Informatics 43

LECTURE 9

"HOW DO WE STRUCTURE THE SOFTWARE IN DETAIL? (PART 1)"

Last lecture

- An architectural style is
  - a named collection of architectural design decisions that result in beneficial qualities in each resulting system
- Styles: model-view-controller, client-server, layered, peer-to-peer, pipe and filter
  (These are not all the styles that exist)
- All successful software evolves
  - We must plan for this
  - Architecture helps us do so

Today's Lecture - **How do we structure the software in detail?**

- Design phase of software engineering
- Designs
- Abstraction

Today's Lecture - **How do we structure the software in detail?**

- Design phase of software engineering
- Designs
- Abstraction

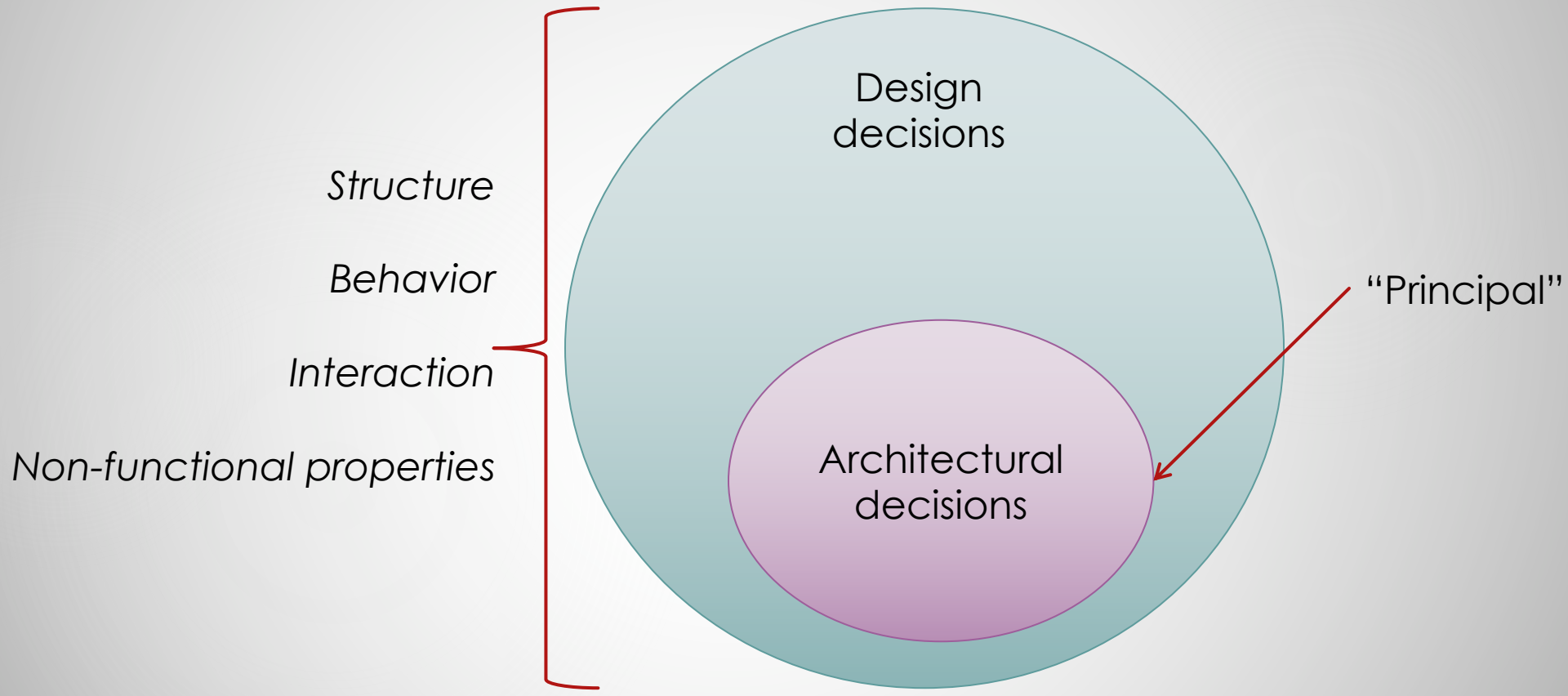
Software design

Requirements

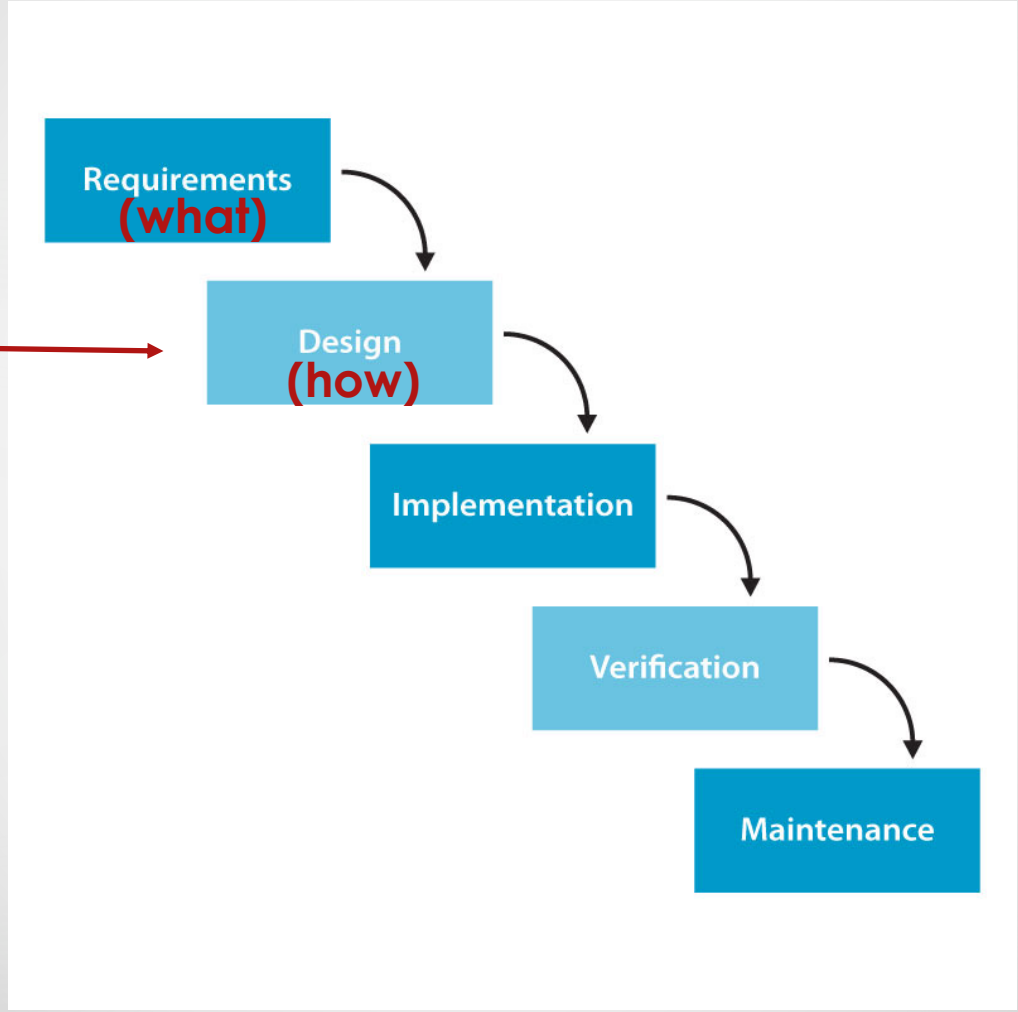
Architecture & **Design**

Code

Software design



Design Phase of Software Engineering



Software Design Goals/Activities

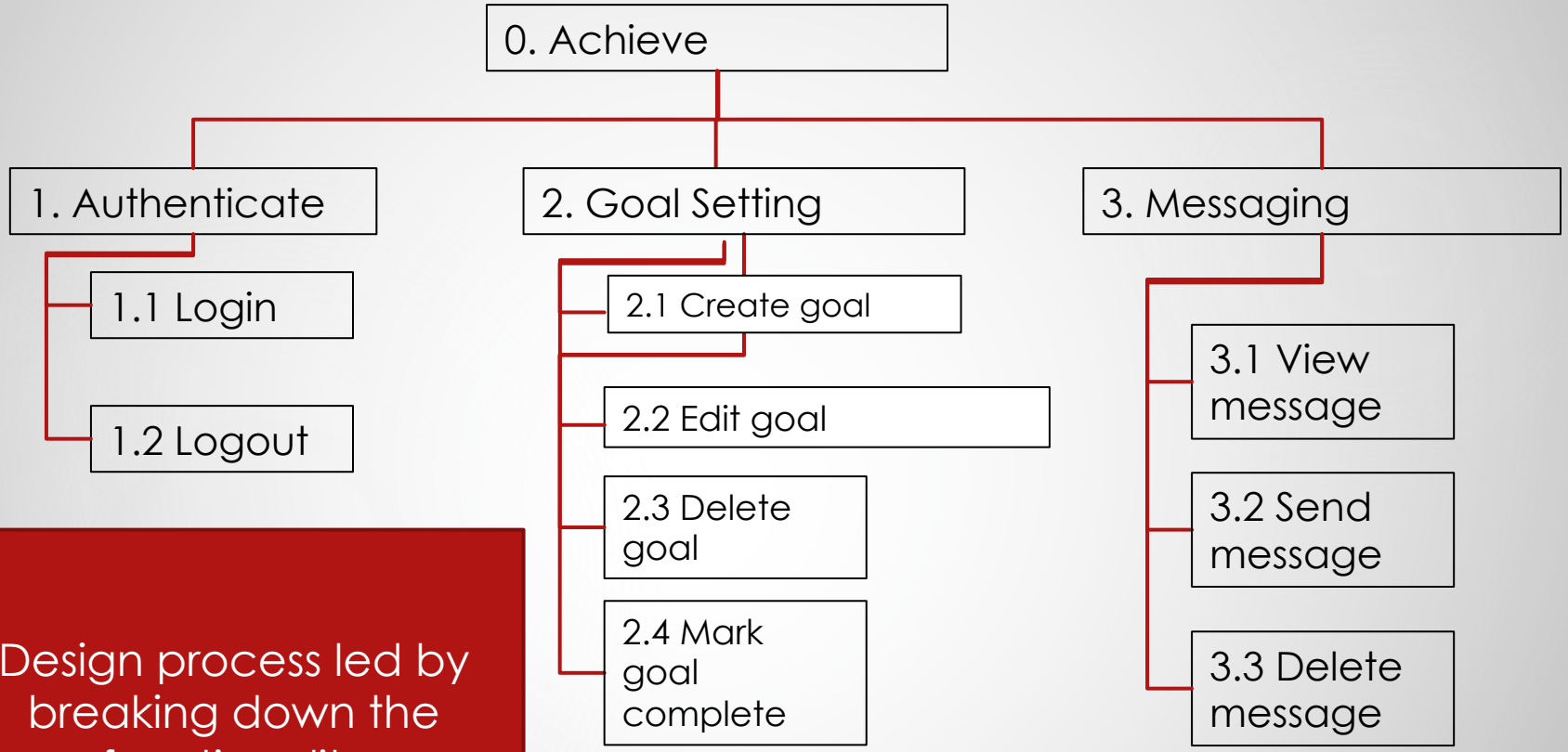
- Making system-wide decisions
  - Architecture, languages, libraries, platforms
- Making lower-level decisions in an *iterative* manner
  - Studying the problem
  - Identifying solutions
  - Creating abstractions
  - Evaluating

Some Approaches to Software Design

- Software architecture
- Functional decomposition
- Relational database design
- Object-oriented design and UML
- User interface design

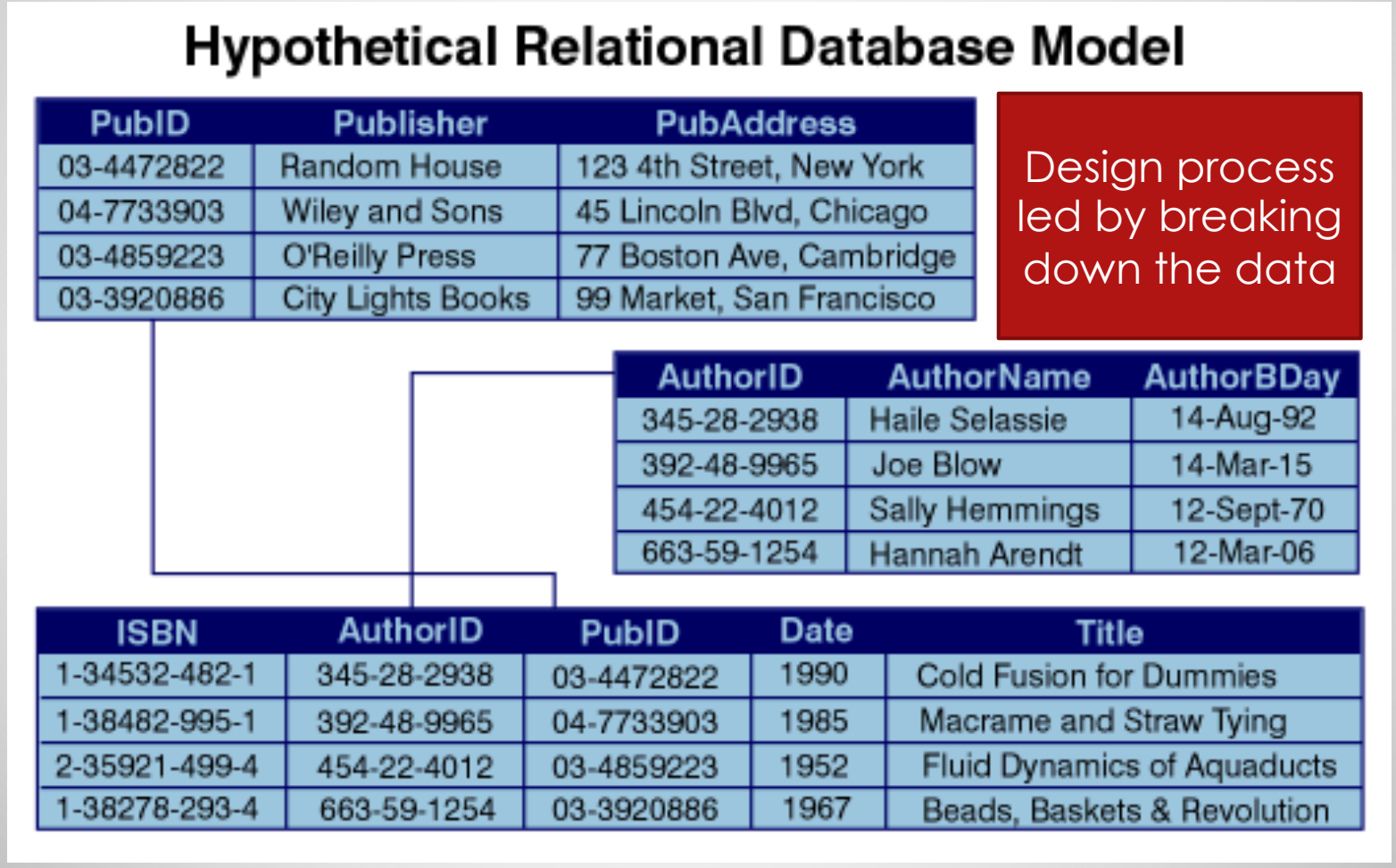
- (Sketching) – Lecture 10

Functional Decomposition



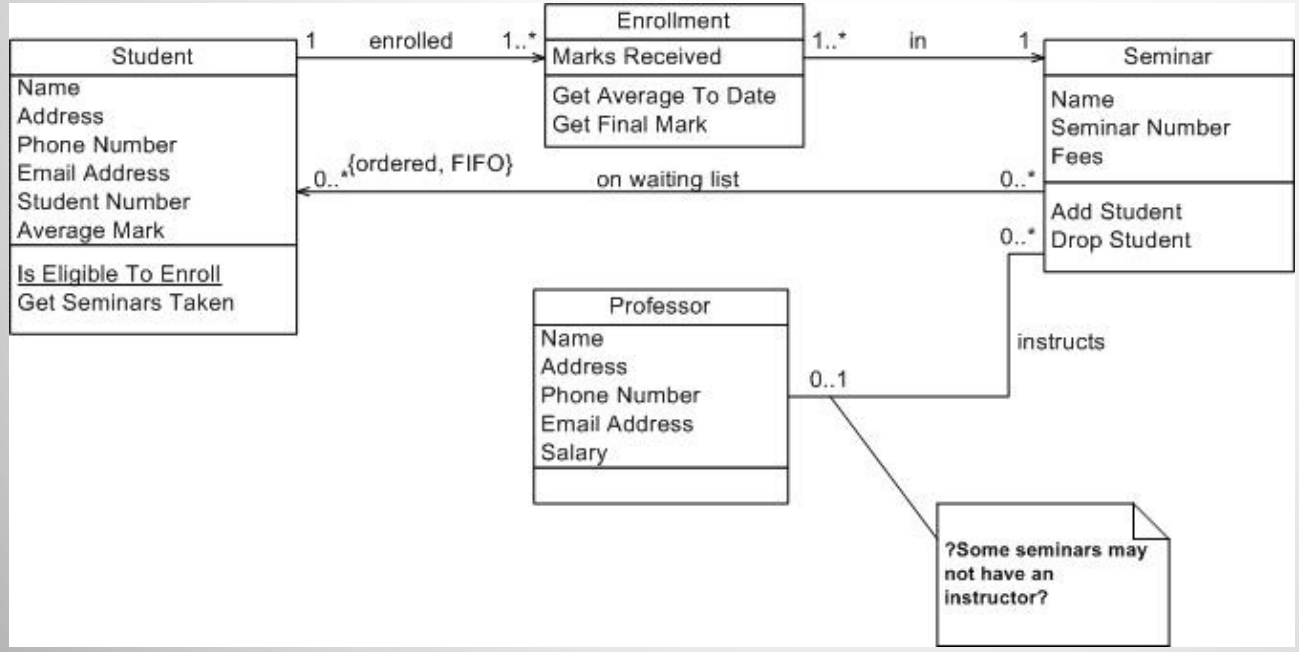
Design process led by breaking down the functionality

Relational Database Design



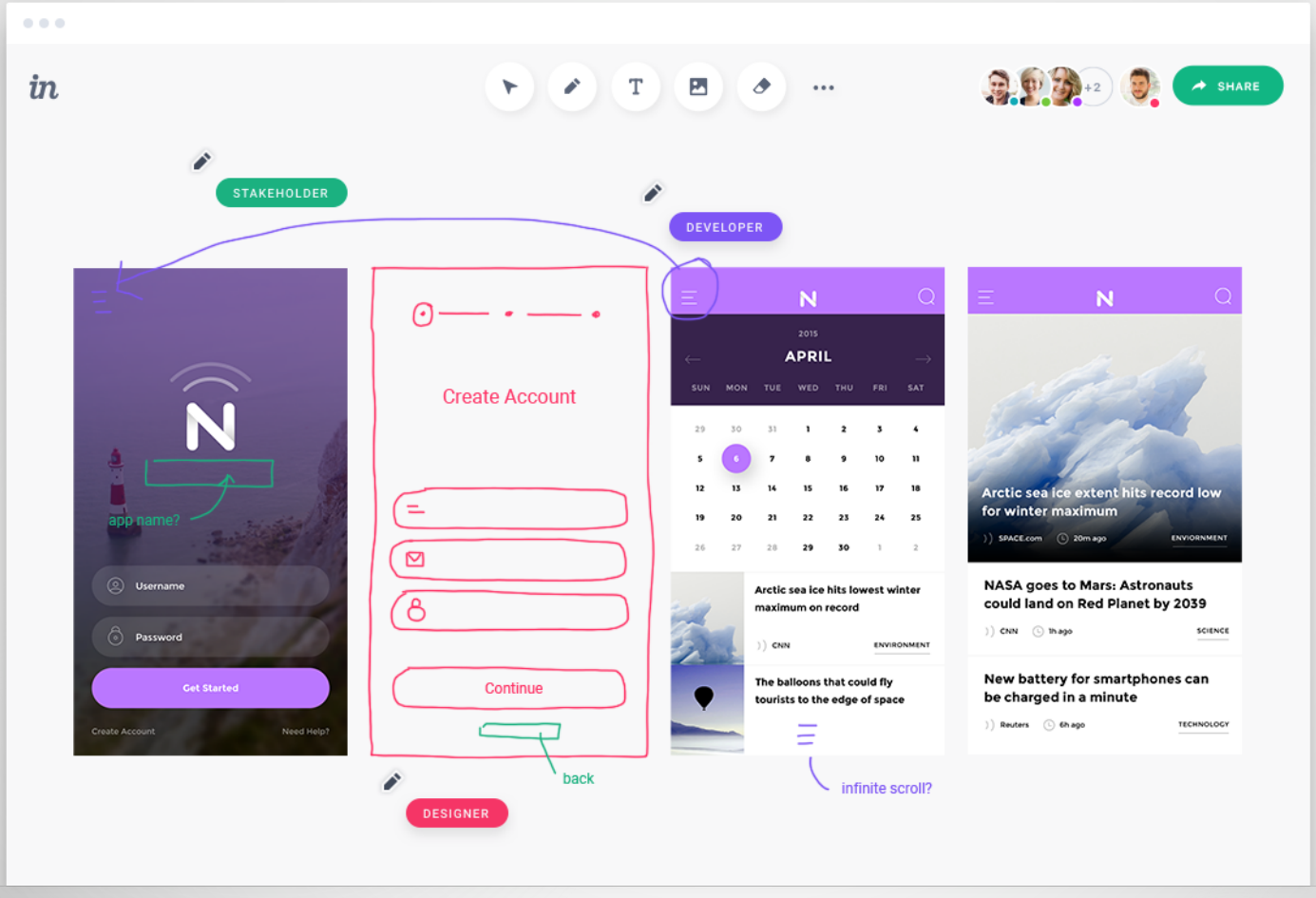
Object-Oriented Design and UML

- An "object" contains both data and methods
- A "class" is a blueprint for making objects



Design process led by breaking down the entities identified in the domain and the functionality that accompanies each entity

User Interface Design

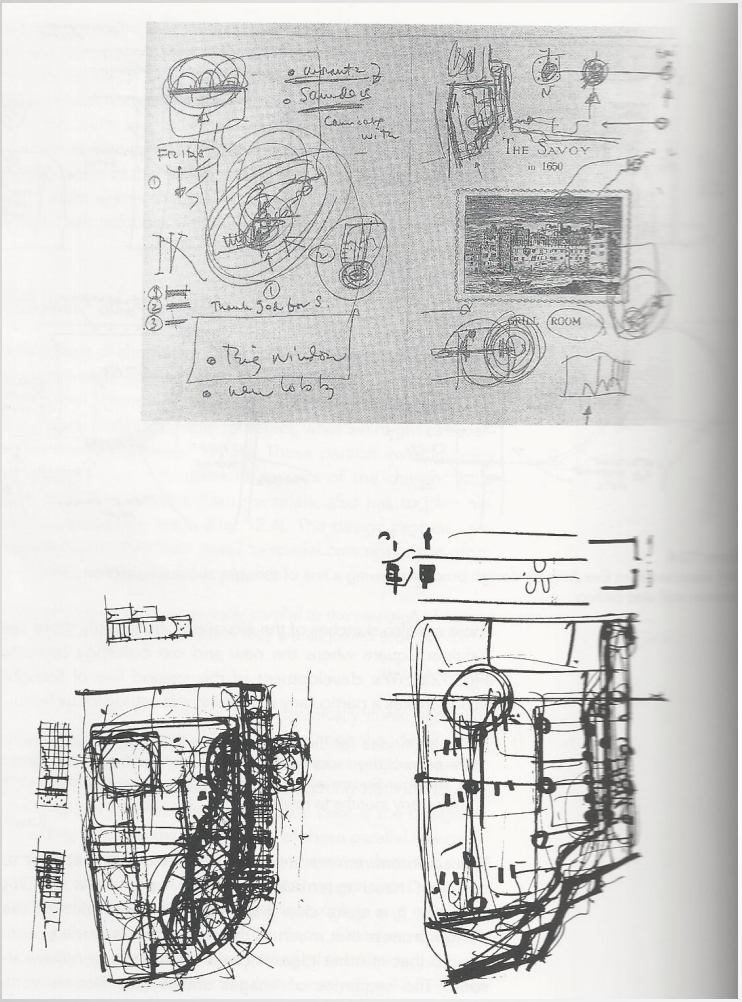


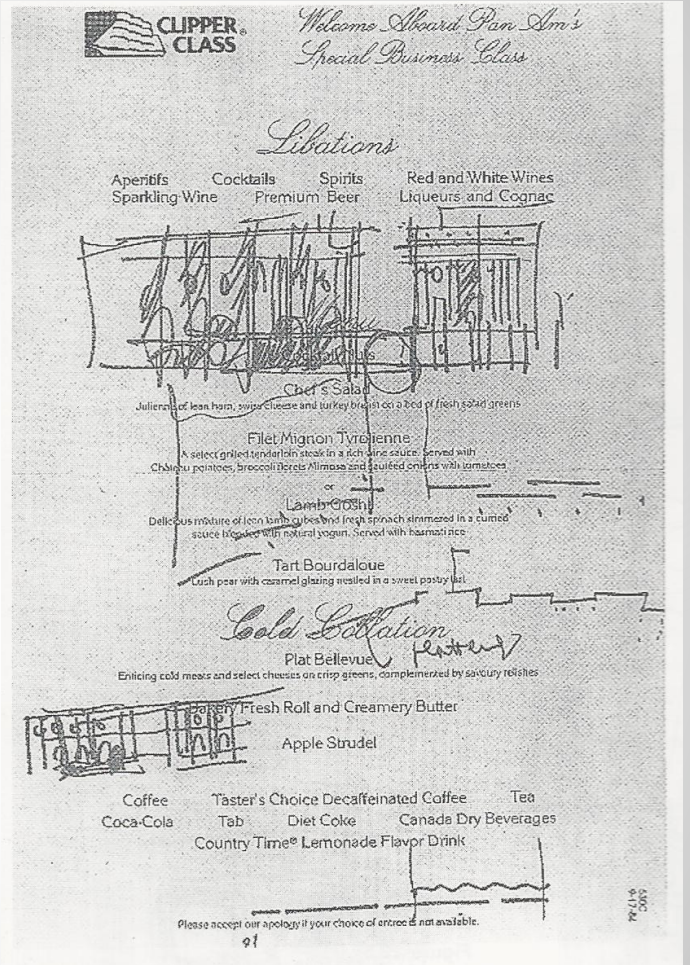
Design process led by envisioning the user interface and iterating

Today's Lecture - **How do we structure the software in detail?**

- Design phase of software engineering
- Designs
- Abstraction

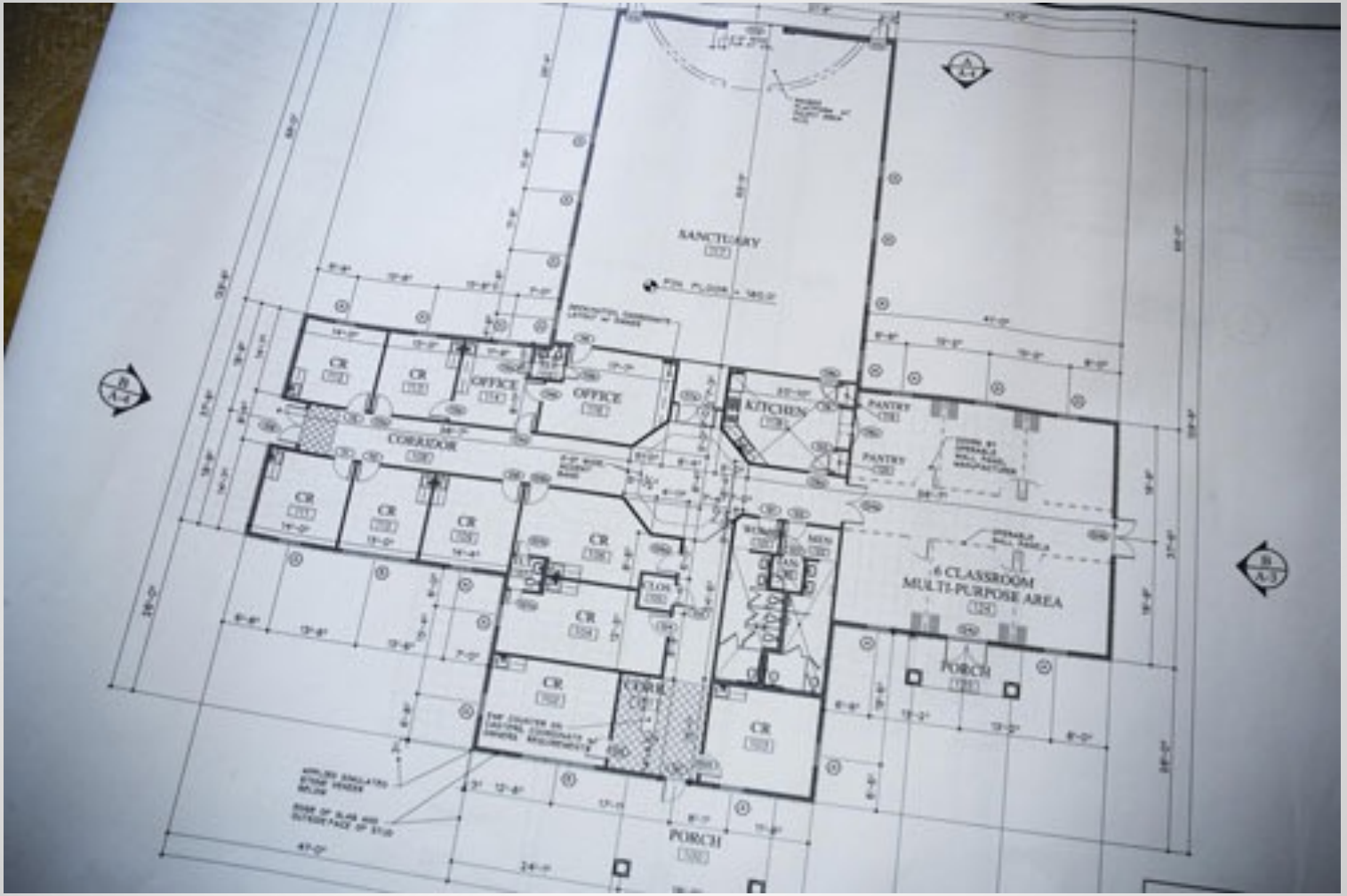
Designs


THE SAVOY
in 1650
GRILL ROOM


CLIPPER CLASS
Welcome Aboard Pan Am's Special Business Class
Libations
Aperitifs Cocktails Spirits Red and White Wines
Sparkling Wine Premium Beer Liqueurs and Cognac
Chef's Salad
Julienne of lean ham, swiss cheese and turkey breast on a bed of fresh salad greens
Filet Mignon Tyrolienne
A select grilled tenderloin steak in a rich wine sauce. Served with
Chateau potatoes, Broccoli Bercis Mimosa and sautéed onions with tomatoes
or
Lamb Coshy
Delicious mixture of lean lamb cubes and fresh spinach simmered in a curried
sauce blended with natural yogurt. Served with basmati rice
Tart Bourdaloue
Lush pear with caramel glazing nestled in a sweet pastry tart
Cold Collation
Plat Bellevue
Enticing cold meats and select cheeses on crisp greens, complimented by savoury relishes
Bakery Fresh Roll and Creamery Butter
Apple Strudel
Coffee Taster's Choice Decaffeinated Coffee Tea
Coca-Cola Tab Diet Coke Canada Dry Beverages
Country Time® Lemonade Flavor Drink
Please accept our apology if your choice of entree is not available.
91

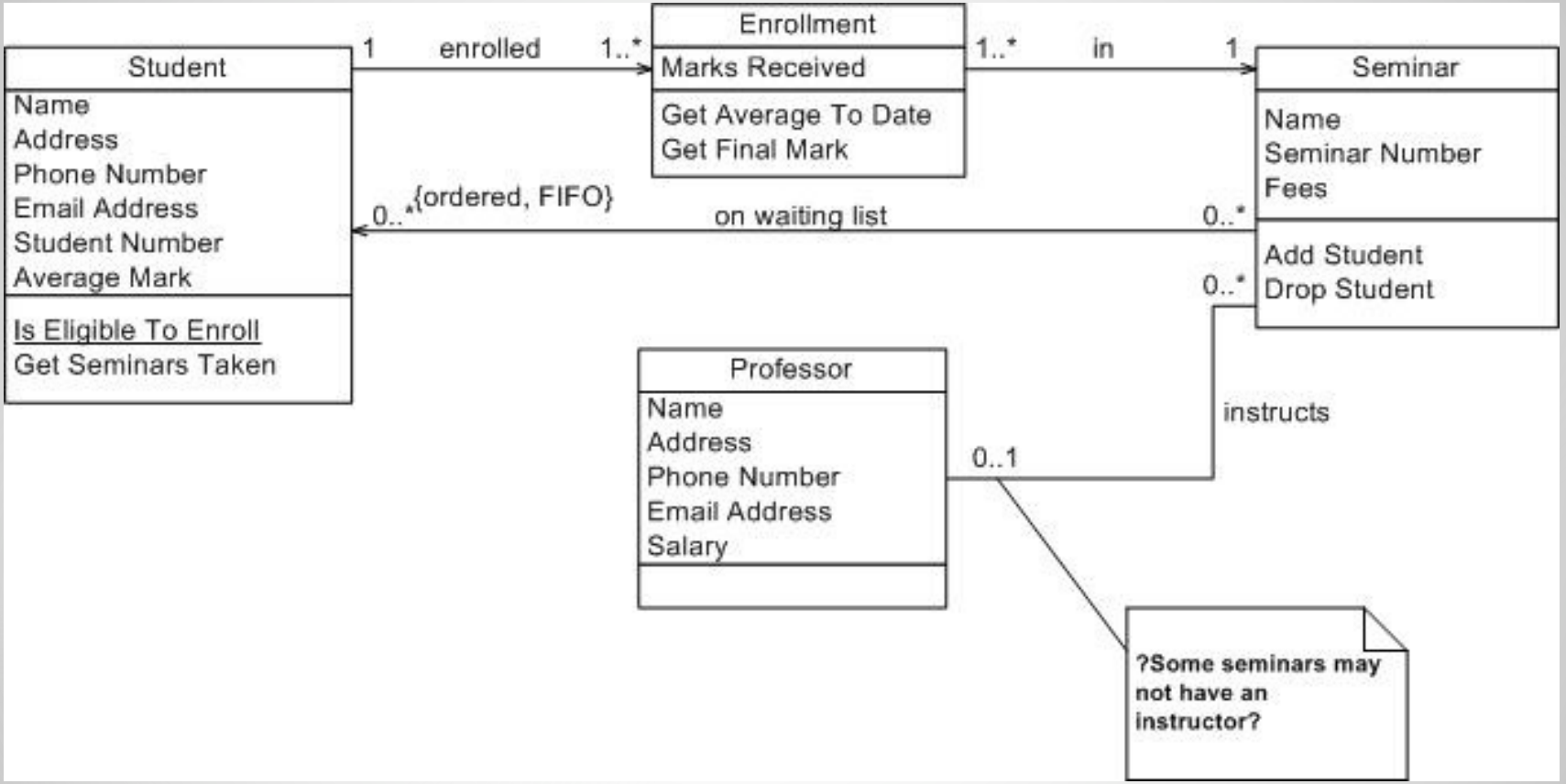
Designs

Designs


SANCTUARY
OFFICE
OFFICE
KITCHEN
PANTRY
PANTRY
CORRIDOR
6 CLASSROOM
MULTI-PURPOSE AREA
PORCH
PORCH
MEN
WOMEN

Designs/Models

Designs/Models



Purpose of designs

- Designs to *think*
- Designs to *talk*
- Designs to *prescribe*

*Designs are developed **iteratively***

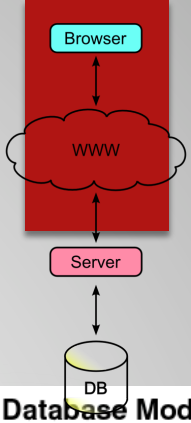
Today's Lecture - **How do we structure the software in detail?**

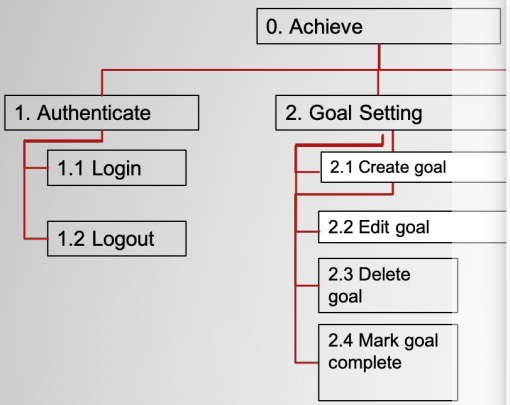
- Design phase of software engineering
- Designs
- Abstraction

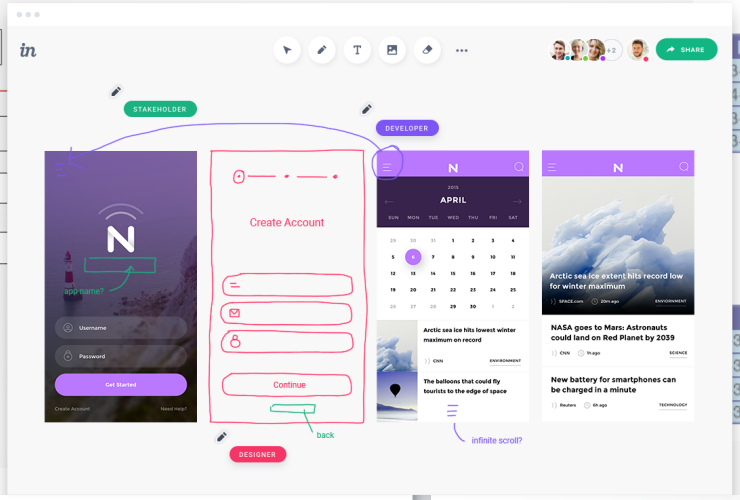
Abstraction

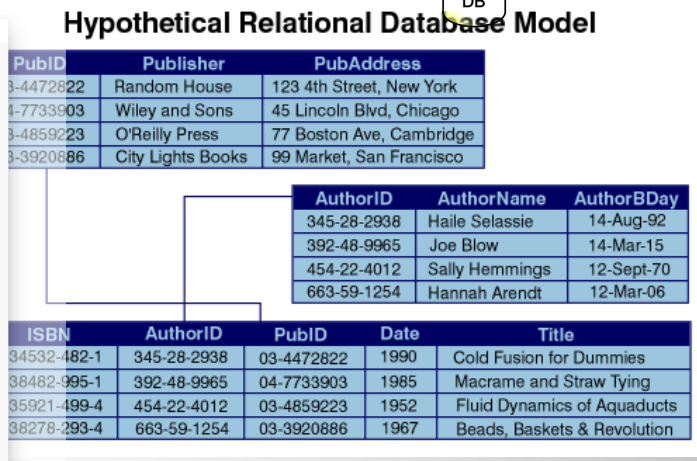
- Abstractions are formed by removing irrelevant information and retaining/highlighting relevant information
- Every design notation supports a certain kind of abstraction
- Software engineering is all about constructing and elaborating abstractions/models

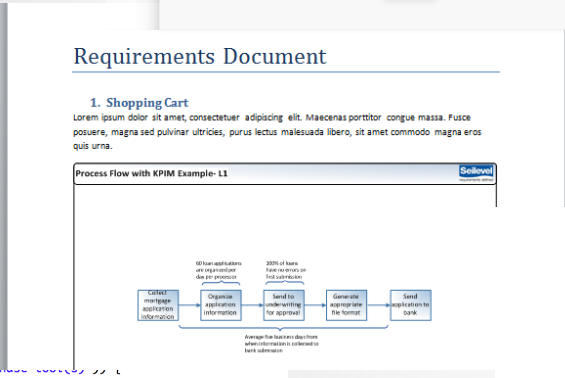
What do each of these highlight/abstract away?



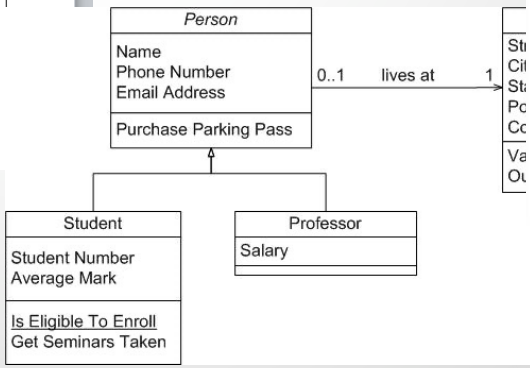


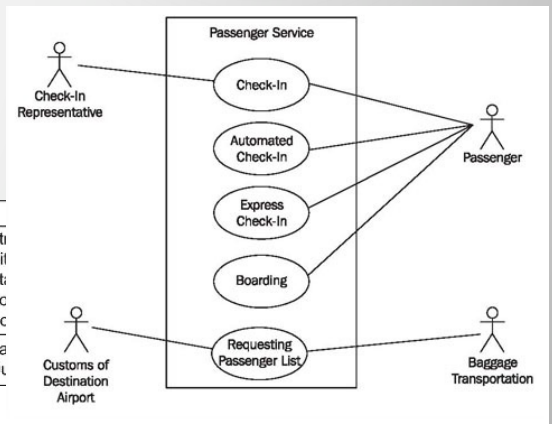







```
470             state.getActionStateRepository()
471                     .add((FireAction) action
472             ruleExec.update(gui, RuleExecuto
473                     "FireDestroyObjectsRuleA
474             destChecker.update(false, gui);
475         } else if (action instanceof Purchas
476             Vector<SSObject> participants =
477             for (int i = 0; i < participants
478                 SSObject obj = participants.
479                 if (obj instanceof Employee)
480                     if (menuText.equals("Pur
481                         ((Employee) obj)
482                                 .setOverhead
483                     }
484                 } else if (obj instanceof Cu
485                     if (menuText.equals("Pur
486                         ((Customer) obj)
487                                 .setOverheadText("Tool(s) have been purchased!");
488                     }
489                 }
490             }
491             state.getActionStateRepository()
492                     .getPurchaseToolActionStateRepository()
493                     .add((PurchaseToolAction) action);
494             ruleExec.update(gui, RuleExecutor.UPDATE_ONE,
495                     "PurchaseToolEffectRuleA", action);
496             destChecker.update(false, gui);
497         }
```





Attendance Quiz

Abstraction of Software Engineering Artifacts

Summary

- Design phase of software engineering
  - The "how" to the "what" of requirements
  - Architecture, functional decomposition, relational database design, OO design/UML, UI design, sketching
- Designs are used iteratively to think, talk, and prescribe
- Software engineering is all about constructing and elaborating abstractions/models

Next time

- Designs, Models, and Notations, Part 2