الخـلاصـة

تتصف مواقع الانترنت الديناميكية بوجود تغيرات على صفحاتها أو قواعد البيانات المرتبطة بها بصورة مستمرة مع تقدم العمل و التفاعل مع هذه المواقع.

حيث ان هذه المواقع شهدت في الآوانة الاخيرة استخداما واسعا وكبيرا من قبل مختلف المستفيدين لهذا ازداد الطلب على تصميم وبناء مثل هذه المواقع. إلا أن ألآمر أدى الى حصول تباين كبير في عمليات التحليل و التصميم و البناء لهذه المواقع و بأستخدام وسائل برمجية مختلفة مثل (ASP,PHP, CGI, Perl ،........الخ) مما افقد هذه المواقع الكثير من المؤشرات الخاصة بهندسة البرامجيات لهذه المواقع.

ولهذا فأن البحث يهدف الى وضع أُسس موضوعية للجوانب التصميمية و التنفيذية معتمدا على المعايير القياسية لهندسة البرامجيات لهذه المواقع. أن الموقع الديناميكي المقترح تم بنائه و تصميمة وفق مبادئ و خصائص هندسة البرامجيات، حيث ان الموقع الديناميكي المقترح خاص بالاخبار و المستجدات على الساحة العلمية حيث يقوم بنشر اخر البحوث و اورواق العمل الخاصة بالمشتركين في الموقع من قبل الاساتذة الموجودين في قسم علوم الحاسبات التابع لجامعة النهرين من خلال ربط الموقع بقاعدة بيانات تخزن عن طريقها المواد العلمية الخاصة بالقسم. و يتمكن المشتركين بالموقع من تصفح وتحميل هذه المعلومات على حاسباتهم الشخصية. كما يعتبر الموقع المقترح كموقع قياسي بالنسبة للمواقع الموجودة على شبكة الانترنت، فيما اذا لو ان هذه المواقع تم تصميمها وبنائها وفق معايير هندسة البرامجيات ام انها مجرد مصصمة من قبل مستفيدين وهي لاتخضع لهذه الأُسس.

حيث تم الاستنتاج ان لبناء و تطوير مواقع ديناميكية ناجحة يجب ان تكون خاضعة لخصائص و قواعد هندسة البرامجيات وإلا يكون الموقع مجرد تطبيق . وكذلك ان معظم هذه المواقع وخاصة المواقع الشخصية منها هي غير خاضعة للمعايير الهندسية الخاصة بهندسة المواقع بعد اختيار مجموعة من المواقع الموجودة على شبكة الانترنت بصورة عشوائية واختبارها. المواقع المختارة هي(http://www.tashian.com ، http://www.cornell.edu ، http://consc.net) ،كما تم نشر الموقع الديناميكي المقترح على شبكة الانترنت و على العنوان التالي (http://www.mawteny.com).

Abstract

Dynamic Website today are large-scale and involve sophisticated interaction with visitors and databases, new dimension of dealing with users in Websites potentially the whole world; such Websites are often regarded as mission critical.

In parallel with this evolution, dynamic Websites loss main basic specification of software engineering, a need for Web engineering has become apparent, to successfully build large-scale, complex Web-based systems and applications.

The aim of the thesis is to put subjective bases for designing and implementation aspects based on standard measures for software engineering of dynamic Websites.

The proposed dynamic Website designed under specification of Web engineering, the proposed dynamic Website published the last scientific news, thesis, and papers of all staff of computer science department of Al-Nahrain University on Website on internet by store the content in a database connected with that Website. All proposed dynamic Website members can browsing, and download the information of the Website, only admin can edit and update the information of the Website by special username and password. There is a registration for new users also.

The proposed dynamic Website consider a standard Website to the others Websites on the internet, and find is that Website designed with sound methodology of software engineering or not.

The concluded issues in this research are: to successfully develop, deploy, and maintain high-quality Web applications must be uses scientific, engineering and management principles and systematic approaches of software engineering. Also most of Website specially the personal Website is designed out of Web engineering principles.

The chosen Websites are (http://www.tashian.com, http://www.cornell.edu, http://consc.net). The proposed dynamic Website is published on the internet at URL (http://www.mawteny.com).

List of Abbreviations

Abbreviations	Meaning
ASP	Active Server Page
CGI	Common Gateway Interface
DHTML	Dynamic Hyper Text Markup Language
GIF	Graphic Interchange Format
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IP	Internet Protocol
ISP	Internet Services Provider
IT	Information Technology
JPG	Join Photograph Group
JSP	Java Server Pages
NSU	Navigation Semantic Units
ODBC	Open Database Connectivity
PDF	Portable Document Format
Perl	Practical Extraction and Report Language
PHP	Personal Home Page
SQL	Structured Query Language
TCP	Transmission Control Protocol
URL	Uniform Resource Locator
WebApps	Web-based system and applications
WebE	Web engineering
WWW	World Wide Web
WYSIWYG	What You See Is What You Get

Acknowledgment

I would like to express my sincere appreciation to my research supervisor, Dr. Moaid A. Fadhil, for giving me the major steps to go on to explore the subject, shearing with me the ideas in my research "Software Engineering Specifications for Dynamic Website Design" and perform the points that I felt were important.

Special thanks for Dr. Ban N. Al-kallak my supervisor for her guidance, help, and advice that she gave to me.

Grateful thanks for the Head of Department of Computer Science Dr. Taha S. Bashaga.

I wish to thank the staff of Computer science Department at Al-Nahrain University for their help.

I would like to thank Mr. Eihab Ahmed for his help and encouragement.

I would like to say "thank you" to my faithful friends for supporting and giving me advice and encouragement.

Taiseer Mohammed Aljadir
October, 2006

Dedication

To every one light a candle in my way to reach my all ambitions…

Taiseer

Table of Contents

Chapter One: Introduction

Chapter Two: Web Engineering

Chapter Three: Web Applications

Chapter four: Proposed Dynamic Website Design and Implementation

Chapter Five: Conclusions and Future Works

جمهورية العراق
وزارة التعليم العالي و البحث العلمي
جامعة النهرين
كلية العلوم

خصائص هندسة البرامجيات لتصميم المواقع الديناميكية

رسالة
مقدمة الى قسم علوم الحاسبات في جامعة النهرين
كجزء من متطلبات نيل درجة الماجستير في
علوم الحاسبات

من قبل

تيسير محمد حميد الجادر

اشراف

د. مؤيد عبد الرزاق فاضل

د. بان نديم الكلاك

تشرين الاول ٢٠٠٦

Republic of Iraq
Ministry of Higher Education
and Scientific Research
Al-Nahrain University
College of Science

Software Engineering Specifications for Dynamic Website Design

A thesis
Submitted to the
College of Science, Al-Nahrain University in
partial Fulfillment of the Requirements
For
The degree of Master of Science in
Computer Science

By

Taiseer Mohamed Hameed Al-Jadir

Supervisors

Dr.Moaid A. Fadhil **Dr. Ban N. Al-kallak**

October 2006

Appendix A

Opinions Poll

1. **Job: programmer**

   Education: B.Sc in computer science

   Age: 29

   Number of year work on Internet: 6 years

Web Engineering Characteristics	Estimated Percentage Degree	How do they do that with HTML	Cornell University	David Chalmers	Proposed Dynamic Website
Content Design	25%	15	20	19	20
Navigation Design	20%	10	19	19	19
Interface Design	20%	15	19	17	19
Unique Template	10%	8	9	7	9
Goal of Website	5%	4	4	4	4
Search Engine	1-10%	5	9	9	7
Download Time	1-10%	9	8	9	9
Total		66%	88%	84%	87%

2. **Job: programmer**

Education: B.Sc in computer science

Age: 33

Number of years work on Internet: 5 years

Web Engineering Characteristics	Estimated Percentage Degree	How do they do that with HTML	Cornell University	David Chalmers	Proposed Dynamic Website
Content Design	25%	15%	20%	22%	22%
Navigation Design	20%	10%	15%	15%	19%
Interface Design	20%	10%	19%	17%	18%
Unique Template	10%	6%	9%	6%	8%
Goal of Website	5%	3%	4%	3%	4%
Search Engine	1-10%	7%	9%	5%	8%
Download Time	1-10%	8%	9%	9%	9%
Total		59%	84%	77%	88%

3. **Job: Engineer**

Education: B.Sc in Electronic Engineering and Communications

Age: 33

Number of years work on Internet: 5 Years

Web Engineering Characteristics	Estimated Percentage Degree	How do they do that with HTML	Cornell University	David Chalmers	Proposed Dynamic Website
Content Design	25%	15%	15%	15%	13%
Navigation Design	20%	12%	17%	15%	17%
Interface Design	20%	5%	17%	5%	15%
Unique Template	10%	5%	7%	5%	8%
Goal of Website	5%	1%	4%	2%	4%
Search Engine	1-10%	5%	7%	5%	7%
Download Time	1-10%	7%	9%	9%	9%
Total		50%	76%	56%	73%

4. Job: programmer

Education: B.Sc in Physics Science

Age: 28

Number of years work on Internet: 4 Years

Web Engineering Characteristics	Estimated Percentage Degree	How do they do that with HTML	Cornell University	David Chalmers	Proposed Dynamic Website
Content Design	25%	15%	20%	22%	22%
Navigation Design	20%	10%	19%	19%	18%
Interface Design	20%	10%	18%	18%	18%
Unique Template	10%	5%	9%	5%	8%
Goal of Website	5%	3%	4%	3%	4%
Search Engine	1-10%	8%	3%	3%	8%
Download Time	1-10%	9%	9%	9%	9%
Total		60%	82%	77%	86%

5. **Job: programmer**

Education: H. Diploma in Computer Science

Age: 33

Number of years work on Internet: 4 Years

Web Engineering Characteristics	Estimated Percentage Degree	How do they do that with HTML	Cornell University	David Chalmers	Proposed Dynamic Website
Content Design	25%	5%	20%	10%	20%
Navigation Design	20%	15%	18%	10%	18%
Interface Design	20%	10%	15%	10%	18%
Unique Template	10%	5%	8%	3%	9%
Goal of Website	5%	5%	5%	5%	5%
Search Engine	1-10%	7%	8%	7%	7%
Download Time	1-10%	9%	9%	9%	9%
Total		56%	83%	54%	86%

6. **Job: programmer**

Education: B.Sc in Computer Science

Age: 25

Number of years work on Internet: 5 Years

Web Engineering Characteristics	Estimated Percentage Degree	How do they do that with HTML	Cornell University	David Chalmers	Proposed Dynamic Website
Content Design	25%	10%	20%	20%	22%
Navigation Design	20%	10%	18%	18%	18%
Interface Design	20%	5%	19%	7%	16%
Unique Template	10%	5%	7%	5%	9%
Goal of Website	5%	2%	4%	3%	4%
Search Engine	1-10%	7%	8%	8%	7%
Download Time	1-10%	8%	8%	8%	9%
Total		47%	84%	71%	87%

7. **Job: programmer**

Education: B.Sc in Computer Science

Age: 33

Number of years work on Internet: 5 Years

Web Engineering Characteristics	Estimated Percentage Degree	How do they do that with HTML	Cornell University	David Chalmers	Proposed Dynamic Website
Content Design	25%	5%	15%	12%	13%
Navigation Design	20%	15%	18%	12%	17%
Interface Design	20%	10%	11%	13%	16%
Unique Template	10%	3%	8%	7%	9%
Goal of Website	5%	5%	4%	5%	5%
Search Engine	1-10%	8%	8%	7%	8%
Download Time	1-10%	9%	9%	9%	9%
Total		55%	74%	65%	77%

8. **Job: programmer (Web Designer)**

Education: B.Sc in Computer Science

Age: 27

Number of years work on Internet: 5 Years

Web Engineering Characteristics	Estimated Percentage Degree	How do they do that with HTML	Cornell University	David Chalmers	Proposed Dynamic Website
Content Design	25%	17%	18%	10%	16%
Navigation Design	20%	15%	17%	10%	18%
Interface Design	20%	10%	17%	10%	15%
Unique Template	10%	5%	10%	3%	10%
Goal of Website	5%	3%	5%	5%	5%
Search Engine	1-10%	7%	8%	8%	7%
Download Time	1-10%	10%	9%	9%	9%
Total		67%	84%	55%	80%

9. **Job: programmer (Web Designer)**

Education: B.Sc in Computer Science

Age: 32

Number of years work on Internet: 6 Years

Web Engineering Characteristics	Estimated Percentage Degree	How do they do that with HTML	Cornell University	David Chalmers	Proposed Dynamic Website
Content Design	25%	10%	20%	18%	20%
Navigation Design	20%	20%	15%	13%	18%
Interface Design	20%	5%	17%	10%	16%
Unique Template	10%	5%	7%	9%	9%
Goal of Website	5%	5%	5%	5%	5%
Search Engine	1-10%	10%	9%	9%	7%
Download Time	1-10%	10%	9%	10%	10%
Total		65%	82%	74%	85%

10. Job: programmer

Education: B.Sc in Computer Science

Age: 28

Number of years work on Internet: 5 Years

Web Engineering Characteristics	Estimated Percentage Degree	How do they do that with HTML	Cornell University	David Chalmers	Proposed Dynamic Website
Content Design	25%	10%	20%	13%	20%
Navigation Design	20%	20%	18%	10%	15%
Interface Design	20%	5%	19%	10%	17%
Unique Template	10%	5%	8%	5%	8%
Goal of Website	5%	5%	5%	5%	5%
Search Engine	1-10%	10%	10%	10%	10%
Download Time	1-10%	10%	10%	10%	10%
Total		65%	80%	63%	85%

- [Ala00] Aladdin Ayesh, “Essential Dynamic HTML Fast”, Springer-Verlag London Limited, 2000.

- [And01] Andrew McDonald and Ray Welland, "Web Engineering in Practice", University of Glasgow, Department of Computing Science, 2001.
  Available at:
  http://www.dcs.gla.ac.uk/~andrew/webe2001.pdf

- [Ath01] Athula Ginige and San Murugesan, “Web Engineering an Introduction”, University of Western Sydney, Australia, IEEE, 2001.
  Available at:
  http://www-itec.uni-klu.ac.at/~harald/proseminar/web1.pdf

- [Beb96] Bebo White, “Web Document Engineering”, Stanford University, Stanford Linear Accelerator Center (SLAC), May 1996.
  Available at:
  http://www.slac.stanford.edu/~bebo/bebo.html

- [Bry96] Bryan Pfaffenberger, "Netscape Navigator 3.0, surfing the web and exploring the Internet", AP Professional, 1996.

- [Cri01] Cristiana Amza and Emmanuel Cecchet, "Bottleneck Characterization of Dynamic Web Site Benchmarks", Rice University, Department of Computer Science, 2001.
  Available at:
  http://rubis.objectweb.org/download/dyna_bottleneck.pdf.

- [Cri02] Cristiana Amza and Annupam Chanada,“Specification and Implementation of Dynamic Web Site Benchmarks”, Rice University, CS Department, IBM Lab, 2002.
  Available at:
  http://infoscience.epfl.ch/getfile.py?mode=best&recid=55719

- [Dan99] Danial A. Tauber and Brenda Kienan, ”FrontPage 2000”,Sybex Inc.,1999.

- [Dan04] Daniel R. Licata," Verifying Interactive Web Programs", M.Sc. Thesis, Brown University ,Department of Computer Science, May 2004.
  Available at:
  http://www.cs.brown.edu/publications/theses/ugrad/2004/dlicata.pdf

- [Dav01] David Wolber, Yingfeng Su, Yih Tsung Chiang, "Designing Dynamic Web Pages in the WYSIWYG Interface", University of San Francisco, Department of Computer Science, 2001.
  Available at:
  http://www.cs.usfca.edu/~wolber/Wolber/Research/WebSheets/webdevshort.pdf

- [Dav00] David Libertone, Andrew Scoppa,"Microsoft Site Server 3.0 Commerce Edition", Prentice Hall PTR, 2000.

- [Den01] Denis Helic, "Introduction to Web Engineering", IICM, 2001.
  Available at:
  http://coronet.iicm.edu/lectures/mmis2/material/slides_we_main.pdf

- [Eih05] Eihab Ahmed Mohammed "Development of a Web Site Search Engine" M.Sc. Thesis, Informatics Institute for Postgraduate studies, Baghdad, IRAQ, 2005.

- [Eng02] Engin Kirda, "Engineering Device-Independent Web Services", Ph.D. Thesis, Technical University of Vienna, 2002.
  Available at:
  http://www.infosys.tuwien.ac.at/Staff/mj/theses/ek-phd.pdf

- [Fil01] Filippo Ricca, "Analysis and testing of Web applications", Ph.D thesis, University degli studi di Genova, IEEE, 2001.
  Available at:
  http://star.itc.it/Abstracts/web-analysis.pdf

- [Ina01] Ina Sommerville, "Software Engineering", Pearson Education, 2001.

- [Int02] Internet Guide,"Static HTML", Internet Guide, 2002.
  Available at:
  http://www.internet-guide.co.uk/static-html

- [Isr05]Isra'a Tahseen Ali Al-Attar, "Enhancement of Ranking and Query Optimizer in internet Search Engine", M.Sc. Thesis, Informatics Institute for Postgraduate studies, Baghdad, IRAQ, 2005.

- [Ist00] Information Society Technologies, IST,"Web Engineering Methodology and Development Manual", IST, 2000.
  Available at:
  http://www.e-negociogalicia.com/proxecto/documentacion/
  Web_Engineering_Methodology_and_Development_Manual.pdf

- [Jam98] James J. Hobuss, "Building Access Web Sites", Prentice Hall PTR, 1998.

- [Jef00] Jeff Cannon,"Make Your Web Site Work for You", McGraw-Hill, 2000.

- [Jim00] Jim conallen, "Building Web Applications with ULM", Addison Wesley Longman Inc., 2000.

- [Jon05] Jon Schlackl, "Static Dynamic Websites" Metamend Software and Design Ltd., 2005.
  Available at:
  http://www. Metamend.com/article.htm

- [Jos04] M. Jose Escalona and Nora Koch, "Requirements Engineering for Web Applications –A Comparative Study", University of Seville Spain and University of Munich(LMU)and F.A.S.T. GmbH, Germany, Rinton press, Journal of Web Engineering, 2004.
  Available at:
  http://www.pst.informatik.uni-muenchen. de/ personen /kochn /KochEscalonaJWE-rev.pdf

- [Jup06] Jupitermedia Corporation, "Server Types", Jupitermedia Corporation, 2006.
  Available at:
  http://www.webopedia.com/quick_ref/servers.asp
  http://www.webopedia.com/TERM/M/middleware.html

- [Lin01] Linda Ericksen,"Web Page Creation & Design", Prentice-Hall, Inc., Second Edition 2001.

- [Mar01] Martin Gaedke and Daniel Schwabe, "Web Engineering: Introduction to minitrack", University of Karlsruhe, IEEE, 2001.
  Available at:
  http://csdl.computer.org/comp/proceedings/hicss/2001/0981/07/09817072.pdf

- [Mic00] Microsoft*, "Web Applications with Microsoft InterDiv 6.0", Microsoft Corporation, 2000.

- [Nic98] Nick Gould, “Web/Database Integration with Active Server Pages”, School of Social Sciences, University of Manchester, 1998.
  Available at:
  http://les.man.ac.uk/course/asp/slides/index.htm
  http://www.socialsciences.man.ac.uk/school/

- [Pat05] Patrick Lynch and Sarah Horton "Web Style Guide, 2nd edition", Copyright Lynch and Horton, 2002.
  Available at:
  http://www.webstyleguide.com/process/plan.html

- [Rog05] Roger S. Pressman "Software Engineering", McGraw-Hill, Sixth Edition, 2005.

- [Rog06] Roger S. Pressman & Associates Inc, “Software Engineering Resources ", S. Pressman & Associates Inc, 2006.
  Available at:
  http://www.rspa.com/spi/webe-process.html

- [Ric96] Rick Stout, "The World Wide Web Complete Reference", Osborne McGraw-Hill, 1996.

- [Rob05] Robert H Zakon, "Hobbes' Internet Timeline", 2005.
  Available at:
  http://www.zakon.org/robert/internet/timeline

- [Rog99] Roger Fournier, “A Methodology for Client/Server and Web Application Development”, Prentice Hall PTR, 1999.

- [San05] San Murugesan and Athula Ginige "Web Engineering: introduction and Perspectives ", Southern Cross University and University of Western Sydney, Australia, Idea Group Inc, 2005.
  Available at:
  http://www.idea-group.com/downloads/excerpts/SuhChapter1.pdf

- [Syb00] Sybex**, "Perl, CGI, JavaScript Complete ", SYBEX Inc., 2000.

*Author is Microsoft Company
** Author is Sybex Company

- [Sve04] Sven Ziemer, “Web Engineering", Norwegian University of Science and Technology, Department of Computer and Information Science, 2004.
  Available at:
  http://csgsc.idi.ntnu.no/2004/data/svenz/WE_csgsc.pdf

- [Tho98] Thomas A. Powell, "Web Site Engineering beyond Web Page Design", Prentice Hall PTR, 1998.

- [Tho00] Thomas A. Powell, “Web Design: The Complete Reference”, McGraw-Hill, 2000.

- [Tho04] Thomas Zwanzinger, “Testing and Improving Web Application Performance ", Institute for Software Technology and Interactive Systems, Vienna University of Technology, 2004.
  Available at:
  http://www.schatten.info/lehre/diplomarbeiten/Zwanziger_Thomas_WebAppTesting.pdf

- [Val04] Vlasios Voudouris," Development of a Prototype of a Database-Driven Website to Assist the Marketing of Properties in London by Providing both Tabular and Spatial Data", M.Sc. Thesis, City University, Information Systems and Technology Department , 2004, Available at:
  http://vega.soi.city.ac.uk/~fd776/mywork/dis.pdf

- [Vir01] Virtualis Glossary,”Web Page”, Virtualis Systems, 2001. Available at:
  http://www.virtualis.com/vr/virtuali/guides_glossary.html#w

- [Wik06] Wikipedia Free Encyclopedia, “Web Site”, Wikimedia Foundation, Inc, 2006.
  Available at:
  http://en.wikipedia.org/wiki/website.

```
[ExtShellFolderViews]
Default={5984FFE0-28D4-11CF-AE66-08002B2E1262}
{5984FFE0-28D4-11CF-AE66-08002B2E1262}={5984FFE0-28D4-11CF-AE66-08002B2E1262}

[{5984FFE0-28D4-11CF-AE66-08002B2E1262}]
PersistMoniker=file://Folder.htt

[.ShellClassInfo]
ConfirmFileOp=0
```

Chapter Five
Conclusions and Future Works

5.1 Conclusions

1. Website are become more like software , like building software, so software engineering can be applied to Website development to help bring the chaotic process under control and minimize the risk of the failed project. Web engineering is an emerging discipline having both theoretical and practical significance.
2. Each Website has a life cycle, the process by which a successful Website lives. The cycle could be analysis, design, implementation, testing, maintenance, and developing. If any component missing, Website will not achieve its target. Website maintenance and developing are the keys to keeping the cycle moving.
3. Content updating and Website development are very important to keep the Website active with many visitors.
4. By comparing the proposed dynamic Website with other Websites found that, many Websites on the World Wide Web build in a chaotic ways without any applying to principles of Web engineering.
5. The Website with great content and a great interface but poor information architecture, it relatively useless.
6. If the user cannot easily find the information, the Website loses its effectiveness, so each Website has great content and has many Web pages must be has a search engine to help the users to find what there want.

5.2 Suggestions for Future Works

By the experiments, several suggestions are identified that could be implemented in the future to make the project more optimal in its activation with the user:

1. To build Website under Web engineering principles, and help all users to design a Website with the basic principles of Web engineering, build an editor contain these basic principles to guidance the designer to the correct way. For example:
   - design content
   - navigation deign
   - Architecture design
   - unique template

لبناء موقع ويب خاضع لآسس هنسة البرامجيات ولمساعدة مستخدمي الانترنت ومصممي مواقع الويب لتصميم و بناء موقع خاضع لمبادئ هندسة الويب نقترح بناء محررنصوص خاص ببناء مواقع البناء يحوي اهم الصفات لبناء مواقع ويب ناجحة ومنها

كيفية تصميم المحتوى

Hyper Link كيفية بناء ادواة الابحار داخل الموقع

كيفية تحديد القلب الموحد بالتصميم لكافة صفحات الويب

2. Because of the limitation of this research, the membership on the proposed dynamic Website is limited, in the future other department of the university can publish there research and papers on that Website by using larger database.

بناء وتوسيع قواعد البيانات وخزنها بخادم server خاص للموقع الديناميكية ذات المحتوى الكبير و ربطها بمحرك خاص يسهل عملية الحصول على المعلومات

3. This research lead to a new research that produce a statistical of all Websites published on World Wide Web at the few last years to give overview of all Websites that construct out of Web engineering and the Websites build with the principles of Web engineering.

هذا البحث الصغير يقودنا الى بحث اخر موسع يمكن تن يكون عبارة عن مجموعة من الاحصائيات التي تجرى على اخر المواقع الموجودة على شبكة الانترنت من خلال هذه الائحصائيات ممكن ان نعرف كم من هذه المواقع مبني و مصمم تحت خصائص انظمة الويب وكم منها مبني بطريقة عشوائية غير خاضعة لهندسة البرامجيات.

Chapter four
Proposed Dynamic Website Design and Implementation

4.1 Introduction

Part one of this works is concerned with design a dynamic Website under principles and specification of software engineering as discussed in chapter two. The second part use the proposed dynamic Website as a standard Website to others Websites on the internet to find out if these sites designed under the software engineering specification or not and estimate a percentage degree for each Website depend upon some chosen characteristics of Web engineering.

4.2 Web Application Life Cycle

The Web engineering steps to build any Web application illustrate in figure (4.1), these steps use to construct the proposed dynamic Website as discussed in chapter two.

4.2.1 Website Planning

At first, the Web engineer must understand the whole idea of the Website, the purpose from that Site and accesses the underlying need for that Website.

The goals of the Website, the goal of proposed dynamic Website is to publish the scientific news thesis, papers, and scientific research to be useful for all students in computer science. After that define the structure of Website; all pages and links are illustrated in a Site map. As shown in figure (4.2)

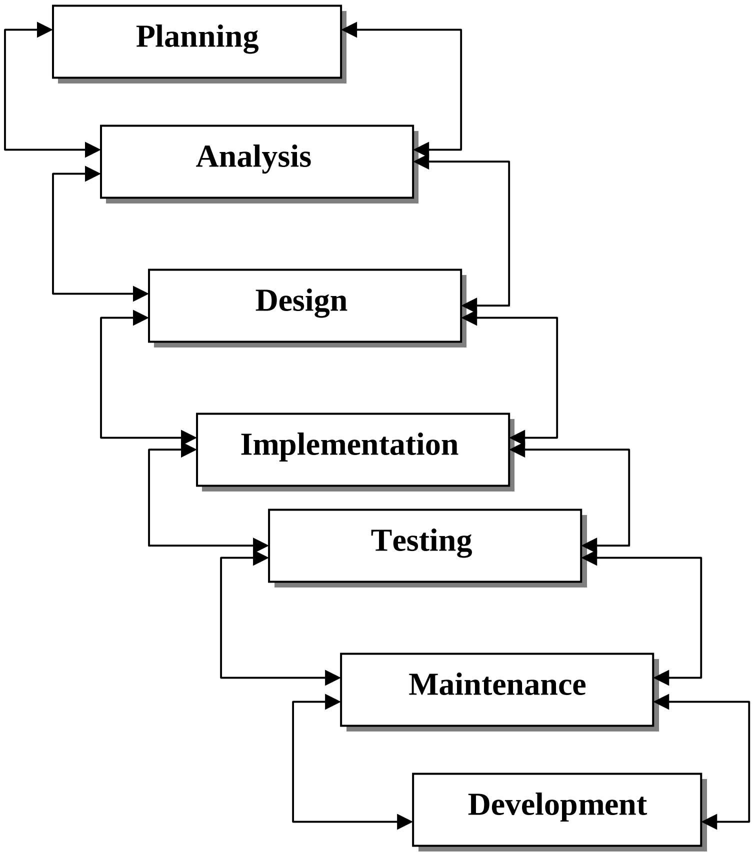


Figure (4.1) Web Application Life Cycle

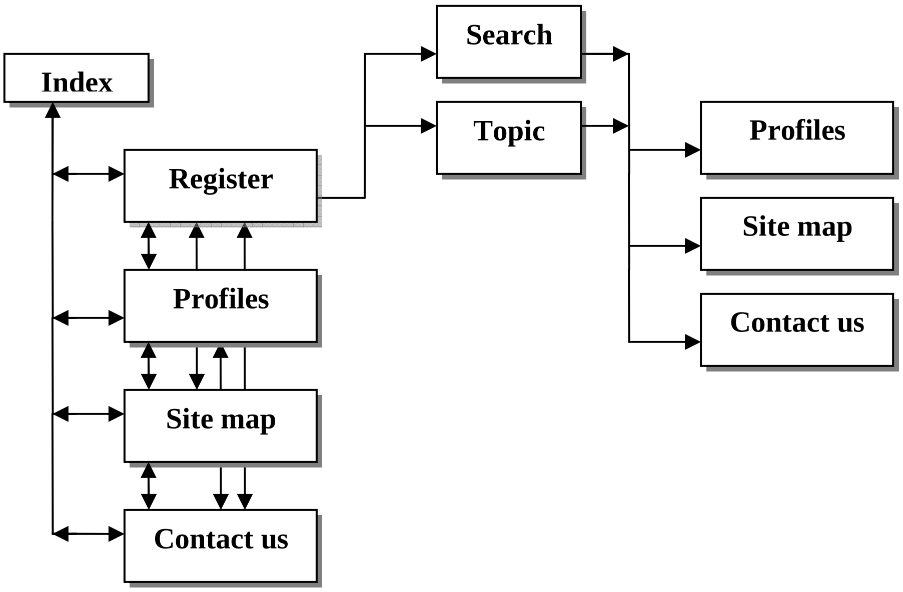


Figure (4.2) Site Map of Proposed Dynamic Website

4.2.2 Website Analysis

During planning phase, the basic goals and structure of Website are identified. In this step, the Web engineer should understand the problems to be addressed by the proposed dynamic Website.

The analysis for the proposed dynamic Website as fallows:

1. The proposed dynamic Website stockholders are: (administrators, members, and gust users) see figure (4.3).
    1. The administrators can edit, add, delete, and update the scientific news and all information in the proposed dynamic Website by using special username and password.
    2. The members can download the information and browse the Website.
    3. Guest users only can browse the Website; they also can register and become a member on the Website.

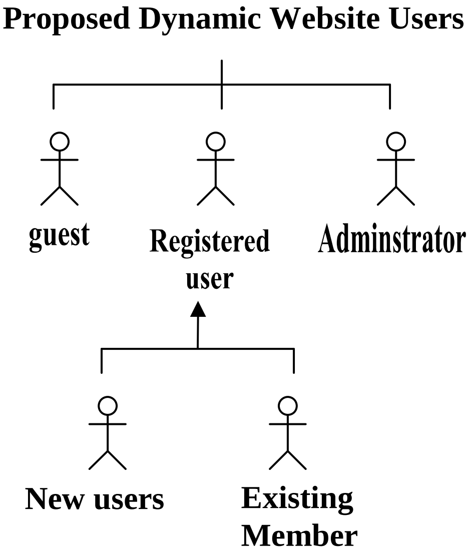


Figure (4.3) The Proposed Dynamic Website stockholders

2. The functions of proposed dynamic Website are:
    1. Register any new user want to be member in the Website.
    2. Edit, delete, and update the scientific information by the administrator of the Website and store them in a database.

3. Store username and password of all members and administrator in a database by using SQL server.
4. Providing all computer scientists and all interesting with computer science with last scientific information by *download* them form Website through Portable Document Format (PDF) files, *upload* these files by using File Transfer Protocol (FTP) to the internet.
5. Store all scientific news in a table on a database.

3. The contents are supplied by the scientific staff of the computer science department and this information always updated from the administrators of the Site.

The full analysis for the proposed dynamic Website is shown in the main algorithm:

1. Main Algorithm

This algorithm describes the work of the proposed dynamic Website in general as shown in figure (4.4):

- When any user enters to the Website, the user is either, administrator, member, or just guest. The Website checks the stockholder first to introduce a proper function to each stockholder.
- If that user is member in the Website, then that member is either administrator or just a member. The Website expects the right username and password when login to the Site, it checks them with a stored username and password in a database.
- If the user is just a member, he can download the scientific files and browse Website, or if the user is an administrator, he can edit, delete, or update the content of the site, then logout from the Website.
- If the user enters the Website as a guest, he can only browse the Website and also can make registration in Website.

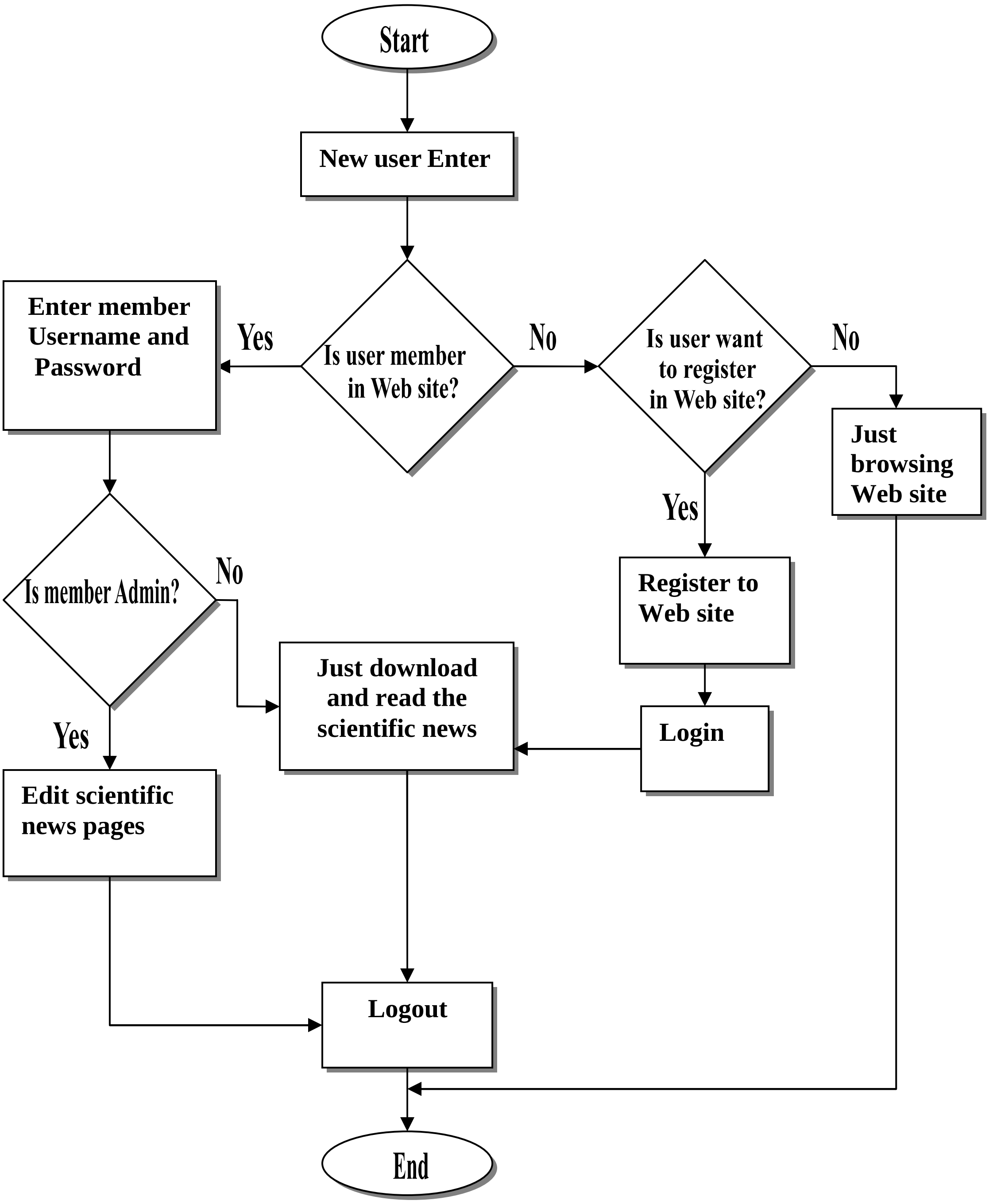

Start
New user Enter
Is user member in Web site?
Yes
No
Enter member Username and Password
Is user want to register in Web site?
No
Just browsing Web site
Yes
Register to Web site
Is member Admin?
No
Just download and read the scientific news
Login
Yes
Edit scientific news pages
Logout
End

Figure (4.4) Analysis Flowchart of the Proposed Dynamic Website

2. Registration Process Algorithm

For more analysis, the algorithm shown in figure (4.5) describes the registration process when any new user wants to be a member in the Website this is done as follows:

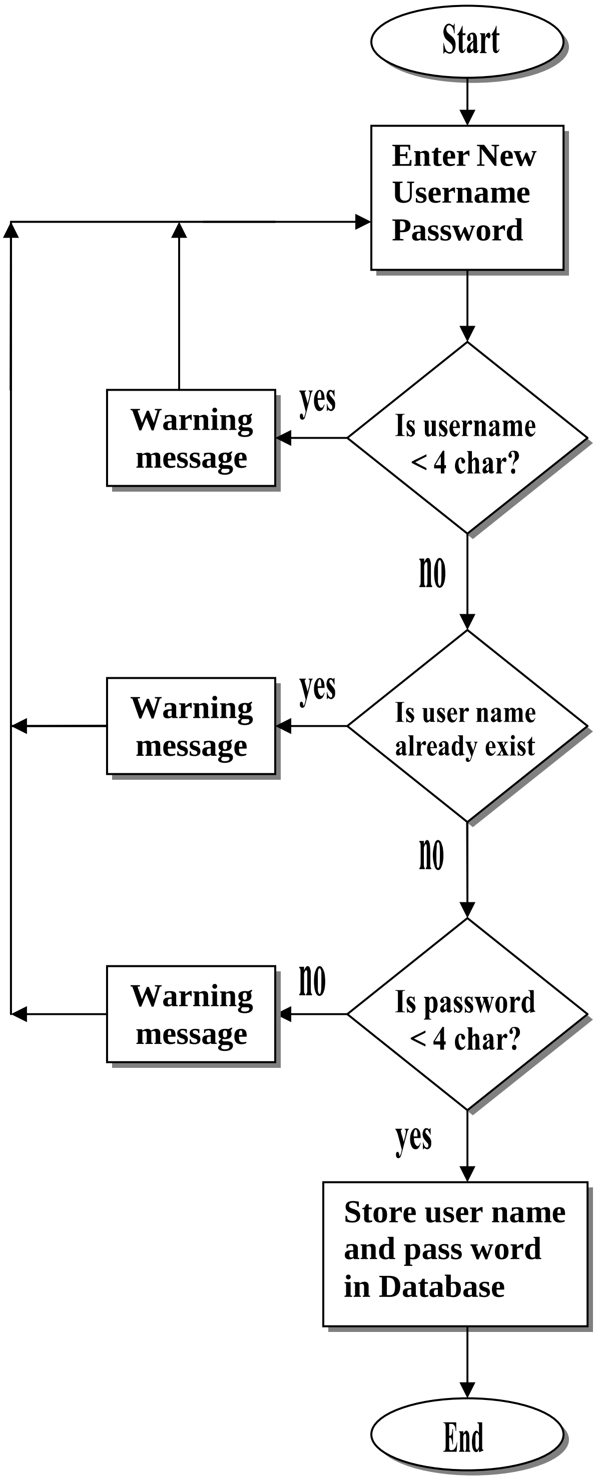


Figure (4.5) Registration Flowchart

- Any new user want to be a member in a Website, first he must be enter a username and a password, the Website check if that username and password is less than four characters, if so there is warning message appear to the user to change them and ask the user to enter new username and password.

- The system also checks if the username is already exists in the database if so, warning message appears to ask the user to enter new username and password.
- Then store the username and password in a database and that user is became a member in the Website.

3. Authorization Algorithm

This algorithm is shown in figure (4.6), it checks the user ID that entered the Website if it is a member or admin by checking the username and password and then give the authorization to that user.

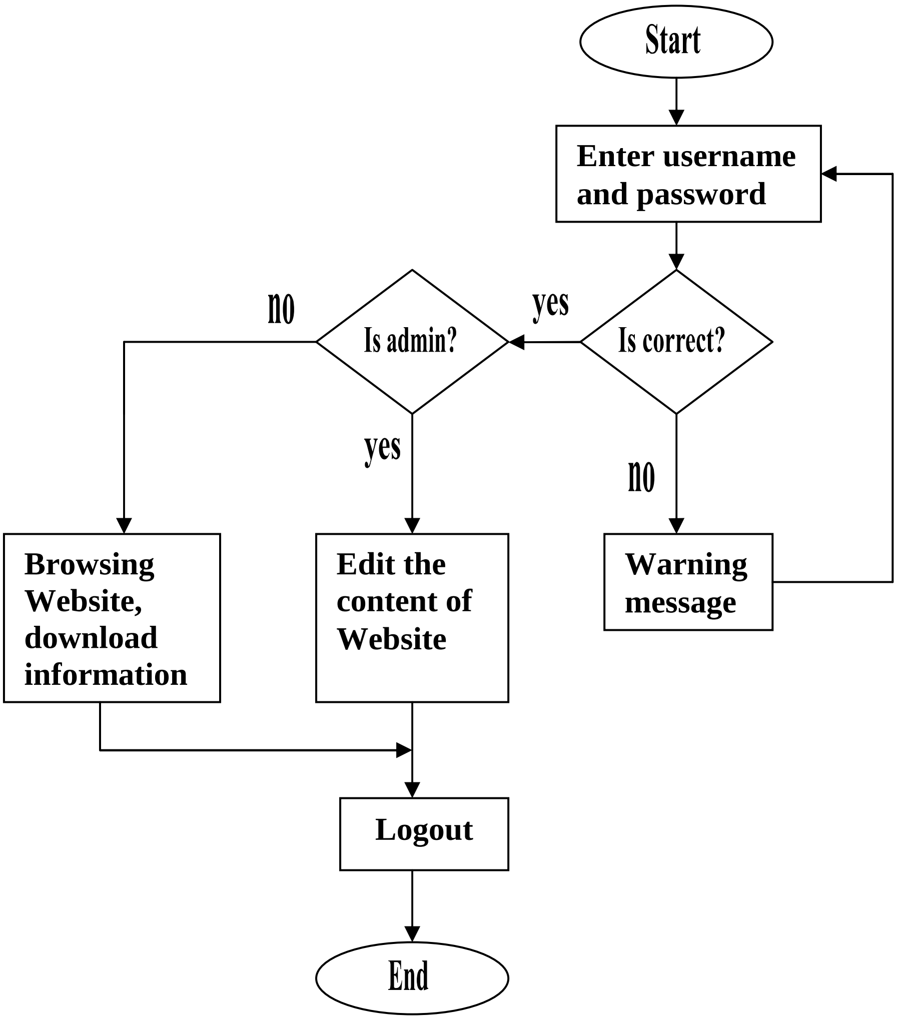

Start
Enter username and password
Is correct?
yes
no
Is admin?
no
yes
Browsing Website, download information
Edit the content of Website
Warning message
Logout
End

Figure (4.6) Authorization Flowchart

- The user enters its username and password, if it is correct; the system gives the authorization to that user.
- Check the user's authorization; if he is an admin then he can edit the content of Website, otherwise he can just browse the Website and download the information.
- If the username and password are not correct, a warning message will appear and retuning to the main page.

In the analysis phase, the Web engineer should understand the requirements of the Web application and ready to the next step to construct the Website.

4.2.3 Website Design

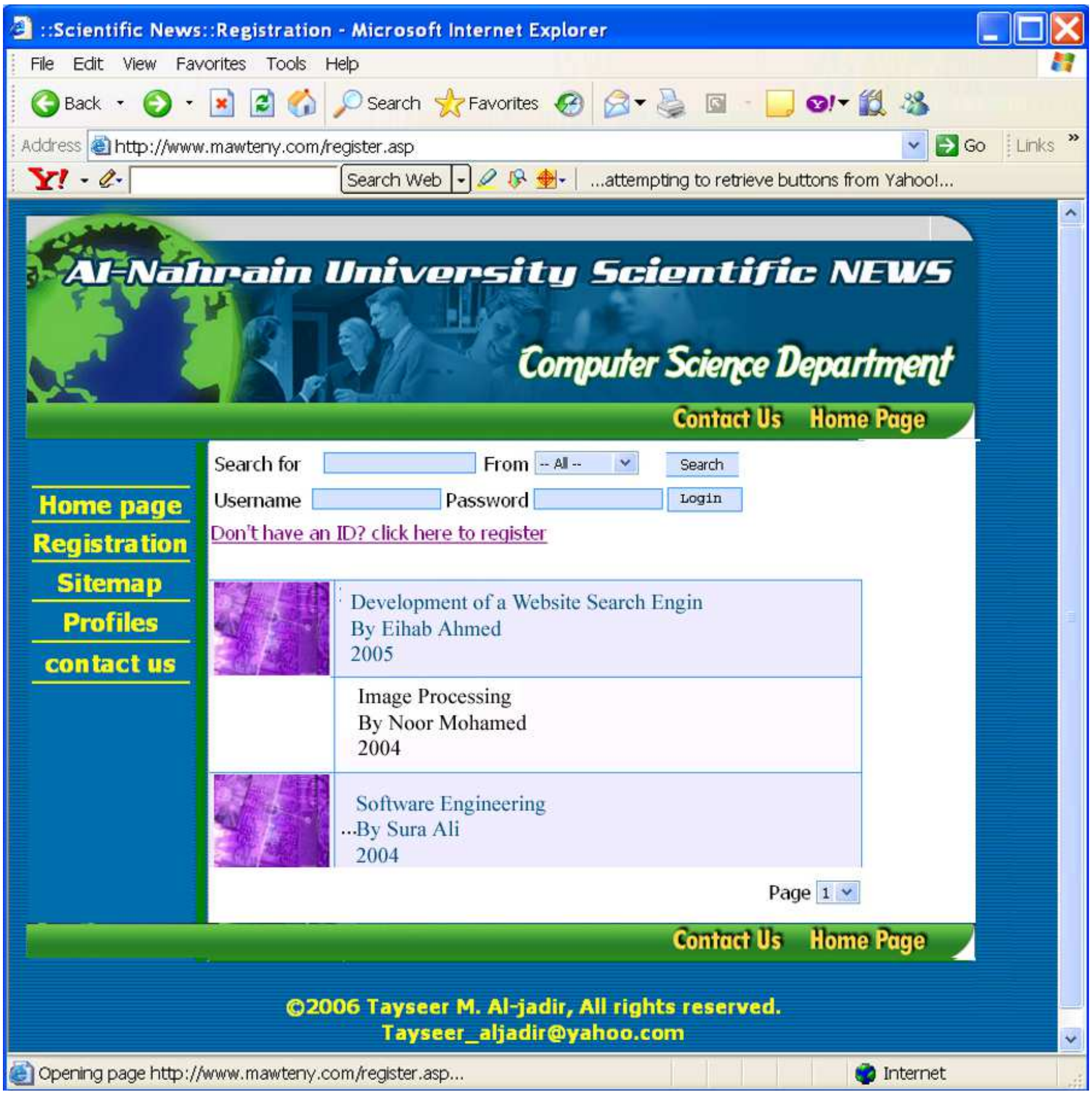
Design is the place where Web application quality is established. Website encompasses six major steps, see figure (2.1) Web engineering pyramid, the steps were used to design the proposed dynamic Website as follows:

a. Interface design

Interface page in the proposed dynamic Website is a good communication between the users and the Website. It gives answer to how the user is going to interact with the Website. Figure (4.7) shows interface page of proposed dynamic Website.

The interface of the proposed dynamic Website is connected with the database by adding member username and password. Also the interface is proved by testing when browsing a site that is easy to use and understand from every one browsing the site. The interface gives the user that enters the Website an idea about the holistic Website, what is the goal of that Website, which information the user can gain, what is the possible links related with that page.

The interface of any Website is the index page of that site. The index page of the proposed dynamic Website is called "*default.asp*".



::Scientific News::Registration - Microsoft Internet Explorer

Address http://www.mawteny.com/register.asp

...attempting to retrieve buttons from Yahoo!...

Al-Nahrain University Scientific NEWS

Computer Science Department

Search for From -- All --

Username Password

Don't have an ID? click here to register

Development of a Website Search Engin
By Eihab Ahmed
2005

Image Processing
By Noor Mohamed
2004

Software Engineering
...By Sura Ali
2004

Page 1

©2006 Tayseer M. Al-jadir, All rights reserved.
Tayseer_aljadir@yahoo.com

Opening page http://www.mawteny.com/register.asp...

Figure (4.7) Interface Page of Proposed Dynamic Website

b. Aesthetic design

The proposed dynamic Website has a good ration of aesthetics, with observation the subject of the Website that is very important point. The images, colors, fonts must be suitable with the subject of the Website. The proposed dynamic Website is scientific site, so the graphics must be chosen in a suitable way.

The banners flashing, graphics twirling, words scrolling, with a string of fireworks chasing your mouse pointer. It makes the user confused and hard to read the content of the Website specially the Site not for advertisement, so the graphical banner of proposed dynamic Website has suitable colors blue and green with an image denotable about the site subject, aesthetic design describes the look and feel of Website, so any new user enter the site may stay or search to another Site if the user don't like that site. Figure (4.8) shows the banner of the propose Website, this graphical banner designed by using Adobe Photoshop 7.0.

Al-Nahrain University Scientific NEWS
Computer Science Department
Contact Us
Home Page

Figure (4.8) Banner of Proposed Dynamic Website

c. Content Design

The proposed dynamic Website content includes text, graphics, and images. The content is either store on *.html* pages or stores in a database. The algorithm shown in figure (4.9), describes the connection between the Website pages and database.

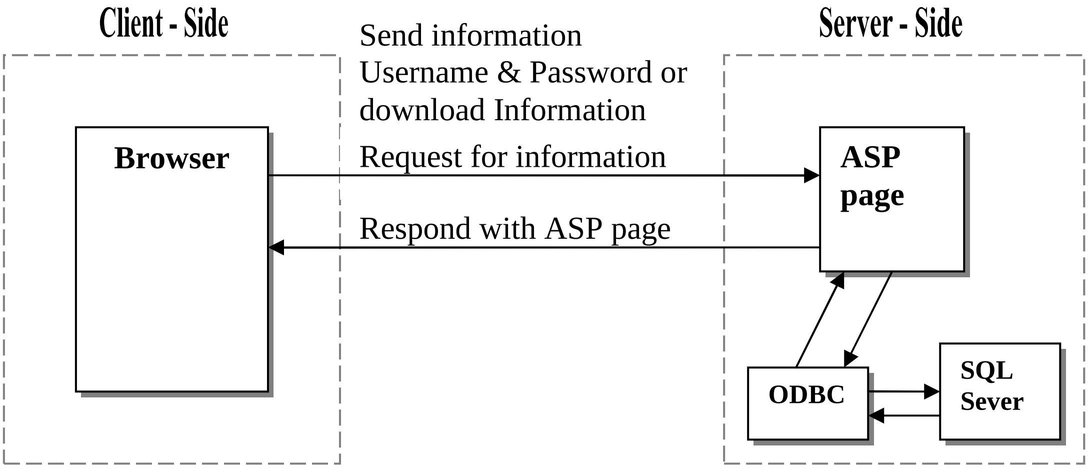


Figure (4.9) Connection between the Website and Database

Two tables are needed to store the content in the database:

1. Users table the format of this table is represented in table (4.1) as follows:

Table (4.1) Users Table

Id	Username (text)	Password (text)	JoinDate (date/time)	IsAdmin (as a flag)

i. Id: represents the unique assigned identification number for each user.

ii. Username: represents the name of the Website members of the Website.

iii. Password: represents the password of all Website members of the Website.

iv. JoinDate: represents the date and time of the registration.

v. IsAdmin: works as an acknowledgment flag to distinguish between the administrator and other members of proposed dynamic Website.

2. Posts table: the format of this table is represented in table (4.2) as follows:

Table (4.2) Posts Table

Id	Title text	PostDate (date/time)	PostHTML (Html ages)	HasImage (As a flag)

i. Id: represents the unique assigned identification number for each topic.

ii. Title: represents the title of the scientific news.

iii. PostDate: represents the published date and time of the scientific news.

iv. PostHTML: represents the details of the news.

v. HasImage: works as an acknowledgment flag to distinguish between if the topic has an image or not.

The research, papers, and thesis stored as PDF file to minimize the size of the text content and make them easy to download from internet. The amount of information in each page is suitable not too large, to help any user to access the necessary and useful information

All graphics and images content are stored as Joint Photograph Group (JPG) or Graphic Interchange File Format (GIF) to minimize the size and avoid the slow load time problems usually caused by too many graphics or un-optimized graphics.

d. Navigation Design

Many people enter a Website, and they get lost after three clicks, when that happens, the user searches another Website. The proposed dynamic Website avoid that by create navigation map over the Website, which connect all pages with each other. In every page, in the header part and footer part there is simple menu in the left of the Website with links to all pages in the Website, that avoid losing the user.

A *Contact us* button is placed in every page, to help the user if he has any question or suggestion about the Website.

e. Architecture Design

The hypertext organizational schemas is hierarchy or tree form. The users don’t need to see all information at once unless that user is a member in the Website. The hierarchical structure as shown in figure (4.9) is used to hide or expose as much information as necessary.

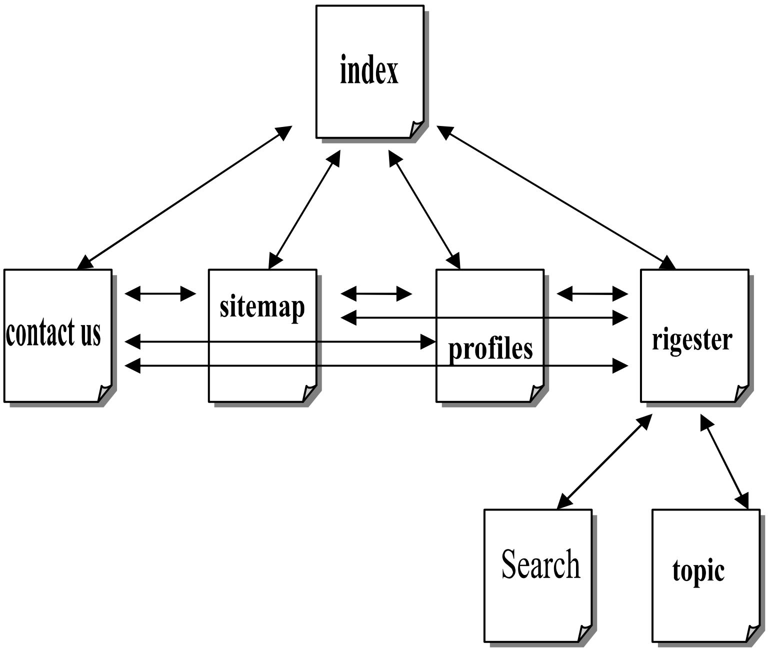


Figure (4.9) Hierarchical Structure

f. Component Design

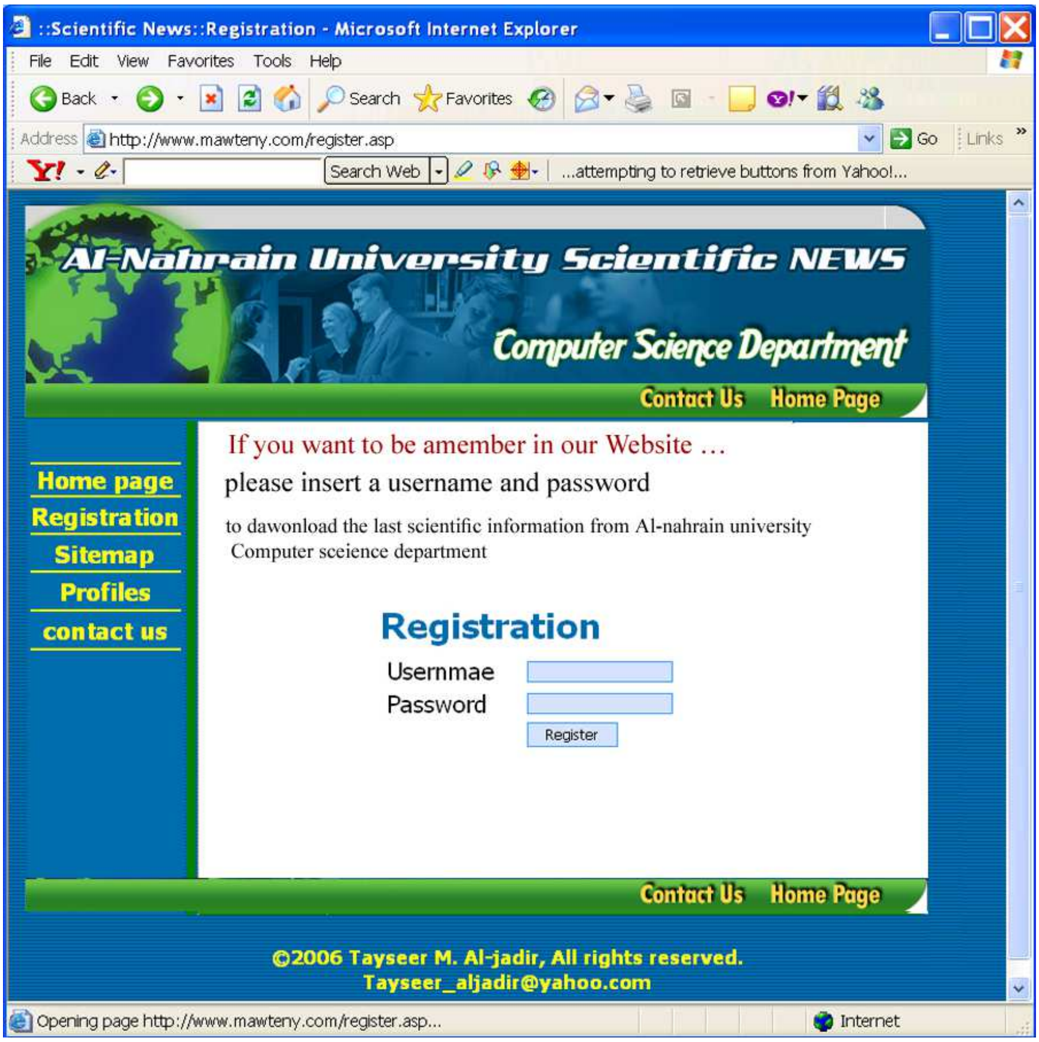
The implemented functional components required in the proposed dynamic Website are:

1. Perform localized processing to generate content and navigation capability in dynamic fashion.
2. Establish interfaces with database.
3. Provide connection between Website and database content.

4.2.4 Web Page Templates Design

A Web page template of the proposed dynamic Website shown in figure (4.10):

- ***Header section*** contains Website title, navigation links that connect all Web pages of the Website.
- ***Main body section*** contains the data entry fields and static information of Website.
- ***Footer section*** contain navigation links, last update date, and copyright marks.



::Scientific News::Registration - Microsoft Internet Explorer

Address http://www.mawteny.com/register.asp

Al-Nahrain University Scientific NEWS

Computer Science Department

If you want to be amember in our Website …

please insert a username and password

to dawonload the last scientific information from Al-nahrain university Computer sceience department

Registration

Usernmae

Password

©2006 Tayseer M. Al-jadir, All rights reserved.
Tayseer_aljadir@yahoo.com

Opening page http://www.mawteny.com/register.asp...

Figure (4.10) Header, Main body, and Footer Section

4.2.5 Website Implementation

In the implementation phase, the information and functionality must be coded in an appropriate format. The proposed dynamic Website use HTML documents to deliver static content, these HTML documents are written by using Microsoft FrontPage 2003 editor.

ASP used to implement server-side scripting language imbedded into HTML code, to construct the pages of Website. On server-side the process involves reach to database from ASP pages of Website, by using open database connectivity ODBC to connect Web pages with database.

The database consists of number of tables that are arranged so as to facilitate faster retrieval of the data. This database is housed in a database server. The proposed dynamic Website uses a Web server as database server because of the small number of Web pages. Each member by enter username and password from Web page; check them with the database by using ODBC.

JavaScript on the other hand, is embedded into HTML and interpreted locally on the user browser.

4.2.6 Testing the Proposed Dynamic Website

The first step toward testing is publishing proposed dynamic Website on an offline Web server (Web server not connected with the Internet).

The Website implemented on network that works on TCP/IP protocol, the proposed dynamic Website work by client/server technique. The proposed dynamic Website published offline on computer system work as client and server at the same time.

Client side must have Internet Explorer and the server side must be have Windows Server 2000 NT and Microsoft SQL Server 7.0. The proposed dynamic Website works on the offline server and the Website engineer must be find the errors and repair them before published proposed dynamic Website on the Internet. The testing was done on the following components:

- The static content of Website (HTML Pages).
- The navigation links of all pages of Website.
- Check the connection between database with Website pages, that is done by checking the dynamic content that stored by database and check (add, delete, and edit) on these content.
- The connection of users with Website database (admin, members, gust user) by adding testing usernames and passwords to database of the proposed dynamic Website, and enter the proposed dynamic Website by these testing username and password.

After checking the proposed dynamic Website from the errors on the offline server, publish it on the Internet.

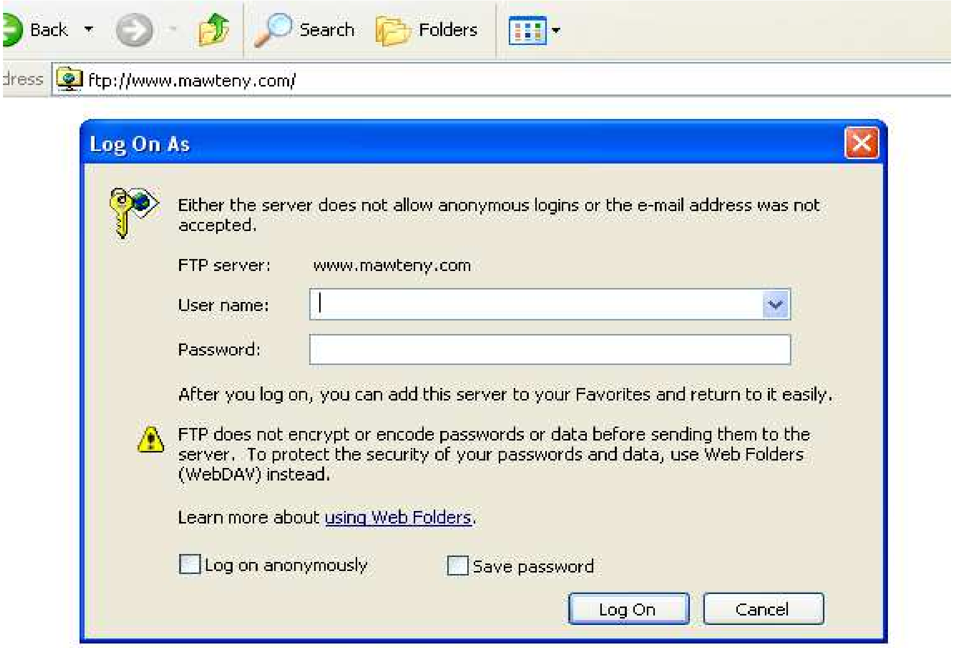
4.3 Publishing Website

To be a part of World Wide Web (WWW), the Website must be published or (uploaded) to Web server.

To publish online, the computer must be connected to the internet, and need an account from an Internet service provider (ISP).

Publishing is the process of copying all files that makes up Website on the proper computer. That is done by either using appropriate program help to transfer the Website to Web server or by using File Transfer Protocol (FTP) by using Web browser.

The proposed dynamic Website published through Web browser address ***ftp://www.mawteny.com*** as shown in figure (4.12), a log on windows will appear and ask about the user name and password, after write them, the user must click on the ***log on*** button. The Web server page will appear with all pages and folder that stored in the server, then the puplisher must copy all pages from his computer to the Web server as shown in figure (4.13).



ftp://www.mawteny.com/

Log On As

Either the server does not allow anonymous logins or the e-mail address was not accepted.

FTP server: www.mawteny.com

User name:

Password:

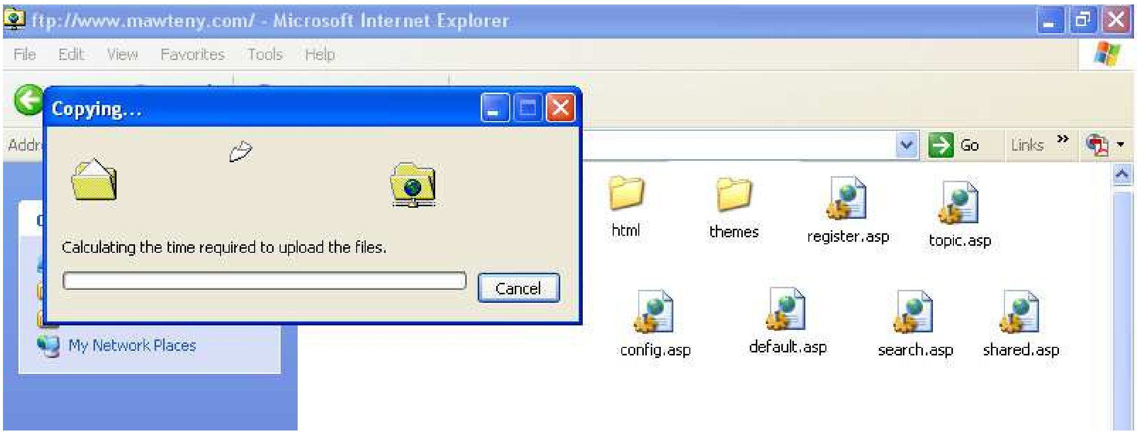
After you log on, you can add this server to your Favorites and return to it easily.

FTP does not encrypt or encode passwords or data before sending them to the server. To protect the security of your passwords and data, use Web Folders (WebDAV) instead.

Learn more about using Web Folders.

□ Log on anonymously □ Save password

Figure (4.12) FTP on Web Browser



Calculating the time required to upload the files.

Figure (4.13) Web Server content

4.4 Comparison Between the Proposed Dynamic Website with other Websites

The proposed dynamic Website was constructed under specifications of Web engineering. This section consider the proposed dynamic Website as a standard Website for other Websites on the World Wide Web, then choose Websites from the Internet, to find if these Websites construct under the specification of the Web engineering. That is done by assigning some of Web engineering characteristics and experiment the chosen Websites, to find how many of these characterstics were applied on these sites, each characteristic has rate degree for comparison depend upon its important, and finally find out the final degree (the percentage degree is between 0-100).

As discussed in chapter two (the principles of Web engineering) and applying these principles on proposed dynamic Website, the first section of chapter four, there are many characteristics in the Web engineering must be applied when construct any Website is going to be constructed, some of these Web engineering characteristics are chosen according to its importance. The reasons of choosing these characteristics are as follows:

1. Content Design

Content include text, graphics, images, video and audio, When any user browsing Website without useful content or they suffering to find the information they want, they search to another Website, so the Web engineer must be careful about many point:

a- Web pages should not be large in order to avoid horizontal and very large vertical scrolling.

b- Poor spelling.

c- Plenty of music.

d- Using huge flash into screen and lots of moving thing like (blinking text or over use of banner advertisements)

e- Out dated information.

The estimated percentage degree of this point is 25%.

2. Navigation Design

To avoid losing users, all pages in the Website must be linked together by correct hyperlinks. The Web engineering must start Website design with a good understanding of the structure of the information space and communicate this structure explicitly to the user. Providing a site map enable users know where they are and where they can go. The estimated percentage degree for this Web engineering characteristics is 20%.

3. Interface Design

Interface of any Website means the index page of that Website, which means it is the first page that any user visit it when he enter the URL of the Website in the address bar of the Web browser. Index page gives the answer to how the user is going to interact with the Website. It is very important characteristics in Web engineering, the estimated percentage degree is 20%.

4. Unique Template

One of the best ways to ensure a consistent application look and feel is to develop a Web page template that is very important to technique provides the users with a secure sense of consistency and structure while interfacing with the graphical user interface.

For instance, the page header and footer sections should contain information that establishes clearly the identity of the Website. It also helps the users to quickly situate themselves visually in the Web application structure. The users should never get confused as to where they located in the Web application. The estimated percentage degree is 10%.

5. Goals of Website

Each Website has a specific goal; the reasons behind constructing the Website, and is to be easy to every one how visiting the Website understand its purpose. The estimated percentage degree is 5%.

6. Search Engine

Provide search engine to Website is very important point. Some Website has large number of Web pages and other Website has no more than five pages, the necessary of adding search engine is different from one Website to another.

Therefore the need of search engine is extrusive proportion depend upon the number of Web pages of the Website. The estimated percentage degree has a range from (1-10%).

7. Download Time

When any user waiting for downloads Website to appear on its computer for long time the user maybe searches to another Website. Traditional human factors guidelines indicate 10-15 seconds as the maximum response time before users lose interest, the Web programmer must avoiding a large number of images, using audio or video that help to slow down time of the Website.

It is extrusive proportion, when images, audio, and video are increasing the download time is increasing also. The estimated percentage degree is range from (1-10%).

These characteristics are used to compare between the proposed dynamic Website and the chosen Websites download from the internet. Three different Websites are chosen, illustrated in table (4.3) as follows:

Table (4.3) Websites Table

	Website Name	Description	URL of Website
1	“How do they do that with HTML”	Tutorial Website, learning some basic of HTML language	http://www.tashian.com
2	Cornell University	Website about Cornell University	http://www.cornell.edu
3	David Chalmers	Personal Website about Professor of Philosophy	http://consc.net

A. Comparison of “How do they do that with HTML” Website

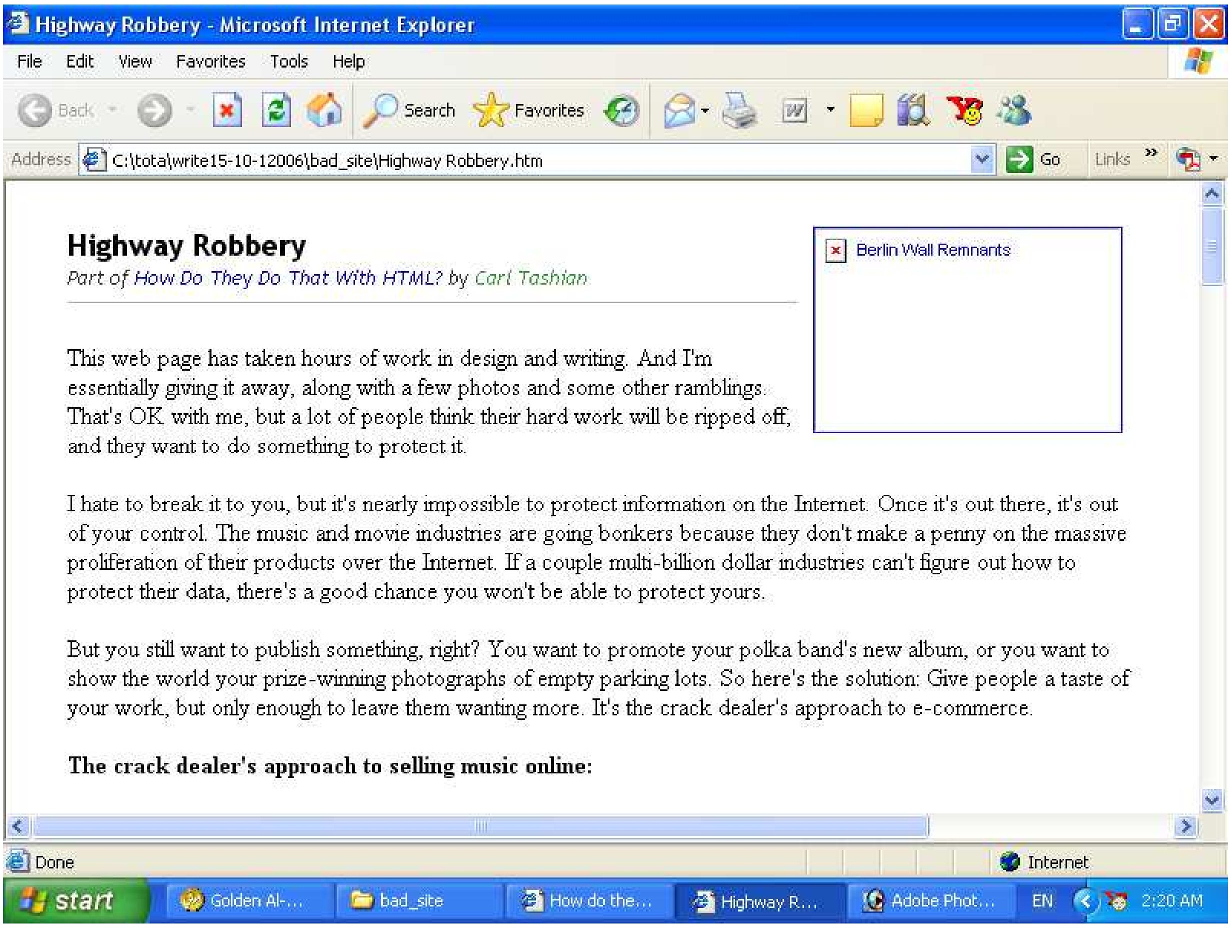
1. Content Design

As shown in figure (4.14) the “How do they do that with HTML” Website has the following description:

- Has too much information in one page with vertical scrolling.
- There is lost image in the top of pages.
- No error in spelling.
- No huge flash into screen or moving components like (blinking text or over use of banner advertisement).

- The information is updated, the last update is 8/6/2006.

The estimated percentage degree is 20% from 25%.



Highway Robbery - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\tota\write15-10-12006\bad_site\Highway Robbery.htm
Go Links
Berlin Wall Remnants
Done Internet
start Golden Al-... bad_site How do the... Highway R... Adobe Phot... EN 2:20 AM

Highway Robbery

Part of How Do They Do That With HTML? by Carl Tashian

This web page has taken hours of work in design and writing. And I'm essentially giving it away, along with a few photos and some other ramblings. That's OK with me, but a lot of people think their hard work will be ripped off, and they want to do something to protect it.

I hate to break it to you, but it's nearly impossible to protect information on the Internet. Once it's out there, it's out of your control. The music and movie industries are going bonkers because they don't make a penny on the massive proliferation of their products over the Internet. If a couple multi-billion dollar industries can't figure out how to protect their data, there's a good chance you won't be able to protect yours.

But you still want to publish something, right? You want to promote your polka band's new album, or you want to show the world your prize-winning photographs of empty parking lots. So here's the solution: Give people a taste of your work, but only enough to leave them wanting more. It's the crack dealer's approach to e-commerce.

The crack dealer's approach to selling music online:

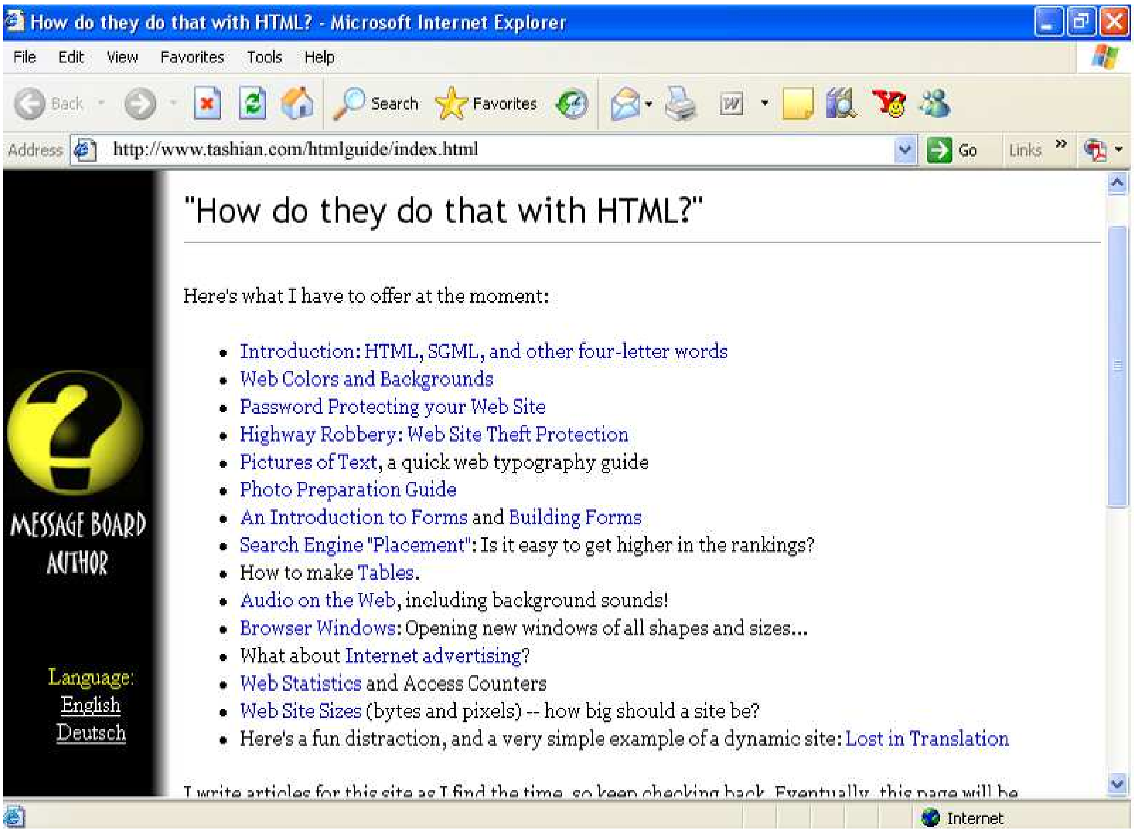
Figure (4.14) One Page of Website

2. Navigation Design

The “How do they do that with HTML” Website has wrong hyperlink in the index pages; also the number of Web pages is 26 pages and has no *Site Map* help the users to navigate all pages of the Website. The estimated percentage degree is 10% from 20%.

3. Interface Design

The interface page of the “How do they do that with HTML” Website it is index page; it contains a list of all links that connected all pages with the index page, with simple banner on the left side of the index page. The estimated percentage degree is 15% from 20%. Figure (4.15) shows interface page of “How do they do that with HTML” Website.



How do they do that with HTML? - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://www.tashian.com/htmlguide/index.html
Go
Links
MESSAGE BOARD
AUTHOR
Language:
English
Deutsch
Internet

"How do they do that with HTML?"

Here's what I have to offer at the moment:

- Introduction: HTML, SGML, and other four-letter words
- Web Colors and Backgrounds
- Password Protecting your Web Site
- Highway Robbery: Web Site Theft Protection
- Pictures of Text, a quick web typography guide
- Photo Preparation Guide
- An Introduction to Forms and Building Forms
- Search Engine "Placement": Is it easy to get higher in the rankings?
- How to make Tables.
- Audio on the Web, including background sounds!
- Browser Windows: Opening new windows of all shapes and sizes...
- What about Internet advertising?
- Web Statistics and Access Counters
- Web Site Sizes (bytes and pixels) -- how big should a site be?
- Here's a fun distraction, and a very simple example of a dynamic site: Lost in Translation

I write articles for this site as I find the time, so keep checking back. Eventually, this page will be

Figure (4.15) Interface Page

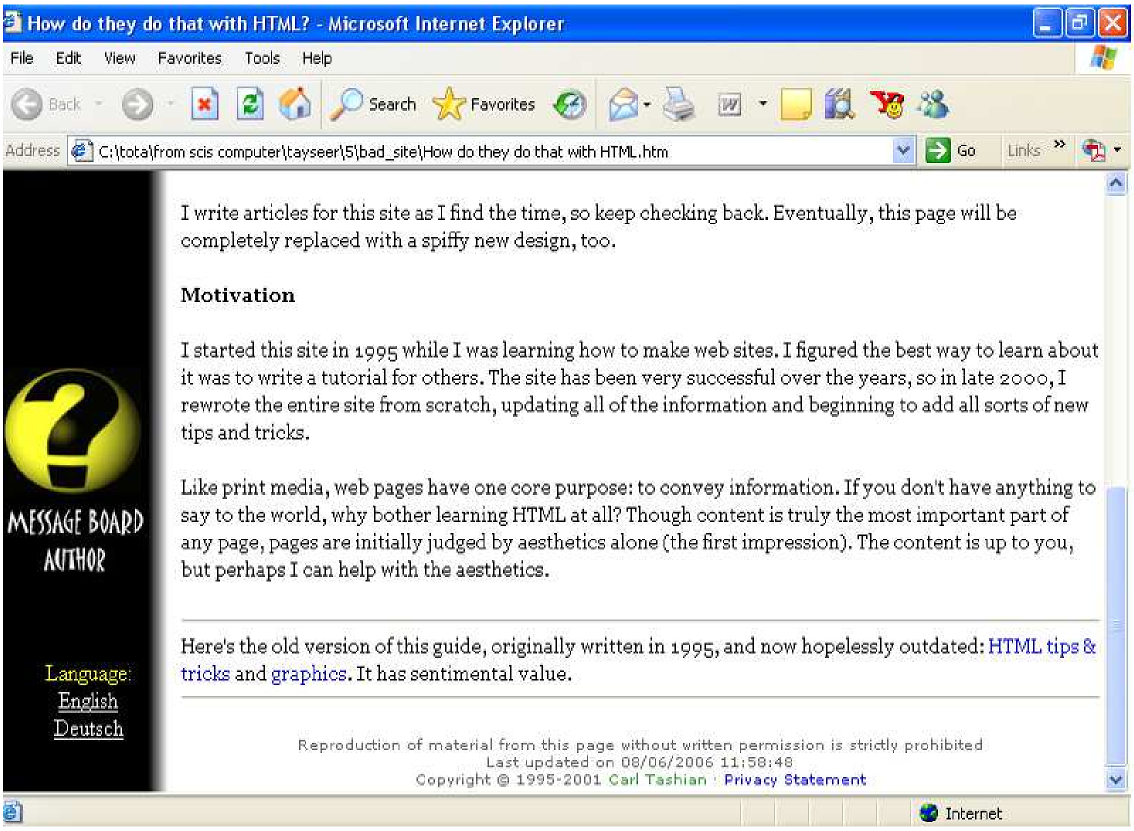
4. Unique Template

- There is no unique template.
- There is no header section contain the name and logo of the Website.
- Simple main body.
- Footer section has good information.

Figure (4.16) shows the index page of the t “How do they do that with HTML” Website establishes footer section. The estimated percentage degree is 4% from 10%.

5. Goals of Website

The goal of the “How do they do that with HTML” Website is clear to the user at the first time enter the Website, is a tutorial Website about HTML language. The estimated percentage degree is 5% from 5%.


How do they do that with HTML? - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\tota\from scis computer\tayseer\5\bad_site\How do they do that with HTML.htm
Go Links
MESSAGE BOARD
AUTHOR
Language:
English
Deutsch
Internet

I write articles for this site as I find the time, so keep checking back. Eventually, this page will be completely replaced with a spiffy new design, too.

Motivation

I started this site in 1995 while I was learning how to make web sites. I figured the best way to learn about it was to write a tutorial for others. The site has been very successful over the years, so in late 2000, I rewrote the entire site from scratch, updating all of the information and beginning to add all sorts of new tips and tricks.

Like print media, web pages have one core purpose: to convey information. If you don't have anything to say to the world, why bother learning HTML at all? Though content is truly the most important part of any page, pages are initially judged by aesthetics alone (the first impression). The content is up to you, but perhaps I can help with the aesthetics.

Here's the old version of this guide, originally written in 1995, and now hopelessly outdated: HTML tips & tricks and graphics. It has sentimental value.

Reproduction of material from this page without written permission is strictly prohibited
Last updated on 08/06/2006 11:58:48
Copyright © 1995-2001 Carl Tashian · Privacy Statement

Figure (4.16) Footer Section

6. Search Engine

The “How do they do that with HTML” Website has no searching tool. The number of Web pages is 26 pages. Those numbers of Web pages need a search engine to help users find the information need from that Website. The search engine is necessary and the Website has no search engine so, the estimated percentage degree is 2%.

7. Download Time

The “How do they do that with HTML” Website has suitable number of pictures, no video or audio. So the download time needed to appear the Website on Web browser take a traditional time, it has a good download time. The estimated percentage degree is 8%.

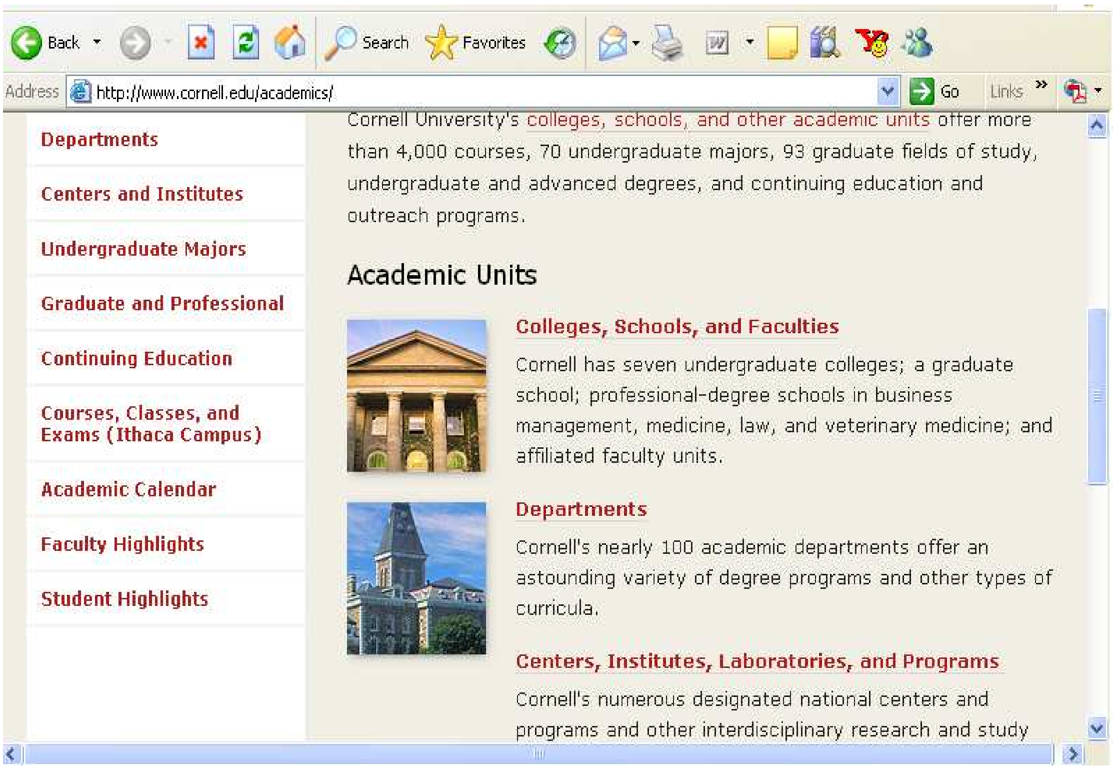
B. Comparison of “Cornell University” Website

1. Content Design

As shown in figure (4.17) the *Cornell University* Website has the following describe:

- Has suitable information in each Web page with suitable vertical scrolling.
- There are no loosing images in all Website pages.
- No error in spelling.
- No huge flash into screen or moving components like (blinking text or over use of banner advertisement).
- The information is updated. The last update was in 2006.

The estimated percentage degree is 23% from 25%.



Address http://www.cornell.edu/academics/

Departments

Centers and Institutes

Undergraduate Majors

Graduate and Professional

Continuing Education

Courses, Classes, and Exams (Ithaca Campus)

Academic Calendar

Faculty Highlights

Student Highlights

Cornell University's colleges, schools, and other academic units offer more than 4,000 courses, 70 undergraduate majors, 93 graduate fields of study, undergraduate and advanced degrees, and continuing education and outreach programs.

Academic Units

Colleges, Schools, and Faculties

Cornell has seven undergraduate colleges; a graduate school; professional-degree schools in business management, medicine, law, and veterinary medicine; and affiliated faculty units.

Departments

Cornell's nearly 100 academic departments offer an astounding variety of degree programs and other types of curricula.

Centers, Institutes, Laboratories, and Programs

Cornell's numerous designated national centers and programs and other interdisciplinary research and study

Figure (4.17) Web Page Content

2. Navigation Design

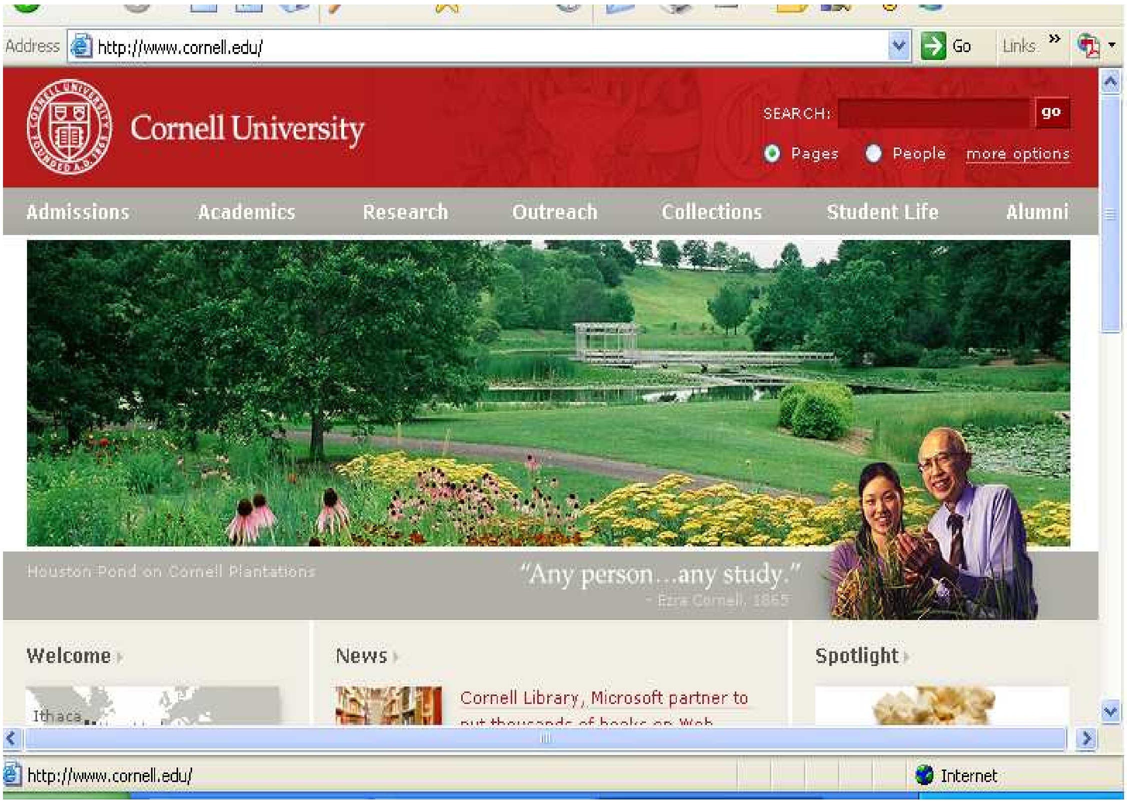
- The *Cornell University* chosen Website has no wrong hyperlink.
- The number of Web pages in *Cornell University* Website is 40 pages and the Website has no *Site Map*, to help the users navigate in all pages of the Website. The estimated percentage degree is 10% from 20%.

3. Interface Design

Figure (4.18) shows the interface page of the *Cornell University* Website:

- It contains a list of all links that connected all pages with each other, in the header, footer and left side of index page.
- It has a good interface that helps all users to interact with the Website.

The estimated percentage degree is 18% from 20%.



Houston Pond on Cornell Plantations

"Any person...any study."
- Ezra Cornell, 1865

Cornell Library, Microsoft partner to put thousands of books on Web

Figure (4.18) Interface Page

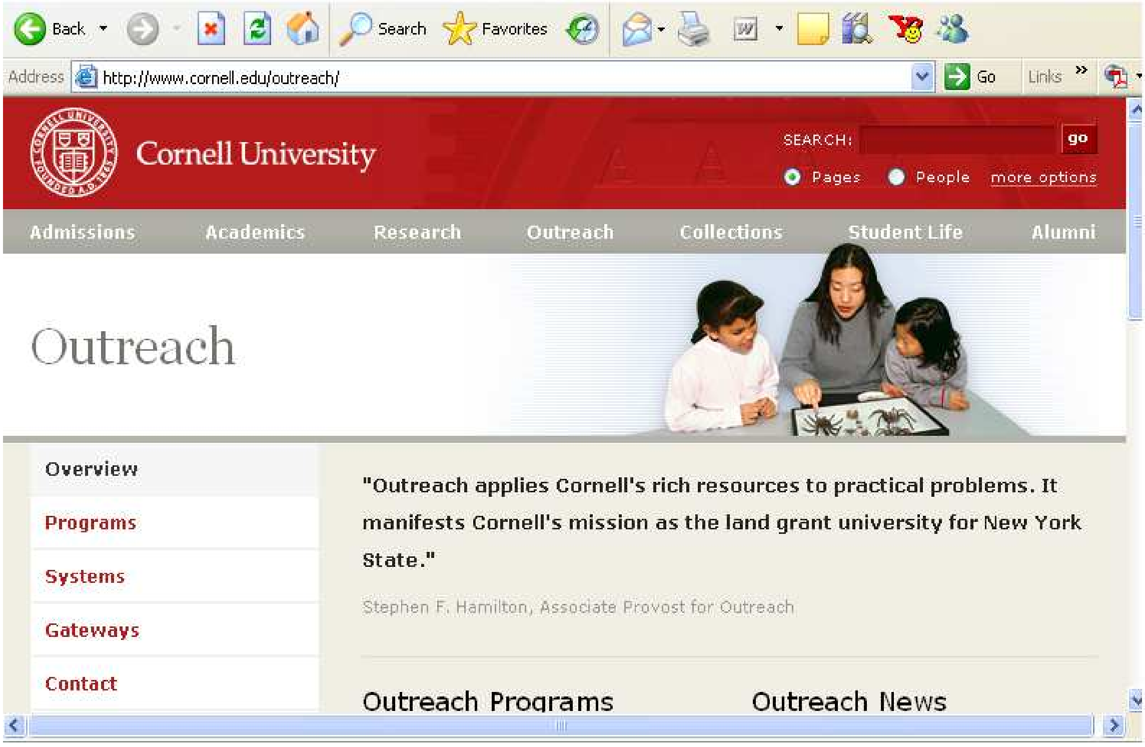
4. Unique Template

- There is a unique template.
- There is simple header section contain the name and logo of the Website. Figure (4.19) shows the Web page of the *Cornell University* Website establish header section.
- Simple main body.
- Footer section has good information. See figure (4.20) shows the one Web page of the *Cornell University* Website establish footer section.

The estimated percentage degree is 10% from 10%.

5. Goals of Website

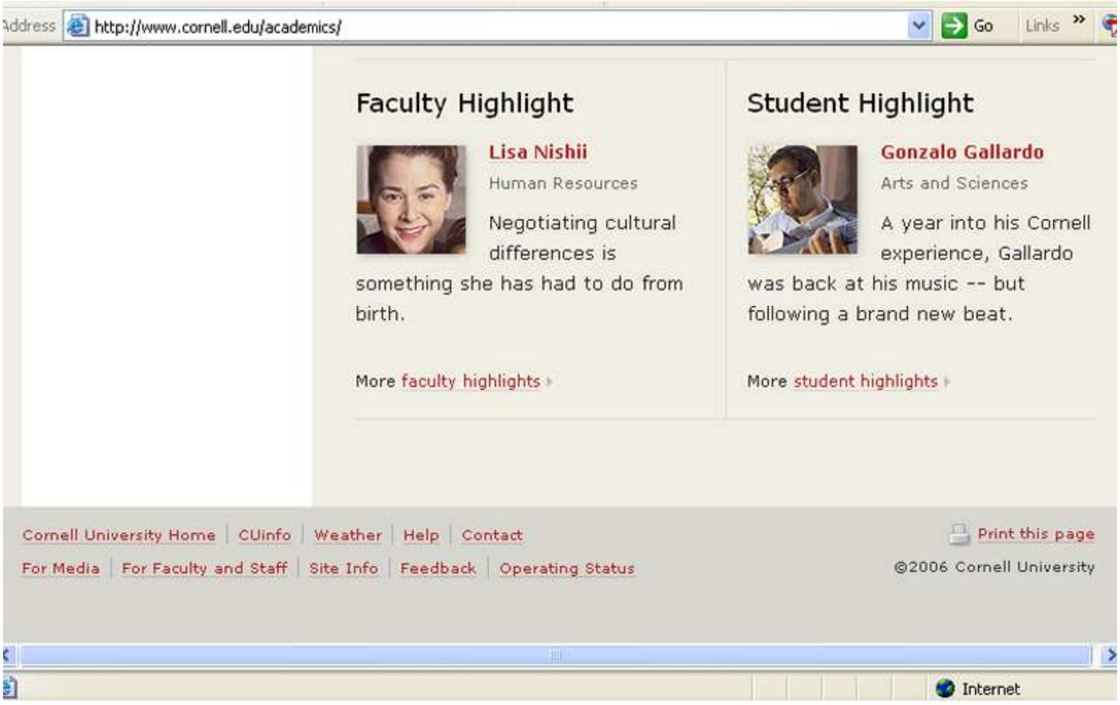
The Website talks about *Cornell University*, and helps users to know all information about that university. The goal of Website is clear. The estimated percentage degree is 5% from 5%.



"Outreach applies Cornell's rich resources to practical problems. It manifests Cornell's mission as the land grant university for New York State."

Stephen F. Hamilton, Associate Provost for Outreach

Figure (4.19) Header Section



Address http://www.cornell.edu/academics/ Go Links

Faculty Highlight

Lisa Nishii

Human Resources

Negotiating cultural differences is something she has had to do from birth.

More faculty highlights

Student Highlight

Gonzalo Gallardo

Arts and Sciences

A year into his Cornell experience, Gallardo was back at his music -- but following a brand new beat.

More student highlights

Cornell University Home | CUinfo | Weather | Help | Contact

For Media | For Faculty and Staff | Site Info | Feedback | Operating Status

Print this page

©2006 Cornell University

Internet

Figure (4.20) Footer Section

6. Search Engine

The *Cornell University* Website has search engine, in the banner of all pages of the *Cornell University* Website see figure (4.21). The number of Web pages of the Website is 40 pages. Those Web pages need a search engine to help users find the needed information from that Website. The search engine is necessary and the Website has search engine so, the estimated percentage degree is 8% from 10%.

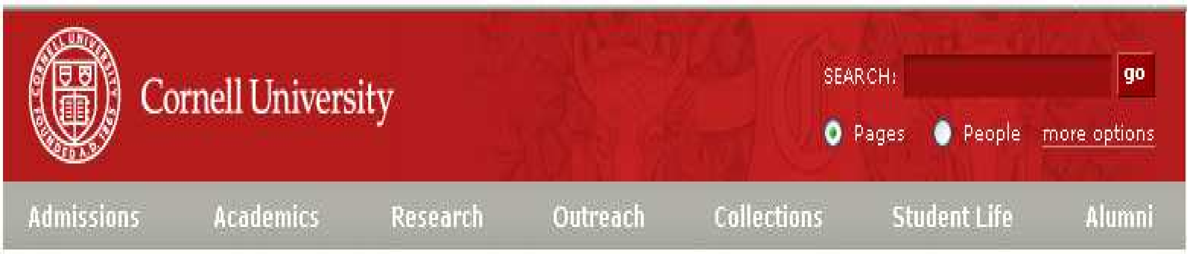


Figure (4.21) Search Engine

9. Download Time

The *Cornell University* Website has a suitable number of pictures, no video or audio. So the download time needed to appear the Website on Web browser take a traditional time, it has a good download time. The estimated percentage degree is 8% from 10%.

C. Comparison of “David Chalmers” Website

1. Content Design

As shown in figure (4.22) the *David Chalmers* Website has the following description:

- Has too much information in one page with vertical scrolling.
- There are no loosing images in all Website pages.
- No error in spelling.
- No huge flash into screen or moving components like (blinking text or over use of banner advertisements).
- The information is updatable.

The estimated percentage degree is 20% from 25.

2. Navigation Design

- The *David Chalmers* Website with no wrong hyperlink.
- The number of Web pages in *David Chalmers* Website is 20 pages and the Website has no *Site Map,* to the help users navigate in all pages of the Website. The estimated percentage degree is 10% from 20%.

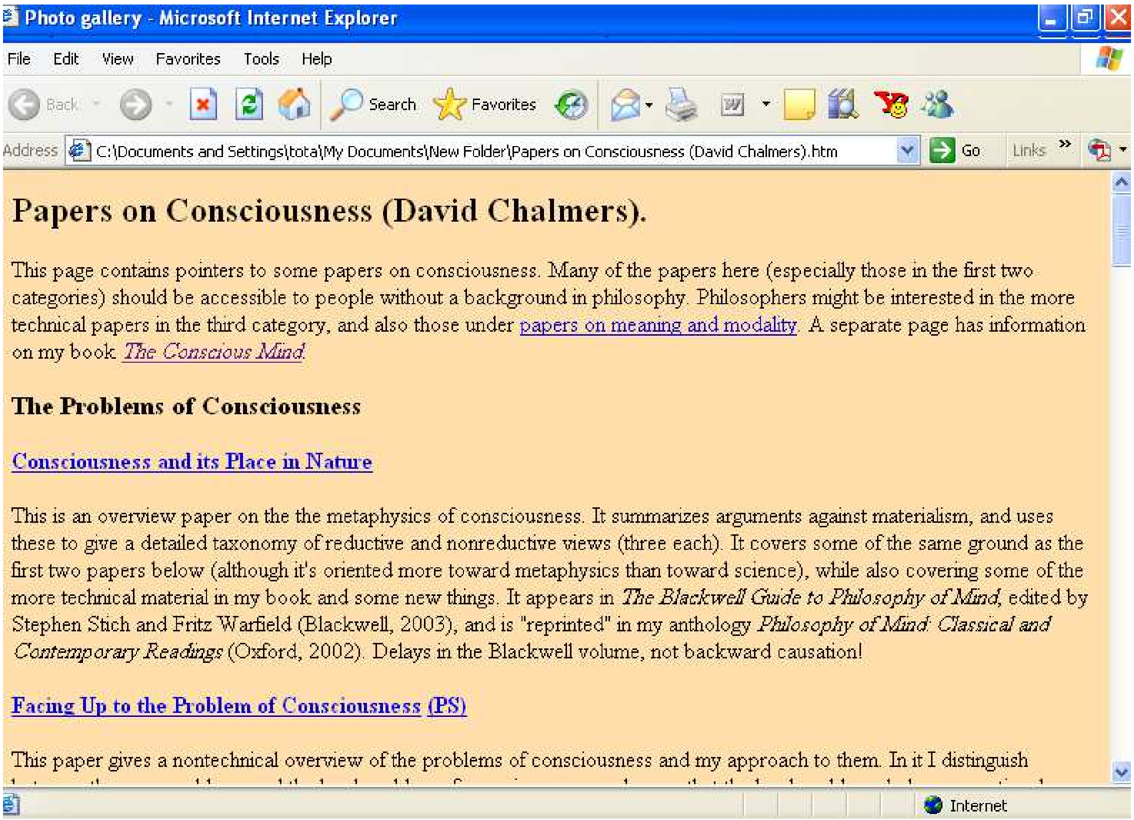


Photo gallery - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\Documents and Settings\tota\My Documents\New Folder\Papers on Consciousness (David Chalmers).htm
Go Links
Internet

Papers on Consciousness (David Chalmers).

This page contains pointers to some papers on consciousness. Many of the papers here (especially those in the first two categories) should be accessible to people without a background in philosophy. Philosophers might be interested in the more technical papers in the third category, and also those under papers on meaning and modality. A separate page has information on my book *The Conscious Mind*.

The Problems of Consciousness

Consciousness and its Place in Nature

This is an overview paper on the the metaphysics of consciousness. It summarizes arguments against materialism, and uses these to give a detailed taxonomy of reductive and nonreductive views (three each). It covers some of the same ground as the first two papers below (although it's oriented more toward metaphysics than toward science), while also covering some of the more technical material in my book and some new things. It appears in *The Blackwell Guide to Philosophy of Mind*, edited by Stephen Stich and Fritz Warfield (Blackwell, 2003), and is "reprinted" in my anthology *Philosophy of Mind: Classical and Contemporary Readings* (Oxford, 2002). Delays in the Blackwell volume, not backward causation!

Facing Up to the Problem of Consciousness (PS)

This paper gives a nontechnical overview of the problems of consciousness and my approach to them. In it I distinguish

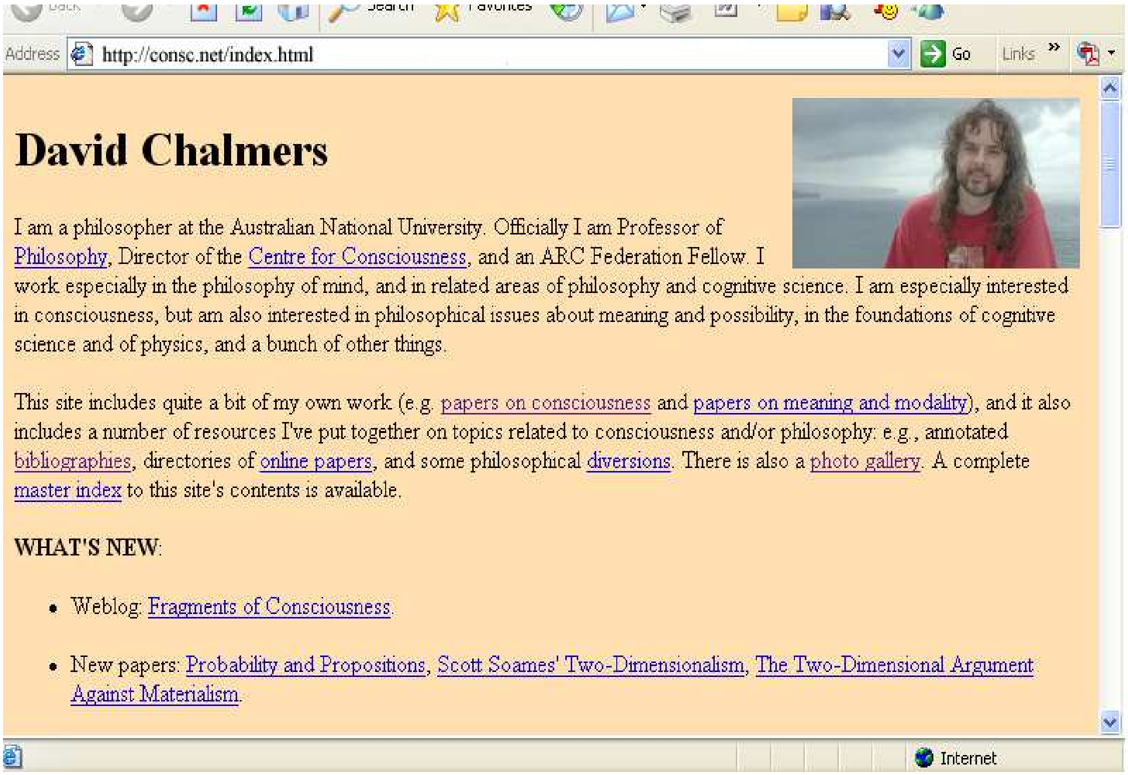
Figure (4.22) One Page of the *David Chalmers* Website

3. Interface Design

Figure (4.23) shows the interface page of the *David Chalmers* Website:

- It contains a list of all links that connected all pages with the index page, and with other pages of Website.
- No regular hyperlink list with the pages of Website.
- No banner in Web pages and the Website name are only on the index page.
- All pages are just document of information.
- The index page contains an explanation about the subject, goal and author of the Website.

The estimated percentage degree is 12% from 20%.



http://consc.net/index.html

David Chalmers

I am a philosopher at the Australian National University. Officially I am Professor of Philosophy, Director of the Centre for Consciousness, and an ARC Federation Fellow. I work especially in the philosophy of mind, and in related areas of philosophy and cognitive science. I am especially interested in consciousness, but am also interested in philosophical issues about meaning and possibility, in the foundations of cognitive science and of physics, and a bunch of other things.

This site includes quite a bit of my own work (e.g. papers on consciousness and papers on meaning and modality), and it also includes a number of resources I've put together on topics related to consciousness and/or philosophy: e.g., annotated bibliographies, directories of online papers, and some philosophical diversions. There is also a photo gallery. A complete master index to this site's contents is available.

WHAT'S NEW:

- Weblog: Fragments of Consciousness.
- New papers: Probability and Propositions, Scott Soames' Two-Dimensionalism, The Two-Dimensional Argument Against Materialism.

Figure (4.23) Interface Page

4. Unique Template

- There is no unique template.
- There is no header section contain the name and logo of the Website.
- Simple main body.
- No Footer section, just in index page only. Figure (4.24) shows the footer of the index page. The estimated percentage degree is 2% from 10%.

5. Goals of Website

The goal of the *David Chalmers* Website is clear to the user at the first time he enter the Website, it is a personal Website about professor of philosophy, the Website contain the books and papers of the professor and others philosophers. The estimated percentage degree is 5% from 5%.

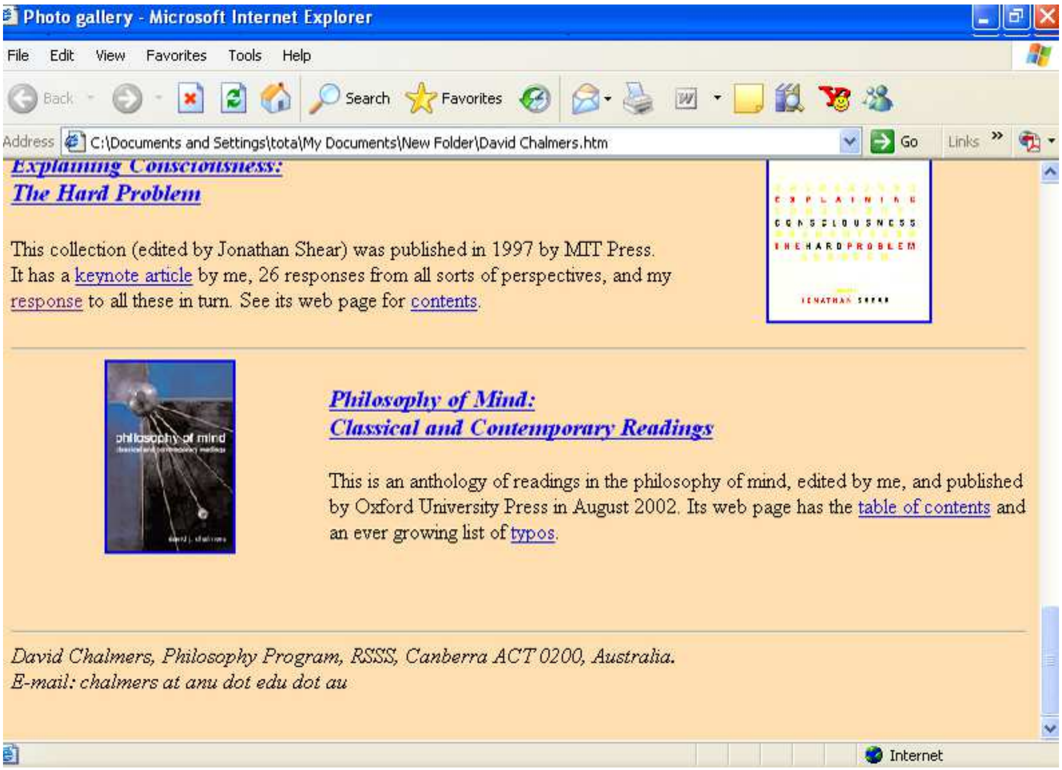


Photo gallery - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\Documents and Settings\tota\My Documents\New Folder\David Chalmers.htm
Go Links
EXPLAINING CONSCIOUSNESS THE HARD PROBLEM
JONATHAN SHEAR
philosophy of mind
david j. chalmers
Internet

Explaining Consciousness:
The Hard Problem

This collection (edited by Jonathan Shear) was published in 1997 by MIT Press. It has a keynote article by me, 26 responses from all sorts of perspectives, and my response to all these in turn. See its web page for contents.

Philosophy of Mind:
Classical and Contemporary Readings

This is an anthology of readings in the philosophy of mind, edited by me, and published by Oxford University Press in August 2002. Its web page has the table of contents and an ever growing list of typos.

David Chalmers, Philosophy Program, RSSS, Canberra ACT 0200, Australia.
E-mail: chalmers at anu dot edu dot au

Figure (4.24) the Footer Section

6. Search Engine

The *David Chalmers* Website has a search engine, in only one Web page, that page helps the user to find the photos from a list of photo gallery stored in a database. Figure (4.25) shows the search engine of the Web page. The number of Web pages of this Website is 20 pages. Those Web pages need a search engine to help users finding their needed information from the Website. The estimated percentage degree is 5% from 10%.

7. Download Time

The *David Chalmers* Website has suitable number of pictures, no video or audio. The download time needed to appear the Website on Web browser take a traditional time; it has a good download time. The estimated percentage degree is 8% from 10%.

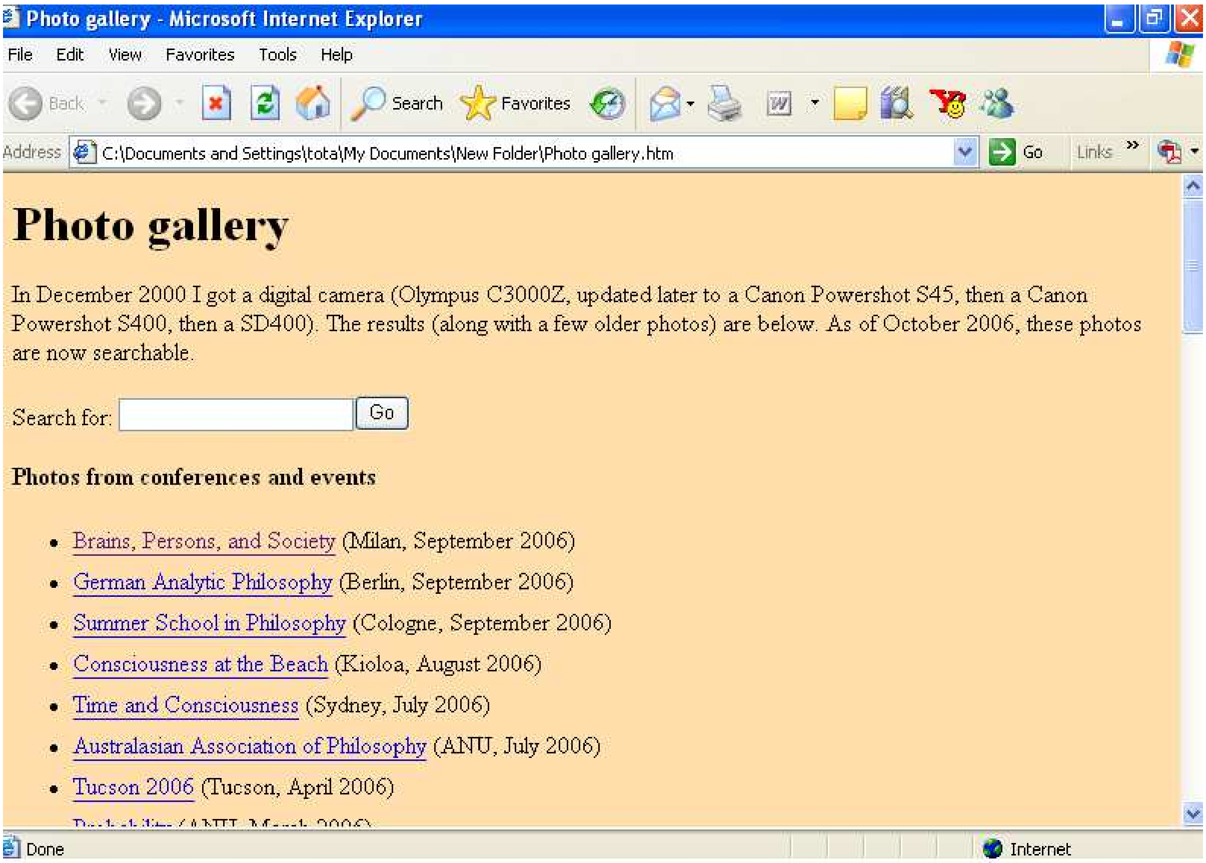


Photo gallery - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\Documents and Settings\tota\My Documents\New Folder\Photo gallery.htm
Go Links
Done
Internet

Photo gallery

In December 2000 I got a digital camera (Olympus C3000Z, updated later to a Canon Powershot S45, then a Canon Powershot S400, then a SD400). The results (along with a few older photos) are below. As of October 2006, these photos are now searchable.

Search for: [ ] Go

Photos from conferences and events

- Brains, Persons, and Society (Milan, September 2006)
- German Analytic Philosophy (Berlin, September 2006)
- Summer School in Philosophy (Cologne, September 2006)
- Consciousness at the Beach (Kioloa, August 2006)
- Time and Consciousness (Sydney, July 2006)
- Australasian Association of Philosophy (ANU, July 2006)
- Tucson 2006 (Tucson, April 2006)
- Probability (ANU, March 2006)

Figure (4.25) Search Engine

Table (4.4) shows the estimation percentage degree for the selected Websites estimated by the researcher. It shows every Web engineering characteristic percentage degree, this degree is suggested from the researcher point of view, and it came from the truth of applying the chosen Web engineering characteristics.

Table (4.5) shows the statistical degrees of the selected Websites as well as the proposed dynamic Website. These degrees are come after selecting ten programmers randomly to estimate them, and then displaying the average degree for each characteristic and the total degree for each Website.

Table (4.4) Comparison Result Table

Web Engineering Characteristics	Estimated Percentage Degree	How do they do that with HTML	Cornell University	David Chalmers	Proposed Dynamic Website
Content Design	25%	20%	23%	20%	22%
Navigation Design	20%	10%	10%	10%	20%
Interface Design	20%	15%	18%	12%	18%
Unique Template	10%	4%	10%	2%	10%
Goal of Website	5%	5%	5%	5%	5%
Search Engine	1-10%	2%	8%	6%	8%
Download Time	1-10%	8%	8%	8%	8%
Total		64%	82%	63%	91%

Table (4.5) Comparison Result Table

Web Engineering Characteristics	Estimated Percentage Degree	How do they do that with HTML *Average*	Cornell University *Average*	David Chalmers *Average*	Proposed Dynamic Website *Average*
Content Design	25%	11.7%	18.8%	15.9%	18.7%
Navigation Design	20%	13.7%	17.4%	14.1%	17.7%
Interface Design	20%	8.5%	17.2%	11.7%	16.8%
Unique Template	10%	5.2%	8.2%	5.5%	8.7%
Goal of Website	5%	3.6%	4.4%	4%	4.5%
Search Engine	1-10%	7.4%	7.9%	7.1%	7.6
Download Time	1-10%	8.9%	8.8%	9.1%	9.2%
Total		59%	82.7%	67.4	83.2

4.5 System Requirement

The proposed dynamic Website specification could be classified into two classes:

4.5.1 Hardware Requirement

The Proposed Dynamic Website as every Web application requires being online on the Internet or at least be implemented on a network that works on TCP/IP protocol. The Website works by Client/Server Technique.

4.5.2 Software Requirement

There are two sides in the proposed dynamic Website:

1. Client-Side: this side must be installed Internet Explorer to be online on the Internet.
2. Server-Side: a large part of the proposed dynamic Website is implemented on the server-side. The server requires:
    - Active server pages (ASP).
    - Hyper Text Mark up Language (HTML).
    - Visual Basic Script.
    - Java Script.
    - Microsoft SQL Server 7.0.
    - ODBC Connections.
    - Windows Server 2000 NT.

Chapter One
Introduction

1.1 Introduction

Within a short period, the Internet and World Wide Web (WWW) have become ubiquitous, surpassing all other technological developments. They've also grown rapidly in their scope and extent of use, significantly affecting all aspects of our lives.

Figure (1.1) [Rob05] presents the growth of the Websites according to a study presented by *Robert H. Zakon*. Robert makes study regarding growth of Websites collected over last thirteen years.

Websites =Number of Websites on the Internet.

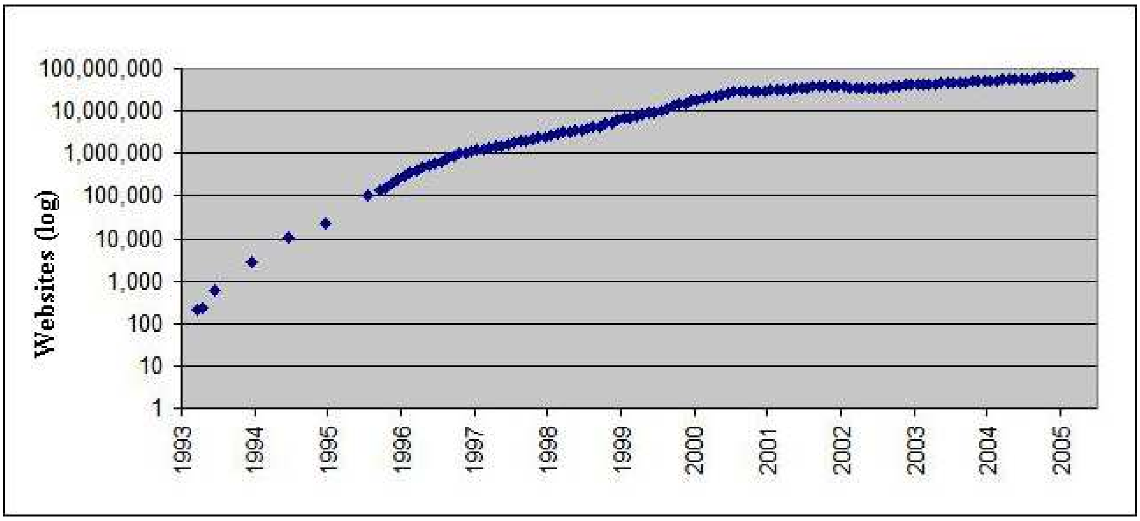


Figure (1.1) Growth of the Websites [Rob05]

Industries such as manufacturing, travel and hospitality, banking, education, and government are Web enabled to improve and enhance their operations. E-commerce has expanded quickly, and database systems have migrated to the Web. Advances in wireless technologies and Web-enabled appliances are triggering a new wave of mobile Web applications. As a result, many of peoples rely on Web-based systems and applications.

To build these systems and applications, Web developers need a sound methodology, a disciplined and repeatable process, better development tools, and a set of good guidelines. The emerging field of Web engineering fulfills these needs. It uses scientific, engineering, and management principles and systematic approaches to successfully develop, deploy, and maintain high quality Web systems and applications. It aims to bring the current chaos in Web-based system development under control, minimize risks, and enhance Website maintainability and quality [Ath01].

The essence of Web engineering is to successfully manage the diversity and complexity of Web application development, and hence, avoid potential failures that could have serious implications [San05].

This thesis apply the fundamentals objectivity for design and execution sides depend upon the software engineering's fundamental concepts and principles for dynamic Web application, and introduce Web engineering as a way of managing complexity and diversity of large-scale Web development.

1.2 Software Engineering Principles and the Web

Software engineering is an engineering discipline which is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use. Software engineering adopt a systematic and organized approach to their work as this is often the most effective way to produce high-quality software. However, engineering is all about selecting the most appropriate method for a set of circumstances and more creative, informal approach to development may be effective in some circumstances [Ina01].

Websites are become more like software. Like building software, a single Website may have to fulfill many roles ranging from document delivery to business process automation. Sites are often building to suite the need of diverse group including potential customers or department in an organization. The many requirements for sites, the number of concerned parties and the pace at which sites must be developed make Websites development at a challenging endeavor.

Software Engineering methods can be applied to Website development to help bring the chaotic process under control and minimize the risk of a failed project. That will help minimize problems and provide a framework to manage the project [Tho98].

1.3 Website

A Website a set of interconnected Web pages, which are individual documents, coded in Hyper Text Markup Language (HTML) and located on the WWW. Web pages are distinct from other pages by their Uniform Resource Locater (URL) [Dan99].

Each Website contains a *Home page***,** which is the first document or Web page that users see when they enter the site [Eih05**]**.The site might also contain additional Web pages, which are sometimes called *Child Pages* [Vir01].

A Website is hosted on a server by its owner or at an Internet Services Provider (ISP). It may share space on the server with other Websites, reside on a server dedicated to that Website only or be on multiple dedicated servers. To qualify as a Website, the Web server must be available on the Internet 24 hours a day [Wik06], Figure (2.1) shows a simple Website.

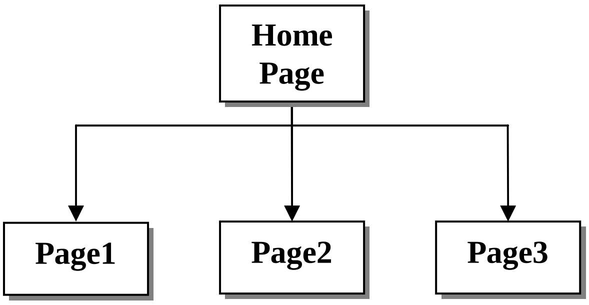


Figure (1.2) Simple Website

There are two kinds of Websites: Ones that store all Web pages in a single document (static) and ones that use a database to store content (dynamic) [Dan99].

1.3.1 Static Website [Int02]

Static Website is the Website whose content doesn't change and stays same when every page is viewed by user. The user cannot interact with the content or change the content and therefore the content stays the same. The reason it is static is because all the content in the page comes from the HTML code and not an external file such as Structured Query Language (SQL) database or java script file.

1.3.2 Dynamic Website

Dynamic Websites use a database to store information; it can change after every page view by a user. It is reacting to the user choices and information entered by a user, or information in a database. Foremost specializes in building dynamic Websites for customers to suit specific needs. Typically the Website will serve as a front-end to a database such as Oracle and SQL. There are plenty of different types of Web based computer languages for creating Dynamic HTML such as Java Server Pages (JSP), Common Gateway Interface (CGI), Personal Home Page (PHP), Active Server Page (ASP), and Cold Fusion [Int02].

Figure (2.1) illustrates typical configuration of a dynamic content Website [Cri01].

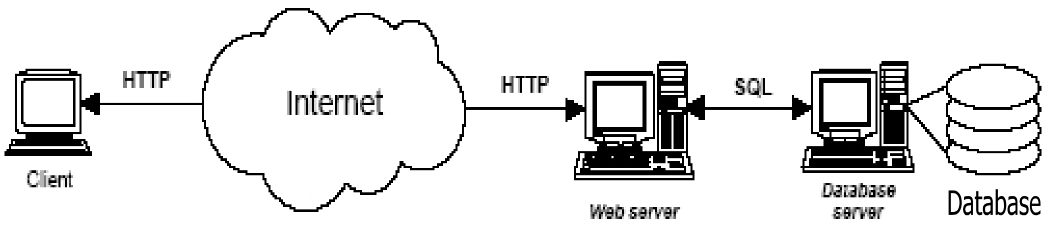


Figure (2.1) Typical Configuration of a Dynamic Content Website

1.3.3 Dynamic Website Benefits [Jon05]

Benefits of using dynamic pages are illustrated below:

- **Databases searchable**: users can search database for keywords/phrases.
- **User controlled content management**: control the text and images on the Website. This saves the time and money in the long run. For example, Web engineer can design a user-friendly content management system that lets any members build their own Web pages. This means that members can edit their content on the Web pages that they create as often as they like.
- **Menu systems update as pages are added and removed:** A static Website's menu system is static so if add a new page must update the entire Website. With dynamic Websites things like member home pages and member product sub-pages are dynamic.

1.4 Website as Software [Tho98]

Websites considered as software. Software is a computer program or a set of computer programs plus associated materials like documentation, which is used to perform some task. Simple Websites may be nothing more than a collection of documents that is retrieved from a remote file server. Once functionally collecting user information with a form, providing database query capabilities, generating information with a form, providing database query capabilities, generating pages dynamically, or even providing an online game is added to the Website, the Website is a collection of pages. It can actually be thought of as a program.

1.5 Web Engineering and Web Applications [Rog06]

Web-based systems and applications (WebApps) deliver a complex array of content and functionality to a broad population of end-users. Web engineering is the process that is used to create high-quality WebApps. Web engineering (WebE) is not a perfect clone of software engineering, but it borrows many of software engineering's fundamental concepts and principles, emphasizing the same technical and management activities. There are subtle differences in the way these activities are conducted, but an overriding philosophy that dictates a disciplined approach to the development of a computer-based system is identical**.**

1.6 Literature Survey

1. Andrew McDonald and Ray Welland (2001) [And01]

The paper describes the background, results and the conclusions that can be drawn about the practice of Web engineering. Also discuss the major characteristics that describe Web-based application development, and the issues that a successful Web engineering process will have to address, depend upon conducted interviews with a number of people within organizations in the United Kingdom who are involved in the development of Web-based applications.

The goals of the interviews is to try to identify more clearly the major issues facing the development of Web-based systems and to see which, if any, traditional software engineering practices and techniques were being successfully applied these. According to the results, the average development life cycle time is less than three months. And Unlike traditional software engineering, Web engineering must cope not only with the development of software components but also with the development of data, and the inter-dependencies between them, with Small development teams working in parallel on similar tasks.

2. David Wolber, Yingfeng Su (2001) [Dav01]

The paper presented a Web Developer (WebDev), it is a programming in the “What You See Is What You Get” (WYSIWYG)

“HTML editor” interface tool for building dynamic Web pages that connect to databases. The system allows designers to "program" by entering query by example (QBE) and spreadsheet formulas into visual components of HTML documents. Then, the system then automatically generates dynamic Web pages that can be executed in a browser. These tools do not, however, help a designer create a complete dynamic Web page, including both presentation and content. The designer is still dependent on a programmer.

3. Filippo Ricca (2002) [Fil01]

This thesis improved the quality factors of Web application such as correctness, reliability, maintainability and usability, by applying analysis and testing techniques. The goal of analysis and testing is to assess and to improve the quality of Web applications generated during development and evolved during the modification phases. The steps toward analysis and testing are the definition of a set of models representing the various entities involved in Web applications and their mutual relationships, entities such as forms, dynamic links and dynamic pages, with the aim of extending known analyses and testing techniques to dynamic Web applications.

4. Thomas Zwanzinger (2004) [Tho04]

A discussion issues of testing Web application and Web database performance that have to be considered while performing load and performance tests on Web application systems. This paper focus on performance evaluation through the eyes of the user, so several things can be measured when evaluating a Web applications performance: resource usage, throughput, response time, and queue lengths, describing the average or maximum number of tasks waiting to be served. Results of performance testing can have serious implications on the design of a Web application and can be used to estimate future workload.

5. Daniel R. Licata (2004) [Dan04]

A description of a model checker designed to identify errors in Web software (Interactive Web), not only Websites are generated by programs, but they are increasingly playing the role of “services”, accepting inputs from users, combining these with information in databases, and dynamically computing results.

The paper present a technique for automatically generating novel models of Web programs from their source code; these models include the additional control flow enabled by these user operations. In this technique, exploit a constraint-based approach to avoid over approximating this control flow; this approach allows to evade exploding the size of the model. The paper discusses the implementation of this model checker and a study of its effectiveness.

6. Vlasios Voudouris (2004) [Val04]

In this thesis a prototype was developed of a database-driven Website specifically designed for the property business. This Website assists the users to find properties that match their criteria. Once they have found a property, an interactive map representing regional facilities was available so that the users can have additional geographic information supporting their decision making process. The study was restricted in scope to the development of a prototype database-driven Website. Three database-driven Websites specifically designed.

1.7 Aim of the Thesis

The aim of the thesis is to put subjective bases for designing and implementation aspects based on standard measures for software engineering of dynamic Websites.

1.8 Thesis Layout

In this section, the contents of individual chapters of the thesis are briefly reviewed:

Chapter Two (Web Engineering)

At first, an introduction and explanation about Web engineering, and define some important basic terms that will be used in Web engineering. Then illustrate the Web Engineering specifications and systematic approaches to how construct successfully and high quality dynamic Web Application.

Chapter Three (Web Applications)

An explanation about Web applications, and how these Web applications use Hyper Text Transfer Protocol (HTTP) to implement communication between browsers and servers, and then illustrate categorization of Web application; at the last part explain the types of Web Application and the complexity of these Websites and illiterated quality requirement for Web applications.

Chapter Four: (Proposed Dynamic Website design and Implementation)

The Proposal Dynamic Website is presented step by step, designing, programming and implementation; the Dynamic Website will be presented and explained explicitly.

Chapter Five (Conclusions and Suggestion for Future Work)

Conclusions that are extracted from this thesis and important remarkable future work point will be illustrated.

Chapter Two
Web Engineering

2.1 Introduction

The Web, or the World Wide Web architecture, has become, undisputedly, the premier delivery platform for the majority of applications today. Applications range from small, simple static Websites to large applications, involving distributed, heterogeneous databases, scattered throughout a wide geographical area.

The constant technological evolution, coupled with the increasing complexity of applications, has stressed the already perceived need for adequate methods and techniques for the development, maintenance and evolution of such applications. The area of Web engineering has focused on such methods and techniques, leveraging existing software engineering practices, enriched with new, Web-specific approaches [Mar01].

Web engineering uses scientific, engineering and management principles and systematic approaches to successfully develop, deploy and maintain high quality Web based systems and application. Web developers, clients, government agencies, users, academics, and researchers have increasingly become interested in Web engineering. In addition, this new field has attracted professionals from other related disciplines such as software engineering, distributed systems, computer science, and information retrieval.

Web engineering is a holistic approach, and it deals with all aspects of Web based systems development, starting from concepts of Web-based systems development to implementation, performance evaluation and continual maintenance. Building and deploying a Web-based system involves multiple, iterative steps.

Most Web-based systems continuously evolve to keep the information current and to meet user needs. Web engineering represents a proactive approach to creating Web application. Web engineering methodologies have been successfully applied in a number of Web applications [Ath01].

2.2 Web Engineering Versus Software Engineering [Ath01]

Web engineering isn't a clone of software engineering although both involve programming and software development. While Web engineering adopts and encompasses many software engineering principles, it incorporates many new approaches, methodologies, tools, techniques, and guidelines to meet the unique requirements of Web-based systems.

Developing Web-based systems is significantly different from traditional software development and poses many additional challenges. There are subtle differences in the nature and life cycle of Web-based and software systems and the way in which they are developed and maintained. Web development is a mixture between print publishing and software development, between marketing and computing, between internal communications and external relations, and between art and technology.

2.3 Terminology

Define some basic terms that will be used in Web Engineering:

2.3.1 Web Design

A multidisciplinary pursuit pertaining to the planning and production of Websites, including, but not limited to, technical development, information structure, visual design, and networked delivery [Tho02].

2.3.2 Content

Content is the information that is offered to the user by Web pages, includes the form and organization of a Website's content. This can range from the way text is written to how it is organized, presented, and structured using a markup technology such as HTML [Tho00].

- **Static Content:** Content that does not change at run-time. It is mainly stored in files on servers or in databases and is presented to the user without any processing (e.g. a home page defined in HTML) [Eng02].
- **Dynamic Content:** Content that generated at run-time based on the interaction with the user (e.g. an e-commerce application that presents a welcome text and lists the current items in a user' shopping cart) [Eng02].

2.3.3 Web Server

Is a program that makes Web pages available to people who are browsing the Web [Bry96].Web server retrieve Web documents in response to browser requests and forward the documents to the requesting browsers via the Internet. Web servers also provide gateways that enable browsers to access Web-related applications such as database searchers [Syb00].

2.3.4 Web browser

Web browser is a program that displays HTML pages. The Web browser requests a pages from a server based on its Internet address. It retrieves the document from the server and displays the content to the user [Mic00]. Is a window onto the WWW. With Web browsers, user can view Web documents containing integrated or linked graphics, or even video and audio clips [Ric96].

2.3.5 Application server

Referred to as a type of middleware, application servers occupy a large chunk of computing territory between database servers and the end user, and they often connect the two[Jup06].

2.3.6 Middleware

Middleware is Software that connects two otherwise separate applications. For example, there are a number of middleware products that link a database system to a Web server. This allows users to request data from the database using forms displayed on a Web browser, and it enables the Web server to return dynamic Web pages based on the user's requests and profile [Jup06].

2.3.7 Stakeholders

There are different stakeholders in Website engineering project: Content managers, graphic designer, Web engineers, visitors (users) [Eng02].

2.3.8 Web engineer

A Web engineer becomes involved in a wide range of activities during the development of a WebApp including requirement elicitation, analysis modeling, architectural, navigational and interface design, WebApp implementation, and testing [Rog05].

2.4 Planning in Web Engineering

Getting started is always difficult. The first Web engineering framework activities emphasize *Planning*. Web Engineers, their managers and non-technical stakeholders all participate in formulation and planning, it provides a map for Web engineering tem. Planning a Website is addresses the things that must be defined to establish a work flow and a schedule, and to track work as the project proceeds [Rog06].

Because anyone can publish on the Web, the Web browsers users see all kinds of poorly designed, hard-to-read pages. So before begin creating the Website, need to do some planning [Lin01]. Planning a Website is two part process:

1. Defining the goals of Website.
2. Define the Structure.

2.4.1 Defining the Goals of Website

Two or three goals should be the foundation of any Website design. The statement should include specific strategies around which the Website will be designed, What is the purpose of this Website, how long the Website design, construction, and evaluation periods will be, and specific quantitative and qualitative measures of how the success of the Website will be evaluated [Lin01].

Building a Website is an ongoing process, not a one-time project with static content. Long-term editorial management and technical maintenance must be covered in the budget and production plans for the Website [Pat05].

2.4.2 Define the Structure

After defining the goals of the Web presentations, should define the structure on paper; that is draw all the pages, define all links, and make all decisions before start to code [Lin01].

A *Site map* is often a useful idea. This can be as a simple as listing all the pages in outline form. It can be as complex as an interactive page that shows all the links from one page to the next [Jef00].

Picking of the correct structure for information is not an easy task. The idea of picking the correct structure for document collection or breaking up information into a collection of pages is called *information mapping*. The first step of the information mapping process is to understand the data that will be included on the Website.

This may include the type of the data such as text, graphic, video, sound and so on as well as the amount of information and its eventual purpose. Often it is very useful to understand if information is to be consumed on screen or printed form. This can be very important because it may influence chunking the data dramatically. The overall purpose of the Website, including the amount of control that the user should have versus the designer, is also important in determining structure [Tho98].

2.5 Analysis Modeling for Web Applications

Analysis is to clearly understand the problems to be addressed by the proposed Web application [And01].

Requirement analysis for WebApp encompasses three major tasks: information, requirements gathering and analysis modeling. During planning, the basic motivation (goals) and objectives for the WebApp are identified, and the categories of users are defined. As requirements gathering begins, communication between the Web engineering team and WebApp stakeholders intensifies.

Context analysis is understood the system’s major objectives and requirements as well as the needs of system’s typical users and organization that needs the system. Analysis for Web application can minimize or eliminate the major problems playing large Web-based application and based on the context analysis, then arrive at the system’s technical and non-technical requirements which in turn influence the system architecture design [San05].

The objectives of context analysis of Web applications are [San05]

- Identify the stakeholders and their broader requirements and experiences.
- Identify the functions the Website needs to provide (immediately, and in the short, medium, and long term).
- Establish what information needs to be on the Website, how to get this information, and how often this information may changed.
- Identify the corporate requirements in relation to look and feel, performance, security, and governance.
- Get a feel of the number of users and anticipated demands on the system.
- Study similar (competitive) Websites to gain an understanding of their functionalities, strengths, and limitations.

2.6 Design Modeling for Web Applications

Design in the context of Web engineering leads to a model that contains the appropriate mix of aesthetics, content, and technology. The mix will vary depending upon the nature of WebApp, and as a consequence the design activities that are emphasized will also vary [Rog05].

When design any Website, create design elements that can be maintained throughout the Website. This includes a color scheme, font style, corporate logos, or even navigation buttons. Anyone should be able to enter the Website at any pages and instantly know what the Website is for [Jef00].

Figure (2.1) depicts a design pyramid for Web engineering. Each level of the pyramid represents one of the following design activities [Rog05]:

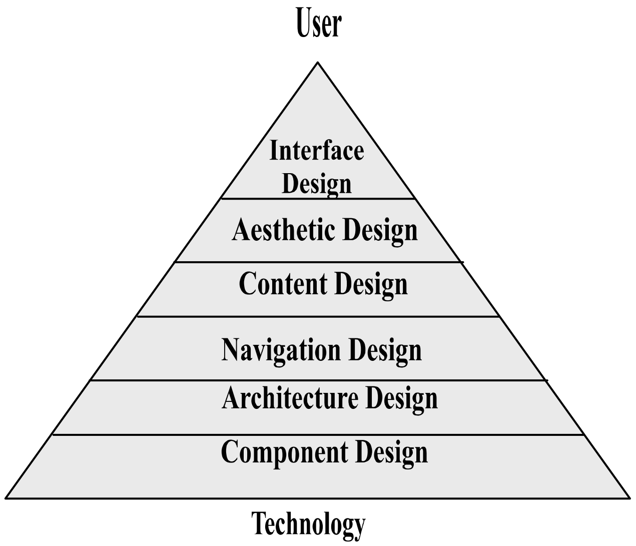


Figure (2.1) Web Engineering Design Pyramid

2.6.1 Interface Design

Describe the structure and organization of the user interface. Include a representation of screen layout a definition of the modes of interaction, and a description of navigation mechanisms [Rog05].

Interface requirement to a user also known as interaction requirements or user's requirements. They give an answer to how the user is going to interact with the Web application [Jos04].

Interface development of a screen layout to serve as an effective communication medium prototype between the user and the system [Ist00].

2.6.2 Aesthetic Design

Aesthetic Design, also called graphic design, describes the "look and feel" of the WebApp, includes color schemes, geometrics layout, text size, font and placement, the use of graphics, and related aesthetic decision [Rog05].

2.6.3 Content Design

Define the layout, structure, and outline for all content that is presented as a part of the WebApp. It establishes the relationships between content objects [Rog05]. Design of the Web application content, which includes text, graphics, images, video and audio data [Ist00].

The amount of information shown on a page should not to be too large for many reasons [Rog99]:

- If too much information is on a single page, the users might simply be confused or overwhelmed by the interface.
- The amount of time needed to download the page from the Web server might be too long.
- The users might have to scroll up and down the page to access all necessary information, which might disorient them. In the worst case scenario, scrolling up and down a page might be unacceptable for Web database application functions that contain repetitive computing tasks such as data entry transactions.

2.6.4 Navigation Design

Navigation design represents the navigational flow between content objects and for all WebApp functions; the designer must define navigation paths way that enable users to access WebApp content and functions.

Each user interacts with Web application encounters a series of *Navigation Semantic Uunits* (NSU) define as a set of information and elated navigation structures that collaborate in the fulfillment of a subset of related user requirements [Rog05].

Too many people enter a Website and then get lost after three clicks for example, when this happens; the user searches another Website. Avoid losing the user by limiting their frustration. Create simple navigation that gives control over Website to the user. Keep it simple. Keep it fast, navigation bars should be labeled clearly to show the visitors where they are and what else is available [Jef00].

Figure (2.2) shows an example of a generic Web application structure that contains an index page that points to all the pages of Web database application [Rog99].

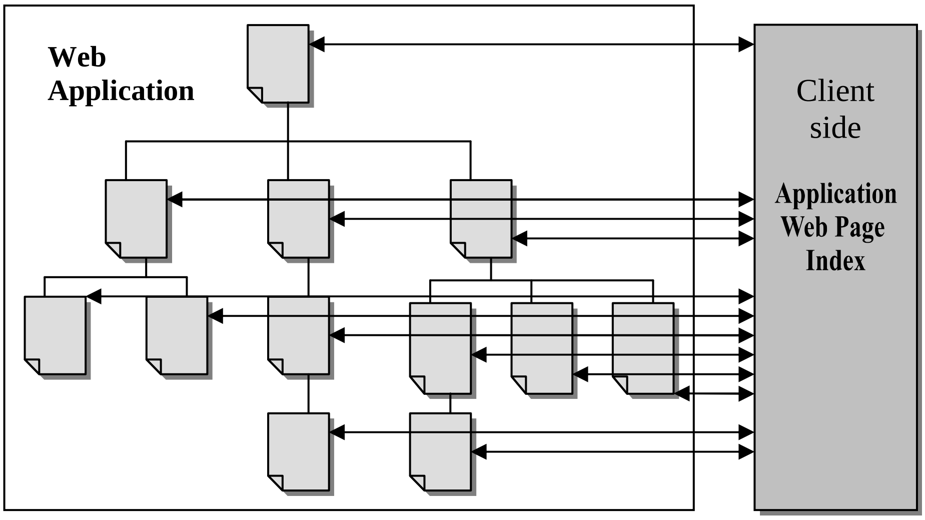


Figure (2.2) Web Application Navigational Design [Rog99]

2.6.5 Architecture Design

Architecture design identifies the overall hypermedia structure for the WebApp [Rog05]. There are four main hypertext organizational schemas in use on the Web:

- Linear forms
- Grids
- Hierarchy
- Pure Web

Slight variations on some of the basic schemas are also common. Choosing the correct Website form is important in making a Website accessible [Tho98].

1. Linear Structure

This structure is the easiest of the structure to design and navigate through because it most closely resembles a conventional paper document. One positive of this Structure in this manner are presented with a familiar and comfortable model. Links between pages are very well defined and navigation is linear. Linear structure is shown in figure (2.3) [Beb96].

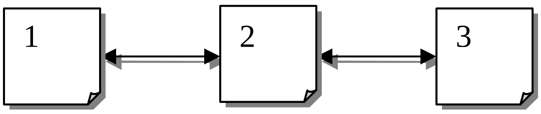


Figure (2.3) Linear Structure [Beb96, Tho98, and Rog05]

2. Grid Structure

A grid is a dual linear structure that presents both horizontal and vertical relationship between items. Because a grid has a spatial organization, it is good for collections of related items; so far a pure grid structure remains uncommon on the Web. When designed properly, a grid provides horizontal and vertical orientation so the user may never feel lost. While a grid structure is highly regular and may be easy for user to navigate, not many types of information are uniform enough to lend themselves well to this organization. Grid structure is shown in figure (2.4).

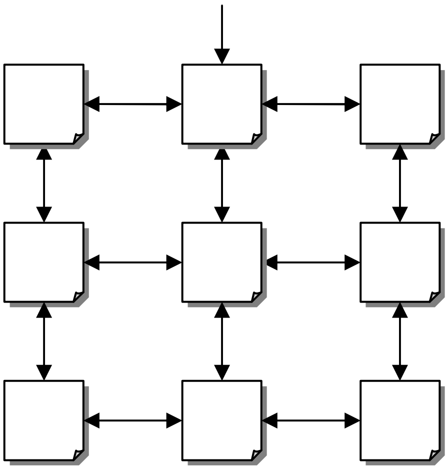


Figure (2.4) Grid Structure [Beb96, Tho98, and Rog05]

3. Hierarchy Structure

The most common hypertext structure on the Web is the tree or hierarchy form. The hierarchy is very important because it can be modified to hide or expose as much information is a necessary. Designer might be tempted to expose all information at once, but remember the hypertext rule that states that users don't need to see all information at once. Hierarchical structure is shown in figure (2.5) [Tho98].

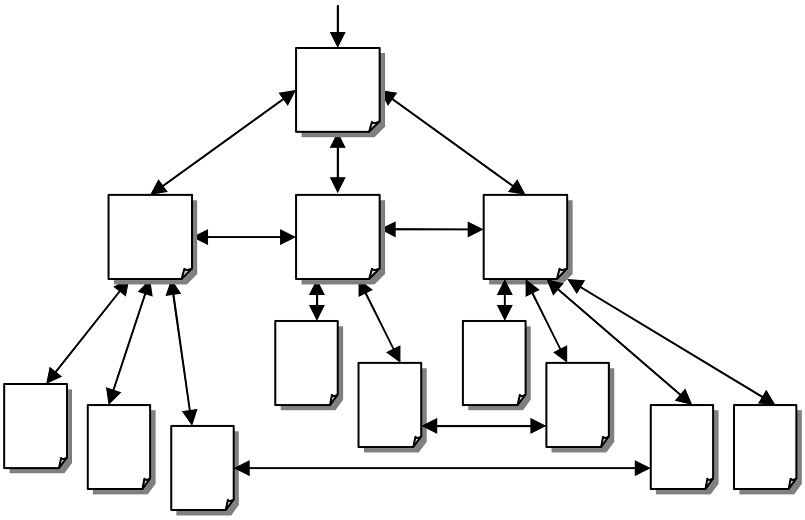


Figure (2.5) Hierarchical Structure [Beb96, Tho98, and Rog05]

4. Pure Web Structure

The Web structure is recognizably the most expressively powerful of the structure as well as the most potentially dangerous to a user. Pages and their links can be organized in any topological pattern which best defines their relationships and the navigational paths within the document. The document can be viewed as multidimensional (hyper-spatial).

Form a user's perspective; the relationships within the document may appear hopelessly confusing. For example, tools for navigation within such a document become critical in order to prevent a user form becoming hopelessly lost or confused.

For the design of complex Web structures, the Relationship Management Methodology Approach may be applicable. In this method, entity-relationship tools are used to define the complex relationships encountered in a hypertext document structured as a Web. Web structure is shown in figure (2.6) [Beb96].

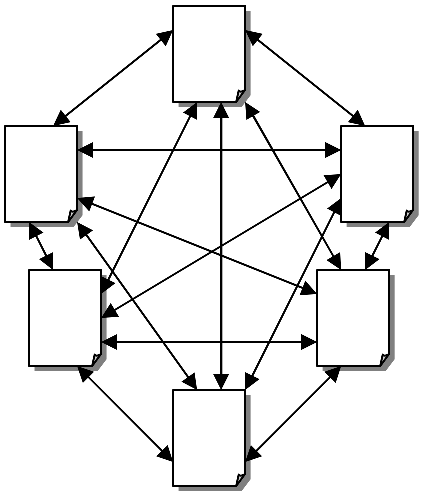


Figure (2.6) Web Structure [Beb96, Tho98, and Rog05]

Figure (2.7) shows the relationship between the expressiveness and predictability of the different organizational structures.

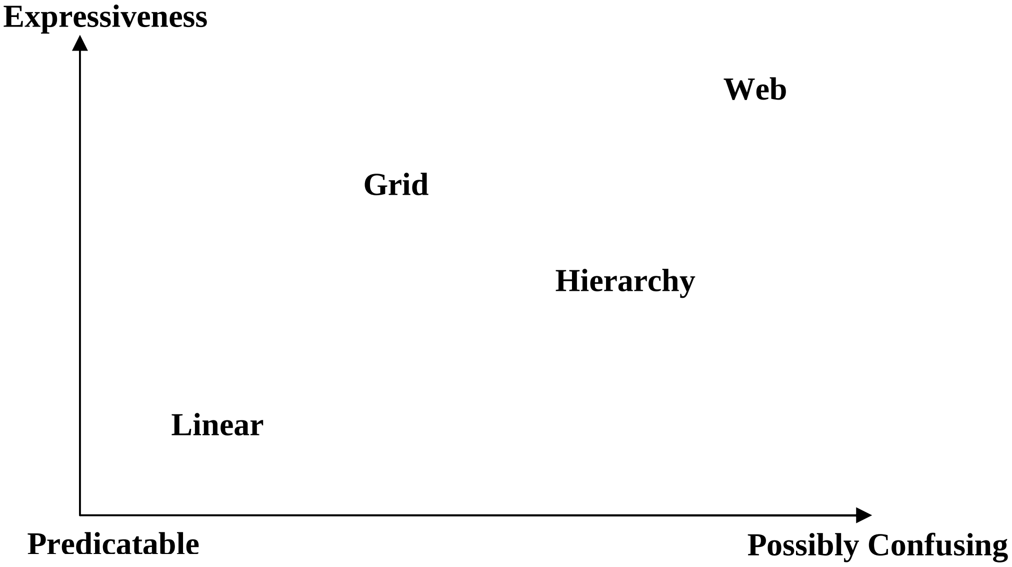


Figure (2.7) Expressive Versus Predictability [Beb96, Tho98]

The linear is very predictable; it provides a limited relation view. While Web is very expressive, it can be confusing. The grid and hierarchy share the middle ground. Most Websites now use a mixed hierarchy approach that is familiar to many Web users. Depend on the goals of the Website; however, the use of several types of structures might be combined.

For example, while the overall structure of a Website might be hierarchy, a pure linear structure could be used to provide an introduction to a company, and a narrow hierarchy could be used in the technical support section [Tho98]. The structure should be planned accordingly and documented in a flowchart as shown in figure (2.8).

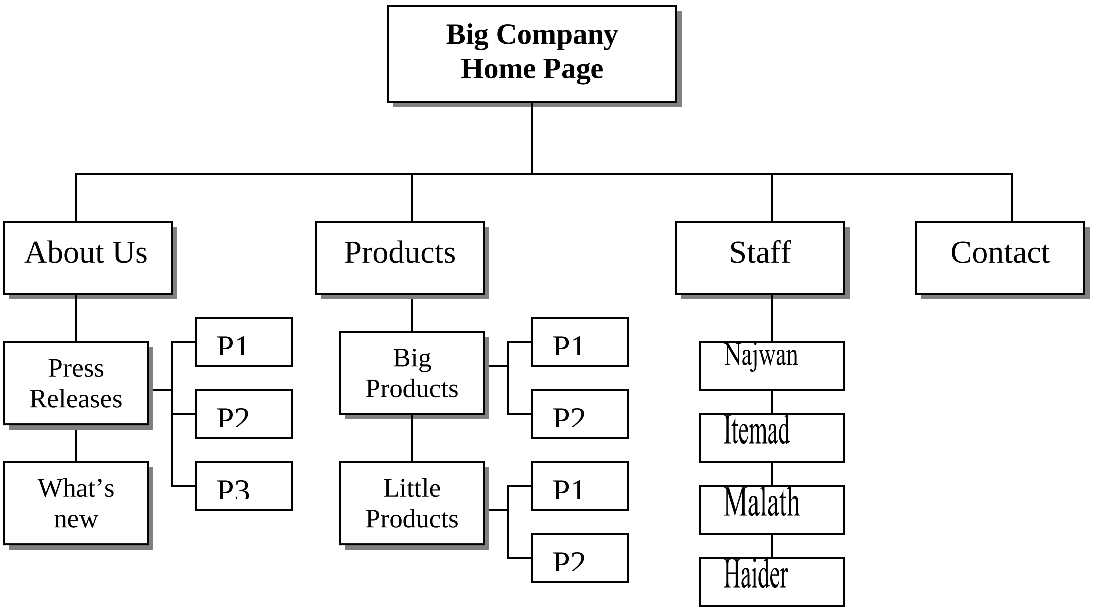


Figure (2.8) Sample Website Flowchart

2.6.6 Component Design [Rog05]

Develops the detailed processing logic required to implement functional components. Modern Web applications deliver increasingly sophisticated processing functions that:

1. Perform localized processing to generate content and navigation capability in dynamic fashion.
2. Provide computation or data processing capability that is appropriate for the WebApp's business domain.
3. Provide sophisticated database query and access.
4. Establish data interfaces with external corporate system.

To achieve these and many other capabilities, the Web engineer must design and construct program components that are identical in form to software components for conventional software.

2.6.7 Web Page Design Templates [Rog99]

One of the best ways to ensure a consistent application look and feel is to develop a set of standard Web page templates that can be constantly reused across the enterprise, for all Web applications. If necessary, the templates can be modified and adapted to satisfy the peculiar needs of each particular application. Atypical template for a Web page application appears in figure (2.9)

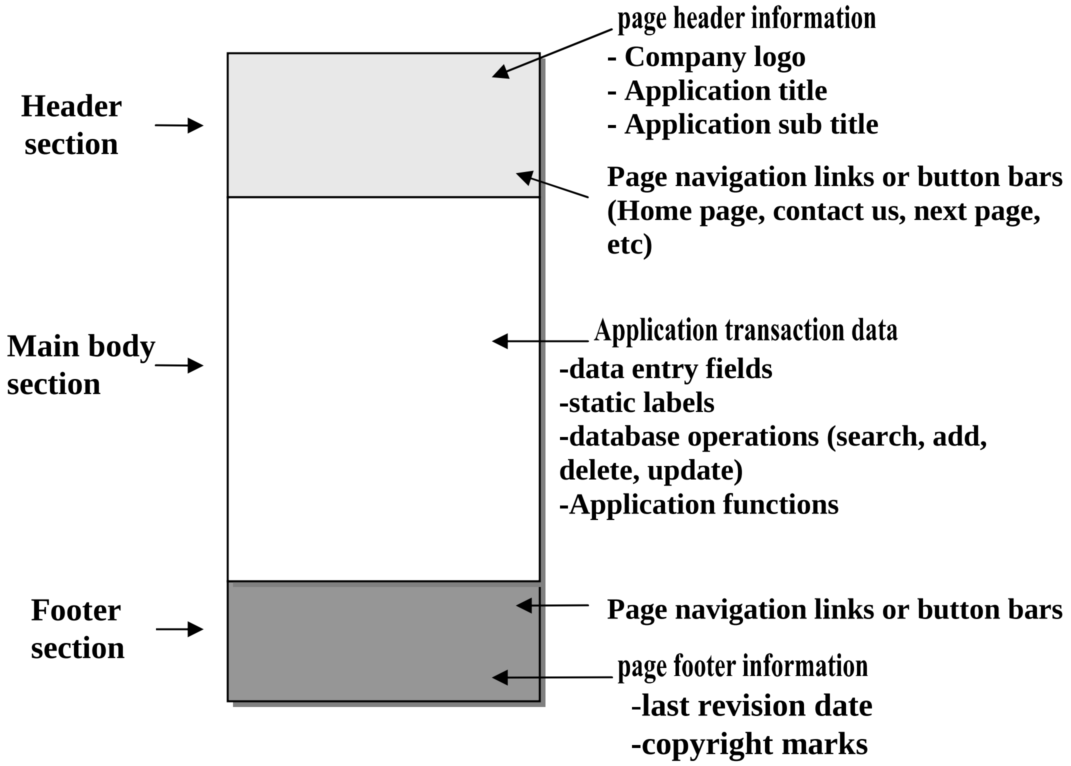


Figure (2.9) Standard Web page Grid

This template diagram illustrates one possible approach for the placement of static text, navigational page controls, functions and data fields. The standardization of the layout of the Web pages, either in a single application or across different applications, greatly helps the users to navigate from one page to another and also quickly recognize the repetitive information and navigation cues that are consistently laid out on each page.

This technique provides the users with a secure sense of consistency and structure while interfacing with the graphical user interface. For instance, the page header and footer sections should contain information that established clearly the identity of the Web application. It also helps the users to quickly situate themselves visually in the Web application structure. The users should never get confused as to where they are located in the Web application.

For example, the header section of each application Web page should always contain a title describes the application and subtitle that describes the business transaction that can be performed with this Web page. If small, recurring graphic theme such as a company logo is used to identify the Web application or the enterprise, and then place it consistently at the same location across all the pages, preferably in the header section.

2.8 Implementation in Web Engineering

The implementation of software system is more than just writing and compiling the code although that is a large part of the workflow. Implementation takes the artifacts of design and applies software development tools to them. The tools are typically editors and compilers [Jim00].

The information and functionality planned and organized in the design phase is coded in an appropriate format. Most Websites use HTML files to deliver static content. These HTML documents can be written using editors or generated from relational databases using widely available Web tools. The functionality and support for interactions is usually implemented using popular Web technologies.

Most of these technologies generate dynamic content by either writing HTML to a stream that is sent back to the calling client or by mixing layout information with application logic in files that are interpreted at run-time by an application server. Examples of the first form of interaction are the Practical Extraction and Report Language (Perl) script, Java servlet and C# technologies and an example of the second form are the PHP, and Coldfusion technologies [Eng02].

2.8 Testing Web Applications

In most Software Engineering Process model, Software testing has an important role in guaranteeing the quality of software product [Tho04].

WebApp testing is a collection of related activities with a single goal, to uncover error in WebApp content, function, usability, navigability, performance, capacity, and security, to accomplish this, a testing strategy that encompasses both reviews and executable testing is applied throughout the Web engineering process. Web engineers and other project stakeholders (managers, customers, end-users) all participate in WebApp testing.

If end-users encounter errors that shake their faith in the WebApp, they will go elsewhere for the content and function they need, and the WebApp will fail. For this reason, Web engineers must work to eliminate as many errors as possible before the WebApp goes on-line [Rog05].

The WebApp testing process begins by focusing on user-visible aspect of the WebApp and proceeds to test that exercise technology and infrastructure. Seven testing steps are performed:

- Content Testing.
- Interface Testing.
- Navigation Testing.
- Component Testing.
- Configuration Testing.
- Performance Testing.
- Security Testing.

Figure (2.10) shows the WebApp testing process with the design pyramid discussed in section [2.6] *Design Modeling for Web Applications* see figure (2.1). The testing flow proceeds from left to right and top to bottom, followed by infrastructure design elements [Rog05].

2.8.1 Content Testing

Content testing insures that the content of the Website is correctly implemented; this includes visual reproduction issues, spelling, grammar, and other important details such as copyright information. Images must also be checked to make sure they are clear and colors appear to reproduce correctly. Other forms of media, like sound and video, would have similar legal checking requirements [Tho98].

Proper information design is a key to the development of a successful Website. If a Website has great content and a great interface but poor information architecture, it may be relatively useless. If the user cannot easily find the information the Website loses its effectiveness [San05].

2.8.2 Interface Testing [Rog05]

Verification and validation of a WebApp user interface occurs at three distinct points in the Web engineering process. During analysis and the interface model is reviewed to ensure that it conforms to customer requirements and to other elements of the analysis model. During design, the interface design model is reviewed to ensure that generic quality criteria established for all user interfaces have been achieved and that application specific interface design issues have been properly addressed.

During testing, the focus shifts to the execution of application specific aspects of user interaction as they are manifested by interface syntax and semantics. In addition, testing provides a final assessment of *usability*.

2.8.3 Navigation Testing [Rog05]

The navigation process can be unpredictable because the visitor, influenced by something he sees or learns, may choose a path or initiate an action that is not typical for the original objective.

The job of navigation testing is:

1. To ensure that the mechanisms that allow the WebApp user to travel through the WebApp are all functional.
2. To validate that each navigation semantic unit defined in section [2.6.4] can be achieved by the appropriate user category.

2.8.4 Component Testing [Rog05]

Component level testing also called function testing, focuses on a set of test that at tempt to uncover errors in WebApp functions. Each WebApp function is a software module (implemented in one of a variety of programming of scripting languages) must be tested. Each component specifies all input values and the expected output to be provided by the component.

In many situations, the correct execution of a WebApp function is tied to proper interfacing with a database that may be external to the WebApp. Therefore, database testing becomes an integral part of the component testing regime.

2.8.5 Configuration Testing [Tho98]

System aspects such as processor power, disk speed, and operating system issues may affect Website use dramatically. System performance is one major consideration during configuration testing. For example, a Website using client-side scripting for animation and timing may find that a particular minimum system memory and processor requirement is required to insure adequate playback. The design document should have considered what the base platform was and testing should be performed to make sure that Website works on the base platform.

2.8.6 Performance Testing [Tho04]

Performance Testing is a crucial part of a Testing Process in a Web Engineering. Several things can be measured when evaluating a Web applications performance: resource usage, throughput, response time and even queue lengths, describing the average or maximum number of tasks waiting to be served. Some resources that ought to be considered include network bandwidth requirements, CPU cycles, disk access operations and memory usage and database access rates. Results of performance testing can have serious implications on the design of a Web application.

2.8.7 Security Testing [Rog05]

To protect Website against external hackers, dishonest competitors, and any one else who wishes to steal sensitive information, modify content, degrade performance, disable functionality, one or more of the following security elements is implemented:

- Firewalls: a filtering mechanism that is a combination of hardware and software that examines each incoming packet of information to ensure that it coming from a legitimate source, blocking any data that are suspect.
- Authentication: a verification mechanism that validates the identity of all clients and servers, allowing communication to occur only when both sides are verified.
- Encryption: an encoding mechanism that protects sensitive data by modifying it in a way that makes it impossible to read by those with malicious intent. Encryption is strengthened by using digital

certificates that allow the client to verify the destination to which the data are transmitted.

- Authorization: a filtering mechanism that allows access to client or server environment only by those individuals with appropriate authorization code (e.g. user ID and password).

2.8.8 Database Testing [Rog05]

Modern Web applications do much more than present static content objects. In many application domains, WebApps interface with sophisticated database management system and build dynamic content objects that are created in real-time using the data acquired from a database.

Testing should ensure that:

1. Valid information is passed between client and server from interface layer.
2. The WebApp process script correctly and properly extracts or formats user data.
3. User data are passed correctly to server side data transformation function that formats appropriate queries (e.g. SQL)

2.8.9 The Result of Web Engineering Testing [Tho98]

The point of testing is to uncover problems with a Website. Invariably, bugs will be found and new features suggested. As testing proceeds, any bugs uncovered should be tracked and potentially dealt with. A bug tracking system is generally necessary for large projects so that bug fixes can be monitored.

The testing phase should yield an improved Website that is relatively free of defects and ready to be delivered. The testing process should produce results that can be used effectively in Website development.

2.9 Web Application Maintenance [San05]

After a Web-based system is developed and deployed online for use, it needs to be maintained. Content maintenance is a continual process. We need to formulate content maintenance policies and procedures, based on the decision taken at the system architecture design stage on how the information content would be maintained, and then we need to implement them.

Further, as the requirements of Web systems grow and evolve, the system needs to be updated and also may be redesigned to cater to the new requirements. It is important to periodically review Web-based systems and applications regarding the currency of information content, potential security risks, performance of the system, and usage patterns (by analyzing Web logs), and take suitable measures to fix the shortcomings and weaknesses.

2.10 Web Application Development [San05]

Web-applications are evolutionary. For many Web applications, it is not possible to specify fully what their requirements are or what these systems will contain at the start of their development and later, because their structure and functionality will change constantly over time.

Further, the information contained within and presented by a Website often changes in some applications as often as every day. Thus, the ability to maintain information and to scale the Website's structure (and the functions it provides) is a key consideration in developing a Web application.

Successful development of Web systems and applications involves multiple interactive steps which influence one another.

These are steps for successful development and deployment of Web applications [San05]:

1. Understand the system's overall function and operational environment.
2. Clearly identify the stakeholders that is, and system's main users.
3. Develop overall system architecture of the Web-based system that meets the technical and non-technical requirements.
4. Develop and implement the subprojects.
5. Refine and update the system.

2.11 The Signs of a Well Engineered Website [Tho98]

There are numerous aspects to the well engineered Website:

- Correct

A Website is correct if it performs properly and functionally and cosmetically error free. In a purely theoretical sense, correctness is hard to define; many Websites do not emphasize correctness when they contain problems like broken links, scripts that don't work, or incomplete navigation. Correctness is never absolute but can be enshroud with testing.

- Testable

Testing comes on many levels ranging from functionality testing to usability testing.

Functionality testing: is defining what the Website should do and then developing a battery of tests to prove that it dose what it was designed for.

Usability testing: Usability testing goes beyond functionality, (must Website be friendly use with any user).

- Maintainable

A well designed Website should be maintainable that is it should be easy to make changes to the Website. Website maintainability not only should provide the possibility of adding or removing section or functionality, but also should provide ways to do it with ease.

- Portable and Scalable

A Website should be *portable*; this means that it performs across platforms and is cross browser safe. Truly portable sites should be easy to move from server to server with a few change required. A truly portable Website implies that the Website is *scalable,* that it can be quickly extended to handle more users.

- Reusable

An important aspect of software is that the components used to build it are reusable. Websites can benefit from this principle of software engineering creating Websites components that are reusable can save time and money.

- Robust and Reliable

Websites should be robust and reliable. This requirement refers to the quality of the visual and technical implementation as well as to the delivery of the Website to the end user. For example , developer often focus greatly on the visual or programming nature of the Website but spend little time on the network issues, such as bandwidth availability and server responsiveness. However, if these issues are not well considered, the Website may not be robust. It also may be unreliable, failing at inopportune moments or not working consistently.

- Efficient

Efficiency is important principle of software engineering. The efficiency or performance of a Website (including the amount of data delivered and the quality of the programming elements) is determined not only by the quality of the implementation, but by the server and network issues over which developers may have limited control.

- Readable

For maintenance purpose, all files used to make a Website should be readable. Readability refers to the understandability of source file.

Readable files provide adequate comments, use judicious white space and logical formatting, and choose meaningful names so that future developers can understand the code.

- Well Documented

Websites should be well documented. This has meaning that any programming code or HTML markup should be commented so that future maintainers can modify the content. Well document Website should also provide documentation in the form of help files or other forms of information to help end users use the Website.

Chapter Three
Web Applications

3.1 Introduction

A Web application is a software application that depend on the Word Wide Web and that offers Web-specific resources like contents and services that can be used by Web browser. Once a new release of a Web application is installed on the server, this release is available to all users. This immediate deployment characteristic is probably one of the most powerful characteristics of a Web application [Sve04].

They use a single client-server model as illustrated in figure (3.1), and run in a Web browser on the client computer.

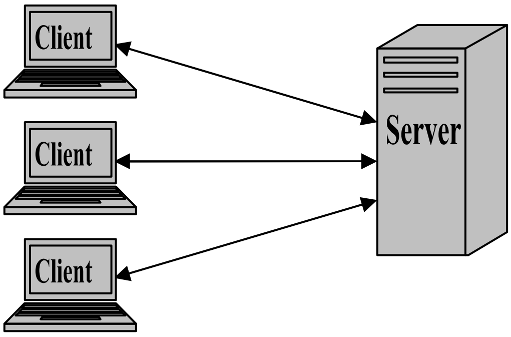


Figure (3.1) Client- Server Model [Eih02]

A Web application is an application delivered to users from a Web server over a network such as the Internet or an Intranet. Web applications are popular due to the ubiquity of the Web browser as a client. The ability to update and maintain Web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity. Web applications are used to implement Web mail, online retail sales, online auctions, discussion boards, and perform many other functions [Wik06].

There are different names in use for what is called a Web application. Names in use are Websites, Website, WWW site [Wik06], Web document [And01], Web service [Eng02], Web-based applications [Rog05].

3.2 Web Application Architecture

There are three significant architectural components to a Web application: the client browser, the Web server, and the application server. It is also most likely that the Web application will also use a database server [Jim00].

Generally Web servers provide two different types of content: static and dynamic content. Static content are Websites or files that can be served directly. Only data transfer occurs on server side. Dynamic pages are completely or partly generated by the Web server or any background server (e.g. application server). While it is possible for a Web server to serve a great number of static content per second, dynamic content suffers from a high processing overhead [Tho04].

Dynamic Web content is typically generated by a combination of front-end Web server, an application server and a back-end database (see figure (3.2).

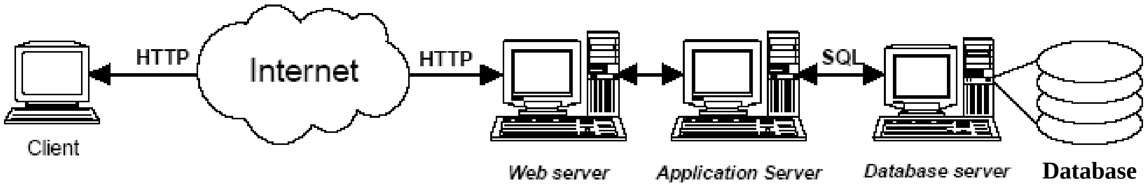


Figure (3.2) Typical Configuration of Dynamic Content Website

The dynamic content of the Website is stored in the database. The application server provides methods that implement the business logic of the application. As part of that the application typically accesses the database [Cri02].

The three servers (Web, application and database server) may all execute on a single machine, or each one of them may execute on a separate machine [Cri02]. Figure (3.3) illustrated the generic architecture of Web applications.

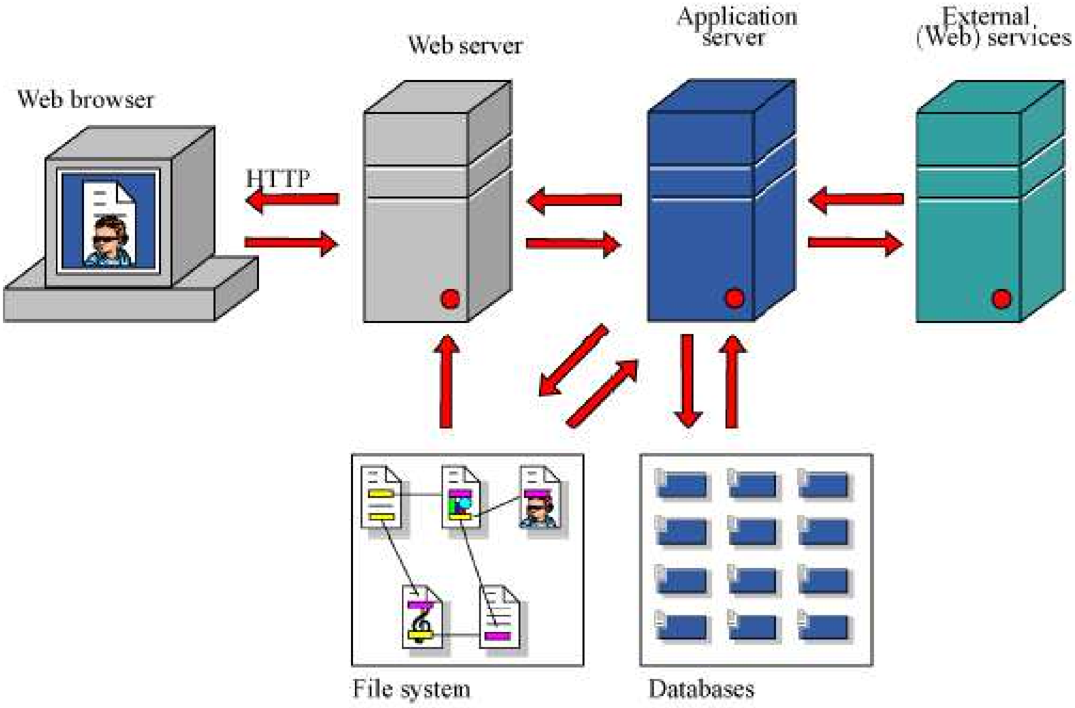


Figure (3. 3) Generic Architecture of Web Applications [Den01]

3.3 Web Applications and HTTP [Mic00]

Web applications use Hyper Text Transfer Protocol (HTTP) to implement communication between browsers and servers. When a user requests a page, the browser creates an HTTP request massage and sends it to the server. The server responds by creating an HTTP response message that is returned to the Web browser. The response message contains an HTML document. The following steps describe this process as illustration in figure (3.4):

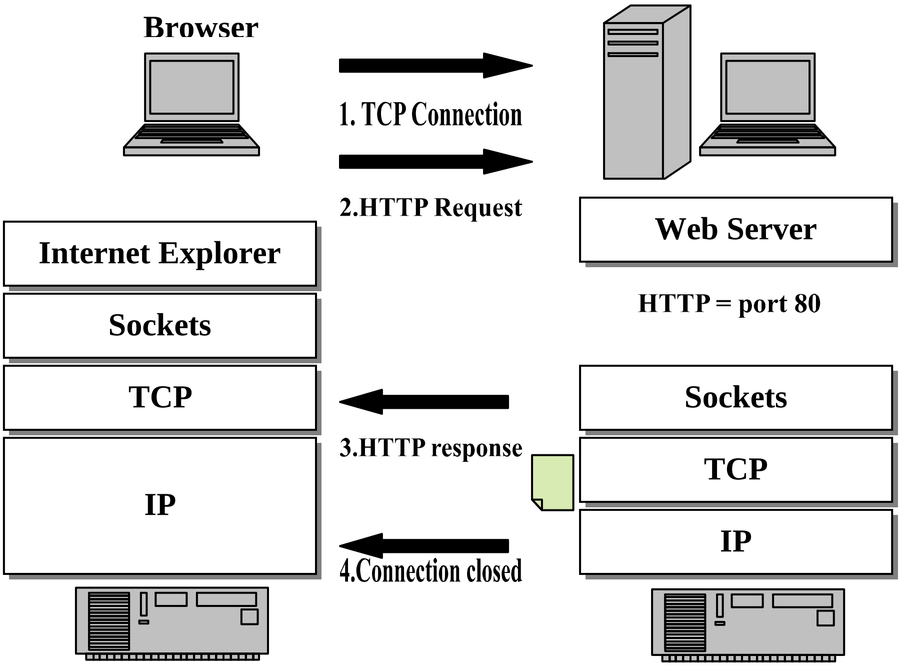


Figure (3.4) Web Application and HTTP

1. The browser creates a Transmission Control Protocol/ Internet Protocol (TCP/IP) connection to the server.
2. The browser packages a request for an HTML document from the server into an HTTP request message, and then sends the message to the server by using a TCP/IP connection. The first line of the message contains the HTTP request method. The Get method is used for a simple page request.
3. The server receives the HTTP request and process it based on the request method contained in the request line.
4. The server then sends back an HTTP response message. Part of the response message is a status line that contain code indicating whether the attempt to satisfy the HTTP request was successful .
5. When the Web browser receives the HTTP response, the TCP/IP connection is closed and the HTTP session terminates.

If the requested HTML document contains embedded objects such as background sounds, the browser makes subsequent requests for each embedded object. For example if a page contains three images, a background sound and an Active X control, six separated HTTP session are required to retrieve the entire page, five for the embedded objects and one for the page itself.

3.4 Categorization of Web Application

Web applications are divided into two groups, the first Web application that has state and use some server-side logic, and the second Website that only have client-side logic [Sve04].

Server script and client script look very much alike because they use the same languages. The main difference is in how script blocks are specified. Server side script is contained either in <% %> in ASP or with Java script in a < SCRIPT > tag, and both gives the following benefits [Mic00]:

- Allows Web engineer to make the Web pages more active.
- Enables Web engineer to interact With Website.
- Allows Web engineer to add forms to the Website to gather data.
- Allows you to use Java and Microsoft Active X technologies.

3.4.1 Server Side Logic

HTML Commands that reside on the Web server are those included in Web pages when they are constructed. They are typically used to hold HTML tag pairs that are common among a group of Web pages. Think of them as the HTML equivalent of subroutines [Jam98].

A Web server, running special software, constructs an HTML page, according to the user's request and possibly other variables, such as time. Suitable scripting languages include for example PHP, ASP, JSP and ColdFusion, and SQL databases, suitable for use with the above. They allow users, subject to access if required, to update content [Wik06]. The flowing ASP code is an example of server side script:

```
<HTML>
 <BODY>
  <% for i=3 to 7 %>
    <FONT SIZE="<% =i %>"> Hello <BR>
  <% next %>
 </BODY>
</HTML>
```

An ASP page consists of scripting languages statement and standard HTML code. To distinguish between the two put the scripting code in brackets <% %> [Nic98].

Active Server Pages or ASP is Scripting environment that can be used to easily create sophisticated Web applications. A key point of Active Server Pages is that ability to accessing a database necessary in complex Website [Dav00]. An Active Server Pages is text file which reside on a Web server. When Web client (browser) calls an Active Server Page the Web server processes the code in the Active Server Page and returns standard HTML to the browser. The advantage Active Server Pages have over standard HTML pages is that they are dynamic [Nic98].

An ASP pages runs on the server before the page is sent to the client. The browser makes a request to the Web server (usually to display a page the user wants to see). And the Web server returns a response (usually an HTML page) to the browser. Using ASP, you can generate a browser-independent page that can vary each time the page is requested (such as returning database). The Web server processes the server script and then sends HTML to the browser for processing.

An ASP script begins to run when a browser requests an *.asp* file from your Web server. The ASP engine then reads through the requested file from top to bottom, execute any script commands, and sends a Web page to the browser as illustrated in figure (3.5) [Mic00].

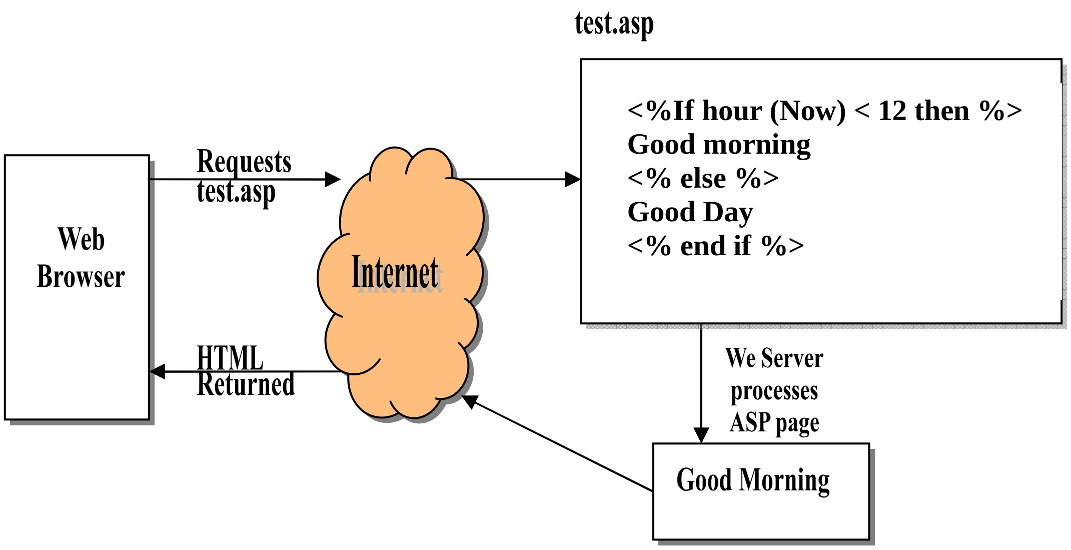


Figure (3.5) How Active Server Pages are processed

3.4.2 Client side Logic

Client side scripting works at the user's browser; Dynamic Hyper Text Markup Language (DHTML) is concerned with client side scripting that reduces the processing time. In other words the time to download the Website will be less because the scripting will be executed on the client side (traditional human factors guidelines indicate 10-15 seconds as the maximum response time before users lose interest). As a result the Website control will be done from the client side rather than the server side [Ala00]. The flowing Java script code is an example of a Client side script:

```
<HTML>
   <HEAD>
      <TITEL> Simple Web page </TITEL>
   </HEAD>
   <BODY>
        <script type=”text/javascript”>
        <! --
         Document.write(“today date is: “ + Date( ));
          //-- >
        </script >
  </BODY>
</HTML>
```

3.5 Web Database Application

A brief mention of database is relevant, considering that they serve as content store for many pages that are dynamically created. Database selection for a Web project depends on the scale of the application. The range of choices includes huge database systems on mainframes to small single-user desktop databases. SQL server, Oracle might be suitable for most Website. Since databases are often the heart of a Website, they should be considered carefully as part of the design phase. Plans should also take into account the performance and maintenance requirements of the database [Tho98].

Web database are accessed indirectly from HTML forms, where HTML specification do not provide for direct database access from within an HTML page. Rather, special "tags", or commands, that are embedded within HTML page trigger another program that actually reads the Web database; the results are then formatted into an HTML page for display back to the person who requested the information.

Web databases are accessed from a Web browser and can have no data components that reside on the user's machine. Web databases come in many different varieties, such as: Microsoft Access, Oracle, and SQL figure (3.6) illustrate the relation ship between Web and Database [Jam98].

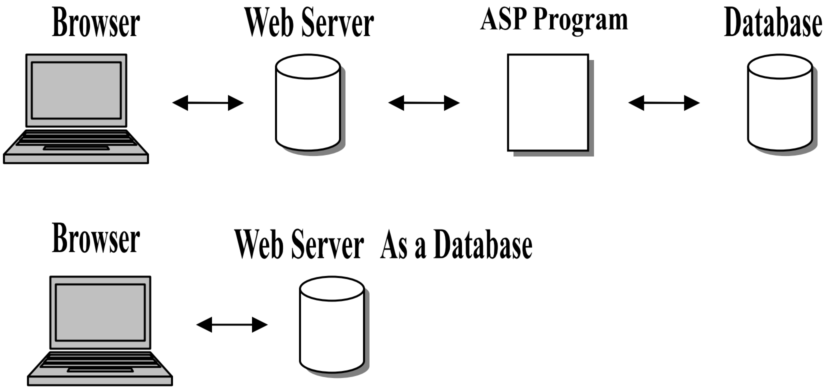


Figure (3.6) Web and Database Relationship [Tho98]

3.5.1 Internet Database Application Benefits [Jam98]

There are a number of advantages to the company that undertakes to develop and deliver a Web database application these are:

- The potential to spend less money by purchasing less expensive client machines as Web applications become more prolific.
- The ability to interface with more potential customers.
- The ability to market products and services globally without having to suffer the burden of establishing overseas offices.
- Increased opportunity to develop systems that meet more of the needs of the company due to the rapid application development nature of HTML and GUI development tools.
- Easy maintenance of Web database applications.

3.5.2 Advantages of Web database Application for the User [Jam98]

There are two groups of users of Web database applications: administrators, employees or members who access Web databases deployed over the internet, and everyone else (gust).

The advantages of Web database application to members are summarized as:

- Access to graphical user interfaces to corporate data.
- Ability to customize their browsers to meet their specific needs.
- Integration of the Web database application with other application running on their machines.

The advantages to every one else accessing Web database applications are:

- Ability to reach an automated sales force at all hours.

- Access to any information from the comforts of home (or their workplace).
- Ability to interface with the Website data without having to purchase expensive equipment or software.

3.5.3 Web Database Processing

Internet database connectivity is usually through some type of middleware that communicates via Open Database Connectivity (ODBC) drivers for a particular database. SQL commands are used for common database functions such as queries, inserts, and updates as well as for accessing stored procedures [Tho98].

A Web database consists of a Web server and a database. Databases are normally used for generating dynamic content and could easily become a bottleneck of the whole Web server system. Figure (3.7) describes a Web database and how a query is normally processed. Such a query is processed in four steps [Tho04]:

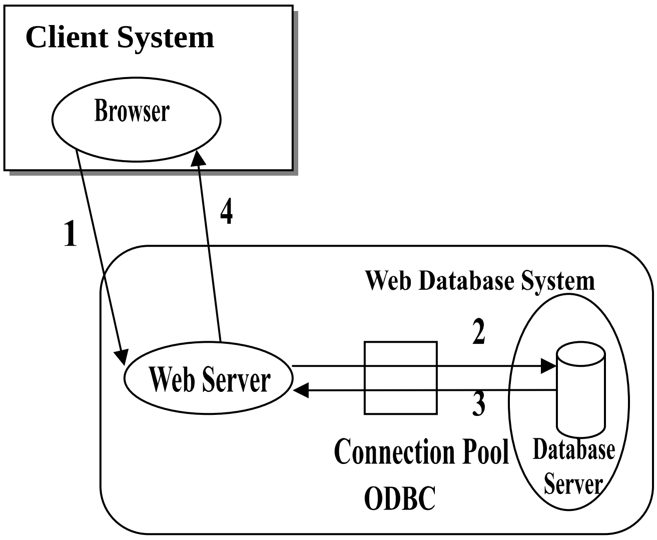


Figure (3.7) Processing a Web Database Query at a Typical Web Database System

1. The query request is send by a client to Web database system via the Internet.
2. Here the request is served by the Web server if it finds a suitable database query and dispatches the request to the database server through an interface.
3. The query is processed by the database server and the result is returned to the Web server.
4. The Web server then uses the result to create a response Web page and returns this page the client.

3.6 Web Application Types [Ist00]

The categorization of the various types of Web applications assists in estimating their level of complexity and the development effort required. In the following most common application categories reported in the Web engineering [Ath01, San05, and Ist00]:

1. Informational – Read-only content with navigation and links for exam.
2. Download – Information available for downloading by the user.
3. Customizable – Content can be customized based on user needs.
4. Interaction – Communication among users via chat rooms, bulletin boards, or instant messaging.
5. User input – Communication via online forms.
6. Transaction oriented – Order processing (products and services).
7. Service oriented – The application provides an online service (e.g. estimating a mortgage payment).
8. Portal – A starting point that channels the user to other Web applications outside the domain of the portal application.
9. Database access – Querying a database and retrieving information.
10. Data warehousing – Querying a collection of large databases and retrieving.

3.7 Web Application Complexity

The scope and complexity of current Web applications vary widely, from small scale, short-lived services to large-scale enterprise applications distributed across the Internet and corporate intranets and extranets. Web-based applications can be grouped into several categories, although a given application may belong to more than one category.

As Web applications have evolved, the demands placed on Web-based systems and the complexity of designing, developing, maintaining, and managing these systems have also increased significantly. They provided vast, dynamic information in multiple media formats (graphics, images, and video).Website design for these and many other applications demand balance among information content, aesthetics, and performance [Ath01]. Figure (3.8) shows a graphical representation for Websites complexity [Tho98].

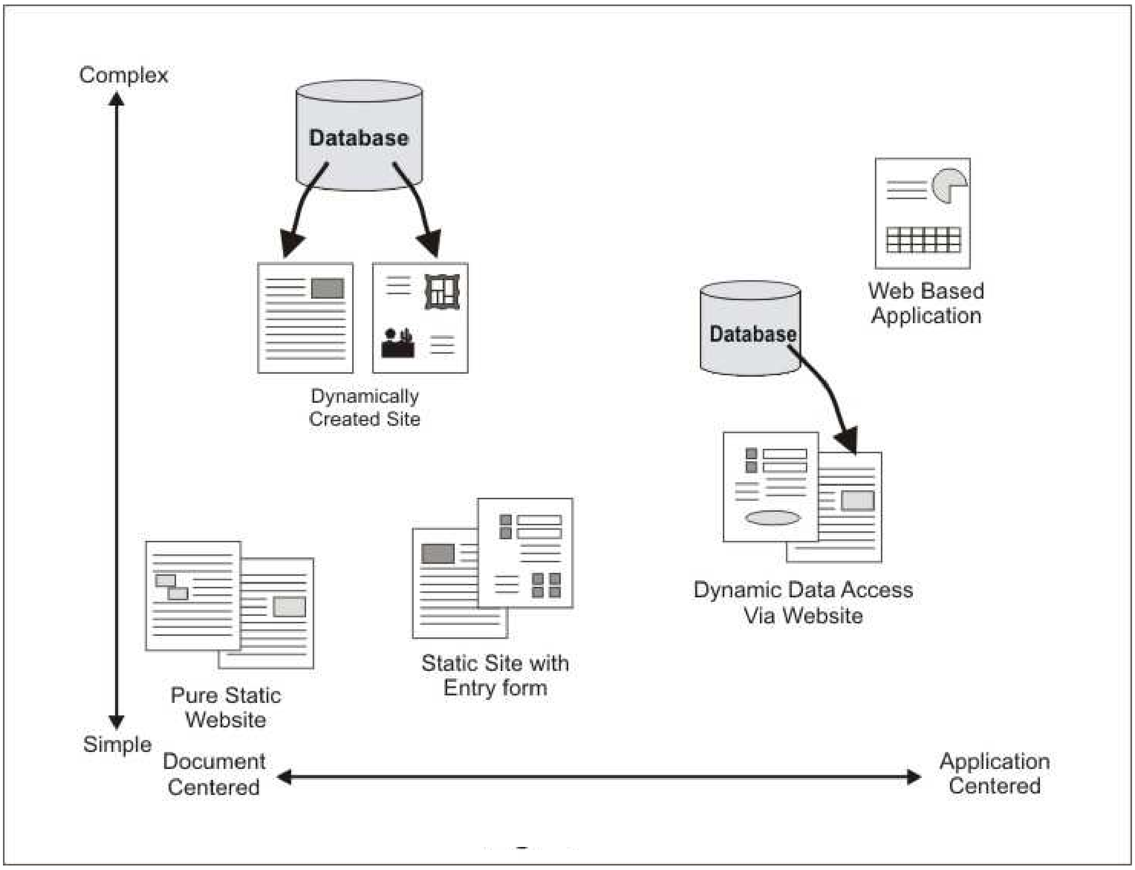


Figure (3.8) Web Application Range Complexity

1. Static Websites

In the most basic form, a Website is little more than a collection of static documents created in HTML and tied together with links. In a purely static Website, much of the emphasis is placed on the information provided and the presentation of that information.

2. Static with Form-Based Interactivity

Limited interactivity is often added to Websites via fill-in forms. Forms are generally used to collect information from the user, including comments or requests for information.

In this case, 90 percent or more of the Websites purpose may be pure document delivery, with a limited emphasis placed on comment forms and other data collection mechanisms.

3. Websites with Dynamic Data Access

The Website provides a front end for accessing a database. Via a Web page, users can search a catalog or perform queries on the contents of a database. Data returned from these user actions is displayed in the form of an HTML document. Like Websites with interactive forms, the division between the dynamically generated aspect of the Website and the static aspect of the Website may be very distinct.

4. Dynamically Generated Websites

One problem of the Websites is that they often have to meet many needs at once. Marketers want to provide customized pages and content based on user preferences in order to foster a one-to-one marketing relationship. Technologists also want to build custom pages dynamically to account for the differences between user browsing environments.

A fully dynamically-generated Website is, at its heart, a software application. The rigors of Web engineering will help ensure that it operates properly.

5. Web-Based Software Application

Web-based software application could be an inventory tracking system in the form of a Web page or a sales force automation tool. Websites that facilitate business processes beyond providing information have more in common with traditional client/ server applications than with static Websites. Because of the emphasis on function, Web engineering principles are mandatory to insure that the resulting product works well and can be maintained.

3.8 Web Application Quality requirements [Ist00]

The complexity of the Web application may be viewed in terms of quality requirements such as usability, functionality, reliability, efficiency and maintainability, see figure (3.9):

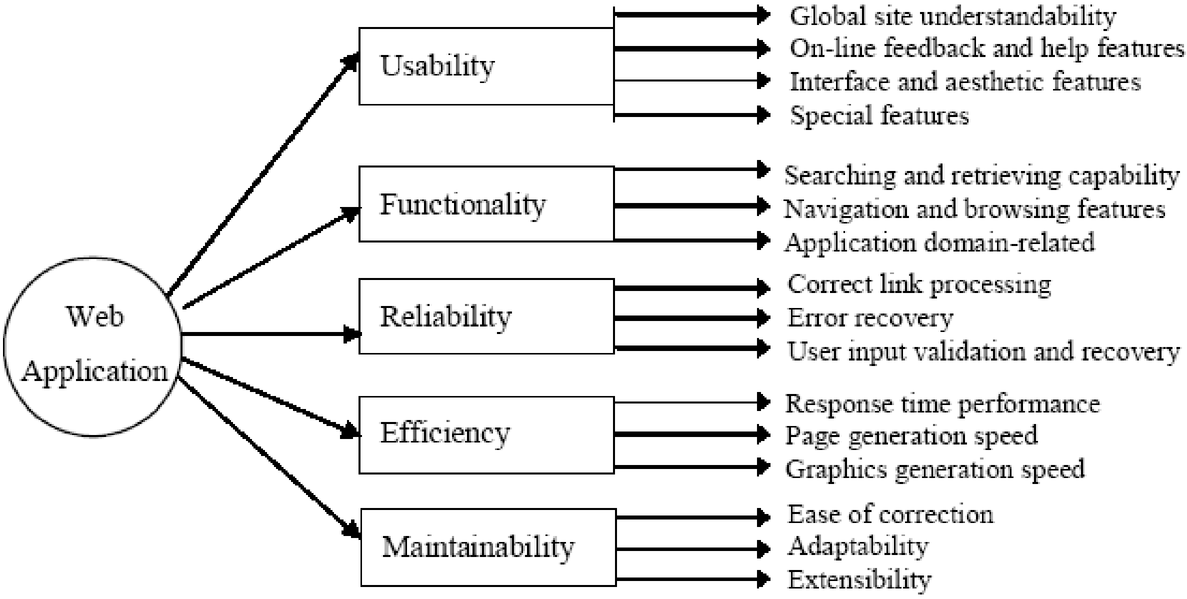

Web Application
Usability
Global site understandability
On-line feedback and help features
Interface and aesthetic features
Special features
Functionality
Searching and retrieving capability
Navigation and browsing features
Application domain-related
Reliability
Correct link processing
Error recovery
User input validation and recovery
Efficiency
Response time performance
Page generation speed
Graphics generation speed
Maintainability
Ease of correction
Adaptability
Extensibility

Figure (3.9) Quality Requirements Tree for Web Applications [Ist00, Rog05]

1. Usability

Issues like understandability, learn ability, friendliness, operability, playfulness and ethics are vital design factors that Web engineers cannot afford to miss. The system must be implemented in such away to allow for easy understanding of its functioning and behaviors even by non-expert Internet users. Aesthetics of user-interface, consistency and ease-of-use are attributes of easy-to-learn systems with rapid learning curve.

2. Functionality

The system must include all the necessary features to accomplish the required tasks. Accuracy, suitability, compliance, interoperability and security are issues that must be investigated in designing a Web application system to ensure that the system will perform as it is expected to. The Web application must have searching and retrieving capabilities, navigation and browsing features and application domain-related features.

3. System Reliability

Producing a reliable system involves understanding issues such as fault tolerance, crash frequency, recoverability and maturity. The system must maintain a specified level of performance in case of software faults with the minimum crashes possible. It also must have the ability to reestablish its level of performance. A system must consistently produce the same results, and meet or even exceed users’ expectations. The Web application must have correct link recognition, user input validation and recovery mechanisms.

4. Efficiency

Users expect the system to run in an efficient manner in order to support their goals. System’s response-time performance, as well as page and graphics generation speed, must be high enough to satisfy user demands. Fast access to information must be examined also throughout the system’s life to ensure that user requirements are continuously met on one hand, and that the system remains competitive and useful on another.

5. Maintainability

The primary target here is to collect data that will assist designers to conceive the overall system in its best architectural and modular form, from a future maintenance point of view. With the rapid technological changes especially in the area of Web engineering, as well as the rigorous user requirements for continuous Website updates, easy system modifications and time enhancements, both in content and in the way this content is presented, are also critical success factors for the development and improvement of a Web application.

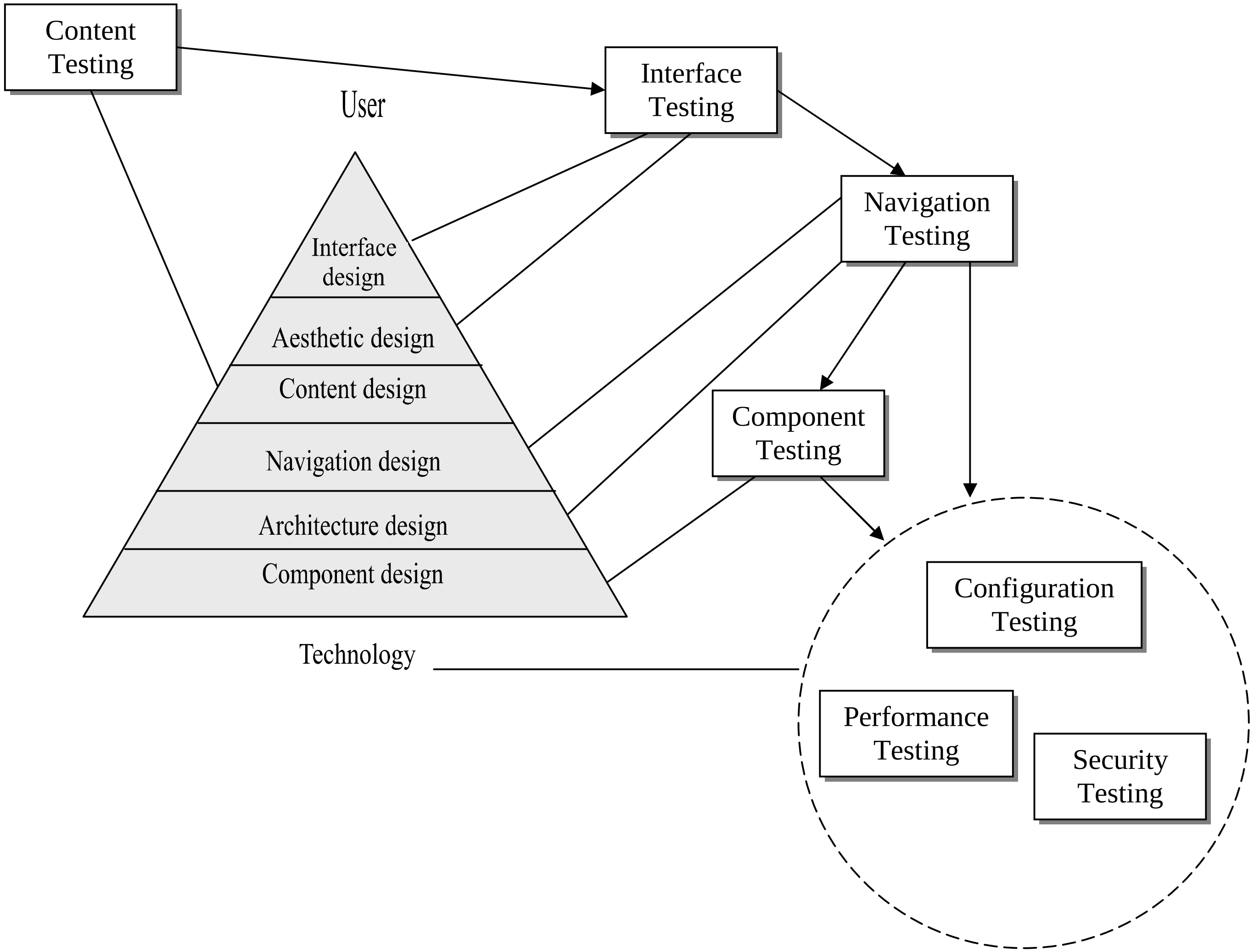


Figure (2.10) Web Engineering Testing Process