A. OVERVIEW

This course is a comprehensive introduction into web application development. The primary focus of the course is the skills and knowledge to understand how to develop modern web applications.

The bulletin description is as follows:

Web application development is a comprehensive review of the concepts and modern web development. Principal topics include major web architecture components, a survey of user interface design principals and guidelines for Web applications, designing for people (human computer interaction), analysis, design and implementation (languages) through graphical user interface (GUI) controls, forms and events. Students are expected to develop web applications using popular commercial products.

B. DEFINITIONS

A Web application is a Web system that allows its customers to execute business logic with a Web browser.

A Web system is a hypermedia system that allows for resources, such as, documents to be linked to one another.

A web application builds on, or extends, a Web system to add business functionality. In essence, a Web application uses a Web site as the front end to a more typical application.

C. GOAL

The primary goal of this course is to prepare students to be knowledgeable in Web application development concepts, terminology, methodology, and techniques.
D. PRIMARY OBJECTIVES

Web Application Development

To provide:

1. An understanding of the web application development process.
2. An understanding of web architectures
3. An understanding of how to develop web applications software.
4. An understanding of web development tools.

E. SECONDARY OBJECTIVES

Graphical User Interface Objectives

To provide:

1. To develop an understanding of graphical user interface development.
2. To develop an understanding of how to develop graphical user interfaces using software engineering processes.
3. To provide an understanding of software quality and requirements engineering with respect to graphical user interfaces.
4. To provide an understanding of the design aspects of graphical user interfaces.
5. To provide an understanding of software tools and how to implement graphical user interfaces.
6. To provide an understanding of human factors.
7. To provide an understanding of theories, principles, and guidelines.
8. To provide an understanding of expert reviews, usability testing, surveys, and continuing assessments.
9. To provide an understanding of direct manipulation, menu selection, form fillin, and dialog boxes.
10. To provide an understanding of command and natural languages.
11. To provide an understanding of interaction devices.
12. To provide an understanding of hypermedia and the World Wide Web.

Software Development

To provide:

1. An understanding of the value of analyzing customer requirements.
2. An understanding of all aspects of analysis, design, coding and testing.
3. An understanding of software tools to facilitate overall software effectiveness.
4. An understanding of how to develop software that meets a customer’s requirements for functionality, performance, usability, reliability and supportability.
E. SECONDARY OBJECTIVES (Continued)

Client/Server

To provide:

1. An understanding of client/server architectures.
2. An understanding of client/server technology.
3. An understanding of how to develop client/server software.
4. An understanding of client/server tools.

Networking

To provide:

1. An understanding of the basics of networking and data communications.
2. An understanding of LAN and WAN architecture and technology.
3. An understanding of the protocols, topologies, transmission media, network operating systems, cabling and standards.

Database

To provide:

1. An understanding of current database concepts and terminology.
2. An understanding of database design methods, techniques and tools.
3. An understanding of database features and implementation issues.
4. An understanding of data modeling and data dictionaries.
F. **LECTURER**

Ralph Brueggemann  
HOME: 871-8477  
BUSINESS: 558-0735  
EMAIL: Ralph.Brueggemann@uc.edu

G. **TIME**

Saturday: 1:00-3:40 PM

H. **PLACE**

203 Sanders

I. **Prerequisite**

Systems Analysis and Design, 30-IT-301 or the equivalent

J. **Text**


K. **Resources**

Blackboard will be used for all course materials. Please make sure that you review it periodically.

Changes to the syllabus will be posted to Blackboard. If there are changes to the syllabus they will not increase the overall assigned work.

[http://blackboard.uc.edu](http://blackboard.uc.edu)

It is recommended that you bring a zip disk or diskette to class to save sample projects that are completed during class.

L. **Attendance Policy**

You are expected to attend each class. If you are unable to attend, please notify the instructor.
M. Examinations

1. Mid-Term: November 1
2. Final: December 13

Both examinations will be comprehensive, covering the text and lecture.

N. Make-Up Examinations

1. Prior Approval

   Students who are unable to make a scheduled examination should get prior approval from the instructor.

2. Unforeseen Events

   If a student cannot attend an examination due to an unforeseen event, such as an illness, the instructor should be contacted as early as possible so that alternate arrangements can be made.

3. The make-up exam sessions will be scheduled. It is your responsibility to arrange for these sessions. These are for emergencies only.
O. Evaluation

Required

The course grade is earned and will be comprised of the following items;

1. Examinations 50%
2. Class Projects 50%

Optional Extra Credit Projects

The extra credit projects are optional and are intended to improve communications and critical thinking skills. This involves collecting and analyzing information and critically evaluating your conclusions.

1. If you are a senior, you may be asked to become a student assistant to help the other students with their projects.

2. Prepare a student journal. You are required to take class notes, handouts, and summaries of in class discussion and related projects and organize them into a notebook by sections. You have complete control over how you organize the material.

   The student journal must include information about some aspect of an event or set of events in the students’ lives and how that event relates to the course. The student should describe how the course was applied to this event.

   The student journal will be evaluated based on completeness (50%) and organization and format (50%)

3. Class participation and discussions.

4. Reading assignment (read a web application development text; write a report-5 page maximum)

5. Software Tool Evaluation Project (evaluate a web application development software product; write a report-5 pages maximum). This could be an evaluation of an Integrated Development Environment (IDE).

6. Reading assignment (read a set of web application development articles; write a report-5 page maximum)
P. Model Agenda

A typical class will be comprised of the following events;

1. Presentation of agenda and goals.
2. Review of prior week key ideas.
3. Class material.
4. Discussion and student interaction to pursue specific areas of interest and encourage further thought, ideas and interest in information technology.
5. Questions.
6. Workshops
7. Post class review for student support.

Q. Approach

1. Spiral (increasing levels of detail).
2. Supplement and not duplicate the text.
3. Encourage reading of text and outside material.
4. Encourage critical thinking.
5. Emphasize the basics.
7. Learn concepts through using software tools.

R. Academic Honesty and Classroom Rules

Academic dishonesty will be dealt with in accordance with the Student Code of Conduct.

The use of PCs during the class is only permitted during workshops and computer-based exams.

*The use of PCs during a lecture will result 1-point reduction for each occurrence.*

During computer-based exams, the use of any written materials or electronic resources will results in a grade of F.

S. Development Tool

Microsoft Active Server Pages (ASP.NET) is an increasingly popular environment for building full-featured Windows applications quickly and easily. ASP are useful for both the novice and the experienced developers to crate applications that provide fast and effective solutions to business needs. This course provides the skills and knowledge required to exploit the features and capabilities of ASP.NET. The course reviews the ASP.NET development environment and its features in order to have a fuller understanding of how to develop Web applications.
T. **The Car Example on User Interface Design**

Car controls and how they are arranged.
Remembering to put on a seat belt.
Response time to turning the wheel to a car or pressing the brakes.
Driving on the right and left side of the road. (established expectations)
Speed (faster more accidents (errors))
Speed (following a slow driver, annoying)
Road signs as progress Indicators (dynamic graphic progress indicators for loading a web page)
A frequently traveled route will necessitate more effort to optimize.

U. **Human Computer Interaction**

**What is Human Computer Interaction concerned with?**

HCI is understanding how people and computers can interactively carry out tasks, and how such interactive systems are designed." (Johnson)

**Why be concerned?**

Without user interfaces computers would be useless. User interfaces are often not as good as they could be: very often they are an afterthought and in themselves may be difficult to understand, causing the user to make unnecessary mistakes." (Thimbleby)  *kegworth air disaster, three mile island, cashpoints, videos, washing machines* ...

**The User Interface** -- the mechanism by which people access the facilities offered by the computer: *reliability, speed, accuracy, automation, enjoyment*
# COURSE ASSIGNMENT SCHEDULE

<table>
<thead>
<tr>
<th>SUBJECT</th>
<th>DATE</th>
<th>ASSIGNMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction to Web Application Development</td>
<td>September 27</td>
<td></td>
</tr>
<tr>
<td>Industrial Revolution-Williamsburg: The Steam Engine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Goals of Software Development</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Goals of Web Application Development</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Theories, Principles and Guidelines</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component Development -Lego Model/Remote Lego Model</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Introduction to HTML</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Introduction to Forms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction styles (Shneiderman-71)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Direct manipulation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Menu selection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Form fillin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command language</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Natural language</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Text Assignment</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Beginning ASP.NET 1.0 with Visual Basic .NET</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Introduction</td>
<td>ASP.NET</td>
<td>Introduction</td>
</tr>
<tr>
<td>(Regarding the Introduction, there are no test questions)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Getting Started with ASP.NET</td>
<td>ASP.NET</td>
<td>Chapter 1</td>
</tr>
<tr>
<td>(Regarding Chapter 1, there are no test questions from the section titled ‘The Installation Process’ from page 22 to 58)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Applying UML and Patterns</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O-O Analysis and Design</td>
<td>Applying UML &amp; Patterns</td>
<td>Chapter 1</td>
</tr>
<tr>
<td>Iterative Development...</td>
<td>Applying UML &amp; Patterns</td>
<td>Chapter 2</td>
</tr>
<tr>
<td><strong>2. Requirements</strong></td>
<td>October 4</td>
<td></td>
</tr>
<tr>
<td>Basic HTML</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Controls, Menus and Dialog Boxes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creating a Web Application</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Theoretical Foundations (Shneiderman-354)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Limitations of short-term and working memory and chunking</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Miller’s Law of Chunking, sources of errors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conditions for optimum problem solving</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Text Assignment</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Beginning ASP.NET 1.0 with Visual Basic .NET</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anatomy of an ASP.NET Page</td>
<td>ASP.NET</td>
<td>Chapter 2</td>
</tr>
<tr>
<td><strong>Applying UML and Patterns</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Case Study: NextGen
Inception

Applying UML & Patterns
Applying UML & Patterns

Chapter 3
Chapter 4
### COURSE ASSIGNMENT SCHEDULE

<table>
<thead>
<tr>
<th>SUBJECT</th>
<th>DATE</th>
<th>ASSIGNMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. The Process</td>
<td>October 11</td>
<td>User Interface Interface Design Concepts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Structured Concepts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Structure Theorem and Data Types</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Writing an Event Driven Program</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Task Analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Object-Action Interface Model</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Text Assignment</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Beginning ASP.NET 1.0 with Visual Basic .NET</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Forms and HTML Server Controls</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Applying UML and Patterns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Case Study: NextGen</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inception</td>
</tr>
<tr>
<td>4. Design</td>
<td>October 18</td>
<td>Interaction Devices (Shneiderman-307)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Keyboards (Fitt’s Law)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shole (QWERTY) and Dvorak</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pointing devices</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Direct-control pointing devices</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lightpen, Touchscreen, Stylus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Indirect-control pointing devices</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mouse, Trackball, Joystick, Trackpoint</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Graphics Tablet, Touchpad</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displays, Digital photography</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Text Assignment</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Beginning ASP.NET 1.0 with Visual Basic .NET</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Storing Information in VB.NET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Control Structures and Procedural Programming</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Applying UML and Patterns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>From Inception to Elaboration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Use-Case Model</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Applying UML and Patterns</td>
</tr>
</tbody>
</table>
## COURSE ASSIGNMENT SCHEDULE

<table>
<thead>
<tr>
<th>SUBJECT</th>
<th>DATE</th>
<th>ASSIGNMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>5. Architecture</td>
<td>October 25</td>
<td>Text Assignment</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Beginning ASP.NET 1.0 with Visual Basic .NET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Event-Driven Programming and Postback</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Applying UML and Patterns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Domain Model: Visualizing the Concepts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>From Requirements to Design</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interaction Design Notation</td>
</tr>
<tr>
<td>6. Mid-Term</td>
<td>November 1</td>
<td></td>
</tr>
<tr>
<td>7. Object Concepts</td>
<td>November 8</td>
<td>Expectations and Attitudes (Shneiderman-360)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Previous experiences</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Individual personality differences</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Task differences</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Task complexity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>User’s familiarity with the task</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Text Assignment</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Beginning ASP.NET 1.0 with Visual Basic .NET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Introduction to Objects</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Objects in ASP.NET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Applying UML and Patterns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GRASP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Iteration 2 and its Requirements</td>
</tr>
</tbody>
</table>

| | | Chapter 7 |
| | | Chapter 10 |
| | | Chapter 14 |
| | | Chapter 15 |
| | | Chapter 8 |
| | | Chapter 10 |
| | | Chapter 16 |
| | | Chapter 21 |
## COURSE ASSIGNMENT SCHEDULE

<table>
<thead>
<tr>
<th>SUBJECT</th>
<th>DATE</th>
<th>ASSIGNMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>8. Database Applications</td>
<td>November 15</td>
<td>Accessing Databases</td>
</tr>
<tr>
<td>Graphics and Animation</td>
<td></td>
<td>Using and Creating Databases</td>
</tr>
<tr>
<td>Text Assignment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beginning ASP.NET 1.0 with Visual Basic .NET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reading from Data Sources</td>
<td>ASP.NET</td>
<td>Chapter 12</td>
</tr>
<tr>
<td>Manipulating Data Sources</td>
<td>ASP.NET (Optional)</td>
<td>Chapter 13</td>
</tr>
<tr>
<td>Applying UML and Patterns</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling Behavior in Statechart Diagrams</td>
<td>Applying UML &amp; Patterns</td>
<td>Chapter 29</td>
</tr>
<tr>
<td>Designing the Logical Architecture</td>
<td>Applying UML &amp; Patterns</td>
<td>Chapter 30</td>
</tr>
<tr>
<td>9. Web Applications</td>
<td>November 22</td>
<td></td>
</tr>
<tr>
<td>Text Assignment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beginning ASP.NET 1.0 with Visual Basic .NET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Introducing XML</td>
<td>ASP.NET</td>
<td>Chapter 5</td>
</tr>
<tr>
<td>Web Services</td>
<td>ASP.NET</td>
<td>Chapter 18</td>
</tr>
<tr>
<td>Applying UML and Patterns</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organizing the Design…</td>
<td>Applying UML &amp; Patterns</td>
<td>Chapter 31</td>
</tr>
<tr>
<td>Introduction to Architectural…</td>
<td>Applying UML &amp; Patterns</td>
<td>Chapter 32</td>
</tr>
<tr>
<td>10. Holiday</td>
<td>November 29</td>
<td></td>
</tr>
</tbody>
</table>
## COURSE ASSIGNMENT SCHEDULE

<table>
<thead>
<tr>
<th>SUBJECT</th>
<th>DATE</th>
<th>ASSIGNMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>11. Multimedia Applications</td>
<td>December 6</td>
<td>PDA Devices</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Speech Interaction Devices</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Digital video, photography (still and motion)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CD-ROM, DVD, MPEG (Compression)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scanners</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Speech (Shneiderman-327)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Discrete-word recognition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Continuous-speech recognition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Speech store and forward</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Speech generation (synthesis)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Kurzweil Reader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Multi-Media</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Streaming Audio and Video</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Virtual Reality</td>
</tr>
</tbody>
</table>

### Text Assignment

**Applying UML and Patterns**

- On Drawings and Tools: Applying UML & Patterns Chapter 35
- Introduction to Iterative Planning: Applying UML & Patterns Chapter 36
- Comments on Iterative Development: Applying UML & Patterns Chapter 37
- More UML Notation: Applying UML & Patterns Chapter 38

| 12. Final Exam                  | December 13 |                                                                            |
**Class Projects**

Your will develop a five-part web application. These five projects will be developed incrementally and when finished will work as a complete application. You are graded based upon your ability to meet the project requirements. The projects are required to also apply principles and guidelines of effective user interface design (human computer interaction):

- If you exceed the project requirements you will receive an (S+) for (11-12 Points)
- If the project requirements are met you will receive an (S) for satisfactory for (7-10 Points).
- If you miss some of the requirements you will receive an (S-) for (5-6 Points)
- If you do not submit the project or it is substantially late you will receive a (U) unsatisfactory for (0-4 Points)

The grade range is based on the how you meet the expectation and incorporate the principles of effective user interface design.

The projects are designed to incrementally add more functionality. For example, start with a simple login-form. This login form will always be included in all 5 projects. Incrementally add the requirements for project two to project one and so on. Therefore, the fifth project will include the features of projects one through four. The project will be the basics of a generic e-business web application.
Project Description

The project is to incrementally develop a web application for a retail business.

1. **Direct Manipulation (Basic controls and forms using basic ASP and HTML)**

   Develop a login web application which allows you to enter data a web form. The web form will have at least three server controls, (label, multiple text boxes and command button)

   Due: October 18

2. **Form Fillin (basic algorithms and data structures)**

   Develop a web application, which performs data entry, and editing of the user name and password. There will be at least three server controls, (label, text box and command button) and an event for editing. There will need to be interaction between the controls. The application should demonstrate the use of the program (control) structures and data types.
   - This means the use of sequence, selection (IF THEN ELSE) and/or iteration (LOOPS) should be demonstrated.
   - You should define and demonstrate the use of the basic variables and data types such as integer, string etc. You are required to use explicit declaration.

   Due: November 1

3. **Menus and Components**

   Develop a web application that demonstrates the use of menus and components (procedures and functions). For example, a menu could be used for show a particular set of sub-forms. This application should demonstrate the use of functions and/or called subprocedures.

   Due: November 15

4. **Database Application**

   Develop a web application that demonstrates the use of a database.

   Due: November 22

5. **Extensible Markup Language (XML)**

   Develop an web application that demonstrates the use XML. For extra credit utilize some form of multi-media such as graphics, video or audio.

   Due: December 6
Development Process

User interface development is accomplished through a **collaborative process**. This process starts with the definition of user needs and requirements that can be accurately understood and defined. Users need to be involved in the process to determine their requirements.

- **Requirements** (what the customer needs)
  - Identify project scope
  - Requirements Definition
    - Identify target audience
    - Define usability specifications
    - Define specific user interface principles
    - Define specific user interface guidelines
  - Requirements Analysis—an attempt to more fully understand the problem
    - Gather Customer Requirements (Needs)
      - Survey (Interview) Customers
      - Analyze Tasks Customers Perform
    - **Describe Current Use Case Scenarios**
    - **Describe Future Use Case Scenarios**
  - Proof of concept

- **Design**—(a synthesis activity of devising a new arrangement, representation or structure, how the application will be implemented, form follows function)
  - Conceptual Design
    - End-users choose all objects
    - End-users choose metaphors
    - **Storyboard (sketches) the major objects and metaphors**
    - End-users approve the storyboards
  - Logical Design
    - Finalize outline of structure and hierarchy
    - End-users approve the outline
    - **Create low fidelity (paper model) prototype**
      - Select Site Structure (Form follows Function)
      - Select Interaction Style
        - Direct manipulation
        - Menu selection
        - Form fillin
        - Command language
        - Natural language
      - End-users review the paper prototype
      - End-users approve the paper prototype
  - **Construction** (creation and testing of incremental prototypes)
    - Physical Design
      - **Create high fidelity computer prototype**
      - Conduct usability testing
      - Revise the high fidelity computer prototype
      - End-users approve the high-fidelity prototype
  - Implementation
    - Complete the application development
    - Conduct system tests
    - Perform Quality Assurance Review
    - Conduct user tests
Definitions

Graphical user interfaces

A graphical user interface is a mechanism for computers and people to communicate with each other. A graphical user interface communicates with users through controls. A graphical user interface is based on the following four features:

- Common user interface
- High resolution, bitmapped, color display
- What you see is what you get.
- Direct manipulation.

In moving into web application development, at least four things are required:

- Hardware that supports the four GUI features.
- Software that makes it easy for application developers to build web applications.
- Viable relational database technology
- Modern local and remote networks.

Interfaces

An interface is a point at which independent systems or diverse groups interact. An interface is a noun which describes the connection between two dissimilar devices. An interface in a personal computer environment consists of a screen display, keyboard and mouse. In advanced systems, the interface might include speakers, microphones, pens, trackers, and touch sensitive areas.

User Interfaces

The user interface is the boundary between the computer and a person working with or through the computer to carry out some task. A user interface is a connection between the display, keyboard and mouse combination and the user.

The graphical user interface allows for a broadening of what is possible. Standard graphical user interfaces have provided an environment in which new ways of interaction with computers are now possible. Today, the GUI is the accepted standard in a large number of environments: Netscape and Internet Explorer browsers, Microsoft Windows Products (Windows, Windows 95, Windows 98, Windows NT), OS/2, the Mac with its own operating system, and on UNIX workstations with several operating systems. Software developers can develop software that allows users to interact with applications in ways much different than in the past.
Visual Programming

*Visual programming* is an implementation technology that uses visual expressions such as diagrams, free-hand sketches, icons, or even graphical manipulations. *Visual programming languages* (vpl) are those that use visual programming expressions. *Visual programming environments* (vpe) provide visual ways of working with the visual programming languages whether they are visual or textual.

Reality Modeling

*Multimedia* is the combination of sound and graphical images within a single application that is used for creating animated applications. *Virtual reality* involves projecting a model of the real world directly for the user.

Software engineering

*Software engineering* is the application of science and mathematics by which the capabilities of computer equipment are made useful to man via computer programs, procedures, and associated documentation. (Boehm) *Software engineering* is the means by which we attempt to produce software in a way that is both cost effective and reliable enough to deserve our trust. Clearly it is a discipline which is important to establish well and perform well. (Boehm) *Software engineering* is the practical application of scientific knowledge in the design and construction of computer programs and the associated documentation required to develop, operate, and maintain them. (Boehm)

Client/server

*Client/server* computing optimizes processing by dividing front-end application logic and user interface processing from back end data management. *Mission critical systems* are reliable software applications.

Networking

A *network* is a connection of computing systems using local and/or wide area networks. A *local network* is a communications network that provides interconnecting of a variety of data communicating devices within a small area. A *local area network* is a general purpose local network that can serve a variety of devices. Typically used for terminals, microcomputers, and minicomputers.
Database

A *database* is a collection of information organized in a way that facilitates access, analysis, and reporting for purposes of improved decision making. A *database* is a set of related data tables and other database objects, such as a data dictionary all of which is organized as a group. A *database* is a single source of centralized information for all applications.

*Structured Query Language* (SQL) is a database language used to manipulate relational databases.

A *relational database* organizes data values into logical tables made up of columns and rows.

Data Modeling

A *conceptual data model* is an organizational view of data. A *logical data model* is a users' view or representation of the data. A *physical data model* is the way the data is stored. *Normalization* is a technique used to simplify the logical data structure. Normalization is the creation of a logical database design using the normal forms.

*Normal forms* are rules used to design relational databases.

Advanced Databases

*Object Oriented databases* more closely model the real world. Object-oriented technologies have the potential of managing more data in an enterprise than any other previous technology. There are three means of delivering object-oriented technology: building an object-oriented database, extending a relational database, and developing a hybrid system that is designed for both relational and object-oriented.
World Wide Web

The Internet, a worldwide computer network, that links thousands of smaller networks. Initially developed by the US government to link its suppliers and the Pentagon. It is now used by millions of businesses and individuals.

A browser is client software designed to communicate with WWW servers and interpret the data received from them.

Hypertext is the creation and representation of links between discrete pieces of data. It is a means to organize information linked by semantic relationships.

Hypermedia is an information management tool that links various types of media such as text, graphics, and sound in an associative way. Using hypermedia technology users can access information by jumping from one point to another nonlinearly. When the data can be graphics or sound, as well as text or numbers, the resulting structure is referred to as hypermedia.

HTML, Hypertext Markup Language, is a set of tags developed by the creators of the Web to mark the structural elements of text documents.
REFERENCES

- Designing the User Interface, Ben Shneiderman, Addison Wesley, 1998
- Microsoft Visual Basic 3, Manual Set, Microsoft Corp.
- Visual Basic 3 Videos, Keystone Learning Systems, Corp.
- Decline and Fall of the American Programmer, Edward Yourdon, Prentice Hall, 1992, Yourdon Press.
- Rise and Resurrection of the American Programmer, Edward Yourdon, Prentice Hall, 1996
- Visual Basic for Windows 95 Insider, Peter G. Aitken, 1996
- Client/Server Strategies, David Vaskevitch, IDG Books, 1993
- OLE 2.0 and DDE Distilled, Al Williams, Addition-Wesley, 1994
- Object-Oriented Information Systems: Planning and Implementation,
- David A. Taylor, John Wiley & Sons, Inc., 1992
REFERENCES

- Fourth Generation Languages, Volume I, Principles, James Martin.
- Fourth Generation Languages, Volume II, Representative 4GLs, James Martin.
- Fourth Generation Languages, Volume III, 4GLs from IBM, James Martin.
- Writings of the Revolution, Earth Yourdon, Yourdon Press, 1982.
- Software Psychology, Ben Schneiderman, Winthrop, 1980.
REFERENCES

REFERENCES

• Information Engineering, James Martin, Prentice Hall.
• Book #1- Introduction: 1989
• Book #2- Planning and Analysis: 1990
• Book #3- Design and Construction: 1990