

ENG EC512 Enterprise Client-Server Software Systems Design

2008-2009 Catalog Data:

Prereq: Senior standing or consent of instructor, programming experience in C++, Java, or C#, basic knowledge of internet protocols and HTML, ENG SC 440 or equivalent is required. ENG SC 441 ENG SC 447 are recommended.. Examination of past, current, and emerging technologies. Client side technologies including DHTML, CSS, scripting, ActiveX, RSS and proprietary applications. Legacy server side technologies including, CGI, ISAPI, and active server pages. Current and emerging server technologies including ASP.NET 2, XML/SOAP web services, wireless and handheld access, WAP/WML, SQL databases, streaming media, CMS, and middleware. Design and implementation of solutions involving database connectivity, session state, security requirements, SSL, and authentication of clients. Small-team projects involving design through implementation.

Status in the Curriculum: Elective

Class/Lab Schedule: Lecture: 4 hours/week

Textbooks and other required materials:

Microsoft Electronic Learning Library, Microsoft Press. (available at no charge through MSDNAA); Software Design with C# .NET, Thomas P. Skinner (to be published)

Reference: Numerous online resources, especially the World Wide Web Consortium standards and protocol documentation; MSDN Library, Microsoft Developer Network. (Online or CD versions); Course notes provided by instructor.

Coordinator: Thomas P. Skinner, Associate Professor, ECE Department

Prerequisites by topic: Students must be fully competent in an object oriented programming language (C++, C#, or Java preferred). Familiarity with web technologies such as HTML, scripting, XML, etc. is helpful. Some programming experience with a graphical user environment, MFC, .NET, AWT, X Toolkit, etc. is also very helpful.

Goals:

1. To explore and use cutting edge technology for enterprise software development in the modern distributed environment afforded by the World Wide Web (WWW).
2. To understand the implementation of these technologies.
3. Integrate basic knowledge of computer networks with software design.
4. Learn modern rapid application development (RAD) techniques.
5. Develop experience in working with current and developing standards, e.g., W3C
6. Gain an understanding of the issues of interoperability, cross platform migration and backward compatibility.

Course Outcomes:

1. Learn state of the art design of client/server systems.

2. Apply knowledge of computer networking, algorithms, and programming to challenging problems.
3. Gain a better understanding of software engineering.
4. Learn to use modern software design and programming tools.

Course Outcomes mapped to Program Outcomes:

| | | | | | | | | | | | |
|--------------------------|-----|---|-----|---|-----|---|-----|---|---|---|---|
| Program Outcomes: | a | b | c | d | e | f | g | h | i | j | k |
| Course Outcomes: | 1-4 | | 1-3 | | 2-3 | 1 | 1,3 | 1 | | 1 | 4 |
| Emphasis: | 5 | 1 | 5 | 1 | 4 | 2 | 3 | 2 | 1 | 2 | 5 |

1=not at all; 5=a great deal;

Topics in Project Assignments:

1. Review of basic TCP/IP networking
2. HTML, HTTP, servers and other basic concepts
3. Using Visual Studio 2008 and Expression Web 2 for HTML authoring
4. Scripting on the client, JavaScript, debugging script with VS2008
5. Dynamic HTML (DHTML), document object models, styles and other advanced client side techniques
6. Legacy server side technologies, CGI, ISAPI etc.
7. .NET technology overview, ASP.NET 2 and 3
8. Alternate technologies, ASP, PHP, JSP, etc. and why I feel .NET is superior
9. Important C# concepts
10. An introduction to the framework class library and ASP.NET 2/3
11. Programming windows applications (Windows Forms – very brief intro, covered extensively in EC447) This is important for applications using Web Services.
12. Web forms using ASP.NET 2/3 and C#
13. Web controls and user controls
14. Custom controls (very brief discussion)
15. Web applications, session state, dynamic content, etc.
16. Database access with ADO.NET and Visual Studio tools (including LINQ)
17. Use of SQL Server and Microsoft Access
18. Encryption and authentication, login and roles
19. XML/SOAP web services
20. XML support in .NET
21. Writing multithreaded applications
22. Delivering media (streaming)
23. Mobile technologies (probably only very brief discussion of the limitations)
24. Content Management Systems (CMS), portals etc.
25. AJAX and other emerging technologies (probably moved earlier now that it is integrated in Visual Studio 2008)
26. Silverlight and competitive technologies

Contribution of Course to Meeting the Professional Component:

Engineering topics: 100%

Prepared by: Thomas P. Skinner

Date: June 12, 2009