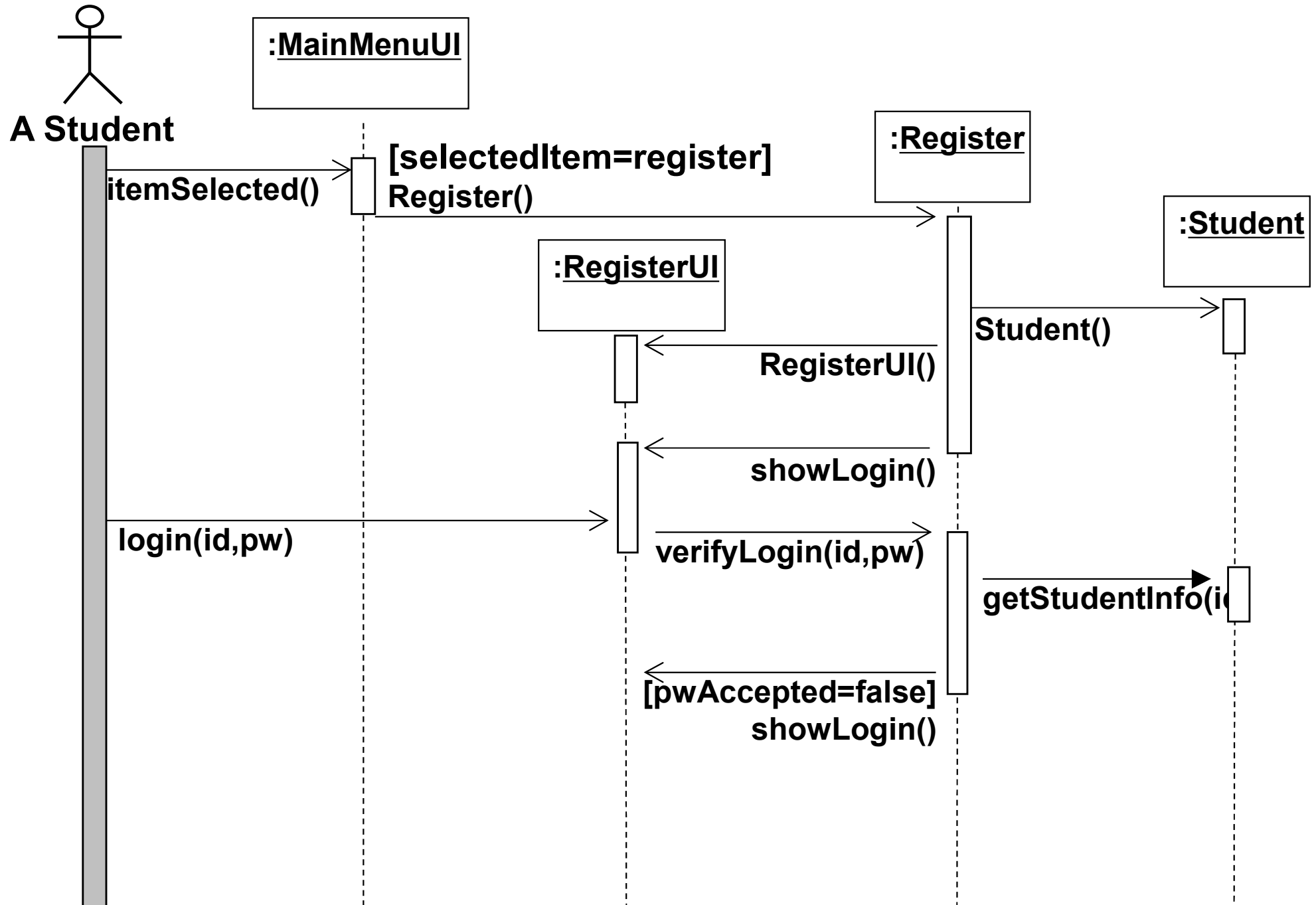


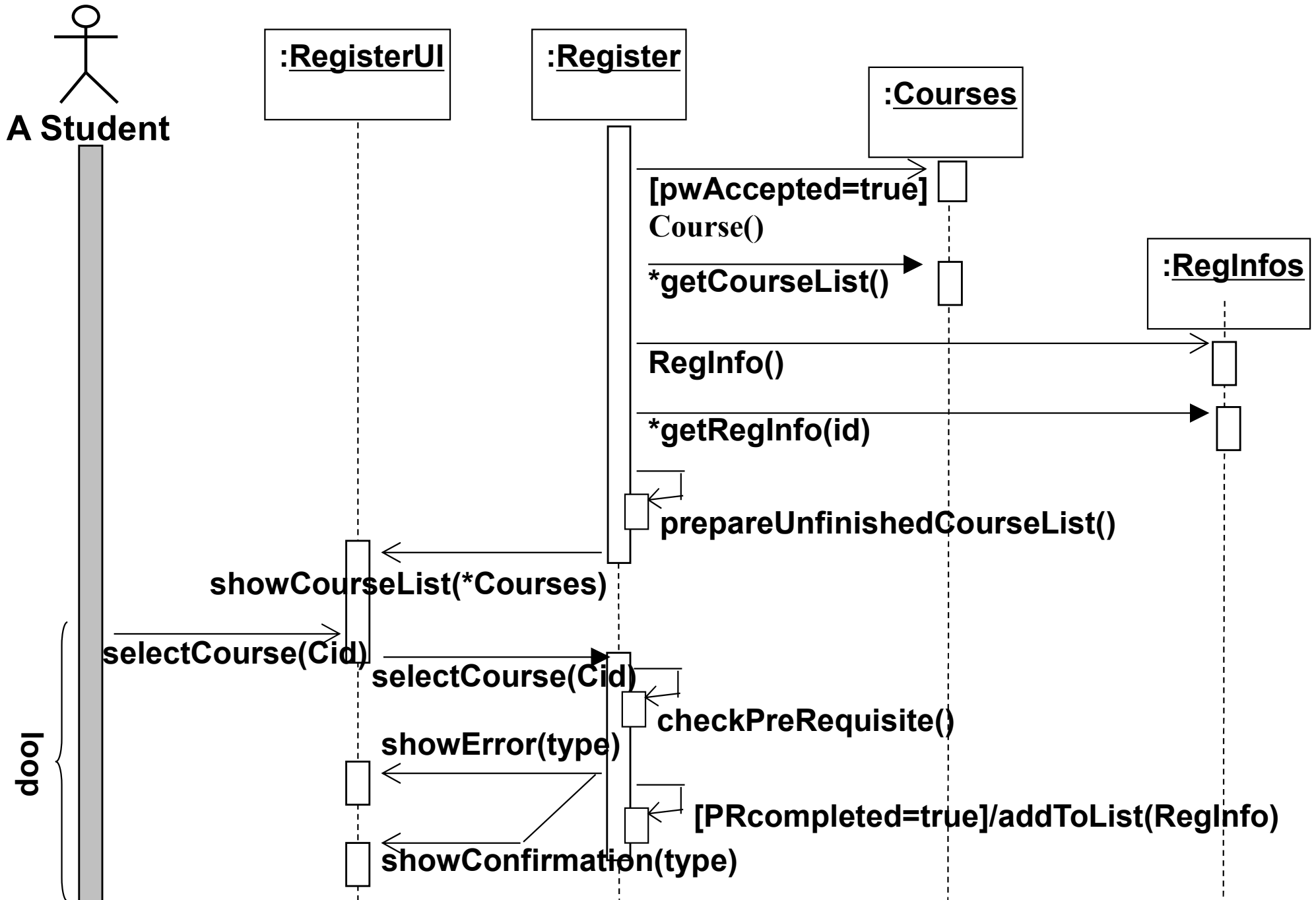
# Register: typical course of events

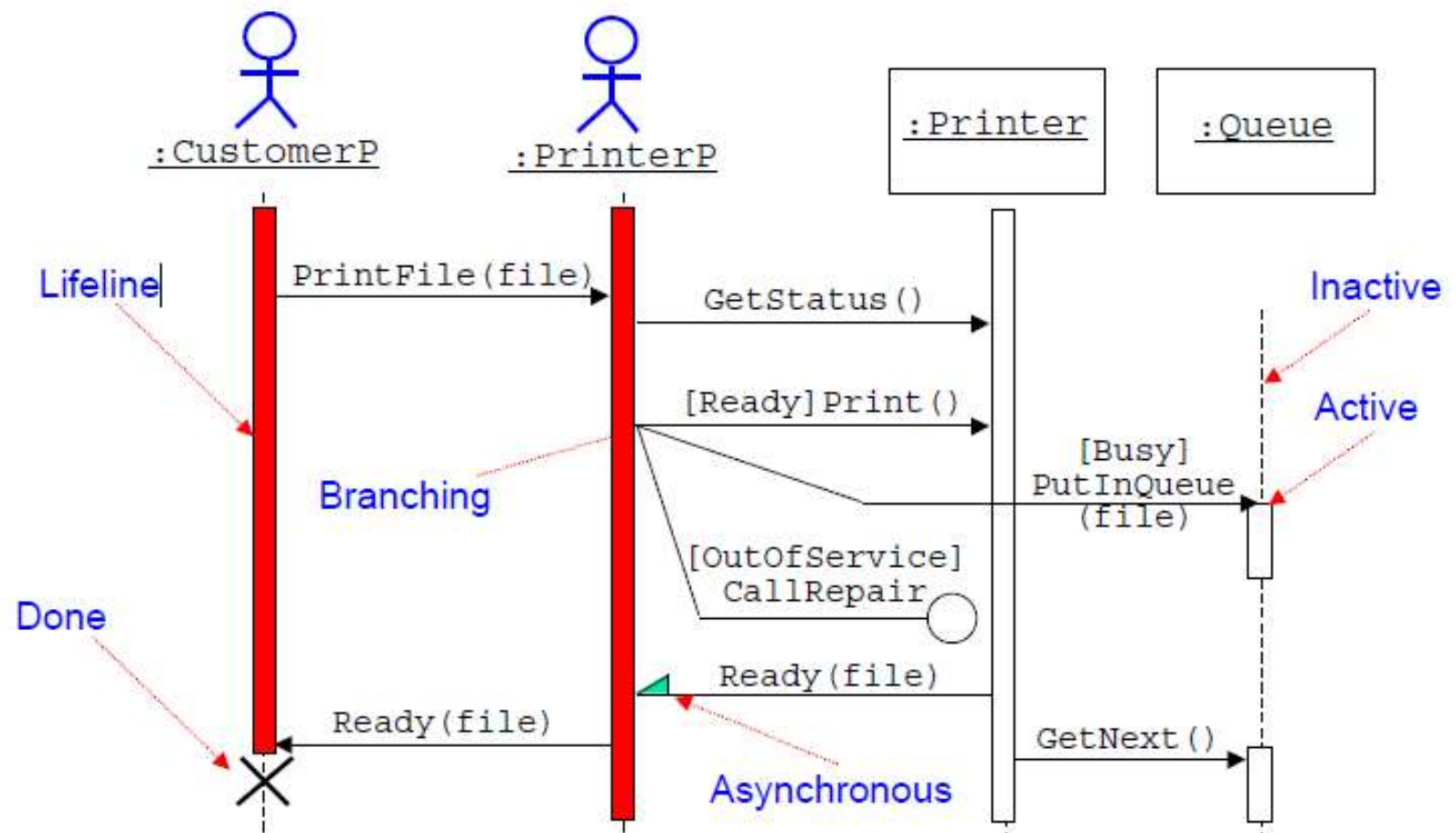
<b>Actor Action (Student)</b>	<b>Actor Action (Adviser)</b>	<b>System Response</b>
<b>1. Open registration Interface</b>		<b>2. Ask for Student Id, Password</b>
<b>3. Provide Id, Password</b>		<b>4. Show List of unfinished courses</b>
<b>5. Select a course</b>		<b>6. Get Pre-requisite list for that course</b>
		<b>7. Check results of those courses</b>
		<b>8. If not passed, then give message to student and deselect the course Otherwise, do nothing</b>
<b>9. Finalize selection</b>		<b>10. Check total credit hour</b>
		<b>11. If ok, then proceed to next step otherwise, give a message requesting the student to revise his selection</b>
		<b>12. Notify Adviser</b>
	<b>13. Ask for details</b>	<b>14. Show details</b>
	<b>15. Authorize</b>	<b>16. Update state</b>
		<b>17. update registration information for selected courses and show confirmation message, Print course card</b>

# Register: Sequence Diagram 1

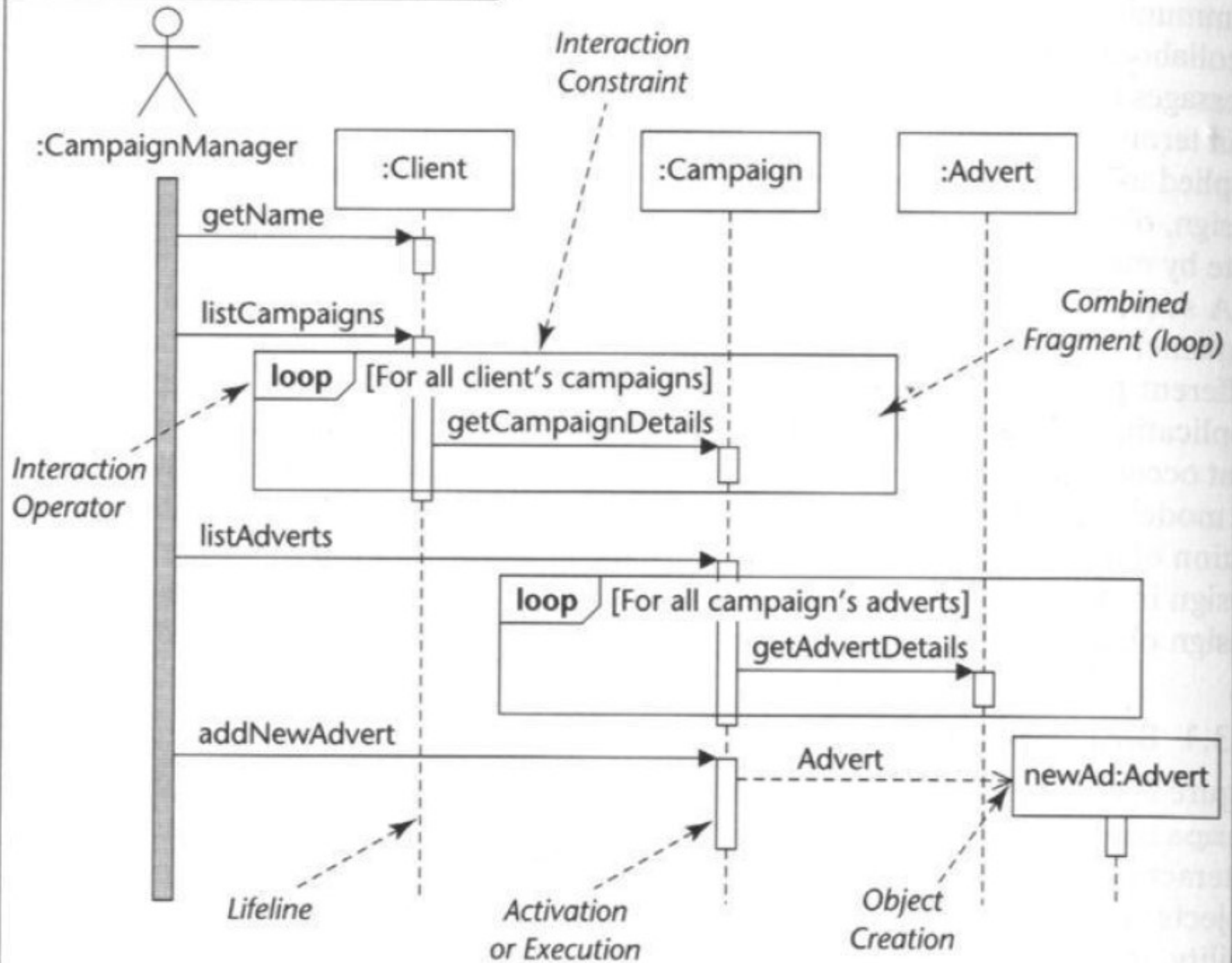


# Register: Sequence Diagram 2





sd Add a new advert to a campaign



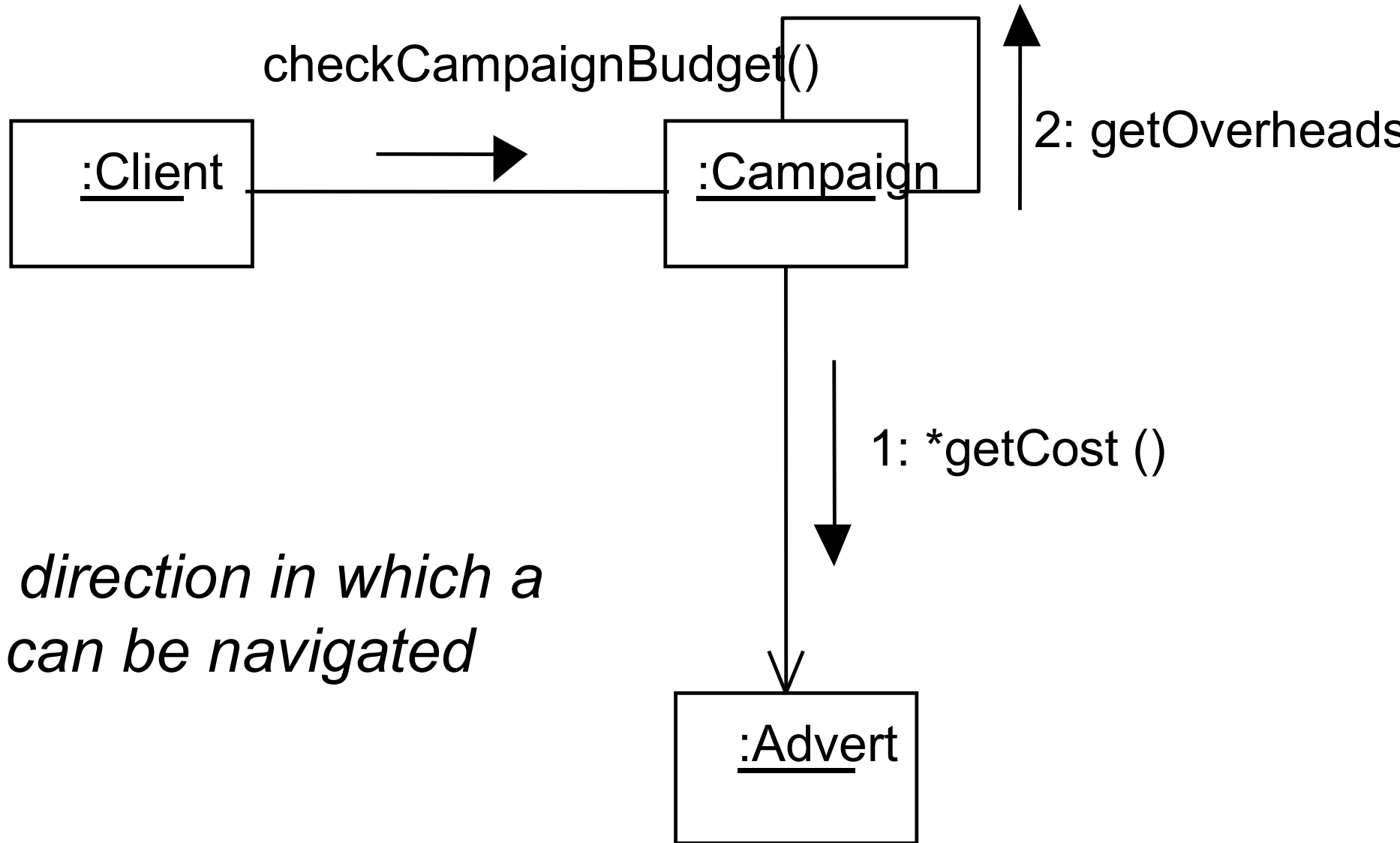
# Collaboration Diagrams

- Hold the same information as sequence diagrams
- Show links between objects that participate in the collaboration
- No time dimension, sequence is captured with sequence numbers
- Sequence numbers are written in a nested style (for example, 3.1 and 3.1.1) to indicate the nesting of control within the interaction that is being modelled

# Message Labels

Type	Example
• <b>Simple message.</b>	4: addItem()
• <b>Nested call with return value.</b> –The return value is placed in the variable name.	3.1.2: name:= getName()
• <b>Conditional message.</b> –This message is only sent if the condition [balance > 0] is true.	[balance > 0] 5: debit(amount)
• <b>Synchronization with other threads.</b> – <i>Message 4: playVideo() is invoked only once the two concurrent messages 3.1a and 3.1b are completed.</i>	3.1a, 3. 1b / 4:playVideo()

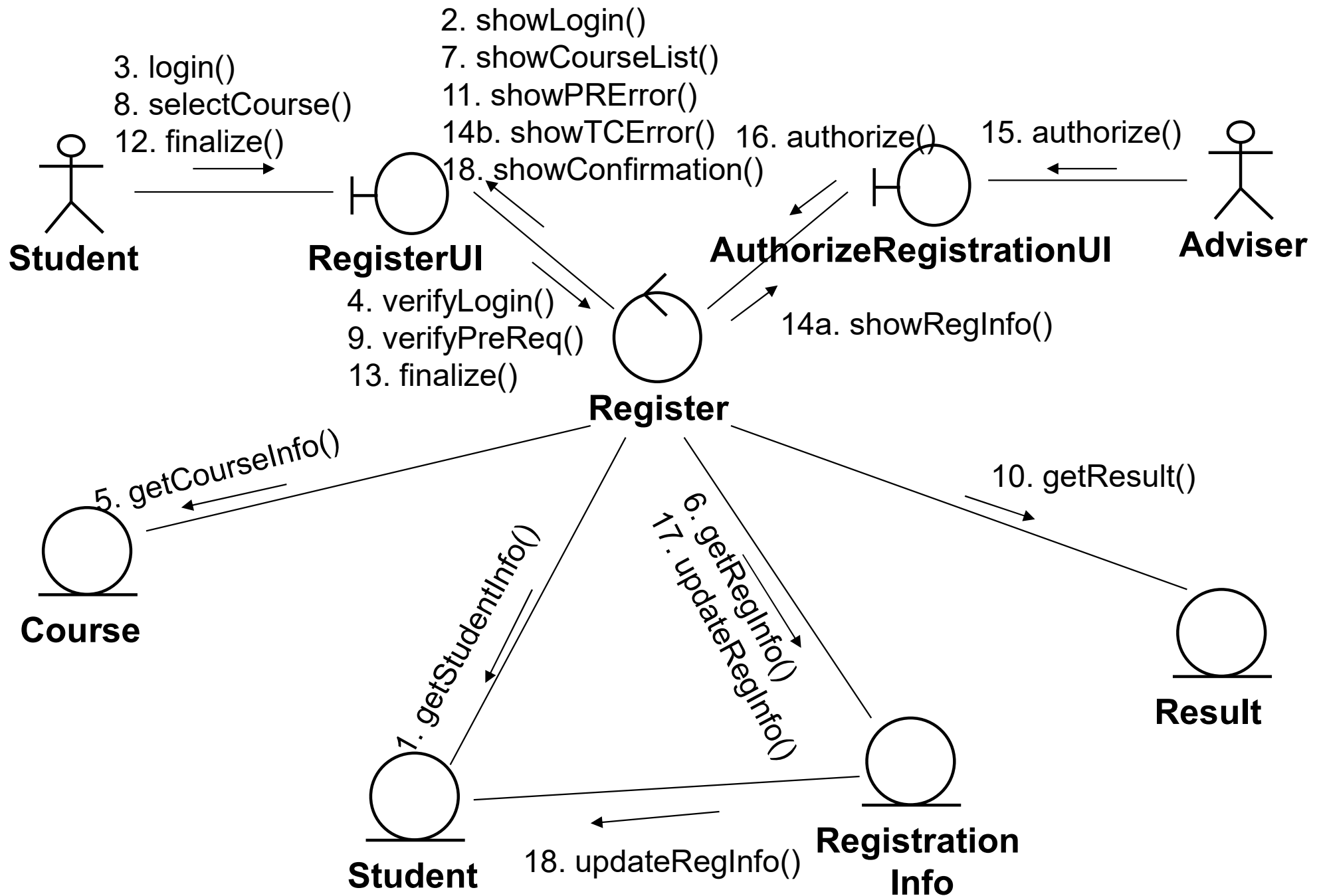
# Navigating Links



*The direction in which a link can be navigated*



# Register: Collaboration Diagram



# Model Consistency

- The allocation of operations to objects must be consistent with the class diagram and the message signature must match that of the operation
- Every sending object must have the object reference for the destination object
  - Either an association exists between the classes or another object passes the reference to the sender
  - Message pathways should be carefully analysed