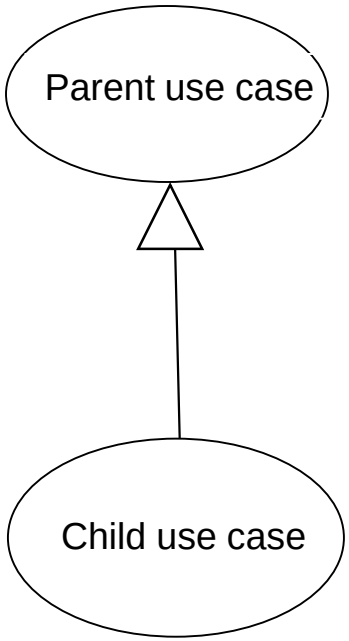
The *<<Association>>* Relationship

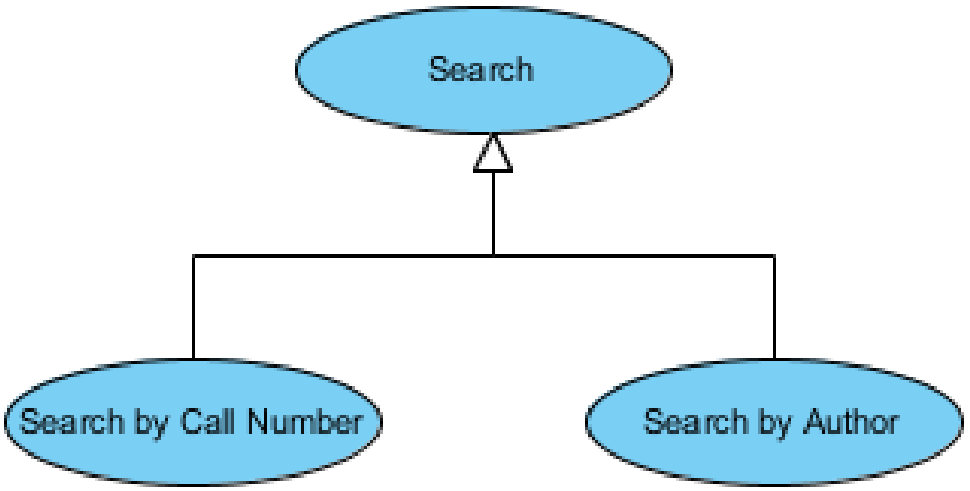
- **Association :** represent communication between actor and use case
- Depicted by line



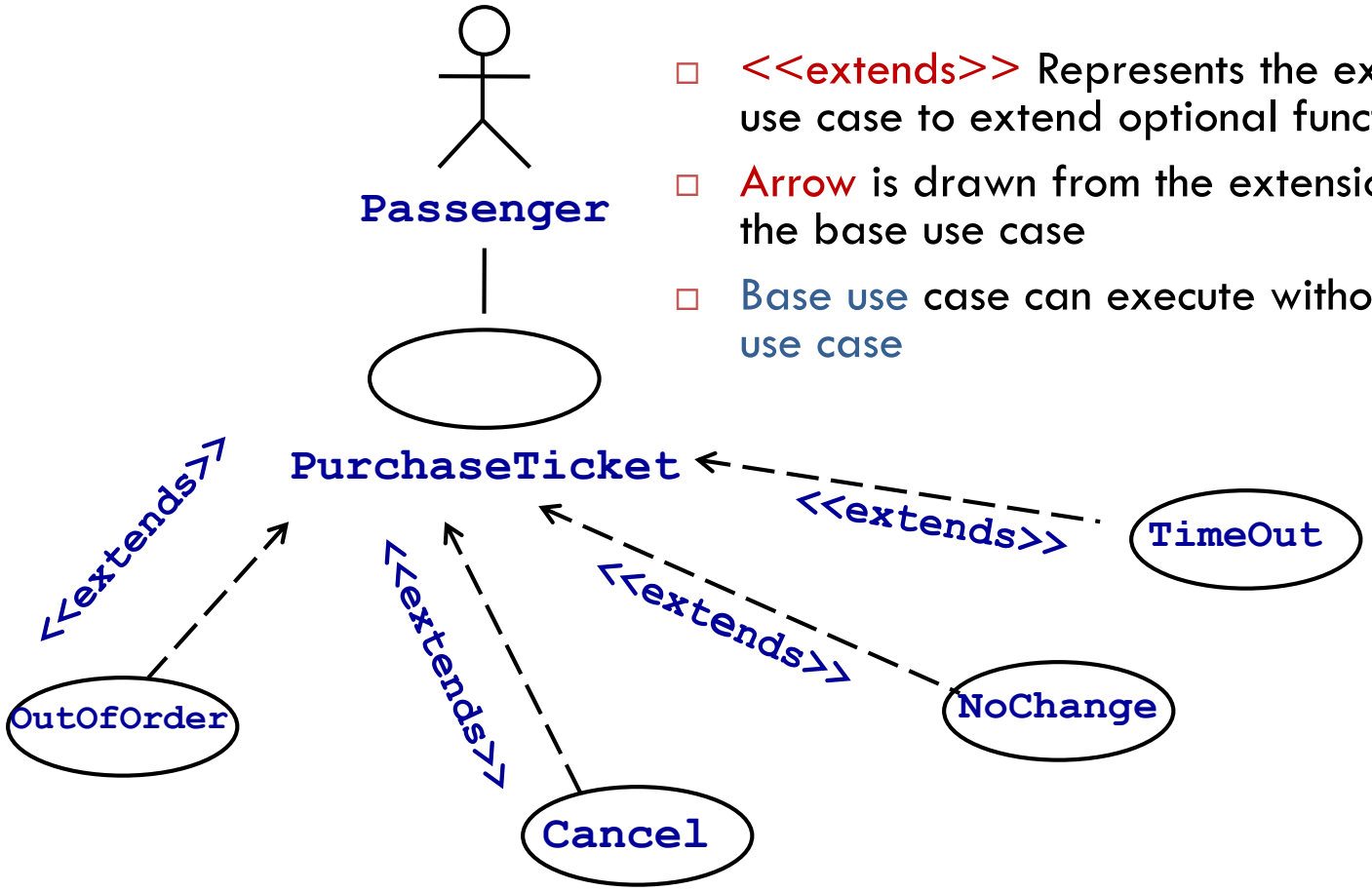
The <<*Generalization*>> Relationship

- Generalization Relationship
  - Represented by a line and a hollow arrow
    - From child to parent
  - The child use case inherits the behavior and meaning of the parent use case.



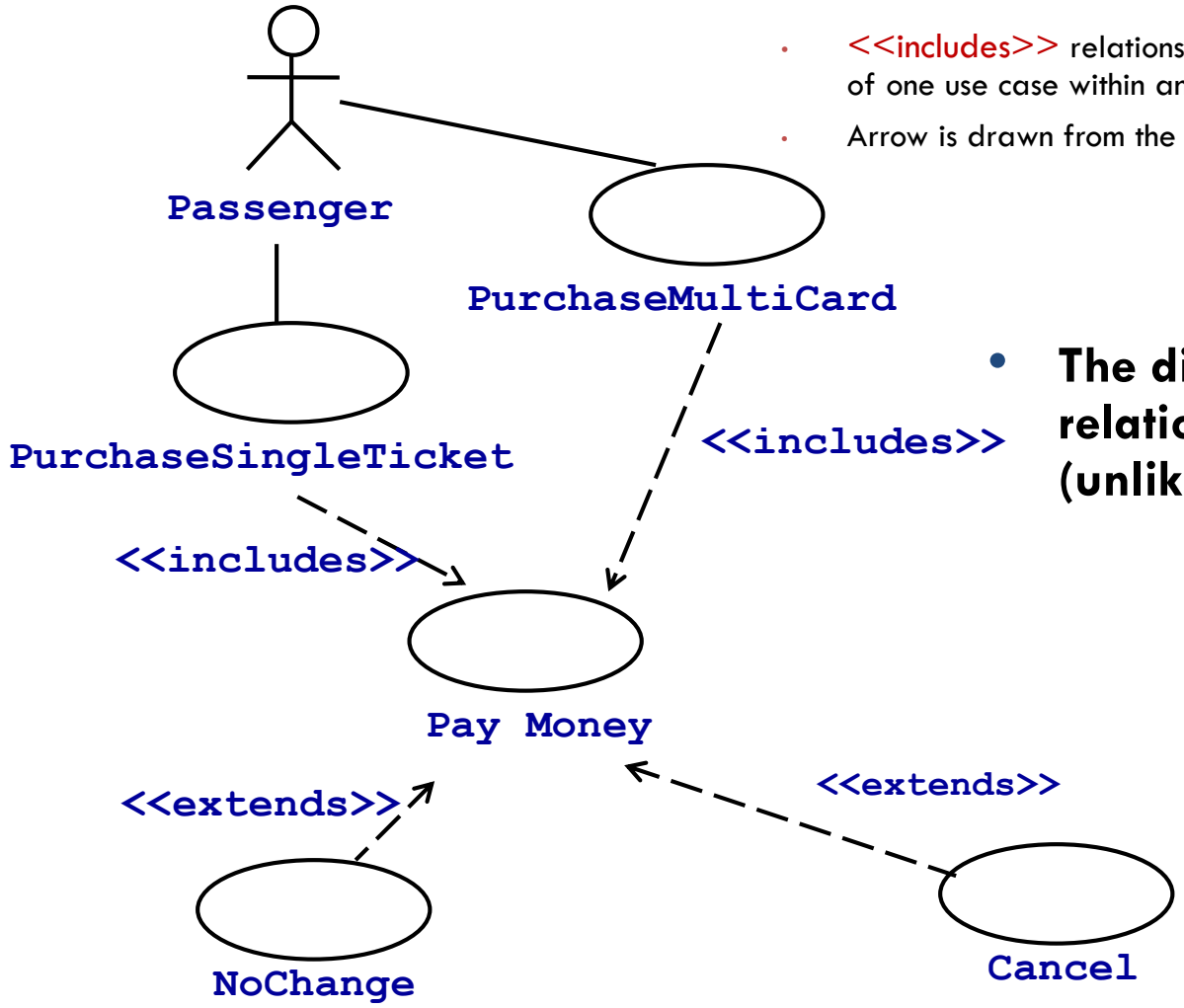


The <<*extends*>> Relationship



- <<extends>> Represents the extension of the use case to extend optional functionality
- Arrow is drawn from the extension use case to the base use case
- Base use case can execute without the extend use case

The *<<includes>>* Relationship



- <<includes>> relationship Represents the inclusion of the functionality of one use case within another
- Arrow is drawn from the base use case to the included use case

- **The direction of an <<includes>> relationship is to the using use case (unlike <<extends>> relationships).**

Case Study

Bank ATM

- Customer can use ATM for different transactions. transactions may be: **check balance, deposit and withdraw money, transfer money** to other accounts. Before ATM takes any action regarding customer transaction, it must authenticate customer first. The bank uses an ATM technician for maintenance and repairing issues. ATM maintenance may include "replenishing" the machine with (money, ink, paper, ..etc) or upgrading its software or even identifying any problems with the machine. Also repairing process may include identifying machine problems.

Bank ATM

Actor	Goal(s)
Customer	1-deposit money, 2-withdraw 3-transfer money 4-check balance
ATM technician	1- maintenance 2-Repair

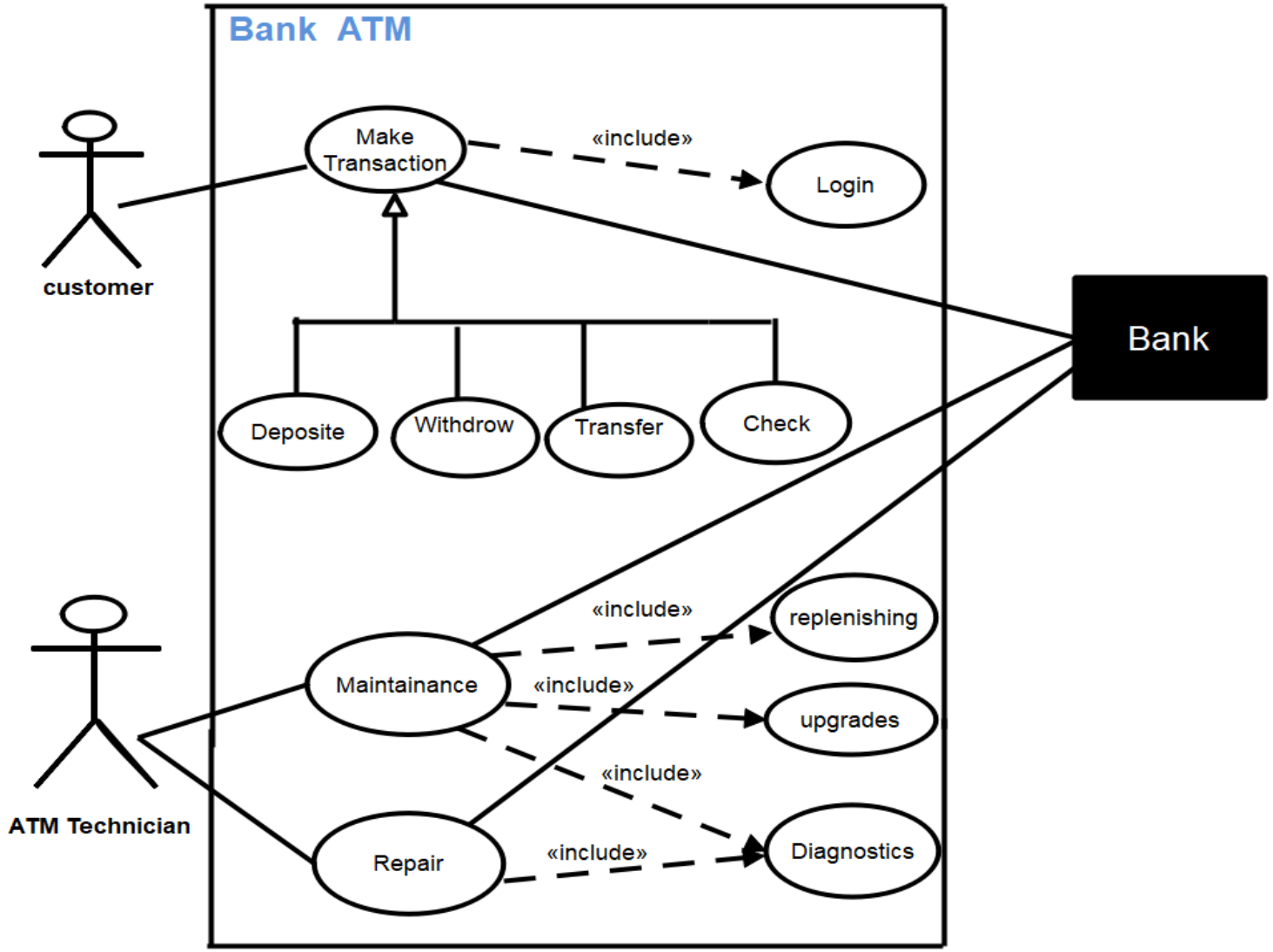
Bank ATM: Withdraw

Use Case Name	Withdraw
Actor(s) Names	Customer.
Trigger	Withdraw money
Pre-condition	Welcome screen is on
Basic Path	1. Customer insert card 2. Machine prompts customer for password 3. Customer enters password 4. System authenticates customer 5. System presents user with a choice menu 6. Customer selects Withdraw Funds option 7. System asks customer to select account 8. Customer selects account 9. System asks customer for amount to withdraw 10.Customer enters amount 11.System dispenses cash and prints receipt 12.System ejects the Customer's **bank card**.

Bank ATM: Withdraw

Alternative Paths	*a. In system failure: 1- Require from the customer to come back at any time later. 4-a) If password is incorrect 1- Go back to step 2 2- If password is incorrect 3 times 2.a) Retain card and notify user 10-a) if customer may not have sufficient funds 1- The system tells the customer that the entered amount is larger than his account balance. 2- The steps 9-10 repeated. 10-b) if machine may not have enough cash 1- The system tells the customer that the entered amount is not available
Post-condition	**Transaction completed and Welcome screen is back on**

Use case diagram



Assignment #3

1- Complete chapter 3 and section 4-1 in chapter 4 in your SRS.

2- Send it.

Thanks