(Producing Collaboration/Communication Diagrams)

Week 18

Collaboration/Communication Diagram

- Collaboration diagrams are used to show how objects interact to perform the behavior of a particular use case, or a part of a use case.
- A collaboration diagram describes a pattern of interaction among objects; it shows the objects participating in the interaction by their links to each other and the messages that they send to each other.
- Along with sequence diagrams, collaborations are used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use case. They are the primary source of information used to determining class responsibilities and interfaces.

Sequence vs Collaboration Diagram

- Sequence diagram and collaboration diagram show the same information, but simply present it differently

- Sequence diagrams visualize the sequence of messages that is used to perform a specific functionality while collaborative diagrams visualize the organization of the object and their interactions

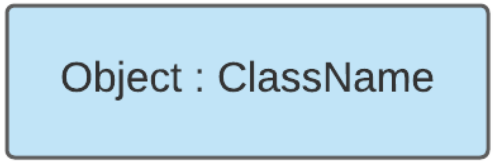
You should remember that a single figure isn’t sufficient to depict most scenarios. Hence, the need of both diagrams

Collaboration Diagram - Notations

- Object
- Actors
- Link
- Message

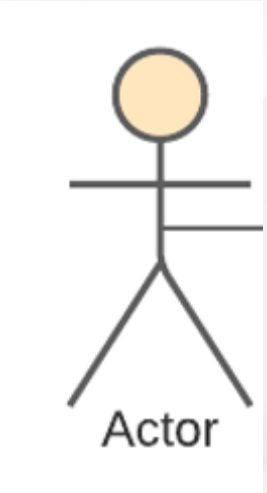
Objects

- Represented as a rectangle with a naming label
- Naming label is written as
  - objectName : className



Actor

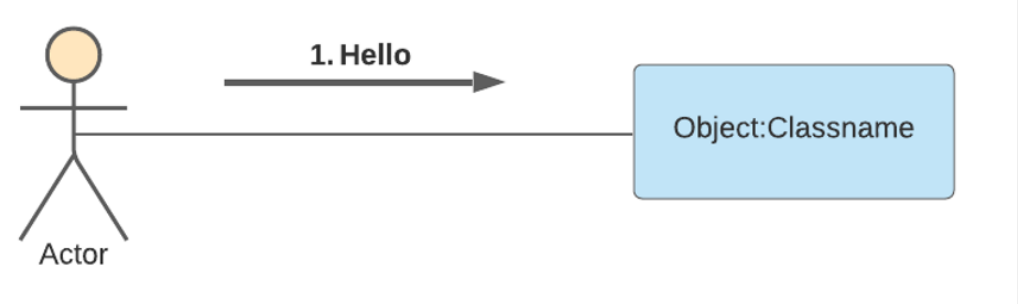
- Represented as a stick figure
- Invokes the interaction
- Has a unique name and role



Links

- Connecting lines
- Drawn between actors or objects, which symbolizes the ability to send messages
- Single link enough to support one or more messages

Messages



- Arrow pointing from the client to the supplier object
- One or more message associated
- Messaged composed of text prefixed by a sequence number

Producing a Collaboration Diagram

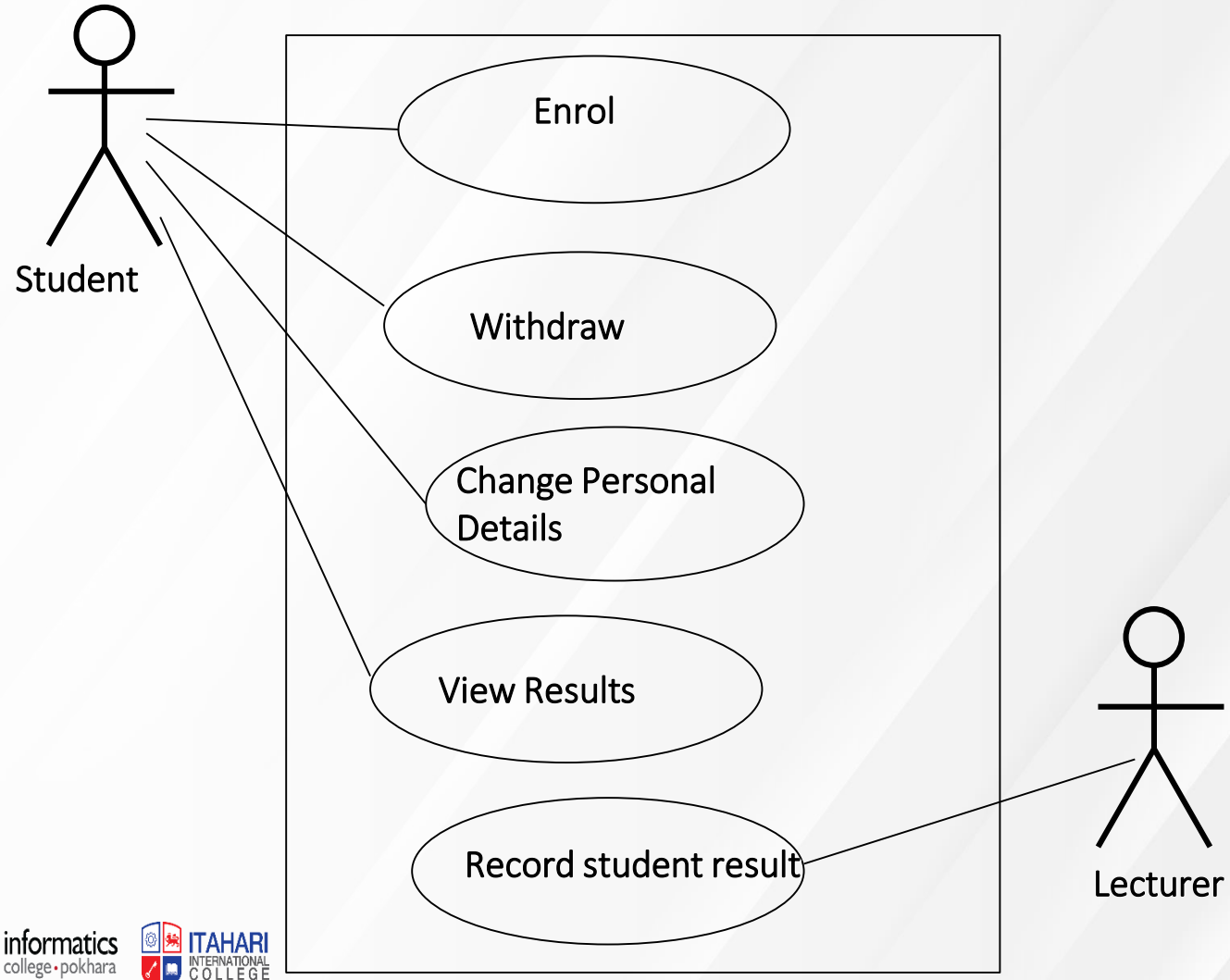
- Read through a Use Case Description and select the Domain Classes

- Consider how Objects of these Domain Classes contribute to and are affected by the Use Case process.

  - Are any objects created/modified/deleted?

  - Does the process obtain any data from any of these objects?

College Student Records System - Required



Picking out the Classes/Objects in a Use Case Description

Consider this Use Case Description. Find all the nouns in the 'Typical Course of Events' description and list those which are important to the processing. Would the system have to process data about the type of person or object named? If so then this is probably a relevant class of objects.

Use Case: Enrol Student
Actors: New Student (initiator)
Description: A new student provides personal details and his/her choice of course to the system. He/she receives confirmation of enrolment.

Typical Course of Events:

Actor Action	System Response
1. The new Student provides personal details for enrolment	2. Presents course options
3 Chooses course	4. provides confirmation of enrolment.

Possible classes for the Enrol Student Use Case:

Student

Course

Enrolment

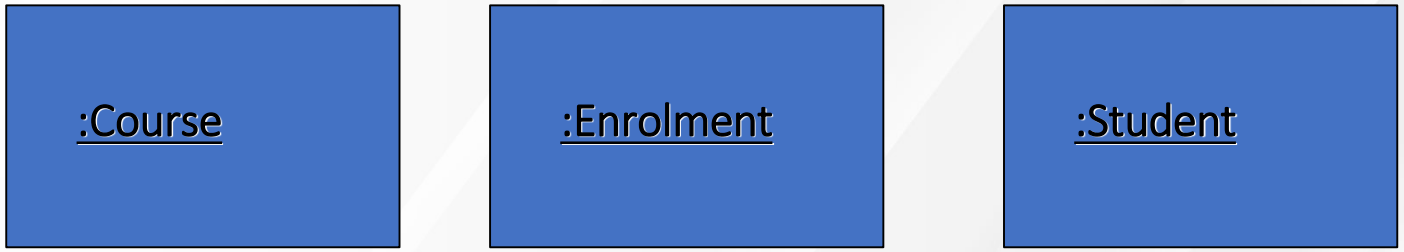
How are these involved in the Enrol Student process?

How are Objects of these Classes affected by the Enrol Student Use Case process?

Consider what might happens when a new student enrols

- An **Object** of Class **STUDENT** will be created (to store the student's personal details)
- One of more **Objects** of Class **COURSE** will be read (so that the list of courses can be presented to the student so that he/she can make a choice)
- One of more **Objects** of Class **ENROLMENT** will be created (to store information showing which student has enrolled on which course, and perhaps date, etc.)

Draw an Object symbol for each of the Domain Class Objects



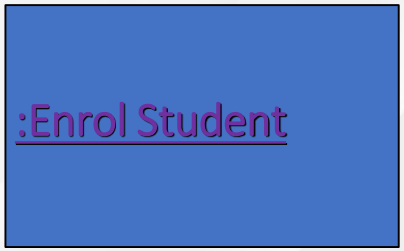
Other types of Class/Object

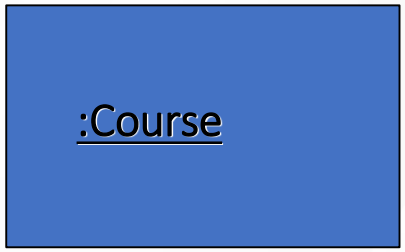
- To manage the Collaboration of Objects which gives effect to the Use Case process – we might have a Control object

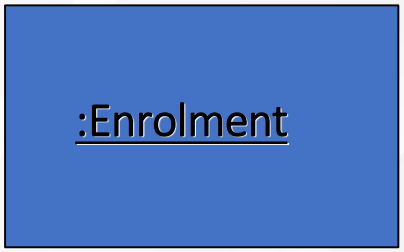
- To manage the screen interaction for the Use Case – we might have a Boundary object

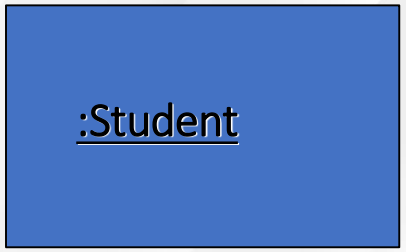
- Would you describe a Control class or a Boundary class as a Domain class? Explain your answer.

Add a Control Object. This will have the same name as the Use Case

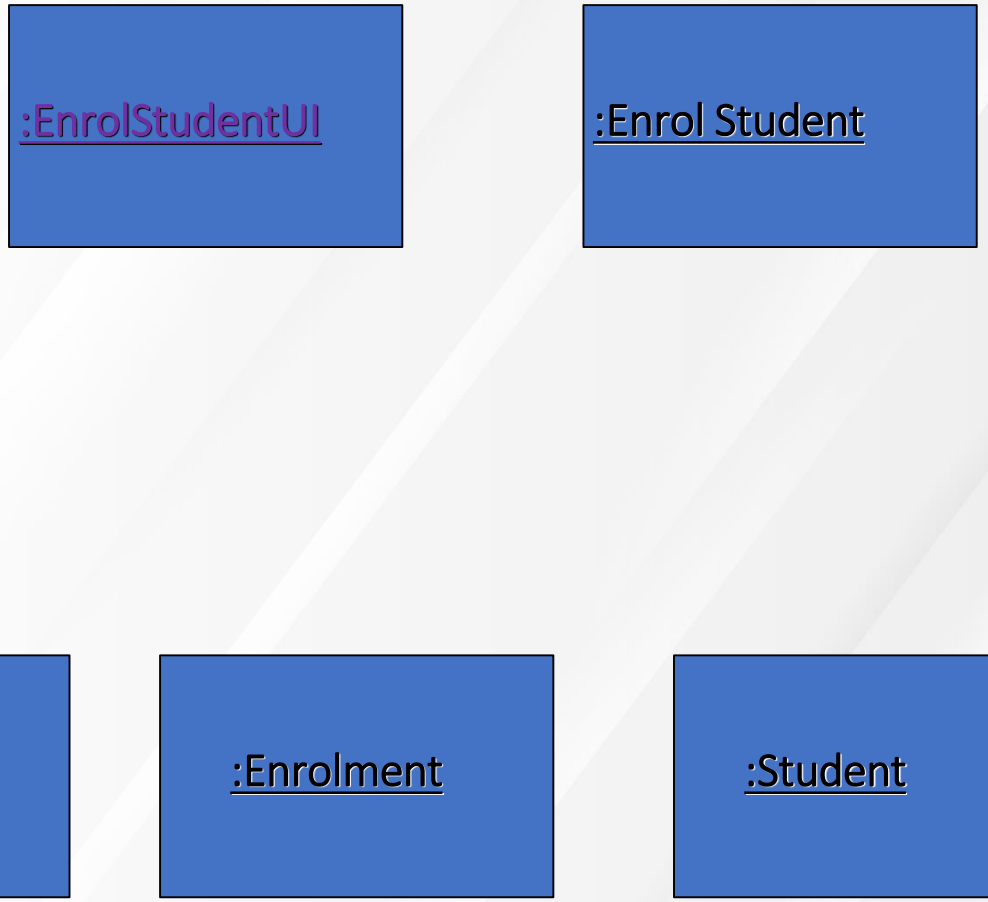




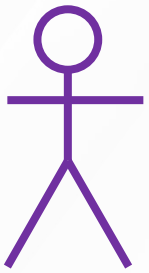




Add a Boundary Object. This will have the same name as the Use Case with the addition of UI (for User Interface)



Add the Actor beside the Boundary object



Student

:EnrolStudentUI

:Enrol Student

:Course

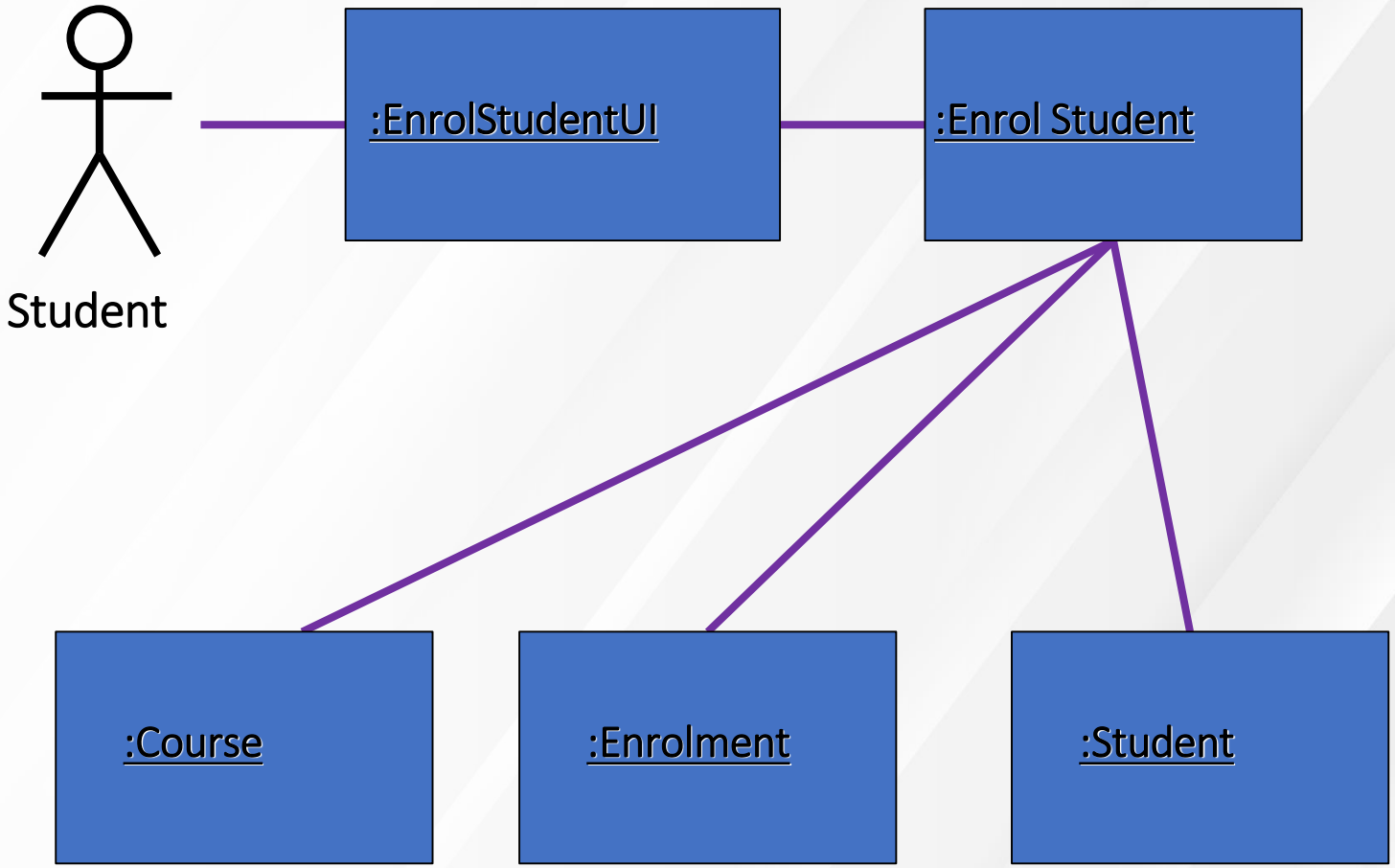
:Enrolment

:Student

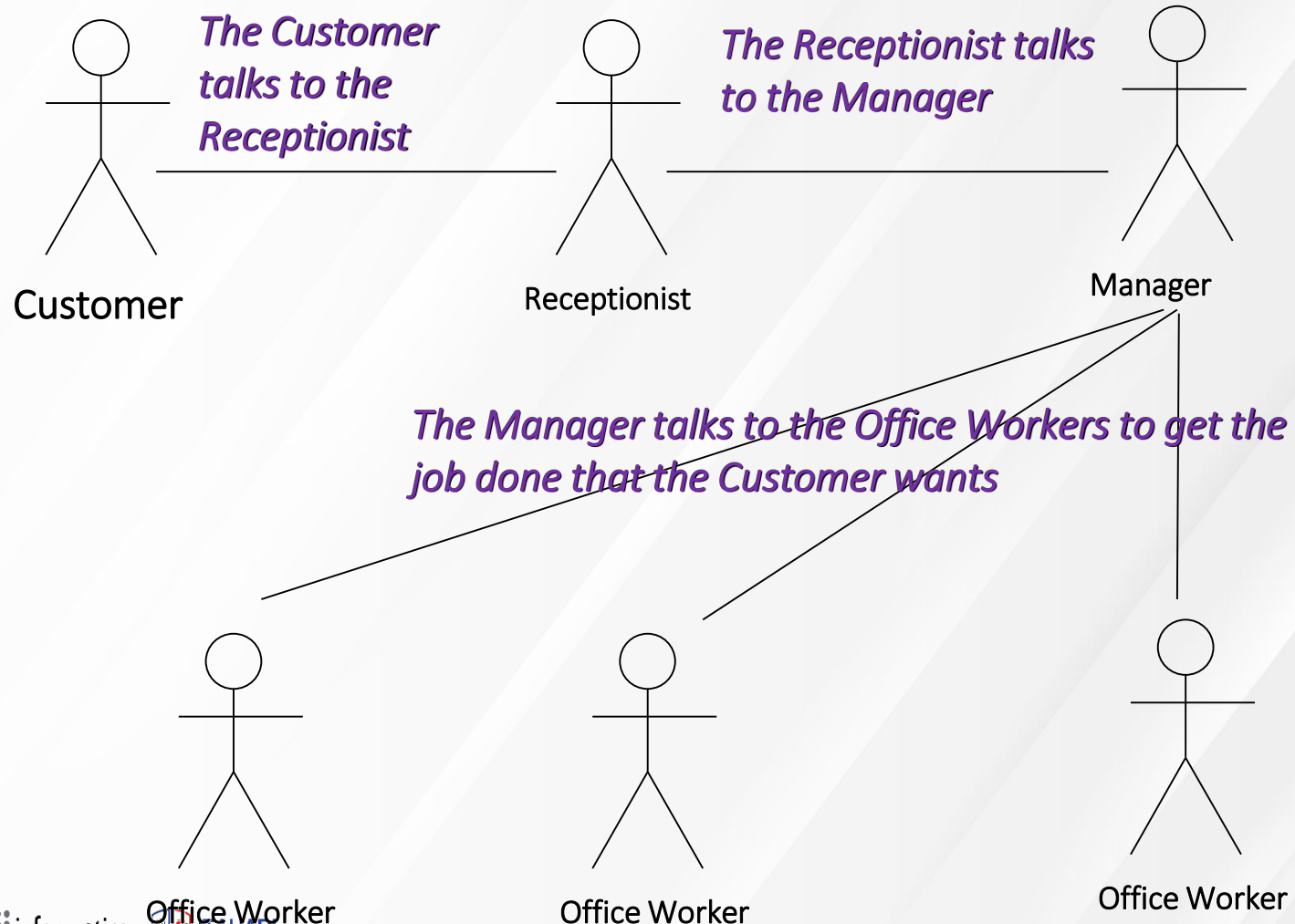
Now add Associations

- Draw a line connecting any pair of Objects which need to 'know' about each other – in order to be able to send a message (perhaps to give an instruction or get information back).
- There are various ways to organize the collaboration. In this case, let's arrange that the Control Object manages all the other objects – so it needs an association with all the others.

Add lines to show connections between objects



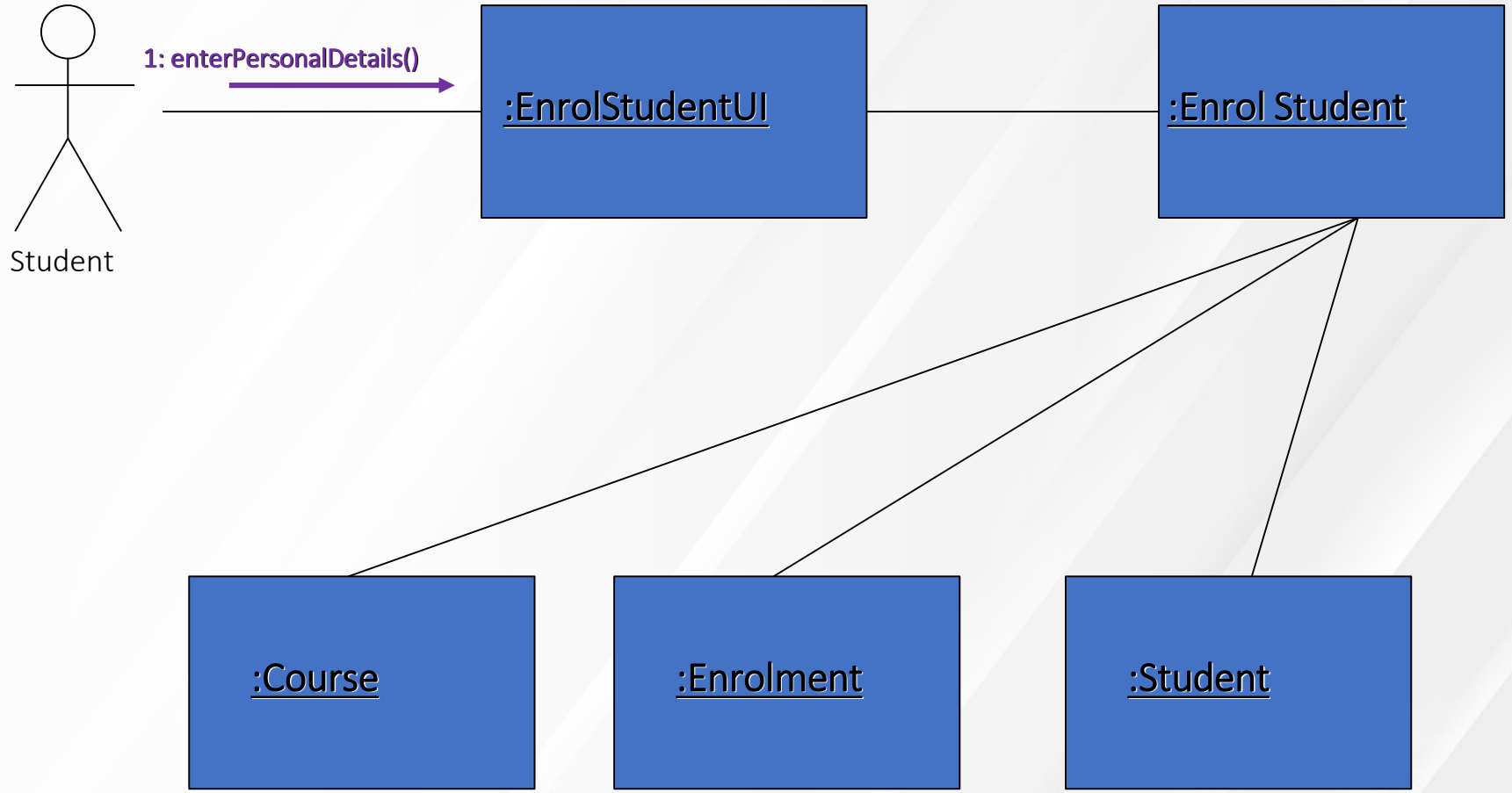
This arrangement resembles the possible situation when a customer visits an office



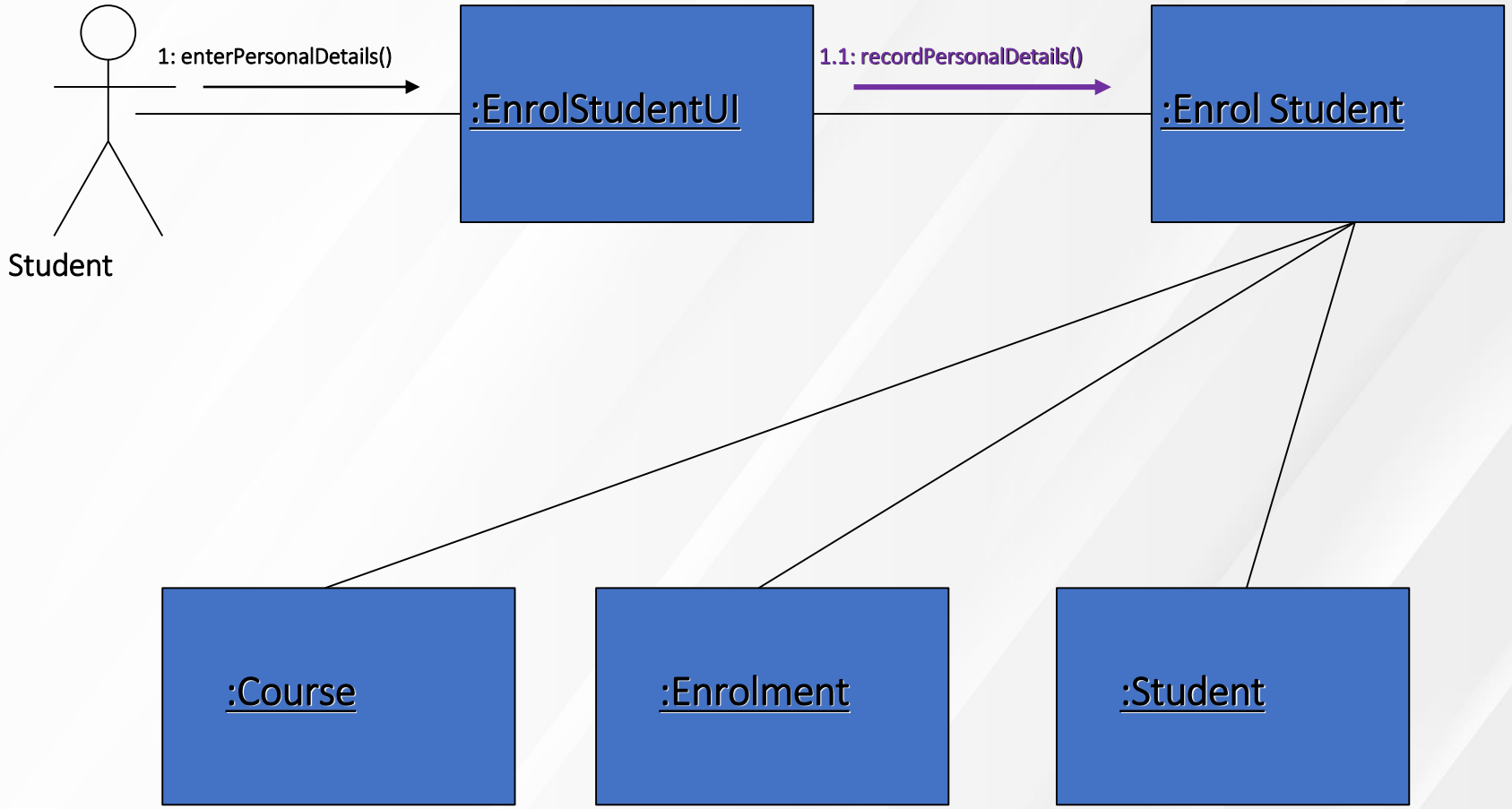
➢However, in the same way that there are other ways to organize an office, there are also other ways to organize Objects in order to carry out a Use Case process

➢We shall not deal with these at the moment. For the moment, let's keep to this model of centralize control.

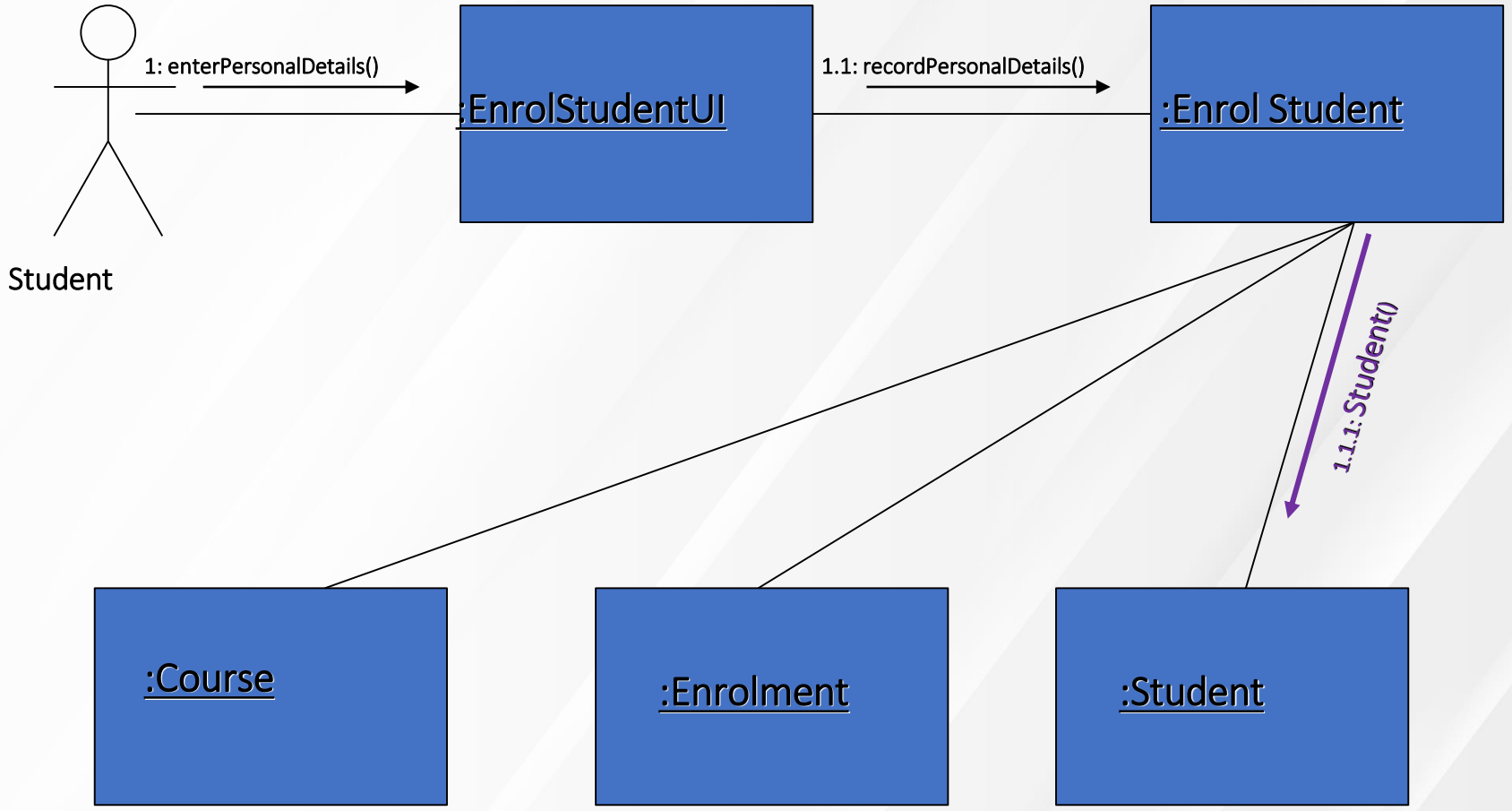
Next start adding messages to the diagram.
If we assume that the User Interface was started up by a menu process – then perhaps the first message might be the User calling a method in the User Interface Class – to enter personal details. Note the numbering and the arrow.



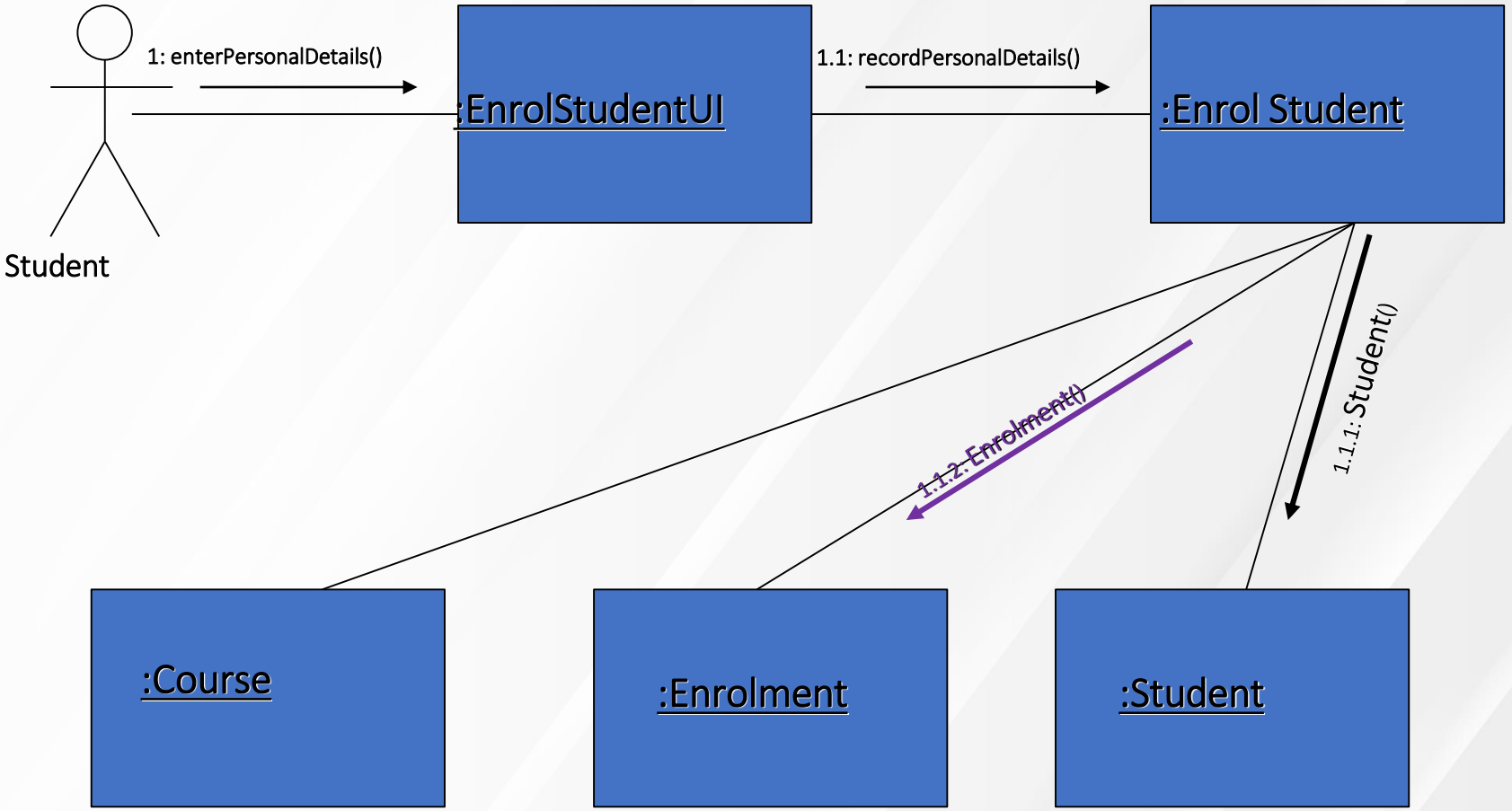
Once the user has entered their name, address, etc, the data might be sent to the Control object using the message **recordPersonalDetails()** *(The parentheses are for parameters – but don't worry about these at this stage).* This message is as a direct result of 1:enterPersonalDetails() and nested within this message – so note the numbering



The control object then will need to create a new Student object in memory to hold the student's personal data. This can be done with **a call to a constructor method – Student()**. Note the numbering – the constructor message is nested as part of recording personal details. Later on we would add parameters to pass the data across to the new Student object.



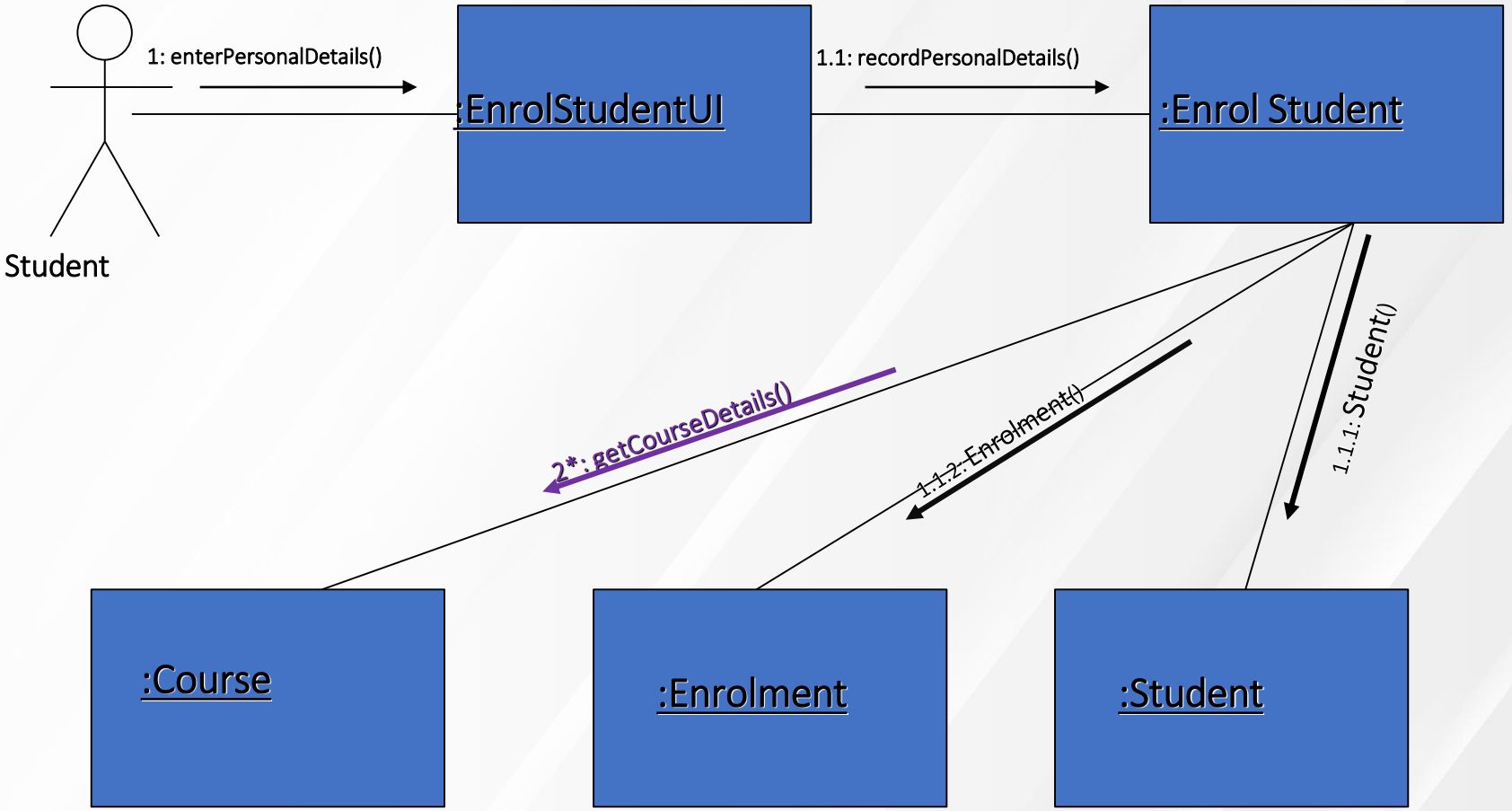
An Enrolment object will be needed to store the information about which course the student is studying – so this could be created now – again with a call to a constructor method. Creating the Enrolment object is the second part of recording personal details – so note the numbering.



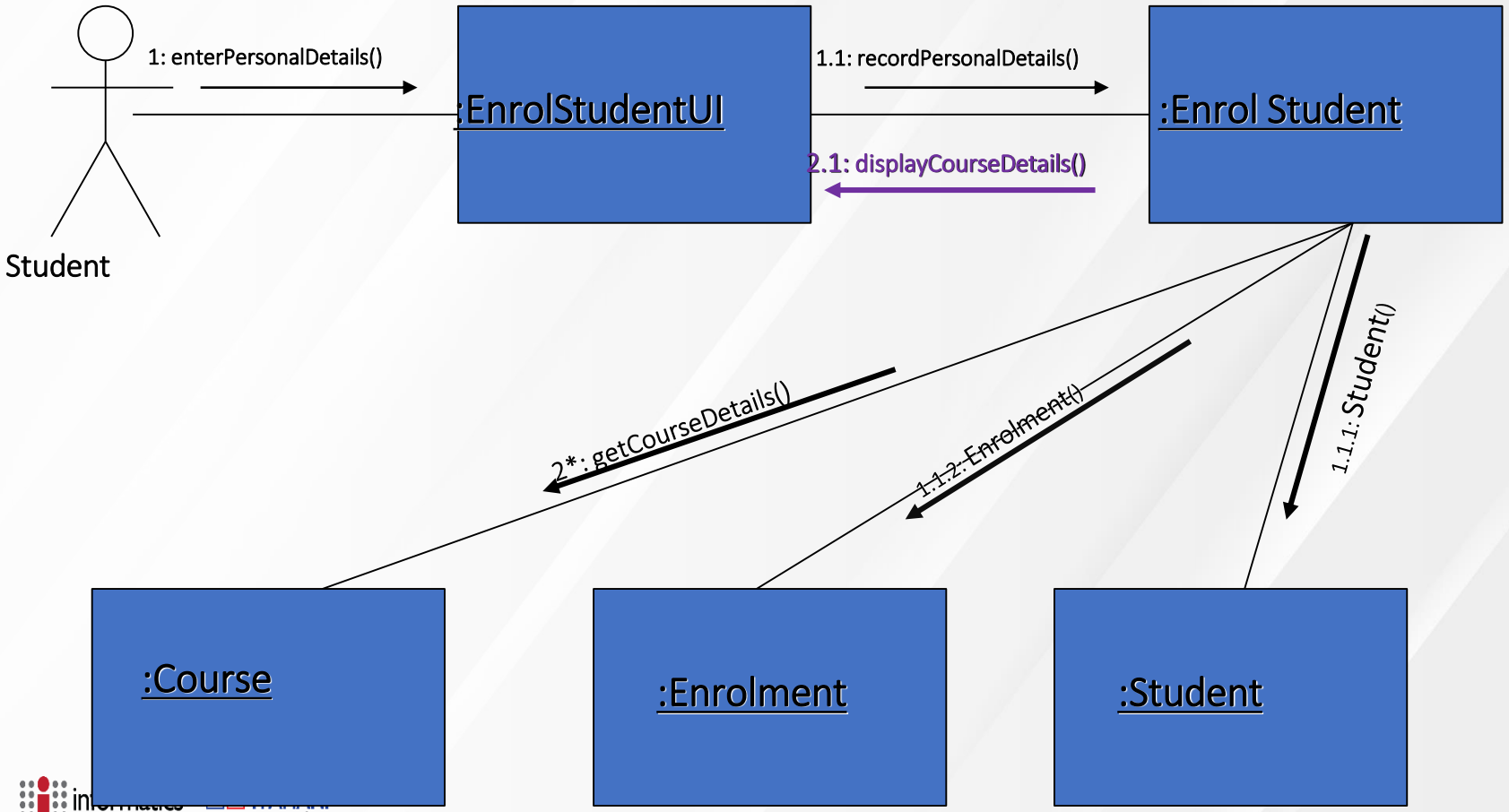
- The diagram now shows all the messages between objects needed to handle the new student's personal details

- We now need to deal with **allowing** the **student** to **choose a course**

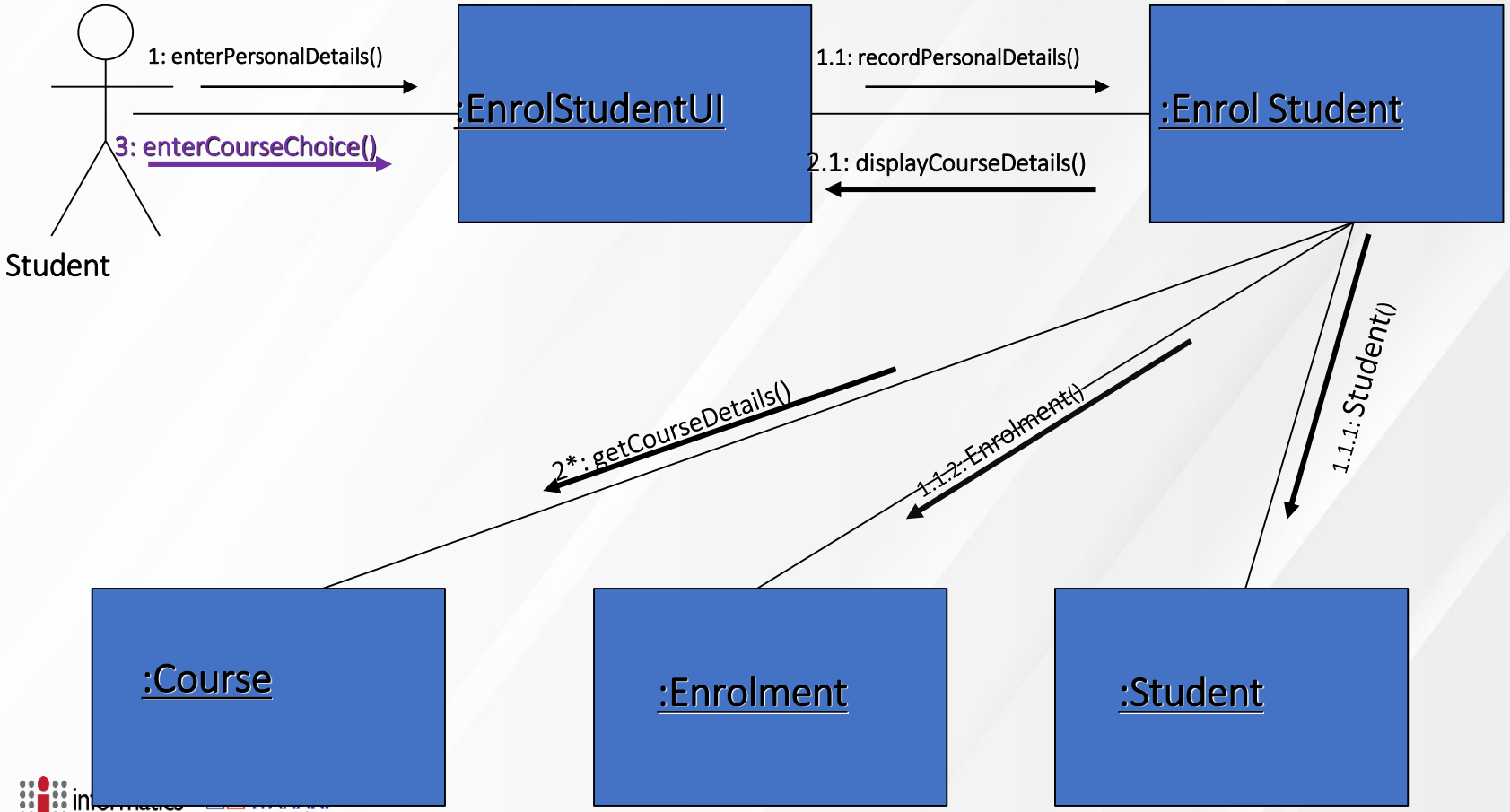
The system needs to put a list of courses on the screen so that the new student can choose one – so this can be obtained by looking at the Course objects. Each object stores the data for one course – so to produce a list, lots of Course objects will be needed. So the getCourseDetails() message contains * to indicate 'repeatedly'. The message will return data – but don't worry about this for the moment.



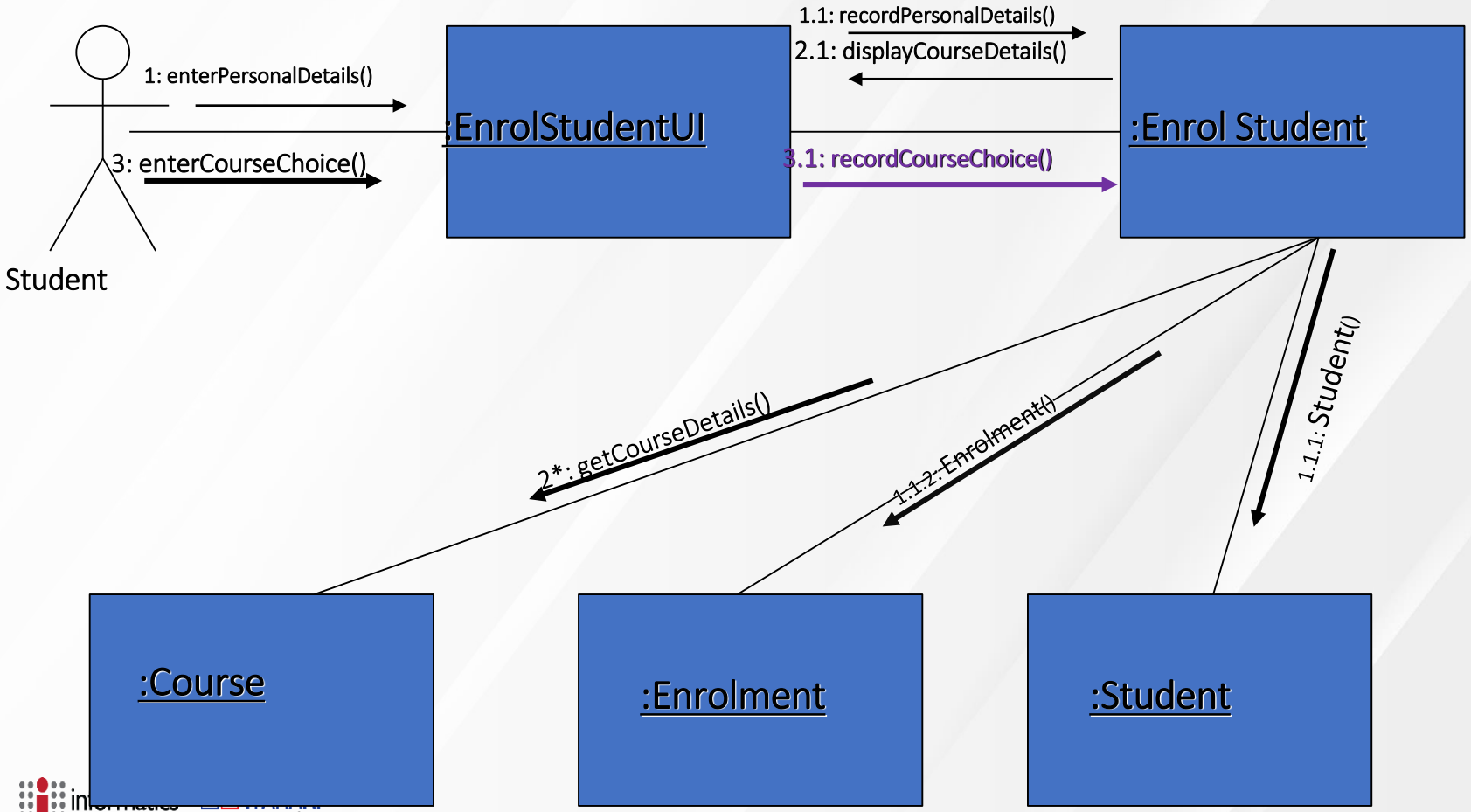
The course data returned by getCourseDetails() needs to be available to the User Interface object so that it can be displayed on the screen. If we assume that the data is sent back one course at a time as it is retrieved from each Course object – then the message that does this displayCourseDetails will also be iterated along with getCourseDetails().



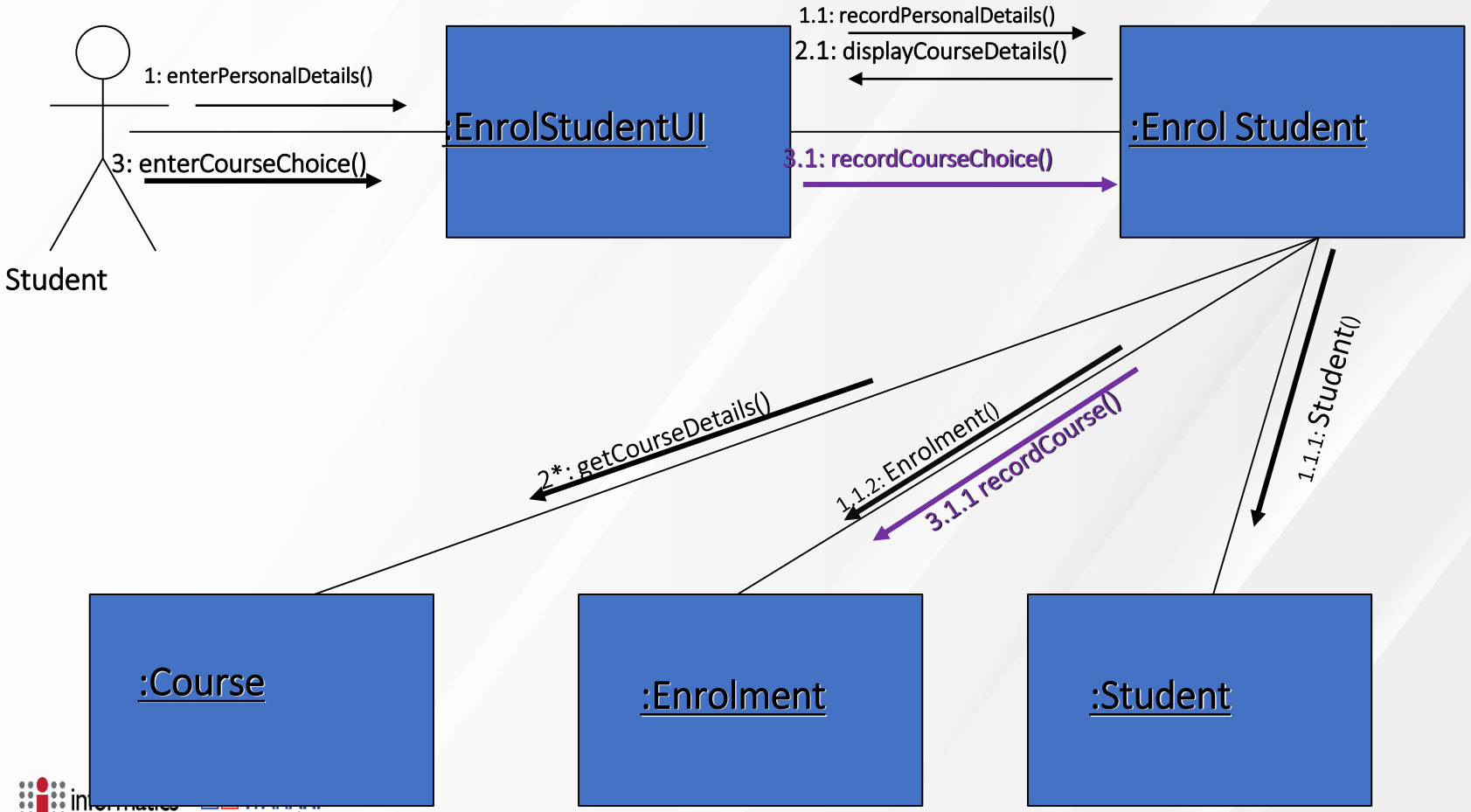
The student having seen the displayed list of courses can then choose his/her course.



This would be processed by a message recordCourseChoice() sent to the Control object – again note the numbering.

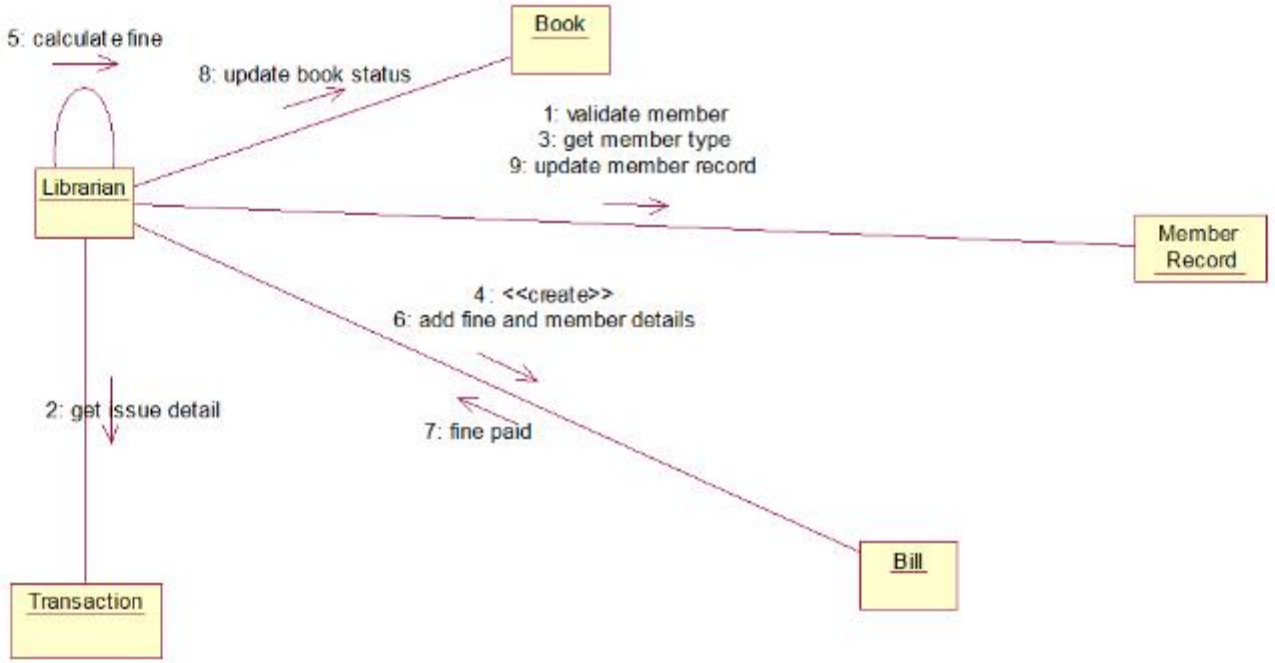


The control object would then send a message to the Enrolment object to save the chosen course. Note the numbering.



Non-MVC approach for collaboration diagram

Collaboration diagram



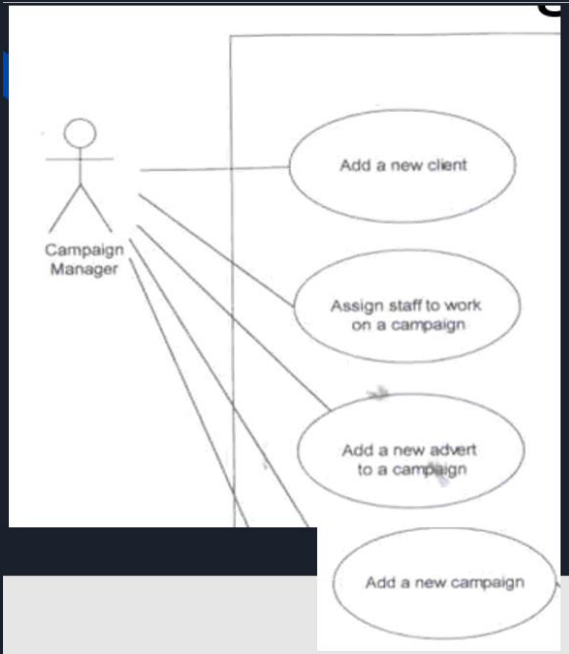
A1.3 Business Activities in the Current System

Agate deals with other companies that it calls clients. A record is kept of each client company, and each client company has one person who is the main contact person within that company. His or her name and contact details are kept in the client record. Similarly, Agate nominates a member of staff—a director, an account manager or a member of the creative team—to be the contact for each client.

Clients have advertising campaigns, and a record is kept of every campaign. One member of Agate's staff, again either a director or an account manager, manages each campaign. Other staff may work on a campaign and Agate operates a project-based management structure, which means that staff may be working on more than one project at a time. For each project they work on, they are answerable to the manager of that project, who may or may not be their own line manager.

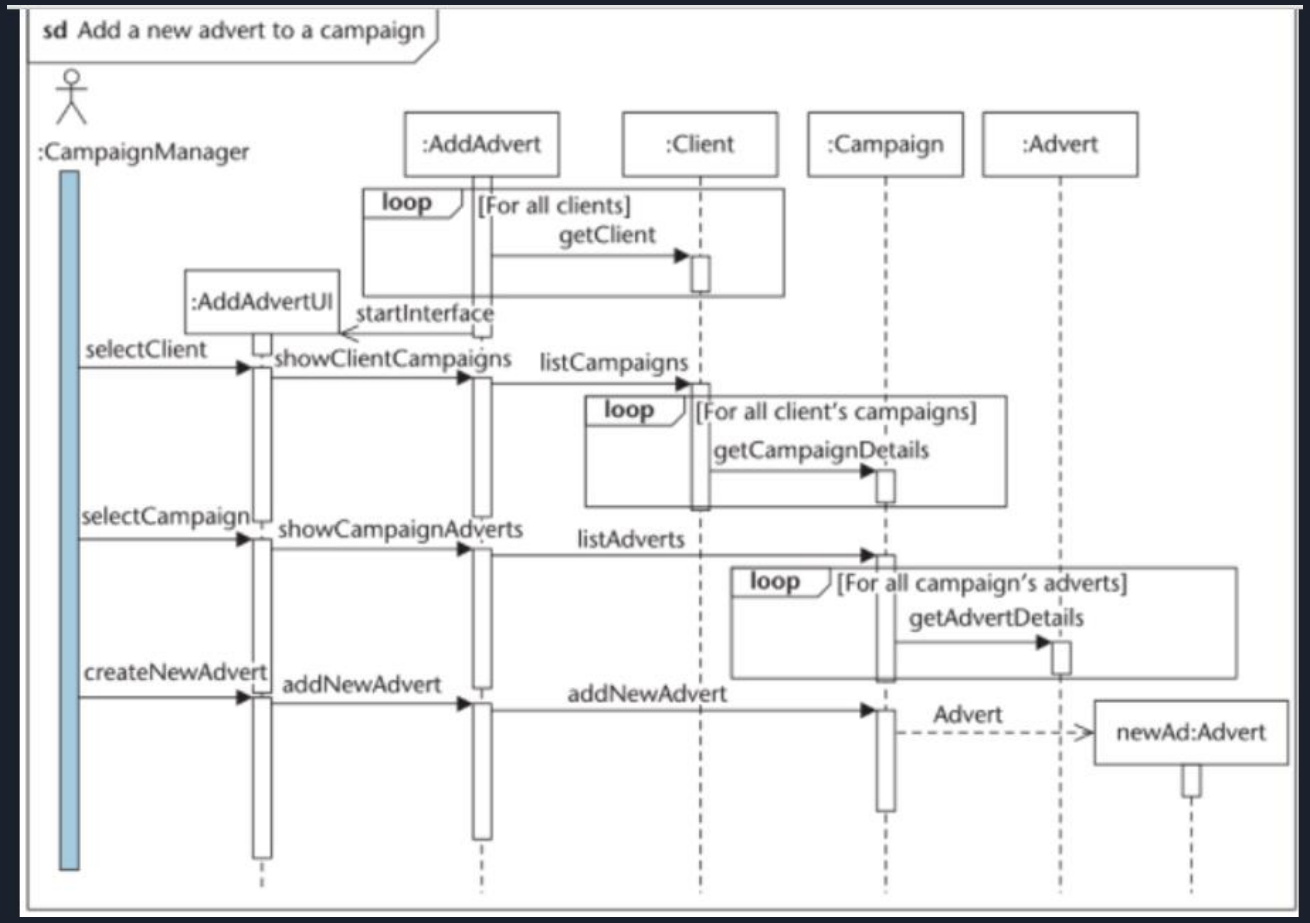
When a campaign starts, the manager responsible estimates the likely cost of the campaign, and agrees it with the client. A finish date may be set for a campaign at any time, and may be changed. When the campaign is completed, an actual completion date and the actual cost are recorded. When the client pays, the payment date is recorded. Each campaign includes one or more adverts. Adverts can be one of several types:

- newspaper advert—including written copy, graphics and photographs
- magazine advert—including written copy, graphics and photographs
- Internet advert—including written copy, graphics, photographs and animations
- TV advert—using video, library film, actors, voice-overs, music etc.
- radio advert—using audio, actors, voice-overs, music etc.
- poster advert—using graphics, photographs, actors

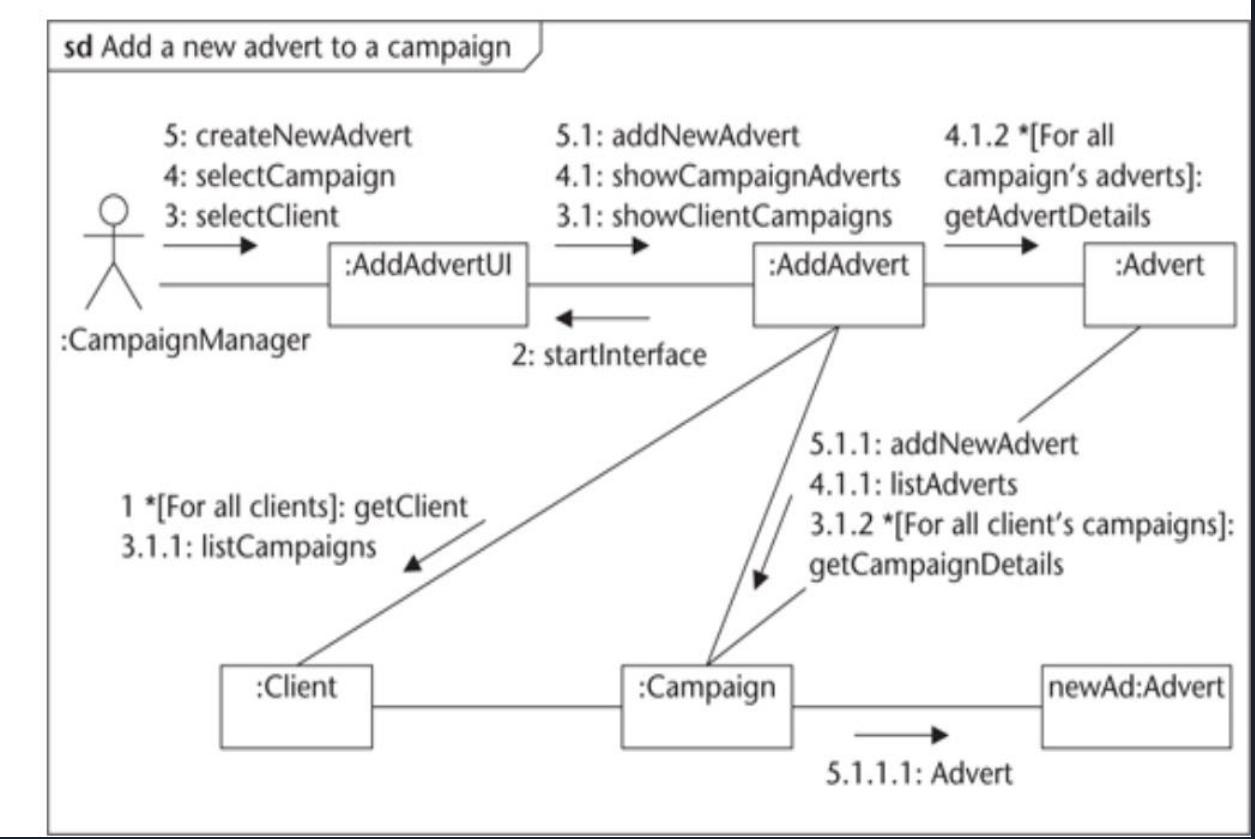


Use Case	Description
Add a new client	When Agate obtains a new client, the full details of the client are entered. Typically this will be because of a new campaign, and therefore the new campaign will be added straight away.
Assign staff to work on a campaign	The campaign manager selects a particular campaign. A list of staff not already working on that campaign is displayed, and he or she selects those to be assigned to this campaign.
Add a new advert to a campaign	A campaign can consist of many adverts. Details of each advert are entered into the system with a target completion date.
Add a new campaign	When Agate gets the business for a new campaign, details of the campaign are entered, including the intended finish date and the estimated cost. The manager for that campaign is the person who enters it.

From last week's example: Sequence diagram for add a new advert use case

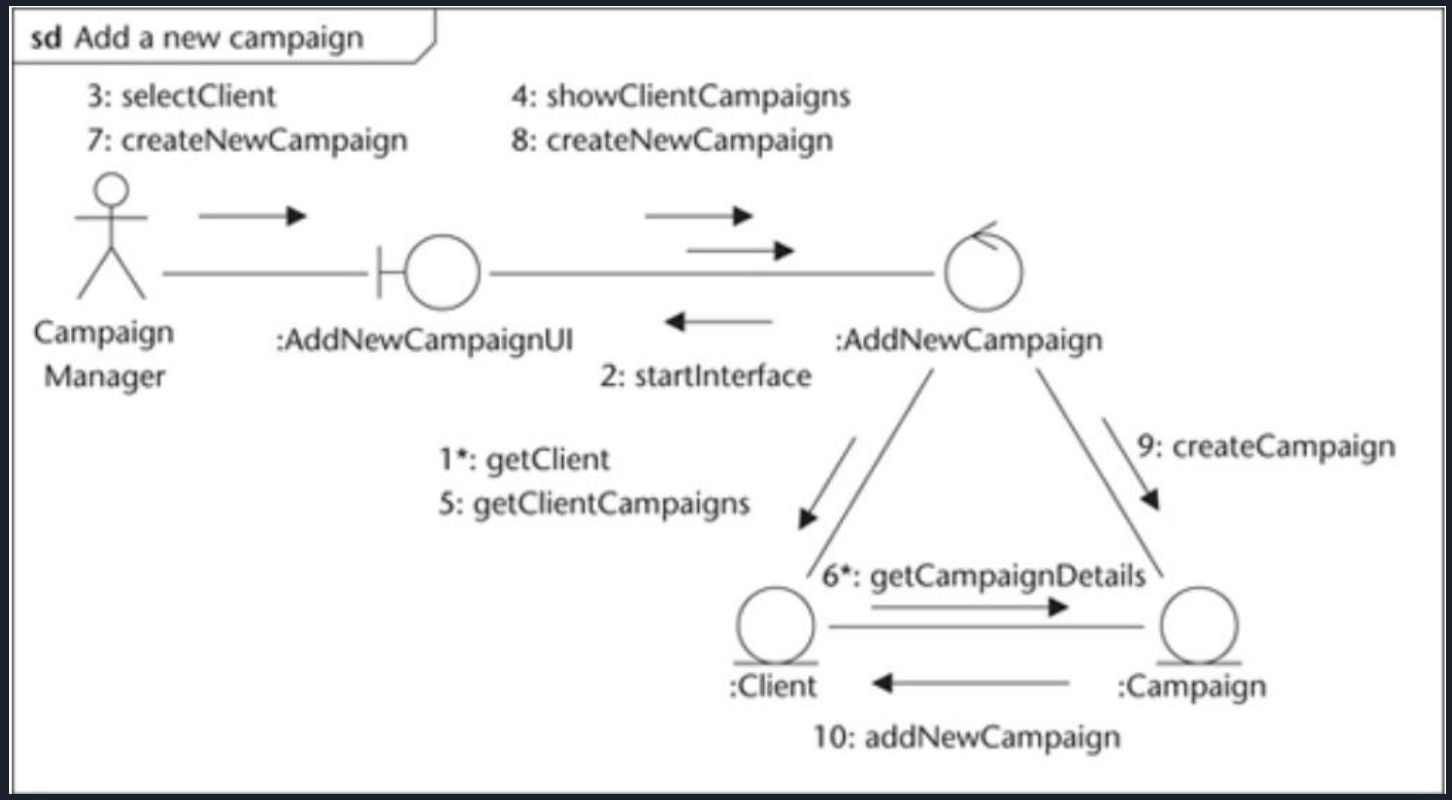


Communication diagram for Add a new advert use case



(Asterik '*') Denotes 'iteration' and represents multiple call

Communication diagram for Add a new campaign



Note that the collaboration diagram depends on how the analyst chooses to allocate responsibilities between the objects and on the chosen order of processing. It would be possible to draw an entirely different collaboration diagram to do the same task.

THANK YOU!