Next week …

- Project 3 presentations are scheduled on Monday (April 9) and on Wednesday (April 11) – see course webpage
- Project 4 description will be made available next week
- Lab7 focuses on testing

WebApp & MobileApp Design

Prof. Alex Bardas

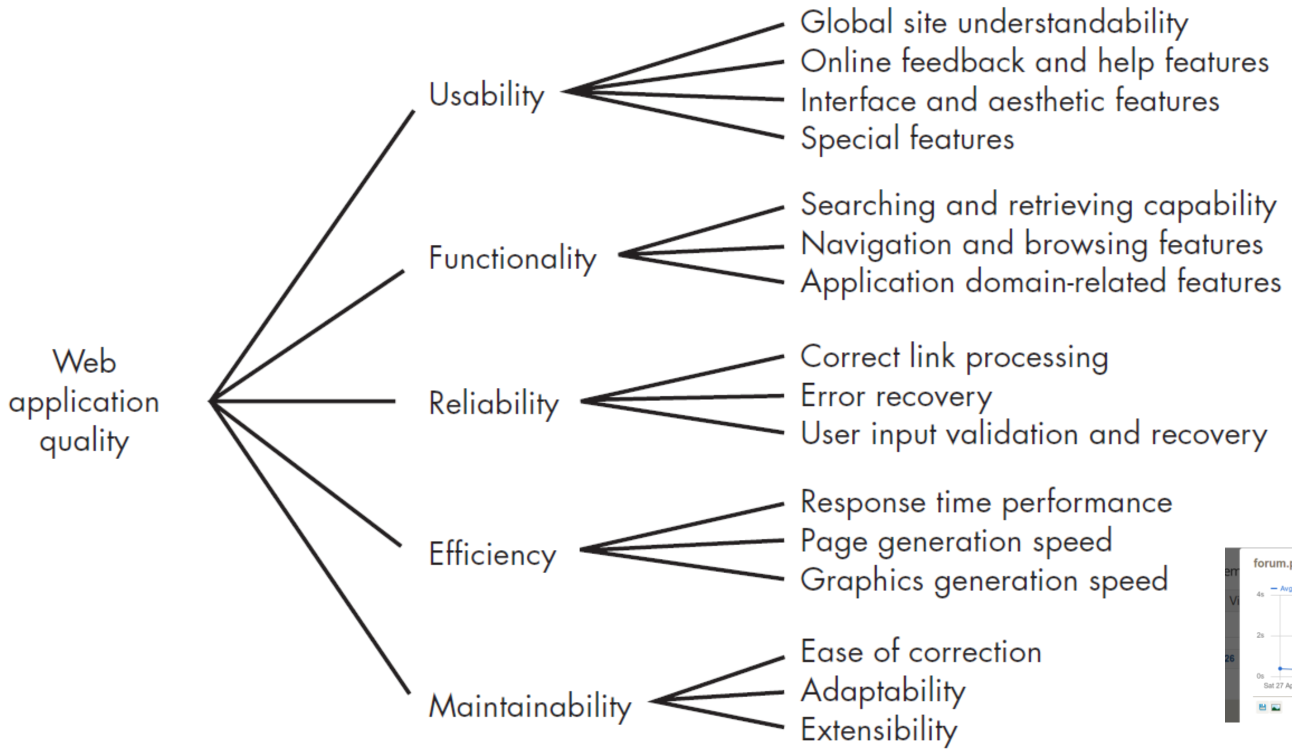
Web Applications

- A **Web application** (WebApp) is an application program that is stored on a remote server and delivered over the internet (world wide web) through a browser interface (i.e. web browser)

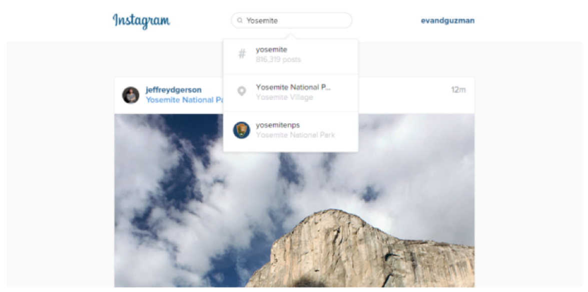
- Tim Berners-Lee’s (inventor of the world wide web) vision of the world wide web was to make all information available to all people/web-clients at all times

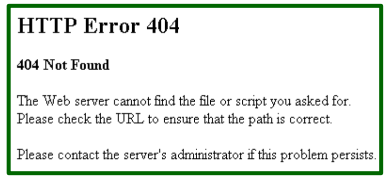
Quality Attributes

- Quality requirement tree



Web application quality
Usability
Global site understandability
Online feedback and help features
Interface and aesthetic features
Special features
Functionality
Searching and retrieving capability
Navigation and browsing features
Application domain-related features
Reliability
Correct link processing
Error recovery
User input validation and recovery
Efficiency
Response time performance
Page generation speed
Graphics generation speed
Maintainability
Ease of correction
Adaptability
Extensibility

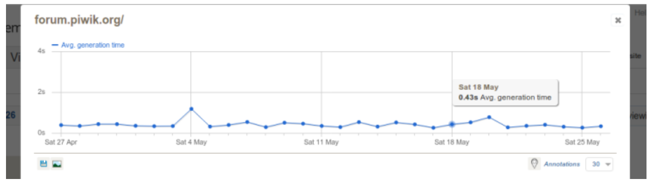



HTTP Error 404

404 Not Found

The Web server cannot find the file or script you asked for. Please check the URL to ensure that the path is correct.

Please contact the server's administrator if this problem persists.



Quality Attributes

- Other quality attributes
  - **Security**
    - Ensure the privacy of users/customers
    - Reject unauthorized access and external malicious attacks
  - **Availability**
    - 24/7/365: percentage of time a WebApp is available for use
    - Accessibility to different browser types
  - **Scalability**
    - Can the WebApp/system handle significant variation in user or transaction volume?
  - **Time to Market**
    - No longer the dominant driver: customers have little “site loyalty” and will switch quickly
    - Still (but less) important

Requirements Modeling

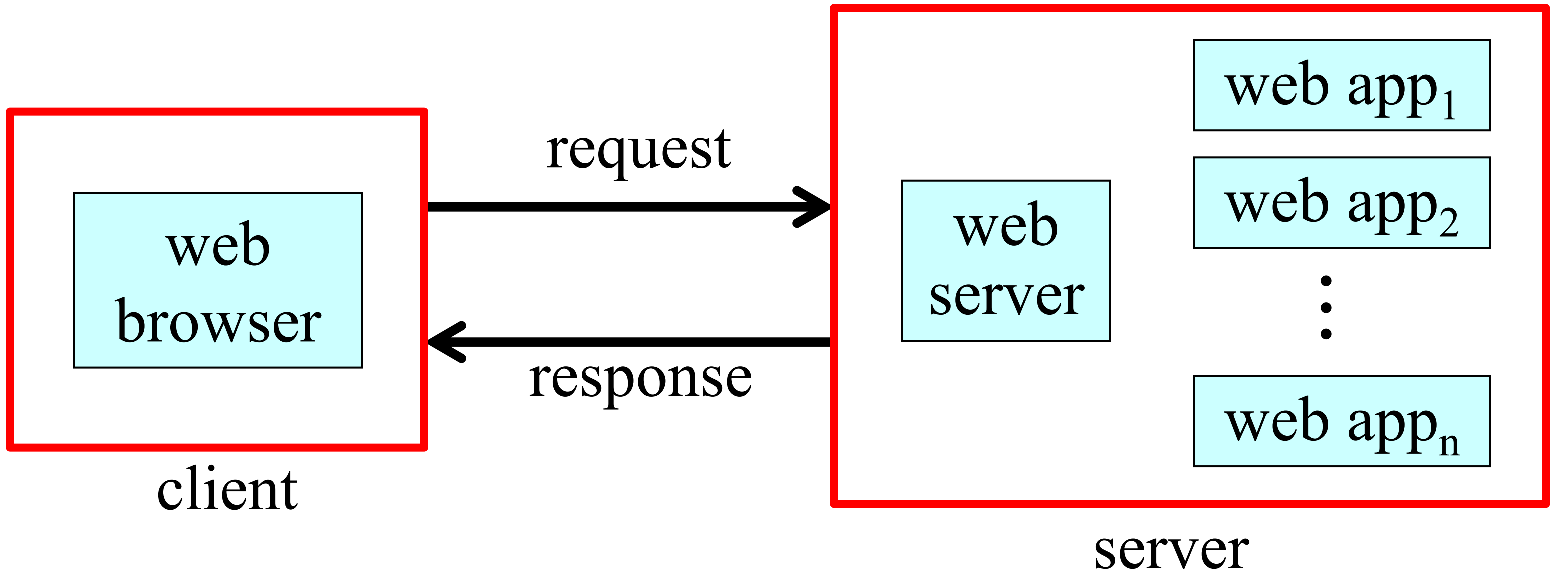
[Chapter 11.5]

For **Web and mobile apps**, analysis and design sometimes merge

- **Step 1**: Identify the goals/objectives, business context, user categories, use cases, and general requirements.
- **Step 2**: Choose right models for your app
  - For the SafeHome example refer to the textbook 11.5.4 - 11.5.8

WebApp Design

- Design in the context of Web engineering
  - Client-Server Model
    - Inherently concurrent and often distributed



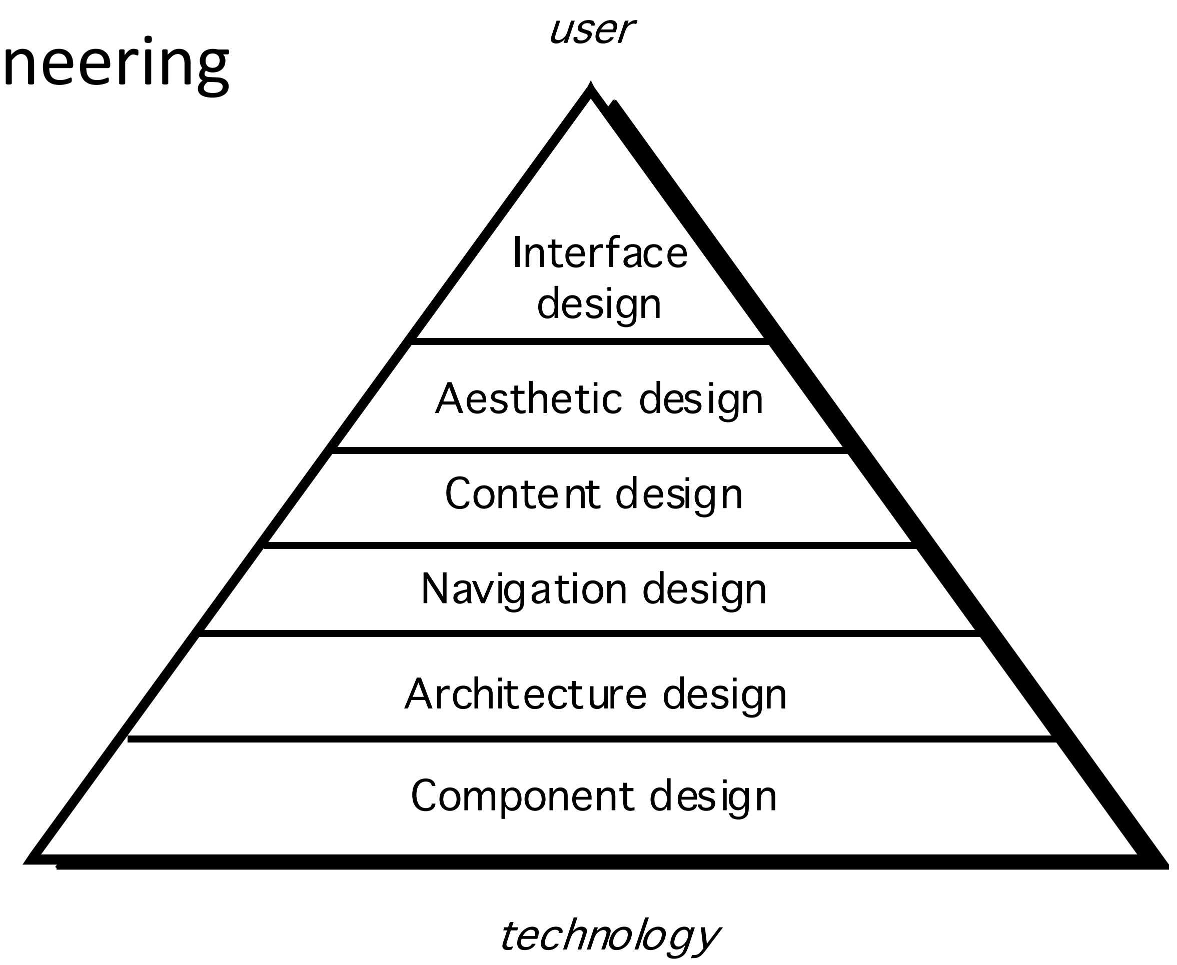
WebApp Design

- Design in the context of Web engineering
  - **Front end**
    - Get user input, render, talk to server
    - HTML, JavaScript, JavaScript libraries, plug-ins, etc.
  - **Back end**
    - Talk to client, perform business logic, talk to DB
    - Java, PHP, C#, etc.
  - **Databases**
    - Store permanent data
    - MySQL, Oracle, JavaDB, etc.

WebApp Design

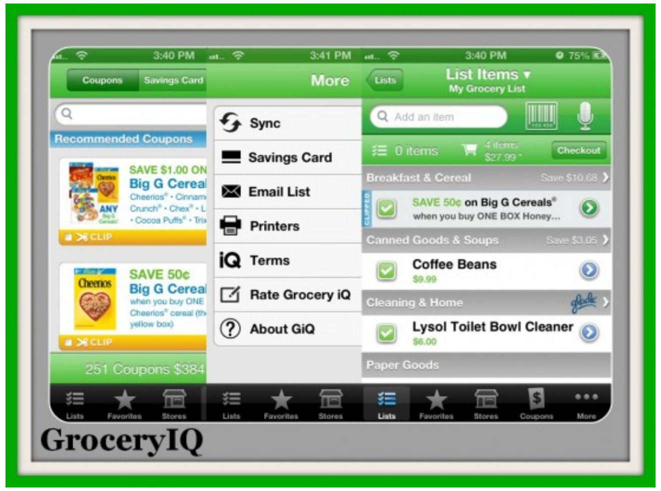
[Textbook: Chapter 17]

- Design in the context of Web engineering



WebApp Design

- **Interface Design**
  - ***Navigability***: move around without help
    - Where am I?
      - Provide an indication of the WebApp that has been accessed
      - Inform the user of the location in the content hierarchy
    - Where have I been, where am I going?
      - Provide a map/navigation mechanism at every level of content hierarchy
      - Navigation menu, graphic icon, or graphic image
  - ***Simplicity***: visually apparent
  - ***Consistency***
    - What can I do now?
      - What functions are available?
      - What links are live?
      - What content is relevant?



WebApp Design

- **Aesthetic Design**
  - ***Simplicity, Navigability, Consistency***
  - **Layout**: no absolute rules, but some tips
    - 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
    - Consider resolution and browser window size
    - Don’t extend with the scrolling bar

    - Bad designs: http://www.webpagesthatsuck.com/bad-web-design.html

WebApp Design

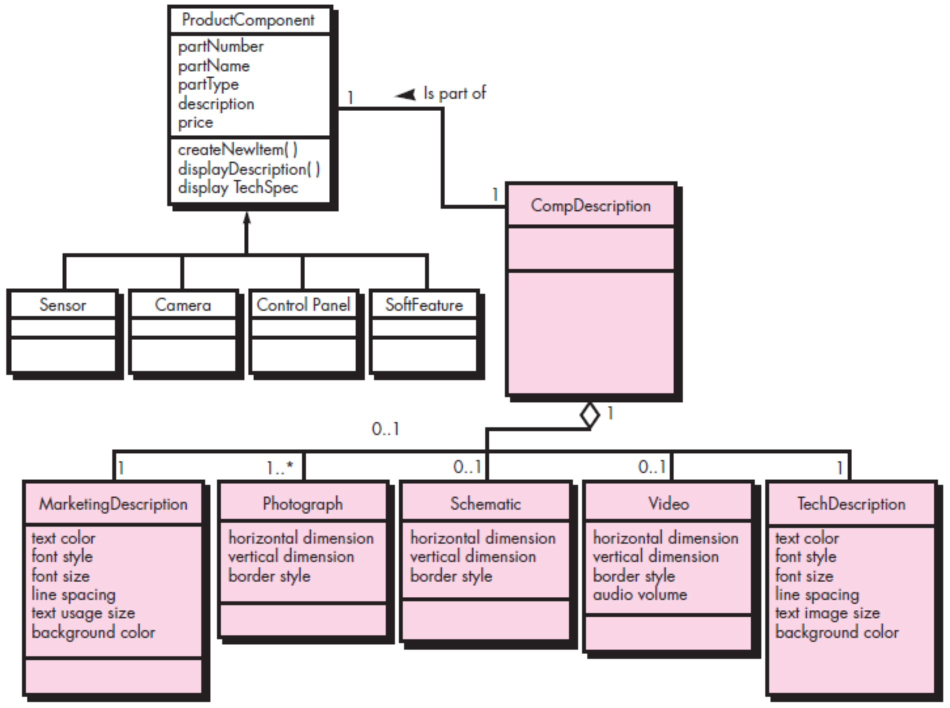
- **Aesthetic Design**
    - ***Visual Appeal, Identity, Compatibility***
    - **Graphical design**
        - Every graphic on a web page should be necessary.
        - Background color and contrast colors.
        - Do not center all text on a page
        - Keep text short
        - Keep width of text to about five inches
        - Don't use blue for regular text color or underline text for emphasis.
        - If using audio files, or movies, give viewers a choice whether or not to use them.
        - User test the visual design thoroughly.

    …

        - More tips at: http://www.graphic-design.com/Web/feature/tips.html

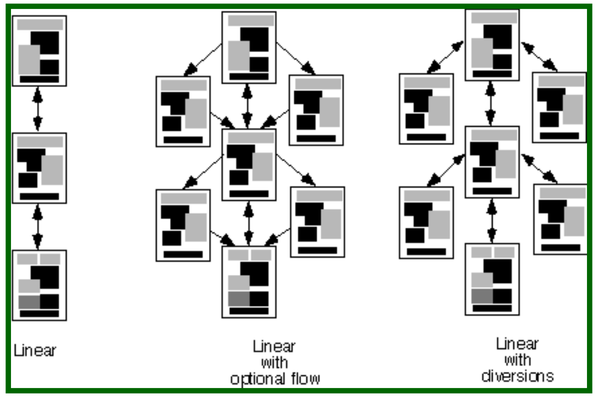
WebApp Design

- **Content Design**
  - How to represent content objects?
    - Identify core information concepts
    - Closely aligned with data objects
    - Defines the template, structure, and sketching of the content
    - Consider restrictions
  - Then, content objects are chunked into web pages



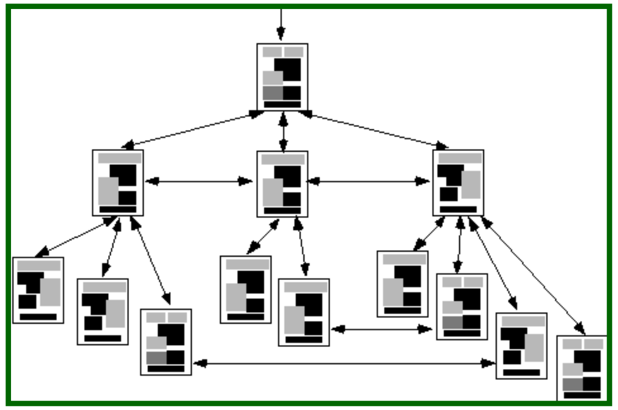
WebApp Design

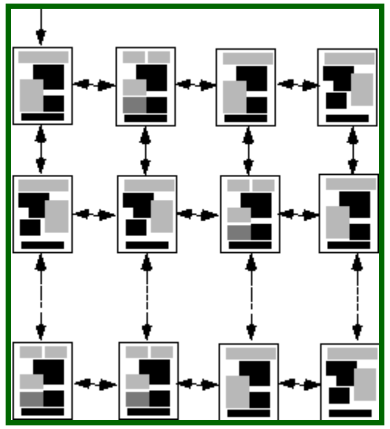
- **Architecture Design**
  - **Content Architecture**
    - How content objects are structured?



Linear

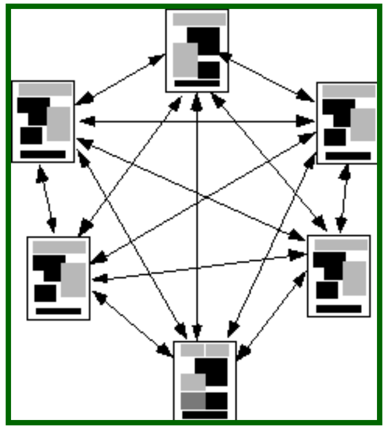
Hierarchical





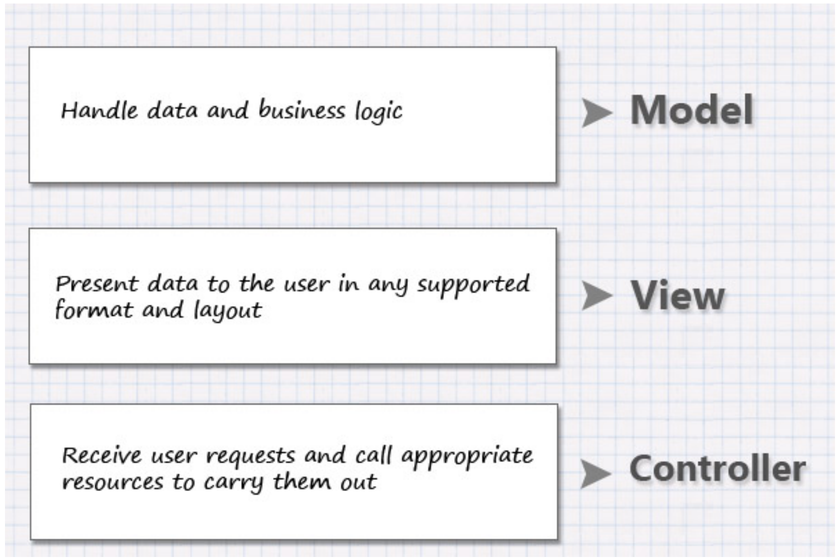
Grid

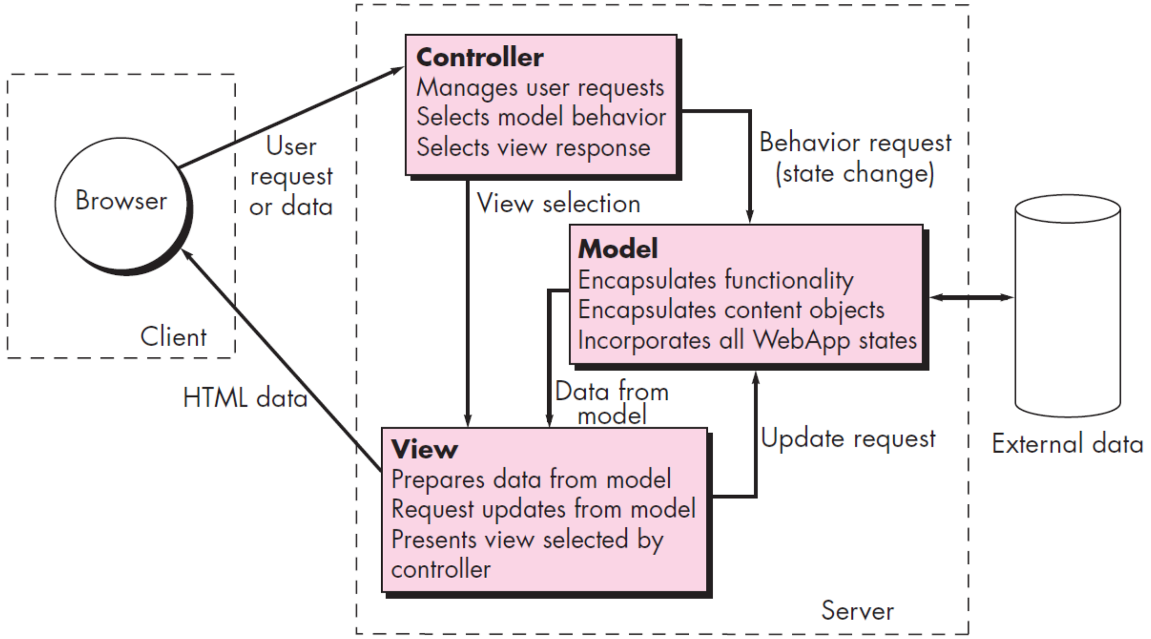
Networked



WebApp Design

- **Architecture Design**
  - **WebApp Architecture**
  - Model-View-Controller (MVC)
    - Separate user interface from navigation and application


Handle data and business logic
Model
Present data to the user in any supported format and layout
View
Receive user requests and call appropriate resources to carry them out
Controller



WebApp Design

- **Architecture Design**
  - **WebApp Architecture**
  - Model-View-Controller (MVC)
    - Model contains all application-specific content and processing logic
    - View contains all interface specific functions
    - Controller manages access to the model and the view and coordinates the flow of data between them - navigation

MVC Pattern

Name	MVC (Model-View-Controller)
Description	Separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other. • The Model component manages the system data and associated operations on that data. • The View component defines and manages how the data is presented to the user. • The Controller component manages user interaction (e.g., key presses, mouse clicks, etc.) and passes these interactions to the View and the Model.
When used	Used when there are multiple ways to view and interact with data. Also used when the future requirements for interaction and presentation of data are unknown.
Advantages	Allows the data to change independently of its representation and vice versa. Supports presentation of the same data in different ways with changes made in one representation shown in all of them.
Disadvantages	Can involve additional code and code complexity when the data model and interactions are simple.

WebApp Design

- **Navigation Design**
  - Define navigation pathways to access content and functions
    - Different roles: visitor, registered, privileged
    - Each associated with different levels of content access
  - Navigation semantics
    - A set of related navigation structures to fulfill requirements
    - Consider use cases!
    - Create a navigation semantic unit (NSU) for each use-case associated with each user role
    - “Way of navigating” (WoN) is the path to achieve a goal
    - Navigation nodes (NN) are connected by navigation links
  - Navigation syntax
    - Options: links, lists, tabs, site map, etc.

WebApp Design

- **Component-Level Design**
  - Apply the previous design methods to WebApp component with little, if any, modification.
  - WebApp components
    - Perform **localized processing** to generate content and navigation capability in a dynamic fashion
    - Provide **computation** or **data processing** capability that are appropriate for the WebApps' business domain
    - Provide appropriate **database** query and access
    - Establish **data interfaces** with external systems

MobileApp Design

- Mobile Applications can be found in any industry
  - Mobile Gaming (see gameloft)
  - Mobile Banking (see RBC)
  - Mobile Text, Presentation, and Spreadsheet (see Microsoft Office Mobile)
  - Social Networking (see Facebook)
  - Mobile News (see Yahoo! Mobile News)
  - Location Aware Services (see Find My Friends)

MobileApp Design

- Mobile application development faces multiple challenges!
- **Heterogeneity of mobile devices**
  - Run on different platforms: hardware and software differ
  - Affect their performance, usability, functionality, etc.

  Need to consider:
    - Display/Screen Size: create layout to the smallest display size
    - Memory: create applications with minimal memory footprint
    - Processing Power: avoid long waiting time
    - Input Devices: text input means, sensors and camera
    - Power management: battery life, functionalities as backlighting, network connection, memory access, specialized hardware

MobileApp Design

- Mobile application development faces multiple challenges!
- **Networking**
  - Bandwidth Usage: keep messages compact and in minimum
    - Avoid high cost/usage especially to financially conscious users
  - Transmission Errors: intermittent connectivity outages
    - Plan for potential problems especially in time/service-sensitive applications
  - Message Latency: overloaded servers, or dead/turned off smart phones, etc.
    - Avoid sending servers or clients stale information
    - Plan for resend

MobileApp Design

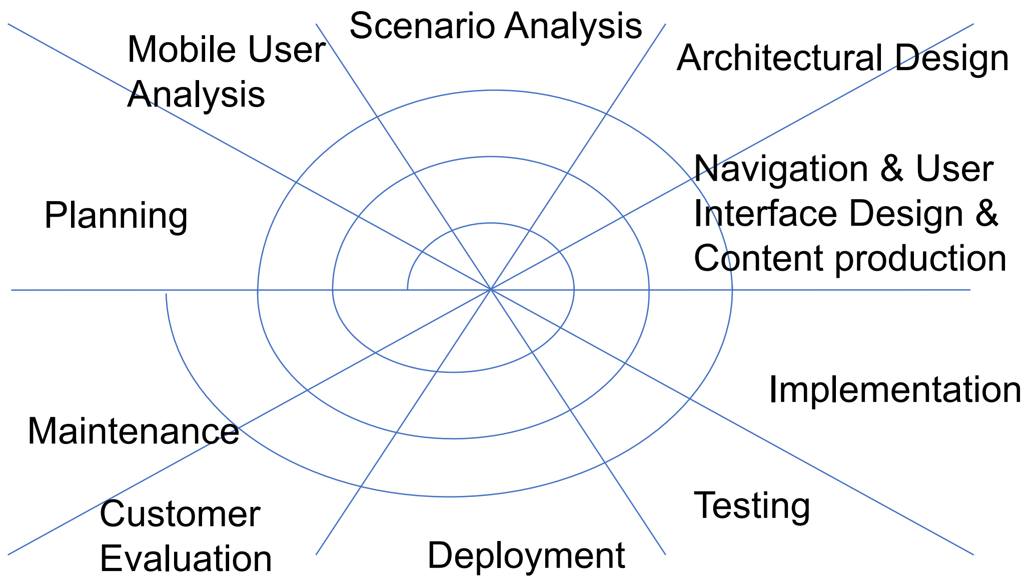
- Mobile application development faces multiple challenges!
- **Non-functional requirements**
  - Security
    - By default wireless networks are not as secure as wired networks
    - Must secure sensitive data over the air
    - Must secure sensitive data stored in device
  - Usability
    - UI limitations: small screens, rich set of interactions
  - Testability

MobileApp Design

- Mobile application development faces multiple challenges!
- **Development considerations**
  - Many development frameworks and languages
    - Android, iOS, used to have Windows, Blackberry, Symbian
    - Java, C#, Objective C
    - Use of emulators
  - Many app stores and different acceptance rules
  - Very short development cycles

MobileApp Design

- A development process for mobile applications
  - Manages a project with multiple stages of application deployment
  - Testing and design take special consideration



-- Mahmoud, Qusay H., and Zakaria Maamar. "Engineering Wireless Mobile Applications." 2006

MobileApp Design

- Planning
  - Identifying the objectives of the application
  - Specifying the scope of the first step
  - Estimating the possible cost of the project
  - Analyzing the possible risks involved
  - Creating a tentative schedule
- Analysis
  - Understand the target audience and examine the types of users
  - Asking experts in the field
  - Reading literature on the topic
  - Asking current users

MobileApp Design

- Analysis
  - Screen and Interaction Analysis
    - How the user will interact with the system, how content will be displayed
  - Usage Analysis
    - The functionality of the system, use cases should be considered to relay this information
  - Environment Analysis
    - The interaction between this system and other networks and devices

MobileApp Design

- Architectural Design
  - Attention must be paid to message latency and application partitioning to ensure performance, reliability, and security
- Navigation & User Interface Design
  - Challenges discussed previously
    - User input, screen sizes, and display characteristics as these play important roles in designing an interface
  - Screen mock-ups
    - Display the look and feel to potential customers

MobileApp Design

- Implementation
  - Use development tools
  - Class and object diagrams
  - API specifications
  - Code conventions
- Testing
  - In an emulator and also on the physical device
  - Use cases are a helpful tool in generating test cases for the system
- Deployment
  - Deploy the application on physical devices

MobileApp Design

- Customer Evaluation
  - Provide feedback
  - Report issues
  - Consider providing an email or web form where users can fill in the necessary info or provide an automated process which sends the error to the server
- Maintenance
  - Resolve any bugs found in the application and creating necessary patches
  - Improve the quality of the application with upgrades
  - Provide new services and capabilities to customers

MobileApp Design

- **Quality factors**: compare to WebApps
  - Functionality
  - Reliability
  - Usability
  - Efficiency
  - Maintainability
  - Portability

WebApps:
Usability
Functionality
Reliability
Efficiency
Maintainability
Security
Availability
Scalability
Time to Market

References

- Prof. Fengjun Li's EECS 448 Fall 2015 slides

- This slide set has been extracted and updated from the slides designed to accompany *Software Engineering: A Practitioner's Approach, 8/e* (McGraw-Hill 2014) by Roger Pressman