

# CSE4006: Software Engineering

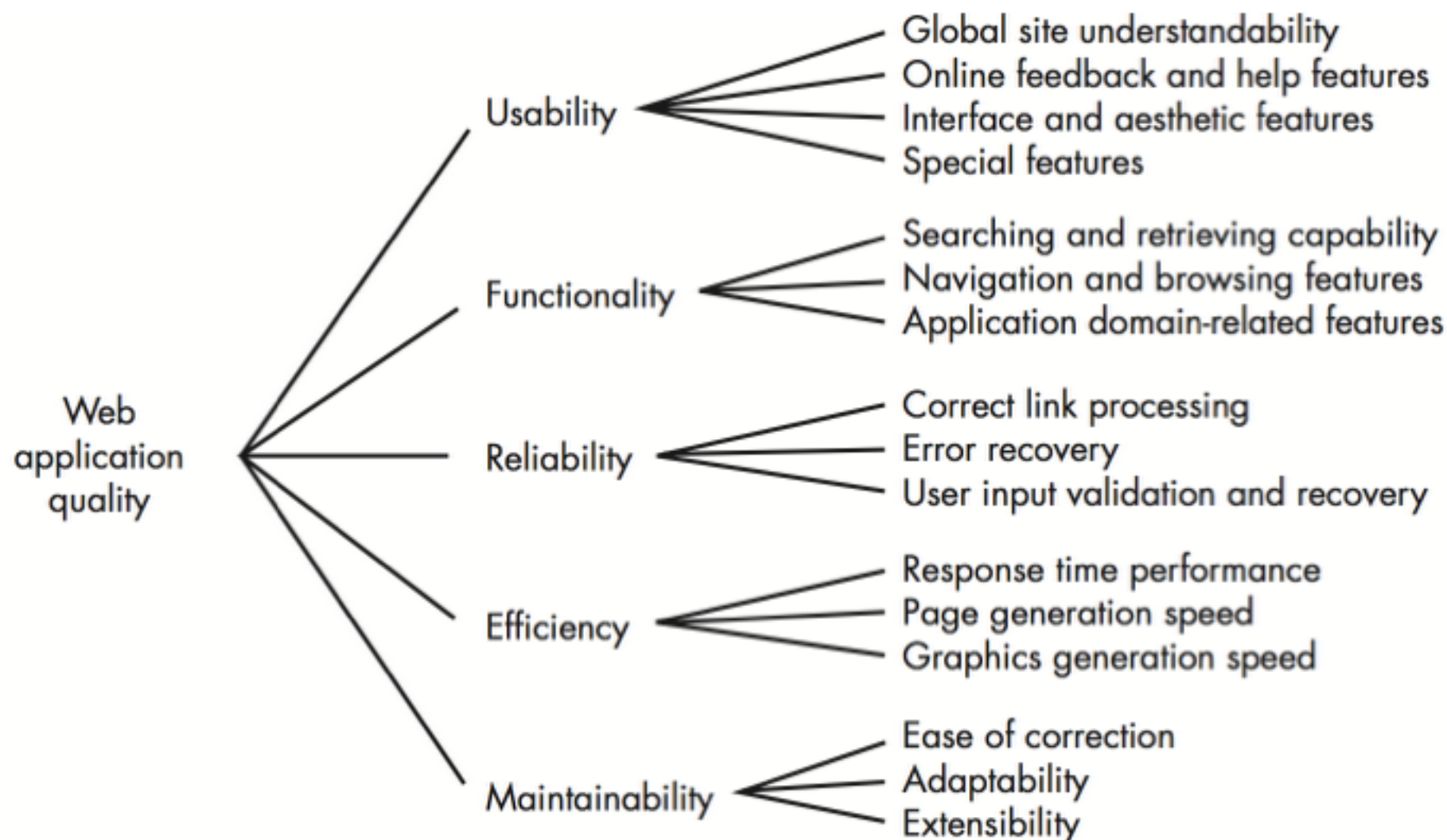
## Lab 9: WebApp Design

### Software Engineering Lab

Except where otherwise noted, the contents of this document are Copyright 2015 Junghoon Lee, Gayeon Kim, Scott Uk-Jin Lee. All rights reserved. Any redistribution, reproduction, transmission, or storage of part or all of the contents in any form is prohibited without the author's expressed written permission.

# WebApp Design Quality

- All of the technical characteristics of the generic quality
- attributes. However, the most relevant of these generic attributes
  - usability, functionality, reliability, efficiency, and maintainability



# WebApp Design Quality (cont)

## Security

- Its server environment to rebuff unauthorized access and block an malicious attack.

## Availability

- The measure of the percentage of time that a WebApp is available for use.

## Scalability

- Will the WebApp and the systems with which it is interfaced handle significant variation in volume or will responsiveness drop dramatically ?

## Time-to-Market

- Not a true quality attribute in the technical sense, it is a measure of quality from a business point of view.

# WebApp Design Goal

## Simplicity

- Content should be informative but succinct and should use a delivery mode that is appropriate to the information that is being delivered.
- Aesthetics should be pleasing, but not overwhelming.

## Consistency

- This design goal applies to virtually every element of the design model. Content should be constructed consistently.(font, color scheme, and style).

## Identity

- The WebApp architecture will be entirely different for each application domain.

# WebApp Design Goal (cont)

## Robustness

- The user expects robust content and functions that are relevant to the user's needs.
- If these elements are missing or insufficient, it is likely that the WebApp will fail.

## Navigability

- It should also be designed in a manner that is intuitive and predictable.

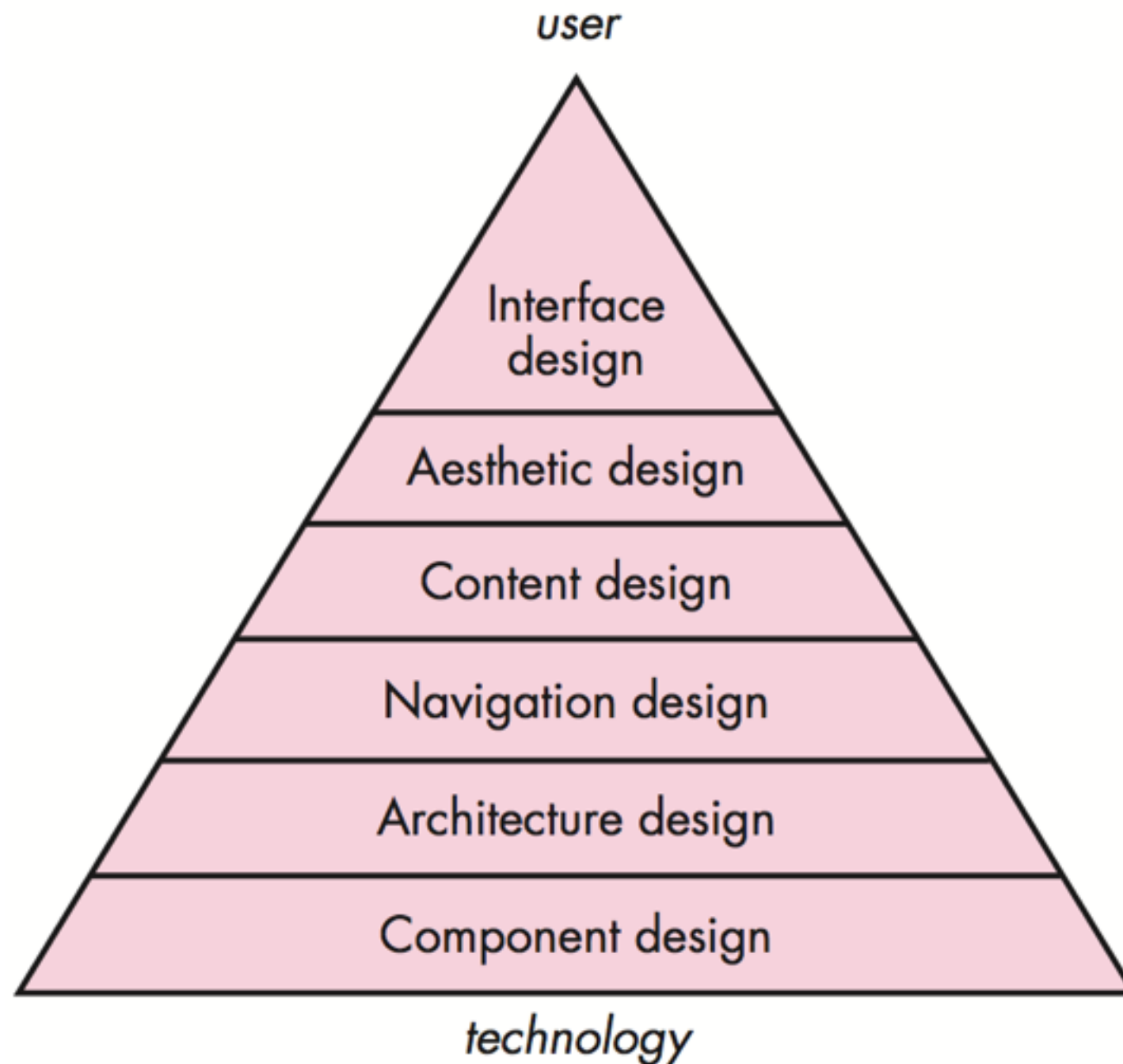
## Visual Appeal

- The look and feel of content; interface layout; color coordination; the balance of text, graphics, and other media; navigation mechanisms.

## Compatibility

- A WebApp will be used in a variety of environments and must be designed to be compatible with each.

# A Design pyramid for WebApps



# WebApp Interface Design

## Goal of a WebApp interface

- Establish a consistent window into the content and functionality provided by the interface.
- Guide the user through a series of interactions with the WebApp.
- Organize the navigation options and content available to the user.

# Aesthetic Design

## Layout Issues

- Don't be afraid of white space.
- Emphasize content.
- Organize layout elements from top-left to bottom-right.
- Group navigation, content, and function geographically within the page.
- Don't extend your real estate with the scrolling bar.
- Consider resolution and browser window size when designing layout.

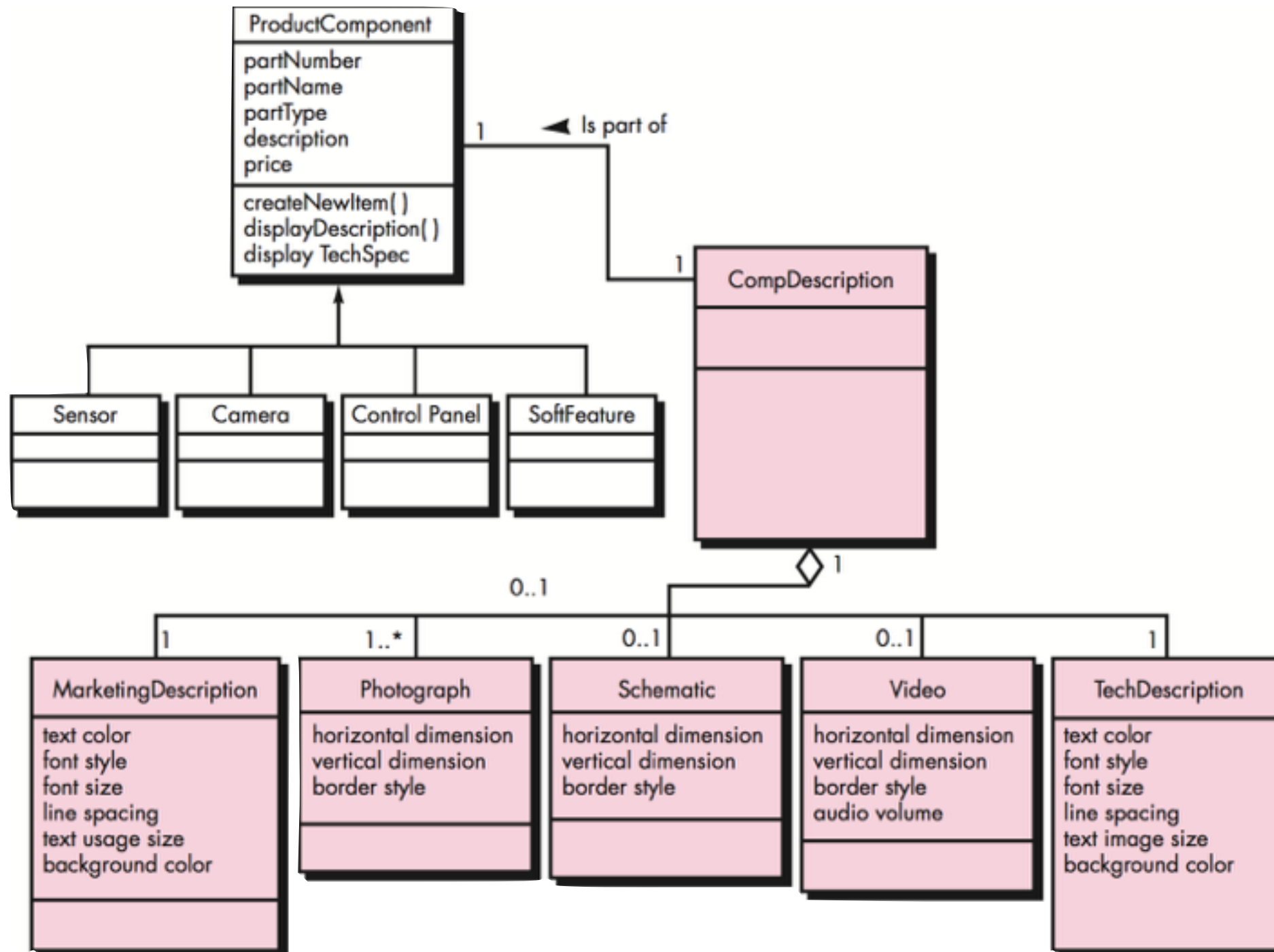


# Content Design

The relationship between content objects defined as part of a requirements model for the WebApp and design objects representing content is analogous to the relationship between analysis classes and design components described

# Content Design (cont)

## Design representation of content objects



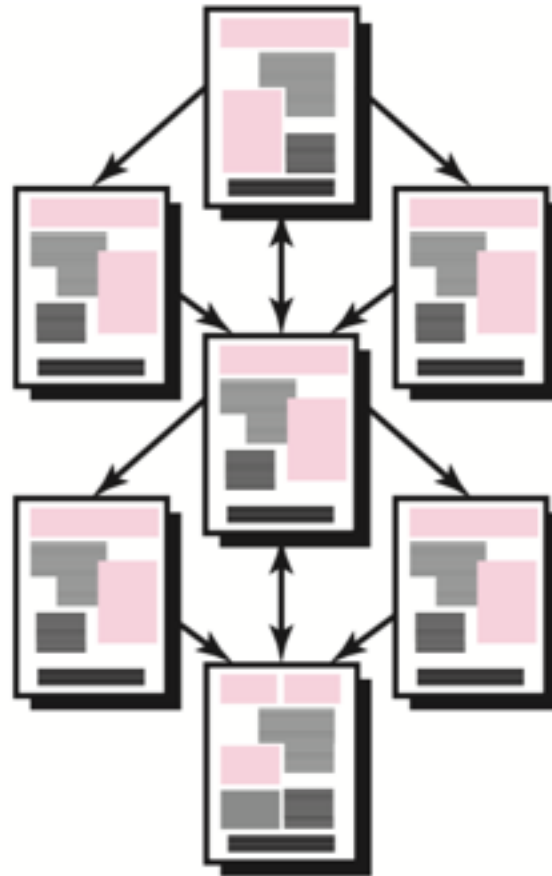
# Architecture Design

## Linear Structures

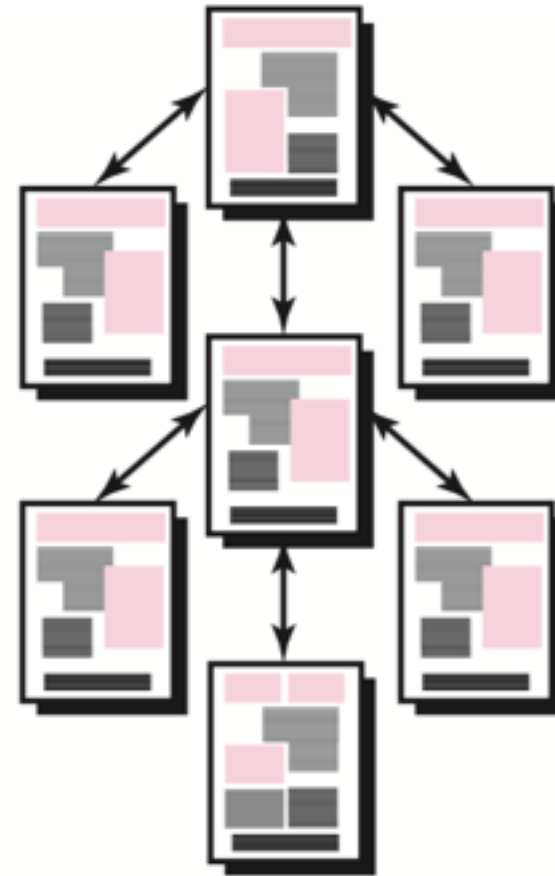
- Linear structures are encountered when a predictable sequence of interactions is common



Linear



Linear  
with  
optional flow

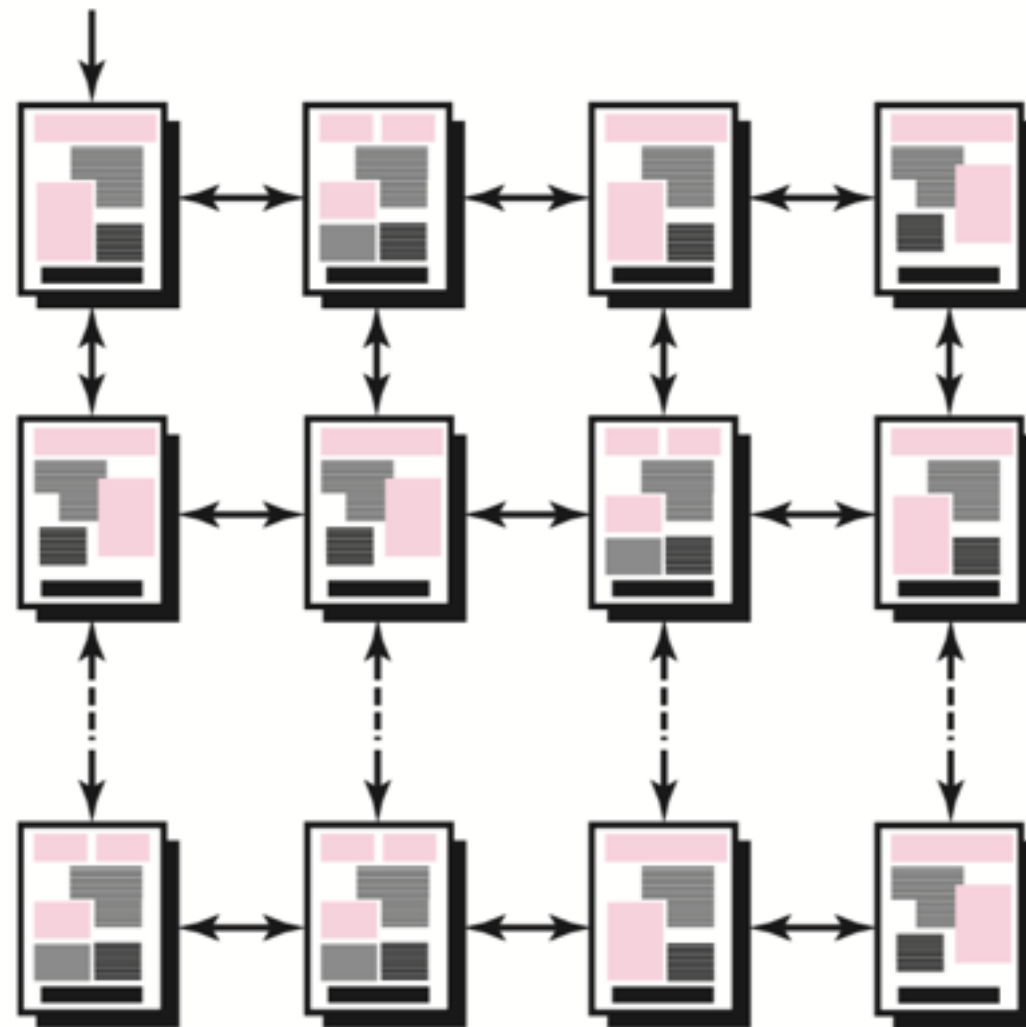


Linear  
with  
diversions

# Architecture Design (cont)

## Grid Structures

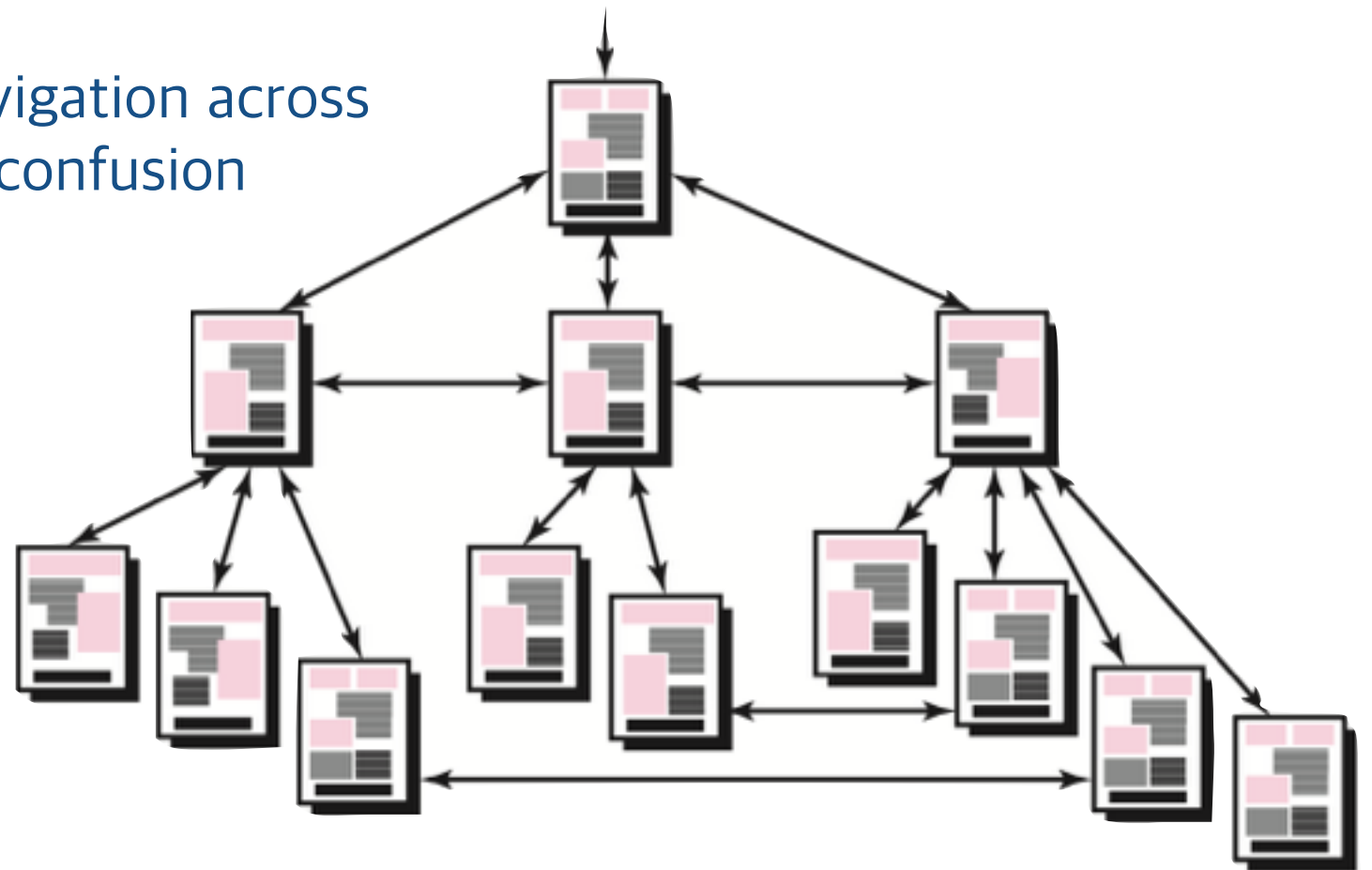
- Grid structures are an architectural option that you can apply when WebApp content can be organized categorically in two (or more) dimensions.



# Architecture Design (cont)

## Hierarchical Structures

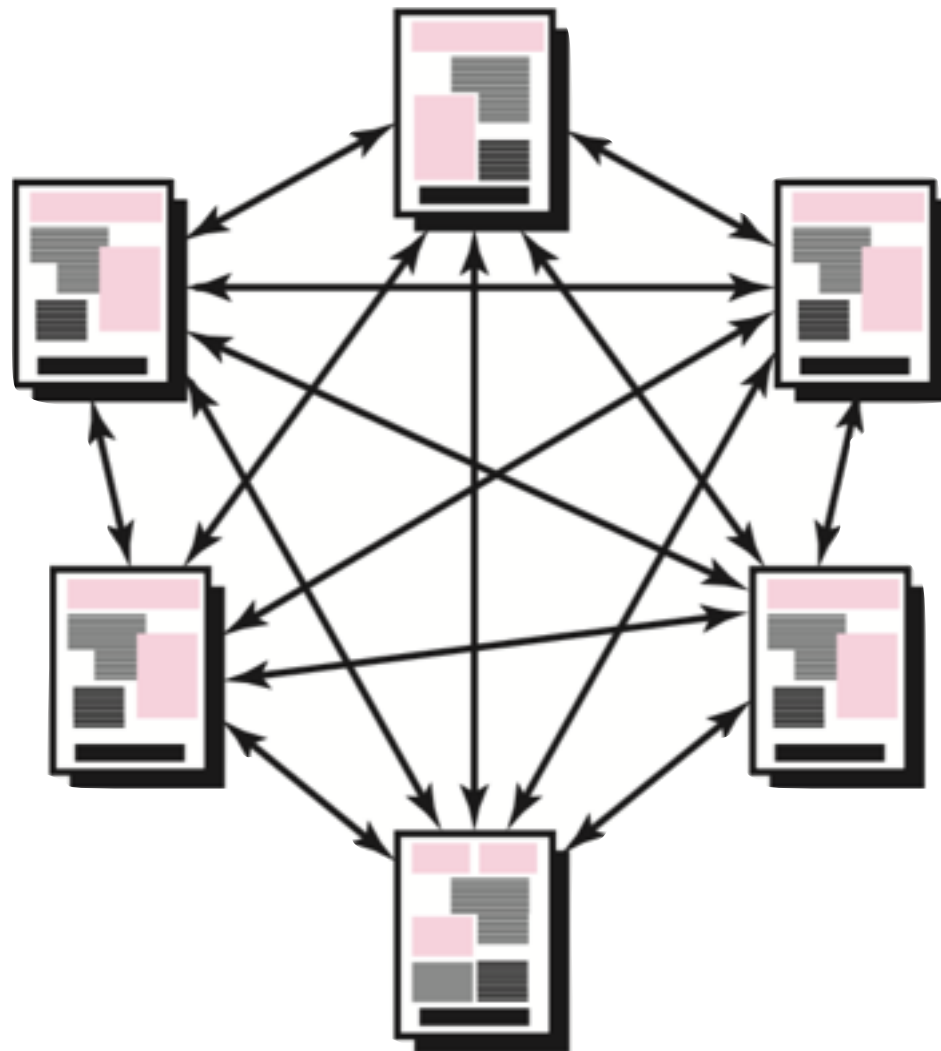
- Hierarchical structures are undoubtedly the most common WebApp architecture.
- A WebApp hierarchical structure can be designed in a manner that enables (via hypertext branching) flow of control horizontally across vertical branches of the structure.
- such branching allows rapid navigation across WebApp content, it can lead to confusion on the part of the user.



# Architecture Design (cont)

## Network Structures (pure web)

- This approach allows considerable navigation flexibility, but at the same time, can be confusing to a user.



# Architecture Design (cont)

## MVC

### Models

- The model contains all application-specific content and processing logic, including all content objects, access to external data/information sources, and all processing functionality that is application specific.

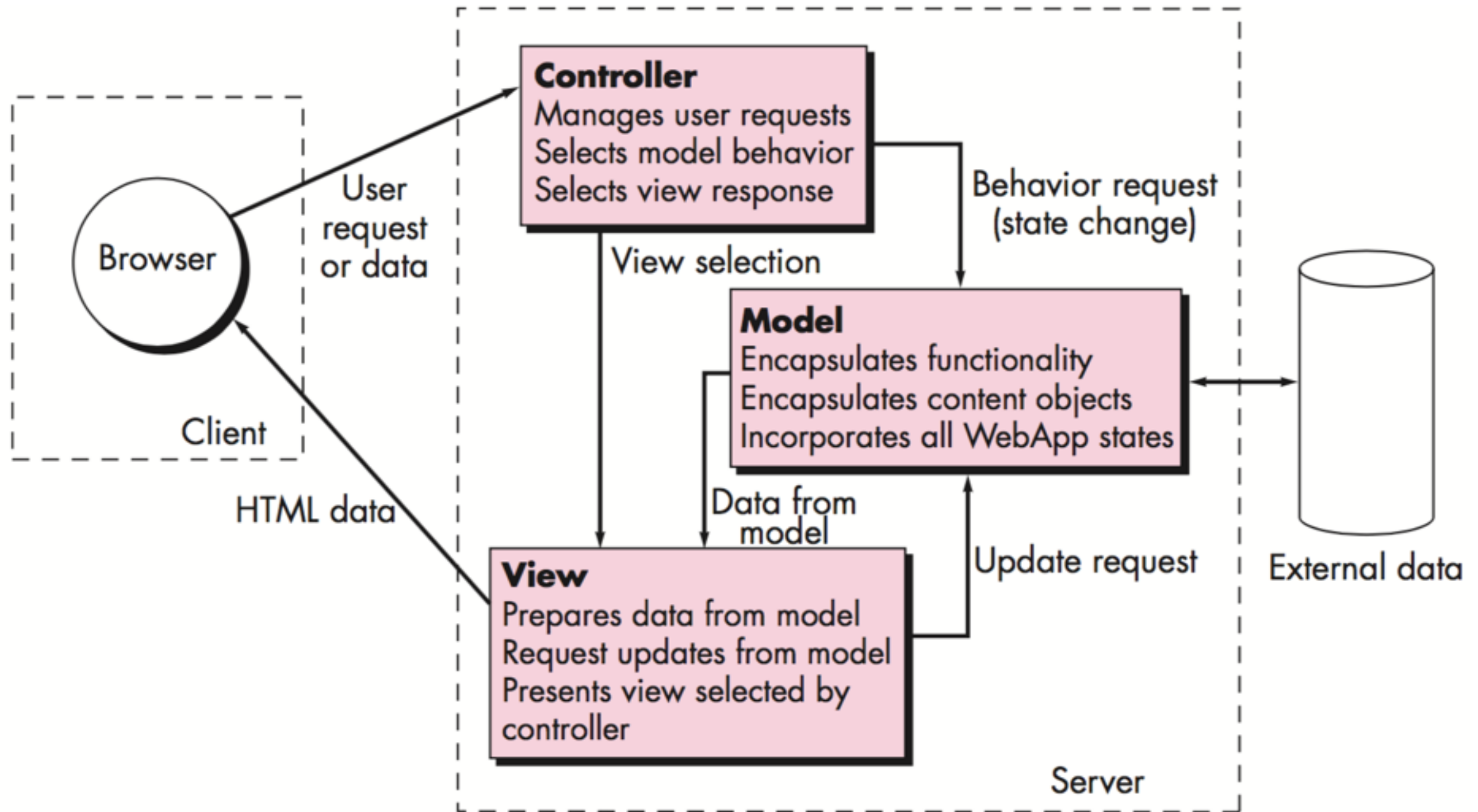
### View

- The view contains all interface- specific functions and enables the presentation of content and processing logic, including all content objects, access to external data/information sources, and all processing functionality required by the end user.

### Controler

- The controller manages access to the model and the view and coordinates the flow of data between them.

# Architecture Design (cont)





# Navigation Design

## Navigation Syntax

- **Individual navigation link**
  - includes text-based links, icons, buttons and switches, and graphical metaphors. You must choose navigation links that are appropriate for the content and consistent with the heuristics that lead to high-quality interface design.
- **Horizontal navigation bar**
  - lists major content or functional categories in a bar containing appropriate links. In general, between four and seven categories are listed.
- **Vertical navigation column**
  - lists major content or functional categories.
  - lists virtually all major content objects within the WebApp. If you choose the second option, such navigation columns can “expand” to present content objects as part of a hierarchy (i.e., selecting an entry in the original column causes an expansion that lists a second layer of related content objects).
- **Tabs**
  - a metaphor that is nothing more than a variation of the navigation bar or column, representing content or functional categories as tab sheets that are selected when a link is required.
- **Site maps**
  - provide an all-inclusive table of contents for navigation to all content objects and functionality contained within the WebApp.

# Ex1. Online Book Store

Design your online book store, and draw architecture.