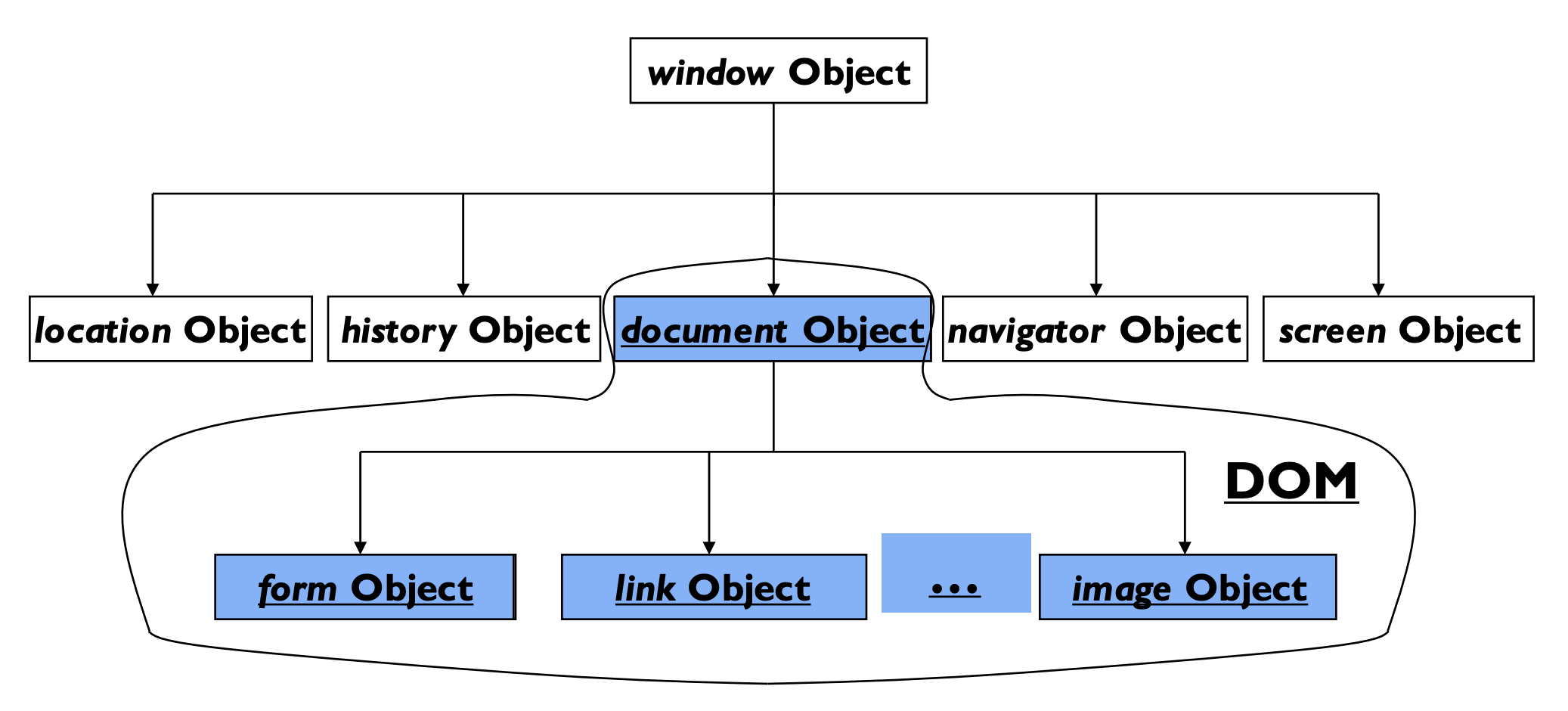
WEB PL

1/17/19

Introduction to WebDev.pdf

1. Server Side
   1. For confidentiality
   2. Can block black door access from client
2. Client Side
   1. More interactive
   2. Must need option for allowing script since some browsers can turn it off
3. Important qualities
   1. Reliability
   2. Usability
   3. Security

Web Software Model.pdf

1. Web Applications
   1. User interactive software programs, deployed on a web server, accessed via a web browser
      1. Set up a server for the application not local files
   2. Let users affect State on the server
   3. Make web content dynamic
2. Object and events
   1. Need descriptions of an object
   2. What does this object do? What are the events. When does this happen?
3. Need to think about
   1. Analyze the object that we need?
   2. Analyze the events with that object?
   3. Analyze the properties of the object?
   4. Analyze what functions need to be called?
   5. The four properties to think about … ?
4. Web Browsers and Objects
   1. Object type: Window
      1. Properties = Location = URL
   2. Object type: Document
      1. Properties
         1. URL
         2. Last Modified= some date
         3. Title = CS4640: WebPL
   3. BOM
      1. Browser Object Model
      2. Collection of objects that the browser makes available to us for use with JavaScript
      3. 
         1. Document Object = DOM
5. Server Side Processing
   1. We need container engine
      1. Understand language
      2. Which components should be called
         1. Which comes first
      3. Where to send information next
   2. Send a response back as html
6. Architectural Styles
   1. Constantly Change
   2. Separate Logic from presentation and to separate as many concers in the logic as possible
7. Page-Centric Design
   1. Requests are made to a main component and the main component response to clients
   2. Follow execution flow
      1. After logic happens then it takes the client back to the first page?
8. Server-Side Scripting Site
   1. Everything happens on the server
   2. Advantages
      1. Server-side processing, browser independent, search optimization improvement, increased security
   3. Disadvantages
      1. Poor modularity, hard to understand, difficult to maintain
9. Server-side Framework
   1. Separate into sections such as presentation, logic, data, …etc
      1. Structure server into tiers, organizes logic into classes
10. Front End Framework Site
    1. Ex. Angular
    2. Advantages
       1. Code organization, reuse, quick and easy to develop
    3. Disadvantages
       1. Duplicate logic in client and server
11. SPA (Single Page Application)
    1. Client-side logic sends messages to server, receives response
    2. Logic is associated with a single html page, written in JavaScript
    3. HTML elements dynamically added and removed through DOM manipulation
    4. Enabling Technologies
       1. AJAX
       2. DOM Manipulation
       3. JSON
       4. jQuery
       5. Angular
    5. Advantage
       1. Fast
    6. Disadvantage
       1. Hard to maintain and reuse, cross-site scripting, memory leak

Users and Usability Principles

1. What is Usability
   1. Quality Attribute
   2. Learnability
      1. How easy is it for users to start using the system?
      2. How easy is it for users to start using Collab?
   3. Efficiency
      1. How quickly can they perform tasks?
      2. Focus on UI
         1. No hardware or CPU
      3. How many steps are there? How many clicks to do a certain function?
   4. Memorability
      1. How easily can returning users reestablish proficiency
   5. Error
      1. How many errors do users make?
      2. How easy can the user come back? How easily can they recover from errors?
      3. Keep typed in input fields?
   6. Satisfaction
      1. How pleasant is it to use the system?
2. Why is it important?
   1. People will want to use the website!
      1. The focus of web dev should be the people using the website
   2. We need to convince the users that they can find all they need in the website
3. What does it mean “USER”
   1. Who is the user?
      1. Work Experience
      2. Computer Experience
      3. Age
      4. Education
      5. Reading/Writing/Language Skills
      6. Work Environment
      7. Task Frequency
   2. Look from the user perspective
   3. User Profile
4. Design of UIs
   1. Inside out design
      1. Develop a system
      2. Then add interface
   2. Outside-in
      1. Design interface
      2. Then build the system
5. Shneiderman’s 5 criteria for Measuring Usability
   1. Time to learn
      1. How long to learn to use the interface
   2. Speed of UI
   3. Avoiding user errors
   4. Retention of Skills
   5. Subjective Satisfaction