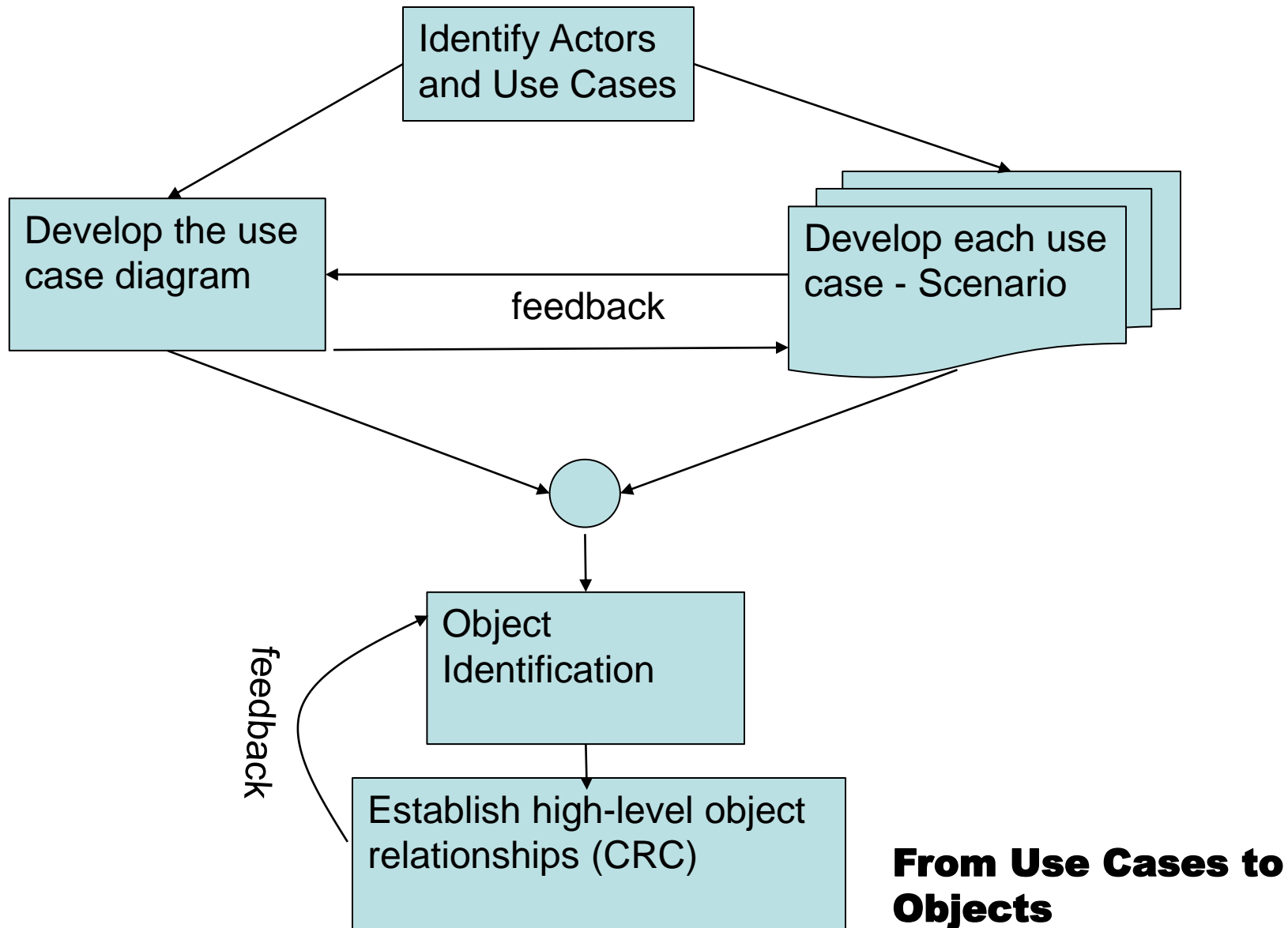


Approach

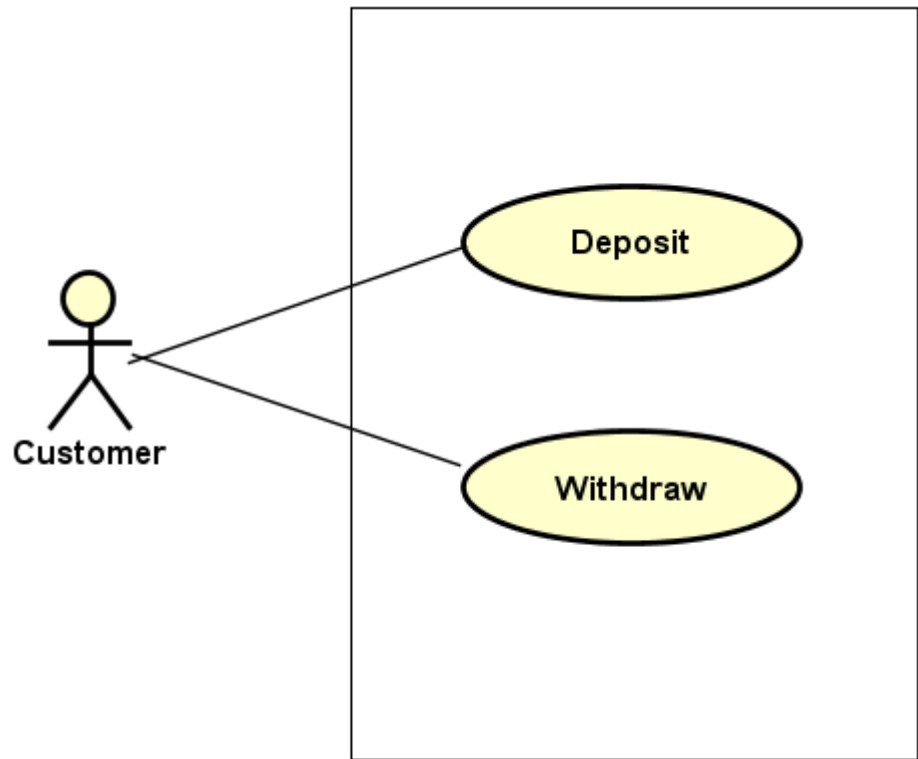


From Use Case Diagram to CRC

Use case diagram represents the high-level dynamic view of the system. It depicts main functionalities, their relationship, and entities (actors) that invoke these functionalities. Use case diagram DOESN'T show how functionalities will be implemented



ATM Menu Interface



**ATM Use Case Diagram:
First Iterations**

Use Case Scenario

For each use case a scenario in which the actor interact with the system need to be written. Sometimes we call it user stories.

Use case name	Main functionality
Description	Describe the purpose of the main functionality
Primary Actor	Main actor for the use case
Basic Flow	Sequence of events and actions that are happening in the use case

Basic Template for a Use Case Scenario

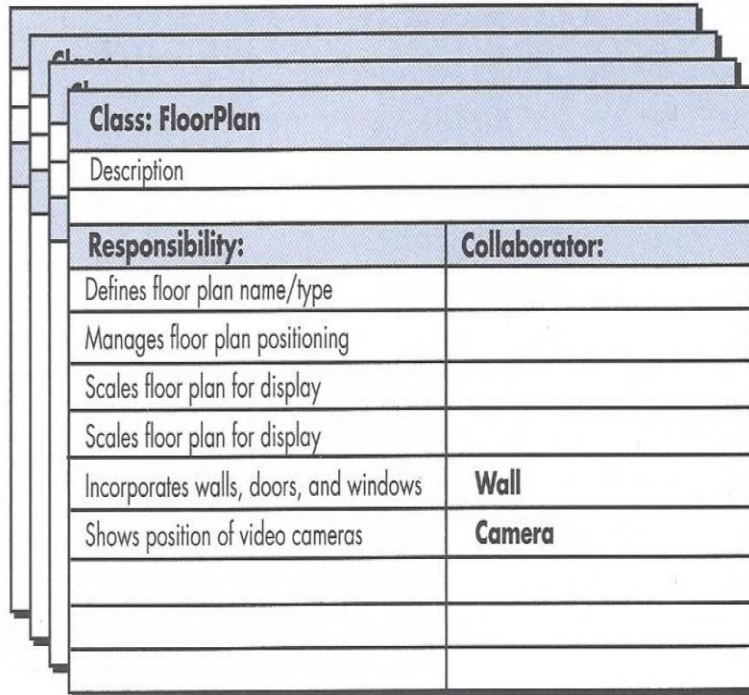
Use Case Scenario : ATM Withdraw Use Case

Use case name	Withdraw Money
Description	Withdraw money from the ATM machine with a bank card
Primary Actor	Bank Customer
Basic Flow	Certain ATM machine is expected to behave as follows: While withdrawing money, first, user enters the card. If the card is valid, then the user will be prompted to enter the PIN. Invalid cards will be returned by the machine. If the user enter an invalid PIN for three times the card will be kept by the ATM machine. Otherwise, the user will be prompted to enter the withdrawal amount. If sufficient balance money will be dispensed and the card will be returned

CRC Diagrams

FIGURE 10.3

A CRC model
index card



Class: FloorPlan	
Description	
Responsibility:	Collaborator:
Defines floor plan name/type	
Manages floor plan positioning	
Scales floor plan for display	
Scales floor plan for display	
Incorporates walls, doors, and windows	Wall
Shows position of video cameras	Camera

CRC diagram Template

Class responsibility collaborator (CRC) diagram is more like a index card set where each index card has the three portions

Class:

Name of the class and a the description of the class indicating main functionality

Responsibility:

List main functionalities (responsibilities) of the class

Collaborator:

Collaborator classes for each functionality (if any).

Simplified ATM System: CRC Example

class: User Menu	
RESPONSIBILITY:	COLLABORATOR:
Insert Card	card reader
deposit withdraw	Transcation
Print Receipt	Print

Simplified ATM CRC Example

Suppose the ATM system has an object called User Menu that is responsible for showing the menu to the user and respond to following key user interactions

Insert Card
Deposit
Withdraw
Print Receipt

Above actions are the **responsibilities** of the class.

Collaborators are other classes that help the User Menu to complete its responsibilities