Lecture 8

- **WebApp and Mobile App Design**

“What are the design considerations for WebApp and MobileApp?”

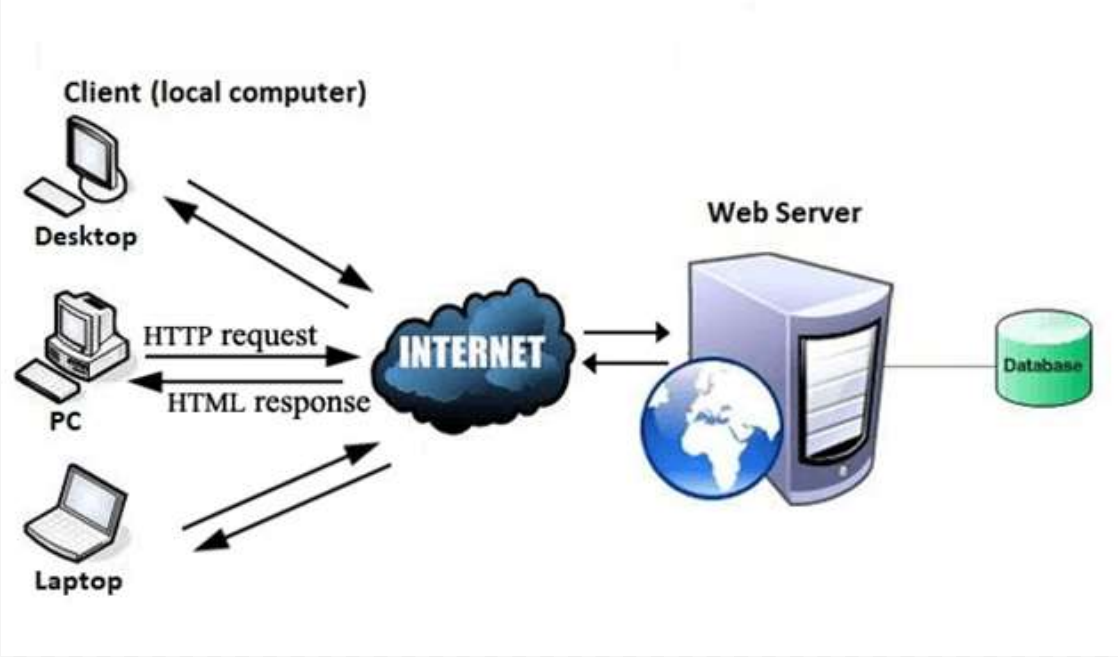
Topics

- WebApp Features
- WebApp Design
- Mobile App Features
- Mobile App Design

1. WebApp Features

World Wide Web

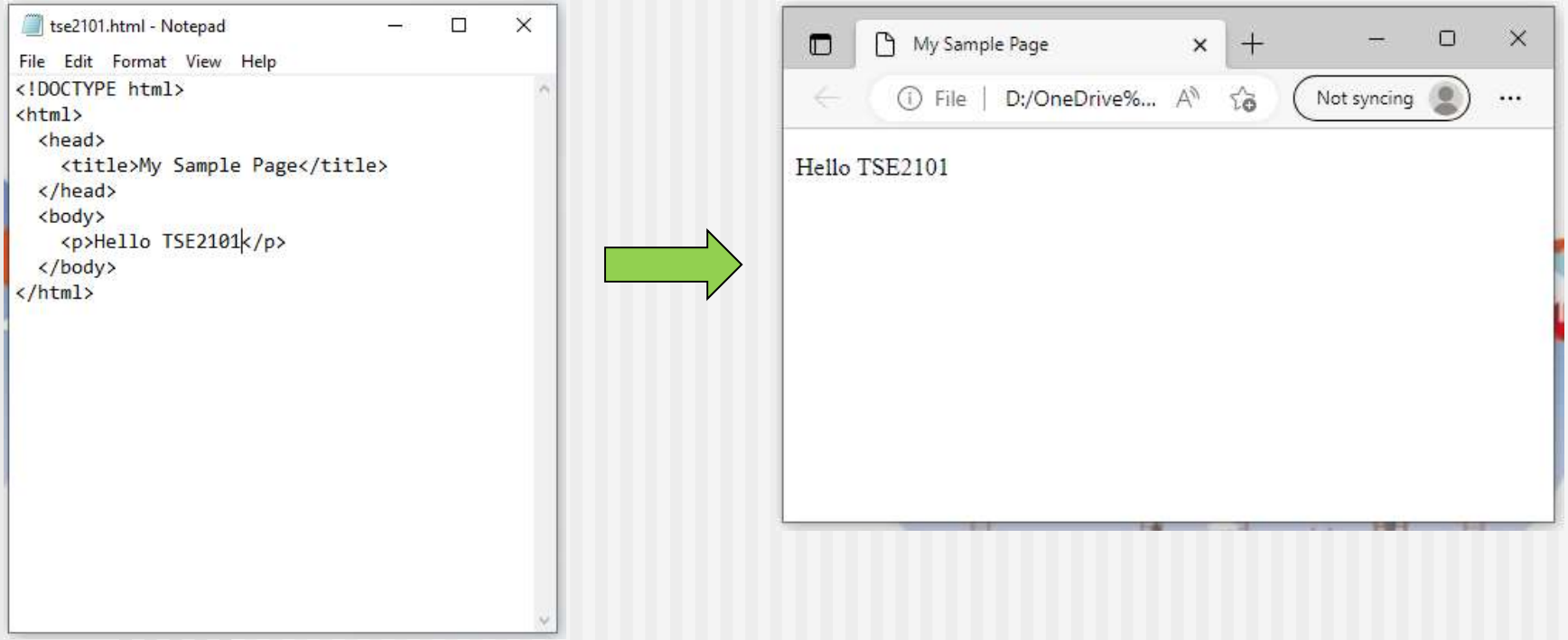
- A collection of websites or web pages stored in web servers and connected to local computers through the Internet



World Wide Web

- Client computers make requests to server computers, and the server computer responds by sending the web page data back to the client.
- Web browsers are computer programs that are installed on client computers to request web page files from servers, interprets the data in the files and displays it on the screen.
- The files in HTML format allows the browsers to display the web pages graphically in the client computers

HTML File Example



Web Applications (WebApp)

- Generates the HTML files dynamically to the browsers
- Inputs from users (as requests from client computers) are processed in the applications to generate the HTML output files – using data from databases or other sources

```
$sql = "SELECT Lastname, Age FROM Persons ORDER BY Lastname";

if ($result = $mysqli -> query($sql)) {
  // Get field information for all fields
  while ($fieldinfo = $result -> fetch_field()) {
    printf("Name: %s\n", $fieldinfo -> name);
    printf("Table: %s\n", $fieldinfo -> table);
    printf("Max. Len: %d\n", $fieldinfo -> max_length);
  }
  $result -> free_result();
}
```

```
<ul>
  {% for x in mymembers %}
    <li>{{ x.firstname }}</li>
  {% endfor %}
</ul>
```

WebApp Main Components

- **Web Browser**: the client-side component or the front-end component - interacts with the user, receives the input and manages the presentation logic while controlling user interactions with the application.
- **Web Server**: the backend component or the server-side component handles the business logic and processes the user requests by routing the requests to the right component and managing the entire application operations.
- **Database Server**: provides the required data for the application. It handles data-related tasks.

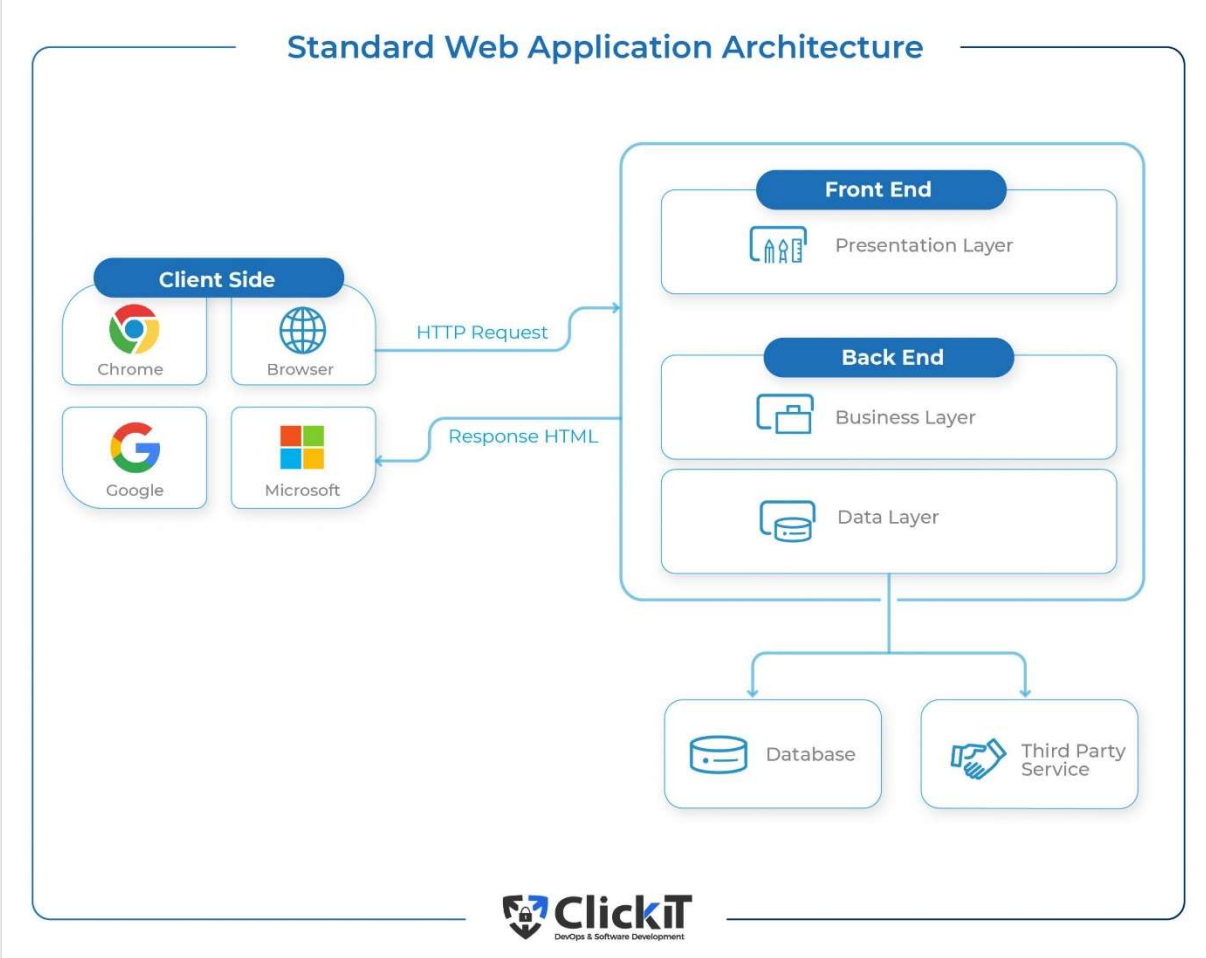
WebApp Architecture

The separation of the components in WebApp uses a 3-Tier architecture with the following layers:

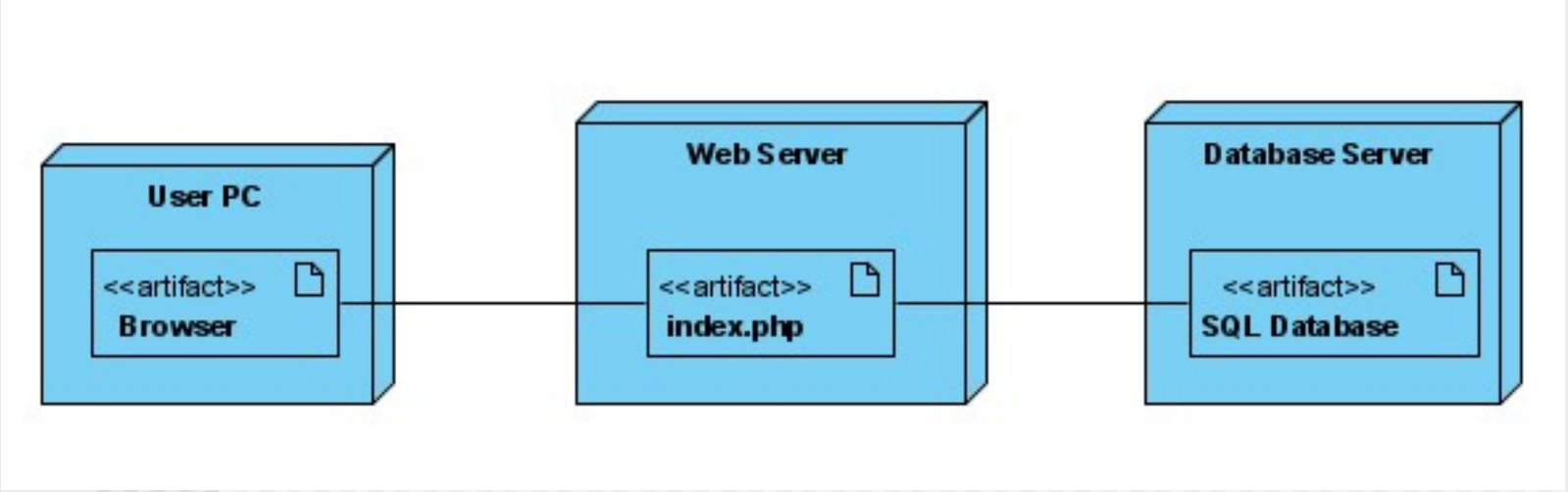
- Presentation layer / Client Layer
- Application Layer / Business Layer
- Data Layer

The development of WebApp involves developing the components in these layers and how they interact to product the required outputs/results

WebApp Architecture



WebApp Deployment Diagram



2. WebApp Design

Design & WebApps

"There are essentially two basic approaches to design: the artistic ideal of expressing yourself and the engineering ideal of solving a problem for a customer."

Jakob Nielsen

- *When should we emphasize WebApp design?*
  - when content and function are complex
  - when the size of the WebApp encompasses hundreds of content objects, functions, and analysis classes
  - when the success of the WebApp will have a direct impact on the success of the business

Design & WebApp Quality

- Security
  - Rebuff external attacks
  - Exclude unauthorized access
  - Ensure the privacy of users/customers
- Availability
  - the measure of the percentage of time that a WebApp is available for use
- Scalability
  - Can the WebApp and the systems with which it is interfaced handle significant variation in user or transaction volume?
- Time to Market

Quality Dimensions for End-Users

- ***Time***
  - How much has a Web site changed since the last upgrade?
  - How do you highlight the parts that have changed?
- ***Structural***
  - How well do all of the parts of the Web site hold together.
  - Are all links inside and outside the Web site working?
  - Do all of the images work?
  - Are there parts of the Web site that are not connected?
- ***Content***
  - Does the content of critical pages match what is supposed to be there?
  - Do key phrases exist continually in highly-changeable pages?
  - Do critical pages maintain quality content from version to version?
  - What about dynamically generated HTML pages?

Quality Dimensions for End-Users

- ***Accuracy and Consistency***
  - Are today's copies of the pages downloaded the same as yesterday's? Close enough?
  - Is the data presented accurate enough? How do you know?
- ***Response Time and Latency***
  - Does the Web site server respond to a browser request within certain parameters?
  - In an E-commerce context, how is the end to end response time after a SUBMIT?
  - Are there parts of a site that are so slow the user declines to continue working on it?

Quality Dimensions for End-Users

- ***Performance***
  - Is the Browser-Web-Web site-Web-Browser connection quick enough?
  - How does the performance vary by time of day, by load and usage?
  - Is performance adequate for E-commerce applications?

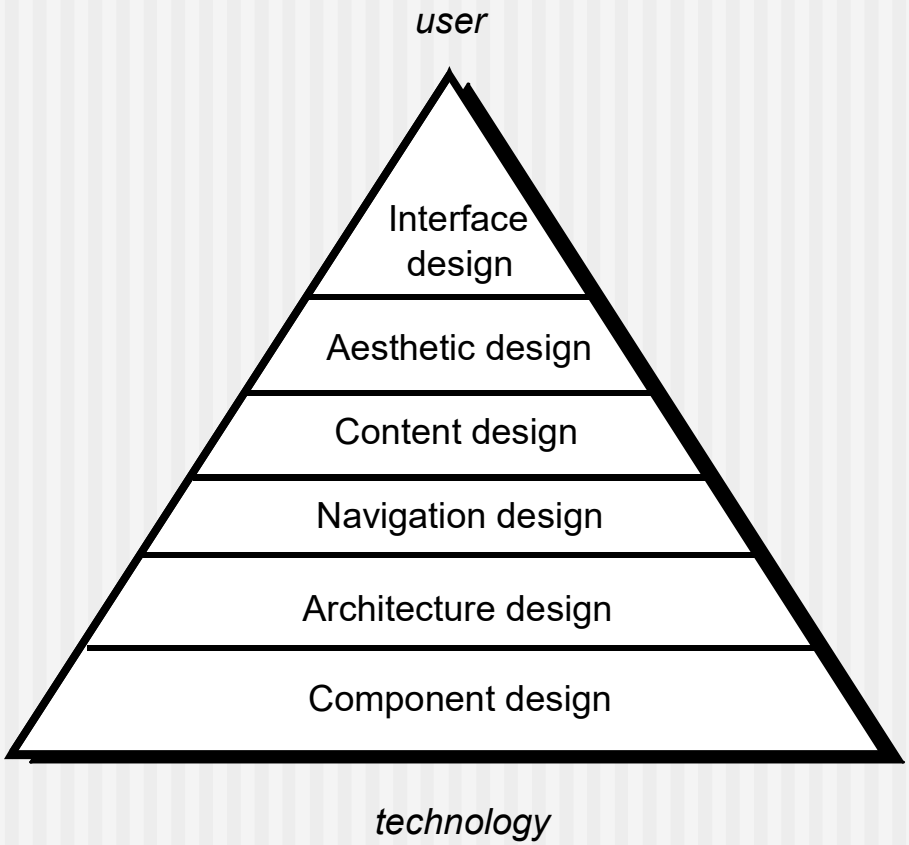
WebApp Design Goals

- Consistency
  - Content should be constructed consistently
  - Graphic design (aesthetics) should present a consistent look across all parts of the WebApp
  - Architectural design should establish templates that lead to a consistent hypermedia structure
  - Interface design should define consistent modes of interaction, navigation and content display
  - Navigation mechanisms should be used consistently across all WebApp elements

WebApp Design Goals

- Identity
  - Establish an “identity” that is appropriate for the business purpose
- Robustness
  - The user expects robust content and functions that are relevant to the user’s needs
- Navigability
  - 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 must appeal to end-users
- Compatibility
  - With all appropriate environments and configurations

WebE Design Pyramid



WebApp Interface Design

- *Where am I?* The interface should
  - provide an indication of the WebApp that has been accessed
  - inform the user of her location in the content hierarchy.
- *What can I do now?* The interface should always help the user understand his current options
  - what functions are available?
  - what links are live?
  - what content is relevant?
- *Where have I been, where am I going?* The interface must facilitate navigation.
  - Provide a “map” (implemented in a way that is easy to understand) of where the user has been and what paths may be taken to move elsewhere within the WebApp.

Effective WebApp Interfaces

- Bruce Tognozzi [TOG01] suggests…
  - Effective interfaces are visually apparent and forgiving, instilling in their users a sense of control. Users quickly see the breadth of their options, grasp how to achieve their goals, and do their work.
  - Effective interfaces do not concern the user with the inner workings of the system. Work is carefully and continuously saved, with full option for the user to undo any activity at any time.
  - Effective applications and services perform a maximum of work, while requiring a minimum of information from users.

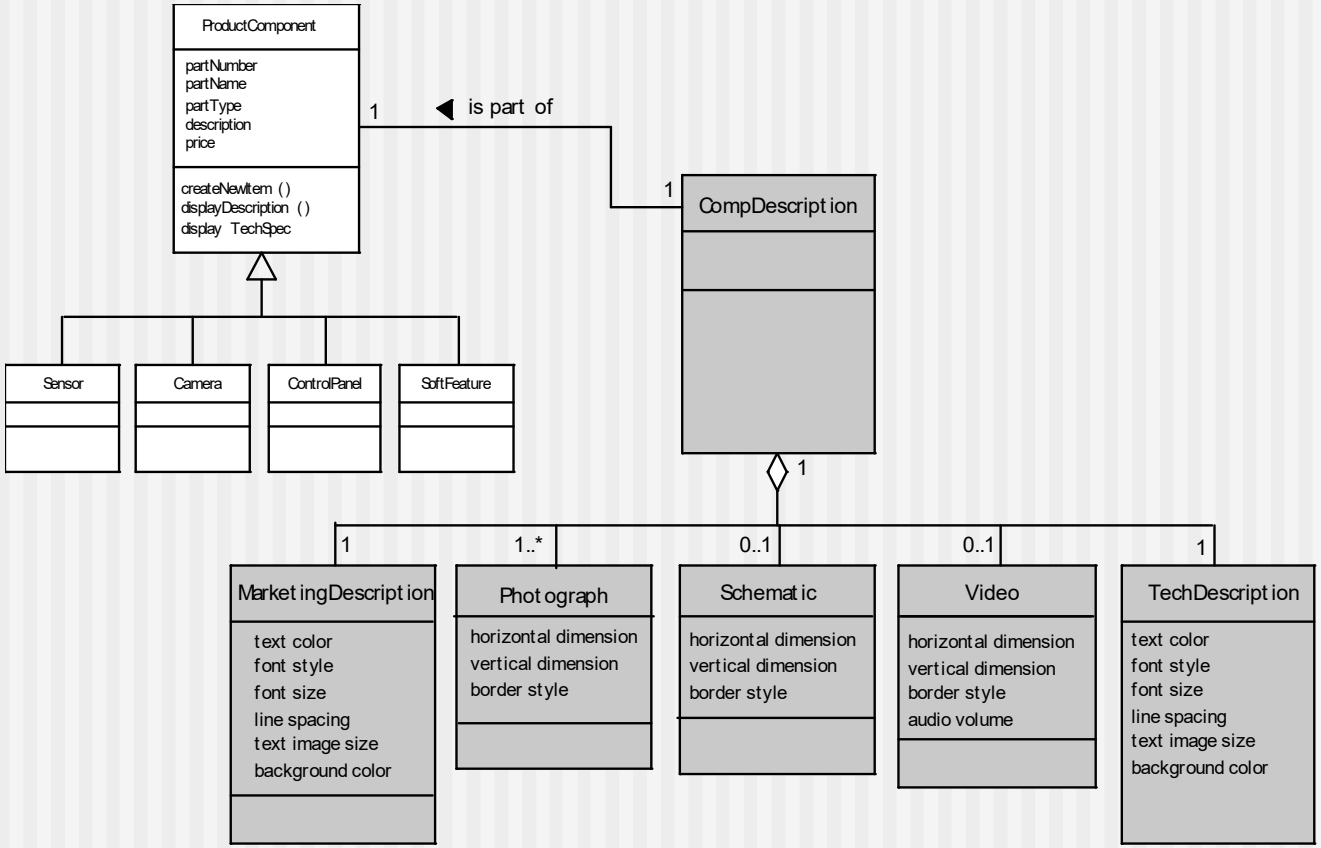
Aesthetic Design

- 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

- Develops a design representation for content objects
  - For WebApps, a content object is more closely aligned with a data object for conventional software
- Represents the mechanisms required to instantiate their relationships to one another.
  - analogous to the relationship between analysis classes and design components described in Chapter 11
- A content object has attributes that include content-specific information and implementation-specific attributes that are specified as part of design

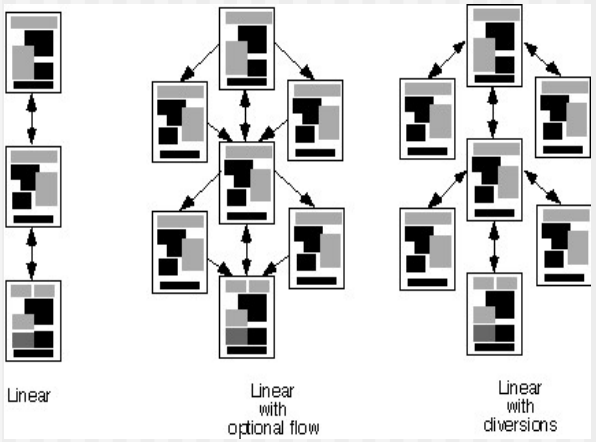
Design of Content Objects

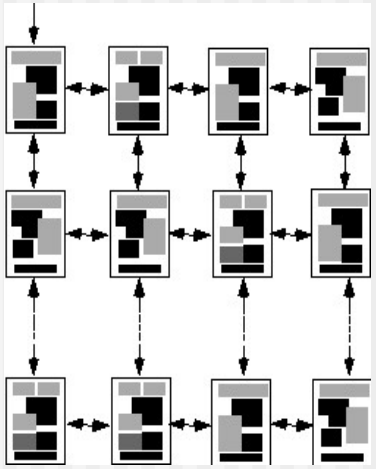


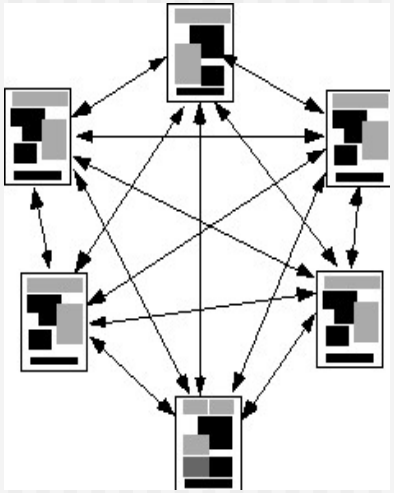
Architecture Design

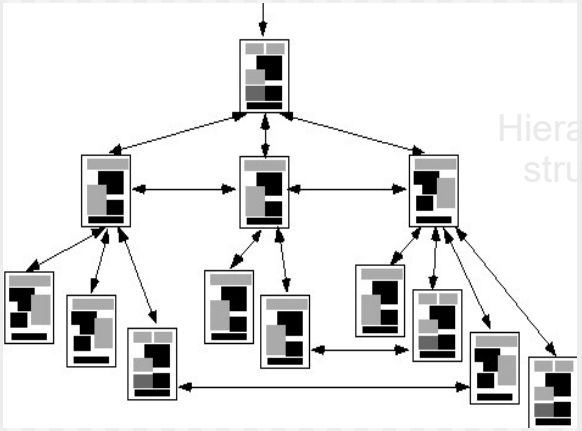
- *Content architecture* focuses on the manner in which content objects (or composite objects such as Web pages) are structured for presentation and navigation.
  - The term information architecture is also used to connote structures that lead to better organization, labeling, navigation, and searching of content objects.
- *WebApp architecture* addresses the manner in which the application is structured to manage user interaction, handle internal processing tasks, effect navigation, and present content.
- Architecture design is conducted in parallel with interface design, aesthetic design and content design.

Content Architecture









Navigation Design

- Begins with a consideration of the user hierarchy and related use-cases
  - Each actor may use the WebApp somewhat differently and therefore have different navigation requirements
- As each user interacts with the WebApp, she encounters a series of *navigation semantic units* (NSUs)
  - NSU—"a set of information and related navigation structures that collaborate in the fulfillment of a subset of related user requirements"

Navigation Syntax

- *Individual navigation link*—text-based links, icons, buttons and switches, and graphical metaphors..
- *Horizontal navigation bar*—lists major content or functional categories in a bar containing appropriate links. In general, between 4 and 7 categories are listed.
- *Vertical navigation column*
  - lists major content or functional categories
  - lists virtually all major content objects within the WebApp.

Navigation Syntax

- *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 tab of contents for navigation to all content objects and functionality contained within the WebApp.

Component-Level Design

- WebApp components implement the following functionality
  - perform localized processing to generate content and navigation capability in a dynamic fashion
  - provide computation or data processing capability that are appropriate for the WebApp's business domain
  - provide sophisticated database query and access
  - establish data interfaces with external corporate systems.

3. Mobile App Features

Mobile App

- A type of application software designed to run on a mobile device, such as a smartphone or tablet computer
- They are generally small, individual software units with limited function
- Limited hardware resources of some mobile devices makes it difficult for mobile applications to have too many functions
- Limited display also reduces the interface options and navigation of these applications

Types of Mobile App

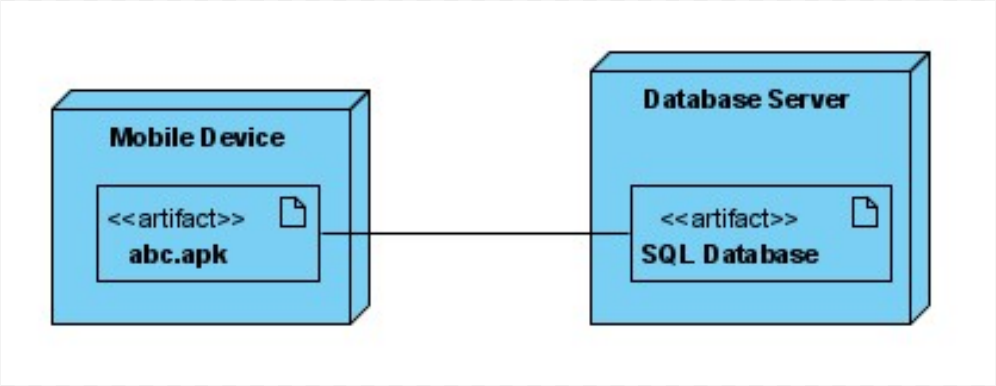
- Native apps
  - built for a specific mobile operating system, usually iOS or Android.
  - better performance and a more finely-tuned user interface (UI), and usually need to pass a much stricter development and quality assurance process before they are released
- Web apps
  - Run through a browser (HTML5 or CSS)
  - User is redirected on a specific web page, and all information is saved on a server-based database
  - Require a stable connection to be used.

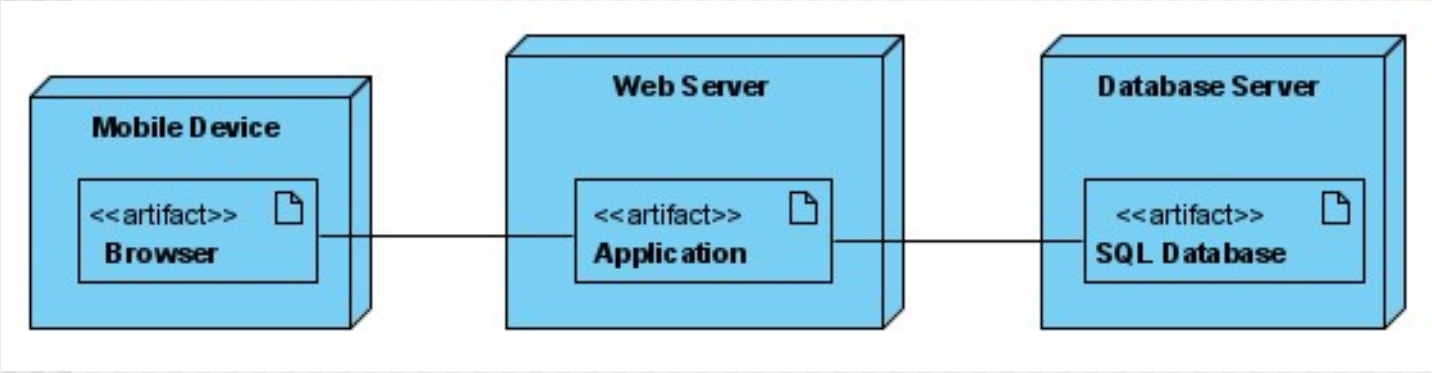
Mobile App Architecture

Similar to WebApp, Mobile App also uses a 3-Tier architecture with the following layers:

- Presentation layer / Client Layer
  - User Interface (UI) and User Experience (UX)
- Application Layer / Business Layer
  - logic and rules responsible for data exchange, operations, and workflow regulation
- Data Layer
  - data utilities, service agents, and data access components to support data transactions

Mobile App Deployment Diagram





4. Mobile App Design

Mobile Development Considerations – 1

- Multiple hardware and software platforms
- Many development frameworks and programming languages.
- Many app stores with differing acceptance rules and tool requirements
- Short development cycles
- User interface limitations

Mobile Development Considerations – 2

- Complex camera/sensor interaction
- Effective use of context
- Power management
- Security and privacy models/policies
- Device limitations (computation and storage)
- Integration of external services
- Texting complexities

MobileApp Development Process Model

- Formulation
- Planning
- Analysis
- Engineering
- Implementation and testing
- User evaluation

MobileApp User Interface Design Considerations

- Define user interface brand signatures
- Focus the portfolio of products
- Identify core user stories
- Optimize UI flows and elements
- Define scaling rules
- Create user performance dashboard
- Rely on dedicated champion with user interface engineering skills

MobileApp Design Mistakes

- Kitchen sink
- Inconsistency
- Overdesigning
- Lack of speed
- Verbiage
- Non-standard interaction
- Help-and –FAQ-itis

MobileApp Design Best Practices

- Identify the audience
- Design for context of use
- Recognize line between simplicity is not laziness
- Use the platform to its advantage
- Allow for discoverability of advanced functionality
- Use clear and consistent labels
- Clever icons should never be developed at the expense of user understanding
- Long scrolling forms trump multiple screens

References

- https://www.javatpoint.com/what-is-world-wide-web
- https://helpsme.com/articles/technology/basic-world-wide-web-concepts
- https://www.techopedia.com/definition/2953/mobile-application-mobile-app