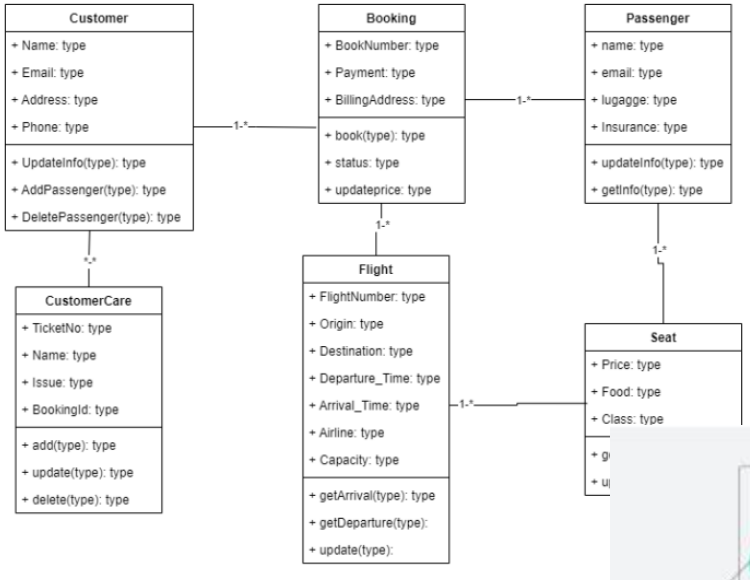
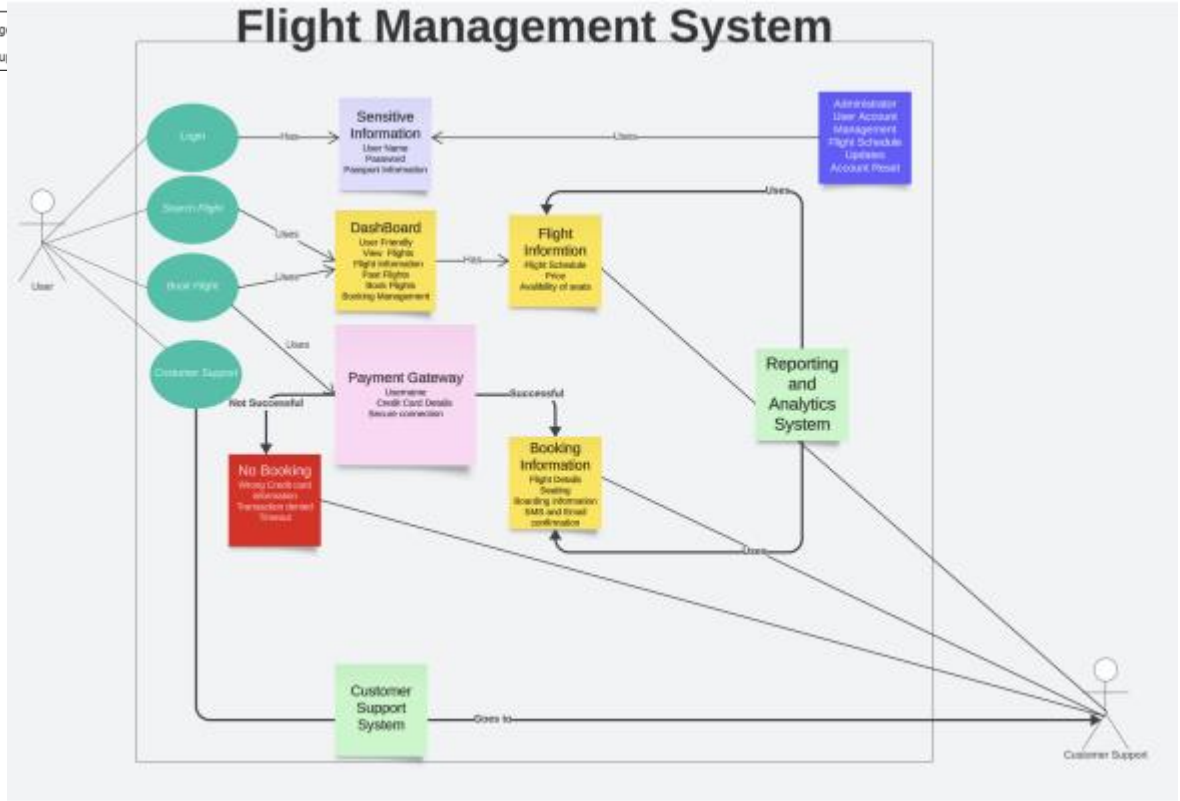
Chapter 5 – System Modeling

Sequence Models


Customer
+ Name: type
+ Email: type
+ Address: type
+ Phone: type
+ UpdateInfo(type): type
+ AddPassenger(type): type
+ DeletePassenger(type): type
Booking
+ BookNumber: type
+ Payment: type
+ BillingAddress: type
+ book(type): type
+ status: type
+ updateprice: type
Passenger
+ name: type
+ email: type
+ lugagge: type
+ Insurance: type
+ updateInfo(type): type
+ getInfo(type): type
CustomerCare
+ TicketNo: type
+ Name: type
+ Issue: type
+ BookingId: type
+ add(type): type
+ update(type): type
+ delete(type): type
Flight
+ FlightNumber: type
+ Origin: type
+ Destination: type
+ Departure_Time: type
+ Arrival_Time: type
+ Airline: type
+ Capacity: type
+ getArrival(type): type
+ getDeparture(type):
+ update(type):
Seat
+ Price: type
+ Food: type
+ Class: type
1-*
-


Software Engineering
Ian Sommerville


Flight Management System
Sensitive Information
DashBoard
Flight Information
Payment Gateway
Reporting and Analytics System
Booking Information
No Booking
Customer Support System
Not Successful
Successful
Goes to
Customer Support

User and system requirements

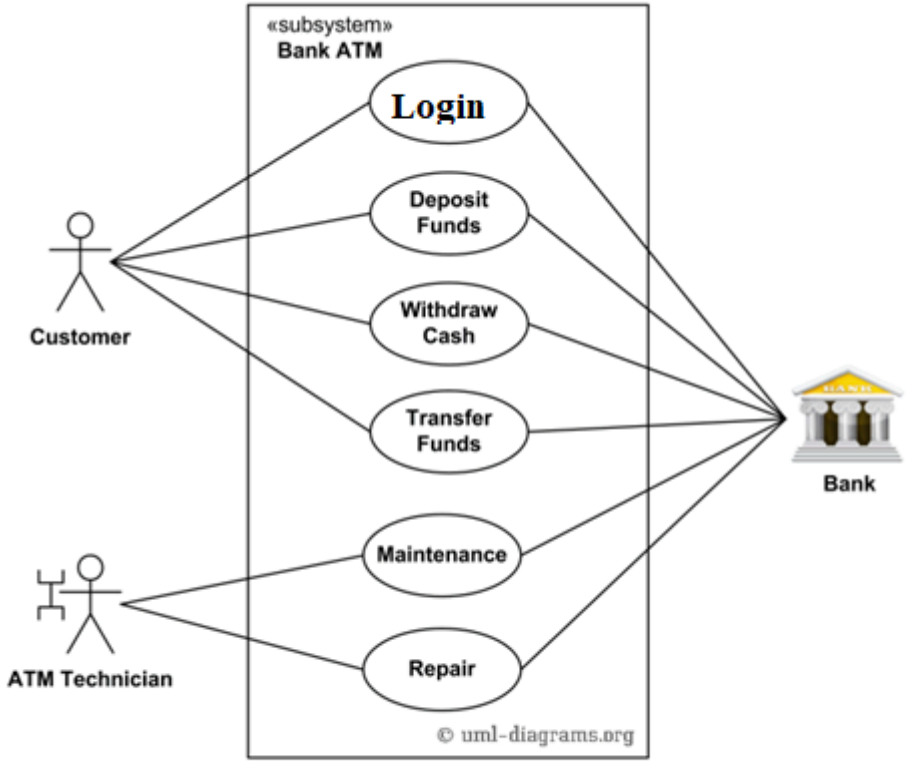
User requirements definition

1. The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

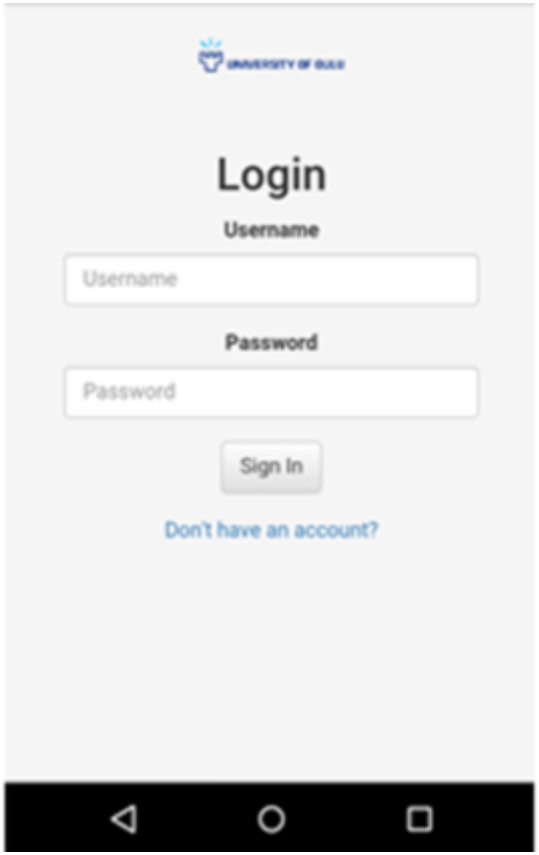
System requirements specification

1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
1.2 The system shall generate the report for printing after 17.30 on the last working day of the month.
1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
1.4 If drugs are available in different dose units (e.g. 10mg, 20mg, etc) separate reports shall be created for each dose unit.
1.5 Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

User Requirements : Use case Diagram



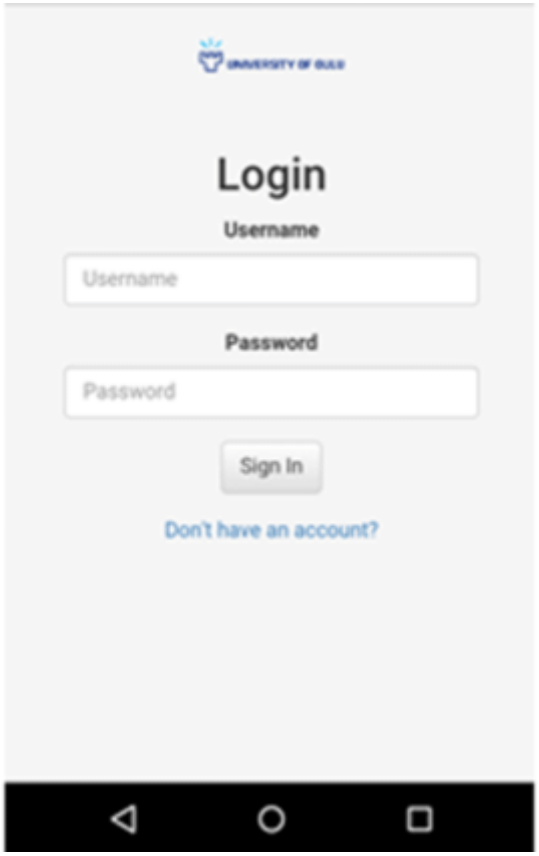
System Requirements: Sequence Diagram/Models



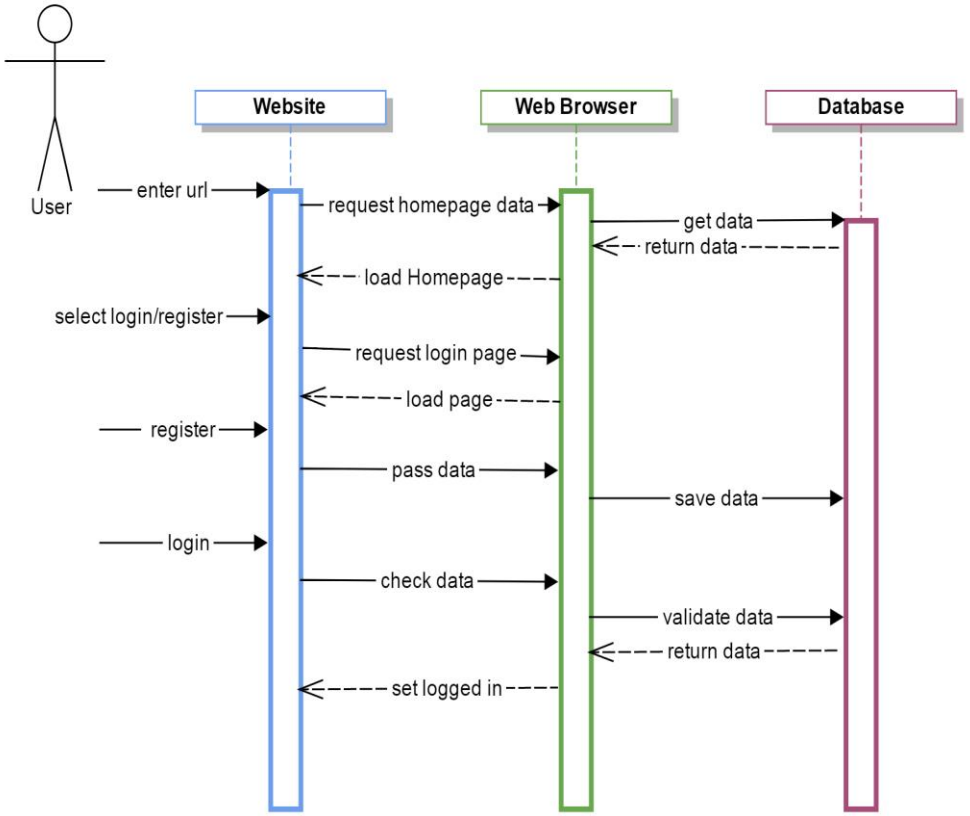
Dev/Soft eng – what do you do next ?

System Requirements: Sequence Diagram/Models

Prototype: Agile



Design : Plan driven



Topics covered

- ✧ Context models
- ✧ Interaction models (Sequence Models)
- ✧ Structural models
- ✧ Behavioral models
- ✧ Model-driven engineering

UML diagram types

- Use case diagrams, which show the interactions between a system and its environment.
- Class diagrams, which show the object classes in the system and the associations between these classes.
- **Sequence diagrams**, which show interactions between actors and the system and between system components.
- State diagrams, which show how the system reacts to internal and external events.

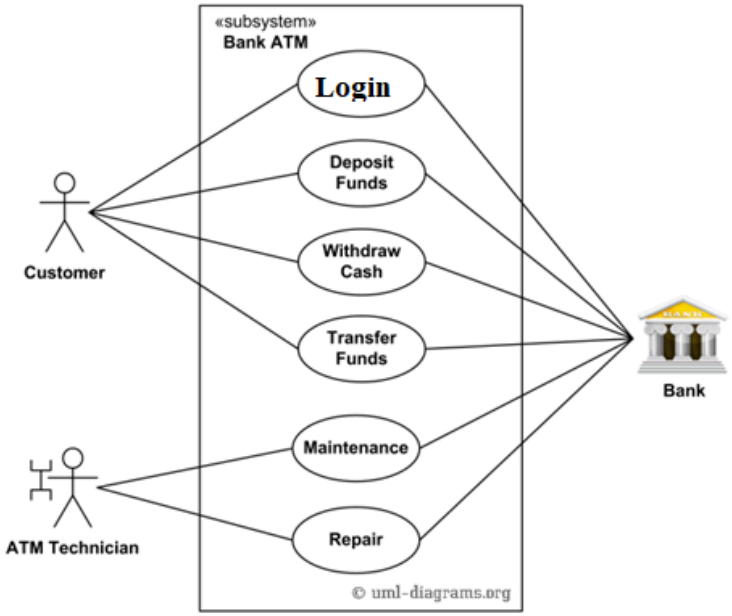
Interaction Diagrams: Sequence diagrams

- UML Specifies a number of interaction diagrams to model dynamic aspects of the system
- Dynamic aspects of the system
  - Messages moving among objects/ classes
  - Flow of control among objects
  - Sequences of events

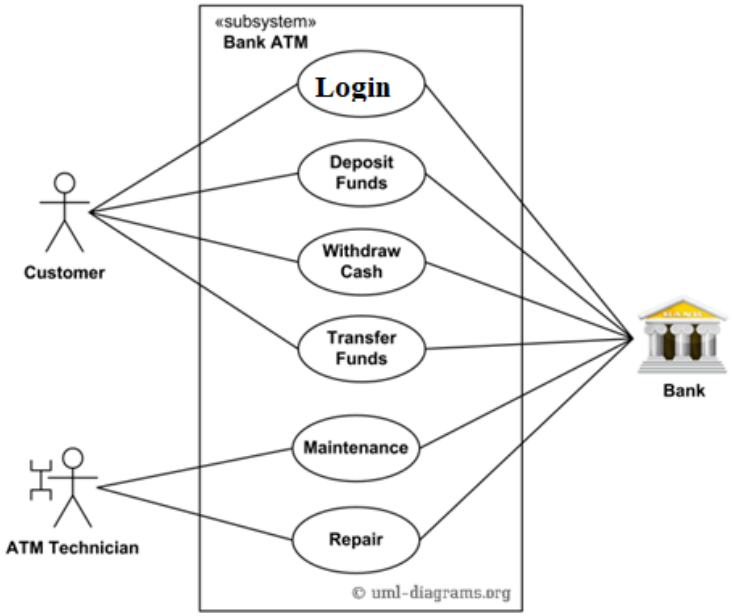
UML sequence diagrams

- **sequence diagram**: an "interaction diagram" that models a single scenario executing in the system
  - perhaps 2nd most used UML diagram (behind class diagram)
  - use cases -> sequence diagrams

use cases -> sequence diagrams

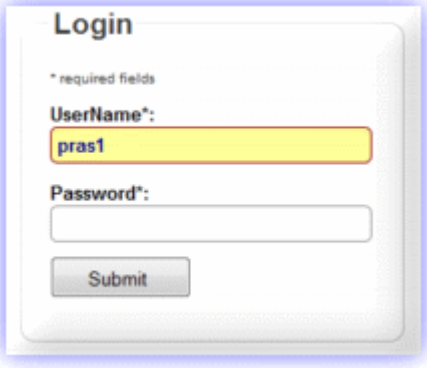


use cases -> sequence diagrams



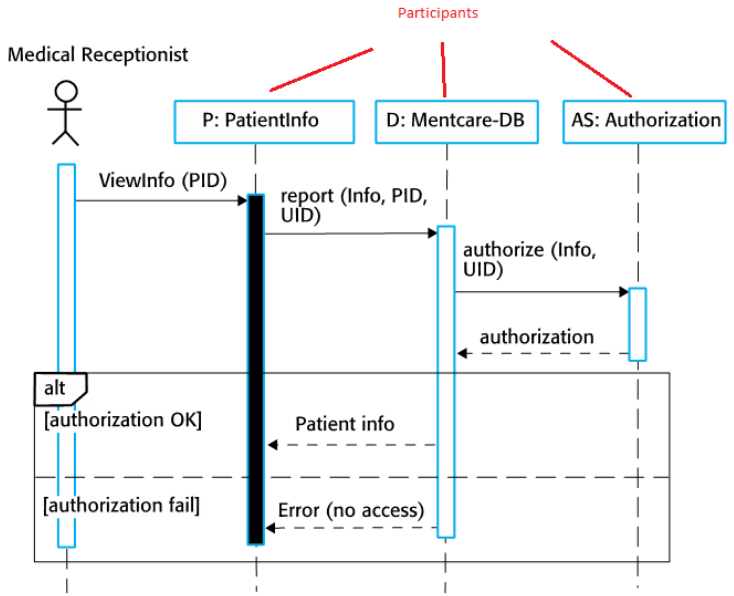
Login

```
<!DOCTYPE html>
<html>
  <head>
    <title>Wikitechy Login Form</title>
1 ->  <link rel="stylesheet" type="text/css" href="login-style.css">
  </head>
  <body>
2 ->  <form class="form container">
      <h2>HTML5 Login Form</h2>
      <label><b>Username</b></label>
3 ->  <input type="text" name="u
      <label><b>Password</b></la
4 ->  <input type="password" name
5 ->  <button type="submit">Logi
    </form>
  </body>
</html>
```



Sequence diagrams

- ✧ Components of Sequence diagrams
- ✧ Message types
- ✧ Model Terminology and Concepts



Sequence diagrams

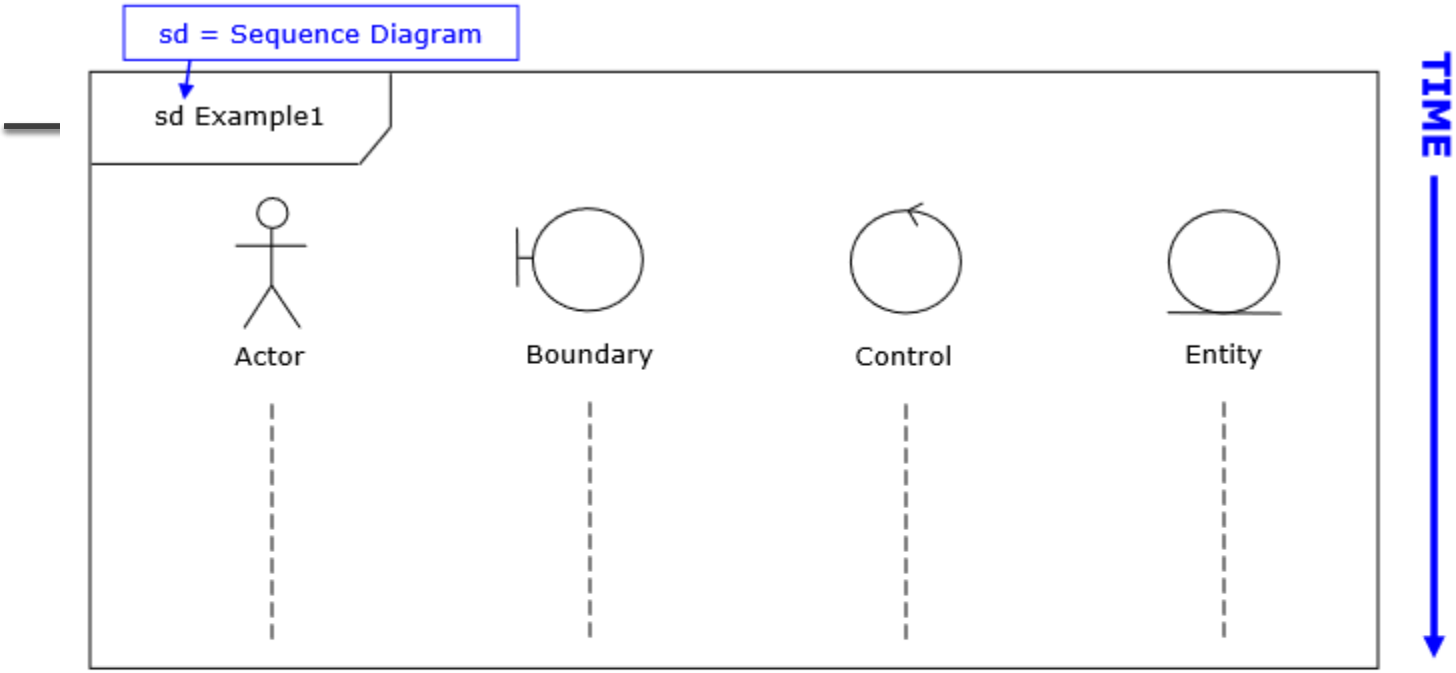
- Sequence diagrams are part of the UML and are used to model the interactions between the actors and the objects within a system.
- A sequence diagram shows the sequence of interactions that take place during a **particular or specific** to a use case or use case instance.
- The objects and actors involved are listed along the top of the diagram, with a dotted line drawn vertically from these.
- Interactions between objects are indicated by annotated arrows.

Sequence diagrams

- Describe the flow of messages, events, actions between objects
- Show time sequences that are not easily depicted in other diagrams
- Typically used during analysis and design to document and understand the **logical flow of your system**

Key parts :Sequence Model

- **participant**: an object or entity that acts in the sequence diagram
  - sequence diagram starts with an unattached "found message" arrow
- **message**: communication between participant objects
- the axes in a sequence diagram:
  - horizontal: which object/participant is acting
  - vertical: time (down -> forward in time)



Actor: Stakeholder within a system

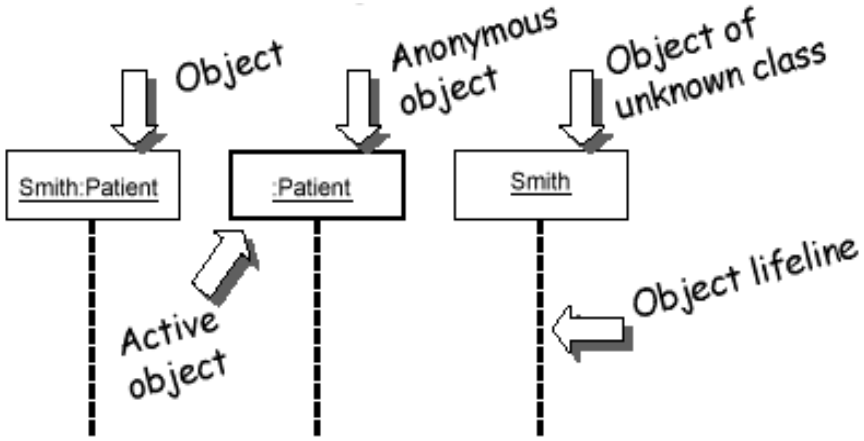
Boundary: Objects that interface with the actors. Examples would include MS Windows, Screens, or Menus.

Control: Mediate between Boundaries and Entities. It captures the control logic.

Entity: Objects representing system data. Can be thought of as the from the Entity-Relation perspective.

Representing objects: Specific (object) Communications

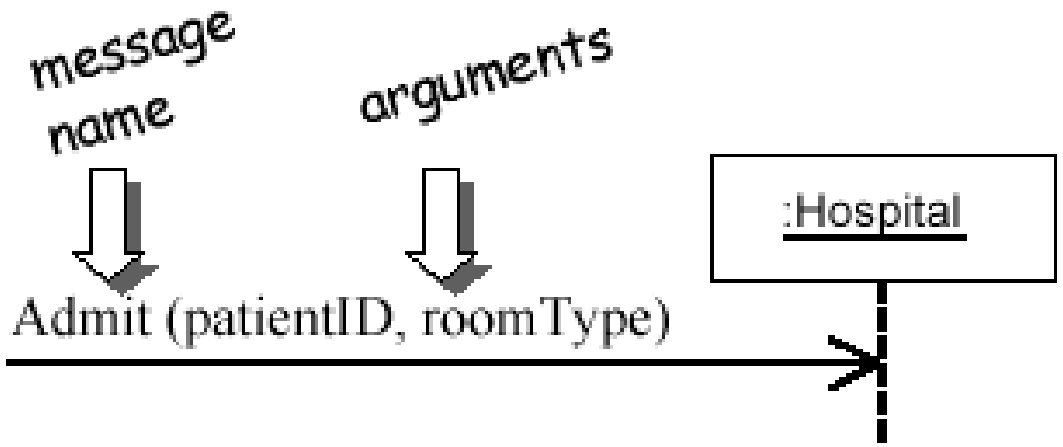
- Squares with object type, optionally preceded by **object name and colon**
  - write object's name if it clarifies the diagram
  - object's "life line" represented by dashed vert. line



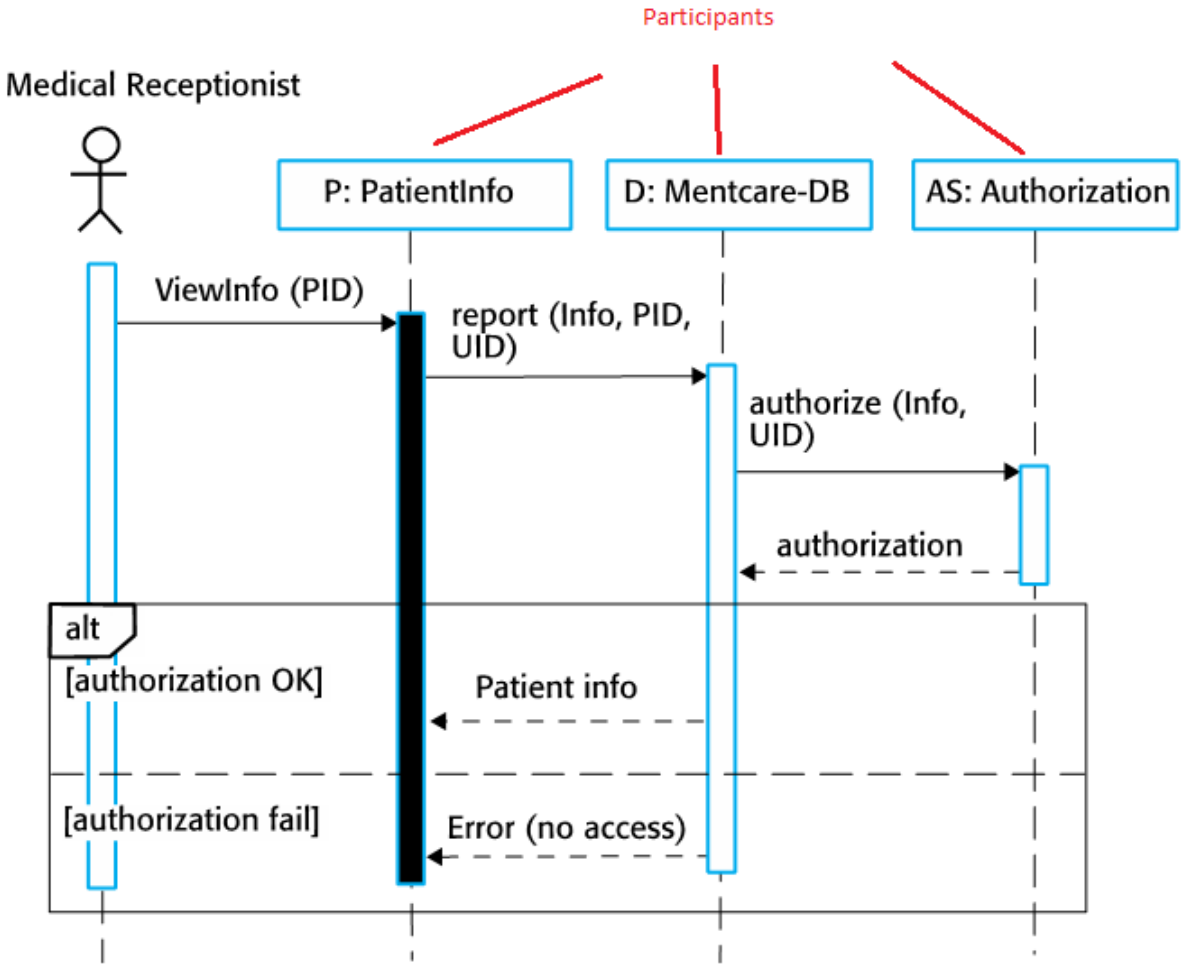
Name syntax: <objectname>:<classname>

Messages between objects

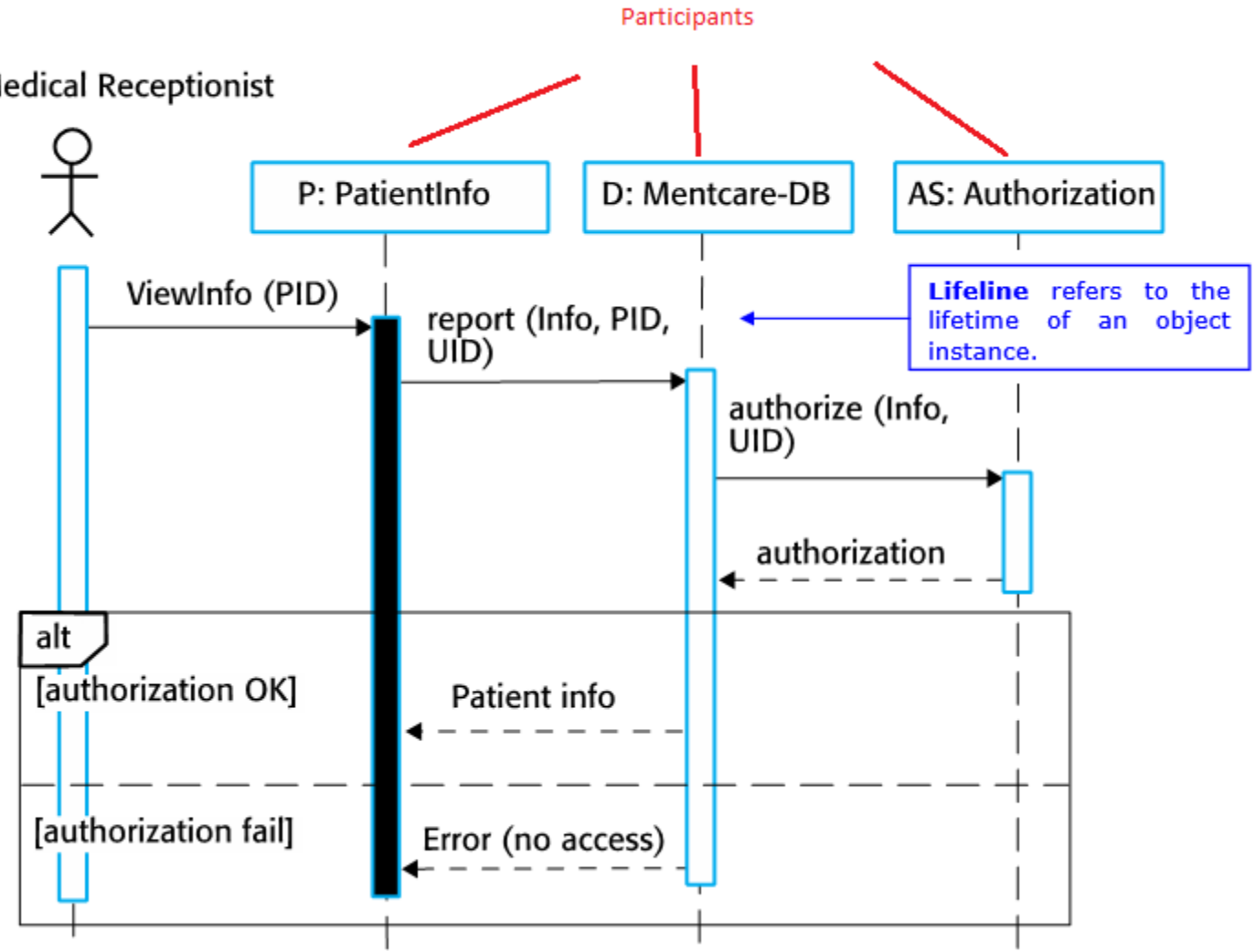
- message (method call) indicated by horizontal arrow to other object
  - write message name and arguments above arrow



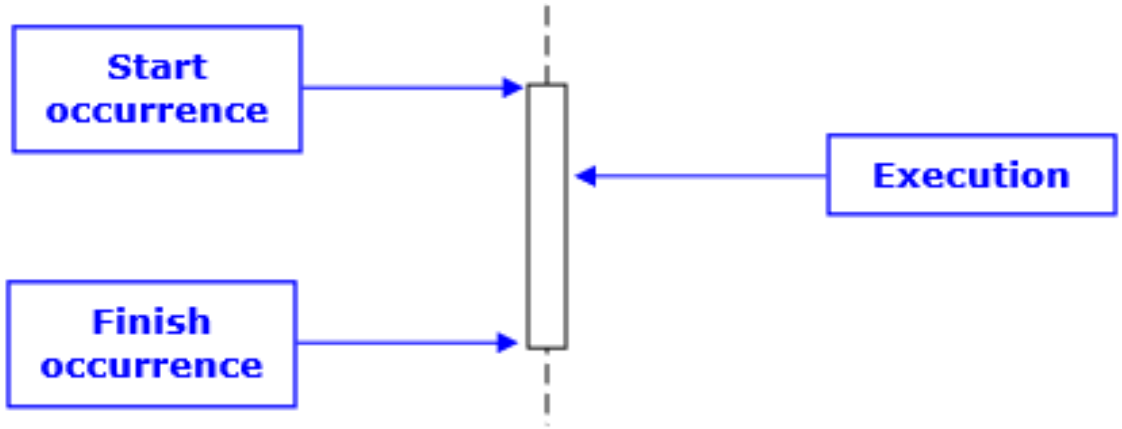
Sequence diagram for View patient information



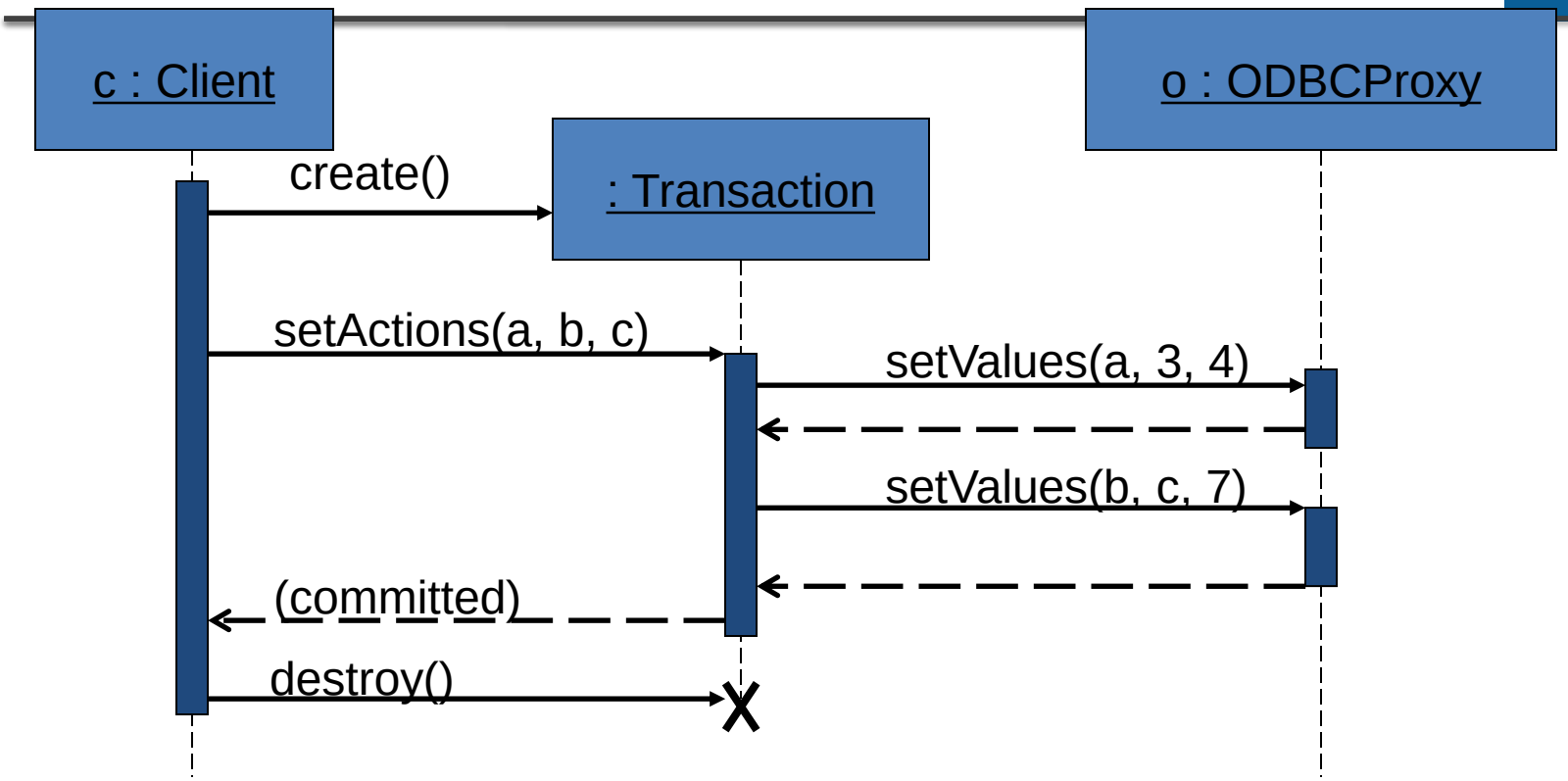

Engineering
Ian Sommerville


Participants
Medical Receptionist
P: PatientInfo
D: Mentcare-DB
AS: Authorization
ViewInfo (PID)
report (Info, PID, UID)
Lifeline refers to the lifetime of an object instance.
authorize (Info, UID)
authorization
alt
[authorization OK]
Patient info
[authorization fail]
Error (no access)

Sequence Model: Life Line



Sequence Model: Message Components



create()
destroy()

Synchronous message

Asynchronous message

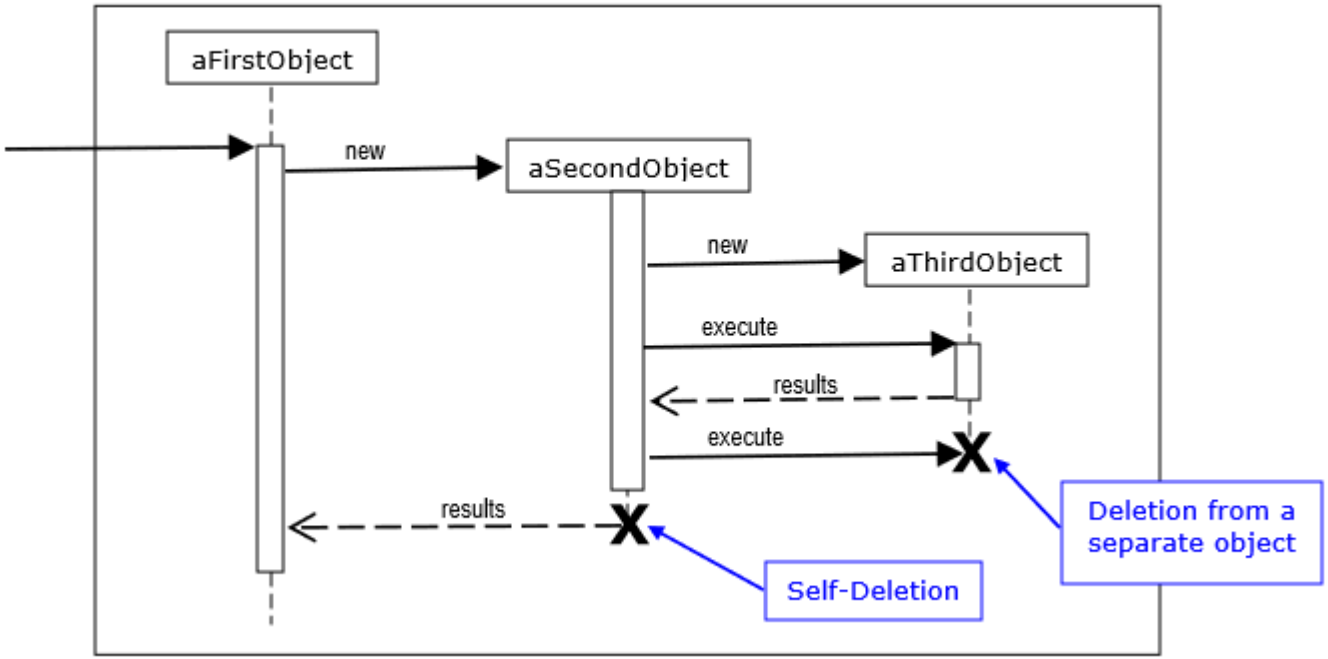
Return message

Communication Types

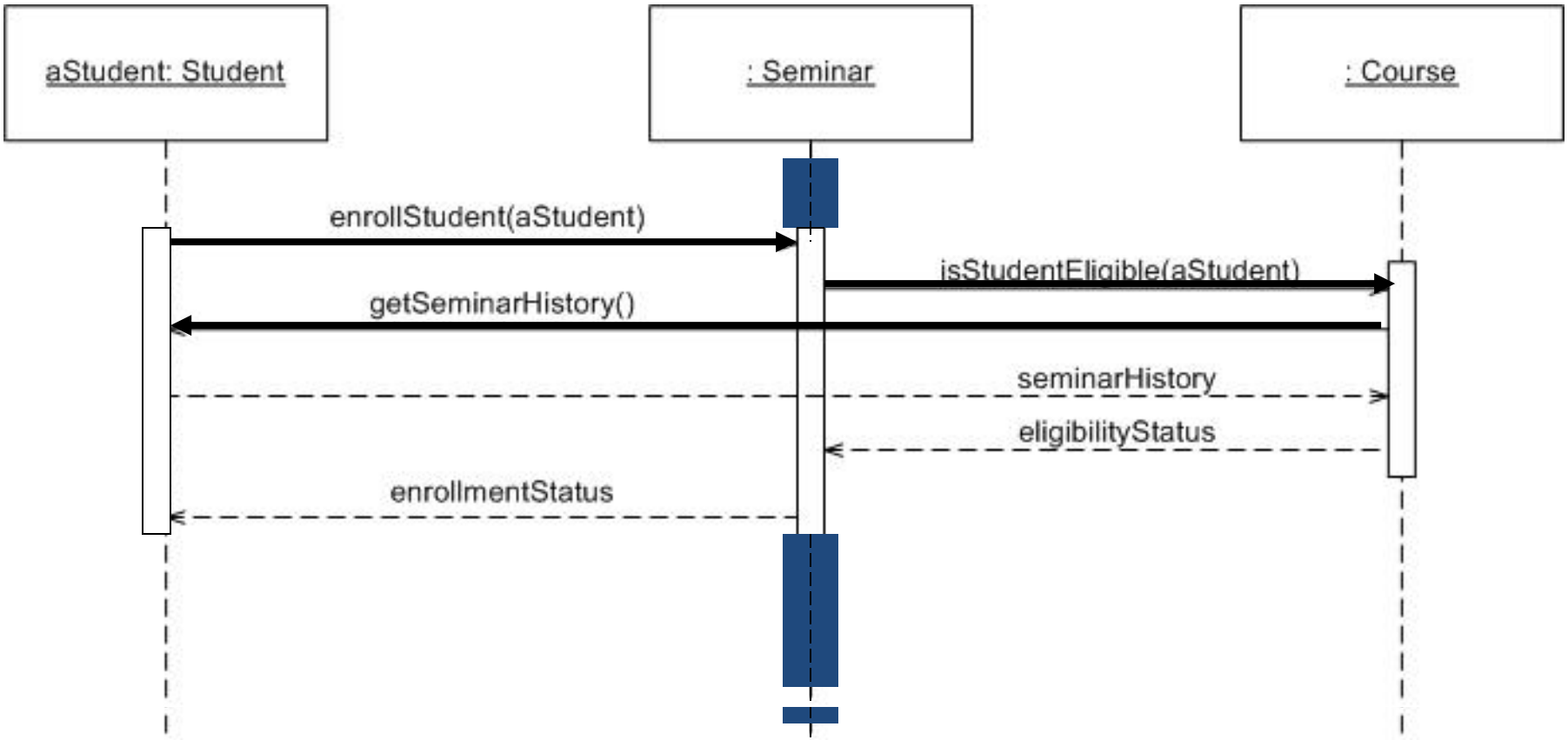
- Synchronous messaging are able to flow in both directions between two components

- Asynchronous messaging are able to flow in one direction [doesn't wait for second party]

Another example of creating/destroying an object

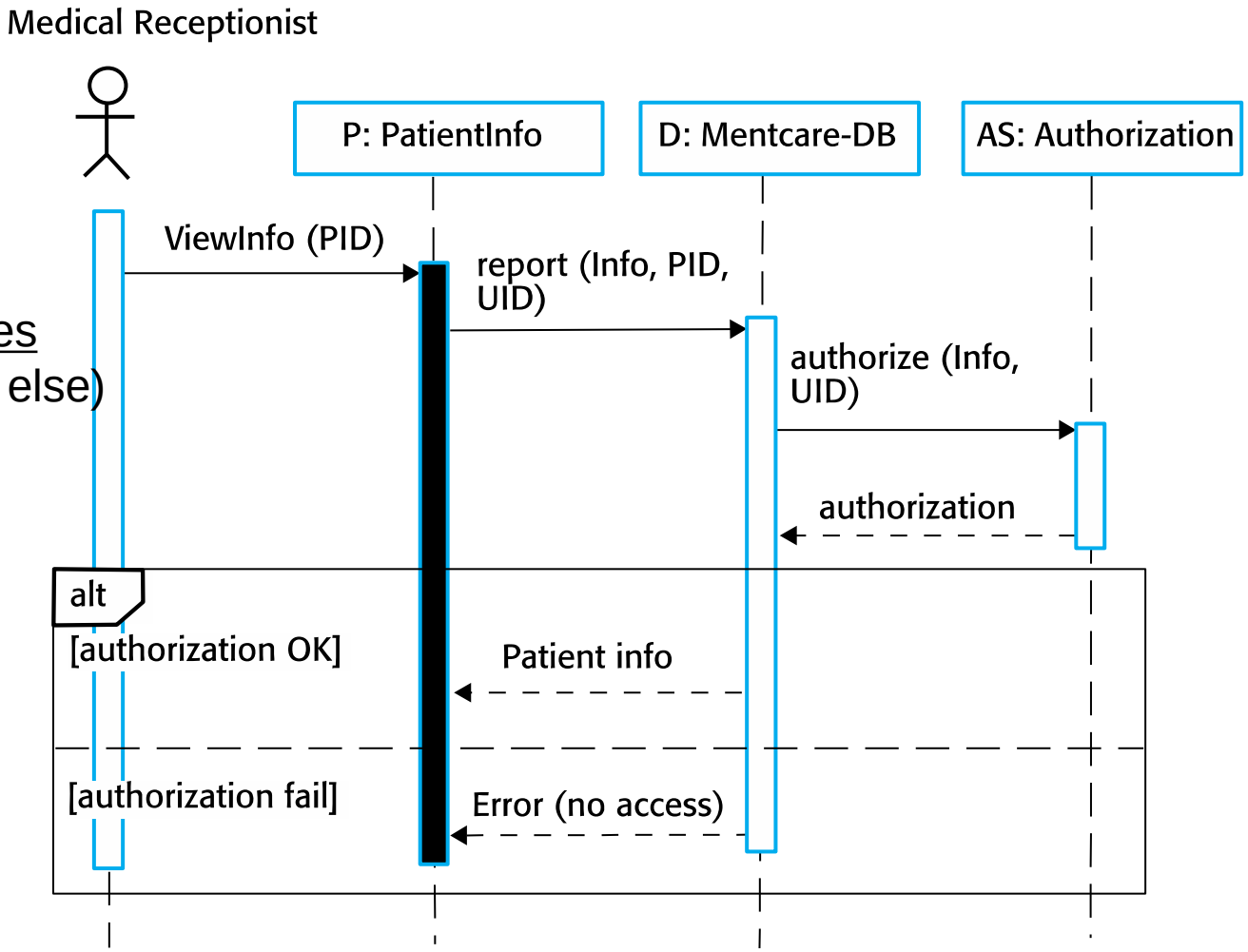


Sequence Diagram: Enroll Student

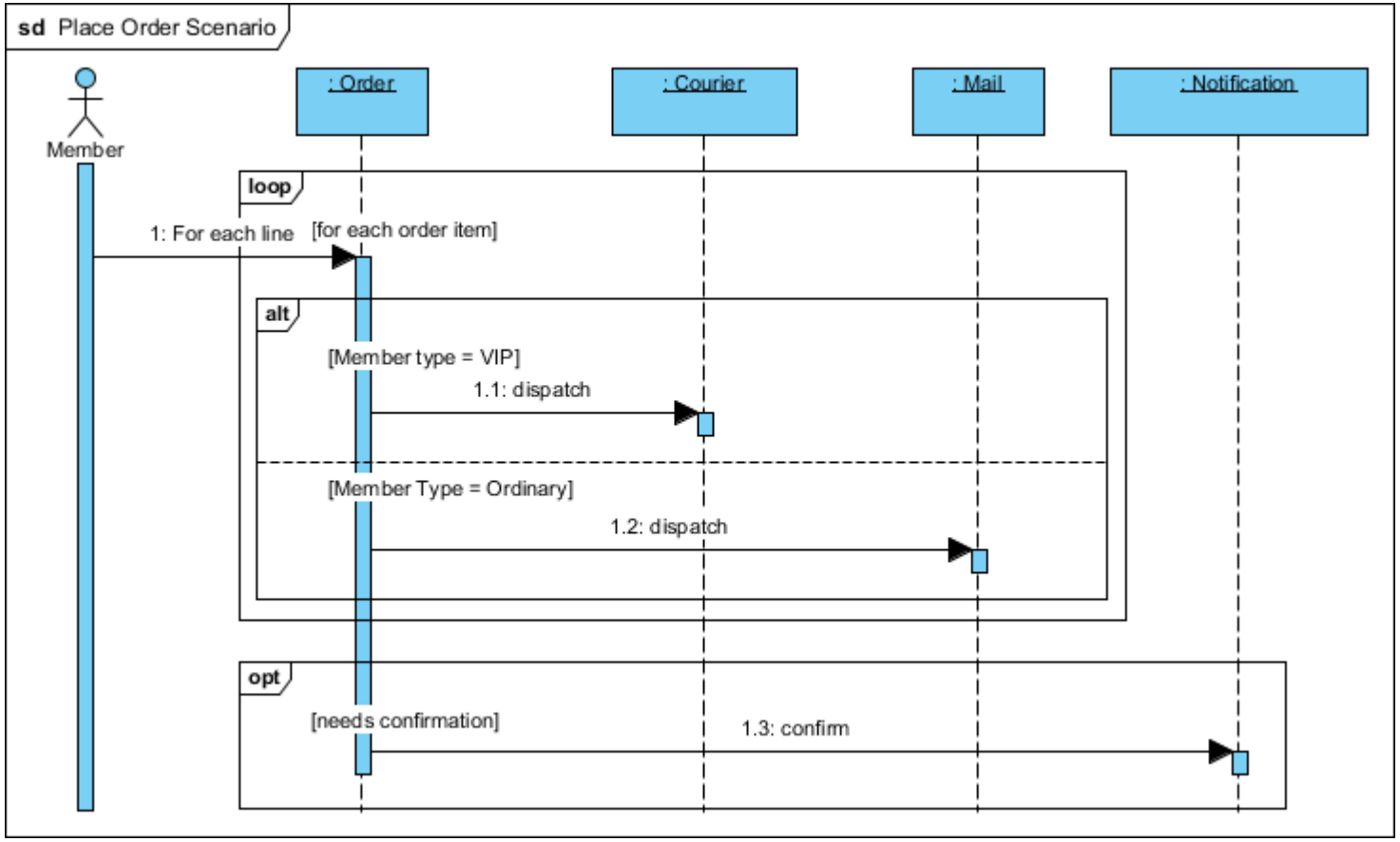


Sequence diagram : Alt: Alternatives

Alt: Alternatives
(example: if & else)



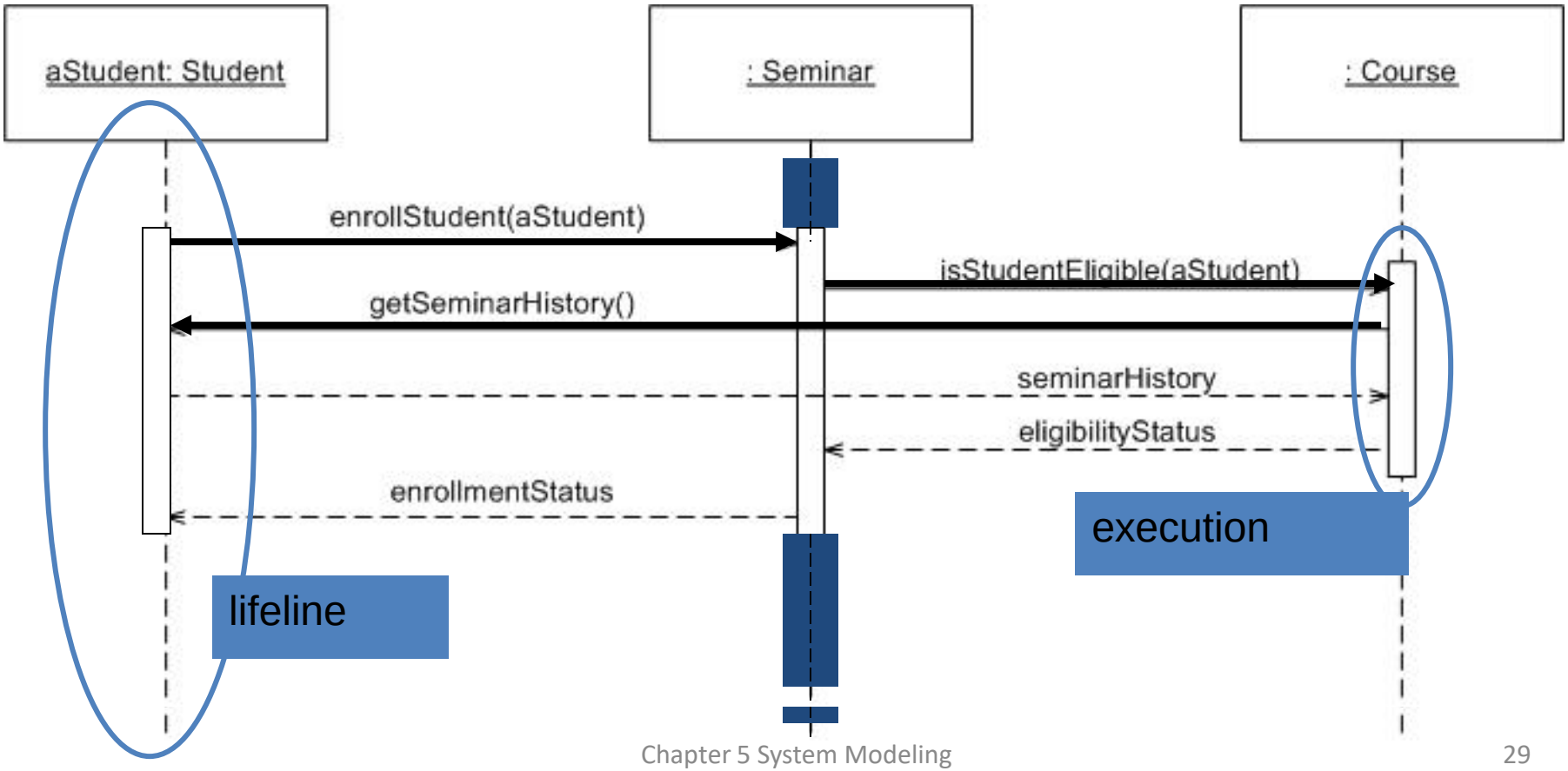
Sequence diagram : Loop & Optional



Optional : mail, text, hard copy

Components


Software Engineering
Ian Sommerville


aStudent: Student
: Seminar
: Course
enrollStudent(aStudent)
isStudentEligible(aStudent)
getSeminarHistory()
seminarHistory
eligibilityStatus
enrollmentStatus
lifeline
execution

Team Activity :Develop a Sequence Model /Design

Case study 1 : Flight Reservation system

✧ User Registration

✧ Purchase flight ticket (use case: book a flight)

Case study 2: Elevator

✧ Press Elevator Button

✧ Press Elevator Button

Case study 3 : Bank ATM

✧ Deposit (ATM)

✧ Withdraw (ATM)

Reference

- Textbook & Prof Boeticher Slides
- [Google] Images
- https://www.uml-diagrams.org/
- Object-Oriented and Classical Software Engineering,8th Edition Author: Stephen R. Schach