FARABI UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE



Object-Oriented Analysis and Design

Lecture 4
Class–Responsibility–Collaborator Cards

Dr. Bazargul Matkerim, Professor-Researcher

Learning Outcomes

Describe	Assign	Evaluate
Explain the three parts of a CRC card.	Allocate responsibilities to suitable classes.	Walk through a scenario and refine collaborations.

Learning task: model a bank-machine interaction with CRC cards.

CRC Connects Components and Responsibilities

Class	Responsibility	Collaborator
Which concept owns a coherent role?	What must it know or do?	Whose help does it need to fulfil that role?

CRC cards support discussion before committing to detailed implementation.

The Anatomy of a CRC Card

Class name	
Responsibilities What the class knows What the class does Keep each statement concise	**Collaborators** Other classes that help fulfil these responsibilities

One card represents one candidate class. Revise cards as the design evolves.

Choosing Candidate Classes

Start with the scenario

Read a concrete user goal.
Identify meaningful concepts
and required behaviour.

Evaluate each candidate

Does it have a clear purpose?
Does it own useful information
or behaviour?
Is it inside the system boundary?

Nouns suggest candidates, but not every noun deserves a software class.

Writing Useful Responsibilities

Knowing	Doing
Know the current session state. Know the requested operation. Know the device status.	Coordinate a balance inquiry. Display an outcome. Request account information.

State intent with a short verb phrase. Leave algorithms and field types for later.

Finding Collaborators

Responsibility	Needed help
ATMController coordinates a balance inquiry.	BankGateway requests the balance from the bank.

A collaborator is justified by a responsibility that requires its help.

A Practical CRC Design Workflow

1 Draft

Choose a scenario.
Write candidate cards.

2 Walk through

Assign roles to people.
Follow each request.

3 Refine

Add missing work.
Redistribute overloaded roles.

Repeat with an alternative or failure scenario.
Keep cards easy to move, replace and rewrite.

The value comes from reasoning about collaboration, not filling in a template.

Case Study: A Bank Machine



User goal

An authenticated customer
wants to view an account balance.

Scope of this example

ATM software, device interfaces
and communication with a bank.
Bank internals are outside scope.

This teaching model omits production banking protocols and security detail.

Actors and Software Classes Are Different



Customer: external actor

Inserts a card and chooses
an operation through the ATM.

ATMController: software class

Coordinates the session and
requests help from other classes.

A Customer data class is needed only if the software must represent customer data.

CRC Card: ATMController

ATMController	
Responsibilities	**Collaborators**
Coordinate the customer session Request authentication and balance Choose the next prompt or outcome	CardReader Keypad Display BankGateway

Coordinate the work while delegating device access and bank communication.

CRC Card: BankGateway

BankGateway	
Responsibilities	**Collaborators**
Send authentication requests Request an account balance Translate bank replies and failures	BankConnection

BankConnection handles communication with the external bank service.

Responsibilities of Device Interfaces

CardReader	Keypad	Display
Read card data Return the card Used by ATMController	Capture customer input Report input events Used by ATMController	Show prompts Show results and errors Used by ATMController

Each device interface has a focused role. Lower-level drivers are omitted here.

Walk Through a Balance Inquiry

ATMController	BankGateway	Display
Receive balance choice Request account balance Ask Display to show result	Request balance through BankConnection Return success or failure	Present the balance or an error message

Precondition: authentication succeeded and an account is selected.

Walkthrough: controller requests the balance, receives the outcome, then updates display.

Role-Playing Reveals Missing Responsibilities

Scenario: the bank does not respond

Walkthrough

BankConnection detects a timeout.
BankGateway reports an unavailable result.
ATMController selects the error response.
Display explains that balance is unavailable.

Review questions

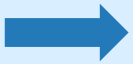
Ask the group:
Who owns each step?
Is any work missing?
Can the session end safely?

A communication failure is not a zero balance. Model it as a distinct outcome.

Refining an Overloaded Class

Overloaded ATM class

Read devices
Manage the session
Communicate with the bank
Format every screen

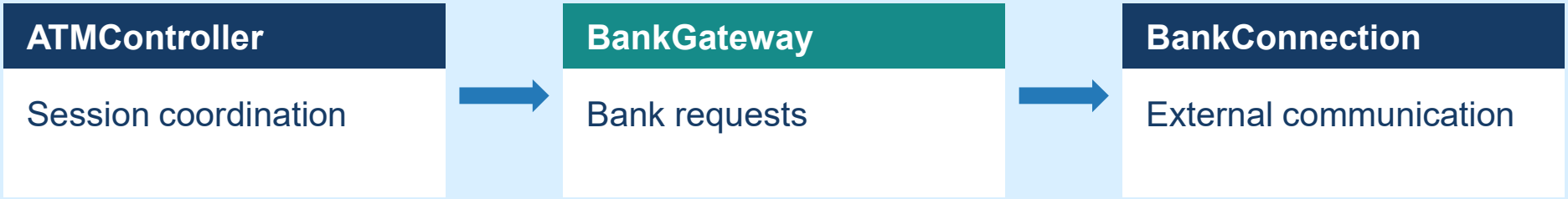


Focused responsibilities

ATMController: coordination
Device interfaces: device access
BankGateway: bank requests
Display: presentation

Split responsibilities when one class has several unrelated reasons to change.

Organizing Collaborating Classes



Device interfaces

CardReader, Keypad and Display support the controller.

Place collaborating cards near each other. Proximity is not a UML relationship type.

From CRC Cards to Detailed Design

CRC statement	Detailed design questions
Request an account balance. Collaborator: BankGateway	What operation is called? What input and result types are used? How is failure represented? What is the interaction order?

Use class and sequence diagrams to add detail after responsibilities are clear.

Practice: Model a Cash Withdrawal

Scenario: an authenticated customer requests a cash withdrawal.

Create or refine cards

ATMController
BankGateway
CashDispenser
Display

Walk through alternatives

The bank declines the request.
The ATM has insufficient cash.
Dispensing fails after authorization.
Identify any additional responsibilities.

State assumptions and record unresolved recovery questions; do not assume success.

Review and Key Ideas

A class owns a coherent set of responsibilities.
Responsibilities describe what it knows and does.
Collaborators provide help needed to fulfil those responsibilities.
Scenario walkthroughs expose missing or misplaced work.

Exit question: how can you tell that a CRC card has too many responsibilities?