Left over

Add/Update/Delete nutrition facts about a product

1

Assignment #8
(1/2)

- Based on the static website that you've built in assignment #7, dynamic content based on information from database is to be added in this assignment.
  - Database is the same as in assignment #6.
  - Basic functions required are listed in the next page (copied from assignment #6) are to be implemented using Microsoft Expression Web.
  - As in assignment #6, two or more extra functions not on the next page's list should be implemented.
- Due: 01/18/2012

2

Assignment #8
(2/2)

  - Add/change/remove a book information in the database.
  - Add/change/remove a user in the trading system.
  - Add/change/remove a sale/buy advertisement in the system.
  - Add/change/remove a trading (history) in the system.
  - Search for a used textbooks in the trading system.
    - By at least four different methods (e.g. by user, by price, by program, …)
- Final presentation on 1/18/2012
  - Each company should prepare a 15 – 20 minutes presentation. The content includes at least the following:
    - Business model (20%)
    - 2-tier application functionality (40%)
    - 3-tier web site functionality (40%)

3

Reminder

- We have one more lecture left (Jan. 4, 2012).
- We have a final exam on Jan. 11, 2012
  - Covering lectures since (and include) SQL, VC#, Expression Web, …
  - It will be mostly a computerized exam, meaning you need to implement something using VC# & Expression Web on computer, on spot, on time!!

4

Lecture 15

Web Services

5

Today's Topic

- Introduction to Web Services
- Web Service Requirements
- SOAP web service
  - XML
  - WSDL
  - SOAP
- REST-style web service

6

Current Status

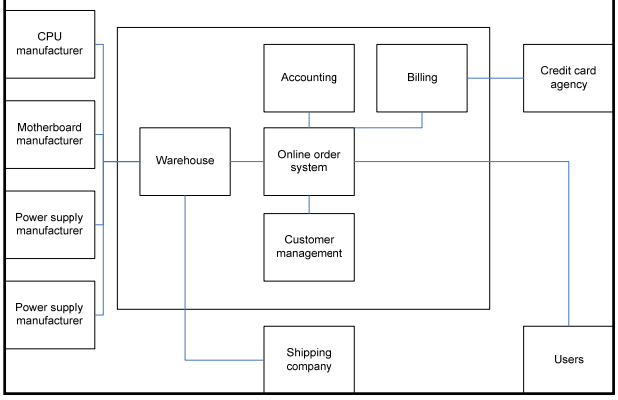
- Software concept → requirement analysis → architecture design → detailed design → implementation
- We've programmed a desktop application and are going to develop a web application for maintain user data, used computer part information, and trading used parts.
- What is left to do?
  - Verification & Validation
  - Electronic payment → data exchange with credit card agency or banks
  - Vertical integration: integrate your supplier, shipping company, … → data exchange with your partners
- Issues
  - Data format (middleware), security, …

7

Introduction

- Web services are services for applications.
  - Web pages and web applications are services for people
  - Can be seen as objects or function calls offered through the Internet using protocols developed for the Web.
  - Web services can be located anywhere on the planet Earth.
- We can use web services to integrate different information systems and applications through the Internet.

8



Web Service Requirements

- Standard RPC mechanism
  - HTTP is the most popular protocol and has been widely adopted on most platforms.
- Standard data format
  - Binary data is hardware dependent and difficult to debug (human unreadable) → use human-readable data format.
  - The data format needs to be extensible for all kinds of purposes (image, video, text, data record, map, …)
- Standard service description language
  - Describe information that the web service is needed and returns
- Standard service discovery mechanism
  - Helps programmers find needed service for his/her applications

10

Two commonly used standards(?)

- SOAP Web Service
  - XML as data format
  - SOAP as RPC mechanism
  - UDDI for service discovery
  - WSDL for service description
- REST-Style Web Service
  - A style, not a standard for RPC mechism
  - Data format are often JSON or XML.
  - No formal description or discovery mechanism

11

Data Format

12

XML
eXtensible Markup Language

- A markup language (recall HTML) that has tags enclosing data.
  - Tags: meaningful named tags to describe data enclosed.
  - Everything is in human-readable format.

```
<BOOK>
<AUTHOR> Aho, A. V. </AUTHOR>
<AUTHOR> Sethi, R. </AUTHOR>
<AUTHOR> Ullman, J. D. </AUTHOR>
<TITLE> Compilers: Principles, Techniques, and
Tools </TITLE>
<PUBLISHER> Addison-Wesley </PUBLISHER>
<YEAR> 1985 </YEAR>
</BOOK>
```

13

Introduction

- Today, we have a lot of information on web pages (HTML). However, the data is difficult to be reused.
  - Mixed data and presentation

Introduction

- XML stands for eXtensible Markup Language
  - Designed to describe data, not their look.
  - XML tags are not pre-defined (thus extensible).
  - It is going to be everywhere for storing, carrying, and exchanging data.
    - Office 2007/2010 now by default saves files as XML format (zip compressed), such as .docx, .xlsx, .pptx, …
  - XML documents has two parts: prolog (HTML head) contains XML declaration and document validation; and body (HTML body) that has tags define data.

15

```
<note>
   <to> Tove </to>
   <from> Jani </from>
   <heading> Reminder </heading>
   <body> Don't forget me this weekend! </body>
</note>
```

16

XML Example

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <note date="12/11/2002">
3    <to> Tove </to>
4    <from> Jani </from>
5    <heading> Reminder </heading>
6    <body> Don't forget me this weekend! </body>
7 </note>
  <!-- This is a remark -->
```

PROLOG

BODY

1. XML Declaration (Prolog),including XML version and encoding
2. Root element: note, date is its attribute, "12/11/2002" is an attribute value for the attribute date.
3 – 6: 4 child elements of the root
7: closing tag

17

XML vs. HTML

- XML contains only data, HTML mixes data with presentation (<H1>, <I> …)
- XML has no predefined tags, HTML tags are pre-defined so web browsers know how to render a HTML document.
- Both XML & HTML are human-readable, machine-independent text files.
- XML has more strict syntax than HTML
  - all elements must have a closing tag
  - Elements must be properly nested
  - Attribute values must be quoted
- Tags in XML are case-sensitive; HTML is case insensitive.

- XHTML → EXtensible HyperText Markup Language, will replace HTML. Nearly identical to HTML 4.01
  - XHTML elements must be properly nested
  - XHTML documents must be well-formed
  - Tag names must be in lowercase
  - All XHTML elements must be closed

18

Relationships

```
<book>
   <title> My First XML </title>
   <prod id="33-657" media="paper"> </prod>

   <chapter> Introduction to XML
      <para> What is HTML </para>
      <para> What is XML </para>
   </chapter>

   <chapter> XML Syntax
      <para> Elements must have a closing tag </para>
      <para> Elements must be properly nested </para>
   </chapter>

</book>
```

Root element: book
Child elements of book: title, prod, chapter
Title, prod, chapter are siblings (or sister elements)

19

Element Contents

- Elements can have different content types.
  - An XML element is everything from (including) the element's start tag to (including) the element's end tag.
  - An element can have element content, mixed content, simple content, or empty content. An element can also have attributes.
  - In the previous example
    - Book has element content, because it contains other elements.
    - Chapter has mixed content (text + other elements).
    - Para has simple (text) content because it contains only text.
    - Prod has empty content, because it carries no information.
    - Prod element has attributes. The attribute named id has the value "33-657". The attribute named media has the value "paper".

20

Element Naming & Attribute

- Element Naming
  - Names can contain letters, numbers, and other characters
  - Names must not start with a number or punctuation character
  - Names must not start with the letters xml (or XML or Xml ..)
  - Names cannot contain spaces
- Attribute
  - <img src="computer.gif">
  - Provide information that is not a part of the data. In the example below, the file type is irrelevant to the data
    - <file type="gif">computer.gif</file>
  - Values of attributes must be enclosed by quotes (single or double)
    - If values contain double quote, then single quotes must be used.
    - If values contain single quote, then double quotes must be used.

<gangster name='George "Shotgun" Ziegler'>

<gangster name="George 'Shotgun' Ziegler">

21

Well Formed and Valid XML

- Well formed XML
  - File content follows XML syntax
- Valid XML
  - Well-formed + conforming to rules in DTD or XSD
  - DTD: Document Type Definition (superseded by XSD)
  - XSD: XML Schema Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE note SYSTEM "note.xsd">
<note>
<to> Tove </to>
<from> Jani </from>
<heading> Reminder </heading>
<body> Don't forget me this weekend! </body>
</note>
```

22

XSD
XML Schema Definition

- XSD defines
  - elements and attributes that can appear in a document
  - which elements are child elements
  - the order and number of child elements
  - whether an element is empty or can include text
  - data types for elements and attributes
  - default and fixed values for elements and attributes
- XSD originally proposed by Microsoft, and became an official W3C recommendation in May 2001.
- XSD uses XML syntax
- Example: http://www.w3schools.com/schema/schema_intro.asp

23

XML Parser & Error Handling

- XML files are read by applications to get data inside.
  - Class libraries or functions designed to read XML data are called “XML parsers”. (need to parse data file to get data)
  - Both Java & .NET have standard XML parsers.
  - Many free or commercial XML parsers exist.
- Error handling
  - W3C XML specification states that a program should not continue to process an XML document if it finds a validation error.

24

XML Transformation

- Most modern browsers can view XML data.
  - Native XML → Mozilla and IE uses tree-view
  - XML can be translated/transformed into HTML through XSLT
- XSLT
  - XSLT is a language for transforming XML documents into XHTML documents or to other XML documents.
  - XSLT = XSL Transformation
  - XSL = Extensible Stylesheet Language
- Demo: XML → XHTML

25

XML Parsing

- Reading data from XML in programming is usually done using parsers.
  - Parser: APIs or classes to help programmers extract data from XML documents.
- Two approaches for parsing XML documents:
  - SAX: Simple API for XML
    - Sequentially read given XML document, no turning back.
    - The parser generates a series of events (in sequence), and it is programmer's responsibility to handle these events.
    - Element starts, element ends, text node
  - DOM: Document Object Model
    - View a document as an object, and each tags in its content is an attribute of the document.

http://everything.explained.at/Simple_API_for_XML/
http://en.wikipedia.org/wiki/Document_Object_Model

26

Reference

- XML:
  http://www.w3schools.com/xml/xml_whatis.asp
- XHTML:
  http://www.w3schools.com/xhtml/
- XSD:
  http://www.w3schools.com/schema/schema_intro.asp
- XSLT:
  http://www.w3schools.com/xsl/

27

JSON

- JavaScript Object Notation. Although it originates from JavaScript standard, it is a compute-rlaugnage neutral standard.
- A light-weight text-based data-interchange format.
  - Easy for humen to read and write.
  - Easy for machines to parse and generate.
- Corresponds to two data types in computer languages:
  - unordered list (name/value pairs) → object, struct, associative array, etc.
  - ordered list → array.
- It is very popular for data interchange between JavaScript on webpages and web services. JavaScript has a eval() function to directly translate JSON data into a native object, and therefore programmers do not need to worry how to parse JSON data format.
- For other languages, there are many JSON parsers that can be used.

http://json.org/

28

JSON

```
{
  "employees": [
    { "firstName":"John"  , "lastName":"Doe" },
    { "firstName":"Anna"  , "lastName":"Smith"},
    { "firstName":"Peter" , "lastName":"Jones"}
  ]
}
```

- In the example above, it describes an object (because of {…}). This object has one name/value pair. The name is "employees".
- The value paired with "employees" is an array (because of […]) containing three objects.
- Each of the contained object has two name/value pairs.
- Values can be nested structures (objects as an array element, array as the value for objects, etc.)

29

JSON

- Name is a string. JSON string mostly follow C/C++ string definitions, including some escape characters.
- Value can be string ("…"), object ({…}), array ([…]), number, true, false, null.

```
 1 {   "intArray": [1,2,3,4,5,6,7,8,9,10],
 2     "father": {
 3         "name": "George the Monkey", "age": 48,
 4         "married": false, "hasChildren": true,
 5         "speedTicket": null,
 6         "children": {
 7             "son1": { "name": "Lala", "age": 18 },
 8             "son2": { "name": "Kiki", "age": 15 },
 9             "daughter": { "name": "Koko", "age": 5 }
10         }
11     }
12 }
```

JSON

- JSON is different from XML in the following aspects:
  - There is no concept as “tag”.
  - It is easier & faster to parse JSON than to parse XML.
  - JSON needs less data size to describe the same data.
- References
  - www.json.org
  - http://www.w3schools.com/json/json_intro.asp

31

WSDL
Web Service Description Language

- XML formatted.
- Describes SOAP web services.
- Content:
  - Description and format of messages that can be passed in <types> and <message> tags
  - Direction of message passingin<portType>:
    - Request-only, request-response, response-only
  - Message encoding in <binding> element (literal, etc.)
  - Location where service is offered in <service> element

32

Function Prototype

- Prototype in C/C++/Java:
  - Defines arguments/parameters and the returning information of functions
  - e.g.
    ```
    int func1(int a, int b, int c);
    ```
- Prototype in Web service
  - Web services provide functions through the Web
  - Prototype is required for each function provided through web services → WSDL!
  - In addition, the prototype also needs to identify where/how to invoke the function (e.g. URLs)

33

WSDL
Web Service Description Language

- WSDL is a document written in XML to describe a Web service. It:
  - Specifies the location of the service
  - Specifies the operations (or methods) the service exposes
  - Describes a set of SOAP messages and how the messages are exchanged
- Six main elements are there in a WSDL document:
  - <definition>: Root WSDL element
  - <types>: The data types used by the web service.
  - <message>: The messages used by the web service. → The input and output of a web service.
  - <portType>: The operations performed by the web service. Similar to defining classes and functions or methods in traditional programming.
  - <binding>: The communication protocols used by the web service
  - <service>: where the service is located

34

WSDL Example

```
<message name="getTermRequest">
   <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
   <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
      <input message="getTermRequest"/>
      <output message="getTermResponse"/>
  </operation>
</portType>
```

Similar to class

A method of class glossaryTerms

35

Online WSDLs & References

- Amazon
  - http://soap.amazon.com/schemas2/AmazonWebServices.wsdl
  - Details at: http://aws.amazon.com/
- eBay
  - http://developer.ebay.com/webservices/latest/eBaySvc.wsdl
- Microsoft TerraService
  - http://terraservice.net/TerraService2.asmx?WSDL
- References:
  - http://www.w3schools.com/wsdl/default.asp
  - http://www.w3.org/TR/wsdl
  - http://www.developer.com/services/article.php/1602051

36